UNIVERSIDAD DE DEUSTO

METODOLOGÍA BASADA EN LA EXPERIENCIA PARA LA MEJORA DE LA EFICIENCIA DE LA EVALUACIÓN DE LA USABILIDAD DE APLICACIONES MÓVILES

Tesis doctoral presentada por D. Iván Pretel García

Dirigida por la Dra. Ana Belén Lago Vilariño

BILBAO

2015

UNIVERSIDAD DE DEUSTO

METODOLOGÍA BASADA EN LA EXPERIENCIA PARA LA MEJORA DE LA EFICIENCIA DE LA EVALUACIÓN DE LA USABILIDAD DE APLICACIONES MÓVILES

Tesis doctoral presentada por D. IVÁN PRETEL GARCÍA

dentro del Programa de Doctorado en

INGENIERÍA INFORMÁTICA Y TELECOMUNICACIONES

Dirigida por la Dra. ANA BELÉN LAGO VILARIÑO

La Directora

El Doctorando

BILBAO

2015

Metodología basada en la experiencia para la mejora de la eficiencia de la evaluación de la usabilidad de aplicaciones móviles

Autor: D. Iván Pretel García
Directora: Dra. Ana Belén Lago Vilariño

Editado con *Microsoft Word* utilizando los tipos de letra *Book Antiqua* para el contenido y *Roboto* para títulos, cuyos derechos pertenecen a Cristiano Robertson - *https://www.google.com/fonts/specimen/Roboto*.

Impreso en Bilbao
Primera edición, octubre 2015

A mis padres

y hermano

Resumen

Dentro de la disciplina de la Interacción Persona-Ordenador o HCI (Human-Computer Interaction), las pruebas de usabilidad de aplicaciones de software desarrolladas para dispositivos móviles es una emergente área de investigación. Esta área se enfrenta a una variedad de problemas debido a las características de los dispositivos móviles, que en los últimos años han revolucionado el concepto de la telefonía, tanto a nivel tecnológico como social.

Estos dispositivos muestran varias limitaciones desde la perspectiva de la usabilidad ya que disponen de interfaces como su pequeña pantalla, difíciles de utilizar. Además, el contexto dinámico que caracteriza a este tipo de dispositivos dificulta la interacción al ser extremadamente variable y mostrar un impacto muy significativo en la usabilidad de los mismos. Por consiguiente, existe una fuerte necesidad de estudiar las características del contexto en el estudio de la usabilidad de aplicaciones móviles. Dentro de las pruebas de usabilidad en entornos reales, se ha detectado un coste en cuanto a tiempo y esfuerzo elevado, además de una deficiencia en la calidad de los datos capturados y una falta de privacidad por parte de los usuarios que realizan las pruebas que dificulta su realización en ciertos entornos reales como es en la propia casa del usuario.

Esta tesis explora los medios para abordar las limitaciones de la evaluación que se han planteado mediante el desarrollo de una novedosa metodología que hace uso de una base de conocimiento de experiencias previas para la evaluación de usabilidad de aplicaciones móviles con un uso reducido de recursos.

En conclusión, esta tesis estudia si es posible reducir los recursos necesarios en la evaluación de la usabilidad de aplicaciones móviles sin comprometer la calidad de los resultados mediante una nueva metodología de evaluación centrada en una base de

conocimiento. Para realizar este trabajo se define una metodología de evaluación de aplicaciones móviles, se implementa una plataforma de soporte a la nueva metodología logrando una aplicabilidad práctica y se verifican los resultados del uso de esta nueva metodología.

Abstract

Inside the HCI (Human-Computer Interaction) field, usability testing of software applications developed for mobile devices is an emerging research area. This area faces a wide variety of problems due to the main characteristics of mobile devices. In recent years, this kind of devices has transformed the concept of telephony not only in technological terms, but also in social ones.

From the usability perspective, these devices show several limitations. This is because their interfaces are difficult to use, such as their small screens. In addition, the dynamic context that characterizes this kind of devices makes interaction very difficult. This context is extremely variable and shows a significant impact on the usability of the mobile applications. Therefore, there is a strong need to study all the context characteristics inside the usability of mobile applications field. Focusing on the usability testing performed in real environments; it has been detected that the used procedures demand a high quantity of time and effort. The captured data have low quality and require a reduction of the privacy of the user who performs the experiments. These problems make the execution of usability experiments in certain environments very difficult (e.g. at user's home).

This dissertation explores several ways to address the limitations that have been explained by developing a new methodology that uses a knowledge base, which is made up by previous experiences, to evaluate the usability of mobile applications using a reduced amount of resources.

In conclusion, this dissertation studies whether it is possible to reduce the needed resources to assess the usability of mobile applications without decreasing the quality of results through a new methodology focused on a knowledge base. In order to do so, a new methodology to evaluate the usability of mobile

applications is developed, a software platform to support the new methodology and achieve practical applicability is implemented, and results of the use of the new methodology are verified.

Agradecimientos

La ansiada finalización de este trabajo ha resultado muy dura, no sólo a mí sino también a todas las personas que en muchas ocasiones han soportado y mermado mis estados emocionales sinusoidales. Es difícil comenzar estas palabras cuando te vienen a la mente tantos momentos y tantas personas a la cabeza.

Voy a comenzar con mis padres Mª Yolanda y Félix, grandes pilares en mi vida, por dármela, por facilitármela y mejorármela tantos años a golpe de una buena educación y estudios. Yeray, mi hermano y eterno cómplice, gracias por estar siempre conectado y dispuesto a desconectar. A mis abuelos, Elvira y Delfín, gracias por codirigir con mis padres mi educación en mis primeros años, enderezando bien esos primeros raíles, no como las carreteras que hacía yo con pinzas en la cocina. Gracias también a mis tíos, tías, primos y primas.

Gracias a David e Israel, la impulsividad y serenidad encarnadas en dos personas que no tienen filtro, algo muy agradecido cuando buscas consejo y algo parecido al ángel bueno y al ángel malo de la conciencia de los dibujos animados.

Gracias a la cuadrilla Basauritarra. Donde me viene a la mente la gran boda en Wisconsin de Iñaki y Molly, gracias por dar ese toque de originalidad al año que al fin termino la tesis. Txema, sin muchas palabras pero un fuerte apoyo desde que comencé la carrera. Edu, con el que he compartido problemas, proyectos y sobretodo el rumbo académico desde que comenzamos en el cole hasta este mismo año. Roberto e Imanol, los gemelos no gemelos de la clase en esos primeros años de universidad. Todos habéis sido un gran apoyo, gracias.

A nivel académico, gracias a todos los compañeros de universidad que formamos grupo, gracias chicos, ya sabéis quienes sois. Y a

todos los profesores que me formaron en la carrera. Especial mención a Begoña Rodríguez-Rey y sus sólidos valores, que fueron un gran ejemplo y apoyo, sobretodo en el desarrollo del proyecto fin de carrera.

Gracias a Diego por creer en mí y haberme ofrecido la oportunidad de formar parte de este equipo. Gracias a Ana por ofrecerme siempre ayuda y comprensión desde mis comienzos en la investigación, sobre todo por converger mi divergencia, muchas veces enredada en sí misma. Gracias a los compis generacionales de doctorado Eduardo Castillejo y Aitor Gómez-Goiri, destacando la ofensiva y culta prosa de este último. Gracias a las correcciones de Aitor Almeida, las lecciones y conversaciones de Pablo Orduña y las compañías en la ingesta de cafeína del grupo de cafeteros. En definitiva, muchísimas gracias a todos los miembros que han formado y forman parte del grupo Morelab (morelab.deusto.es).

Koldo y Pablo, compañeros de escuadra de batalla con las propuestas y proyectos, muchísimas gracias por allanarme el camino a la meta liberándome de carga de trabajo. Sin vosotros este trabajo todavía no estaría completado. Agradeceros, no solo facilitarme el día a día laboral, sino también el personal, con esas escapadas a los Pirineos en la temporada de invierno y con esa compañía en esos intentos de cuidar la salud y "comer de tuper".

Janire, una gran compañera de trabajo y mejor compañera de aventuras, con la que puedes compartir no solo esas escapadas de esquí a los Pirineos o esas travesuras de Boly, sino también una infinidad de problemas. Gracias por todo lo compartido, aconsejado y vivido, sin ello esta tesis no hubiera tenido forma y seguramente yo tampoco.

Muchas gracias a Laurentzi, por su tiempo y ayuda al desarrollar una de las aplicaciones de la experimentación. Y a todos aquellos que participaron desinteresadamente en todos los experimentos de esta tesis.

Igualmente debo nombrar a todos los amigos de Hormiguera que siempre han intentado que salga de la cueva en verano e invierno cuando he estado agobiado, muchas gracias.

Todas las porciones que aparecen en este pastel son igual de dulces, pero sin duda el toque final que da la guinda lo das tú, Irene. Muchísimas gracias por una infinidad de cosas pero sobre todo por simplificar lo complicado, aguantar lo inaguantable y tranquilizarme en esos momentos de alegría, cabreos, lloros y un largo etcétera en los que cualquier otra persona hubiera salido corriendo. Como diríais por Falces, tú junto con todos los Autor, especialmente Mari Jose, habéis conseguido que logre terminar este encierro.

He querido resumir en este poco espacio con un orden algo aleatorio a todos aquellos que han aportado, también con aleatoriedad, granos o sacos de arena a este trabajo. A todos los que he mencionado y a todos los que no, gracias.

Eskerrik asko,
Iván Pretel
Bilbao, octubre 2015

Índice general

CAPÍTULO 1 **1**
1. Introducción 1
   1.1. Motivación 3
   1.2. Hipótesis 6
   1.3. Objetivos 7
   1.4. Metodología 11
   1.5. Estructura de la tesis 13

CAPÍTULO 2 **15**
2. Estado del arte 15
   2.1. Introducción a la usabilidad 16
      2.1.1. Definición de la usabilidad 16
      2.1.2. Dimensiones de usabilidad 18
      2.1.3. Ingeniería de la usabilidad 19
      2.1.4. Métodos de evaluación de usabilidad 19
   2.2. Usabilidad en los dispositivos móviles 36
      2.2.1. Limitaciones de las aplicaciones móviles 37
      2.2.2. Evaluación de la usabilidad de aplicaciones móviles 40
      2.2.3. Limitaciones de la evaluación de la usabilidad 44
      2.2.4. Importancia del contexto en las aplicaciones móviles 46
   2.3. Metodologías de evaluación 48
      2.3.1. Criterios de análisis 48
      2.3.2. Métodos de evaluación de usabilidad 51
   2.4. Conclusiones y solución propuesta 60
      2.4.1. Análisis y conclusiones 60
      2.4.2. Solución propuesta 62

CAPÍTULO 3 **69**
3. Metodología de evaluación de usabilidad en entornos móviles 69
   3.1. Descripción general 71
      3.1.1. Fase de definición 73
      3.1.2. Fase de ejecución 76
      3.1.3. Fase de análisis 78
   3.2. Base de conocimiento 80
      3.2.1. Modelo de contexto 81

3.2.2. Modelo de interacción 90
3.2.3. Modelo de análisis 98
3.2.4. Modelo de casos favorables 109
3.3. Relación de las fases con la base de conocimiento 131

CAPÍTULO 4 **135**
4. Plataforma de soporte 135
4.1. Plataforma Android como elección 137
4.2. Descripción general 140
4.3. Librería de integración 143
4.3.1. Funciones de la librería 145
4.3.2. Integración de la aplicación evaluada 147
4.4. Aplicación web de gestión de la base de conocimiento 150
4.5. Herramientas de desarrollador 152
4.5.1. Funcionalidad 152
4.5.2. Arquitectura de las herramientas de desarrollador 161
4.6. Herramienta de usuario de pruebas 163
4.6.1. Funcionalidad 163
4.6.2. Arquitectura de la aplicación 168

CAPÍTULO 5 **171**
5. Experimentación y validación 171
5.1. Validación de la estrategia de captura de los modelos 174
5.1.1. Descripción del experimento 175
5.1.2. Resultados 176
5.1.3. Consideraciones de la estrategia de captura 185
5.2. Validación del uso de la base de conocimiento 186
5.2.1. Descripción del experimento 187
5.2.2. Resultados 190
5.2.3. Consideraciones del uso de la base de conocimiento 199
5.3. Validación de la metodología 200
5.3.1. Descripción del experimento 202
5.3.2. Resultados 215
5.3.3. Consideraciones de la metodología 221

CAPÍTULO 6 **225**
6. Conclusiones 225
6.1. Visión general del trabajo 226
6.2. Contribuciones 227
6.3. Resultados obtenidos 231
6.4. Trabajo futuro 234
6.5. Consideraciones finales 237

Índice de figuras

Figura 1.1	Principales eventos en la evolución de los teléfonos inteligentes	3
Figura 1.2	Fases de la metodología de investigación seguida en esta tesis	11
Figura 2.1	Dimensiones de usabilidad de los principales autores	18
Figura 2.2	Taxonomías más relevantes para clasificar los métodos de evaluación de la usabilidad	28
Figura 2.3	Componentes de la solución propuesta en esta tesis	64
Figura 2.4	Resumen de la solución propuesta en esta tesis y sus requisitos	67
Figura 3.1	Fases de la metodología de evaluación de usabilidad en entornos móviles	71
Figura 3.2	Fases en las que se involucran los agentes de la metodología	73
Figura 3.3	Modelos que forman la base de conocimiento	80
Figura 3.4	Pilares del modelo de contexto	81
Figura 3.5	Conjuntos de atributos de la aplicación	82
Figura 3.6	Conjuntos de atributos del usuario	83
Figura 3.7	Conjuntos de atributos del dispositivo	84
Figura 3.8	Conjuntos de atributos del entorno	86
Figura 3.9	Conjuntos de atributos de la tarea	88
Figura 3.10	Diagrama de estados de la tarea y eventos de tarea	91
Figura 3.11	Ejemplo del flujo básico de los cambios de estado de una tarea	93
Figura 3.12	Ejemplo de interfaces para generación de camino de interacción correcta	96
Figura 3.13	Ejemplo de grafo de camino de interacción correcta	97
Figura 3.14	Criterio de decisión para definir el tipo de evento de interacción	98
Figura 3.15	Intersección entre las 18 dimensiones de Baharuddin et al. y los estándares formales	99
Figura 3.16	Función del número de casos posibles en base al número de entornos	112
Figura 3.17	Función de probabilidad de que al menos un error aparezca una vez	114
Figura 3.18	Función de probabilidad para los casos posibles de ejemplo	116
Figura 3.19	Elección de la metodología de análisis de relación	120
Figura 3.20	Pasos de la metodología de análisis tipo I	125
Figura 3.21	Pasos de la metodología de análisis tipo II	128
Figura 3.22	Pasos de la metodología de análisis tipo III	131
Figura 4.1	Número total de desarrolladores por plataforma	139
Figura 4.2	Elementos de la plataforma de soporte	140
Figura 4.3	Modelo de datos	141
Figura 4.4	Uso de la librería por una aplicación evaluada	145
Figura 4.5	Ciclo de vida de un Activity y funciones de librería ejecutadas en las transiciones de estado	147
Figura 4.6	Arquitectura de la aplicación web de gestión de la base de conocimiento	150
Figura 4.7	Ventana de inicio de sesión en la aplicación de escritorio	153
Figura 4.8	Panel principal de evaluaciones del desarrollador	154
Figura 4.9	Formulario de datos generales de evaluación y dimensiones de contexto intrascendentes	155
Figura 4.10	Formulario de definición de usuarios y entornos	156
Figura 4.11	Formulario de grabación y generación de caminos de interacción correcta	158

Figura 4.12	Grabación de eventos desde la aplicación móvil de grabación de caminos de interacción	159
Figura 4.13	Interfaz de análisis de resultados	160
Figura 4.14	Arquitectura de la aplicación de escritorio y grabación de tareas	161
Figura 4.15	Asistente de registro de un usuario de pruebas	164
Figura 4.16	Descarga de evaluaciones a realizar por el usuario de pruebas	165
Figura 4.17	Selección de tarea y entorno	166
Figura 4.18	Finalización de tarea y cuestionario	167
Figura 4.19	Subida de resultados a la aplicación web	167
Figura 4.20	Arquitectura de la aplicación móvil de captura de pruebas	168
Figura 5.1	Esquema de la validación	173
Figura 5.2	Evolución del uso medio estimado de procesador por dispositivo	178
Figura 5.3	Comparación del uso de memoria física con y sin captura de modelos	179
Figura 5.4	Comparación del tiempo de carga de interfaz con y sin herramienta	184
Figura 5.5	Fases del experimento de validación del uso de la base de conocimiento	187
Figura 5.6	Estimaciones de errores de interacción encontrados mediante la base de conocimiento	190
Figura 5.7	Comparación de la proporción del número de errores de interacción detectados en los grupos de casos favorables elegidos con el resto de casos posibles	195
Figura 5.8	Proporción del número de errores de interacción detectados dentro de los límites acotados	198
Figura 5.9	Analogía entre los pasos de la metodología definida y MUSiC	203
Figura 5.10	Prototipo de grabación creado para la captura de la interacción en vídeo	213
Figura 5.11	Respuestas a las afirmaciones del cuestionario de aceptación tecnológica	220

Índice de tablas

Tabla 1.1	Objetivos operacionales y sus relaciones con los objetivos específicos	9
Tabla 2.1	Propiedades de los principales métodos de la evaluación de usabilidad (1/2)	27
Tabla 2.2	Propiedades de los principales métodos de la evaluación de usabilidad (2/2)	27
Tabla 2.3	Limitaciones de las aplicaciones móviles desde el punto de vista de la usabilidad	38
Tabla 2.4	Limitaciones de la evaluación de la usabilidad de aplicaciones móviles identificadas	45
Tabla 2.5	Características de las soluciones comerciales	55
Tabla 2.6	Características de las soluciones del ámbito científico	59
Tabla 2.7	Retos abordados por las soluciones estudiadas en comparación con la solución propuesta	62
Tabla 3.1	Resumen de los requisitos que debe cumplir la metodología	70
Tabla 3.2	Ejemplo de definición de nuevo entorno	75
Tabla 3.3	Ejemplo de atributos del pilar usuario del modelo de contexto	84
Tabla 3.4	Ejemplo de atributos del pilar dispositivo del modelo de contexto	86
Tabla 3.5	Ejemplo de atributos del pilar entono del modelo de contexto	88
Tabla 3.6	Ejemplo de atributos del pilar tarea del modelo de contexto	90
Tabla 3.7	Ejemplo de atributos de eventos de tarea	93
Tabla 3.8	Ejemplo de atributos de eventos de interacción	94
Tabla 3.9	Métricas del componente sumativo	100
Tabla 3.10	Preguntas del cuestionario USE adaptadas para calcular la satisfacción	103
Tabla 3.11	Ejemplo de métricas para 6 tareas agrupables	104
Tabla 3.12	Ejemplo de cálculo de variables para la agrupación de 6 tareas agrupables	104
Tabla 3.13	Ejemplo de componente formativo	106
Tabla 3.14	Criterio de nivel de severidad del problema de usabilidad	108
Tabla 3.15	Ejemplo de combinación de entornos	111
Tabla 3.16	Ejemplo de combinación de tipos de usuario	111
Tabla 3.17	Ejemplo de casos posibles con tres entornos y dos tipos de usuario	111
Tabla 3.18	Ejemplo de cálculo de la tabla de la probabilidad de detección	115
Tabla 3.19	Ejemplo de clasificación de casos posibles	116
Tabla 3.20	Ejemplo de valores de las variables ruido e iluminancia en los momentos de causar errores	117
Tabla 3.21	Ejemplo de estadísticos descriptivos de dos variables de contexto cuantitativas	118
Tabla 3.22	Ejemplo de tabla de frecuencia para una variable de contexto cualitativa	119
Tabla 3.23	Ejemplo de muestra con variable de contexto cuantitativa	122
Tabla 3.24	Criterio de aceptación en base al tamaño del efecto de variable independiente cuantitativa	124
Tabla 3.25	Ejemplo de muestra con variable de contexto dicotómica	126
Tabla 3.26	Criterio de aceptación en base al tamaño del efecto de variable independiente dicotómica	127
Tabla 3.27	Ejemplo de muestra con variable de contexto policotómica	129

Tabla 3.28	Criterio de aceptación en base al tamaño del efecto de variable independiente policotómica	130
Tabla 3.29	Pasos de consulta de la base de conocimiento	132
Tabla 3.30	Pasos de agregación de la base de conocimiento	133
Tabla 4.1	Resumen de los requisitos que debe cumplir la plataforma de soporte	136
Tabla 4.2	Relación de pasos con la funcionalidad de la aplicación de escritorio de gestión de evaluación	153
Tabla 4.3	Relación de pasos con la funcionalidad de la aplicación móvil de captura de pruebas	163
Tabla 4.4	Variables de contexto capturadas en el registro del usuario de pruebas	164
Tabla 4.5	Variables de contexto capturadas durante la ejecución de una tarea	166
Tabla 5.1	Ejecuciones de tareas desarrolladas para el estudio del sesgo en el dispositivo, aplicación e interacción	175
Tabla 5.2	Modelos utilizados para el estudio del sesgo en el dispositivo, aplicación e interacción	176
Tabla 5.3	Diferencias del uso medio estimado de procesador con y sin herramienta	178
Tabla 5.4	Diferencias del uso medio estimado de memoria física con y sin herramienta	179
Tabla 5.5	Kilobytes estimados requeridos para el almacenamiento de los modelos capturados por tarea realizada	180
Tabla 5.6	Intervalos de confianza del 95% de los milisegundos estimados para llamar a las funciones	181
Tabla 5.7	Diferencias de los segundos estimados de duración de las ejecuciones con y sin herramienta	182
Tabla 5.8	Diferencias de los milisegundos estimados de carga de interfaz con y sin herramienta	184
Tabla 5.9	Descripción del grupo de usuarios de pruebas que generó la base de conocimiento del experimento	187
Tabla 5.10	Descripción de los dispositivos utilizados en la generación de la base de conocimiento del experimento	188
Tabla 5.11	Descripción de las tareas realizadas con la aplicación Maicbay en el experimento	188
Tabla 5.12	Descripción de los entornos del experimento	189
Tabla 5.13	Descripción del grupo de usuarios de pruebas de la evaluación de Maicvalia	189
Tabla 5.14	Descripción de los dispositivos utilizados en la evaluación de Maicvalia	190
Tabla 5.15	Estimaciones obtenidas de la base de conocimiento para casos de un entorno	191
Tabla 5.16	Estimaciones obtenidas de la base de conocimiento para casos de dos entornos	192
Tabla 5.17	Estimaciones obtenidas de la base de conocimiento para casos de tres entornos	192
Tabla 5.18	Número de errores de interacción encontrados en la evaluación de Maicvalia agrupados por severidad	193
Tabla 5.19	Ejemplo de combinaciones para la validación de la base de conocimiento	193
Tabla 5.20	Mejores casos posibles de la estimación de la base de conocimiento asignados como casos favorables	196
Tabla 5.21	Estimaciones de la variable iluminancia obtenidas de la base de conocimiento generada	197
Tabla 5.22	Tabla de frecuencias obtenidas de la base de conocimiento generada para las variables cualitativas	198

Tabla 5.23	Proporción del número de errores de interacción detectados dentro de los estados elegidos	199
Tabla 5.24	Resumen de los tiempos que componen las fases de las metodologías a comparar	207
Tabla 5.25	Resumen de variables de tiempo a cuantificar en el experimento de validación de la metodología	211
Tabla 5.26	Descripción del grupo de desarrolladores que realizó el experimento de validación de la metodología	211
Tabla 5.27	Descripción del conjunto de tareas del experimento de validación de la metodología	212
Tabla 5.28	Cuestionario Likert del estudio de la aceptación tecnológica	214
Tabla 5.29	Resumen de variables medidas de valor constante ajenas al desarrollador	215
Tabla 5.30	Resumen de variables referentes al desarrollador	215
Tabla 5.31	Duración estimada de ambas evaluaciones y su diferencia	216
Tabla 5.32	Valores de las variables con las que estudiamos el comportamiento de la diferencia de tiempos	217
Tabla 5.33	Matriz de correlaciones de las variables en estudio	218

Acrónimos

ACM	Association for Computing
ANOVA	ANalysis Of VAriance
API	Application Programming Interface
APK	Application Package File
A2DP	Advanced Audio Distribution Profile
CPU	Central Processing Unit
CRUD	Create, Read, Update and Delete
dB	Decibelios
DRUM	Diagnostic Recorder for Usability Measurement
EUCSI	End-User Computing Satisfaction Instrument
GNU	GNU's Not Unix
GPRS	General Packet Radio Service
HCI	Human-Computer Interaction
IP	Internet Protocol
kB	Kilobytes
lux	Luxes
MB	Megabytes
MUSIC	Measurement of USability In Context
QUIS	Questionnaire for User Interface Satisfaction
REST	REpresentational State Transfer
SATURN	Software ArchitecTure analysis of Usability Requirements realization
SDK	Software Development Kit
SIERRA	Seguro, Invisible, Eficiente, Robusto, Remotamente controlable y Autónomo
SIGCHI	Special Interest Group on Computer–Human Interaction
SQuaRE	Software product Quality Requirements and Evaluation
SUMI	Software Usability Measurement Inventory
SUS	System Usability Scale
TCP	Transmission Control Protocol
UMTS	Universal Mobile Telecommunications System
URL	Uniform Resource Locator
USE	Usefulness, Satisfaction, and Ease of Use

CAPÍTULO 1

INTRODUCCIÓN

«Un camino de mil millas comienza con un primer paso»,
Francesc Miralles Contijoch (1968-)

ÍNDICE DE CAPÍTULO 1

1.1. Motivación	3
1.2. Hipótesis	6
1.3. Objetivos	7
1.4. Metodología	11
1.5. Estructura de la tesis	13

Mediante este capítulo proporcionaremos una primera toma de contacto dentro de los límites de la presente tesis, presentando el ámbito de trabajo de la misma y su motivación, la hipótesis formulada junto con los objetivos y la metodología seguida para la validación de la misma y la consecución de los objetivos.

Situamos el trabajo desarrollado en esta tesis doctoral dentro del campo de la *Interacción Persona-Ordenador* o *HCI (Human-Computer Interaction)*, introducido originalmente por Card et al. [Card+83]:

> *La Interacción Persona-Ordenador es un proceso con el carácter de un modelo de diálogo conversacional en el que el usuario (alguien que quiere realizar alguna tarea con la ayuda de un ordenador) proporciona información codificada (p.ej., de entrada) al ordenador, que le responde con información y datos.*

Posteriormente, el grupo de interés en *HCI* de la asociación *ACM* (*Association for Computing Machinery*) llamado *SIGCHI* (*Special*

Interest Group on Computer–Human Interaction), proporcionó una definición más formal [Hewett+92]:

> *Una disciplina que se centra en el diseño, evaluación e implementación de sistemas de computación interactiva para el uso humano y el estudio de los principales fenómenos que los rodea.*

Este campo se considera la intersección entre las ciencias de la computación, ciencias de la conducta, diseño, y varios otros campos de estudio. Por ello se muestra como una disciplina muy extensa donde se reconocen varias vertientes.

> *Dentro de la definición formal explicada, el trabajo desarrollado en esta tesis se centra en nuevas técnicas de evaluación, concretamente en el diseño de métodos específicos para las evaluaciones y técnicas de pruebas de usabilidad y explícitamente, la usabilidad de aplicaciones móviles.*

Como veremos posteriormente en las definiciones de la comunidad especializada en la usabilidad, explicamos informalmente lo siguiente:

> *El campo de la usabilidad estudia la capacidad de un sistema de computación interactiva de ser usado fácilmente y efectivamente por un rango específico de usuarios, dando una ayuda y soporte específicos, para completar un rango específico de tareas en un rango de escenarios específico.*

Mediante la siguiente sección presentaremos la motivación de la elección de este ámbito de trabajo y las razones que justifican el motivo para centrarlo concretamente en la evaluación de aplicaciones móviles.

1.1. MOTIVACIÓN

En los últimos años, los dispositivos móviles han revolucionado el concepto de la telefonía, tanto a nivel tecnológico como social. Antes de centrarnos en la problemática de la evaluación de este tipo de aplicaciones, debemos estudiar cómo han evolucionado este tipo de dispositivos.

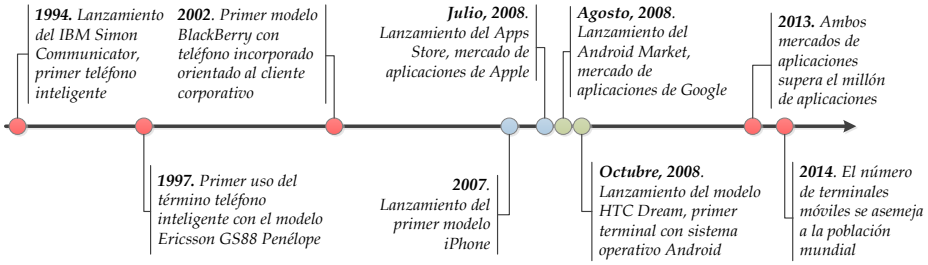

Figura 1.1 Principales eventos en la evolución de los teléfonos inteligentes

El 16 de agosto de 1994, la empresa IBM lanzó al mercado estadounidense un terminal móvil conocido como IBM Simon Communicator. Dicho terminal, con su medio kilo de peso ofrecía características actuales como interacción mediante pantalla táctil y funciones de correo electrónico (sólo si estaba conectado a un ordenador con conexión a Internet). Este modelo fue posteriormente catalogado como el primer Smartphone de la historia. En 1997, se utilizó por primera vez el término Smartphone o teléfono inteligente, cuando la compañía Ericsson describió su modelo GS88 Penélope como un teléfono inteligente. En 2002, se lanza el primer dispositivo BlackBerry con teléfono incorporado orientado al cliente corporativo. Éste podía conectarse a internet por GPRS y contaba con funcionalidad para proporcionar herramientas de ayuda a la comunicación empresarial como un organizador personal que integraba funciones de escritorio.

Hasta entonces los terminales presentaban una evolución constante, no es hasta 2007 cuando comienza la verdadera

revolución y expansión de los teléfonos inteligentes a manos de Steve Jobs con la presentación del primer modelo del terminal iPhone[1]. Éste se presentó como una combinación entre un reproductor de música portátil, teléfono y dispositivo de navegación con acceso a servicios de Internet. Dicho terminal marcó un antes y un después en la evolución de los teléfonos inteligentes, abriendo un año después nuevos modelos de negocio como el famoso mercado de aplicaciones Apps Store[2] lanzado el 10 de julio de 2008. Como respuesta al avance en el sector, en agosto de 2008 se lanzó el mercado de aplicaciones Android Market, conocido actualmente como Google Play[3] que serviría dos meses después, para complementar al terminal HTC Dream o Google Phone, el primero con sistema operativo Android, que permitía acceso a aplicaciones de Google como Gmail[4] y Google Maps[5] y todas las ofertadas en su mercado de aplicaciones.

Estas dos compañías junto con sus mercados de aplicaciones comenzaron a crecer a un ritmo impensable, superando el millón de aplicaciones disponibles tanto en el mercado de Apple[6] como en el de Google[7]. Además, la importancia y relevancia de estos dispositivos dentro del mercado, según las cifras proporcionadas por el informe anual de Ericsson [Ericsson15], se consolida igualando el número de terminales móviles operativos a la población mundial.

En términos de innovación tecnológica, dichos dispositivos han adquirido propiedades y capacidades muy propias de un ordenador personal, siendo hoy en día un elemento clave en términos de tráfico de datos por Internet. Analizando los datos del último informe de CISCO sobre tráfico móvil [Cisco14], el tráfico

---

[1]http://www.youtube.com/watch?v=t4OEsI0Sc_s
[2]http://play.google.com
[3]http://www.apple.com/itunes
[4]http://www.google.com/mobile/gmail
[5]http://www.google.com/mobile/maps
[6]http://tnw.to/p3I1O
[7]http://mashable.com/2013/07/24/google-play-1-million

mundial de datos móviles creció un 81% en 2013. Además, el tráfico mundial de datos móviles alcanzó 0.82 exabytes[8] por mes a finales de 2012, 1.5 por mes a finales de 2013 y un 55% más a finales del 2014 respecto al año anterior [Ericsson15].

Viendo estas tendencias en cuanto a conexiones y evolución de los terminales, se ha producido un importante cambio en el desarrollo de aplicaciones tanto web como móvil. Este tipo de aplicaciones han dado un gran salto del contexto típico de escritorio a cualquier tipo de entorno (incluso exterior). El vertiginoso aumento del número de negocios que migran sus servicios a Internet y a los dispositivos móviles ha causado una gran demanda en el desarrollo de aplicaciones de movilidad. Siendo el desarrollo de las mismas cada vez más exigente en términos de tiempo, demandando fases de pruebas cada vez más rápidas.

Desde el punto de vista de los dispositivos en los que residen las aplicaciones móviles, detectamos una gran heterogeneidad existiendo numerosos modelos con características muy diversas. Además, las interfaces tanto de entrada como de salida son pequeñas, por lo que requieren especial atención. En cuanto a los entornos donde estas aplicaciones son utilizadas, muestran una conectividad extremadamente variable, una elevada probabilidad de que el usuario sea interrumpido (p.ej., llamadas de teléfono o demanda de atención del entorno como al cruzar un paso de cebra) y un tiempo muy limitado en la ejecución de las tareas que son realizadas con estas aplicaciones. En definitiva, un fuerte impacto de los elementos del contexto en la interacción del usuario con las aplicaciones móviles.

Aunque la investigación en usabilidad móvil ha constituido una nueva área de firme expansión, existen claras limitaciones dentro del campo. En primer lugar, existe una fuerte necesidad en el estudio de las características del entorno dentro del estudio de la

[8]Un exabyte es una unidad de medida de almacenamiento de datos equivalente a 10^9 gigabytes

usabilidad de aplicaciones móviles. El coste en cuanto a tiempo y esfuerzo de estas evaluaciones es muy elevado si se requieren resultados fiables en las pruebas realizadas en entornos reales.

Por ello, es necesario un mayor esfuerzo en la línea de buscar una adecuación de las técnicas y métodos de usabilidad (hasta ahora débilmente establecidos a las particularidades de este tipo de dispositivos) así como en la integración de aspectos relacionados con la variabilidad propia y el estudio de un contexto móvil.

Dicho esto concluimos formalmente la problemática que abordaremos en este trabajo:

> *No existe una metodología de evaluación de aplicaciones móviles que requiera pocos recursos en términos de tiempo y esfuerzo, que muestre resultados fiables para el estudio de la usabilidad de las mismas y estudie las características propias de su contexto.*

Para abordar dicha problemática, proponemos una hipótesis en la siguiente sección.

1.2. HIPÓTESIS

Podemos resumir la hipótesis cuya verificación promueve la presente investigación de la siguiente forma:

> *Es posible reducir los recursos necesarios en la evaluación de la usabilidad de aplicaciones móviles sin comprometer la calidad de los resultados mediante una nueva metodología de evaluación centrada en una base de conocimiento.*

Como consecuencia de la formulación de dicha hipótesis se deben abordar y aclarar varios aspectos. Por un lado, debemos aclarar que entendemos como recursos al conjunto de elementos disponibles para resolver las necesidades de llevar a cabo una evaluación de usabilidad. Dentro de estos elementos se encuentran: el conjunto de agentes y dispositivos de ayuda y

soporte a la evaluación, el número de usuarios necesarios, el tiempo de evaluación necesario y el coste económico de la evaluación. Por otro lado, entendemos que no se compromete la calidad de los resultados cuando el número de problemas de usabilidad encontrados en una evaluación no desciende al reducir los recursos utilizados en la misma y los datos de interacción capturados no han sufrido ningún sesgo por la estrategia de captura utilizada.

1.3. OBJETIVOS

Con el fin de enmarcar con mayor precisión el ámbito de este trabajo, presentamos en esta sección el objetivo general fijado para la tesis que hará posible su validación. Lo especificamos de la siguiente manera:

> *El objetivo general de esta tesis es desarrollar una metodología basada en una base de conocimiento para la evaluación de usabilidad de aplicaciones móviles que haga uso de una cantidad de recursos reducida.*

Para ello, el diseño debe cumplir ciertas necesidades identificadas en el análisis del estado del arte (ver sección 2.3) que se detallan a continuación:

- *Debe ser capaz de ofrecer resultados para evaluaciones cuya finalidad sea tanto formativa como sumativa*. Es decir, mediante esta solución se debe poder tanto identificar los problemas de usabilidad más significativos que hace que los usuarios no completen las tareas o sean más ineficientes (p.ej., botones mal situados) como el grado en el que una aplicación cumple ciertos requisitos (p.ej., el usuario debe completar una tarea en menos de un minuto).
- *La cantidad de recursos necesaria debe ser reducida*. Entendiendo como recursos a reducir el tiempo de la evaluación y el equipamiento necesario para desarrollar la misma.

- *La calidad de los resultados no debe disminuir*. Aunque la cantidad de recursos sea reducida, la fiabilidad de los datos debe ser alta. Es decir, durante la realización de las pruebas no debe generarse ningún sesgo por las herramientas de captura de las mismas y los datos analizados no deben mostrar variaciones debidas al evaluador de las mismas en su análisis.
- *Se debe preservar la privacidad de los usuarios que realizan las pruebas*. Los usuarios podrán realizar pruebas de usabilidad en cualquier contexto sin que se vea comprometida la privacidad del mismo (p.ej., en su propia casa).
- *Se debe posibilitar el estudio de un modelo de contexto detallado*. Se debe estudiar un modelo de contexto que abarque no solo características básicas del usuario y las tareas realizadas, sino también características del entorno donde desarrolla el usuario la interacción con la aplicación.

Para la realización del objetivo definido y la satisfacción de las necesidades concretadas, se definen varios objetivos específicos. En consonancia con el objetivo general, los tres objetivos específicos definidos son:

- *OE1. Definir una metodología de evaluación de aplicaciones móviles*. Dicho de otro modo, plantear el modelo teórico que servirá de base para la nueva metodología de evaluación. Se incluirá en dicho modelo tanto el conjunto de fases y pasos que conformarán toda la evaluación, como la definición de la base de conocimiento que será el principal cimiento de la misma.
- *OE2. Implementar una plataforma de soporte a la nueva metodología*. Con ello dispondremos de una plataforma software que permita la automatización y ejecución paralela de ciertos pasos definidos en el modelo teórico y que ayude tanto a los usuarios que realizan las pruebas como a los evaluadores de las aplicaciones móviles. Con ello lograremos una mejora en la aplicabilidad práctica de la solución teórica definida.

- *OE3. Verificar los resultados del uso de la nueva metodología*. Para lograr este fin, se diseñarán y ejecutarán experimentos con usuarios reales y con herramientas de simulación que permitan verificar tanto las diferentes propiedades clave de la metodología definida como la validez de la hipótesis planteada.

Para alcanzar los objetivos específicos definidos y por consiguiente, la consecución del objetivo general y la validación de la hipótesis, mostramos varios objetivos operacionales mediante la tabla 1.1.

Objetivo específico	*Objetivo Operacional*
OE1	*Definir una nueva metodología de evaluación de usabilidad para aplicaciones móviles*
OE1	*Definir una base de conocimiento y sus modelos para componer la metodología propuesta*
OE2	*Implementar una herramienta de captura de los modelos que conforman la base de conocimiento*
OE2	*Implementar una plataforma de soporte que ayude a la correcta ejecución de la nueva metodología*
OE3	*Realizar un experimento en el que validemos la correcta captura de los modelos que conforman la base de conocimiento*
OE3	*Realizar un experimento con usuarios reales que estudie si el uso de la base de conocimiento nos permite definir una evaluación que permita encontrar más errores de interacción*
OE3	*Diseñar y ejecutar un experimento con usuarios reales en el que validemos de un modo empírico la principal hipótesis definida en este trabajo*

Tabla 1.1 Objetivos operacionales y sus relaciones con los objetivos específicos

Cabe decir que el objetivo de la información presentada no corresponde al plan del trabajo detallado que hemos mantenido en esta tesis, sino tan solo mostrar los objetivos operacionales más relevantes desde el punto de vista científico junto con la relación con el objetivo específico correspondiente.

Por otro lado, en las *contribuciones* que ofrecemos en este trabajo (detalladas en la sección 6.2) destacamos dos principales *contribuciones científicas* mediante el capítulo 3:

- *Definición teórica de la metodología con la cual podemos utilizar conocimiento previo para mejorar la eficiencia de la evaluación de la usabilidad de aplicaciones móviles* (ver sección 3.1)
- *Definición de una base de conocimiento compuesta por cuatro modelos* (ver sección 3.2).

En el capítulo 4, destacamos la principal *contribución técnica:*

- *Construcción de un conjunto de herramientas para posibilitar el uso de la metodología tanto a desarrolladores* (ver secciones 4.3 y 4.5) *como a usuarios de pruebas* (ver sección 4.6).

Para concluir con esta sección, consideramos importante especificar que el enfoque en el que se centra este trabajo es la evaluación de la usabilidad de aplicaciones software diseñadas para dispositivos móviles.

Debemos añadir dentro del marco de trabajo de esta tesis doctoral dos principales detalles:

- Este trabajo se centra en evaluaciones de usabilidad donde los usuarios que realizan las pruebas son reales, al igual que los entornos donde se realizan las mimas.
- Limitamos la atención a las aplicaciones evaluadas cuyo estado dentro de su implementación corresponda a un estado funcional (sean prototipos funcionales o versiones finales) con el que se puedan desempeñar las tareas para las cuales han sido creadas.

1.4. METODOLOGÍA

Sabiendo qué objetivos vamos a abordar en este trabajo procedemos a explicar cómo los vamos a desarrollar. Para esto, es necesaria una metodología que busca una continua retroalimentación.

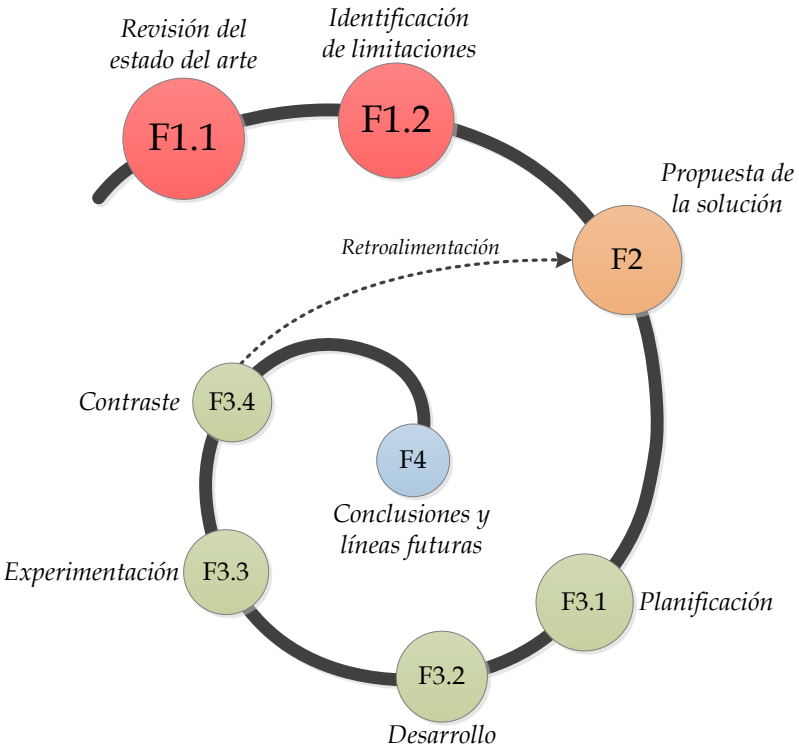

Figura 1.2 Fases de la metodología de investigación seguida en esta tesis

Como se muestra en la figura 1.2, hemos desarrollado una metodología cíclica basada en las siguientes fases:

- *FASE 1. Actualización de nuestro conocimiento sobre la materia.* Esta actualización se realiza mediante una revisión del estado del arte referente al campo de la usabilidad y más concretamente al de usabilidad en dispositivos móviles. Para ello nos hemos basado en la búsqueda, estudio y análisis de las publicaciones de la comunidad científica en este ámbito, explorando las revistas y actas de los congresos más relevantes. Con ello hemos tomado consciencia de las

limitaciones y problemas de los trabajos existentes y hemos identificado las potenciales áreas de contribución científica. Dicha fase queda plasmada en el capítulo 2.

- *FASE 2. Propuesta de la solución que afronte los problemas encontrados*. La segunda fase comienza con la propuesta de la solución a la problemática afrontada y el enunciado de la hipótesis. Gracias a la difusión mediante publicación en congresos científicos y al desarrollo de proyectos de investigación ligados al ámbito de esta tesis, hemos mantenido reuniones con expertos que han originado una retroalimentación que nos ha permitido refinar y ajustar la solución planteada.
- *FASE 3. Planificación, desarrollo, evaluación y contraste*. Dentro de esta fase se incluyen la planificación y el desarrollo de los diferentes elementos que conforman la solución. La planificación implica la definición de cómo se van a desarrollar dichos elementos y el modo en el que se va a realizar su validación. El desarrollo toma en cuenta tanto el diseño de las diferentes partes como el desarrollo en sí. El resultado de este desarrollo se muestra en el capítulo 3 y 4. Por otro lado, la evaluación abarca el diseño de los experimentos y su ejecución para medir de forma cuantitativa los indicadores que validarán la solución, cuyos resultados se exponen en capítulo 5. Además, contrastamos con la comunidad científica los resultados obtenidos determinando la relevancia de la investigación y originando la retroalimentación necesaria para refinar y ajustar la solución planteada.
- *FASE 4. Elaboración de conclusiones y líneas futuras*. Finalmente, cuando se han realizado las mejoras de la solución y se han obtenido los resultados que han demostrado la validez de la solución, se procede a elaborar las conclusiones finales y plantear las posibles líneas futuras de esta investigación. El resultado de esta fase se expone en el capítulo 6 de este trabajo.

1.5. ESTRUCTURA DE LA TESIS

La estructura del documento que explica la presente tesis doctoral está organizada de la siguiente forma:

Capítulo 1. Hemos proporcionado una visión inicial del estado general del ámbito de trabajo de la tesis, así como las motivaciones generales que han propiciado su realización. Hemos resumido además la principal problemática abordada que esta tesis pretende resolver así como las características más relevantes de la solución propuesta, la hipótesis de trabajo de la tesis y los objetivos que guían su desarrollo.

Capítulo 2. Proporcionamos una visión general del estado del arte relativo al área de estudio. En primer lugar realizamos una introducción de la usabilidad y su evaluación en términos generales para fijar de este modo los pilares conceptuales de este trabajo. Posteriormente centramos la atención en las aplicaciones móviles, estudiando sus limitaciones desde la perspectiva de la usabilidad y los retos que se deben abordar en la evaluación de las mismas. Finalmente, ofrecemos un análisis de los retos y de cómo son afrontados en los trabajos más relevantes para inspirar el enfoque propuesto en esta tesis.

Capítulo 3. Describimos en este capítulo el grueso del trabajo de la tesis: la metodología de evaluación de usabilidad en entornos móviles. Primeramente se exponen las principales características que debe cumplir para lograr completar los requisitos específicos de la misma. Posteriormente, presentamos la metodología con su descripción general, el detalle de cada una de las fases que la conforman y una explicación de la composición de la base de conocimiento, principal pilar de la misma. Posteriormente, detallaremos los modelos que componen dicha base de conocimiento. Finalmente, describiremos la relación de las fases de la metodología con la base de conocimiento presentada.

Capítulo 4. Describimos la herramienta de soporte que ha sido implementada como ayuda a la metodología. Comienza con la recopilación de los requisitos que debe cumplir. Después,

describimos la plataforma móvil elegida para la implementación de los componentes móviles de la plataforma de soporte. Habiendo justificado esta elección, realizamos una descripción general de la plataforma para finalmente detallar los elementos que la conforman, describiendo su funcionalidad y su arquitectura.

Capítulo 5. En este capítulo abordamos la experimentación como medio para la validación de la hipótesis enunciada en la tesis. Analizamos la hipótesis, seleccionamos y justificamos las estrategias de verificación más apropiadas, describimos las actividades y los experimentos realizados, mostramos los resultados de éstos y concluimos la veracidad o no de la hipótesis.

Capítulo 6. Para finalizar este trabajo, en este capítulo recogemos las conclusiones de la tesis, así como las líneas futuras de trabajo y líneas de mejora.

Finalmente, se concluye con un apartado dedicado a la bibliografía, con todas las referencias mencionadas en esta tesis.

CAPÍTULO 2

ESTADO DEL ARTE

«Si no quieres repetir el pasado, estúdialo»,
Baruch Benedict Spinoza (1632-1677)

ÍNDICE DE CAPÍTULO 2

2.1. Introducción a la usabilidad 16
2.1.1. Definición de la usabilidad 16
2.1.2. Dimensiones de usabilidad 18
2.1.3. Ingeniería de la usabilidad 19
2.1.4. Métodos de evaluación de usabilidad 19
2.2. Usabilidad en los dispositivos móviles 36
2.2.1. Limitaciones de las aplicaciones móviles 37
2.2.2. Evaluación de la usabilidad de aplicaciones móviles 40
2.2.3. Limitaciones de la evaluación de la usabilidad 44
2.2.4. Importancia del contexto en las aplicaciones móviles 46
2.3. Metodologías de evaluación 48
2.3.1. Criterios de análisis 48
2.3.2. Métodos de evaluación de usabilidad 51
2.4. Conclusiones y solución propuesta 60
2.4.1. Análisis y conclusiones 60
2.4.2. Solución propuesta 62

Mediante este capítulo, realizaremos una revisión superficial de los principales conceptos que engloban el campo de la usabilidad. Una vez explicado el campo a nivel genérico nos introduciremos en detalle en el campo de la evaluación de la usabilidad de aplicaciones móviles. Posteriormente, realizaremos un análisis de los principales problemas para identificar los que se abordarán dentro de este trabajo.

2.1. INTRODUCCIÓN A LA USABILIDAD

Jakob Nielsen[9], una de las personas más respetadas en el ámbito mundial sobre usabilidad y gran defensor de la misma, afirma en uno de sus libros más conocidos [Nielsen94a] que "*tu mejor suposición no es lo suficientemente buena*". Esta idea o eslogan es la principal razón de la existencia de la evaluación de la usabilidad de los productos, ya que todos los usuarios tienen una capacidad extrema para encontrar malas interpretaciones acerca del producto evaluado. Por ello, es de vital importancia que estudiemos cómo los usuarios realmente utilizan nuestro producto y si este uso cumple con las expectativas supuestas y los objetivos del mismo.

2.1.1. DEFINICIÓN DE LA USABILIDAD

Centrándonos en este problema definimos la usabilidad. Aunque podemos definir coloquialmente la usabilidad como la propiedad que tiene un determinado sistema de ser fácil de usar y fácil de aprender, debemos detallar su significado mediante definiciones más formales. Dentro de la comunidad científica se ha invertido mucho esfuerzo en definir correctamente el término usabilidad, presentamos los más relevantes.

Uno de los principales autores en reconocer la importancia de la usabilidad fue Shackel [Shackel91], la define como *la capacidad de un sistema en términos funcionales humanos de ser usado fácilmente y efectivamente por un rango específico de usuarios, dando una ayuda y soporte específicos, para completar un rango específico de tareas en un rango de escenarios específico*. Además, descompone la usabilidad en cuatro dimensiones: *efectividad* (rendimiento en realizar las tareas), *facilidad de aprendizaje*, *flexibilidad* (adaptación a varias tareas) y *actitud* (del usuario frente al sistema).

[9]http://www.nngroup.com/people/jakob-nielsen

Bevan et al. [Bevan+91] definen la usabilidad *como la facilidad de uso (incluyendo la facilidad de aprendizaje cuando sea relevante) y el nivel de aceptabilidad de un producto para una clase de usuarios concretos realizando tareas específicas en entornos específicos*.

Por otro lado, el mencionado Nielsen [Nielsen03] la define como *un atributo de la calidad que define cómo de fácil es de usar una interfaz de usuario*, añadiendo cinco componentes: *facilidad de aprendizaje*, *eficiencia*, *facilidad de ser recordado*, *errores* y *satisfacción*. Mediante esta definición percibimos la evolución en el campo y el acuerdo con los estándares originados, como comprobaremos a continuación.

Dentro de los estándares existentes en el campo, destacamos la definición de usabilidad más conocida: la proporcionada por el estándar ISO 9241-11:1998 [ISO98]:

> *La usabilidad es el nivel con el que un producto es usado por determinados usuarios para conseguir objetivos concretos con efectividad, eficiencia y satisfacción en un contexto en uso específico.*

Dicha definición es tomada en cuenta en numerosos estándares [ISO99a, NIST07] pero es de especial mención el estándar ISO 9126-1 [ISO01]. En él se define la calidad en uso del mismo modo que la usabilidad, con la diferencia de que considera la usabilidad como una simple característica de calidad en uso, un término más general. Concretan que la calidad en uso puede ser afectada tanto por la calidad interna del software como por la externa. Por ello, redefine el término usabilidad como *la capacidad que tiene un producto software para ser entendido, aprendido, usado y considerado atractivo por el usuario cuando es utilizado bajo unas condiciones específicas*. Por otro lado, Quesenbery [Quesenbery04] desglosa la definición de la usabilidad en cinco dimensiones, atribuyéndolas el nombre de las "*5Es*": *efectiva* (*Effective*), *eficiente* (*Efficient*), *interesante* (*Engaging*), *tolerante a fallos* (*Error tolerant*) y *fácil de aprender* (*Easy to learn*).

Posteriormente, surgió el estándar ISO/IEC 25000 SQuaRE [ISO14] con el objetivo de sustituir a la serie de estándares ISO/IEC 9126 y el actualmente descartado ISO/IEC 14598 [ISO99b]. Sin embargo, consideran la definición del estándar ISO 9241-11:1998 [ISO98] lo suficientemente completa como para conservarla en la nueva especificación SQuaRE.

Como resumen de las definiciones detectamos que predominan tres elementos críticos para los cuales un producto está diseñado, sea cual sea su naturaleza: *usuarios determinados*, *objetivos concretos* y *contexto de uso específico*.

2.1.2. DIMENSIONES DE USABILIDAD

Recapitulando las definiciones de usabilidad de la comunidad especializada en este campo, observamos que muchos de ellos ofrecen la descripción de la usabilidad mediante la definición de los atributos que la componen.

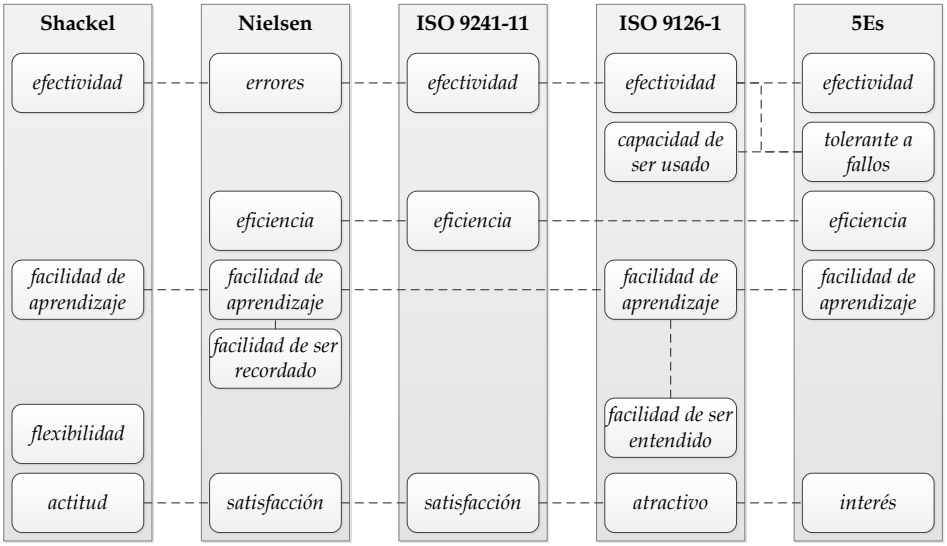

Figura 2.1 Dimensiones de usabilidad de los principales autores

Estudiando los atributos con los cuales la definen los diferentes trabajos expuestos, percibimos que hay autores que nombran ciertos aspectos relacionados (ver líneas discontinuas de la figura 2.1). Además podemos observar que existe una ligera discrepancia en la interpretación de qué atributos son más importantes. Por

ejemplo, dentro de la definición de usabilidad más aceptada (ISO 9241-11:1998 [ISO98]), la *facilidad de aprendizaje* no está tan considerada.

2.1.3. INGENIERÍA DE LA USABILIDAD

Con el objetivo de mejorar la usabilidad de las interfaces de usuario nace la disciplina de la ingeniería de la usabilidad. Se define como sigue [Whiteside+87, Nielsen94a]:

> *La ingeniería de la usabilidad es una disciplina que proporciona procedimientos y métodos para conseguir una usabilidad aceptable en el diseño de interfaces de usuario.*

Dentro de los principales beneficios que incluye el uso de esta disciplina es la reducción del tiempo desperdiciado en el rediseño de una interfaz y del proceso de desarrollo en general, además de proveer métodos objetivos y no sesgados [Mayhew99, Rosson+02] y ahorrar recursos desde el punto de vista de los usuarios finales del producto evaluado [Lederer+92]. Aunque en sus comienzos el foco de esta disciplina era la ayuda en la fase de diseño de la interfaz de usuario, se ha ampliado a todas las fases del proceso, cobrando especial fuerza en la evaluación [Folmer+04].

2.1.4. MÉTODOS DE EVALUACIÓN DE USABILIDAD

La evaluación de la usabilidad es un área dentro de la Ingeniería de usabilidad compuesta de métodos para medir aspectos de la usabilidad de la interfaz de usuario de un sistema e identificar problemas específicos en diferentes etapas del desarrollo software [Nielsen94a]. Según Hilbert y Redmiles [Hilbert+00] se define del modo siguiente:

> *La evaluación de la usabilidad es el acto de medir atributos de la usabilidad o identificar problemas potenciales que afectan a la usabilidad de un sistema o dispositivo respecto a unos usuarios concretos,*

> *desarrollando unas tareas concretas, en un contexto particular.*

Realizar una evaluación de usabilidad en sí es una tarea que consta de diferentes pasos dependiendo del método utilizado. Estos pasos suelen consistir en una cronología de captura de datos de usabilidad (p.ej., tiempo de ejecución de tareas, errores, satisfacción del usuario, incumplimiento de pautas de usabilidad…), análisis o interpretación de los datos capturados para inferir problemas de usabilidad; y crítica, que consiste en sugerir medidas o soluciones para paliar los problemas identificados [Ivory01, Perallos07].

2.1.4.1. PRINCIPALES MÉTODOS DE EVALUACIÓN DE USABILIDAD

Para tomar conciencia de los métodos de evaluación que existen, se ha realizado un estudio de los métodos más extendidos dentro de la literatura. Para realizar una enumeración organizada presentamos la descripción de los principales métodos en base al tipo de técnica utilizada: métodos de inspección, métodos de indagación y métodos de pruebas de usabilidad.

2.1.4.1.1. MÉTODOS DE INSPECCIÓN

Los métodos de inspección son métodos en los que expertos en usabilidad o desarrolladores software estudian y examinan los elementos que componen una interfaz de usuario o prototipo para determinar si cumplen ciertos principios de usabilidad establecidos. Enumeramos los principales métodos dentro de este tipo:

- *(1) Evaluación heurística.* Es un método de inspección que consiste en analizar la interfaz de usuario y comprobar que cumple una serie de principios reconocidos o heurísticas centradas en la usabilidad. Las heurísticas más extendidas dentro de este método son las 10 heurísticas de Nielsen [Nielsen95a], el principal inconveniente es que presentan heurísticas universales y presentadas a cualquier ámbito. Por ello han surgido nuevas heurísticas aplicadas a

dominios concretos, incluido el dominio de las aplicaciones móviles [Bertini+06, Korhonen+06, Inostroza+12]. Otro inconveniente es que son necesarios expertos para la evaluación con dicho método.

- *(2) Inspección de estándares.* Similar a la evaluación heurística, el método de inspección de estándares realiza un examen detallado de la interfaz del software con el objeto de validar el cumplimiento de todos los puntos definidos en un estándar [Wixon+94]. La principal desventaja de este método es que el evaluador que realice la inspección debe ser experto en el estándar o estándares seleccionados. Dentro de este método se incluye la inspección de consistencia, que consiste en un examen en el que se valide que tanto el diseño, estructura e interacción siga una coherencia dentro de toda la aplicación [Wixon+94].
- *(3) Recorrido cognitivo (Cognitive walkthrough).* El recorrido cognitivo es una alternativa a la evaluación heurística e inspección de estándares (ya que no se basa en ninguna pauta establecida) donde el evaluador empatiza simulando que es un usuario, realizando una revisión de todos los pasos que debe realizar el usuario para lograr los objetivos de todas las tareas que pueden realizar con el sistema a evaluar [Rieman+95]. Con este trabajo el evaluador intenta predecir problemas de usabilidad sin necesidad de usuarios. Para ponerse en el papel del usuario, el evaluador debe disponer de una descripción general de cómo son los usuarios, una descripción específica de qué tareas se pueden realizar y una lista de acciones para realizar satisfactoriamente estas tareas. Desafortunadamente, los resultados muestran mucha dependencia del evaluador.
- *(4) Recorrido pluralista (Pluralistic walkthrough).* Al igual que el recorrido cognitivo, los participantes asumen el papel de usuario. Sin embargo, en este caso [Bias94] participan en un grupo de análisis usuarios representativos junto con desarrolladores y expertos en usabilidad. Cada participante va anotando la forma en la que realizaría cada una de las

tareas que se presentan en el grupo. Una vez finalizadas, se presentan las anotaciones para debatir sobre ello e identificar posibles problemas. Este método es muy estimado por su capacidad para ser utilizado en las etapas más tempranas de diseño, lo que permite la resolución de problemas de usabilidad rápida y temprana. Además, permite la detección de un mayor número de problemas de usabilidad debido a la intervención de varios participantes de diferentes puntos de vista.

2.1.4.1.2. MÉTODOS DE INDAGACIÓN

Los métodos de indagación son métodos en los que expertos en usabilidad obtienen información acerca de los gustos, necesidades y modelos mentales de los usuarios frente al sistema a evaluar. Esto es obtenido mediante la observación de los usuarios en sus entornos o formulándoles preguntas a los mismos. Enumeramos los principales métodos dentro de este tipo:

- *(5) Indagación contextual*. La indagación contextual [Holtzblatt+93] es un método estructurado de entrevista de campo, basado en tres principios fundamentales: comprender el contexto en el que se utiliza el nuevo producto, el usuario es parte del proceso de diseño y que este proceso debe estar centrado en la usabilidad. Este método ayuda al evaluador a comprender y mejorar el diseño del producto. El principal inconveniente de este método es que requiere tiempo ya que es necesario hacer entrevistas a los usuarios y desplazarse al entorno real donde se utilizará el producto. Este método suele utilizarse en la etapa de captura de requisitos.
- *(6) Observación de campo*. Es un método que se usa en las primeras fases de desarrollo muy similar a la indagación contextual con la diferencia de que se centra en las actividades y tareas que realizan los usuarios [Liberman99]. Para realizar una observación de campo el evaluador se desplaza al entorno donde se utilizará el producto y se estudia a los usuarios. En este entorno el evaluador

desempeña una actividad de observación con la que recupera información y la complementa mediante entrevistas. Mediante este método se estudian todos los pasos de cada tarea, el contexto de su realización y los diferentes modelos mentales.

- *(7) Grupos focales (focus groups).* Es un método en el que se forma un grupo de usuarios representativos y se discuten aspectos relacionados con el sistema propuestos por un evaluador experto, el cual actúa de moderador [Kitzinger95]. Mientras el grupo debate sobre los aspectos planteados, el evaluador recoge la información generada para obtener como resultado un conjunto de opiniones y actitudes. Este método es utilizado en las últimas fases del desarrollo.
- *(8) Entrevistas.* Varios métodos como la indagación contextual y la observación de campo utilizan como parte de ellos las entrevistas. Si la estudiamos como un método más, la entrevista consiste en una conversación donde usuarios reales responden a una serie de preguntas sobre el producto a evaluar formuladas por el evaluador. Dentro de las entrevistas existen dos tipos: estructuradas y no estructuradas. La entrevista no estructurada no dispone de guion y adopta una fluidez similar a una conversación. Por el contrario, en una entrevista estructurada existe una agenda preestablecida con cuestiones específicas. Aunque su uso es más frecuente en las fases finales para estudiar la satisfacción con el sistema y en el estudio de los requisitos, puede ser utilizado en cualquier etapa del desarrollo [Preece+94, Macaulay12]. Una ventaja de las entrevistas es que los errores y malentendidos pueden ser identificados y aclarados de un modo sencillo. Desafortunadamente, la naturaleza no estructurada de los datos resultantes hace muy factible la malinterpretación u omisión de resultados.
- *(9) Cuestionarios*. Este método consiste en la simple distribución de cuestionarios compuestos por una lista de preguntas que los usuarios del sistema a evaluar deben

responder. La mayor parte de los cuestionarios desarrollados para este fin disponen de respuestas con escala Likert [Likert32], siendo ésta una escala numérica. Esto nos permite cuantificar ciertos aspectos como la satisfacción del usuario con el sistema de un modo sencillo, como es el caso del cuestionario QUIS (Questionnaire for User Interface Satisfaction) [Chin+88] y EUCSI (End-User Computing Satisfaction Instrument) [Doll+94]. Es de especial mención el cuestionario SUS (System Usability Scale) [Brooke96], debido a su amplio uso ya que nos permite medir mediante diez preguntas, la usabilidad de cualquier tipo de sistema. Este tipo de método es utilizado en las últimas fases de desarrollo donde ya se puede probar el sistema con usuarios.

