FENG MOBILE

"Acceso a Feng Office para dispositivos móviles"

Annia Díaz - Vicente Castelló

Instituto de Computación,
Facultad de Ingeniería,
Universidad de la República,
Montevideo, Uruguay
2012

Tutor

Eduardo Fernández

Resumen

Recientemente se ha vuelto popular el uso de herramientas colaborativas basadas en la web. Feng Office es una de ellas, brinda las funcionalidades necesarias para administrar tareas, calendario, contactos, documentos entre otros elementos que están presentes en la gestión de un proyecto o negocio. El uso de dispositivos móviles y sus prestaciones también se han visto incrementados, al igual que su uso como herramienta de trabajo. Revisar el correo, agendar reuniones y manejar contactos son tareas en las que comúnmente pueden ser utilizados. Dadas las tendencias mencionadas surge la necesidad por parte de Feng Office de brindar a los usuarios la posibilidad de poder utilizar las funcionalidades de dicha herramienta en dispositivos móviles.

En este proyecto se diseña e implementa una solución que permita el acceso a Feng Office desde dispositivos móviles. Las actividades realizadas durante el proyecto fueron: relevamiento de plataformas y tecnologías aplicables, estudio de buenas prácticas y patrones de diseño de interacción, análisis de los enfoques tomados por la competencia de Feng Office, diseño de una solución y desarrollo de un prototipo que permita probar los aspectos más importantes del sistema diseñado.

Como resultado del diseño de la solución y desarrollo del prototipo, se creó una aplicación para Android, se diseñó un algoritmo de sincronización genérico que implementa la sincronización de objetos entre cliente y servidor, se extendió la aplicación web de Feng Office para exponer una interfaz que permite acceder a la sincronización. Consideramos que los resultados obtenidos por este proyecto son un buen comienzo para la construcción de una solución a brindar a usuarios finales.

Palabras clave: Sistemas colaborativos, dispositivos móviles, sincronización, diseño de interacción.

Índice general

1. Introducción **1**
1.1. Objetivos . . . 1
1.2. Proyecto . . . 2
1.2.1. Análisis . . . 2
1.2.2. Diseño y construcción . . . 3
1.2.3. Integración y documentación . . . 3
1.3. Organización del documento . . . 3

2. Estado del Arte **5**
2.1. Dispositivos móviles . . . 5
2.1.1. Smartphones . . . 5
2.1.2. Tabletas . . . 6
2.1.3. Celulares con plataforma J2ME . . . 7
2.1.4. Comparativo . . . 7
2.2. Evaluación de plataformas . . . 10
2.2.1. Aplicaciones nativas . . . 11
2.2.2. Generadores de aplicaciones . . . 21
2.2.3. Aplicaciones web - HTML5 . . . 27
2.3. Análisis del mercado de S.O. para móviles . . . 32
2.3.1. Mercado mundial . . . 32
2.3.2. Mercado América Latina . . . 34
2.4. Análisis de la competencia a Feng Mobile . . . 34
2.4.1. 37signals - Basecamp . . . 34
2.4.2. Central Desktop . . . 37
2.4.3. Zoho . . . 39
2.4.4. Google Apps . . . 41
2.4.5. Comparativo . . . 42
2.5. Diseño de interacción . . . 43
2.5.1. Patrones de diseño . . . 43
2.5.2. Diseño de interfaces para móviles . . . 44

3. Requerimientos del sistema y alcance del proyecto **47**
3.1. Requerimientos del sistema . . . 47
3.1.1. Requerimientos funcionales . . . 47
3.1.2. Requerimientos no funcionales . . . 50

3.2. Alcance del proyecto . . . 51
3.2.1. Ingresar una tarea . . . 51
3.2.2. Buscar una tarea . . . 52
3.2.3. Ver detalles de una tarea . . . 52
3.2.4. Completar una tarea . . . 52
3.2.5. Reabrir una tarea . . . 52
3.2.6. Editar tarea . . . 52
3.2.7. Ver tareas pendientes . . . 52
3.2.8. Comentarios . . . 52
3.2.9. Notificaciones . . . 53
3.2.10. Sincronización . . . 53

4. Solución propuesta **55**
4.1. Selección de Android como plataforma . . . 55
4.2. Diseño de interfaz de usuario . . . 56
4.2.1. Lineamientos . . . 56
4.2.2. Bosquejos . . . 59
4.2.3. Subtareas . . . 61
4.3. Arquitectura . . . 63
4.3.1. Cliente . . . 63
4.3.2. Servidor . . . 64
4.3.3. Comunicación y sincronización . . . 65
4.3.4. Seguridad . . . 65

5. Implementación del prototipo **67**
5.1. Cliente . . . 67
5.1.1. Presentación . . . 67
5.1.2. Acceso a datos . . . 68
5.1.3. Autenticación y sincronización . . . 69
5.2. Servidor . . . 70
5.3. Seguridad . . . 71
5.4. Sincronización . . . 72
5.5. Entorno de desarrollo . . . 73

6. Pruebas **75**
6.1. Casos de pruebas . . . 75
6.2. Resultados obtenidos . . . 82

7. Conclusiones y trabajo futuro **83**
7.1. Conclusiones . . . 83
7.2. Trabajo futuro . . . 84

A. Glosario **87**

B. Patrones de diseño **91**

C. Algoritmo de sincronización **95**

Bibliografía **97**

1. Introducción

En los últimos años se ha vuelto popular el uso de las herramientas de gestión de proyectos colaborativas que permiten gestionar recursos, clientes y requerimientos. Sirven para realizar el seguimiento de proyectos mostrando a qué se le dedica más tiempo, obtener indicadores, estimar, tomar decisiones y proyectarse en el tiempo. Facilitan la comunicación en el equipo de trabajo y el registro de información.

Feng Office es un sistema de gestión colaborativa desarrollado inicialmente en la Facultad de Ingeniería de la UdelaR como un proyecto de grado. Al culminar el proyecto su código fuente fue liberado bajo el nombre de OpenGoo, fundando una comunidad que permitió difundir la herramienta. Por medio de una interfaz web, permite a un grupo de personas geográficamente distribuidas trabajar de forma colaborativa compartiendo información a través de internet. Cuenta con varios módulos que permiten manejar: notas, contactos, calendarios, documentos y tareas entre otros. Sobre los objetos que maneja cada módulo existen diferentes acciones: comentar, suscribirse a cambios, relacionar objetos y búsquedas.

Con el creciente uso de "smartphones" y tabletas (ver sección 2.1), surge la necesidad de contar con una solución que permita acceder a Feng Office desde estos dispositivos. Esto permitiría a los usuarios acceder a la información en cualquier momento y lugar. Utilizando el foro de anuncios de Feng Office se realizó una consulta a los usuarios acerca de una versión para móviles [36]. Los usuarios respondieron demostrando interés y proponiendo funcionalidades.

Este proyecto de grado tiene como propósito construir una solución que permita acceder a Feng Office mediante el uso de dispositivos móviles.

1.1. Objetivos

- El alto nivel de fragmentación existente en el mercado de los dispositivos móviles determina que uno de los principales objetivos de este proyecto sea hacer un relevamiento de las plataformas sobre las cuales es posible implementar el sistema requerido. En base a dicho estudio se seleccionará la plataforma que más se adecue a las necesidades del cliente.
- Se espera como resultado de este proyecto diseñar y construir una solución que permita el acceso a Feng Office.

- Hacer un relevamiento de las mejores prácticas y patrones de diseño de interacción más comunes en el desarrollo de aplicaciones para este tipo de dispositivos.

1.2. Proyecto

Para este proyecto fue definida una metodología que permitiera alcanzar los objetivos planteados, identificando tres fases principales:

1. Análisis
2. Diseño y construcción
3. Integración y documentación

En la siguiente figura se muestra el cronograma seguido.

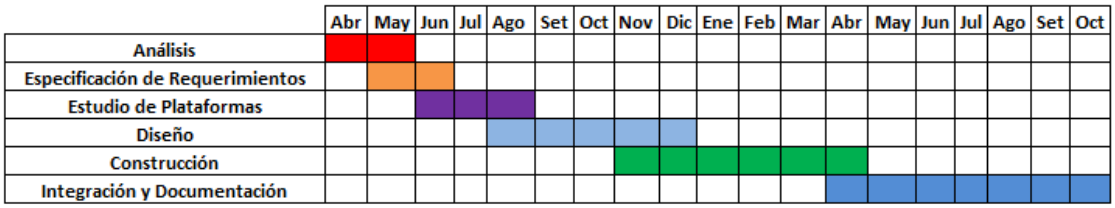

Figura 1.1.: Cronograma

A continuación se detallan las actividades realizadas en cada una de las fases.

1.2.1. Análisis

En esta fase se obtuvieron los requerimientos y el conocimiento necesario para llevar a cabo el proyecto.

En un principio se estudiaron las funcionalidades de Feng Office, y se definió que se utilizaría dicha herramienta para gestionar este proyecto, con lo que el equipo se familiarizó con el uso de todas las funcionalidades que proporciona la herramienta. Se utilizó para organizar el trabajo, almacenar documentos y registrar algunas tareas.

El relevamiento de requerimientos se hizo con el cliente del proyecto mediante reuniones semanales, a partir de las cuales se establecieron los objetivos y se generó la especificación de requerimientos.

Se realizó un estudio de las posibles plataformas y enfoques de desarrollo para el planteo de una solución. También fueron analizadas las aplicaciones para móviles desarrolladas por la competencia.

En base a bibliografía recomendada[1] e información de los sitios de las distintas plataformas y sistemas operativos se estudiaron los conceptos principales de diseño de interfaz. Se analizaron recomendaciones y patrones de diseño aplicables al tipo de sistema a desarrollar.

Teniendo en cuenta los objetivos y el relevamiento realizado se definió el alcance del proyecto.

1.2.2. Diseño y construcción

Tomando como base los conocimientos adquiridos en la etapa de análisis se seleccionó la plataforma para la que sería implementada la solución. Luego fueron definidos los estándares a utilizar para el diseño de la interfaz gráfica. Se diseñaron algunas pantallas y la interacción.

Se diseñó la arquitectura del sistema, los mecanismos de comunicación con el servidor de Feng Office y la estrategia de sincronización. Se implementó un prototipo según lo definido en el alcance.

1.2.3. Integración y documentación

En esta etapa se realizó la integración del prototipo con el servidor de Feng Office. Por lo que se extendió una API que permite la comunicación desde el cliente. Se hicieron pruebas del prototipo y se corrigieron algunos errores.

También en esta etapa se escribió el informe del proyecto y se preparó la presentación.

1.3. Organización del documento

Los capítulos en este documento están organizados según el orden cronológico en que fueron ejecutadas las actividades que se describen las cuales son:

Estado del arte: Muestra el relevamiento obtenido en la fase de análisis.

Requerimientos del sistema y alcance del proyecto: Muestra los requerimientos detectados durante reuniones realizadas con el cliente, define cuales fueron implementados en el marco de este proyecto y los resultados esperados.

Solución propuesta: Se muestran los detalles de diseño tanto de interfaz como de arquitectura de la solución recomendada.

Implementación del prototipo: Contiene información acerca del prototipo, diseño e implementación.

[1] Recomendada por el cliente, tutor y obtenida en internet.

Pruebas: Muestra las pruebas que se realizaron para evaluar la solución desarrollada.

Conclusiones y trabajo futuro: Se evalúan los resultados obtenidos en el proyecto y se plantea el camino a seguir para el desarrollo del sistema completo.

2. Estado del Arte

Hoy en día, los dispositivos móviles se han convertido en una de las herramientas más utilizadas a nivel mundial que nos permiten estar comunicados en todo momento y lugar. La gran gama de funcionalidades de los equipos, las diferentes alternativas de precios y las modalidades de pago han sido tal vez las variables que han tenido mayor peso para explicar tal difusión.

Con las capacidades que tienen los smartphones y otros dispositivos móviles es cada vez más frecuente su uso para acceder a información en cualquier momento y lugar. Es común que para usuarios de Feng Office sea necesario acceder a dicho sistema cuando no están utilizando una computadora con acceso a la web. Usos típicos serían agendar una reunión, ingresar una tarea mientras se está hablando con otra persona, acceder a un documento durante una presentación o registrar un contacto luego de una charla con un cliente. Por lo anterior es que surge la necesidad de contar con un acceso a Feng Office desde dispositivos móviles.

En este capítulo se muestra el relevamiento hecho de los dispositivos móviles y las plataformas en las cuales se podría implementar la aplicación. También se muestran las aplicaciones más usadas en la actualidad que se asemejan a Feng Office. Por último se muestran conceptos referentes al diseño de interfaces para móviles.

2.1. Dispositivos móviles

Dentro de los dispositivos móviles más utilizados se destacan smartphones, tabletas y teléfonos celulares con plataforma J2ME. A continuación se muestran las características principales que los definen.

2.1.1. Smartphones

Un smartphone, o teléfono inteligente, es además de un teléfono celular, un dispositivo que ofrece un conjunto de funcionalidades y capacidad de procesamiento que lo asemejan en su uso, al de una computadora portátil. Entre sus características principales están:

- Soportan completamente un cliente de correo electrónico, navegador, lista de contactos, calendarios, notas y recordatorios.

- Permiten el agregado de programas por parte del usuario para adaptarlo a sus necesidades.
- Usualmente cuentan con hardware que extiende las posibilidades de uso del dispositivo, como ser cámara, GPS, acelerómetro y brújula entre otros.
- Cuentan con una interfaz para el ingreso de datos más avanzada que un teléfono convencional como ser: teclados querty, teclado en pantalla y pantallas multitouch.
- Permiten conectarse a internet de varios modos: 3G, WiFi, GPRS. Algunos modelos permiten compartir la conexión a internet con otros dispositivos.
- Incluyen software para leer documentos en distintos formatos, como planillas, documentos de texto y presentaciones.
- Permiten reproducir varios formatos de audio y video, e incluyen también un reproductor multimedia. Como suelen tener una pantalla con buena resolución pueden ser usados para mirar videos o televisión por internet.
- Permite ejecutar varias aplicaciones al mismo tiempo. Por ejemplo recibir llamadas, revisar la agenda, ver videos, mientras se ejecutan tareas de fondo.

Ejemplos de smartphones son: Serie iPhone de Apple, Serie Galaxy de Samsung, Serie Galaxy Nexus de Google/Samsung, Serie BlackBerry de Research In Motion, Serie Lumia de Nokia, Serie Ascend de Huawei, Serie Defy de Motorola, Serie Optimus de LG, Serie One de HTC, Serie Xperia de Sony Mobile Communications.

Se realizó un relevamiento de los sistemas operativos y fabricantes más usados por los smartphones que se muestra en la siguiente tabla.

Sistema Operativo	Fabricante	Lenguaje de Programación
Android	Google	Java
iOS	Apple	Objective-C
BlackBerry OS	RIM	Java
Symbian OS	Symbian Foundation	C++
Windows Phone	Microsoft	C#

Cuadro 2.1.: Sistemas operativos de distintos smartphones

2.1.2. Tabletas

Luego de los smartphones surgieron los “tablets” o tabletas, dispositivos que se asemejan más a una computadora portátil por tener una pantalla más grande, más capacidad de almacenamiento y gráficos de mejor calidad que los smartphones pero aún mantienen las funciones de teléfono.

En las tabletas se interactúa por medio de una pantalla táctil. No tienen teclado pero a muchas se les puede conectar uno convirtiéndolas prácticamente en una computadora portátil. Al tener una pantalla grande son más prácticas para la lectura y para ver videos, suelen ser utilizadas también para realizar dibujos, muchas veces ayudándose de un lápiz especial.

Muchos fabricantes han incursionado en la producción de tabletas, por ejemplo, Samsung, BlackBerry, Sony, Toshiba, Acer, Hewlett Packard, Microsoft, entre otros.

2.1.3. Celulares con plataforma J2ME

Es la plataforma de desarrollo que comúnmente estaba disponible en los teléfonos celulares previo a la aparición de los smartphones.

Java 2 Micro Edition (J2ME) es una versión de la máquina virtual de Java pensada para correr en dispositivos con hardware y recursos limitados [34]. Es una de las plataformas predominantes dentro del mercado de dispositivos móviles [31] y eso se debe principalmente a que es incorporada en dispositivos "no-smartphones" de bajo costo, que aún son vendidos a millones de personas en el mundo [40].

Se utiliza la misma tecnología para ejecutar aplicaciones J2ME en dispositivos móviles e integrados: teléfonos móviles, reproductores Blu-ray, dispositivos multimedia digitales, entre otros [41].

2.1.4. Comparativo

Los dispositivos móviles con la plataforma J2ME están muy limitados en comparación con los smartphones y tabletas. Las tabletas y los smartphones son dispositivos bastante diferentes en cuanto a lo que es el tamaño y principalmente la pantalla, sin embargo en lo que se refiere al hardware son muy parecidos (procesador, memoria RAM, incluso la capacidad de almacenamiento).

Se realizó un relevamiento que se resume en las tablas 2.2 y 2.3, donde se muestran las características principales de smartphones y tabletas a fin de poder hacer una comparación de sus prestaciones [22, 30, 8, 18, 23, 10, 29, 54]. Se evalúan los siguientes conceptos: S.O., procesador, pantalla, RAM, capacidad de almacenamiento, conectividad, cámara, GPS, WiFi y Bluetooth.

Samsung Galaxy S3	
S.O.	Android 4.0 Ice Cream Sandwich con TouchWiz
Procesador	Cuatro núcleos Exynos Quad Core 1,4 Ghz
Pantalla	4.8" HD Super AMOLED (1280x720) display
RAM	1 GB
Memoria	16 o 32 GB ranura microSD
Conectividad	4G LTE
Cámara	8 mpx Sensor retroiluminado, Flash LED, , detector de rostros, grabación de video en HD hasta 30 frames por segundo, geoetiquetado y autoenfoque
GPS	GPS asistido y GLONASS
WiFi	802.11 a/b/g/n
Bluetooth	4.0
iPhone 4S	
S.O.	iOS 5
Procesador	A5 dual core
Pantalla	Multi-Touch panorámica de 3,5 '' (diagonal)
RAM	512MB
Memoria	16, 32 o 64 GB
Conectividad	3.5G
Cámara	iSight de 8 mpx, Flash LED, detector de rostros, grabación de video en HD (1080p) hasta 30 frames por segundo con audio, geoetiquetado de fotos y videos
GPS	GPS asistido y GLONASS
WiFi	802.11 b, g, n (802.11n solo a 2,4 GHz)
Bluetooth	4.0
BlackBerry Curve 9380	
S.O.	BlackBerry OS 7.0
Procesador	800MHz
Pantalla	3.2'' (360x480)
RAM	512MB
Memoria	hasta 32GB ranura microSD
Conectividad	3G
Cámara	5 mpx, 2592x1944 px, flash LED, detección de rostro, estabilizador de imagen, video VGA, geoetiquetado y autoenfoque
GPS	GPS asistido
WiFi	802.11 b/g/n
Bluetooth	v2.1 A2DP, EDR
Nokia Lumia 900	
S.O.	Microsoft Windows Phone 7.5 Mango
Procesador	1.4GHz
Pantalla	ClearBlack de 4.3 '', AMOLED touchscreen capacitivo, 16M colores
RAM	512MB
Memoria	16GB
Conectividad	GSM 850/900/1800/1900 HSDPA (High Speed Downlink Packet Access, es una tecnología basada en la red 3G que pueden soportar velocidades de hasta 7,2 Mbits por segundo) 850/900/1900/2100
Cámara	8 mpx, 3264x2448 px, óptica Carl Zeiss, flash LED dual, video 720p@30 frames por segundo, cámara frontal 1.3mpx 720p, geoetiquetado y autoenfoque.
GPS	GPS asistido
WiFi	802.11 b/g/n
Bluetooth	v2.1 A2DP, EDR

Cuadro 2.2.: Características principales de Smartphones

Samsung Galaxy TAB 10.1	
S.O.	Android 3.2 HoneyComb
Procesador	1GHz dual core NVIDIA Tegra 2
Pantalla	10.1'' widescreen, (1280x800) WXGA TFT LCD, 4-way rotation
RAM	1GB
Memoria	16,32,64 GB ROM
Conectividad	EDGE/GPRS 850/900/1800/2100, HSPA +21 850/900/1900/2100
Cámara	3mpx, flash LED, grabación de video en HD, cámara frontal 2 mpx y autoenfoque
GPS	GPS asistido
WiFi	802.11 a/b/g/n
Bluetooth	3.0
BlackBerry PlayBook	
S.O.	BlackBerry PlayBook OS 2.0
Procesador	doble núcleo a 1GHz
Pantalla	7'' LCD, (1024 x 600)
RAM	1GB
Memoria	16,32,64 GB
Conectividad	4G, 4G LTE
Cámara	5 mpx, cámara frontal de 3 mpx, grabación en HD
GPS	
WiFi	802.11 a/b/g/n
Bluetooth	2.1 + EDR
iPad	
S.O.	iOS 5
Procesador	A5 dual core
Pantalla	Muti-Touch panorámica de 9,7 ” (en diagonal) retroiluminada por LED con tecnología IPS, 2.048 x 1.536 px a 264 px por pulgada
RAM	512MB
Memoria	16,32,64 GB
Conectividad	UMTS/HSPA/HSPA+/DC-HSDPA 850/900/1900/2100 MHz, GSM/EDGE 850/900/1800/1900 MHz, 4G LTE 700/ 2100 MHz sólo en EE. UU. y Canadá
Cámara	iSight de 5 mpx, grabación de vídeo en HD (1080p) de hasta 30 frames por segundo con audio, estabilización de video, geoetiquetado de fotos y videos.
GPS	GPS asistido y GLONASS
WiFi	802.11a/b/g/n
Bluetooth	4.0
Motorola Xoom	
S.O.	Android 3.1 Honeycomb
Procesador	1GHz Nvidia Tegra Dual Core
Pantalla	10.1'' WXGA (1280x 800 px a 150 px por pulgada)
RAM	1GB
Memoria	hasta 32 GB
Conectividad	3G
Cámara	5mpx, Dual-LED flash, grabación en HD 720p
GPS	GPS asistido
WiFi	802.11 b/g/n
Bluetooth	Estéreo 2,1+EDR+HID

Cuadro 2.3.: Características principales de Tabletas

2.2. Evaluación de plataformas

En esta sección se presenta un análisis de los enfoques de desarrollo considerados: aplicaciones nativas, generadores de aplicaciones y aplicaciones web (HTML5). Dicho análisis fue realizado en octubre del 2011, con algunas actualizaciones en junio del 2012.

La siguiente figura fue extraída del informe Developer Economics 2012 publicado en Junio de 2012 por la consultora Vision Mobile [25], hace una comparación de las distintas plataformas en base a ciertos factores como la llegada a los usuarios, posibilidades de generar ganancias, la facilidad de programación, el costo de desarrollo, la facilidad de aprendizaje, el acceso a SDK's o API's y el soporte de la comunidad. En ella se puede apreciar que plataformas como la web, iOS y JavaME tienen una mayor llegada a los usuarios. iOS también se presenta como una de las que más brinda posibilidades para generar ingresos. Por otro lado Android figura como la plataforma en la que es más difícil generar ingresos, pero este es el único aspecto negativo que se le identifica. Seguramente esto se deba a la cantidad de aplicaciones disponibles. Esta gráfica resulta útil para conocer algunos de los criterios por los que se suelen evaluar las plataformas.

Figura 2.1.: Plataformas elegidas por desarrolladores en base a cinco factores [25]

La figura 2.2, también extraída del informe Developer Economics 2012 publicado en Junio de 2012 por la consultora Vision Mobile[25], muestra el porcentaje de desarrolladores de aplicaciones para las diez plataformas más usadas entre 2010 y el primer semestre del 2012. En esta gráfica lo que se puede ver es que Android, iOS y la web son las plataformas más utilizadas por los desarrolladores. También se puede apreciar que plataformas como JavaME, Symbian y Flash van comenzando a perder popularidad entre los desarrolladores.

