

Sistema de evaluación de la efectividad del usuario sensible al contexto para aplicaciones móviles

Authors Name/s per 1st Affiliation (Author)

line 1 (of Affiliation): dept. name of organization

line 2: name of organization, acronyms acceptable

line 3: City, Country

line 4: e-mail address if desired

Authors Name/s per 2nd Affiliation (Author)

line 1 (of Affiliation): dept. name of organization

line 2: name of organization, acronyms acceptable

line 3: City, Country

line 4: e-mail address if desired

Abstract—El número de dispositivos móviles con conectividad a Internet aumenta cada año de forma exponencial. Además, estos dispositivos son cada vez más importantes tanto en contextos particulares como corporativos. En el caso de la empresa, ofrecen un canal de comunicación perfecto para interactuar con potenciales clientes o con los mismos empleados. Por ello, es muy importante sacarle el máximo partido ya que un buen diseño puede influir directamente en los beneficios de la empresa. Dependiendo del contexto, la eficiencia del usuario a la hora de realizar una tarea con el dispositivo puede variar drásticamente. Por ello, es vital la medición de la eficiencia del usuario respecto al contexto que le rodea. En este artículo se presenta un sistema centrado en móviles donde se hace un modelo de contexto e interacción para una posterior captura de los mismos y así medir la eficacia del usuario sin añadir desviaciones en el resultado.

Keywords—HCI; servicios móviles, evaluación; context-aware; efectividad; calidad

I. INTRODUCCIÓN

En los últimos años, los dispositivos móviles están ganando cada vez más importancia en la realización de tareas tanto personales y domésticas como de empresa. Hace muchos años que las empresas están expandiendo los límites de sus modelos de negocio a la red pero en los últimos años, gracias a la masificación de los terminales Smartphones, ha aumentado progresivamente el número de servicios conectados a través de los terminales móviles. Además, los nuevos modelos de negocio que han crecido a lo largo de este nuevo ecosistema de aplicaciones, como son Google Play¹ y Apple App Store² han permitido que mediante el desarrollo de una pequeña y sencilla aplicación los usuarios puedan acceder a ella e instalársela en un tiempo récord.

Por otro lado, las aplicaciones han evolucionado drásticamente a entornos cada vez más complicados, comenzando por aplicaciones de escritorio donde no había conectividad y el entorno no variaba drásticamente, ya que los ordenadores de sobremesa estaban en interiores. Posteriormente, han evolucionado a entornos de conectividad pero seguían siendo en su mayoría ordenadores de sobremesa y portátiles, que en su mayoría seguían estando en interiores. Finalmente, las aplicaciones han evolucionado a entornos

móviles donde la mayor parte del tiempo los usuarios hacen uso de aplicaciones con su terminal Smartphone. Sin embargo, las herramientas de evaluación software han evolucionado centrándose en el dominio web y no toman en cuenta el nuevo dominio de los dispositivos móviles. Según el Índice de Networking de Cisco Visual pasado [1], el uso del este tipo de dispositivos se ha triplicado en el año 2011 y la cantidad media de tráfico en 2011 fue de 150 MB al mes cada Smartphone, frente a los 55 MB por mes en 2010. Debido a este justificado y preocupante crecimiento, las herramientas de evaluación de software deben centrarse más en los nuevos modelos que crean las nuevas aplicaciones móviles.

El principal problema de las aplicaciones móviles es que los contextos en los que la aplicación se usa difieren entre ellos de un modo extraordinario y además, los diferentes entornos en los que la aplicación va a usarse son impredecibles. Debido a esto, la calidad de la experiencia de usuario y la efectividad de las tareas realizadas con las aplicaciones móviles pueden variar dependiendo del contexto. Por consiguiente, la captura y medición de la efectividad junto con su modelo de contexto sin realizar modificaciones en el mismo es extremadamente importante.

En este artículo se expone una herramienta de evaluación automática de la eficacia basada en la captura de métricas del contexto. Dicha aplicación está formada por una pequeña librería de captura para terminales Android que simplifica a los desarrolladores de aplicaciones la captura de las diferentes interacciones y el modelo de contexto. Esta pequeña librería facilita el envío de la información capturada a un servidor web, que forma parte de la herramienta, para almacenar toda la información de forma remota. También se muestra la implementación de una versión preliminar del sistema que captura un subconjunto de variables del modelo de contexto. Además, se realizó la evaluación del prototipo mediante la implementación de un pequeño juego. Gracias a los resultados obtenidos se demuestra que esta herramienta puede almacenar la información necesaria para la evaluación de la eficacia, sin reducir el rendimiento del dispositivo utilizado y por lo tanto, sin modificar el modelo de contexto.