- *(10) Self-reporting logs.* Es un método sencillo donde los usuarios del producto evaluado completan un informe en el que registran las acciones que realizan y las observaciones dentro de esa interacción [Hom98]. Posteriormente, el evaluador analiza los datos que proporcionan los usuarios. La principal desventaja es que un usuario se ve constantemente interrumpido al tener que documentar todas las acciones. Además, hay una probabilidad alta de omitir aspectos de la evaluación relevantes.
- *(11) Screen Snapshots.* Este método sirve de complemento al anterior, es un método en el que el usuario realiza capturas de pantalla en diferentes momentos durante la ejecución de una tarea o una serie de tareas [Hom98].

2.1.4.1.3. MÉTODOS DE PRUEBAS DE USABILIDAD (USABILITY TESTING)

Estos métodos, conocidos como "*Empirical Usability testing*" o "*Usability testing*", consisten en reclutar a una muestra representativa de usuarios y estudiar cómo desarrollan tareas típicas utilizando el sistema. Destacamos los siguientes:

- *(12) Medidas de rendimiento.* Algunas pruebas de usabilidad están dirigidas a determinar el rendimiento del sistema desde la perspectiva de la usabilidad con datos

cuantitativos [Hom98]. A menudo, estas métricas se utilizan como objetivos durante el diseño de un producto, los cuales deben ser cuantificables. En este método se debe disponer de un sistema que tenga la funcionalidad para cuantificar estas medidas, ya sea un sistema implementado o un prototipo. Para aplicar este método los usuarios realizan tareas mientras su interacción es grabada mediante algún tipo de técnica o herramienta. Posteriormente el evaluador analiza las medidas cuantificadas.

- *(13) Protocolo de pensamiento en voz alta (Thinking Aloud Protocol).* En este método, mientras los usuarios realizan pruebas con el producto a evaluar, expresan en voz alta y libremente sus pensamientos, sentimientos y opiniones sobre cualquier aspecto del sistema [Hom98]. Esta técnica permite a los evaluadores comprender el modelo mental del usuario, siendo utilizado en cualquier etapa de desarrollo.
- *(14) Protocolo de emisión de preguntas (question-asking protocol).* Como extensión al protocolo de pensamiento en voz alta, los evaluadores realizan preguntas mientras los usuarios realizan pruebas con el producto a evaluar [Dumas+99]. La capacidad del usuario para responder a las preguntas formuladas puede ayudar a ver qué partes de la interfaz del producto son obvias y cuales problemáticas.
- *(15) Aprendizaje por co-descubrimiento.* Es otro método que se extiende del protocolo en voz alta, donde dos participantes intentan realizar tareas conjuntamente mientras son observados por el evaluador. La principal ventaja de este método sobre el protocolo de pensamiento en voz alta es que la interacción entre los dos participantes puede suponer registrar más ideas que con uno solo.
- *(16) Test retrospectivo.* Esta técnica consiste en grabar al usuario realizando tareas y posteriormente analizar con él la grabación de la sesión. Este método, siendo utilizado en cualquier etapa de desarrollo, dispone de comentarios más extensos ya que en el momento del análisis se puede

interrumpir el video y analizar con tranquilidad situaciones concretas [Nielsen94a]. El principal problema es la demanda de tiempo requerido, ya que como mínimo se dobla por prueba realizada.

- *(17) Coaching method.* En este método el usuario realiza tareas donde el propio evaluador le guía para el desarrollo de las mismas [Nielsen94a]. Durante la prueba el usuario realiza preguntas relacionadas con cualquier aspecto del sistema. Este método se centra en la ayuda a usuarios inexpertos, analizando las características del sistema para proporcionar un sistema más fácil de comprender y aprender.
- (18) *Análisis de trazas (log analysis).* Normalmente con la ayuda de algún tipo de herramienta de captura y análisis, el evaluador estudia la interacción de los usuarios con el sistema mediante la revisión de las trazas generadas cuando realizan una serie de tareas con el mismo [Vokar+01]. La principal ventaja es que se pueden extraer conclusiones de interacciones reales sin sesgos añadidos, ya que el evaluador analiza posteriormente los resultados. Una de las principales ventajas de este método es la sencillez y rapidez de la recogida de datos si es realizada de un modo remoto. Concretamente, a esta variación se la conoce en la literatura como *evaluación remota (remote testing)*, muy utilizada en la evaluación de usabilidad web [Hong+01a, Cuddihy+05, Symonds11].

Dependiendo de los recursos que se dispongan, serán más idóneos unos métodos u otros. Para disponer de una visión más clara resumimos en la tabla 2.1 y tabla 2.2, los diferentes métodos junto con las etapas dentro del ciclo de vida software, los recursos y tiempo necesarios, si son necesarios usuarios finales y el grado de

experiencia de los evaluadores para realizar el método (según datos extraídos de Folmer y Bosch [Folmer+04] y UsabilityNet[10]).

Método[11]	*(1)*	*(2)*	*(3)*	*(4)*	*(5)*	*(6)*	*(7)*	*(8)*	*(9)*
F. captura de Requisitos					*x*	*x*		*x*	
F. de Diseño	*x*	*x*	*x*	*x*				*x*	
F. de Codificación	*x*	*x*	*x*				*x*	*x*	
F. de Pruebas	*x*	*x*	*x*				*x*	*x*	*x*
Tiempo/recursos necesarios	*alto*	*alto*	*alto*	*alto*	*alto*	*alto*	*alto*	*alto*	*alto*
Usuarios finales	*no*	*no*	*no*	*sí*	*sí*	*sí*	*sí*	*sí*	*sí*
Experiencia del evaluador	*alta*	*alta*	*alta*	*alta*	*alta*	*alta*	*alta*	*alta*	*alta*

Tabla 2.1 Propiedades de los principales métodos de la evaluación de usabilidad (1/2)

Método[11]	*(10)*	*(11)*	*(12)*	*(13)*	*(14)*	*(15)*	*(16)*	*(17)*	*(18)*
F. captura de Requisitos									
F. de Diseño			*x*	*x*	*x*	*x*	*x*	*x*	*x*
F. de Codificación	*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*
F. de Pruebas	*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*	*x*
Tiempo/recursos necesarios	*alto*	*alto*	*bajo*	*alto*	*alto*	*alto*	*alto*	*alto*	*alto*
Usuarios finales	*sí*	*sí*	*no*	*sí*	*sí*	*sí*	*sí*	*sí*	*sí*
Experiencia del evaluador	*baja*	*baja*	*baja*	*alta*	*alta*	*alta*	*alta*	*alta*	*alta*

Tabla 2.2 Propiedades de los principales métodos de la evaluación de usabilidad (2/2)

Como apreciamos en las propiedades, numerosos métodos sólo pueden ser aplicados cuando se dispone de un diseño de interfaz de usuario o incluso un prototipo implementado pudiéndose aplicar en las fases de pruebas, siendo una minoría los que se centrar en fases preliminares. También observamos que requieren de evaluadores con ciertos conocimientos y experiencia. Finalmente, vemos que la mayoría de ellos requieren de usuarios finales en su ejecución.

[10]http://www.usabilitynet.org/tools/methods.htm

[11]Los métodos son: (1) Evaluación heurística, (2) Inspección de estándares, (3) Recorrido cognitivo, (4) Recorrido pluralista, (5) Indagación contextual, (6) Observación de campo, (7) Grupos focales, (8) Entrevistas, (9) Cuestionarios, (10) Self-reporting logs, (11) Screen Snapshots, (12) Medidas de rendimiento, (13) Protocolo de pensamiento en voz alta, (14) Protocolo de emisión de preguntas, (15) Aprendizaje por co-descubrimiento, (16) Test retrospectivo, (17) Coaching method y (18) Análisis de trazas (remota).

2.1.4.2. TAXONOMÍAS DE CLASIFICACIÓN DE MÉTODOS

En la revisión de la literatura referente a la evaluación de la usabilidad, hemos encontrado numerosos criterios y taxonomías con las que los autores clasifican los métodos, estas taxonomías son desarrolladas en función de ciertas características del método.

Un ejemplo es la clasificación en base al tipo de colaborador que participa en la evaluación de la usabilidad: con usuarios (métodos en los que se evalúa el producto con la interacción directa de usuarios representativos) o sin usuarios (no intervienen usuarios representativos, participan evaluadores expertos). Otro ejemplo es el propuesto por Coutaz [Coutaz95], que propone una clasificación en dos grupos: métodos experimentales, basados en la existencia de dispositivos físicos que utilizan usuarios para el estudio de la usabilidad; y métodos predictivos, basados en modelos teóricos de evaluación.

A continuación estudiamos las taxonomías más relevantes (ver figura 2.2) para realizar un posterior estudio de los métodos y herramientas de evaluación de aplicaciones móviles, principal área de estudio de esta tesis (ver apartado 2.2.2).

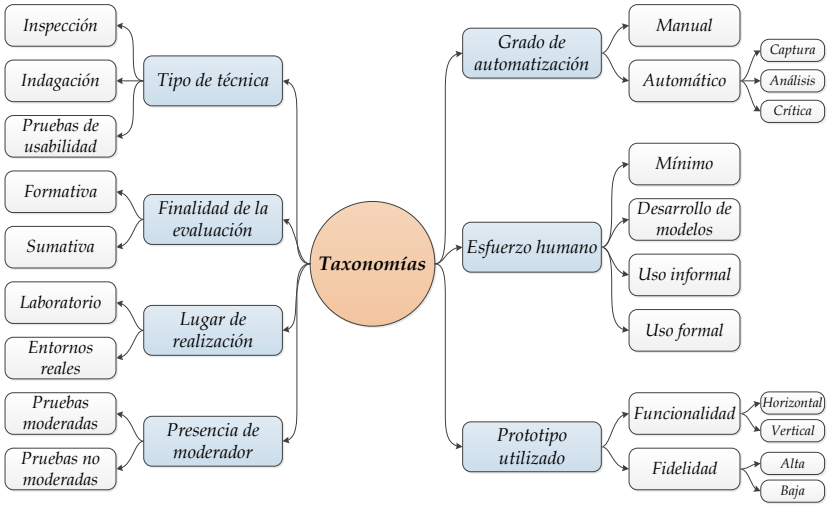

Figura 2.2 Taxonomías más relevantes para clasificar los métodos de evaluación de la usabilidad

2.1.4.2.1. TIPO DE TÉCNICA

Este tipo de criterio es uno de los más utilizados [Nielsen94a, Ivory03, Folmer+04]. Como ya hemos visto en la presentación de los principales métodos (ver apartado 2.1.4.1), se diferencian en tres principales grupos:

- *Métodos de inspección*. Son métodos en los que un grupo reducido de evaluadores, siendo estos expertos en usabilidad o desarrolladores software, estudian y examinan los elementos que componen una interfaz de usuario o prototipo. Mediante este examen, determinan en base a su propia experiencia y fundamentándose en reconocidos principios de usabilidad (heurísticos), si cumplen ciertos criterios para disponer de una usabilidad aceptable, recomendando mejoras si se consideran.
- *Métodos de indagación*. Con estos métodos los expertos en usabilidad estudian y observan a los usuarios y sus entornos para obtener información acerca de los gustos, necesidades y modelos mentales de los mismos frente al sistema a evaluar.
- *Métodos de pruebas de usabilidad*. Estos métodos, conocidos como "*Empirical Usability testing*" o "*Usability testing*", consisten en reclutar a una muestra representativa de usuarios que realiza tareas concretas utilizando el sistema (o el prototipo) y los evaluadores utilizan los resultados para ver cómo la interfaz de usuario da soporte a estos con sus tareas. Son considerados los métodos más efectivos [Tsai06] y disponen de un gran número de tipos.

Estos tres grupos son los que más presencia muestran en la literatura referente a la ingeniería de la usabilidad. Sin embargo, Ivory [Ivory03] propone otros dos grupos dentro de esta agrupación: el *modelado analítico*, donde el evaluador utiliza modelos de usuario e interfaz para predecir errores de usabilidad; y la *simulación*, donde el evaluador utiliza modelos de usuario e interfaz para imitar la interacción de un usuario y generar conclusiones.

2.1.4.2.2. FINALIDAD DE LA EVALUACIÓN

Numerosos autores [Nielsen94b, Bowman+02, Barnum10, Albert+13] utilizan como criterio de clasificación la finalidad de la evaluación del método. Existen principalmente dos tipos de finalidad:

- *Evaluación formativa*. Es un estudio observacional cuyo objetivo es *estudiar la interacción del usuario mediante la observación de usuarios representativos en escenarios basados en tareas para identificar problemas de usabilidad*. Es una evaluación empírica que también sirve para evaluar la capacidad del diseño de apoyar la exploración del usuario, el aprendizaje y la ejecución de tareas. Las evaluaciones formativas ofrecen resultados de carácter cualitativo: objetos de interfaz críticos, comentarios de usuarios, reacciones generales, etc. También ofrecen (aunque en menor medida) resultados cuantitativos: completitud de tareas, frecuencia de los errores, etc. Con estos resultados se pretende identificar cuáles son los problemas de usabilidad más significativos que hace que los usuarios no completen las tareas o sean más ineficientes.
- *Evaluación sumativa*. Es un estudio comparativo cuyo objetivo es *estudiar el grado en el que el producto evaluado logra unos objetivos.* Siendo estos objetivos predefinidos (p.ej. el usuario debe completar una tarea en menos de un minuto). Al igual que con la anterior, los usuarios representativos realizan tareas en escenarios de trabajo y los evaluadores recogen y calculan datos cuantitativos: tiempo de tarea, número de errores generados, etc.

Como muestra la explicación en el libro de Albert y Tullis [Albert+13], la evaluación formativa se asemeja a cómo un chef está constantemente probando su elaboración (punto de sal, pimienta, cocción…) hasta que la termina ya que está constantemente evaluando ciertos parámetros y los va ajustando. Por otro lado y con la misma analogía, la evaluación sumativa se asemeja a la evaluación realizada por un crítico culinario que prueba la elaboración del chef y la compara con otras

elaboraciones similares ofertadas en otros restaurantes o con la misma elaboración probada tiempo atrás.

2.1.4.2.3. LUGAR DE REALIZACIÓN

Este criterio clasifica en función de la localización donde se realizan las pruebas de evaluación de la usabilidad del producto. Se diferencian principalmente dos grupos.

- *Pruebas en laboratorio*. Las pruebas en laboratorio se realizan en un entorno y ambiente controlado. Dentro de este lugar, se utilizan dispositivos específicos y usuarios concretos si los requiere el método utilizado.
- *Pruebas en entornos reales*. Al contrario que la categoría anterior, en este tipo de métodos las tareas de evaluación de la aplicación se realizan en entornos reales. Esto quiere decir que se efectúan en el lugar de interacción habitual del usuario.

Ambos grupos presentan varias ventajas e inconvenientes. En primer lugar, los métodos realizados en laboratorio se llevan a cabo en un contexto altamente controlable [Zhang+05] y la captura de datos es sencilla [Nielsen+06]. Por el contrario, en los entornos reales la captura es más complicada y la mayoría de parámetros del contexto no son controlables, lo que impide que los investigadores se centren en fenómenos específicos de interés [Razak+10]. En un laboratorio, al poder controlar la mayoría de parámetros del contexto, todos los sujetos experimentan exactamente la misma experiencia [Razak+10]. Por lo que la comparación entre sujetos es más ajustada. Por el contrario en los entornos reales, aunque no podamos realizar una comparación tan ajustada, se ofrece una visión más profunda de los problemas durante el uso cotidiano de la tecnología, llegándose a producir fenómenos en situaciones reales no reproducibles en laboratorio y muy relevantes para la evaluación [Razak+10].

Viendo las principales ventajas e inconvenientes de ambos tipos, *las carencias de uno son las virtudes del otro*. Consecuentemente, miembros de la comunidad científica han optado por intentar unir

las pruebas en entornos reales con las de laboratorio, desplazando éste a entornos reales mediante laboratorios de usabilidad portátiles. Un laboratorio de usabilidad portátil puede ser montado en un periodo de tiempo muy corto [Rowley94]. Los principales problemas que esto acarrea son principalmente los relacionados con el uso del equipo de captura de datos, ya que es necesario tiempo tanto en la preparación de las pruebas como en la captura de los datos. Esto hace más difícil llevar a cabo observaciones de uso dentro de un tiempo realista y dispone de datos en los que el usuario no se comporta con naturalidad debido a la presencia de objetos de captura (p.ej., cámaras de vídeo) [Mackay95, Jordan+99].

2.1.4.2.4. GRADO DE AUTOMATIZACIÓN

Considerando como criterio el grado de automatización disponible para los métodos. Siguiendo el esquema de clasificación propuesto por Balbo [Balbo95], se reconocen cuatro principales categorías ligadas a la fase del método automatizada:

- *No automáticos*: Son métodos que son realizados manualmente por expertos en el área. Por lo tanto, no hay ningún tipo de automatización siendo el tipo de métodos que más tiempo requiere según esta clasificación.
- *Captura automática*: Son los métodos que utilizan algún tipo de herramienta de soporte para capturar información relevante sobre el usuario y el sistema (p.ej. teclas del teclado pulsadas, datos visuales, etc.)
- *Análisis automático*. En este grupo están aquellos métodos que tienen la capacidad de identificar problemas de usabilidad de un modo automático.
- *Crítica automática*. Estos métodos ofrecen además de identificar problemas, ofrece recomendaciones de mejora.

Por otro lado, la taxonomía más genérica es la clasificación en *métodos automáticos* (donde estarían los tres últimos grupos de Balbo) y *métodos manuales o no automáticos*.

El principal problema de los métodos automáticos es que necesitan un sistema implementado o prototipos preliminares funcionales para poder realizar su trabajo. Sin embargo, son de especial ayuda ya que se ejecutan tareas rápidamente y los resultados siempre provienen de los mismos parámetros, sin apreciaciones subjetivas. Por el contrario, los métodos manuales requieren recursos más costosos (tiempo, evaluadores, etc.) pero resultan más flexibles al poder evaluar con ellos aspectos no convencionales y específicos de un determinado sistema [Perallos07].

Los intentos más ambiciosos en la automatización de pruebas de usabilidad provienen de la comunidad de inteligencia artificial [Norman+06]. Sin embargo, se adoptan enfoques más modestos donde se focaliza en el registro, análisis e interpretación de los eventos de interfaz de usuarios reales que realizan bien las tareas simuladas en el laboratorio o las tareas reales en un entorno (análisis automático). Área donde se centra este trabajo junto con la captura automática.

2.1.4.2.5. PROTOTIPO UTILIZADO

En primer lugar debemos aclarar que dentro del área de la usabilidad *un prototipo es una representación limitada de un producto que permite estudiarlo y explorar su uso*. Mediante los hallazgos de este estudio, podremos generar un producto con mayor calidad. Pudiendo en este caso, evaluar la usabilidad sin necesidad de esperar a la implementación total del producto. Existen dos principales enfoques para la agrupación dentro de esta taxonomía.

Por un lado, se ofrece una agrupación en función del nivel de funcionalidad reproducida [Van89, Floria00].

- *Horizontal*: Se reproduce gran parte del aspecto visual del producto pero no hay una funcionalidad detallada.
- *Vertical*: Se reproduce el aspecto visual de sólo una parte del producto y la parte reproducida tendrá casi la totalidad de la funcionalidad a implementar.

Por otro lado, el grado de fidelidad o calidad del prototipo marcará la facilidad con la que un prototipo se distingue del producto final. A mayor fidelidad, mayor probabilidad de confundir el prototipo con el producto final [Walker+02]:

- *Alta fidelidad*: El prototipo será muy parecido al producto final ya terminado. Las pruebas que se realizan con este tipo de prototipos suelen dirigirse a un estudio más detallado y de aspecto visual.
- *Baja fidelidad*: El aspecto del prototipo será muy distinto del producto final. Frecuentemente se utiliza en las primeras etapas de desarrollo para estudiar la arquitectura de la información o el diseño de la interacción a alto nivel [Landay+01], concretamente el uso del *prototipado de baja fidelidad en papel* [Snyder03], que consiste en reproducir los aspectos básicos de la interfaz en un papel.

Estudios previos han abordado las diferencias potenciales entre baja y alta fidelidad sin manipular el medio de los prototipos donde no se encontraron diferencias significativas entre ambos tipos de prototipos en el número de problemas de usabilidad encontrados [Virzi+96, Hong+01b]. La principal diferencia radica en el coste de la elaboración de un prototipo, en el caso de la alta fidelidad el coste es elevado debido a su nivel de detalle [Rudd+96]. Por el contrario, la realización de prototipos de baja fidelidad es de elaboración muy rápida y económica, por lo que ayudan en las primeras fases del ciclo de vida del desarrollo software. Otra diferencia importante es que las pruebas realizadas con prototipos de alta fidelidad pueden verse suspendidas debido a errores del propio prototipo, cuyos cambios llevan mucho trabajo [Rettig94].

2.1.4.2.6. PRESENCIA DE MODERADOR

Este criterio de agrupación es utilizado en varios trabajos [Dumas+08, Barnum10], concretamente para el estudio de pruebas de usabilidad remotas, por lo que solo es aplicable a este tipo de evaluaciones. Se basa en dos principales grupos.

- *Pruebas moderadas o síncronas*. Este tipo de evaluaciones disponen de un evaluador que actúa de moderador que interactúa en el momento de la realización de las pruebas con los participantes. La diferencia que presenta respecto a las pruebas de laboratorio es la localización de los participantes en las pruebas.
- *Pruebas no moderadas o asíncronas*. Este tipo de evaluaciones no disponen de un moderador. Por ello también son denominadas como pruebas asíncronas al ser realizadas cuando el participante lo desea y no requerir de la presencia de nadie en el momento de las pruebas.

La principal ventaja dentro de los métodos remotos es que el evaluador no tiene que viajar a los entornos donde se realizan las pruebas, abaratando significativamente el coste de las pruebas. Sin embargo, ambos tipos muestran varios inconvenientes. Las pruebas moderadas requieren un evaluador experimentado para llevar a cabo correctamente las mismas. Esto es debido a los problemas relacionados con el uso de algún sistema de telecomunicación para la interacción con el usuario: la comunicación no es tan fluida como en un modo presencial y es común experimentar fallos técnicos. Por otro lado, las pruebas no moderadas posibilitan un alcance de un gran número de usuarios en poco tiempo, al no requerir de un moderador. La principal desventaja es que aunque los resultados no presentan sesgos, no se identifican los motivos de las acciones de los usuarios al no disponer de comunicación directa con ellos.

2.1.4.2.7. ESFUERZO HUMANO REQUERIDO

Otro criterio expuesto en la taxonomía ofrecida por Balbo [Balbo95] y tomada en cuenta por Ivory [Ivory03] es el esfuerzo humano requerido. Indica el grado de esfuerzo humano que requiere un método para llevarse a cabo.

- *Esfuerzo mínimo*. No requieren uso de interfaz gráfica ni uso de ningún modelo de la misma.

- *Desarrollo de modelos*. Necesitan que el evaluador utilice o desarrolle un modelo de interfaz de usuario para ejecutar estos métodos.
- *Uso informal*. Es necesario, además de una interfaz, la ejecución de tareas elegidas libremente por el evaluador o por el usuario.
- *Uso formal*. Es necesario, además de una interfaz, la ejecución de tareas que han sido definidas en el método en sí.

Habiendo expuesto las diferentes taxonomías utilizadas para la clasificación de los métodos de evaluación de la usabilidad concluimos la introducción a la usabilidad desde un punto de vista general y damos comienzo al estudio de esta disciplina dentro del área de los dispositivos móviles.

2.2. USABILIDAD EN LOS DISPOSITIVOS MÓVILES

Para tener una visión clara de las limitaciones que tienen este tipo de dispositivos debemos marcar los límites entre lo que es un teléfono móvil convencional, un teléfono inteligente o Smartphone y teléfonos de pantalla completa.

- Los *teléfonos móviles convencionales*, son terminales que permiten realizar llamadas de teléfono, disponen de una pantalla pequeña y permiten realizar una interacción muy deficiente con servicios de Internet.
- En segundo lugar, disponemos del *teléfono inteligente o Smartphone*. Desde un punto de vista superficial, el diccionario de Cambridge[12] define el concepto de teléfono inteligente como un teléfono móvil que puede ser utilizado como un pequeño ordenador y dispone de conexión a Internet. Coincidiendo con esta definición, diversos trabajos

[12]http://dictionary.cambridge.org/dictionary/british/smartphone

como [Raento+09] y [Boulos+11] lo definen como una extensión de la funcionalidad típica de un terminal móvil con avanzadas capacidades de computación y comunicación. Además, amplían la definición añadiendo que disponen de sensores avanzados y pantallas con buena definición. Revisando varias definiciones, definimos un teléfono inteligente presentando las tres características mostradas por Sanz [Sanz12]: funcionalidad avanzada (realización de tareas complejas como gestión de correo electrónico o reproducción de contenido multimedia), hardware especializado (mediante una arquitectura con sensores avanzados como acelerómetro, giroscópico o GPS) y alta capacidad de cómputo.

> *Un teléfono inteligente o Smartphone es un teléfono móvil que, gracias a una alta capacidad de cómputo y al hardware especializado que dispone, es capaz de realizar funciones avanzadas en movilidad.*

- Finalmente, los *teléfonos de pantalla completa* son teléfonos inteligentes que disponen de una pantalla de alta resolución y táctil que permite una interacción sencilla mediante manipulación directa de los dedos con la pantalla y gestos intuitivos. Aunque comúnmente este tipo de dispositivos no muestren una gran diferencia con los teléfonos inteligentes, hemos decidido remarcar la diferencia ya que el disponer de este tipo de interacción mejora de un modo considerable la experiencia en el uso de las aplicaciones [Nielsen+13].

Habiendo definido los diferentes tipos de dispositivo móvil procedemos a mencionar las limitaciones generadas por las características de los mismos.

2.2.1. LIMITACIONES DE LAS APLICACIONES MÓVILES

Como ya hemos avanzado, las pruebas de usabilidad de aplicaciones de software desarrollado para dispositivos móviles es una emergente área de investigación que se enfrenta a una

variedad de problemas debido a las características únicas de los dispositivos móviles.

Revisando los diferentes trabajos de la literatura, hemos clasificado las limitaciones encontradas en dos principales grupos mostrados en la tabla 2.3.

Relacionadas con el dispositivo	*Relacionadas con la movilidad y su contexto*
Heterogeneidad en los dispositivos	*Conectividad extremadamente variable*
Tamaño de pantalla reducido	*Probabilidad elevada de interrupción*
Interfaces de entrada reducidas y complicadas	*Tareas con tiempo muy limitado*
	Impacto de los elementos del contexto muy significativo

Tabla 2.3 Limitaciones de las aplicaciones móviles desde el punto de vista de la usabilidad

- *Limitaciones relacionadas con el dispositivo*. Como hemos revisado al comienzo de esta sección, al hablar de los tipos de dispositivos móviles hay una gran diversidad de dispositivos, tanto en software como en hardware y ergonomía. Debido a esta *heterogeneidad en los dispositivos*, resulta difícil tanto generalizar hallazgos como proporcionar una evaluación de una misma aplicación en diferentes terminales ya que al disponer de terminales con diferentes capacidades, la usabilidad se ve comprometida [Pendell+12].

  Dentro de las características que más agravan la usabilidad de las aplicaciones móviles es el *tamaño reducido de la pantalla de los dispositivos*, numerosos trabajos han demostrado que la usabilidad se ve afectada por esta característica [Bickmore+97, Jones+99, Kim+01, Tsiaousis+10, Hong+11] ya que aumenta la carga cognitiva del usuario al disponer de menos superficie de lectura [Nielsen+13] y el usuario debe realizar acciones adicionales como el escalado en la navegación web [Al-Ismail+14].

  Al igual que la pantalla dificulta la visualización de datos, también existe *dificultad con las interfaces de entrada*. Por un

lado, se sustituye el ratón de escritorio por una interfaz más intuitiva pero menos precisa, donde no disponemos de la visión del puntero en todo momento y se produce fatiga en el brazo si se hace un uso prolongado [Nielsen+13]. Por otro lado, la introducción de texto alfanumérico dentro de las pantallas táctiles es dificultosa y lenta [Longoria01, Shrestha07]. Como se dispone de poca superficie para presentar botones, se debe cambiar el formato de teclado si queremos introducir números u otros caracteres especiales [Hong+11]. Aunque existen técnicas ya patentadas y explotadas comercialmente como Swipe [Westerman+11], sigue suponiendo una fuerte limitación.

- *Limitaciones relacionadas con la movilidad y su contexto*. Dentro de este grupo están las limitaciones que hacen referencia a las posibles limitaciones que se producen al ser un objeto que es utilizado en casi cualquier lugar. Primeramente, debido a la posibilidad de que puede ser utilizado en constante movimiento (p.ej., en un medio de transporte) *la conectividad es extremadamente variable*, pudiendo en ocasiones incluso perderse. Como se demuestra en el estudio de Kristjánsdóttir et al. [Kristjánsdóttir+11], una mala conectividad puede desencadenar frustración en el usuario. Además, la condición de movilidad incide directamente sobre las características de las tareas que se realizan con este tipo de dispositivos. En primer lugar, dispone de una *probabilidad muy elevada de que la tarea sea interrumpida* (p.ej., por una llamada de teléfono) [Brown+10]. En segundo lugar, las tareas que se realizan en estas condiciones son *tareas de duración corta cuya limitación puede inferir negativamente en la eficiencia* de la tarea [Kaasalainen10]. Finalmente, existen diferentes elementos que se sitúan en el lugar de la interacción (personas, objetos, temperatura, etc.). Estos *elementos ejercen un impacto muy significativo* sobre la usabilidad de la aplicación [Ryan+05, Tsiaousis+08, Wigelius+09].

El pequeño tamaño de los dispositivos móviles y los entornos donde se utilizan han limitado las formas en que los usuarios pueden interactuar con ellos. Habiendo estudiado estas limitaciones desde el punto de vista de la usabilidad, a continuación revisaremos cómo son estudiadas y evaluadas.

2.2.2. EVALUACIÓN DE LA USABILIDAD DE APLICACIONES MÓVILES

La usabilidad en el área de las aplicaciones móviles muestra un mayor impacto que en otro tipo de áreas [Coursaris+11]. Sin embargo, a pesar de existir un volumen considerable de investigación en la usabilidad desde una perspectiva general hay pocos trabajos desarrollados centrados en la usabilidad de esta área [Nayebi+12], existiendo una necesidad de estudiar y mejorar los problemas y limitaciones asociadas a este tipo de aplicaciones que hemos mencionado anteriormente (ver apartado 2.2.1).

Con este fin, las dos principales plataformas de aplicaciones móviles ofrecen guías para el diseño y desarrollo de las aplicaciones móviles de su plataforma (iOS Human Interface Guidelines[13] y Android Developers[14]). Dichas guías son de gran utilidad, ya que evitan varios problemas pero no sirven para una evaluación de usabilidad fuera de las técnicas de inspección (ver apartado 2.1.4.2.1). Como ya hemos examinado en el apartado 2.1.4, existen numerosos métodos para realizar una evaluación de usabilidad. Nos disponemos a resolver *cuáles son los más utilizados e idóneos para este tipo de aplicaciones y cómo han evolucionado en el área de las aplicaciones móviles*.

Afortunadamente, la revisión de la literatura realizada por Coursaris et al. [Coursaris+11] realiza una revisión en la que contrasta los resultados obtenidos con otra realizada en 2006

[13]https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG
[14]https://developer.android.com/design/index.html

[Coursaris+06] con los nuevos resultados para deducir la evolución del campo. En dicho trabajo, formado por la comparación de más de 100 estudios empíricos de usabilidad móvil, concluyeron varios hallazgos.

Primeramente, descubrieron una *carencia en la existencia de marcos de evaluación de la usabilidad de aplicaciones móviles sólidos, habiendo detectado una falta calificada por los autores como crítica*. A pesar de ello, esta carencia no fue resuelta al pasar las metodologías a segundo plano, como afirmó Kjeldskov et al. [Kjeldskov+12] en 2012.

En contraste con el contexto típico de uso general con un ordenador, los contextos en los que se interactúa con aplicaciones móviles son heterogéneos y dinámicos [Isomäki+11]. Desafortunadamente, uno de los principales problemas encontrados es que los atributos de usabilidad de una aplicación móvil a menudo son estudiados como medidas de usabilidad en las aplicaciones de escritorio [Alshehri+12]. Esto *revela una tendencia hacia los métodos de evaluación que se centran principalmente en el dispositivo en vez de otros aspectos del contexto* [Avouris+08, Billi+10]. Si bien sí se detectó un aumento en el número de estudios que tienen en cuenta el entorno en la evaluación, siguieron afirmando que *existe una falta de investigación empírica sobre la relevancia de las características tanto del usuario como del entorno que rodea la interacción* (incluyendo en este las tecnologías asociadas al dispositivo móvil). Aunque en su comparativa detectaron un incremento de los trabajos donde se estudian ciertas características del entorno, consideraron que *el estudio de las características del entorno es el área con mayor potencial para la futura investigación dentro de la evaluación de la usabilidad móvil*. Finalmente, centrándonos en el tipo de método utilizado, contrastaron que *predominaban el uso los estudios que utilizan la evaluación en laboratorios y/o estudios de campo*, siendo utilizados en el 78% de los estudios analizados. Más adelante y del mismo modo, Harrison et al. [Harrison+13] plantearon investigar cómo se llevan a cabo actualmente las evaluaciones de este tipo de aplicaciones. Revelaron que aproximadamente el 59% de los

estudios realizan una evaluación de la usabilidad mediante experimentos controlados, lo cual *indica un fuerte interés por este tipo de métodos*. Por otro lado, destacan los estudios a través de estudios de campo, siendo estos el 27% de los trabajos analizados. Con ello, al igual que el trabajo de Coursaris et al. [Coursaris+11], concluyeron que *los dos métodos más utilizados en la evaluación de la usabilidad de aplicaciones móviles son la evaluación en laboratorios y los estudios de campo*, con un 90% de predominio (163 trabajos analizados donde 141 utilizan uno de los dos tipos de método mencionados).

Viendo los dos métodos más utilizados, *detectamos un gran desacuerdo en la elección del tipo de método en función de la localización*. Dicho desacuerdo ha sido explícitamente estudiado y detallado por los trabajos realizados por Jesper Kjeldskov, desde su trabajo de 2004 [Kjeldskov+04] donde formula una polémica pregunta: "*Is it worth the hassle?*". Con esta pregunta, los autores cuestionaban si merece la pena el valor añadido de realizar un estudio de usabilidad de aplicaciones móviles en entornos reales. En este estudio se dispuso a comparar el resultado de la evaluación de la usabilidad de un sistema móvil en un entorno de laboratorio y en el entorno real en relación con problemas de usabilidad identificados y el tiempo dedicado. En dicho trabajo afirmaban que *las técnicas de recolección de datos en entornos reales son extremadamente difíciles y que se debía dedicar esfuerzo a mejorar y resolver dicha dificultad*. Este trabajo impactó en la comunidad científica originando numerosos estudios (191 citas al artículo desde su publicación hasta febrero de 2014 de acuerdo con Google Scholar[15]). Recapitulando estas menciones, Kjeldskov et al. [Kjeldskov+14] las analiza para concluir el estado de la investigación en esta área.

[15]Google Scholar es un buscador que está centrado en el mundo académico y especializado en literatura científica.

Bertini et al. [Bertini+06] afirmaron que hay una clara *necesidad de que el método de evaluación de usabilidad móvil debe incluir a los usuarios reales, contextos reales o simulados y tareas reales con dispositivos reales*. El principal problema para llevar a cabo dicha propuesta radica en la *fiabilidad de los datos capturados* en el desarrollo de este tipo de evaluación [Alshehri+12].

Mediante el análisis de Kjeldskov et al. [Kjeldskov+14] *se percibe un fuerte interés en encontrar métodos que permitan aumentar el realismo en métodos de evaluación en laboratorio incluyendo simulaciones de contexto*, como la reproducción de ruido y presencia de gente [Kondratova+06, Leitner+07] o la proyección de obstáculos en el suelo para simular la acción de caminar [Lumsden+07].

Centrados en la captura de los datos en entornos reales, se realiza mediante cámaras, grabaciones de audio, trazas de datos, cuestionarios y entrevistas [Oulasvirta+05, Dearman+05] aunque también *se detectan varios trabajos que capturan información de contexto a través de los sensores* de los propios dispositivos [Petersen+10, Jambon+09]. Relacionado con la captura se argumenta, al igual que en su revisión de 2012 [Kjeldskov+12], que *la complejidad y riqueza del mundo real no es capturada ni reproducida en un laboratorio* [Rogers+07b].

Además, la comunidad científica *no considera fundamental la replicación de los experimentos dentro del área de la evaluación de la usabilidad, argumentando que los factores sociales, humanos y tecnológicos poseen excesiva variabilidad como para ser reproducibles* [Brown+11].

Si bien la importancia de los estudios de campo es bastante evidente (varios trabajos demuestran que se identifican problemas adicionales en entornos reales [Duh+06, Nielsen+06]) sigue habiendo varios problemas sin resolver. Por un lado, *la falta de control* (p.ej. el clima) [Kellar+04], posiblemente solo adquirida en un laboratorio. Por otro lado, *el coste en términos de tiempo y esfuerzo necesario para realizar las pruebas en entornos reales* [Oulasvirta12, Rogers+07b]. Finalmente, varias revisiones de la literatura

[Kjeldskov+04] y trabajos notifican problemas relacionados con la privacidad del usuario dentro de las evaluaciones de la usabilidad debido a la *imposibilidad de observar a los usuarios interactuando con la aplicación móvil en ciertos contextos donde requieren privacidad* (p.ej. en su propia casa) [deSa+08].

Habiendo examinado los principales trabajos relacionados con el área, nos disponemos a resumir las limitaciones que han sido encontradas en la literatura.

2.2.3. LIMITACIONES DE LA EVALUACIÓN DE LA USABILIDAD

Dentro de las limitaciones expuestas referentes a las aplicaciones móviles, destaca la indudable importancia del entorno dentro de este tipo de software, ya que es dinámico y heterogéneo (por ello, las aplicaciones móviles deben cubrir la mayor posibilidad de situaciones [Kim+02]). Dicha importancia no pasa desapercibida dentro de las investigaciones en el campo de la usabilidad, donde se concluye lo siguiente:

> *Hay una fuerte necesidad en el estudio de las características del entorno dentro del estudio de la usabilidad de aplicaciones móviles.*

Centrados en las evaluaciones realizadas en laboratorios se identifican las siguientes:

- De cara a los métodos en laboratorio, la principal limitación que presenta es la *baja calidad de la simulación del entorno*, ya que presentan simulaciones que muestran pocas similitudes con la realidad. La calidad de la simulación puede ser incrementada aunque su coste es muy elevado.

Centrados en las evaluaciones realizadas en entornos reales se identifican las siguientes:

- Se ha identificado que *el coste en cuanto a tiempo y esfuerzo de estas evaluaciones es muy elevado*. Para la reducción del mismo existe cierta tendencia a automatizar fases del proceso de evaluación de la usabilidad.

- Una limitación derivada de la propia naturaleza de este tipo de evaluaciones es la *falta de control de los factores de contexto*, ya que al ser pruebas realizadas en entornos reales el control es muy complicado e incluso imposible (p.ej. factores climatológicos).
- También se han identificado limitaciones en la captura de las propiedades del contexto. *La fiabilidad de los datos capturados es baja* ya que dependiendo del método o herramienta de captura elegido se muestran sesgos relacionados con la alteración del entorno real (p.ej. añadiendo una cámara de grabación altera la ergonomía del dispositivo, un usuario siendo estudiado por un observador humano mientras interactúa con la aplicación no muestra el mismo comportamiento [Cassady+02]).
- Además, *la privacidad de los usuarios se ve amenazada* en diversos métodos si son realizados en entornos reales críticos como en la propia casa del usuario que utiliza el sistema.

Habiendo identificado las diferentes limitaciones resumidas en la tabla 2.4, y en relación con la limitación de carácter general, a continuación nos disponemos a estudiar más en detalle la importancia del contexto dentro de las aplicaciones móviles.

General	*Entornos de laboratorio*	*Entornos reales*
Hay una fuerte necesidad en el estudio de las características del entorno dentro del estudio de la usabilidad de aplicaciones móviles	*Baja calidad de la simulación del entorno*	*El coste en cuanto a tiempo y esfuerzo de estas evaluaciones es muy elevado* *Falta de control de los factores de contexto* *La fiabilidad de los datos capturados es baja* *La privacidad de los usuarios se ve amenazada*

Tabla 2.4 Limitaciones de la evaluación de la usabilidad de aplicaciones móviles identificadas

2.2.4. IMPORTANCIA DEL CONTEXTO EN LAS APLICACIONES MÓVILES

Desde que hemos definido formalmente la usabilidad (ver apartado 2.1.1), hemos sido conscientes de que el contexto toma un fuerte papel dentro de esta área. Además de los trabajos expuestos en las limitaciones de los terminales móviles relacionadas con la movilidad y su contexto (ver apartado 2.2.1), numerosos trabajos [Roto+11, Avouris+08, Isomäki+11] demuestran y afirman que el contexto es parte de la experiencia subjetiva del usuario, y por lo tanto, un fuerte elemento que influye directamente en la usabilidad de las aplicaciones móviles. Debido a esto, la investigación centrada en la evaluación de la usabilidad de aplicaciones móviles está cada vez más centrada en las variables contextuales que afectan a la interacción desde una perspectiva holística [Bernhaupt+08]. Debemos estudiar qué elementos conforman el contexto y cómo es definido.

La comunidad científica ha dedicado mucho esfuerzo en acotar y definir el contexto en numerosas disciplinas [Bradley+05]. Como afirma Dey en su trabajo ampliamente considerado [Dey01], el contexto es una palabra que se entiende fácilmente pero es difícil de definir con claridad. En dicho trabajo se presenta la siguiente definición:

> *El contexto es cualquier información que se pueda utilizar para caracterizar la situación de una entidad. Una entidad es una persona, lugar u objeto que se considera relevante para la interacción entre un usuario y una aplicación, incluyendo el usuario y las propias aplicaciones.*

Con la finalidad de ofrecer una aproximación multidisciplinar, Bradley y Dunlop [Bradley+05] proponen un modelo dinámico de contexto sobre la base de las definiciones tomadas y analizadas de la lingüística, la informática y la psicología. Según ellos, el contexto que rodea al usuario está formado por la tarea, el entorno físico, social, temporal y cognitivo.

Enfocando la atención en los dispositivos móviles, el concepto de contexto surgió del trabajo de varios estudiosos como Maguire [Maguire01] o Bevan y Macleod [Bevan+94], que trataban de identificar las variables adicionales que pueden afectar a la usabilidad. En estos trabajos se identifican cuatro principales elementos: usuarios, tareas, equipamiento y entorno. Schilit y Theimer [Schilit+94], definen el contexto en función de la ubicación del usuario, quién está con el usuario y los recursos disponibles. También afirmaron que el elemento más explicativo del contexto era la ubicación física. Después, Brown et al. [Brown+97] añaden a la ubicación más atributos tales como el tiempo meteorológico, para lograr una definición más precisa. En 2001, Dey propuso la definición explicada anteriormente [Dey01], la cual ha sido ampliamente estudiada.

Por otro lado, la comunidad de estándares internacionales, como ya se ha detectado, también han reconocido el papel de contexto dentro de la usabilidad. Según el estándar ISO 9241-11:1998 [ISO98] *el contexto en uso se compone de los usuarios, tareas y equipos (hardware, software y materiales), y los entornos físicos y sociales en los que se utiliza un producto*. Esta definición de contexto es utilizada por posteriores estándares como [ISO04] y [NIST07].

Centrándonos concretamente en la usabilidad de aplicaciones móviles, Roto [Roto06] identifica cuatro dimensiones del contexto móvil que afecta a la experiencia del usuario: el contexto físico (p.ej., condiciones meteorológicas, luz, ruido,…), el contexto social (p.ej., amigos presentes, familiares,…), el contexto temporal (atributos que afecta el tiempo: viaje en autobús) y las tareas (p.ej., objetivo general del uso de la aplicación móvil). Otra representación es la propuesta por Kim et al. [Kim+02], que divide el contexto en dos grupos. El contexto personal, dividido a su vez en interno (p.ej., motivación del usuario, sentimientos,…) y externo (p.ej., manos o piernas), representa los aspectos intrínsecos del usuario. El contexto ambiental, dividido a su vez en físico (p.ej., ruido, luminosidad,…) y social (p.ej., personas que rodean al usuario), muestra los elementos que rodean al usuario. Un enfoque de definición del contexto más centrado en el dispositivo

y entornos laborales es el propuesto por Zheng y Yuan [Zheng+06], donde presentan cuatro elementos: los trabajadores, las tareas móviles, el contexto móvil (subdividido en localización y estructura temporal) y tecnología móvil. Savio y Braiterman [Savio+07] sugirieron elementos como la cultura, la atención y el dispositivo. Con el mismo enfoque, Kankainen [Kankainen02] añadió a la definición del contexto la composición emocional del usuario y la comunidad que le rodea.

Durante las últimas décadas la definición de los diferentes factores que componen el contexto ha sido ampliamente estudiada. A pesar de existir numerosos trabajos que describen los elementos del contexto de un modo muy detallado, no hay un marco común de referencia ampliamente utilizado. Sin embargo, sí hemos identificado varios factores comunes, como los expuestos por Coursaris y Kim [Coursaris+11]: usuario, dispositivo, entorno y tarea. A los cuales añadimos el factor aplicación, como en Biel et al. [Biel+10]. Como consecuencia, identificamos cinco principales factores que modelan el contexto para aplicaciones móviles, cuyo detalle será presentado en la definición del contexto de la metodología desarrollada (ver apartado 3.2.1): *aplicación, usuario, dispositivo, entorno y tarea.*

2.3. METODOLOGÍAS DE EVALUACIÓN

Ya disponemos tanto de una visión general de la usabilidad como una visión específica dentro de las aplicaciones móviles. Con esta base nos disponemos a analizar ciertas características de los trabajos existentes más relevantes que nos permita estudiar si dichos trabajos abordan las limitaciones presentadas.

2.3.1. CRITERIOS DE ANÁLISIS

En primer lugar, presentamos los criterios en base a los cuales vamos a realizar dicho análisis.

- *Tipo de aplicación*. Mediante este criterio disponemos a visualizar si dentro de las aplicaciones evaluadas, el trabajo

analizado evalúa aplicaciones *de tipo web*, *aplicaciones nativas* o *ambas*. Cabe añadir que una aplicación nativa es una aplicación que reside en el dispositivo con el que se interactúa, desarrollada específicamente para una plataforma, siendo su acceso mediante su invocación dentro del terminal móvil. Las aplicaciones web son sitios web que, en muchos sentidos, se sienten como aplicaciones nativas, pero no se implementan como tal ya que se interactúa con ellas mediante un navegador web y son accedidas mediante el acceso a una URL desde el navegador.

- *Método de captura*. Mediante este criterio nos disponemos a visualizar cómo se generan los datos que sirven de base para el análisis de las pruebas realizadas. Dentro de esta captura destacan seis métodos. Mediante *cuestionarios* (C), como el método con igual nombre (ver método 9 del apartado 2.1.4.1.3). La *generación de trazas* (TZ) captura los eventos y los registra para su posterior interpretación, como en el método de análisis de trazas (ver método 18 del apartado 2.1.4.1.3). La *grabación del usuario en vídeo* (GUV) o la *grabación del usuario en audio* (GUA) que sirve para un posterior análisis como en el método de análisis retrospectivo (ver método 16 del apartado 2.1.4.1.3) y la *grabación de la pantalla del usuario* (GP), siendo una extensión dinámica del método Screen Snapshots (ver método 11 del apartado 2.1.4.1.3).
- *Equipamiento necesario*. Para la realización de la evaluación estudiamos si es necesario algún equipamiento adicional que no sea ni el terminal móvil ni un servidor donde almacenar la información. Dentro de esta característica predominan dos principales elementos: una *aplicación móvil de captura* (App) y una *cámara web* (Cam).
- *Grado de automatización*. Indica si alguna fase se realiza de un modo automático (ver apartado 2.1.4.2.4).

- *Lugar de realización*. Indica principalmente si las pruebas se realizan en laboratorio o en entornos reales (ver apartado 2.1.4.2.3).
- *Finalidad de la evaluación*. Nos indica si el trabajo analizado soporta un objetivo de tipo *formativo*, *sumativo* o *ambos*.
- *Conocimientos adicionales*. Estudia si el evaluador que realiza las pruebas de usabilidad debe tener conocimientos adicionales fuera de los conocimientos de usabilidad para realizar la evaluación (p.ej., conocimientos de desarrollo para integrar algún tipo de herramienta en la aplicación).
- *Tiempo requerido*. Estudia la cantidad de tiempo requerida para realizar la totalidad de la evaluación de usabilidad respecto a la totalidad de los trabajos analizados. Hemos realizado una escala de tres niveles. *Alto*, cuando se requiere una cantidad de tiempo igual o similar al trabajo que más tiempo demanda. *Bajo*, cuando el tiempo requerido es similar al trabajo que menos tiempo demanda. *Medio*, cuando no se encuentra en ninguno de los anteriores.
- *Fiabilidad de los datos*. Mediante esta característica analizamos si los datos extraídos tanto de la captura como de su análisis son objetivos y no muestran desviaciones. Para ello hemos realizado una escala de dos niveles. Nivel *alto* implica que no existe ninguna desviación significativa en los datos obtenidos. Nivel *bajo*, indica alguna desviación significativa, como la obtenida por la subjetividad añadida por la interpretación de los datos capturados al realizar el análisis de grabaciones por parte del evaluador.
- *Grado de privacidad*. Estudia si mediante el método de captura propuesto por el trabajo la privacidad de los usuarios se ve amenazada. *Alta*, cuando no hay ningún tipo de amenaza. *Media*, cuando la privacidad del usuario se ve sensiblemente amenazada (p.ej. se obtienen grabaciones de audio del usuario). *Baja*, cuando la privacidad del usuario

se encuentra totalmente amenazada (p.ej. se obtienen grabaciones de audio y vídeo del usuario).

- *Modelo de contexto*. Mediante esta característica se estudia los diferentes elementos del contexto que utiliza el trabajo analizado. Dentro de estos elementos se distingue seis: *usuario* (USR), *dispositivo* (DIS), *aplicación* (APP), *equipamiento* (EQU), *tareas* (TAR) y *entorno* (ENT).
- *Uso de experiencia previa*. Se estudia si se hace uso de conocimiento generado con anterioridad para agilizar el proceso de evaluación.

2.3.2. MÉTODOS DE EVALUACIÓN DE USABILIDAD

Dentro de los trabajos encontrados que presentan algún tipo de metodología de evaluación de usabilidad las clasificamos en comerciales, ya que proveen una plataforma con la que los desarrolladores de software realizan un estudio de usabilidad para aplicaciones reales; y plataformas del ámbito científico, cuyo desarrollo es puramente académico y del sector de la investigación.

2.3.2.1. SOLUCIONES COMERCIALES

En este apartado describiremos las plataformas comerciales de evaluación de usabilidad remota más relevantes.

- *Loop11*[16]. Es una plataforma de evaluación de usabilidad remota que permite crear pruebas de usabilidad en la que usuarios remotos realizan tareas concretas en sitios web. Se dota al evaluador de la posibilidad de crear un objetivo para el usuario en cada tarea y de mezclar preguntas con las tareas en el orden que desee. El evaluador realiza una configuración de las pruebas y las lanza. El modo de reclutamiento es sencillo ya que solo difundiendo un

[16]http://www.loop11.com

enlace, potenciales usuarios pueden realizar las pruebas con cualquier dispositivo, incluyendo terminales móviles. Por ello, permite realizar pruebas de usuario paralelas con un gran volumen de los mismos. De cara a los resultados, ofrece resultados que soportan tanto el enfoque sumativo (completitud de tarea, tiempo de tarea) como el formativo (lista y número de páginas por las que ha pasado el usuario para completar la tarea). La principal limitación es que no se sabe si realmente las tareas se realizaron correctamente ya que el usuario que realiza la prueba es quien decide si la tarea se ha completado satisfactoriamente. Esto posibilita la situación en la que un usuario haya realizado mal una tarea y notifique una completitud positiva de la misma. Otra limitación es que el tiempo de respuesta de la aplicación que se evalúa se ve disminuido considerablemente por la penalización añadida a los tiempos de carga, propiciando una penalización en la duración de la tarea. Las pruebas deben ser cuidadosamente diseñadas e implementadas ya que la monitorización se realiza mediante la ejecución de un código propietario de JavaScript[17].

- *TryMyUI*[18]. Es otra plataforma comercial que ofrece evaluación remota de sitio web para móviles y aplicaciones nativas Android. Para evitar las restricciones de Apple en aplicaciones de grabación de otras aplicaciones, disponen de un kit de desarrollo de software que pueden agregar en el código de la aplicación a evaluar para permitir la captura y su análisis. Al igual que el anterior dispone de una fase de configuración. En ella se configuran las pruebas a realizar y se especifica el entorno donde deben de estar realizadas (p.ej., se le ha invitado a una fiesta muy elegante y está buscando zapatos negros volviendo a casa en el tren) para

[17]JavaScript es un lenguaje de programación interpretado orientado a objetos, débilmente tipado y dinámico que es utilizado principalmente para la mejora de la interfaz de usuario y páginas dinámicas en entorno web.
[18]http://trymyui.com

que posteriormente las hagan los usuarios y se generen los vídeos correspondientes. Dichos vídeos tienen una duración máxima de 20 minutos por usuario. La principal ventaja es que además de grabar los gestos e interacciones del usuario con la interfaz, graban los comentarios que realizan los usuarios. Con ello se dispone de los motivos por los cuales se realizan las acciones. Sin embargo, el principal problema es el tiempo de análisis ya que para disponer de los resultados se deben analizar la totalidad de los vídeos generados. Por lo que si el número de usuarios es elevado, el tiempo de análisis aumentará considerablemente.

- *OpenHallway*[19]. Esta plataforma permite crear un escenario de pruebas y enviar por correo electrónico dicho escenario a potenciales usuarios que considere el evaluador. Este sistema está completamente basado en la web y permite que los usuarios realicen las pruebas mediante cualquier equipo con una versión de Java mayor que 1.5. Mediante su herramienta de grabación de pantalla y captura de voz los evaluadores pueden analizar la interacción del usuario con el sistema que queda registrada mediante un vídeo, además permite realizar cuestionarios. El número de usuarios que realicen las tareas es ilimitado, sin embargo, hay un límite de almacenamiento por usuario (20 minutos). Las ventajas e inconvenientes que muestra esta plataforma son las mismas que la anterior. Sin embargo, no dispone de un enfoque de evaluación de aplicaciones móviles nativas.
- *Userlytics*[20]. Es otra plataforma cuyo funcionamiento es idéntico al de las anteriormente explicadas. El evaluador crea un entorno de pruebas mediante la especificación de los datos demográficos de los usuarios objetivo y lanza las pruebas. Posteriormente se permite añadir un cuestionario

[19]http://openhallway.com
[20]http://www.userlytics.com

de hasta 100 preguntas. La principal ventaja frente a otras plataformas es que permite hacer uso de plantillas de tarea predefinidas para configurar las pruebas. Como resultado se obtienen los vídeos del usuario grabados mediante cámara web y las capturas de pantalla del dispositivo. Desafortunadamente, el análisis no es automático y debe ser realizado por un evaluador, lo que implica que también requiere mucho tiempo para el análisis de todas las grabaciones.

- *Userzoom*[21]. Ésta es otra compañía de pruebas de usabilidad orientada a la prueba de aplicaciones web y móviles. También registran la navegación del usuario y graba en vídeo a los usuarios realizando las pruebas. Para ello utiliza su herramienta UserZoom Recorder, con la que también permite realizar cuestionarios. Una de las ventajas que le hace destacar a esta plataforma es la variedad de resultados que ofrece: métricas cuantitativas (ratios de éxito, eficiencia, eficacia y satisfacción) y vídeos de los usuarios junto con la captura de pantalla. Sin embargo, sigue necesitando un análisis manual de todos los vídeos, por lo que consideramos que siguen requiriendo mucho tiempo.

En las principales plataformas comerciales explicadas, al igual que en muchas otras (*WhatUsersDo*[22], *Validately*[23], *YouEye*[24], etc.), identificamos una metodología común: definición de tareas, ejecución de las tareas por parte de usuarios clasificados en base a datos demográficos, y un posterior análisis de las grabaciones (en su mayoría vídeos de la interacción) para finalmente crear los resultados relevantes para el estudio tanto sumativo como formativo.

[21]http://www.userzoom.com
[22]http://whatusersdo.com
[23]http://validately.com
[24]https://www.youeye.com

Como apreciamos en el resumen de la tabla 2.5, hemos identificado tres limitaciones que afectan mayoritariamente a estas aplicaciones. En primer lugar, el análisis requiere mucho tiempo, al deber realizar el análisis de las grabaciones manualmente. Relacionado con el análisis, los resultados no son fiables ya que dependen de la objetividad del evaluador al realizarlo. Finalmente, no se estudia el entorno fuera del modelo de usuario (características demográficas).

Característica estudiada	*Loop11*	*TryMyUI*	*OpenHallway*	*Userlytics*	*UserZoom*
Tipo de aplicación	*Web*	*Ambas*	*Web*	*Ambas*	*Ambas*
Método de captura	*C y TZ*	*C, GUV y GP*	*C, GUV y GP*	*C, GUA y GP*	*C, GUV y GP*
Equipamiento necesario	*-*	*App*	*-*	*App y Cam*	*App y Cam*
Grado de automatización	*Captura y análisis*	*Captura*	*Captura*	*Captura*	*Captura y análisis*
Lugar de realización	*E. Reales*	*E. Reales*	*E. Reales*	*E. Reales*	*E. Reales*
Finalidad de la evaluación	*Ambos*	*Ambos*	*Ambos*	*Ambos*	*Ambos*
Conocimientos adicionales	*Sí*	*Sí*	*No*	*No*	*No*
Tiempo requerido	*Bajo*	*Alto*	*Alto*	*Alto*	*Alto*
Fiabilidad de los datos	*Baja*	*Baja*	*Baja*	*Baja*	*Baja*
Grado de privacidad	*Alto*	*Medio*	*Medio*	*Bajo*	*Bajo*
Modelo de contexto	*USR, TAR y DIS*	*USR, TAR, DIS y ENT*	*USR, TAR y DIS*	*USR, TAR y DIS*	*USR, TAR y DIS*
Uso de experiencia previa	*Plantillas de pruebas anteriores*	*Lista predefinida de entornos*	*No*	*Plantillas de tareas y cuestionario*	*No*

Tabla 2.5 Características de las soluciones comerciales

2.3.2.2. SOLUCIONES DEL ÁMBITO CIENTÍFICO

En este apartado detallamos las principales metodologías extraídas de la literatura científica.

- *MUSiC* [Macleod+97]. Es una rigurosa metodología de ámbito general para la medición de la usabilidad que permite especificar los requisitos de usabilidad de un producto para posteriormente evaluarlo y saber si cumple

con estos requisitos. El enfoque del proyecto MUSIC es similar al ofrecido por el estándar ISO 9241-11:1998 [ISO98], define la usabilidad como la calidad de la interacción en un contexto particular, midiendo eficacia, eficiencia y satisfacción. El proyecto proporciona herramientas que permiten identificar y definir los requisitos de las métricas de usabilidad y los componentes del contexto. En primer lugar se debe definir el producto a evaluar. Posteriormente, se define el contexto en uso donde se deben realizar las pruebas mediante una guía [Thomas+96]. Después se define el contexto de evaluación y los objetivos del mismo para consecutivamente preparar la evaluación y ejecutar las pruebas, donde se generan vídeos de la ejecución de las pruebas hechas por los usuarios y los usuarios responden a un cuestionario SUMI (Software Usability Measurement Inventory) [Kirakowski+93]. Una vez ejecutadas se realiza un análisis e interpretación de las métricas para finalmente elaborar un informe de usabilidad. Para la ejecución de esta metodología se desarrolló una herramienta denominada DRUM (Diagnostic Recorder for Usability Measurement) [Macleod+93], cuyo objetivo era simplificar los pasos de la metodología facilitando el análisis y la anotación de los vídeos. Al igual que hemos observado con las soluciones comerciales, el análisis de los vídeos debe realizarse de un modo manual, por ello, una de sus principales limitaciones es el tiempo requerido. Por otro lado, la principal ventaja es que aunque no captura factores del contexto, sí que los define detalladamente y son tomados en cuenta. Dicho contexto está basado en usuarios, equipamiento, tareas, entorno organizacional, entorno técnico y entorno físico.

- *RECON* [Jensen09, Jensen12]. El objetivo de RECON, al igual que otras plataformas más antiguas en las que está basada (*ContextPhone* [Raento+05], *MyExperience* [Froehlich+07], *SocioXensor* [Mulder+05], etc.), es capturar información del contexto para combinarla con conjuntos de datos de interacción sobre las situaciones en las que ha sido realizada. Los requisitos los resume con el acrónimo

SIERRA: Seguro, invisible, eficiente, robusto, remotamente controlable y autónomo. La metodología que sigue es el configurar los terminales móviles con la aplicación a evaluar y una librería, ejecutar las pruebas y llevar a cabo el análisis de los resultados. Mediante la librería que se vincula a la aplicación, se captura la interacción a través del código de la aplicación y un simple API de registro con el que se informan de los eventos de interacción. Los principales beneficios de este método son que permite el estudio del uso a largo plazo y la interacción en contextos reales, no requiere tiempo y esfuerzo. Es posible anonimizar los datos automáticamente y realizar estudios en ámbitos donde la observación normal no sería posible sin comprometer la privacidad. Sin embargo, está dirigido específicamente para datos objetivos y cuantitativos, lo que indica que principalmente sirve para un enfoque sumativo.

- *Lettner y Holzmann* [Lettner+12]. Presenta una metodología y un prototipo formado por un conjunto de herramientas que permite a los desarrolladores realizar experimentos de usabilidad. En este trabajo, los pasos para llevar a cabo el análisis son: integrar la aplicación a evaluar con el conjunto de herramientas de este trabajo, ejecutar las pruebas y generar los análisis. Con este enfoque se registran los datos de interacción del usuario y se calculan métricas de usabilidad de un modo automático. También presentan la navegación de usuario por lo que se considera que puede llevarse a cabo un análisis formativo. Aunque no hace uso de ningún conocimiento previo para ayudar a la evaluación, el tiempo requerido es bajo ya que los desarrolladores no tienen que agregar código manualmente para evaluar la aplicación y los datos cuantitativos son presentados de un modo automático. Desafortunadamente, esta solución no tiene en cuenta el contexto aunque sí lo mencionan como posible mejora.
- *Carta et al.* [Carta+11]. El enfoque propuesto en este trabajo, al igual que el planteado una década atrás por Hong et al.

[Hong+01a], se basa en un servidor proxy intermedio cuyo objetivo es añadir a las páginas web visitadas código que registre el comportamiento real del usuario sin necesidad de llevar a cabo ninguna instalación o configuración específica. El servidor cuenta con una aplicación web que gestiona las pruebas de usabilidad. Los pasos que se deben realizar son una fase de especificación de las tareas a realizar y los factores a capturar (orientación y movimiento del dispositivo, toques en la pantalla…), la realización de las mismas por parte de los usuarios y el análisis, para el cual se presentan gráficos de la interacción mediante líneas de tiempo. Mediante esta solución se pueden realizar estudios de tipo formativo. Este trabajo presenta como principal problema el tiempo que deben invertir los evaluadores en realizar el análisis.