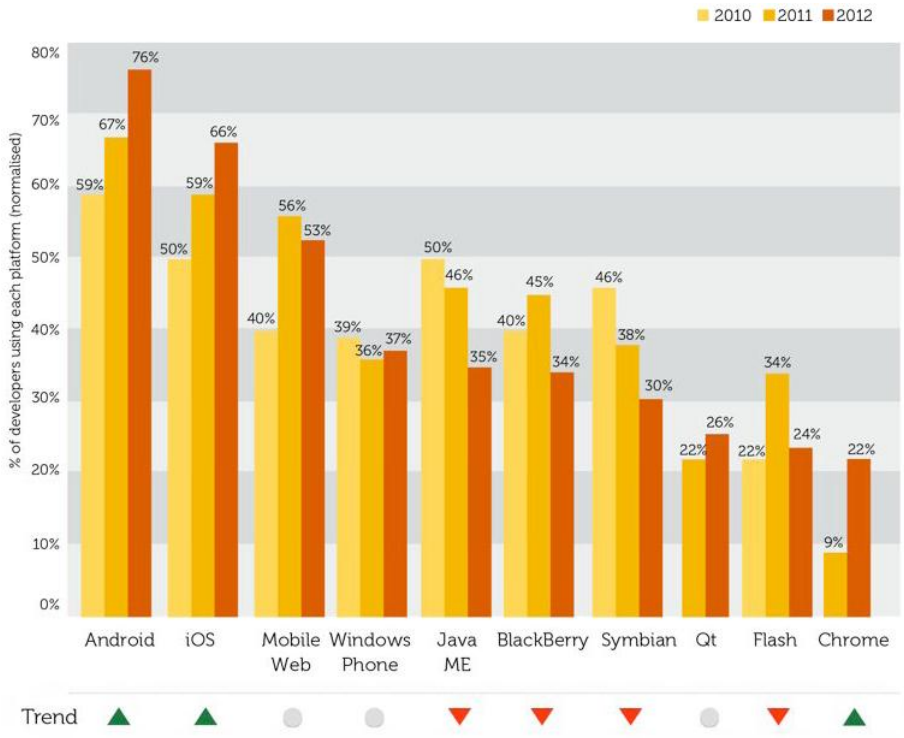

Figura 2.2.: Porcentaje de uso de plataformas entre 2010 y primer semestre de 2012 [25]

2.2.1. Aplicaciones nativas

En esta sección se presentan las distintas alternativas para el desarrollo de aplicaciones nativas de plataformas específicas de los diferentes fabricantes, las cuales analizamos en base a los siguientes conceptos: posibilidades que brinda, facilidad de desarrollo y aprendizaje, usabilidad y diseño, cobertura del mercado y flexibilidad.

Una ventaja que tienen las aplicaciones nativas, es que lucen y se comportan como las aplicaciones que ya trae el sistema operativo, por lo que se integran de forma coherente con el resto del sistema. Las aplicaciones nativas pueden acceder a funcionalidades como ser sistema de archivos, gps, acelerómetro o cámara, mediante

interfaces brindadas por el sistema permitiendo aprovechar todas las prestaciones del dispositivo.

Una desventaja que presenta este enfoque de desarrollo es que el conocimiento adquirido para desarrollar un sistema no puede ser reutilizado para desarrollar otro. Si se desea desarrollar una aplicación para que corra en iOS y Android, hay que programarla dos veces: una en Objective-C y la otra en Java.

Android

Android es un sistema operativo para dispositivos móviles inicialmente desarrollado por Android Inc. una firma adquirida en el 2005 por Google. Google define a Android como un "stack de software" para dispositivos móviles ya que está formado por un sistema operativo, un middleware que facilita el desarrollo de aplicaciones y un conjunto de aplicaciones base [3].

Posibilidades que brinda la plataforma

- Es un sistema de código abierto y corre sobre el kernel de Linux. Al ser código abierto hay versiones de Android desarrolladas por la comunidad.
- El SDK de Android provee las herramientas necesarias para desarrollar aplicaciones para la plataforma utilizando el lenguaje Java, permitiendo la reutilización de librerías y componentes.
- Incorpora de manera nativa herramientas como SQLite para almacenamiento en base de datos, soporta varios formatos de audio y video, utilidades para acceder a la red y acceso al hardware.
- El desarrollador tiene acceso completo a la misma API utilizada por las aplicaciones que vienen con el sistema operativo, lo que hace que el usuario pueda instalar nuevos programas que le permitan modificar por completo el funcionamiento del teléfono de acuerdo a sus necesidades.

Facilidad de desarrollo y aprendizaje

- Fuerte soporte para el desarrollo de módulos y aplicaciones en Java, generó una gran comunidad de desarrolladores.
- Tiene buena documentación disponible en la web, en la que se pueden encontrar varios tutoriales y guías para aprender a programar para Android.
- Permite desarrollar secciones de la aplicación en C/C++ en los casos en que se necesite prestar especial atención al rendimiento.
- Provee un entorno de desarrollo completo que incluye un emulador y un plugin para Eclipse.

Usabilidad y diseño

- Dada la gran comunidad de desarrolladores, existen sitios como androidpatterns.com [4] que ofrecen patrones de diseño para aplicaciones Android.
- La documentación de Android Developers cuenta con varias secciones referentes dando guías de diseño y usabilidad [3].
- Una de las desventajas es que hay muchas versiones de Android ejecutando en los dispositivos y cada versión agrega funcionalidades que no están presentes en las anteriores. Los componentes gráficos van variando y resulta difícil decidir qué versión de la API utilizar. La siguiente figura fue tomada de developer.android.com [3] y muestra el porcentaje de uso de cada versión liberada.

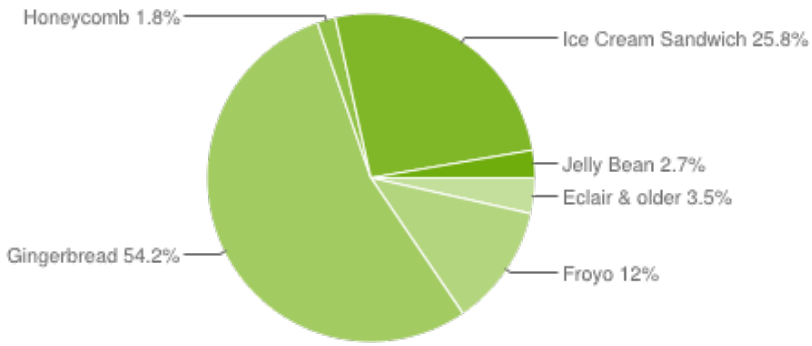

Figura 2.3.: Versiones de Android liberadas y su porcentaje de uso [3]

Cobertura del mercado

- En Android Market existen más de 300.000 aplicaciones para descargar y gran parte de ellas son gratuitas. Existen otros mercados como Amazon Android Appstore, GetJar y AndroidPIT.
- En marzo de 2012, Google unificó el nombre de sus tiendas de aplicaciones, Android Market, Google Music y Google eBookStore, pasando a ser llamado Google Play Store.
- Es la plataforma que más está creciendo en popularidad en el mundo. En la siguiente figura, obtenida de IDC [39], se puede apreciar como Android va liderando el mercado de sistemas operativos para smartphones y continúa ganando mercado.

Flexibilidad

- Por medio de la API se puede hacer uso del hardware y aprovechar todas las prestaciones del dispositivo, permitiendo desarrollar todo tipo de aplicaciones.

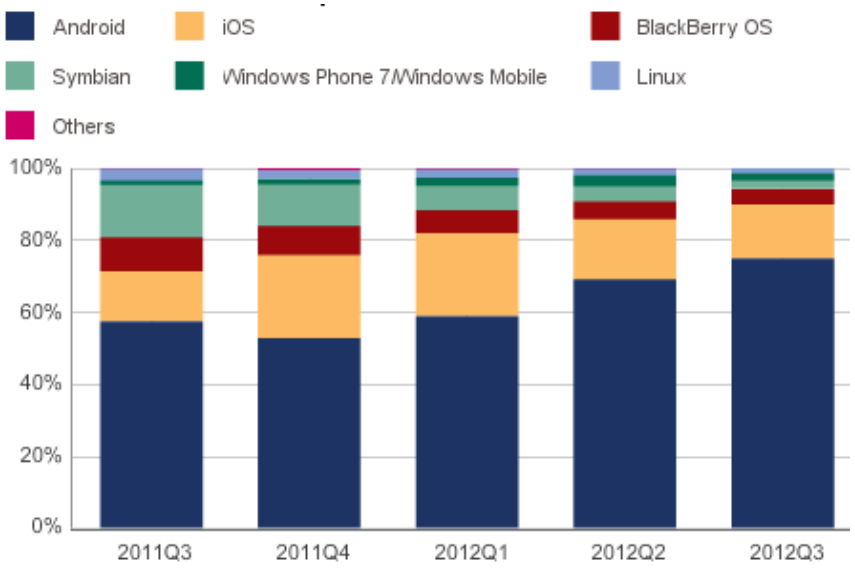

Figura 2.4.: Mercado de sistemas operativos para smartphones [39]

- Es posible reutilizar componentes de aplicaciones de terceros sin necesidad de implementar todas las funcionalidades requeridas por el sistema. Por ejemplo si es necesario tomar una foto dentro una aplicación, es posible solicitarle al sistema que lance la aplicación de tomar fotos y el sistema cargará la aplicación de tomar fotos preferida del usuario.
- Provee soporte para diseñar para múltiples tamaños de pantallas. Es fácil definir diferentes disposiciones (layouts) según tamaño de pantalla, de esta forma adaptamos la aplicación al tipo de dispositivo en el que ejecuta.

Apple iOS

iOS es un sistema operativo para dispositivos móviles desarrollado por Apple. Aunque comparte un origen común con muchas de las tecnologías de Mac OS, iOS fue diseñado para trabajar en dispositivos móviles donde las necesidades del usuario son diferentes [6].

Posibilidades que brinda la plataforma

- El SDK de iOS permite la creación de aplicaciones gráficas (graphically-oriented applications) que corren de forma nativa en iOS.
- Para desarrollar aplicaciones para iOS es necesario contar con una computadora Mac OS X corriendo las herramientas Xcode, el cual es un IDE que contiene las herramientas necesarias para crear proyectos para iOS y su código fuente, permite compilar el código, ejecutar y debuguear la aplicación tanto en un dispositivo como en un simulador de iOS.

- El lenguaje utilizado para programar para iOS es Objective-C, un lenguaje orientado a objetos que extiende el estándar ANSI C, provee sintaxis y características de la orientación a objetos.

Facilidad de desarrollo y aprendizaje

- Para poder publicar las aplicaciones en la tienda de aplicaciones, es necesario pasar por un proceso de aprobación establecido por Apple.
- En la página de desarrollo de Apple hay varios recursos para aprender a desarrollar para esta tecnología.

Usabilidad y diseño

- Apple ofrece el "iPhone developer program" donde los desarrolladores se pueden registrar para acceder a recursos de aprendizaje y desarrollo.
- En el sitio para desarrolladores de Apple se muestran varias guías en lo que respecta en diseño y usabilidad. Definen un conjunto de principios para que las aplicaciones que se desarrollan para iOS se integren de forma consistente con el resto del dispositivo.

Cobertura del mercado

- Según la empresa Quantcast[63], en mayo de 2010 en Estados Unidos, el 58.8 % de los accesos web desde dispositivos móviles eran provenientes de iOS como se muestra en la figura 2.5 en la página siguiente.
- Desde App Store se pueden descargar muchas aplicaciones gratis y con licencia [14]. Apple también ofrece la store iTunes para la descarga de música.

Flexibilidad

- La interfaz es intuitiva, todo lo necesario para comenzar a utilizar el dispositivo lo tenemos a golpe de vista, contactos, mensajes, teléfono, mail, App Store y demás aplicaciones instaladas de fábrica se organizan en una sola pantalla.
- Solo esta disponible en los dispositivos de la marca Apple.

Symbian

Symbian es un sistema operativo con código abierto diseñado para dispositivos móviles mantenido por Nokia [58].

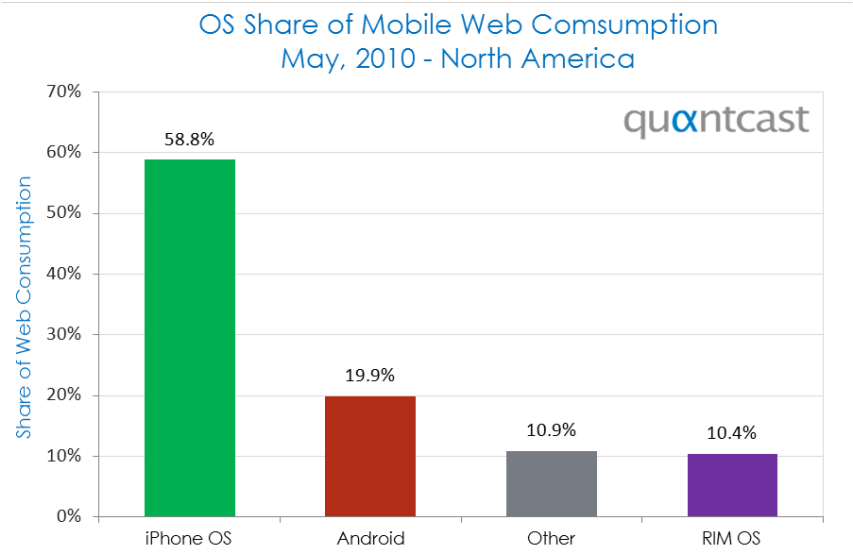

Figura 2.5.: iOS lidera en 2010 los accesos web desde dispositivos móviles en Estados Unidos [63]

Posibilidades que brinda la plataforma

- Dentro de las posibilidades que brinda la plataforma es que soporta varios lenguajes de programación y frameworks para el desarrollo de aplicaciones como Java, QT y Symbian C++, Adobe Flash Lite.
- Las ventajas que ofrece Symbian, es la capacidad que tiene el sistema para adaptar e integrar todo tipo de aplicaciones que pueden ser programadas por diferentes desarrolladores.

Facilidad de desarrollo y aprendizaje

- Permite utilizar lenguajes de alto nivel como Java, C++, Visual Basic o Perl.
- Proporciona herramientas para desarrollar aplicaciones para este tipo de sistema operativo, el inconveniente en este punto es que existen entornos de desarrollo y API's diferentes en función de la interfaz sobre la cual se va a implementar la aplicación a realizar.
- En la página para desarrolladores de Nokia se encontraron guías para traducir aplicaciones Android e iOS para el framework QT.

Usabilidad y diseño

- No se encontraron guías de usabilidad para Symbian C++ y diseño, como en el caso de Android e iOS, solo un listado, si las hay para el desarrollo con QT.

- Como se mencionó anteriormente permite el uso de varios frameworks para la creación de aplicaciones que permiten especificar las interfaces de manera diferentes:
  - QT: Al ser QT portable el mismo código puede ser utilizado para otro tipo de dispositivos que soporten QT o incluso como aplicaciones de escritorio.
  - Symbian C++: Utilizando las API C/C++ nativas de Symbian y S60, permite acceder a las funcionalidades de bajo nivel del dispositivo, puede ser útil para desarrollar juegos.
  - Java – Utilizando la especificación MIDP 2.1 (Mobile Information Device Profile) para Java [42], permite usar los componentes gráficos definidos por Java.
  - Web Runtime – Utilizando tecnología web estándar.
  - Adobe Flash Lite – La tecnología flash es soportada por todas las plataformas de Nokia y está integrada con el navegador basado en WebKit incorporado en celulares Nokia.

Cobertura del mercado

- En el 2007 Symbian tenía el 61.7 % del mercado de smartphones en el mundo, pero en la actualidad ha bajado considerablemente. Aún así continúa teniendo una fuerte presencia en dispositivos que no son considerados smartphones.
- Desde Ovi Store o Symbian App Store se pueden descargar aplicaciones gratis y con licencia [59].

Flexibilidad

- La tecnología del sistema operativo Symbian se ha diseñado teniendo en cuenta puntos clave como el poder proporcionar la energía, memoria y gestión de entrada y salida de recursos requeridos específicamente en los dispositivos móviles.
- Las ventajas de desarrollar aplicaciones para Symbian es la rapidez de ejecución.
- La flexibilidad de Symbian permite que sea corrido tanto en celulares con un teclado estándar con una pantalla de 320 x 240 píxeles como en smartphones con pantallas nHD táctiles, teclados querty y dispositivos con variedad de sensores.

BlackBerry OS

BlackBerry OS es un sistema operativo desarrollado por la compañía canadiense Research in Motion (RIM) [9]. La plataforma BlackBerry soporta diferentes enfoques

para el desarrollo de aplicaciones dentro de los cuales se destacan el desarrollo de aplicaciones web, el desarrollo de aplicaciones Java y los widgets híbridos.

Posibilidades que brinda la plataforma

- Es posible crear aplicaciones nativas en C/C++ y aplicaciones Java.
- Provee un SDK para crear aplicaciones usando Adobe AIR.
- Con BlackBerry WebWorks Application Platform los desarrolladores tienen la posibilidad de crear los programas aprovechando las ventajas de HTML5, CSS, JavaScript, ofrece un simulador y una guía para interfaz de usuario.
- La plataforma WebWorks para BlackBerry funciona utilizando el motor de HTML WebKit. Esta plataforma permite a los desarrolladores crear aplicaciones cliente-servidor utilizando las tecnologías estándar de la web. La plataforma permite:
  - Acceder al almacenamiento del dispositivo como tarjetas microSD o base de datos SQLite.
  - Acceso seguro a los servidores remotos.
  - Instalar la aplicación a través del BlackBerry App World™.
  - Agregar funcionalidades con extensiones JavaScript.
- En la web para desarrolladores brindan herramientas para reempaquetar aplicaciones Android de forma de poder ejecutarlas en un BlackBerry. BlackBerry 10 y BlackBerry Tablet OS poseen un runtime que permite ejecutar aplicaciones Android 2.3.3[9].

Facilidad de desarrollo y aprendizaje

- Como se mostró en el punto anterior permite desarrollar con varias tecnologías lo que hace que programadores con conocimiento en diferentes lenguajes de programación puedan desarrollar aplicaciones.
- Para cada lenguaje o tecnología que la plataforma soporta, el sitio de desarrolladores contiene herramientas, aplicaciones de ejemplo, documentación y referencias de la API.
- Hay emuladores tanto para smartphones como tabletas ya que tienen sistemas operativos diferentes.

Usabilidad y diseño

- Este sistema operativo está más orientado a clientes empresariales, principalmente usado por su plataforma de mensajería instantánea y correo electrónico.

- Permite en el desarrollo C/C++ expresar las interfaces utilizando QT y QML.
- Como el resto de las plataformas, contiene guías de diseño y usabilidad para crear aplicaciones con un diseño consistente con el resto del sistema.

Cobertura del mercado

- Actualmente cuenta con un 16 % en el mercado mundial.
- Desde AppWorld BlackBerry Web Store los usuarios pueden descargar aplicaciones, algunas son gratis y otras con licencia [7].

Flexibilidad

- Para usar los servicios de comunicación de BlackBerry hay que tener un plan de datos especial.
- Las aplicaciones solo corren en celulares con sistema operativo BlackBerry.

Windows Phone

Windows Mobile fue un sistema operativo para móviles creado en el año 2000 por Microsoft y tuvo su último release el 2 de Febrero del 2010. No seguirán saliendo a la venta nuevos equipos con este sistema operativo. Tuvo tres ambientes de trabajo que fueron: PocketPC para dispositivos móviles que no tuviesen funciones de teléfono, Windows Mobile Standard para teléfonos clásicos sin pantallas táctiles y Windows Mobile Professional para smartphones y teléfonos con touchscreen [50, 43].

Microsoft comenzó a trabajar desde 2008 en lo que debía ser un replanteamiento de su sistema operativo, y en 2010 en el Mobile World Congress en Barcelona presentó Windows Phone. En setiembre de ese mismo año liberó el SDK (el kit para que comiencen a trabajar los desarrolladores) [51].

Posibilidades que brinda la plataforma

- Contiene un conjunto de aplicaciones básicas para dispositivos móviles basados en la API Win32 de Microsoft.
- El SDK de Windows Phone incluye Visual Studio 2010 Express para Windows Phone, Windows Phone Emulator que permite hacer deploy, debug y testear la aplicación, XNA Game Studio, Expression Blend para Windows Phone, ejemplos y documentación.

Facilidad de desarrollo y aprendizaje

- Para desarrollar aplicaciones es necesario tener la herramienta profesional de desarrollo (visual studio.net professional) ya que las versiones de desarrollo gratuitas ofrecidas por Microsoft (express edition) no incluyen estas herramientas.
- En Windows Developer Center hay guías paso a paso como desarrollar una aplicación con Windows Phone.

Usabilidad y diseño

- Algo que hace único a Windows Phone es su interfaz de usuario, llamada "Estilo Metro" porque ha sido inspirada en la tipografía e iconografía de los sistemas de transporte globales. En la edición 2011 del Reconocimiento Internacional a la Excelencia en Diseño (International Design Excellence Awards – IDEA), Windows Phone ganó tres premios: Oro en Experiencia de Producto Interactivo, Plata en Investigación y Bronce en Estrategia de Diseño.
- Cuenta con varias guías en la página para desarrolladores las cuales hacen especial hincapié en la simplicidad del diseño, en la diagramación y en los tamaños y tipo de fuente, son más bien guías generales que pueden aplicarse al diseño de cualquier aplicación para smartphones. Tiene también una parte más avanzada donde describe en detalle los componentes de interfaz de usuario específicos de la plataforma.

Cobertura del mercado

- Cuenta con Windows Phone Market Place donde se pueden adquirir aplicaciones hechas por terceros, pero al igual que la tienda de BlackBerry está un poco limitada, no cuenta con tantas aplicaciones como Android y Apple [52].

Flexibilidad

- Hay una sección en el sitio dedicada e usuarios empresariales.
- Si bien no hay una manera directa de portar aplicaciones desarrolladas para otras plataformas, existe una herramienta que muestra la correspondencia con componentes de las API de otros sistemas.

Análisis comparativo de aplicaciones nativas

A continuación mostramos una tabla en la cual se hizo un relevamiento de las distintas características de las aplicaciones nativas.

	Android	iOS	Symbian	BlackBerry	Windows Phone
Fabricante	Google	Apple	Nokia	RIM	Microsoft
Lenguaje de Programación	Java	Objective C	C++	Java	C#
Tipo de S.O.	Abierto	Cerrado	Abierto	Cerrado	Cerrado
Soporte de memoria externa	SI	NO	SI	SI	SI*
Multitarea	SI	SI	SI	SI	NO
Tienda de Aplicaciones	Google Play	AppStore	Ovi Store	AppWorld	Market Place
Soporte Adobe Flash	SI	NO	SI	NO	NO
Interfaz Personalizable	SI	NO	SI	SI	SI

Figura 2.6.: Comparación Aplicaciones Nativas

2.2.2. Generadores de aplicaciones

Un generador de aplicaciones es un software que genera programas a partir de una descripción del problema, en lugar de hacerlo desde un lenguaje de programación convencional. Funcionan a un nivel de abstracción más alto que un lenguaje de programación de alto nivel. Una sentencia o línea descriptiva puede generar una enorme rutina o todo un programa, sin embargo, los generadores de aplicaciones siempre tienen límites en cuanto a su posible uso. También se conocen como generadores de aplicaciones multiplataforma a software que permite programar con un lenguaje de alto nivel y generar aplicaciones nativas para varios sistemas.