El artículo se estructura como se detalla a continuación. En la siguiente sección se estudian las principales definiciones de eficacia, calidad, y calidad en uso basado en los principales estándares de calidad. En la Sección 3, los diferentes tipos de

¹ Google Play. <https://play.google.com>

² Apple App Store. <http://itunes.apple.com/es/browse/>

definiciones de modelos de contexto son explicados, se proporciona una nueva definición del mismo basado en entornos móviles y se estudia la forma de capturarlo, siempre analizando los métodos de captura ya existentes. En la Sección 4, se presenta la implementación del prototipo desarrollado. En la Sección 5, se expone el experimento y se analizan los resultados obtenidos con el mismo. Por último, se describen las conclusiones obtenidas con este trabajo y se discuten las posibles mejoras y trabajos futuros.

II. CALIDAD SOFTWARE

A. Calidad en Uso

Dentro de la calidad software, se han estudiado los diferentes tipos de calidad, calidad en uso y eficacia, así como las diferentes conexiones entre ellas.

En primer lugar, según el estándar ISO 9241-11 [2], la usabilidad es el grado en que un producto puede ser utilizado por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso específico. Dentro de este estándar se destaca la efectividad como el nivel de precisión y exhaustividad con la que los usuarios alcanzan objetivos específicos. En el estándar ISO 9126-4 [3] se utiliza la misma definición cuando se concreta la eficacia, pero no se proporciona una regla general compuesta por atributos medibles. De hecho, la responsabilidad de desarrollar las métricas adecuadas recae en los expertos en calidad software. De acuerdo con la norma ISO / IEC 9126 se define la calidad representa una propiedad del producto de software que se define en términos de un conjunto de atributos interdependientes (como la usabilidad, la seguridad, la fiabilidad, el rendimiento, la complejidad, legibilidad, reutilización) expresado en diferentes niveles de detalle donde es tenido en cuenta el contexto en que se usa.

Los diferentes atributos pueden medir tres principales tipos de calidad: calidad interna, calidad externa y calidad en uso:

- La calidad interna es el conjunto de atributos del software desde un punto de vista interno (recursos consumidos, escalabilidad...). Se mide y es mejorado durante la implementación del código, la revisión y las pruebas.
- La calidad externa es aquella que se muestra cuando el software se está ejecutando en términos de su comportamiento, como son el número de respuestas no esperadas. Se mide y se evalúa durante las pruebas de software.
- La calidad en uso es la calidad del software que el usuario puede percibir cuando el software se utiliza en un contexto explícito. Mide el grado en el que los usuarios pueden realizar sus tareas en un ambiente particular. Se mide en cuatro principales capacidades en un contexto de uso: la eficacia, productividad, seguridad y satisfacción.

La calidad en uso se puede descomponer en varias métricas que pueden medirse mediante el diseño y la realización de experimentos. Este trabajo se centra concretamente en la medición de una de esas capacidades: la eficacia dentro de la

calidad en el aspecto de uso. Esta capacidad a su vez es formada por de tres indicadores: Efectividad de tareas (TE), finalización de tareas (TC) y la frecuencia de errores (EF).

B. Métricas de efectividad

Estos indicadores miden el grado de exactitud y completitud con la que los objetivos pueden ser alcanzados. 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.

La eficacia de la tarea (1) mide la cantidad de los objetivos conseguidos por un usuario. Se mide sumando el número de errores (A_i) ocurridos durante la tarea. Algunos errores pueden ser más importantes que otros. En el ejemplo del mensaje en el terminal, una palabra mal escrita es menos importante que no escribirla. Para distinguir los diferentes errores, cada tipo de error tiene su propio peso asociado. El resultado de la métrica está entre 0 y 1, cuanto más cerca de 1, mejor.

$$TE = |1 - \sum A_i| \quad (1)$$

La completitud de la tarea (2) mide el nivel de éxito que el usuario consigue en la realización de tareas. A diferencia de la métrica anterior, esta medición supone que las tareas se pueden realizar sin la posibilidad de ser completadas parcialmente. En este caso, se calcula dividiendo el número de tareas completadas (TXM) por el número de intentos de realización de esa tarea (TA). El resultado de la métrica también está entre 0 y 1, cuanto más cerca de 1, mejor.

