



**Pontificia Universidad Católica de Valparaíso**

Facultad de Ingeniería

Escuela de Ingeniería Informática

Ingeniería Civil en Informática

**APOYO A LAS EVALUACIONES DE USABILIDAD  
DE APLICACIONES PARA DISPOSITIVOS  
MÓVILES**

Autor:

Cristian Mauricio Urbina Guerra

Informe final del Proyecto para optar al Título profesional de  
Ingeniero Civil en Informática

Profesor Guía:

Alexandru Cristian Rusu

**DICIEMBRE 2007**

**Agradecimientos**

*Agradezco a mi familia, amigos y compañeros  
por su apoyo y fe.*

**Resumen**

Cuando hablamos de usabilidad en aplicaciones para móviles, nos referimos a facilidad y satisfacción con que un usuario puede usar dicho software bajo circunstancias específicas. Esto es de vital importancia para asegurar el éxito de aplicaciones para dispositivos móviles. Algunos métodos de evaluación de usabilidad pueden ser agilizados si se automatizan algunos de sus procesos, como la captura de información relevante durante una prueba y el cálculo de algunas estadísticas.

En este trabajo se investigó sobre el concepto de usabilidad, los métodos de evaluación de usabilidad más conocidos y sobre las técnicas que se están ocupando en las evaluaciones para aplicaciones de dispositivos móviles. Posteriormente, se identificaron restricciones y problemas que presentaban estas técnicas y se presenta una solución software, la cual fue desarrollada pasando por tres prototipos.

El último cumple con la mayoría de los requerimientos definidos durante el trabajo y permite un real apoyo en la captura automática de información y evaluación de dicha información para las evaluaciones de usabilidad de aplicaciones para dispositivos móviles.

## Abstract

When we talk about usability in applications for mobile devices, we refer to the satisfaction and the easiness that an user can get by using said software under certain circumstances. This is of vital importance to secure the success of applications for mobile devices. Some evaluation methods of usability may be sped up if some of the processes are automated, such as the capture of relevant information during a test, and the calculation of some statistics.

The concept of usability is investigated in this project, along with the more known evaluation methods, and the techniques that are being used in evaluations for mobile devices. Subsequently, restrictions and issues that these techniques presented were identified, and a software solution is presented, which was developed going through three prototypes.

The last prototype fulfills with most of the requirements defined during the project and allows a real support in the automatic capture of information. It also allows assessing said information for evaluations in usability of applications for mobile devices.

# Capítulo 1

## Introducción

La historia en los últimos 25 años ha mostrado grandes cambios en relación a las tecnologías. A comienzos de los 80, el primer PC (Computador Personal) salió al mercado haciendo posible que oficinas, casas y hasta salas de clases pudiesen contar con un computador para poder realizar ciertas tareas. A principios de los 90, la *World Wide Web*, que gracias a Internet, revolucionó la disponibilidad y la entrega de información. Ahora una nueva ola nos está enfrentando. Los dispositivos móviles están llegando a ser bastante pequeños permitiendo ser transportados sin ningún problema y tener incluso, la posibilidad de poder interactuar con dispositivos implantados en el ambiente, pero su gran relevancia es que está permitiendo que las personas puedan estar conectadas con la información en cualquier momento y lugar.

El mercado de estos dispositivos está cambiando muy rápidamente. Cada vez se añaden más aplicaciones y servicios a estos: juegos, cámaras fotográficas, navegar por Internet, servicios basados en la localización, etc. Funcionalidades que hacen que ésta tecnología sea más compleja cada día. Esto, unido al hecho que cada vez son más pequeños, que tienen un número limitado de controles y que son usados generalmente con una mano, hace que sea más su uso, tanto los dispositivos como las aplicaciones que se desarrollan para estos. ¿Cómo hacer que un pequeño dispositivo, que tiene gran potencial de entregar múltiples y variadas funciones sea fácil de usar?

La usabilidad se encarga de este tema, dando las técnicas para poder medir, en pocas palabras, que tan fácil de usar puede ser un producto. El diseño para estas terminales es difícil, pero tienen un gran potencial que, para ser explotado, requiere de una investigación mucho más cuidadosa. Como señaló Jakob Nielsen en una entrevista para el diario online Elpais.com, ‘los servicios o aplicaciones para este tipo de dispositivos que no tengan un alto grado de usabilidad fracasarán’ [N4].

Es por todo esto, que la Ingeniería de la Usabilidad implantada en el desarrollo, tanto del aparato o terminal, como de la interfaz de navegación y de las aplicaciones, es un elemento crítico para asegurar que los productos sean sencillos de aprender a utilizar, para nuevos usuarios, eficaces, para usuarios experimentados, y de recordar, cuando no se tiene mucho contacto con el producto.

Actualmente se logra encontrar muchos documentos que logran explicar claramente el concepto de usabilidad y sus métodos de evaluación, siendo esta muy variada, pero, en la mayoría de los casos, destinada al desarrollo de aplicaciones web. Como por ejemplo Nielsen, quien ha escrito varios libros dedicados a las evaluaciones heurísticas, pruebas de usuario y otros métodos de evaluaciones de la usabilidad, da soluciones que se pueden utilizar durante el ciclo de vida a la usabilidad.

Con respecto a documentos que ayuden en la integración de la usabilidad en el desarrollo de aplicaciones para dispositivos móviles, ésta, es muy escasa. Nokia, una de las empresas más importantes en el mercado de los teléfonos celulares y que ha integrado la usabilidad dentro de su cultura, tiene dedica en su página web, una sección especial sobre este tema. Acá se puede obtener pequeños documentos que ayudan en el desarrollo de aplicaciones destinadas a sus productos. Además, Anne Kaikkonen, Titti Kalli y Mihael Cankar, entre otros, presentaron un estudio comparativo sobre las pruebas de usabilidad en aplicaciones móviles dentro de un laboratorio versus realizarlas fuera, donde no se intenta de desechar una de las alternativas, sino, solo analizar las desventajas y ventajas que tiene cada técnica y poder, de esta manera, optar por la más indicada para una aplicación determinada.

Con este trabajo se intenta comprender a grandes rasgos los factores claves de la usabilidad en el proceso de desarrollo de aplicaciones móviles para, posteriormente, poder identificar algún punto problemático o situación mejorable dentro de las evaluaciones de usabilidad de estas aplicaciones y finalmente presentar una solución o mejora de lo antes señalado desarrollando un o más prototipos del mismo el cual será probado en una prueba real, para poder determinar las potencialidades de las funciones del prototipo y, de esta manera, apoyar a las evaluaciones de usabilidad para aplicaciones de dispositivos móviles.

## Capítulo 2

### Descripción de Objetivos

#### **2.1 Objetivo general**

El objetivo general de este proyecto es apoyar las pruebas de usabilidad de aplicaciones para dispositivos móviles a través de un prototipo de herramienta software para ello.

Se estableció dicho objetivo general dada la importancia que los dispositivos móviles están tomando en la vida de las personas y ya que es necesario que estos cuenten con aplicaciones que sean sencillas de utilizar. Para conseguir esto, cae la necesidad de una herramienta que apoye de manera concreta a las evaluaciones de usabilidad de software para estos dispositivos.

#### **2.2 Objetivos específicos**

1. Conocer el concepto de usabilidad y su aplicación en el proceso de desarrollo de aplicaciones para dispositivos móviles.
2. Conocer las características de los dispositivos móviles que dificulten la usabilidad de ellos y de las aplicaciones que contienen.
3. Identificar problemas de las evaluaciones de usabilidad de aplicaciones para dispositivos móviles.
4. Definir los requerimientos que la herramienta debe cumplir para poder apoyar las evaluaciones de usabilidad de aplicaciones para dispositivos móviles.
5. Comprobar la utilidad del prototipo desarrollado en el apoyo de las evaluaciones de usabilidad.

# Capítulo 3

## Metodología

El Proyecto contempla los siguientes métodos para alcanzar los objetivos:

- **Estudio Teórico sobre el estado del arte.**

En esta etapa se planteo analizar la literatura disponible para poder alcanzar una mejor comprensión del concepto de usabilidad y el dominio en el cual se desenvuelve. Dicho estudio fue realizado con respecto a:

- Estudio del concepto de usabilidad y sus complementos.
- Estudio de las técnicas de evaluación de usabilidad de sistemas de software más comunes en la actualidad.
- Estudio de técnicas de evaluación de la usabilidad utilizadas en el desarrollo de aplicaciones para dispositivos móviles.

- **Identificación de problemas o puntos de mejora.**

Con los conocimientos adquiridos en la etapa anterior se plantea la identificación de problemas o puntos de mejora que se pueden realizar en las evaluaciones de usabilidad de aplicaciones para dispositivos móviles.

De esta manera se puede presentar una solución software para poder mejorar dichos problemas y los limites del dominio sobre el cual se va a desenvolver el trabajo en su completitud.

- **Implementación de la solución.**

Dentro de la implementación se plantea llevar a cabo el análisis, el modelado y el desarrollo de la solución a tal punto, para que esta alcance los objetivos planteados.

- **Análisis de resultados.**

En el Análisis de resultados se plantea la realización de pruebas de utilidad para verificar si la solución desarrollada realmente apoya a las evaluaciones de aplicaciones para dispositivos móviles.

# Capítulo 4

## Plan de trabajo

### 1. Desarrollo del marco teórico (Marzo de 2007 – Abril de 2007)

1.1. Investigación del concepto de usabilidad y la necesidad de su implementación en el desarrollo de aplicaciones software.

1.2. Análisis de los métodos más comunes ocupados para la evaluación de usabilidad.

1.3. Investigación de métodos específicos para la evaluación de usabilidad de aplicaciones para dispositivos móviles.

1.4. Investigación de las características de los dispositivos móviles que afectan negativamente a la usabilidad de aplicaciones móviles.

1.5. Investigación de otras herramientas que ayuden en el proceso de evaluación de usabilidad de aplicaciones para terminales móviles.

### 2. Presentación de propuesta (Abril de 2007 – Junio de 2007)

2.1. Presentación de una propuesta de solución para resolver los problemas encontrados o mejorar herramientas ya existentes.

2.2. Análisis de los requerimientos de la herramienta solución según los conocimientos adquiridos.

### 3. Desarrollo del primer prototipo (Agosto de 2007 – Octubre de 2007)

3.1. Refinación de la propuesta de solución presentada en la primera parte.

3.2. Modelado de la primera versión del prototipo.

- 3.3. Desarrollo del prototipo.
- 3.4. Validación de la primera versión del prototipo a nivel de utilidad.
- 4. Desarrollo del segundo prototipo(Octubre de 2007 – Diciembre de 2007)
  - 4.1. Refinación del modelo.
  - 4.2. Desarrollo de la segunda versión del prototipo.
  - 4.3. Validación de la segunda versión del prototipo tanto a nivel de funcionalidad como de utilidad.
- 5. Desarrollo del tercer prototipo(Marzo de 2007 – Mayo de 2007)
  - 5.1. Refinación del modelo.
  - 5.2. Desarrollo de la tercera versión del prototipo.
  - 5.3. Validación de la tercera versión del prototipo tanto a nivel de funcionalidad como de utilidad.

## Capítulo 5

### Dispositivos móviles

Los dispositivos móviles son aparatos de pequeño tamaño, con algunas capacidades de procesamiento con conexión permanente o intermitente a una red, con memoria limitada, diseñados específicamente para una función, pero que pueden llevar a cabo otras funciones más generales.

Para comprender mejor lo crítico que puede llegar a ser la usabilidad en aplicaciones para este tipo de dispositivos, se presentan los diferentes tipos de terminales móviles que se pueden encontrar en el mercado en la actualidad. La mayoría de estos dispositivos están siendo cada día más accesibles para el consumidor, puesto que sus precios han bajado muy rápidamente así como también aparecen rápidamente nuevos dispositivos y nuevas tecnologías que son incorporadas a estos dispositivos.

#### **5.1 Teléfonos celulares**

Fue inventado en 1947 por la empresa norteamericana AT&T, pero no se hizo portátil de manera práctica hasta 1983 cuando Motorola culmina el proyecto DynaTAC 8000X, el que es presentado oficialmente en 1984. El DynaTAC 8000X pesaba cerca de 1 kg, tenía un tamaño de 13 x 1,75 x 3,5 pulgadas, y rendía una hora de comunicación y ocho horas en stand-by, con pantalla de LED.

El teléfono celular consiste en un dispositivo de comunicación electrónico con las mismas capacidades básicas de un teléfono de línea telefónica convencional. Además de ser portátil es inalámbrico al no requerir cables conductores para su conexión a la red telefónica.

La red de telefonía móvil o celular consiste en un sistema telefónico en el que mediante la combinación de una red de estaciones transmisoras-receptoras de radio (estaciones base) y una serie de centrales telefónicas de conmutación, se posibilita la comunicación entre

terminales telefónicos portátiles (teléfonos móviles) o entre terminales portátiles y teléfonos de la red fija tradicional.

## **5.2 Personal Digital Assistant (PDA)**

PDA, del inglés Personal Digital Assistant (Ayudante personal digital), es un computador de mano originalmente diseñado como agenda electrónica (calendario, lista de contactos, bloc de notas y menos) con un sistema de reconocimiento de escritura. Hoy día se puede usar como una computadora doméstica (ver películas, crear documentos, juegos, correo electrónico, navegar por Internet, etc.). En la actualidad las PDA tienen características como:

**Movilidad:** Es un dispositivo de pequeño tamaño que puede ser transportado y utilizado en todo momento.

**Potencia de cómputo:** Los procesadores que incorporan los PDA de última generación son incluso más potentes que los de los ordenadores personales de hace sólo algunos años. Esto hace que los PDA actuales puedan abordar con suficiente rapidez tareas muy complejas, con las limitaciones lógicas del tamaño de pantalla y de los procedimientos de entrada de datos.

**Posibilidades de expansión:** Los PDA actuales poseen ranuras de expansión tipo Compactflash o MMC-SD, que permiten añadir al dispositivo una gran cantidad de ampliaciones, desde ampliaciones de memoria hasta tarjetas de comunicaciones inalámbricas, receptores GPS e incluso cámaras fotográficas y módulos GPRS.

**Conectividad:** Los PDA, aparte de las comunicaciones tradicionales vía serie, USB e infrarrojos, poseen conexiones inalámbricas, como Bluetooth o Wi-Fi.

## **5.3 Smartphones**

También conocidos como teléfonos inteligentes son dispositivos híbridos entre las PDA y los teléfonos celulares, rescatando las ventajas de ambos dispositivos. Ofrecen sustanciales ventajas, no sólo porque evitan la necesidad de transportar varios dispositivos portátiles,

sino porque ofrecen una integración real de aplicaciones de voz y datos en un mismo dispositivo.

El futuro de un smartphone es convertirse en un pc de mano, integrando todo lo que los ordenadores de sobremesa y portátiles ofrecen y todo lo que los aparatos de mano ya integran. Es por ello que cada vez tienen mejores procesadores, sistemas operativos más potentes e incluso ya están empezando a desarrollarse computadores del tamaño de un smartphone hasta el punto en el que se conviertan en una misma cosa, una unidad de comunicaciones, trabajo y ocio portátil.

## **5.4 Jukebox**

Son dispositivos orientados fundamentalmente al almacenamiento y reproducción de archivos de audio, video y fotografías, aunque algunos de ellos disponen también de interfaz de juegos.

Los formatos más comúnmente aceptados por estos dispositivos son:

- Audio: MP3, WMA, WAV, OGG, AAC.
- Video: MPEG-1, MPEG-2 y MPEG-4 (Divx, Xvid,...).

## **5.5 Dispositivos GPS**

Los dispositivos portátiles de localización han evolucionado considerablemente. Hoy en día, los dispositivos de localización basados en GPS (Global Position System), se pueden dividir en tres tipos fundamentales:

- *Dispositivos GPS dedicados: Se trata de pequeños dispositivos de mano dedicados únicamente a la localización GPS.*
- *PDA's con funciones adicionales de GPS: PDA's que añaden a las funcionalidades propias del PDA una antena receptor GPS, así como software de posicionamiento en mapas y gestión de rutas.*

- *Ampliaciones GPS para PDA: Se trata de módulos externos que, o bien se conectan directamente a un PDA a través de la ranura Compactflash, o bien vía Bluetooth.*

## Capítulo 6

### Usabilidad

En este proyecto se trabajó con la definición que da la *International Organization for Standardization* dentro su estándar titulado *Ergonomics of Human System Interaction* (ISO 9241), al concepto de usabilidad. En el define a la usabilidad como:

"La medida con la que un producto puede ser utilizado por determinados usuarios para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso específico."

En esta definición se puede identificar cuatro factores claves de la usabilidad:

- La efectividad, la cual se refiere a la precisión con que el usuario logra realizar un objetivo específico.
- La eficiencia, que es la cualidad tiene el software para poder cumplir estos objetivos sin malgastar recursos y realizándolo de buena manera.
- La satisfacción del usuario, que se refiere a la actitud positiva del usuario frente al producto gracias a la ausencia de incomodidad.
- El contexto de uso, el cual hace referencia al ambiente que se produce la interacción entre el usuario y el sistema.

La usabilidad no es el único factor importante en el éxito de un sistema. Obviamente su contenido y los servicios y funcionalidades disponibles contribuyen a este. Sin embargo, ante dos aplicaciones que ofrezcan servicios similares, los usuarios optarán por aquel que sea más fácil de aprender, eficiente en su uso, efectivo en los resultados y satisfactorio en la experiencia. Es más, muchas veces se puede ver casos donde es elegido una aplicación que

sea fácil de usar frente a otros que entregan una gran gama de funcionalidades, pero que al momento de utilizar, llegan a ser un gran desafío.

La usabilidad no solo se preocupa de la interfaz gráfica, es decir, de cómo está estructurada, los colores que se ocupan, el tamaño de la letra, el diseño de los iconos, imágenes, etc. La usabilidad va más allá. Busca que la interfaz sea un fiel rostro de lo que el sistema le ofrece al usuario, de cómo ayuda ésta al usuario al momento de realizar sus tareas, de la documentación para el usuario y del proceso de instalación.

La usabilidad hace referencia a la rapidez y facilidad con que los usuarios son capaces de realizar sus tareas utilizando un determinado sistema sin la necesidad de tener que recurrir periódicamente al manual de usuario de dicha aplicación. Con ello podemos desprender que si un sistema de información es fácil de usar y eficiente en su interacción, este aumentará la productividad del usuario. Por este motivo, es el usuario y no los diseñadores ni desarrolladores, el interesado en contar con un sistema con un grado de usabilidad aceptable y es él, entonces, quien determina cuando un producto es fácil de usar.

Una aplicación con un grado de usabilidad aceptable se puede aprender mejor y su aprendizaje perdura más en la memoria. La usabilidad reduce los errores cometidos por los usuarios y lleva a que estos realicen las tareas deseadas de manera más eficiente y efectiva, aumentando así su satisfacción y mejorando su experiencia global con el sistema y por ende, aumentando su productividad personal. Si un sistema no es considerado una herramienta de apoyo, es decir, que realmente ayude al usuario a realizar sus tareas cumpliendo con sus expectativas, el sistema tiene muchas posibilidades de ser simplemente rechazado.

Además las ventajas antes mencionadas, que se resumen en permitir un incremento en la productividad del usuario, podemos observar que reduce los costos de capacitación ya que, teóricamente, no necesitaría de un manual o un curso especializado para poder utilizarlo si el usuario conoce el área en que se desempeña el sistema, y ya que, llega a ser grato de utilizar en muy poco tiempo de utilización, reduce la resistencia al cambio producida al implantar un nuevo sistema. Estos factores son importantes para el cliente o para los dueños del sistema. Por otra parte, aumenta la facilidad de uso, disminuye los errores cometidos

por el usuario y/o ayuda a recuperarse rápida y efectivamente si llegasen a ocurrir, aumentando la satisfacción de trabajar con el producto y generando un alza en la confianza del usuario con el sistema, factores importantes para el usuario. [N4]

Todos estos beneficios implican una reducción y/u optimización de costos tanto en la producción como en la puesta en marcha del sistema, así como también un aumento en la productividad individual de cada usuario y en la de toda la organización donde se implemento el sistema. La usabilidad permite mayor rapidez para realizar tareas sencillas y específicas y reduce las pérdidas de tiempo.

Un software de buena calidad no implica que su usabilidad sea buena. Existen sistemas de gran calidad, pero baja usabilidad y viceversa, sistemas con una gran usabilidad pero de poca calidad. Debemos considerar el concepto de usabilidad como atributo de calidad de una aplicación, consecuentemente, como disciplina o enfoque de diseño y evaluación. Se suele hablar entonces de la Ingeniería de la Usabilidad, conjunto de fundamentos teóricos y metodológicos que aseguren el cumplimiento de los niveles de usabilidad requeridos para la aplicación y que ayuda a incorporar las evaluaciones de usabilidad dentro del ciclo de desarrollo.

Para poder incorporar la ingeniería de usabilidad dentro del ciclo de desarrollo de un software, es necesario un experto en usabilidad o por lo menos, alguien que ya tenga experiencia práctica suficiente en esta área para poder dirigir este proceso. No basta con solo leer una gran cantidad de textos que traten sobre usabilidad y sobre el diseño y desarrollo orientado al usuario, para decir que uno ya conoce a los usuarios, esto no lo hace a uno un experto en el tema. El grado de conocimiento sobre el comportamiento de los usuarios se va adquiriendo y aumentando mientras más experiencia se tenga en usabilidad.

Para profundizar este tema, se puede mencionar que no existe ningún usuario exactamente igual que otro, por lo que si existe alguien que diga que es un experto en el comportamiento de los usuarios y que conoce todos los tipos de usuario, estaría mintiendo y blufando sobre el tema. Tal vez, después de un par de años trabajando en el desarrollo de software similar

o destinado a un mismo tipo de usuario, se pudiese decir que uno conoce como reaccionaria la mayoría de usuarios que pertenecen a este perfil frente a este software específico.

Ya que la usabilidad busca realizar un sistema que sea de fácil utilización, esta debería estar presente en todas las etapas del ciclo de desarrollo del sistema, incluyendo la generación de la documentación para el usuario. Antes de iniciar el proceso de análisis de requerimientos es bueno tener una idea de las características de los usuarios que van a interactuar con el sistema y de la idea que tienen estos usuarios sobre el sistema. Teniendo en cuenta estas consideraciones se ahorrará tiempo y dinero en el transcurso del proceso, puesto que la implantación de nuevos aspectos de usabilidad en etapas posteriores implica un enorme esfuerzo adicional.

## 6.1 Atributos de usabilidad

La ingeniería de la usabilidad ha logrado cuantificar la interacción hombre-sistema a través de varios atributos que son bastantes generales para un sistema. Algunos de estos son:

- **Facilidad de aprendizaje:** es la facilidad que tiene un usuario nuevo, en la utilización del sistema, en poder desarrollar una interacción efectiva con el sistema, utilizando funcionalidades básicas del sistema. Esta es medida por el tiempo utilizado para desarrollar una tarea específica por un usuario inexperto, en relación al tiempo que le tomaría a un usuario más experto en la misma tarea.
- **Eficiencia:** que tan productivo es el sistema cuando un usuario quiere realizar una tarea específica. Este intenta de medir que tan rápido puede, un usuario experimentado, realizar un conjunto de tareas típicas.
- **Carga de memoria:** que tanto le cuesta recordar el como usar el sistema a un usuario esporádico sin la necesidad de tener que utilizar un manual, u otro documento, para poder aprender su funcionamiento cada vez que lo utiliza.
- **Errores:** Este atributo intenta medir que tanto el sistema evita que el usuario cometa errores y que tan fácil es la recuperación cuando ocurre uno. Se mide por la cantidad de errores menores y catastróficos cometidos por un usuario al realizar tareas específicas. Los

errores reducen la eficiencia del sistema y la satisfacción del usuario frente al sistema. A nadie le gusta cometer errores.

- **Satisfacción de uso:** que tan agradable o placentero puede resultar para el usuario determinado el uso del sistema. Es el atributo más subjetivo de todos, puesto que intenta medir la impresión del usuario frente al sistema. Para esto se utiliza encuestas para poder obtener la opinión del usuario después de haber utilizado el sistema.

## 6.2 Reglas de usabilidad

Con los atributos definidos anteriormente, se han fijado algunas reglas de usabilidad que parten desde la práctica y que permiten a los diseñadores de software establecer caminos a seguir en busca de tener una usabilidad mayor en sus sistemas. Estas reglas no son suficientes para asegurar un grado de usabilidad de alta aceptabilidad, pero si ayudan a conseguirla. Algunas de estas reglas son:

- **Acceso:** Esta regla indica que los usuarios con experiencia y conocimiento en el área de la aplicación no deberían tener problemas para poder interactuar con el sistema sin la necesidad de ayuda o instrucciones de terceros. Por esto, el sistema debe presentarse en forma natural y utilizando un lenguaje conocido por los usuarios, para que ellos no necesiten dirigirse a manuales para poder utilizar el sistema. Las interfaces que cumplen con esta regla, se les llama intuitivas y ayudan a que los usuarios realicen tareas de forma más correcta que incorrecta. Esta regla es difícil de cumplir en un dispositivo móvil puesto que el tamaño de un dispositivo móvil es relativamente pequeño, por lo mismo, es difícil mostrar la información necesaria y poder hacer que la interfaz sea intuitiva.

- **Eficacia:** Se refiere a que el sistema no impida al usuario experto a realizar las tareas que le interesa de forma eficiente. Hay que tener cuidado con esta regla, puesto que la influye negativamente para que usuarios inexpertos puedan tener una buena interacción. Por esto es necesario que poder tener en cuenta esta regla como apoyo para todos los niveles de usuarios. Esta regla llega a ser crítica al momento de desarrollo de aplicaciones para dispositivos móviles ya que la incomodidad que tienen estos dispositivos para poder

presentar información al usuario y la incomodidad que tiene los usuarios para seleccionar la opción que desea, disminuyen la eficiencia del sistema.

- **Progresión:** Esta regla se refiere a que el sistema debe permitir que el usuario crezca en conocimientos y experiencia y que mientras lo haga, este se adecue al avance continuo del usuario. Se tiene que tener mucho cuidado con cosas que agraden a usuarios principiantes y que desagraden a los avanzados o viceversa. La idea es buscar un equilibrio que consiga proveer un proceso continuo de adquisición de habilidad. Como en las reglas antes mencionadas, la condición de mostrar poca, pero necesaria, información en la pantalla en los dispositivos móviles, dificulta poder llevar a cabo esta regla en todo su apogeo. Además la restricción de tamaño del software que tienen muchos dispositivos dificulta que se puedan incorporar extras para poder cumplir exitosamente esta regla.

- **Soporte:** Se refiere a que el sistema debe apoyar el trabajo que el usuario desea realizar de manera que se haga más fácil para este. No siempre es posible cumplir esta regla, pero se debe buscar la forma de proporcionar las herramientas que permitan lograr realizar la misma tarea de formas diferentes, es decir dándole al usuario la posibilidad de elegir el camino que más le convenga.

- **Contexto:** Implica que el sistema debe adaptarse a las condiciones de uso reales en el entorno donde se va a utilizar. Si no se cumple con esta indicación, el sistema puede ser muy bueno, pero puede resultar ineficaz su uso. Teniendo en cuenta las diferencias en el ambiente de trabajo, flujos de entrada y salida de información, la población esperada de los usuarios y que cada contexto es diferente para un mismo sistema, entre otros, permitirá poder cumplir con esta regla. Un dispositivo móvil se utiliza en un ambiente dinámico, no es un ambiente fijo como en una aplicación tradicional. Por este motivo, hay que fijarse en que tan dinámico puede llegar a ser el contexto de uso, es decir, si se desarrolla una aplicación para una empresa, es muy difícil que se utilice en ambientes donde es muy complicado utilizar, generalmente se utilizará en ambientes tranquilos por ejemplos, oficinas, salas de reuniones, restaurantes, etc.

## 6.3 Principios de usabilidad

Los principios de usabilidad son abstracciones generalizables que tienen como objetivo orientar a los diseñadores en varios aspectos en el ciclo de desarrollo de un sistema.

Estos principios vienen de la teoría, la experiencia y el sentido común y diversos autores han propuesto diferentes conjuntos de principios de usabilidad a través de los cuales evaluar la usabilidad.

Para mostrar los distintos principios que pueden guiar, a continuación se presenta un conjunto de ellos en los cuales se encuentran presentes principios propuestos tanto por Nielsen como por otros autores.

- **Visibilidad:** Todos los controles tienen que ser claramente visibles, con una buena representación sobre sus efectos. En dispositivos móviles, por el tamaño limitado de la pantalla, es difícil de poder mostrar iconos de gran tamaño y calidad, y por ende, pueden llegar a ser poco representativos.
- **Feedback:** Es la información que se devuelve al usuario sobre que acción se ha realizado y como se ha llevado a cabo, permitiendo al usuario continuar con su actividad. La idea es mantener al usuario informado sobre lo que está pasando en el sistema. El problema está en donde y como mostrar esta información cuando tenemos una pantalla relativamente pequeña y la mayoría la ocupamos para mostrar otro tipo de información.
- **Coincidencia (“mapping”):** Los controles tienen que relacionarse con sus efectos en el mundo. Si ocupamos controles que sean representativos en sus efectos para una gran cantidad de usuarios, no debería haber problema. Este principio está muy ligado al de Visibilidad, teniendo también las mismas complicaciones para el desarrollo de aplicaciones de dispositivos móviles.
- **Uso (“affordance”):** Atributo que tienen los objetos que permiten saber como funcionan y que limitaciones tienen.
- **Control del usuario y libertad:** Se debe proporcionar a los usuarios mecanismos de escape cuando se llegan a situaciones no deseadas. Es decir situación como las de un error

o simplemente que llegaron a un lugar de la aplicación que no deseaban. En las terminales móviles esta más limitada por las restricciones de peso que debe tener la aplicación en algunas terminales. Hay que tener cuidado, mientras más código y recursos tenga nuestra aplicación, más pesada será y más restringido será el grupo de terminales donde podrá implementarse.

- **Flexibilidad y eficiencia en el uso:** Proporcionar aceleradores que sean invisibles a los novatos. No tiene mucha complicación para dispositivos móviles, pero hay que tener cuidado al agregar mucho código extra a la aplicación.
- **Lenguaje común entre el sistema y el usuario:** El sistema debe hablar el lenguaje del usuario, huyendo de tecnicismos incomprensibles o mensajes crípticos. Si este lenguaje es muy largo y no cabe dentro de la ventana de la aplicación, en especial en aplicaciones para dispositivos móviles, estaremos cometiendo una gran falta. Por esto, hay que tener mucho cuidado con el lenguaje que se ocupará, en que este sea representativo y no ocupe mucho espacio en la interfaz de la aplicación.
- **Consistencia y estandarización:** Evitar que diferentes palabras, acciones y situaciones tengan el mismo significado. No tiene muchas complicaciones para dispositivos móviles, solo hay que ser cuidadoso.
- **Estética y diseño minimalista:** Prohibir el uso de información que no es relevante o que se necesite raramente. El diseño minimalista es fundamental en aplicaciones para dispositivos móviles.
- **Impedir errores:** Intentar impedir al usuario cometer errores cuando sea posible. Es decir, por ejemplo, avisándole que el procedimiento que está haciendo es incorrecto en esta fecha y puede estar infringiendo ciertas reglas internas de la organización.
- **Ayudar a los usuarios a reconocer, diagnosticar y recuperar la situación cuando se produce un error.** Usar un lenguaje sencillo para describir la naturaleza del error y sugerir la forma de resolverlo. El lenguaje sencillo también debe ser preciso y corto para aplicaciones de dispositivos móviles, puesto que estos están limitados por el tamaño de pantalla, y por lo tanto, de la interfaz.

- **Ayuda y documentación:** Proporcionar información que se pueda alcanzar fácilmente y que proporcione ayuda siguiendo unos pasos concretos. La ayuda ofrecida por las aplicaciones móviles debe ser muy corta y precisa, puesto que esto significa agregar más código o más recursos y por ende, aumentar el tamaño de la aplicación.