La principal ventaja que promueve este tipo de sistemas es la posibilidad de crear aplicaciones nativas, o que luzcan como nativas, para todas las plataformas con una única especificación del sistema. De esta manera es posible cubrir una gran variedad de los dispositivos que se encuentran en el mercado con una versión nativa generada del sistema. No se requieren grandes conocimientos de lenguajes de programación ya que no es necesario codificar en los lenguajes específicos de cada plataforma. Muchos generadores permiten el acceso a las funcionalidades del dispositivo e integración con las aplicaciones ya existentes, cámara, gps, contactos, entre otros.

Si bien la especificación de la aplicación es independiente de la plataforma en la que se va a ejecutar, se crea una gran dependencia con el generador. Si surge un nuevo sistema operativo, o una nueva versión hay que esperar una actualización del generador para poder portar las aplicaciones. Al ser generadas automáticamente las aplicaciones no están optimizadas para la plataforma objetivo y pueden ser lentas. El acceso a las funcionalidades que brinda el dispositivo no suele ser completo y se accede a un subconjunto para cada plataforma. En algunos casos, se requiere licencia para poder usar los generadores de código de aplicaciones multiplataforma y se corre el riesgo de desarrollar una aplicación que puede quedar obsoleta con el

paso del tiempo debido que pueden surgir nuevas versiones de sistemas operativos, dispositivos o por cambios en el generador.

A continuación se muestran los generadores que se evaluaron para este proyecto.

Genexus

Genexus es una herramienta de desarrollo de software ágil, multiplataforma, basada en conocimiento, orientada principalmente a aplicaciones web empresariales, plataformas Windows y smartphones. Es el producto principal de la compañía uruguaya Artech. El desarrollador describe sus aplicaciones en alto nivel, de manera declarativa, a partir de lo que Genexus genera código para múltiples plataformas (Windows, iSeries, Web, dispositivos Móviles) [65].

Dentro de la documentación mostrada en la web se destacan las siguientes características[1]:

- Incrementar la llegada a usuarios finales de las aplicaciones portando proyectos ya existentes a dispositivos móviles y generando la aplicación para varias plataformas. Escalar la aplicación colocando servicios consumidos desde el móvil en la nube.
- Mejora en la experiencia del usuario. Según la documentación, por medio del uso de las interfaces nativas se aumenta la usabilidad, la performance y se aprovechan las capacidades del hardware.
- Posee funcionalidades para publicar servicios al market lo que facilitaría las iteraciones de desarrollo y el prototipado.
- Si bien recomiendan la generación de aplicaciones nativas, también permiten generar aplicaciones en HTML5.

Genexus para smartdevices[2] permite generar aplicaciones para los siguientes sistemas operativos:

- Android, generando código Java.
- BlackBerry, generando código Java.
- iOS, generando código Objective-C.

Para cada sistema operativo la documentación de Genexus presenta ciertas características que se muestran en la tabla 2.4 en la página siguiente.

La principal ventaja que presenta Genexus en el desarrollo de aplicaciones para dispositivos móviles (y para el desarrollo en Genexus en general) es la facilidad y velocidad para crear aplicaciones relativamente sencillas.

[1] Todas las características mencionadas para Genexus fueron extraídas de [65]
[2] Así llaman en la documentación a “smartphones” y “tablets”

La principal desventaja es que la generación de código para smartdevices es un feature relativamente nuevo dentro de Genexus por lo que aún no se cuenta con mucha documentación y sólo permite crear aplicaciones muy simples.

En el siguiente cuadro se muestran las características de Genexus para dispositivos móviles, relevadas de [65].

Android	**iOS**	**BlackBerry**
Experiencia de usuario Android Nativa.	Experiencia de usuario iOS Nativa.	Experiencia de usuario BlackBerry Nativa.
Aplicaciones en el lenguaje nativo de Android.	Aplicaciones en el lenguaje nativo de iOS (Objective-C).	Aplicaciones en el lenguaje nativo de BlackBerry.
Versatilidad: Cree sin esfuerzo aplicaciones para los distintos teléfonos y tabletas.	Versatilidad: Cree sin esfuerzo aplicaciones para iPad, iPhone y iPod Touch.	Versatilidad: Cree sin esfuerzo aplicaciones para PlayBooks y teléfonos BlackBerry.
Fácilmente integre funcionalidades móviles a sistemas existentes.	KB Navigator, para testear la aplicación rápidamente en la nubes.	Fácilmente integre funcionalidades móviles a sistemas existentes.
Acceso a Cámara, GPS y Teléfono.	Acceso a Cámara, GPS y Teléfono.	Acceso a Cámara, GPS y Teléfono.
Click to Cloud.	Click to Cloud.	Integración con servicios de mensajería e internet de BlackBerry.
		Click to Cloud.

Cuadro 2.4.: Características de Genexus para cada plataforma de dispositivos móviles soportadas [65]

Mobincube

Mobincube es un software web desarrollado por la compañía Mobimento que permite a personas sin conocimiento técnico crear aplicaciones para dispositivos móviles [53].

Dentro de las funcionalidades principales con las que cuenta la herramienta se incluyen:

- Una herramienta para editar y definir el diseño de aplicaciones.
- Posibilidad de publicar la aplicación en todos los markets directamente desde el sistema, aunque también es posible descargar el instalador.

- Permite construir aplicaciones para: iPhone, iPad, Android, Windows Phone, BlackBerry.
- Permite crear aplicaciones web basadas en HTML5.
- Provee herramientas para ganar dinero incluyendo publicidad en las aplicaciones generadas.
- Permite postulase como desarrollador "oficial" y la empresa comienza a derivar trabajo al desarrollador.

En el sitio web de Mobincube muestran que se puede desarrollar todo tipo de aplicaciones pero recorriendo las descripciones de las funcionalidades se puede ver que las aplicaciones generadas son más bien orientadas al marketing y publicidad.

PhoneGap

PhoneGap es una herramienta que permite desarrollar aplicaciones utilizando HTML5 y JavaScript. Las aplicaciones pueden ser empaquetadas como aplicaciones nativas para diferentes plataformas. Desarrollado por la compañía Nitobi Software, empresa que fue comprada por Adobe, ahora es una plataforma de código abierto bajo licencia Apache con el nombre Apache Cordova[28]. La idea es poder aplicar los conocimientos de los desarrolladores de aplicaciones web al desarrollo para dispositivos móviles [19].

Permite desarrollar aplicaciones para iOS, Android, BlackBerry, Windows Phone, Symbian, Bada y WebOS. La aplicación se desarrolla enteramente en HTML5 y JavaScript utilizando algunas API JavaScript definidas por la herramienta que le permiten al desarrollador utilizar las funciones del dispositivo, el código escrito para esas plataformas es luego empaquetado dentro de una aplicación nativa. La aplicación nativa lo que hace es cargar el código HTML5 del almacenamiento local e implementar la API JavaScript expuesta a la aplicación de forma tal que invoque a los servicios requeridos en el sistema operativo del móvil.

Se puede aprovechar completamente el código específico de la aplicación para ser usado en todas las plataformas ya que todas tienen navegadores que soportan HTML5 y JavaScript. Si alguna plataforma no provee alguna de las funciones expuestas por la API estas son implementadas por la herramienta lo que garantiza que la API funcione en todos los dispositivos. Una excepción a esto es el acceso a ciertos componentes de hardware que pueden no estar presentes en algunos dispositivos. Sólo es necesario escribir código nativo para inicializar los componentes que cargan la aplicación desarrollada, y en el sitio web de la herramienta [19] hay varios tutoriales que muestran cómo implementar ese código nativo que es prácticamente igual para todas las aplicaciones que se desarrollen.

PhoneGap brinda una herramienta online para hacer el empaquetado nativo de la aplicación llamada Adobe PhoneGap Build[2]. También existe una extensión llamada

Ripple para el navegador GoogleChrome que puede ser utilizado como entorno de desarrollo para programar y debuguear la aplicación.

Como principal ventaja de esta herramienta se destaca el hecho de poder reutilizar los conocimientos de desarrollo web para desarrollar aplicaciones para prácticamente todos los sistemas operativos de dispositivos móviles más populares del mercado. El hecho de ser un proyecto que esté bajo la incubación de la Apache Software Foundation garantiza que el sistema va a continuar siendo de código abierto y por consiguiente puede ser desarrollado por la comunidad. Ya cuenta con varias herramientas y extensiones para facilitar el desarrollo.

Se identifica como una desventaja que usar el motor web/JavaScript de cada dispositivo es más costoso, en lo que refiere a performance, que ejecutar directamente código nativo. A medida que los navegadores van agregando soporte a nuevos features de HTML5, el uso de esta herramienta como intermediario entre la aplicación web y el código nativo va dejando de tener sentido. Al ser aplicaciones web no lucen como nativas.

Appcelerator Titanium

Es una plataforma para desarrollar aplicaciones móviles y de escritorio utilizando tecnologías web. Está desarrollado por Appcelerator Inc. y fue lanzado en diciembre del 2008. En junio de 2009 se añadió soporte para el desarrollo de aplicaciones móviles para iPhone y Android. En abril de 2010 se añadió soporte para el desarrollo de aplicaciones para iPad [5].

Es una herramienta de código abierto que permite crear aplicaciones nativas para móviles y tabletas usando JavaScript, HTML y CSS, también permite crear aplicaciones de escritorio utilizando los lenguajes Python, PHP y Ruby.

El lenguaje principal utilizado en esta plataforma es JavaScript. Las aplicaciones desarrolladas se ejecutan principalmente sobre un motor JavaScript que invoca a la API nativa. De esta manera se pueden crear con esta plataforma aplicaciones cuya interfaz de usuario sea nativa. Es posible también desarrollar interfaces con HTML y CSS [55].

Appcelerator brinda un servicio “en la nube” que permite ejecutar componentes de backend de las aplicaciones móviles. Cuenta con un conjunto de servicios ya implementados como manejo de usuarios, almacenamiento de fotos, almacenamiento clave-valor y chat entre otros. Permite desarrollar estos servicios utilizando el framework node.js [57].

Rhodes

Rhomobile es una solución para desarrollo multiplataforma de aplicaciones móviles. Rhodes es un framework de código abierto para construir rápidamente aplicaciones

para los sistemas operativos para dispositivos móviles como: iPhone, BlackBerry, Windows Phone, Symbian y Android[21].

Para desarrollar aplicaciones se sigue el patrón MVC (Model-View-Controller). Las vistas son definidas utilizando mayormente HTML, JavaScript y CSS. El código HTML puede tener código Ruby embebido para poder incluir la lógica de negocio necesaria de la aplicación dentro de las vistas. Los controllers y la definición del modelo son implementados con código Ruby. Si se requiere sincronización con un servidor remoto se puede hacer utilizando un producto llamado RhoSync, que es un producto pago.

Según Naranjo Javier [55], las interfaces gráficas desarrolladas no tienen un aspecto completamente nativo. En muchos casos la apariencia es similar, pero las diferencias son apreciables. El acceso a las funcionalidades brindadas por el dispositivo es realizado por medio de una API desarrollada en Ruby. De esta manera el código escrito para la aplicación es independiente de la plataforma en que ejecuta.

El desarrollo se puede realizar utilizando un plugin para Eclipse llamado RhoStudio. Se puede probar la aplicación con el emulador o en los dispositivos, usando el emulador de cada plataforma.

Luego de desarrollada la aplicación con la herramienta RhoHub se genera el código nativo para la plataforma destino. No todo el código generado es nativo, parte de la aplicación corre con un intérprete Ruby, lo que puede afectar la performance.

MotherApp

Es una plataforma que permite generar aplicaciones nativas para Android, iPhone y Windows Mobile desde un lenguaje parecido a HTML que llaman MotherApp HTML. Este lenguaje es un subconjunto de HTML con algunas extensiones que permiten acceder a funcionalidades de dispositivos móviles [17]. La aplicación nativa que se instala en el dispositivo, contiene la información necesaria para cargar el HTML del servidor, y el motor que convierte el HTML cargado en una interfaz nativa.

El código de la aplicación es cargado desde un servidor de la misma manera que se carga una página web, este código es interpretado con un motor llamado “MotherApp devEngine” que traduce el XML de la aplicación en componentes nativos del sistema que está ejecutando.

También se puede conseguir que la aplicación resultante tenga acceso al GPS y a la cámara del teléfono. Además la aplicación generada no es estática, ya que cualquier actualización de datos en la página web se reflejará automáticamente en las aplicaciones de los teléfonos. Si bien el modelo que muestran es similar al de la web donde cada página es cargada del servidor, también proveen mecanismos para indicar qué páginas deben ser guardadas en el almacenamiento del dispositivo.

Permite además, acceder al motor de base de datos del dispositivo haciendo uso de URL's especiales para tal fin.

La principal ventaja que tiene esta herramienta es la posibilidad de realizar cambios en la aplicación sin solicitar a los usuarios que la actualicen. Si bien no hay que desarrollar con código nativo de ninguna plataforma en particular y es posible reutilizar los conocimientos de programación web, es necesario aprender un nuevo lenguaje que solo es útil en esta herramienta.

2.2.3. Aplicaciones web - HTML5

Se denomina aplicación web a aquellas herramientas que los usuarios pueden utilizar accediendo a un servidor web a través de internet mediante un navegador. En este informe consideramos a las aplicaciones web a aquellas que se basan en HTML dado que el uso de tecnologías alternativas como pueden ser Flash o Applets de Java no suelen estar presentes en los dispositivos móviles.

La web es la única plataforma que está presente y funciona en todos los dispositivos móviles. Se utilizan los mismos protocolos y estándares tanto en la web para móviles como en PC de escritorio. Las aplicaciones web son populares debido a lo práctico del navegador web como cliente ligero, a la independencia del sistema operativo, así como a la facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software a miles de usuarios. Permite a los desarrolladores tener una comunicación directa entre el móvil y el sistema, facilitando así la puesta en producción de cambios y actualizaciones sin necesidad de involucrar al usuario [31].

Es una plataforma con tecnologías fáciles de aprender, los costos de producción de aplicaciones web son bajos, es la más estandarizada, es la que está más disponible en todos los móviles y es la forma más fácil de distribuir las aplicaciones.

Si bien la web para móviles está disponible en todos los dispositivos sigue existiendo un problema a resolver que es la variedad de dispositivos y además se suman los navegadores, que existen varias alternativas. Hay que dedicar cierto esfuerzo para lograr adaptarse a la mayoría de los dispositivos de forma de lograr la mejor experiencia de usuario posible. De todos modos, el esfuerzo necesario para adaptarse a la variedad de dispositivos es mucho menor que en el desarrollo de aplicaciones nativas.

En el desarrollo de aplicaciones web se pierde la posibilidad de acceder a las funcionalidades brindadas por el dispositivo, como ser cámara, acelerómetro, acceso a contactos del teléfono entre otros.

Las aplicaciones nativas se integran mejor con las aplicaciones incorporadas en el dispositivo al estar construidas utilizando los mismos componentes gráficos. Al diseñar aplicaciones web para móviles se utilizan los controles brindados por HTML o librerías JavaScript y es posible que difieran de los utilizados por las aplicaciones nativas perdiendo la sensación de integración con el resto del sistema.

Para acceder a una aplicación web es necesario contar con una conexión a internet. Los dispositivos móviles al cambiar de entorno constantemente pueden perder la señal o cambiar de red fácilmente. Algunos navegadores que soportan HTML5 incluyen funcionalidades que permiten realizar aplicaciones que puedan trabajar de manera offline. Esta capacidad de los navegadores aún no es muy popular y es necesario definir mecanismos para que el usuario comprenda que la aplicación seguirá funcionando con el dispositivo offline.

HTML5

HTML5 es la nueva versión de HTML, el lenguaje utilizado para la web. Esta nueva versión proporciona mecanismos que simplifican el trabajo y la inclusión de elementos multimedia. El principal criterio de diseño de HTML5 ha sido el de resolver problemas prácticos, y con este objetivo adopta soluciones dirigidas a facilitar el trabajo en situaciones reales [46]. Surge de un trabajo en conjunto entre el World Wide Web Consortium [67] (W3C) y Web Hypertext Application Technology Working Group [71] (WHATWG). WHATWG estuvo trabajando en lo referente a formularios y aplicaciones, el W3C estaba trabajando en XHTML 2.0, en 2006 comenzaron a cooperar y crear una nueva versión de HTML [70]. Se definió un conjunto de reglas para definir a HTML5:

- Las nuevas funcionalidades deberían estar basadas únicamente en HTML, CSS, DOM, y JavaScript
- Reducir la necesidad de utilizar plugins externos al navegador (como Flash y Java).
- Mejorar el manejo de errores.
- Más tags en el lenguaje para reducir la necesidad de usar scripts.
- HTML5 debería ser independiente de la plataforma en la que se ejecute.

Algunos elementos (Nuevas API's y JavaScript) que provee HTML5 que hacen que sea un paso importante en el desarrollo de aplicaciones móviles son:

- API para hacer Drag & Drop mediante eventos.
- API para trabajar offline. Se proveen mecanismos de almacenamiento del lado del cliente que permiten a los desarrolladores manejar las posibles interrupciones de conectividad y mejorar la experiencia de usuario al no depender del estado de la conexión para finalizar una transacción.
- Canvas y Video: Estas funcionalidades están diseñadas para poder embeber gráficos y video sin la necesidad de que el dispositivo cliente tenga instalado algún plugin.
- Formularios Avanzados: Existen más tipos de inputs y validaciones incorporados que pueden ser manejados por el navegador lo que implica descargar menos código JavaScript.

- API de Geoposicionamiento para dispositivos que lo soporten.
- API Storage. Facilidad de almacenamiento persistente en local, con bases de datos (basadas en SQLite) o con almacenamiento de objetos por - aplicación o por dominio web (Local Storage y Global Storage). Se dispone de una Base de Datos con la posibilidad de hacer consultas SQL.
- Soporte para pantallas táctiles.

El sitio html5test.com permite hacer una prueba del soporte que tienen los navegadores web a las características que se van definiendo. La figura 2.7 muestra cómo los distintos navegadores para dispositivos móviles han ido implementando las diferentes funciones definidas para HTML5 basadas en un tope máximo de 500 puntos[3]. Se puede apreciar en la gráfica la tendencia que tienen los desarrolladores de navegadores y sistemas operativos para móviles a implementar la mayor cantidad de características posible. Esto es una ventaja ya que se estaría cada vez más cerca de llegar a tener una plataforma estándar entre los diferentes dispositivos.

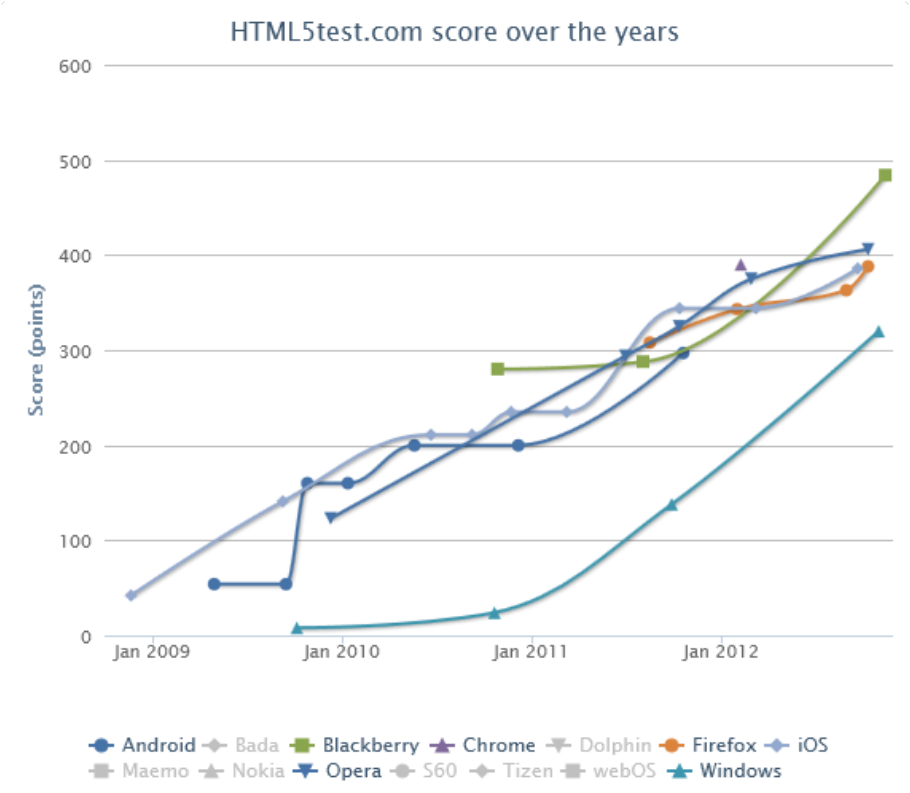

Figura 2.7.: Adopción de las funcionalidades definidas para HTML5 en los navegadores para dispositivos móviles [13]

[3]Cada característica tiene un cierto puntaje, a medida que se van agregando características a la especificación se va aumentando el tope máximo.

Una ventaja que tiene el desarrollo en HTML5 para móviles, que también la tiene para el desarrollo para navegadores de escritorio, es que no es necesario mantener diferentes versiones de la aplicación para cada dispositivo. En el caso de ocurrencia de nuevas versiones no es necesario que el cliente descargue e instale la nueva aplicación, ya que el navegador descarga los contenidos actualizados por los desarrolladores desde el servidor de la aplicación. Como contrapartida se tiene que al momento de iniciar el desarrollo hay que tener mucho cuidado con el framework a utilizar (en caso de utilizar alguno), ya que en este momento la mayoría no está totalmente estable o está en una primera versión. Para el desarrollo de aplicaciones en HTML5 no es necesario disponer de licencias de desarrollo.

A continuación se describen algunos frameworks para desarrollo de aplicaciones HTML5 para dispositivos móviles.

Sencha Touch Es un framework para construir aplicaciones basadas en HTML5, CSS y JavaScript. Está disponible tanto en una versión con licencia comercial como opensource según el uso que se le de [24]. Dentro de las funcionalidades que brinda se encuentran:

- Soporte a la interacción por medio de pantallas táctiles manejando eventos como: como toque, doble toque, arrastrar, tocar y mantener, pellizcar y girar.
- Permite interactuar con diferentes tipos de fuentes de datos como AJAX, JSONP, YQL, con la capacidad de mapear los datos a componentes del framework.
- Ofrece funcionalidades para incorporar audio y video utilizando los componentes especificados por HTML5 para tal fin.
- Una API para hacer uso del almacenamiento local (en el dispositivo) especificado por HTML5 [49].
- Incluye un estilo por defecto, pero además viene con dos temas más: uno que se ajusta mejor al look and feel de Android y otro para iOS, cambiar entre uno y otro es sólo cuestión de reemplazar el CSS.

Actualmente es compatible con Apple iOS, Android y BlackBerry.

Tiene muchas ventajas, es fácil de usar, flexible y cuenta con muchas características y funciones interesantes, sólo funciona con los navegadores basados en WebKit como lo son la mayoría de los navegadores que vienen incluidos con los dispositivos móviles.

JQuery Mobile Es un framework JavaScript para creación de sitios webs optimizados, para los dispositivos móviles más populares. Se basa principalmente en el framework para JavaScript JQuery y JQuery UI [47].

Este framework está orientado aplicaciones para pantallas táctiles, intenta simplificar el código fuente e incluye herramientas y controles para la construcción de interfaces para móviles [16].

Algunas características son:

- Está desarrollado para funcionar con HTML5.
- Implementa un sistema para realizar transiciones de pantallas utilizando AJAX.
- Soporte para eventos táctiles.
- Componentes de interfaz gráfica que extienden los nativos de HTML que son independientes de la plataforma.
- Personalización de temas: igual que ocurre con jQueryUI, hay varios temas para JQuery Mobile, también se pueden crear temas personalizados.
- Permite que el mismo código fuente funcione tanto para dispositivos móviles como pantallas de escritorio.
- Funciones de accesibilidad como ser soporte para lectores de pantalla.