$$TX = TXM / TA. \quad (2)$$

Por último, la frecuencia de error (3) mide el número de veces que se comete un error en un plazo determinado. Se calcula dividiendo el número de errores (E) por el tiempo de la tarea o el número de pasos totales (T). Este indicador es muy útil para hacer comparaciones si los errores tienen la misma importancia. Al igual que las otras métricas, el resultado también está entre 0 y 1, pero en este caso cuanto más cerca de 1, peor.

$$EE = E / T. \quad (3)$$

Centrándonos en la calidad en uso, la eficacia no tiene en cuenta cómo los objetivos se han alcanzado, sólo la medida en la que se han logrado. La eficacia, define y forma parte de la usabilidad y la calidad en uso. Además, ésta puede estar sesgada por el contexto en uso (el usuario actual y su conocimiento, su dispositivo móvil, la conexión de red, el ambiente, etc.).

III. CAPTURA DEL CONTEXTO

A. Definición del modelo

A lo largo de las diferentes definiciones de la efectividad y calidad, el contexto de uso ha estado apareciendo como un elemento que puede alterar notablemente los resultados. El contexto de uso ha sido un elemento que se ha tenido en cuenta y se ha definido en numerosas ocasiones por una gran cantidad

de investigadores, expertos y comunidades. La norma ISO 9241-11 define el contexto en uso como cada usuario de la tarea, el equipo y el medio tanto físico como social que pueden afectar y ser afectados por la ejecución de la tarea. La mayoría de las definiciones estudiadas dan mucha importancia al ambiente que rodea usuario, pero no es el único factor como veremos a continuación.

Por un lado, varios estudios mantienen que un contexto es sólo la ubicación física [4]. La ubicación es uno de los aspectos que más cambia en la mayoría de los contextos, pero no es el único. También agregan a la ubicación más atributos, como es el tiempo meteorológico [5] para lograr una definición más precisa del entorno y el contexto físico. Por otra parte, otros estudios añaden al mismo la comunidad [6] y todos los agentes afectados por el sistema [7]. También mantienen que el contexto debe definirse respondiendo dónde está el usuario, quién es el usuario, y qué recursos están disponibles. Otros, como Nadav Savio [8] también definen el contexto enumerando más parámetros, tales como la cultura, la actividad, la atención, la interfaz, el dispositivo, la conexión...

Después de ver los diferentes modelos y definiciones de contexto se ha definido un modelo preliminar centrado en el campo móvil. Se concluye que los componentes que definen el contexto son: el usuario, el dispositivo móvil, la aplicación que ejecuta y el entorno (físico, ambiental, técnico y sociocultural).

El usuario está definido por cuatro grandes grupos de atributos: información personal, conocimientos, habilidades y actitudes. Atributos tales como la edad y el sexo, que ofrecen buenos criterios para clasificar a los usuarios, forman el grupo de información personal. Los conocimientos son aquellos atributos respecto al nivel de formación del usuario, como el nivel de idiomas o nivel de estudios. El grupo de habilidades se refiere a las capacidades y/o discapacidades físicas y mentales. El grupo de actitudes está formado por las motivaciones personales, la experiencia pasada y la esperada.

El dispositivo móvil está formado por seis principales grupos: conexiones, cuerpo, entradas, salidas, batería y características de software. El grupo de conexiones define las distintas interfaces de comunicación que dispone el dispositivo móvil, como son la conexión 3G o Wi-fi. El cuerpo del dispositivo define las dimensiones, color y peso del dispositivo. Dos de los elementos más importantes para definir a un dispositivo móvil son sus interfaces de salida (por ejemplo, la pantalla y los altavoces), así como las interfaces de entrada (por ejemplo, la pantalla táctil, teclado T9...). El grupo de la batería se describe la información acerca de la batería y su estado: si está cargándose, la tecnología de la misma... Por último, pero no menos importante, el grupo de atributos de características software cuenta con la información referente a la versión del sistema operativo (OS), su lenguaje, etc.