- *Biel et al.* [Biel+10]. El método propuesto por estos autores se basa en la combinación de un análisis de la arquitectura del software y una evaluación de la usabilidad en forma de pruebas de usuario. En primer lugar se definen los requisitos funcionales, de usabilidad y los factores de contexto (usuarios, entornos, tareas, dispositivos y aplicación). En segundo lugar se lleva a cabo una evaluación de la arquitectura de la información mediante su sistema conocido como SATURN (Software ArchitecTure analysis of Usability Requirements realizatioN) y paralelamente una evaluación de usabilidad. En este trabajo no especifica un método de evaluación concreto ya que delegan la elección al evaluador, que debe elegir el más idóneo dependiendo del objetivo de la evaluación. Presenta un modelo de contexto muy elaborado y dividido en cinco grupos: usuario, entorno, tarea, dispositivo y aplicación. Para la validación de su trabajo realiza un caso de estudio donde la evaluación de usabilidad está realizada por una herramienta conocida por

MacEval[25], fruto de la investigación de Thomas Grill cuyo análisis es realizado mediante el estudio de contenido audiovisual generado por la misma. En la metodología presentada en este trabajo apreciamos una fuerte importancia del contexto en el proceso de evaluación. Sin embargo, muestra una restricción en cuanto al tiempo de análisis ya que tanto la revisión de la arquitectura de la información como el análisis de la usabilidad requieren una elevada cantidad de tiempo.

En la revisión de estos trabajos (ver tabla 2.6) hemos detectado un mayor interés en la elaboración y estudio del contexto añadiendo más elementos fuera de los datos demográficos del usuario.

Característica estudiada	*MUSiC*	*RECON*	*Lettner y Holzmann*	*Carta et al.*	*Biel et al.*
Tipo de aplicación	*Ambas*	*Nativas*	*Nativas*	*Web*	*Ambas*
Método de captura	*C, GUV y GP*	*TZ*	*TZ*	*TZ*	*C, GUV y GP*
Equipamiento necesario	*Cam*	*App*	*App*	*-*	*Cam*
Grado de automatización	*Captura*	*Captura y análisis*	*Captura y análisis*	*Captura*	*Captura*
Lugar de realización	*E. Reales*	*E. Reales*	*E. Reales*	*E. Reales*	*E. Reales*
Finalidad de la evaluación	*Ambos*	*Sumativo*	*Ambos*	*Formativo*	*Formativo*
Conocimientos adicionales	*No*	*Sí*	*No*	*No*	*No*
Tiempo requerido	*Alto*	*Bajo*	*Bajo*	*Medio*	*Alto*
Fiabilidad de los datos	*Baja*	*Baja*	*Alta*	*Baja*	*Baja*
Grado de privacidad	*Bajo*	*Alto*	*Alto*	*Alto*	*Bajo*
Modelo de contexto	*USR, TAR, EQU y ENT*	*USR, TAR, EQU y ENT*	*No*	*USR, TAR y DIS*	*USR, TAR, DIS, APP y ENT*
Uso de experiencia previa	*No*	*No*	*No*	*No*	*No*

Tabla 2.6 Características de las soluciones del ámbito científico

[25]http://www.tomgrill.info/maceval

También hacen mayor uso de métodos de captura basados en la generación de trazas, lo que ayuda a la mejora del grado de privacidad del usuario. Otra cuestión que debemos destacar es que no hemos detectado que se haga uso de conocimiento previo para agilizar la evaluación.

2.4. CONCLUSIONES Y SOLUCIÓN PROPUESTA

A través de este capítulo hemos introducido de un modo general el área de la usabilidad y de un modo más específico dentro del campo de las aplicaciones móviles. En primer lugar, hemos estudiado las limitaciones que presentan las aplicaciones móviles en sí (ver apartado 2.2.1), tomando consciencia de *la importancia de la movilidad y su contexto para este tipo de aplicaciones*. Éste presenta un impacto significativo de los elementos del mismo en la usabilidad, donde existe una conectividad extremadamente variable y una elevada probabilidad de interrupción de las tareas, cuyo tiempo es limitado. Posteriormente hemos identificado los principales retos en el campo de la evaluación de la usabilidad de aplicaciones móviles (ver apartado 2.2.3).

2.4.1. ANÁLISIS Y CONCLUSIONES

A nivel general se ha concluido que *existe una fuerte necesidad en el estudio de las características del entorno dentro del estudio de la usabilidad de aplicaciones móviles*, donde predominan las evaluaciones en entornos de laboratorio y en entorno reales. En estas últimas, las cuales son abordadas en este trabajo, existen varios retos. Por un lado, el elevado coste en cuanto a tiempo, la baja fiabilidad de los datos capturados y la amenazada privacidad de los usuarios que realizan las tareas de la evaluación.

Las posibles aproximaciones para hacer frente a los retos planteados son las siguientes:

- Para reducir el *coste* de la evaluación en términos de tiempo debemos reducir la duración de las diferentes fases de la evaluación. Para ello existen tres principales opciones. En

primer lugar, podemos *reducir el número de participantes o pruebas* que deben realizar reduciendo la calidad de los datos obtenidos. También podemos *automatizar diferentes pasos de la evaluación* con lo que obtendríamos una reducción de recursos significativo. Además, podemos hacer *uso de conocimiento previo* (p.ej., plantillas de preguntas o tareas predefinidas) para agilizar los procesos. En términos de recursos, se puede reducir el equipamiento necesario para la evaluación a un *equipamiento necesario mínimo*. Consideramos que para las pruebas de usabilidad en entornos reales sólo es necesario el uso de los terminales móviles y un servidor web para centralizar los resultados.

- La *fiabilidad* de los resultados puede verse comprometida por dos principales causas. Por un lado, el sesgo de la interacción ocasionada por los agentes que capturan los datos (p.ej., cámaras adheridas al dispositivo móvil o usuarios nerviosos por sentirse observados). Podemos reducir dicho sesgo *utilizando métodos de captura poco intrusivos* como la captura de trazas y *automatizando el proceso*. Por otro lado, puede haber un mal registro en la captura o análisis de la interacción (p.ej., un observador omite errores observando a un usuario por distracción). Para solucionar el mal registro, al igual que el problema anterior, podemos *utilizar herramientas automáticas* que realicen dicho trabajo.
- La extrema *relevancia del contexto* en este tipo de aplicaciones hace indispensable y obligatoria su consideración. Para ello *se deben estudiar modelos de contexto complejos*, es decir, modelos que estén compuestos por conjuntos de factores que no se limiten a características demográficas y detallen características del entorno.
- La *privacidad* es un aspecto que puede verse comprometido si registramos la interacción del usuario mediante su grabación en vídeo mientras realiza las pruebas en entornos reales (p.ej., en la casa del usuario). Sin embargo, si *sustituimos la grabación por la captura de los factores relevantes*

del contexto no comprometeremos gravemente la privacidad del usuario.

- Por otro lado, debido a la existencia de las dos principales finalidades (sumativa y formativa), *debemos proporcionar una solución integral que permita cubrir ambas finalidades de evaluación*.

Como podemos observar en el resumen del análisis expuesto en la tabla, *no existe una metodología que afronte la totalidad de los retos planteados*.

	Loop11	*TryMyUI*	*OpenHallway*	*Userlytics*	*UserZoom*	*MUSiC*	*RECON*	*Lettner y Holzmann*	*Carta et al.*	*Biel et al.*	***Solución propuesta***
Equipamiento físico necesario mínimo	✓	✓	✓				✓	✓	✓		✓
Automatización de captura y análisis	✓				✓		✓	✓			✓
Ambas finalidades de evaluación	✓	✓	✓	✓	✓	✓		✓			✓
Poco tiempo requerido de evaluación	✓						✓	✓		✓	✓
Fiabilidad de los datos alta								✓			✓
Grado de privacidad alto	✓						✓	✓	✓		✓
Modelo de contexto complejo		✓				✓	✓			✓	✓
Uso de experiencia previa	✓	✓		✓							✓

Tabla 2.7 Retos abordados por las soluciones estudiadas en comparación con la solución propuesta

2.4.2. SOLUCIÓN PROPUESTA

Habiendo identificado esta limitación, para el desarrollo de esta tesis se propone lo siguiente:

> *Desarrollar un nuevo enfoque que permita la evaluación de la usabilidad de aplicaciones móviles persiguiendo reducir el coste de la evaluación en términos de recursos (equipamiento y tiempo necesarios) sin comprometer la fiabilidad de los resultados ni la privacidad de los usuarios que participen en las pruebas, tomando en cuenta el contexto y la experiencia previa, ofreciendo resultados*

que soporten tanto evaluaciones formativas como sumativas.

Dentro de este trabajo hemos definido varios requisitos generales que debe cumplir el enfoque propuesto.

- *R1. Debe ser capaz de ofrecer resultados para evaluaciones cuya finalidad sea tanto formativa como sumativa*.

  Esto se conseguirá implementando los cálculos de las principales métricas para la finalidad sumativa y un modelo de identificación de errores de interacción para la formativa.

- *R2. La cantidad de recursos necesaria debe ser reducida*. Entendiendo como recursos a reducir el tiempo de la evaluación y el equipamiento necesario para desarrollar la misma.

  Se logrará mediante cuatro principales frentes. En primer lugar, gracias a la automatización de varios pasos de la metodología se reducirá el tiempo. Además, se añadirá una base de conocimiento mediante la cual se logrará reducir el número de tareas y usuarios para obtener los mismos resultados. También, se ofrecerán procedimientos y herramientas simples y fáciles de aprender para agilizar el proceso. Finalmente, se realizará la evaluación con un equipamiento mínimo: utilizando solamente los terminales de los usuarios que realizan las pruebas y un servidor web para centralizar los datos capturados.

- *R3. La calidad de los resultados no debe disminuir*. Aunque la cantidad de recursos sea reducida, la fiabilidad de los datos debe ser alta. Es decir, durante la realización de las pruebas no debe generarse ningún sesgo por las herramientas de captura de las mismas y los datos analizados no deben mostrar variaciones debidas al evaluador de las mismas en su análisis.

  Se logrará cumplir este requisito mediante el uso de herramientas de captura poco intrusivas basadas en una

aplicación móvil de captura de interacción y contexto y una librería de integración que residirán en los terminales móviles. Además, se ofrecerá una herramienta de gestión y análisis con la que automatizará el cálculo de los resultados.

- *R4. La privacidad de los usuarios que realizan las pruebas debe ser preservada*.

  Se cumplirá este requisito realizando la captura sin almacenar grabaciones en vídeo o audio de las pruebas almacenando y ocultando datos que dificulten la identificación individual del usuario.

- *R5. El estudio de un modelo de contexto detallado debe ser posible*. Como ya se ha explicado, se debe estudiar un modelo de contexto que abarque no solo características básicas del usuario y las tareas desarrolladas, sino también características del entorno donde desarrolla el usuario la interacción con la aplicación.

  Dicho requisito se logrará mediante el diseño de un modelo de contexto centrado en la evaluación de aplicaciones móviles y la captura del mismo durante el desarrollo de las pruebas.

Para completar la totalidad de requisitos proponemos un nuevo enfoque compuesto por una metodología de evaluación de usabilidad de aplicaciones móviles y una plataforma de soporte que permita llevarla a cabo. Estos componentes se complementarán permitiendo lograr los requisitos definidos.

Figura 2.3 Componentes de la solución propuesta en esta tesis

2.4.2.1. METODOLOGÍA DE EVALUACIÓN DE USABILIDAD

Ésta metodología, por sí sola, quizás no constituiría un trabajo demasiado innovador puesto que ya existen otras metodologías utilizadas para este mismo ámbito. Desafortunadamente, *las metodologías existentes hacen un uso muy superficial del conocimiento previo*, limitándose al reaprovechamiento de conocimiento estático (p.ej., plantillas predefinidas de preguntas de cuestionario, tareas populares, etc.). Sin embargo, en esta nueva aproximación hacemos un mayor uso del conocimiento previo mediante la generación de una base de conocimiento con evaluaciones anteriores.

Esta principal característica condiciona la formulación de la hipótesis cuya verificación promueve la presente investigación:

> *Es posible reducir los recursos necesarios en la evaluación de la usabilidad de aplicaciones móviles sin comprometer la calidad de los resultados mediante una nueva metodología de evaluación centrada en una base de conocimiento.*

Para afrontar esta hipótesis planteamos una metodología cuyas características deben satisfacer los siguientes requisitos para la metodología (RM).

- *RM1*. Debe proponer un modelo teórico que ofrezca procedimientos y cálculos para generar resultados de evaluación sumativa y formativa.
- *RM2*. Debe proporcionar un proceso simple y fácil de estudio y elección de los atributos de las pruebas que generen mejores resultados.
- *RM3*. La metodología debe ofrecer procedimientos de análisis que no deterioren la calidad de los datos de la evaluación en la fase de captura ni en la fase de análisis.
- *RM4*. Los modelos con los que se realicen la evaluación no deben demandar datos cuya captura comprometan la

privacidad del usuario que realiza las pruebas (p.ej., datos personales que identifiquen al usuario).

- *RM5*. La metodología debe contener y estudiar un modelo de contexto complejo que abarque los factores más relevantes que compongan el mismo.

Explicaremos los procedimientos y modelos de esta metodología en detalle en el capítulo 3.

2.4.2.2. PLATAFORMA DE SOPORTE

La metodología propuesta en esta tesis, no puede cumplir los objetivos definidos por sí sola. Para ello debe ser soportada por un conjunto de herramientas software que ayuden en la ejecución de los diferentes pasos a realizar en la metodología.

> *Dicho conjunto de herramientas automatizarán ciertos pasos de la metodología y harán uso del menor equipamiento posible para realizar tanto la captura de los datos de las pruebas de evaluación de un modo remoto (principal reto de la plataforma de soporte al no sesgar la interacción), como la gestión de la metodología.*

Explicaremos esta plataforma en detalle en el capítulo 4. Para cumplir satisfactoriamente con su principal objetivo debemos cumplir los siguientes requisitos para la plataforma (RP) extraídos de los generales, anteriormente propuestos para este trabajo.

- *RP1*. Debe automatizar el proceso de cálculo de resultados de evaluación sumativa y formativa definidos en la metodología.
- *RP2*. Debe ofrecer herramientas que faciliten el proceso de análisis y gestión de las pruebas de usabilidad que permita llevar a cabo los procesos de estudio y elección de atributos de las pruebas que generen mejores resultados.
- *RP3*. Debe automatizar tanto la captura de la interacción del usuario como la captura de los factores que compongan un

modelo de contexto complejo que no añada sesgos a la interacción del usuario con la aplicación.

- *RP4*. La captura de la interacción del usuario se debe realizar con herramientas que no amenacen la privacidad del usuario que realiza las pruebas.
- *RP5*. Las herramientas que componen la plataforma de soporte deben capturar los elementos del modelo de contexto definido en la metodología.

Resumiendo la solución propuesta a desarrollar en esta tesis mediante la figura 2.4, recapitulamos lo avanzado. Hemos presentado la hipótesis que abordaremos en el presente trabajo, junto con unos objetivos generales que la solución debe satisfacer, fruto del estudio previo del estado del arte. Para cumplir estos requisitos se han definido dos principales componentes (una nueva metodología y una plataforma de soporte) que cumplirán sus propios requisitos para lograr en conjunto los generales.

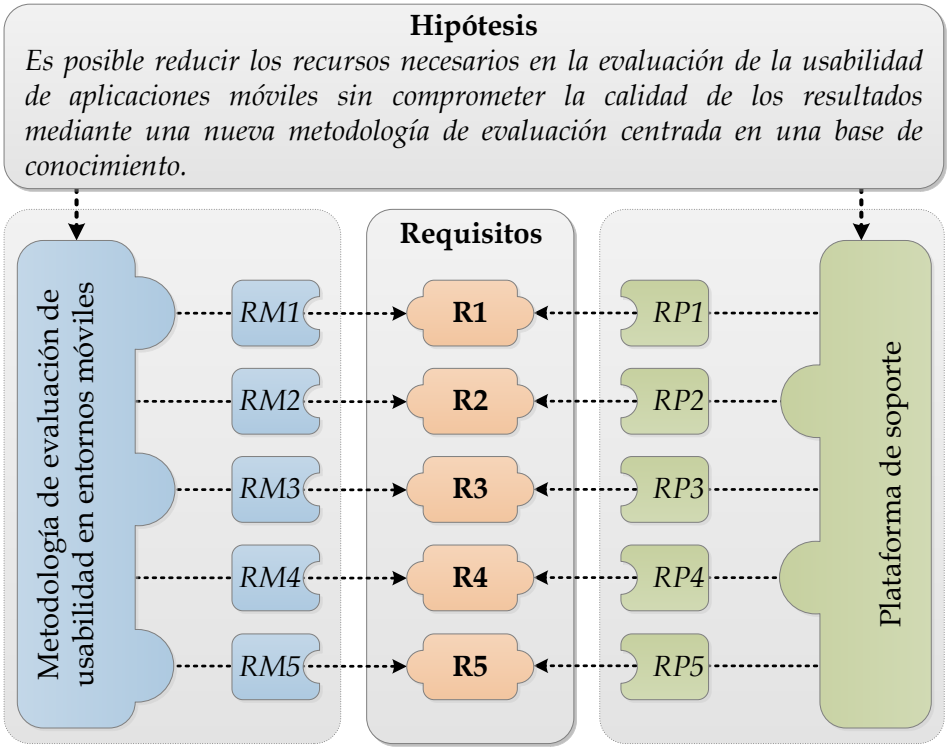

Figura 2.4 Resumen de la solución propuesta en esta tesis y sus requisitos

CAPÍTULO 3

METODOLOGÍA DE EVALUACIÓN DE USABILIDAD EN ENTORNOS MÓVILES

«Lo importante en ciencia no es tanto obtener nuevos hechos como descubrir nuevas formas de pensar sobre ellos»,
William Lawrence Bragg (1890-1971)

ÍNDICE DE CAPÍTULO 3

3.1. Descripción general 71
3.1.1. Fase de definición 73
3.1.2. Fase de ejecución 76
3.1.3. Fase de análisis 78
3.2. Base de conocimiento 80
3.2.1. Modelo de contexto 81
3.2.2. Modelo de interacción 90
3.2.3. Modelo de análisis 98
3.2.4. Modelo de casos favorables 109
3.3. Relación de las fases con la base de conocimiento 131

Habiendo abordado en los anteriores capítulos los conceptos fundamentales sobre la usabilidad, su situación y relación con el ámbito móvil además de los retos más íntimamente ligados con la presente tesis, explicamos en este capítulo la descripción de la metodología propuesta y desarrollada.

Antes de comenzar, recapitulemos que la metodología debía cumplir ciertos requisitos propios (ver apartado 2.4.2.1) para que este trabajo pudiera lograr los requisitos generales expuestos en detalle en la presentación de la solución propuesta del final del capítulo anterior (ver apartado 2.4.2). Estos requisitos junto con los

requisitos generales a los que están ligados, son resumidos mediante la tabla 3.1.

Requisito general	*Requisito de la metodología*
R1. Debe ser capaz de ofrecer resultados para evaluaciones cuya finalidad sea tanto formativa como sumativa.	*RM1. Debe proponer un modelo teórico que ofrezca procedimientos y cálculos para generar resultados de evaluación sumativa y formativa.*
R2. La cantidad de recursos necesaria debe ser reducida.	*RM2. Debe proporcionar un proceso simple y fácil de estudio y elección de los atributos de las pruebas que generen mejores resultados.*
R3. La calidad de los resultados no debe disminuir. Aunque la cantidad de recursos sea reducida, la fiabilidad de los datos debe ser alta.	*RM3. La metodología debe ofrecer procedimientos de análisis que no deterioren la calidad de los datos de la evaluación en la fase de captura ni en la fase de análisis.*
R4. La privacidad de los usuarios que realizan las pruebas debe ser preservada.	*RM4. Los modelos con los que se realice la evaluación no deben demandar datos cuya captura comprometan la privacidad del usuario que realiza las pruebas (p.ej., datos personales que identifiquen al usuario).*
R5. El estudio de un modelo de contexto detallado debe ser posible.	*RM5. La metodología debe contener y estudiar un modelo de contexto complejo que abarque los factores más relevantes que compongan el mismo.*

Tabla 3.1 Resumen de los requisitos que debe cumplir la metodología

Para cumplir los requisitos definidos, la metodología presenta varios modelos que cumplen con los mismos:

- Un *modelo de análisis* provisto de procedimientos tanto sumativos como formativos (RM1).
- Un *modelo de casos favorables* para el estudio de las mejores situaciones con las que proporcionar criterios objetivos de elección de los atributos de las pruebas que generen mejores resultados (RM2).
- Un *modelo de interacción* que se compone de ciertas reglas y métodos que determinan cuándo se generan errores de interacción y la severidad de los mismos de un modo objetivo, eliminando el posible sesgo de los evaluadores (RM3).
- Un *modelo de contexto* en el que se definen y estudian varios factores relevantes para este tipo de aplicaciones (RM5). Además, está compuesto por datos que no facilitan la identificación del usuario (RM4).

En la siguiente sección presentaremos la metodología con su descripción general, el detalle de cada una de las fases que la conforman y una explicación de la composición de la base de conocimiento, principal pilar de la misma. Posteriormente, detallaremos los modelos que componen la base de conocimiento utilizada que da soporte a la metodología. Finalmente, describiremos la relación de las fases de la metodología con la base de conocimiento presentada.

3.1. DESCRIPCIÓN GENERAL

El nuevo enfoque que proponemos mediante esta metodología tiene como *principales objetivos simplificar las evaluaciones de aplicaciones móviles y reducir la necesidad de recursos*. Para ello, la principal base de la misma es el uso de información de evaluaciones anteriores. Con ello fundamentaremos la elección de los diferentes elementos que componen los contextos cuando debamos definir una evaluación de usabilidad en entornos de movilidad.

Como se distingue en la figura 3.1, la metodología consta principalmente de tres fases. Dichas fases persiguen proporcionar una evaluación rápida y eficaz, además de nutrir y utilizar la base de conocimiento definida posteriormente en la sección 3.2.

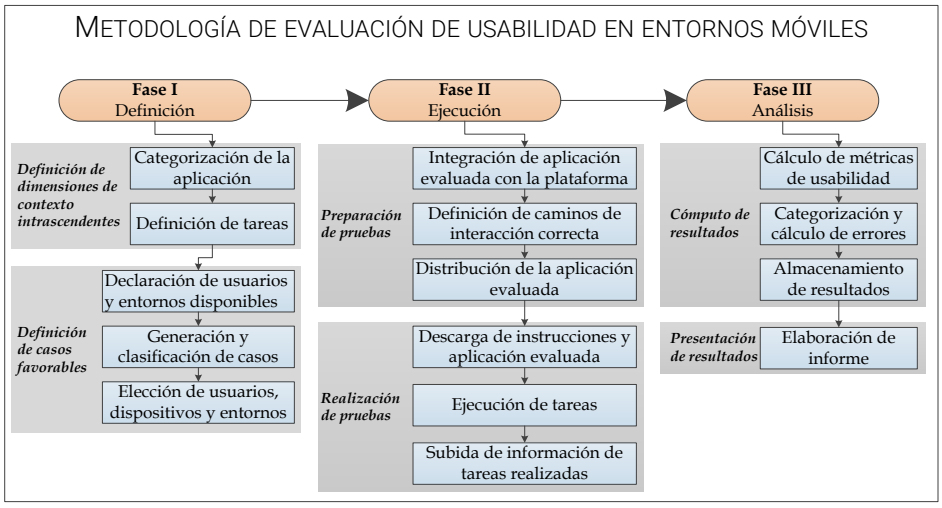

Figura 3.1 Fases de la metodología de evaluación de usabilidad en entornos móviles

Antes de comenzar la descripción de las fases, debemos aclarar ciertos conceptos y agentes que son mencionados a lo largo de este trabajo.

- El término aplicación puede referirse a cualquier desarrollo software. Sin embargo y debido al enfoque de esta tesis, acotamos este término con la siguiente definición:

  Entendemos una aplicación o aplicación evaluada como un producto software que es ejecutado en terminales móviles para lograr realizar tareas concretas y es objeto de evaluación mediante la metodología propuesta.

- Dichas aplicaciones son creadas por desarrolladores de software en fases anteriores a la evaluación. Dentro de este trabajo acotamos lo siguiente:

  Un desarrollador es una persona con conocimientos en desarrollo de aplicaciones móviles que ha creado o ha colaborado en la creación de una aplicación evaluada.

- Otro término muy importante es el de evaluador. Dichos evaluadores son los principales agentes que deben llevar a cabo la metodología propuesta.

  Entendemos que un evaluador es una persona interesada en la evaluación de la usabilidad mediante la metodología propuesta de una aplicación.

  En este trabajo, como no son necesarios grandes conocimientos de usabilidad, *el propio desarrollador puede ser el evaluador.*

- Finalmente, para llevar a cabo el correcto desarrollo de la metodología, los usuarios de pruebas deben utilizar la aplicación evaluada.

  Un usuario de pruebas es un potencial usuario final de la aplicación que realiza las tareas definidas para la

evaluación de la aplicación mediante el uso de la misma.

Como se muestra en la figura 3.2, el evaluador de la aplicación realiza una definición de las pruebas a realizar en una primera fase. Posteriormente, se realizará una preparación de las pruebas por parte del desarrollador y una realización de las mismas por parte de los usuarios de pruebas. Finalmente, se llevará a cabo un análisis automático donde los resultados finales serán mostrados al evaluador.

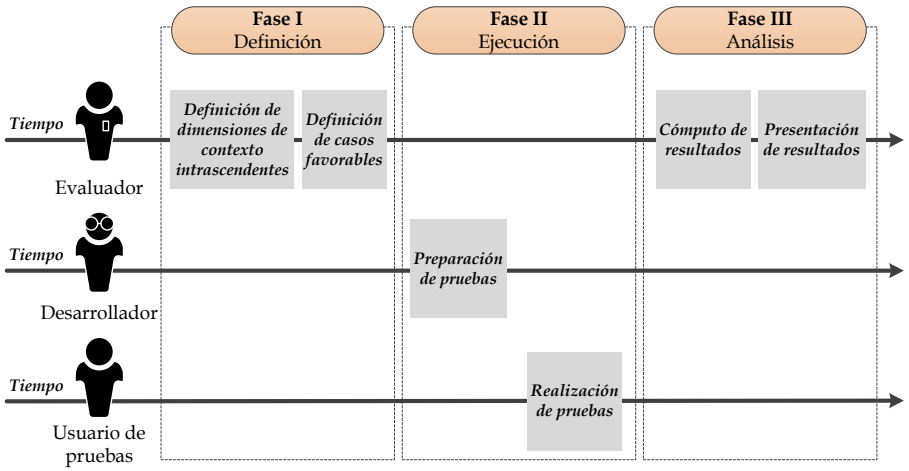

Figura 3.2 Fases en las que se involucran los agentes de la metodología

Habiendo explicado los diferentes términos y agentes que están envueltos en la metodología, procedemos a detallar de un modo más explícito las fases de la metodología y el conjunto de pasos que las componen.

3.1.1. FASE DE DEFINICIÓN

La fase de definición es la más crítica de cara a realizar una evaluación de usabilidad con la nueva metodología. *El objetivo de la misma es definir el modelo de contexto* que explicaremos posteriormente en el apartado 3.2.1. Esta fase está dividida principalmente por dos conjuntos de pasos: definición de dimensiones de contexto intrascendentes y definición de casos favorables.

3.1.1.1. DEFINICIÓN DE DIMENSIONES DE CONTEXTO INTRASCENDENTES

Dentro de la definición de dimensiones de contexto intrascendentes debemos aclarar este concepto.

> *Entendemos dimensiones de contexto intrascendentes como las dos dimensiones del modelo de contexto (aplicación y tareas) que son consideradas poco importantes desde el punto de vista de ahorro de recursos en la evaluación.*

Para su definición destacamos dos principales pasos: definición de la categoría de la aplicación y definición de tareas.

- Primeramente, detallamos la aplicación y definimos la *categoría de la aplicación* correspondiente (p.ej. compras, comunicación, deportes,…) acorde a las definidas en la base de conocimiento.
- Una vez categorizada la aplicación a evaluar, debemos *definir las tareas* que posteriormente realizarán los usuarios. Aunque como ya hemos mencionado, no son asumidas como importantes de cara a la optimización de recursos, sí que deben tener una correcta definición para una evaluación de la aplicación sin sesgos. Dichos sesgos pueden estar ocasionados por una mala definición de las tareas, ya que pueden no ser interpretadas correctamente. Para ello se hace uso de las tareas definidas anteriormente para aplicaciones de igual categoría de aplicación. Al basarse en la teoría de que *los usuarios con tareas de la misma categoría realizan tareas similares*, esta metodología muestra las tareas realizadas con otras aplicaciones similares. Por ejemplo, las tareas comunes de la mayoría de las aplicaciones categorizadas como compras son: buscar un producto, comprar un producto o marcar un producto como favorito. En el caso de no disponer de ninguna tarea acorde a las que se deben realizar, se crearán nuevas.

3.1.1.2. DEFINICIÓN DE CASOS FAVORABLES

Dentro del paso de definición de casos favorables se encuentran las definiciones de las dimensiones más críticas. *La correcta definición de dichas dimensiones es la que realmente va a marcar la diferencia en el ahorro de tiempo y recursos durante la evaluación de la aplicación.* Como hemos hecho en el apartado anterior, antes de comenzar es muy importante aportar la siguiente definición:

> *Un caso favorable es una combinación de entornos y usuarios que muestran una posible inducción al aumento del número de errores detectados en la evaluación de una aplicación móvil.*

Para la elección de las restantes dimensiones de contexto, se presta especial atención a los casos favorables generados mediante un cálculo de probabilidades y una descripción de los valores más adecuados de las variables de contexto (ver apartado 3.2.4). Destacamos tres principales pasos: la declaración de los usuarios y entornos disponibles, la generación y clasificación de los casos y la elección final de los usuarios de pruebas, dispositivos y entornos.

- En este primer paso, *se declaran los tipos de usuario y entornos disponibles* para la realización de las pruebas. Los tipos de usuario son definidos en función del atributo género, por lo que las tres opciones disponibles son: hombres, mujeres o ambos. Por otro lado, los tipos de entorno se definen en función de un nombre genérico (p.ej. caminando por la calle). Los entornos que hayan sido utilizados para la evaluación de aplicaciones anteriores son recuperados de la base de conocimiento. Si no se disponen de los entornos que el evaluador desea, éste debe crearlos mediante un nombre y una descripción del mismo. A modo ejemplo se muestra en la tabla 3.2 la definición de un nuevo entorno.

Nombre	*Descripción*
Tumbado en casa	*Estando en casa, lo importante es que el usuario esté tumbado (sea boca arriba, de costado o boca abajo) en una cama o sofá.*

Tabla 3.2 Ejemplo de definición de nuevo entorno

- Una vez declarados los usuarios y tareas, *se generan y clasifican los casos posibles* correspondientes.

  > *Entendemos como caso posible, una combinación de entornos y usuarios que son candidatos a ser caso favorable.*

  Después, clasificamos los casos posibles y concluimos los casos favorables mediante el cálculo de probabilidades. Finalmente, calculamos los diferentes valores que las variables de contexto adquieren cuando hay más posibilidad de originarse errores. Todos los cálculos, hacen uso de los resultados del modelo de análisis de la base de conocimiento (ver apartado 3.2.3) y el modelo de casos favorables (ver apartado 3.2.4).

- Habiendo obtenido los casos favorables y la descripción de sus valores, realizaremos una observación y estudio de los mismos. Mediante este estudio podremos realizar la correcta *elección de los usuarios, dispositivos y entornos*. Es importante señalar que los usuarios han sido previamente registrados para ser seleccionados, al igual que los dispositivos, estrechamente ligados al usuario. Por ello, consideramos la elección del dispositivo ligada al usuario de pruebas.

Como resultado de esta fase, se obtiene una completa definición del conjunto de entornos, usuarios, tareas y dispositivos en los que se realizarán las pruebas. El siguiente paso será llevar a cabo la ejecución de las mismas.

3.1.2. FASE DE EJECUCIÓN

En esta fase, los usuarios de pruebas ejecutarán las tareas en los entornos especificados en la fase anterior. Esta fase está dividida en dos principales bloques: la preparación de las pruebas y la realización de las mismas. Éstos están estrechamente ligados al objetivo de la correcta ejecución de las pruebas.

3.1.2.1. PREPARACIÓN DE PRUEBAS

Antes del lanzamiento de las pruebas, debemos realizar una preparación de la aplicación evaluada para que esté integrada con la plataforma de soporte y podamos realizar su posterior lanzamiento y captura de interacción en los terminales de los usuarios de pruebas.

- En primer lugar, el desarrollador debe realizar una *integración de la aplicación evaluada con la plataforma de soporte* (los detalles de la integración mediante la librería de integración implementada se especifican en la sección 4.3). Dicha integración nos permite capturar las variables relevantes para generar los modelos de contexto e interacción mediante una aplicación móvil de captura de pruebas (ver sección 4.6).
- Después, el desarrollador debe efectuar una *definición de los caminos de interacción correcta* (ver apartado 3.2.2.3) del modelo de interacción (definido en el apartado 3.2.2) correspondiente a las tareas que deben realizar los usuarios. Con dichos caminos, capturaremos el desarrollo de la interacción y detectaremos los errores que se cometan durante la ejecución de las pruebas.
- Finalmente, el desarrollador realiza la *distribución de la aplicación* a los usuarios correspondientes mediante la plataforma de soporte, descrita más adelante.

3.1.2.2. REALIZACIÓN DE LAS PRUEBAS

Cuando se han realizado los pasos de preparación de pruebas por parte del desarrollador, los usuarios de pruebas deben ejecutar las mismas. Es importante señalar que *este conjunto de pasos es realizado únicamente por los usuarios de prueba*.

Si es la primera vez que un usuario de pruebas se involucra con esta metodología, debe instalar y registrar en su dispositivo la herramienta de usuarios de pruebas definida en la sección 4.6 del siguiente capítulo. Gracias a esta herramienta, la interacción y el contexto podrán ser registrados durante el uso de las aplicaciones

evaluadas. Además, durante la instalación, el usuario llevará a cabo su registro en la plataforma, quedando almacenados sus datos en la base de conocimiento. Consta de tres pasos:

- El usuario de pruebas debe *descargarse las instrucciones de la nueva evaluación y la aplicación evaluada* para posteriormente instalarla mediante la herramienta de usuario de pruebas definida posteriormente en la sección 4.6.
- Cuando el usuario de pruebas dispone de la aplicación evaluada y las instrucciones descargadas, el siguiente paso es *ejecutar las tareas* correspondientes. Para ello, el usuario de pruebas debe leer y comprender las instrucciones de las tareas antes de dar comienzo a las mismas mediante la herramienta de usuario de pruebas.
- Una vez realizadas las tareas correspondientes a la aplicación evaluada, el usuario de pruebas debe agregar los datos a la base de conocimiento mediante *la subida de la información de las tareas*. Para ello, una vez más hará uso de la herramienta de usuario de pruebas.

3.1.3. FASE DE ANÁLISIS

Finalmente, en la fase de análisis se calcularán y expondrán los resultados finales de la metodología. Esta fase concierne únicamente al evaluador de la aplicación y su objetivo es el análisis de los resultados. Consta de dos principales pasos: cómputo de resultados y presentación de los mismos.

3.1.3.1. CÓMPUTO DE RESULTADOS

En primer lugar, el cómputo de resultados presenta dos pasos que pueden ser ejecutados de un modo paralelo: cálculo de métricas de usabilidad, y cálculo y categorización de errores. Cuando se obtienen los resultados, se recogen en la base de conocimiento.

- Para realizar el *cálculo de las métricas de usabilidad* se hace uso del modelo de análisis presentado en el apartado 3.2.3. Gracias a dicho modelo se obtienen las métricas que serán

utilizadas para describir la usabilidad de la aplicación y para la elaboración del informe.

- Al igual que las métricas de usabilidad, se hace uso de los modelos de análisis e interacción para realizar el *cálculo y categorización de los errores detectados*. Sus resultados también son utilizados en el informe.
- Una vez calculados los dos tipos de resultados, se procede al *almacenamiento* de los mismos en la base de conocimiento.

3.1.3.2. PRESENTACIÓN DE RESULTADOS

Cuando realizamos todos los cálculos y almacenamos los resultados, los presentamos en un informe. Dicho informe consta de dos principales secciones que dan soporte a los dos principales enfoques de evaluación: enfoque sumativo y formativo.

- En el enfoque sumativo, se presentan las métricas de usabilidad más relevantes. A su vez, son presentadas desde un punto de vista general y en tres principales tipos de segmentación: por tipo de usuario de pruebas, tarea y entorno. Todas las métricas son descritas tanto a nivel individual como colectivo. Con el nivel individual, el evaluador puede centrar su atención explícitamente en una propiedad concreta. Con el nivel colectivo, el evaluador dispone de una visión global.
- Por otro lado, en el enfoque formativo nos muestra una catalogación y descripción de los errores de interacción que pueden estar asociados a problemas de usabilidad. Para ello, hacemos un informe en el que aparecen ordenados los errores detectados en función de su severidad descrita en el modelo de análisis (ver apartado 3.2.3.2). Conjuntamente, mostramos sus frecuencias de aparición, la interfaz y los objetos involucrados. Además, se ofrece una segmentación por tarea y entorno.

Una vez presentados los resultados, el evaluador dispone de la descripción necesaria como para realizar recomendaciones y mejoras. Consecuentemente, finalizamos la evaluación.

3.2. BASE DE CONOCIMIENTO

Como podemos apreciar en la descripción general de la metodología, la base de conocimiento es el principal pilar de la misma. Por ello, es de vital importancia que comprendamos su composición y objetivos. Antes de continuar, proponemos y entendemos como base de conocimiento lo siguiente:

> *Conjuntos de información representados y clasificados por cuatro modelos (contexto, interacción, análisis y casos favorables), adquiridos y generados a través de la experiencia con evaluaciones de usabilidad.*

Dicho de otro modo, la base de conocimiento almacena el conjunto de evaluaciones ya realizadas para poder generar conocimiento, cuyo objetivo es definir las situaciones en las cuales la usabilidad se ve más afectada y los problemas de usabilidad son propensos a aparecer. Para representar dichas situaciones se han desarrollado varios modelos que se complementan entre sí para formar en un conjunto la base de conocimiento de la metodología.

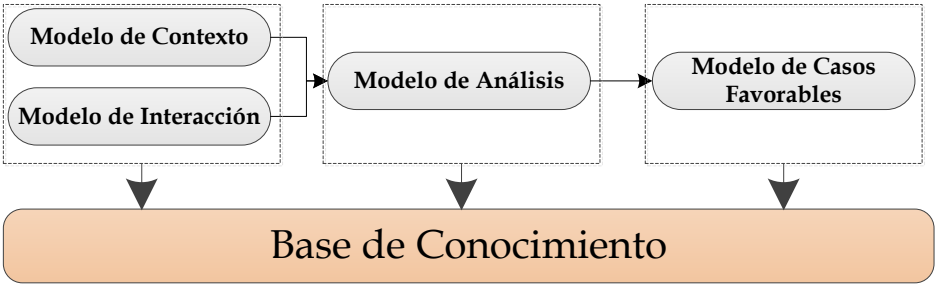

Figura 3.3 Modelos que forman la base de conocimiento

En primer lugar, el *modelo de contexto* (ver apartado 3.2.1) centrado en aplicaciones móviles permite describir los diferentes elementos que forman parte y afectan a la interacción. Para complementar este primer modelo, el *modelo de interacción* (ver apartado 3.2.2) añade la representación de la interacción del usuario de pruebas con la aplicación evaluada mientras desarrolla las tareas. Alimentado por estos dos modelos, el *modelo de análisis* (ver apartado 3.2.3) representa la usabilidad de la aplicación evaluada.

Finalmente y haciendo uso de los modelos anteriores, el *modelo de casos favorables* (ver apartado 3.2.4) representa las situaciones donde la usabilidad se ve más afectada. Mediante los siguientes apartados describimos explícitamente los modelos desarrollados.

3.2.1. MODELO DE CONTEXTO

Como ya se ha explicado, una de las principales bases para representar y almacenar la información de las evaluaciones de usabilidad de entornos móviles es el modelo de contexto.

Habiendo estudiado los modelos de contexto existentes en el capítulo anterior (ver apartado 2.2.4) nos centramos en la definición proporcionada por el estándar ISO 9241-11 [ISO98]. El contexto es determinado de la siguiente manera:

> *Las características de los usuarios, tareas, equipos (hardware, software y materiales), y entornos tanto físicos como sociales donde un producto es utilizado.*

Siendo conscientes de dicha definición, adaptamos la misma al campo de este trabajo junto con los factores definidos por Biel et al. [Biel+10]. Como se aprecia en la figura 3.4, definimos el contexto mediante los cinco pilares básicos que conforman la interacción con aplicaciones móviles: aplicación, usuario, dispositivo, entorno y tarea. Mediante estos pilares podremos generar todas las descripciones de cómo deben realizarse las pruebas.

Figura 3.4 Pilares del modelo de contexto

A continuación, procedemos a describir más en detalle cada uno de ellos.

3.2.1.1. APLICACIÓN

En primer lugar, se dispone del pilar que describe la aplicación a evaluar.

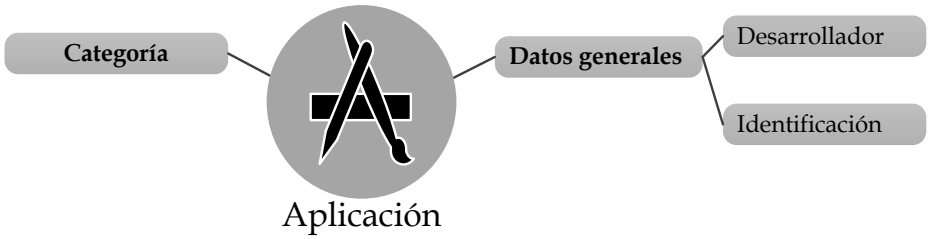

Figura 3.5 Conjuntos de atributos de la aplicación

Como muestra la figura 3.5, dicho pilar consta de dos principales grupos de atributos: categoría de la aplicación y los datos generales.

- La *categoría de la aplicación* muestra el tipo de aplicación al que pertenece la aplicación a evaluar. Esta categorización está basada en la propuesta por Google mediante su mercado de aplicaciones[26]. Se muestran 27 posibles categorías de aplicación: bibliotecas y demos, compras, comunicación, cómics, deportes, educación, entretenimiento, estilo de vida, finanzas, fondos animados, fotografía, herramientas, libros y obras de consulta, medicina, multimedia y vídeo, música y audio, negocios, noticias y revistas, personalización, productividad, salud y bienestar, sociedad, tiempo, transporte, viajes y guías, widgets, y finalmente juegos.
- El conjunto de atributos de los *datos generales* muestra datos con fines identificativos y descriptivos. Por un lado, muestra los datos del desarrollador de la aplicación mediante su nombre y correo electrónico. Por otro lado, se presenta el nombre de la aplicación, el identificador único y la descripción de la misma.

[26] https://play.google.com/store/apps

3.2.1.2. USUARIO

En segundo lugar, se dispone del usuario de pruebas. Dicho pilar describe las características de los usuarios que deben realizar las pruebas.

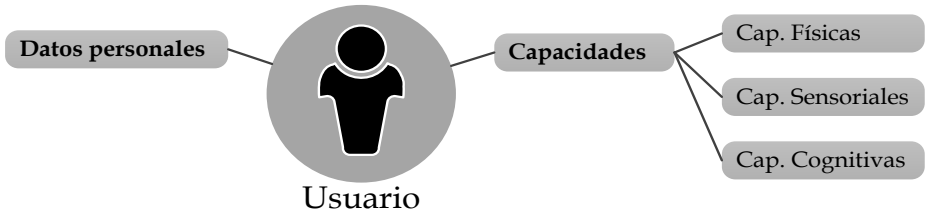

Figura 3.6 Conjuntos de atributos del usuario

Como se muestra en la figura 3.6, describimos este pilar mediante dos conjuntos de atributos: datos personales y capacidades.

- Dentro del conjunto de *datos personales* se describen los datos más comunes de un usuario. Éstos son el nombre o alias, fecha y lugar de nacimiento, altura, peso y género. Este último será considerado como importante criterio de agrupación al haber mostrado diferencias significativas en varios trabajos [Simon00, Cyr+05, Djamasbi+07].
- Los atributos del conjunto de *capacidades* son aquellos que definen el nivel de destreza o conocimiento que posee el usuario para ejecutar acciones concretas. En este caso se distinguen principalmente tres subconjuntos: las capacidades físicas, capacidades sensoriales y capacidades cognitivas. Las capacidades físicas incluyen las habilidades motrices como la lateralidad de un usuario. Las capacidades sensoriales describen principalmente los niveles de discapacidad sensorial como sordera, mudez, ceguera o daltonismo. Finalmente, las capacidades cognitivas refieren a lo relacionado con el procesamiento de la información recibida. Forman parte de este subconjunto tanto la comprensión o nivel de un idioma, como la habilidad de interpretación cartográfica.

A modo resumen, se muestra un ejemplo de los atributos de los conjuntos datos personales (DP) y capacidades (C) que componen el pilar usuario mediante la tabla 3.3.

Conjunto (subconjunto)	*Atributo*	*Valor ejemplo*
DP	*Alias*	*usuario1*
DP	*Fecha de nacimiento*	*23-05-1984*
DP	*Lugar de nacimiento*	*37.429,-122.128*
DP	*Altura*	*165cm*
DP	*Peso*	*65.5kg*
DP	*Género*	*Masculino*
C(física)	*Lateralidad*	*Diestro*
C(sensorial)	*Visión*	*Daltonismo acromático*
C(cognitiva)	*Nivel de castellano*	*Nativo*
C(cognitiva)	*Nivel de inglés*	*B2*

Tabla 3.3 Ejemplo de atributos del pilar usuario del modelo de contexto

3.2.1.3. DISPOSITIVO

Como apreciamos en la definición de contexto expuesta anteriormente, unos de los principales componentes es el conjunto de equipos. En este caso, asumimos que los equipos son los dispositivos móviles. Definimos dispositivo móvil como:

> *Artefacto de pequeño o medio tamaño, portátil e inalámbrico donde reside la aplicación a evaluar y que es usado por el usuario de pruebas para la realización de las tareas.*

Figura 3.7 Conjuntos de atributos del dispositivo

Como vemos en la figura 3.7, para definir el dispositivo hemos dividido este pilar en varios conjuntos de atributos.

- El conjunto de atributos denominados como *datos generales*, son los atributos referentes a las características de

producción, características físicas y de posición del dispositivo. Atributos como las dimensiones, peso, color y aceleración del terminal componen este subconjunto.

- El conjunto de atributos *entradas* son aquellos que definen la entrada de datos en la interacción del usuario con el dispositivo. Dentro de éste entran el número de pulsaciones simultáneas detectadas en una pantalla táctil, la disposición de altavoz, el tipo de teclado, etc.
- Al contrario que el subconjunto de atributos anterior, el conjunto *salidas* posee los atributos que definen la salida de datos y por lo tanto, el modo mediante el cual el usuario recibe la información. Los atributos más importantes en este subconjunto son los referentes a la descripción de la pantalla del dispositivo: dimensiones de la pantalla, modelo, resolución. También es importante que mencionemos atributos referentes al audio: volumen del sistema, conexión de auriculares, manos libres Bluetooth, etc.
- Las *características software* muestran la descripción del sistema operativo del terminal. Aquí destacamos el tipo de sistema operativo del sistema, la versión y el idioma del mismo.
- La *batería* de los dispositivos móviles modernos, es una de las principales causas de disconformidad. Dicha disconformidad es propiciada por la poca duración de la misma. Por ello, asignamos un conjunto de atributos referentes a la misma. Además del nivel de carga, es interesante describir el estado de la fuente de alimentación: si simplemente está descargándose la batería, si está cargándose en una toma de corriente,…
- Finalmente y no por ello menos importante, las *conexiones* del dispositivo es un conjunto de atributos que describe los medios de conexión que el terminal tiene habilitados. Dentro de este conjunto entran el estado de las interfaces de comunicación (WiFi, Bluetooth, etc…).

Al igual que el anterior pilar, se muestra un ejemplo de los atributos de los conjuntos que describen el pilar dispositivo mediante la tabla 3.4.

Conjunto	*Atributo*	*Valor ejemplo*
Datos generales	*Modelo*	*Nexus 5*
Datos generales	*Fabricante*	*LG*
Datos generales	*Dimensión (Altura)*	*137.9 mm*
Datos generales	*Dimensión (Anchura)*	*69.2 mm*
Datos generales	*Dimensión (Largura)*	*8.6 mm*
Entradas	*Puntos táctiles*	*5*
Entradas	*Estado micrófono*	*Encendido*
Salidas	*Volumen del sistema*	*50%*
Salidas	*Auriculares*	*Conectados*
Software	*Versión del S.O.*	*5.1*
Software	*Idioma*	*ES*
Batería	*Nivel de carga*	*88%*
Batería	*Estado*	*Enchufada a A/C*
Conexiones	*Estado interfaz datos móviles*	*Conectada*
Conexiones	*Estado interfaz WiFi*	*Conectada*

Tabla 3.4 Ejemplo de atributos del pilar dispositivo del modelo de contexto

3.2.1.4. ENTORNO

El tercer pilar, cuya importancia en esta tesis es esencial, se encuentra el entorno. Interpretamos como entorno lo siguiente:

> *El entorno es el conjunto de condiciones que rodean al usuario de pruebas y su dispositivo en el desarrollo de las tareas.*

Tomando como base la clasificación mostrada en el estándar ISO 9241-11 [ISO98], dichas condiciones son clasificadas en tres principales tipos de entorno. Además se añade un grupo de descripción general para poder identificar el entorno de un modo genérico.

Figura 3.8 Conjuntos de atributos del entorno

Como se muestra en la figura 3.8, detectamos el grupo de datos generales y tres tipos de entorno: entorno físico, técnico y social.

- El grupo de *datos generales* dispone solo de dos atributos. El nombre del entorno y la descripción genérica del mismo. Dichos atributos deben ser autoexplicativos.
- El *entorno físico* describe las condiciones y circunstancias relativas al espacio físico que rodea al usuario de pruebas. Este conjunto está descrito tanto por atributos de localización (longitud, latitud, altitud) como medioambientales (iluminancia, ruido, temperatura, humedad, condición meteorológica,…).
- El *entorno técnico* describe las condiciones de los sistemas con los que el dispositivo móvil se comunica. Dentro de este conjunto destacamos el tipo de red con el que el dispositivo se comunica (WiFi, GPRS, UMTS, Bluetooth…) y la velocidad del mismo. Es fácil confundir el entorno técnico con las conexiones del pilar dispositivo, por ello aclaramos que el entorno técnico describe los componentes fuera del límite del dispositivo y las conexiones del dispositivo el estado de conexión ceñido a las interfaces del mismo. Por ejemplo, podemos disponer de la interfaz de datos móviles del dispositivo (conexión del dispositivo) habilitado conectada a celdas tipo GPRS (entorno técnico).
- Finalmente, el *entorno sociocultural* describe las condiciones tanto sociales como culturales. Dentro de los atributos de este grupo destacan los idiomas del entorno y la estructura social organizacional del momento (amigos, familia, compañeros de trabajo…).

Habiendo explicado y estudiado los diferentes tipos de entorno, es importante mencionar que dentro del desarrollo de tareas y el uso de aplicaciones móviles, consideramos el entorno sociocultural como el menos dinámico. Con ello no pretendemos añadir que no sesgue la interacción pero sí asumimos que lo van a hacer en mucha menos medida que el entorno técnico y físico.

A continuación, mediante la tabla 3.5 mostramos un ejemplo de los atributos de los conjuntos que describen el pilar entorno.

Conjunto	*Atributo*	*Valor ejemplo*
Datos generales	*Nombre*	*Tumbado en casa*
Datos generales	*Descripción*	*Estando en casa, el usuario está tumbado…*
Entorno físico	*Localización*	*43.2687485, -2.9397425*
Entorno físico	*Ruido*	*46dB*
Entorno físico	*Iluminancia*	*100 lux*
Entorno técnico	*Conexión Bluetooth*	*A2DP(Advanced Audio Distribution Profile)*
Entorno técnico	*Tipo de red*	*WiFi*
Entorno técnico	*Velocidad de red*	*54Mbps*
Entorno social	*Idioma del entorno*	*ES*
Entorno social	*Círculo social actual*	*Padre, madre y hermano.*

Tabla 3.5 Ejemplo de atributos del pilar entono del modelo de contexto

3.2.1.5. TAREA

El último pilar del modelo de contexto desarrollado es la tarea. La definición formal de tarea asumida es la siguiente:

> *Una tarea es el conjunto de actividades realizadas para lograr los objetivos que se buscan al utilizar la aplicación a evaluar.*

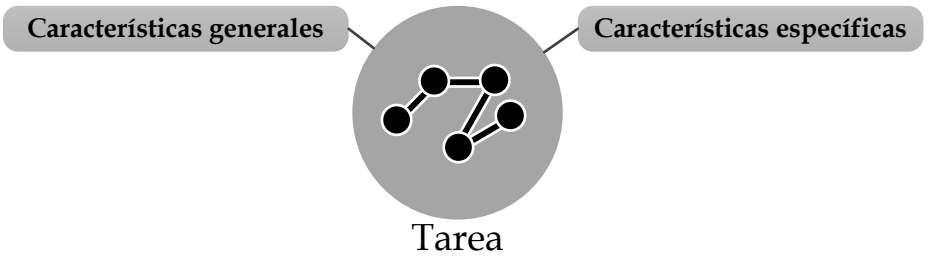

Figura 3.9 Conjuntos de atributos de la tarea

Como se ve en la figura 3.9, este pilar dispone de dos conjuntos de atributos: los que definen las características generales de la tarea y los que definen las características específicas.

- Dentro del conjunto de *características generales* diferenciamos dos principales atributos: nombre y tipo. El atributo nombre define el identificador principal de la tarea. Además de identificar a la tarea, dicho atributo tiene como principal objetivo describir. Por ello, se define

mediante una taxonomía basada en la definición de un verbo seguido de un nombre como objeto directo (p.ej. publicar anuncio). Para disponer de un criterio de clasificación más genérico, usamos como base la clasificación para tareas con dispositivos móviles táctiles descrita en el trabajo de Yáñez Gómez et al. [Yañez+14]. Basándonos en dicho trabajo, proporcionamos una clasificación de seis tipos de tareas: búsqueda, navegación, comunicación, transacción, diversión y otros. El tipo búsqueda refiere a tareas cuyos objetivos están relacionados con encontrar un elemento del modelo de la aplicación; un ejemplo es buscar una canción en una aplicación de reproducción de música. Las tareas de tipo navegación tienen como objetivo el desplazamiento del usuario dentro de la aplicación; un ejemplo de este tipo de tarea es abrir el menú de preferencias. El tipo comunicación define las tareas que están relacionadas con el intercambio de mensajes con otros usuarios; un ejemplo es el envío de un comentario de valoración de fotografía en una red social de publicación de fotos. El tipo transacción abarca a las tareas relacionadas con movimientos de elementos de valor dentro de la aplicación a evaluar (p.ej. puntos de un sistema de fidelización, dinero en compra electrónica,…); un ejemplo es comprar un libro en una aplicación de compras. Tareas de tipo diversión son aquellas cuyo objetivo no tiene un criterio de finalización concreto y son puramente para pasar el tiempo; un ejemplo es jugar a un juego dentro de una aplicación móvil. Finalmente, las tareas que no cumplan ningún criterio expuesto en los demás tipos, forma parte del tipo otros.

- El conjunto de atributos ligados a las *características específicas* son los referentes a una tarea de una aplicación a evaluar concreta. Define aquellos atributos que detallan el objetivo específico de la tarea y el conjunto de pasos que el usuario de pruebas debe realizar para lograr dicho objetivo. Dentro de este conjunto de atributos destacamos el nombre detallado e instrucciones. Ligado a las instrucciones y

miembro de este conjunto, se menciona el modelo de interacción, detallado más adelante (ver apartado 3.2.2).

A continuación, mediante la tabla 3.6 mostramos un ejemplo de los atributos que describen el pilar tarea.

Conjunto	*Atributo*	*Valor ejemplo*
Características generales	*Nombre*	*Buscar artículo*
Características generales	*Tipo*	*Búsqueda*
Características específicas	*Nombre detallado*	*Comprar de artículo deportivo*
Características específicas	*Instrucciones*	*Realizar la búsqueda del artículo cuyo título sea 'Bicicleta Y-020594'.*
Características específicas	*Modelo de interacción*	*(no definido)*

Tabla 3.6 Ejemplo de atributos del pilar tarea del modelo de contexto

Concluyendo con la descripción del modelo de contexto, debemos remarcar que el pilar tarea dispone del modelo de interacción, que ha sido superficialmente mencionado para ofrecer una descripción explícita a continuación.

3.2.2. MODELO DE INTERACCIÓN

El modelo de interacción tiene como principal objetivo representar la interacción del usuario de pruebas con la aplicación a evaluar. Para ello, recuperamos la definición de tarea ofrecida en el modelo de contexto. Dentro de la misma destaca el conjunto de pasos y actividades que son realizados para completar el objetivo de la misma. Cada uno de los pasos o actividades que el usuario realiza con la aplicación genera eventos. Entendemos un evento del modelo de interacción dentro de este trabajo como lo siguiente:

> *Un evento es un suceso ligado al cambio de estado de una tarea y el avance de la misma mediante la interacción del usuario de pruebas.*

Como se muestra en la definición, dentro de los diferentes estados que una tarea puede adquirir, se distinguen los estados que refieren al estado general de la tarea (si está siendo realizada) y el avance dentro de la misma (si se van realizando los pasos para lograr su objetivo). De un modo genérico, para describir correctamente la ocurrencia de los diferentes eventos, asignamos a cada uno el momento exacto en el que se ha producido, el motivo

por el cual se ha originado, su clasificación y el tipo dentro de la misma. Esta clasificación se realiza mediante dos principales grupos: eventos de tarea y eventos de interacción.

Gracias a la definición de estos eventos, somos capaces de describir dos situaciones distintas. Por un lado, *podemos describir cómo un usuario de pruebas desempeña una tarea concreta en la fase de ejecución de las pruebas*. Por otro lado, mediante el camino de interacción correcta descrito en el apartado 3.2.2.3, *podemos definir cómo debe completarse una tarea satisfactoriamente*. Para lograr una mejor comprensión, a continuación comenzamos la descripción en detalle de ambos tipos de eventos y el camino de interacción correcta.

3.2.2.1. EVENTOS DE TAREA

Mediante los eventos de tarea y sus estados representamos las diferentes etapas por las que pasa una tarea. Concretamente, los eventos de tarea son aquellos *sucesos originados por un cambio en el estado de la tarea*. Como se muestra en la figura 3.10, una tarea puede adquirir cuatro principales estados: no iniciada, pausada, en ejecución y finalizada.

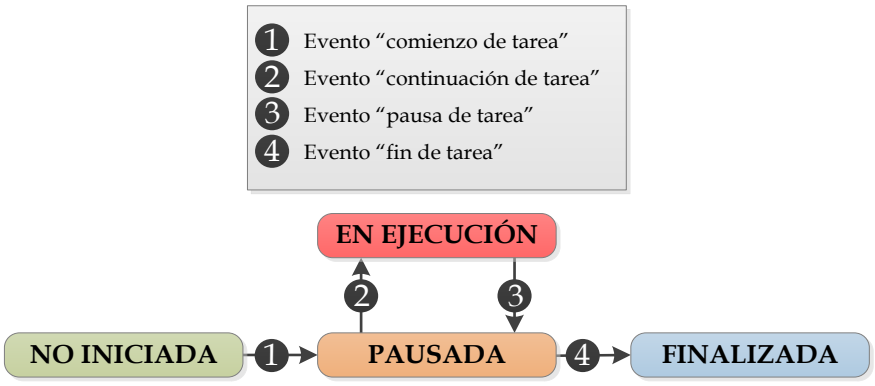

Figura 3.10 Diagrama de estados de la tarea y eventos de tarea

El primer estado por el cual pasa una tarea es el estado *no iniciada*. Dicho estado indica que la tarea no ha dado comienzo, por lo que el usuario todavía no ha interactuado con la aplicación para

realizarla pero comienza a tomar consciencia de la misma (p.ej. leyendo el enunciado de la tarea).

Una vez el usuario de pruebas decide realizar la tarea, da comienzo la misma. Por lo tanto, la tarea pasa a un nuevo estado denominado *pausada* y debido a este cambio se genera el primer evento: *comienzo de tarea*. Este nuevo estado muestra que la tarea ha sido iniciada pero el usuario no está interactuando directamente con la aplicación para completarla. Estas situaciones pueden darse en dos casos. Por un lado, cuando se ha generado el evento *comienzo de tarea* y el usuario de pruebas todavía no puede interactuar directamente con la aplicación (p.ej. buscando el icono de la aplicación evaluada para lanzarla o la espera a su carga inicial). Por otro lado, cuando se ha generado el evento *pausa de tarea* porque el usuario abandona la aplicación a evaluar a causa de cualquier interrupción externa (p.ej. llamada entrante, encontrarse con un conocido en la calle…). Cuando una tarea se encuentra en este estado, el usuario de pruebas tiene opción de continuarla o finalizarla.

En el caso de que el usuario piense que ya ha logrado el objetivo de la tarea o decida abandonarla, ésta adquiere el último estado posible llamado *finalizada*, cuyo cambio al mismo implica la generación del evento *fin de tarea*.

Si el usuario de pruebas elije continuar la tarea, éste restaurará la aplicación evaluada y se generará el evento *continuación de tarea* ya que la tarea adoptará el estado *en ejecución*. En este estado, el usuario de pruebas puede interactuar con la aplicación evaluada. Debido a esto, en este estado es donde realmente se dan los pasos para avanzar en la realización de la tarea y es donde realmente vamos a monitorizar la interacción con la aplicación.

Complementando la descripción, resumimos mediante la figura 3.11 el flujo básico de los cambios de estado de una tarea mediante un ejemplo con una interrupción de una llamada de teléfono.

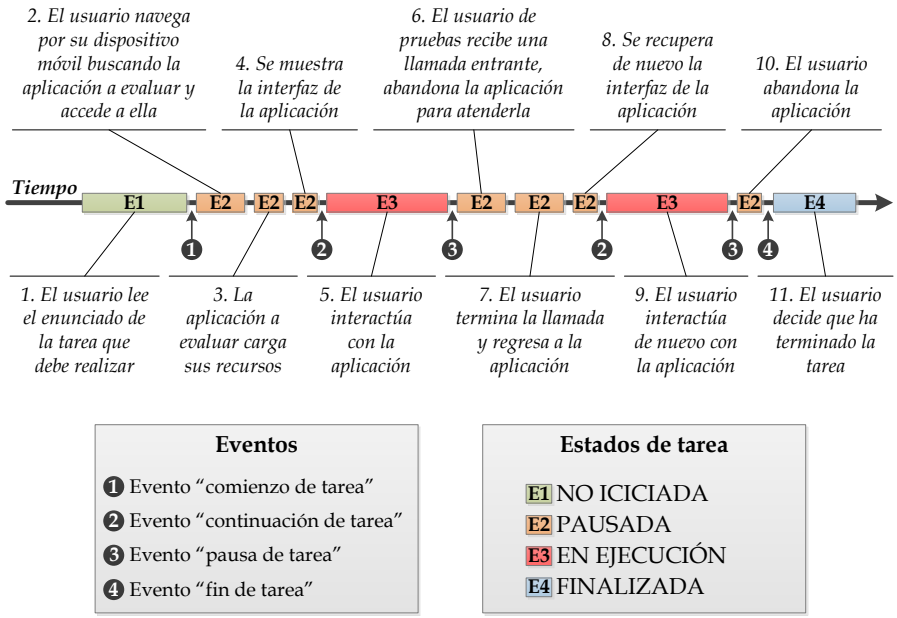

Figura 3.11 Ejemplo del flujo básico de los cambios de estado de una tarea

Dichos eventos son descritos mediante dos simples atributos: el tiempo en el que han sido originados y el tipo de evento. A modo de ejemplo, se muestra mediante la tabla 3.7 la descripción de los estados de tarea originados en el ejemplo de la figura 3.11.

Evento	*Marca temporal (s)*	*Tipo*
1	*0*	*Comienzo tarea*
2	*0,98*	*Continuación de tarea*
3	*2,56*	*Pausa de tarea*
4	*15,13*	*Continuación de tarea*
5	*23,03*	*Pausa de tarea*
6	*23,56*	*Fin de tarea*

Tabla 3.7 Ejemplo de atributos de eventos de tarea

Dentro de los estados de tarea estudiados, destaca el estado en el que la tarea se encuentra en ejecución, donde el usuario realiza la verdadera interacción con la aplicación a evaluar. Dentro de este estado es donde se producen los eventos de interacción, explicados a continuación.