Existen diversos grados de compatibilidad para cada navegador dentro de cada familia de dispositivos móviles. En la siguiente imagen podemos ver los logos de algunos de los sistemas operativos que jQuery Mobile soporta.

Figura 2.8.: Sistemas Operativos que soportan jQuery Mobile

JQTouch Es un plugin para jQuery específicamente diseñado para navegadores móviles basados en WebKit (iPhone, Android y Palm Pre). Es fácil de usar y está bien documentado [15].

Adopta un enfoque progresivo de mejora, busca llegar a que las aplicaciones tengan una experiencia de usuario similar a las nativas de iPhone en base a su estructuración del HTML. Efectos de animación, diseño, UI, formularios entre otras cosas que emulan la interfaz nativa de aplicaciones móviles. Depende mucho de capacidades "únicas" del WebKit de iPhone, por lo que no funciona muy bien en Android. Es simple, y proporciona un conjunto básico de widgets y animaciones y provee el suficiente control programático para dar la posibilidad a los desarrolladores de generar un comportamiento dinámico.

Jo Framework Jo es un framework para desarrollar aplicaciones web en HTML5 para dispositivos móviles. Fue desarrollado específicamente para hacer aplicaciones y no sitios web. Soporta los navegadores de iOS, Android, webOS, BlackBerry, Chrome OS y otros navegadores que soporten HTML5 y CSS3. También funciona bien con PhoneGap si se lo quiere integrar en una aplicación nativa [45].

Algunas de las funcionalidades que brinda son:

- Componentes gráficos independientes de la plataforma para crear aplicaciones web.
- Animaciones en las transiciones entre pantallas.
- No requiere grandes conocimientos del DOM.
- Implementa su propia jerarquía de objetos para representar la interfaz por lo que no depende de otras bibliotecas.
- Soporte para almacenamiento local en el browser y acceso a base de datos SQLite del navegador WebKit.
- Estilo personalizable.

2.3. Análisis del mercado de S.O. para móviles

Una de las principales necesidades que surgen a la hora de desarrollar para dispositivos móviles es conocer a cuantos usuarios se va a alcanzar con el producto desarrollado. Dada la gran cantidad de dispositivos existentes y la variedad de plataformas y enfoques de desarrollo posible, es necesario conocer cómo está distribuido el mercado para tener una idea de qué oportunidades se ganan (o se pierden) al elegir cualquiera de las alternativas.

En esta sección se presentan las porciones del mercado de los distintos sistemas operativos para dispositivos móviles, tanto a nivel mundial como en el mercado local.

2.3.1. Mercado mundial

Según el informe IDC's Worldwide Quarterly Mobile Phone Tracker [39] se obtuvieron los datos que se despliegan a continuación.

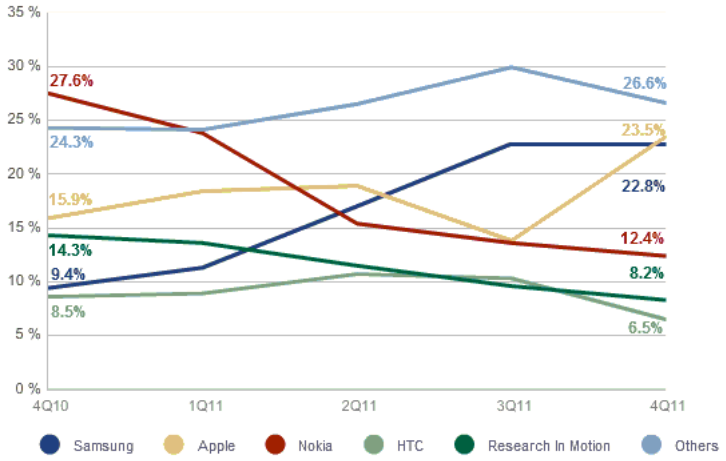

Figura 2.9.: Cinco principales fabricantes de smartphones [39]

Vendor	Q4 2011 Shipment Volumes	Q4 2011 Market Share	Q4 2010 Shipment Volumes	Q4 2010 Market Share	Year Over Year Change
Apple	37.0	23.5%	16.2	15.9%	128.4%
Samsung	36.0	22.8%	9.6	9.4%	275.0%
Nokia	19.6	12.4%	28.1	27.6%	-30.6%
Research In Motion	13.0	8.2%	14.6	14.3%	-11.0%
HTC	10.2	6.5%	8.7	8.5%	17.2%
Others	42.0	26.6%	24.8	24.3%	69.4%
Total	**157.8**	**100.0%**	**102.0**	**100.0%**	**54.7%**

Figura 2.10.: Cinco principales proveedores a nivel mundial de smartphones, envíos y cuota de mercado, en 2011 y 2010 (unidades en millones)[39]

Según Infographic en su informe "Smartphone Users around the world - Statistics and facts" [33] el 80 % de las personas a nivel mundial tiene un dispositivo móvil. Existen 5 billones de dispositivos móviles en el mundo, donde 1.08 billones son smartphones.

A nivel de plataformas en el año 2011 1.6 % Symbian, 5.2 % Microsoft, 16.6 % RIM, 28.7 % iPhone y 46.9 % Android. El 89 % de las personas usa su smartphone durante todo el día. Según sexo: 47 % son mujeres y 53 % son hombres.

	iPhone	Android	RIM	Microsoft
18-34	43 %	50 %	39 %	38 %
35-44	24 %	21 %	20 %	25 %
45-54	16 %	16 %	21 %	20 %
55+	17 %	13 %	20 %	18 %

Cuadro 2.5.: Uso de sistemas operativos según edad

2.3.2. Mercado América Latina

Los smartphones se han impuesto fuertemente en América Latina. Según una investigación hecha en el primer trimestre del 2012 por Our Mobile Planet [62] desarrollado por Ipsos MediaCT, la cantidad de usuarios con smartphones ha crecido en América Latina. Siendo Brasil y México los países que cuentan con la mayor cantidad de smartphones, alrededor de 27 millones y 23 millones respectivamente. Dicho informe contó con el apoyo de la MMA (Mobile Marketing Association) y el IAB (Interactive Advertising Bureau).Según el informe varios países en América Latina cuentan con mayor cantidad de usuarios con smartphones que otros países del mundo.

	%
Android	46.0
iOS	18.9
Symbian	18.7
RIM	10.9
Windows Mobile	2.1
Otros	3.4

Cuadro 2.6.: Market Share en plataformas en América Latina

2.4. Análisis de la competencia a Feng Mobile

En esta sección se presentan los sistemas que se asemejan a Feng Office, tienen una interfaz para dispositivos móviles y se consideran como los más relevantes, ya sea por sus funcionalidades o por popularidad. Los sistemas analizados son: BaseCamp, Central Desktop, Zoho y Google Apps.

2.4.1. 37signals - Basecamp

Basecamp es una herramienta web para la gestión de proyectos desarrollada por la empresa 37signals [1]. Dentro de las funcionalidades que ofrece se encuentran lista de pendientes, edición de documentos web similar a wikipedia, gestión de milestones, permite compartir archivos, gestión de tiempo y un sistema de mensajería [64].

Para realizar esta evaluación se creó una cuenta gratis que está limitada a un único proyecto y 10 mb de almacenamiento. Utilizando esta cuenta es posible evaluar las versiones para móvil y PC del sistema. Para evaluar la versión móvil se accedió utilizando el navegador de un celular con Android 2.1.

Inicialmente la aplicación permite seleccionar el proyecto en el que se va a trabajar, además de mostrar la lista de pendientes, los milestones y configuraciones de la cuenta.

La interfaz de la aplicación cuenta con un conjunto de pestañas que permite acceder a las funcionalidades mencionadas en el párrafo anterior. Las pestañas son:

- Overview – Muestra un resumen de lo último ocurrido en el proyecto.
- Messages – Despliega los mensajes creados por los integrantes del proyecto.
- To-Dos – Muestra la lista de tareas pendientes a hacer en el proyecto.
- Milestones – Permite gestionar los milestones.
- Writeboards – Permite crear espacios similares a una wiki para escribir documentos.
- Files – Muestra los archivos subidos al sistema.
- People & Permissions - Permite administrar las personas relacionadas con el proyecto y asignar los permisos a los usuarios.
- Search – Buscador dentro del proyecto.

La aplicación oficial para dispositivos móviles es una aplicación web en HTML5 especialmente diseñada para estos dispositivos. La figura 2.11 muestra algunas capturas de pantalla de Basecamp.

Cobertura de funcionalidades

A continuación se muestran las funcionalidades de la aplicación que están presentes en la versión para móviles.

Seguridad Se identifica al usuario por medio de nombre de usuario y contraseña. El acceso es mediante HTTPS.

Dashboard En la pantalla inicial, luego del login, se permite al usuario seleccionar el proyecto en el que se va a trabajar. Por defecto se muestran los proyectos recientes pero es posible verlos todos. Además de estas opciones solo se encuentra la de Sign Out para salir del sistema.

Una vez que se selecciona a un proyecto no se muestra un overview y se muestra el listado con las funcionalidades disponibles y una pestaña para ver las personas involucradas en el proyecto.

Messages Es posible crear mensajes y comentarlos, al igual que en la web para PC. No está disponible la opción de editar, ni borrar los mensajes. Tampoco es posible adjuntar archivos, aunque sí es posible ver los archivos que fueron adjuntados a mensajes escritos desde la web estándar.

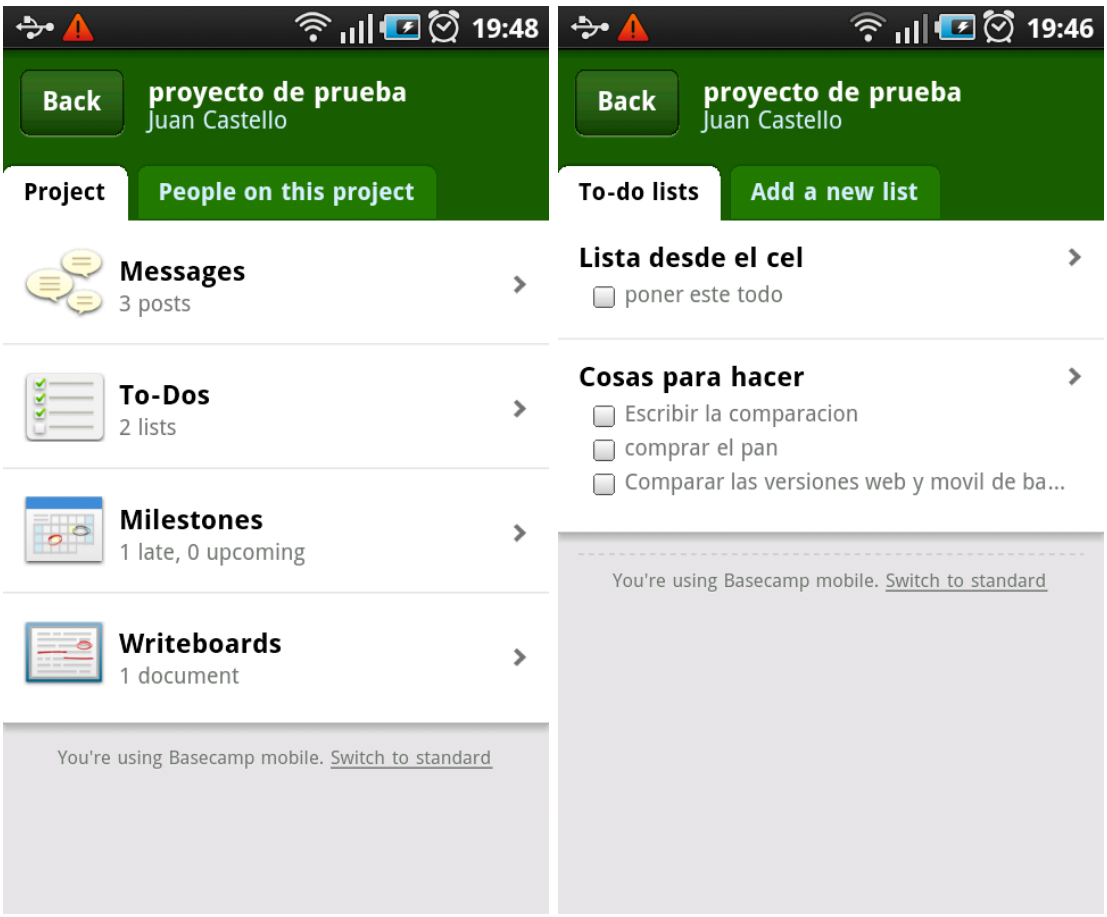

Figura 2.11.: Pantallas de acceso, principal y listado de tareas pendientes en la versión móvil de Basecamp

To-Dos Para la lista de tareas pendientes es posible crear, editar y marcar como completadas las tareas. Es posible hacer comentarios. No es posible eliminar listas ni tareas.

Milestones Para los milestones es posible filtrar por completados, atrasados o a alcanzar. La funcionalidad de milestones es de consulta, pero es posible hacer comentarios sobre los milestones. Es posible exportar los milestones utilizando iCal [64].

Writeboards Los writeboards permiten editar un documento de forma similar a wikipedia. En la versión para móviles del sistema son sólo de lectura. No es posible hacer comentarios sobre los mismos.

People on this project A esta funcionalidad se accede desde una pestaña en la pantalla principal del proyecto. Despliega un listado con las personas involucradas. Al tocar sobre el correo de una persona abre la aplicación de enviar correo del móvil.

Diseño de la interfaz

Dentro de una pantalla, las distintas funcionalidades se muestran como pestañas. Los módulos como ser mensajes, To-Dos, milestones se muestran como un listado al contrario que en la web que se utilizan pestañas.

Para mostrar todos los listados se utiliza el patrón "Thumbnail-and-Text List" que se puede ver en [44]. Un ejemplo de esto es el listado de módulos donde se muestra un icono que los identifica y un texto con el nombre del módulo, debajo de este texto se muestra un resumen del módulo, por ejemplo: en el caso de los mensajes se muestra la cantidad de mensajes.

Al acceder a una página se muestra una animación en la cual se desliza la pantalla de derecha a izquierda. En todas las pantallas se muestra en la parte superior izquierda un botón "Back" que permite volver a la pantalla anterior. Cuando se toca sobre el botón back se muestra una transición de izquierda a derecha.

Al utilizar la aplicación no es necesario esperar que cargue la página. Una vez que carga el marco principal si hay contenido que debe actualizarse se muestra lo que se puede mostrar. Cuando se está cargando información desde el server se muestra un icono en la parte superior derecha que muestra que se están actualizando los datos.

En la parte inferior de la pantalla se muestra la opción de ver el sistema en la versión HTML estándar.

Otro aspecto a destacar es que la aplicación se ve bien sin importar la orientación de la pantalla.

2.4.2. Central Desktop

Central Desktop permite utilizar workspaces para representar equipos de trabajo o áreas dentro de la compañía [11]. Dentro de cada workspace se pueden utilizar las siguientes funcionalidades:

- Files & Discussions – Repositorio donde es posible almacenar archivos y conversaciones. Dentro de esta opción es posible crear documentos.
- Tasks – Permite crear tareas y listas de tareas.
- Milestones – Administración de milestones.
- Calendar – Permite ver en el calendario, eventos milestones y tareas.
- People – Permite ver las personas de las empresas asociadas al workspace.

La versión móvil de Central Desktop es una aplicación web para celulares que se muestra en la figura 2.12.

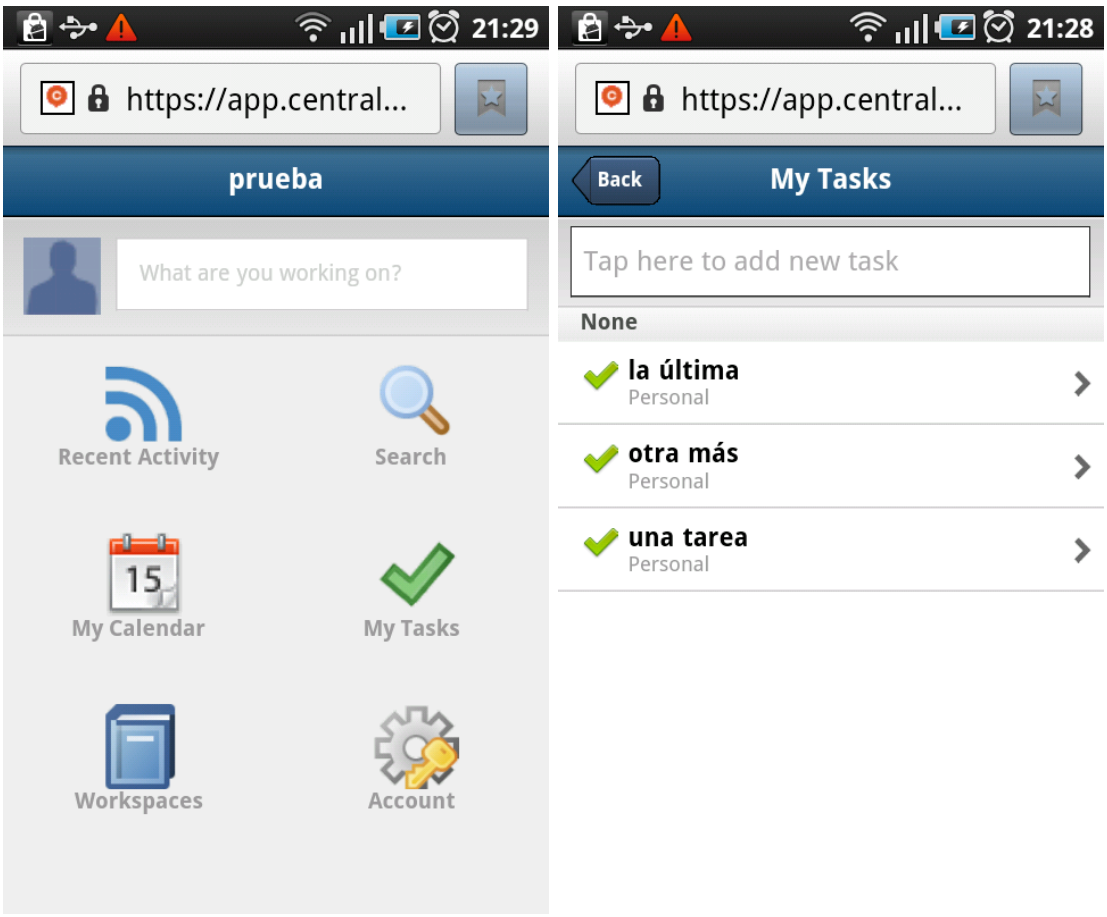

Figura 2.12.: Capturas de pantalla de Central Desktop

Cobertura de funcionalidades

A continuación se muestran las funcionalidades de Central Desktop que se incluyen en su interfaz para móviles.

Seguridad Como se puede ver en la figura 2.12 se accede por HTTPS.

Workspaces Es posible seleccionar los workspaces existentes dentro de la cuenta, y para cada uno de ellos acceder a los eventos, calendarios y tareas. No es posible crear ni modificar workspaces.

Tasks Para acceder a la vista de tareas es posible seleccionar las tareas asociadas a un workspace o seleccionar las tareas que tiene asignadas el usuario. El sistema permite crear y modificar tareas. No es posible eliminarlas.

Events Para acceder a la vista de eventos es posible seleccionar los eventos dentro de un workspace o ver los eventos del usuario. Es posible ver y modificar los eventos, pero no eliminarlos.

Calendar Muestra un calendario que permite, al hacer clic sobre un día del mes, acceder a los eventos agendados. Una vez seleccionado el día es posible también crear eventos. No es posible eliminar eventos del calendario.

Files Con la opción de files se permite una navegación por los directorios donde se almacenan los archivos. Es posible descargar los archivos (esta opción no funciona con el navegador por defecto de Android). No es posible agregar archivos ni modificar.

Account Permite ver datos de la cuenta del usuario.

Diseño de la interfaz

La interfaz está diseñada para dispositivos móviles Apple (iPhone o iPod touch), se utiliza un botón de back en la parte superior izquierda con el título de la ventana anterior. En la parte superior derecha se muestra una opción sobre el contenido actual (como puede ser el comando Guardar) sino se muestra la opción de logout.

Para mostrar todos los listados se utiliza el patrón "Thumbnail-and-Text List". Al editar los eventos hay opciones como la fecha de inicio y fin que quedan fuera de la pantalla, esto posiblemente se deba a que se encuentra optimizado para iPhone y las pruebas se realizaron utilizando el browser de Android.

2.4.3. Zoho

Zoho es una plataforma web que integra herramientas colaborativas para la administración de empresas. Incluye gestión de proyectos, documentos, chat, email, calendario, entre otras [26].

Cobertura de funcionalidades

Mail Posee un cliente de correo electrónico muy completo, con toda la funcionalidad básica para la comunicación vía email. Permite escribir los correos en texto plano, permite hacer las operaciones comunes sobre un correo como reenviar, responder, mover de carpeta, marcar como leído etc.

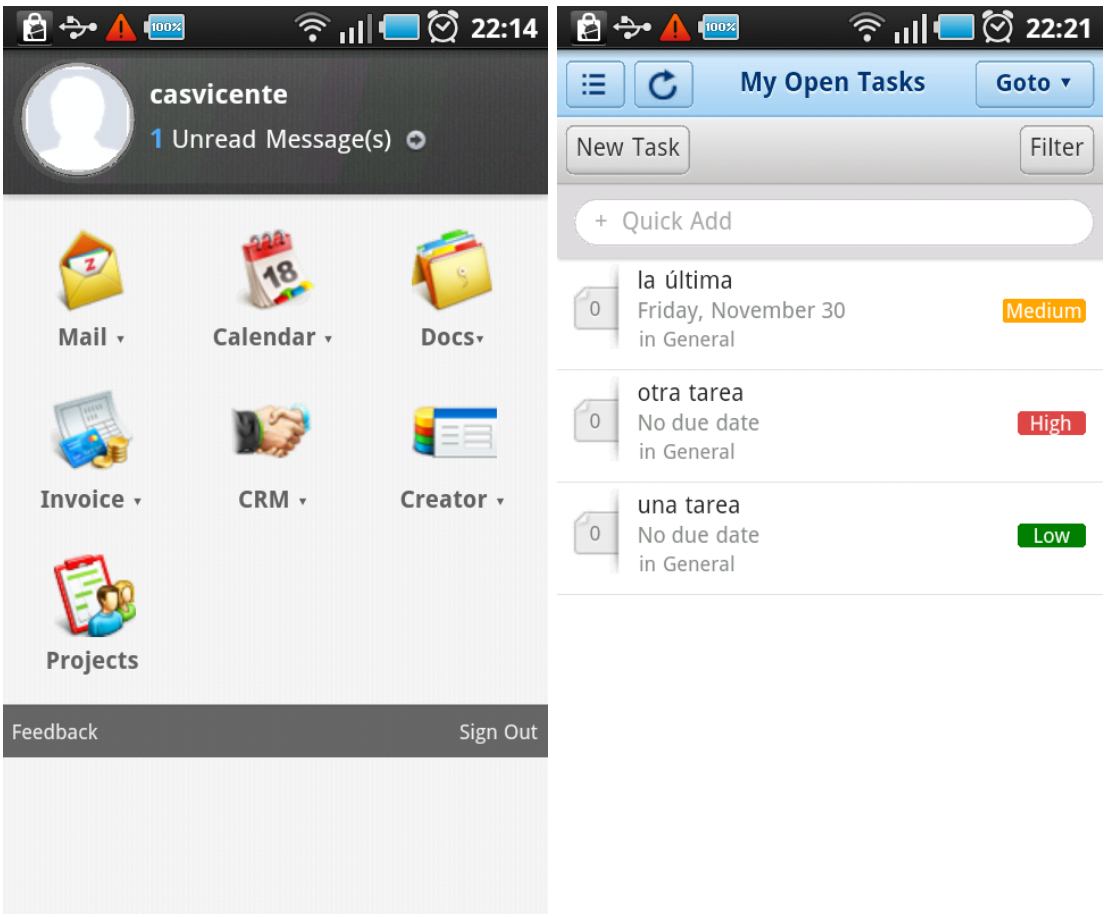

Figura 2.13.: Capturas de pantalla de la versión para móviles de Zoho

Calendar Un calendario en el que se tiene una vista para ver los eventos del día y otra para ver el mes completo. Permite agregar y editar eventos. Permite crear varios calendarios.