Por último, el entorno se define mediante cuatro grandes grupos: entorno físico, ambiental, técnico y sociocultural. El entorno físico define todos aquellos elementos tangibles que nos rodean. El grupo ambiental contiene todos los atributos que pueden describir las condiciones meteorológicas y del ambiente (ruido, la temperatura del medio ambiente, su humedad...). El grupo técnico mantiene todas las características técnicas excluyendo al dispositivo móvil: los

atributos de conectividad, hardware y características de software adicional, etc. El grupo sociocultural define los agentes culturales y sociales que pueden determinar la interacción (por ejemplo, los hábitos culturales o la religión).

Después de definir el modelo se estudiará la forma de capturarlo. Tenemos que tener en cuenta que el contexto puede estar sesgada en función de la forma en que utiliza para capturar la información.

B. Métodos de captura

Una vez que el contexto de uso ha sido definido el siguiente paso es diseñar el método de captura. El primer paso es estudiar los métodos existentes, elegir los más relevantes y definir en base a lo analizado el método más relevante para aplicaciones móviles.

Por un lado, los métodos basados en laboratorios (tales como Morae Observer [9]) capturan información en ambientes controlados con dispositivos y usuarios específicos. Las principales ventajas de estos métodos son que el medio es muy controlable y la recogida de datos, es barata y fácil. Sin embargo, no se puede emular con precisión las interacciones reales y los modelos de contexto en los que la aplicación va a ser usada.

Por otra parte, los métodos basados en entornos no controlados [10] [11] capturan la información en entornos donde realmente se ve el comportamiento de la aplicación. Se suelen utilizar cámaras añadidas al dispositivo u observadores humanos para capturar información. Este tipo de métodos al final tratan de llevar el laboratorio al entorno real, utilizando técnicas como encuestas y entrevistas que se mezclan con la captura de contexto.

Para capturar el modelo de contexto y las interacciones, es esencial capturar información objetiva, utilizando el dispositivo móvil en entornos reales. Este tipo de información no puede ser capturada con un método basado en laboratorio. En cambio los otros métodos, puede proporcionar información más objetiva, pero agentes tales como cámaras añadidas o métodos más invasivos de evaluación como las encuestas durante las tareas, han de ser eliminados. Por lo tanto, la mejor manera de capturar la interacción de datos y el modelo de contexto es mediante la captura de información a través de un dispositivo móvil, utilizando una pequeña aplicación de captura.

IV. SISTEMA DE CAPTURA

Si se desea implementar una herramienta de captura sin influir el contexto, hay que eliminar todos los elementos que influyen y utilizar solamente los elementos que componen el contexto (en este caso, el dispositivo móvil). Esta herramienta debe capturar el modelo de contexto a través de los sensores incorporados en el móvil. Aunque se ha definido un modelo de contexto detallado, la primera versión de la herramienta sólo puede capturar un pequeño subconjunto de los datos. Además, esta versión asume que todos los errores tienen la misma relevancia.

El sistema propuesto (ver Fig. 1) está formado por una pequeña librería de desarrollado para aplicaciones móviles Android y un servidor para almacenar y registrar la información capturada. Los usuarios se descargan la aplicación

a evaluar desde el servidor y mediante la misma se registran en el sistema. La aplicación de prueba debe tener integrada la biblioteca desarrollada. Esta biblioteca captura automáticamente la información de contexto e interacción a través de un módulo de captura de contexto y lo almacena en una base de datos local mediante el módulo de persistencia de datos. Cuando el dispositivo tiene conexión a Internet y su propietario lo desea, toda la información almacenada se transfiere al servidor.

Con el fin de capturar la información de modelo de contexto nuestra biblioteca permite a los desarrolladores abstraerse de la captura de las interacciones y el modelo de contexto. Sólo necesitan implementar sus propias aplicaciones e insertar pequeñas líneas de registro dentro de su lógica de aplicación para, de un modo sencillo, registrar los diferentes pasos que un usuario está realizando. Los desarrolladores pueden trabajar con la librería con sólo tres instrucciones básicas: *configure*, *send_local_info* y *log*.

La instrucción *configure* establecen los datos del capturador con los parámetros necesarios para su posterior ejecución: establece el nombre de usuario, contraseña e identificador de dispositivo. *Send_local_info* envía la información capturada y almacenada en la base de datos local al servidor principal. La información capturada es el timestamp de la interacción, el modelo de contexto definido y el objeto con el que el usuario interactúa. Esta instrucción sólo requiere tres parámetros: el tipo de evento, que se explicará más adelante, el identificador de la interfaz de usuario y el objeto (botones, cuadros de texto...) con el cual el usuario está interactuando.