3.2.2.2. EVENTOS DE INTERACCIÓN

Mediante los eventos de tarea, hemos representado de un modo teórico los diferentes estados que pueden adquirir las mismas y las diferentes secuencias posibles. En este caso, definimos eventos

de interacción como *los sucesos de interacción que se producen en la comunicación entre el usuario de pruebas y la aplicación evaluada*.

Para describir dichos sucesos, se definen seis atributos. En primer lugar, se muestra el tiempo exacto en el que se ha producido el evento. Referente a la interfaz, se define la interfaz gráfica mostrada en ese momento y el objeto de la interfaz que ha disparado el mismo. Además, una breve descripción del evento y el tipo de evento de interacción dentro de la tarea.

Marca temporal (s)	*Interfaz*	*Objeto*	*Valor*	*Tipo*	*Descripción*
0	*Interfaz A*	*BTN_5*		*Paso*	*Pulsar 'Nueva Búsqueda'*
1,87	*Interfaz B*	*TXT_8*	*"título"*	*Paso*	*Introducir Título*
16,01	*Interfaz B*	*SPN_10*	*"sub_1"*	*Error*	*Seleccionar Subcategoría*
16,92	*Interfaz B*	*SPN_9*	*"cat_3"*	*Paso*	*Seleccionar Categoría*
17,97	*Interfaz B*	*SPN_10*	*"sub_1"*	*Paso*	*Seleccionar Subcategoría*
19,26	*Interfaz B*	*SPN_11*	*"León"*	*Paso*	*Seleccionar Provincia*
23,01	*Interfaz B*	*BTN_12*		*Fin*	*Pulsar 'Buscar'*

Tabla 3.8 Ejemplo de atributos de eventos de interacción

Un evento de interacción puede ser de tres principales tipos: *tipo paso*, *tipo error* y *tipo fin*. El *tipo paso* es un evento que se genera cuando hay una interacción que propicia un avance hacia el objetivo de una tarea. El *tipo error* se genera cuando el usuario realiza una interacción con la aplicación evaluada errónea. Finalmente, el *tipo fin* indica que se ha originado un evento de interacción que propicia lograr el objetivo de la tarea y por lo tanto, completarla.

3.2.2.3. CAMINO DE INTERACCIÓN CORRECTA

Para deducir el tipo de evento de interacción que es, debemos tener definido qué secuencia de eventos es la correcta para completar la tarea. Logramos dicho objetivo mediante la especificación del camino de interacción correcta, definido de la siguiente manera:

> *Un camino de interacción correcta es el conjunto de secuencias de eventos de interacción que deben ser realizadas para el logro del objetivo de la tarea a la que pertenecen.*

Gracias a este elemento, comprobaremos si la interacción se desarrolla de un modo correcto hasta finalizar la tarea.

Para realizar correctamente ciertas tareas, debemos ser conscientes del orden de los eventos de interacción. En ciertas interacciones el orden en el que los diferentes eventos se generan no importa, como en un formulario de acceso a una red privada (se puede introducir primero la contraseña y después el nombre de usuario). En cambio, en otros casos sí que importa, como al rellenar un formulario de dirección postal (antes de seleccionar una ciudad, debemos seleccionar la provincia para cargar las ciudades correspondientes). En este caso, definimos que el evento "selección de ciudad" depende del evento "selección de provincia".

Para representar dicho camino y las dependencias entre los eventos de interacción, se han revisado trabajos que representan conceptos similares. En la literatura relacionada, se han detectado numerosos trabajos que representan la interacción mediante diferentes tipos de modelos, tomando como base los grafos dirigidos y las máquinas de estados finitos. Dentro de estos trabajos, destaca el objetivo común de realizar una generación automática de casos de prueba. Como en los modelos de grafos de flujos de eventos [Memon+01], grafos de interacción semántica de eventos [Yuan+10] o los de interacción de eventos [Katayama+95]. Para ello, los eventos son asociados a funciones y operaciones de la interfaz gráfica.

Estos modelos, aunque manifiestan una clara facilidad para interpretar las secuencias gráficamente, muestran excesiva complejidad para representar los diferentes pasos, ya que muchas de ellas intentan representar todas las posibles interacciones. Como el enfoque de este trabajo es más restrictivo y solo intenta definir el camino correcto de interacción, se presenta un modelo basado en grafos dirigidos. En éste, *los nodos representan los diferentes eventos* necesarios para completar la tarea y *las aristas las dependencias que existen* entre ellos.

Para aclarar la generación del camino de interacción correcta, supongamos las dos interfaces mostradas en la figura 3.12.

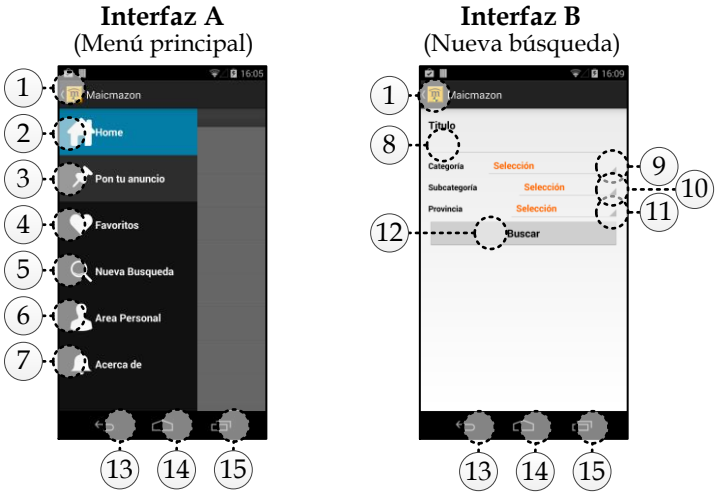

Figura 3.12 Ejemplo de interfaces para generación de camino de interacción correcta

Éstas forman parte de un pequeño fragmento de una aplicación en la que se pueden realizar búsquedas de productos. Si disponemos de la interfaz "A" y pulsamos el botón "Nueva Búsqueda" (evento 5), la aplicación mostraría la interfaz "B". Además, disponemos de una tarea cuyo enunciado es: realizar la búsqueda de un producto cuyo título es "título", categoría "cat_3", subcategoría "sub_1" y provincia "León".

Los pasos que el usuario de pruebas debe seguir para realizar dicha tarea son los siguientes. Primero, debe pulsar el botón "Nueva Búsqueda" (evento 5) en la interfaz "A". Después, debe introducir el texto "título" en el campo título de la interfaz "B", con ello se genera el evento 8. Posteriormente, debe seleccionar el valor "cat_3" en la lista categoría (evento 9) y el valor "sub_1" en la lista de subcategorías (evento 10). Finalmente, debe seleccionar el valor "León" en la lista de provincias (evento 11) y pulsar el botón "Buscar" (evento 10).

Habiendo explicado los pasos que deben realizarse para cumplir el objetivo de la tarea, el grafo del camino de interacción correcta correspondiente se refleja en la figura 3.13.

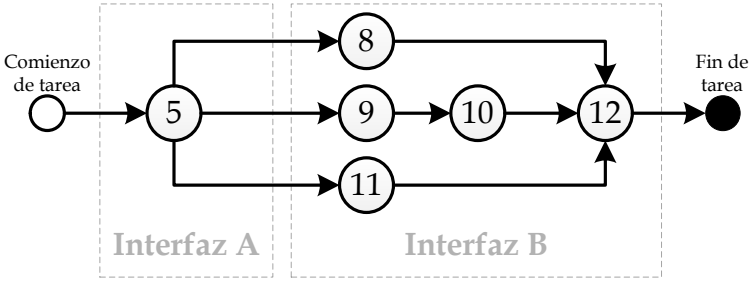

Figura 3.13 Ejemplo de grafo de camino de interacción correcta

Como apreciamos en el mismo, es necesario pulsar el botón "Nueva Búsqueda" antes de rellenar ningún campo (eventos 8, 9 y 11 dependen de 5). Además, distinguimos que rellenar el título, seleccionar la categoría y la provincia no importa el orden. En cambio antes de seleccionar la subcategoría, sí es obligatorio tener seleccionada la categoría (evento 10 depende de 9). Para pulsar el botón "Buscar" debemos tener correctamente cumplimentados los campos anteriores (evento 12 depende de 8, 10 y 11).

Habiendo definido y explicado el camino de interacción correcta, aclaramos que pulsar cualquier botón que se salga del mismo, originará eventos de tipo error. Además, cualquier evento originado dentro del camino de interacción correcta que no cumpla con el valor que debe introducir (p.ej. se selecciona "Vizcaya" en vez de "León" dentro de la provincia) también originará un evento de tipo error. Por otro lado, el pulsar el botón "Buscar" (evento 12) cuando todas sus dependencias han sido cumplidas satisfactoriamente (los eventos de los que depende han sido originados con los valores correctos), origina un evento de tipo fin.

Concluimos la explicación con el resumen del criterio de decisión para definir el tipo de evento de interacción con la figura 3.14.

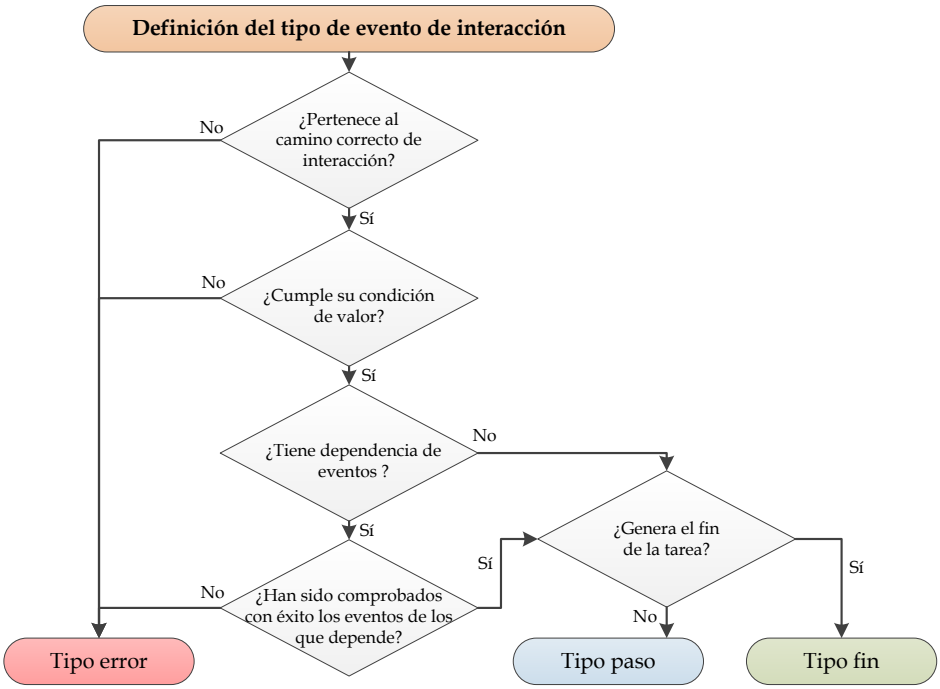

Figura 3.14 Criterio de decisión para definir el tipo de evento de interacción

3.2.3. MODELO DE ANÁLISIS

Habiendo descrito los dos modelos anteriores disponemos del conocimiento suficiente para completar el modelo de análisis. Este modelo tiene como objetivo representar los resultados que se obtienen mediante las pruebas. Consta principalmente de dos componentes: componente sumativo y componente formativo. Dichos componentes nos servirán de ayuda para presentar los resultados finales de la evaluación.

3.2.3.1. COMPONENTE SUMATIVO

Este tipo de componente nos ayuda a definir los resultados de carácter sumativo. Dicho de otro modo:

> *El objetivo que tiene este componente es describir la usabilidad de la aplicación a evaluar mediante las métricas que la componen.*

Para realizar un correcto modelado del componente mediante la adherencia de métricas, hemos realizado una priorización de las

dimensiones de usabilidad más relevantes en base al estudio realizado por Baharuddin et al. [Baharuddin+13]. En él se identificaron 18 dimensiones para buscar la respuesta a la siguiente pregunta de investigación:

> *¿Cuáles son las dimensiones de usabilidad necesarias para las aplicaciones móviles?*

Es importante añadir que en dicho trabajo se analizan 9 estudios empíricos, de los cuales uno de ellos [Coursaris+06] realizó un análisis previo de 45 estudios más. Las cinco primeras dimensiones clasificadas son las que han sido evaluadas en más del 50% de los trabajos analizados: la efectividad, eficiencia, satisfacción, utilidad y estética.

Saltando la frontera del campo específico de las aplicaciones móviles, continuamos el análisis retomando las definiciones de usabilidad que proporcionan los estándares formales anteriormente mencionados (ver apartado 2.1.2). Habiendo identificado las cinco dimensiones anteriores, analizamos su importancia en los estándares revisados.

Como se muestra en la figura 3.15, detectamos que hacen especial mención a las dimensiones de efectividad, eficiencia y satisfacción dentro de la definición de usabilidad.

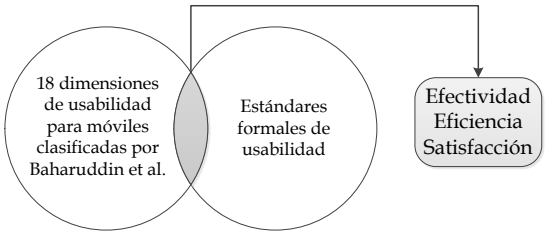

Figura 3.15 Intersección entre las 18 dimensiones de Baharuddin et al. y los estándares formales

Por lo tanto, habiendo realizado una búsqueda de las dimensiones más descriptivas, concluimos lo siguiente:

> *Las dimensiones más relevantes y utilizadas para describir la usabilidad son la efectividad, eficiencia y satisfacción.*

Dando un paso más en el correcto enfoque sumativo, se estudian los modos de medición y cálculo de estas tres dimensiones. Actualmente, dentro de la división 2502n del estándar SQuaRE [ISO14] donde se especificarán las mediciones de calidad, se están revisando las métricas que componen la calidad en uso en el estándar ISO/IEC 9126-4 [ISO04]. Desafortunadamente, como esta división se encuentra en desarrollo, hemos optado por describir las dimensiones con este último estándar. Explorando su especificación, la calidad en uso responde a cuatro dimensiones: efectividad, productividad, seguridad y satisfacción. En ella se incluye una explícita definición de las métricas que conforman las mismas. Tomando como base dicho estándar, se muestran las métricas que forman el componente sumativo del modelo de análisis, todas ellas compuestas por variables tomadas a nivel de tarea realizada.

Dimensión	*Métrica*	*Fórmula*	*Pregunta*
Efectividad (completitud)	*Completitud de la Tarea(CT)*	Tc	*¿Qué proporción de la tarea es completada correctamente?*
Efectividad (exactitud)	*Frecuencia de Error(FE)*	$\frac{E}{Tt}$	*¿Cuál es la proporción de errores relativos al tiempo de la tarea?*
Eficiencia	*Tiempo de Tarea (TT)*	Tt	*¿Cuánto tiempo es invertido para lograr realizar la tarea?*
Satisfacción	*Escala de Satisfacción (ES)*	Q	*¿Cómo de satisfecho está el usuario realizando la tarea?*

Tabla 3.9 Métricas del componente sumativo

Una vez mostradas mediante la anterior tabla, a continuación detallamos su significado y composición.

- La dimensión de la *efectividad* se define de la siguiente manera:

  > *La capacidad del producto software para facilitar a los usuarios alcanzar metas específicas con exactitud y completitud.*

Por ejemplo, si el objetivo deseado es escribir un mensaje en un terminal móvil, la exactitud se puede medir por el número de palabras mal escritas, y la completitud por el número de palabras escritas. Dicha dimensión, como se especifica en la tabla 3.9, la detallamos con dos principales métricas: completitud de la tarea (CT) y frecuencia de error (FE).

La *completitud de la tarea (CT)* mide el nivel de éxito que el usuario consigue en la realización de la misma, cubriendo la completitud. Esta medición supone que las tareas se pueden realizar sin la posibilidad de ser completadas parcialmente: o se completa o no lo hace. En este caso, la medición es un simple valor: 1 en el caso de lograr los objetivos de la misma y 0 en caso contrario. Para deducir la completitud de la tarea, habiendo explicado el modelo de interacción referente, *basta con asegurarse que se ha producido un evento de interacción de tipo fin* (ver apartado 3.2.2.2).

En segundo lugar, la *frecuencia de error (FE)* mide el número de veces que se comete un error en un periodo determinado, cubriendo la exactitud. Se calcula dividiendo el número de errores cometidos (E), por el tiempo de la tarea (Tt). *El número de errores es obtenido al contabilizar los eventos de interacción de tipo error* durante la tarea. *El tiempo de la tarea*, definido posteriormente, *se obtiene contabilizando el tiempo en el que la tarea se encuentra en estado "en ejecución"* (ver apartado 3.2.2.1). Al contrario que la otra métrica, el resultado es dado mediante una escala continua donde el valor 0 es el caso óptimo.

Es importante añadir que en términos prácticos, la eficacia no tiene en cuenta cómo los objetivos se han alcanzado, sólo la medida en la que han sido logrados.

- La siguiente dimensión es la *eficiencia*. Este término, referido en la propia especificación como productividad, muestra la siguiente definición:

> *La capacidad del producto software para invertir la cantidad apropiada de recursos en relación a la eficacia alcanzada en un contexto específico de uso.*

Siguiendo con el ejemplo cuyo objetivo es escribir un mensaje en un terminal móvil, podemos medir la eficiencia midiendo el tiempo necesario (interpretado éste como un recurso) en relación al número de palabras del mensaje escritas. Aunque en el estándar muestran diversas métricas como la efectividad de tarea, productividad económica, proporción productiva o la eficiencia relativa a usuario experto, tomamos simplemente el *tiempo de tarea (TT)*. Esta métrica de rendimiento es la más extendida y mide el tiempo (*Tt*) que el usuario tarda en realizar la tarea (véase la tabla 3.2), cuyo cálculo ha sido explicado en la descripción de la dimensión anterior.

- Finalmente, la dimensión *satisfacción* se define del siguiente modo:

> *La capacidad del producto software para satisfacer a los usuarios en un contexto específico de uso.*

Puede especificarse y medirse mediante la valoración subjetiva usando cuestionarios con escala psicométrica. Particularmente, en este trabajo (véase tabla 3.9) se utiliza una métrica: *la escala de satisfacción (ES)*. Esta se mide mediante un cuestionario que produce escalas psicométricas y la satisfacción del usuario (*Q*) con la tarea que ha realizado. Hay diversos cuestionarios como el SUS (System Usability Scale) [Brooke96], cuyo objetivo es obtener una medición genérica en base a 10 preguntas; o el cuestionario QUIS (Questionnaire for User Interface Satisfaction) [Chin+88], que realiza una medición centrada en la satisfacción pero excesivamente extensa (27 preguntas).

Buscando un equilibrio entre el número de preguntas y el enfoque de medir sólo la satisfacción, hemos optado por el

uso del cuestionario llamado *USE (Usefulness, Satisfaction, and Ease of Use)* [Lund08]. Dicho cuestionario dispone de 30 preguntas distribuidas en 4 categorías (*Usefulness, Satisfaction, Ease of Use* y *Ease of Learning*) respondidas con una escala Likert de 7 puntos.

En el caso de este trabajo tomamos como base las 7 preguntas que corresponden a la categoría *Satisfaction*. Mediante la tabla 3.10 se muestran las preguntas adaptadas.

Pregunta original USE	*Pregunta adaptada*
I am satisfied with it	*Estoy satisfecho con la tarea que he realizado*
I would recommend it to a friend	*Recomendaría a un amigo usar esta aplicación para realizar esta tarea*
It is fun to use	*Me he divertido llevando a cabo esta tarea*
It works the way I want it to work	*Funciona del modo que espero que funcione cuando efectúo esta tarea*
It is wonderful	*Me ha parecido una aplicación maravillosa para efectuar esta tarea*
I feel I need to have it	*Siento que necesito tener esta aplicación para hacer este tipo de tarea*
It is pleasant to use	*Para llevar a cabo esta tarea, es una aplicación agradable*

Tabla 3.10 Preguntas del cuestionario USE adaptadas para calcular la satisfacción

Una vez hemos definido las diferentes métricas, es importante destacar que todas son calculadas a nivel de tarea. Esto nos permite poder exponer los datos tanto a nivel individual, como colectivo realizando las agrupaciones en función de los tres criterios de segmentación definidos en la fase de análisis: por tipo de usuario de pruebas, tarea y entorno (véase apartado 3.1.3).

Para realizar el cálculo de los datos descriptivos a nivel colectivo distinguimos por un lado los cálculos relacionados con la completitud de tarea y por otro lado el resto.

- En el caso de la completitud de la tarea, al tratarse de datos que forman una distribución binomial, no se considera relevante el cálculo de su desviación típica ni mediana. En cambio, sí se calculan el intervalo de confianza en base a la

distribución binomial. Con ello, describimos entre qué valores caerá el porcentaje real de tareas realizadas de un modo satisfactorio.

- Por otro lado, la frecuencia de error, el tiempo de tarea y la escala de satisfacción dispondrán de la media, mediana, desviación típica e intervalos de confianza en base a la distribución T de Student, al no poder garantizar en la mayoría de las muestras un número de casos igual o superior a 30.

Finalmente, concluimos este apartado con un ejemplo de datos del componente sumativo del modelo. Supongamos que disponemos de los resultados mostrados en la tabla 3.11 de 6 tareas de un experimento que cumplen los requisitos de pertenecer al mismo grupo desde la perspectiva de un criterio de agrupación.

Métrica	*Unidad*	*T1*	*T2*	*T3*	*T4*	*T5*	*T6*
Completitud de Tarea (CT)	*Proporción de tareas completadas*	*0*	*1*	*1*	*1*	*1*	*1*
Frecuencia de Error (FE)	*Errores/segundo*	*0*	*0.455*	*0.222*	*0.125*	*0.583*	*0.143*
Tiempo de Tarea (TT)	*Segundos*	*12*	*11*	*9*	*24*	*12*	*14*
Escala de Satisfacción (ES)	*Puntos Likert[1-7]*	*5*	*5*	*3*	*4*	*5*	*7*

Tabla 3.11 Ejemplo de métricas para 6 tareas agrupables

Como resultado de la agrupación de estas tareas calculamos las variables que describen las métricas, mostradas en la tabla 3.12.

Métrica	*Media*	*Mediana*	*Desviación típica*	*Intervalo de confianza (95%)*
Completitud de Tarea (CT)	*0.833*	-	-	*[0.622, 1.044]*
Frecuencia de Error (FE)	*0.22*	*0.134*	*0.235*	*[0.070, 0.369]*
Tiempo de Tarea (TT)	*14.25*	*13.5*	*4.224*	*[11.566, 16.934]*
Escala de Satisfacción (ES)	*4.833*	*5*	*1.528*	*[3.863, 5.804]*

Tabla 3.12 Ejemplo de cálculo de variables para la agrupación de 6 tareas agrupables

Gracias a estos cálculos, podremos describir la usabilidad mediante sus tres dimensiones más relevantes: eficacia, eficiencia

y satisfacción. Además, obteniendo la visión a nivel colectivo en base a diferentes criterios de agrupación, podemos interpretar los resultados en base a un colectivo o entorno concreto (p. ej., habiendo obtenido un límite inferior del intervalo de confianza en el tiempo de tarea de 11.566 segundos, aseguramos que en el 95% de las tareas, el grupo estudiado mediante el criterio de clasificación del ejemplo, van a realizar la tarea como mínimo en 11 segundos y medio).

3.2.3.2. COMPONENTE FORMATIVO

Este componente se centra en descubrir los posibles problemas de usabilidad que posee la aplicación a evaluar. *Asumimos que un problema de usabilidad está ligado a errores de interacción cometidos por el usuario de pruebas durante la ejecución de las tareas en interfaces específicas*. Por ello, entendemos que varios errores cometidos pueden estar ocasionados por el mismo problema de usabilidad base. Consecuentemente, especificamos lo siguiente:

> *El objetivo que tiene este componente es definir una catalogación y descripción de los errores de interacción que pueden estar asociados a problemas de usabilidad de la aplicación a evaluar.*

Imaginemos que disponemos de una aplicación con una lista desplegable en el que se encuentran las provincias españolas y uno de los pasos de una tarea es seleccionar la provincia de Cantabria. Si varios usuarios de pruebas seleccionan la provincia de Cádiz o Castellón, posiblemente sea debido al mismo problema de usabilidad: no existe mucha separación entre las opciones de la lista, por lo que el usuario de pruebas selecciona las opciones próximas accidentalmente. Por consiguiente, disponemos de varios errores de interacción que son originados por el mismo problema de usabilidad.

Desafortunadamente, debido a las limitaciones de esta metodología, no podemos asignar automáticamente a los errores de interacción una asociación de pertenencia a un problema de usabilidad concreto. Sin embargo, se ofrece una aproximación

donde se asume que todas las veces que se origine el mismo error de interacción, serán dadas por el mismo problema de usabilidad.

Tipo de error	*Tarea*	*Interfaz*	*Objeto*	*Valor*	*Nivel severidad*	*Descripción*
1	*T1*	*Interfaz A*	*BTN_5*		*Alto*	*Pulsar 'Búsqueda'*
5	*T1*	*Interfaz B*	*SPN_10*	*"sub_1"*	*Medio*	*Sel. Subcategoría*
6	*T1*	*Interfaz B*	*SPN_11*	*"León"*	*Bajo*	*Sel. Provincia*

Tabla 3.13 Ejemplo de componente formativo

Como se muestra en la tabla 3.13, este componente está formado por un conjunto de tipos de error que son descritos mediante la interfaz donde se originaron, el objeto que ha propiciado el error de interacción y la severidad calculada del mismo.

Se consideran del mismo tipo de error a aquellos errores que hayan sido originados por el mismo objeto dentro de la misma interfaz desempeñando la misma tarea. Para llevar a cabo esta clasificación, tomamos como base los eventos de interacción de tipo error (ver apartado 3.2.2.2) y realizamos una agrupación de todos los eventos de este tipo considerados iguales. *Dos eventos de interacción de tipo error se consideran iguales cuando han sido producidos durante la ejecución de la misma tarea, en la misma interfaz y además, han sido originados por el mismo objeto de la interfaz*.

Haciendo esta primera agrupación identificamos los diferentes tipos de errores que pueden asociarse a un problema de usabilidad. Añadiendo a cada tipo el objeto y la interfaz, podemos deducir en qué interfaces debemos centrar más la atención y en qué objetos de las mismas podemos caracterizar más conflictivos. Para priorizar los tipos de error, se realiza una catalogación en base a *la severidad del posible problema de usabilidad* asociado al error de interacción.

Para concluir un criterio de cálculo, hemos revisado la literatura referente al cálculo del nivel de severidad de los problemas de usabilidad. Dentro de esta revisión, se han detectado que todos los autores estudiados utilizan una escala que se compone de al menos tres niveles de severidad, de menos severo a más. También destacamos que los trabajos más relevantes utilizan una combinación de varios factores para calcular el nivel, como el

cálculo en base al dominio del problema de usabilidad (si afecta a una pequeña parte de la aplicación o a toda ella) combinado con el impacto del problema en el desarrollo de la tarea [Dumas+99], este último factor mayoritariamente utilizado.

En trabajos como el realizado por Nielsen [Nielsen95b], un problema de usabilidad con un nivel de severidad ínfimo o muy bajo equivale a problemas de usabilidad que no suponen casi ningún impacto y son incluso cuestionados como problema de usabilidad. En otros, la escala comienza en un nivel de severidad bajo, denominado a estos problemas como cosméticos [Nielsen94a] o irritantes [Rubin+08], que son conocidos como posibles problemas que originan simples molestias o incomodidades en la interacción con la aplicación (p.ej. un color y tamaño de letra que dificulta la lectura de los datos). En cambio, el nivel más alto de severidad muestra problemas que llegan a impedir que el usuario realice la tarea satisfactoriamente [Molich+03]. (p.ej. un botón de "login" imposible de pulsar que está oculto en un formulario de iniciar sesión en una pantalla con un tamaño reducido), o incluso fallos que lleguen a corromper el sistema dando lugar a fallos o pérdidas de datos [Wilson+01]. Por lo tanto, los problemas cuya severidad sea mayor deben solucionarse rápidamente, ya que tienen un gran impacto en la usabilidad de la aplicación. Uno de los principales inconvenientes de esta escala es la subjetividad del criterio ya que estos grados son definidos por los evaluadores de la aplicación.

Un criterio que sale de la subjetividad es el que utiliza la frecuencia de aparición del problema. Entendiendo como frecuencia de aparición al número de usuarios que experimentan una problemática concreta respecto al número total de los mismos. Destaca el criterio propuesto por Rubin y Chisnell [Rubin+08], que cuantifica las proposiciones cualitativas del trabajo de Nielsen [Nielsen95b] y de Molich y Jeffries [Molich+03].

Habiendo revisado los criterios más relevantes, se propone un criterio de tres niveles (bajo, medio y alto) basado en dos factores, cada uno con dos niveles (bajo o alto). Los dos factores son el

cálculo del impacto en la usabilidad, debido a su amplio uso; y la frecuencia de aparición, debido a su objetividad (ver tabla 3.14).

	Baja frecuencia	*Alta frecuencia*
Bajo impacto	*Bajo*	*Medio*
Alto impacto	*Medio*	*Alto*

Tabla 3.14 Criterio de nivel de severidad del problema de usabilidad

- Para realizar un cálculo objetivo del impacto en la usabilidad y deducir los dos niveles, nos basamos en la definición del nivel catastrófico dentro del factor impacto de Molich y Jeffries [Molich+03], donde indican que dichos problemas de usabilidad impiden que el usuario termine la tarea. Interpretamos con ello que en las tareas en las que aparece ese problema, muy pocas veces son completadas. Por ello, establecemos que un problema tiene bajo impacto si menos del 80% de las tareas en las que aparece ese error, no son completadas. Si deseamos calcular el impacto de un determinado problema de usabilidad ($Impacto_{PU}$), dicho factor se calculará dividiendo el número de tareas no completadas en las que aparece el problema de usabilidad ($TareasPU_{NC}$) entre el número total de tareas en las que aparece el problema de usabilidad ($TareasPU$).

$$Impacto_{PU} = \frac{TareasPU_{NC}}{TareasPU} \quad (3.1)$$

  Supongamos un conjunto de 40 tareas, del cual 10 han experimentado un problema de usabilidad concreto. Dentro de del conjunto que ha experimentado el problema de usabilidad, 9 no han llegado a completarse. Dados estos datos, el impacto del problema será un 90% (9/10), por lo que el nivel es de alto impacto.

  En el cálculo de los dos niveles de la frecuencia del problema de usabilidad hemos transformado la propuesta de cuatro niveles de Rubin y Chisnell [Rubin+08] en los dos niveles de esta aproximación. Consideramos altos los dos niveles más altos de ese trabajo, donde caen los problemas cuya frecuencia es igual o mayor al 51%. Para el cálculo de

la frecuencia del problema de usabilidad ($Frecuencia_{PU}$) debemos dividir el número de tareas donde aparece el problema de usabilidad ($TareasPU$) por el número total de tareas realizadas ($TareasTotales$).

$$Frecuencia_{PU} = \frac{TareasPU}{TareasTotales} \quad (3.2)$$

Siguiendo con el ejemplo mostrado en el cálculo del impacto, en este caso obtendríamos una frecuencia del 25% (10/40), por lo que la frecuencia asociada al problema de usabilidad es baja.

Concluyendo con la explicación de nuestro criterio de nivel de severidad y siguiendo con el ejemplo obtenemos un nivel de severidad medio. Si volvemos a la tabla 3.14, vemos que la intersección entre un problema de usabilidad con bajo impacto pero alta frecuencia muestra un nivel medio de severidad.

Con ello concluimos la explicación del componente, y por lo tanto, del modelo de análisis.

3.2.4. MODELO DE CASOS FAVORABLES

Para el cálculo de los atributos de este modelo usaremos los modelos anteriores para exponer, en función de los recursos que se dispongan, la descripción de los contextos más propensos a generar errores.

Por ello aclaramos lo siguiente:

> *El modelo de casos favorables tiene como objetivo ofrecer una descripción de los usuarios y entornos en los que se detectan más errores y las métricas de usabilidad presentan resultados extremos.*

Por un lado, se definen los casos favorables donde un usuario de pruebas es más propenso a cometer errores con una categoría de aplicación concreta. Por otro lado, dentro de los casos favorables

se realiza una descripción de los valores de los atributos de los entornos que componen cada caso favorable.

Con ello se ofrece una descripción de qué variables requieren especial atención. A continuación, se explica cómo se genera dicho modelo.

3.2.4.1. CÁLCULO DE CASOS FAVORABLES

Lindgaard y Chattratichart [Lindgaard+07] concluyeron que *no es tan importante la elección del número de participantes como la elección de los más adecuados*. Aplicando esta suposición al otro componente fundamental del contexto móvil, proponemos la siguiente teoría:

> *No es tan importante el número de entornos como la elección de los más adecuados.*

Siguiendo esta línea y simplificando el objetivo de este enfoque, podemos formular la siguiente pregunta:

> *¿Qué combinación de entornos y usuarios es la más acertada para detectar el mayor número de errores?*

Para hallar la respuesta llevaremos a cabo dos principales pasos: el cálculo de los diferentes casos posibles y su ordenación en base a un criterio de clasificación.

Nos basamos en la teoría de que el número de errores depende del contexto y que aplicaciones similares muestran resultados similares. Por ello se recupera información de las aplicaciones de igual categoría para calcular el factor de clasificación.

3.2.4.1.1. GENERACIÓN DE CASOS POSIBLES

Como primer paso nos centramos en los diferentes entornos posibles, con estos se generan todas las posibles combinaciones de los mismos. Imaginemos que disponemos de tres contextos denominados como S (sentado), W (caminando) y T (medio de transporte). El número de combinaciones posibles, como se muestra en la tabla 3.15, es de 7.

Combinación	*Entornos*	*Nº entornos*
1	*S*	*1*
2	*T*	*1*
3	*W*	*1*
4	*S y T*	*2*
5	*S y W*	*2*
6	*T y W*	*2*
7	*S, T y W*	*3*

Tabla 3.15 Ejemplo de combinación de entornos

Una vez generadas todas las posibles combinaciones de los entornos, se añaden las combinaciones de los tipos de usuarios.

En este trabajo, el criterio de agrupación de los usuarios es por género: hombre (H) y mujer (M). Por lo tanto, dispondremos de tres posibles combinaciones (ver tabla 3.16).

Combinación	*Tipos*	*Nº tipos*
1	*H*	*1*
2	*M*	*1*
3	*H y M*	*2*

Tabla 3.16 Ejemplo de combinación de tipos de usuario

Si generamos las posibles combinaciones, como resultado obtenemos los 21 casos posibles mostrados en la tabla 3.17.

Caso posible	*Entornos*	*Tipos de usuario*	*Nº entornos*	*Nº tipos de usuario*
1	*S*	*H*	*1*	*1*
2	*T*	*H*	*1*	*1*
3	*W*	*H*	*1*	*1*
4	*S y T*	*H*	*2*	*1*
5	*S y W*	*H*	*2*	*1*
6	*T y W*	*H*	*2*	*1*
7	*S, T y W*	*H*	*3*	*1*
8	*S*	*M*	*1*	*1*
9	*T*	*M*	*1*	*1*
10	*W*	*M*	*1*	*1*
11	*S y T*	*M*	*2*	*1*
12	*S y W*	*M*	*2*	*1*
13	*T y W*	*M*	*2*	*1*
14	*S, T y W*	*M*	*3*	*1*
15	*S*	*H y M*	*1*	*2*
16	*T*	*H y M*	*1*	*2*
17	*W*	*H y M*	*1*	*2*
18	*S y T*	*H y M*	*2*	*2*
19	*S y W*	*H y M*	*2*	*2*
20	*T y W*	*H y M*	*2*	*2*
21	*S, T y W*	*H y M*	*3*	*2*

Tabla 3.17 Ejemplo de casos posibles con tres entornos y dos tipos de usuario

Como dicta la matemática discreta, es importante señalar que el número de posibles combinaciones sigue la función $(2^n - 1) * 3$, siendo *n* el número de entornos posibles.

Al ser una función exponencial, el número de casos posibles aumenta considerablemente (ver figura 3.16).

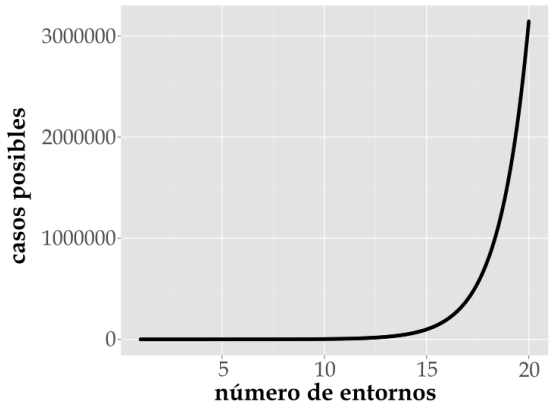

Figura 3.16 Función del número de casos posibles en base al número de entornos

Como se aprecia, al aumentar exponencialmente el número de casos posibles, también lo hará el tiempo de cálculo del valor de ordenación de los mismos. A continuación procedemos a explicar dicho cálculo.

3.2.4.1.2. CRITERIO DE ORDENACIÓN DE LOS CASOS POSIBLES

Habiendo visto la estructura de la tabla de casos posibles, el siguiente paso es dotar a dichos casos de un valor de ordenación en función de un criterio de clasificación.

En este trabajo hemos establecido como criterio una adaptación del modelo probabilístico descrito por Lewis [Lewis82]. El propósito de este trabajo es justificar la elección del número de usuarios en las pruebas de usabilidad mediante el cálculo de la probabilidad de detección de errores, concepto que emplearemos como criterio de clasificación. Su trabajo se basa en la teoría de que *el número mínimo de participantes depende del número de veces que un problema debe ser observado antes de que sea considerado problema y la*

magnitud de la proporción de la población de usuarios para la que se desean detectar. Define una fórmula (ver fórmula 3.3) que calcula la probabilidad de que un evento concreto (p.ej. la detección de un problema de usabilidad específico) aparezca al menos una vez en un número determinado de intentos. Esta fórmula explica que la probabilidad depende de dos variables: la probabilidad de que un evento suceda (p) y el número de iteraciones (n).

$$P\ (x \geq 1) = 1 - (1 - p)^n \quad (\ 3.3\ )$$

Aplicando la fórmula a un ámbito más práctico y centrado en nuestro trabajo, ésta representa la probabilidad $P\ (x \geq 1)$ de que un error de interacción o problema de usabilidad sea detectado o aparezca al menos una vez durante la realización de las pruebas por n usuarios. Por otro lado, *la probabilidad de detección (p)* define la probabilidad de que un error sea detectado. Para explicar el cálculo de p planteamos un sencillo ejemplo.

Si hemos elaborado un estudio de una aplicación donde se ha realizado una tarea concreta 12 veces, de las cuales 3 se han experimentado problemas graves, la probabilidad de experimentar problemas graves es la proporción de esos 3 eventos positivos con los 12 intentos; es decir, $p = 3\ /12 = 0.25$.

Comprendiendo las variables, mediante este modelo podemos calcular el número de usuarios que requerimos para asegurarnos la detección de problemas concretos con una probabilidad específica. Continuando con el ejemplo, si queremos asegurarnos con una probabilidad del 90% de que los problemas graves al menos ocurran una vez, sustituimos los valores en la fórmula y despejamos el valor de n, el cual nos hace afirmar que serán necesarios alrededor de 8 usuarios.

$$n = \frac{\ln(1 - P\ (x \geq 1))}{\ln(1 - p)} = \frac{\ln(1 - 0.9)}{\ln(1 - 0.25)} = 8.0039 \simeq 8\ usuarios$$

Habiendo visto cómo trabaja este modelo probabilístico, debemos ser conscientes de que muestra la probabilidad de solo un evento concreto.

Si representamos la probabilidad de que un error sea detectado al menos una vez (ver figura 3.17) como una función del número de usuarios n y la probabilidad de detección p, podemos comprobar que cuanto más valor tenga p, más probabilidades de detección y por lo tanto, mejor entorno para realizar las pruebas y encontrar errores.

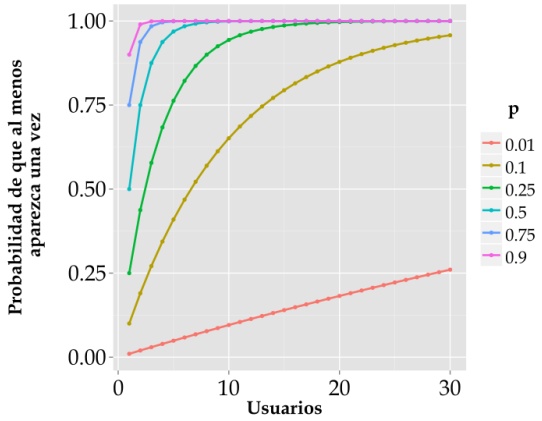

Figura 3.17 Función de probabilidad de que al menos un error aparezca una vez

Gracias a ello podemos centrarnos en el valor de p como criterio de clasificación. Al estar directamente relacionado $P\ (x \geq 1)$ con p, nos da el mismo orden de clasificación y por lo tanto, podemos omitir ese cálculo.

Para estudiar la probabilidad de detección de varios errores debemos hacer uso de un valor de p compuesto p_{comp}. En la literatura relacionada con este modelo, destacan varios trabajos [Hwang+07, Hwang+09, Hwang+10], donde hacen uso del mismo modelo probabilístico para estimaciones en evaluaciones heurísticas, al igual que [Nielsen+90], y además aplicado a estudios de usabilidad basados en escenarios, como en [Lewis94, Lewis01, Lewis12]. Otro ejemplo, más distante del foco de este trabajo es [Guest+06], donde hace uso del modelo para la planificación de entrevistas.

Aunque cabe la posibilidad de estimar el valor compuesto eligiendo la probabilidad de detección mínima mostrada por los errores, la mayoría de los trabajos revisados y mencionados

realizan *la estimación mediante el cálculo de la media de las probabilidades de detección de los errores individuales*. Por ello, asumimos el cálculo de la probabilidad de detección compuesta mediante la media. Para realizar dicho cálculo se compone una tabla en la cual contrastamos los errores descubiertos por los participantes. Por cada intersección entre un error y un participante marcamos la ocurrencia del mismo.

	Error 1	*Error 2*	*Error 3*	*Total*
Usuario 1	*x*	*x*		*2*
Usuario 2	*x*		*x*	*2*
Usuario 3		*x*		*1*
Usuario 4			*x*	*1*
Total	*2*	*2*	*2*	*6*

Tabla 3.18 Ejemplo de cálculo de la tabla de la probabilidad de detección

Una vez realizada la tabla, contabilizamos las ocurrencias sumándolas todas, en este caso 6. Después dividimos por el total de ocurrencias posibles que coincide con el número de celdas posibles en la tabla, calculado multiplicando el número de errores por el número de usuarios. Simplificándolo en una fórmula (ver fórmula 3.4), donde s es el número total de ocurrencias, e el número de errores y n el número de usuarios, podemos representar el valor del siguiente modo:

$$p_{comp} = \frac{s}{e * n} \quad (3.4)$$

Comprendiendo el cálculo de p_{comp}, procedemos a calcular dicho valor para el total de las posibles combinaciones.

Para el cálculo de cada valor correspondiente a cada combinación o caso posible, debemos generar una tabla de la probabilidad de detección. En primer lugar, generamos tantas columnas como errores hayan sido detectados. En segundo lugar, formamos tantas filas como usuarios hayan evaluado la aplicación. Para rellenar las intersecciones, revisamos las tareas realizadas por ese usuario en los contextos del caso posible y marcamos el error como encontrado si al menos en alguna de las tareas dicho error ha sido detectado.

Gracias a este método obtendremos el valor p_{comp} para la tabla generada con todas las posibles combinaciones. Por consiguiente, podremos llevar a cabo una clasificación eligiendo los casos posibles en los que el valor calculado sea mayor. Como dicho valor oscila entre 0 y 1, nos permite obtener una clasificación acorde al número de entornos que disponemos y a la vez elegir el que más se ajuste a los requisitos de la evaluación.

Caso posible	p_{comp}	*Entornos*	*Tipos de usuario*	*Nº usuarios necesarios para* $P(x \geq 1) = 0.8$
CP1	*0.309*	*S, T y W*	*H y M*	*4.35*
CP2	*0.298*	*S y T*	*H y M*	*4.54*
CP3	*0.210*	*W*	*H y M*	*6.80*
CP4	*0.119*	*T*	*M*	*12.70*
CP5	*0.087*	*S*	*H*	*17.70*

Tabla 3.19 Ejemplo de clasificación de casos posibles

Supongamos 5 casos posibles con las propiedades de la tabla 3.19. En este ejemplo, la elección del mejor caso puede variar si nos fijamos en el número de entornos o usuarios necesarios. En el ejemplo, si elegimos el caso posible CP1 debemos reproducir las pruebas en tres contextos. En cambio, si omitimos uno de los entornos (W) obteniendo el caso CP2, podremos ahorrar la ejecución de las tareas en un entorno disminuyendo insignificantemente la probabilidad de encontrar errores. Además, al estudiar las probabilidades obtenidas mediante la figura 3.18, bastaría con realizar las pruebas mediante el caso probable CP3 con sólo el entorno W en el caso de disponer de 20 usuarios.

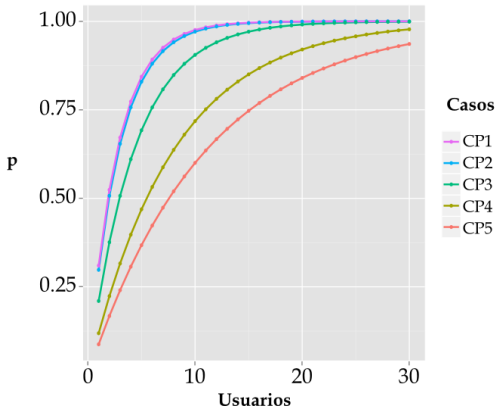

Figura 3.18 Función de probabilidad para los casos posibles de ejemplo

Como hemos explicado, éstas son las bases en las que el modelo de análisis se sustenta para la priorización de los mejores casos posibles a probar. Habiendo descrito su cálculo, a continuación explicamos las variables de contexto del caso probable elegido.

3.2.4.2. DESCRIPCIÓN Y CÁLCULO DE LAS VARIABLES DE CONTEXTO

Mediante este enfoque describiremos de un modo más explícito los casos favorables elegidos. Describiremos las variables de contexto que conforman dichos casos mediante dos principales niveles: nivel descriptivo y nivel inferencial. Mediante el *nivel descriptivo* describimos los estados y valores que adoptaban las variables de contexto cuando han aparecido errores en aplicaciones anteriores y similares. Para complementar esta información, el *nivel inferencial* estudia la relación de la variable de contexto con el número de errores detectados.

3.2.4.2.1. NIVEL DESCRIPTIVO

Para describir los valores o estados que adoptan las variables de contexto cuando se detectan errores en la interacción, se hace un estudio en función del tipo de variable del modelo de contexto explicado en el apartado 3.2.1.

3.2.4.2.1.1. DESCRIPCIÓN DE VARIABLES CUANTITATIVAS

Dentro de las variables cuantitativas descritas en el modelo, se utilizan las principales medidas que ofrece la estadística descriptiva. Para calcularlas, se extraen todos los valores de las variables tomadas justo en el momento de originarse el error.

Tiempo (s)	*Ruido (dB)*	*Iluminancia (lux)*
3.61	*63.05*	*6.17*
8.41	*91.27*	*19.70*
23.14	*89.28*	*45.81*
25.95	*63.80*	*46.36*
27.82	*74.07*	*11.89*
30.53	*65.48*	*12.47*
38.13	*64.79*	*7.71*
38.49	*67.00*	*11.62*
44.78	*64.83*	*10.58*
49.39	*65.36*	*73.23*

Tabla 3.20 Ejemplo de valores de las variables ruido e iluminancia en los momentos de causar errores

Supongamos que tenemos los datos de la tabla 3.20 y que deseamos estudiar el nivel de ruido y la iluminancia. En esta descripción, se calculan la *media* y la *mediana* como medidas de tendencia central. Añadimos el cálculo del *primer cuartil* y *tercer cuartil* a la descripción de los datos, además de los valores *máximos* y *mínimos*. Finalmente, para tener una visión de la dispersión de los datos añadimos el cálculo de la *desviación típica*. Por lo tanto, los resultados de los datos supuestos de ejemplo corresponden a la tabla 3.21.

Estadístico	*Ruido (dB)*	*Iluminancia (lux)*
Mínimo	*63.05*	*6.17*
Cuartil Q1	*64.80*	*10.84*
Mediana o Cuartil Q2	*65.42*	*12.18*
Media	*70.89*	*24.55*
Cuartil Q3	*72.30*	*39.28*
Máximo	*91.27*	*73.23*
Desviación típica	*10.67*	*22.63*

Tabla 3.21 Ejemplo de estadísticos descriptivos de dos variables de contexto cuantitativas

3.2.4.2.1.2. DESCRIPCIÓN DE VARIABLES CUALITATIVAS

Para describir las variables cualitativas del modelo, no se pueden hacer uso de las medidas anteriores. En este caso generamos la *tabla de frecuencia* de cada una de las variables. Para construirla, asignaremos tantas filas como estados pueda adquirir la variable de contexto analizada. Calcularemos dos medidas por cada estado.

Primero calcularemos la *frecuencia absoluta simple* ($f_{absoluta}$), haciendo un simple recuento del número total de errores detectados cuando la variable de contexto se encontraba en dicho estado. A continuación, calculamos la *frecuencia relativa simple* ($f_{relativa}$) mediante la fórmula 3.5, donde dividimos la frecuencia absoluta de este estado por el número total de errores detectados (n).

$$f_{relativa_i} = \frac{f_{absoluta_i}}{n} \quad (3.5)$$

Como resultado, dispondremos de una tabla similar a la tabla 3.22. Es importante recalcar que en este ejemplo se dispone de una variable cuantitativa policotómica de tres posibles estados. Si dispusiéramos de una variable dicotómica, generaríamos una tabla de frecuencia de solo dos filas, sin variar el método en absoluto.

Estado de variable	*Frecuencia absoluta simple*	*Frecuencia relativa simple*
Red GPRS	*94*	*0.51*
Red 3G	*32*	*0.17*
Red 4G	*57*	*0.31*

Tabla 3.22 Ejemplo de tabla de frecuencia para una variable de contexto cualitativa

Gracias a los cálculos expuestos, se ofrece una visión más explícita del contexto donde los usuarios han presentado más predisposición a cometer errores de interacción.

3.2.4.2.2. NIVEL INFERENCIAL

Habiendo visto qué valores tienen las variables de contexto cuando se producen errores, gracias a este nivel calcularemos cuáles de las variables requieren especial vigilancia al mostrar un mayor efecto sobre el número de errores. Interpretando este efecto como:

> *El grado en el que una variable puede afectar en el incremento del número de errores cometidos por el usuario.*

Haciendo un análisis de la dependencia entre los errores y las variables de contexto, podremos orientarnos a la hora de centrar nuestra atención en las mismas. Para ello tomamos como base las pruebas de contraste de hipótesis que dicta la estadística inferencial. Fundamentaremos el análisis estableciendo hipótesis basadas en pruebas matemáticas que hacen posible asumirlas como afirmaciones válidas dentro de unos márgenes de confianza definidos. Formulando las hipótesis sobre la relación entre dos variables, este procedimiento nos permite verificar si esa relación existe o no y con qué margen de error.

Aunque tomaremos simultáneamente medidas múltiples de cada interacción, estudiaremos las variables de contexto de un modo independiente. Por ello, el enfoque explicado en este apartado excluye de su base el análisis multivariante.

De cara al estudio correlacional, definimos *el número de errores cometidos como la variable dependiente* que es influenciada, y *las variables de contexto como variables independientes*, que afectarán a la dependiente durante la interacción con la aplicación móvil.

Para extraer y analizar satisfactoriamente las muestras debemos centrarnos en el tipo de variable independiente en estudio. La variable dependiente va a ser siempre la misma variable cuantitativa (número de errores), pero en función de la naturaleza de las variables independientes debemos realizar el estudio de un modo u otro.

Como resultado final del estudio, asignaremos la prioridad con dos simples estados: prioritario o no prioritario.

3.2.4.2.2.1. ELECCIÓN Y EJECUCIÓN DEL TIPO DE METODOLOGÍA DE CÁLCULO

Como se aprecia en el diagrama mostrado en la figura 3.19, si disponemos de una variable independiente cuantitativa optamos por la metodología de análisis tipo I. Por otro lado, si se dispone de una variable independiente cualitativa debemos usar la metodología tipo II o tipo III en función del número de posibles valores que pueda adoptar.

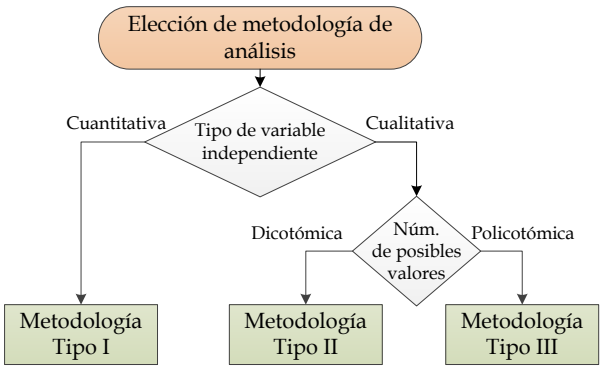

Figura 3.19 Elección de la metodología de análisis de relación

Una vez elegida la metodología debemos realizar su análisis partiendo siempre de una hipótesis. Deberemos rechazar o aceptar dicha hipótesis en función de las evidencias encontradas en los datos. Posteriormente, estudiaremos el efecto de la variable para finalmente concluir su nivel de prioridad. De un modo genérico las tres metodologías de análisis siguen un procedimiento base:

- *Extracción de la muestra*. Dependiendo del tipo de variable a estudiar, debemos extraer la muestra relevante del conjunto de datos crudos capturados en la fase de ejecución de las pruebas.
- *Análisis de relación entre variables*. Una vez generada la muestra a analizar, aplicamos un procedimiento estadístico de análisis de relación acorde con el tipo de variable independiente presentada. En los procedimientos utilizados, adquirimos un enfoque en el que no asumimos dependencia entre las variables hasta que el conjunto de muestras no expongan evidencias suficientes. Consecuentemente, formulamos de un modo genérico las siguientes hipótesis, que posteriormente serán definidas de un modo más explícito en la descripción de las tres metodologías de análisis:

  H_0: *Las dos variables en estudio son independientes*
  H_a: *Las dos variables en estudio están relacionadas*

- *Normalización e interpretación del resultado*. Una vez que tomamos una decisión respecto a la dependencia entre las variables en estudio (si están relacionadas o no) mediante el procedimiento estadístico relevante, debemos cuantificar dicha relación. Para ello hacemos uso del tamaño del efecto y las interpretaciones del mismo propuestas por Jacob Cohen [Cohen88, Cohen92]. Dicho autor propone ciertas equivalencias en las que transforma ciertos resultados estadísticos que cuantifican el tamaño del efecto en una escala de tres niveles: pequeño, medio y grande.

Dependiendo del tamaño del efecto, deduciremos si la variable de contexto analizada muestra un efecto significativo sobre el número de errores y decidiremos priorizar o no dicha variable.

Habiendo dado una idea general de los diferentes pasos que debemos seguir para el cálculo de este valor, procedemos a explicar las diferentes metodologías de cálculo en función de la naturaleza de la variable.

3.2.4.2.2.2. METODOLOGÍA DE ANÁLISIS TIPO I

Haremos uso de esta metodología cuando dispongamos de una variable independiente cuantitativa.

El primer paso que debemos realizar es extraer los datos para elaborar la muestra. Para este tipo de datos, se genera un caso por cada tarea realizada. Debido a que solo es de vital interés los casos en los que el usuario realiza algún tipo de interacción con el dispositivo, se extraen todas las entradas correspondientes a los pasos de interacción y los valores que la variable independiente tenía durante la ejecución de la tarea. Como se muestra en el ejemplo de la tabla 3.23, contabilizamos el número de errores cometidos durante la tarea y calculamos la media de los valores de la variable independiente cuando se realiza algún tipo de interacción, tanto errores como avances en la misma (ver apartado 3.2.2).

Identificador de tarea	*Media de iluminancia en las interacciones (lux)*	*Número de errores*
22877	*56.4*	*2*
40847	*67*	*4*
69002	*102*	*5*
148519	*32.8*	*0*

Tabla 3.23 Ejemplo de muestra con variable de contexto cuantitativa

Una vez obtenidas las muestras correspondientes, se procede al estudio de relación entre variables para averiguar si existen evidencias significativas que afirmen que el número de errores depende de la variable cuantitativa en estudio.

Para formular la hipótesis se hace un estudio de la correlación de un modo analítico. Primeramente disponemos del cálculo de la covarianza, que permite determinar si existe una dependencia entre ambas variables. Su valor es expresado en la escala de medida de las variables en estudio y no sabemos su cota superior, por ello no podemos cuantificar en qué medida hay una dependencia. Debido a los inconvenientes de la covarianza, hemos optado por el cálculo del *coeficiente producto-momento de Pearson* de la muestra, asignado por la letra *r* (fórmula 3.6). Corrige y añade un valor adimensional al dividir la covarianza (S_{xy}) entre las desviaciones típicas de cada variable (S_xy S_y), ya que todas tienen las mismas dimensiones. Se expresa con la siguiente fórmula:

$$r_{xy} = \frac{S_{xy}}{S_x * S_y} \quad (3.6)$$

Dicho coeficiente se traduce como la expresión numérica que nos indica el grado de relación existente entre ambas variables.

Varía entre los límites 1 y -1. Su magnitud indica el grado de asociación entre las variables, si el coeficiente toma el valor 0, implica que no existe una relación lineal. El signo muestra, al igual que la covarianza, el tipo de relación lineal. Si el valor es 1 implica correlación lineal perfecta directa (cuanto más aumenta una variable, más aumenta la otra) y si es -1 implica correlación lineal perfecta inversa (cuanto más aumenta una variable, más desciende la otra).

Desafortunadamente, este coeficiente estudia relaciones lineales, por lo que, en este trabajo quedan descartadas las relaciones no lineales. Por tanto, mediante este método vamos a asumir que las dependencias entre variables cuantitativas son lineales.

Concretamente se especifican las siguientes hipótesis:

H_0: *El coeficiente de correlación es cero (r=0)*
H_a: *El coeficiente de correlación difiere significativamente de cero (r!=0)*

Una vez calculado el coeficiente de correlación se debe comprobar su significación estadística mediante el cálculo del *p-valor*, que nos indica el riesgo que corremos al rechazar la hipótesis nula con la información que nos proporciona la muestra. En el caso de este trabajo, se asume un riesgo de rechazar la hipótesis nula de un 5% (*p-valor* < *0.05*). Con esto podremos decir que los resultados de estas pruebas que alcanzan este nivel de significación tienen menos de un 5% de probabilidad de que no haya una correlación real.

En el supuesto caso en el que los valores calculados nos permitan asumir con una probabilidad menor del 5% que existe una relación entre las dos variables cuantitativas podremos cuantificar de un modo muy sencillo, la magnitud de la correlación. Como en el caso de este trabajo no consideramos relevante si la relación lineal es directa o inversa, estudiamos simplemente el valor absoluto del coeficiente |r|, que es el valor que nos indica la magnitud.

Gracias a la prueba de significación averiguamos si existen evidencias para afirmar que el número de errores depende significativamente de la variable independiente en estudio. Desafortunadamente, aunque podamos rechazar la hipótesis nula, puede darse el caso en el que exista una dependencia despreciable. Dicha situación es dada cuando |r| tiene un valor muy pequeño. Para establecer un criterio en el que podamos afirmar una dependencia con una magnitud significativa al interpretar |r|, consideramos los valores proporcionados por Cohen [Cohen92]. La decisión de los valores ajustados queda reflejada en la tabla 3.24.

Interpretación de Cohen	*Conclusión de la metodología*	*Rango de valores*
No relevante	*Variable no prioritaria*	$0 \leq \|r\| < 0.10$
Pequeño	*Variable no prioritaria*	$0.10 \leq \|r\| < 0.30$
Medio	*Variable prioritaria*	$0.30 \leq \|r\| < 0.50$
Grande	*Variable prioritaria*	$0.50 \leq \|r\| \leq 1$

Tabla 3.24 Criterio de aceptación en base al tamaño del efecto de variable independiente cuantitativa

Por lo tanto, asumimos lo siguiente:

> *Con un valor* $|r| \geq 0.30$ *es suficiente para suponer una relación entre el número de errores y la variable independiente lo suficientemente fuerte como para considerarla prioritaria.*

Concluyendo con el desarrollo de esta metodología, se exponen los diferentes pasos mediante la figura 3.20.

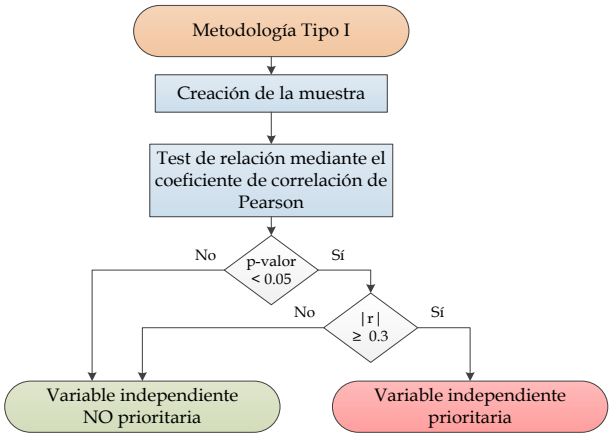

Figura 3.20 Pasos de la metodología de análisis tipo I

3.2.4.2.2.3. METODOLOGÍA DE ANÁLISIS TIPO II

En segundo lugar, cuando disponemos de una variable independiente de tipo cualitativo dicotómico se realiza el análisis como se describe.

Para la creación de la muestra se dispondrá de uno o dos casos por tarea realizada. Uno de los casos se forma realizando el sumatorio de los errores producidos en dicha tarea cuando la variable adopta uno de los estados. El otro caso es el sumatorio de los errores cuando la variable adopta el otro valor. Durante toda la ejecución de una tarea puede que no se modifique el valor de la variable independiente. Esto no quiere decir que debamos añadir un caso donde la variable tenga el valor que no ha adoptado con 0 errores porque estaríamos añadiendo un sesgo positivo a ese estado (ya que no se ha realizado la tarea con ese estado sin errores).

Como resultado, se obtendrá una muestra como la expuesta en el ejemplo de la tabla 3.25. En ella hay cuatro tareas realizadas donde en sólo una (tarea 22877) la variable ha cambiado de valor.

Identificador de la tarea	*Posición de pantalla*	*Número de errores*
22877	*horizontal*	*2*
22877	*vertical*	*4*
69002	*vertical*	*8*
40847	*horizontal*	*5*
148519	*horizontal*	*0*

Tabla 3.25 Ejemplo de muestra con variable de contexto dicotómica

Habiendo explicado la obtención de la muestra correspondiente a las tareas a analizar, se procede al estudio de la relación de las variables. En este caso, nos fundamentamos en una deducción bastante sencilla. Tomando como ejemplo la tabla anterior, si el número de errores que se produce cuando la posición de la pantalla es horizontal es muy similar al número de errores medio en vertical significa que no hay variación y por lo tanto, el número de errores no depende de la posición de la pantalla.

En esta metodología de análisis debemos llevar a cabo una agrupación de los casos de la muestra. Éstos se agrupan en función del estado de la variable independiente. Si tomamos la tabla 3.25 de ejemplo, la agrupación resultante estaría formada por un grupo con los tres casos donde la variable independiente adopta un valor (pantalla horizontal) y otro con los dos casos donde adopta el otro (pantalla vertical).

Al igual que la anterior metodología, al asumir que la distribución de la muestra es normal, centramos la atención en la media de los grupos. Después, especificamos las siguientes hipótesis donde μ_1 y μ_2 corresponden respectivamente a las medias del grupo 1 (pantalla horizontal) y del grupo 2 (pantalla vertical):

H_0: *La media de los dos grupos son iguales* ($\mu_1 = \mu_2$)
H_a: *La media de los dos grupos son significativamente diferentes* ($\mu_1 \neq \mu_2$)

Dependiendo de la homogeneidad de la varianza de la distribución deberemos ceñirnos a una prueba estadística u otra.

Esto es debido a que ambas pruebas asumen que la muestra sigue una distribución normal pero en términos de homogeneidad de la varianza, la prueba de relación de *t de Student* asume homocedasticidad (las varianzas son muy similares) y la *t de Welch* heterocedasticidad (varianzas diferentes). Para ello se hace uso del procedimiento estadístico conocido como *la prueba de Levene* [Levene60]. Dicho procedimiento asume también el supuesto de una distribución normal y será el término que nos guíe para el uso de un test de relación u otro. También se basa en pruebas de contraste de hipótesis donde se define una hipótesis nula que afirma la igualdad de varianzas. De cara a la obtención del *p-valor*, si es mayor o igual que 0.05 suponemos varianzas distintas, en caso contrario las supondremos iguales.