Documents Permite ver los documentos agrupados por diferentes directorios, también es posible crear y editar documentos de texto.

En cuanto al conjunto de funcionalidades provista en la versión móvil, en algunos módulos es mucho más reducida que la versión estándar para PC.

La versión móvil provee el conjunto de funcionalidades básicas para que el usuario pueda interactuar con el sistema, sin ofrecer demasiada complejidad en las mismas pero brindando la posibilidad de consultar la información en todo momento. Sobre el diseño de la interfaz, está hecho específicamente para dispositivos móviles, no hereda los patrones utilizados en el cliente para PC. Es intuitivo y claro para el usuario.

Diseño de la interfaz

La interfaz móvil se desarrolló utilizando HTML5. Aplica un diseño específico para los clientes móviles. Las interfaces se basan en listas de items de una columna con menús desplegables para realizar acciones sobre los elementos de la lista. Cada menú lleva a una pantalla nueva con las acciones propias del caso de uso. Utiliza un menú en la parte superior para cambiar de módulo.

2.4.4. Google Apps

Es un servicio que ofrece Google que incluye las siguientes herramientas: correo con GMail, calendario, almacenamiento en la nube con Google Drive, edición de documentos y creación de sitios entre otros [38]. Cuenta con aplicaciones nativas para todas las plataformas, muchas de ellas vienen instaladas con los dispositivos. También existe una versión web para móviles que es lo que se muestra en la figura 2.14.

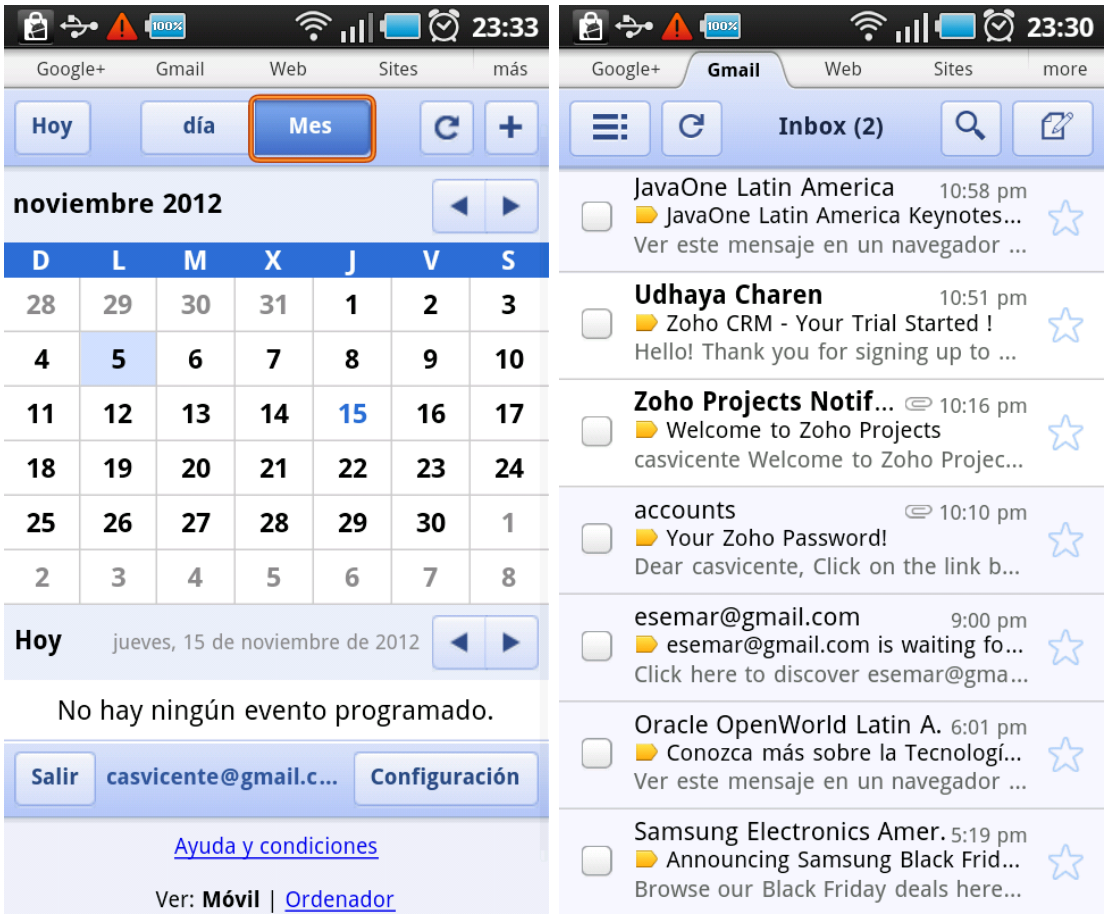

Figura 2.14.: Captura para Google Apps

Cobertura de funcionalidades

Para las distintas plataformas en algunos casos contamos con aplicaciones nativas y en otros hay que utilizar la web. A continuación se muestra para cada plataforma el tipo de acceso que se tiene.

Android Algunos dispositivos que funcionan con Android incluyen de fábrica las aplicaciones de Google pero la mayoría se pueden descargar del market.

BlackBerry GMail se puede descargar la versión gratuita desde el navegador web. Se puede sincronizar información en Google Calendar con el calendario del BlackBerry, para esto se debe descargar e instalar Google Sync. También es posible descarar Google Talk. Para acceder a Google Docs se hace directamente desde el navegador del BlackBerry.

iPhone e iPod Touch Están disponibles con una interfaz web simplificada. Google Sync ofrece una sincronización bidireccional entre calendario, contactos y correo electrónico del iPhone o del iPod Touch y Google Apps.

Windows Phone GMail, Google Calendar y Contactos ya están preinstalados en dichos dispositivos. Al igual que iPhone e iPod con Google Sync se pueden sincronizar bidireccionalmente el correo, calendario y contactos entre Windows Phone y Google Apps.

Diseño de la interfaz

Las versiones nativas se adaptan al estilo de cada plataforma. Para las versiones web se puede ver que son una adaptación de la web. El diseño de la interfaz no se identifica con ninguna plataforma sino que es más bien independiente.

2.4.5. Comparativo

Estudiadas las características de las aplicaciones de la competencia se creó la siguiente tabla que muestra las plataformas para las que están desarrolladas.

Aplicación	Plataforma
37signals - Basecamp	HTML5
Central Desktop	HTML5 y iOS
Zoho	HTML5
Google Apps	BlackBerry, Android, iOS, Windows Phone y HTML5

Cuadro 2.7.: Aplicaciones y plataforma en la que corren

2.5. Diseño de interacción

El diseño de las interfaces de usuario es una parte muy importante de toda aplicación, define la manera en la cual los usuarios interactúan con el sistema para ejecutar las acciones o procesos sobre el mismo. El usuario debe poder realizar las acciones de la manera más sencilla y eficiente posible, esto es lo que motiva a tener un buen diseño de interfaces.

Para poder llegar a un buen diseño primero se deben conocer bien los requerimientos de la aplicación y cual será la plataforma o el tipo de aplicación (aplicación de escritorio, web o móvil).

A partir de estos puntos se deben pensar cuales son las formas más útiles para el usuario para presentarle la aplicación. Algo muy importante a tener en cuenta son las maneras más usuales en que los usuarios intentan interactuar con los sistemas, cuales son los objetivos de ellos al utilizarlos y cual es el lenguaje utilizado según el tipo de sistema (con lenguaje nos referimos al conjunto términos que aplican al negocio). Estos parámetros dependen del tipo de sistema y de las funcionalidades del mismo.

Para obtener este tipo de información se utilizan diversos métodos como realizar encuestas, realizar casos de estudio con grupos de personas utilizando un sistema, métodos de observación directa de los procesos que ejecutan las personas para realizar sus tareas, y otros. Estas formas de relevar las formas de uso son muy costosas.

Con el tiempo ya están bastante universalizados los modos en los cuales los usuarios utilizan los sistemas, a partir de esta información se han formalizado muchos patrones de diseño contemplando las diversas formas de presentar las aplicaciones según su uso [44]. A continuación se presentarán una serie de patrones de diseño de interfaces para aplicaciones móviles.

2.5.1. Patrones de diseño

Los patrones son soluciones óptimas a problemas comunes. Cuando surge un problema común junto con él, espontáneamente, surgen de la comunidad un conjunto

de soluciones comunes. De las soluciones identificadas posiblemente haya un subconjunto que resalta como las mejores, y luego de un proceso de refinación llegan a convertirse en patrones [56]. En el diseño de interfaces, los patrones intentan hacer que lo que se está diseñando sea más fácil de entender o más bello, hacer que sea más útil y usable. Los patrones pueden ser vistos como un conjunto de mejores prácticas en un dominio dado. Al ser soluciones comunes a problemas comunes no son novedosas ni originales. No pretenden ser soluciones listas para utilizar en un contexto. Cada implementación de un patrón difiere de las demás. Los patrones no son reglas a seguir ni una indicación paso a paso de como diseñar una interfaz [44]. En el anexo B se muestran algunos patrones aplicables al sistema a desarrollar.

Fuentes de patrones

Existen abundantes libros y sitios web de donde se pueden obtener patrones, ejemplos y recomendaciones acerca del diseño de interfaces, en la siguiente lista se muestran los que fueron consultados para este proyecto.

- Designing Interfaces (2011) de Jennifer Tidwell [44] es un libro que incluye una recopilación de patrones de diseño de interfaz. Contiene un capítulo dedicado a patrones de diseño para dispositivos móviles.
- Mobile Design Pattern Gallery (2012) de Theresa Neil es una recopilación de patrones específicos para dispositivos móviles. Contiene más de 400 ejemplos de patrones para las diferentes plataformas categorizados y ordenados. [68]
- Sitios propios de las plataformas, contienen guías y patrones característicos de las aplicaciones nativas, algunos de ellos pueden verse en las referencias [3, 58, 6].
- Yahoo! Design Pattern Library - Es una recopilación de patrones de diseño de yahoo. Si bien no son específicos para móviles muchos de ellos pueden ser aplicables[56].
- androidpatterns.com - Es un sitio con un conjunto de patrones aplicables a la plataforma Android creado por la empresa Unitid[4].

2.5.2. Diseño de interfaces para móviles

El diseño de interfaces para dispositivos móviles presenta ciertas dificultades que no se dan a la hora de diseñar aplicaciones para computadoras convencionales. En [44] se presenta el siguiente listado con los desafíos que plantea la plataforma:

Pantallas pequeñas: Los dispositivos móviles no ofrecen demasiado lugar donde poner toda la información y opciones que se desearían. Hay que dejar las funciones más importantes en la página principal y descartar o mover el resto a algún lugar mas escondido en el sistema.

Ancho de pantalla variable: Es difícil crear un diseño que funcione en pantallas de diferentes anchos. Hacer scroll vertical de la pantalla no es costoso para el usuario, pero el ancho de la pantalla debe ser utilizado inteligentemente a la hora de diseñar. Algunos sitios terminan diseñando versiones diferentes para cada tipo de dispositivo. En [32] se presenta una estrategia de diseño para múltiples tamaños de pantalla.

Pantallas táctiles: Es posible navegar por los links de un sitio web utilizando un móvil con teclado. En los dispositivos con pantalla táctil tocar áreas pequeñas resulta dificultoso. Hay que hacer los botones y los links lo suficientemente grandes para que sean fáciles de usar. Esto además reduce el espacio disponible para otras cosas.

Dificultad a la hora de ingresar texto: A los usuarios no les gusta escribir texto en pantallas táctiles ni teclados de celular. Es necesario diseñar la interacción de forma de reducir el ingreso de texto. Usar campos que se auto-completen es una recomendación.

Condiciones del entorno: Las personas usan los dispositivos móviles en diferentes lugares. Las diferentes condiciones de luz pueden variar la visibilidad de los colores en la pantalla. El ruido del ambiente puede generar que los usuarios no oigan señales sonoras en el dispositivo. El movimiento del aparato hace que el texto con tamaño de fuente pequeño sea difícil de leer.

Influencia social y atención limitada: Hay que diseñar considerando que los usuarios no van a pasar mucho tiempo usando la aplicación. Los usuarios usan las aplicaciones móviles mientras realizan otras tareas o están con otras personas. Hay que diseñar asumiendo que los usuarios se distraen, hay que hacer que las tareas dentro del sistema sean fáciles y rápidas.

Para encarar el diseño de aplicaciones para dispositivos móviles se sugiere:

- Diseñar para lo que el usuario realmente necesita en su contexto.
- Reducir la aplicación hasta tener solo lo relevante para el usuario.
- Poner lo más relevante arriba en la pantalla principal.
- En lo posible usar las prestaciones del hardware.
- Linearizar el contenido. El ancho de pantalla de muchos dispositivos no va a permitir diseños horizontales flexibles. Pensar en un diseño vertical.
- Optimizar las secuencias de interacción lo más posible. Evitar la escritura y tratar de cargar pocas páginas.

3. Requerimientos del sistema y alcance del proyecto

Teniendo en cuenta los objetivos planteados para el proyecto y el relevamiento del estado del arte realizado, se definieron los requerimientos del sistema y el alcance del proyecto.

3.1. Requerimientos del sistema

En esta sección se describen los requerimientos que fueron detectados en las reuniones con el cliente.

3.1.1. Requerimientos funcionales

Tareas

Ingresar una tarea Se deberá poder ingresar una tarea especificando los siguientes datos:

- Nombre
- Fecha de comienzo
- Fecha límite, prioridad
- Descripción
- Tiempo estimado
- Recordatorios

Además de los datos característicos de la tarea deberá ser posible:

- Especificar si la tarea se repite y con que período de tiempo (diariamente, semanalmente, mensual, etc.).
- Indicar si es una tarea asociada a una cuenta de Feng Office o es una tarea almacenada personal (almacenada únicamente en el móvil).
- Asignar una persona a la tarea.
- Asociar los suscriptores de la tarea.

Buscar una tarea El sistema deberá poder desplegar un listado de tareas dentro del cual se podrá:

- Buscar una tarea recorriendo la lista manualmente.
- Filtrar las tareas. Ver las asignadas al usuario, o al equipo del usuario.
- Buscar una tarea ingresando un texto.

Ver detalles de una tarea El sistema deberá permitir ver un resumen con los datos de una tarea. En dicho resumen se podrán ver los datos característicos de la tarea y acceder a los comentarios asociados.

Registro de tiempo de la tarea Se deberá brindar una funcionalidad que permita registrar el tiempo de trabajo sobre la tarea esto es:

- Poder registrar el momento en que se comenzó a trabajar en la tarea.
- Poder pausar el registro de tiempo.
- Poder reanudar el trabajo.
- Poder marcar el trabajo como finalizado ingresando un comentario del trabajo realizado.

Completar una tarea El usuario deberá poder marcar una tarea como completada. Esta funcionalidad deberá ser intuitiva y de fácil acceso. Al marcar como completada una tarea se detiene el registro de tiempo si es que se está llevando a cabo.

Reabrir una tarea Debe ser posible reabrir una tarea que ha sido marcada como completada. Al igual que la funcionalidad anterior deberá ser intuitiva y de fácil acceso.

Editar tarea Se deberá poder editar una tarea, modificando los datos ingresados en la creación de la misma. También se podrá editar los suscriptores asociados.

Ver tareas pendientes El usuario deberá poder consultar un listado con las tareas pendientes para el día, semana y mes.

Comentarios Se deberá poder ver y realizar nuevos comentarios sobre una tarea.

Notificaciones El usuario deberá recibir una notificación con:

- Recordatorios de tareas.
- Cuando se modifica una tarea a la que se esta suscrito o tiene asignada.

Sincronización Se deberán sincronizar las tareas con las almacenadas en Feng Office. El usuario deberá especificar los datos necesarios para poder configurar la sincronización de la cuenta. Se deberá poder especificar filtros que permitan definir que tareas quiere sincronizar.

Contactos

Agregar contacto Se deberá poder ingresar un contacto especificando los siguientes datos:

- Nombre
- Apellido
- Correo electrónico
- Tipo (Persona o Empresa)

Relacionado a su información personal puede ingresar los siguientes datos:

- Dirección
- Ciudad
- Estado
- Código postal
- País
- Sitio web
- Teléfono
- Fax
- Celular
- Fecha de nacimiento
- Zona horaria
- Suscriptores

Relacionado a su trabajo puede ingresar los siguientes datos:

- Nombre de la empresa
- Departamento
- Título
- Dirección
- Ciudad
- Estado

- Código postal
- País
- Sitio web
- Teléfono
- Fax
- Asistente
- Callback

Obtener número de teléfono o mail del contacto Se deberá poder seleccionar un contacto, y poder enviar un mail o llamarlo.

Documentos

Buscar y descargar un documento El sistema deberá poder desplegar un listado de documentos dentro del cual se podrá:

- Buscar y descargar un documento recorriendo la lista manualmente.
- Buscar y descargar un documento ingresando un texto.

Notas

Ingresar una nota Se deberá poder ingresar una nota especificando los siguientes datos:

- Título
- Etiqueta
- Texto
- Si tiene cuenta de Feng Office debe especificar workspace, habilitar comentarios de usuarios, comentarios anónimos y asociar suscriptores.

3.1.2. Requerimientos no funcionales

Robustez

El sistema deberá ser robusto de tal forma de no permitir la pérdida de datos tales como tareas, notas, contactos y documentos.

Facilidad de uso

El sistema deberá ser amigable e intuitivo para el usuario.

Internacionalización

El sistema deberá permitir cambiar fácilmente de idioma.

Portabilidad

El sistema deberá funcionar correctamente en el sistema operativo Android.

Concurrencia

El sistema deberá funcionar correctamente cuando varios usuarios se conecten al sistema al mismo tiempo.

Desempeño

El sistema garantizará la confiabilidad, la seguridad y el desempeño del sistema informático a los diferentes usuarios. En este sentido la información almacenada podrá ser consultada y actualizada permanente y simultáneamente, sin que se afecte el tiempo de respuesta.

3.2. Alcance del proyecto

El alcance del proyecto incluye:

- La selección de una plataforma para el desarrollo de la solución.
- El diseño de la solución.
- La construcción de un prototipo que pruebe los aspectos principales de la solución diseñada.

La construcción del prototipo abarca un subconjunto de los requerimientos que se describieron en la sección anterior.

3.2.1. Ingresar una tarea

Se deberá poder ingresar una tarea especificando los siguientes datos: nombre, fecha de comienzo, fecha límite, prioridad, descripción, tiempo estimado y recordatorios.

3.2.2. Buscar una tarea

El sistema deberá poder desplegar un listado de tareas dentro del cual se podrá:

- Buscar una tarea recorriendo la lista manualmente.
- Filtrar las tareas.
- Ver las asignadas al usuario o al equipo del usuario.
- Buscar una tarea ingresando un texto.

3.2.3. Ver detalles de una tarea

El sistema deberá permitir ver un resumen con los datos de una tarea. En dicho resumen se podrá ver los datos característicos de la tarea y acceder a los comentarios asociados.

3.2.4. Completar una tarea

El usuario deberá poder marcar una tarea como completada. Esta funcionalidad deberá ser intuitiva y de fácil acceso. Al marcar como completada una tarea se detiene el registro de tiempo si es que se está llevando a cabo.

3.2.5. Reabrir una tarea

Debe ser posible reabrir una tarea que ha sido marcada como completada. Al igual que la funcionalidad anterior deberá ser intuitiva y de fácil acceso.

3.2.6. Editar tarea

Se deberá poder editar una tarea, modificando los datos ingresados en la creación de la misma. También se podrán editar los suscriptores asociados.

3.2.7. Ver tareas pendientes

El usuario deberá poder consultar un listado con las tareas pendientes para el día, semana y mes.

3.2.8. Comentarios

Se deberá poder ver y realizar nuevos comentarios sobre una tarea.

3.2.9. Notificaciones

El usuario deberá recibir una notificación con: recordatorios de tareas. Cuando se modifica una tarea a la que se esta suscrito o tiene asignada.

3.2.10. Sincronización

Se deberán sincronizar las tareas con las almacenadas en Feng Office. El usuario deberá especificar los datos necesarios para poder configurar la sincronización de la cuenta. Se deberá poder especificar filtros que permitan definir qué tareas quiere sincronizar.

4. Solución propuesta

En esta sección se muestran los aspectos de la solución propuesta que se consideran más relevantes en relación a los objetivos planteados para el proyecto.

Se muestra en la primer subsección los lineamientos a seguir en lo que respecta al diseño de interfaz de usuario. Esto es, diseño de las pantallas y detalles de interacción. Luego, la arquitectura de la aplicación mostrando cómo interactúan los componentes que forman parte del sistema.

4.1. Selección de Android como plataforma

Dentro de cada uno de los enfoques mostrados en el estado del arte hay varias alternativas, en este documento se mostraron algunas, las que se consideraron más importantes e interesantes. Como se menciona en [55] la documentación suele ser demasiado abundante y poco formal, lo que dificulta muchas veces realizar el relevamiento y puede llevarnos a caer en una elección incorrecta. Para escoger alguno de los enfoques anteriores hay que considerar cuales son las necesidades de la aplicación a desarrollar.

Como menciona Brian Fling en [31] el desarrollo de aplicaciones nativas se recomienda en los siguientes casos:

- Se desea cobrar por utilizar la aplicación.
- Desarrollo de videojuegos.
- Acceder a la ubicación del usuario.
- Tener acceso a la cámara.
- Utilizar acelerómetros.
- Acceder al sistema de archivos.
- Uso offline de la aplicación

También se recomienda desarrollar aplicaciones web en los casos que no se cumplan las condiciones anteriores para el desarrollo de aplicaciones nativas. Para desarrollar de forma rápida aplicaciones para varias plataformas que no tengan mucha complejidad se podrían utilizar los generadores de código, debido a las ventajas mencionadas.

Consideramos a HTML5 como una buena alternativa para el desarrollo de este proyecto.

Es viable el desarrollo de aplicaciones web para dispositivos móviles, si bien hay que tener en cuenta las limitaciones de cada cliente web, al desarrollar una aplicación web se está desarrollando con una tecnología que está presente en todos los dispositivos. Para el tipo de sistema que se pretende desarrollar en este proyecto no aplican varias de las nuevas funcionalidades que provee HTML5, como canvas o video, pero lo positivo de utilizar este enfoque de desarrollo es la portabilidad en los distintos dispositivos y la prometedora escalabilidad que se tiene a futuro, ya que los navegadores móviles en cada liberación aumentan la compatibilidad con HTML5. Por lo anterior se llegó a la conclusión de que el enfoque más adecuado para este proyecto es desarrollar una aplicación web basada en HTML5.

Cuando se propuso utilizar dicha plataforma, el cliente informó que en Feng Office ya tenían la intención de realizar el desarrollo de una aplicación para móviles basada en HTML5. Se acordó realizar un prototipo para Android porque presentaba ciertas ventajas para la realización de este proyecto:

- Contábamos con dispositivos para probar.
- Las herramientas de desarrollo son gratuitas.
- Hay buena documentación.
- Se contaba con cierta experiencia.

4.2. Diseño de interfaz de usuario

4.2.1. Lineamientos

Lo que siguen son algunos lineamientos a tener en cuenta a la hora de construir interfaces gráficas para dispositivos móviles. En particular se muestra la forma en que serían aplicables dentro del diseño para Feng Mobile.

Autocompletar campos

Autocompletar los campos ahorra tiempo y esfuerzo del usuario. Convierte la tarea de ingresar un texto relativamente complejo en seleccionar un elemento de una lista [44].