## **6.4 Características de dispositivos móviles que afectan la usabilidad.**

En el capítulo 5 se dieron varios ejemplos de dispositivos móviles, pero estos, por su pequeño tamaño, tienen varias limitaciones que necesitan ser consideradas al momento de desarrollar aplicaciones y al hablar de usabilidad:

- El tamaño pequeño de la Pantalla: Esto puede estar muy limitado, por ejemplo, en teléfonos móviles.
- Capacidad limitada: En términos de la capacidad del procesador, la memoria disponible, el espacio de almacenamiento, y la vida de la batería.
- Baja conectividad: Relativamente desacelerada, por el momento, es la conectividad con Internet donde se quiera. La conectividad inalámbrica de la red de área local, como 802.11, provee solución simple y fidedigna para la comunicación, pero localizada. [C1]

Además, estas características varían de un tipo de dispositivo móvil a otro, produciendo una alta heterogeneidad entre ellos. Por ejemplo:

- El tamaño de los caracteres en la pantalla.
- Tamaño, diseño y distribución de las teclas.
- El grado de dificultad para escribir texto.
- La cantidad de colores con que trabaja.
- La cantidad de líneas y caracteres por línea.

- Algunos soportan escritura predictiva, otros no.
- El formato de los enlaces y de las barras de scroll puede diferir en función del móvil.
- Muchos fabricantes implementan extensiones propias del lenguaje estándar.

Todas estas características, dificultan el desarrollo de una aplicación genérica para todos los dispositivos móviles con un grado de usabilidad aceptable. Algunas empresas han incorporado algunos botones que ayuden en la navegación del usuario dentro del teléfono, pero se debería modificar las aplicaciones para poder utilizar en forma correcta estas cualidades para cada uno de estos diferentes dispositivos.

# Capítulo 7

## Evaluaciones de usabilidad

Una evaluación de la usabilidad, la etapa más importante en el proceso de Diseño Centrado en el Usuario, se puede realizar a través de varios métodos o técnicas y sobre diferentes representaciones del sistema (prototipos en papel, prototipos software, incluso sobre otros sistemas que trabajen en la misma área para poder obtener ventajas y desventajas de este y tenerlas en cuenta en nuestro sistema). Es importante tener en cuenta que cuanto más esperamos para su realización, más costoso resultará la reparación de los errores de diseño descubiertos.

En muchos de estas evaluaciones en que se trabaja con usuarios, es muy importante que al momento de realizar estas experiencias, no se encuentre presente nadie que no este directamente vinculado con la evaluación, incluyendo al jefe del participante, puesto que puede cambiar la actitud del usuario.

Existe una gran diversidad de métodos para evaluación de usabilidad, aunque en el presente trabajo se describen aquellos que se cree de más utilidad y que dan una idea clara del concepto de evaluación. Se describen dichas evaluaciones bajo la clasificación de métodos de inspección y métodos de prueba.

### **7.1 Métodos de inspección**

Este conjunto de métodos se caracteriza en que en él, un grupo de expertos en usabilidad se dedican a la inspección del software o prototipos del mismo que se tengan al momento de la evaluación.

Este tipo de métodos no presenta ningún tipo de problema para poder ser aplicado en evaluaciones de usabilidad de aplicaciones para dispositivos móviles, ya que la experiencia que un experto le ayuda a omitir los problemas referentes al dispositivo y concentrarse más

en la aplicación y por otra parte, no es necesario crear un respaldo, para su posterior análisis, de la experiencia que tiene el experto con la aplicación que se está evaluando.

Dentro de estos métodos podemos encontrar:

### **7.1.1 Evaluación Heurística**

En donde los especialistas en usabilidad, basándose en su propia experiencia y guiándose por un conjunto de heurísticas establecidas, juzgan si cada elemento de la interfaz de usuario sigue los principios de usabilidad establecidos identificando errores y ventajas del sistema. La evaluación heurística puede ser utilizada en, prácticamente, cualquier momento del ciclo de desarrollo, aunque probablemente se adapta mejor en etapas tempranas, cuando no hay material lo suficientemente firme para efectuar alguna prueba. Este tipo de evaluación o requiere gran despliegue de medios, pero su ventaja es la rapidez con la que se puede llevar a cabo.

Muchos autores proponen conjuntos de heurísticas diseñadas para la evaluación de usabilidad tanto para sistemas en general como para tipos de sistemas específicos, como por ejemplo, sistemas basados en páginas web.

Algunas de las principales etapas de una evaluación heurística son:

- **Planificación:** *En esta etapa se deben fijar algunos objetivos de la inspección en base al dominio de la aplicación, el perfil de los expertos que realizarán la evaluación y elaborar un paquete de inspección con los principios que guiarán a los expertos en el análisis.*
- **Evaluación:** *Acá el experto primero se familiariza con el paquete de inspección repasando los puntos que este contiene. Luego el evaluador asume el papel de un usuario específico y se concentra en el contexto o escenario en que se analizará el sistema para poder realizar la evaluación. Los evaluadores deben realizar este proceso individualmente y señalar las tareas que no logró concretar satisfactoriamente.*
- **Reunión:** *Después de que cada experto haya realizado la evaluación, se reúnen para poder proporcionar la información obtenida y poder analizarla. Cabe destacar que antes*

*de este paso, los expertos no deberían comentar los resultados entre si. Esta información puede ser entregada a través de informes formales o a través de una conversación. Además se pueden entregar los problemas clasificándolos según categorías fijadas con anterioridad.*

- **Nivelación:** *se proporcionan niveles de severidad de los problemas encontrados en la evaluación, sacando un promedio de severidad por defecto. Generalmente se clasifican desde problemas de nivel catastróficos, fallos a gran escala que impiden la realización de trabajo de los usuarios, hasta de nivel menor, no se produce una barrera para la realización de una tarea pero producen insatisfacción del usuario.*

- **Resultados:** *El encargado de la evaluación reúne la información obtenida y genera un informe con los datos obtenidos y con sugerencias para el rediseño del sistema.*

Las ventajas y desventajas están señaladas en Tabla 7 6.

### **7.1.2 Recorrido Cognitivo**

En este método el grupo de especialistas que realizan la evaluación, recorre un escenario de tareas determinado por como habría de hacerlo un usuario principiante en el sistema. Por lo cual, ayuda a medir la facilidad de aprendizaje del sistema para usuarios inexpertos en él. Consta de una navegación por las interfaces del sistema, sean esto en prototipos de papel o prototipos funcionales.

El experto debe considerar que la interfaz esta terminada debiendo asumir el papel de usuario tipo y trabajar a través de las tareas que el usuario realizaría comúnmente. El objetivo es encontrar las barreras potenciales que impidan a un usuario a aprender a utilizar el sistema.

Esta técnica igualmente se puede aplicar en cualquier etapa del desarrollo del sistema y, al igual que con los otros métodos, siempre teniendo en cuenta que mientras más temprano mejor.

Las etapas para desarrollar un recorrido cognitivo son:

- **Planificación:** *Se analiza la población de usuarios que interactuarán con el sistema, se definen las tareas representativas que realizarían los usuarios y la sucesión de acciones correctas para cada tarea.*
- **Evaluación:** *Se selecciona una tarea específica para su evaluación, luego se realizan las acciones definidas para esta tarea en particular. Esta evaluación intenta identificar los objetivos de los usuarios y los propósitos de cada tarea.*
- **Resultados:** *Se analizan los resultados obtenidos y se rediseña el sistema. La idea es reparar aquellas deficiencias encontradas en la evaluación.*

Las ventajas y desventajas están señaladas en Tabla 7 6.

### 7.1.3 Inspecciones formales

Esta técnica se llama formal, por la formalidad que tienen sus procesos, etapas y documentación. Esto no quiere decir que los otros métodos no sean formales, sino, que esta se caracteriza por esta condición. Con este método se pueden encontrar gran cantidad de problemas de usabilidad pero los recursos necesarios para aplicarlo son mucho mayores a otros métodos.

Las siguientes son las etapas de una inspección formal:

- **Constitución del equipo de trabajo:** *se conforma un equipo de trabajo donde se pueden encontrar diseñadores, ingenieros, etc.*
- **Asignación de funciones:** *Cada miembro del equipo debe adoptar una función durante la evaluación y las reuniones de equipo que se hagan. Algunas de estas funciones son: Moderador, Propietario, Encargado de registro, Inspector.*
- **Distribución de documentación:** *Es la documentación necesaria para realizar las evaluaciones, como por ejemplo: prototipos en papel, perfil del usuario, contextos de uso, heurísticas, etc.*

- **Inspección:** *Cada evaluador realiza su trabajo en forma individual asumiendo la función que se le fue asignada y de acuerdo a los documentos de evaluación. El evaluador debe registrar todos los defectos encontrados en un formulario entregado al comienzo de la evaluación.*
- **Reunión Formal:** *Finalmente se realiza una reunión final donde el moderador conduce al grupo a través de cada escenario y sus respectivas tareas y los evaluadores informan de los defectos encontrados en los escenarios.*
- **Asignación de prioridades:** *a los defectos encontrados se les asigna prioridad y pasan a ser de responsabilidad del equipo pertinente.*

Las ventajas y desventajas están señaladas en Tabla 7-6.

#### **7.1.4 Inspección de consistencia**

El objetivo principal de esta técnica es verificar la consistencia entre un grupo de interfaces procedentes del mismo tipo de sistema en terminología, colores, esquemas, entradas, manuales de usuario, sistema de ayuda, etc.

Este análisis se utiliza para decidir la forma de realización de una determinada función de los sistemas. Se utiliza en etapas previas de desarrollo, donde el sistema no se haya comenzado a desarrollar, debido a que así si se requieran grandes cambios con respecto a consistencia no tengan que ser modificados en su totalidad.

Hay que poner énfasis al momento de pasar de una interfaz a otra, evaluando al sistema de acuerdo a estas interfaces. Cada inspector crea un documento con los hallazgos y la forma de utilización del sistema. Se realiza una reunión entre los integrantes del equipo de inspección y se discute sobre los sistemas analizados y se llega a un acuerdo unánime en cuanto a la utilización de los productos.

Las ventajas y desventajas están señaladas en Tabla 7-6.

## 7.2 Métodos de pruebas

Acá se reúnen los métodos que se basan en la observación y análisis de la interacción entre el usuario final del sistema y el sistema mismo, anotando los problemas de uso con los que se encuentran para poder solucionarlos posteriormente. Generalmente estas pruebas se realizan dentro de un laboratorio de usabilidad el cual consiste en un área o espacio tranquilo donde el usuario que ayuda en la prueba, puede concentrarse en las tareas dadas. Esta área esta conectada a un área de monitoreo donde el examinador va monitoreando el transcurso de la prueba.

Estos métodos son complementarios a los demás, pero un test con usuarios es más costoso, por lo que es recomendable realizarlo siempre después de otro método de evaluación, ya que sería desperdiciar tiempo y dinero utilizarlo para descubrir errores de diseño motivados por el no cumplimiento de principios generales de usabilidad.

La ventaja que ofrecen los pruebas frente a otro tipo de evaluaciones es que por un lado es una demostración con hechos, por lo que sus resultados son más fiables, y por otro porque posibilitan el descubrimiento de errores de diseño imposibles o difíciles de descubrir mediante otros métodos.

Para que estas pruebas tengan un alto grado de fiabilidad, es necesario realizar una buena selección de los perfiles de usuarios que interactuarán con el sistema y así reclutar participantes que entreguen información relevante.

Las técnicas que prueba de usabilidad generalmente tiene un plan de trabajo muy similar, entre estos pasos podemos encontrar:

- **Fijar objetivos:** *Se determina el propósito del prueba y los objetivos a lograr durante la prueba identificando las cosas que se desean evaluar.*
- **Reclutar a usuarios:** *Identificar a los usuarios a participar en relación a los perfiles necesarios para la ejecución de la prueba.*
- **Diseñar la prueba:** *Determinas los escenarios o contextos y las tareas que se va a evaluar, identificando lo que se necesita para la prueba (equipos, sistemas a analizar,*

*prototipos, etc). Identificar que significa una tarea terminada exitosamente e identificar quien será el conductor de la prueba.*

- **Realizar la prueba:** *Conducir a los usuarios en la realización de las tareas mientras se reúne la información relevante de las observaciones.*
- **Analizar los datos:** *Se reúnen los resultados, se clasifican los problemas y se presentan conclusiones junto con soluciones.*

En el caso específico de la aplicación de estos métodos a aplicaciones de dispositivos móviles, se puede determinar que son más difíciles de llevar a cabo. Esto se debe a varios puntos que deben ser tomados en cuenta. Entre otros podemos encontrar:

- Un dispositivo móvil generalmente no tiene la capacidad de poder ejecutar la aplicación que se está poniendo en prueba y, al mismo tiempo, ejecutar otra aplicación que respalde toda la interacción que ocurre entre el usuario y la aplicación.
- Generalmente el dinamismo del ambiente en una experiencia real, se pierde al realizar estas pruebas en un ambiente controlado.

Entre estos métodos podemos encontrar:

### **7.2.1 Pensamiento en voz alta**

En este método el usuario que interactúa con el sistema expresa en voz alta lo que opina, siente y piensa mientras realiza una tarea determinada en un contexto específico y mientras es observado por un examinador. Esto ayuda a conocer lo que el usuario quiere y necesita del sistema y entender el modelo mental del usuario y su interacción con el sistema a través de la interfaz.

Esta técnica puede ser utilizada en cualquier etapa dentro del ciclo de vida de desarrollo del producto, siendo una forma eficaz y económica de obtener grandes cantidades de información cualitativa. Para su realización, es necesario reclutar entre 3 y 5 usuarios por cada perfil para así obtener diferentes puntos de vista.

Antes de comenzar con la evaluación, es necesario preparar el material para la prueba y establecer el escenario bajo el cual se realizará la prueba. Al llevar a cabo la evaluación es necesario registrar todo lo que el usuario va comentando durante la ejecución de las tareas y al finalizar, se recomienda solicitar al usuario que responda un cuestionario completo con respecto a la experiencia realizada detallando los defectos y hallazgos encontrados.

Las ventajas y desventajas están señaladas en Tabla 7-6.

### **7.2.2 Interacción constructiva**

Este es un método que deriva del método anterior. Aquí la prueba es realizada por 2 usuarios que, conjuntamente, van interactuando con el sistema y pensando en voz alta mientras son observados por un examinador. Además de potenciar las ventajas del pensamiento en voz alta, ayuda a que los usuarios se expresen de mejor manera, puesto que les es natural verbalizar conjuntamente. Generalmente es preferente aplicar este método a sistemas que van a ser utilizados por usuarios en un entorno de trabajo de equipo.

Al momento de reclutar a los usuarios se debe tener presente que se deben identificar de a parejas y determinar las tareas que van a realizar. Se debe registrar todo lo que hablan los participantes durante la evaluación, incluyendo los comentarios individuales y las conversaciones entre ellos, permitiéndoles ayudarse entre ellos, para analizar la forma en que se haría si ellos estuvieran trabajando en conjunto tras un objetivo común mediante el uso del producto.

Las ventajas y desventajas están señaladas en Tabla 7-6.

### **7.2.3 Técnica de interrogación:**

Es una técnica en la que el usuario es interrogado directamente sobre el sistema utilizado por el moderador de la prueba. Esta técnica es generalmente aplicada después de algún otro método de prueba, como por ejemplo, pensamiento en voz alta. Con ella es fácil detectar que partes del sistema resultan obvias y cuales son más difíciles de manejar. Es una técnica que, en comparación con otras, va más allá debido a que al usuario se le interroga

directamente sobre el sistema utilizado. Esta técnica puede ser utilizada en cualquier parte del ciclo de desarrollo.

Las ventajas y desventajas están señaladas en Tabla 7-6.

Por último, cabe destacar que, aunque estas no son todas las técnicas que existen, si son algunas de los métodos más reconocidos y utilizados. Es importante destacar que estos métodos no son exclusivos entre si. Se pueden unir, complementar o simplemente tener en cuenta mientras se aplican otros, dependiendo de la experiencia de los examinadores, los diseñadores y desarrolladores y dependiendo del tiempo disponible para realizar las evaluaciones, entre otros.

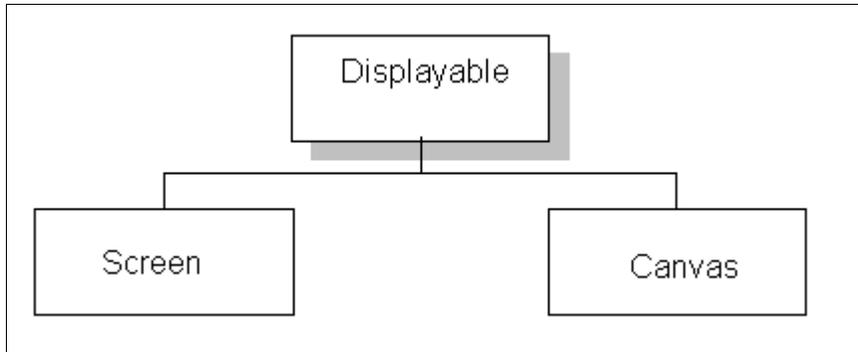
### **7.3 Evaluaciones de usabilidad en dispositivos móviles**

En el momento en que comenzó con la realización de este trabajo, no se lograron encontrar documentos que se refieran a métodos de evaluación diseñados específicamente para aplicaciones en dispositivos móviles y mucho menos, del como poder hacer compatibles la ingeniería de usabilidad y la ingeniería de software de este tipo de aplicaciones. Por otra parte, si es posible encontrar documentos que ayudan, a los implicados en el proceso de diseño y desarrollo, a tener en cuenta ciertos conceptos importantes de usabilidad y orientados a aplicaciones de dispositivos móviles. Por ejemplo, la empresa Nokia, que está muy comprometida con el concepto de usabilidad en terminales móviles, ha proporcionado una serie de documentos en su página web dedicada a la usabilidad para aplicaciones desarrolladas para sus teléfonos celulares. Uno de estos documentos, por ejemplo, contiene un *checklist* para el caso de desarrollar aplicaciones empresariales en sus terminales series 80 y 60.

Nokia marca la diferencia entre aplicaciones empresariales, o comunes, y juegos, porque los juegos tienen algunas etapas necesarias en el ciclo de desarrollo que no tienen las aplicaciones empresariales como lo es el diseño conceptual del juego, que son propios del diseño de cualquier tipo de juego para cualquier dispositivo (PC, consolas de juego, etc.). Además, el desarrollo de software se realizan sobre distintos tipos de 'Entornos gráficos'. Por ejemplo Java 2 Micro Edition(J2ME) trabaja con 2 tipos: Screen (alto nivel) que nos

ofrece trabajar colocando elementos de formulario de una interfaz de usuario, como textos, menús, etc. y Canvas (bajo nivel) con la que podemos trabajar a nivel gráfico.

Ilustración 7-1: Entornos gráficos de J2ME



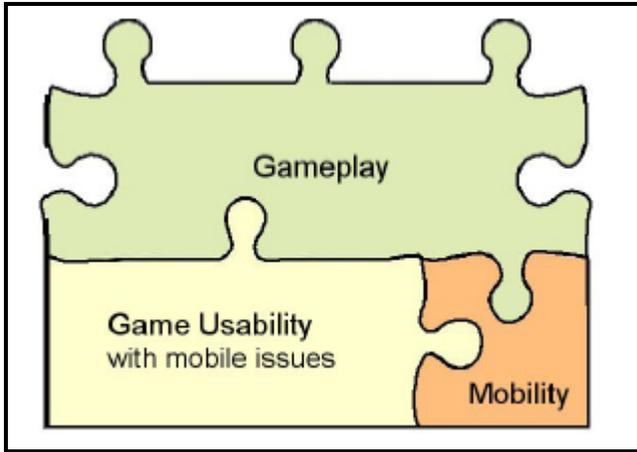
En el caso de aplicaciones empresariales, Nokia señala seguir, entre otros, los siguientes principios de diseño:

- La vista principal aplicativa debería tener, en gran parte del área del indicador, el nombre e icono de aplicación. El resto del área puede ser usado en subsiguientes puntos de vista.
- El usar la tecla de *Scroll* (control de navegación) como tecla predeterminada para la confirmación en diálogos, debería ser considerado cuidadosamente caso por caso.
- En el caso del menú, ordenar los ítems según la frecuencia de uso.
- Evitar utilizar el *scrolling* en el menú, especialmente en los menús de cascada.

Por otra parte, Nokia presenta un conjunto de artículos relacionados con la experiencia, información y herramientas que ayudan a diseñadores y desarrolladores en todas las etapas de diseño y desarrollo de juegos para teléfonos celulares, para poder obtener juegos que tengan una gran usabilidad y además una gran jugabilidad.

Dentro de esta biblioteca se presenta un artículo con un conjunto de heurísticas que consta de tres módulos: *Gameplay*, *Mobility* y *Game Usability* (Ilustración 7-2).

Ilustración 7-2: Módulos de Heurísticas propuestas por Nokia



El listado de estas heurísticas se presenta en las tablas: Tabla 7-1, Tabla 7-2 y Tabla 7-3.

Tabla 7-1: Listado de heurísticas para la movilidad

<b>Heurísticas de movilidad</b>	
1	Las sesiones de juego y de obra teatral (videos, introducción, etc.) pueden comenzar rápidamente.
2	El juego no es incomodo jugarlo en un ambiente no estático.
3	Las interrupciones son manejadas razonablemente.

Tabla 7-2: Listado de heurísticas para Juegos

<b>Heurísticas de Usabilidad para Juegos</b>	
1	La representación audiovisual soporta el juego.
2	La disposición de la pantalla es eficiente y visualmente agradable.
3	La UI(Interfaz de usuario) del dispositivo y la UI de juego, sirven para sus propósitos.
4	Los indicadores son visibles.

5	El jugador entiende la terminología.
6	La navegación es coherente, lógica, y minimalista.
7	Las llaves de control son consistentes y siguen a las convenciones estándar.
8	Los controles de juego son convenientes y flexibles.
9	El juego da información retroactiva en las acciones de los jugadores.
10	El jugador no puede hacer errores irreversibles.
11	El jugador no tiene que aprender de memoria cosas innecesariamente.
12	El juego contiene ayuda.

Tabla 7-3: Listado de heurísticas para el Gameplay

<b>Heurísticas para el <i>Gameplay</i></b>	
1	El juego provee metas evidentes o soporta metas creadas por el jugador.
2	El jugador ve el progreso en el juego y puede comparar los resultados.
3	Los jugadores son recompensados y las recompensas son significativas.
4	El jugador tiene el mando.
5	El reto, la estrategia y el paso están en balance.
6	La experiencia de primera instancia es alentadora.
7	La historia de juego soporta el gameplay y es significativa.
8	No hay tareas repetitivas o aburridas.

9	Los jugadores pueden expresarse.
10	El juego soporta estilos diferentes de modo de jugar.
11	El juego no se estanca.
12	El juego es coherente.
13	El juego usa diferenciación ortogonal de la unidad.
14	El jugador no pierde cualquier posesión ganada a duras penas.

Por otra parte, Mauro Marinilli en su libro ‘Profesional Java User Interfez’, presenta la importancia del diseño orientado al usuario las GUI desarrolladas en JAVA. Además, el capítulo 10 está dedicado al diseño de GUI de aplicaciones J2ME ocupados en aplicaciones para dispositivos móviles.

En el apéndice A, se presenta un cuestionario recomendado por Mauro Marinilli para poder complementar alguna prueba de usabilidad.

En el caso de realizar pruebas de usabilidad, el problema que existe es que están se desarrollan generalmente dentro de un laboratorio de usabilidad. Este laboratorio presenta la deficiencia de ser un ambiente estático y controlado a diferencia del ambiente real de uso de una aplicación para dispositivos móviles. Hay que recordar que el objetivo de este tipo de dispositivos es permitir que el usuario tenga acceso en cualquier momento y en cualquier lugar a la información que desee. Esto incluye situaciones donde no existe una gran comodidad para poder manipular el dispositivo o simplemente no se pueda comprender la interfaz de la aplicación. Es decir, el ambiente circundante y la movilidad establecen requisitos especiales para aplicaciones de dispositivos móviles. Este ambiente dinámico es difícil de emular dentro de un laboratorio. A lo más, se puede simular algunas condiciones dinámicas, como lo es la inestabilidad de la señal de la conexión, pero en cambio, no podemos simular el movimiento de un vehículo en un laboratorio.

Es por este motivo que se deberían realizar algunas pruebas fuera de un laboratorio, donde el contexto de uso sea lo más semejante al verdadero, pero ¿es igual, menos o más conveniente el realizar pruebas dentro de un laboratorio o fuera de él?

Anne Kaikkonen, y otros, en su texto titulado “*Usability Testing of Mobile Applications: A Comparison between Laboratory and Field Testing*” [K1], muestran los resultados obtenidos en un estudio comparativo, realizado por ellos, entre realizar pruebas de usabilidad a aplicaciones para dispositivos móviles dentro de un laboratorio y realizarla fuera de este, en un ambiente dinámico. Ellos se preguntaron si las pruebas de campo son críticas en las evaluaciones de usabilidad de aplicaciones de dispositivos móviles o si las pruebas realizadas en laboratorio enlantan la suficiente validez ecológica.

En dicho experimento solo se cambió el ambiente en que se realizó la prueba, es decir, las tareas, el método de pensamiento en voz alta, etc. permanecieron constantes. Ellos se basaron en la premisa que si se adecua más un laboratorio al ambiente real del contexto de uso, menor será la diferencia de los resultados obtenidos. Por este motivo, decidieron utilizar un laboratorio normal de usabilidad, es decir, lo más estándar posible. Además, si se considera que para el equipo de evaluadores el modificar para cada aplicación el laboratorio de usabilidad para hacerlo lo más parecido al contexto real de uso, se incurriría en muchos costos y tiempo que muchas veces no esta considerados en los presupuestos ni en las fechas de entrega.

En una prueba de usabilidad se recomienda utilizar aprox. 5 usuarios por cada ronda de tareas, pero para este caso, donde se quería estudiar los métodos de prueba, se decidió ocupar un mayor número de usuarios para así obtener mejores resultados.

Los examinadores, que actuaban de moderadores de las pruebas, tenían entre 5 y 13 años de experiencia en pruebas de usabilidad. Cada examinador dirigió un número igual de pruebas ambos en el laboratorio y en el campo.

Los dos grupos de usuarios utilizados (para las pruebas de campo y laboratorio) estaban constituidos por 20 individuos por grupo, de los cuales no se diferenció por experiencia en la utilización de dispositivos móviles. Las edades de todos los usuarios fluctuaban entre los

22 y 35 años. Para la prueba de laboratorio, la edad promedio era de 28,5 años con desviación estándar de 3,5, mientras que en la prueba de campo, la media era de 28,8 con 4 años de desviación estándar. La cantidad de hombres y mujeres era igual en cada grupo.

Para conseguir el objetivo de dicha prueba, se realizaron las siguientes preguntas:

- ¿Se encuentran la misma cantidad de problemas en ambas pruebas?
- ¿Son los mismos problemas y los fenómenos encontrados en el laboratorio y en el campo? ¿En caso de que no, cuál es la diferencia?
- ¿Si hay diferencias entre las conclusiones de ambas pruebas, es diferente también su severidad?
- ¿Son diferentes los tiempos de ejecución de tarea?
- ¿Afecta el ambiente la actuación experimental del usuario?

Las pruebas del laboratorio fueron realizadas en un ambiente típico de prueba de usabilidad. El trasfondo estaba bien controlado: No hubo interrupciones externas inesperadas, distracciones por ruidos, condiciones del alumbrado que varía, etc.

Los usuarios recibieron una introducción breve por el teléfono móvil y estaban adiestrados para pensar en voz alta durante la prueba. Las tareas fueron dadas oralmente en un orden predefinido las cuales fueron diseñadas de esta forma para ser usadas también en la pruebas de campos.

Las pruebas de campo fueron realizadas completamente en el exterior donde los usuarios podían caminar, permanecer inmóvil, sentarse o hacer lo que normalmente harían al realizar las tareas. Como en el laboratorio, el moderador dio las tareas definidas oralmente usando la redacción convenida.

Las sesiones experimentales empezaron en una oficina alrededor de la hora pico diaria. Los usuarios estaban adiestrados para tomar el metro y salir al encuentro de un amigo en un centro de compras. Los usuarios recibieron la primera tarea mientras caminaron para la

estación del metro. El moderador siguió al usuario algunos pasos atrás y dio las tareas durante el viaje mientras iban terminando las anteriores. Otra parte de las tareas fueron realizadas al circular en el centro comercial. Durante las pruebas, los usuarios tuvieron que cruzar una calle ocupada, una escalera mecánica atestada de gente, transpórtese en un metro y caminar en un centro comercial grande lleno de los disturbios.

Para esta prueba, los usuarios experimentales tuvieron que traer puesto equipo especial para registrar la grabación de los datos de la prueba mientras estaban en movimiento. El equipo puede ser visto en la Ilustración 7-3. El equipo constó de dos partes: La unidad experimental del usuario y la unidad del moderador. La unidad experimental del usuario tuvo tres cámaras de vídeo. Como en el laboratorio, la primera captó la pantalla y el teclado del teléfono móvil, mientras la segunda cámara registró la cara del usuario. La tercera cámara se utilizó para dar vista a las afueras del usuario. La unidad tuvo también un micrófono, un dispositivo audio de la grabación y de vídeo, un transceptor inalámbrico de la señal del video y baterías para proveer poder.

La unidad del moderador constó de uno monitor electroluminiscente de 6", una cámara de vídeo, un transceptor de vídeo inalámbrico y una batería. La cámara registró las afueras del usuario de perspectiva del moderador. El monitor le permitió al moderador ver lo que el usuario estaba haciendo con la aplicación móvil cuando el usuario.

La aplicación que se evaluó, era una que todavía se encontraba en etapa de prototipo, lo cual fue adecuado para esta prueba, así se aseguraba que los usuarios no estuviesen familiarizados con la aplicación. El nombre de la aplicación era Mobile Wire y su idea era el de poder transmitir archivos entre el teléfono móvil y una PC, ya sea desde una PC al teléfono, del teléfono a una PC o de una PC a otra PC.

Ilustración 7-3: Equipamiento para una prueba de campo



El listado de tareas utilizado para ambas pruebas, fue el siguiente:

1. Enviar un SMS.
2. Abriendo el SMS recibido, abriendo la página WAP mencionada en el mensaje.
3. descargar la aplicación experimental para el dispositivo desde la página WAP.
4. Encontrar y abrir al público la aplicación de la que se descargo.
5. Usar la aplicación para ver las imágenes compartidas por un amigo.
6. Descargar una imagen compartida al teléfono.
7. Sacar una foto usando con la cámara del teléfono y guardarla en una computadora doméstica.
8. Otorgar los derechos a un amigo para mirar la imagen guardada en la computadora doméstica.
9. Cerrar la aplicación.
10. Eliminar la aplicación.

En los resultados obtenidos del experimento se destaca que se encontraron la misma cantidad de errores en ambas pruebas y que, además, eran del mismo tipo. Estos problemas se encuentran enumerados más abajo en la Tabla 7-4 y la frecuencia con que se presentaron es presentada en la Ilustración 7-4.