Una vez concluido el estudio de la homogeneidad de la varianza debemos realizar el test de relación correspondiente. Independientemente del test utilizado debemos comprobar, al igual que con la metodología tipo I, su significación estadística mediante el cálculo del *p-valor*. De nuevo el riesgo de rechazar la hipótesis nula es 5% (*p-valor* < *0.05*).

En el caso de que detectemos una relación significativa, procedemos a cuantificar el efecto mediante el uso de la *d de Cohen* [Cohen88]. Esta medida representa el número de desviaciones típicas que separan los dos grupos. Recuperando el ejemplo, si obtenemos un valor de 0.5 indica que la diferencia entre las medias del número de errores con la pantalla horizontal y con la pantalla vertical es de media desviación típica.

Considerando los valores proporcionados por el mismo Cohen [Cohen92], la decisión de los valores ajustados queda reflejada en la tabla 3.26.

Interpretación de Cohen	*Conclusión de la metodología*	*Rango de valores*
No relevante	*Variable no prioritaria*	$d < 0.20$
Pequeño	*Variable no prioritaria*	$0.20 \leq d < 0.50$
Medio	*Variable prioritaria*	$0.50 \leq d < 0.80$
Grande	*Variable prioritaria*	$0.80 \leq d$

Tabla 3.26 Criterio de aceptación en base al tamaño del efecto de variable independiente dicotómica

Haciendo uso de los valores y las interpretaciones mostradas asumimos lo siguiente:

> *Con un valor de la d de Cohen mayor que media desviación ($d \geq 0.50$) es suficiente para aceptar la suposición de que existe relación entre el número de errores y la variable independiente lo suficientemente fuerte como para considerarla prioritaria.*

Finalmente se describen los pasos de esta metodología de análisis mediante la figura 3.21.

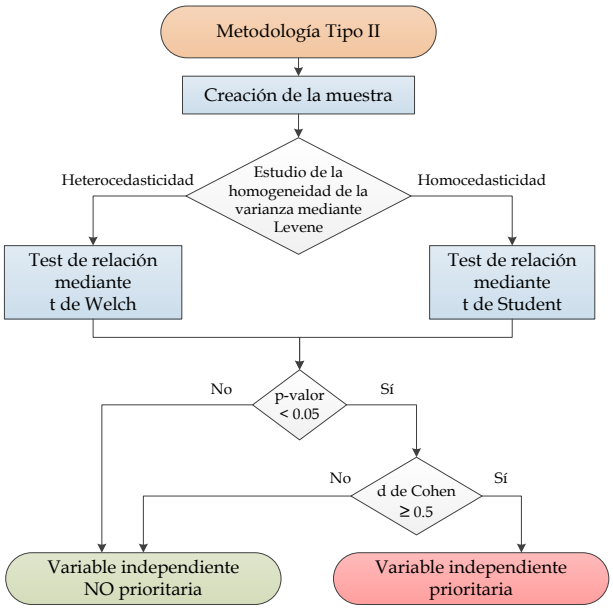

Figura 3.21 Pasos de la metodología de análisis tipo II

3.2.4.2.2.4. METODOLOGÍA DE ANÁLISIS TIPO III

Finalmente, la última metodología de análisis se utiliza cuando disponemos de una variable independiente de tipo policotómico. En esta metodología de análisis, se dispondrá como máximo de tantos casos como el número de posibles valores que puede adoptar la variable independiente. Al igual que la metodología anterior, los casos se forman realizando el sumatorio de los errores producidos cuando la variable adopta ese valor. Igualmente, para evitar añadir sesgos (ver apartado 3.2.4.2.2.3), sólo se añaden los casos cuyos valores de la variable independiente han sido

adoptados. Un ejemplo de cómo es el resultado de la creación de la muestra se expone en la tabla 3.27.

Identificador de la tarea	*Tipo de red conectada*	*Número de errores*
22877	*Red GPRS*	*7*
22877	*Red 3G*	*0*
22877	*Red 4G*	*4*
40847	*Red 3G*	*1*
148519	*Red 3G*	*1*
148519	*Red 4G*	*3*

Tabla 3.27 Ejemplo de muestra con variable de contexto policotómica

Partiendo de la metodología de análisis de tipo II, debemos realizar una agrupación de los casos de la muestra también en función del valor que variable independiente haya adoptado. En este caso, en vez de disponer de dos grupos, dispondremos de tantos como posibles valores pueda adoptar la variable. Agrupando los casos de ejemplo expuestos en la tabla 3.27, la agrupación resultante estaría formada por tres grupos: un grupo con los valores de GPRS (un caso), otro con los valores de 3G (tres casos) y otro con los valores de 4G (dos casos).

Al disponer de más de dos grupos, el análisis de relación no debe llevarse a cabo por las pruebas de *Student* o *Welch*, sino que debe recurrirse al *análisis de la varianza de un factor (ANOVA)* descrito por Chambers et al. [Chambers+92]. Esta prueba, también con base y atención en la media de los grupos, formula las siguientes hipótesis:

H_0: *La media de los grupos son iguales* ($\mu_1 = \mu_2 = \mu_n$)
H_a: *La media de todos los grupos son significativamente diferentes* ($\mu_1 \neq \mu_2 \neq \mu_n$)

Al tratarse de una prueba donde uno de sus supuestos es la homogeneidad de la varianza debemos realizar la prueba de Levene [Levene60] al igual que en la anterior metodología. Al contrario que la anterior, asumimos que si no hay una situación de

homocedasticidad no disponemos de pruebas suficientes como para continuar con el análisis ya que dentro de este trabajo quedan descartadas las pruebas de relación para distribuciones con varianzas heterogéneas en variables policotómicas. Si por el contrario podemos asumir homocedasticidad, llevamos a cabo el estudio en base a *ANOVA*. Realizando este test nos centramos de nuevo en su significación estadística mediante el cálculo del *p-valor*, con el cual asumimos de nuevo un 5% (*p-valor* < *0.05*) de riesgo de rechazar la hipótesis nula.

En el caso de *ANOVA* para cuantificar la relación hacemos uso del valor *"eta cuadrado"* (η^2). Esta medida es definida en [Jimenez+02] como *la proporción de la variabilidad total de la variable dependiente atribuible a la variable independiente*. Considerando de nuevo los valores proporcionados por Cohen [Cohen88], la decisión queda reflejada en la tabla 3.28.

Interpretación de Cohen	*Conclusión de la metodología*	*Rango de valores*
No relevante	*Variable no prioritaria*	$\eta^2 < 0.01$
Pequeño	*Variable no prioritaria*	$0.01 \leq \eta^2 < 0.06$
Medio	*Variable prioritaria*	$0.06 \leq \eta^2 < 0.14$
Grande	*Variable prioritaria*	$0.14 \leq \eta^2$

Tabla 3.28 Criterio de aceptación en base al tamaño del efecto de variable independiente policotómica

Para este tipo de metodología asumimos lo siguiente:

> *Con un valor* $\eta^2 \geq 0.06$ *es suficiente para aceptar la suposición de que existe relación entre el número de errores y la variable independiente lo suficientemente fuerte como para considerarla prioritaria.*

A modo resumen se muestra la figura 3.22 donde se exponen los pasos de esta metodología.

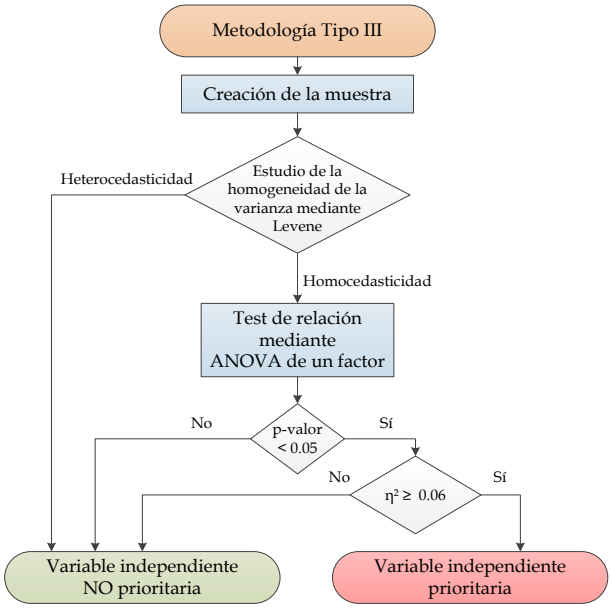

Figura 3.22 Pasos de la metodología de análisis tipo III

3.3. RELACIÓN DE LAS FASES CON LA BASE DE CONOCIMIENTO

Aunque ya han sido explicados los diferentes pasos de la metodología y la base de conocimiento, debemos manifestar la relación entre ellos. Se distinguen principalmente dos tipos de relación entre las fases de la metodología y la base de conocimiento: relación de consulta y relación de agregación de datos.

- Los pasos que manifiestan la *relación de consulta* son aquellos que demandan información de la base de conocimiento. Esta relación está presente en las tres fases de la metodología. En la primera fase se demanda información para la definición de las pruebas. En la segunda se necesita información ya almacenada para la generación del camino de interacción correcta de cada tarea y las tareas con las instrucciones definidas para la posterior ejecución de mismas. En la última fase, se necesita recuperar la información almacenada en la fase anterior para generar el

análisis. Explícitamente, mediante la tabla 3.29 mostramos los pasos y la información recuperada en cada fase.

Fase	*Paso*	*Información recuperada*
Definición	*Categorización de la aplicación*	*Tipos de aplicación*
Definición	*Definición de tareas*	*Tareas por tipo de aplicación*
Definición	*Declaración de usuarios y entornos disponibles*	*Entornos disponibles por tipo de aplicación*
Definición	*Generación y clasificación de casos*	*Evaluaciones anteriores de las aplicaciones del mismo tipo*
Ejecución	*Definición de caminos de interacción correcta*	*Tareas definidas*
Ejecución	*Descarga de instrucciones y aplicación evaluada*	*Instrucciones de la evaluación*
Análisis	*Cálculo de métricas de usabilidad*	*Pruebas ejecutadas almacenadas*
Análisis	*Cálculo de métricas de usabilidad*	*Pruebas ejecutadas almacenadas*
Análisis	*Categorización y cálculo de errores*	*Pruebas ejecutadas almacenadas*

Tabla 3.29 Pasos de consulta de la base de conocimiento

- El conjunto de pasos que se relacionan con la base de conocimiento mediante la *agregación de datos* es de vital importancia. Dichos pasos forman parte de todas las fases de la metodología y su principal objetivo es alimentar los modelos de la base de conocimiento. Individualmente ninguno completa ningún modelo pero sí en su conjunto.

  En la primera fase se agrega la información con la nueva aplicación a evaluar y las pruebas a realizar junto con sus contextos. En la segunda fase, se generan los caminos de interacción correcta y las tareas que realizan los usuarios. Finalmente, en la última fase se alimenta la base de conocimiento con los resultados de la evaluación.

  Mediante la tabla 3.30 mostramos los pasos y los modelos relacionados con los mismos que conforman la agregación de datos.

Fase	*Paso*	*Información agregada*
Definición	*Categorización de la aplicación*	*Nueva aplicación*
Definición	*Definición de tareas*	*Nuevas tareas*
Definición	*Declaración de usuarios y entornos disponibles*	*Usuarios y entornos donde realizar las tareas*
Ejecución	*Definición de caminos de interacción correcta*	*Caminos de interacción correcta*
Ejecución	*Subida de información de tareas realizadas*	*Pruebas ejecutadas capturadas para el cálculo de métricas*
Análisis	*Almacenamiento de resultados*	*Resultados del análisis*

Tabla 3.30 Pasos de agregación de la base de conocimiento

Concluyendo la descripción de la metodología, procedemos al detalle de la plataforma de soporte.

CAPÍTULO 4

PLATAFORMA DE SOPORTE

«Saber no es suficiente; tenemos que aplicarlo.
Tener voluntad no es suficiente: tenemos que implementarla»,
Johann Wolfgang Goethe (1749-1832)

ÍNDICE DE CAPÍTULO 4

4.1. Plataforma Android como elección 137
4.2. Descripción general 140
4.3. Librería de integración 143
4.3.1. Funciones de la librería 145
4.3.2. Integración de la aplicación evaluada 147
4.4. Aplicación web de gestión de la base de conocimiento 150
4.5. Herramientas de desarrollador 152
4.5.1. Funcionalidad 152
4.5.2. Arquitectura de las herramientas de desarrollador 161
4.6. Herramienta de usuario de pruebas 163
4.6.1. Funcionalidad 163
4.6.2. Arquitectura de la aplicación 168

Una vez presentada y descrita la metodología, a continuación se presenta la plataforma software que da soporte a todas las fases que la componen.

Como ya hemos mencionado, la metodología por sí sola no puede lograr los requisitos definidos. Por ello, se presenta esta plataforma que complementará a la misma y lograrán en conjunto satisfacer los requisitos generales definidos en el apartado 2.4.2. Al igual que en el capítulo anterior, recapitulamos los requisitos propios de la plataforma de soporte (ver apartado 2.4.2.2) acorde a los requisitos generales a los que están ligados. Se exponen en la tabla 4.1.

Requisito general	*Requisito de la metodología*
R1. Debe ser capaz de ofrecer resultados para evaluaciones cuya finalidad sea tanto formativa como sumativa.	*RP1. Debe automatizar el proceso de cálculo de resultados de evaluación sumativa y formativa definidos en la metodología.*
R2. La cantidad de recursos necesaria debe ser reducida.	*RP2. Debe ofrecer herramientas que faciliten el proceso de análisis y gestión de las pruebas de usabilidad que permita llevar a cabo los procesos de estudio y elección de atributos de las pruebas que generen mejores resultados.*
R3. La calidad de los resultados no debe disminuir. Aunque la cantidad de recursos sea reducida, la fiabilidad de los datos debe ser alta.	*RP3. Debe automatizar tanto la captura de la interacción del usuario como la captura de los factores que compongan un modelo de contexto complejo que no añada sesgos a la interacción del usuario con la aplicación.*
R4. La privacidad de los usuarios que realizan las pruebas debe ser preservada.	*RP4. La captura de la interacción del usuario se debe realizar con herramientas que no amenacen la privacidad del usuario que realiza las pruebas.*
R5. El estudio de un modelo de contexto detallado debe ser posible.	*RP5. Las herramientas que componen la plataforma de soporte deben capturar los elementos del modelo de contexto definido en la metodología.*

Tabla 4.1 Resumen de los requisitos que debe cumplir la plataforma de soporte

Para cumplir los requisitos definidos, la plataforma de soporte presenta un conjunto de herramientas que cumplen con los mismos. Las *herramientas de desarrollador* tienen como objetivo facilitar la gestión del proceso de las pruebas de usabilidad. Para ello ofrece varias funcionalidades entre las que destacan: el cálculo automático de resultados tanto sumativos como formativos (RP1), la gestión de las evaluaciones y la ayuda en la elección de las pruebas más favorables (RP2). La *herramienta de usuario de pruebas* captura las pruebas realizadas automáticamente (RP3) haciendo uso sólo del terminal móvil del propio usuario y sin necesidad de generar grabaciones de audio o vídeo (RP4). Además, registra tanto la interacción del mismo como el modelo de contexto que forma parte de la base de conocimiento de la metodología (RP5).

Aunque la metodología presentada se ha centrado en aplicaciones móviles independientemente de la plataforma, la diferencia entre las mismas obliga a elegir una concreta para desarrollar la plataforma de soporte. Por ello, en primer lugar describimos la plataforma móvil elegida para la implementación de la plataforma de soporte. A continuación, realizamos una descripción general de la misma para finalmente detallar los elementos que la conforman.

4.1. PLATAFORMA ANDROID COMO ELECCIÓN

Android[27] fue creado por una compañía llamada Android Inc. en Palo Alto, California. En 2003, cuatro jóvenes socios llamados Andy Rubin, Rich Miner, Nick Sears y Chris White pusieron en marcha el desarrollo de un sistema operativo abierto que fuese capaz de generar una experiencia de usuario innovadora, basándose en el sistema GNU y el núcleo de Linux. Tiempo después, la empresa Google Inc.[28], fundada en 1998 por dos estudiantes de la Universidad de Standford, Larry Page y Sergei Brin respaldó económicamente el proyecto de Android Inc. y adquirió la compañía en 2005.

En 2007, la fundación Open Handset Alliance[29] fue creada por numerosas compañías de hardware, software y operadoras de telefonía (LG, HTC, Intel, Texas Instruments, T-Mobile, ARM, Garmin, Sony Ericsson y Toshiba entre ellas) para avanzar en los estándares abiertos de los dispositivos móviles. En diciembre de ese mismo año la primera versión del sistema operativo Android se dio a conocer. El sistema operativo continúa creciendo y evolucionando a medida que la comunidad de desarrolladores trabaja en conjunto para crear aplicaciones móviles innovadoras y adaptar nuevas tecnologías al proyecto.

Las aplicaciones Android están principalmente desarrolladas en lenguaje de programación Java[30]. Las herramientas del SDK (Software Development Kit) de Android compilan el código de la nueva aplicación móvil junto con todos los datos y recursos en un APK (Application Package File). Un archivo APK tiene todo el contenido de una aplicación para Android y es el archivo que los dispositivos utilizan para instalar la aplicación.

---

[27] https://www.android.com
[28] https://www.google.com
[29] http://www.openhandsetalliance.com
[30] http://java.com

Cada aplicación dispone de varios *componentes que interactúan entre sí mediante mensajes asíncronos o Intents*. En su conjunto permiten definir el comportamiento general de la aplicación. Existen cuatro tipos diferentes de componentes, cada uno con un propósito concreto.

- Un componente actividad o *Activity* representa una única pantalla con una interfaz de usuario con la que los usuarios pueden interactuar con el fin de realizar una tarea o varios pasos de tarea (p.ej. sacar una foto, enviar un correo electrónico, ver un mapa…). Una aplicación, por lo general consiste en múltiples componentes Activity que normalmente están ligados entre sí. El comportamiento típico de una aplicación es especificar un Activity como principal, que se presenta al usuario al iniciar la aplicación por primera vez. A partir del principal y en función de la interacción del usuario con el mismo, pueden comenzar otros componentes Activity para realizar otras acciones.
- Un componente de tipo *Service* se ejecuta en segundo plano para realizar operaciones de larga duración. Un servicio no proporciona una interfaz de usuario, su principal valor es que puede realizar operaciones sin bloquear la interacción del usuario con una actividad. Otro componente, como una actividad, puede iniciar el servicio.
- Los componentes de tipo *Content Provider* tienen como finalidad gestionar un conjunto de datos para que la aplicación o aplicaciones con los permisos adecuados puedan consultarlos y modificarlos.
- Un *Broadcast Receiver* es un componente que responde a mensajes transmitidos por el sistema. Estos mensajes anuncian eventos en el sistema (p.ej., la pantalla se ha apagado, la batería está baja, etc.). Otros componentes como las actividades o los servicios también pueden iniciar mensajes para anunciar ciertos eventos como la finalización de una descarga. Normalmente se utilizan para iniciar servicios o actividades en función del mensaje enviado.

De cara a las diferentes plataformas móviles que existen, varios motivos justifican la elección de ésta.

- En primer lugar, al tratarse de un sistema de código abierto posibilita estudiar su implementación y acceder a múltiples módulos del sistema. Gracias a su acceso a una amplia gama de librerías y herramientas, Android ha suscitado un gran interés dentro de la comunidad científica, y más en las que comparten intereses centrados en soluciones móviles.
- Desde un punto de vista comercial y no tan importante, Android y su mercado de aplicaciones Google Play[31] ha crecido a un ritmo muy elevado, lanzando numerosas versiones con las que se perciben una gran evolución, siendo la 5.1 Lollipop la última versión a fecha de 6 de abril de 2015. Además, en la figura de appFigures[32] (ver figura 4.1), vemos que el número de desarrolladores Android ha crecido considerablemente, superando los 375000 en 2014.

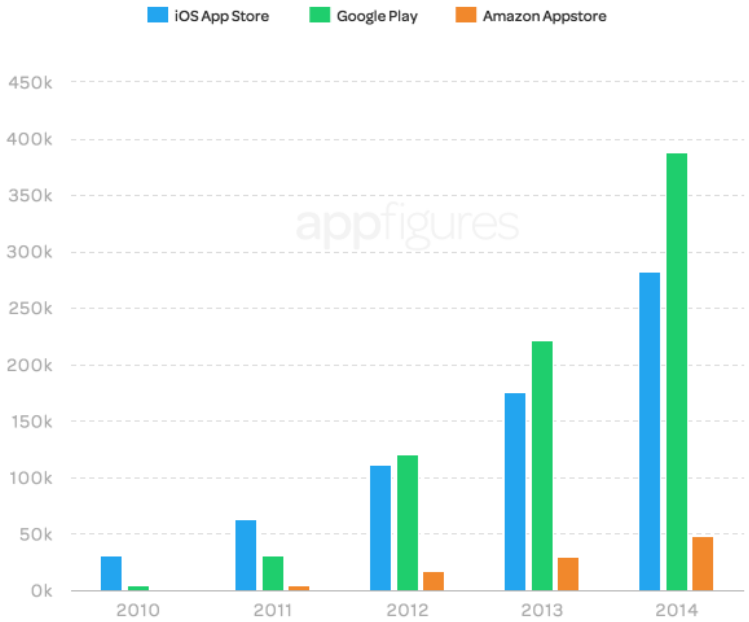

Figura 4.1 Número total de desarrolladores por plataforma

[31]https://play.google.com
[32]http://blog.appfigures.com/app-stores-growth-accelerates-in-2014

4.2. DESCRIPCIÓN GENERAL

La plataforma de soporte está formada por un *conjunto de elementos software que interactúan entre sí para componer la base de conocimiento y ayudar en el proceso de evaluación de la usabilidad de aplicaciones móviles*.

Como se menciona en el capítulo anterior, el propio desarrollador puede ser el evaluador (ver apartado 3.1). Debido a esto, el diseño de la plataforma de soporte asume que el desarrollador adquiere también el rol de evaluador. Por lo tanto, a partir de ahora el término desarrollador incluirá también al perfil de evaluador.

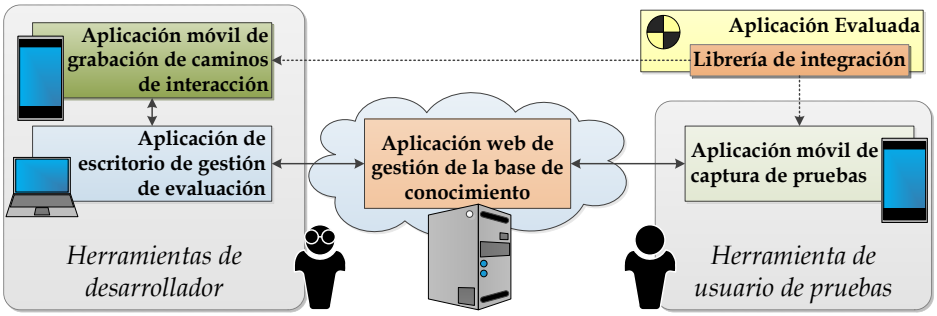

Figura 4.2 Elementos de la plataforma de soporte

La plataforma consta de varios elementos (ver figura 4.2): la aplicación web de gestión de la base de conocimiento, la aplicación evaluada con una librería de integración añadida y el conjunto de herramientas de soporte tanto de desarrollador como de usuario de pruebas.

- La *aplicación web de gestión de la base de conocimiento* es un software en el que se centralizan todas las evaluaciones realizadas mediante esta plataforma. Su principal objetivo es dar soporte a las diferentes necesidades de las herramientas (ya sean móviles o de escritorio) mediante una capa de integración basada en API (Application Programming Interface).
- La *librería de integración* consiste en un pequeño conjunto de funciones que ayudan a integrar la aplicación evaluada con la plataforma de soporte.

Clasificamos las herramientas de soporte en función del tipo de usuario que la utiliza: herramientas de desarrollador y herramientas de usuario de pruebas.

- Las *herramientas de desarrollador* son las que ayudan principalmente al desarrollador a gestionar el proceso de evaluación durante todas las fases de la metodología.

  La principal herramienta es la aplicación de escritorio de gestión de evaluaciones. Ésta ayuda al desarrollador en las tres fases de la metodología.

  Concretamente, para el apoyo en la fase de definición de los caminos de interacción correcta se ha implementado la aplicación móvil de grabación de tareas, que complementa a la aplicación de escritorio de gestión de evaluación.

- La *herramienta de usuario de pruebas* está centrada en la generación de datos de interacción y contexto, además de la ayuda al usuario de pruebas con su registro en la plataforma, descarga de aplicaciones y subida de datos.

Habiendo introducido los diferentes elementos, debemos aclarar el modelo de datos que dichos elementos generan y hacen uso.

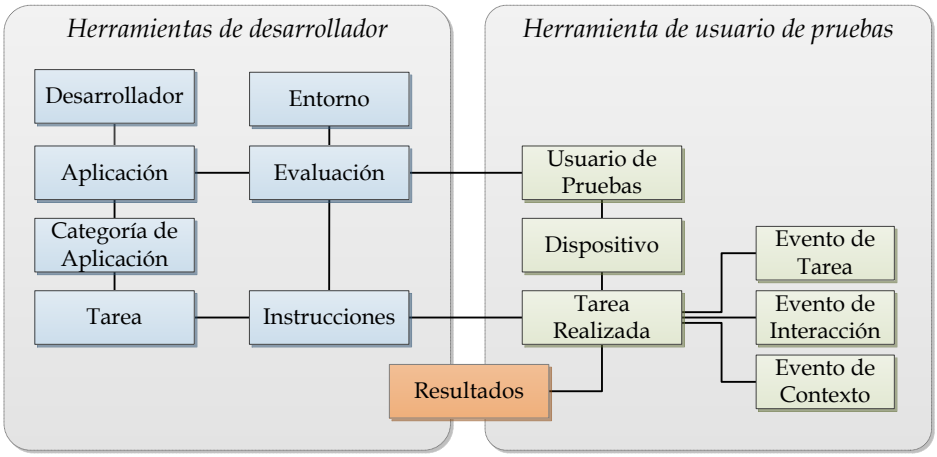

Figura 4.3 Modelo de datos

El modelo de datos de la plataforma (ver figura 4.3) consta principalmente de tres grupos, diferenciados por el objetivo de los mismos.

- En primer lugar, presentamos los elementos que conforman la definición de las evaluaciones generadas mediante las herramientas de desarrollador. El elemento *desarrollador* muestra los atributos correo electrónico y contraseña para validar las credenciales en el acceso a la plataforma y para asignarle propietario a las aplicaciones creadas. Cada objeto *aplicación* solo pertenecerá a un desarrollador y dispone de un identificador único, nombre, paquete de aplicación, icono, descripción y una categoría asignada. Cada *categoría* define un tipo de aplicación por la cual van a estudiarse los casos posibles. Sólo dispone de un identificador único y un nombre. Las *tareas* definen tareas genéricas que se realizan con las aplicaciones de una categoría determinada, por lo que cada tarea pertenece a una categoría concreta. Cuando se definen nuevas tareas dentro de una evaluación, deben asignarse instrucciones concretas para la aplicación a evaluar, para ello se crea el elemento *instrucciones* que consta, además de las instrucciones, del camino de interacción correcta con el que posteriormente se podrán clasificar los tipos de eventos de interacción. El elemento *evaluación* contiene la total descripción de la misma, consta de nombre, descripción, el progreso, la referencia a la aplicación evaluada, el conjunto de tareas con las instrucciones de las mismas y los entornos donde debe ser evaluada. Cada *entorno* consta de la etiqueta del mismo y de la descripción.
- En segundo lugar se presentan los elementos que conforman la ejecución de las pruebas. Constan de un elemento *usuario* referente a la descripción del usuario de pruebas mediante varios atributos: alias único y contraseña para el acceso a la plataforma, la fecha de nacimiento, el género, su altura, peso, lateralidad, los tipos de problemas de audición, los problemas de visión, el nivel de inglés, castellano, portugués, alemán y francés. Cada usuario tiene asignado un *dispositivo*. Aunque es posible que un usuario de pruebas disponga de varios dispositivos, asumimos que sólo usará uno para las pruebas. El dispositivo está

compuesto por los siguientes atributos: identificador único, el fabricante del terminal, el modelo, país de fabricación, pantalla, versión del sistema operativo y lenguaje del terminal. Cuando se realiza una tarea asignada a una evaluación concreta se genera un elemento de tipo *tarea realizada*, que detalla la ejecución de una tarea descrita en la fase de definición. Este elemento consta de un identificador único, la etiqueta del entorno en el que ha sido realizada la tarea, la referencia a las instrucciones de la misma y el conjunto de eventos producidos en la interacción. Este conjunto de eventos está formado por tres tipos posibles. Un *evento de tarea* refiere a los eventos de este tipo, consta de la marca temporal del evento y del tipo de evento de tarea. Al igual que el anterior, el *evento de interacción* refiere a los eventos de ese mismo tipo, consta también de la marca temporal del evento, del tipo de evento de interacción, de la interfaz donde se ha generado, el objeto que lo genera y el valor del mismo. El *evento de contexto* describe el cambio del valor de una variable de contexto mediante su marca temporal, la variable que cambia de valor y el nuevo valor adquirido. Los tipos de variable de contexto que se han monitorizado se describen, junto con la fuente utilizada, más adelante en la tabla 4.5 de la herramienta de usuario de pruebas.

- Finalmente, la mayor parte de los *resultados* son generados por la herramienta de aplicación de escritorio y una pequeña parte referente a las respuestas del cuestionario de satisfacción, generada por la herramienta del usuario de pruebas.

A continuación, complementamos la visión general de la plataforma de soporte expuesta mediante la descripción y detalle de los elementos que la conforman de un modo más explícito.

4.3. LIBRERÍA DE INTEGRACIÓN

Las herramientas de naturaleza móvil introducidas en la descripción general deben completar los eventos descritos en el

modelo de interacción (ver apartado 3.2.2). Por ello, debemos tener la capacidad de detectar tanto los eventos de tarea como los eventos de interacción producidos en la aplicación evaluada y ser capaces de transmitirlos a estas herramientas. En otras palabras, debemos ser capaces de integrar la aplicación evaluada con las herramientas de la plataforma de soporte.

> *Esta librería tiene como objetivo simplificar la integración de la aplicación evaluada con la plataforma de soporte.*

Utilizaremos esta librería en la fase de ejecución de la metodología desarrollada (ver apartado 3.1.2) por lo que será integrada por el desarrollador de la aplicación evaluada y utilizada por los usuarios de pruebas en la realización de las tareas. Consecuentemente, debe cumplir dos principales requisitos:

- Por un lado, debemos *ofrecer una integración rápida y sencilla*. El paso de integración de la aplicación evaluada con la plataforma es un paso específico de esta metodología que puede suponer un cuello de botella, ya que sin esta preparación e integración, las pruebas no pueden dar comienzo, retrasando el fin de la metodología. Por ello, presentamos un diseño de librería basado en tres simples comandos que deben ser añadidos en el código de la aplicación evaluada.
- Por otro lado, *la interacción del usuario de pruebas con la aplicación evaluada no debe ser sesgada por la captura*. Por ello, debemos tener especial cuidado en la modificación del contexto en el cual se llevan a cabo las pruebas. Si realizamos la captura de datos mediante software sin añadir componentes físicos, los diferentes componentes del contexto más propensos a ser modificados son: el usuario de pruebas, el rendimiento del propio dispositivo y la aplicación evaluada.

  Por motivos de privacidad, cualquier usuario debe ser notificado si se realiza la captura de información, lo que

hace imposible eliminar dicho sesgo. El rendimiento del dispositivo va a verse afectado en términos de memoria, batería y procesador. La aplicación evaluada puede verse afectada en el tiempo de respuesta, ya sea por el sesgo del dispositivo o por la adhesión de código dentro de la aplicación cuya ejecución sea pesada.

Para satisfacer ambos requisitos, se ha decidido eliminar completamente toda la funcionalidad relacionada con la captura del contexto, cuya responsabilidad queda asignada a la herramienta móvil de captura de pruebas. Gracias a ello, los procesos de la aplicación evaluada no se verán sobrecargados con códigos referentes a la captura del contexto. Sin embargo, la funcionalidad de propagación de eventos de interacción queda asignada a dicha librería.

Acorde a lo expuesto, se ha diseñado una librería ligera cuyo tamaño es inferior a 4kB. Como se muestra en la figura 4.4, ésta librería ofrece tres funciones básicas, llamadas desde los componentes de la aplicación evaluada que generan mensajes asíncronos del sistema (Intents). Posteriormente, estos mensajes son tratados por las herramientas de la plataforma para definir su comportamiento y generar la información correspondiente. A continuación se describen las funciones a introducir en el código.

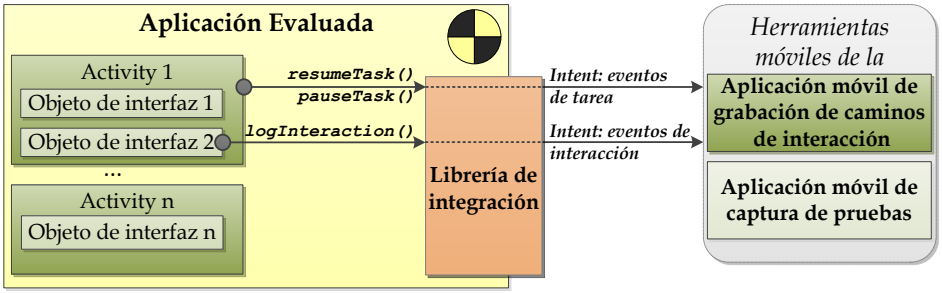

Figura 4.4 Uso de la librería por una aplicación evaluada

4.3.1. FUNCIONES DE LA LIBRERÍA

Estas funciones contienen la lógica necesaria para propagar los eventos de interacción y tareas generados por la aplicación evaluada. Existen tres funciones básicas. Las dos primeras generan

los eventos de tarea necesarios, llamadas cuando una interfaz es presentada o retirada. La última genera los eventos de interacción, llamada por cada componente de interfaz con el que interactúa el usuario.

- La función *resumeTask()* permite propagar un evento de tarea *continuación de tarea*. El único parámetro que necesita es el objeto que define el contexto de la aplicación, usado para enviar Intents. La llamada a esta función será realizada cuando la aplicación evaluada se muestre en primer plano y el usuario pueda seguir interactuando mediante interfaces de la misma.
- La función *pauseTask()* permite propagar un evento de tarea *pausa de tarea*. Al igual que la anterior, el único parámetro que necesita es el contexto de la aplicación. La llamada a esta función será realizada cuando la aplicación deja de mostrar interfaz con la cual el usuario puede interactuar.
- La función *logInteraction(),* al contrario que las anteriores, propaga eventos de interacción en vez de tarea. Dicha función es llamada cuando el usuario interactúa con un objeto de una interfaz de la aplicación. Los parámetros necesarios están estrechamente ligados a la estructura de los eventos de interacción (ver apartado 3.2.2.2). En primer lugar se necesita el identificador de la interfaz en la cual se encuentra el objeto con el que ha interactuado, el identificador del mismo, su valor y una descripción del evento. Cabe destacar que se generan todos los atributos del evento de interacción exceptuando su tipo, al estar estrechamente ligado a la tarea realizada.

Observamos que dentro de las funciones de la librería sólo se disponen de dos eventos de tarea (pausa de tarea y continuación de tarea) siendo cuatro (ver apartado 3.2.2.1).

> *Debido a que los eventos de tarea comienzo de tarea y fin de tarea son generados cuando el usuario de pruebas decide comenzar o finalizar la tarea, no son contemplados por la librería.*

El usuario de pruebas notificará estos eventos mediante la aplicación móvil de captura de pruebas (ver apartado 4.6.1.3).

Habiendo descrito el comportamiento y las funciones de la librería, detallamos el uso de ésta para la integración de la aplicación evaluada con la plataforma de soporte.

4.3.2. INTEGRACIÓN DE LA APLICACIÓN EVALUADA

El desarrollador llevará a cabo la integración mediante llamadas a las funciones de la librería desde los componentes de tipo Activity (eventos de tarea) y desde los eventos generados por los componentes de interfaz (eventos de interacción).

Como se ha explicado en la descripción del sistema Android, el componente Activity es el principal elemento que gestiona la interacción del usuario con las aplicaciones y el comportamiento de las interfaces de la aplicación. El ciclo de vida de un Activity se puede relacionar con los eventos de tarea (ver figura 4.5).

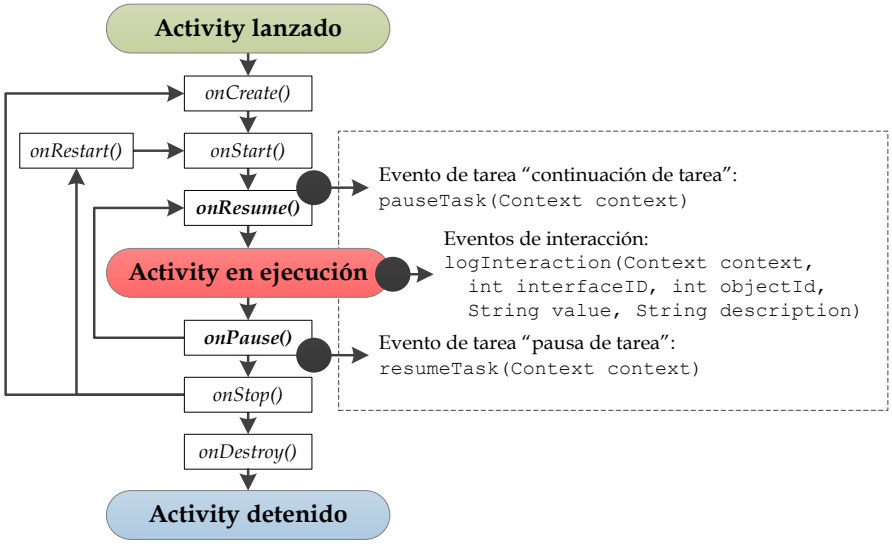

Figura 4.5 Ciclo de vida de un Activity y funciones de librería ejecutadas en las transiciones de estado

Un Activity pasa por tres estados: en ejecución, pausado y detenido. Cuando un Activity se encuentra en ejecución significa que está situado en primer plano de la pantalla y tiene la atención del usuario. Cuando se encuentra pausado, hay sobre éste otro

Activity, por el cual el usuario no puede interactuar con el evaluado. Finalmente, un Activity en estado detenido implica que no es visible por el usuario y no puede interactuar con el mismo.

Observamos que el único estado en el que el usuario puede interactuar con la aplicación es cuando el Activity se encuentra en el estado de ejecución. Este estado marca a su vez cuándo una tarea se encuentra también en el estado de ejecución. Por lo tanto, *si determinamos la entrada y salida en el estado de ejecución de los componentes Activity, podremos notificar los eventos de tarea que determinan el estado de la tarea*. Estas notificaciones serán producidas mediante la llamada a la librería en esas situaciones.

Afortunadamente, como muestra la figura 4.5, la implementación de un componente Activity dispone de varias funciones que ejecutan código en las transiciones de estado del Activity. Concretamente, dos funciones marcan la entrada y salida al estado en ejecución. Respectivamente, el código dentro de *onResume()* y *onPause()* se ejecuta justo antes de que el Activity comience y el usuario pueda interactuar, o cuando el Activity pasa a segundo plano y el usuario no puede interactuar. Por ello las llamadas a *pauseTask()* y *resumeTask()* se sitúan como muestra el código 4.1.

```
public class ActivityA extends Activity {
    …
    @Override
    protected void onResume() {
        super.onResume();                 // El usuario puede
        IntegrationLib.resumeTask(this); // interactuar con la
        …                                 // interfaz.
    }
    @Override
    protected void onPause() {
        super.onPause();                  // Otro componente va a
        IntegrationLib.pauseTask(this);  // pasar a primer plano
                                          // y el usuario deja de
    }                                     // poder interactuar
    …
}
```

Código 4.1 Ejemplo de funciones de librería para eventos de tarea en un Activity

Todos los elementos de una interfaz (botones, campos de texto,…) generan eventos de interacción cuando el usuario interactúa con ellos. Por lo tanto, también debemos añadir llamadas a la función

logInteraction() de la librería en el código donde se capturen dichos eventos.

Imaginemos un botón con identificador 111 en una interfaz con identificador 222. Éste botón realiza una búsqueda cuando el usuario lo pulsa. Dentro del código que se ejecuta cuando el botón es pulsado, se debe introducir una llamada a la función *logInteraction()*, añadiendo los parámetros necesarios para que se pueda generar el evento de interacción (ver código 4.2): el contexto de la aplicación, el identificador de la interfaz (222), el identificador del botón (111), el valor nulo y la descripción del objeto. El valor del objeto se introduce como nulo porque un botón no dispone de ningún valor. Por el contrario, si el objeto fuera un campo de texto, deberíamos introducir en este parámetro el valor del texto introducido. En el caso de una lista, introduciríamos el valor del componente seleccionado en el parámetro valor.

```
Button btnSearch;
btnSearch = (Button) rootView.findViewById(R.id.btn_search);
int INTERFACEID_CS = 222;
int OBJECTID_BTN = 111;

btnSearch.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        IntegrationLib.logInteraction(
            mHostActivity,           // Objeto contexto
            INTERFACEID_CS,          // Identific. De interfaz
            OBJECTID_BTN,            // Identific. De objeto
            IntegrationLib.NO_VALUE, // Valor
            "Botón búsqueda");       // Descripción
        searchAnnouncement();
    }
});
```

Código 4.2 Ejemplo de función de librería para eventos de interacción en un botón

Gracias a esta librería podemos capturar y propagar los eventos referentes al modelo de interacción y tarea a las herramientas móviles de la plataforma para procesarlos.

4.4. APLICACIÓN WEB DE GESTIÓN DE LA BASE DE CONOCIMIENTO

La aplicación web de gestión de la base de conocimiento reside en un servidor Apache Tomcat[33] con acceso a Internet. Ésta aplicación permite que dicha base sea utilizada por las herramientas de soporte.

> *La aplicación web tiene como objetivo gestionar toda la información de las evaluaciones realizadas de un modo centralizado y satisfacer las diferentes necesidades de las herramientas de soporte.*

Para lograr su objetivo la aplicación expone la funcionalidad CRUD (Create, Read, Update and Delete) sobre las diferentes entidades que conjuntamente componen de la base de conocimiento mediante la arquitectura de tres capas presentada en la figura 4.6: capa de persistencia, capa de lógica y capa de exposición de servicios.

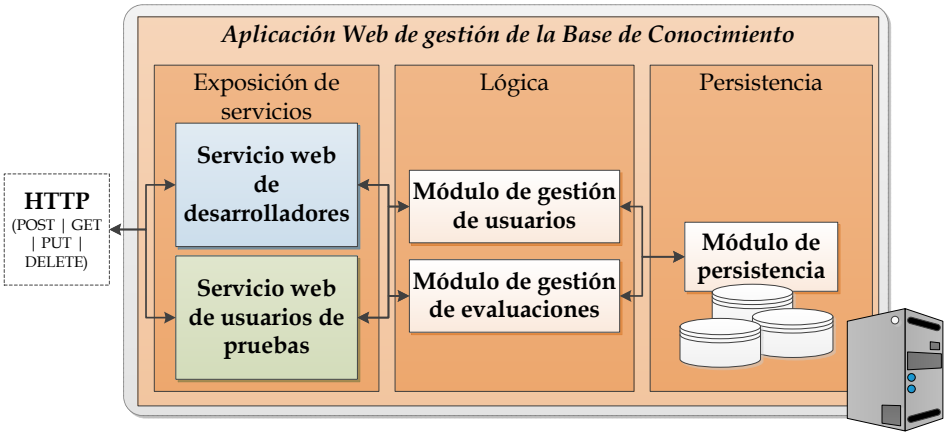

Figura 4.6 Arquitectura de la aplicación web de gestión de la base de conocimiento

[33] https://tomcat.apache.org

- La *capa de persistencia* gestiona una base de datos MySQL[34] que reside en el servidor y contiene toda la información de la base de conocimiento. Todas las especificaciones de evaluaciones, usuarios registrados y tareas realizadas son almacenadas en dicha base de datos.
- La *capa de lógica* realiza mediante dos módulos que se comunican con el módulo de persistencia, las operaciones necesarias para gestionar los usuarios de pruebas y las evaluaciones que son llevadas a cabo mediante la plataforma. Las operaciones relacionadas con las lecturas, altas, bajas y modificaciones de los usuarios de pruebas y los desarrolladores, son gestionadas por el módulo de gestión de usuarios. Además, mediante el módulo de gestión de evaluaciones se llevan a cabo todas las operaciones relacionadas con la gestión de las mismas, tanto la creación de las evaluaciones como la subida de pruebas realizadas por los usuarios.
- La *capa de exposición de servicios* expone toda la funcionalidad de la capa de lógica mediante dos servicios web que muestran un API con el estilo arquitectural REST (REpresentational State Transfer). Mediante el servicio web de desarrolladores la aplicación de escritorio de gestión de evaluación podrá acceder a la base de conocimiento para crear nuevas evaluaciones para aplicaciones evaluadas, recuperar datos de evaluaciones ya definidas y registrar tanto a desarrolladores como aplicaciones evaluadas. Por otro lado, el servicio web para usuarios de pruebas permite a la herramienta del usuario de pruebas tres principales operaciones: registrar nuevos usuarios de pruebas y dispositivos móviles, descargarse las instrucciones de las nuevas evaluaciones a realizar y almacenar los resultados de las pruebas en la base de conocimiento.

[34] http://www.mysql.com

4.5. HERRAMIENTAS DE DESARROLLADOR

Para llevar el control de una correcta ejecución de la metodología propuesta y todas las fases de una evaluación, hemos implementado la aplicación de escritorio de gestión de evaluación. Ésta aplicación ayuda al desarrollador en las tres fases de la metodología.

Dentro de la aplicación de gestión de evaluación, concretamente en la definición de los caminos de interacción correcta de la fase de ejecución, necesitaremos detectar los mensajes de sistema producidos por la librería para definir de un modo sencillo los caminos de interacción correcta. Con este fin, hemos implementado la aplicación móvil de grabación de caminos de interacción. Dicha aplicación móvil propagará los eventos de interacción a la aplicación de escritorio para crear y asignar los caminos de interacción correcta a cada tarea definida.

Consecuentemente, definimos lo siguiente:

> *La aplicación de escritorio de gestión de evaluación ayuda al desarrollador a definir una evaluación de una aplicación evaluada, crear los caminos de interacción correcta con la ayuda de la aplicación móvil de grabación de caminos de interacción, y analizar los resultados de la evaluación.*

Ésta aplicación gestiona los registros de los desarrolladores que deseen evaluar una aplicación con esta metodología, la definición de la información de nuevas evaluaciones y su almacenamiento.

4.5.1. FUNCIONALIDAD

Ésta aplicación de escritorio ofrece cinco funcionalidades principales para el desarrollo de la metodología a través de sus tres fases. Mediante la creación de la tabla 4.2, exponemos todas las funcionalidades acorde a la cronología de los pasos de la metodología desarrollada. Posteriormente se detallará cada una de ellas.

Fase	*Paso de la metodología*	*Funcionalidad*
-	-	*Registro de desarrollador e inicio de sesión*
Definición	*Categorización de la aplicación*	*Definición de nueva evaluación*
Definición	*Definición de tareas*	*Definición de nueva evaluación*
Definición	*Declaración de usuarios y entornos disponibles*	*Definición de nueva evaluación*
Definición	*Generación y clasificación de casos*	*Estudio de casos*
Definición	*Elección de usuarios, dispositivos y entornos*	*Definición de nueva evaluación*
Ejecución	*Definición de camino de interacción correcta*	*Grabación de tareas*
Análisis	*Cálculo de métricas de usabilidad*	*Análisis de resultados*
Análisis	*Categorización y cálculo de errores*	*Análisis de resultados*
Análisis	*Almacenamiento de resultados*	*Análisis de resultados*
Análisis	*Elaboración de informe*	*Análisis de resultados*

Tabla 4.2 Relación de pasos con la funcionalidad de la aplicación de escritorio de gestión de evaluación

4.5.1.1. REGISTRO DE DESARROLLADOR E INICIO DE SESIÓN

El primer paso que un desarrollador debe dar con la plataforma de soporte es su registro e identificarse en el sistema para dar comienzo a la creación de una evaluación. Consecuentemente, estos dos pasos se consideran necesarios dentro de la plataforma de soporte pero fuera de la metodología. Para ello, la aplicación de escritorio muestra una ventana de inicio de sesión (ver figura 4.7).

Figura 4.7 Ventana de inicio de sesión en la aplicación de escritorio

Si el desarrollador no dispone de credenciales, debe registrarse en la plataforma mediante el botón "Registrarse", donde debe introducir un nuevo correo electrónico y una contraseña. Estas credenciales serán almacenadas y servirán al nuevo desarrollador para posteriormente acceder a la plataforma.

4.5.1.2. DEFINICIÓN DE NUEVA EVALUACIÓN

Una vez el desarrollador dispone de credenciales y ha iniciado sesión, dan comienzo los pasos de la metodología.

En primer lugar se muestra una interfaz donde se muestran las diferentes evaluaciones realizadas (ver figura 4.8). Si se desea comenzar la evaluación de una nueva aplicación, se pulsa el botón "Nueva evaluación".

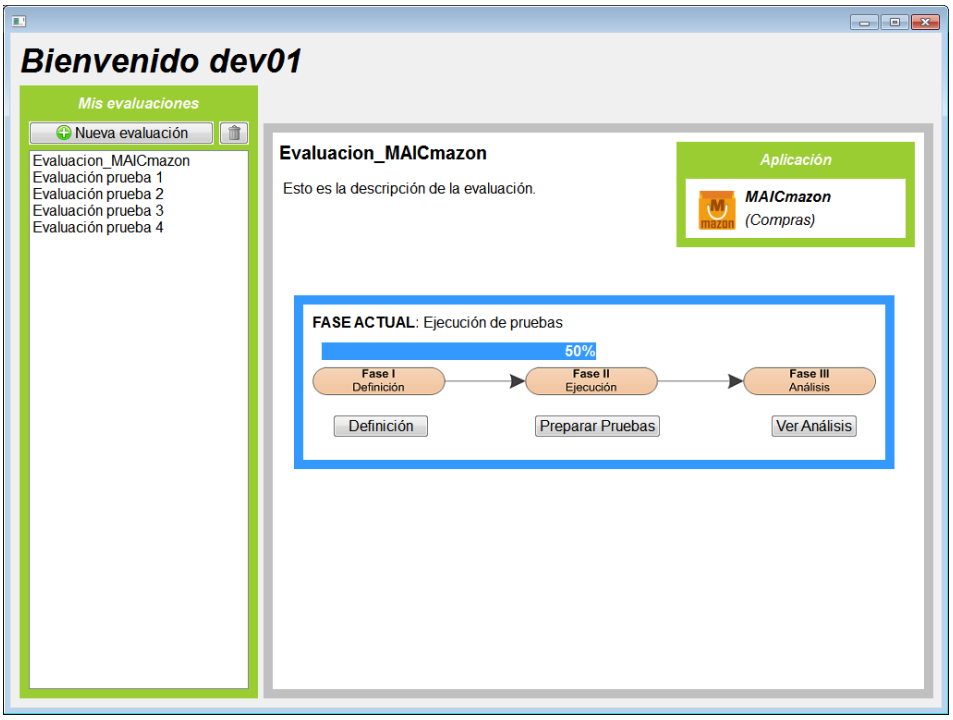

Figura 4.8 Panel principal de evaluaciones del desarrollador

Inmediatamente comenzará un asistente que nos guiará en la definición de la evaluación en 2 pasos.

- Primeramente, se muestra un formulario (ver figura 4.9) donde añadir los datos generales de la evaluación y realizar

la definición de las dimensiones de contexto intrascendentes (ver apartado 3.1.1.1).

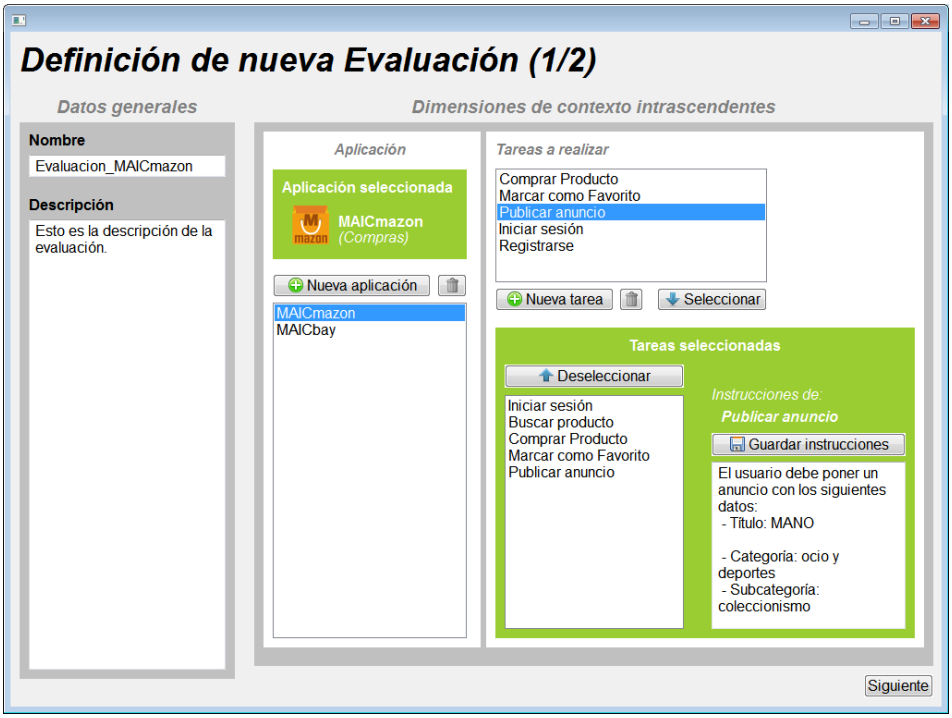

Figura 4.9 Formulario de datos generales de evaluación y dimensiones de contexto intrascendentes

Por un lado, los datos generales de la evaluación son introducidos mediante dos campos de texto que el desarrollador debe completar: nombre y descripción.

Por otro lado, las dimensiones de contexto intrascendentes se rellenan mediante dos bloques: bloque de aplicación y bloque de tareas. En el bloque de aplicación el desarrollador selecciona de una lista que muestra las aplicaciones del desarrollador la que se desea evaluar.

Si la aplicación evaluada todavía no se encuentra en la lista, el desarrollador debe agregarla. Para ello, se pulsa en el botón agregar aplicación y se insertan los diferentes atributos relacionados con la misma: nombre de la aplicación, identificador, paquete, icono, descripción y categoría. Una vez seleccionada la aplicación evaluada, se deben seleccionar las tareas a realizar mediante el segundo bloque.

En el bloque de tareas, se muestra una lista que expone las tareas previamente realizadas con aplicaciones de la misma categoría que la aplicación evaluada seleccionada. En la zona verde de la interfaz aparecerán las tareas que el desarrollador ha seleccionado. Para seleccionar alguna de las tareas de la lista, basta con seleccionarla y pulsar el botón "Seleccionar". Una vez está seleccionada, el desarrollador debe introducir las instrucciones concretas para la aplicación evaluada seleccionada mediante el bloque de texto de instrucciones y pulsando el botón "Guardar instrucciones". Si el desarrollador desea introducir tareas que no estén previamente evaluadas, debe añadir una nueva pulsando el botón "Nueva tarea". En este caso se introducen los datos de nombre de tarea y descripción. Una vez completados todos los datos se pulsa el botón "Siguiente".

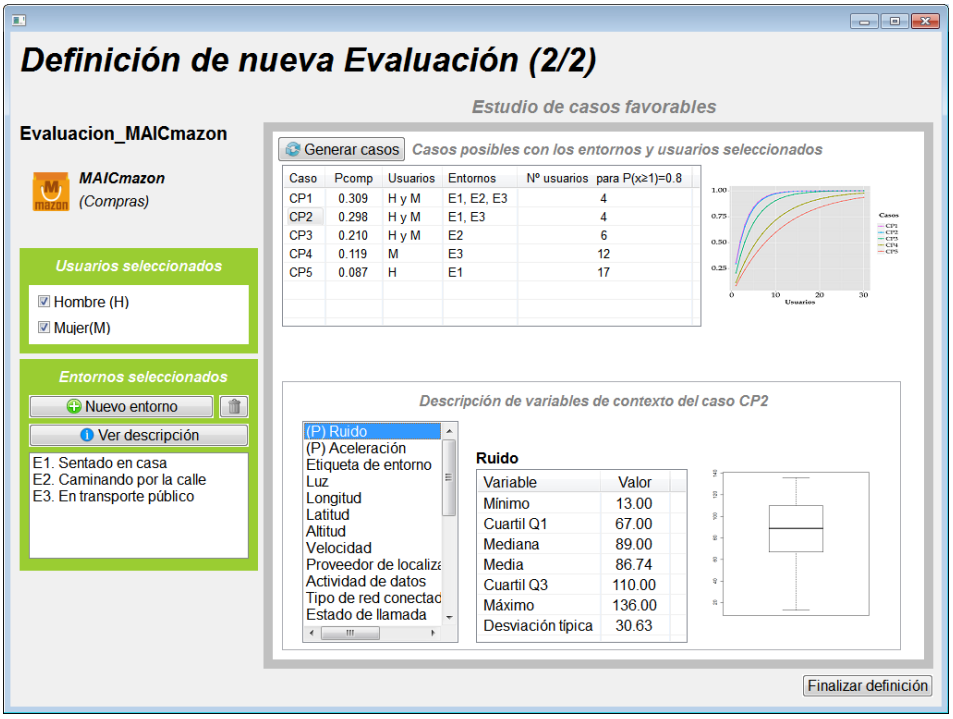

Figura 4.10 Formulario de definición de usuarios y entornos

El segundo y último paso a realizar para completar la definición de la nueva evaluación es la definición de los tipos de usuario y entornos (ver figura 4.10). El

desarrollador debe seleccionar los tipos de usuario para realizar las pruebas. Estos se especifican mediante el bloque "Usuarios seleccionados". Además, también debe escoger los entornos en los cuales desea que los usuarios realicen las pruebas. Se seleccionan mediante el botón "Agregar entorno" del bloque "Entornos seleccionados". Cuando se pulsa dicho botón, se seleccionará un entorno que ya ha sido probado con otras aplicaciones y se pulsará "Aceptar". En el caso de que el desarrollador no encuentre algunos de los entornos que desea probar, debe introducirlos añadiendo a cada uno los atributos de nombre y descripción.

Una vez definidos los tipos de usuario y entornos se podrá realizar un estudio de los casos favorables en base a lo seleccionado para modificar la selección si es necesario.

4.5.1.3. ESTUDIO DE CASOS

Para realizar el estudio de las combinaciones de usuarios y entornos más propensas a detectar errores se hace uso del bloque "Estudio de casos favorables" del formulario expuesto anteriormente en la figura 4.10. Mediante el botón "Generar casos" se generan los casos posibles (ver apartado 3.2.4.1.1) en función de los datos seleccionados. Todos los casos son clasificados en una tabla mediante el criterio de ordenación p_{comp} explicado en el apartado 3.2.4.1.2.

En esta tabla, se podrán comprobar si realmente son necesarios tantos entornos o el uso de ambos tipos de usuario. Una vez presentados los casos, desde un punto de vista sumativo se presentan las variables de contexto para poder visualizar qué valores habían adquirido cuando se han producido errores en aplicaciones anteriores. Para visualizar los datos, simplemente se selecciona un caso y se listan en la parte inferior del formulario las diferentes variables de contexto. En el caso de haber clasificado alguna variable como prioritaria mediante la descripción a nivel inferencial (ver 3.2.4.2.2), aparecerá en el primer lugar de la lista precedida de una P. Si el desarrollador selecciona cualquiera de

las variables que se presentan en la lista, aparecerán los valores correspondientes a la misma. En función de los valores presentados, el desarrollador podrá eliminar algún contexto o tipo de usuario de las pruebas si lo considera necesario. Una vez realizado esto, pulsando el botón "Finalizar definición" se terminará la fase de definición y todos los datos quedarán almacenados en la aplicación web para su posterior uso.

4.5.1.4. GRABACIÓN DE TAREAS

La ayuda que presenta la aplicación de escritorio referente a la fase de ejecución alberga únicamente el primer bloque de pasos: la preparación de las pruebas. Una vez el desarrollador ha integrado la aplicación a evaluar con la librería de integración descrita más atrás, se deben grabar las tareas descritas en la fase de definición para generar los caminos de interacción correcta (ver apartado 3.2.2.3) y posteriormente detectar los posibles errores. Para ello, se debe instalar en un terminal móvil la aplicación evaluada y la aplicación móvil de grabación de caminos de interacción.

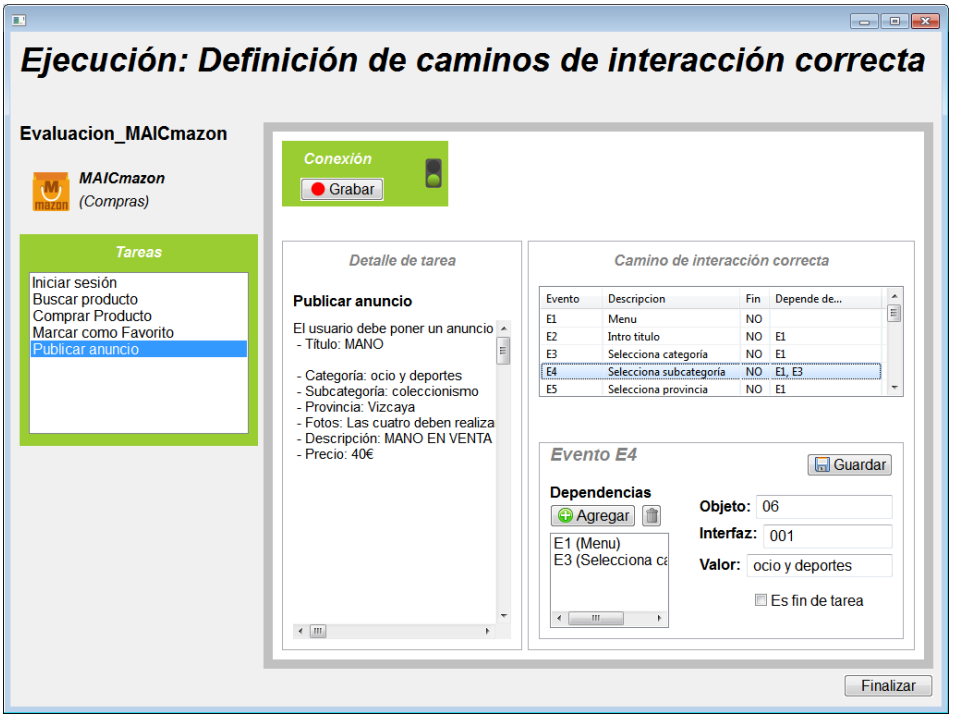

Figura 4.11 Formulario de grabación y generación de caminos de interacción correcta

Una vez instaladas las dos aplicaciones, el desarrollador debe entrar en la interfaz de definición de caminos de interacción correcta (ver figura 4.11) y completar todos los caminos de todas las tareas. Para ello, el desarrollador debe seleccionar una tarea y pulsar el botón "Grabar" dentro del bloque "Conexión", con ello la aplicación se pondrá a la escucha de eventos propagados desde la aplicación móvil de grabación de caminos de interacción.

Paralelamente, el desarrollador debe iniciar en el terminal móvil la aplicación de móvil de grabación de tareas (ver figura 4.12) y conectarse a la aplicación de escritorio introduciendo dirección IP y puerto correspondientes. En el momento de la conexión, se muestra la aplicación a lanzar y bastará con pulsar el botón "Rec" para realizar la tarea y capturar los eventos en la aplicación de escritorio. En el momento en el que el desarrollador haya realizado correctamente la tarea, volverá a la aplicación móvil de grabación y pulsará el botón "Stop".

Figura 4.12 Grabación de eventos desde la aplicación móvil de grabación de caminos de interacción

Gracias a la aplicación móvil de grabación de caminos de interacción, la aplicación de escritorio puede mostrar el camino de interacción correcta mediante la tabla superior. Sin embargo, aunque muestra los valores de los eventos para que se den por satisfactorios, no es posible detectar las dependencias, por lo que inicialmente el campo "Depende de…" aparecerá vacío. Es tarea del desarrollador asignar los eventos de los que depende cada uno de ellos mediante la modificación de las propiedades de los eventos en la zona inferior. Una vez estén todas las dependencias

agregadas de todas las tareas definidas en la fase anterior, el desarrollador pulsará el botón "Finalizar" para comenzar la distribución de la aplicación y el comienzo de la realización de las pruebas por parte de los usuarios mediante la herramienta de usuario de pruebas (ver apartado 4.6).