Una tarea puede pasar a través de cuatro estados principales: Cuando una tarea no se ha iniciado todavía (*no_iniciada*), cuando una tarea se inicia y el usuario está interactuando para lograr el objetivo de la tarea (*iniciada*), cuando una tarea se ha iniciado, pero su usuario no está interactuando para lograrlo (*pausada*) y cuando la tarea está finalmente terminado (*finalizada*). Después de ver los principales estados, veremos eventos que pueden ser lanzados para cambiar el estado de una tarea.

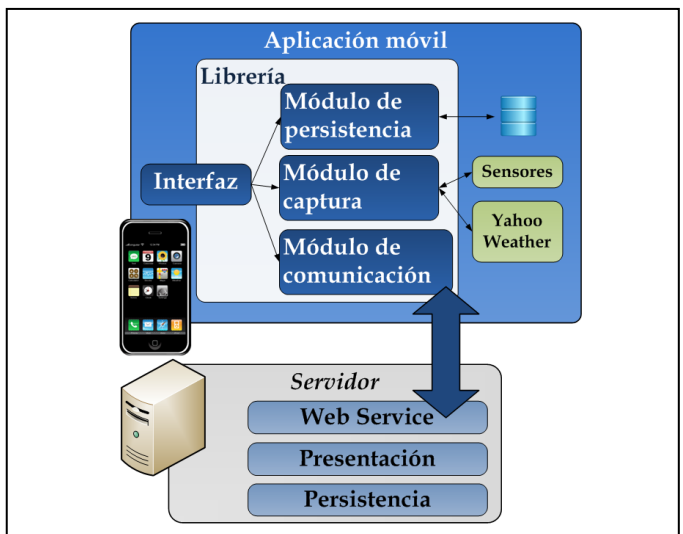


Figura 1. Arquitectura del sistema

Durante la realización de una tarea un usuario puede lanzar dos eventos principales: *START_TASK* (al comienzo de la tarea) y *END_TASK* (al final de la tarea). Por otra parte existen otros dos eventos: si el usuario sale de la tarea (por ejemplo, por una llamada de teléfono entrante) *PAUSE_TASK* se lanza. Cuando el usuario decide continuar con la tarea se produce el evento *RESUME_TASK*.

Cuando una tarea se inicia pueden lanzarse dos eventos relacionados con la interacción. El evento *INTERACTION* significa que el usuario está interactuando de la forma correcta. Este evento se activa cuando un usuario está avanzando en el ciclo de la tarea. El evento *ERROR* indica que un usuario ha cometido un error durante el proceso de interacción.

El modelo de contexto se captura mediante el uso de todos los sensores incorporados y las interfaces de programación de aplicaciones proporcionadas por Android. La información relacionada con el usuario se recupera durante el registro debido a que la frecuencia de cambio de esta información es muy baja. Se guardan el apodo, sexo, fecha de nacimiento, altura, peso, si el usuario es zurdo, diestro o ambidiestro y el nivel europeo de inglés, español, portugués, alemán y francés. La información relevante al dispositivo móvil se captura en dos fases: también durante el registro y durante la ejecución de tareas. Se almacenan el identificador de dispositivo, versión de Sistema Operativo, fabricante, país de procedencia, modelo e idioma configurado. Cuando el usuario está realizando las tareas la configuración relativa al volumen de sonido es capturada (volumen de alarma, timbre, modo de vibración, si hay auriculares conectados...). También se captura la información relacionada con la batería (nivel de tensión, temperatura, si la batería se está cargando o no...). Las propiedades de la pantalla también se capturan (densidad, altura, anchura, brillo). La información relacionada con el medio ambiente es capturada durante la ejecución de la tarea debido a la alta frecuencia de los cambios. En este grupo entran los niveles de ruido y la luz, que son capturados por los sensores. La ubicación y la información de conexión se capturan con los servicios internos de Android. Se puede capturar si el dispositivo está conectado a alguna red, el tipo de conexión, su operador, etc. Si el dispositivo está conectado a Internet las condiciones meteorológicas son capturadas mediante servicios web de Yahoo Weather (descripción de las condiciones meteorológicas, temperatura, humedad, velocidad del viento...).