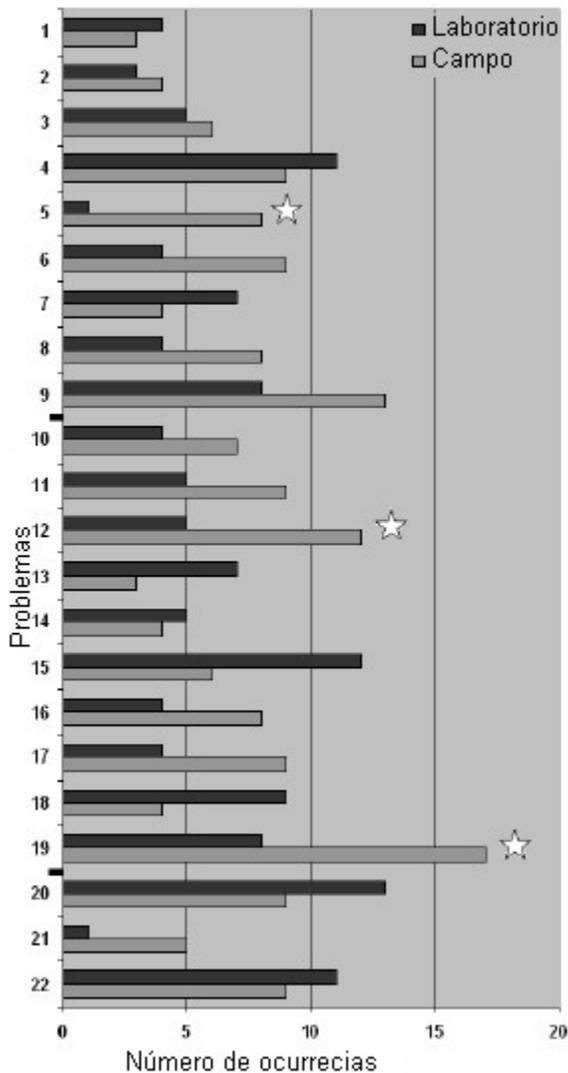
<b>#</b>	<b>Problema</b>	<b>Grado de criticidad</b>
1	<i>No puede encontrar el link de descarga de la página WAP</i>	1
2	<i>No se encuentra el precio</i>	1
3	<i>Las dificultades para salir del navegador</i>	1
4	<i>Las dificultades para encontrar la aplicación después de descargarla</i>	1
5	<i>Las dificultades en entender el concepto de 'otorgar derechos de acceso'</i>	1
6	<i>Se agregó el amigo pero no se le proporcionó los derechos de acceso a la computadora</i>	1
7	<i>No salvaron los cambios después de dar las derechos</i>	1
8	<i>Renuncia al intento de supresión de la aplicación</i>	1
9	<i>Renuncia a por lo menos una tarea durante la prueba</i>	1
10	<i>Sumando depósito problemático</i>	2
11	<i>El movimiento del cursor causa problemas</i>	2
12	<i>Al principio es poco claro el proceso de descarga</i>	2
13	<i>El diálogo de descarga es problemático</i>	2

14	<i>Re-carga de la aplicación, no es esta seguro si ya fue descargada</i>	2
15	<i>Tratar de enviar la foto desde la aplicaciones de la cámara (lugar equivocado)</i>	2
16	<i>Tratar de enviar la foto desde lista de dispositivos (lugar equivocado)</i>	2
17	<i>Tratar de enviar la foto desde la etiqueta (agra lugar equivocado)</i>	2
18	<i>Tratar de enviar la aplicación hasta la trash bin</i>	2
19	<i>Tratar de eliminar la aplicación desde carpeta equivocada</i>	2
20	<i>El tiempo de descarga es problemático</i>	3
21	<i>Diferentes menús desde el joystick y las softkey, causan problemas</i>	3
22	<i>Se lee la ayuda al menos una vez durante la prueba</i>	3

Tabla 7-4: Listado de Problemas en ambas pruebas (Campo v/s Laboratorio)

Los primeros nueve problemas, de la Tabla 7-4, fueron conciderados críticos (el problema impidió que usuarios completaran la tarea). Estos son marcados con número 1 en la lista de problemas. Los problemas de 10 para 19 fueron problemas severos (le causaron problemas evidentes a los usuarios al realizar las tareas). Estos son marcados con número 2. Los último tres fueron problemas menores causando incomodidad al usar el sistema por primera vez, pero los usuarios las aprendieron a superar durante la prueba.

Ilustración 7-4: Frecuencia de problemas de ambos casos (Campo v/s Laboratorio)



Para los problemas marcados con estrella en la Ilustración 7-4, las diferencias estadísticamente significativas fueron encontradas en el número de ocurrencias.

Promediando, los problemas en el campo no fueron más severos que los encontrados en el laboratorio.

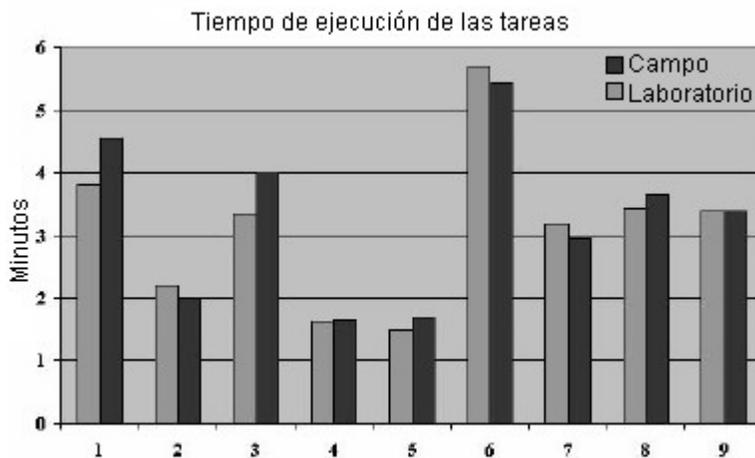
Un asunto interesante está relacionado con problema 20: En el ambiente del laboratorio 13 usuarios se quejaron del tiempo de descarga, en el campo, 9 usuarios comentaron acerca del mismo asunto. La redacción usada en laboratorio demostró más frustración que en ambiente de campo. Basados en la reacción de los usuarios experimentales, los expertos habrían evaluado la severidad del problema diferentemente en dos posiciones: En

laboratorio el problema se consideró severo, pero en el campo la evaluación fue problema menor.

No hubo diferencias significativas en los tiempos de ejecución de tareas individuales entre los dos trasfondos experimentales. Estos tiempos de ejecución pueden verse en la Ilustración 7-5.

La posición experimental significativamente no afectó la velocidad y el éxito de la ejecución de tarea de los usuarios. De cualquier forma, el ambiente pareció tener un mayor impacto en las conclusiones cualitativas de la prueba.

Ilustración 7-5: Tiempo de ejecución de tareas



Cuando la tarea dada no les fue familiar, los usuarios podrían haber clavado los ojos en la pantalla del teléfono o podría haberse movido a un lado para tener un lugar más tranquilo para que puedan terminar la tarea. Este comportamiento da compenetración en el nivel de dificultad de las tareas, y no se hizo pública en un laboratorio sedimentándose.

Las interrupciones potenciales no parecieron molestar a los usuarios durante la prueba: Aun los circunstantes chillones que captaron la atención del moderador no quebrantaron la concentración de los usuarios. Continuaron realizando la tarea sin interrupciones significativas. Hubo sólo pocos pasajeros curiosos en el metro.

En el campo, los usuarios experimentales hablaron más libremente acerca de la aplicación, y si la podrían usar.

Al hacer una descarga de algo o al esperar que la descarga terminara, los usuarios no siempre prestaban atención a la tarea que estaban esperando terminar. Podían recoger un periódico gratis o podían comprobar las llamadas perdidas en su teléfono personal. Ésta fue también la situación cuando miraron alrededor y observaron a los otros pasajeros. Este comportamiento fue sólo observado en el campo y puede tener implicaciones en cómo notan los usuarios los indicadores de progreso en la aplicación: En el campo pueden perder las indicaciones que desaparecen con oportunidad del momento.

Aun si no hubo diferencias significativas entre los dos tipos de prueba, el tiempo experimental total fue más largo en el campo (45 min. aprox.) que en el ambiente del laboratorio (35 min. aprox.). La diferencia está explicada por el trasfondo experimental en el campo: Ciertas tareas estaban sincronizadas para tomar lugar en posiciones específicas. Muchos usuarios experimentales de la prueba de campo, esperaban tiempo extra para poder verificar el estado de su tarea (por ej. la descarga de un archivo).

También, el tiempo destinado para acomodar las herramientas utilizadas en las pruebas experimentales fue más largo en comparación con el de las pruebas de laboratorio. En las pruebas de campo se tomó más tiempo instalando el equipo, por ejemplo la mochila con los dispositivos de la grabación necesitó estar ajustada para cada usuario. Las pruebas de campo siempre se iniciaron en el mismo sitio y siguiendo la misma ruta, queriendo decir que los moderadores mal gastaron más tiempo viajando de vuelta al punto de partida.

Tabla 7-5: Tiempo de la prueba

Lugar	Tiempo Total (promedio)	Instrucciones y preparación (estimado)	Viaje estimado para el moderador (estimado)

Laboratorio	35 min	10 min	-
Campo	45 min	20 min	40 min

En la Tabla 7-5 se resumen los tiempos necesitados para las tareas, preparaciones y viajes durante las pruebas. En el laboratorio las pruebas podrían ser arregladas para iniciarse cada hora, pero en las pruebas de campo el intervalo entre las pruebas fue de dos horas.

En conclusión:

- El realizar pruebas de usabilidad tanto en un laboratorio como fuera de este, no afecta la severidad de los problemas encontrados.
- Los tiempos ocupados en las tareas individuales fueron prácticamente los mismos, pero el tiempo total necesitado para la experimentación fue mucho más largo para el campo que para el laboratorio.
- Aunque las acciones del moderador en las pruebas estaban definidas para ser las mismas en el campo y en el laboratorio, la prueba de campo pareció ser más casual. Los usuarios tendieron a comentar más libremente acerca de la aplicación.
- Los problemas que fueron encontrados más frecuentemente en el campo parecen estar relacionados con entender la lógica de la aplicación, pero, por otra parte, hubo también asuntos complicados sin diferencias entre los dos trasfondos experimentales.
- Las pruebas de campo no se vieron afectadas por la cantidad y potencialidad de las interrupciones que se produjeron.
- Al enfrentarse a tareas más complicadas, los usuarios buscaban refugio seguro para poder realizarlas tranquilamente.
- En las pruebas de campo los usuarios se concentraron con exceso en la prueba y pocos usuarios pudieron realizar todas las tareas mientras caminaban.

- El tiempo y los esfuerzos ocupados en las pruebas de campo son más altos que los ocupados en las pruebas de laboratorio.

## 7.4 Análisis comparativo de los métodos de evaluación de usabilidad

En el siguiente cuadro, se muestra una tabla comparativa entre los distintos tipos de evaluaciones de usabilidad.

Tabla 7-6: Cuadro comparativo de evaluaciones de usabilidad

<b>Método</b>	<b>Grupo</b>	<b>Posibles etapas de aplicación</b>	<b>Ventajas</b>	<b>Desventajas</b>	<b>Aplicación en móviles</b>
Evaluación heurística.	Método de inspección.	Todo el ciclo de desarrollo.	-Bajo costo en comparación con otras técnicas. -Utilizable en etapas tempranas del ciclo. -Encuentra muchos problemas tipo.	-Puede obviar problemas del dominio específico. -Si aumentan los evaluadores, más errores se encontrarán, pero los costos se dispararán.	Existen listados de heurísticas creados para este tipo de aplicaciones que pueden ser utilizados.
Recorrido cognitivo.	Método de inspección.	Todo el ciclo de	-Utilizable en etapas	-El tiempo consumido en la	Aplicable sin mayores

		desarrollo.	tempranas. -Hallazgos en relación a las tareas necesarias para un usuario.  -Identificar más claramente las necesidades de los usuarios.	evaluación es elevado.  -Solo identifica problemas de aprendizaje.  -Puede pasar por alto otros problemas.	inconvenientes.
Inspección formal.	Método de inspección.	Todo el ciclo de desarrollo.	-Se encuentran una gran cantidad de problemas tipo.  -Aplicable desde etapas tempranas de desarrollo.	-Dificultad para conseguir el equipo de colaboración.  -El tiempo consumido en procesos burocráticos de la evaluación es elevado.	Aplicable sin mayores inconvenientes.
Inspección de consistencia	Método de inspección	Etapas previas al desarrollo.	-Se consigue información para lograr que productos similares se comporten de	-Necesidad de grupo grande de inspectores.  -Se centra	Aplicable sin mayores inconvenientes.

			<p>forma similar.</p> <p>-Se puede utilizar en etapas previas.</p>	<p>más en la forma de utilización del producto que en las posibilidades de mejoras.</p>	
Pensamiento en voz alta.	Método de prueba.	Cuando se tenga algún prototipo funcional o no.	<p>-Se encuentra una gran cantidad de problemas de usabilidad.</p> <p>-Se puede determinar el porque de los problemas encontrados.</p> <p>-La terminología utilizada puede ser utilizada en el diseño y documentación.</p> <p>-Fácil y económico en su aplicación.</p>	<p>-Puede tornarse un poco lenta debido a su forma de ejecución</p> <p>-Los diferentes usuarios pueden tener diferentes formas de aprendizaje</p>	<p>-Difícil ver lo que esta respondiendo el sistema durante la prueba.</p> <p>-No existen buenas herramientas que ayuden en el proceso.</p>
Interacción	Método de	Cuando se tenga	-Puede revelar más	-Puede tornarse un	Lo mismo que en el caso del

constructiva.	prueba.	algún prototipo funcional o no.	información con respecto a la técnica de pensamiento en voz alta.  -Se encuentra una gran cantidad de problemas de usabilidad.	poco lenta debido a su forma de ejecución.  -Los diferentes usuarios pueden tener diferentes formas de aprendizaje.	pensamiento en voz alta.
Técnica de interrogación.	Método de prueba.	Todo el ciclo de desarrollo.	-Obtención de información suplementaria de gran utilidad.  -Simple y económico.	-Los datos obtenidos son subjetivos.	-Sin mayores inconvenientes.

## 7.5 Herramientas actuales

Por lo que se ha logrado apreciar en el transcurso de este trabajo, es que las herramientas ocupadas para las evaluaciones de usabilidad de aplicaciones para este tipo de dispositivos, no es la más adecuada. En el experimento realizado por Anne Kaikkonen el tiempo total utilizado en las pruebas de campo excedía al tiempo ocupado en las pruebas de laboratorio. Esta fue la clave de porque no es muy conveniente realizar las pruebas fuera de un laboratorio. Gran parte de este tiempo perdido se debió a la preparación que se debía hacer antes de comenzar la realización de una prueba de campo (la otra parte fue porque no

siempre se estaban realizando las tareas como en un laboratorio). El revisar que las baterías estén suficientemente cargadas (sino, reemplazarlas), revisar que las cámaras estén capturando lo que se desea capturar, ajustar la mochila con el equipo necesario al usuario, fueron procesos que se pudiesen evitar si se contara con una herramienta que estuviera diseñada para ser más práctica en esas situaciones.

Las únicas herramientas encontradas, no están diseñadas para resolver este problema, pero sí para apoyar las pruebas de usabilidad con usuarios.

La primera es una interfaz de la aplicación que se desea evaluar para poder realizar pruebas de usuario remotos de usabilidad. Esta interfaz, que se presenta en una página web, es apoyado por el software UserZoom, el cual ayuda tanto en la automatización de la captura de información en pruebas sobre grandes muestras de usuarios, geográficamente dispersos, tanto a nivel nacional como internacional, de manera detallada, rápida y eficaz, como en el análisis de los datos capturados. Este software no solamente es utilizado para las evaluaciones de usabilidad de dispositivos móviles, sino que más bien, fue desarrollada para las evaluaciones de aplicaciones tipo WEB. Esta herramienta ayuda tanto en la captura de información como en el análisis de la información capturada.

UserZoom fue desarrollado por Xperience Consulting, empresa española que ofrece el servicio de gestión y medición de experiencia de usuario. Esta empresa ofrece un abanico de servicios con este objetivo, entre ellas se encuentran Test de usuarios en remoto, Test de usabilidad en laboratorio, apoyo en la Usabilidad para móviles, Eye Tracking y apoyo en el Diseño Centrado en el Usuario.

Para el servicio de Usabilidad para móviles, Xperience Consulting, crea un emulador de la interfaz de la aplicación que se desea evaluar. Esta está desarrollada en un software de edición multimedia como por ejemplo Adobe Flash. Este emulador es puesto en una página web y gracias a UserZoom es posible capturar una amplia gama de información que después es analizada con el mismo software.

UserZoom esta compuesto por:

- Barra UZ: este es el primer componente del sistema. Se trata de un ligero pero sofisticado 'plug-in' que solicita a los usuarios que completen tareas y contesten a una serie de preguntas online. Mientras los usuarios participan en el test el sistema captura, se graba toda la interacción con la web o aplicación online, incluyendo si cumplen o no las tareas, n° de clicks y tiempo necesitado, caminos recorridos, clicks realizados, dudas y recomendaciones, etc. Este componente se aprecia como una barra en parte inferior de la página WEB.
- Plataforma UZ: es el 'backoffice' de UserZoom, es decir, un sistema interno de gestión de tests remotos, que permite realizar inserción de 'scripts' (guión del test), monitorización de avances y análisis post-realización de tests. La plataforma cuenta con 2 componentes, UZ Log Manager, encargado del filtrado de datos, y UZ Analytics , encargado del analizados de los datos. A través de la Plataforma UZ realizamos el análisis y segmentación de resultados, así como la exportación automática de información, para generar un informe entregable final en formatos como Word y PowerPoint.

En la Ilustración 7-6 se muestra una captura de pantalla de una versión de muestra del emulador de interfaces para teléfono Sony Ericsson P900. Esta muestra se encuentra en la página WEB de la empresa (<http://www.xperienceconsulting.com>). Con el mouse se puede controlar el puntero del teléfono y así poder seleccionar lo que se desee en la pantalla táctil del teléfono y poder apretar las teclas que el usuario necesite.

Ilustración 7-6: Captura de pantalla del emulador de interfaces móviles de Xperience Consulting



Hay que recalcar que es el emulador de la interfaz de la aplicación, la herramienta diseñada para este tipo evaluaciones de usabilidad. Es esta, la que le informa al software UserZoom sobre los eventos que ocurren en la interacción con el usuario.

La segunda, y última, es un dispositivo que ayuda acoplar una pequeña cámara de video, como una cámara web, a un dispositivo móvil. El objetivo de este dispositivo es poder capturar, en video, la interacción que tienen los dedos del usuario con las teclas del dispositivo y los eventos que ocurren en la pantalla del dispositivo.

Xperience Consulting, también cuenta con este tipo de dispositivos y es utilizado en las pruebas realizadas dentro de un laboratorio y la han llamado Mobile Xperience Cam.

Estos dispositivos alteran el terminal móvil, afectando los resultados de las pruebas, pero por lo visto en el experimento realizado por Anne Kaikkonen, la alteración en los resultados obtenidos es bastante baja y podrían llegar a ser despreciables.

Ilustración 7-7: Mobile Xperience Cam



Otros dispositivos similares a la Mobile Xperience Cam, como los mostrados en la Ilustración 7-8, pueden ser encontrados en la red, pero todos son muy parecidos y ninguno de ellos elimina la alteración que se realiza al equipo.

Ilustración 7-8: Otros dispositivos como la Mobile Xperience Cam



## 7.6 Problemas identificados

Las herramientas antes mencionadas son bastantes útiles al momento de evaluar aplicaciones para dispositivos móviles, pero definitivamente se les puede sacar mayor provecho o, sencillamente, ser mejoradas. El equipo ocupado en las pruebas de campo del estudio antes presentado, presentaba las desventajas de ser incomodo (andar con una mochila y muy probablemente pesada) y el de ocupar mucho tiempo en su preparación para

el comienzo de la prueba. Por otra parte, no contaba con un mecanismo para capturar eventos de forma automática, para que los resultados obtenidos tuviesen más peso.

Por otra parte, la Mobile Xperience Cam, o alguno de sus similares, ayudan a poder capturar la interacción de los dedos con el teléfono y lo que va ocurriendo en la pantalla de la terminal en video, lo cual entrega información valiosa para su análisis. Generalmente estas cámaras son utilizadas dentro de un laboratorio y al momento de realizar estas pruebas, no se cuenta con una herramienta capaz de capturar otro tipo de información relevante para la evaluación.

El uso del emulador de interfaz móvil tiene un gran potencial, gracias a la herramienta UserZoom, en lo que es la automatización de la captura de datos y del análisis de dicha información, pero su desventaja es que esta diseñada para poder trabajar interfaces, donde el usuario no tiene contacto directo con el dispositivo móvil. Es decir, en estas pruebas no existe una real interacción entre el usuario y el dispositivo, solo con su interfaz grafica.

## Capítulo 8

### Solución Propuesta

Considerando las ventajas y desventajas que presentan las herramientas antes mencionadas, y teniendo en cuenta la falta de un software que apoye las pruebas de usabilidad de aplicaciones para dispositivos móviles, se propone desarrollar una herramienta software capaz de apoyar las pruebas de usabilidad de este tipo de software.

Según Melody Ivory en su documento titulado *The State of the Art in Automating Usability Evaluation of User Interfaces*, existen cuatro niveles de automatización que una herramienta para el apoyo de evaluaciones de usabilidad puede llegar a alcanzar: sin automatización, automatización en la captura, automatización en el análisis y automatización en las críticas.

Esta herramienta tiene como objetivo alcanzar el tercer nivel de automatización, es decir, automatización en el análisis.

La herramienta se compondrá de dos componentes importantes:

Una API, que entregue una interfaz a la aplicación que se desea evaluar, para que esta informe de los eventos que se producen. Esta solo se encargará enviar a otra aplicación los eventos informados por la aplicación.

Una aplicación capaz de recepcionar dichos eventos, ordenarlos, guardarlos y procesarlos. Toda la carga de procesamiento será realizado en este aplicación.

La API será la encargada de automatizar el proceso de captura, mientras que la aplicación en el computador, será la encargada de automatizar el proceso de análisis.

Esta herramienta reflejará su apoyo en las evaluaciones que sean del tipo Test de Usabilidad, donde un usuario experimental tenga contacto directo con la aplicación a evaluar.

Lo ideal para esta herramienta será trabajar de una manera que no sea intrusiva dentro del código de la aplicación que se va a probar, es decir, que no se tenga que modificar el programa a evaluar. Una forma de realizar esto, es crear una pequeña aplicación que trabaje en segundo plano dentro del dispositivo móvil y que sea capaz de realizar la captura de eventos. Esta idea no es factible por el hecho de que J2ME y Java en general, no permite que un proceso pueda interferir o manipular a otro proceso. Es por este motivo, que se plantea, por el momento, crear esta herramienta intrusiva para realizar el proceso de captura de los eventos y se deja para el final, y si el tiempo lo permite, intentar buscar una forma de realizar dicho proceso en forma no intrusiva.

Ya que en la actualidad los juegos para este tipo de dispositivos son cada vez más populares y se han vuelto bastantes rentables para sus desarrolladores, existen cada vez más en el mercado. Por este motivo, acotaremos esta herramienta en el sentido que el prototipo que se presente al final de la evaluación deberá, por lo menos, poder acoplarse a este tipo de aplicaciones. Además el juego debe encontrarse en etapas de desarrollo, que es donde más necesaria son las pruebas de usabilidad, para poder contar con el código fuente de la aplicación y así poder integrar fácilmente la API.

## **8.1 Estudio de Factibilidad**

### **8.1.1 Factibilidad Técnica**

En esta solución el punto más riesgoso es la implementación de la comunicación entre el dispositivo móvil y el computador. Por este motivo se ha tomado la decisión de trabajar con Bluetooth.

La tecnología de comunicación inalámbrica Bluetooth, es inferior a la tecnología Wi-Fi y superior a la tecnología de infrarrojos. Bluetooth está diseñada para dispositivos de bajo consumo y para conexiones de corta distancia (10 metros). A diferencia de los infrarrojos, Bluetooth es omnidireccional y tiene un mayor ancho de banda (máximo 720 kb/s de capacidad bruta).

Bluetooth es una tecnología ideal para la conexión de dispositivos de bajas prestaciones (móviles, cámaras de fotos, auriculares manos libres, impresoras,...).

Uno de los mayores ámbitos de utilización de Bluetooth es sin duda los teléfonos móviles. Cada vez es más común encontrar terminales móviles con soporte para Java y Bluetooth. Por este motivo se ha desarrollado un JSR (que es una especie de pequeña API para J2ME) que cubre la necesidad de programar en Java con esta tecnología y fue identificado como JSR-82.

En estos momentos se cuenta con un dispositivo USB-Bluetooth para poder potenciar un computador con esta tecnología y que trabaja con el Stack Bluetooth Bluesoleil. Además se cuenta con un dispositivo móvil que cuenta con Bluetooth y con la API JSR-82 para poder utilizarla.

Para poder trabajar con esta tecnología, nos encontramos con Bluecove. La cual es una API que implementa la JSR-82 de J2ME, pero para J2SE, ayudando a la conexión J2SE y J2ME de una manera más transparente. En estos momentos se encuentra en su etapa Beta y trabaja solo con solo algunos Stack Bluetooth (Windows Bluetooth, WIDCOMM y BlueSoleil). Este producto se encuentra bajo licencia LGPL (Lesser General Public License).

Ya que las evaluaciones que se desean apoyar están dirigidas a juegos para dispositivos móviles, la tecnología Bluetooth presenta algunos inconvenientes como por ejemplo: Si se desea respaldar en video lo que ocurre en la pantalla del dispositivo durante la ejecución del juego y cantidad de cuadros por segundos requerida es aproximadamente superior a 10, la transferencia de dichas imágenes puede hacer que la aplicación evaluada se vea afectada. Por este motivo esta solución quedará limitada a esta condición.

En conclusión es posible señalar que la solución propuesta si es factible de realizar aún considerando sus limitaciones.

### **8.1.2 Factibilidad Económica**

El proyecto cuenta con el respaldo de la Escuela de Ingeniería Informática en cuanto a los costos que presenta el desarrollo completo. Esto es válido en el ámbito de la construcción

del prototipo, considerando el uso de las dependencias de la Escuela de Ingeniería Informática tanto en equipos, y software asociado al proyecto. Lo anterior permite el desarrollo completo del proyecto desde el punto de vista económico.

Aunque si se llevara a cabo este proyecto en un ambiente no académico, los costos de adquisición serian:

Tabla 8-1: Estimación de costos del proyecto

<i>ITEM</i>	<i>VALOR(pesos)</i>
Notebook Dell Inspiron 1501 (AMD64, 1GB RAM, 80GB HD)	\$ 500.000
3 meses de Mano de obra (desarrollador Java)	\$ 960.000
Teléfono Celular con dispositivo BlueTooth y soporte a J2ME  (con implementación de JSR82 y MIDP 2.0 o superior)	\$ 100.000
Unidad BlueTooth USB	\$ 10.000
Licencia para IDE Netbeans 5.5	LICENCIA LIBRE
Licencia para Bluecove	LICENCIA LIBRE
Licencia para Sun Java Wireless Toolkit	LICENCIA LIBRE
Licencia Para Rational Rose	LICENCIA ESTUDIANTE
<b>TOTAL</b>	<b>\$ 1.570.000</b>

Los precios tanto del hardware como del software implementado, son referenciales a los sitios oficiales de las empresas (Dell, Motorola).

### **8.1.3 Factibilidad Legal**

La naturaleza del proyecto no presenta problemas de acuerdo a restricciones que presentan las licencias de las herramientas ocupadas ni con la legislación chilena en el ámbito de las tecnologías de la información. Dentro de las leyes involucradas se encuentran:

- **LEY-19223:** Tipifica figuras penales respecto a la informática. Fecha de publicación: 7.06.1993. Establece cuatro tipos penales en el ámbito informático: daño informático, espionaje informático, sabotaje informático, revelación y divulgación de datos.
- **LEY-19628:** Sobre protección de la vida privada. Fecha de publicación: 28.08.1999. Permite al titular de datos personales tener control sobre la información que le concierne, aún cuando ella es tratada por terceros en alguna actividad económica o por el Estado.

## **8.2 Paradigma, Metodología y Herramientas**

### **8.2.1 Alternativas de paradigma**

Ahora se presentan algunos de los paradigmas más ocupados en el desarrollo de software y que se tuvieron en cuenta al momento de elegir el paradigma que guiará el proceso de desarrollo. Otros paradigmas se mantuvieron al margen de esta proceso por no tener experiencia con ellos.

#### **8.2.1.1 Modelo Cascada**

El paradigma clásico. Plantea el desarrollo de software en una serie secuencial de etapas (Análisis, Diseño, Implementación, Pruebas, Mantenimiento), presume que los resultados de cada etapa permanezcan sin grandes modificaciones una vez realizados. Al no tener los requerimientos claramente definidos desde el comienzo del proyecto, presenta una gran desventaja.

### **8.2.1.2 Proceso Unificado**

Este es un paradigma de desarrollo iterativo e incremental, conducido por los Casos de Uso y basado en la Arquitectura del software. Una de sus principales ventajas es que puede ser amoldado de acuerdo a la organización o al proyecto al cual se va a aplicar. Ya que este proyecto se va a llevar a acabo por un solo desarrollador quien generará toda la documentación y la implementación del prototipo, puede ser un problema en términos de tiempo y esfuerzo. La empresa Rational a propuesto su propio Proceso Unificado que deriva del genérico y que es uno de las más conocidos y utilizados.

### **8.2.1.3 Prototipos.**

El desarrollo de Prototipos es un paradigma el cual permite capturar de una manera iterativa los requerimientos del cliente mediante la construcción iterativa del producto. De esta manera, el producto adquiere flexibilidad ante cambios por parte del cliente, exploración de diferentes alternativas o cambios debido a otros factores externos.

El desarrollo de prototipos tiene dos posibilidades, una es trabajar con prototipos desechables, que se utiliza para crear prototipos para mostrar al cliente el como se implementarían sus requerimientos en el futuro sistema, la segunda es trabajar con prototipo evolutivo, donde se comienza la iteración con un prototipo básico que contempla los requerimientos que más se manejan y en cada iteración, donde se van probando los prototipos, se va profundizando en los demás requerimientos y agregándolos en el prototipo.

## **8.2.2 Paradigma de desarrollo: Desarrollo basado en prototipo evolutivo**

Para el desarrollo de este proyecto se ha decidido utilizar el paradigma de prototipo evolutivo, ya que presenta la oportunidad de comenzar a trabajar rápidamente con solo algunos de los requerimientos finales. Además, este paradigma presenta la posibilidad de tener un adelanto funcional en las instancias de presentación de avances de este proyecto.

El trabajar con prototipos evolutivos, aprovechamos al máximo el trabajo realizado anteriormente, siempre y cuando, el desarrollador tenga presente que se van a ser

constantemente modificaciones del software y por lo tanto mantenga el código lo más ordenado y organizado posible y bien comentado.

### **8.2.3 Metodología de Programación: Programación Orientada a Objetos**

El trabajar con orientación a objetos consiste en entender un programa como un conjunto de unidades (objetos) que interactúan entre si mediante la recepción, proceso y envío de mensajes. Cada objeto posee una responsabilidad diferente lo cual le otorga flexibilidad y una facilidad de mantención a los programas. La elección de este paradigma esta influenciada en gran parte por el amplio soporte existente, desde lenguajes de programación hasta herramientas de análisis y diseño de software.

Otra forma sería realizar el modelado y la programación de la herramienta con métodos estructurados, pero no se cuenta con mucha experiencia con este tipo de procesos.

### **8.2.4 Herramientas a Utilizar**

#### **8.2.4.1 Modelado y diseño**

UML: El lenguaje seleccionado para el modelado orientado a objetos es UML (Unified Modelling Language) debido al estándar internacional que posee y al amplio soporte que se ofrece.

Rational Rose: La elección de Rational Rose como herramienta CASE para el análisis y diseño en UML del proyecto, se debe a la experiencia que se tiene trabajando ella, para así ahorrar tiempo de aprendizaje en otras herramientas similares.

#### **8.2.4.2 Lenguaje de Programación**

Java 2 Micro Edition: Se ha decidido trabajar con este lenguaje de programación, por la mayoría de los juegos para teléfonos celulares en la actualidad están desarrollados bajo este lenguaje, lo que facilitaría la integración de la herramienta al tipo de aplicaciones que se desea evaluar. Además de contar con la experiencia suficiente para poder utilizar este lenguaje y ahorrar tiempo de aprendizaje de otro lenguaje. Por otra parte, es necesario el

aprender a utilizar JSR-82 de J2ME, la cual es necesaria para poder trabajar con la tecnología Bluetooth.

Java Standart Edition: Este lenguaje es el padre de J2ME y es compatible con esta última, lo que facilitaría evitar muchos problemas de compatibilidad. Por otro parte, su alta portabilidad evitará depender de un solo sistema operativo. Además de la experiencia adquirida a través de la carrera en la utilización de este lenguaje. Para poder trabajar con Bluetooth y J2SE se cuenta con la API Bluecove que implementa la API JSR-82 para J2SE.