4.5.1.5. ANÁLISIS DE RESULTADOS

Una vez todos los usuarios de pruebas han terminado de realizar todas las tareas, el desarrollador ya podrá realizar el análisis final de los resultados. Cuando el desarrollador se ha asegurado de que los usuarios han terminado, puede acceder al análisis pulsando el botón "Ver Análisis" de la ventana principal (ver figura 4.8).

Una vez en la interfaz de análisis (ver figura 4.13), el usuario dispone de los dos tipos de análisis descritos en el capítulo anterior: análisis de carácter sumativo y de carácter formativo.

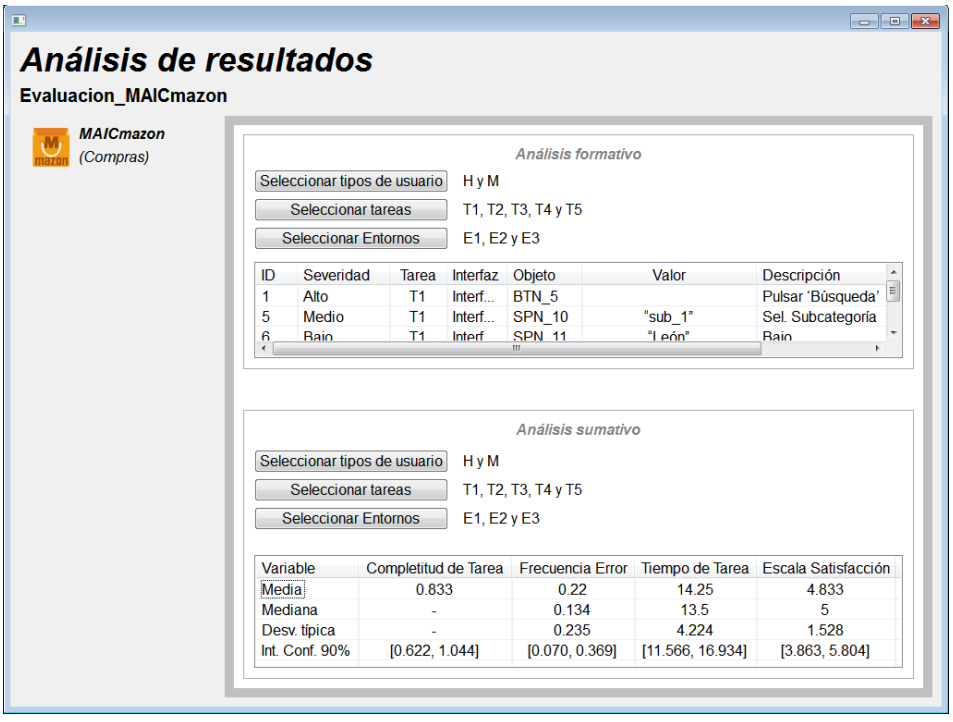

Figura 4.13 Interfaz de análisis de resultados

- En el bloque de análisis sumativo dispone de las variables explicadas en el modelo de análisis sumativo junto con los filtros necesarios para ofrecer tanto la visión global como la

individual (ver apartado 3.2.3.1). Dichos filtros pueden cambiarse mediante los botones visualizados en el bloque.

- El bloque de análisis formativo (ver apartado 3.2.3.2), al igual que el anterior, dispone de los filtros para ofrecer ambas visiones de los problemas originados, mostrando en una tabla los tipos de error, severidad, interfaz, objeto, etc.

Una vez descrita toda la funcionalidad, finalmente presentamos la arquitectura que la sustenta.

4.5.2. ARQUITECTURA DE LAS HERRAMIENTAS DE DESARROLLADOR

Para dar soporte a toda la funcionalidad expuesta, presentamos la arquitectura implementada. Como se muestra en la figura 4.14, la aplicación de escritorio está compuesta por varios módulos cuya función está claramente definida acorde a las fases de la metodología. La aplicación móvil de grabación de caminos de interacción no dispone de suficiente complejidad como para mostrar una división modular detallada.

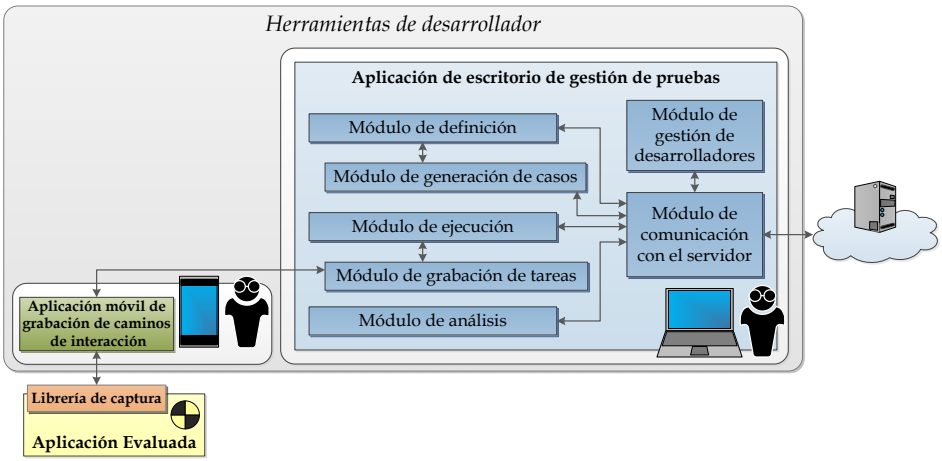

Figura 4.14 Arquitectura de la aplicación de escritorio y grabación de tareas

- El *módulo de gestión de desarrolladores* tiene como principal objetivo realizar el registro de los nuevos desarrolladores en la plataforma de soporte y la modificación de las credenciales de los mismos.

- El *módulo de definición* registra todas las propiedades de la definición de una nueva evaluación. Esto incluye la definición y categorización de la aplicación evaluada, las tareas que los usuarios deben realizar, además de la definición de los usuarios y entornos.
- El *módulo de generación de casos* es el encargado de realizar los cálculos para la ayuda en la definición de las pruebas. Por ello, es un módulo que está estrechamente ligado al módulo de definición ya que los casos favorables son generados mediante los datos introducidos en este módulo y las evaluaciones similares almacenadas en la base de conocimiento.
- El *módulo de ejecución* dispone de las operaciones necesarias para la creación de los caminos de interacción correcta de cada tarea. Para ello recupera las tareas asignadas en el módulo de definición para la nueva evaluación y completa el camino de interacción correcta con las dependencias entre eventos. El camino de interacción correcta es grabado mediante el módulo de grabación de tareas.
- El *módulo de grabación de tareas* se comunica con la aplicación móvil de grabación de caminos de interacción para grabar los eventos de interacción recibidos y generar el camino de interacción correcta.
- La *aplicación móvil de grabación de caminos de interacción* se comunica mediante un socket TCP con el módulo de grabación de tareas para saber qué aplicación debe ejecutar y grabar los eventos de interacción generados por la misma. Los mensajes de sistema que contienen eventos de interacción y son generados por la librería de integración se propagan al módulo de grabación de tareas, para así crear el camino de interacción correcta y ser completado en el módulo de ejecución.
- El *módulo de análisis* recupera las pruebas ejecutadas y enviadas por los usuarios de pruebas. Con ellas, analiza y expone los resultados.

4.6. HERRAMIENTA DE USUARIO DE PRUEBAS

En la realización de las pruebas referente al segundo conjunto de pasos dentro de la fase de ejecución de la metodología, los usuarios de pruebas deben realizar las pruebas con su terminal móvil y enviar los resultados de las mismas. Para realizar esto, deben descargar e instalar la aplicación evaluada en sus terminales, tener claras las tareas que deben realizar y en qué contextos para posteriormente enviar los resultados una vez terminadas las pruebas.

> *La herramienta de usuario de pruebas consiste en una aplicación móvil cuyo objetivo es ayudar a los usuarios en el proceso de la realización de pruebas.*

Ésta aplicación registra a nuevos usuarios de pruebas, gestiona las pruebas que deben realizarse, captura los datos de la ejecución de las tareas y los envía a la aplicación web.

4.6.1. FUNCIONALIDAD

Como se ha mencionado, esta aplicación móvil ofrece cuatro tipos de funcionalidades. Son mostradas a continuación cronológicamente y en función de los pasos de la metodología.

Fase	*Paso de la metodología*	*Funcionalidad*
-	-	*Registro de usuario de pruebas*
Ejecución	*Descarga de instrucciones y aplicación evaluada*	*Gestión de pruebas*
Ejecución	*Ejecución de tareas*	*Captura de pruebas*
Ejecución	*Subida de información de tareas realizadas*	*Subida de resultados*

Tabla 4.3 Relación de pasos con la funcionalidad de la aplicación móvil de captura de pruebas

4.6.1.1. REGISTRO DE USUARIO DE PRUEBAS

La primera acción que debe realizar un usuario de pruebas es la instalación de la aplicación móvil de captura de pruebas y el registro dentro de la plataforma de soporte.

Para realizar el registro del mismo, la aplicación muestra un pequeño asistente (ver figura 4.15) que consta de tres principales pasos. Con éstos se inserta manualmente la información referente al usuario de pruebas y automáticamente las propiedades del dispositivo.

Figura 4.15 Asistente de registro de un usuario de pruebas

Ésta información se almacena en la base de conocimiento del servidor mediante la comunicación de esta aplicación con la aplicación web de gestión de la base de conocimiento. La información almacenada mediante este asistente es la mostrada en la tabla 4.4.

Nombre de variable	*Modelo de contexto*	*Fuente*
Alias de usuario	*Usuario(datos personales)*	*Usuario(manual)*
Contraseña	*--*	*Usuario(manual)*
Fecha nacimiento	*Usuario(datos personales)*	*Usuario(manual)*
Género	*Usuario(datos personales)*	*Usuario(manual)*
Altura	*Usuario(datos personales)*	*Usuario(manual)*
Peso	*Usuario(datos personales)*	*Usuario(manual)*
Lateralidad	*Usuario(capacidades físicas)*	*Usuario(manual)*
Problema de audición	*Usuario(capacidades sensoriales)*	*Usuario(manual)*
Problema de visión	*Usuario(capacidades sensoriales)*	*Usuario(manual)*
Nivel de inglés	*Usuario(capacidades cognitivas)*	*Usuario(manual)*
Nivel de castellano	*Usuario(capacidades cognitivas)*	*Usuario(manual)*
Nivel de portugués	*Usuario(capacidades cognitivas)*	*Usuario(manual)*
Nivel de alemán	*Usuario(capacidades cognitivas)*	*Usuario(manual)*
Nivel de francés	*Usuario(capacidades cognitivas)*	*Usuario(manual)*
Identificador de terminal	*Dispositivo(datos generales)*	*Sistema Operativo*
Fabricante	*Dispositivo(datos generales)*	*Sistema Operativo*
Modelo	*Dispositivo(datos generales)*	*Sistema Operativo*
País de fabricación	*Dispositivo(datos generales)*	*Sistema Operativo*
Pantalla	*Dispositivo(salidas)*	*Sistema Operativo*
Versión del sistema operativo	*Dispositivo(software)*	*Sistema Operativo*
Lenguaje del terminal	*Dispositivo(software)*	*Sistema Operativo*

Tabla 4.4 Variables de contexto capturadas en el registro del usuario de pruebas

4.6.1.2. GESTIÓN DE PRUEBAS

Cuando el usuario ya está registrado en la plataforma de soporte, deben descargarse las evaluaciones en fase de ejecución con las tareas a realizar (ver figura 4.16). Para ello, se dispone de una interfaz que comprueba si nuevas evaluaciones han sido asignadas al usuario. En caso afirmativo, éste podrá descargarlas mediante un simple botón. Cuando ya se disponen de las evaluaciones descargadas, se presenta un listado de las mismas. Pulsando sobre cada una de ellas se muestran las tareas a realizar en dicha evaluación junto con sus detalles.

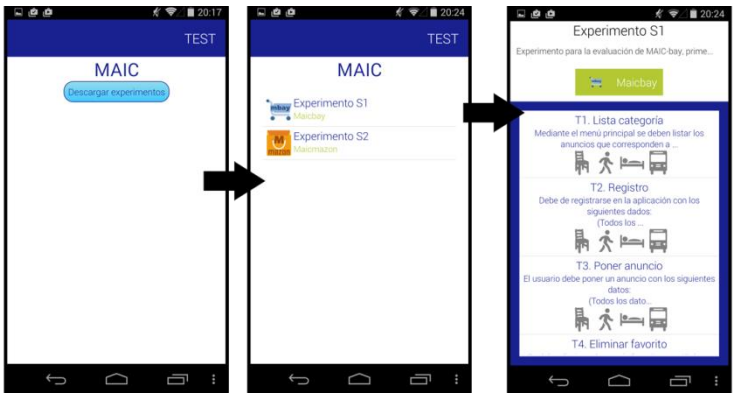

Figura 4.16 Descarga de evaluaciones a realizar por el usuario de pruebas

Además, el usuario de pruebas podrá descargar e instalar las aplicaciones evaluadas que deba usar para desarrollar las tareas de las evaluaciones descargadas en la fase de ejecución de la metodología. El usuario accederá al panel de aplicaciones donde aparecerán las aplicaciones necesarias a descargar. Pulsando en el botón "Instalar" se descargará la aplicación y se ejecutará su instalador.

4.6.1.3. CAPTURA DE PRUEBAS

Cuando el usuario desea realizar una tarea y se encuentra en el entorno adecuado, debe seleccionar la tarea que desea realizar. Después, selecciona el entorno en el que está, lee el detalle de las instrucciones de la tarea y cuando comprende lo que debe hacer, pulsa el botón "Comenzar" (ver figura 4.17).

Figura 4.17 Selección de tarea y entorno

En ese momento, la aplicación móvil de captura de pruebas lanzará la aplicación evaluada correspondiente a la tarea a realizar (ver figura 4.18). Además, en ese preciso momento se generará y almacenará el evento de tarea "comienzo de tarea". Paralelamente, se comenzará a capturar las variables dinámicas de contexto.

En la tabla 4.5 se muestran las diferentes variables de contexto que son capturadas y el medio usado para obtener su valor.

Nombre de variable	*Modelo de contexto*	*Fuente*
Etiqueta de entorno	*Entorno (datos generales)*	*Usuario(manual)*
Iluminancia	*Entorno (físico)*	*(Sensor) Sensor de luz*
Ruido	*Entorno (físico)*	*(Sensor) Micrófono*
Longitud	*Entorno (físico)*	*(Sensor) GPS y celdas*
Latitud	*Entorno (físico)*	*(Sensor) GPS y celdas*
Altitud	*Entorno (físico)*	*(Sensor) GPS y celdas*
Velocidad	*Entorno (físico)*	*(Sensor) GPS y celdas*
Aceleración	*Dispositivo (datos generales)*	*(Sensor) Acelerómetro*
Proveedor de localización(GPS/red)	*Entorno (técnico)*	*Sistema Operativo*
Actividad de datos	*Entorno (técnico)*	*Sistema Operativo*
Tipo de red conectada	*Entorno (técnico)*	*Sistema Operativo*
Estado de llamada	*Entorno (técnico)*	*Sistema Operativo*
Orientación de pantalla	*Dispositivo (entradas)*	*Sistema Operativo*
Modo de timbre	*Dispositivo (salidas)*	*Sistema Operativo*
Estado de altavoces(on/off)	*Dispositivo (salidas)*	*Sistema Operativo*
Estado de micrófono (on/off)	*Dispositivo (entradas)*	*Sistema Operativo*
Estado de música (on/off)	*Dispositivo (salidas)*	*Sistema Operativo*
Volumen de timbre	*Dispositivo (salidas)*	*Sistema Operativo*
Auriculares conectados (on/off)	*Dispositivo (salidas)*	*Sistema Operativo*
Volumen del sistema	*Dispositivo (salidas)*	*Sistema Operativo*
Volumen de llamada	*Dispositivo (salidas)*	*Sistema Operativo*
Nivel de batería	*Dispositivo (batería)*	*Sistema Operativo*
Voltaje de batería	*Dispositivo (batería)*	*Sistema Operativo*
Temperatura de batería	*Dispositivo (batería)*	*Sistema Operativo*
Fuente de carga	*Dispositivo (batería)*	*Sistema Operativo*
Salud de la batería	*Dispositivo (batería)*	*Sistema Operativo*

Tabla 4.5 Variables de contexto capturadas durante la ejecución de una tarea

Una vez el usuario de pruebas cree que ha realizado la tarea, vuelve a la aplicación móvil de captura y notifica que ha finalizado pulsando el botón "Finalizar tarea", con ello se genera el evento de tarea "fin de tarea". Seguidamente, la captura de las variables de contexto finaliza y se presenta el cuestionario de 7 preguntas expuestas en la tabla 3.10, que ayuda en el cálculo de nivel sumativo (ver apartado 3.2.3.1). Después de responder, los resultados se almacenan localmente para su posterior subida.

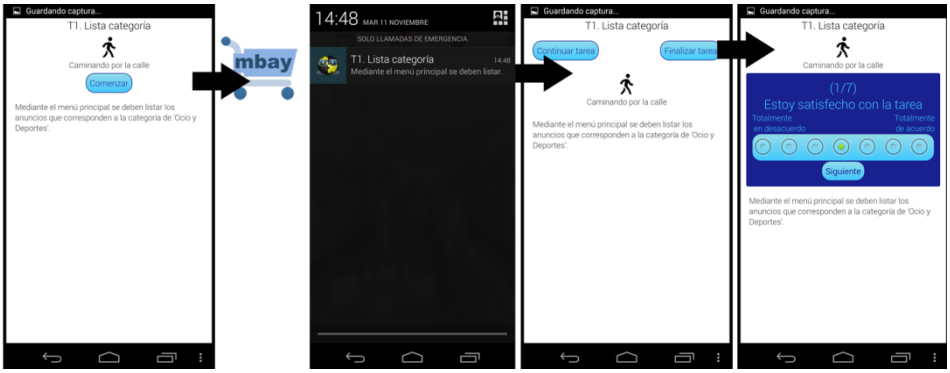

Figura 4.18 Finalización de tarea y cuestionario

4.6.1.4. SUBIDA DE RESULTADOS

Cuando las tareas de una evaluación han sido desarrolladas, el usuario de pruebas debe transferir todos los resultados a la base de conocimiento. Debido a esto, si hay tareas ya realizadas y resultados no transferidos, aparecerá el mensaje "*Hay datos para enviar*" en la parte superior de la pantalla principal. Pulsando este mensaje aparecerá el botón "Enviar datos" que inicia la subida.

Figura 4.19 Subida de resultados a la aplicación web

El objetivo de dar la opción de subir los resultados cuando el usuario de pruebas lo desee es el no afectar a la conexión de datos en el caso de estar realizando tareas de alguna evaluación, limitada en el caso de no utilizar conexión de tipo WiFi.

Una vez explicada la funcionalidad de la aplicación, describimos la arquitectura implementada que la posibilita.

4.6.2. ARQUITECTURA DE LA APLICACIÓN

Como se muestra en la figura 4.20, la aplicación móvil de captura de pruebas está compuesta por dos principales grupos de módulos: grupo de gestión y grupo de captura.

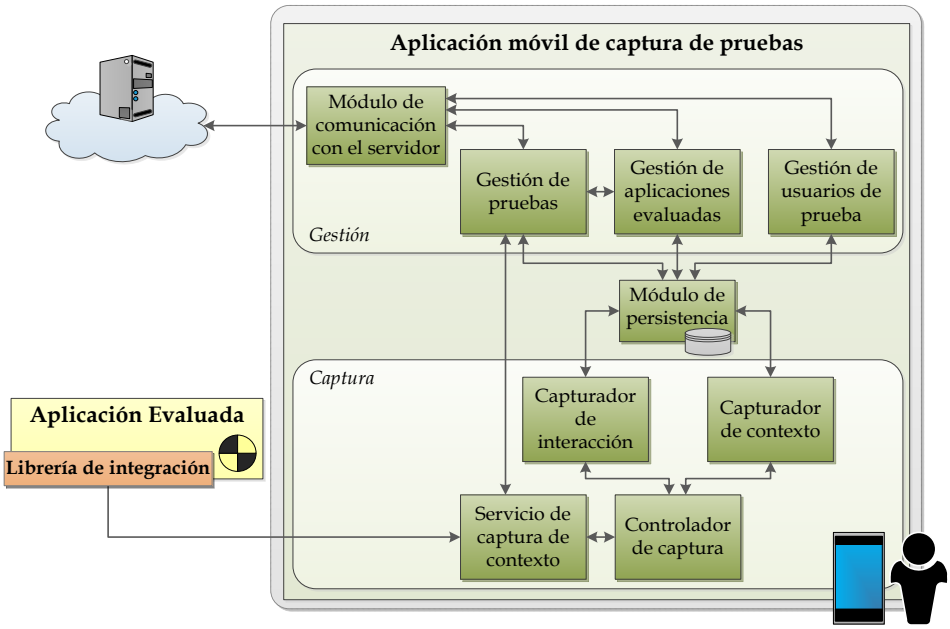

Figura 4.20 Arquitectura de la aplicación móvil de captura de pruebas

Todos estos módulos giran en torno a una pequeña base de datos que almacena tanto los datos generados como resultados como los datos referentes a las pruebas a realizar.

- El *módulo de persistencia* tiene como objetivo gestionar la pequeña base de datos que almacena los datos generados por la aplicación y satisfacer las demandas de lectura y

escritura de los diferentes hilos de ejecución durante las capturas de contexto e interacción.

El grupo de gestión lo componen módulos que se centran en la administración de los usuarios, pruebas y aplicaciones. Esta administración está ligada a la base de conocimiento, por lo que estos módulos se comunican con la aplicación web de gestión de la base de conocimiento (ver apartado 4.4).

- El *módulo de comunicación con el servidor* tiene como objetivo realizar las llamadas al servicio web de usuarios de pruebas para satisfacer la funcionalidad de los siguientes módulos de gestión: módulo de gestión de usuarios de pruebas, módulo de gestión de aplicaciones evaluadas y módulo de gestión de pruebas.
- El *módulo de gestión de usuarios de pruebas* registra este tipo de usuarios y modifica sus atributos. En el proceso de registro, éste módulo se comunica con la aplicación web de pruebas mediante el módulo de comunicación con el servidor para dejar constancia del registro.
- El *módulo de gestión de aplicaciones* es el encargado de cerciorarse de que las aplicaciones evaluadas están correctamente instaladas en el sistema y lanzarlas cuando el usuario decide realizar una tarea de una aplicación evaluada concreta.
- El *módulo de gestión de pruebas* es el módulo que administra las diferentes pruebas que el usuario debe realizar. Descarga mediante el módulo de comunicación con el servidor, muestra las instrucciones de las tareas a realizar y decide cuándo deben almacenarse los resultados de pruebas realizadas en la base de conocimiento.

Por otro lado, el grupo de captura contiene los módulos cuyo objetivo es capturar la interacción y el contexto.

- El *servicio de captura de contexto* es un componente Android de tipo Service que recibe los mensajes del sistema enviados por la librería de integración (mensajes con

eventos de pausa de tarea, continuación de tarea e interacción) y del módulo de gestión de pruebas (mensajes con eventos de comienzo de tarea y fin de tarea). Los eventos que contienen dichos mensajes (tanto de interacción como de tarea), son extraídos y propagados al controlador de captura.

- El *controlador de captura* es el encargado de orquestar a los dos módulos que capturan los datos de la realización de la tarea (capturador de interacción y capturador de contexto) en función de los eventos que le propague el servicio de captura de contexto.
- El *capturador de interacción* recibe los eventos de tipo interacción. La principal función que desempeña este módulo es el determinar el tipo de evento de interacción que es (ver apartado 3.2.2.2) mediante el camino de interacción correcta correspondiente a la tarea en ejecución. Una vez completados todos los atributos del evento, se propaga al módulo de persistencia para su almacenamiento.
- El *capturador de contexto* recoge los valores de todas las variables dinámicas de contexto (ver tabla 4.5). Estos valores son recogidos mediante consultas al sistema operativo y el acceso a los diferentes sensores del terminal. Este módulo es el más complejo y el que más carga de trabajo demanda debido al número de hilos de ejecución que se originan y al número de sensores que se activan.

Con esta descripción queda completada la exposición de la plataforma de soporte.

CAPÍTULO 5

EXPERIMENTACIÓN Y VALIDACIÓN

«Nada ocurre porque sí. Todo en la vida es una sucesión de hechos que, bajo la lupa del análisis, responden perfectamente a causa y efecto»,
Richard Feynman (1918-1988)

ÍNDICE DE CAPÍTULO 5

5.1. Validación de la estrategia de captura de los modelos 174
5.1.1. Descripción del experimento 175
5.1.2. Resultados 176
5.1.3. Consideraciones de la estrategia de captura 185
5.2. Validación del uso de la base de conocimiento 186
5.2.1. Descripción del experimento 187
5.2.2. Resultados 190
5.2.3. Consideraciones del uso de la base de conocimiento 199
5.3. Validación de la metodología 200
5.3.1. Descripción del experimento 202
5.3.2. Resultados 215
5.3.3. Consideraciones de la metodología 221

Al igual que en cualquier trabajo de investigación, una de las fases clave es la validación de las hipótesis formuladas. Por ello, el objetivo de este capítulo es precisamente describir los procesos realizados para esta validación, dividida en tres partes: validación de la estrategia de captura que genera la base de conocimiento, su uso y la metodología en sí. Por cada una de las partes, presentaremos un análisis del objeto de validación y la conclusión de las estrategias de validación más adecuadas. Después, presentaremos las actividades de experimentación utilizadas junto con los resultados para finalmente concluir la veracidad o falsedad de las afirmaciones presentadas.

Para la validación de esta tesis debemos cerciorarnos de que la solución propuesta cumple las características presentadas en el capítulo 1 (ver sección 1.3). Recapitulando las mismas, algunas de estas características pueden ser confirmadas mediante el propio diseño de la metodología y de la plataforma de soporte, los cuales han sido expuestos en los capítulos 3 y 4 respectivamente.

Las características que se confirman con el diseño son:

- *Debe ser capaz de ofrecer resultados para evaluaciones cuya finalidad sea tanto formativa como sumativa*. Gracias tanto a los componentes del modelo de análisis (ver apartado 3.2.3) como a la funcionalidad de análisis de la herramienta de desarrollador (ver apartado 4.5.1.5) es posible realizar ambos tipos de evaluación.
- *Se debe preservar la privacidad de los usuarios que realizan las pruebas*. Mediante la funcionalidad de registro que ofrece la herramienta de usuario de pruebas mediante su propio terminal móvil (ver apartado 4.6.1) y el modelo de contexto de la base de conocimiento diseñado para esta metodología (ver apartado 3.2.1), el usuario puede realizar pruebas sin haber contactado nunca con el evaluador y sin registrar datos que faciliten la identificación del mismo.
- *Se debe posibilitar el estudio de un modelo de contexto detallado*. Gracias al modelo de contexto y el modelo de casos favorables de la metodología (ver apartado 3.2.4), la funcionalidad de captura de la herramienta de usuario de pruebas y las funcionalidades de las herramientas de desarrollador tanto de análisis como de estudio de casos (ver apartado 4.5.1) se posibilita el estudio de las principales características del contexto de entornos móviles.

Por el contrario, otras características deben ser validadas de un modo más detallado mediante experimentación. Dentro de este grupo entran las características que se ligan directamente con la hipótesis formulada en este trabajo (ver sección 1.2).

- *La calidad de los resultados no debe disminuir*. Al realizar la captura de las pruebas de usabilidad mediante el uso de los diferentes terminales, debemos asegurar que no se genera ningún sesgo por la estrategia de captura planteada. Por ello, *planteamos confirmar que podemos capturar la interacción y el contexto sin sesgar los resultados utilizando el propio dispositivo móvil* (ver sección 5.1).
- *La cantidad de recursos necesaria debe ser reducida*. En términos de equipamiento, el diseño de la plataforma de soporte solo necesita el uso de los dispositivos móviles de los usuarios de pruebas y un servidor para centralizar la información. Sin embargo, si nos centramos en la reducción del tiempo es necesario validar la utilidad de la base de conocimiento. Como novedad en esta solución, hacemos uso de la misma para elegir las características de las pruebas más adecuadas. Por ello, *debemos validar que podemos detectar los casos favorables y valores de las variables de contexto más adecuados para la evaluación de una nueva aplicación* (ver sección 5.2).

Una vez planteada las validaciones de las dos características anteriores de un modo independiente, también planteamos la validación integral de la metodología (ver sección 5.3) para confirmar la principal hipótesis de esta tesis.

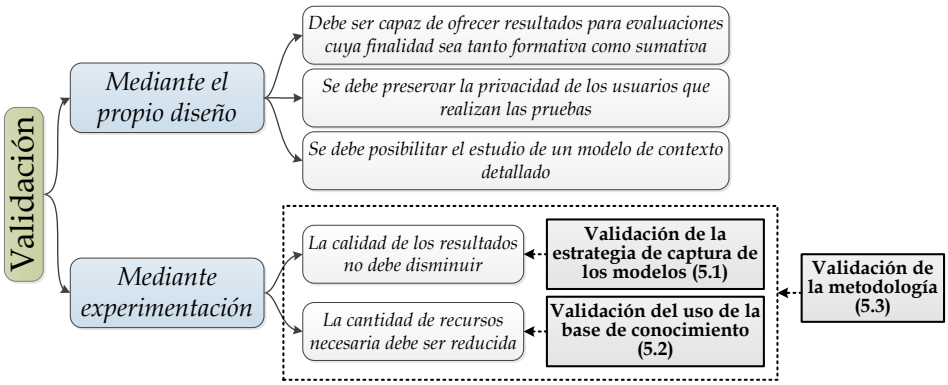

Figura 5.1 Esquema de la validación

Resumiendo lo planteado mediante la figura 5.1, procedemos a explicar las validaciones mediante la experimentación definida.

5.1. VALIDACIÓN DE LA ESTRATEGIA DE CAPTURA DE LOS MODELOS

Para poder cumplir con el objetivo de capturar los modelos de contexto e interacción y poder posteriormente hacer un análisis, debemos asegurarnos de validar lo siguiente:

> *Es posible capturar la interacción y el contexto realizando tareas para evaluaciones de usabilidad de aplicaciones móviles en entornos reales sin sesgar los resultados utilizando el propio dispositivo móvil.*

Para ello debemos estudiar cómo afecta la ejecución de la herramienta de usuario de pruebas en la captura de información referente al modelo de contexto e interacción.

Dentro de la definición del modelo de contexto en el apartado 3.2.1, recordamos que disponíamos de cinco principales pilares que pueden ser afectados por la herramienta de pruebas: aplicación, usuario, dispositivo, entorno y tarea. Dentro de estos pilares consideramos que solo tres son afectados por la plataforma. El *usuario de pruebas* porque siempre es consciente de que está realizando pruebas para una evaluación de usabilidad. El *dispositivo* por la reducción de los recursos del mismo al ejecutar y llevar instalada la propia herramienta de usuario de pruebas. Y finalmente, el proceso de la *aplicación evaluada* al tener que llamar a funciones de la librería de integración y al disponer de menos recursos del dispositivo, utilizados por la herramienta de usuario de pruebas. Además, dentro del modelo de interacción definido en el apartado 3.2.2, el tiempo en el que se producen los eventos de interacción y tarea también puede verse afectados al modificarse la interacción por los sesgos añadidos en el modelo de contexto.

Por ello se plantea una validación que nos permitirá estudiar estos elementos en base a la medición del rendimiento de los dispositivos y la medición del tiempo de ejecución de ciertos procesos. Por un lado, *estudiaremos que el sesgo que se produce en la aplicación, dispositivo y eventos de interacción no muestra un efecto*

significativo. Por otro lado, *analizaremos la percepción del usuario mediante la comprobación de que se mantiene la instantaneidad de la interacción con las aplicaciones a pesar de añadir la captura de datos*.

5.1.1. DESCRIPCIÓN DEL EXPERIMENTO

Para lograr una cuantificación exacta y objetiva, optamos por la simulación de la interacción de usuarios de pruebas utilizando la herramienta Robotium[35]. Dicha herramienta permite una automatización de pruebas robustas de interfaz que simula interacciones reales. Para capturar las diferentes variables analizadas, se conecta el dispositivo a un ordenador que almacena la salida estándar del dispositivo (por donde van quedando registradas las variables medidas) en un fichero para su posterior análisis. Mediante el uso de la simulación de pruebas conseguimos realizar exactamente la misma tarea con las mismas interacciones justo en el mismo instante, prescindiendo así de usuarios reales. Gracias a ello, realizaremos una comparación objetiva y con suma precisión, garantizando así la imparcialidad de los resultados.

El conjunto de ejecuciones de tareas que hemos programado y definido corresponden a una de las aplicaciones desarrolladas para la validación del uso de la base de conocimiento: *Maicbay*. Como se muestra en la tabla 5.1, disponemos de cuatro ejecuciones que corresponden a dos tareas. Una de ellas consiste en el listado de productos de una categoría concreta y la otra en el registro de un nuevo usuario dentro de la aplicación.

Ejecución	*Tarea a realizar*	*Características*
T1_1	*Listado de productos*	*No hay errores de interacción*
T1_2	*Listado de productos*	*Se selecciona incorrectamente un menú pero se corrige*
T2_1	*Registro de usuario*	*No hay errores de interacción*
T2_2	*Registro de usuario*	*Se escribe incorrectamente el nombre y se corrige el error*

Tabla 5.1 Ejecuciones de tareas desarrolladas para el estudio del sesgo en el dispositivo, aplicación e interacción

[35] https://code.google.com/p/robotium

Debido a la heterogeneidad de los dispositivos móviles, se ha optado por la ejecución de las tareas en varios modelos. Para disponer de una muestra significativa, se realizan 300 repeticiones de las cuatro ejecuciones capturando los modelos con la herramienta y otras 300 sin la herramienta, todas ellas por cada dispositivo. Los terminales móviles han avanzado en capacidad muy rápidamente por lo que hemos realizado las diferentes simulaciones y mediciones con cuatro modelos de terminal cuyos lanzamientos distan de un año: desde 2010 a 2013.

Dispositivo (fecha lanzamiento)	*Procesador(núcleos)*	*Memoria*	*Almacenamiento*
HTC Desire HD (09/2010)	*1 GHz (1)*	*768 MB*	*8 GB*
Samsung Galaxy Nexus (10/2011)	*1.2 GHz (2)*	*1 GB*	*16 GB*
Samsung Nexus 10 (10/2012)	*1.7 GHz (2)*	*2 GB*	*32 GB*
LG Nexus 5 (10/2013)	*2.3 GHz (4)*	*2 GB*	*32 GB*

Tabla 5.2 Modelos utilizados para el estudio del sesgo en el dispositivo, aplicación e interacción

Una vez realizadas las simulaciones, cuya duración fue de más de 14 horas, generamos 1200 tareas realizadas y capturadas con la herramienta de usuario de pruebas y otras 1200 tareas manteniendo la aplicación evaluada sin cambios para realizar la captura. A continuación procedemos a su análisis en función del elemento afectado mediante el programa estadístico R [Team12].

5.1.2. RESULTADOS

Como ya hemos avanzado, realizaremos el análisis de los resultados mediante dos perspectivas: analizaremos que el sesgo que se produce en la aplicación, dispositivo y eventos de interacción no muestra un efecto significativo. Por otro lado, analizaremos que la percepción del usuario no se altera por la captura de los modelos.

5.1.2.1. SESGO EN LA APLICACIÓN, DISPOSITIVO Y EVENTOS DE INTERACCIÓN

Dentro de este grupo de pilares destacamos la dependencia que hay entre los mismos. Por un lado, si los recursos del dispositivo móvil se ven reducidos considerablemente por los procesos generados por la herramienta de usuario de pruebas, los procesos de la aplicación evaluada también dispondrán de menos recursos.

Debido a esto, demandarán más tiempo para realizar las actividades de proceso correspondientes y finalmente se verá traducido en una interacción ligeramente más lenta.

Para asegurarnos de que los sesgos añadidos entran dentro de unos límites aceptables, debemos cuantificarlos. Por ello, identificamos varias variables a estudiar en función del elemento que se ve involucrado.

- Desde el punto de vista del dispositivo entendemos que los recursos críticos del mismo que pueden sufrir una reducción significativa son: *el uso de procesador, la memoria física disponible y el espacio disponible en el almacenamiento interno del dispositivo*. Descartamos la reducción del ancho de banda ya que en el momento de la ejecución de las tareas de la evaluación, la plataforma de soporte no demanda el uso de la conectividad móvil. Para realizar dicha cuantificación, debemos comparar el estado de los recursos cuando la herramienta de usuarios de prueba está en ejecución con el estado de los mismos cuando no lo está.
- Desde el punto de vista de aplicación evaluada, debemos ser conscientes de que puede disminuir la velocidad de sus procesos, no sólo por la reducción de recursos del dispositivo, sino también por los ciclos de ejecución dedicados a las llamadas a las funciones de la librería de integración. Por lo tanto, deberemos medir el *tiempo que dedican los procesos de la aplicación evaluada a realizar las llamadas a las funciones de la librería de integración*.
- Finalmente, debemos medir si existe un aumento significativo del *tiempo requerido para la ejecución de las tareas* por parte de los usuarios de pruebas. De un modo similar al de los recursos del dispositivo, debemos comparar el tiempo necesario para la ejecución de unas tareas concretas cuando la herramienta de usuarios de prueba está en ejecución con el tiempo necesario cuando no lo está.

5.1.2.1.1. SESGO EN EL DISPOSITIVO

En primer lugar, procedemos al estudio de la capacidad de proceso necesaria para capturar los modelos con la herramienta.

Si observamos los datos correspondientes al uso del procesador, sí detectamos fuertes cambios en el uso del mismo al comparar ambos grupos: con uso de la herramienta y sin su uso. Observamos en la tabla 5.3 que en los intervalos de confianza de la diferencia de medias estimadas no aparece el valor cero y todas las estimaciones son positivas, por lo que *sí notificamos un incremento significativo en el uso de procesador*.

Dispositivo	*Uso con herramienta (%)*	*Uso sin herramienta (%)*	*Intervalo de confianza [95%] de la diferencia estimada (%)*
HTC Desire HD	*64.91*	*28.93*	*[35.44,36.53]*
S. Galaxy Nexus	*60.36*	*26.06*	*[33.9,34.7]*
S. Nexus 10	*39.00*	*24.29*	*[14.29,15.13]*
LG Nexus 5	*26.83*	*17.14*	*[9.37,10.03]*

Tabla 5.3 Diferencias del uso medio estimado de procesador con y sin herramienta

Por otro lado, plasmando las distribuciones en la figura 5.2 apreciamos que a medida que mejoramos las capacidades del terminal, lógicamente reducimos la diferencia entre los grupos. Cuantificando esta tendencia apreciamos una sustancial mejora donde *disminuimos de una demanda estimada en el terminal más antiguo en torno al 36% a una demanda estimada en el terminal más moderno en torno al 9.7% de su capacidad de proceso*.

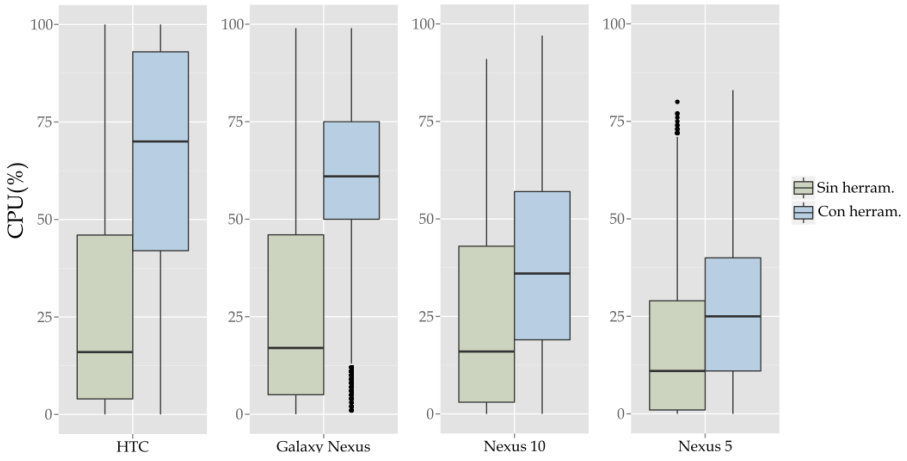

Figura 5.2 Evolución del uso medio estimado de procesador por dispositivo

Para el estudio de la memoria física se ha realizado el mismo procedimiento que con la capacidad de proceso. Observando los resultados mediante la descripción de los mismos en la figura 5.3, apreciamos que existen múltiples casos. Por un lado, encontramos resultados que indican un ligero aumento, como en el caso del Nexus 5. Por otro lado, observamos el caso contrario donde disminuye la memoria como en el Nexus 10 producido posiblemente por el recolector de basura o Garbage Collector del dispositivo. Además, nos encontramos un caso donde no hay una diferencia significativa como en el terminal Galaxy Nexus.

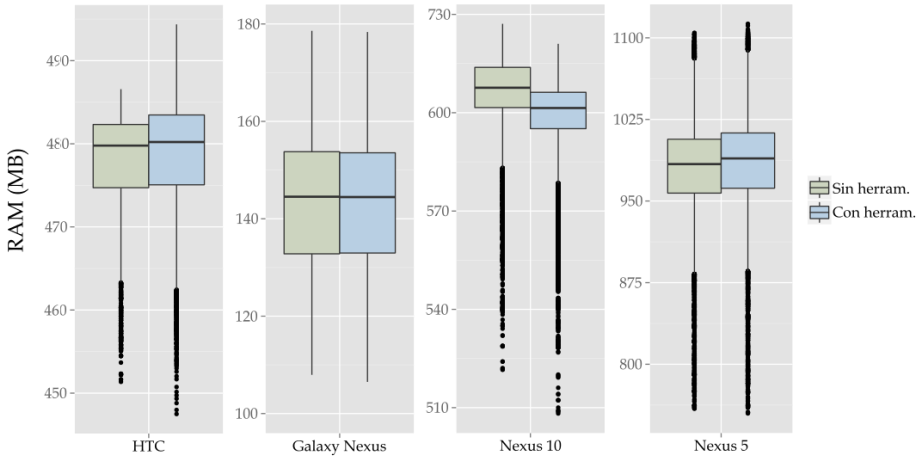

Figura 5.3 Comparación del uso de memoria física con y sin captura de modelos

Observando las diferencias estimadas mediante la tabla 5.4 y viendo que sólo hay un aumento significativo y nunca superior a los 5.8 MB, llegamos a la siguiente conclusión: *con las evidencias de los datos recuperados del experimento deducimos que no hay un aumento significativo de la memoria física*.

Dispositivo	*Memoria con herramienta (MB)*	*Memoria sin herramienta (MB)*	*Intervalo de confianza [95%] de la diferencia estimada (MB)*
HTC Desire HD	*478.94*	*477.82*	*[1.01, 1.24]*
S. Galaxy Nexus	*554.89*	*555.06*	*[-0.46, 0.12]*
S. Nexus 10	*599.65*	*606.4*	*[-6.94, -6.55]*
LG Nexus 5	*983.8*	*978.89*	*[4.02, 5.8]*

Tabla 5.4 Diferencias del uso medio estimado de memoria física con y sin herramienta

Para el estudio del almacenamiento, debemos añadir al tamaño fijo de la herramienta de usuario de pruebas (2.34 MB), el ligero tamaño de las instrucciones de las tareas (4kB cada conjunto de instrucciones) y el tamaño de los datos referentes a los modelos capturados por cada ejecución de tarea, ya que hacemos uso del propio terminal móvil. Hemos decidido utilizar el almacenamiento del propio dispositivo en el momento de la captura de los modelos para no reducir el ancho de banda de la conexión, al ser éste un recurso más sensible. Por ello se ha realizado la medición del espacio de almacenamiento disponible en los dispositivos antes y después de cada tarea capturada.

El tiempo de las ejecuciones con algunos dispositivos duró más de 14 horas. Dentro de este tiempo, recordamos que se capturaron 1200 tareas simuladas con la herramienta de usuario de pruebas capturando. Mencionamos estos datos porque asumimos que la situación de realizar más de 14 horas de tareas sin antes haber realizado una subida de los resultados generados es poco probable. Viendo que el modelo de datos es el mismo independientemente del dispositivo se ha agrupado el espacio que se demanda por tipo de tarea realizada.

Como muestra la tabla 5.5, *la ejecución de tareas dentro de unos márgenes normales no implica la generación de una amplia demanda de almacenamiento*, siendo ésta para cada tarea realizada de 9.5kB estimados. Suponiendo las 2400 tareas totales como capturadas, el almacenamiento demandado total no superaría los 23MB.

Tarea	*Uso de memoria con herramienta (KB)*	*Intervalo de confianza [95%] del espacio requerido estimado (kB)*
T1_1	*4.84*	*[4.65, 5.02]*
T1_2	*9.56*	*[9.28, 9.85]*
T2_1	*10.03*	*[9.75, 10.31]*
T2_2	*13.67*	*[13.34, 14]*
Todas	*9.52*	*[9.35, 9.69]*

Tabla 5.5 Kilobytes estimados requeridos para el almacenamiento de los modelos capturados por tarea realizada

Suponiendo un caso excepcional con un terminal con instrucciones de 100 tareas similares a realizar en 4 entornos diferentes, su demanda de almacenamiento estimada no superaría

los 6.5MB. Siendo 2.34 MB de la aplicación con 0.39MB de instrucciones almacenadas (100 tareas de 4kB cada una) y 3.72MB de modelos capturados (400 ejecuciones de 9.5kB cada una).

Una vez realizadas las mediciones, observamos que el recurso que más se ve reducido es el uso del procesador. No habiendo observado grandes diferencias significativas en cuanto a la demanda de memoria física. De cara al almacenamiento, no supone un incremento sustancial debido a las capacidades de almacenamiento de los terminales.

5.1.2.1.2. SESGO EN LA APLICACIÓN

Examinando cómo afecta la captura de los modelos al rendimiento del dispositivo, es importante distinguir si las llamadas a las funciones de la librería de integración añaden más lentitud en la ejecución de la aplicación. Extrayendo las mediciones de tiempo se han recuperado por cada dispositivo 10800 cálculos de tiempos de llamada a funciones de la librería distribuidos en 6600 llamadas a *logInteraction()*, 2100 a *resumeTask()* y otros 2100 a *pauseTask()*.

Exponemos mediante la tabla 5.6 los intervalos de confianza del 95% de los milisegundos estimados y clasificados por función de librería.

Dispositivo	*resumeTask()*	*pauseTask()*	*logInteraction()*	*total*
HTC Desire HD	*[4.95, 5.29]*	*[5.04, 5.87]*	*[4.48, 5]*	*[4.77, 5.13]*
S. Galaxy Nexus	*[1.57, 1.72]*	*[1.55, 1.67]*	*[1.5, 1.58]*	*[1.54, 1.6]*
S. Nexus 10	*[1.95, 2.02]*	*[1.94, 2.01]*	*[2.04, 2.09]*	*[2.02, 2.05]*
LG Nexus 5	*[0.99, 1.05]*	*[1.01, 1.07]*	*[1.04, 1.08]*	*[1.04, 1.06]*

Tabla 5.6 Intervalos de confianza del 95% de los milisegundos estimados para llamar a las funciones

Estos datos muestran que *no hay un aumento apreciable en el tiempo de ejecución del proceso de la aplicación evaluada debido a un aumento significativo de menos de 6 milisegundos por llamada*. Podemos afirmar dicha conclusión al observar que no supone un aumento significativo. Con una penalización estimada de 5.87 milisegundos como máximo por llamada en el dispositivo con menos recursos, cuando las llamadas a la librería se realizan con muy baja frecuencia en comparación con el tiempo de llamada de las mismas.

5.1.2.1.3. SESGO EN EL TIEMPO DE TAREA

El siguiente punto es medir si existe un aumento significativo de tiempo en la ejecución de las tareas debido a la disminución de recursos. Debemos tener presente que el factor que más condiciona al tiempo es el tipo de ejecución a simular. También es condicionante el tipo de dispositivo por lo que las comparaciones se harán por cada dispositivo y por cada ejecución de tarea. Para ello, realizaremos un estudio de dependencia en el que estudiaremos el tiempo de ejecución de la tarea como variable dependiente.

Analizando los datos expuestos en la tabla 5.7, percibimos que en la mayoría de los casos hay un aumento estadísticamente significativo a excepción de algunos casos del dispositivo más moderno, donde curiosamente hay una disminución significativa en las ejecuciones *T1_2* y *T2_2*. Sin embargo, no percibimos evidencias mediante Pearson para afirmar un aumento de la diferencia de tiempos estimada en función de la duración de la ejecución, indicando contrariamente un valor de relación inversa no significativo: $r=-0.04$, $p=0.8$. Por ello, deducimos que no hay una relación de dependencia significativa para tareas con aplicaciones móviles.

Dispositivo	*Ejecución*	*Duración con herramienta (s)*	*Duración sin herramienta (s)*	*Intervalo de confianza [95%] de la diferencia estimada (s)*
HTC Desire HD	*T1_1*	*7.6*	*7.58*	*[0.01, 0.02]*
S. Galaxy Nexus	*T1_1*	*7.78*	*7.39*	*[0.37, 0.41]*
S. Nexus 10	*T1_1*	*7.32*	*7.02*	*[0.28, 0.33]*
LG Nexus 5	*T1_1*	*6.58*	*6.52*	*[0.05, 0.07]*
HTC Desire HD	*T1_2*	*18.93*	*18.25*	*[0.66, 0.69]*
S. Galaxy Nexus	*T1_2*	*17.77*	*17.52*	*[0.24, 0.27]*
S. Nexus 10	*T1_2*	*18.12*	*17.08*	*[0.31, 0.34]*
LG Nexus 5	*T1_2*	*17.07*	*17.1*	*[-0.03,-0.02]*
HTC Desire HD	*T2_1*	*17.44*	*17.39*	*[0.05, 0.07]*
S. Galaxy Nexus	*T2_1*	*17.93*	*17.82*	*[0.08, 0.14]*
S. Nexus 10	*T2_1*	*18.67*	*18.43*	*[0.18, 0.23]*
LG Nexus 5	*T2_1*	*17.2*	*17.21*	*[-0.03, 0.01]*
HTC Desire HD	*T2_2*	*20.43*	*20.35*	*[0.06, 0.08]*
S. Galaxy Nexus	*T2_2*	*21.08*	*20.99*	*[0.06, 0.12]*
S. Nexus 10	*T2_2*	*21.83*	*21.67*	*[0.13, 0.18]*
LG Nexus 5	*T2_2*	*20.17*	*20.21*	*[-0.05, -0.02]*

Tabla 5.7 Diferencias de los segundos estimados de duración de las ejecuciones con y sin herramienta

Por todo ello, concluimos que *hay evidencias de un aumento significativo en el tiempo de tarea no superior a 0.69 segundos (independientemente del tiempo de la duración)*, lo cual puede acarrear un aumento apreciable en tareas de muy corta duración.

5.1.2.2. ESTUDIO DE LA PERCEPCIÓN DEL USUARIO

Un usuario que va a ser monitorizado siempre debe ser notificado de tal actividad, por ello no podemos eliminar que el usuario sea consciente de que está siendo monitorizado. Sin embargo, además de este sesgo, puede darse el caso en el que se pierda la percepción de instantaneidad de la respuesta de la aplicación a la hora de actuar con la interfaz.

Anteriormente, se evaluó una versión preliminar [Pretel+14] de la herramienta de usuario de pruebas de un modo cualitativo. Ésta fue integrada con una aplicación móvil híbrida real con la que 12 usuarios realizaban tareas con dos versiones de la aplicación aparentemente igual. Una de ellas realizaba la captura de los modelos y otra no. Al finalizar el experimento los usuarios debían elegir qué versión de la aplicación realizaba la captura, el 100% de los sujetos del experimento añadieron que no notaron la diferencia. El 100% de los sujetos respondieron con un 30% de seguridad, donde el 100% de seguridad correspondía a estar completamente seguro.

Complementando la evaluación preliminar, debemos asegurar de un modo cuantitativo la instantaneidad de la interfaz, utilizando como criterio el concretado en [Miller68], [Card+91] y [Nielsen91], donde definen que *un usuario percibe una interacción instantánea cuando su tiempo de respuesta no excede de 0.1 segundos*.

Para ello, hemos medido el tiempo de penalización añadido al proceso de la aplicación evaluada que realiza las llamadas a la librería de integración, el cual nunca excedía los 6 milisegundos por llamada (ver apartado 5.1.2.1.3). Aunque esta medida dista considerablemente de los 100 milisegundos permitidos (0.1 segundos), debemos asegurarnos de que el descenso de los recursos de procesador no implica una disminución significativa

en la respuesta general de la interfaz. Considerando que en la generación y carga de la interfaz es la situación más sensible, durante las simulaciones calculamos el tiempo que tarda la aplicación evaluada desde que se le notifica que debe mostrar una interfaz hasta que la muestra.

Agrupando las 4200 mediciones de carga tomadas por dispositivo, percibimos en la figura 5.4 que el tiempo, en la mayoría de los casos es superior a 100 milisegundos. El sistema Android, para mantener la espontaneidad muestra durante la carga de la misma una animación dando la sensación de respuesta instantánea.

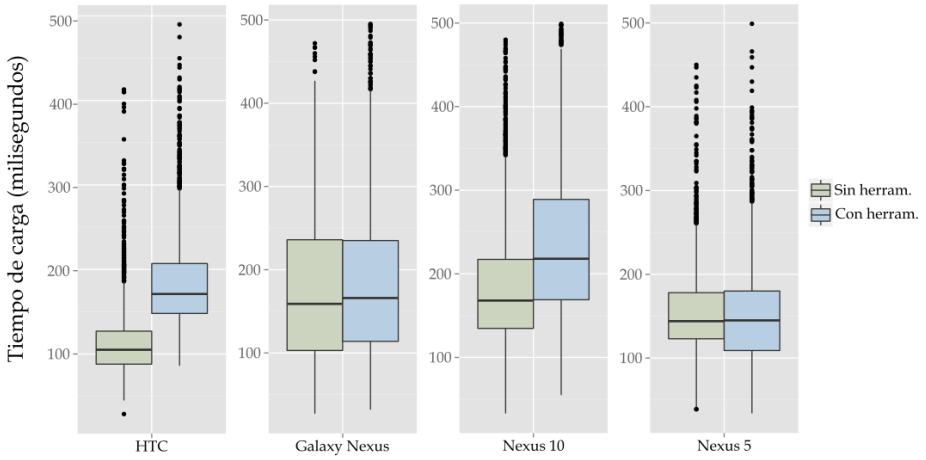

Figura 5.4 Comparación del tiempo de carga de interfaz con y sin herramienta

Sin embargo, hay un aumento significativo en el tiempo de carga de la interfaz cuando se realiza la captura de los modelos mediante la herramienta. Estudiando el aumento por dispositivo (ver tabla 5.8), la diferencia estimada entre los dos grupos muestra que en el dispositivo más moderno no se manifiesta un aumento significativo. Sin embargo, en los otros dispositivos utilizados sí.

Dispositivo	*Tiempo de carga con herramienta (ms)*	*Tiempo de carga sin herramienta (ms)*	*Intervalo de confianza [95%] de la diferencia estimada (ms)*
HTC Desire HD	*186.71*	*114.28*	*[69.29, 75.58]*
S. Galaxy Nexus	*199.05*	*174.45*	*[17.57, 31.64]*
S. Nexus 10	*246.47*	*190.04*	*[50.33, 62.52]*
LG Nexus 5	*152.34*	*173.78*	*[-57.28, 14.4]*

Tabla 5.8 Diferencias de los milisegundos estimados de carga de interfaz con y sin herramienta

Por el contrario, las estimaciones en función de los datos calculados indican que en ciertos dispositivos hay un aumento significativo de 75.58 milisegundos como máximo.

Por lo tanto, *podemos afirmar que la herramienta de usuario de pruebas en el momento de la captura de los modelos no es apreciable por el usuario al no aumentar el tiempo de respuesta de la interfaz en más de 100 milisegundos.*

5.1.3. CONSIDERACIONES DE LA ESTRATEGIA DE CAPTURA

Concluyendo con el estudio de los diferentes elementos que pueden verse afectados por la captura de los modelos, debemos tener en cuenta varias cuestiones.

Por un lado, no se ha detectado un incremento significativo en la memoria física, por lo que consideramos que no requiere especial atención al igual que el almacenamiento. La demanda de espacio por la herramienta de usuario de pruebas no es considerada de importancia al compararla con las capacidades reales de los terminales móviles. Sin embargo, dependiendo del tipo de dispositivo que el usuario de pruebas disponga, se percibirá un incremento significativo en la demanda de procesador, siendo incrementado desde un 9.7% en el terminal más moderno hasta un 36% percibido en el más antiguo. Dicho incremento puede penalizar el tiempo de ejecución de la aplicación evaluada junto con las llamadas a funciones de la librería. Afortunadamente, el tiempo de penalización por las llamadas (estimado en 6 milisegundos por llamada) tampoco se considera apreciable salvo en tareas de muy corta duración (menos de 10 segundos), donde puede añadirse hasta medio segundo al tiempo real de la tarea.

Por otro lado, aunque el rendimiento se vea afectado, en términos generales la percepción del usuario de pruebas no se ve alterada ya que no hay un aumento significativo en el tiempo de respuesta de la interfaz mayor a 0.1 segundos.

Gracias a este estudio, *la afirmación presentada al comienzo de esta sección queda confirmada.*

5.2. VALIDACIÓN DEL USO DE LA BASE DE CONOCIMIENTO

Una vez estudiado el sesgo que nos permite asegurar la veracidad de los datos extraídos mediante la estrategia de captura implementada en este trabajo, se procede a validar la base de conocimiento presentada en la sección 3.2.

Para finalmente poder validar la metodología, debemos asegurarnos de que la base de conocimiento cumple con el objetivo de calcular casos favorables y describir los valores de las variables de contexto en los que realmente exista una mayor probabilidad de encontrar errores. En otras palabras, lo que validaremos en este apartado es lo siguiente:

> *Es posible detectar los casos favorables y valores de las variables de contexto más adecuados para la evaluación de una nueva aplicación mediante el uso de la base de conocimiento generada por aplicaciones evaluadas anteriormente de la misma categoría.*

Abordando esta idea proponemos un estudio comparativo basado en un experimento donde evaluaremos una aplicación móvil mediante la metodología expuesta, cuantificaremos el número de errores de interacción encontrados en todos los casos posibles resultantes y compararemos el número de errores detectados en los casos hipotéticamente elegidos en función de los datos obtenidos mediante la base de conocimiento. Como mostramos en la figura 5.5, el experimento se divide en dos fases:

- En una primera fase, generaremos una base de conocimiento mediante la evaluación de dos aplicaciones móviles desarrolladas desde cero.
- En una segunda fase, evaluaremos una tercera aplicación con la cual estudiaremos los resultados obtenidos y los contrastaremos con las elecciones basadas en la base de conocimiento.

Figura 5.5 Fases del experimento de validación del uso de la base de conocimiento

5.2.1. DESCRIPCIÓN DEL EXPERIMENTO

En primer lugar, para la generación de la base de conocimiento se han estudiado las aplicaciones de la categoría compras del mercado de aplicaciones Google Play[36] y se han implementado dos aplicaciones móviles: la aplicación ya definida y utilizada en las pruebas de validación de la estrategia de captura llamada *Maicbay* y una nueva implementación denominada *Maicmazon*. Estas aplicaciones están basadas en las interfaces y arquitecturas de la información de varias aplicaciones ya existentes, concretamente de las aplicaciones *Amazon compras*[37], *eBay*[38] y *segundamano.es*[39]. Estas aplicaciones fueron evaluadas por un grupo de 24 usuarios anónimos con las propiedades que se muestran en la tabla 5.9.

Característica	*Descripción*
Edad	*29.94 años de media y 6.81 años de desviación típica*
Género	*15 hombres y 9 mujeres*
Procedencia	*20 del País Vasco, 3 de Navarra y 1 de Madrid*
Lateralidad	*19 diestros y 5 zurdos*

Tabla 5.9 Descripción del grupo de usuarios de pruebas que generó la base de conocimiento del experimento

Los dispositivos detallados en la tabla 5.10 y utilizados en el experimento fueron los propios de los usuarios anónimos.

[36]https://play.google.com
[37]https://play.google.com/store/apps/details?id=com.amazon.mShop.android.shopping
[38]https://play.google.com/store/apps/details?id=com.ebay.mobile
[39]https://play.google.com/store/apps/details?id=com.anuntis.segundamano

Fabricante	*Modelo*	*Versión del S.O.*
HTC	*HTC Desire 816*	*4.4.2*
HTC	*HTC Desire HD A9191*	*2.3.5*
LEOTEC	*LESPH5001B*	*4.2.1*
LGE	*LG-D802*	*4.4.2*
LGE	*Nexus 4*	*4.4.4*
LGE	*Nexus 5*	*4.4.4*
Samsung	*Galaxy Nexus*	*4.3*
Samsung	*GT-I9300*	*4.3*
TCT	*ALCATEL ONE TOUCH 7041X*	*4.2.2*
THL	*thl T11*	*4.2.2*

Tabla 5.10 Descripción de los dispositivos utilizados en la generación de la base de conocimiento del experimento

Este grupo de usuarios de pruebas desempeñaron un conjunto de tareas utilizando las aplicaciones en sus dispositivos móviles. El grupo de tareas estaba formado por 5 tareas típicas detectadas dentro de las aplicaciones de categoría compras. En primer lugar, estas tareas (ver tabla 5.11) fueron realizadas utilizando la aplicación *Maicbay*. Una vez realizadas todas las tareas con la aplicación, los usuarios hicieron las mismas tareas con *Maicmazon* y con sus instrucciones correspondientes. Por lo tanto, las tareas fueron realizadas por cada usuario de pruebas en 4 entornos, generando por cada aplicación y usuario un mínimo de 20 tareas realizadas.

Nombre	*Instrucciones*
T1. Lista categoría	*Mediante el menú principal se deben listar los anuncios que corresponden a la categoría de 'Ocio y Deportes'.*
T2. Registro	*Debe registrarse en la aplicación con los siguientes datos:* *(Todos los datos en minúsculas)* *- Nombre: pedro* *- Contraseña: pedro84* *- Correo: pedro@pedro.com*
T3. Poner anuncio	*El usuario debe poner un anuncio con los siguientes datos:* *- Título: MANO* *- Categoría: ocio y deportes* *- Subcategoría: coleccionismo* *- Provincia: Vizcaya* *- Fotos: Las cuatro deben hacerse con la cámara.* *- Descripción: MANO EN VENTA* *- Precio: 40€*
T4. Eliminar favorito	*Se debe eliminar el anuncio favorito cuyo título es 'Comedero para perros electrónico'*
T5. Buscar y contactar	*Buscar el anuncio cuyo título contenga las palabras 'tortuga andador' en Guipúzcoa y contactar con el vendedor*

Tabla 5.11 Descripción de las tareas realizadas con la aplicación Maicbay en el experimento

Para evitar el efecto aprendizaje, el cual añade un mayor número de errores en las primeras tareas que realiza el usuario de pruebas, tanto el orden de las tareas como el de los entornos son asignados aleatoriamente para cada usuario.

El grupo elegido de 4 entornos para el desarrollo de las pruebas está formado por dos entornos interiores y dos entornos exteriores, sus detalles se especifican en la tabla 5.12.

Entorno	*Descripción*
Sentado en casa	*Estando en casa, lo que interesa es que el usuario esté sentado ya sea en una silla, sillón, sofá...*
Tumbado en casa	*Estando en casa, lo importante es que el usuario esté tumbado (sea boca arriba, de costado...). Puede estar tumbado en cualquier sitio: sofá, cama, suelo...*
Caminando por la calle	*Efectuando cualquier desplazamiento caminando fuera de un edificio.*
Viajando en un medio de transporte	*Haciendo algún desplazamiento en cualquier tipo de transporte, ya sea público o privado. Incluyendo tren, coche, autobús...*

Tabla 5.12 Descripción de los entornos del experimento

Una vez generada la base de conocimiento, se procede al estudio del uso de la misma. Para ello, nos pusimos en contacto con un desarrollador especializado en la plataforma Android y con experiencia en aplicaciones de producción. Este desarrollador, de una empresa que realiza soluciones centradas en dispositivos móviles, implementó una nueva aplicación denominada *Maicvalia*. Dicha aplicación, también de la categoría compras, fue probada mediante la metodología expuesta en este trabajo.