Debe ser posible ingresar de forma rápida una tarea ya que es común que el usuario disponga de un corto período de tiempo para ingresar una o varias. Un ejemplo de uso sería un usuario en una reunión ingresando tareas mientras habla con el resto de los participantes. Este patrón adquiere valor cuando el dato que se quiere ingresar posee cierta complejidad como ser:

- Nombres
- URL

- Direcciones de correo
- Textos muy largos

Al autocompletar los datos de la lista también se previenen posibles errores de escritura. En los dispositivos móviles este patrón es de gran importancia. Escribir en celulares no es una tarea sencilla y al evitarla se ahorra tiempo y frustración del usuario.

El patrón autocompletar se podría aplicar en el campo de texto de "Ingresar tarea". Para ello debería desplegar una lista de sugerencias a medida que el usuario escribe. Estas sugerencias podrían ser:

- Texto de tareas ya ingresadas. Útil para usuarios con tareas repetitivas (no periódicas).
- Nombres de contactos. Se pueden mostrar los nombres de contactos del teléfono y de Feng Office.
- Acciones sobre los contactos. Al sugerir un contacto también sugerir una acción por ejemplo: "Llamar a Juan", "Reunión con Juan".
- Correos. Sugerir direcciones de email de los contactos. También podría incluir una acción por ejemplo: "Mandar mail a ejemplo@gmail.com".
- URL. Sugerir URL's ya visitadas o asociadas a algún contacto, o incluir una acción por ejemplo: "Visitar www.fengoffice.com".

Autocompletar (AutoComplete, o Autocompletion en inglés) es un patrón de diseño de interfaz. En [44] y [73] se puede acceder a información más detallada de dicho patrón.

Ayuda multinivel

Con el fin de brindar ayuda que guíe a usuarios con diferente nivel de experiencia (desde principiantes a expertos) se plantea el uso de un sistema de ayuda multinivel. En un sistema de ayuda multinivel se utilizan distintos tipos de ayuda de forma de brindar la ayuda apropiada a cada nivel de experiencia [44]. A continuación se muestran los distintos tipos de ayuda a utilizar en la aplicación.

Input prompts La idea de este patrón es precargar un campo de texto con un mensaje que indique al usuario qué se debe escribir en el. Hace que los campos de texto sean auto explicativos.

Esto puede ser implementado por ejemplo en el campo de texto utilizado para ingresar una nueva tarea. De esta forma no es necesario que incluya un label quedando bien claro su propósito.

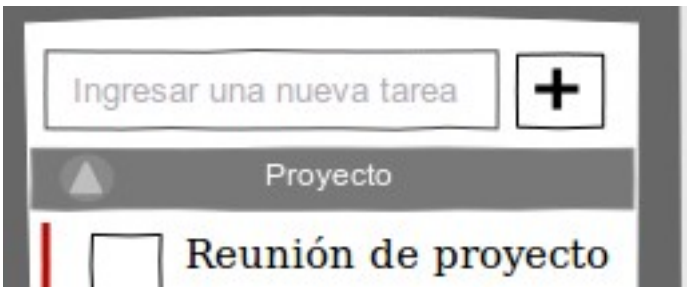

Figura 4.1.: Input prompt en el campo de ingresar tarea

Cuadros con ayuda contextual La intención de los cuadros de ayuda contextual es desplegar información introductoria para algunas secciones de la aplicación. Mostrando un cuadro con texto explicando elementos de la interfaz relevantes cuando estos se presentan al usuario por primera vez. Se plantean como un híbrido entre los Inline Help Box y Dynamic Input Hints presentados en [69, 61] referencias tomadas de [44].

En la figura 4.2 se muestra un cuadro de ayuda contextual de ejemplo. Se exagera el botón "OK" para mostrar que debe existir la posibilidad de eliminar el cuadro para que no se vuelva a mostrar.

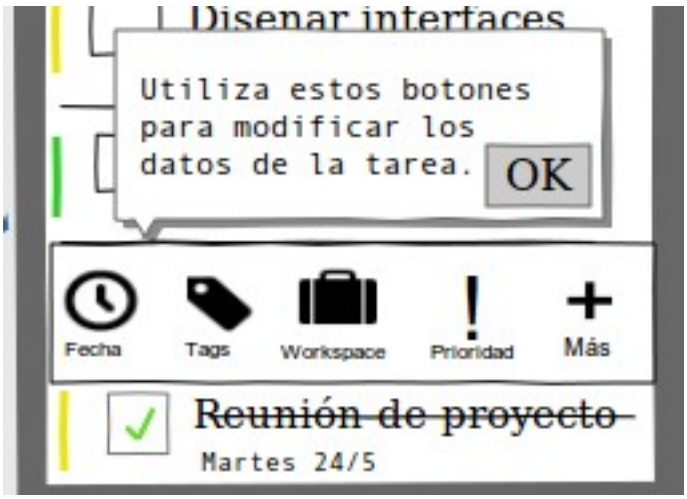

Figura 4.2.: Cuadro con ayuda contextual mostrado la primera vez que se despliegan los botones de edición rápida

Sugerencias Se plantea incluir en el menú un botón que permita acceder a una pantalla donde se despliegan consejos sobre el uso de la aplicación. Debe haber botones que permitan navegar por las sugerencias. Una vez mostrados todas las sugerencias se accede a una ayuda completa donde se describe en detalle las funcio-

nalidades de la aplicación. Este último es el nivel más completo de ayuda del sistema multinivel planteado para la aplicación.

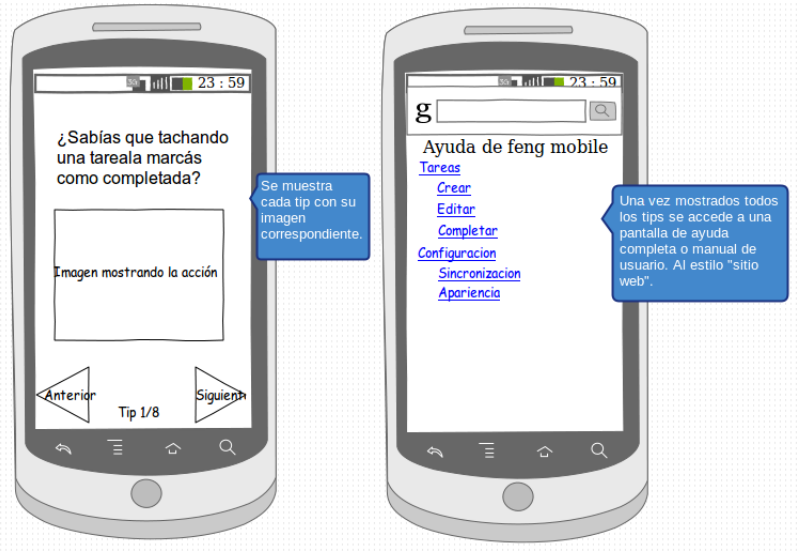

Figura 4.3.: Bosquejos mostrando las sugerencias

4.2.2. Bosquejos

Para el diseño de la interfaz gráfica de la aplicación se realizaron algunos bosquejos de las pantallas principales, los cuales fueron validados con Feng Office.

La primera pantalla que se diagramó fue el listado de tareas. Se definió que esta pantalla sea la pantalla principal del módulo de tareas. De esta forma el usuario puede acceder de forma rápida a las tareas pendientes. En la parte superior de la pantalla hay un campo de texto que permite ingresar una tarea sin necesidad de ingresar datos adicionales. Al crear una tarea, esta se despliega en el listado de tareas y debajo se muestra un menú con opciones que permiten modificar sus datos. A dicho menú lo llamamos “menú de edición rápida”.

Para cada tarea del listado, el diseño es el mismo: se muestra el título de la tarea y un dato extra como la fecha límite y la persona asignada. A la izquierda se coloca un checkbox para indicar si la tarea está completada. Se subraya el título de la tarea cuando está completada (checkbox tildado). Se definió un código de colores para indicar la prioridad de cada tarea dentro del listado: rojo para urgente, amarillo para prioridad alta, verde para normal y celeste para baja. Estos colores son mostrados en una barra contra el borde del recuadro que muestra cada tarea.

Al presionar de forma prolongada sobre una tarea se muestra un menú contextual que permite marcar una tarea como completada, ir a una pantalla de edición avanzada, comentar y eliminar una tarea.

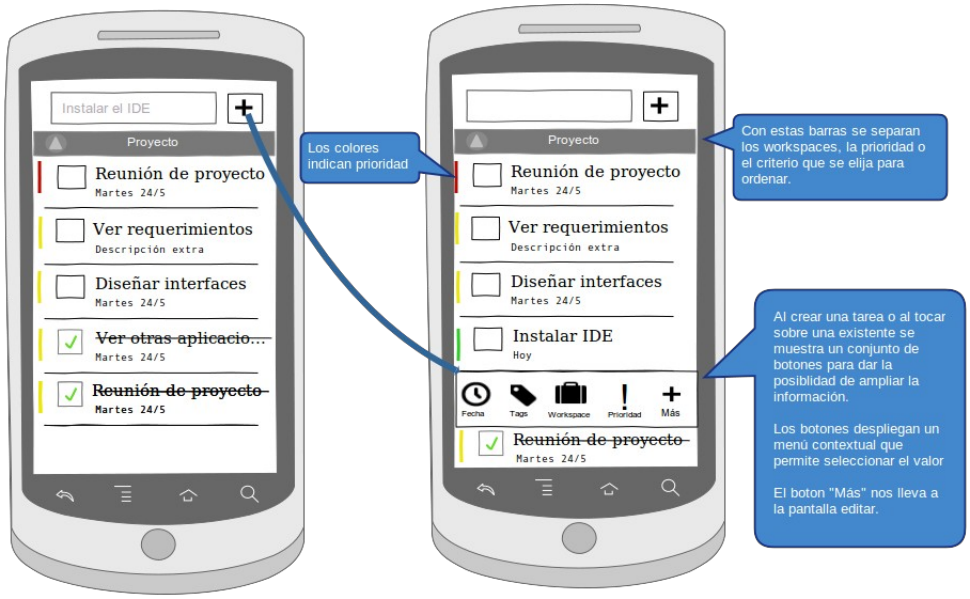

Figura 4.4.: Listado de tareas y menú de edición rápida

Seleccionando la opción “Más” del menú de edición rápida o la opción “Editar” del menú contextual se accede a la pantalla de edición de tareas (“Edición avanzada”). Esta pantalla es un formulario que permite editar todos los datos asociados a una tarea. En la figura 4.5 se muestra un bosquejo de la pantalla de “Edición avanzada”.

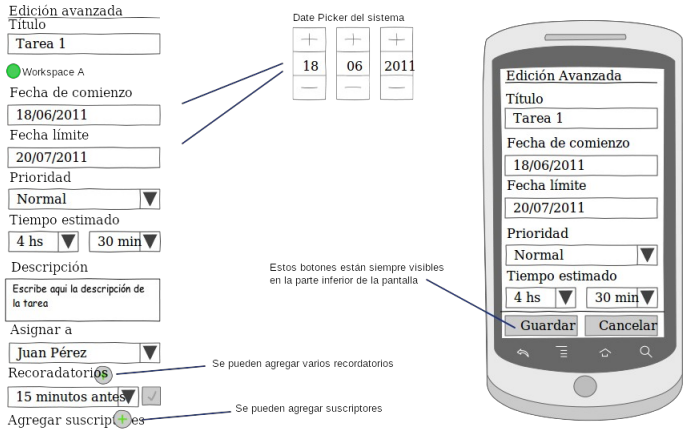

Figura 4.5.: Formulario de edición de tareas

Realizar comentarios es una acción que en Feng Office se puede aplicar a todos los objetos de los diferentes módulos. Por ello se hicieron bosquejos de la pantalla “Comentar Tarea”. Esta pantalla es accesible desde el menú contextual. Consiste de un listado de comentarios, con el texto, el autor y la fecha. Al final de la lista se despliega un botón que permite cargar más comentarios (ver patrón Infinite List

en [44]). En la parte inferior se muestra un cuadro de texto para realizar un nuevo comentario.

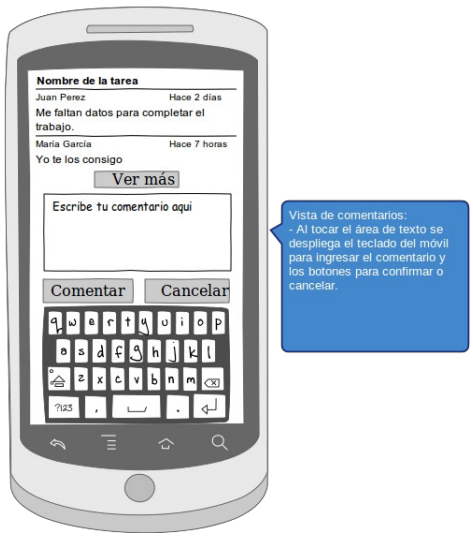

Figura 4.6.: Pantalla para ver y hacer comentarios

4.2.3. Subtareas

El módulo de tareas de Feng Office permite representar las tareas como una composición de subtareas. Las subtareas son tareas, poseen los mismos atributos y permiten las mismas acciones. Dadas las restricciones de tamaño de pantalla existentes en muchos de los dispositivos móviles es importante definir la representación de la estructura de subtareas.

Para definir la forma en que se representarán las subtareas se hizo un relevamiento de aplicaciones de tareas para celulares que permitan registrar subtareas el cual se resume a continuación.

Ultimate ToDo-List Al igual que en Feng Office las subtareas son tareas. Hace notar la jerarquía cambiando la indentación de las tareas en el listado. Como muestra todas las tareas en el listado principal se pueden ver las subtareas que forman parte de una tarea en particular.
El problema principal que tiene esta representación es que a medida que se avanza en los subniveles queda menos espacio disponible para los datos. Para registrar una subtarea primero hay que crearla como tarea, y luego asignarle la tarea "padre".

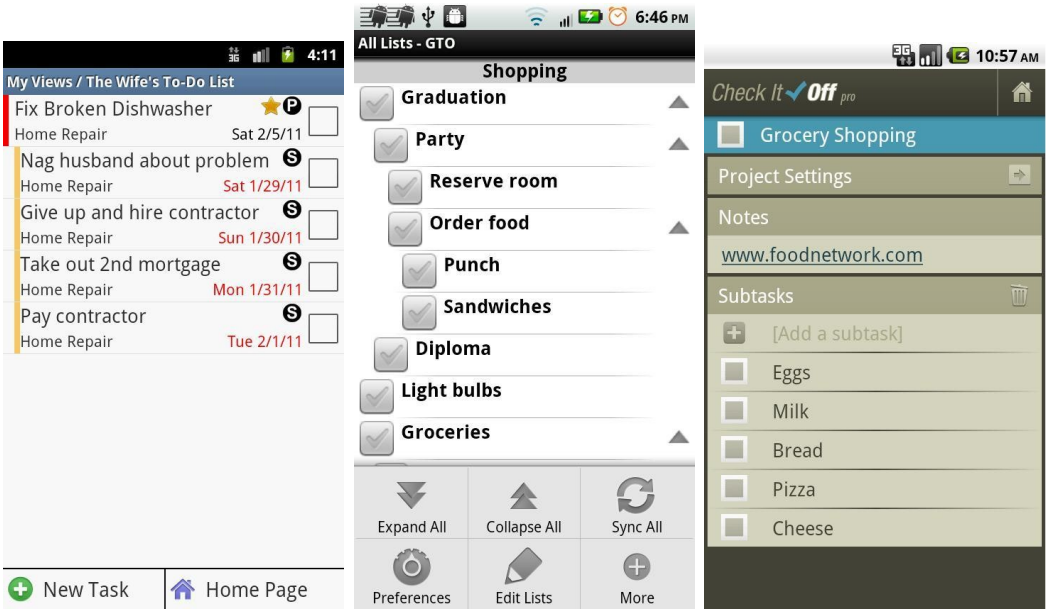

Figura 4.7.: Ultimate ToDo-List, Google Task Organizer y CheckitOff Pro

Google Task Organizer Es una aplicación que permite manejar las tareas de Google Tasks desde un celular Android. Tiene una interfaz muy sencilla que muestra sólo texto y un checkbox para marcarla como completada.
Muestra las subtareas utilizando indentación. Permite colapsar las subtareas que están debajo de una tarea padre para poder visualizar mejor el listado. Como se muestra muy poco para cada tarea es posible mostrar más subniveles en pantalla. Una dificultad que presenta es que al no estar bien demarcados los bordes se pierde noción de los niveles en la indentación.

CheckitOff Pro Las subtareas en esta aplicación son un dato de la tarea padre. No es posible representar una jerarquía de subtareas ya que se permite solo un nivel. Crear una subtarea es fácil ya que no hay que ingresar más datos que el título.
Como las subtareas son un dato de las tareas no se modifica el listado para representar la jerarquía. Para cada tarea hay una pantalla que permite editar y ver las subtareas.

En base a estas tres aplicaciones se propone: ingresar las subtareas dentro de la pantalla de editar tareas. De esta manera no hay que agregar funcionalidades extras al listado de tareas. Las subtareas se despliegan en el listado de tareas y se muestran utilizando indentación. En el listado también se deberá poder tratar a las subtareas como tareas, de manera que sea posible editarlas y agregarles más subtareas.

4.3. Arquitectura

Se plantea una arquitectura cliente-servidor en la que el subsistema cliente interactúa con el servidor por medio de un proceso de sincronización. El cliente es una aplicación Android nativa que ejecuta en el móvil y el servidor es una instancia del sistema web Feng Office. A continuación se muestran las características principales de los componentes y los aspectos de diseño de la arquitectura que se consideraron importantes para el desarrollo de esta aplicación.

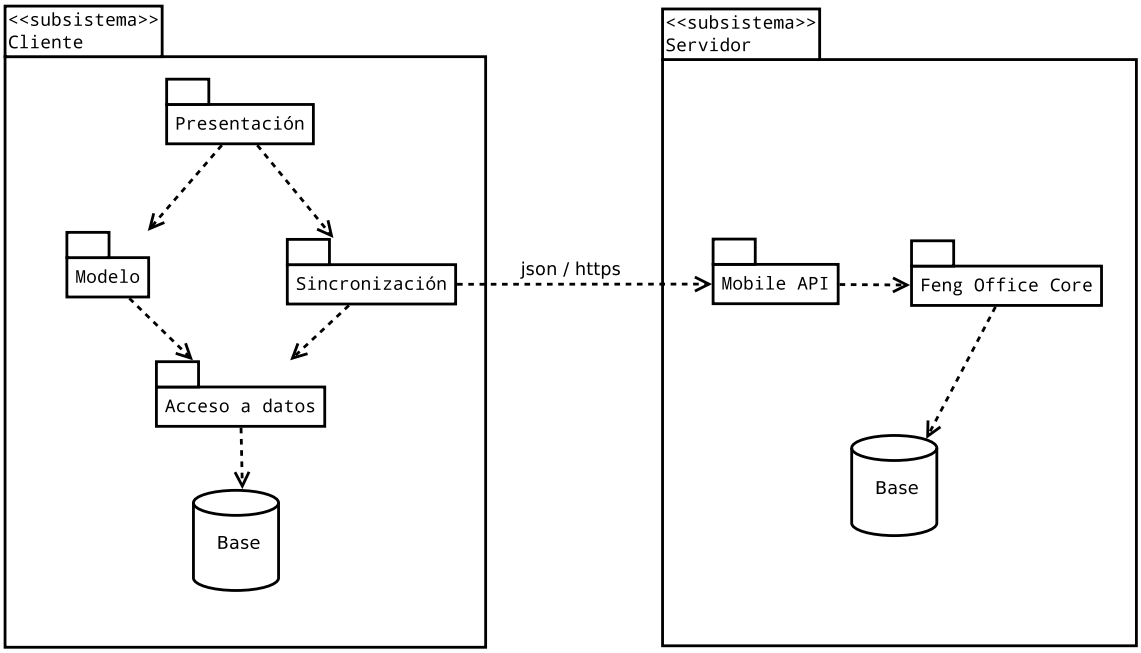

Figura 4.8.: Componentes de la arquitectura propuesta

4.3.1. Cliente

El componente cliente es una aplicación nativa Android. Si quisiéramos desarrollar un componente cliente para otro tipo de plataforma, sería razonable que tuviera las características que se definen en esta sección.

El cliente está formado principalmente por:

- Una interfaz de usuario o presentación.
- Una base de datos local.
- Un proceso en background para ejecutar la sincronización.

La interfaz de usuario deberá cumplir con las decisiones de diseño definidas en la sección anterior. Al tratarse de una aplicación nativa, es deseable que cumpla con las recomendaciones de diseño de la plataforma para la que se desarrolla, en este

caso Android [20]. De esta manera se tendrá una interfaz más consistente con el resto del sistema que el usuario ya está acostumbrado a usar.

La interfaz de usuario debe poder acceder a un almacenamiento local de los datos. Esto permite la utilización del sistema cuando el dispositivo no tenga comunicación con el servidor. Para esta aplicación se utiliza el motor de base de datos provisto por la plataforma Android llamado SQLite3 y otro tipo de almacenamiento clave-valor, para preferencias de usuario y otros valores que son necesarios para el funcionamiento de la aplicación.

Para mantener sincronizados los datos almacenados localmente con los del servidor el cliente deberá contar con un proceso de sincronización que ejecute en background. Este proceso envía al servidor los cambios realizados en el móvil y recibe los del servidor, notifica los cambios a la presentación para que se actualice la interfaz de usuario.

Las características definidas anteriormente pueden representarse como una arquitectura en tres capas que se puede ver en la figura 4.8. En la capa de presentación se implementa la interfaz de usuario. La capa de negocio está formada por el módulo llamado "Modelo" que contiene la lógica de negocio implementada por la aplicación cliente y el módulo "Sincronización" que contiene la lógica necesaria para implementar el proceso de sincronización. En la capa de persistencia se resuelve el acceso a los datos almacenados localmente los cuales son accedidos desde la capa superior.

4.3.2. Servidor

El componente servidor es un módulo de Feng Office desarrollado en PHP. Este módulo expone la API que es utilizada por el componente Cliente para la sincronización. Utiliza funcionalidades internas de Feng Office para poder manipular los objetos que la aplicación gestiona.

Además de la API para realizar la sincronización, el componente servidor también expone funciones para poder realizar la autenticación de los clientes.

La integración del componente servidor con la aplicación web Feng Office se realizó teniendo en cuenta los siguientes criterios:

- Minimizar los cambios en el código de Feng Office. Esto evita que se alteren las funcionalidades ya existentes de la aplicación web. También simplifica la integración del módulo servidor con nuevas versiones de Feng Office.
- Maximizar la reutilización de la lógica ya existente. Al no conocer en profundidad la lógica interna de la aplicación web de Feng Office, la reutilización de código existente facilita el desarrollo de la API del servidor.
- Evitar modificaciones en la base de datos. Por el mismo motivo que se evitan cambios en el código fuente, se trató de evitar cambios en los esquemas de la base de datos.

4.3.3. Comunicación y sincronización

Feng Office es una aplicación web programada en PHP, que ejecuta en un servidor web. Por otra parte, la plataforma Android contiene dentro de su API todas las librerías necesarias para invocar web services sobre HTTP. Se decidió que la comunicación entre los componentes se haga a través del protocolo HTTP.

Como el componente cliente va a correr en un dispositivo móvil debemos tener en cuenta que la sincronización consume dos recursos importantes: batería y conexión de datos. Se usa el formato JSON para intercambiar los datos en la sincronización ya que es un formato que es sencillo de interpretar y requiere pocos bytes [48]. Android y PHP incluyen funciones y parsers que permiten utilizar JSON.