## 8.3 Requerimientos

A continuación se presentan los requerimientos que debe cumplir la herramienta.

### 8.3.1 Requerimientos funcionales:

1. *Aplicación móvil* – requerimientos para la API estará en el dispositivo móvil.

1.1. Debe informar los eventos que ocurran por la interacción con el usuario, como lo es el presionado de teclas y cambios de estado, todo el tiempo en que se este ejecutando la aplicación.

1.2. Debe poder capturar en video lo visualizado en la pantalla.

2. *Evaluador* – requerimientos para la aplicación que se encuentra en el computador.

2.1. Captura

2.1.1. Debe poder iniciar la captura de información, además de pararla cuando desee.

2.1.2. Debe poder informar de algunos eventos referentes a la prueba misma (cumplimientos de tareas, frustraciones, etc.).

2.2. Gestión de Evaluaciones

2.2.1. Se deberá poder ingresar información referente a la evaluación y a cada prueba de la evaluación.

2.2.2. El evaluador podrá guardar toda la información capturada para la evaluación.

2.2.3. En una evaluación ya existente se debe poder agregar más pruebas.

2.2.4. Se debe poder reproducir una prueba y tener control sobre la reproducción.

2.2.5. Se debe poder exportar el video capturado a un formato estándar.

### 2.3. Análisis de Evaluaciones

2.3.1. Se debe poder generar un resumen de las pruebas realizadas en una evaluación.

2.3.2. Se debe poder generar estadísticas necesarias para la facilidad de análisis de una evaluación.

2.3.3. Se debe poder generar un documento XML ya sea de toda la evaluación o de alguna prueba en particular.

## **8.3.2 Requerimientos no funcionales:**

### ***1. Desempeño:***

1.1. Garantizar que la información capturada es fidedigna y no se ve alterada en el proceso de captura.

1.2. La captura de la información debe ser en tiempo real.

1.3. El desempeño de la aplicación a evaluar, no debe verse afectada por la herramienta.

1.4. La conexión entre el dispositivo móvil que contiene la aplicación a evaluar y el dispositivo que contiene la aplicación receptora, debe ser inalámbrica.

### ***2. Integración***

2.1. Se debe buscar o establecer un estándar al cual poder exportar los eventos capturados en las pruebas. Para asegurar compatibilidad con otras herramientas de usabilidad.

### ***3. Mantenibilidad***

3.1. La herramienta deberá estar documentada, cada uno de los componentes de software que forman parte de la solución propuesta deberán estar debidamente documentados tanto en el código fuente como en los manuales de usuario.

# Capítulo 9

## Modelado de la herramienta

### 9.1 Identificación de actores

Aplicación Móvil: Es la aplicación para dispositivos móviles a la que se le estará evaluando su usabilidad. Es uno de los ejes principales en una prueba de usabilidad.

Usuario Experimental: Es el segundo eje principal en una prueba de usabilidad. Es el encargado de interactuar directamente con la Aplicación Móvil, para así poder encontrar problemas de usabilidad que pudiese presentar la dicha Aplicación.

Evaluador: Es el moderador de la prueba de usabilidad, es el encargado de guiar al Usuario Experimental a través del desarrollo de las tareas.

### 9.2 Casos de uso

En esta sección se mostraran todos los casos de usos que se presentan según los requerimientos obtenidos. Muchos de los casos de uso son comprensibles por la mayoría de los que intervienen en un proceso de desarrollo. Por este motivo, no serán analizados en su profundidad. Como por ejemplo, en el Diagrama 2: Caso de uso: Gestionar Evaluación, por ‘Gestionar Archivo’ se comprende que es el caso de uso encargado de abrir, guardar, cerrar archivos, es decir, todo lo que tenga que ver con el manejo de los respaldos de las evaluaciones.

Ilustración 9-1: Caso de uso Alto Nivel

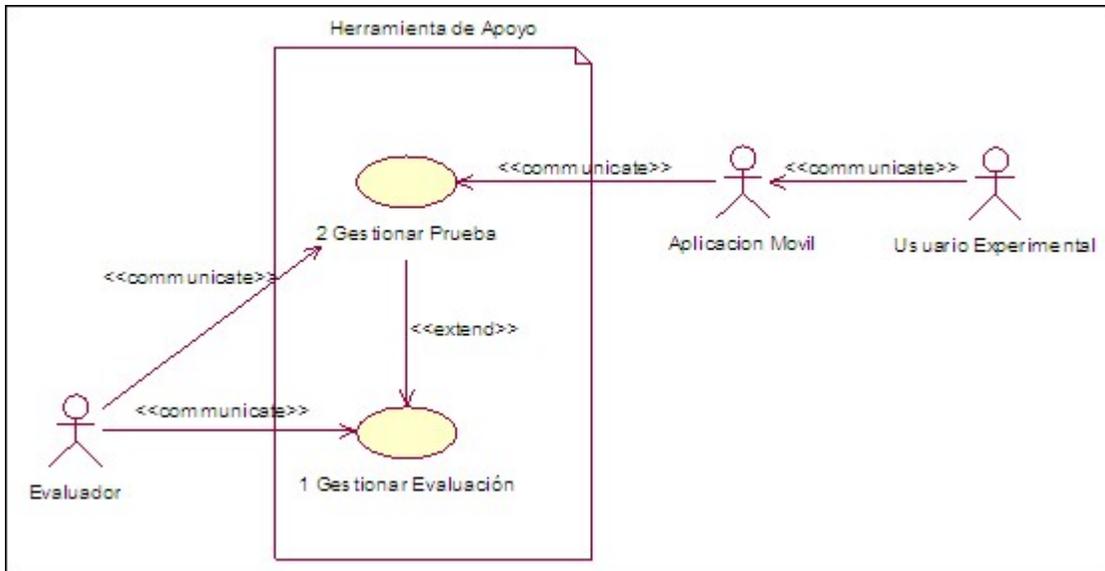
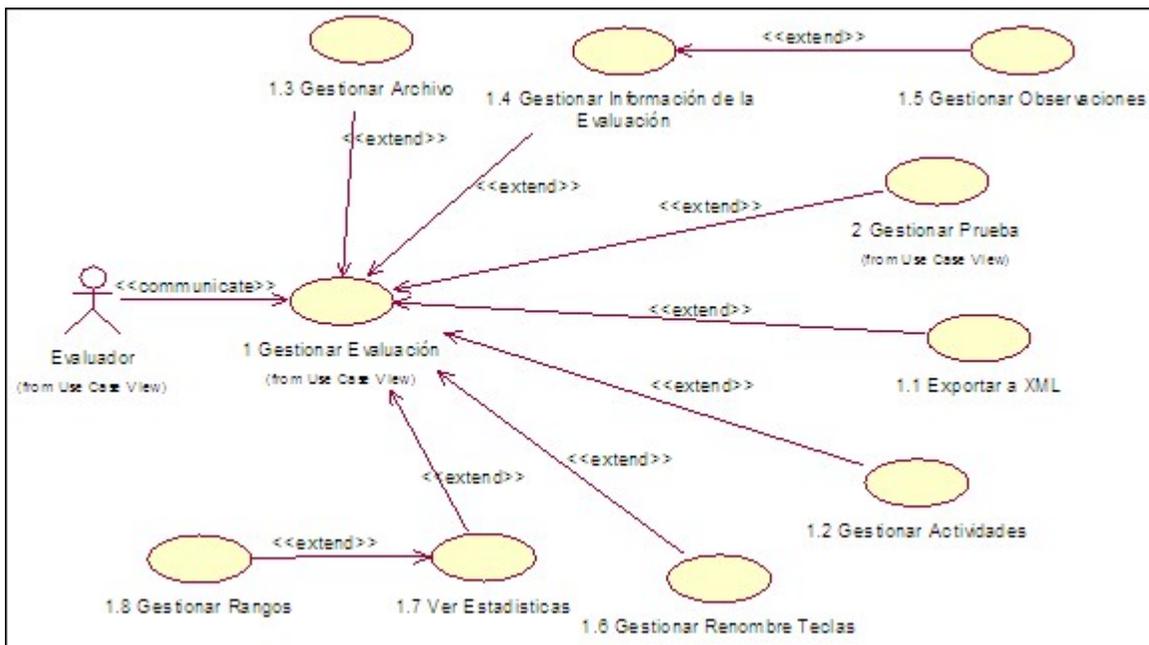
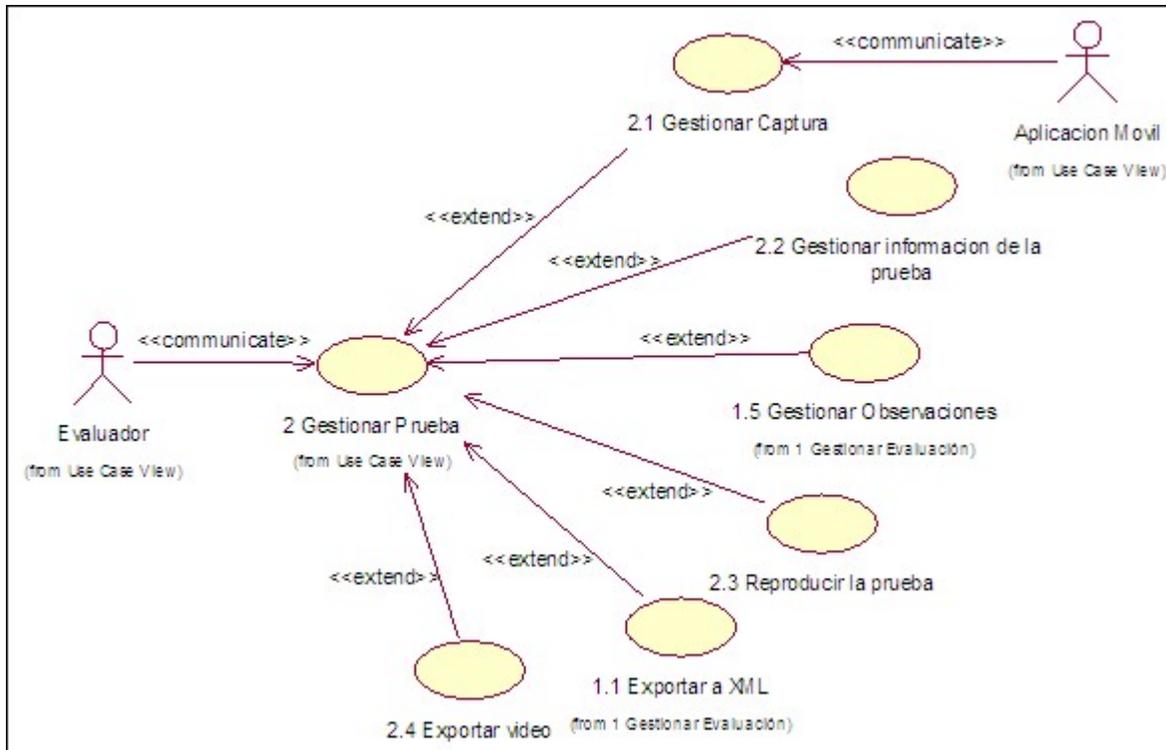


Ilustración 9-2: Caso de uso 1 - Gestionar Evaluación



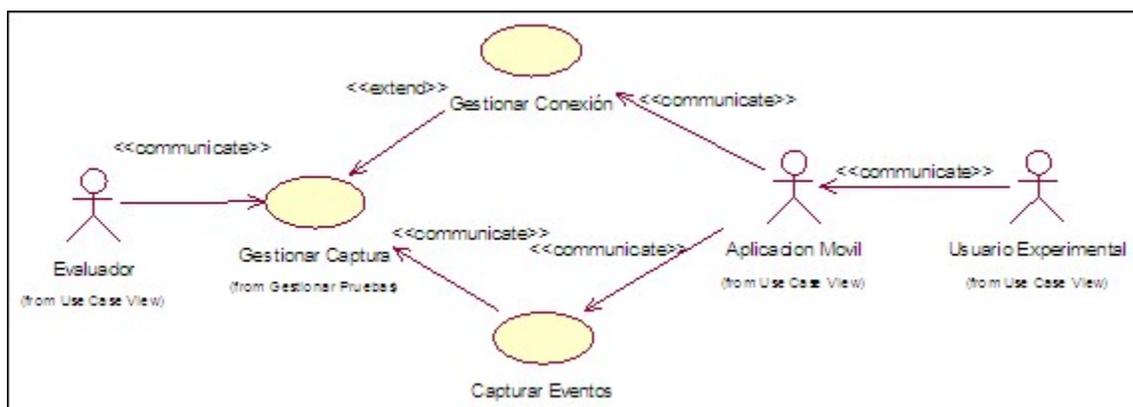
En la Ilustración 9-2, se puede apreciar los casos uso en que esta compuesto el caso de uso de Gestionar Evaluación. El caso de uso Gestionar Prueba se encuentra expandido en la Ilustración 9-3.

Ilustración 9-3: Caso de uso 2 - Gestionar Prueba



En la Ilustración 9-3, se aprecian los casos de uso que componen el caso de uso Gestionar Prueba. El más crítico de estos casos de uso, es el Gestionar Captura que tiene que con la gestión de la conexión inalámbrica con el dispositivo inalámbrico y el envío y recepción de los eventos capturados.

Ilustración 9-4: Caso de uso 2.1 - Gestionar Captura



Por último, en la Ilustración 9-4, se puede ver los casos de uso que están involucrados con el caso de uso Gestionar Captura y las relaciones que tienen los actores con estos. En este

modelado se ha considerado a la aplicación que se desea evaluar como un actor que a través de la API se comunicará con la aplicación receptora.

### 9.3 Casos de uso extendido

En los siguientes cuadros se darán a conocer los casos de uso extendido de los casos de uso más importantes o más críticos para conseguir el objetivo de este proyecto.

<i>ID</i>	<i>Exportar a XML</i>
<i>Nivel</i>	<i>1.1</i>
<i>Descripción</i>	<i>Este caso de uso se encargar de exportar una evaluación o una prueba a un documento XML.</i>
<i>Iniciador</i>	<i>Actor Evaluador</i>
<i>Secuencia</i>	<i>1 El actor indica al sistema que desea obtener el documento XML de la evaluación o prueba Actual.</i>
	<i>2 El sistema consulta al actor la ubicación donde se desea guardar el documento XML.</i>
	<i>3 El sistema genera el documento XML</i>
	<i>4 El sistema entrega dicho documento al usuario.</i>
<i>Excepciones</i>	<i>No tiene</i>
<i>Importancia</i>	<i>Importante</i>

Tabla 9-1: Caso de uso extendido - 1.1 Exportar a XML

Tabla 9-2: Caso de uso extendido - 1.2 Gestionar Actividades

<i>ID</i>	<i>Gestionar Actividades</i>
<i>Nivel</i>	<i>1.2</i>
<i>Descripción</i>	<i>Este caso de uso se encargar agregar, eliminar y modificar el listado de actividades asociado a una evaluación. Las actividades se identifican con descripción, tiempo máximo, criterio de éxito y criterio de error.</i>
<i>Iniciador</i>	<i>Actor Evaluador</i>
<i>Importancia</i>	<i>Importante</i>

Tabla 9-3: Caso de uso extendido - 1.3 Gestionar archivo

<i>ID</i>	<i>Gestionar Archivo</i>
<i>Nivel</i>	<i>1.3</i>
<i>Descripción</i>	<i>Este caso se encarga de guardar un respaldo de la evaluación actual y abrir un respaldo de alguna evaluación que se haya realizado.</i>
<i>Iniciador</i>	<i>Actor Evaluador</i>
<i>Importancia</i>	<i>Importante</i>

Tabla 9-4: Caso de uso extendido - 1.4 Gestionar información de la evaluación

--	--

<i>ID</i>	<i>Gestionar información de la evaluación</i>
<i>Nivel</i>	<i>1.4</i>
<i>Descripción</i>	<i>Este caso se encarga de gestionar(modificar, mostrar, etc.) toda la información de la Evaluación, ya sea nombre del software, versión del software, tipo de evaluación, objetivo de la evaluación, etc.</i>
<i>Iniciador</i>	<i>Actor Evaluador</i>
<i>Importancia</i>	<i>Normal</i>

Tabla 9-5: Caso de uso extendido - 1.5 Gestionar observaciones

<i>ID</i>	<i>Gestionar observaciones evaluación</i>
<i>Nivel</i>	<i>1.5</i>
<i>Descripción</i>	<i>Este caso se encarga de gestionar (mostrar, agregar, eliminar, editar) las observaciones correspondientes tanto de una evaluación como de una prueba.</i>
<i>Iniciador</i>	<i>Actor Evaluador</i>
<i>Importancia</i>	<i>Normal</i>

Tabla 9-6: Caso de uso extendido - 1.6 Gestionar renombre de teclas

<i>ID</i>	<i>Gestionar renombre de teclas</i>

<i>Nivel</i>	<i>1.6</i>
<i>Descripción</i>	<i>Este caso de uso se encarga de gestionar (agregar, eliminar, editar), los renombres que el evaluador hace a las teclas para asimilar mejor los eventos de teclado capturados.</i>
<i>Iniciador</i>	<i>Actor Evaluador</i>
<i>Importancia</i>	<i>Importante</i>

Tabla 9-7: Caso de uso extendido - 2.2 Gestionar información de la prueba

<i>ID</i>	<i>Gestionar información de la prueba</i>
<i>Nivel</i>	<i>2.2</i>
<i>Descripción</i>	<i>Este caso se encarga de gestionar (modificar, mostrar, etc.) toda la información de una prueba, ya sea nombre del moderador entre otros.</i>
<i>Iniciador</i>	<i>Actor Evaluador</i>
<i>Importancia</i>	<i>Normal</i>

Tabla 9-8: Caso de uso extendido - 2.3 Reproducir la prueba

<i>ID</i>	<i>Reproducir la prueba</i>
<i>Nivel</i>	<i>2.3</i>

<i>Descripción</i>	<i>Caso que se encarga de controlar la reproducción de la prueba y sus eventos. Esto incluye iniciar, continuar, pausar, detener la reproducción.</i>
<i>Iniciador</i>	<i>Actor Evaluador</i>
<i>Importancia</i>	<i>Importante</i>

Tabla 9-9: Caso de uso extendido - 2.4 Exportar Video

<i>ID</i>	<i>Exportar Video</i>
<i>Nivel</i>	<i>2.4</i>
<i>Descripción</i>	<i>Caso de uso que se encarga de exportar, a un formato estándar de video, el video capturado de lo visualizado en la pantalla del dispositivo móvil.</i>
<i>Iniciador</i>	<i>Actor Evaluador</i>
<i>Importancia</i>	<i>Importante</i>

Tabla 9-10: Caso de uso extendido - 2.1.1 Gestionar la conexión

<i>ID</i>	<i>Gestionar la conexión</i>
<i>Nivel</i>	<i>2.1.1</i>
<i>Descripción</i>	<i>Caso que se encarga de realizar la conexión entre el dispositivo móvil y el computador.</i>

<i>Iniciador</i>	<i>Actor Evaluador</i>	
<i>Precondición</i>	<i>Debe existir una prueba que no haya pasado por la captura de sus eventos.</i>	
<i>Secuencia</i>	<i>Paso</i>	<i>Acción</i>
	<i>1</i>	<i>El Evaluador indica que desea realizar una captura.</i>
	<i>2</i>	<i>El sistema comprueba que la prueba actual no ha pasado por la etapa de captura.</i>
	<i>2a</i>	<i>Si la prueba ya a pasado por la etapa de captura, el sistema devuelve un mensaje al usuario con lo sucedido y se cancela este caso de uso.</i>
	<i>2b</i>	<i>Si la prueba no ha pasado por el proceso de captura, el sistema inicia el dispositivo Bluetooth.</i>
	<i>3</i>	<i>El Evaluador inicia la aplicación en el dispositivo móvil, dicha aplicación espera que otra aplicación se acople.</i>
	<i>4</i>	<i>El sistema busca dispositivos remotos y recupera información sobre sus servicios.</i>
	<i>5</i>	<i>El sistema muestra una lista de los dispositivos remotos que tienen el servicio necesario para acoplarse a ellos.</i>
<i>6</i>	<i>El Evaluador selecciona el dispositivo móvil con el que desea acoplar la aplicación.</i>	

	7	<i>El sistema establece la conexión con el dispositivo seleccionado</i>
	8	<i>La aplicación móvil sigue su curso normal y se inicia el caso de uso 'Capturar Eventos'</i>
<i>Excepciones</i>	<i>Paso</i>	<i>Acción</i>
	4	<i>Si el sistema no encuentra dispositivos con el servicio necesario, da la posibilidad al usuario de realizar una nueva búsqueda o abortar este caso de uso.</i>
	5	<i>Si el Evaluador no encuentra el dispositivo móvil con el que desea realizar la conexión, puede realizar una nueva búsqueda o puede abortar este caso de uso.</i>
<i>Importancia</i>	<i>Vital</i>	

Tabla 9-11: Caso de uso extendida - 2.1.2 Capturar Eventos

<i>ID</i>	<i>Capturar Eventos</i>
<i>Nivel</i>	<i>2.1.2</i>
<i>Descripción</i>	<i>Este caso de uso se refiere al proceso de la captura de los eventos producidos en el dispositivo móvil, enviarlos a la aplicación ubicada en el computador y guardadas en el mismo.</i>
<i>Iniciador</i>	<i>Actor Aplicación Móvil</i>

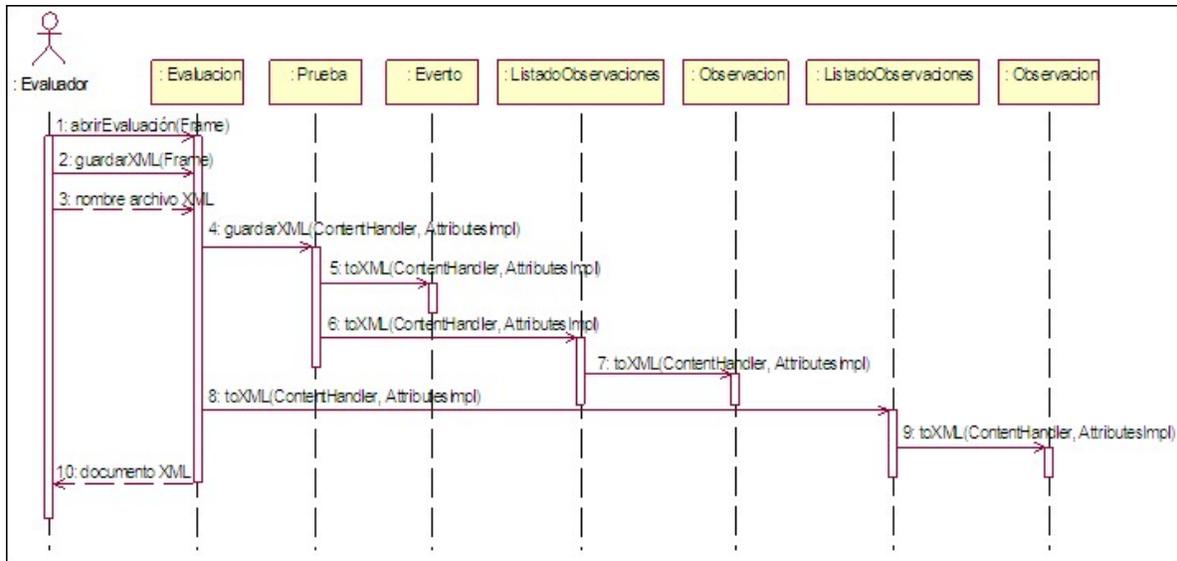
<i>Precondición</i>	<i>La aplicación en el dispositivo móvil y la aplicación receptora deben estar acopladas.</i>	
<i>Secuencia</i>	<i>Paso</i>	<i>Acción</i>
	<i>1</i>	<i>La aplicación móvil informa sobre un nuevo evento ocurrido durante la ejecución.</i>
	<i>2</i>	<i>Se encapsula el evento y es enviado a través de la conexión a la aplicación en el computador.</i>
	<i>3</i>	<i>El sistema recibe el evento y lo almacena en la prueba</i>
<i>Excepción</i>	<i>El Evaluador también puede informar sobre un evento de cumplimiento de actividad.</i>	
<i>Importancia</i>	<i>Vital</i>	

## 9.4 Diagramas de secuencia

En los siguientes diagramas muestran la interacción entre los distintos actores y las clases del software. Solo se muestran los diagramas más importantes o críticos del desarrollo de la herramienta.

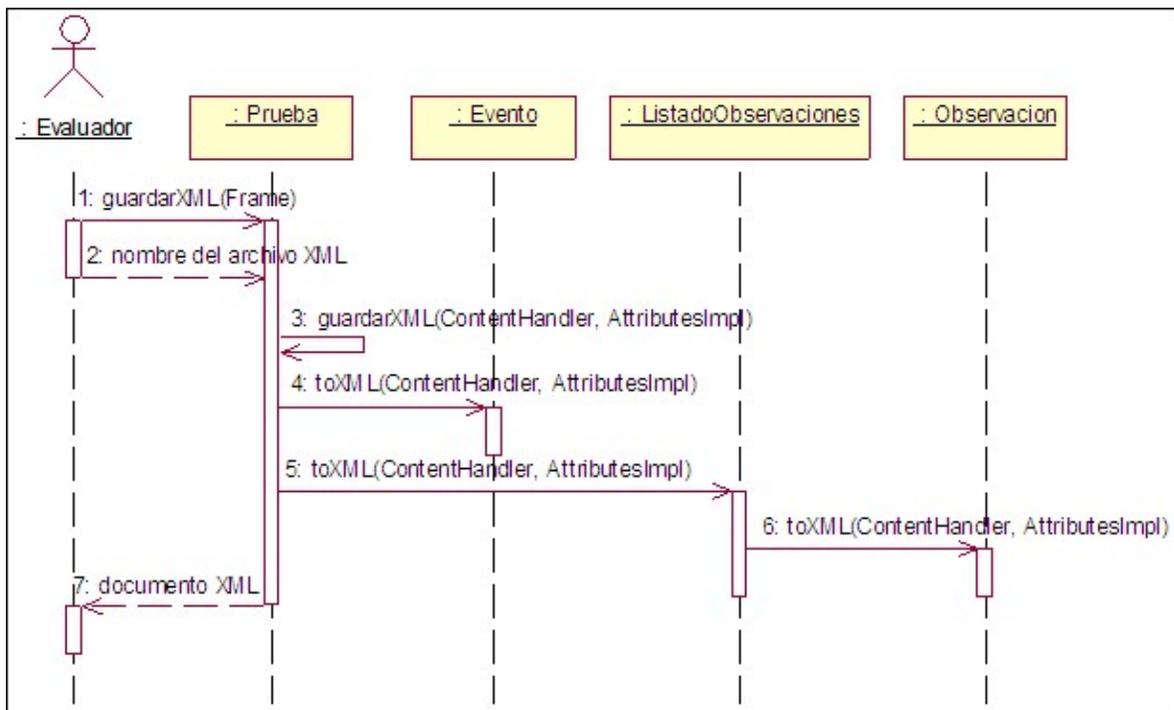
### 9.4.1 Caso de uso: 1.1 Exportar a XML

Ilustración 9-5: Diagrama de secuencia - Exportar una Evaluación a XML



En la Ilustración 9-5, se aprecia el proceso de exportación de una evaluación a un documento XML. En este diagrama se ve claramente como cada elemento importante en una evaluación, pruebas y sus eventos, observaciones, etc., son guardadas en un documento XML.

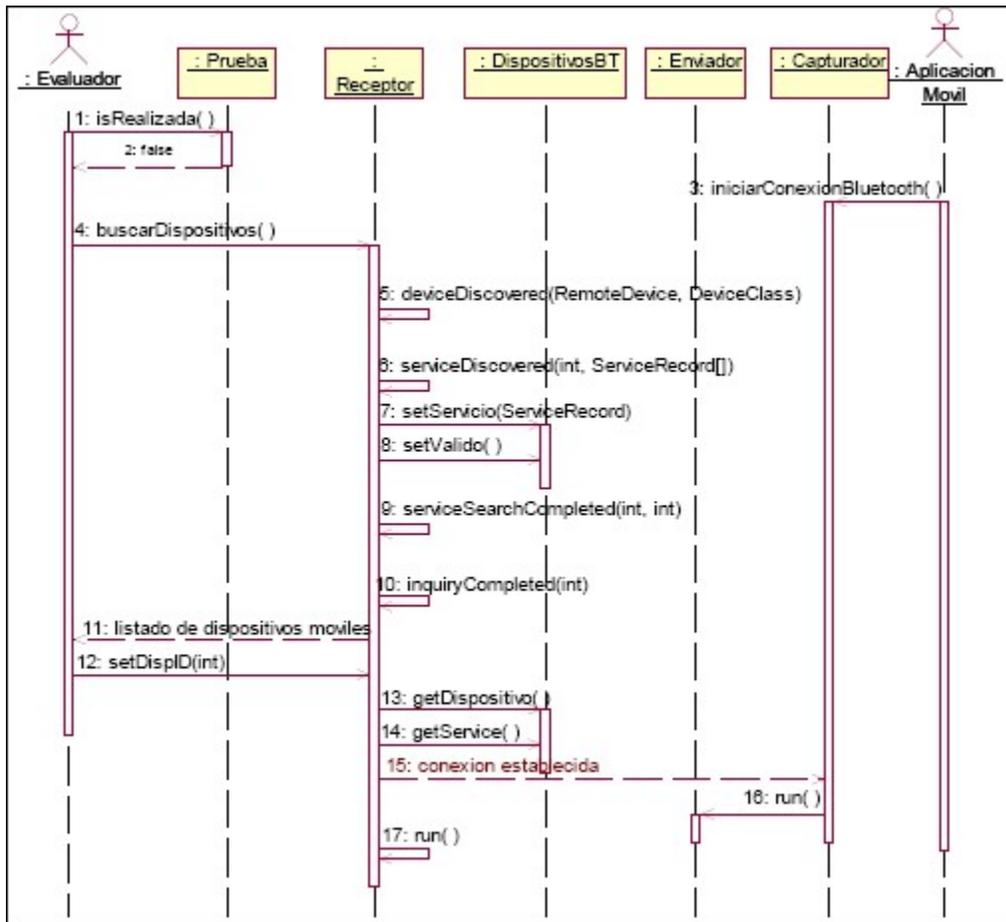
Ilustración 9-6: Diagrama de secuencia - Exportar una Prueba a XML



En la Ilustración 9-6, se puede apreciar el proceso de exportación de una prueba determinada a un documento XML. En este diagrama se ve claramente como cada elemento importante en una prueba, sus eventos, observaciones, etc., son guardadas en un documento XML.