Un grupo de 12 usuarios (ver detalles en la tabla 5.13) realizaron en los entornos previamente definidos en la tabla 5.12 las 4 tareas de similar nombre a las anteriormente expuestas en la tabla 5.11 pero con diferentes instrucciones.

Característica	*Descripción*
Edad	*28.17 años de media y 3.86 años de desviación típica*
Género	*8 hombres y 4 mujeres*
Procedencia	*11 del País Vasco, 1 de Navarra*
Lateralidad	*10 diestros y 2 zurdos*

Tabla 5.13 Descripción del grupo de usuarios de pruebas de la evaluación de Maicvalia

Los dispositivos utilizados fueron también los propios de los usuarios de pruebas (ver tabla 5.14).

Fabricante	*Modelo*	*Versión del S.O.*
HTC	*HTC Desire HD A9191*	*2.3.5*
LEOTEC	*LESPH5001B*	*4.2.1*
LGE	*Nexus 4*	*4.4.4*
LGE	*Nexus 5*	*4.4.4*
Samsung	*GT-I9300*	*4.3*
THL	*thl T11*	*4.2.2*

Tabla 5.14 Descripción de los dispositivos utilizados en la evaluación de Maicvalia

5.2.2. RESULTADOS

Una vez realizadas las fases descritas del experimento, procedemos al estudio de los resultados obtenidos mediante dos perspectivas: *casos favorables* y *variables de contexto*.

5.2.2.1. CASOS FAVORABLES

En primer lugar, estudiamos la base de conocimiento generada mediante las aplicaciones *Maicbay* y *Maicmazon*. Haciendo uso del modelo de casos favorables (ver apartado 3.2.4), calculamos todos los casos posibles y las probabilidades de originar errores de interacción para posteriormente concluir los casos favorables. Todos los casos son agrupados por número de entornos. En la figura 5.6 apreciamos la evidencia lógica que dicta que a medida que aumentamos el número de entornos, también lo hace la probabilidad de encontrar más errores (cuantos más entornos, las estimaciones van más a la izquierda precisando menos usuarios).

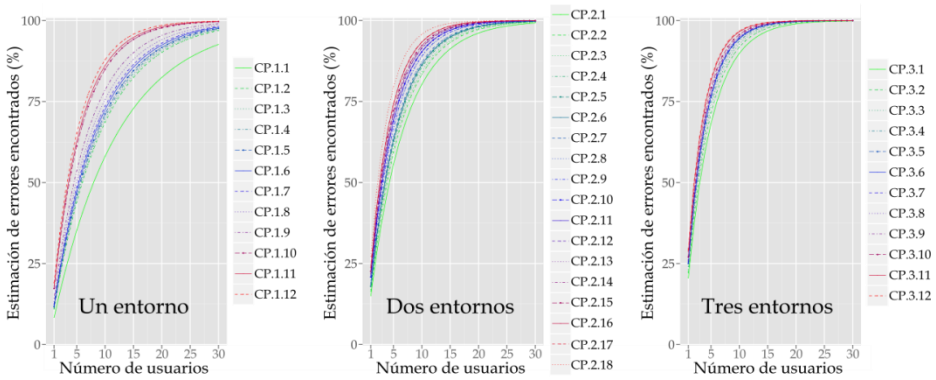

Figura 5.6 Estimaciones de errores de interacción encontrados mediante la base de conocimiento

Analizando en detalle las estimaciones de la tabla 5.15, disponemos de los entornos ya definidos (ver tabla 5.12) denominados como "*en la cama*" (B), "*sentado*" (S), "*caminando*" (W) y "*en medio de transporte*" (T) y de los tres tipos de grupos de usuario existentes: "*hombre*" (H), "*mujer*" (M) y "*ambos*" (H y M).

Apreciamos que el entorno referente a "*sentado*" es el que más induce a los usuarios de pruebas a producir errores. Esto es debido a que los usuarios de pruebas se encuentran más fácilmente en este tipo de entorno y por lo tanto, es el primero en el que realizan las tareas designadas. Consecuentemente, debido al efecto aprendizaje, cometen más errores en este tipo de entorno. Afortunadamente y aunque sea debido a este efecto, también se producirá en evaluaciones posteriores por lo que también se obtendrá mayor número de errores de interacción en este tipo de entorno.

Concretamente, si nos centramos en el estudio del caso posible más adecuado para realizar la evaluación dentro de los entornos mostrados en la tabla 5.15 en los que se usa sólo un entorno, observamos que el mejor caso posible, es el CP.1.12. Explícitamente, dicho caso posible dispone de un valor p_{comp} utilizado como criterio de clasificación de 0.190. Por lo tanto, es considerado como caso favorable.

Caso posible	*Entornos*	*Tipos de usuario*	p_{comp}
CP.1.1	*B*	*M*	*0.083*
CP.1.2	*W*	*M*	*0.110*
CP.1.3	*T*	*M*	*0.112*
CP.1.4	*W*	*H y M*	*0.114*
CP.1.5	*W*	*H*	*0.117*
CP.1.6	*B*	*H y M*	*0.121*
CP.1.7	*T*	*H y M*	*0.125*
CP.1.8	*T*	*H*	*0.132*
CP.1.9	*B*	*H*	*0.143*
CP.1.10	*S*	*H*	*0.173*
CP.1.11	*S*	*H y M*	*0.179*
CP.1.12	*S*	*M*	*0.190*

Tabla 5.15 Estimaciones obtenidas de la base de conocimiento para casos de un entorno

Una vez identificado el caso favorable para los casos posibles de un entorno, estudiamos los casos posibles de dos entornos mediante los datos mostrados en la tabla 5.16. En estos resultados

el p_{comp} más alto lo adquiere el caso el CP.2.18, con un valor de 0.271. Consecuentemente, marcamos dicho caso posible como el caso favorable de los casos de dos entornos.

Caso posible	*Entornos*	*Tipos de usuario*	p_{comp}
CP.2.1	*B y W*	*M*	*0.150*
CP.2.2	*B y T*	*M*	*0.164*
CP.2.3	*T y W*	*M*	*0.169*
CP.2.4	*T y W*	*H y M*	*0.176*
CP.2.5	*T y W*	*H*	*0.180*
CP.2.6	*B y W*	*H y M*	*0.181*
CP.2.7	*B y T*	*H y M*	*0.185*
CP.2.8	*B y T*	*H*	*0.197*
CP.2.9	*B y W*	*H*	*0.199*
CP.2.10	*S y W*	*H*	*0.208*
CP.2.11	*B y S*	*M*	*0.210*
CP.2.12	*S y T*	*H*	*0.218*
CP.2.13	*S y W*	*H y M*	*0.219*
CP.2.14	*B y S*	*H y M*	*0.220*
CP.2.15	*B y S*	*H*	*0.226*
CP.2.16	*S y T*	*H y M*	*0.237*
CP.2.17	*S y W*	*M*	*0.238*
CP.2.18	*S y T*	*M*	*0.271*

Tabla 5.16 Estimaciones obtenidas de la base de conocimiento para casos de dos entornos

Finalmente, en el caso de tres entornos calculamos mediante la base de conocimiento un valor p_{comp} máximo de 0.29. Como apreciamos en la tabla 5.17, este valor corresponde al caso posible CP.3.12, considerado por lo tanto el caso favorable en el grupo de casos de tres entornos.

Caso posible	*Entornos*	*Tipos de usuario*	p_{comp}
CP.3.1	*B, T y W*	*M*	*0.205*
CP.3.2	*B, T y W*	*H y M*	*0.222*
CP.3.3	*B, T y W*	*H*	*0.231*
CP.3.4	*S, T y W*	*H*	*0.241*
CP.3.5	*B, S y W*	*M*	*0.250*
CP.3.6	*B, S y W*	*H y M*	*0.253*
CP.3.7	*B, S y W*	*H*	*0.254*
CP.3.8	*S, T y W*	*H y M*	*0.259*
CP.3.9	*B, S y T*	*H*	*0.261*
CP.3.10	*B, S y T*	*H y M*	*0.270*
CP.3.11	*B, S y T*	*M*	*0.286*
CP.3.12	*S, T y W*	*M*	*0.290*

Tabla 5.17 Estimaciones obtenidas de la base de conocimiento para casos de tres entornos

Como resultado de este análisis, debemos validar que en la aplicación *Maicvalia* se originan más errores de interacción reales

en los casos favorables elegidos (CP.1.12, CP.2.18 y CP.3.12) que en el resto de casos.

Para validar estas elecciones, extraemos los errores de interacción encontrados por los usuarios de pruebas de la aplicación. Con estos datos extraídos del análisis automático realizado por la herramienta de soporte de la metodología, efectuamos una agrupación de los 89 errores de interacción registrados utilizando la severidad calculada de los errores de interacción encontrados como criterio.

Severidad	*Errores de interacción encontrados*
Baja	*32*
Media	*54*
Alta	*3*
Total	*89*

Tabla 5.18 Número de errores de interacción encontrados en la evaluación de Maicvalia agrupados por severidad

Habiendo clasificado los diferentes errores de interacción, nos centramos en el estudio de las diferentes permutaciones que pueden darse dentro de los cuatro entornos y los usuarios de pruebas que han realizado las tareas propuestas para la evaluación de *Maicvalia*.

Para ello, se realizan todas las combinaciones posibles de usuarios y entornos. Después se calcula por cada combinación, el número de errores de interacción de cada tipo de severidad detectados. Es decir, si dispusiéramos por ejemplo de dos usuarios (U1 y U2) y dos entornos (E1 y E2), generaríamos una muestra de 9 combinaciones (ver tabla 5.19).

Usuarios	*Entornos*
U1	*E1*
U1	*E2*
U1	*E1 y E2*
U2	*E1*
U2	*E2*
U2	*E1 y E2*
U1 y U2	*E1*
U1 y U2	*E2*
U1 y U2	*E1 y E2*

Tabla 5.19 Ejemplo de combinaciones para la validación de la base de conocimiento

En el caso de los resultados disponemos de doce usuarios y cuatro entornos, por lo que generamos una muestra de 61425 casos. Posteriormente los dividimos en seis grupos. Por un lado, agrupamos todos los casos de un solo entorno que sean el caso favorable CP.1.12 en un grupo denominado CF1. Por otro lado, seleccionamos los casos restantes de un solo entorno en el grupo RC1. Realizamos la misma operación para crear los grupos CF2 (para los casos CP.2.18) y RC2 referentes a los grupos de dos entornos e igualmente con los grupos de tres entornos para CF3 (para los casos CP.3.12) y RC3.

Una vez agrupados los diferentes casos, estimamos el número de errores de interacción medio encontrados en función del número de usuarios utilizado y el nivel de severidad de los mismos. Como el experimento lo han realizado cuatro usuarios de pruebas femeninos, se estudiarán los casos hasta cuatro usuarios ya que no se pueden realizar comparaciones con casos con entornos de más usuarios.

> *Con estos grupos nos disponemos a analizar si los casos que cumplen el perfil de caso favorable y los que no lo cumplen difieren de tal modo que la media estimada de los casos elegidos es superior a la del resto.*

Plasmando los datos mediante la figura 5.7, notificamos que las medias estimadas de los grupos elegidos como casos favorables sí adquieren un valor más elevado que el resto de casos que poseen el mismo número de entornos.

Consecuentemente, podemos afirmar que los casos favorables elegidos logran mejores resultados que el resto de casos que poseen su mismo número de entornos. En lo referente a la severidad de los errores de interacción, los casos favorables calculados logran mejores resultados con los errores de severidad alta que con los de media y baja.

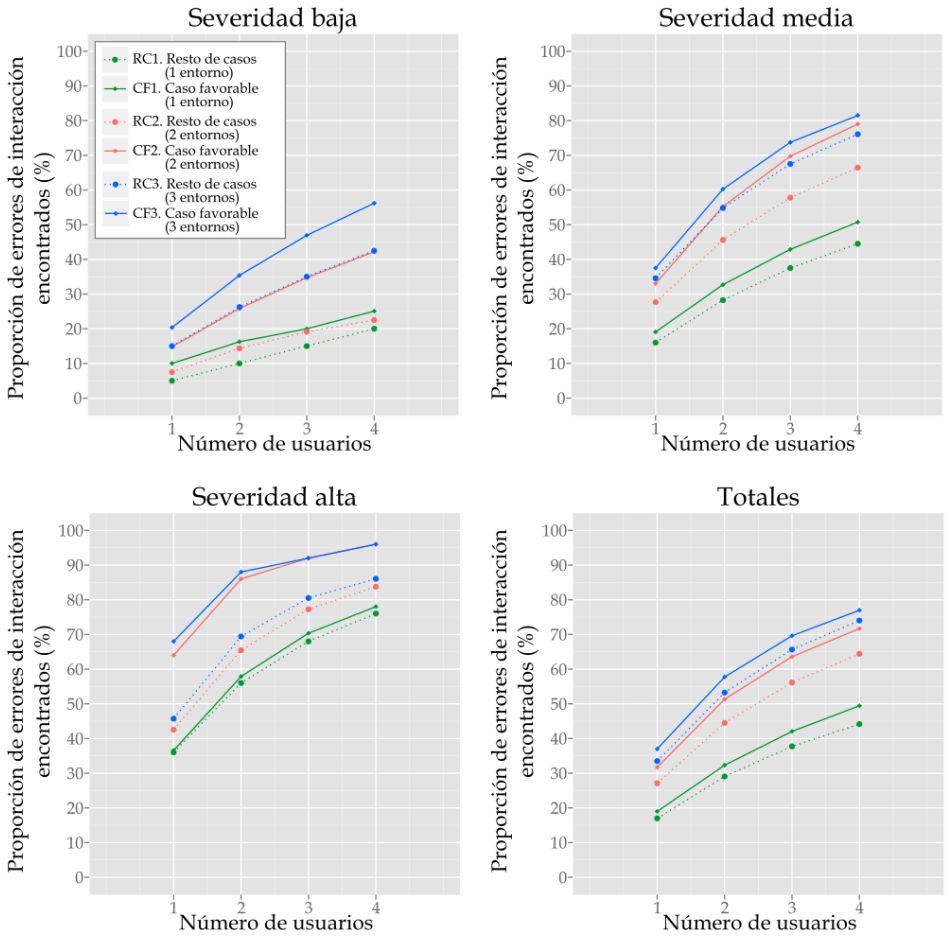

Figura 5.7 Comparación de la proporción del número de errores de interacción detectados en los grupos de casos favorables elegidos con el resto de casos posibles

Centrándonos en la selección del mejor caso favorable, si tuviéramos tres tipos de entorno a nuestra disposición, el caso favorable que mejor resultado ofrece es el CP.3.12.

Como vemos resumido en la tabla 5.20, el mejor caso posible de dos entornos CP.2.18 tiene un valor p_{comp} de 0.271. La diferencia entre este valor y los valores de los dos casos que ofrecen mejores condiciones no es excesiva: 0.015 y 0.019. Además, dicho valor es incluso mejor que el de la mayoría de los casos con tres entornos (ver valores p_{comp} de la tabla 5.17). Como apreciamos en la figura 5.7, si decidimos utilizar un entorno menos mediante la elección de CP.2.18 (CF2), podremos obtener resultados similares al uso de los tres entornos. Como las medias estimadas son muy similares

podríamos ahorrarnos un entorno en la evaluación de la aplicación.

Caso posible	*Entornos*	*Tipos de usuario*	p_{comp}
CP.2.17	*S y W*	*M*	*0.238*
CP.3.10	*B, S y T*	*H y M*	*0.270*
CP.2.18	*S y T*	*M*	*0.271*
CP.3.11	*B, S y T*	*M*	*0.286*
CP.3.12	*S, T y W*	*M*	*0.290*

Tabla 5.20 Mejores casos posibles de la estimación de la base de conocimiento asignados como casos favorables

Una vez validado el cálculo de casos favorables podemos afirmar que obtenemos como casos favorables, casos que obtienen mejores resultados que la mayoría de casos posibles.

5.2.2.2. VARIABLES DE CONTEXTO

A continuación, procedemos a un análisis de la descripción de las variables de contexto visto en el apartado 3.2.4.2. Para ello se propone el análisis de varias variables de contexto tanto de tipo cuantitativo y como cualitativo.

En el caso de las *variables cuantitativas*, proponemos el estudio de las variables de contexto más relevantes en este experimento: "*Iluminancia*", "*Ruido*", "*Aceleración en eje x*", "*Aceleración en eje y*" y "A*celeración en el eje z*".

Para ello, con la ayuda de la base de conocimiento generada, definiremos rangos de valores en los que debemos intentar que las variables fluctúen. Una vez definidos, se contabilizarán todos los errores de interacción detectados dentro de los mismos y se calculará la proporción del total de errores detectados sin restricciones de rangos.

> *Si la proporción de los errores de interacción dentro de los rangos definidos es elevada significa que podemos estimar los valores más adecuados de estas variables para detectar errores de interacción.*

Para la acotación de los límites se proponen tres modos. A modo de ejemplo, se describen en la tabla 5.21 los valores de la variable

"*Iluminancia*" donde hay más posibilidad de ocasionar errores de interacción.

- El límite menos restrictivo es acotar en base al valor máximo y mínimo de la variable, en el caso de la "*Iluminancia*" los valores 0 y 10240 (equivalente a un día nublado o lluvioso).
- En un término medio (debido a su elevada desviación típica) es acotar en base a la media obtenida sumando y restando su desviación típica. En el caso de ejemplo, de 0 a 1749.28 (equivalente a una sala de lectura).
- El modo más restrictivo para este caso es mediante la definición de los cuartiles Q1 y Q3, donde abarcamos el 50% de los valores recuperados, en este caso de 10 (equivalente a noche despejada) a 99.25 (equivalente a la iluminación tenue de una habitación).

Estadístico	*Iluminancia (lux)*
Mínimo	*0*
Cuartil Q1	*10*
Mediana o Cuartil Q2	*20*
Media	*312.34*
Cuartil Q3	*99.25*
Máximo	*10240*
Desviación típica	*1436.94*

Tabla 5.21 Estimaciones de la variable iluminancia obtenidas de la base de conocimiento generada

Una vez definidos los tres modos de acotación, calculamos la proporción de errores de interacción que se detectan dentro de los límites especificados respecto a los totales detectados. Como muestra la figura 5.8, dependiendo del modo de acotación elegido se pierden errores de interacción mostrando un descenso en el porcentaje similar en las cinco variables estudiadas.

Centrándonos en la severidad, se aprecian peores resultados en los errores de severidad alta, independientemente del modo de acotación escogido. En cuanto al modo de acotación, los modos menos restrictivos muestran resultados similares mientras que hay un descenso de forma significativa en el modo más restrictivo, aunque nunca por debajo del 45%.

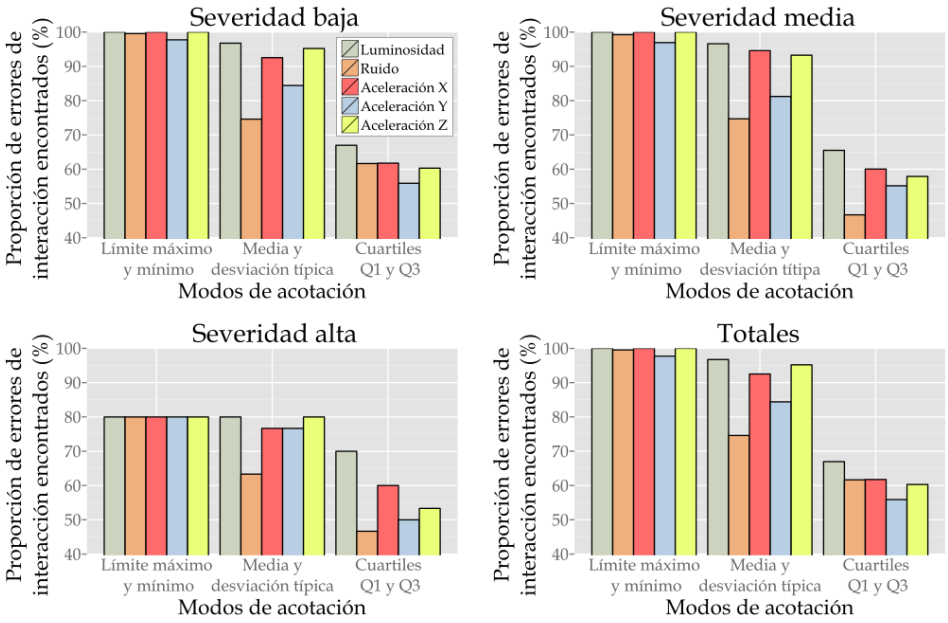

Figura 5.8 Proporción del número de errores de interacción detectados dentro de los límites acotados

Estudiando las *variables de contexto cualitativas*, el procedimiento es más sencillo al no tener que acotar ningún valor y tener simplemente que elegir entre los diferentes estados finitos que pueden adquirir. En el caso de este experimento, aunque no hemos considerado ninguna variable relevante, se ha optado por analizar las variables "*Auriculares conectados*" y "*Estado de música*".

En este caso, las frecuencias relativas expuestas en la tabla 5.22 muestran que si tuviéramos la opción de elegir los estados de las dos variables, elegiríamos "*no*" para "*Auriculares conectados*" y "*apagada*" para "*Estado de música*".

Variable	*Estado de variable*	*Frecuencia relativa simple (%)*
Auriculares conectados	*sí*	*8.15*
Auriculares conectados	*no*	*91.85*
Estado de música	*encendida*	*5.8*
Estado de música	*apagada*	*94.2*

Tabla 5.22 Tabla de frecuencias obtenidas de la base de conocimiento generada para las variables cualitativas

En estos dos casos, si estudiamos la proporción de errores de interacción detectados en la evaluación de la aplicación *Maicvalia*, notificamos en la tabla 5.23 valores superiores al 80% para los

errores de interacción de menos severidad y un 80% de los errores de severidad alta detectados.

Variable (estado)	*Severidad baja (%)*	*Severidad media (%)*	*Severidad alta (%)*	*Total (%)*
Auriculares conectados (no)	*91.73*	*82.89*	*80*	*85.55*
Estado de música (apagada)	*95*	*86.35*	*80*	*88.54*

Tabla 5.23 Proporción del número de errores de interacción detectados dentro de los estados elegidos

Con estos datos podemos afirmar que se han designado los valores más adecuados.

5.2.3. CONSIDERACIONES DEL USO DE LA BASE DE CONOCIMIENTO

Mediante este estudio hemos validado tanto el cálculo de los propios casos favorables del apartado 3.2.4.1 como la descripción de las variables de contexto del apartado 3.2.4.2.

Por un lado, se ha validado el uso de p_{comp} de la base de conocimiento como criterio de clasificación de casos posibles para calcular los más adecuados ya que los casos con p_{comp} mayor muestran mejores resultados que la mayoría. Por ello afirmamos que *es posible detectar los casos favorables más adecuados para la evaluación de una nueva aplicación utilizando la base de conocimiento.*

Por otro lado, hemos validado que tanto en variables de contexto cuantitativas como cualitativas podemos aproximar los valores más adecuados de las mismas para encontrar errores de interacción. Aunque el número de errores de interacción ocasionados depende del nivel de restricción del evaluador a la hora de acotar los rangos en las variables de contexto cuantitativas o elegir los estados de las variables de contexto cualitativas, podemos afirmar que *es posible detectar los valores de las variables de contexto más adecuados para la evaluación de una nueva aplicación utilizando la base de conocimiento.*

Gracias a estas afirmaciones y este estudio, *la afirmación planteada en la validación del uso de la base de conocimiento queda confirmada.*

5.3. VALIDACIÓN DE LA METODOLOGÍA

Una vez validada tanto la estrategia de captura de los datos para la generación de la base de conocimiento como el uso de la misma, se procede a validar la principal hipótesis presentada en este trabajo. Recordando la hipótesis definida en la sección 1.2, planteábamos lo siguiente:

> *Es posible reducir los recursos necesarios en la evaluación de la usabilidad de aplicaciones móviles sin comprometer la calidad de los resultados mediante una nueva metodología de evaluación centrada en una base de conocimiento.*

Afrontando esta idea, realizaremos un estudio comparativo que contrastará desde el punto de vista de los recursos necesarios, las diferencias existentes entre la evaluación con la metodología descrita en esta tesis y otra evaluación remota de aplicaciones móviles. Igualmente, se debe estudiar que el número de problemas detectados no se reduce utilizando la metodología definida en este trabajo. Consecuentemente, se comparará el número de dichos problemas detectados en ambas evaluaciones.

En primer lugar, debemos identificar los recursos que vamos a medir dentro de este enfoque. Como hemos especificado en la definición de la hipótesis, distinguimos tres principales recursos: el *coste general de la evaluación*, el *número de usuarios de pruebas y entornos necesarios* y el *tiempo de evaluación*.

- Dentro del *coste general de la evaluación* existe el coste fijo de la evaluación (coste de material, mantenimiento, etc.) y el coste asociado a la compensación de los usuarios de pruebas. Asumimos que el coste fijo relacionado con el material no es elevado al no requerir costosas instalaciones como en un Living Lab. Sin embargo, el coste asociado a los usuarios sí que lo es, en el informe de Sova y Nielsen [Sova+08] presentan varios modos de compensación. Aunque existen incentivos no monetarios (p.ej., vales de compra, comida y refrescos, etc.), nos centramos en una

compensación económica. Este informe engloba 165 evaluaciones donde han compensado económicamente a los usuarios de pruebas. La estimación media basada en estos datos es de 64$ cada hora de prueba por usuario. Viendo esta información consideramos que este coste está relacionado directamente con el número de usuarios de pruebas requeridos y el tiempo que debe dedicar cada uno de ellos. Por ello, asociamos la reducción de este recurso a la reducción del *número de usuarios de pruebas y entornos necesarios*. Añadimos además de los usuarios los entornos ya que a más entornos, más tiempo deben dedicar los usuarios al deber reproducir las tareas en cada uno de ellos.

- Centrándonos en el *número de usuarios de pruebas y entornos necesarios*, mediante la validación del uso de la base de conocimiento realizada anteriormente (ver sección 5.2), reducimos su número sin comprometer los problemas de usabilidad detectados al elegir los casos en los que existe mayor probabilidad de generar errores de interacción. Por lo tanto, la reducción de estos recursos y del coste de la evaluación (al estar relacionada directamente), dependerá de la elección del caso favorable a realizar, pudiendo elegir casos de una probabilidad de encontrar errores de interacción similar pero con un número de entornos menor.
- Finalmente, consideramos el *tiempo de evaluación* como el recurso más crítico, al cual dedicaremos especial atención. Para ello descompondremos las diferentes fases de la metodología y estudiaremos el tiempo requerido en cada una de ellas, lo mediremos y estudiaremos su comportamiento en función de las características de la evaluación.

Recapitulando lo expuesto anteriormente, se realizará un *experimento comparativo* donde mediremos, tanto la diferencia del *tiempo necesario* en cada una de las fases de las dos evaluaciones, como los *problemas de usabilidad encontrados*. Además, presentaremos adicionalmente una evaluación basada en modelos de aceptación tecnológica para comprender la predisposición de

los desarrolladores de aplicaciones móviles a utilizar la metodología y su plataforma de soporte en un entorno laboral. Dicha evaluación consiste en la cumplimentación de un cuestionario por parte de los desarrolladores.

5.3.1. DESCRIPCIÓN DEL EXPERIMENTO

En este apartado, antes de realizar la definición del experimento, hemos justificado la elección de un método de evaluación similar a la metodología definida para realizar una correcta comparación. Se ha realizado una asociación de los pasos del método a comparar con los de la metodología definida. Después se ha estudiado el tiempo de evaluación de cada uno de los métodos y se ha descompuesto para posteriormente expresar la diferencia de tiempos a medir y seguidamente estimar las diferencias.

5.3.1.1. ELECCIÓN DEL MÉTODO MUSIC

Para realizar esta comparativa, realizamos una adaptación del método MUSiC [Macleod+97]. Consideramos idónea la elección de este método por varios motivos:

- *Muestra varias similitudes con la metodología definida*. Al igual que ésta, el método MUSiC afronta tanto el enfoque sumativo como el formativo. Además, siendo un método basado en usuarios también permite la grabación de la interacción en entornos reales y sus posteriores revisiones.
- *Dispone de una herramienta de diseño modular cuya funcionalidad es fácil de implementar*. Esta herramienta, conocida como DRUM [Macleod+93] dispone de grabación de la interacción en vídeo y el registro de los eventos de interacción para una posterior reproducción y análisis.
- *Existe una clara analogía entre los pasos de MUSiC y la metodología desarrollada*. Como muestra la figura 5.9, MUSiC dispone de ocho principales pasos que encajan con los especificados en la metodología definida, lo que propicia una fácil comparación de los tiempos de cada fase.

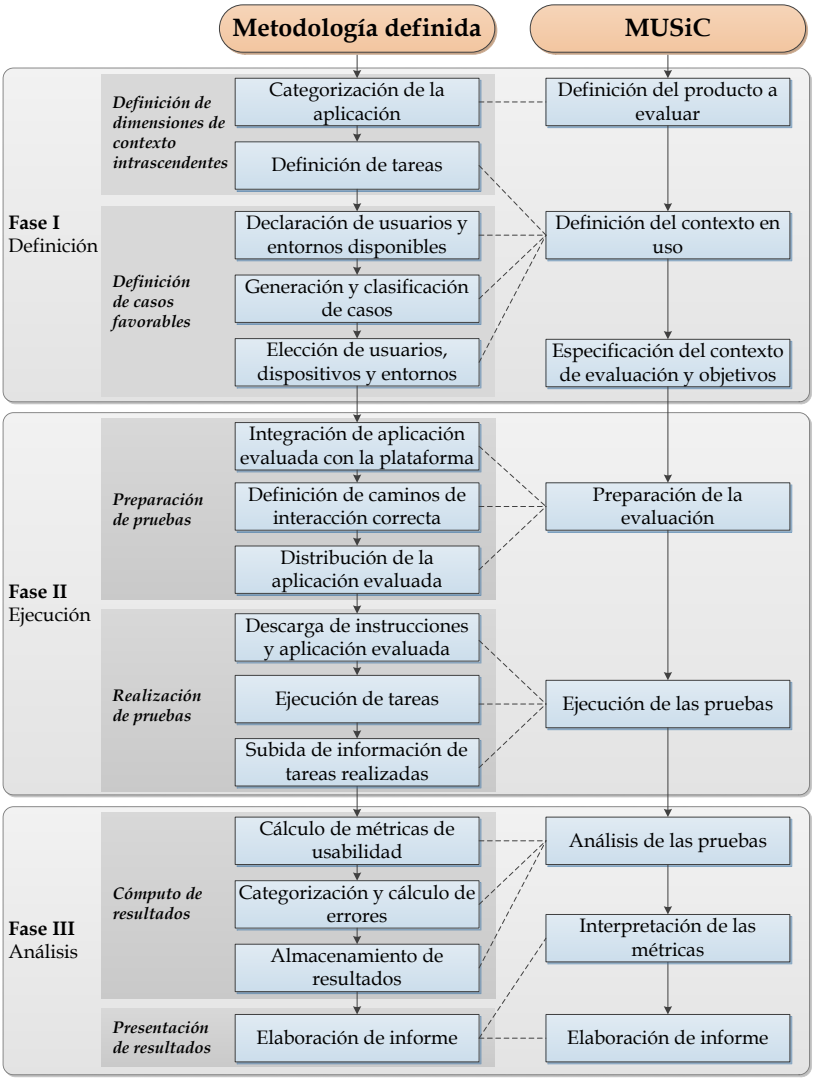

Figura 5.9 Analogía entre los pasos de la metodología definida y MUSiC

Procedemos a continuación a analizar explícitamente cada paso dentro de las fases y explicar la analogía existente entre ellos para posteriormente extraer los diferentes tiempos a cuantificar.

- En primer lugar, dentro de la metodología MUSiC los tres primeros pasos (definición del producto, definición del contexto en uso y especificación del contexto de evaluación y objetivos) se realizan mediante una guía [Thomas+96]. Estos pasos corresponden a la fase de definición de la metodología definida en este trabajo y tienen como objetivo definir la aplicación, usuarios, tareas y entornos en los que

se va a realizar la evaluación. Dentro del paso de especificación del contexto de evaluación y los objetivos, se definen principalmente las métricas que van a ser cuantificadas. Este paso queda excluido del análisis al no poder realizar una comparación directa, ya que no se realiza dentro de la nueva metodología. Desafortunadamente, dentro de MUSiC no se realiza la elección de los usuarios, dispositivos y entornos mediante el estudio de casos posibles y elección de casos favorables. Por ello será el principal paso donde se perciba una penalización en el tiempo requerido por la metodología definida.

- Dentro de la segunda fase, los pasos de preparación de la evaluación y ejecución de pruebas corresponden a los pasos de preparación de pruebas y realización de pruebas respectivamente. Dentro de la preparación de la evaluación asumimos que la metodología MUSiC no requiere de una cantidad de tiempo significativamente superior a la distribución de la aplicación. Por ello y debido a que no son realizados en MUSiC, el paso de integración con la plataforma y el paso de definición de caminos de interacción correcta, son los que penalizarán el tiempo de la metodología definida.
- En la última fase es donde mayor diferenciación se espera obtener. Debido al modo de la captura de la interacción mediante DRUM en MUSiC, al disponer del procesador de métricas de DRUM asumimos que no se obtendrán diferencias en el análisis sumativo. Sin embargo, se debe realizar un análisis de los vídeos de las interacciones capturadas para obtener los errores de interacción y poder realizar un análisis formativo. Por ello, *se medirá el tiempo de evaluación desde el enfoque del análisis formativo*.

5.3.1.2. ESTUDIO DEL TIEMPO DE EVALUACIÓN

Para lograr estudiar el *tiempo de evaluación* (T) debemos dividirlo en el tiempo que requiere cada una de las fases que componen las

dos evaluaciones. Retomando la descripción general de la metodología descrita en este trabajo (ver sección 3.1) y viendo la correspondencia de los pasos de ambas, podemos descomponer el *tiempo de evaluación* (T) de ambas metodologías en la suma del *tiempo de definición* (TD), *tiempo de ejecución* (TE) y *tiempo de análisis* (TA). Gracias a esto podremos realizar la comparativa de tiempos.

$$T = TD + TE + TA \quad (\ \mathbf{5.1}\ )$$

Una vez estudiados los pasos de ambas metodologías y generados los tiempos de un modo genérico, los detallamos explícitamente.

- Centrándonos en el *tiempo de definición* (TD) de las metodologías, la composición del mismo en la metodología definida es el *tiempo de definición de tareas y aplicación* ($T_{1.1}$) y el *tiempo de definición de los casos favorables* ($T_{1.2}$).

  $$TD_{M.Def} = T_{1.1} + T_{1.2} \quad (\ \mathbf{5.2}\ )$$

  Para MUSiC, la composición de este tiempo está formada por el *tiempo de definición de tareas y aplicación* ($T_{1.1}$). Asumimos a favor de MUSiC que el tiempo de definición de los usuarios, entornos y dispositivos es muy pequeño al no tener que realizar el estudio de los casos favorables, por eso lo consideramos despreciable.

  $$TD_{MUSiC} = T_{1.1} \quad (\ \mathbf{5.3}\ )$$

- El *tiempo de ejecución* (TE) de la metodología definida consta de varios pasos que no son realizados en MUSiC: el *tiempo de integración de la aplicación evaluada con la plataforma* ($T_{2.1}$) y el *tiempo de definición de caminos de interacción correcta* ($T_{2.2}$). El único tiempo que es común es el *tiempo de ejecución de las pruebas por parte de los usuarios de prueba* ($T_{2.3}$).

  $$TE_{M.Def} = T_{2.1} + T_{2.2} + T_{2.3} \quad (\ \mathbf{5.4}\ )$$

  Posicionándonos en el peor de los casos, asumimos que el tiempo de preparación de la evaluación de MUSiC es muy

pequeño, por lo que lo descartamos. Además, aunque de cara a la captura y grabación de los vídeos de la interacción de los usuarios requiere disponer de cámara de vídeo o algún dispositivo de captura, asumimos que ésta se paraleliza al igual que con la herramienta de usuario de pruebas (ver sección 4.6) de la plataforma de soporte. Por consiguiente, el *tiempo de ejecución de las pruebas por parte de los usuarios de prueba* ($T_{2.3}$) será de la misma duración que en la metodología definida y definirá la duración total del tiempo dedicado a la ejecución de las pruebas en MUSiC. Por ello, indirectamente asumimos que $T_{2.3}$ de la metodología definida y el de MUSiC no muestran una diferencia significativa en su duración.

$$TE_{MUSiC} = T_{2.3} \quad (5.5)$$

- Finalmente, el *tiempo de análisis* (TA) de la metodología definida consta principalmente del tiempo del cálculo de los resultados y el tiempo de elaboración del informe. Ambos se realizan automáticamente por la aplicación de escritorio de gestión de evaluación (ver sección 4.5), por lo que descartamos su tiempo. Sin embargo, consideramos de especial relevancia interpretar los errores de interacción del informe para poder notificar problemas de usabilidad. Por ello, añadimos al tiempo de análisis el *tiempo de interpretación del informe de errores de interacción* ($T_{3.1}$).

$$TA_{M.Def} = T_{3.1} \quad (5.6)$$

Centrándonos en MUSiC y posicionándonos en el peor de los casos, descartamos el tiempo de interpretación de métricas y la elaboración del informe al asumir que son automáticos. Esto nos dirige a limitar la duración de la fase de análisis en MUSiC al *tiempo de análisis de los vídeos generados por los usuarios de pruebas* ($T_{3.2}$).

$$TA_{MUSiC} = T_{3.2} \quad (5.7)$$

Mediante la tabla 5.24 exponemos los diferentes tiempos que componen las duraciones totales de ambas metodologías, asociando cada uno de ellos a la metodología correspondiente.

Fase	*Tiempo*	*Descripción*	*Metodología*
Definición	$T_{1.1}$	*Definición de tareas y aplicación*	*Ambas*
Definición	$T_{1.2}$	*Definición de casos favorables*	*M. Definida*
Ejecución	$T_{2.1}$	*Integración de la aplicación evaluada*	*M. Definida*
Ejecución	$T_{2.2}$	*Definición de caminos de interacción correcta*	*M. Definida*
Ejecución	$T_{2.3}$	*Ejecución y realización de pruebas*	*Ambas*
Análisis	$T_{3.1}$	*Interpretación del informe de errores de interacción*	*M. Definida*
Análisis	$T_{3.2}$	*Análisis de los vídeos generados por los usuarios*	*MUSiC*

Tabla 5.24 Resumen de los tiempos que componen las fases de las metodologías a comparar

Como conclusión de este análisis, expresamos el tiempo de evaluación de la metodología definida ($T_{M.Def.}$) descomponiendo los tiempos de las fases en los explicados:

$$T_{M.Def} = (T_{1.1} + T_{1.2}) + (T_{2.1} + T_{2.2} + T_{2.3}) + T_{3.1} \quad (5.8)$$

Análogamente, la expresión correspondiente al tiempo de evaluación de MUSiC (T_{MUSiC}) es:

$$T_{MUSiC} = T_{1.1} + T_{2.3} + T_{3.2} \quad (5.9)$$

Para realizar la comparación de los tiempos de evaluación, estudiamos la diferencia entre ambos ($T_{dif.}$). Centrando la atención en $T_{1.1}$ y $T_{2.3}$, advertimos que ambos tiempos forman parte de ambas evaluaciones. Dentro de la comparación, asumimos que el *tiempo de definición de tareas y aplicación* ($T_{1.1}$) es similar en ambas evaluaciones por lo que es descartado. Al igual que $T_{1.1}$, el *tiempo de ejecución de las pruebas por parte de los usuarios de prueba* ($T_{2.3}$) se considera igual en ambas evaluaciones ya que asumimos la posibilidad de paralelizar las pruebas en MUSiC. Despejando los tiempos descartados obtenemos la siguiente expresión:

$$T_{dif} = T_{MUSiC} - T_{M.Def} = T_{3.2} - (T_{1.2} + T_{2.1} + T_{2.2} + T_{3.1}) \quad (5.10)$$

Mediante esta expresión podemos medir la diferencia de tiempos entre MUSiC y la metodología definida.

Siendo los resultados positivos los casos en los que la metodología definida requiere menos tiempo.

Debido a la naturaleza de los pasos de las evaluaciones, se detecta una gran dificultad: *resulta difícil requerir del tiempo necesario para realizar un número elevado de tareas en varios contextos utilizando algún sistema de grabación* debido al *tiempo de análisis de los vídeos generados por los usuarios de pruebas* ($T_{3.2}$). Esta tarea al no poder paralelizarse, requiere el tiempo total de todos los vídeos generados anteriormente, lo que hace que una evaluación completa sea costosa e inviable de forma voluntaria.

Por ello *optamos por realizar los pasos de la evaluación con dos tareas y realizar las estimaciones de las duraciones totales*. A continuación, estudiamos los tiempos que componen T_{dif} para extraer las variables a cuantificar y disponer de un modelo con el que realizar las estimaciones.

Analizando la parte de la expresión referente a MUSiC disponemos sólo de $T_{3.2}$.

- El *tiempo de análisis de los vídeos generados por los usuarios de pruebas* ($T_{3.2}$) se traduce en el tiempo que el desarrollador necesita para visualizar todos los vídeos de todos los usuarios de pruebas. Para estimar este tiempo se multiplica el *tiempo total de todos los vídeos generados* (tv) por el *factor de análisis de vídeo* (FAV), siendo este factor la proporción por la cual incrementa el tiempo de análisis en función de la duración de un vídeo. Es decir, si dispusiéramos de una hora de vídeos y un factor de análisis de vídeo de 2.5, estimaríamos que el desarrollador necesitaría 2 horas y media para analizar dicha hora.

$$T_{3.2} = tv * FAV \quad (\ \mathbf{5.11}\ )$$

En cuanto a la parte de la expresión T_{dif} referente a la metodología definida disponemos de $T_{1.2}, T_{2.1}, T_{2.2}$ y $T_{3.1}$.

- En primer lugar, asumimos que *tiempo de definición de los casos favorables* ($T_{1.2}$) es independiente del número de usuarios, entornos y dispositivos ya que el estudio se realiza sobre los resultados de los casos generados automáticamente. Al estar los casos clasificados por orden de probabilidad siempre se realizará el estudio sobre los primeros casos mostrados, siendo éstos los más relevantes. Por ello, asumimos que su tiempo es constante (T_{cf}).

$$T_{1.2} = T_{cf} \quad (5.12)$$

- En segundo lugar, el *tiempo de integración de la aplicación evaluada con la plataforma* ($T_{2.1}$) se traduce en el tiempo que el desarrollador necesita para integrar los diferentes eventos generados por los componentes de su aplicación con la librería de integración (ver apartado 4.3.2). A su vez, podemos estimar dicho tiempo multiplicando el *número total de los eventos a integrar* (ne) por el *factor de integración de un evento* (FE), que es el tiempo medio que tarda un desarrollador en integrar un evento.

$$T_{2.1} = ne * FE \quad (5.13)$$

- El *tiempo de definición de caminos de interacción correcta* ($T_{2.2}$) se traduce en el tiempo que el desarrollador necesita para grabar los caminos de interacción correcta de todas las tareas (ver apartado 4.5.1.4) y añadir las propiedades de fin de tarea y dependencias entre eventos. Para estimar este tiempo multiplicamos el *tiempo de grabación de todas las tareas* (tg) por el *factor de grabación* (FG), siendo este factor la proporción por la cual incrementa el tiempo de grabación en función de la duración de una tarea. Si un desarrollador grabara varias tareas cuya duración total es de 10 minutos y un factor de grabación de 1.5, estimaríamos en 15 minutos la definición de los caminos de interacción correcta.

$$T_{2.2} = tg * FG \quad (5.14)$$

- El *tiempo de interpretación del informe de errores de interacción* ($T_{3.1}$) se traduce en el tiempo que el desarrollador necesita para analizar los errores de interacción encontrados mediante la herramienta de escritorio. Para estimarlo, multiplicamos el *número de errores de interacción encontrados* (neg) por el *factor de análisis de un error de interacción encontrado* (FA), siendo éste el tiempo que se tarda en evaluar un error de interacción encontrado.

$$T_{3.1} = neg * FA \quad (5.15)$$

Finalizando con la identificación de las variables, sustituimos los valores en la expresión 5.10 y concluimos la fórmula presentada a continuación.

$$T_{dif} = tv * FAV - \left(T_{cf} + ne * FE + tg * FG + neg * FA\right) \quad (5.16)$$

Esta fórmula de T_{dif} más detallada nos permite estimar la diferencia de tiempos entre ambas evaluaciones de un modo más explícito y con la cual realizaremos el diseño de las pruebas y el experimento.

5.3.1.3. CARACTERÍSTICAS DEL EXPERIMENTO

Resumiendo los diferentes valores de tiempo a cuantificar en la tabla 5.25, *presentamos un experimento en el que varios desarrolladores y usuarios realizan los pasos de la metodología definida*. Gracias a ello, no sólo nos permitirá medir las variables de tiempo y el número de problemas de usabilidad detectados, sino también dotar a los sujetos del experimento de la experiencia suficiente para contestar al cuestionario.

Para la realización de este experimento hemos necesitado a usuarios de pruebas para realizar la ejecución de las pruebas de la metodología y la grabación de los vídeos de MUSiC. Para desarrollar los pasos restantes de los dos métodos, también hemos necesitado a participantes con un perfil técnico de desarrolladores

al tener que realizar la integración de la aplicación evaluada con la plataforma dentro de uno de los pasos del experimento.

Metodología	*Variable*	*Descripción de la cuantificación*
MUSiC	tv	*Medición de la duración de vídeos de usuarios realizando tareas de una evaluación*
MUSiC	FAV	*División entre la duración de los vídeos grabados y la duración de sus respectivos análisis*
M. Definida	T_{cf}	*Medición de la duración de la definición de casos favorables por parte del desarrollador*
M. Definida	ne	*Número de eventos a integrar en la aplicación evaluada*
M. Definida	FE	*División entre la duración de la integración de la aplicación evaluada y el número de eventos a integrar en la aplicación* (ne)
M. Definida	tg	*Medición de la duración de la grabación de tareas con la aplicación de grabación (ver apartado 4.6.1.3)*
M. Definida	FG	*División entre la duración de la generación completa de los caminos de interacción correcta y la grabación de las tareas* (tg)
M. Definida	neg	*Número total de errores de interacción generados en la fase de evaluación*
M. Definida	FA	*División entre la duración del análisis de los eventos de interacción encontrados y el número de eventos de interacción generados*(neg)

Tabla 5.25 Resumen de variables de tiempo a cuantificar en el experimento de validación de la metodología

Dentro de los participantes con el rol de desarrolladores, hemos dispuesto de 12 voluntarios que han trabajado en el sector de las TICs. Sus características se muestran en la tabla 5.26.

Característica	*Descripción*
Nivel de estudios	*2 doctores* *7 postgraduados* *2 graduados universitarios* *1 con titulación superior no universitaria*
Nivel de programación	*8 han programado aplicaciones Android* *4 han programado aplicaciones Java*
Conocimientos de usabilidad	*8 con nivel bajo* *3 con nivel medio* *1 con nivel alto*

Tabla 5.26 Descripción del grupo de desarrolladores que realizó el experimento de validación de la metodología

En cuanto a participantes con el rol de usuario de pruebas, hemos dispuesto de los 12 usuarios que evaluaron la aplicación *Maicvalia* (ver tabla 5.13). Para evadir el efecto aprendizaje, se desarrolla este experimento utilizando la aplicación *Maicbay*, que consta de 46 eventos de interacción y tarea a integrar y debe utilizarse en tres entornos: sentado, tumbado y caminando (ver tabla 5.12).

A continuación, mediante la tabla 5.27 se describen las tareas del experimento junto con las variables cuantificadas.

Tareas del experimento	*Objetivo*	*Descripción*
U1. Ejecución y realización de pruebas	tv, neg	*Los usuarios de pruebas deben realizar las tareas definidas en los pasos anteriores en tres contextos diferentes (sentado, tumbado y caminando) utilizando al mismo tiempo un prototipo de grabación construido (ver figura 5.10) y la herramienta de usuario de pruebas (ver sección 4.6) de la plataforma de soporte de nueva metodología.*
D1. Definición de tareas y aplicación	*Cuestionario*	*Los desarrolladores deben definir una nueva aplicación y dos tareas (registro de usuario y búsqueda de producto) utilizando la aplicación de escritorio de gestión de evaluación (ver apartado 4.5.1.2).*
D2. Definición de usuarios y entornos	T_{cf}	*Los desarrolladores deben definir y razonar un caso favorable basado en los resultados expuestos en la aplicación de escritorio de gestión de evaluación generados a partir de la base de conocimiento.*
D3. Integración de la aplicación evaluada	ne, FE	*Con el código fuente de una de las interfaces de la aplicación Maicbay, los desarrolladores deben añadir la librería de integración (ver sección 4.3) dentro del código en una de sus interfaces.*
D4. Definición de caminos de interacción	tg, FG	*Los desarrolladores deben generar los caminos de interacción correcta correspondientes a las tareas definidas en el primer paso haciendo uso de la aplicación de escritorio de gestión de evaluación y la aplicación móvil de grabación de caminos de interacción.*
D5. Análisis y presentación de resultados	FAV, FA	*(A) Por un lado, mediante el uso de las grabaciones anteriores, los desarrolladores realizan el análisis de los vídeos generados por el prototipo de grabación construido. (B) Por otro lado, analizan los resultados obtenidos por la herramienta de usuario de pruebas y expuestos mediante la aplicación de escritorio de gestión de evaluación (ver apartado 4.5.1.5).*

Tabla 5.27 Descripción del conjunto de tareas del experimento de validación de la metodología

Por un lado y aunque corresponde a fases más avanzadas, los usuarios de pruebas realizan el primer paso del experimento para generar el material necesario (vídeos y errores de interacción) que será utilizado en las sesiones con los desarrolladores. Para la realización de la captura de la interacción en formato vídeo, hemos creado el prototipo de grabación mostrado en la figura 5.10. Este prototipo consta principalmente de una cámara GoPro[40]

[40] http://www.gopro.com

(modelo Hero 3+ Black) que ha sido adherida a un soporte en el que también ha sido fijado un dispositivo móvil LG Nexus 5, ofreciendo la misma funcionalidad de captura de interacción que expone la herramienta DRUM. Aunque hemos asumido que los usuarios de pruebas pueden realizar las tareas de forma paralela, al haber realizado sólo un prototipo esto no es posible dentro de este experimento. Por lo tanto, el prototipo es cedido a los usuarios de prueba uno a uno para que generen el material a analizar por los desarrolladores.

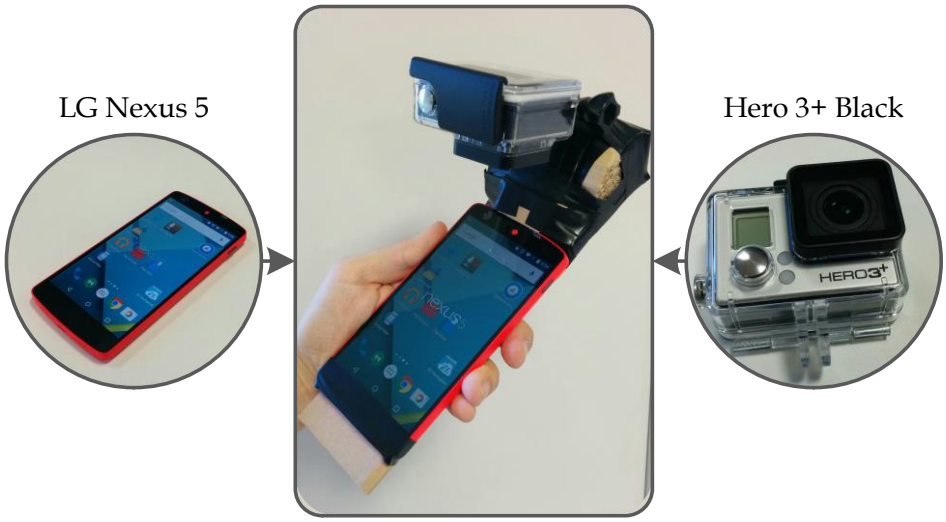

Figura 5.10 Prototipo de grabación creado para la captura de la interacción en vídeo

Una vez completada la información por parte de los usuarios de pruebas, se realizan los pasos del experimento restantes en sesiones individuales de 45 minutos con los desarrolladores. En estas sesiones se les explica el objetivo, duración de la sesión y los pasos a realizar. Después se cuantifican los resultados mediante la observación del participante. Es importante señalar que en el último paso del experimento (D5. Análisis y presentación de resultados) *se cuantifican los problemas de usabilidad generados tanto con el análisis de los vídeos como con el análisis de los resultados de la nueva metodología para validar que no hay un descenso significativo en el número de problemas de usabilidad detectados por la metodología definida en comparación con MUSiC.*

Al finalizar la sesión, se pide a los desarrolladores que rellenen un cuestionario para el estudio de la aceptación tecnológica. Dicho cuestionario dispone de varias afirmaciones que el desarrollador debe leer y responder sobre el grado de conformidad o disconformidad con las mismas mediante una escala Likert de 7 puntos. Con él se examinan las características de un sistema a través de la facilidad de uso y la utilidad percibida. Partiendo de [Davis+89] definimos tres parámetros a medir: la facilidad de uso, la utilidad percibida y la intención de conducta.

- La *facilidad de uso* nos permite medir el grado de esfuerzo que una persona percibe al estar usando un sistema particular. En este caso, la plataforma de soporte a la metodología. Para medirlo, se han formulado las afirmaciones F1, F2, F3, F4 y F5 de la tabla 5.28.
- La *utilidad percibida* refiere al grado en el que una persona cree que usando un sistema particular mejoraría su trabajo. Para medirlo, se han formulado las afirmaciones U1, U2, U3, U4 y U5 de la tabla 5.28.

  La *intención de conducta* nos permite estudiar la disposición de una persona para adquirir una conducta determinada, en este caso una buena actitud frente a la plataforma de soporte y la metodología. Para medirlo, se han formulado las afirmaciones I1 e I2 de la tabla 5.28.

Id	*Afirmación*
F1	*Aprender a usar el sistema y la metodología ha sido fácil para mí*
F2	*Es un sistema fácil de manejar y de interactuar con él*
F3	*Sería fácil llegar a ser un usuario experto en el manejo del sistema*
F4	*La metodología de evaluación facilita el realizar evaluaciones*
F5	*En general, tanto la metodología como el sistema son sencillos de utilizar*
U1	*El uso de la metodología y sistema permitiría evaluar aplicaciones móviles con mayor rapidez*
U2	*Esta metodología ahorra recursos (tiempo, usuarios, entornos, tareas a realizar) en la evaluación de la usabilidad de una aplicación móvil*
U3	*El uso de la metodología y sistema permitiría que desarrollara aplicaciones de mayor calidad más rápido*
U4	*El uso de la metodología y sistema es sencillo y fácil*
U5	*Encuentro este sistema útil en mi trabajo*
I1	*Recomendaría el sistema a otros usuarios que requieran evaluar sus aplicaciones móviles*
I2	*Utilizaría el sistema en futuras evaluaciones de aplicaciones móviles*

Tabla 5.28 Cuestionario Likert del estudio de la aceptación tecnológica

5.3.2. RESULTADOS

Después de realizar el experimento y obtener todos los resultados gracias a la colaboración desinteresada de los participantes que han ayudado a realizar los experimentos de este trabajo, procedemos al análisis de los resultados.

Primeramente, validaremos que la diferencia de tiempos entre la metodología definida y MUSiC es positiva y analizaremos el comportamiento de dicha diferencia. En segundo lugar, estudiaremos el número de problemas de usabilidad encontrados en ambas evaluaciones y comprobaremos si hay una diferencia significativa entre ellos. Finalmente analizaremos las respuestas al cuestionario de aceptación tecnológica.

5.3.2.1. TIEMPO DE EVALUACIÓN

Habiendo medido la totalidad de las variables expuestas en la tabla 5.27 y sustituyendo las mismas en la fórmula 5.16 mostrada anteriormente, se estima la diferencia de tiempos.

En primer lugar, mostramos mediante la tabla 5.29 el resumen de las variables medidas ajenas al desarrollador que van a adquirir un valor constante.

Variable	*Valor*
tv	*76min. 24s. de vídeo grabado por los usuarios de pruebas*
ne	*46 eventos a integrar en la aplicación evaluada*
neg	*15 errores de interacción registrados por la nueva metodología*

Tabla 5.29 Resumen de variables medidas de valor constante ajenas al desarrollador

Por otro lado, mostramos mediante la tabla 5.30 los diferentes valores medios estimados que presentan las variables cuantificadas referentes al desarrollador.

Variable	*Media estimada*	*Intervalo de confianza [95%] de la media estimada*
FAV	*1.90*	*[1.48, 2.33]*
T_{cf}	*78*	*[52.54,103.46]*
FE	*54.85*	*[43.8, 65.9]*
FG	*1.29*	*[1.10, 1.48]*
FA	*14.17*	*[9.91, 18.42]*
tg	*228.67*	*[225.18, 232.15]*

Tabla 5.30 Resumen de variables referentes al desarrollador

Observando el *factor de análisis de vídeo* (FAV), percibimos que el aumento del tiempo de análisis de los vídeos es de casi el doble de la duración del vídeo analizado (1.9), resultado similar al manifestado con la implementación real de DRUM [Macleod+93], donde documentan un aumento del doble o incluso el triple. El *tiempo de definición de los casos favorables* (T_{cf}) oscila entre los 53 segundos y casi los 2 minutos. También percibimos que el *factor de integración de un evento* (FE) muestra que un desarrollador tarda en integrar un evento de su aplicación una media estimada de 55 segundos. Además, dentro de la definición de los caminos de interacción, el *factor de grabación* (FG) indica que el tiempo medio de generación del camino de interacción correcta es de entre 1.10 y 1.48 veces el tiempo que tarda en grabar el desarrollador la tarea. Finalmente, el *factor de análisis de un error de interacción encontrado* (FA) indica que los desarrolladores tardan entre 10 y 20 segundos en analizar un error de interacción.

	Media estimada (min)	*Intervalo de confianza [95%] de la media estimada (min)*
MUSiC	*145.8*	*[113.34, 178.27]*
M. Definida	*51.85*	*[42.66, 61.04]*
Diferencia	*93.96*	*[62.88, 125.03]*

Tabla 5.31 Duración estimada de ambas evaluaciones y su diferencia

Aunque el único valor real en la tabla 5.31 es la diferencia de tiempos (ya que hemos descartado el tiempo de varios pasos), decidimos plasmar la duración estimada con un fin ilustrativo. Recordamos que estas estimaciones eran para una evaluación en la que participaban 12 usuarios que realizaban 2 tareas en 3 entornos distintos, la diferencia estimada calculada en función de los datos presentados es de 93.96 minutos con un intervalo de confianza del 95% entre 62.88 y 125.03 minutos donde el tiempo estimado de MUSiC es de 2 horas y 25 minutos y el de la metodología definida 52 minutos. Por ello afirmamos lo siguiente:

> *Deducimos y estimamos que la metodología desarrollada, en el caso de una validación con las características planteadas, requiere el 35.5% del tiempo que necesita MUSiC para realizar la misma evaluación.*

Habiendo comprobado en las condiciones del experimento que la diferencia es positiva y favorable para la metodología definida, a continuación procedemos a estudiar el comportamiento de dicha diferencia en función de tv, tg, ne y neg (ver tabla 5.29 y tabla 5.30). De cara a ofrecer un estudio más claro, centrándonos en tg y tv, podemos asumir que el *tiempo de grabación de todas las tareas* (tg) es el *tiempo medio de grabación de una tarea* (T_{mg}) por el número total de las tareas a realizar.

$$tg = T_{mg} * nºtareas \quad (5.17)$$

Además, también podemos asumir que el *tiempo total de todos los vídeos generados* (tv) es el tiempo medio de grabación del vídeo de una tarea (T_{mgv}) por el número de tareas, el número de entornos y número de usuarios.

$$tv = T_{mgv} * nºtareas * nºusuarios * nºentornos \quad (5.18)$$

En una evaluación real, el *tiempo medio de grabación de una tarea* (T_{mg}) será inferior al tiempo medio de grabación del vídeo de una tarea. No obstante, serán de duración similar por lo que asumimos que su duración es la misma.

Con este nuevo enfoque, estudiaremos la correlación que existe entre las variables y el comportamiento de la diferencia de las evaluaciones. Para ello generamos por cada resultado obtenido de desarrollador, la diferencia de tiempos en toda la combinación de casos con los valores mostrados en la tabla 5.32.

Variable	*Valores*
Tiempo medio de grabación de tarea	*10, 20, 30, 40, 50, 100 y 150*
Número de tareas	*1, 5, 10 y 15*
Número de usuarios	*3, 5, 8, 10, 15, 20 y 30*
Número de entornos	*1, 2, 3, 4, 5 y 10*
Número de elementos a integrar	*15, 30, 50, 100 y 150*
Número de errores de interacción registrados	*15, 30, 50, 100 y 150*

Tabla 5.32 Valores de las variables con las que estudiamos el comportamiento de la diferencia de tiempos

Una vez obtenidas todas las combinaciones estudiamos la correlación existente entre las variables en estudio mediante la matriz de correlaciones mostrada en la tabla 5.33.

En ésta se indican todas las correlaciones entre (A) el tiempo medio de grabación de tarea, (B) el número de tareas, (C) el número de usuarios, (D) el número de entornos, (E) el número de eventos a integrar, (F) el número de errores de interacción registrados, (G) el tiempo requerido de MUSiC, (H) el tiempo requerido por la metodología definida y (I) la diferencia de tiempos entre ambas.

	A	*B*	*C*	*D*	*E*	*F*	*G*	*H*	*I*
A	*1*								
B	*0*	*1*							
C	*0*	*0*	*1*						
D	*0*	*0*	*0*	*1*					
E	*0*	*0*	*0*	*0*	*1*				
F	*0*	*0*	*0*	*0*	*0*	*1*			
G	*0.411*	*0.225*	*0.357*	*0.388*	*0*	*0*	*1*		
H	*0.189*	*0.103*	*0*	*0*	*0.805*	*0.208*	*0.136*	*1*	
I	*0.409*	*0.223*	*0.358*	*0.389*	*-0.016*	*-0.004*	*0.999*	*0.117*	*1*

Tabla 5.33 Matriz de correlaciones de las variables en estudio

Viendo las diferentes correlaciones notificamos que la diferencia de tiempos sufre una gran alteración por el tiempo requerido por MUSiC (0.999) en contraste con el tiempo requerido por la metodología definida (0.117). Esto nos indica que el tiempo de MUSiC es mucho más elevado que el de la nueva metodología.

Observamos que el tiempo de la nueva metodología aumenta considerablemente en función del número de eventos a integrar (0.805), donde se estima como ejemplo que un desarrollador requerirá alrededor de 45 minutos para integrar una aplicación con 46 eventos. Además, el tiempo de la metodología definida es penalizado por el número de errores de interacción registrados (0.208). Aunque se perciben otras variables (como el número de tareas y el tiempo medio de la tarea) que aumentan el tiempo requerido por la metodología definida, aumentan también en mayor medida el tiempo requerido de MUSiC.

Consecuentemente, *deducimos que las variables que realmente penalizan significativamente el tiempo requerido de la nueva metodología son el número de eventos a integrar y el número de errores de interacción registrados*. Sin embargo, MUSiC dispone de una fuerte penalización en el incremento del tiempo total de las tareas, donde interviene el tiempo medio de las tareas, el número de tareas, número de entornos y número de usuarios. Lo que le hace ser menos eficiente que la metodología definida al ser dichos elementos más propensos a aumentar.

5.3.2.2. NÚMERO DE PROBLEMAS DE USABILIDAD

El otro punto a validar dentro de la hipótesis es que el número de problemas de usabilidad encontrados en las dos evaluaciones no difieren significativamente.

Para ello se ha procedido a comparar los problemas de usabilidad encontrados por los desarrolladores mediante ambas evaluaciones. Al disponer de una muestra de solo 12 casos se ha utilizado el test de Student para datos emparejados. En este caso, comprobamos la diferencia de medias mediante la prueba de significación. En ésta, obtenemos un valor p muy superior a 0.05, lo que implica que no podemos rechazar la hipótesis de igualdad de medias ya que no hay evidencias que muestren una diferencia significativa: $t(11) = 0.43$, $p=0.67$. Además, su intervalo de confianza del 95% de [-0.341, 0.508] presenta una diferencia no superior a 1 (un problema de usabilidad).