Es importante mencionar que esta herramienta puede capturar información privada y personal. Por lo tanto, los desarrolladores que deseen utilizar la biblioteca deben pedir varios permisos de Android. Es muy importante ya que el instalador Android muestra todos los permisos de la aplicación necesita para funcionar correctamente y si el usuario acepta dichos permisos y la aplicación los tiene, significa que el usuario es completamente consciente de qué tipo de información está disponible para la misma.

El permiso de Internet se usa para abrir sockets de red, consultar el clima y cargar la información registrada en el servidor principal. También es necesario el permiso para acceder a la ubicación mediante fuentes de ubicación como GPS celdas y Wi-fi. Además, el permiso de grabar audio se utiliza para medir el nivel de ruido. Con el permiso de acceso al

estado de red nuestra herramienta tiene acceso a información detallada sobre las redes. Por último, el permiso de acceso a estado de redes inalámbricas se utiliza para recuperar información sobre redes Wi-fi.

El servidor está desarrollado con la tecnología de Google App Engine. Su principal objetivo es ser el principal nexo entre los usuarios y los experimentadores. De forma remota se almacenan todos los datos de interacción en la nube. Esto es posible gracias a varios servicios a través de una interfaz web. A través de estos servicios los usuarios pueden registrarse, iniciar interacciones y ver las interacciones ya registradas.

V. VALIDACIÓN DEL SISTEMA

La versión preliminar del sistema se validó realizando, en primer lugar, un experimento de medición de rendimiento. La única manera de obtener resultados sesgados es cambiar el comportamiento del dispositivo disminuyendo su rendimiento. Con el fin de asegurar que la reducción de dicho rendimiento es mínima se ha medido en dispositivos provistos de con la librería registrando en la base de datos local el modelo de contexto. Esta medición del rendimiento (ver Fig. 2) fue realizada utilizando varios dispositivos móviles con diferentes versiones de Android OS: HTC Desire (versión 2.2.2), HTC Wildfire (2.2.1), Desire HD (2.3.5), Desire Z (2.2), ZTE Skate (2.3.5), Xoom (4.2), Galaxy Nexus (4.2) y Nexus 10 (4.2).

Aunque se perciben grandes diferencias entre los modelos en cuanto a tiempo y variaciones, se concluye que el tiempo de almacenamiento es muy reducido y que mejora con el tiempo, como se aprecia en el resumen general del gráfico inferior. El tiempo más reducido (MIN) fue de 10 milisegundos y el más largo (MAX) fue de 349. Dentro de las muestras había una desviación típica media (SD) bastante estable (22.06 ms.) y una media (MEAN) muy aceptable 46.21 ms. En términos generales viendo estos resultados se puede afirmar que nuestro sistema puede almacenar más de 20 líneas de log por segundo. Por consiguiente, se demostró que es posible almacenar más interacciones por segundo de las que un usuario estándar necesita para interactuar con una aplicación móvil sin influir en el rendimiento del dispositivo.

Por otro lado, se hizo un experimento en entornos reales donde 4 sujetos se registraron desde la plataforma, se descargaron una aplicación que contiene la biblioteca de desarrollo y la utilizaron. La aplicación es un pequeño juego de memoria cuyas partidas se estuvieron registrando durante un día, generando cientos de líneas de interacción registradas. Jugaron en cuatro contextos diferentes: en el hogar (H), caminando por la calle (C), en un transporte público (P) y en el trabajo (T), concretamente en entorno de oficina donde la gente está trabajando con ordenadores y el nivel de ruido es bajo. El experimento se dividió en cuatro etapas principales. En primer lugar, todos los usuarios se tenían que descargar la aplicación desde el servidor con su propio dispositivo móvil. Después de descargarla se instalaba aceptando los permisos anteriormente explicados. Por último, el usuario se registra mediante la aplicación y se presiona el botón de jugar para iniciar el juego. Se selecciona el contexto y se inicia el juego.

La aplicación consiste en un juego de cartas en donde se reparten una serie de cartas boca abajo. En un turno, el jugador

voltea dos cartas. Si coinciden, el jugador las deja boca arriba. Si no coinciden, el jugador voltea de nuevo las cartas boca abajo. El objetivo de este juego es terminar con todas las cartas boca arriba en menos de 15 turnos.

Centrándose en la medición de la efectividad el objetivo de la tarea es terminar con todas las cartas boca arriba en menos de 15 turnos. Si las cartas volteadas en un turno no coinciden, se comete un error. Hay 8 pares de cartas, por lo que la partida perfecta se hace en 8 movimientos y ningún error. El número máximo de turnos es 15, por lo que si un usuario no lo consigue en 15, pierde y la tarea no queda completada.