### 9.4.2 Caso de uso: 2.1.1 Gestionar Conexión

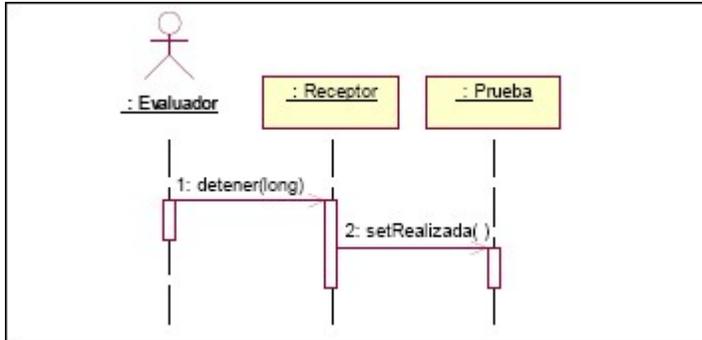
Ilustración 9-7: Diagrama de secuencia - Buscar aplicación remota, acoplar e iniciar captura



En la Ilustración 9-7, se puede apreciar el proceso para poder establecer la conexión Bluetooth entre la API que está en el dispositivo móvil y la aplicación que se encuentra en el computador. La mayoría de los métodos que se pueden apreciar en este diagrama, son parte de la interfaz entregada por la API de la especificación JSR-82 de Java. Al finalizar el proceso de conexión, la clase receptor estará recepcionando los eventos enviados en un

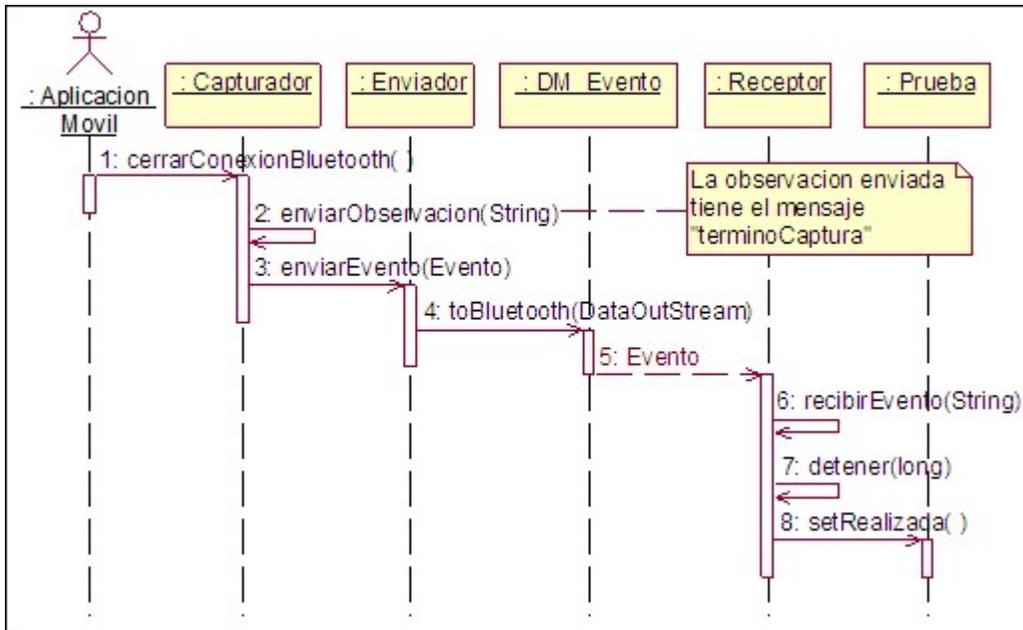
Thread (hilo de ejecución) distinto e independiente del normal. Lo mismo sucede con la clase Enviador en el dispositivo móvil.

Ilustración 9-8: Diagrama de secuencia - El Evaluador termina la captura



En la Ilustración 9-8, se ve como el Evaluador puede detener la captura de información. En este caso la aplicación en el computador cierra la conexión Bluetooth con el dispositivo móvil, termina su ejecución en el Thread independiente y establece la conexión como realizada.

Ilustración 9-9: La aplicación móvil termina la captura.



En la Ilustración 9-9, muestra como la aplicación en el dispositivo móvil puede llegar a terminar el proceso de captura. El Capturador envía un mensaje de término en forma de

evento observación a la aplicación en el computador y esta por su parte, realiza el proceso de término de captura.

### 9.4.3 Caso de uso: 2.1.2 Capturar Eventos

En los siguientes diagramas se pueden apreciar como es el proceso de captura de cada uno de los diferentes eventos.

Ilustración 9-10: Diagrama de secuencia - Capturar Evento Estado

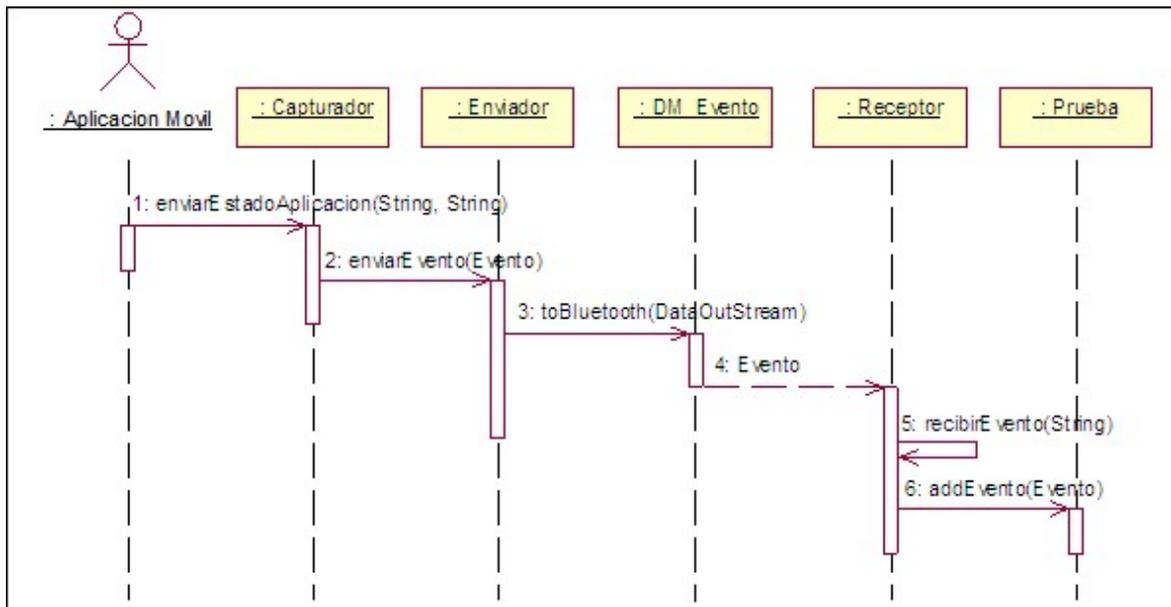


Ilustración 9-11: Diagrama de secuencia - Capturar Evento Observación

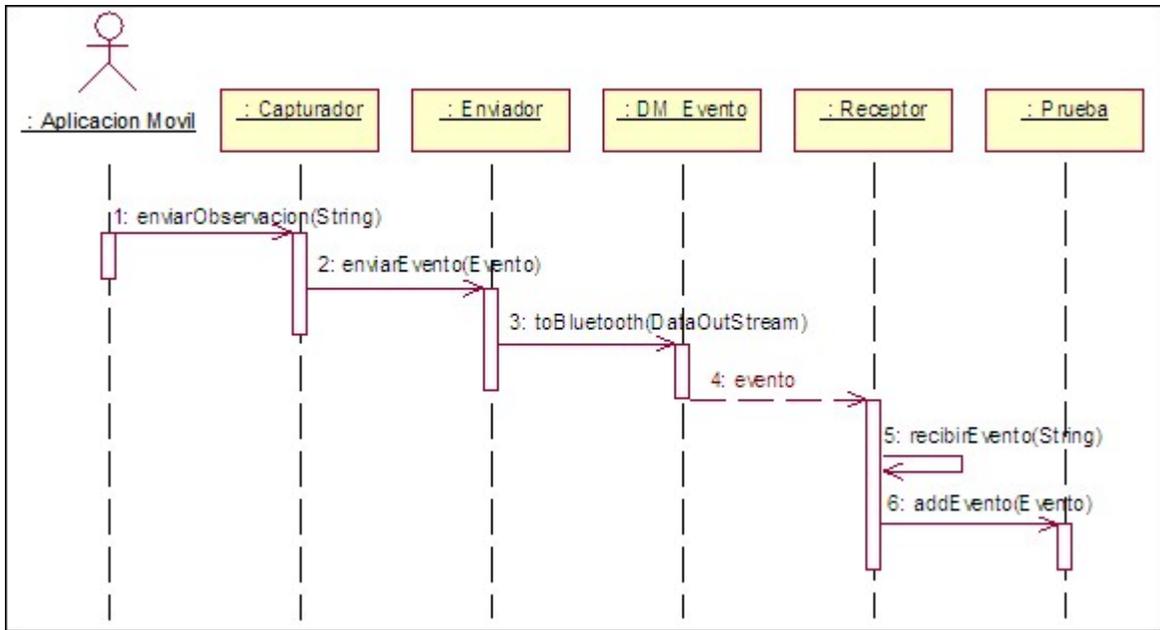


Ilustración 9-12: Diagrama de secuencia - Capturar Evento Teclado

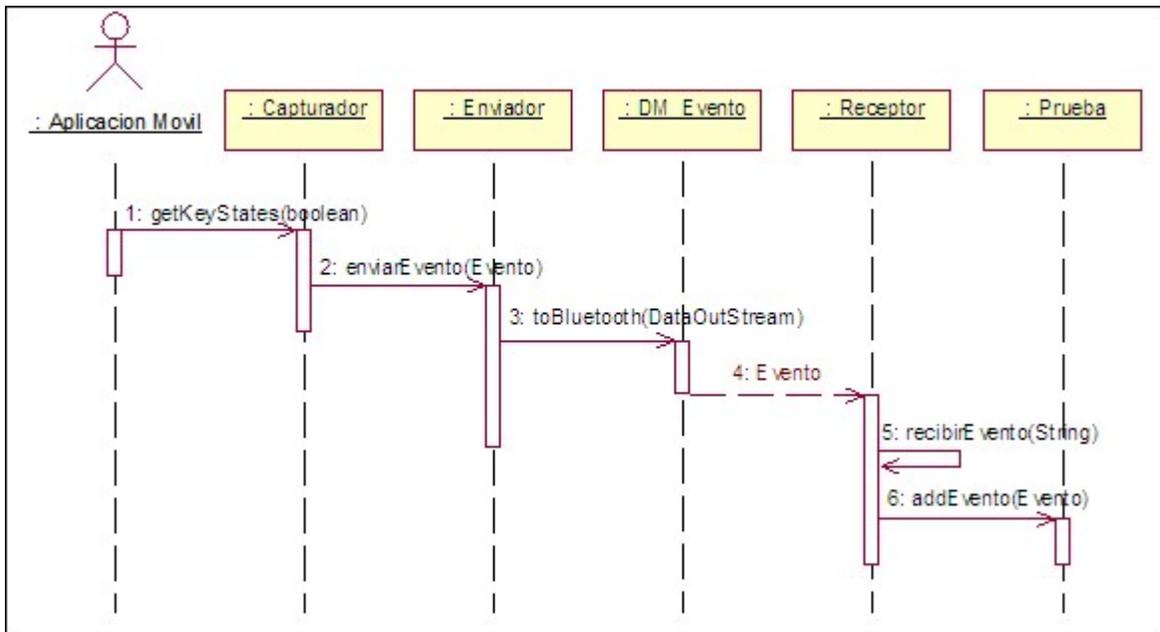


Ilustración 9-13: Diagrama de secuencia - El evaluador indica un cumplimiento de actividad

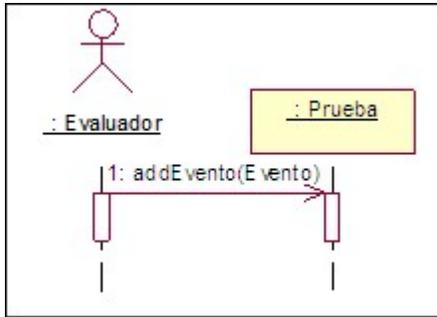
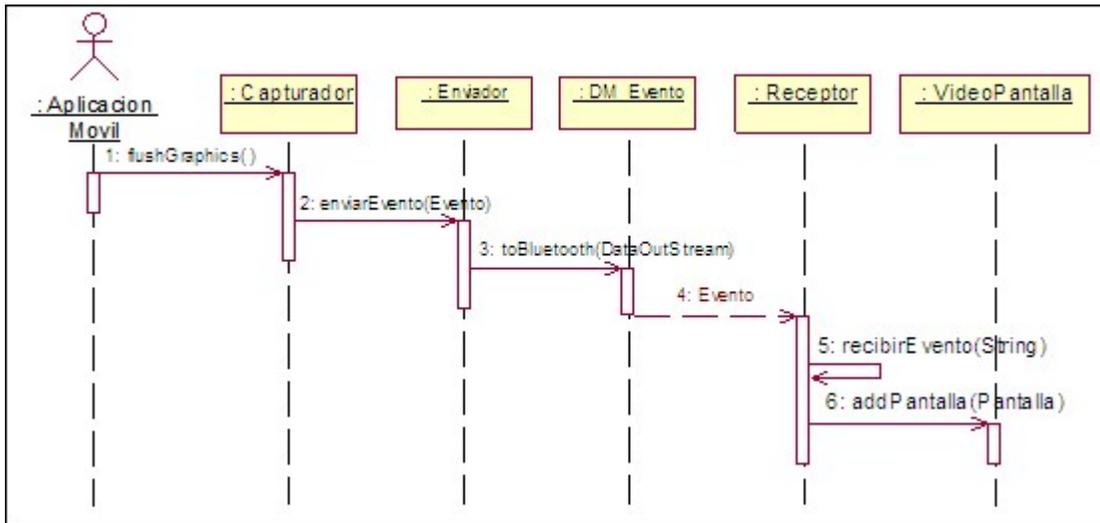


Ilustración 9-14: Diagrama de secuencia - Capturar Pantalla



En la Ilustración 9-14 se puede apreciar como es el proceso de captura de lo visualizado en la pantalla del dispositivo móvil. Cada vez que la aplicación móvil flushGraphics(), que es el método de la clase Canvas encargada de la actualización de lo visualizado en pantalla, la clase Capturador hará una copia de lo visualizado y lo enviará a la aplicación en el computador.

## 9.5 Diagramas de colaboración

### 9.5.1 Caso de uso: 1.1 Exportar a XML

Ilustración 9-15: Diagrama de colaboración - Exportar una evaluación a XML

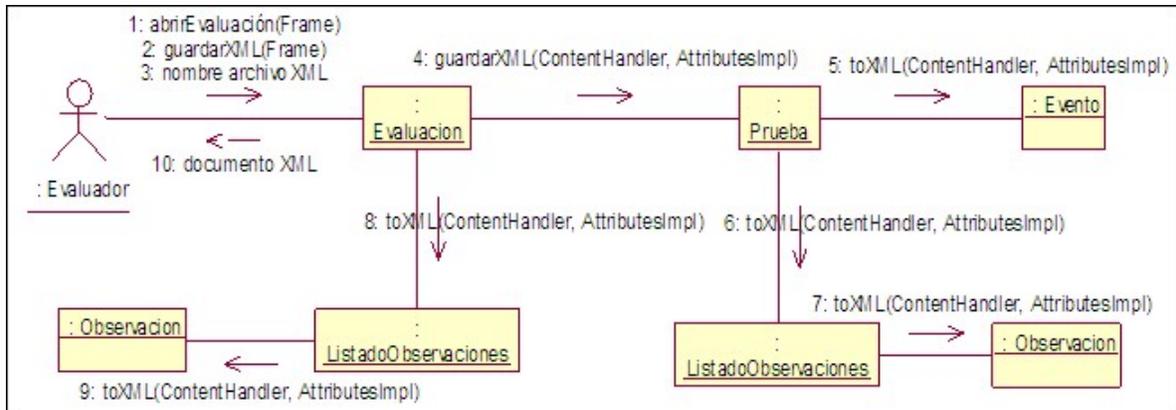
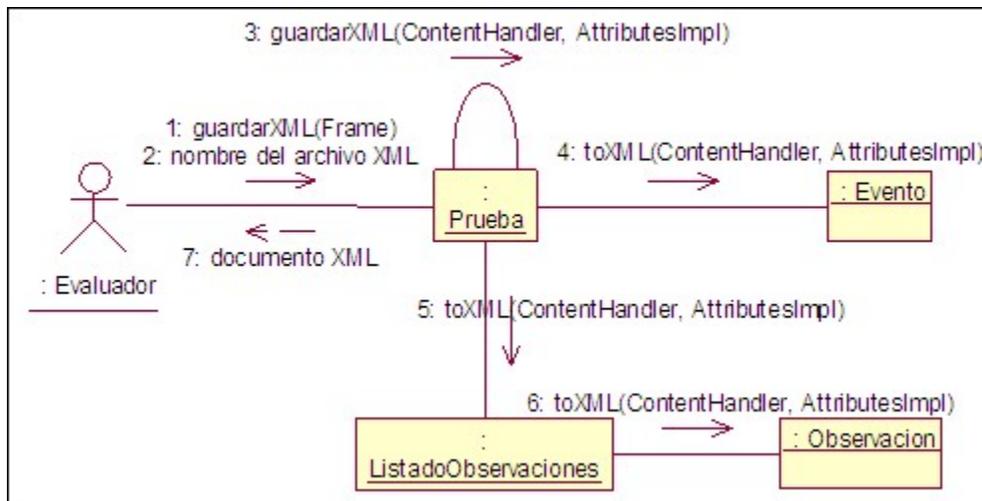


Ilustración 9-16: Diagrama de colaboración - Exportar una prueba a XML



### 9.5.2 Caso de uso: 2.1.1 Gestionar conexión

Ilustración 9-17: Diagrama de colaboración - Buscar aplicación remota, acoplar e iniciar captura

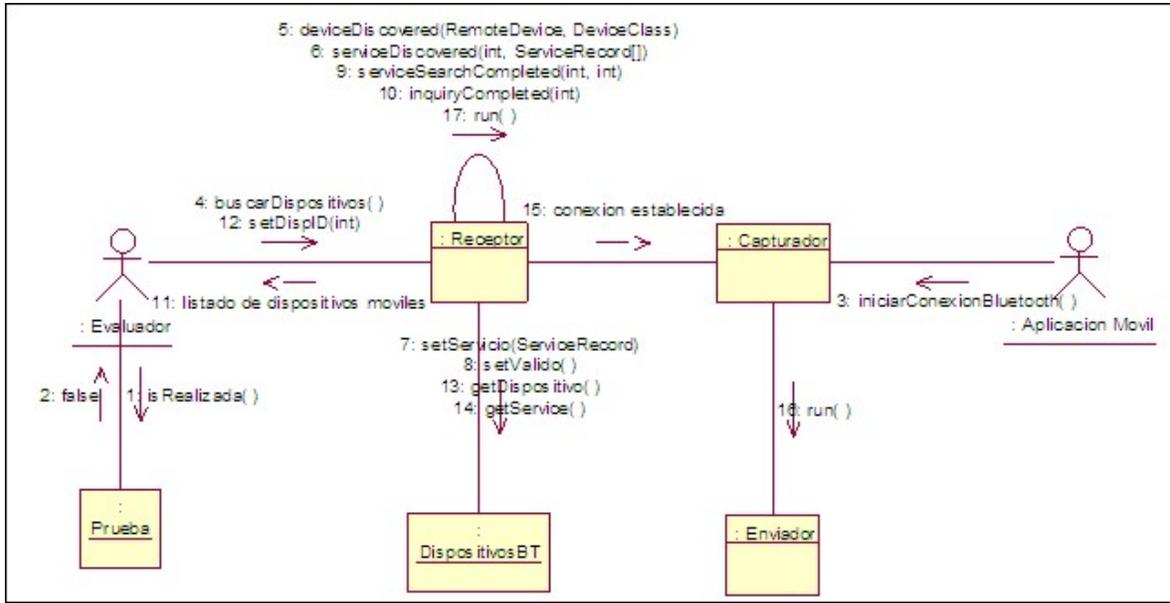


Ilustración 9-18: Diagrama de colaboración - El evaluador termina la captura

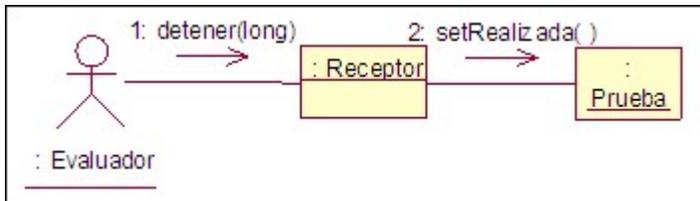
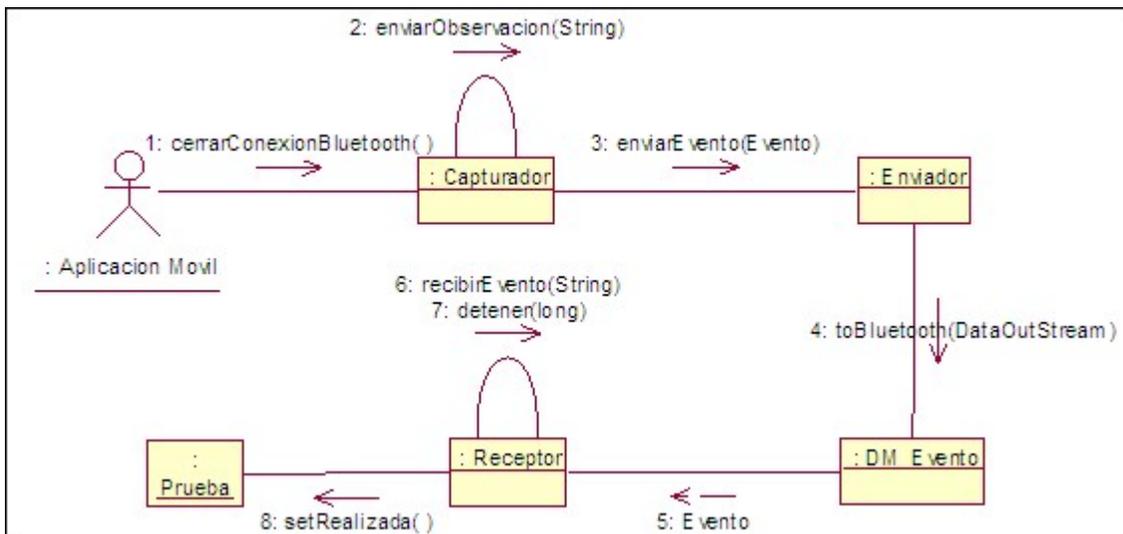


Ilustración 9-19: Diagrama de colaboración - La aplicación móvil termina la captura



### 9.5.3 Caso de uso: 2.1.2 Capturar Eventos

Ilustración 9-20: Diagrama de colaboración - Capturar Evento Estado

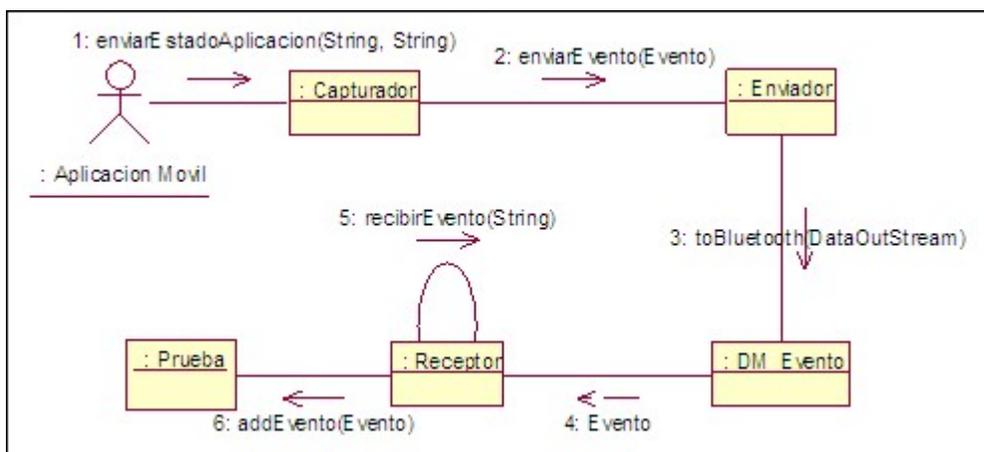


Ilustración 9-21: Diagrama de colaboración - Capturar Evento Observación

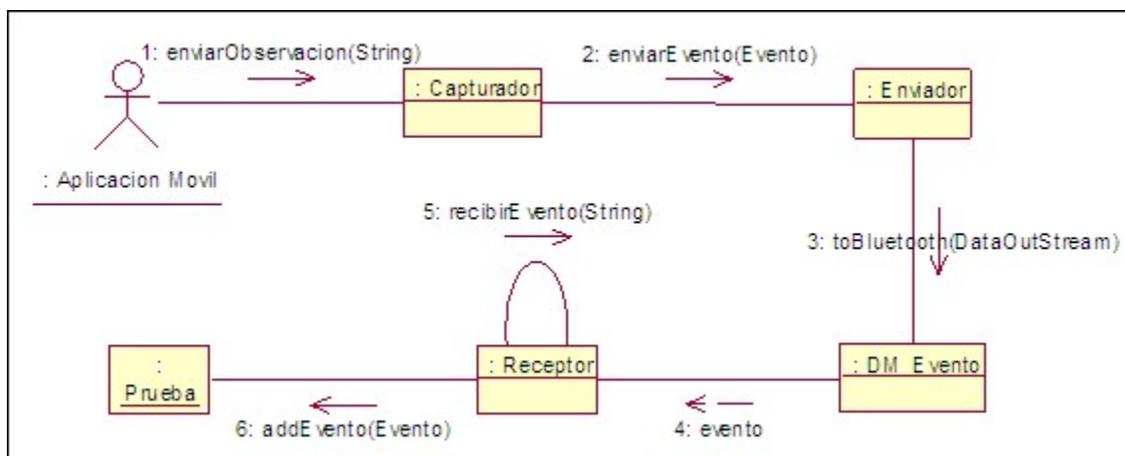


Ilustración 9-22: Diagrama de colaboración - Capturar Evento Teclado

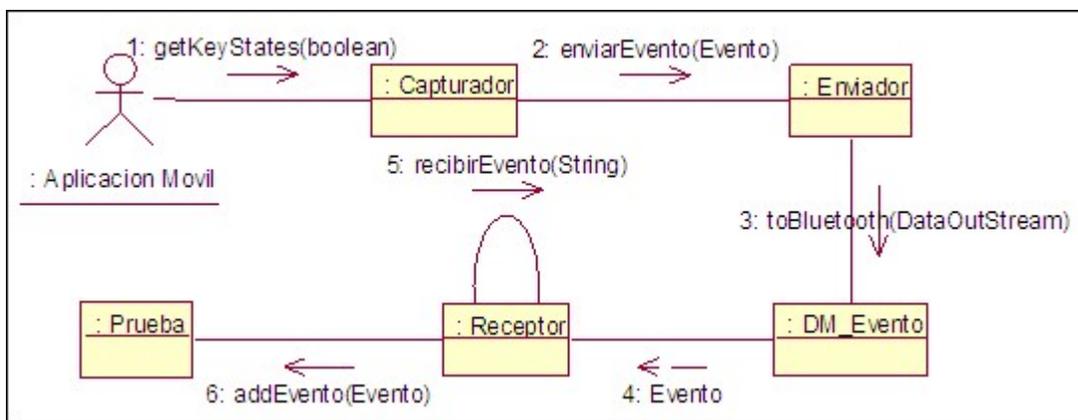
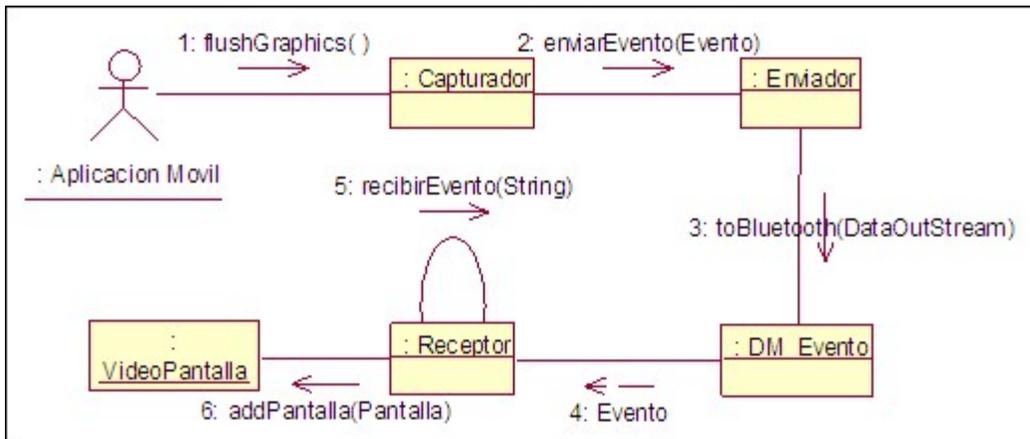


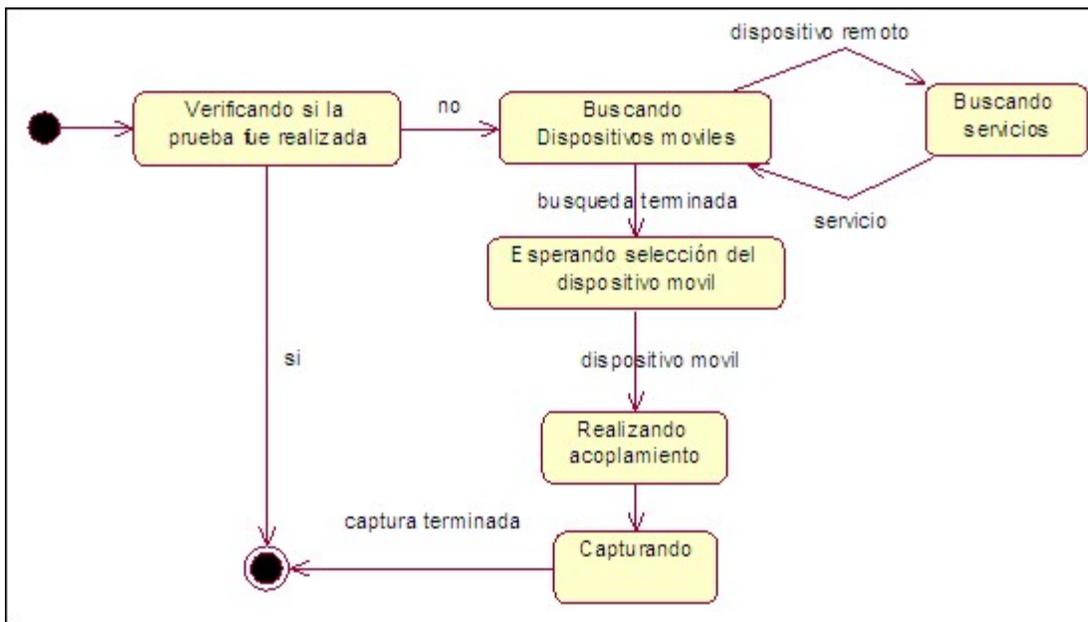
Ilustración 9-23: Diagrama de colaboración - Capturar Pantalla



## 9.6 Diagramas de estado

Solo se presenta el diagrama de estado de Buscar aplicación remota, acoplar e iniciar captura puesto que su funcionamiento esta vinculado a la API JSR82. Así de esta manera se comprende mejor y se puede implementar correctamente. Los demás diagramas no se realizaron, por el hecho de que los demás requerimientos han quedado claros.

Ilustración 9-24: Diagrama de estado - Buscar aplicación remota, acoplar e iniciar captura



## 9.7 Diagramas de componentes

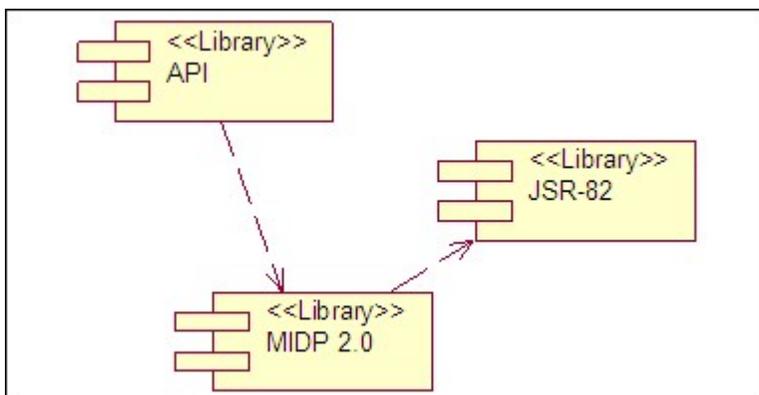
En los siguientes diagramas se puede apreciar los distintos componentes software y sus relaciones.

Ilustración 9-25: Diagrama de componentes - General



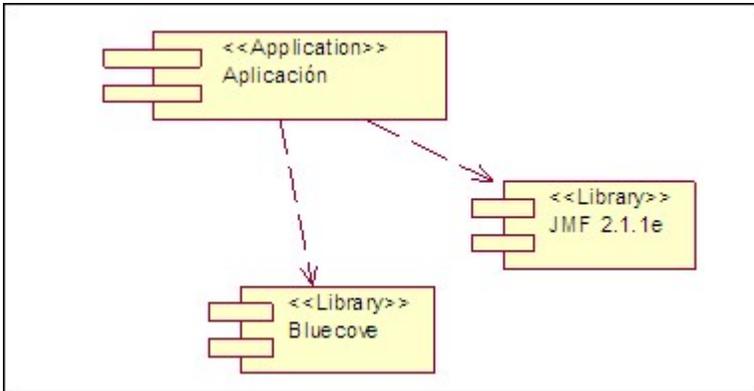
En la Ilustración 9-25, se puede apreciar los dos componentes principales de esta herramienta. El Capturador que se encontrará en el dispositivo móvil y el Receptor que se encontrará en el computador.