Se definió que el intercambio de objetos en la sincronización se haga de manera incremental, intercambiando únicamente los objetos que se modificaron tanto en el cliente como en el servidor. Para ello se implementó un mecanismo de control de versiones de los objetos que se detalla más adelante en la sección 5.4. Esto también ayuda a minimizar la cantidad de datos transmitidos.

La sincronización entre el cliente y el servidor se ejecuta de forma periódica, y las conexiones se inician desde el cliente. El cliente ejecuta la sincronización si el estado actual del dispositivo móvil y la conexión de datos lo permiten. El período de sincronización se configura desde las preferencias de la aplicación para que el usuario lo pueda adaptar a sus necesidades.

4.3.4. Seguridad

Seguridad en la conexión

Se definió que se asegurarán las conexiones utilizando SSL. El componente cliente debe acceder a la URL correcta (indicando el protocolo HTTPS) para que la conexión sea segura. Esto es transparente para el desarrollo del componente cliente ya que las librerías utilizadas por Android para establecer conexiones HTTP soportan también conectarse a servidores usando HTTPS. Para configurar SSL en el componente servidor hay que configurar un certificado en el servidor web sobre el cual ejecuta, no es necesario modificar código.

Autenticación

Se utiliza autenticación por medio de usuario y contraseña para asegurar el acceso a la API. Para autenticar a un usuario, el componente cliente envía nombre de usuario y contraseña al servidor. El servidor, en caso de que el usuario y la contraseña sean correctos retorna un token. El cliente debe almacenar ese token mediante el cual

puede acceder a la API de sincronización autenticado como el usuario. El componente cliente de esta manera no almacena la contraseña del usuario ya que solo es necesario almacenar el token de autenticación.

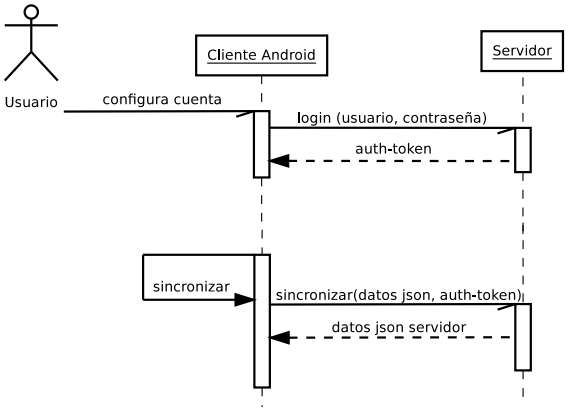

Figura 4.9.: Diagrama de secuencia para la autenticación

5. Implementación del prototipo

Para probar el diseño de la interfaz de usuario y la arquitectura propuesta se desarrolló un prototipo. El mismo está formado por una aplicación nativa para Android y el módulo MobileAPI que se muestra en la arquitectura. La aplicación Android implementa los requerimientos asociados con el módulo de tareas y la sincronización. El módulo MobileAPI implementa la API de sincronización dentro del código de Feng Office.

5.1. Cliente

El prototipo fue desarrollado tratando de aprovechar al máximo las facilidades brindadas por la plataforma Android. Es recomendable que la interfaz de usuario de una aplicación nativa siga las convenciones de la plataforma para la cual está desarrollada. De esta forma la aplicación se integra de forma consistente con el resto del sistema [20]. La interfaz de usuario fue creada usando en su mayoría los componentes gráficos que se incluyen en Android y siguiendo las recomendaciones presentes en la documentación.

La lógica del prototipo fue desarrollada extendiendo los componentes provistos por el SDK de Android. Los componentes mostrados en la arquitectura propuesta se implementaron haciendo uso de diferentes partes del framework provisto por dicha plataforma. A continuación se muestra cómo fueron implementados dichos componentes.

5.1.1. Presentación

En Android las pantallas que forman la interfaz de usuario pueden ser declaradas desde el código y también especificadas por medio de un archivo en formato XML. En la construcción del prototipo se utilizó mayormente la especificación de interfaces con XML ya que es simple y fácil de entender. En casos como la representación de las subtareas y el menú de edición rápida fue necesario adaptar la forma en que se despliegan algunos de los componentes gráficos nativos.

Los componentes incluidos en Android que más se utilizaron fueron: *ExpandableListActivity*, *ListActivity*, *Dialogs* y *Notifications*. *ExpandableListActivity* fue utilizada en la actividad principal de la aplicación para desplegar las tareas separadas según

el orden seleccionado (prioridad, fecha o workspace). *ListActivity* fue utilizada en las pantallas en las que se despliega únicamente un listado, como pueden ser la de Comentarios y la de Editar Subtareas. Se utilizaron *Dialogs* cuando fue necesario solicitar al usuario que seleccione un valor de una lista. El componente *Notifications* se usó para mostrar en la barra de notificaciones el alta de tareas que se reciben desde la sincronización.

Las pantallas de configuración fueron desarrolladas utilizando un tipo de actividad de Android llamado *PreferenceActivity*. Esta actividad funciona tomando de un XML la especificación de los campos que se van a configurar y despliega la pantalla de configuración al usuario. *PreferenceActivity* también resuelve el almacenamiento de los valores configurados. Los datos almacenados por la actividad de configuración que se desarrolle de esta manera, pueden ser accedidos desde otras actividades de la aplicación. La ventaja principal de utilizar este tipo de actividad es que el diseño de la interfaz generada y la interacción son iguales a las mostradas por la configuración del sistema Android.

Figura 5.1.: Configuración de la aplicación usando *PreferenceActivity*

5.1.2. Acceso a datos

Para almacenar los datos de las tareas se utilizó el motor de base de datos SQLite3 que se incluye en Android. La aplicación no accede directamente a la base de datos

sino que lo hace a través de un *ContentProvider* que se desarrolló para tal fin. Los *ContentProviders* de Android son objetos que manejan datos compartidos de las aplicaciones. Se pueden almacenar datos en el file system, en la base de datos, en la web o en cualquier otro medio de almacenamiento que la aplicación pueda acceder. A través de los *ContentProviders* otras aplicaciones pueden consultar o modificar datos. El uso de *ContentProviders* para acceder a las tareas es requerido para la implementación de la sincronización como se muestra en la sección 5.1.3.

Se utilizaron *SharedPreferences* para almacenar algunos valores relacionados con la configuración y sincronización. Las *SharedPreferences* permiten almacenar datos de tipos primitivos. Se trata de una API que permite almacenar valores en un archivo en formato XML que representa una estructura de tipo mapa clave-valor. Fueron almacenados de esta manera las preferencias de sincronización, la URL del servidor, el número de revisión de la sincronización (ver sección 5.4).

5.1.3. Autenticación y sincronización

La autenticación y sincronización se implementaron haciendo uso de los componentes de Android llamados *AccountManager* y *SyncAdapter*. De esta manera, la aplicación obedece a la configuración que el usuario haga en el sistema operativo para habilitar o deshabilitar la sincronización automática.

El proceso de sincronización definido en la arquitectura se implementó con un *SyncAdapter*. El cual es un servicio de la aplicación que es invocado por un manejador de sincronizaciones que forma parte del sistema operativo y mantiene una cola de *SyncAdapters* a invocar. De esta forma se delega al sistema operativo la responsabilidad de elegir el momento en el que se va a sincronizar. El sistema operativo decide si es conveniente ejecutar la sincronización dependiendo de si hay otras aplicaciones sincronizando, o si el estado actual del sistema lo permite.

El *SyncAdapter* creado está asociado en su declaración al *ContentProvider* de tareas, esto es necesario para indicar al sistema operativo qué tipo de contenido se sincroniza. Para agregar otro contenido a sincronizar, se debe crear otro *SyncAdapter* y asociarlo al *ContentProvider* correspondiente. Como se muestra en la figura 5.2 se configuró la cuenta de Feng Office en el sistema y el contenido “Tareas” para sincronizar.

Es necesario tener una cuenta de usuario para poder utilizar un *SyncAdapter*. Existe en Android un componente llamado *AccountManager* que se encarga de administrar las credenciales de usuarios almacenadas en el sistema. Las cuentas registradas en el *AccountManager* aparecen en la sección de “Cuentas y sincronización” dentro de la pantalla de ajustes del dispositivo. Se creó un tipo de cuenta para Feng Office y se implementaron las clases necesarias para poder ejecutar la autenticación del usuario y registrar las credenciales en el *AccountManager*.

Para enviar los cambios realizados en las tareas desde el cliente al servidor, se notifica al sistema qué contenido ha sido modificado desde el *ContentProvider*. Al recibir

esta notificación, el sistema registra que el contenido debe ser sincronizado y ejecuta una sincronización ni bien sea posible. Para recibir los cambios del servidor se decidió ejecutar una sincronización periódica. La API brindada por la versión de Android usada en el prototipo (ver sección 5.5) no permite configurar el período de la sincronización, pero si solicitar una. Se implementó un proceso que solicita sincronizaciones periódicas según el intervalo que el usuario configure en las preferencias.

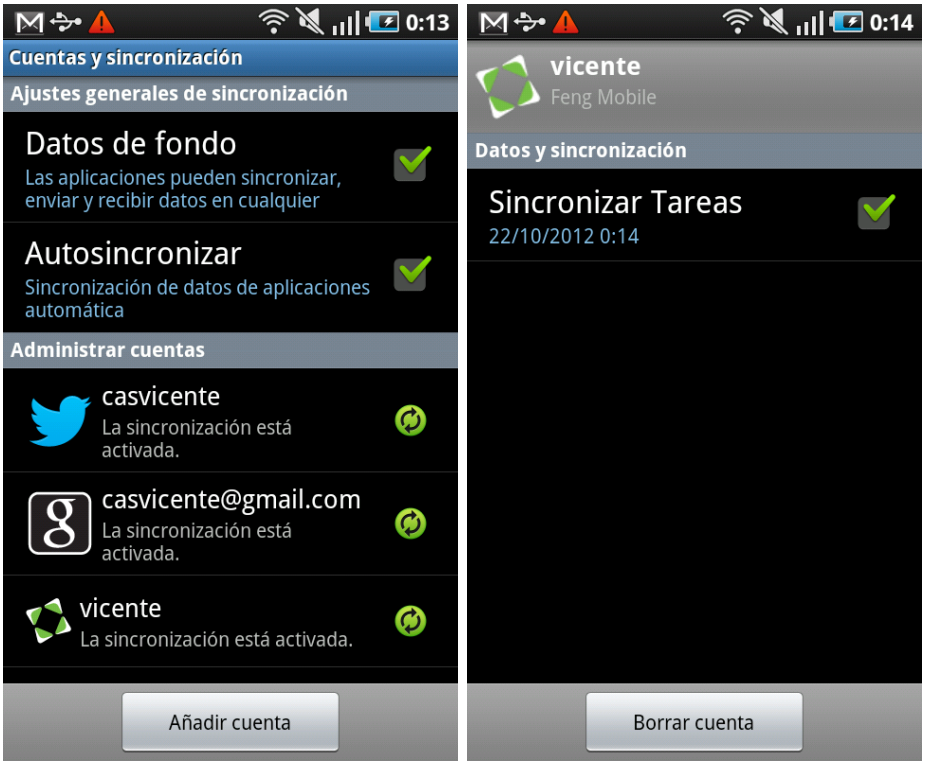

Figura 5.2.: Cuenta de Feng Office en la pantalla de Cuentas y Sincronización

5.2. Servidor

El componente servidor descrito en la arquitectura se implementó agregando algunas funcionalidades en el código de Feng Office con el fin de exponer la API necesaria para la sincronización. La principal dificultad encontrada a la hora de implementar esta parte del prototipo fue entender el funcionamiento interno de Feng Office, que si bien no es una arquitectura compleja no se tenía mucho conocimiento en profundidad.

El código fuente de la aplicación web Feng Office está organizado en una arquitectura Model View Controller. En dicho código hay controladores para cada funcionalidad brindada por la aplicación, modelos para cada tipo de objeto manejado por el sistema y vistas para generar las pantallas. En la wiki de Feng Office hay documentación

de una API [60] que se encontraba en desarrollo al momento que comenzamos la construcción del prototipo. Dicha API está programada en un controlador llamado ApiController.php, y se accede a la misma por HTTP a través de una URL cuyos parámetros permiten seleccionar el controlador y el método que se quiere ejecutar.

Para implementar la sincronización se modificó el controlador ApiController agregándole un método que implementa el algoritmo que se describe en la sección 5.4. También se agregaron otros métodos para poder obtener los workspaces y hacer comentarios, objetos que no se incluyeron en la sincronización para el desarrollo del prototipo.

Como se indica en la wiki de Feng Office anteriormente mencionada, para invocar a los métodos de la API es necesario agregar un token de autorización como parámetro. El token de autorización se encuentra almacenado en la base de datos, dentro de la tabla que almacena los usuarios del sistema. Resolver la autorización implica entonces obtener dicho token a partir del nombre de usuario y la contraseña, por lo que se decidió agregar un método a la API llamado "synclogin". Como hay que pasar como parámetro el token de autorización para invocar métodos del ApiController, el método synclogin debe ser implementado en otro controlador. Observando el código fuente de la página de autenticación en la instalación de Feng Office utilizada, se encontró que el controlador utilizado para autenticar al usuario es AccessController.php. El método synclogin que retorna el token de autorización se agregó en dicho controlador.

5.3. Seguridad

Con el fin de probar la conexión segura entre el cliente y el servidor se decidió habilitar en el servidor web utilizado (Apache2) el módulo que implementa SSL. De esta forma configurando en el cliente una URL que indique el protocolo HTTPS se asegura la conexión sin necesidad de hacer cambios en el código fuente. Lo anterior funciona solamente si el certificado es válido, es decir si fue firmado por una autoridad certificadora [27]. Con un certificado que no es válido el cliente HTTP de Android lanza una excepción y no permite la conexión. Se plantearon las siguientes alternativas:

- Utilizar un certificado gratuito de prueba. Algunas empresas emisoras de certificados como Symantec y Geotrust [66, 12] permiten crear un certificado gratuito de treinta días de duración. Esta opción tiene la ventaja de no requerir cambios en el código fuente. La expiración del certificado es un inconveniente, ya que una vez pasada la fecha habría que solicitar uno nuevo.
- Modificar la aplicación para dar la posibilidad de aceptar o rechazar un certificado. Esta opción requiere que cuando el usuario configura la URL del servidor en la pantalla de preferencias, se le de la opción de aceptar o no el certificado, del mismo modo que lo hacen los navegadores. Si bien consideramos que es

una buena opción de cara al usuario, para implementarla habría que hacer cambios en el código que nos desviarían del objetivo de probar el uso de SSL en el prototipo.

- Modificar la aplicación para permitir un certificado no válido creado por nosotros. Lo que se plantea en esta alternativa es crear un certificado y hacer pocos cambios en la aplicación para aceptarlo. Con herramientas como OpenSSL y keytool [35, 72] es bastante sencillo crear y manejar certificados SSL. Los cambios a realizar en el código fuente son pocos. Esta fue la opción que se implementó en el prototipo.

Entonces, para habilitar SSL en el prototipo lo que se hizo fue:

1. Crear un certificado autofirmado utilizando OpenSSL como se muestra en [72].
2. Configurar mod_ssl [37] en el servidor Apache que corre Feng Office.
3. Incluir el certificado dentro de la aplicación Android cargando un KeyStore que lo incluya y configurar el cliente HTTP de la aplicación Android para confiar en ese certificado como se indica en [27].

5.4. Sincronización

Se definió que el algoritmo que implementa la sincronización sea genérico, es decir que pueda usarse para sincronizar todos los tipos de objetos manejados por Feng Office, de esta forma la lógica de la sincronización es reutilizable permitiendo agregar nuevos tipos de objetos sin mayor dificultad. También, se decidió que se intercambie entre cliente y servidor solamente los objetos que fueron modificados para evitar transferencias innecesarias de datos.

Para implementar la sincronización se desarrolló un algoritmo, en el que tanto el cliente como el servidor almacenan un valor que corresponde con la última versión de los datos, a dicho valor lo llamamos revisión. La idea del algoritmo es que en cada sincronización el cliente envía al servidor los datos que no están sincronizados junto con el número de revisión de la última sincronización ejecutada. El servidor aplica los cambios enviados por el cliente, incrementa el número de revisión y retorna los objetos con revisión mayor a la recibida desde el cliente. En el anexo C se muestran los pseudocódigos de los algoritmos usados en el cliente y servidor.

Para evitar realizar cambios en la base de datos de Feng Office, se utilizó como número de revisión la fecha de modificación que ya estaba presente en el esquema de datos de Feng Office para todos los objetos. La mayor fecha de modificado es la retornada por el servidor luego de cada sincronización a modo de “última revisión”.

Si bien el algoritmo de sincronización fue planteado de forma genérica, hay partes en el código del prototipo que son específicas de las tareas, sobre todo en lo que refiere a la obtención de los datos ya que los métodos que se encontraban disponibles en

el código no permitían acceder a todos los campos necesarios para implementar la sincronización. Por lo dicho anteriormente la sincronización entre cliente y servidor se implementó sólo para las tareas, datos como los workspaces disponibles y los comentarios asociados a las tareas, se obtienen por fuera de la sincronización. Si no hay comunicación con el servidor no se puede ver, ni hacer comentarios sobre las tareas en el prototipo, sí se podrán crear tareas que serán enviadas al servidor por el *SyncAdapter* cuando sea posible establecer la conexión con el servidor.

5.5. Entorno de desarrollo

En esta sección se muestran las herramientas utilizadas para el desarrollo del prototipo.

Herramienta	**Versión**	**Descripción**
Feng Office	2.1-beta	Desarrollo del componente servidor
Android SDK	2.1	API de Android utilizada para el desarrollo del componente cliente
Eclipse	Juno (4.2)	IDE
Apache	2.2.22	Servidor web donde corre la instalación de Feng Office
OpenSSL	1.0.1	Herramienta utilizada para crear el certificado SSL de prueba
PHP	5.3.10-1ubuntu3.4	Lenguaje utilizado por Feng Office
mysql-server	5.5.24-0ubuntu0.12.04.1	Servidor de base de datos
Sistema operativo	Linux 3.2.0-32-generic Ubuntu	Utilizado para correr el servidor y el emulador de Android
Móvil de prueba	GT-I9000T, Firmware 2.1-update1	Utilizado para probar el cliente Android
Subversion	1.6.17	Utilizado para manejar las versiones del código y documentos
LyX	2.0.4	Utilizado para redactar la documentación.

Cuadro 5.1.: Herramientas utilizadas

Fortalezas:

- son todas herrmientas de software libre que se pueden utilizar de forma gratuita y adaptarlas a la solución.

- están disponibles tanto para Linux como para Windows.

Debilidades:

- Trabajamos con máquinas con distinto sistema operativo, y tuvimos ciertas dificultades para configurar algunas herramientas.

6. Pruebas

En esta sección se muestran los casos de pruebas realizados a distintos usuarios. Las pruebas fueron realizadas con un celular Samsung Galaxy S con sistema operativo Android 2.1.

A continuación detallamos los casos de prueba realizados:

6.1. Casos de pruebas

Ingresar tarea

Permite al usuario ingresar nombre de la tarea, ingresando automáticamente la fecha actual como fecha límite.

1. Ingresar nombre de la tarea (identificar si se despliega la opción autocompletar de contactos).
2. Hacer clic en el botón "+" (se da de alta la tarea con la fecha actual como fecha límite).

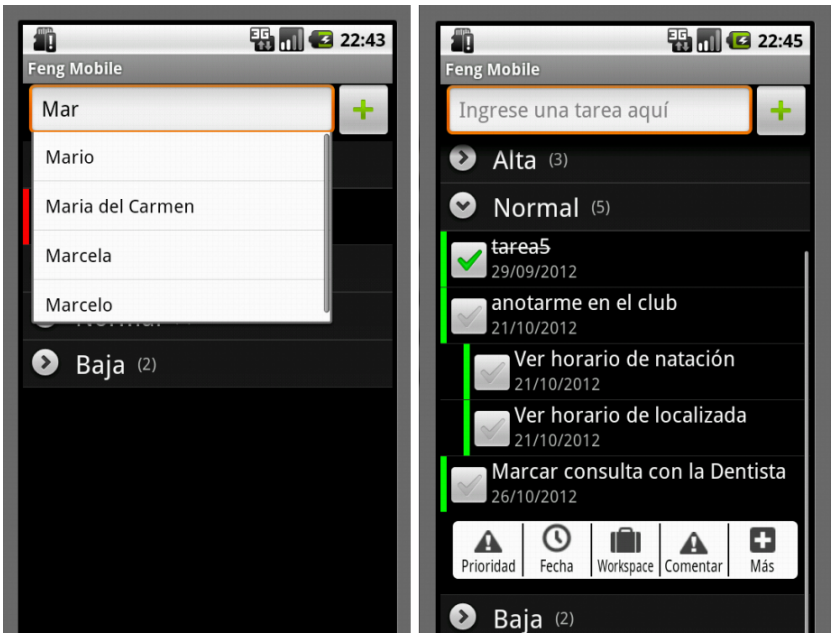

Figura 6.1.: Ingresar Tarea

Modificar tarea

1. Seleccionar una tarea.
2. Seleccionar del menú la opción “Más”.
3. Modificar datos de la tarea.

Figura 6.2.: Modificar Tarea

Comentar tarea

1. Seleccionar una tarea.
2. Seleccionar del menú la opción “Comentar”.
3. Ingresar comentario.

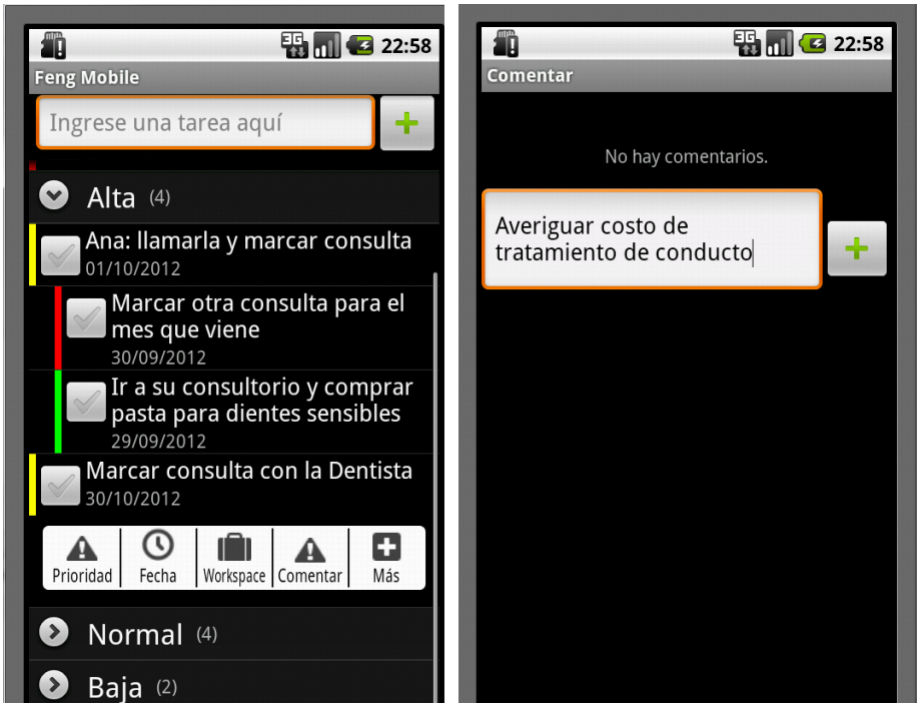

Figura 6.3.: Comentar Tarea

Modificar prioridad

1. Seleccionar una tarea.
2. Seleccionar del menú la opción “Prioridad”.
3. Seleccionar la Prioridad deseada.

Figura 6.4.: Modificar Prioridad

Modificar fecha límite

1. Seleccionar una tarea.
2. Seleccionar del menú la opción “Fecha”.
3. Modificar la fecha límite.

Figura 6.5.: Modificar Fecha Límite

Modificar workspace

1. Seleccionar una tarea.
2. Seleccionar del menú la opción “Workspace”.
3. Modificar Workspace.

Ingresar subtarea

1. Seleccionar una tarea.
2. Seleccionar del menú la opción “Más”.
3. Seleccionar la opción “Editar subtareas”.
4. Ingresar Subtarea.