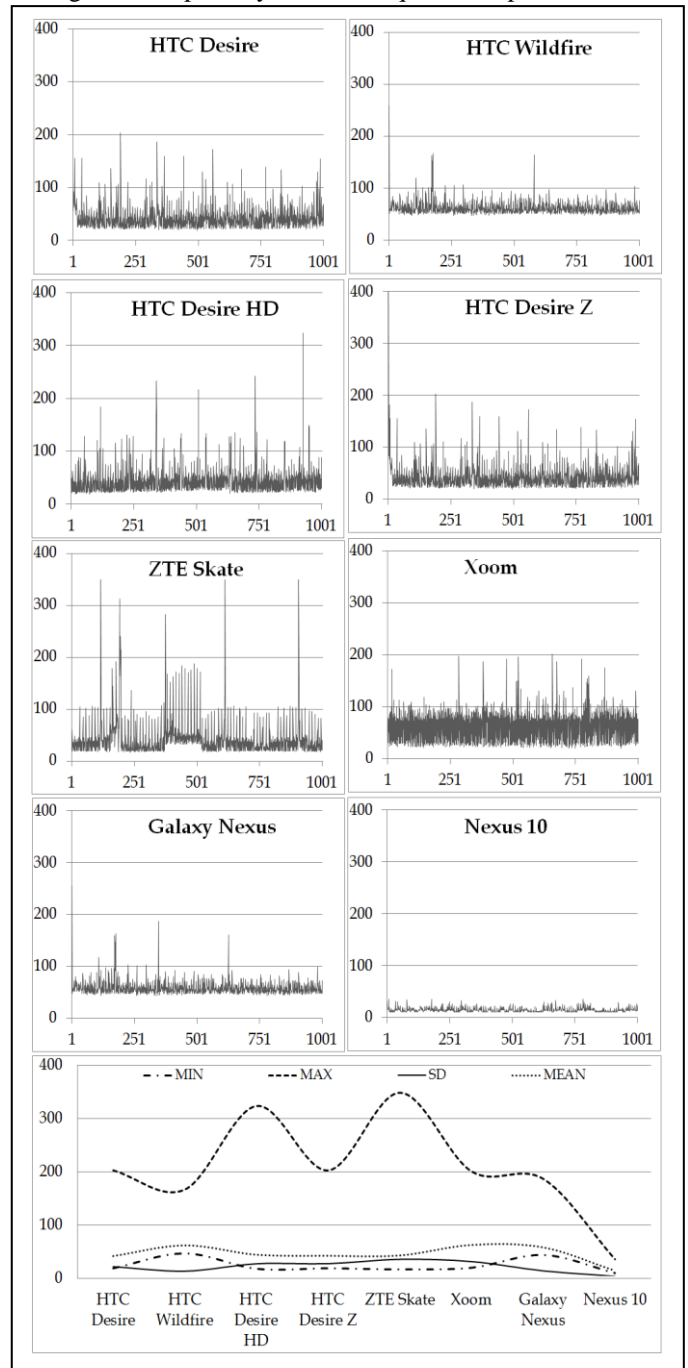


Figura 2. Resultados del experimento de rendimiento

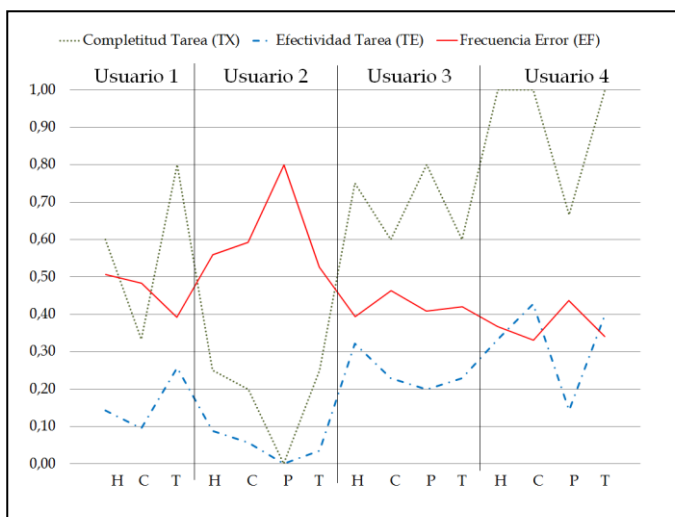


Figura 3. Resultados de la evaluación de la efectividad en los diferentes contextos