Ilustración 9-26: Diagrama de componentes - Paquete Capturador



En la Ilustración 9-26, se puede apreciar la composición del paquete Capturador. En este se ven el componente MIDP 2.0 que es la librería que entrega las clases principales para la realización de juegos para dispositivos móviles y la librería JSR-82 que es la encargada de gestionar la conexión Bluetooth.

Ilustración 9-27: Diagrama de componentes - Paquete Receptor

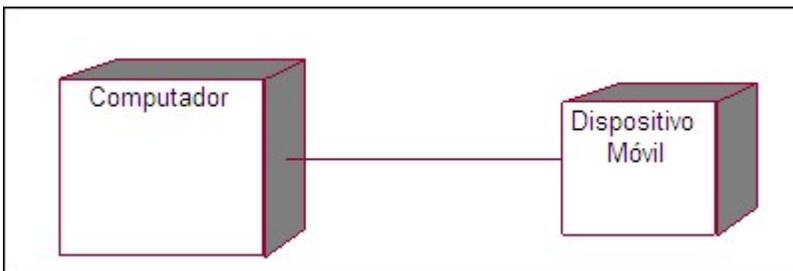


En la Ilustración 9-27 se puede apreciar la composición del paquete Receptor que se encuentra en el computador. La librería Bluecove es una implementación del estándar JSR-82 en J2SE para la gestión de conexión Bluetooth y la librería JMF es el Java Media Framework para el manejo de archivo de multimedia. Este último se ocupa para la exportación del video capturado a formato AVI.

## 9.8 Diagrama de distribución

En el siguiente diagrama se puede ver los elementos físicos de la herramienta. El computador que es donde se encontrará la aplicación receptora y el dispositivo móvil donde se encontrará la aplicación a evaluar y la API encargada de la captura de los eventos.

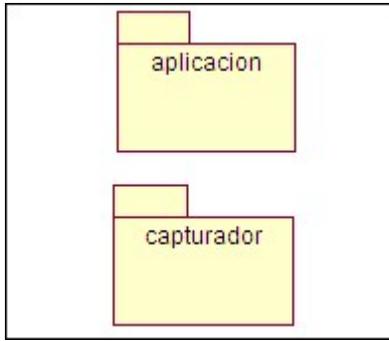
Ilustración 9-28: Diagrama de distribución



## 9.9 Diagrama de Clases

En los siguientes diagramas se muestran las clases que van a ser parte de la herramienta y que se van a hacer cargo de la lógica de la misma.

Ilustración 9-29: Diagrama de clases - General



En la Ilustración 9-30 se muestra una abstracción del diagrama de clases del paquete aplicación. En este diagrama solo se muestran las clases y sus relaciones. En la Ilustración 9-31 y la Ilustración 9-32 se muestra con más detalle este diagrama. En el paquete aplicación se ven las clases que están en la aplicación receptora en el computador.

Ilustración 9-30: Diagrama de clases – Paquete aplicación (abstracto)

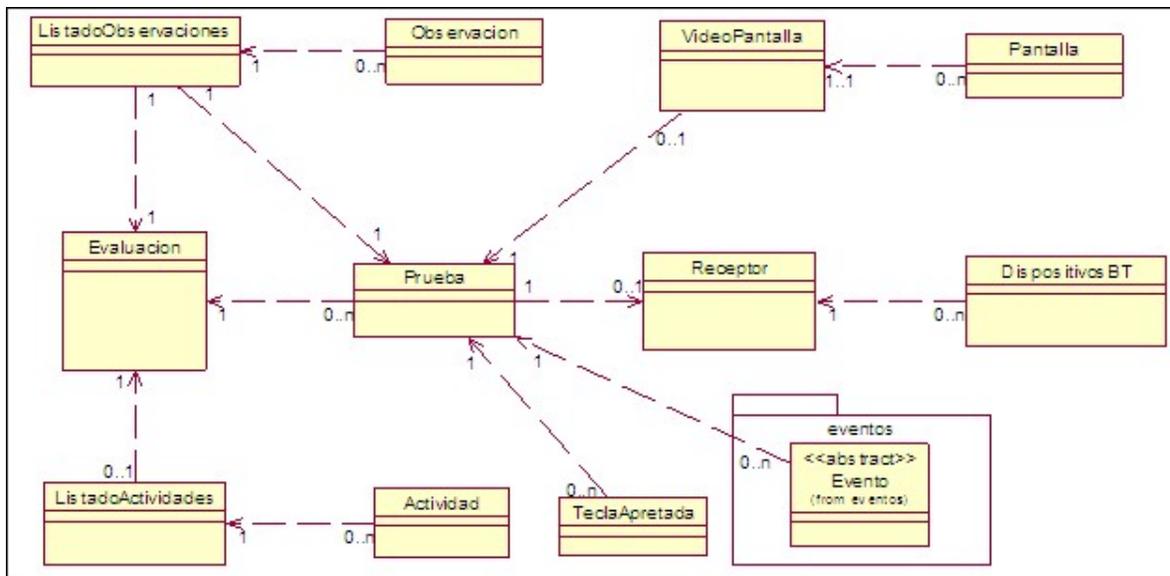


Ilustración 9-31: Diagrama de clases - Paquete aplicación (Parte 1)

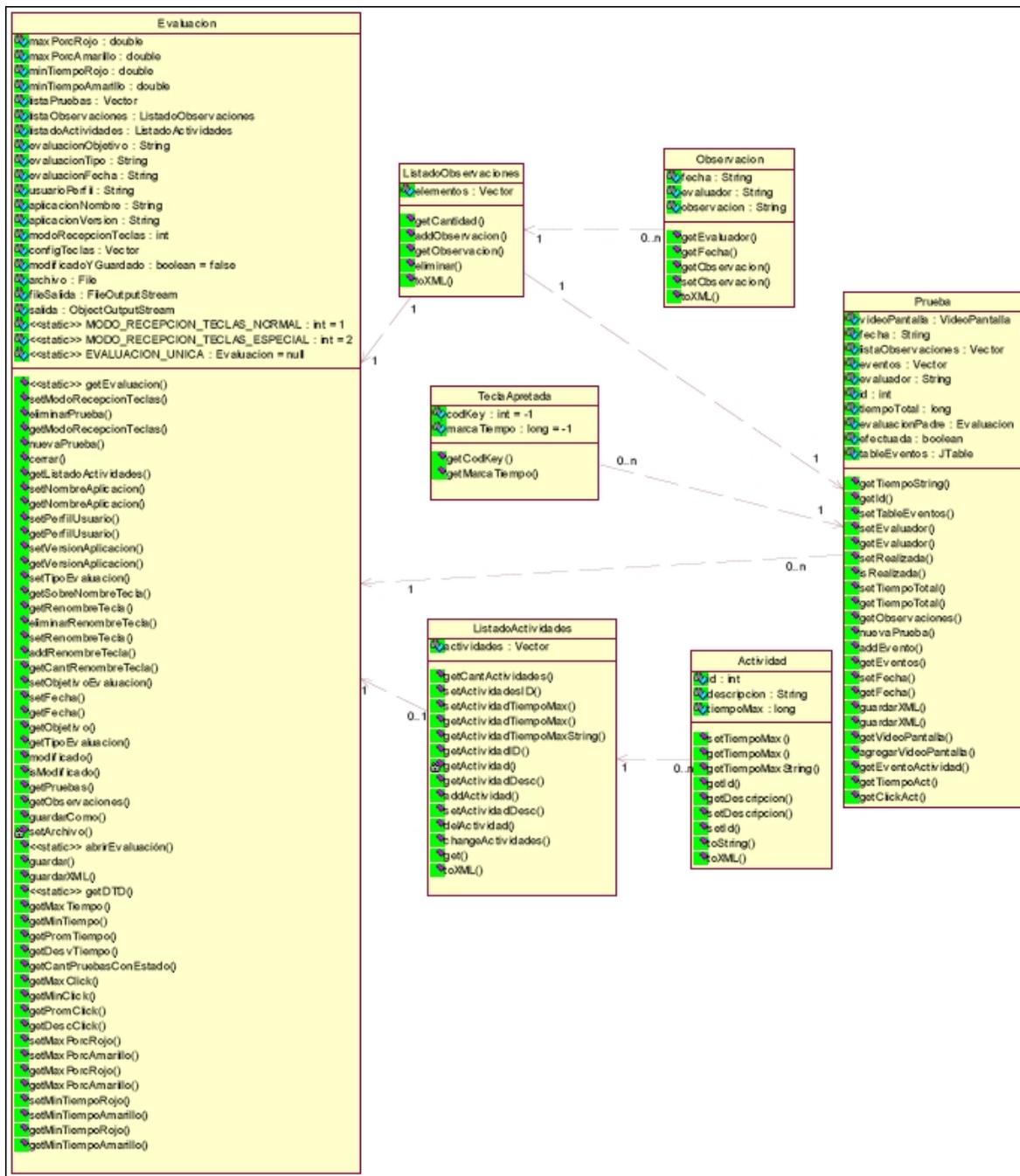


Ilustración 9-32: Diagrama de clases - Paquete aplicación (Parte 2)

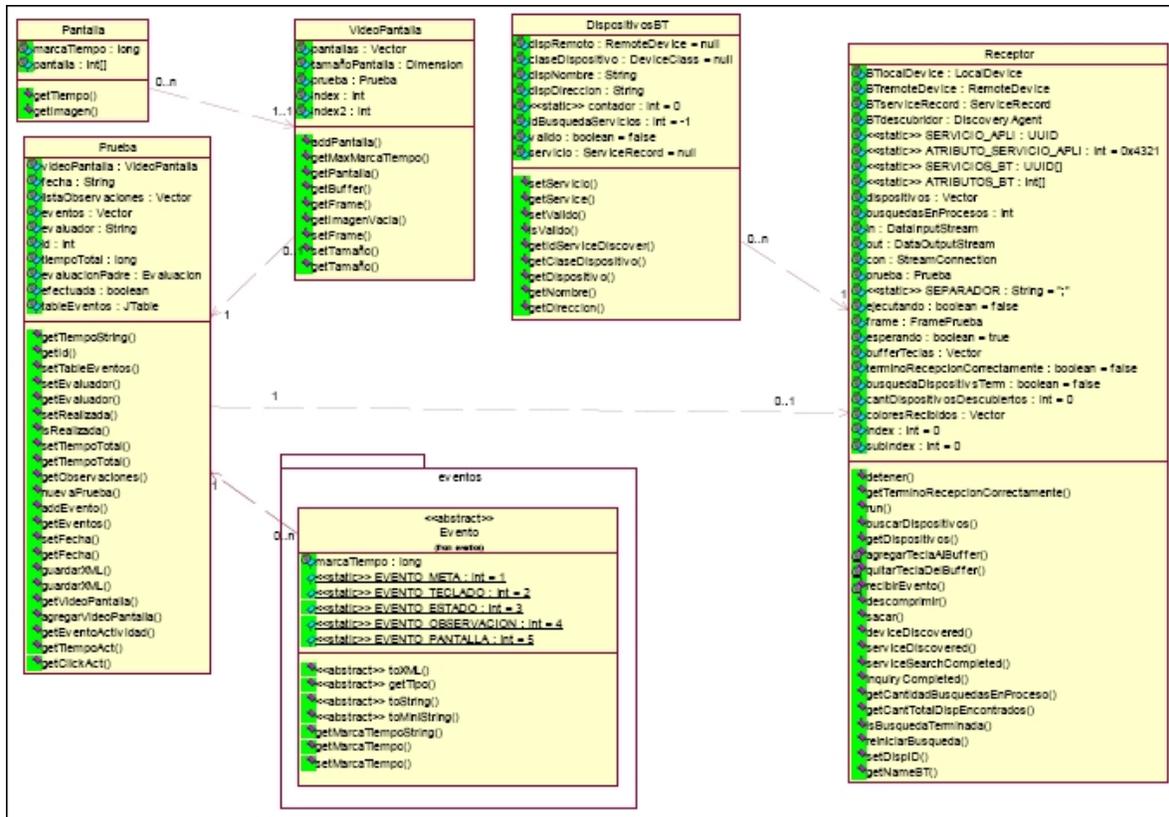
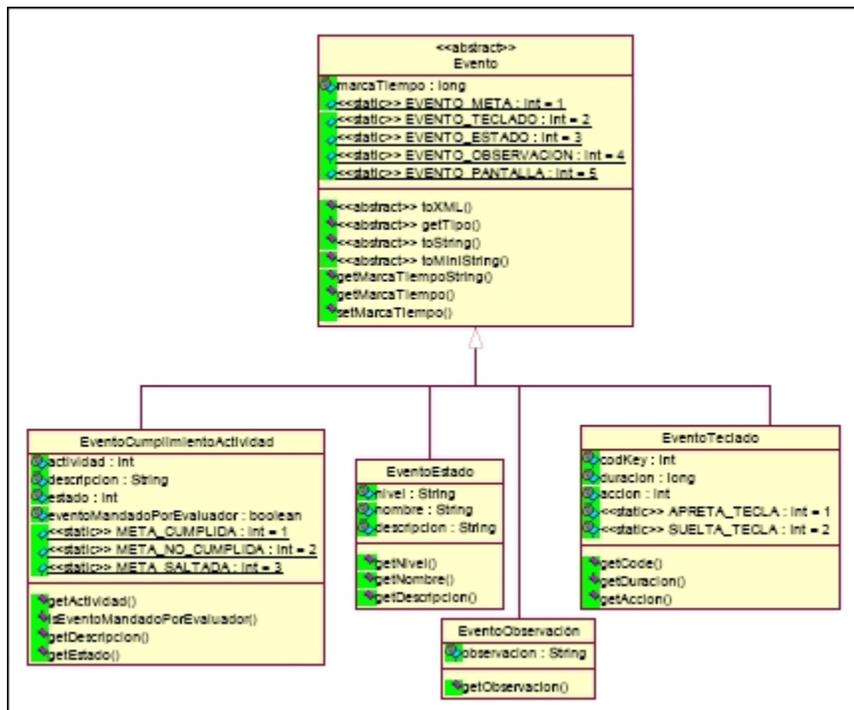
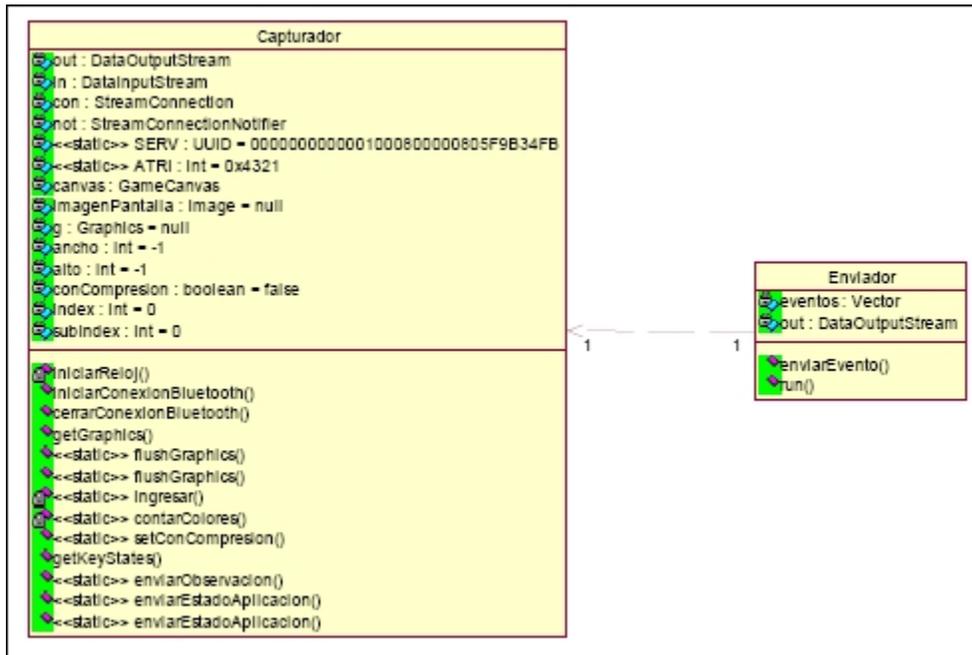


Ilustración 9-33: Diagrama de clases - Paquete aplicacion.eventosReceptor



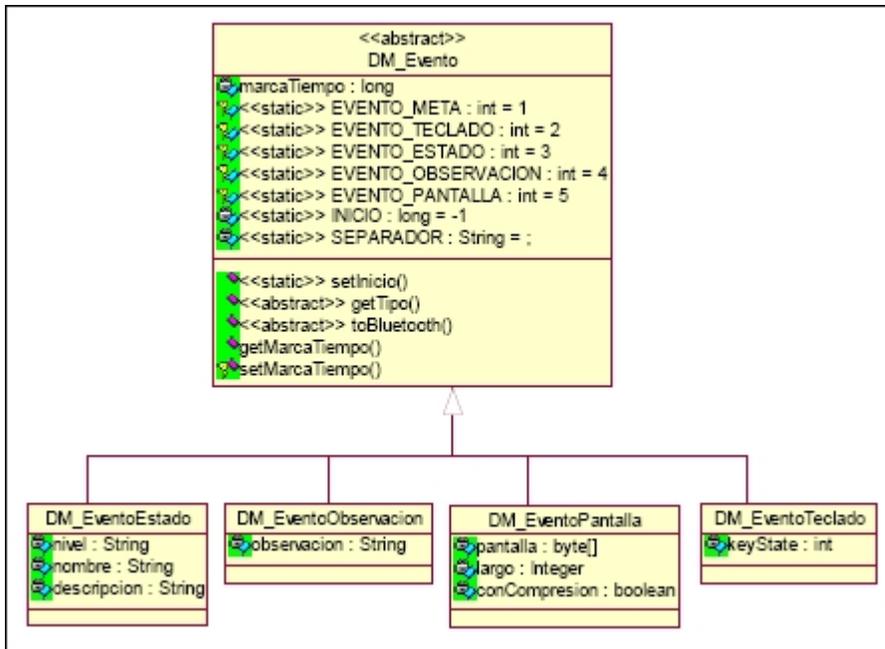
En la Ilustración 9-33 se ve el paquete eventosReceptor que es parte del paquete aplicación. Este paquete contiene las clases que definen los eventos que ocurren en el transcurso de la prueba y que sirven para el procesamiento en la aplicación en el computador.

Ilustración 9-34: Diagrama de clases - Paquete ApiBrotherEyesJr



En la Ilustración 9-34 se ve el paquete ApiBrotherEyesJr que contiene a las clases principales encargadas de la captura de los eventos y el envío de estos al computador.

Ilustración 9-35: Diagrama de clases - Paquete ApiBrotherEyesJr.eventosCapturador



En la Ilustración 9-35 se ve el paquete eventosCapturador que es parte del paquete ApiBortherEyesJr. Este paquete contiene las clases que definen los eventos que ocurren en el transcurso de la prueba y que serán enviados a la aplicación.

# Capítulo 10

## Desarrollo del primer prototipo

El desarrollo del primer prototipo se completó aproximadamente a mediados de septiembre del 2007 y este contenía algunos de los principales requerimientos, como el realizar la captura de los eventos ocurridos en el dispositivo móvil y la conexión vía Bluetooth para el envío de los mismos a la aplicación receptora.

Por otra parte, crea el respaldo de las evaluaciones, sus pruebas y los eventos de estas últimos a excepción de los videos capturados.

Uno de los requerimientos con el que mayores problemas se tuvo que enfrentar, fue lograr la conexión vía Bluetooth y se debió a las restricciones que tiene la librería de Java Bluecove que es una implementación en J2SE de la API JSR-82. Esta librería presenta problemas con el stack de Bluetooth Bluesoleil que es el stack con que trabaja el dispositivo Bluetooth USB con que se cuenta para este proyecto.

Otro requerimiento produjo complicaciones, fue el Debe poder capturar en video lo visualizado en la pantalla. Ya que el envío de las imágenes a través del Bluetooth tardaba demasiado tiempo, se decidió buscar una forma de comprimir las imágenes antes de ser enviadas. J2ME, al ser creado para dispositivos de bajas prestaciones, no tiene un completo manejo de las imágenes, es decir, no contempla una forma para comprimir imágenes ni realizar mayores modificaciones en ella. Para solucionar este problema, se decidió mandar las imágenes en escalas de grises, lo cual disminuyó el tamaño de la imagen a un tercio. El tiempo necesitado para transformar las imágenes a estas tonalidades y enviarlas, resulto no ser mayor al tiempo necesitado para enviar las imágenes sin transformarlas, por lo que es una alternativa aceptable.

De los requerimientos funcionales, este primer prototipo cumple:

*1.1. Debe informar los eventos que ocurran por la interacción con el usuario, como lo es el presionado de teclas y cambios de estado, todo el tiempo en que se este ejecutando la aplicación.*

*1.2. Debe poder capturar en video lo visualizado en la pantalla.*

El capturador captura el video, pero no es visible en el receptor hasta una vez terminada la captura de información.

*2.1.1. Debe poder iniciar la captura de información, además de pararla cuando desee.*

*2.2.1. Se deberá poder ingresar información referente la evaluación y a cada prueba de la evaluación.*

*2.2.2. El evaluador podrá guardar toda la información capturada para la evaluación.*

La herramienta respalda los eventos e información de una prueba, pero no guarda las imágenes capturadas.

*2.2.3. En una evaluación ya existente se debe poder agregar más pruebas.*

*2.2.5. Se debe poder reproducir una prueba y tener control sobre la reproducción.*

Sobre el cumplimiento de los requerimientos no funcionales, se puede mencionar lo siguiente:

*1.1. Garantizar que la información capturada es fidedigna y no se ve alterada en el proceso de captura.*

No se podía asegurar a cabalidad el cumplimiento de este requerimiento. Puesto que el tiempo de transmisión de la información afectaba al buen funcionamiento de la aplicación a evaluar.

*1.2. La captura de la información debe ser en tiempo real.*

La captura de la información era en tiempo real, pero no podía ser vista en tiempo real. Es decir, por el lado del receptor no se podía ver lo capturado hasta una vez terminada la captura.

*1.3. El desempeño de la aplicación a evaluar, no debe verse afectada por la herramienta.*

Existían severos problemas en este punto.

*1.4. La conexión entre el dispositivo móvil que contiene la aplicación a evaluar y el dispositivo que contiene la aplicación receptora, debe ser inalámbrica.*

Este requerimiento fue alcanzado sin mayores problemas. Para ello se utilizó la tecnología Bluetooth.

*3.1. La herramienta deberá estar documentada, cada uno de los componentes de software que forman parte de la solución propuesta deberán estar debidamente documentados tanto en el código fuente como en los manuales de usuario.*

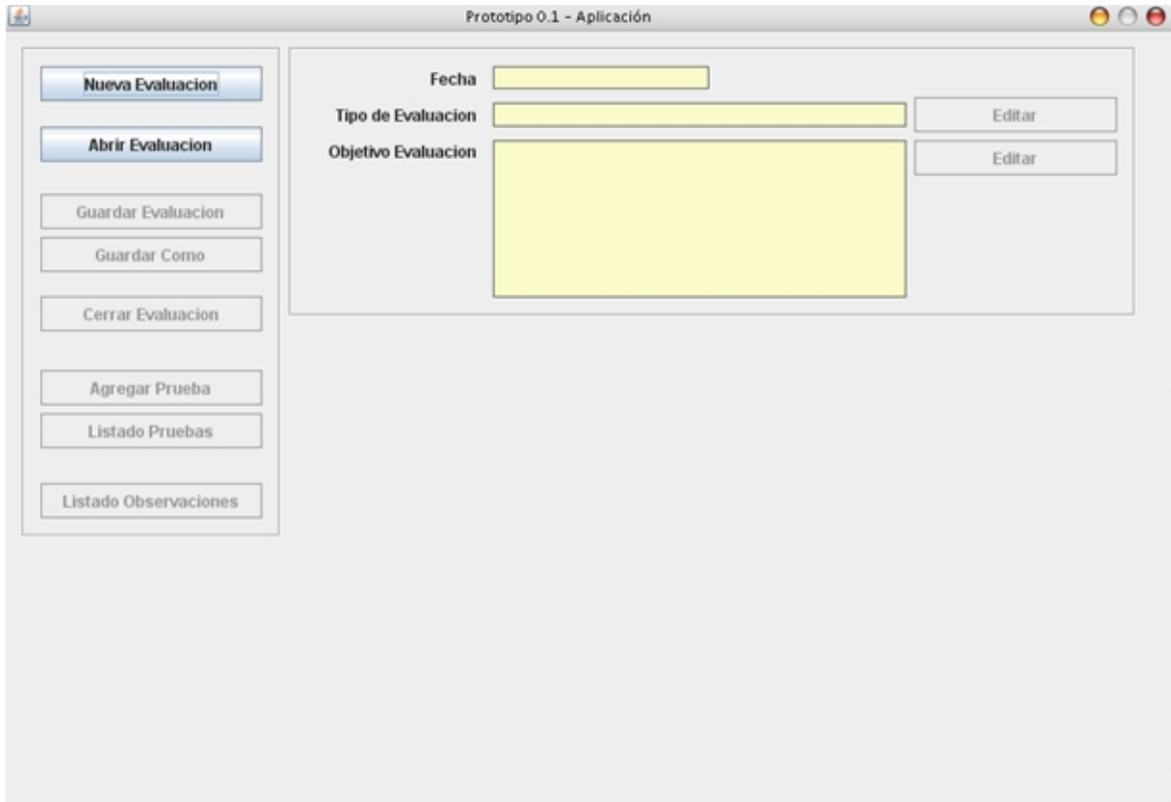
El código fuente fue comentado, pero otros documentos anexos, como el manual de usuario y el manual de instalación no fueron creados.

## **10.1 Comentarios sobre la interfaz de usuario**

Ahora se mostrarán algunas ilustraciones donde se pueden ver algunos de las ventanas más importantes del prototipo.

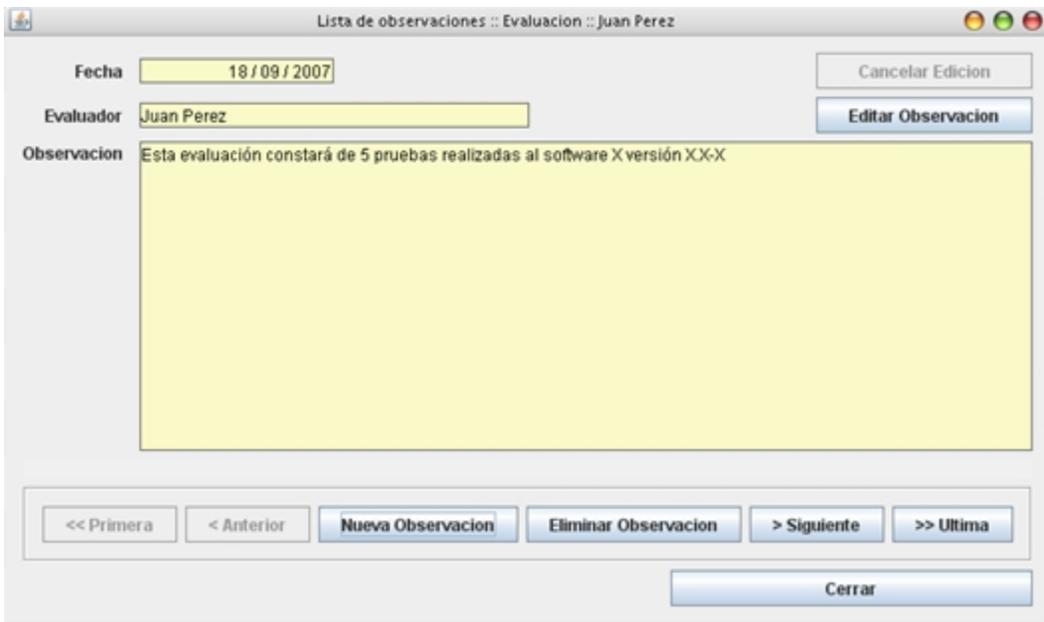
En la Ilustración 10-1, se puede observar la pantalla principal del prototipo, la cual, en su lado izquierdo, tiene el conjunto de botones que ayuda al usuario a realizar sus objetivos.

Ilustración 10-1: Primer prototipo



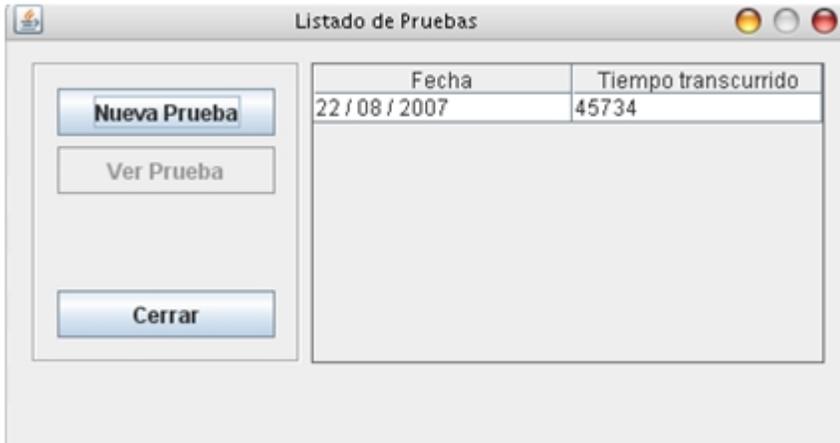
En la Ilustración 10-2, se puede observar la interfaz encargada de gestionar las observaciones, ya sea de la evaluación o de una de sus pruebas.

Ilustración 10-2: Primer prototipo - Lista de observaciones



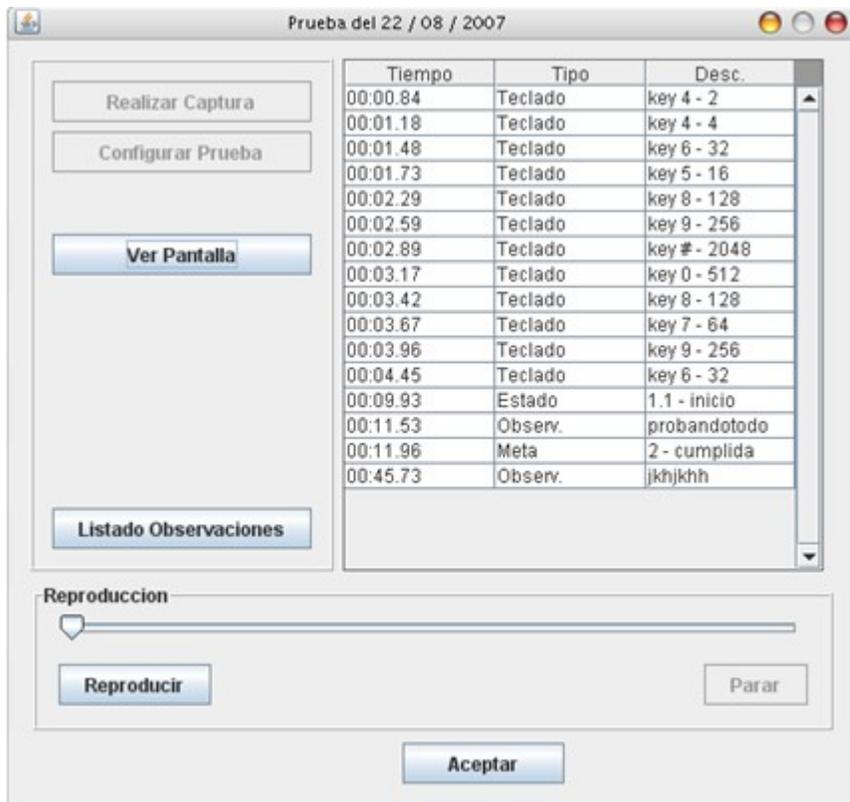
En la Ilustración 10-3, se puede apreciar la pantalla de la aplicación que se encarga de mostrar el listado de pruebas que tiene una evaluación, además de dar la opción de agregar una nueva prueba y revisar alguna prueba que se haya seleccionado.