Por ello, mediante este análisis comprobamos que no hay una diferencia significativa entre el número de problemas detectados entre ambas metodologías, por lo que afirmamos lo siguiente:

> *Tanto la evaluación de aplicaciones mediante la metodología definida como mediante MUSiC presentan una detección de problemas de usabilidad similar, por lo que consideramos a ambas igual de eficaces en cuanto a la detección de problemas.*

5.3.2.3. ACEPTACIÓN TECNOLÓGICA

Una vez estudiado el tiempo y el número de errores de usabilidad detectados, nos disponemos a estudiar la aceptación tecnológica de la metodología definida mediante el análisis expuesto en la figura 5.11, donde se muestran las respuestas a las afirmaciones planteadas al final de la descripción del experimento en el apartado 5.3.1.3 (ver tabla 5.28).

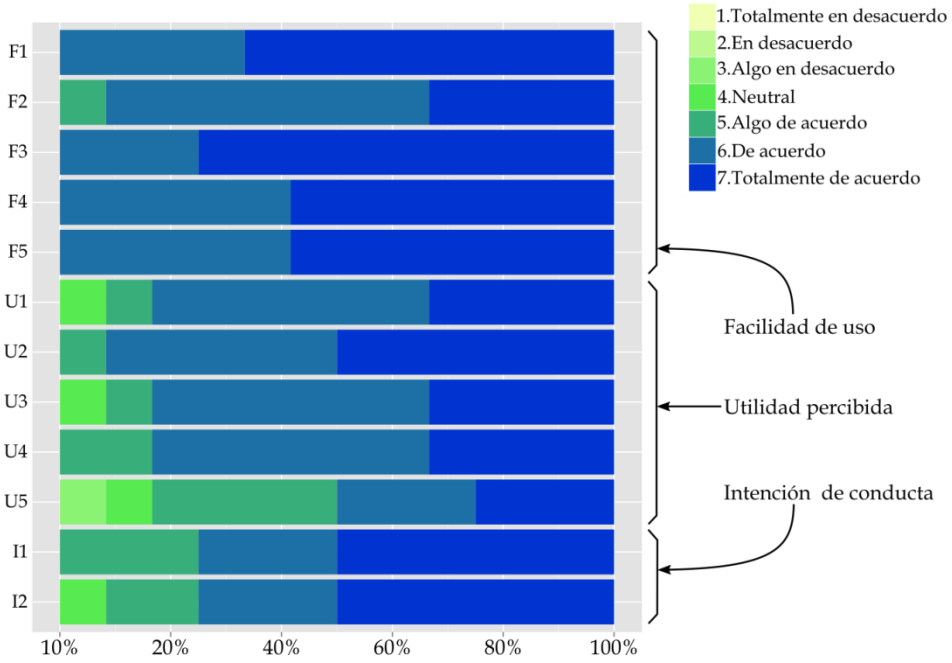

Figura 5.11 Respuestas a las afirmaciones del cuestionario de aceptación tecnológica

En primer lugar, percibimos la *facilidad de uso* como el parámetro mejor valorado donde las afirmaciones disponen de las dos máximas valoraciones (más del 50% de los desarrolladores mostraban estar totalmente de acuerdo) exceptuando la afirmación F2 (es un sistema fácil de manejar y de interactuar con él), donde solo uno de los 12 desarrolladores ha manifestado estar algo de acuerdo frente al resto que está de acuerdo o totalmente de acuerdo (91.7%).

En cuanto a la *utilidad percibida*, aunque los resultados no son tan buenos se muestran también muy positivos. Exceptuando la afirmación U5 (encuentro este sistema útil en mi trabajo), el resto de las afirmaciones mostraban las dos máximas valoraciones en el

83.3% de las respuestas. En el caso de U5, solo uno de los desarrolladores ha valorado negativamente la afirmación, siendo éste un desarrollador que no ha implementado aplicaciones móviles, comentando que en su trabajo "*no realizan evaluaciones de interfaz o usabilidad*".

Finalmente, en la intención de conducta percibimos que 11 de los 12 desarrolladores utilizarían el sistema en futuras evaluaciones y el 100% de los desarrolladores recomendaría el sistema a otros que requieran evaluar sus aplicaciones móviles.

Viendo esta aceptación positiva, finalizamos el análisis de los resultados obtenidos.

5.3.3. CONSIDERACIONES DE LA METODOLOGÍA

Durante esta sección hemos estudiado cómo se reducen los recursos mediante el uso de esta metodología frente al uso del método MUSiC. Primeramente, hemos analizado los principales recursos, el *coste general de la evaluación*, el *número de usuarios de pruebas y entornos necesarios* y el *tiempo de evaluación*.

Asociamos el mayor coste de la evaluación al número necesario de usuarios y entornos. Dicho número es reducido eligiendo un caso favorable, como ha sido validado en la sección de validación del uso de la base de conocimiento (ver sección 5.2).

Para realizar un análisis más explícito nos hemos centrado en el tiempo de la evaluación, donde se ha calculado que *la nueva metodología, en el caso planteado, requiere el 35.5% del tiempo que requiere MUSiC*, ya que la metodología definida tarda una hora y media menos en realizar una evaluación cuya duración con MUSiC se estima en 2 horas y 25 minutos.

Además, para entender más en detalle el tiempo necesario se ha estudiado la relación del mismo con las diferentes variables que lo modifican. En este análisis hemos concluido que la duración total de la nueva metodología (desde que comienza la evaluación hasta que termina) no depende del número de usuarios de pruebas

necesarios ni del número de entornos. Sin embargo, sí detectamos dependencia con otras variables.

Dentro de la fase de definición no se ha considerado ningún paso crítico aunque la definición de las tareas dependa del número de las mismas. En la fase de ejecución hemos observado una fuerte dependencia dentro del paso de integración de la aplicación con la plataforma, donde se ha estimado que un desarrollador tardaría alrededor de 45 minutos en integrar una aplicación móvil con 46 eventos a registrar, ya que se ha estimado una media de 55 segundos por evento. Además, en la generación de los casos favorables se ha estimado que el tiempo de generación de un camino de interacción correcta es 1.48 veces el tiempo que tarda el desarrollador en grabar la tarea. Finalmente en la fase de análisis, el enfoque formativo es el que en este caso ofrece un cuello de botella debido a que el evaluador debe interpretar los errores de interacción registrados para inferir el problema de usabilidad asociado. Se ha estimado que la fase de análisis de una evaluación donde se hayan registrado 100 errores de interacción, durará alrededor de 25 minutos, habiendo estimado 14.17 segundos de análisis por error de interacción. Resumiendo el estudio, *afirmamos que las variables que realmente penalizan significativamente el tiempo requerido de la nueva metodología son el número de eventos a integrar y el número de errores de interacción registrados*.

Una vez comprobado y validado que la metodología requiere menos tiempo, hemos validado que su eficacia no disminuye. Para ello hemos contrastado la diferencia de problemas de usabilidad encontrados con ambas metodologías mediante una prueba de significación. En ella se ha demostrado que *no hay una diferencia significativa en el número de problemas de usabilidad detectados* por lo que conservan ambos métodos la misma eficacia.

Ajeno a la validación de la hipótesis, se ha realizado adicionalmente un cuestionario para estudiar la facilidad de uso, la utilidad percibida y la intención de conducta del desarrollador mostrando una aceptación considerable.

Gracias a las afirmaciones deducidas y a este estudio, *la hipótesis expuesta en la validación de la metodología, que es la principal hipótesis formulada en esta tesis, queda confirmada.*

CAPÍTULO 6

CONCLUSIONES

«Debe quedar claro que con todo lo que he dicho hasta en mi investigación tengo una deuda enorme con el trabajo de otros, mis colegas que han aportado muchas de las ideas que he usado y muchos ejemplos interesantes de análisis y mis colaboradores, sin cuyo cerebro, ojos y manos muy poco se habría hecho»,
Dorothy Crowfoot Hodgkin (1910-1994)

ÍNDICE DE CAPÍTULO 6

6.1. Visión general del trabajo 226
6.2. Contribuciones 227
6.3. Resultados obtenidos 231
6.4. Trabajo futuro 234
6.5. Consideraciones finales 237

Para empezar con este capítulo debemos releer la cita con la que comenzamos ya que ni una sola afirmación concluida ha sido aquí plasmada sin ser conscientes y ser identificados con las palabras de Dorothy Crowfoot Hodgkin.

Durante los capítulos anteriores hemos ido describiendo en detalle todas las actividades realizadas durante el desarrollo de esta tesis doctoral. Así, comenzamos exponiendo la introducción general del entorno de trabajo y las motivaciones iniciales dentro de esta área de conocimiento. Después, realizamos el análisis del estado del arte e identificamos las limitaciones y necesidades que existen dentro de la evaluación de la usabilidad de aplicaciones móviles, lo que nos permitió identificar las oportunidades de mejora para formular la hipótesis y proponer una solución con las características y requisitos que debe cumplir. Seguidamente detallamos las características generales de la metodología de

evaluación de la usabilidad de aplicaciones móviles, su diseño y la herramienta de soporte construida para finalmente comentar los experimentos y pruebas realizadas con el fin de obtener unos resultados que empíricamente permitan comprobar las bondades de la solución. En este capítulo ofrecemos un resumen de todo ello, pero centrando la atención en los resultados obtenidos durante el desarrollo de esta tesis doctoral, principalmente en el uso de la solución desarrollada y en las conclusiones que pueden extraerse de estas experiencias junto con las aportaciones finales atribuibles al trabajo realizado. Asimismo planteamos algunas posibles acciones de mejora futuras que podrían realizarse en torno a esta tesis.

6.1. VISIÓN GENERAL DEL TRABAJO

El principal producto del trabajo de investigación realizado en esta tesis es una metodología para la evaluación de la usabilidad de aplicaciones móviles. Revelamos en el análisis del estado del arte realizado en el capítulo 2 varias limitaciones relacionadas con las aplicaciones móviles y la evaluación de usabilidad de las mismas.

Las principales limitaciones que presentan las aplicaciones móviles desde el punto de vista de la usabilidad (ver apartado 2.2.1) son la heterogeneidad existente en los dispositivos móviles, el reducido tamaño se sus pantallas, las complicadas interfaces de entrada, un significativo impacto de los elementos del contexto (p.ej., la conectividad extremadamente variable) y la elevada probabilidad de interrupción de las tareas, las cuales tienen un tiempo limitado.

Las limitaciones identificadas dentro de las evaluaciones de usabilidad de este tipo de aplicaciones son, desde un punto de vista general, la fuerte necesidad en el estudio de las características del entorno; y más concretamente dentro de las evaluaciones de usabilidad en entornos reales identificamos que el coste en cuanto a tiempo y esfuerzo de estas evaluaciones es muy elevado, que la fiabilidad de los datos capturados es baja y que la

privacidad de los usuarios se ve amenazada dentro de las grabaciones de las pruebas.

Después del estudio e identificación de las mencionadas limitaciones y habiéndolas contrastado con las soluciones existentes, concluimos que *no existe ningún método de evaluación de usabilidad de aplicaciones móviles que permita la evaluación en entornos reales persiguiendo reducir el coste de la evaluación en términos de recursos (equipamiento y tiempo necesarios) sin comprometer la fiabilidad de los resultados ni la privacidad de los usuarios que participen en las pruebas, tomando en cuenta el contexto y la experiencia previa, ofreciendo resultados que soporte tanto evaluaciones formativas como sumativas*.

A raíz de lo anterior, concebimos varios objetivos. En primer lugar, *planteamos como objetivo general desarrollar una metodología basada en una base de conocimiento para la evaluación de usabilidad de aplicaciones móviles que haga uso de una cantidad de recursos reducida*. Acorde con este objetivo general fueron establecidos tres objetivos específicos: *definir una metodología de evaluación de aplicaciones móviles*, *implementar una plataforma de soporte a la nueva metodología* y *verificar los resultados del uso de la nueva metodología*. Estos objetivos han sido cumplidos tal y como quedará patente a continuación.

6.2. CONTRIBUCIONES

Este trabajo doctoral ha dado como resultado varias contribuciones:

- En el capítulo 2 hemos realizado un estudio en el que hemos identificado las principales limitaciones desde la perspectiva de la usabilidad de las aplicaciones móviles y las limitaciones en la evaluación de las mismas (ver sección 2.2), estudiando los retos que se deben asumir en la evaluación y cómo los abordan en las soluciones existentes (ver sección 2.3) para finalmente proponer una nueva aproximación basada en la experiencia previa, describiendo

los componentes de la misma y los requisitos que deben cumplir (ver sección 2.4).

- En el capítulo 3 exponemos las principales *contribuciones científicas*. Destacamos especialmente la *definición teórica de la metodología con la cual podemos hacer uso de conocimiento previo para la mejora de la eficiencia de la evaluación de la usabilidad de aplicaciones móviles* (ver sección 3.1). Dentro de la base de conocimiento (ver sección 3.2) destacan varios modelos con los cuales hemos realizado varias aportaciones. En primer lugar, la *definición de un modelo de contexto centrado en la evaluación de aplicaciones móviles* (ver apartado 3.2.1). La *definición de un modelo de interacción* que permite modelar e identificar los diferentes errores de interacción que puede originar un usuario de pruebas (ver apartado 3.2.2). La *definición de un modelo de análisis integral* (ver apartado 3.2.3), recopilando tanto evaluaciones sumativas como formativas de aplicaciones móviles. Y por último, la *definición de un modelo de casos favorables* (ver apartado 3.2.4) que estudia y describe los casos (usuarios y entornos) en los que se detectan más errores de interacción.
- En el capítulo 4 exponemos las principales *contribuciones técnicas*. Remarcamos esencialmente la plataforma de soporte en sí, es decir, *la construcción de un conjunto de herramientas desarrolladas para posibilitar el uso de la metodología*. Dicho conjunto de herramientas automatizan ciertos pasos de la metodología y hacen uso del menor equipamiento posible para realizar tanto como la gestión de la metodología como la captura de los datos de las pruebas de evaluación de un modo remoto (principal reto de la plataforma de soporte al no sesgar la interacción).

  Dentro de esta plataforma aportamos el *desarrollo de una herramienta de usuarios de prueba basada en la plataforma Android que permite automatizar la captura de las pruebas de usabilidad* (ver apartado 4.6), capturando tanto el modelo de interacción como el modelo de contexto sin requerir otro

equipamiento que no sea el propio dispositivo del usuario de pruebas.

Además, ofrecemos una *implementación de una herramienta de desarrollador que permite agilizar las fases definidas en la metodología* (ver sección 4.5), ofreciendo funcionalidad que facilita la definición de nuevas evaluaciones, el estudio de los casos favorables, facilita la ejecución de las pruebas y permite automatizar el análisis de las mismas una vez los usuarios de pruebas han generado sus resultados.

Finalmente, aportamos un *desarrollo de una herramienta basada en una librería de integración* (ver sección 4.3) *y una aplicación Android de grabación de tareas* que permite la generación de caminos de interacción correcta (ver apartado 4.5.1.4) *para la detección y registro de errores de interacción de modo automático*. Esto permite agilizar la fase de definición y ejecución de la metodología definida.

- En el capítulo 5 exponemos la validación de la solución propuesta en esta tesis, donde aportamos tres estudios con base en la experimentación. Ofrecemos un *estudio basado en un experimento comparativo que analiza el sesgo producido por la captura de las pruebas de usabilidad mediante la herramienta de usuario de pruebas* (ver sección 5.1). Ofrecemos otro *estudio basado en un experimento que demuestra el uso de la base de conocimiento para calcular casos favorables y describir los valores de las variables de contexto en los que realmente exista una mayor probabilidad de encontrar errores* (ver sección 5.2). Finalmente, aportamos un *estudio que analiza la validez de la hipótesis planteada en esta tesis mediante un estudio basado en un experimento comparativo entre la solución propuesta y la metodología MUSiC* (ver sección 5.3).

Todas las contribuciones que hemos presentado han sido fruto de varias actividades de investigación dentro del área de la tesis que han dado lugar a varias publicaciones científicas que han servido tanto para confirmar el interés del tema y de la idea desarrollada como para adquirir la opinión de la comunidad e identificar, gracias a ello, varios puntos de mejora que han servido para

aumentar la calidad de este trabajo. Estas publicaciones se dividen en 2 publicadas en Journals y 8 en conferencias internacionales:

Journals

- Pretel, I., & Lago, A. B. (2012). Mobile Interaction Capturing System in Real Environments. *Computer Technology and Application, 8*, 533–543.
- Klein, B., Pretel, I., Vanhecke, S., Lago, A. B., & Lopez-de-Ipiña, D. (2013). Analysis of Log File Data to Understand Mobile Service Context and Usage Patterns. *International Journal of Interactive Multimedia and Artificial Intelligence, 2*(3), 15–22.

Conferencias internacionales

- Pretel, I., & Lago, A. B. (2010). Framework para Evaluación de la Calidad en Uso de Servicios Móviles. In *5th Iberian Conference on Information Systems and Technologies (CISTI)*. Santiago de Compostela, Spain.
- Pretel, I., & Lago, A. B. (2011). Capturing Mobile Devices Interactions Minimizing the External Influence. In *UBICOMM 2011, The Fifth International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies* (pp. 200–205). Lisbon, Portugal.
- Pretel, I., & Lago, A. B. (2012). Mobile-Human Interaction Monitoring System. In *Mobile Lightweight Wireless Systems* (Vol. 81, pp. 198–205). Springer Berlin Heidelberg.
- Klein, B., Pretel, I., Reips, U.-D., Lago, A. B., & Lopez-de-Ipiña, D. (2013). The Behaviour Assessment Model for the Analysis and Evaluation of Pervasive Services. In Á. Rocha, A. M. Correia, T. Wilson, & K. A. Stroetmann (Eds.), *Advances in Information Systems and Technologies* (Vol. 206, pp. 1129–1140). Springer Berlin Heidelberg.
- Pretel, I., & Lago, A. B. (2013). Effectiveness Measurement Framework for Field-Based Experiments Focused on Android Devices. In C. Collazos, A. Liborio, & C. Rusu (Eds.), *Human Computer Interaction* (Vol. 8278, pp. 123–130). Springer International Publishing.
- Pretel, I., & Lago, A. B. (2013). Sistema de evaluación de la efectividad del usuario sensible al contexto para aplicaciones móviles. In *CISTI 2013: 8ª Conferencia Ibérica de Sistemas y Tecnologías de Información. Evento de la IEEE*. Lisboa.
- Curiel, P., Pretel, I., & Lago, A. B. (2014). Interfaz orientada a la persona: acceso transparente a servicios de comunicación interpersonal. In *Actas de la 9ª Conferencia Ibérica de Sistemas y Tecnologías de Informacion* (Vol. 1, pp. 536–542). Barcelona, Spain.
- Pretel, I., & Lago, A. B. (2014). Evaluación remota de aplicaciones móviles híbridas: nueva aproximación en entornos reales. In *Actas de la 9ª Conferencia Ibérica de Sistemas y Tecnologías de Informacion* (Vol. 1, pp. 383–388). Barcelona, Spain.

Cabe destacar que varias actividades relacionadas con el área de esta tesis han sido aplicadas en el desarrollo de varios proyectos de investigación subvencionados tanto por el Gobierno de España (p.ej., *NEURONA*[41]) como por el Gobierno Vasco (p.ej., *Xperience*[42], *Q-Apps*[43] y *HiSozial*[44]). La realización de ciertas actividades relacionadas con el área dentro de proyectos de investigación nos ha servido para contrastar la aplicabilidad del trabajo desarrollado en la industria del software.

6.3. RESULTADOS OBTENIDOS

En primer lugar, debemos subrayar que los objetivos específicos definidos que propiciaban el cumplimiento el objetivo general de este trabajo (ver sección 1.3) han sido logrados satisfactoriamente.

- Hemos conseguido *definir una metodología de evaluación de aplicaciones móviles (OE1)* ya que se ha definido un conjunto de pasos y una base de conocimiento junto con sus modelos para componer la misma. Mediante esta definición se logran los requisitos de la metodología (RMs) definidos en el apartado 2.4.2.1.
- Hemos logrado *implementar una plataforma de soporte a la nueva metodología (OE2)* al lograr desarrollar una herramienta de captura de los modelos que conforman la base de conocimiento de la nueva metodología y una plataforma de soporte que ayude a la correcta ejecución de la nueva metodología. Mediante estos desarrollos se logran

[41] Proyecto NEURONA: Interconexión de infraestructuras abiertas para la validación integral de servicios móviles convergentes de nueva generación: calidad de experiencia, modelos de negocio y modelos de gestión. (2008-2010)
[42] Proyecto Xperience: Metodología para la evaluación de la experiencia del usuario con servicios móviles de nueva generación. (2009-2010)
[43] Proyecto Q-Apps: Plataforma para la evaluación de la calidad en uso de aplicaciones móviles. (2012-2013)
[44] Proyecto HiSozial: Plataforma de interacción para promover la inclusión social de las personas mayores. (2012-2013)

los requisitos de la plataforma (RPs) definidos en el apartado 2.4.2.2.

- Hemos logrado *verificar los resultados del uso de la nueva metodología (OE3)* mediante la realización de un experimento en el que validamos la correcta captura de los modelos que conforman la base de conocimiento, un segundo experimento con usuarios reales que estudia si el uso de la base de conocimiento nos permite definir una evaluación que permita encontrar más errores de interacción y finalmente, un tercer experimento con usuarios reales en el que validamos de un modo empírico la principal hipótesis definida en este trabajo.

La mayor parte de los resultados han sido obtenidos mediante la experimentación con la metodología y la plataforma de soporte ligada al objetivo específico OE3 y definida en el capítulo 5. Exponemos los resultados en base a los tres estudios presentados en la experimentación.

- *Análisis del sesgo producido por la captura de las pruebas de usabilidad mediante la herramienta de usuario de pruebas*. El sesgo producido puede deberse a la modificación de los elementos muy sensibles al cambio que conforman el contexto de las pruebas: la propia aplicación evaluada, el usuario de pruebas, su dispositivo y el entorno. En el caso de este trabajo, el entorno no se modifica al no requerir elementos adicionales (p.ej., cámaras, observadores humanos,…). Sin embargo, debíamos estudiar al usuario, su dispositivo y la aplicación evaluada. En este estudio, no detectamos un incremento significativo ni en la memoria física ni en el almacenamiento (la demanda de espacio no es considerada de importancia al compararla con las capacidades reales de los terminales móviles). Sin embargo, sí percibimos un incremento significativo en la demanda de procesador dependiendo del tipo de dispositivo: desde un 9.7% en terminales modernos (2013) hasta un 36% en antiguos (2010). Este incremento puede penalizar el tiempo de ejecución de la aplicación evaluada. Por un lado, no

detectamos una penalización en el tiempo por la librería de integración de la plataforma de soporte en tareas de larga duración (más de 10 segundos). Sin embargo, sí apreciamos una ligera penalización de hasta medio segundo en el tiempo total de tareas de corta duración. Por otro lado, aunque se demostró que se modifica el rendimiento, la percepción del usuario de pruebas no se vio alterada de un modo significativo ya que el tiempo de respuesta no aumenta más de 0.1 segundos.

- *Análisis del uso de la base de conocimiento*. Mediante este estudio validamos tanto el cálculo de los propios casos favorables como la descripción de las variables de contexto. Por un lado, hemos validado el uso de p_{comp} como criterio de clasificación de casos posibles para calcular los más adecuados, ya que dentro del mismo número de entornos, la elección del mejor caso genera aproximadamente hasta un 5% más de errores de interacción en el caso del escenario diseñado para el experimento. Por otro lado, hemos validado que tanto en variables de contexto cuantitativas como cualitativas podemos aproximar los valores más adecuados de las mismas para encontrar errores de interacción, dependiendo el número de los mismos del nivel de restricción a la hora de acotar rangos o elegir estados.
- *Análisis de la reducción de los recursos mediante el uso de la nueva metodología frente al uso del método MUSiC*. En este caso nos centramos en el estudio del tiempo de la evaluación, donde se ha calculado que la nueva metodología, en el caso planteado para el experimento, requiere el 35.5% del tiempo que requiere MUSiC (estimado el tiempo requerido de 2 horas y 25 minutos). Además, hemos estudiado la relación del tiempo con las diferentes variables que lo modifican. En este análisis hemos concluido que el tiempo de la nueva metodología no depende del número de usuarios de pruebas necesarios ni del número de entornos. Sin embargo, sí detectamos que en

el paso de definición de las tareas depende del número de las mismas. Además, hemos observado que las variables que realmente penalizan significativamente el tiempo requerido de la nueva metodología son el número de eventos a integrar en la fase de integración con la librería desarrollada (55 segundos estimados por evento a integrar) y el número de errores de interacción registrados en la fase de análisis de los errores, habiendo estimado 14.17 segundos de análisis por error de interacción. También se concluyó que no hay una diferencia significativa en el número de problemas de usabilidad detectados por lo que consideramos ambos métodos igual de eficaces. Finalmente, mediante un estudio con cuestionario de la facilidad de uso, la utilidad percibida y la intención de conducta concluimos una aceptación considerable.

Una vez expuestos los resultados, identificamos finalmente las oportunidades de mejora mediante la siguiente sección.

6.4. TRABAJO FUTURO

Mediante esta tesis ha quedado demostrado que utilizando la experiencia previa de evaluaciones anteriores podemos reducir los recursos necesarios en términos de tiempo y equipamiento. Sin embargo, durante el desarrollo de este trabajo se han identificado varias limitaciones, de las cuales varias han sido solventadas y otras han quedado identificadas como nuevas oportunidades de mejora.

- *Etiquetado automático de entornos*. La dinámica de las pruebas de evaluación desarrollada dicta que el usuario de pruebas realice conscientemente las tareas en un escenario específico descrito dentro de las instrucciones. Consideramos el uso de algoritmos de detección automática y etiquetado de los diferentes entornos en los que se realizan las pruebas una significativa mejora ya que el usuario podría realizar las pruebas sin necesidad de especificar el entorno. Además, esto permitiría notificar al

usuario que se encuentra en un entorno que pertenece a las pruebas, lo que aumentaría las probabilidades de completar la totalidad de las tareas en menor tiempo.

- *Cambios de entorno*. Mediante esta metodología hemos realizado varios modelos que han compuesto una base de conocimiento en la que el contexto juega un importante papel. Sin embargo, este conocimiento está diseñado para afrontar el dinamismo de los cambios de valor en variables dentro de un entorno concreto pero no un cambio total de entorno (p.ej., comenzando una tarea en la parada de autobús y terminándola en el mismo autobús). Por ello, se ha detectado que aunque las tareas realizadas con aplicaciones móviles son de duración relativamente corta, pueden darse situaciones como la del ejemplo. Debido a esto, se debe prestar especial atención al estudio de los cambios de entorno, su detección y su captura.
- *Grado de automatización*. Los análisis que permite la metodología definida en la presente tesis permite automatizar todo el proceso de cálculo de variables sumativas. Por el contrario, el grado de automatización de la evaluación formativa no llega a la crítica automática, que además de identificar problemas, ofrece recomendaciones de mejora (ver apartado 2.1.4.2.4). De cara a la dificultad que demanda este tipo de automatización, queda abierta una fuerte oportunidad de mejora.
- *Simplificación de la fase de integración con la plataforma*. Hemos definido una librería de integración que debe ser adherida a la aplicación evaluada. Para ello, se requieren conocimientos de programación y se demanda una cantidad de tiempo que depende de los eventos de interacción que puedan generarse con la aplicación. El primer paso para la mejora de este tiempo es el desarrollo de herramientas que realicen la integración de un modo automático. Siendo un mejor avance el no requerir ningún tipo de integración.

- *Interfaces adaptativas*. La adaptación de interfaces dentro de los entornos de movilidad no es un elemento novedoso, como se aborda en el trabajo realizado por Castillejo et al. [Castillejo+14], la variabilidad del contexto requiere una atención extra. Detectamos un fuerte potencial en la aplicación de la base de conocimiento y la detección automática de errores de interacción implementada en la adaptación automática de interfaces para ofrecer nuevos métodos de adaptación basados en el conocimiento colectivo de la experiencia previa adquirida.
- *Estudio de variables de contexto y recomendaciones de evaluación mediante guías*. La base de conocimiento que se ha generado para la experimentación y validación del uso de la misma está en una fase muy temprana. Si alimentamos dicha base con un mayor número de evaluaciones se podrán inferir casos posibles con características concretas cuya probabilidad de encontrar errores de interacción sea alta en un número significativo de pruebas, pudiendo generar con ello guías y recomendaciones formales de evaluación. Además, en el caso de esta tesis se ha realizado una agrupación de los casos posibles basada en el género de los usuarios y el entorno etiquetado. Sin embargo, no se han probado otras combinaciones de variables de contexto que permitan una mejor agrupación. El estudio de las variables de contexto y su efecto en la usabilidad de un modo más explícito presenta una gran oportunidad ya que a lo largo de la literatura relacionada ha quedado demostrada su importancia en este tipo de dispositivos.
- *Expansión del dominio de aplicación*. Aunque la metodología presentada se ha centrado en aplicaciones móviles independientemente de la plataforma, no ha sido el caso en el desarrollo de ciertas herramientas de la plataforma de soporte, como la herramienta del usuario de pruebas, que en esta tesis se han realizado para la plataforma Android. Además, el desarrollo de la plataforma de soporte se ha centrado en aplicaciones nativas, dejando de lado las

aplicaciones móviles web. Consideramos interesante la expansión de la plataforma de soporte a otro tipo de plataformas móviles y sobre todo a la evaluación de aplicaciones de tipo web con dispositivos móviles.

6.5. CONSIDERACIONES FINALES

Esta tesis es el resultado de años de trabajo desarrollando una metodología que permita el estudio de la usabilidad de aplicaciones móviles y su relación con el contexto. Cuando comencé en esta área todavía no era consciente de todo lo que implicaba desarrollar una tesis doctoral donde veía, permitiéndome la analogía, una minúscula ración de comida dentro de un plato minimalista. Después de comenzar a desarrollar este trabajo y al terminar el mismo, me sentí profundamente identificado con las palabras de Alberto Chicote[45] criticando los platos minimalistas de Ferran Adrià[46]:

> *"Detrás de una sencillez insultante hay mucho conocimiento"*

Esta idea es la que más ha penetrado en mi mente dentro de mis estudios en este campo, junto con la suma importancia con la que influye el trabajo de otros investigadores: en forma de crítica o golpe de realidad, pero sobretodo motivación e inspiración. Por ello, intento agradecer el trabajo de la comunidad científica ofreciendo este trabajo, que espero sea fuente de inspiración o aprendizaje con la que otros miembros de la comunidad puedan ver algo de conocimiento dentro de esta pequeña porción de comida.

---

[45]Alberto Chicote es un cocinero, chef y restaurador famoso por mezclar la cocina tradicional con las nuevas tecnologías y ser el pionero de lo que se bautizó como cocina fusión en España, que consiste en aplicar técnicas y productos foráneos a la cocina española.

[46]Ferran Adrià Acosta es un cocinero español considerado por muchos como el mejor chef del mundo, destacado por su minimalismo en la presentación de sus platos.

Referencias bibliográficas

[Albert+13] Albert, W., & Tullis, T. (2013). *Measuring the user experience: collecting, analyzing, and presenting usability metrics*. Newnes.

[Alshehri+12] Alshehri, F. & Freeman, M. (2012). Methods for usability evaluations of mobile devices. In *J. W. Lamp (Eds.), 23rd Australian Conference on Information Systems* (pp. 1-10). Geelong: Deakin University.

[Al-Ismail+14] Al-Ismail, M., & Sajeev, A. S. M. (2014, November). Usability challenges in mobile web. In *Communication, Networks and Satellite (COMNETSAT), 2014 IEEE International Conference on* (pp. 50-55). IEEE.

[Avouris+08] Avouris, N., Fiotakis, G., & Raptis, D. (2008). *On measuring usability of mobile applications. In International Workshop on* (p. 38).

[Baharuddin+13] Baharuddin, R., Singh, D., & Razali, R. (2013). Usability dimensions for mobile applications—A review. *Res. J. Appl. Sci. Eng. Technol, 5*, 2225-2231.

[Balbo95] Balbo, S. (1995). Automatic evaluation of user interface usability: Dream or reality. In *Proceedings of the Queensland Computer-Human Interaction Symposium* (Vol. 7).

[Barnum10] Barnum, C. M. (2010). *Usability testing essentials: ready, set... test!*. Elsevier.

[Bernhaupt+08] Bernhaupt, R., Mihalic, K., & Obrist, M. (2008). Usability evaluation methods for mobile applications. *Handbook Res User Interface Des Evaluation for Mobile Technology*, 44, 745-758.

[Bertini+06] Bertini, E., Gabrielli, S., & Kimani, S. (2006). Appropriating and assessing heuristics for mobile computing. In *Proceedings of the working conference on Advanced visual interfaces* (pp. 119-126). ACM.

[Bevan+91] Bevan, N., Kirakowsky, J., & Maissel, J. (1991). What is Usability?. In *Proceedings of the 4th International Conference on Human Computer Interaction*. Elsevier.

[Bevan+94] Bevan, N., & Macleod, M. (1994). Usability measurement in context. *Behaviour & information technology, 13*(1-2), 132-145.

[Bias94] Bias, R. G. (1994). The pluralistic usability walkthrough: coordinated empathies. In *Usability inspection methods* (pp. 63-76). John Wiley & Sons, Inc..

[Bickmore+97] Bickmore, T. W., & Schilit, B. N. (1997). Digestor: device-independent access to the World Wide Web. *Computer Networks and ISDN Systems, 29*(8), 1075-1082.

[Biel+10] Biel, B., Grill, T., & Gruhn, V. (2010). Exploring the benefits of the combination of a software architecture analysis and a usability evaluation of a mobile application. *Journal of Systems and Software, 83*(11), 2031-2044.

[Billi+10] Billi, M., Burzagli, L., Catarci, T., Santucci, G., Bertini, E., Gabbanini, F., & Palchetti, E. (2010). A unified methodology for the evaluation of accessibility and usability of mobile applications. *Universal Access in the Information Society, 9*(4), 337-356.

[Boulos+11] Boulos, M. N., Wheeler, S., Tavares, C., & Jones, R. (2011). How smartphones are changing the face of mobile and participatory healthcare: an overview, with example from eCAALYX. *Biomedical engineering online, 10*(1), 24.

[Bowman+02] Bowman, D. A., Gabbard, J. L., & Hix, D. (2002). *A survey of usability evaluation in virtual environments: classification and comparison of methods. Presence: Teleoperators and Virtual Environments, 11*(4), 404-424.

[Bradley+05] Bradley, N. A., & Dunlop, M. D. (2005). Toward a multidisciplinary model of context to support context-aware computing. *Human-Computer Interaction, 20*(4), 403-446.

[Brooke96] Brooke, J. (1996). SUS-A quick and dirty usability scale. *Usability evaluation in industry, 189*(194), 4-7.

[Brown+97] Brown, P. J., Bovey, J. D., & Chen, X. (1997). Context-aware applications: from the laboratory to the marketplace. *Personal Communications, IEEE, 4*(5), 58-64.

[Brown+10] Brown, A., Yesilada, Y., Jay, C., Harper, S., & Chen, A. Q. (2010). The blind leading the blind: Web accessibility research leading mobile Web usability. *Mobile Web, 2*, 71-93.

[Brown+11] Brown, B., Reeves, S., & Sherwood, S. (2011). Into the wild: challenges and opportunities for field trial methods. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 1657-1666). ACM.

[Card+83] Card, S. K., Newell, A., & Moran, T. P. (1983). The psychology of human-computer interaction.

[Card+91] Card, S. K., Robertson, G. G., & Mackinlay, J. D. (1991, April). The information visualizer, an information workspace. In *Proceedings of the SIGCHI Conference on Human factors in computing systems* (pp. 181-186). ACM.

[Carta+11] Carta, T., Paternò, F., & Santana, V. (2011). Support for remote usability evaluation of web mobile applications. In *Proceedings of the 29th ACM international conference on Design of communication* (pp. 129-136). ACM.

[Cassady+02] Cassady, J. C., & Johnson, R. E. (2002). Cognitive test anxiety and academic performance. *Contemporary Educational Psychology, 27*(2), 270-295.

[Castillejo+14] Castillejo, E., Almeida, A., & López-de-Ipiña, D. (2014). Ontology-Based Model for Supporting Dynamic and Adaptive User Interfaces. *International Journal of Human-Computer Interaction, 30*(10), 771-786.

[Chambers+92] Chambers, J. M., Freeny, A., & Heiberger, R. M. (1992). Analysis of variance; designed experiments. *Statistical Models in S*, 145-193.

[Chin+88] Chin, J. P., Diehl, V. A., & Norman, K. L. (1988). Development of an instrument measuring user satisfaction of the human-computer interface. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 213-218). ACM.

[Cisco14] Cisco, C. V. N. I. (2014). Global mobile data traffic forecast update, 2013–2018. *white paper*.

[Cohen88] Cohen, J. (1988). Statistical Power Analysis for the Behavioral Sciences, 2nd edR Erlbaum. *New Jersey*.

[Cohen92] Cohen, J. (1992). A power primer. *Psychological bulletin, 112*(1), 155.

[Coutaz95] Coutaz, J. (1995). Evaluation techniques: Exploring the intersection of HCI and software engineering. In *Software Engineering and Human-Computer Interaction* (pp. 35-48). Springer Berlin Heidelberg.

[Cuddihy+05] Cuddihy, E., Wei, C., Barrick, J., Maust, B., Bartell, A. L., & Spyridakis, J. H. (2005). Methods for assessing web design through the internet. In *CHI'05 Extended Abstracts on Human Factors in Computing Systems* (pp. 1316-1319). ACM.

[Coursaris+06] Coursaris, C., & Kim, D. (2006). A qualitative review of empirical mobile usability studies. *AMCIS 2006 Proceedings*, 352.

[Coursaris+11] Coursaris, C. K., & Kim, D. J. (2011). A meta-analytical review of empirical mobile usability studies. *Journal of usability studies, 6*(3), 117-171.

[Cyr+05] Cyr, D., & Bonanni, C. (2005). Gender and website design in e-business. *International Journal of Electronic Business, 3*(6), 565-582.

[Davis+89] Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, 319-340.

[Dearman+05] Dearman, D., Hawkey, K., & Inkpen, K. M. (2005). Rendezvousing with location-aware devices: Enhancing social coordination. *Interacting with computers, 17*(5), 542-566.

[deSa+08] de Sá, M., & Carriço, L. (2008). Lessons from early stages design of mobile applications. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services* (pp. 127-136). ACM.

[Dey01] Dey, A. K. (2001). Understanding and using context. *Personal and ubiquitous computing, 5*(1), 4-7.

[Djamasbi+07] Djamasbi, S., Tullis, T., Hsu, J., Mazuera, E., Osberg, K., & Bosch, J. (2007). Gender preferences in web design: usability testing through eye tracking. In *Proceedings of the Thirteenth Americas Conference on Information Systems (AMCIS)* (p. 1).

[Doll+94] Doll, W. J., Xia, W., & Torkzadeh, G. (1994). A confirmatory factor analysis of the end-user computing satisfaction instrument. *Mis Quarterly*, 453-461.

[Duh+06] Duh, H. B. L., Tan, G. C., & Chen, V. H. H. (2006). Usability evaluation for mobile device: a comparison of laboratory and field tests. In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services* (pp. 181-186). ACM.

[Dumas+99] Dumas, J. S., & Redish, J. (1999). *A practical guide to usability testing*. Intellect Books.

[Dumas+08] Dumas, J. S., & Loring, B. A. (2008). *Moderating usability tests: Principles and practices for interacting*. Morgan Kaufmann.

[Dumas+99] Dumas, J. S., & Redish, J. (1999). *A practical guide to usability testing*. Intellect Books.

[Ericsson15] Ericsson, A. B. (2015). Ericsson mobility report: On the pulse of the Networked Society.

[Floria00] Floría Cortés, A. (2000). Recopilación de métodos de usabilidad.

[Folmer+04] Folmer, E., & Bosch, J. (2004). Architecting for usability: a survey. *Journal of systems and software, 70*(1), 61-78.

[Froehlich+07] Froehlich, J., Chen, M. Y., Consolvo, S., Harrison, B., & Landay, J. A. (2007). MyExperience: a system for in situ tracing and capturing of user feedback on mobile phones. In *Proceedings of the 5th international conference on Mobile systems, applications and services* (pp. 57-70). ACM.

[Granollers04] Granollers i Saltiveri, T. (2004). MPIu+ a. Una metodología que integra la Ingeniería del Software, la Interacción Persona-Ordenador y la Accesibilidad en el contexto de equipos de desarrollo multidisciplinares.

[Guest+06] Guest, G., Bunce, A., & Johnson, L. (2006). How many interviews are enough? An experiment with data saturation and variability. *Field methods, 18*(1), 59-82.

[Harrison+13] Harrison, R., Flood, D., & Duce, D. (2013). Usability of mobile applications: literature review and rationale for a new usability model. *Journal of Interaction Science, 1*(1), 1-16.

[Hewett+92] Hewett, T. T., Baecker, R., Card, S., Carey, T., Gasen, J., Mantei, M., Perlman, G., Strong, G., & Verplank, W. (1992). *ACM SIGCHI curricula for human-computer interaction.* ACM.

[Hilbert+00] Hilbert, D. M., & Redmiles, D. F. (2000). Extracting usability information from user interface events. *ACM Computing Surveys (CSUR), 32*(4), 384-421.

[Holtzblatt+93] Holtzblatt, K., & Jones, S. (1993). Contextual inquiry: A participatory technique for system design. *Participatory design: Principles and practices,* 177-210.

[Holzinger05] Holzinger, A. (2005). Usability engineering methods for software developers. *Communications of the ACM, 48*(1), 71-74.

[Hom98] Hom, J. (1998). The usability methods toolbox handbook.

[Hong+01a] Hong, J. I., Heer, J., Waterson, S., & Landay, J. A. (2001). WebQuilt: A proxy-based approach to remote web usability testing. *ACM Transactions on Information Systems, 19*(3), 263-285.

[Hong+01b] Hong, J. I., Li, F. C., Lin, J., & Landay, J. A. (2001). End-user perceptions of formal and informal representations of web sites. In *CHI'01 Extended Abstracts on Human Factors in Computing Systems* (pp. 385-386). ACM.

[Hong+11] Hong, S., & Kim, S. C. (2011). Mobile web usability: developing guidelines for mobile web via smart phones. In *Design, User Experience, and Usability. Theory, Methods, Tools and Practice* (pp. 564-572). Springer Berlin Heidelberg.

[Hwang+07] Hwang, W., & Salvendy, G. (2007). What makes evaluators to find more usability problems?: a meta-analysis for individual detection rates. In *Human-Computer Interaction. Interaction Design and Usability* (pp. 499-507). Springer Berlin Heidelberg.

[Hwang+09] Hwang, W., & Salvendy, G. (2009). Integration of usability evaluation studies via a novel meta-analytic approach: what are significant attributes for effective evaluation?. *Intl. Journal of Human–Computer Interaction, 25*(4), 282-306.

[Hwang+10] Hwang, W., & Salvendy, G. (2010). Number of people required for usability evaluation: the 10±2 rule. *Communications of the ACM, 53*(5), 130-133.

[Inostroza+12] Inostroza, R., Rusu, C., Roncagliolo, S., Jimenez, C., & Rusu, V. (2012). Usability heuristics for touchscreen-based mobile devices. In *Information Technology: New Generations (ITNG), 2012 Ninth International Conference on* (pp. 662-667). IEEE.

[ISO98] ISO. (1998). *ISO 9241-11:1998(E): Ergonomic requirements for office work with visual display terminals (VDTs) Part 11: Guidance on usability*.

[ISO99a] ISO. (1999). *ISO 13407:1999 Human-centred design processes for interactive systems.*

[ISO99b] ISO. (1999). *ISO/IEC 14598-1:1999 Information technology -- Software product evaluation -- Part 1: General overview.*

[ISO01] ISO. (2001). *ISO/IEC 9126-1:2001 Software engineering -- Product quality -- Part 1: Quality model.*

[ISO04] ISO. (2004). *ISO/IEC TR 9126-4:2004(E): Software engineering — Product quality — Part 4: Quality in use metrics).*

[ISO14] ISO. (2014). *ISO/IEC 25000:2014 Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE.*

[Isomäki+11] Isomäki, H., & Pekkola, S. (2011). Reframing humans in information systems development. Springer.

[Ivory01] Ivory, M. Y. (2001). *An empirical foundation for automated web interface evaluation* (Doctoral dissertation, University of California, Berkeley).

[Ivory03] Ivory, M. Y. (2003). *Automated Web site evaluation: researchers' and practitioners' perspectives* (Vol. 4). Springer Science & Business Media.

[Jääskeläinen10] Jääskeläinen, R. (2010). Think-aloud protocol. *Handbook of translation studies, 1*, 371-373.

[Jambon+09] Jambon, F., & Meillon, B. (2009). User experience evaluation in the wild. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems* (pp. 4069-4074). ACM.

[Jensen09] Jensen, K. L. (2009). RECON: capturing mobile and ubiquitous interaction in real contexts. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services* (p. 76). ACM.

[Jensen12] Jensen, K. L. (2012). Remote and autonomous studies of mobile and ubiquitous applications in real contexts. *Developments in Technologies for Human-Centric Mobile Computing and Applications*, 79.

[Jimenez+02] Jiménez, J. M., Herrera, A. S., & Rojas, W. S. (2002). Fuentes de varianza e índices de varianza explicada en las ciencias del movimiento humano. *Pensar en Movimiento: Revista de Ciencias del Ejercicio y la Salud, 2*(2), 70-74.

[Jones+99] Jones, M., Marsden, G., Mohd-Nasir, N., Boone, K., & Buchanan, G. (1999). Improving Web interaction on small displays. *Computer Networks, 31*(11), 1129-1137.

[Jordan+99] Jordan, B., & Henderson, A. (1995). Interaction analysis: Foundations and practice. *The journal of the learning sciences, 4*(1), 39-103.

[Kaasalainen10] Kaasalainen, J. P. (2010). Designing for Mobility. M*obile Web, 2*, 1-32.

[Kankainen02] Kankainen, A. (2002). *Thinking model and tools for understanding user experience related to information appliance product concepts*. Helsinki University of Technology.

[Katayama+95] Katayama, T., Furukawa, Z., & Ushijima, K. (1995, December). Event interactions graph for test-case generations of concurrent programs. In *Software Engineering Conference, 1995. Proceedings., 1995 Asia Pacific* (pp. 29-37). IEEE.

[Kellar+04] Kellar, M., Inkpen, K., Dearman, D., Hawkey, K., Ha, V., MacInnes, J., Nunes, M. N., Parker, K., Reilly, D., Rodgers, M., & Whalen, T. (2004). *Evaluation of Mobile Collaboration: Learning from our Mistakes*. Technical Report 2004-13, Dalhousie University.

[Kim+01] Kim, L., & Albers, M. J. (2001). Web design issues when searching for information in a small screen display. In *Proceedings of the 19th annual international conference on Computer documentation* (pp. 193-200). ACM.

[Kim+02] Kim, H., Kim, J., Lee, Y., Chae, M., & Choi, Y. (2002). An empirical study of the use contexts and usability problems in mobile Internet. In *System Sciences, 2002. HICSS. Proceedings of the 35th Annual Hawaii International Conference on* (pp. 1767-1776). IEEE.

[Kirakowski+93] Kirakowski, J., & Corbett, M. (1993). SUMI: The software usability measurement inventory. *British journal of educational technology, 24*(3), 210-212.

[Kitzinger95] Kitzinger, J. (1995). Qualitative research. Introducing focus groups. *BMJ: British medical journal, 311*(7000), 299.

[Kjeldskov+04] Kjeldskov, J., Skov, M. B., Als, B. S., & Høegh, R. T. (2004). Is it worth the hassle? Exploring the added value of evaluating the usability of context-aware mobile systems in the field. In *Mobile Human-Computer Interaction-MobileHCI 2004* (pp. 61-73). Springer Berlin Heidelberg.

[Kjeldskov+12] Kjeldskov, J., & Paay, J. (2012, September). A longitudinal review of Mobile HCI research methods. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services* (pp. 69-78). ACM.

[Kjeldskov+14] Kjeldskov, J., & Skov, M. B. (2014). Was it worth the hassle?: ten years of mobile HCI research discussions on lab and field evaluations. In Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services (pp. 43-52). ACM.

[Kondratova+06] Kondratova, I., Lumsden, J., & Langton, N. (2006). Multimodal field data entry: performance and usability issues.

[Korhonen+06] Korhonen, H., & Koivisto, E. M. (2006). Playability heuristics for mobile games. In *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services* (pp. 9-16). ACM.

[Kristjánsdóttir+11] Kristjánsdóttir, Ó. B., Fors, E. A., Eide, E., Finset, A., van Dulmen, S., Wigers, S. H., & Eide, H. (2011). Written online situational feedback via mobile phone to support self-management of chronic widespread pain: a usability study of a Web-based intervention. *BMC musculoskeletal disorders, 12*(1), 51.

[Krug05] Krug, S. (2005). *Don't make me think: A common sense approach to web usability.* Pearson Education India.

[Landay+01] Landay, J., & Myers, B. (2001). Sketching interfaces: Toward more human interface design. *Computer, 34*(3), 56-64.

[Lederer+92] Lederer, A. L., & Prasad, J. (1992). Nine management guidelines for better cost estimating. *Communications of the ACM, 35*(2), 51-59.

[Leitner+07] Leitner, G., Ahlström, D., & Hitz, M. (2007). *Usability of Mobile Computing in Emergency Response Systems–Lessons Learned and Future Directions* (pp. 241-254). Springer Berlin Heidelberg.

[Lettner+12] Lettner, F., & Holzmann, C. (2012). Automated and unsupervised user interaction logging as basis for usability evaluation of mobile applications. In *Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia* (pp. 118-127). ACM.

[Levene60] Levene, H. (1960). Robust tests for equality of variances1. *Contributions to probability and statistics: Essays in honor of Harold Hotelling, 2*, 278-292.

[Lewis82] Lewis, J. R. (1982, October). Testing small system customer set-up. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* (Vol. 26, No. 8, pp. 718-720). SAGE Publications.

[Lewis94] Lewis, J. R. (1994). Sample sizes for usability studies: Additional considerations. *Human Factors: The Journal of the Human Factors and Ergonomics Society, 36*(2), 368-378

[Lewis01] Lewis, J. R. (2001). Evaluation of procedures for adjusting problem-discovery rates estimated from small samples. *International Journal of Human-Computer Interaction, 13*(4), 445-479.

[Lewis12] Lewis, J. R. (2012). Usability testing. *Handbook of Human Factors and Ergonomics*, 1267-1312.

[Liberman99] Liberman, K. (1999). From walkabout to meditation: Craft and ethics in field inquiry. *Qualitative Inquiry, 5*(1), 47-63.

[Likert32] Likert, R. (1932). A technique for the measurement of attitudes. *Archives of psychology*.

[Lindgaard+07] Lindgaard, G., & Chattratichart, J. (2007, April). Usability testing: what have we overlooked?. En *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 1415-1424). ACM.

[Longoria01] Longoria, R. (2001). Designing mobile applications: challenges, methodologies, and lessons learned. *Usability evaluation and interface design: Cognitive engineering, intelligent agents and virtual reality*, 91-95.

[Lumsden+07] Lumsden, J., Kondratova, I., & Durling, S. (2007). Investigating microphone efficacy for facilitation of mobile speech-based data entry. In *Proceedings of the 21st British HCI Group Annual Conference on People and Computers: HCI... but not as we know it-Volume 1* (pp. 89-97). British Computer Society.

[Lund08] Lund, A. M. (2001). Measuring Usability with the USE Questionnaire. *STC Usability SIG Newsletter*.

[Macaulay12] Macaulay, L. A. (2012). *Requirements engineering*. Springer Science & Business Media.

[Mackay95] Mackay, W. E. (1995). Ethics, lies and videotape…. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 138-145). ACM Press/Addison-Wesley Publishing Co..

[Macleod+93] Macleod, M., & Rengger, R. (1993). The development of DRUM: A software tool for video-assisted usability evaluation. *People and Computers*, 293-293.

[Macleod+97] Macleod, M., Bowden, R., Bevan, N., & Curson, I. (1997). The MUSiC performance measurement method. *Behaviour & Information Technology, 16*(4-5), 279-293.

[Maguire01] Maguire, M. (2001). Context of use within usability activities. *International Journal of Human-Computer Studies, 55*(4), 453-483.

[Mayhew99] Mayhew, D. J. (1999, May). The usability engineering lifecycle. In *CHI'99 Extended Abstracts on Human Factors in Computing Systems* (pp. 147-148). ACM.

[Memon+01] Memon, A. M., Pollack, M. E., & Soffa, M. L. (2001). Hierarchical GUI test case generation using automated planning. *Software Engineering, IEEE Transactions on*, 27(2), 144-155.

[Miller68] Miller, R. B. (1968). Response time in man-computer conversational transactions. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I* (pp. 267-277). ACM.

[Molich+03] Molich, R., & Jeffries, R. (2003). Comparative expert reviews. In *CHI'03 Extended Abstracts on Human Factors in Computing Systems* (pp. 1060-1061). ACM.

[Mulder+05] Mulder, I., Ter Hofte, G. H., & Kort, J. (2005). SocioXensor: Measuring user behaviour and user eXperience in conteXt with mobile devices. In *Proceedings of Measuring Behavior* (pp. 355-358).

[Nayebi+12] Nayebi, F., Desharnais, J. M., & Abran, A. (2012). The state of the art of mobile application usability evaluation. In *CCECE* (pp. 1-4).

[Nielsen+90] Nielsen, J., & Molich, R. (1990, March). Heuristic evaluation of user interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 249-256). ACM.

[Nielsen91] Nielsen, J. (1993). Response times: The 3 important limits. *Usability Engineering*.

[Nielsen92] Nielsen, J. (1992). Finding usability problems through heuristic evaluation. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 373-380). ACM.

[Nielsen94a] Nielsen, J. (1994). *Usability engineering*. Elsevier.

[Nielsen94b] Nielsen, J. (1994). Usability laboratories. *Behaviour & Information Technology, 13*(1-2), 3-8.

[Nielsen95a] Nielsen, J. (1995). 10 usability heuristics for user interface design. *Fremont: Nielsen Norman Group*.

[Nielsen95b] Nielsen, J. (1995). Severity ratings for usability problems. *Papers and Essays*, 54.

[Nielsen03] Nielsen, J. (2003). Usability 101: Introduction to usability.

[Nielsen+06] Nielsen, C. M., Overgaard, M., Pedersen, M. B., Stage, J., & Stenild, S. (2006). It's worth the hassle!: the added value of evaluating the usability of mobile systems in the field. In *Proceedings of the 4th Nordic conference on Human-computer interaction: changing roles* (pp. 272-280). ACM.

[Nielsen+13] Nielsen, J., & Budiu, R. (2013). *Mobile usability*. MITP-Verlags GmbH & Co. KG.

[NIST07] National Institute of Standards and Technology. (2007). *Common Industry Specification for Usability – Requirements (NISTIR 7432)*.

[Norman+06] Norman, K. L., & Panizzi, E. (2006). Levels of automation and user participation in usability testing. *Interacting with computers, 18*(2), 246-264.

[Oulasvirta+05] Oulasvirta, A., Tamminen, S., Roto, V., & Kuorelahti, J. (2005). Interaction in 4-second bursts: the fragmented nature of attentional resources in mobile HCI. In P*roceedings of the SIGCHI conference on Human factors in computing systems* (pp. 919-928). ACM.

[Oulasvirta12] Oulasvirta, A. (2012). Rethinking experimental designs for field evaluations. *IEEE Pervasive Computing,* (4), 60-67.

[Pendell+12] Pendell, K. D., & Bowman, M. S. (2012). Usability Study of a Library's Mobile Website: An Example from Portland State University.

[Perallos07] Perallos Ruiz, A. (2007). *Metodología ágil y adaptable al contexto para la evaluación integral y sistemática de la calidad de los sitios web* (Doctoral dissertation, Universidad de Deusto, Bilbao).

[Petersen+10] Petersen, J. E. L. M. K., & Zandi, R. H. N. (2010). Observing the Context of Use of a Media Player on Mobile Phones using Embedded and Virtual Sensors. *Observing the mobile user experience*, 33.

[Preece+94] Preece, J., Rogers, Y., Sharp, H., Benyon, D., Holland, S., & Carey, T. (1994). *Human-computer interaction*. Addison-Wesley Longman Ltd.

[Pretel+14] Pretel, I., & Lago, A. B. (2014). Evaluación remota de aplicaciones móviles híbridas: nueva aproximación en entornos reales. In *Actas de la 9a Conferencia Ibérica de Sistemas y Tecnologías de Informacion* (Vol. 1, pp. 383–388). Barcelona, Spain

[Quesenbery04] Quesenbery, W. (2004). Balancing the 5Es of Usability. *Cutter IT Journal, 17*(2), 4-11.

[Raento+05] Raento, M., Oulasvirta, A., Petit, R., & Toivonen, H. (2005). ContextPhone: A prototyping platform for context-aware mobile applications. *Pervasive Computing*, IEEE, 4(2), 51-59.

[Raento+09] Raento, M., Oulasvirta, A., & Eagle, N. (2009). Smartphones an emerging tool for social scientists. *Sociological methods & research, 37*(3), 426-454.

[Razak+10] Razak, F. H. A., Hafit, H., Sedi, N., Zubaidi, N. A., & Haron, H. (2010). Usability testing with children: Laboratory vs field studies. In *User Science and Engineering (i-USEr), 2010 International Conference on* (pp. 104-109). IEEE.

[Rettig94] Rettig, M. (1994). Prototyping for tiny fingers. *Communications of the ACM, 37*(4), 21-27.

[Rieman+95] Rieman, J., Franzke, M., & Redmiles, D. (1995). Usability evaluation with the cognitive walkthrough. In *Conference companion on Human factors in computing systems* (pp. 387-388). ACM.

[Rogers+07a] Rogers, Y., Sharp, H., Preece, J., & Tepper, M. (2007). Interaction design: beyond human-computer interaction. *netWorker: The Craft of Network Computing, 11*(4), 34.

[Rogers+07b] Rogers, Y., Connelly, K., Tedesco, L., Hazlewood, W., Kurtz, A., Hall, R. E., Hursey, J., & Toscos, T. (2007). *Why it's worth the hassle: The value of in-situ studies when designing ubicomp* (pp. 336-353). Springer Berlin Heidelberg.

[Rosson+02] Rosson, M. B., & Carroll, J. M. (2002). *Usability engineering: scenario-based development of human-computer interaction*. Morgan Kaufmann.

[Roto06] Roto, V. (2006). *Web browsing on mobile phones: Characteristics of user experience*. Helsinki University of Technology.

[Roto+11] Roto, V., Väätäjä, H., Jumisko-Pyykkö, S., & Väänänen-Vainio-Mattila, K. (2011). Best practices for capturing context in user experience studies in the wild. In *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments* (pp. 91-98). ACM.

[Rowley94] Rowley, D. E. (1994). Usability testing in the field: bringing the laboratory to the user. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 252-257). ACM.

[Rubin+08] Rubin, J., & Chisnell, D. (2008). *Handbook of Usability Testing: Howto Plan, Design, and Conduct Effective Tests* (2nd ed.). Wiley Publishing.

[Rudd+96] Rudd, J., Stern, K., & Isensee, S. (1996). Low vs. high-fidelity prototyping debate. *interactions, 3*(1), 76-85.

[Ryan+05] Ryan, C., & Gonsalves, A. (2005). The effect of context and application type on mobile usability: an empirical study. In *Proceedings of the Twenty-eighth Australasian conference on Computer Science-Volume 38* (pp. 115-124). Australian Computer Society, Inc..

[Sanz12] Sanz Urquijo, B. (2012). Rubicon: un nuevo enfoque para la seguridad en las aplicaciones de smartphones.

[Savio+07] Savio, N., & Braiterman, J. (2007). Design sketch: The context of mobile interaction. In *Mobile HCI* (pp. 284-286).

[Schilit+94] Schilit, B. N., & Theimer, M. M. (1994). Disseminating active map information to mobile hosts. *Network, IEEE, 8*(5), 22-32.

[Shackel91] Shackel, B. (1991). Usability-context, framework, definition, design and evaluation. *Human factors for informatics usability*, 21-37.

[Shrestha07] Shrestha, S. (2007). Mobile web browsing: usability study. In *Proceedings of the 4th international conference on mobile technology, applications, and systems and the 1st international symposium on Computer human interaction in mobile technology* (pp. 187-194). ACM.

[Simon00] Simon, S. J. (2000). The impact of culture and gender on web sites: an empirical study. *ACM SIGMIS Database, 32*(1), 18-37.

[Snyder03] Snyder, C. (2003). *Paper prototyping: The fast and easy way to design and refine user interfaces*. Morgan Kaufmann.

[Sova+08] Sova, D. H., & Nielsen, J. (2008). *233 Tips and Tricks for Recruiting Users as Participants in Usability Studies*. Nielsen Norman Group.

[Symonds11] Symonds, E. (2011). A practical application of SurveyMonkey as a remote usability-testing tool. *Library Hi Tech, 29*(3), 436-445.

[Team12] Team, R. C. (2012). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria, 2012.

[Thomas+96] Thomas, C., & Bevan, N. (1996). Usability context analysis: a practical guide.

[Tsai06] Tsai, P. (2006). A survey of empirical usability evaluation methods. *GSLIS Independent Study*.

[Tsiaousis+08] Tsiaousis, A. S., & Giaglis, G. M. (2008). Evaluating the effects of the environmental context-of-use on mobile website usability. In Mobile Business, 2008. ICMB'08. 7th International Conference on (pp. 314-322). IEEE.

[Tsiaousis+10] Tsiaousis, A. S., & Giaglis, G. M. (2010). An empirical assessment of environmental factors that influence the usability of a mobile website. In *Mobile Business and 2010 Ninth Global Mobility Roundtable (ICMB-GMR), 2010 Ninth International Conference on* (pp. 161-167). IEEE.

[Van89] van Harmelen, M. (1989). Exploratory user interface design using scenarios and prototypes. In *People and computers V: proceedings of the fifth conference of the British Computer Society Human-Computer Interaction Specialist Group, University of Nottingham* (p. 191).

[Virzi+96] Virzi, R. A., Sokolov, J. L., & Karis, D. (1996). Usability problem identification using both low-and high-fidelity prototypes. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 236-243). ACM.

[Vokar+01] Vokar, S., Mariage, C., & Vanderdonckt, J. (2001). Log files analysis to measure the utility of an intranet. In *Proceedings of 9th Int. Conf. on Human-Computer Interaction HCI International'2001, Lawrence Erlbaum Associates, Mahwah, 2001* (vol. 1 pp. 1185-1189).

[Walker+02] Walker, M., Takayama, L., & Landay, J. A. (2002). High-fidelity or low-fidelity, paper or computer? Choosing attributes when testing web prototypes. In Proceedings of the Human Factors and Ergonomics Society Annual Meeting (Vol. 46, No. 5, pp. 661-665). SAGE Publications.

[Westerman+11] Westerman, W. C., Lamiraux, H., & Dreisbach, M. E. (2011). *U.S. Patent No. 8,059,101.* Washington, DC: U.S. Patent and Trademark Office.

[Whiteside+87] Whiteside, J., & Wixon, D. (1987). The dialectic of usability engineering. In *Human-computer Interaction--INTERACT'87: proceedings of the Second IFIP Conference on Human-Computer Interaction, held at the University of Stuttgart, Federal Republic of Germany, 1-4 September, 1987* (Vol. 2, p. 17). Elsevier Science Ltd.

[Wigelius+09] Wigelius, H., & Väätäjä, H. (2009). Dimensions of context affecting user experience in mobile work. In *Human-Computer Interaction–INTERACT 2009* (pp. 604-617). Springer Berlin Heidelberg.

[Wilson+01] Wilson, C., & Coyne, K. P. (2001). The whiteboard: Tracking usability issues: to bug or not to bug?. *interactions, 8*(3), 15-19.

[Wixon+94] Wixon, D., Jones, S., Tse, L., & Casaday, G. (1994). Inspections and design reviews: framework, history and reflection. In *Usability inspection methods* (pp. 77-103). John Wiley & Sons, Inc..

[Yañez+14] Yáñez Gómez, R., Cascado Caballero, D., & Sevillano, J. L. (2014). Heuristic Evaluation on Mobile Interfaces: A New Checklist. *The Scientific World Journal, 2014*.

[Yuan+10] Yuan, X., & Memon, A. M. (2010). Generating event sequence-based test cases using GUI runtime state feedback. *Software Engineering, IEEE Transactions on, 36*(1), 81-95.

[Zhang+05] Zhang, D., & Adipat, B. (2005). Challenges, methodologies, and issues in the usability testing of mobile applications. *International Journal of Human-Computer Interaction, 18*(3), 293-308.

[Zheng+06] Zheng, W., & Yuan, Y. (2006). Identifying the differences between stationary office support and mobile work support: a conceptual framework. *International Journal of Mobile Communications, 5*(1), 107-122.

Esta tesis fue terminada de escribir en Bilbao el 7 de octubre de 2015