La efectividad de la tarea (TE) se calcula restando 1 al número de errores cometidos durante la tarea multiplicado por el peso del error. El peso de un error es $1/7$ debido a que el número máximo de errores que puede cometer es 6 con 8 interacciones correctas. Si se cometen 7 errores TE debe ser 0 ($1 - (7 * 1/7) = 0$). La completitud de la tarea (TX) se calcula contando todas las partidas ganadas por un usuario divididas por todas las partidas intentadas. La frecuencia de error (EF) se calcula dividiendo el número de errores por el número de partidas totales.

En el último gráfico (ver Fig. 3) se muestran los resultados obtenidos del experimento agrupado por usuarios y contexto. A través de éste se puede ver claramente que el peor contexto para ser eficientes es el de ir caminando por la calle (C), donde se registran las mayores frecuencias de error y tareas menos completadas. En cambio, en el trabajo se registran los mejores resultados.

A través de estas pruebas se demuestra que la eficacia depende del contexto y que se puede medir automáticamente a través de una herramienta software automática.

VI. CONCLUSIONES Y TRABAJO FUTURO

A través de este trabajo se han definido indicadores de eficacia, que pueden influir fácilmente en la evaluación de la calidad en uso en aplicaciones móviles. También hemos estudiado los modelos de contexto y hemos definido uno basada en entornos móviles. Además se ha definido un método de captura y medición que no altera los resultados,

concluyendo que la mejor manera de capturar la interacción y el contexto es mediante el propio terminal móvil.

También se ha mostrado un sistema de captura compuesto por una pequeña biblioteca Android para aplicaciones móviles que captura la información y un servidor para almacenar todos los datos. Además se ha llevado a cabo una evaluación preliminar del sistema mediante el desarrollo de un pequeño juego.

Gracias a este trabajo se prueba que la eficacia depende del modelo de contexto y que automáticamente se puede medir a través de una herramienta software. Además se ha concluido que se pueden capturar más de 20 interacciones por segundo.

El siguiente paso es estudiar todos los atributos capturados del modelo de contexto para realizar un análisis más detallado de las diferentes variables que varían el resultado de la eficacia. También se debe trabajar en el aumento del número de atributos capturados sin dejar desatendido el rendimiento del sistema, que es el principal agente que puede alterar los resultados.

REFERENCIAS

- [1] CISCO, "Cisco Visual Networking Index: Forecast and Methodology, 2011–2016." http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360.pdf, 2012, [Último acceso: 20 de febrero de 2013].
- [2] "ISO 9241-11:1998(E). Ergonomic requirements for office work with visual display terminals (VDTs) Part 11: Guidance on usability", 1998.
- [3] "ISO/IEC TR 9126-4:2004(E). Software engineering — Product quality — Part 4: Quality in use metrics", 2004.
- [4] B.N Schilit and M.M. Theimer, "Disseminating active map information to mobile hosts.", Network IEEE, vol.8, no.5, pp. 22-32, 1994.
- [5] P.J. Brown, J.D. Bovey and Xian Chen, "Context-aware applications: from the laboratory to the marketplace", Personal Communications IEEE, vol.4, no.5, pp.58-64, 1997.
- [6] A. Kankainen, "Thinking model and tools for understanding user experience related for information appliance product concepts.", Dissertation of Anu Kankainen, 2002.
- [7] "NISTIR7432. Information Access Division – Information Technology Laboratory, Common Industry Specification for Usability – Requirements", 2007.
- [8] N. Savio and J. Braiterman, "Design sketch: The context of mobile interaction." Proceedings of MobileHCI 2007, pp. 284-286, 2007.
- [9] TechSmith, "Morae usability testing software from TechSmith", <http://www.techsmith.com/morae.html> 2012, [Último acceso: 20 de febrero de 2013].
- [10] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith and P. Steggles, "Towards a Better Understanding of Context and Context-Awareness.", Proceedings HUC '99, Hans-Werner Gellersen (Ed.). Springer-Verlag, London, UK, UK, pp. 304-307. 1999.
- [11] K.L. Jensen, "RECON: capturing mobile and ubiquitous interaction in real contexts.", Proceedings MobileHCI '09. ACM, New York, NY, USA, 2009.