Ilustración 10-3: Primer prototipo - Listado de pruebas



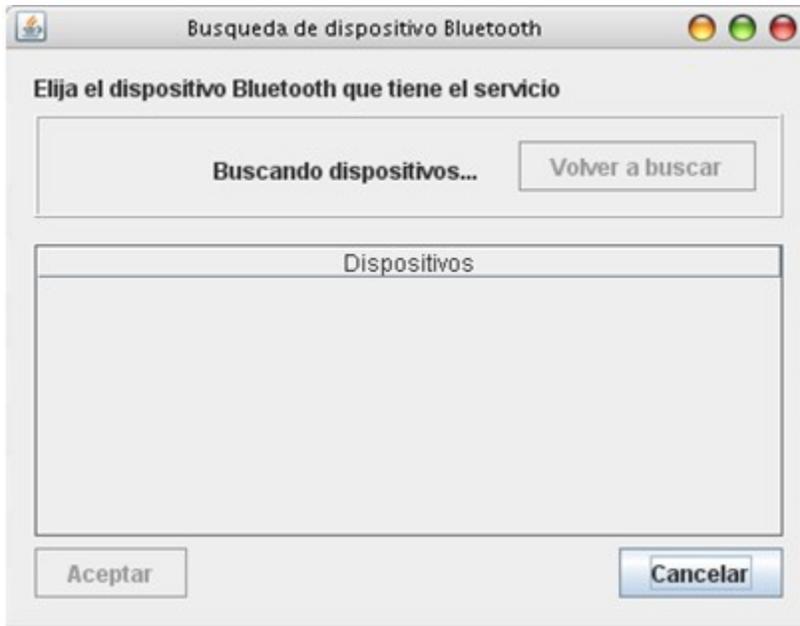
La Ilustración 10-4, muestra la ventana donde se ven los eventos capturados en la prueba y la sección encargada de la reproducción de la prueba. Además se puede visualizar el botón “Ver Pantalla” con el cual se abre una nueva ventana que muestra el video capturado.

Ilustración 10-4: Primer prototipo – Prueba



En la Ilustración 10-5, se puede apreciar la ventana que informa del proceso de búsqueda de los dispositivos Bluetooth y sus servicios. Además es la encargada de iniciar el proceso de captura de los eventos.

Ilustración 10-5: Primer prototipo - Búsqueda de dispositivos Bluetooth



## 10.2 Comentarios sobre la utilidad del primer prototipo

Solo por el hecho de que con anterioridad de iniciar este proyecto no se conocía ningún software que intentara apoyar las evaluaciones de usabilidad de aplicaciones para dispositivos móviles como lo intenta esta herramienta, es posible señalar que esta herramienta ha dado un gran avance y que realmente tiene un grado de utilidad, pero ya que se encuentra en la primera etapa de desarrollo, la utilidad que presenta no es suficiente.

Por ejemplo, el hecho de que pueda recibir las capturas de pantalla del dispositivo móvil y que puedan ser respaldadas para una futura revisión, limita a esta herramienta. Además, ya que la aplicación todavía no es capaz de realizar una exportación a XML, no permite una integración con otras herramientas.

Por otra parte, la aplicación ha sido probada son con teléfonos de la marca Sony Ericsson y por ende, no es posible asegurar que funcione con otros dispositivos, aunque se ha hecho lo posible por que la API que se desarrolló sea lo más genérica posible.

La prioridad que se tenía hasta esta etapa era que la aplicación cumpliera con los requerimientos funcionales más importante junto con algunos no funcionales. Por esto, es

que no se tenía en mente un diseño centrado en el usuario y por ende se puede decir que la usabilidad de la aplicación es muy baja. Esto perjudica la utilidad de la herramienta.

### **10.3 Tareas para el próximo prototipo**

Para la segunda parte del desarrollo del prototipo quedan bastantes tareas que hacer, de las cuales cabe destacar:

- Mejorar la interfaz grafica.
- Terminar de implementar la gestión de archivo.
- Implementar la exportación de la evaluación o prueba a un documento XML.
- Implementar la función de que, mientras se este recibiendo los eventos en la aplicación receptora, se pueda ir viendo lo capturado.
- La exportación del video capturado a algún formato estándar.

# Capítulo 11

## Desarrollo del segundo prototipo

En la última etapa de este proyecto, se tiene un prototipo de la herramienta mucho más completa. Además, se desarrollo un pequeño juego para teléfonos celulares en J2ME para poder probar la herramienta.

De los requerimientos funcionales, este segundo prototipo cumple:

*1.1. Debe informar los eventos que ocurran por la interacción con el usuario, como lo es el presionado de teclas y cambios de estado, todo el tiempo en que se este ejecutando la aplicación.*

*1.2. Debe poder capturar en video lo visualizado en la pantalla.*

*2.1.1. Debe poder iniciar la captura de información, además de pararla cuando desee.*

*2.1.2. Debe poder informar de algunos eventos referentes a la prueba misma (cumplimientos de tareas, frustraciones, etc.).*

*2.2.1. Se deberá poder ingresar información referente la evaluación y a cada prueba de la evaluación.*

*2.2.2. El evaluador podrá guardar toda la información capturada para la evaluación.*

*2.2.3. En una evaluación ya existente se debe poder agregar más pruebas.*

*2.2.4 Se debe poder exportar el video capturado a un formato estándar.*

*2.2.5. Se debe poder reproducir una prueba y tener control sobre la reproducción.*

*2.3.3. Se debe poder generar un documento XML ya sea de toda la evaluación o de alguna prueba en particular.*

Sobre el cumplimiento de los requerimientos no funcionales, se puede mencionar lo siguiente:

*1.1. Garantizar que la información capturada es fidedigna y no se ve alterada en el proceso de captura.*

No se podía asegurar a cabalidad el cumplimiento de este requerimiento. Este problema será explicado detalladamente en el punto 11.3.

*1.2. La captura de la información debe ser en tiempo real.*

Este requerimiento fue alcanzado sin mayores problemas.

*1.3. El desempeño de la aplicación a evaluar, no debe verse afectada por la herramienta.*

No se podía asegurar a cabalidad el cumplimiento de este requerimiento. Este problema será explicado detalladamente en el punto 11.3.

*1.4. La conexión entre el dispositivo móvil que contiene la aplicación a evaluar y el dispositivo que contiene la aplicación receptora, debe ser inalámbrica.*

Este requerimiento fue alcanzado sin mayores problemas. Para ello se utilizó la tecnología Bluetooth.

*2.1. Se debe buscar o establecer un estándar al cual poder exportar los eventos capturados en las pruebas. Para asegurar compatibilidad con otras herramientas de usabilidad.*

Este requerimiento fue alcanzado sin mayores problemas. Para ello se estableció una definición de tipo de documento (DTD) para la herramienta, ya que no se logró encontrar un estándar existente. En el manual de usuario, se encuentra el DTD establecido.

*3.1. La herramienta deberá estar documentada, cada uno de los componentes de software que forman parte de la solución propuesta deberán estar debidamente documentados tanto en el código fuente como en los manuales de usuario.*

Este requerimiento fue alcanzado medianamente. El código no fue comentado completamente, solo se comentaron las funciones que se consideraban más importantes, pero se realizaron los comentarios necesarios para poder generar los Java Doc tanto de la aplicación como de la API. Y, por último, se documentó el manual de usuario.

Para crear el instalador de la aplicación, se utilizó el software CreateInstall versión 4.12.2, el cual se descargó la versión free de su página oficial <http://www.createinstall.com/cifree/index.htm>.

## 11.1 Comentarios sobre la interfaz de usuario

La Ilustración 11-1, muestra una imagen de la pantalla principal de la herramienta. Se puede observar que a diferencia del primer prototipo, esta versión tiene un menú con las opciones referentes a una Evaluación.

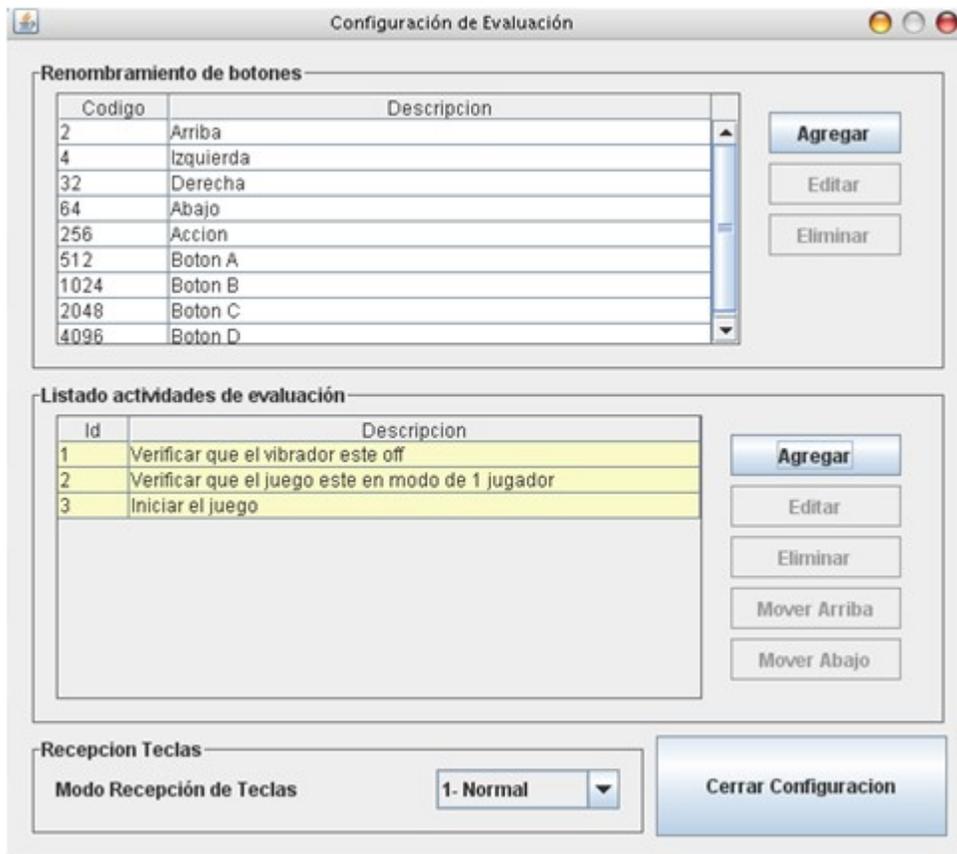
Además, se pueden apreciar algunos botones de acceso rápido a algunas opciones y una pequeña tabla que muestra las pruebas que pertenecen a la evaluación.

Ilustración 11-1: Segundo prototipo

Id	Fecha	Evaluador	Realizada
----	-------	-----------	-----------

La Ilustración 11-4, muestra la ventana de configuración de una evaluación, la cual no existía en la antigua versión. En esta, el evaluador podrá agregar, modificar y eliminar los seudónimos de las teclas, agregar, modificar, eliminar y ordenar las actividades de una evaluación y seleccionar el modo de recepción de las teclas.

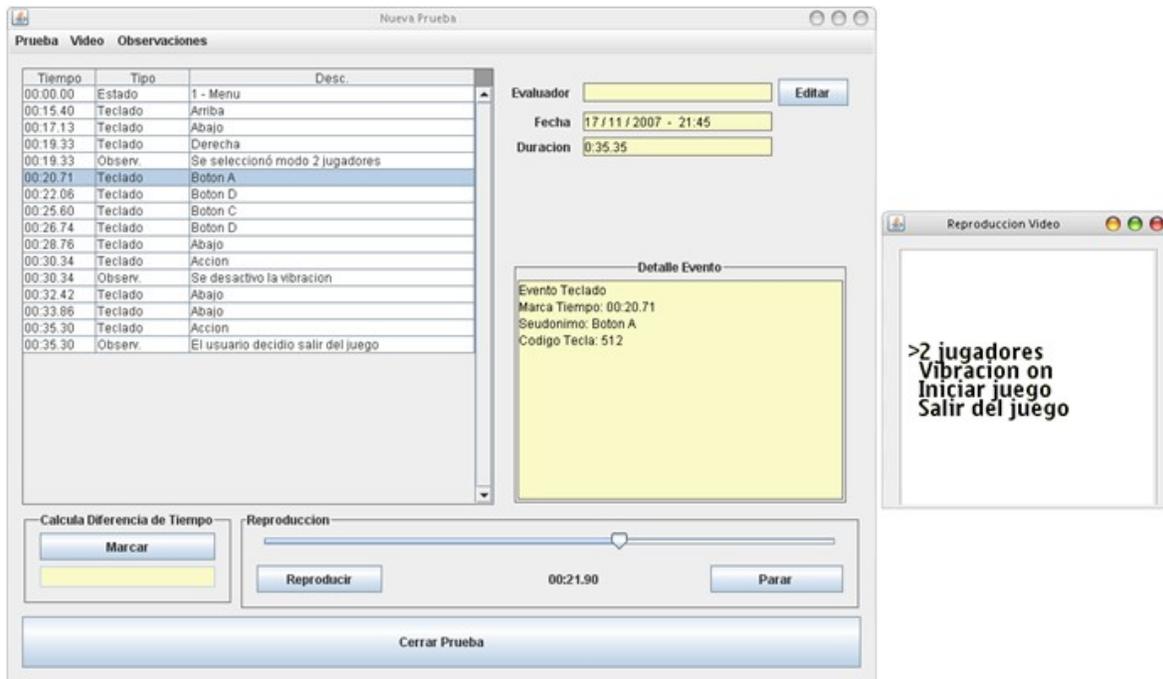
Ilustración 11-2: Segundo prototipo - Ventana de configuración de una evaluación



La Ilustración 11-3 muestra la ventana de un prueba perteneciente a una evaluación, que a diferencia de la que existía en la versión anterior, esta presenta algunos pequeño cambios. Tiene una barra de menú con las opciones referentes a una prueba, una sección para calcular la diferencia en tiempo entre 2 eventos, información asociada a la prueba, como el nombre del evaluador, la fecha en que se creo la prueba y la duración de la misma.

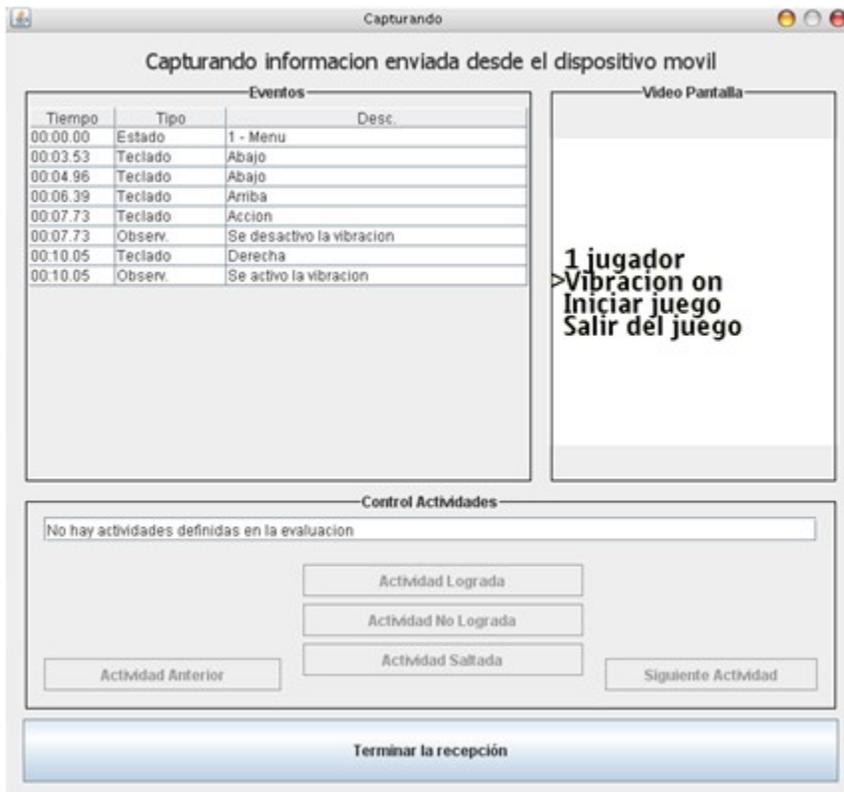
Además, en la misma ilustración se puede apreciar la ventana que muestra el video capturado de lo visualizado en el dispositivo móvil durante la prueba.

Ilustración 11-3: Segundo prototipo - Ventana de una prueba



En la Ilustración 11-4, se puede apreciar ventana que muestra al evaluador el proceso de captura. En esta interfaz, el evaluador puede ir marcando si alguna de las actividades fue lograda, no lograda o saltada. Para esto, el evaluador debería con anterioridad haber ingresado las especificaciones de la lista de actividades en la configuración de la evaluación.

Ilustración 11-4: Segundo prototipo - Ventana capturando



De las demás ventanas, no se mostrarán Screenshot, porque no han reflejado mayores cambios.

## 11.2 Pruebas de utilidad

Para estas pruebas, se desarrollo un juego en J2ME basado en el juego de papel conocido como gato o tres en línea. Este juego se caracteriza por que no envía las imágenes cada cierto tiempo, sino más bien, cada vez que el jugador aprieta alguna tecla del dispositivo, el juego realiza los cambios necesarios, realiza el dibujado de la pantalla y se aprovecha de enviar la imagen a la aplicación receptora.

Para analizar la utilidad de la herramienta, se ha diseñado algunas pruebas de usabilidad que analizarán el juego del gato. Para esto, se les pidió la participación voluntaria a alumnos de la escuela de Ingeniería Informática de la Universidad que hayan cursado el ramo optativo Human-Computer Interaction. Ellos, aunque no tiene la suficiente experiencia para ser llamados expertos en usabilidad, tienen los conocimientos teóricos para poder dar una opinión más o menos objetiva sobre esta herramienta.

También se les pidió la cooperación a otros alumnos de la escuela para actuar de tester dentro de las pruebas de usabilidad, es decir, como jugadores del juego gato.

El notebook Dell Inspiron 1501, un dispositivo Bluetooth USB y un teléfono móvil con Bluetooth y la API JSR82, fueron utilizados para la realización de las pruebas.

La pauta que se siguió en cada una de las pruebas fue la siguiente.

Primero se le mostraba el software al evaluador. Indicándole algunas de las funcionalidades más importantes. Esto se hacía puesto que el objetivo es analizar la utilidad y no la usabilidad de la herramienta.

Después se deja al evaluador que inicie las actividades indicadas en el documento entregado. Cuando se encuentre listo, iniciar la prueba de usabilidad con el jugador voluntario.

Al terminar todas las actividades, el evaluador, si lo desea, puede recorrer la herramienta para conocerla más detalladamente.

Finalmente, el evaluador deberá contestar un pequeño cuestionario que se encuentra en el documento entregado. Este cuestionario consta de 7 preguntas las cuales pueden ser respondidas utilizando una escala que va desde 1 a 7.

Se realizaron un total de 5 pruebas, en un ambiente fuera de un laboratorio de usabilidad, pero estático. En la Tabla 11-1, se muestran las evaluaciones hechas por los evaluadores gracias al cuestionario que se les entregó.

Tabla 11-1: Resultados prueba de utilidad

<i>N°</i>	<i>Pregunta</i>	<i>Evaluador 1</i>	<i>Evaluador 2</i>	<i>Evaluador 3</i>	<i>Evaluador 4</i>	<i>Evaluador 5</i>	<i>Promedio</i>
<i>1</i>	<i>¿La información que identifica una evaluación le parece</i>	<i>5</i>	<i>5</i>	<i>6</i>	<i>6</i>	<i>7</i>	<i>5,80</i>

	<i>correcta?</i>						
2	<i>¿Qué tanto afecta a la prueba el tiempo que se demora una imagen, el ser enviada desde dispositivo móvil a la aplicación?</i>	2	6	4	4	5	<b>4,20</b>
3	<i>¿Qué tan valiosa encuentra la información entregada por el evento de tipo Observación?</i>	7	6	5	5	7	<b>6,00</b>
4	<i>¿Qué tan valiosa encuentra la información entregada por el evento de tipo Actividad?</i>	7	6	5	7	5	<b>6,00</b>
5	<i>¿Qué tan valiosa encuentra la información entregada por el evento de tipo Estado?</i>	4	5	5	5	6	<b>5,00</b>
6	<i>¿Qué tan valioso encontró la función de poder marcar el cumplimiento o no de las actividades definidas?</i>	7	7	7	7	7	<b>7,00</b>
7	<i>¿Cómo clasificaría en general a la aplicación?</i>	5	5	6	6	6	<b>5,60</b>

El promedio más bajo, que es un 4,2, corresponde a la pregunta 2. Dicho problema se encuentra comentado en el punto 11.3.

Por otra parte, la clasificación más alta se encuentra en la pregunta 6, que hace referencia a la posibilidad, que mientras se esta realizando la captura, el evaluador desde el computador que realiza la recepción de los eventos, puede ir informando a la aplicación cuando el

usuario ha completado alguna actividad. Cabe mencionar eso si, que varios evaluadores comentaron que se podría mejorar esta funcionalidad, puesto que cuando se marca algún actividad completada, no se puede desmarcar.

2 de los 5 evaluadores, dejaron observaciones. El primero hace referencia al problema que se mencionó en el párrafo anterior. Y el segundo, hace referencia a algunos problemas de usabilidad que encontró durante la prueba.

Finalmente, los evaluadores han calificado esta herramienta con una nota promedio de 5,6.

### **11.3 Problemas del segundo prototipo**

El problema que tiene esta herramienta, es la funcionalidad de poder enviar lo visualizado en la pantalla del dispositivo móvil durante la prueba. Esto se debe a que la baja tasa de transferencia de la tecnología Bluetooth, no permite enviar grandes cantidades de información en poco tiempo.

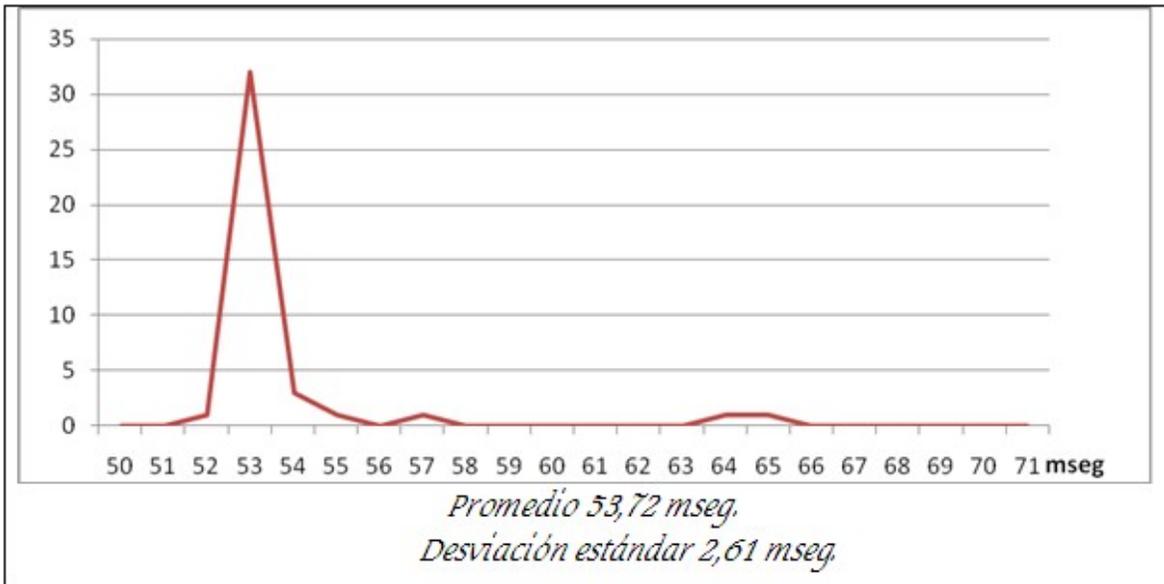
Por ejemplo, si consideramos la pantalla del teléfono que se ocupó en las pruebas, cuya dimensión de la pantalla es de 128x160, y considerando que se ocupan 3 Byte por pixel, el tamaño del buffer de la pantalla es de 491.520 bites.

Se decidió enviar lo visualizado en la pantalla en blanco y negro, para ocupar un Byte por pixel. De esta forma, se ahorraría tiempo en el envío de la información y no se afectaría demasiado la interacción del jugador con la aplicación.

Algunas estadísticas que ayudan a comprender un poco el problema serian:

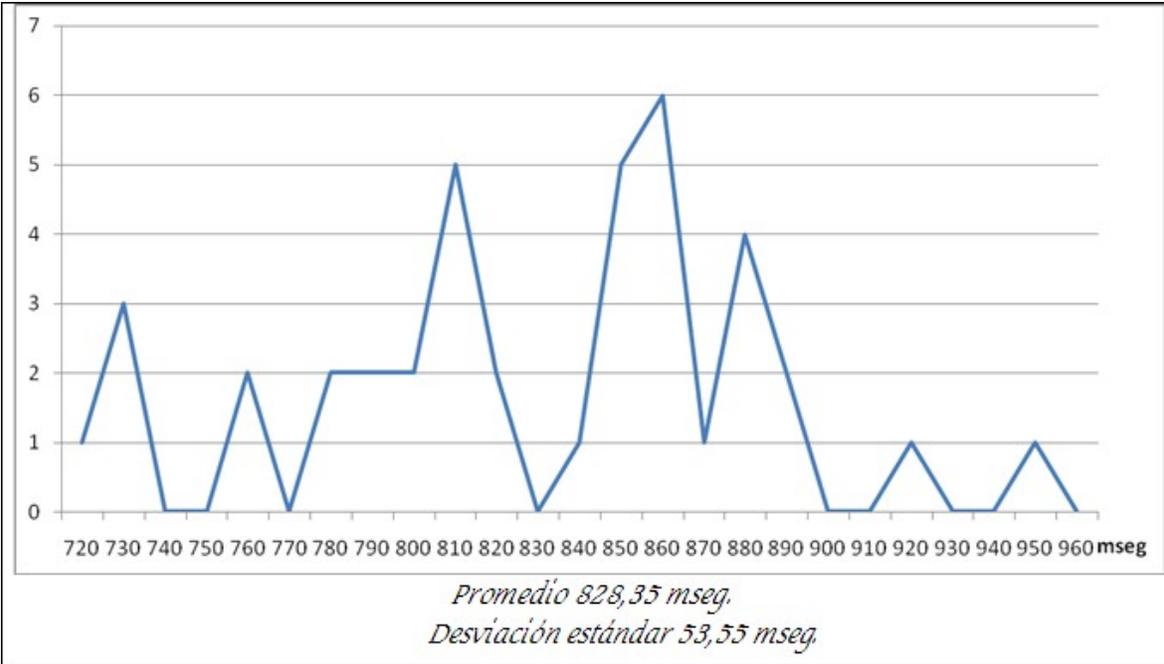
En la Ilustración 11-5, se puede ver un gráfico de frecuencias de los tiempos que se demora el dispositivo móvil en transformar la imagen a blanco y negro, haciendo un promedio de 53,72 milisegundos, lo cual es aceptable para un juego de pocos frames por segundos. Este proceso no afecta, en mayor parte, la interacción.

Ilustración 11-5: Tiempos en comprimir imagen (en msec.)



Pero como se puede apreciar en gráfico de frecuencias de la Ilustración 11-6, el tiempo en que se demora la herramienta en pasar las imágenes desde el dispositivo móvil a la aplicación receptora rodea los 828 milisegundo, lo cual es demasiado, si consideramos que para que un juego tenga la suficiente suavidad en sus movimientos, es necesario que dibuje por lo menos, unos 14 cuadros por segundos (71,4 mseg.), donde lo ideal sería capturar todos estos cuadros.

Ilustración 11-6: Tiempos de envío de imagen (en mseg.)



Si sumamos los tiempos mínimo en ambos procesos obtenemos 776 mseg. Lo cual permite tener una velocidad de imagen de 1,29 cuadros por segundos. Y si sumamos los tiempos máximos, nos daría un total de 1019 mseg, lo que equivale a una velocidad de 0,98 cuadros por segundos.

En ambos casos, podemos ver que nos encontramos frente a un problema bastante complicado al momento de intentar no alterar la interacción entre la aplicación que se desea evaluar y el usuario.

Para poder superar este problema, se debería contar con una tecnología que tenga una capacidad mayor a la de Bluetooth. La tecnología WiFi tiene esta capacidad. En su estándar IEEE 802.11b alcanza los 11Mbps y en el IEEE 802.11g, los 54Mbps, los cuales llegan a superar con creces la capacidad de Bluetooth.

El problema de utilizar esta tecnología es que muy pocos dispositivos móviles tienen integrada esta tecnología. Generalmente son SmartPhone, PDA y BlackBerry, lo cual limita el uso de la herramienta. Además estos dispositivos, no son tan populares como los teléfonos celulares.

Por otra parte, la utilización de la librería de Java Bluecove para poder ocupar JSR82 (Api de Java para Bluetooth), presenta problemas con el stack de Bluetooth (Bluesoleil) con que trabaja el dispositivo Bluetooth USB con que se cuenta para este proyecto. Aunque no afectó en gran medida a la herramienta, se recomienda utilizar WIDCOM, puesto que es el stack que presenta mayor compatibilidad con Bluecove.

## Capítulo 12

### Desarrollo del tercer prototipo

En esta tercera iteración se implementó la posibilidad de modificar los eventos de cumplimiento de actividad de una prueba. Ahora se puede modificar la marca de tiempo a la que corresponde, el estado del cumplimiento (meta cumplida exitosamente, erróneamente y meta saltada) y agregar o eliminar este tipo de eventos. La idea es que cada prueba tenga una única vez un evento de cumplimiento de actividad por cada actividad asociada a la evaluación. Además se le da la posibilidad al evaluador de agregar eventos de observación a la prueba y posteriormente modificarlos. Estos eventos de observación son distintos a las observaciones generales que se puede asociar a una prueba. Todo esto se puede realizar una vez ya realizada la captura de la prueba y las veces que se estimen necesarias.

Por otra parte, al documento XML exportado de una evaluación, no de una prueba en particular, se le ha agregado la lista con las actividades asociadas a esa evaluación.

A la interfaz de la aplicación se le agregó la posibilidad de hacer doble click a elementos que se encuentren en tablas y, de esta manera, acceder a determinadas opciones de forma más rápida con respecto a estos elementos. Por ejemplo en el frame de configuración de la evaluación, se puede hacer doble click a actividades que se encuentran en la tabla de actividades, para poder editarlas.

La interfaz también fue modificada en su aspecto. Se cambio el LookAnd de la interfaz a una más amistosa para el usuario. Ahora los botones son curvos en sus esquinas y no completamente rectos como en las versiones anteriores. Esto ayuda a presentar una interfaz mucho más amistosa y llamativa al usuario.

Una mejora radical que tuvo la aplicación fue el cambio que se realizó a la exportación del video de una prueba. En el prototipo anterior el video exportado llegaba a tener un tamaño excesivamente grande, llegando a pesar aprox. 300 MB por un minuto de grabación,

llegando a ser difícil su transporte. En esta versión se cambió el formato del video. Pasó de ser QuickTime a AVI<sup>1</sup> (Audio Video Interleave). Aunque el cambio en el peso del archivo de video fue bastante drástico, todavía es bastante pesado en proporción al tiempo que dura el video, pero en estos momentos es más transportable que en la versión anterior. Para disminuir más el tamaño del archivo, el evaluador puede optar por comprimir dicho video a otro formato con un programa externo, como por ejemplo DivX. Este fue el motivo por el que se decidió realizar el cambio de formato. Además en estos momentos, la aplicación, a través de una barra de avance, muestra el proceso de exportación del video.

Por otra parte, se trabajó arduamente en una forma de disminuir el peso de la imagen que se captura y envía a la aplicación receptora para, de esta manera, disminuir el tiempo de transferencia de dicha imagen. Para ello, se implementó un algoritmo de compresión de indexación de colores. Este algoritmo reemplaza el identificador de color de cada pixel por uno que ocupe menos bit. Para esto se tiene que almacenar y contar la cantidad de colores que tiene la imagen para posteriormente decidir cuantos bits ocupar para la nueva identificación de los colores y, por último, reemplazar los índices de color de la imagen por los nuevos. Aunque este algoritmo produce una compresión de aprox. 70%, dependiendo de la cantidad de colores que tenga la imagen original, el tiempo que se necesita para realizar la compresión es elevado. Este fue el motivo de porque este algoritmo resultó infructuoso para el objetivo que se necesitaba. Al enviar directamente la imagen a la aplicación receptora, el tiempo que se necesitaba para capturar la imagen era de 53,72 mseg. y el tiempo para enviar la imagen por Bluetooth era de 828,35 mseg. Con la implementación del algoritmo de compresión se utiliza un tiempo de 1.087,79 mseg. en capturar la imagen y comprimirla y 499,03 mseg. en enviarla. Al buscar otra forma de envío de la imagen, se implementó un nuevo thread encargado de realizar el envío. En estos momentos, la API reúne los eventos a ser enviados en una lista-buffer y este thread se encarga de ir enviando los eventos sin interrumpir la ejecución normal de la aplicación. Aunque este método tiene la restricción de la cantidad de memoria con la que cuenta el dispositivo móvil, para aplicaciones que no realizan un alto número de refrescos de pantalla es una excelente solución. Este no altera la ejecución del juego mientras es capturado, los eventos llegan con cierto desfase al capturador con respecto al tiempo en que ocurrieron, pero no llega a ser un gran problema puesto que los eventos traen su marca de tiempo ya establecida.

En estos momentos, la herramienta cumple con todos los requerimientos funcionales, siendo el último que faltaba, el 2.3.2 “Se debe poder generar estadísticas necesarias para facilitar el análisis de una evaluación”. Para este requerimiento se implementó un nuevo frame el cual puede ser llamado desde el menú Evaluación del frame principal de la aplicación. Este frame da la opción de ver ciertas estadísticas con respecto a toda la evaluación y de cada una de las actividades asociadas a la evaluación. En estas estadísticas se puede ver el porcentaje de satisfacción de las actividades. Es decir, el porcentaje de pruebas que tienen la actividad cumplida exitosamente y el promedio de tiempo que se demoraron los tester en cumplir la actividad en forma exitosa. Además cada actividad es analizada con más detalle en sus tres estados (meta cumplida exitosamente, erróneamente y saltada) entregando el máximo y mínimo tiempo que se ocuparon, el promedio del tiempo y su desviación estándar, la máxima y mínima cantidad de click necesarios y su promedio y desviación estándar. Se espera que esta información ayude al evaluador a analizar más rápidamente el software.

Otro cambio que se realizó a la aplicación tiene que ver con la estructura del documento XML. El DTD fue modificado para hacer un documento más estándar a diferencia del que se tenía. Además dicho DTD no es incorporado dentro del documento XML. Al momento que se realiza la exportación de la evaluación o de la prueba, se generan dos archivos: el primero es el documento xml con el nombre escogido por el usuario y por otra parte, se crea en la misma ubicación donde se creó el xml, el archivo usabilidad.dtd que contiene el DTD del archivo xml.

Los requerimientos definidos al comienzo de este proyecto se han logrado cumplir completamente. Los requerimientos funcionales que faltaban por implementar fueron implementados (resumen de las pruebas de una evaluación que se puede ver en la ventana principal de la aplicación y la implementación de la ventana con información estadísticas de la evaluación y las actividades asociadas a dicha evaluación).

El requerimiento no funcional que hace mención a que el desempeño de la aplicación a evaluar no debe verse afectado por la utilización de la herramienta, fue cubierto gracias a la implementación del thread dedicado al almacenamiento y exportación de los eventos en forma independiente.

En estos momentos la herramienta cumple con todos los requerimientos tanto funcionales como no funcionales. En la siguiente sección se verán algunos screenshots de la herramienta y posteriormente los resultados de una evaluación de utilidad que se realizó a la herramienta.

## 12.1 Comentarios sobre la interfaz grafica

En la Ilustración 12-1 y en la Ilustración 12-2, se puede ver el proceso de exportación del video capturado de lo visualizado en la pantalla del dispositivo móvil durante una prueba. En los prototipos anteriores este proceso no era informado y por ende, el usuario no podía saber cuando el proceso había terminado.

Ilustración 12-1: Tercer Prototipo – Proceso exportación 1

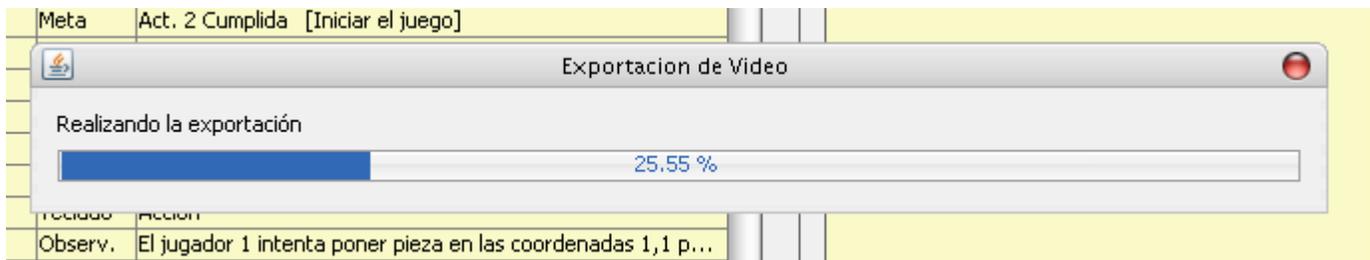
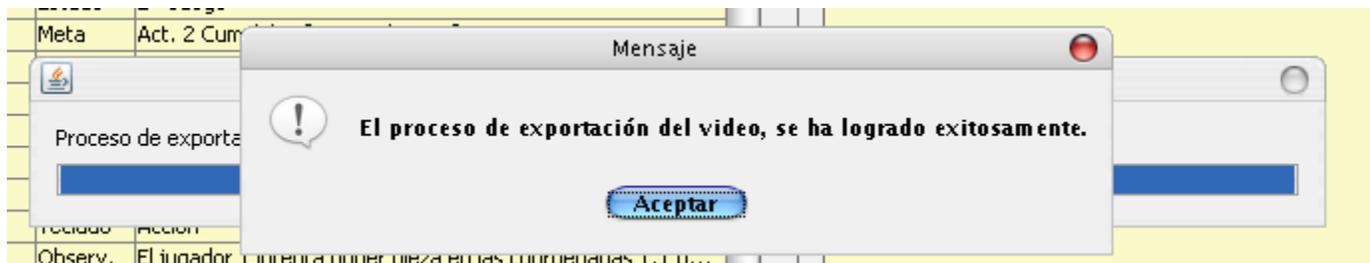


Ilustración 12-2: Tercer Prototipo – Proceso exportación 2



La Ilustración 12-3, muestra el frame que entrega información estadísticas de las evaluaciones y sus actividades asociadas. En este frame se pueden ver 2 secciones. La primera muestra estadísticas a nivel de toda la evaluación, donde muestra el grado de satisfacción con respecto a una actividad, es decir, la cantidad de pruebas en que se cumplieron exitosamente la actividad y el porcentaje que estas representan, y el tiempo a favor que se tiene con respecto al promedio obtenido entre las pruebas que lograron la actividad exitosamente. En esta sección se puede identificar el grado de aceptabilidad de

estos rangos a través de colores en las celdas. Estos rangos de colores pueden ser modificados a través de la configuración a la que se puede ir a través del menú de este frame. En la Ilustración 12-4 se puede ver este frame de configuración.

Una segunda sección muestra más detalles sobre una actividad específica, pudiendo analizar los tres estados de una actividad.

Ilustración 12-3: Tercer prototipo – Ventana de estadísticas

**Estadísticas**

**Configuración**

Estadísticas generales de la evaluación

Actividad	Satisfacción (*)	Dif. media tiempo (**)
1.-Dejar el modo de vibración en off	3/4 75,00 %	-00:01.805
2.-Iniciar el juego	2/4 50,00 %	00:02.007
3.-Terminar el juego sin importar quien gane	2/4 50,00 %	00:35.935
4.-Dejar el modo vibración en on	3/4 75,00 %	00:01.395
5.-Salir del juego	3/4 75,00 %	00:00.548

(\*) Muestra el total de pruebas que tienen la actividad completada exitosamente.  
 (\*\*) Es el tiempo a favor que se tiene entre el tiempo esperado en una actividad y el promedio de las pruebas con la actividad cumplida exitosamente.

Actividad  
 3.-Terminar el juego sin importar quien gane

Tiempo Máximo Estimado para esta actividad: 01:00.000

Total de pruebas de la evaluación: 4

Analizar pruebas que:  
 estan con la actividad saltada

Cantidad de pruebas: 1      Porcentaje Correspondiente: 25,00 %

**Tiempos Ocupados**

Tiempo Máximo: 00:04.531

Tiempo Mínimo: 00:04.531

Tiempo Promedio: 00:04.531

Tiempo a favor: -

Desviación Estandar: 00:00.000

**Cantidad de Clicks**

Máximo de Click: 0

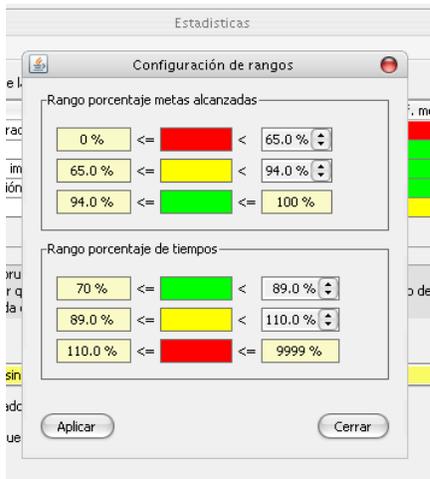
Mínimo de Click: 0

Promedio de Click: 0,00

Desviación Estandar: 0,00

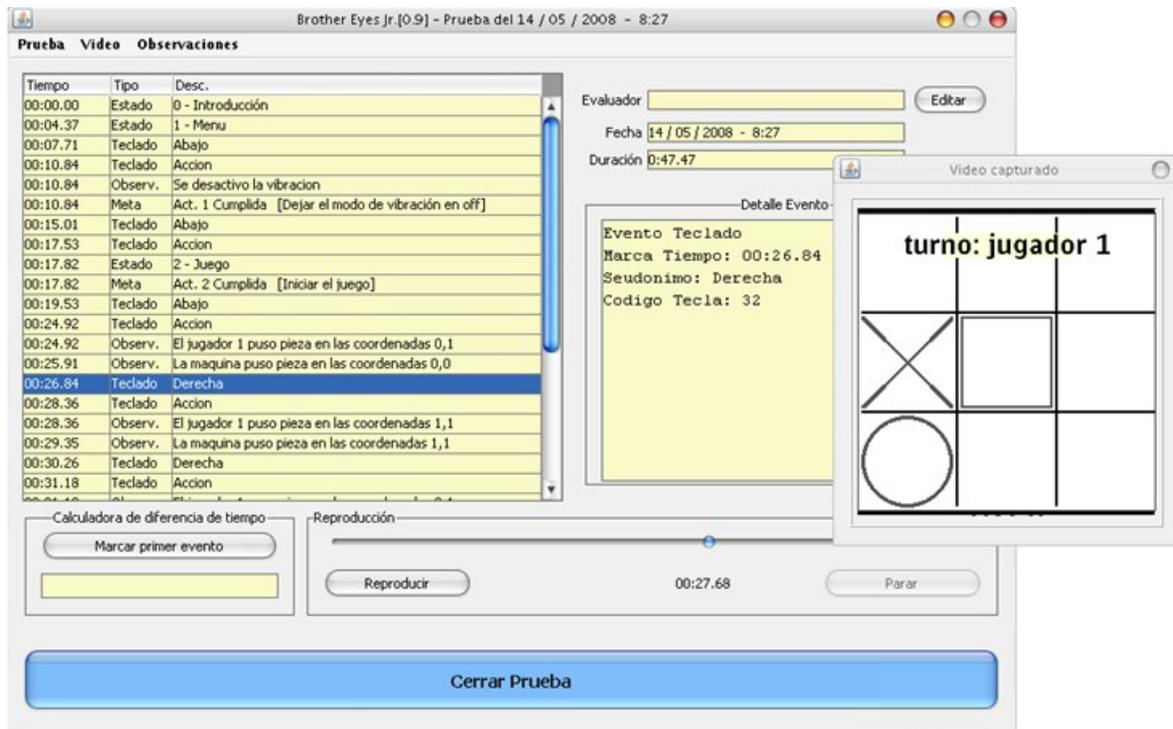
Cerrar

Ilustración 12-4: Tercer prototipo – Configuración de rangos de estadística



En la Ilustración 12-5 se puede ver el frame encargado de administrar una prueba. En él se puede ver una tabla con los eventos capturados, los controles de la reproducción de la prueba y otro frame con el video capturado. Este frame no recibió mayor modificación desde el segundo prototipo, pero se muestra en estos momentos para mostrar el cambio de LookAndFeel de la aplicación. El tipo de letra del menú, la forma de los botones son algunos de los cambios más evidentes de haber realizado esta modificación. Se espera que esto ayude a que la aplicación sea de mayor agrado al usuario.

Ilustración 12-5: Tercer prototipo – Ventana de una prueba



## 12.2 Evaluación de utilidad

Para realizar una evaluación mucho más objetiva de la herramienta, es que se la ha podido su cooperación a la empresa Amable (<http://www.amable.info>), la cual lleva varios años trabajando con el concepto de usabilidad y experiencia de usuario. Esta empresa, la cual se encuentra establecida en el centro de Santiago, ofrece servicios de testeo orientados a la experiencia de usuario. Tres representantes de esta empresa fueron los que se presentaron al momento de la evaluación y actuaron como grupo evaluador.

Las etapas para esta evaluación no difieren mucho de las utilizadas en la evaluación del segundo prototipo y estas fueron.

1. Dar una presentación para dar a conocer la herramienta y su funcionalidad.
2. Primera ronda de preguntas.
3. Demostración de la funcionalidad a través de un caso práctico.
4. Segunda ronda de preguntas.

## 5. Tiempo para contestar cuestionario de evaluación.

La presentación duró aproximadamente 15 minutos, donde mostró los principales conceptos con que trabaja la herramienta (evaluación, prueba, tipos de eventos). Posteriormente se mostró la herramienta y cada una de sus ventanas señalando la funcionalidad que esta tenían y como podían ser aprovechados.

La 1° ronda de preguntas duró no más de 10 minutos, ya que los integrantes del grupo de evaluadores, querían ver como funcionaba la herramienta en un caso práctico.

El caso práctico que se tenía preparado fue el mismo que se utilizó para la evaluación del segundo prototipo, con algunas pequeñas modificaciones en la lista de actividades asociadas a la evaluación. Esta vez ya se tenía un archivo creado con un par de pruebas ya ingresadas y donde los eventos de actividad ya habían sido establecidos en estas pruebas. Ya que el grupo era pequeño, se le pidió a uno de ellos ser el usuario-tester de la aplicación. En la aplicación se abrió la evaluación ya guardada y se creó una nueva prueba. En el dispositivo móvil se inicio la aplicación y se inicio busco dicho dispositivo desde la aplicación receptora. Cuando el proceso se encontró listo para iniciar la captura, se le entregó al usuario-tester el dispositivo móvil y el listado de las actividades que debería realizar y cuando el estuviera listo, se inició la prueba y la captura. Cuando el usuario-tester terminó de realizar las actividades indicadas, se terminó la captura de la prueba. Después se revisó, las estadísticas que producían todas las pruebas de la evaluación, se realizó una exportación de la evaluación a XML y la exportación del video capturado a un archivo AVI.

La 2° ronda de preguntas fue más que nada referente a algunas dudas técnicas que se tenía con respecto a la herramienta y de cómo ellos podrían adaptarla para su forma de trabajo. Por ejemplo, a ellos no les importa que la prueba se este capturando en tiempo real. Ellos solo necesitan un registro de ella para su posterior análisis. Es decir, mandar a un usuario-tester con el dispositivo a caminar por la ciudad o a que realice sus labores diarias, que realice la prueba guardando tanto los eventos como las capturas de pantalla en el dispositivo y una vez regresado el teléfono, trasladar esta información a la herramienta para su análisis.

Al momento de responder el cuestionario de evaluación uno de los evaluadores tuvo que marcharse por tener otros compromisos. Así que la evaluación se realizó solo con dos evaluadores.

Este cuestionario constaba con 10 preguntas que tenían que ser respondidas con una escala de 1 a 7, donde cada una de estas preguntas tenía un carácter bipolar. En la Tabla 12-1 se puede ver las calificaciones de cada uno de los evaluadores a las preguntas y el promedio final de estas.

<i>n°</i>	<i>Pregunta</i>	<i>Nota 1</i>	<i>Nota 7</i>	<i>Evaluador 1</i>	<i>Evaluador 2</i>	<i>Promedio</i>
1	<i>¿La información que identifica una evaluación (conjunto de pruebas de usabilidad) le parece correcta?</i>	<i>Correcta</i>	<i>Innecesaria</i>	7	7	7,0
2	<i>¿Qué tanto afecta a la prueba el tiempo que se demora una imagen, el ser enviada desde el dispositivo móvil a la aplicación?</i>	<i>No afecta</i>	<i>La aplicación es inutilizable</i>	7	7	7,0
3	<i>¿Qué tanto provecho le puede sacar el evento de tipo Observación?</i>	<i>Aprovechable al 100%</i>	<i>No justificable su presencia</i>	6	4	5,0
4	<i>¿Qué tanto provecho le puede sacar al evento de tipo Actividad?</i>	<i>Aprovechable al 100%</i>	<i>No justificable su presencia</i>	7	4	5,5

5	<i>¿Qué tanto provecho le puede sacar al evento de tipo Estado?</i>	<i>Aprovechable al 100%</i>	<i>No justificable su presencia</i>	4	4	4,0
6	<i>¿Qué tan valioso encontró la función de poder marcar el cumplimiento de las actividades asociadas a la evaluación?</i>	<i>Aprovechable al 100%</i>	<i>No justificable su presencia</i>	7	5	6,0
7	<i>¿Qué tan familiar le es a Ud. el formato de archivo de video al que se puede exportar una prueba (AVI)?</i>	<i>Es un formato bastante conocido</i>	<i>Nunca lo había escuchado</i>	7	7	7,0
8	<i>¿Le parece correcto el exportar una evaluación o prueba a un documento XML para poder utilizar los datos con otra aplicación?</i>	<i>Aprovechable al 100%</i>	<i>No justificable su presencia</i>	7	7	7,0
9	<i>¿Las estadísticas entregadas por el software, le parecen útiles al momento de evaluar la usabilidad de un software?</i>	<i>Aprovechable al 100%</i>	<i>No justificable su presencia</i>	6	7	6,5
10	<i>¿Cómo clasificaría en general a la aplicación?</i>	<i>Excelente</i>	<i>Basura</i>	7	7	7,0

Tabla 12-1: Tabla de resultados de evaluación

Como se puede ver, en general la herramienta a sido bien evaluada, llegando a tener varios puntos con nota igual a 7 y ninguno de los puntos fue evaluado con nota menor de 4. Lo cual habla muy bien de la herramienta.

Al final de la hoja con el cuestionario se dejó espació para que el evaluador pudiera dar a conocer alguna observación con respecto al software. De ambas observaciones se rescata que ambos piensan que no es necesario darle tanta importancia a la transmisión en tiempo real de los datos capturados por la API. Es más, con solo tener después de la prueba el video capturado no existiría mayor problema. Además, uno de ellos aconseja tomar el rumbo de las pruebas de usabilidad de aplicaciones web para dispositivos móviles. Él cree que este es el gran futuro de esta herramienta. Para esto, recomiendo implementar la API desarrollada en este proyecto dentro de un browser web de código abierto para dispositivos móviles. De esta manera se pueden realizar pruebas de aplicaciones web aprovechando lo realizado en este proyecto.

# Capítulo 13

## Conclusiones

Dentro del estudio del estado del arte que se realizó, se puede destacar la importancia de la usabilidad como factor primario dentro de la calidad de un software. Se dio a entender que sin él, las aplicaciones para dispositivos software fracasarán. Por este motivo, es necesario mejorar las técnicas de evaluación de usabilidad que se aplican a este tipo de software.

La variedad de métodos de evaluación de usabilidad, con sus ventajas y desventajas, permiten a los encargados poder elegir, según: las experiencias del equipo de trabajo, lo que se desea evaluar o medir, las características del software, los costos asociados, las restricciones de tiempo, presupuesto, entre otros, el método que más se acomode y no estar dependiendo de uno solo que puede traer más problemas que soluciones porque no se adecua a las necesidades.

Los dispositivos móviles, que por sus pequeños tamaños, limitada cantidad de teclas, variados tamaños de pantalla, incomodidad de uso, etc., son un verdadero reto para los diseñadores de software al momento de hablar de usabilidad. Además, la heterogeneidad de estos, dificulta aún más cuando se desea desarrollar una aplicación genérica o que por lo menos funcione en un conjunto de terminales.

No existe mucha información sobre métodos especialmente diseñados para la evaluación de usabilidad de aplicaciones de dispositivos móviles o, mejor dicho, que se adapten correctamente a las características de la interacción entre el usuario y la aplicación para móviles. Solo es posible encontrar pequeñas adaptaciones a los métodos tradicionales para poder utilizarlos en aplicaciones móviles y que no respetan, por ejemplo, el contexto de uso u otro factor importante en una interacción de este tipo. Dichas adaptaciones tienen ventajas bastantes considerables, pero también tienen desventajas que no permiten un verdadero aprovechamiento de las mismas.

Lo más problemático no solo es lo mencionado en el párrafo anterior, sino, que aunque se ocupen métodos tradicionales de evaluación de usabilidad, no se cuenta con herramientas adecuadas que ayuden correctamente a desarrollar dicha evaluación de forma correcta. Esto se ve con mayor énfasis en las pruebas de usabilidad.

Por este motivo, se decidió abarcar este último problema se presentó una propuesta de solución, la cual es un software para poder automatizar algunos procesos de captura de eventos en el transcurso de las pruebas de usabilidad.

Se realizó un estudio de factibilidad de la solución y se establecieron los requerimientos más importantes como lo es la captura de los eventos, el envío de los eventos a la aplicación receptora, el sistema de respaldo y la exportación para poder asegurar una compatibilidad con otras aplicaciones. En el análisis de los requerimientos se le dio mayor énfasis a la captura y al envío de estos a la aplicación principal. Lo mismo se hizo con el diseño del software.

En este informe se presentaron los diagramas hechos tanto para el análisis como para el diseño, además de los casos de uso extendidos.

Se lograron desarrollar tres prototipos de la herramienta. El primer, es capaz de capturar los eventos ocurridos en la aplicación que se desea evaluar, enviar dichos eventos a la aplicación principal y recibir los eventos. Además se ha logrado presentar un pequeño reproductor de la prueba capturada, la cual puede ser controlada completamente por el usuario que esta interesado en dicha prueba.

Al segundo prototipo se le incorporó la opción de exportar el video capturado a un formato estándar, para no obligar al evaluador el ocupar esta herramienta para ver el video. Además, se mejoró la implementación de la gestión de archivos de la herramienta. Ahora se puede guardar un respaldo de la evaluación para poder ser abierto más tarde.

Se hicieron pruebas de utilidad sobre el segundo prototipo, donde los evaluadores calificaron la herramienta desarrollada. Estas han concluido que la herramienta no es útil para poder capturar lo visualizado en la pantalla del dispositivo móvil. Aún en el caso de que el juego que se esté evaluando realice pocos redibujados de la pantalla, el tiempo que

demora la transferencia de la misma, provocan una alteración no aceptable en la interacción con el jugador. La captura de los demás eventos es bastante aceptable.

El que la herramienta sea intrusiva en el juego, no permite evaluar la usabilidad de aquellos juegos de los que no se tiene el código fuente, pero, por otra parte, nos ayuda a obtener datos más valiosos y así poder realizar un mejor análisis.

Por último, en la tercera versión se implementó todo lo que faltaba por cumplir como por ejemplo, el resumen de las pruebas realizadas y el cálculo de algunas estadísticas que ayuden al análisis de usabilidad. Además se mejoraron los problemas que se encontraban en el segundo prototipo, como por ejemplo, se disminuyó el tamaño del video exportado e, implementando un thread (hilo de ejecución) independiente para el tráfico por Bluetooth por parte de la API, se disminuyó considerablemente el tiempo de pausa que existía entre cada refresco de pantalla haciendo, de esta forma, que la API no perjudicará la ejecución de la aplicación a evaluar.

Con el estudio del estado del arte sobre el tema se ha logrado cumplir con los tres primeros objetivos específicos planteados al comienzo del proyecto. Con estos mismos conocimientos se lograron definir los requerimientos mínimos que debería cumplir la solución.

Tras el desarrollo del tercer prototipo se puede afirmar que se han logrado cumplir todos los requerimientos establecidos, tanto los funcionales como los no funcionales. Por este motivo el objetivo que hacía mención a desarrollar un prototipo que cumpliera con los requerimientos propuestos, ha logrado ser cumplido exitosamente.

Y, por último, gracias a la empresa Amable, se realizó una prueba de utilidad de la herramienta logrando obtener muy buenas calificaciones y llegando a ser muy bien evaluado por el grupo.

Con todo esto, se puede afirmar, que el objetivo, *apoyar las pruebas de usabilidad de aplicaciones para dispositivos móviles a través de un prototipo de herramienta software para ello*, ha sido alcanzado satisfactoriamente.

## Referencias

- [C1] Ghaoui, C., *Encyclopedia of Human Computer Interaction*, Liverpool John Moores University, Uk, 2006
- [I1] Ivory, M., *The State of the Art in Automating Usability Evaluation of User Interfaces*, University of California, Berkeley, 2001.
- [K1] Kaikkonen, A., “Usability Testing of Mobile Applications: A Comparison between Laboratory and Field Testing”, *Journal of Usability Studies*, 2005.
- [M1] Marinilli, Mauro, “*Professional Java User Interfaces*”, 2006.
- [N1] \*\*\*, *Enterprise Applications User Experience Library*, NOKIA, fecha de revisión abril de 2007, disponible en <http://www.forum.nokia.com/main/resources/documentation/usability/enterprise.html>
- [N2] \*\*\*, *Game User Experience Library*, NOKIA, fecha de revisión abril de 2007, disponible en <http://www.forum.nokia.com/main/resources/documentation/usability/games.html>
- [N3] \*\*\*, *Mobile Game Playability Heuristics 1.0*, NOKIA, fecha de revisión abril de 2007, disponible en <http://www.forum.nokia.com/main/resources/documentation/usability/games.html>
- [N4] Nielsen, J., *Entrevista a Jakiob Nielsen, Experto en usabilidad*, diario online ElPais.com. Disponible en [http://www.elpais.com/articulo/internet/servicios/moviles/sean/faciles/usar/fracasaran/elppor tec/20051126elpepunct\\_1/Tes](http://www.elpais.com/articulo/internet/servicios/moviles/sean/faciles/usar/fracasaran/elppor tec/20051126elpepunct_1/Tes), fecha de revisión abril de 2007.
- [N5] Nielsen, J., *Ten Usability Heuristics*, 1994, Disponible en su página personal: [http://www.useit.com/papers/heuristic/heuristic\\_list.html](http://www.useit.com/papers/heuristic/heuristic_list.html)
- [N6] \*\*\*, *User Experience Checklist for Java ME Applications 1.0*, NOKIA, fecha de revisión abril de 2007. Disponible en <http://www.forum.nokia.com/main/resources/documentation/usability>
- [N7] Nuez, A., *Usabilidad Estratégica*, Xperience Consulting, 2002
- [P1] Portillo, J., Carretero, N., *Dispositivos portátiles y usabilidad*, Universidad Politécnica de Madrid, disponible en <http://www.ceditec.etsit.upm.es>
- [R1] Rusu, C., *Interacción persona-computador MII712*, Escuela de Ingeniería Informática, Pontificia Universidad Católica de Valparaíso. 2006.

[S1] Sayago, Sergio, *Técnicas de Ingeniería de Usabilidad y metodología de diseño conceptual en algunas aplicaciones informáticas*, Universitat Pompeu Fabra, 2003.

Disponible en su web <http://www.tecn.upf.es/~ssayag>

[U1] Ugalde, C., *Metodología de evaluación de usabilidad de sistema de software*, Pontificia Universidad Católica de Valparaíso, 2004

[U2] User Vision, *Mobile Usability White Paper*, Edinburgh, Febrero 2006.

## Apéndice A: Cuestionario para evaluaciones de aplicaciones

### J2ME

Este cuestionario fue propuesto por

This questionnaire can be used as an acceptance test at the end of a test session with users, or for a first-cut usability evaluation. It should not be used as a substitute of usability tests.

#### **Section A: Your experience with the program**

How long have you worked with the program?

- 1 hour or less
- 1 hour to 1 day
- 1 day to 1 week
- 1 week to 1 month
- 1 to 6 months
- more than 6 months

#### **Section B: Your overall reactions**

Please quantify your reactions to the program:

- terrible – wonderful
- difficult – easy
- frustrating – satisfying
- boring – stimulating
- rigid – flexible
- unhelpful – productive

**Section C: Your past experience**

How many cell similar devices have you used before?

- 1
- 2
- 3
- 4 or more

Rate your familiarity with Java applets: never used before – very high

In the following items check those that you have personally used and with which you are familiar:

PC	Smart phone	Handheld device	Portable Game Console
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SMS	Web Browser	Portable MP3 Player	Digital Camera
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Java	Java Virtual Machine (JVM)	J2ME applet	WiFi Network
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

**Section D: Terminology**

How close is the program's terminology to what you would expect?

not at all – very close

- 

The terminology is:

ambiguous – precise

- 

Technical terminology is used:

too frequently – appropriately

-

Messages are:

confusing – clear

Messages positions on the screen are:

consistent – inconsistent

How often do error messages clarify the problem?

never – always

Error messages seem:

annoying – pleasant

Error messages help to solve the problem?

never – always

**Section E: Program feedback**

The program keeps you informed about what it is doing?

never – always

How frequently does the pointer shape (if any) help in showing the current application state?

never – always

How often does the program freeze without showing what it is doing?

never – always

**Section F: Learning to use the application**

Learning to use the application was:

difficult – easy

Getting started with the application was:

difficult – easy

Learning advanced features was:

difficult – easy

Exploring the features by trial and error was:

risky – safe

Discovering new features was:

difficult – easy

Configuring the application's preferences was:

difficult – easy

**Section G: Display organization**

The display organization (screens, forms, etc.) was:

confusing – clear

Characters and icons were:

hard to read – very readable

The command icons were:

confusing – clear

The overall graphic appearance was:

annoying – pleasing

The application had the same look as other programs:

totally different – exactly the same

**Section H: Navigation**

It was possible to cancel an operation or navigate back to a previous screen:

never – always

Navigation keys and navigation commands were:

confusing – clear

confusing – clear

The number of screens was:

too many – about right

Reaching a given screen was:

difficult – easy

**Section I: Help support**

Help content was:

confusing – clear

Other manuals (if any) were:

confusing – clear

Help material covers the program features:

inadequately – completely

Help activation was:

slow – quick

Help material was easy to find:

never – always

Learning to use the application by using the help support was:

difficult – easy

**Section J: Deployment**

Installation was:

difficult – easy

Launching the application the first time was:

confusing – clear

Launching the application was:

tricky – straightforward

The waiting time for launching the application was:

extremely long – reasonable

Upgrading to a newer version was:

difficult – easy

**Section K: Mobile experience**

Switching the application on or off (pausing and restoring it) was:

difficult – easy

Operations required extra attention:

never – always

How many times did you have to start an operation all over again?

never – more than three times

The application respected my privacy:

never – always

The application handled interruptions such as phone warnings, phone calls, other external situations:

badly – very well

Remote connections were signaled:

confusingly – clearly

The application asked permission before making remote connections:

never – always

The application respected the current phone settings, such as ringer off:

never – always

## Apéndice B: Manual de usuario

Ver *Manual de usuario* en:

---

<sup>1</sup> El formato AVI más que un formato, es un formato contenedor, que permite tener un formato de video y varios de audio