Figura 6.6.: Ingresar Subtarea

Borrar tarea

1. Seleccionar una tarea.
2. Seleccionar del menú la opción “Borrar”.

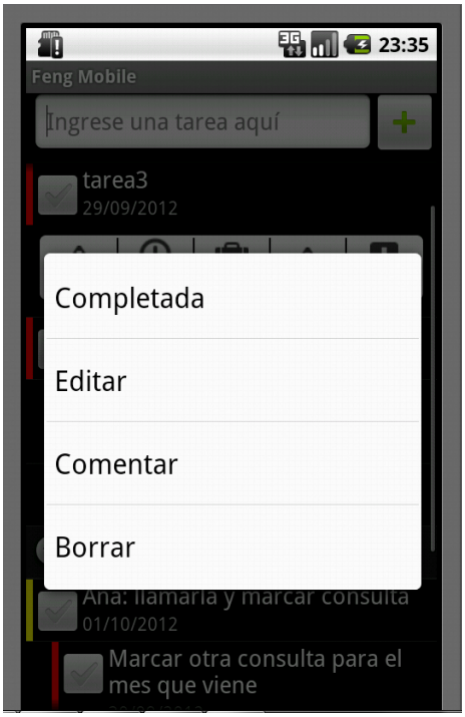

Figura 6.7.: Borrar Tarea

Completar tarea

1. Seleccionar una tarea.

2. Seleccionar del menú la opción “Completada”.

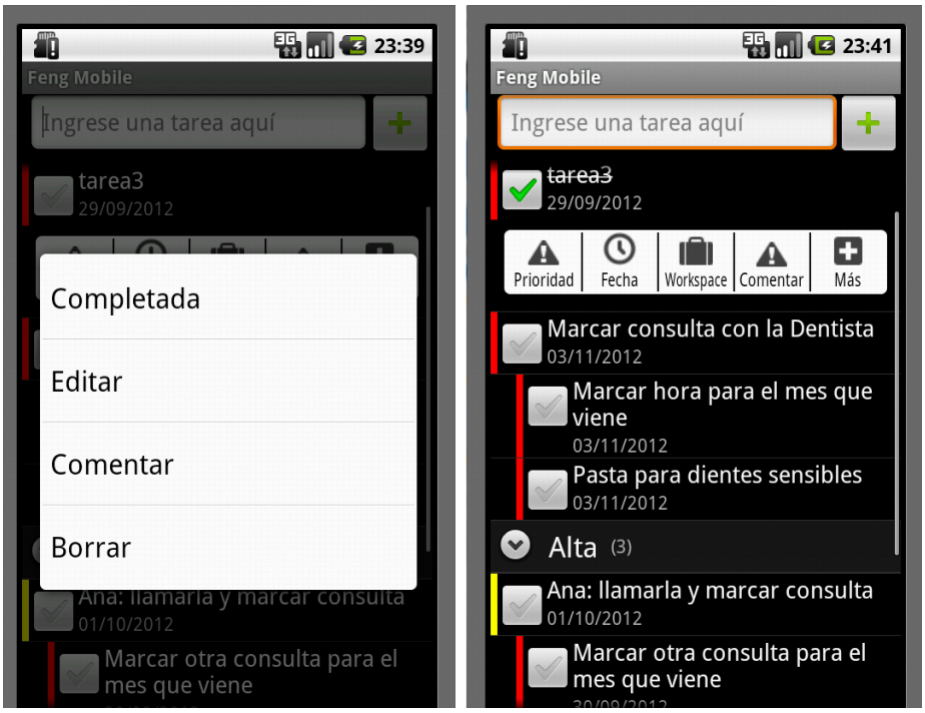

Figura 6.8.: Completar Tarea

Ordenar tareas por prioridad

1. Seleccionar del menú la opción “Ordenar”.
2. Seleccionar la opción “Prioridad”.

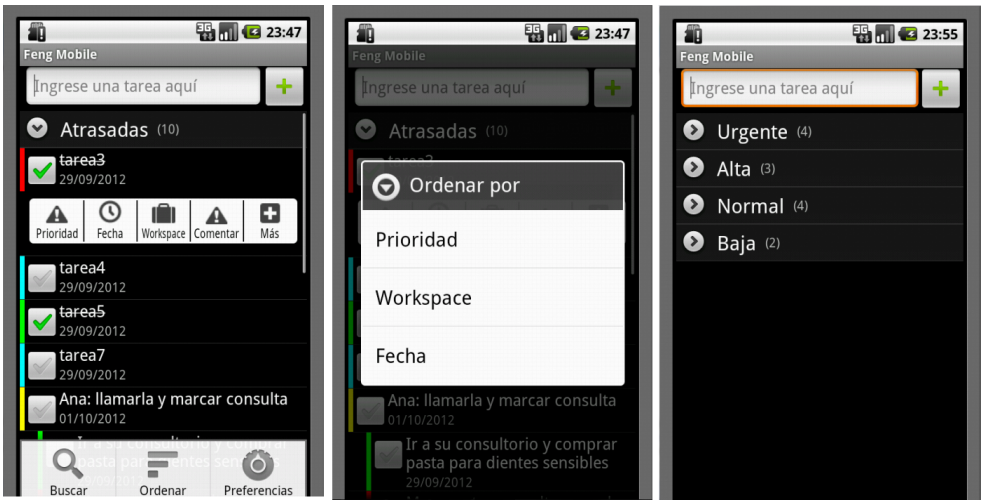

Figura 6.9.: Ordenar por Prioridad

Ordenar por fecha

1. Seleccionar del menú la opción “Ordenar”.
2. Seleccionar la opción “Fecha”.

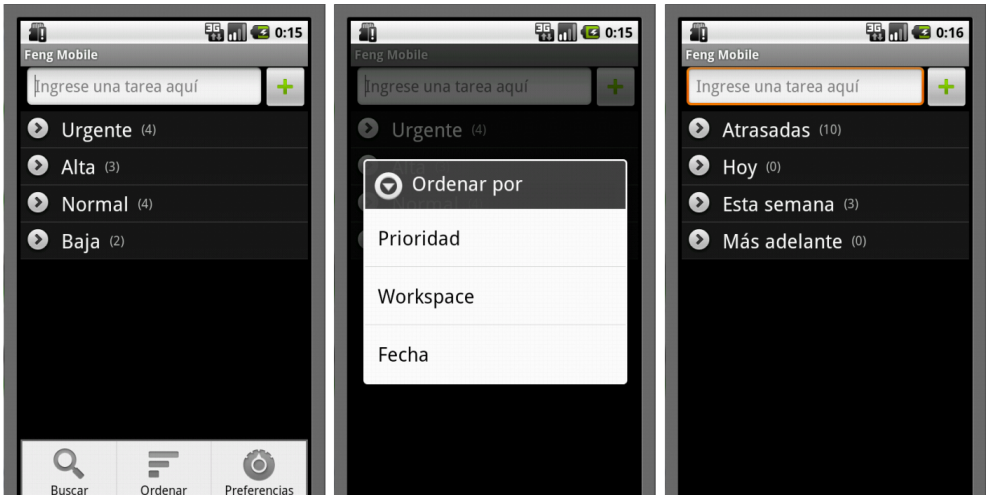

Figura 6.10.: Ordenar por Fecha

Ordenar por workspace

1. Seleccionar del menú la opción “Ordenar”.
2. Seleccionar la opción “Workspace”.

Ingresar una tarea desde la web de Feng Office

1. Ingresar a la web de Feng Office, loguearse.
2. Dar de alta una tarea.
3. Chequear que en el dispositivo llegue la notificación de que la tarea es nueva.
4. Identificar en el dispositivo si se despliega en el listado de tareas.

Ingresar una subtarea desde la web de Feng Office

1. Ingresar a la web de Feng Office, loguearse.
2. Seleccionar la tarea.
3. Ingresar una subtarea.
4. Chequear que en el dispositivo llega la notificación y si se despliega en el listado de tareas como subtarea.

Modificar el padre de una tarea desde la web de Feng Office

1. Ingresar a la web de Feng Office, loguearse.

2. Seleccionar la tarea.
3. Eliminar el padre de la tarea y asignar otra tarea como padre.
4. Chequear que en el dispositivo en el listado de tareas, se actualice el padre de la tarea.

6.2. Resultados obtenidos

A continuación se detallan algunos resultados obtenidos:

- Se detectó que la primera vez que se ejecuta la sincronización, si en la web habían tareas dadas de alta se generaban demasiadas notificaciones.
- Se tuvo un buen tiempo de respuesta al momento de sincronizar la información.
- No se observaron dificultades al utilizar la aplicación Feng Mobile en el dispositivo.
- Observando la forma en como los usuarios se desenvolvieron utilizando la aplicación móvil, podemos decir que los resultados obtenidos han sido satisfactorios y evaluamos satisfactoriamente al prototipo.

7. Conclusiones y trabajo futuro

En esta sección se presentan las conclusiones de nuestro proyecto y algunos puntos que consideramos se deberían tener en cuenta para continuar con el desarrollo de esta solución.

7.1. Conclusiones

Se hizo un relevamiento de las plataformas que permiten desarrollar aplicaciones para dispositivos móviles, como resultado de esto se pudieron hacer los cuadros comparativos 2.1, 2.2, 2.3, 2.6, 2.7 y 5.1. También se hizo un estudio de algunos patrones de diseño que se aplican al desarrollar para estos dispositivos. Se construyó un prototipo que permite acceder a funcionalidades del módulo de tareas de Feng Office. Por lo anterior, consideramos que se cumplieron los objetivos planteados para el proyecto.

Hecho el estudio de las tecnologías disponibles para el desarrollo de la aplicación se pudo apreciar la importancia de la selección de una plataforma. Una elección incorrecta puede llevar a que la aplicación no alcance la cantidad de usuarios esperada, o que en poco tiempo quede obsoleta por la tecnología utilizada. No hay indicios claros de que se esté en vías de definir un estándar, más bien cada fabricante trata de fomentar el desarrollo de aplicaciones nativas para su propia plataforma. Se vio a HTML5 como una buena alternativa, que si bien aún no es adoptada completamente, como a medida que pasa el tiempo los distintos dispositivos van soportando cada vez más funcionalidades especificadas por esta tecnología parece ser una buena opción a futuro.

El relevamiento hecho en relación al diseño de interacción para dispositivos móviles nos permitió familiarizarnos con el tema. Si bien no se profundizó en esta área, fue de utilidad para comprender los principales problemas y plantearnos inquietudes para continuar investigando.

La construcción del prototipo nos permitió adquirir experiencia en el desarrollo de aplicaciones nativas para Android. Se implementó la sincronización de tareas entre cliente y servidor, por lo que tuvimos que hacer modificaciones en el código de la aplicación web de Feng Office. El algoritmo utilizado es extensible y se puede reutilizar para sincronizar otro tipo de objetos manejados por el resto de los módulos de Feng Office. Creemos que el relevamiento hecho, la solución diseñada y el prototipo

desarrollado en este proyecto van a ser de utilidad para el desarrollo de una solución completa para brindar a los usuarios finales.

7.2. Trabajo futuro

Soporte para el resto de los módulos

Agregar soporte para el resto de los módulos como ser notas, contactos, calendarios y documentos extendiendo la interfaz de usuario y la sincronización. Para extender la interfaz de usuario se plantea dividir la aplicación en diferentes actividades para cada módulo. Se propone utilizar pestañas para seleccionar el módulo sobre el cual se quiere trabajar. Hay que incorporar al esquema de base de datos las tablas necesarias para soportar los objetos manejados por los nuevos módulos. Se plantea agregar para cada nuevo módulo un ContentProvider en los casos que corresponda y configurar la aplicación para poder habilitar y deshabilitar la sincronización por módulo en la pantalla que se muestra en la figura 5.2.

Para sincronizar otros objetos además de las tareas es necesario hacer algunas modificaciones a lo que se desarrolló para el prototipo. Principalmente hay que modificar la lógica de la API del lado del servidor para poder obtener los datos que se quieran utilizar de los distintos tipos de objetos. Se propone modificar el modelo de Feng Office para que la jerarquía de objetos tenga métodos que permitan obtener los datos necesarios para la sincronización. Podría ser de utilidad que la revisión esté representada como un atributo de los objetos del servidor específico para ese fin y no utilizar la fecha de modificado.

Resolución de conflictos en la sincronización

Cuando un usuario hace cambios en un objeto desde la aplicación cliente y el mismo objeto está modificado del lado del servidor va a haber un conflicto en la sincronización y hay que resolverlo. La estrategia que se tomó en el prototipo es que la última modificación es la que permanece.

Es necesario implementar un mecanismo de resolución de conflictos. Una posibilidad es trasladar la decisión al usuario y solicitarle la acción a tomar (Ej.: traer la versión del servidor, subir de todos modos o corregir). Esto puede ser engorroso para el usuario. Otra opción sería permitir configurar la estrategia a tomar y que se resuelva el conflicto automáticamente (Ej.: lo que está en el servidor tiene preferencia, duplicar el objeto manteniendo las dos versiones, tratar de resolver el conflicto de forma automática).

Requerimientos que quedaron fuera del alcance

Las siguientes funcionalidades o bien se dejaron fuera del alcance o se identificaron como necesarias durante el desarrollo de la solución:

- Desarrollo de la ayuda como se la describió en la sección 4.2.1.
- Asignación de tareas y suscriptores.
- Integración con los contactos del celular. Se planteó en un principio la idea de sincronizar los contactos de Feng Office con los contactos del móvil pero se dejó fuera del alcance.
- Configuración de preferencias de seguridad. Desarrollar una interfaz para que el usuario pueda aceptar, rechazar certificados del servidor.

Uso de prestaciones de los dispositivos

Permitir ingresar tareas utilizando reconocimiento de voz, tomar fotos con la cámara e incorporarlas al sistema, registrar la ubicación de un evento con el GPS, utilizar el acelerómetro para interactuar con la aplicación, son funcionalidades que se pueden ver en muchas aplicaciones del mercado y sería interesante que estuvieran presentes, por lo menos en una versión nativa de la aplicación.

A. Glosario

Este apéndice tiene como objetivo definir conceptos básicos mencionados a lo largo de toda la documentación.

3G Es la abreviación de tercera generación de transmisión de voz y datos a través de telefonía móvil mediante UMTS (Servicio Universal de Telecomunicaciones Móviles).

Acelerómetro Es un sensor que mide constantemente la posición relativa del dispositivo móvil.

AJAX Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas.

Android Sistema Operativo desarrollado por Google.

API Application Programming Interface. Es una interfaz por la cual se expone un conjunto de funcionalidades a ser consumidas por otro sistema.

Appcelerator Titanium Plataforma para desarrollar aplicaciones móviles y de escritorio utilizando tecnologías web, desarrollada por Appcelerator Inc.

Basecamp Herramienta web para la gestión de proyectos desarrollada por la empresa 37signals.

BlackBerry OS Sistema Operativo móvil desarrollado por la compañía Research in Motion.

CSS Cascading Style Sheets, lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML.

CSS3 Cascading Style Sheets Level 3, ofrece gran variedad de opciones (redondeado, sombreado, funciones avanzadas de movimiento y transformación) importantes para las necesidades del diseño web actual.

DOM Document Object Model. es una interfaz de programación de aplicaciones (API) que proporciona un conjunto estándar de objetos para representar documentos HTML y XML.

Genexus Herramienta de desarrollo de software ágil, multiplataforma, basada en conocimiento, orientada principalmente a aplicaciones web empresariales, plataformas Windows y smartphones.

Google Apps Servicio que ofrece Google que incluye las siguientes herramientas: correo con GMail, calendario, almacenamiento en la nube con Google Drive,

edición de documentos y creación de sitios entre otros. Cuenta con una versión web para móviles.

GPRS General Packet Radio Services.

GPS Global Positioning System.

HTML HyperText Markup Language, lenguaje básico de la World Wide Web.

HTML5 HyperText Markup Language Version 5, es la quinta versión del lenguaje básico de la World Wide Web, HTML.

HTTP Hypertext Transfer Protocol. Es el protocolo que se usa para acceder a la Web.

HTTPS HyperText Transfer Protocol Secure.

IDE Integrated Development Environment.

iOS iPhone Operating System, Sistema Operativo móvil desarrollado por Apple.

J2ME Java 2 Micro Edition es una versión de máquina virtual de Java pensada para correr en dispositivos con hardware y recursos limitados.

JavaScript Lenguaje de programación interpretado, orientado a objetos, basado en prototipos, imperativo y dinámico.

Jo Framework Framework para desarrollar aplicaciones web en HTML5 para dispositivos móviles.

JQTouch Plugin para jQuery específicamente diseñado para navegadores móviles basados en WebKit.

JQuery Biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

JQuery Mobile Framework JavaScript para creación de sitios webs optimizados, para los dispositivos móviles más populares.

JSONP JavaScript Object Notation with Padding, formato ligero para el intercambio de datos, subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

MIDP Mobile Information Device Profile.

Milestone Hito se utiliza para resaltar momentos importantes del proyecto, y de esta manera, ayuda a mejorar el control del proyecto.

MMA Mobile Marketing Association.

Mobincube Software web desarrollado por la compañía Mobimento, que permite crear aplicaciones para dispositivos móviles.

MotherApp Plataforma que permite generar aplicaciones nativas para Android, iPhone y Windows Mobile desde un lenguaje parecido a HTML llamado MotherApp HTML.

MVC Model View Controller, patrón de desarrollo de software que separa los datos de la aplicación, la interfaz de usuario y la lógica de negocio en tres componentes distintos.

Objective-C Lenguaje orientado a objetos que extiende el estándar ANSI C, provee sintaxis y características de la orientación a objetos.

Patrones de diseño Los patrones son soluciones óptimas a problemas comunes. Cuando surge un problema común junto con él, espontáneamente, surgen de la comunidad un conjunto de soluciones comunes.

PhoneGap Herramienta desarrollada por la compañía Nitobi Software, que permite desarrollar aplicaciones utilizando HTML5 y JavaScript.

PHP HyperText Preprocessor, lenguaje de código abierto adecuado para desarrollo web y que puede ser incrustado en HTML.

PocketPC O PDA (Personal Digital Assistant), es un pequeño ordenador, diseñado para ocupar el mínimo espacio y ser fácilmente transportable que ejecuta el sistema operativo Windows CE o Windows Mobile de Microsoft.

QML Es un metalenguaje de programación basado en JavaScript, pensado para el diseño de interfaz de usuario de aplicaciones que tienen entrada táctil.

QT QT es un framework sobre el que se construyen interfaces de aplicaciones de primer nivel.

RAM Random Access Memory.

Rhodes Framework de código abierto para construir aplicaciones para dispositivos móviles.

SDK Software development kit, son herramientas de desarrollo para crear aplicaciones.

Sencha Touch Framework para construir aplicaciones basadas en HTML5, CSS y JavaScript.

Smartphone Dispositivo que ofrece un conjunto de funcionalidades y capacidad de procesamiento que lo asemejan en su uso, al de una computadora portátil.

SQLite Sistema de gestión de bases de datos relacional, contenida en una pequeña biblioteca escrita en C.

SSL Secure Sockets Layer, Protocolo de Capa de Conexión Segura, protege la información que se comparte entre el servidor y el usuario en la web.

Symbian Sistema operativo con código abierto diseñado para dispositivos móviles mantenido por Nokia.

Tabletas Dispositivos que se asemejan más a una computadora portátil por tener una pantalla más grande, más capacidad de almacenamiento y gráficos de mejor calidad que los smartphones pero aún mantienen las funciones de teléfono.

Teclado QWERTY Es la distribución del teclado más común.

URL Uniform Resource Locator. Es una cadena de caracteres que se usa para asignar una dirección única a cada uno de los recursos que están disponibles en Internet.

W3C World Wide Web Consortium, consorcio internacional que produce recomendaciones para la World Wide Web.

WebKit Motor de navegación web de código abierto y además, un framework de Mac OS X que se usó para construir aplicaciones como el mencionado Safari, Dashboard, Mail y otras.

WHATWG Web HyperText Application Technology Working Group, comunidad de personas interesadas en la evolución de HTML y las tecnologías conexas.

WiFi Es una tecnología que permite a los dispositivos conectarse a una red de forma inalámbrica.

Windows Mobile Sistema Operativo móvil desarrollado por Microsoft.

Windows Phone Sistema Operativo móvil desarrollado por Microsoft.

XML Extensible Markup Language.

YQL Yahoo Query Languaje, lenguaje de sintaxis similar a SQL que Yahoo utiliza para filtrar y unificar datos de Web Services de internet. Tiene una API que nos permite realizar consultas complejas y obtener los datos en XML o JSON.

Zoho Plataforma web que integra herramientas colaborativas para la administración de empresas.

B. Patrones de diseño

Los patrones incluidos en este anexo fueron tenidos en cuenta porque se consideró que podrían ser aplicables en el sistema a desarrollar. Se los menciona mostrando su nombre y una descripción.

Vertical Stack Ordenar el contenido de la página en una columna vertical tratando de no colocar un elemento al lado de otro. Dejar que los textos se ajusten al ancho de la pantalla y permitir que se pueda hacer scroll vertical. Usar cuando se quiere que la aplicación sea accedida desde dispositivos con diferente tamaño de pantalla, también cuando pasar de una pantalla a otra es muy costoso, por ejemplo si lleva demasiado tiempo de carga.

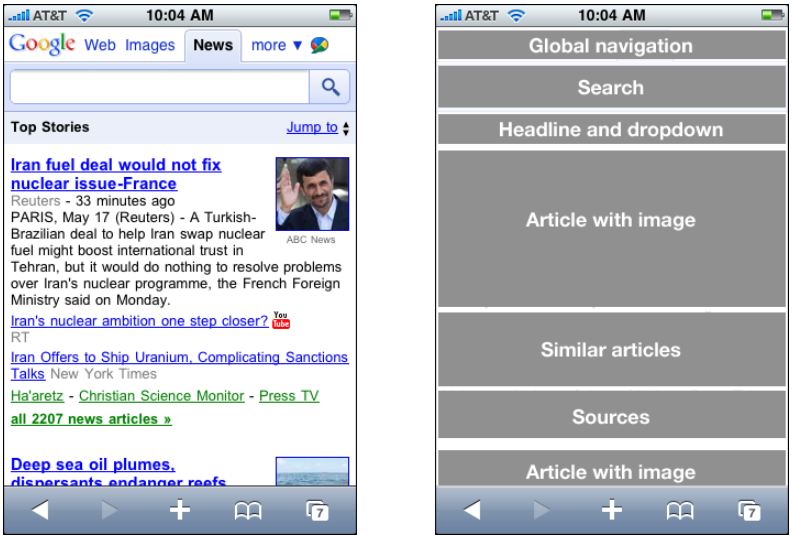

Figura B.1.: Patrón Vertical Stack[44]

Thumbnail-and-Text List Una lista de elementos seleccionables en la que cada uno contiene una imagen, un texto y posiblemente texto más chico conteniendo una descripción. Utilizar colores vistosos, íconos y otros componentes visuales si aplica. Utilizar cuando se quiere mostrar una lista de contenido complejo, cuando se quiere invitar a que el usuario acceda a ellos.

Figura B.2.: Patrón Thumbnail-and-Text List[44]

Infinit List Al final de una lista larga, poner un botón que carga y concatena más datos a la lista. Utilizar cuando se tienen que mostrar largas listas de mensajes de correos, resultados de búsquedas, etc. Utilizar cuando se tienen que mostrar largas listas de mensajes de correos, resultados de búsquedas, etc. Los usuarios suelen encontrar lo que desean en la parte superior de la lista, pero a veces es necesario buscar por más resultados.

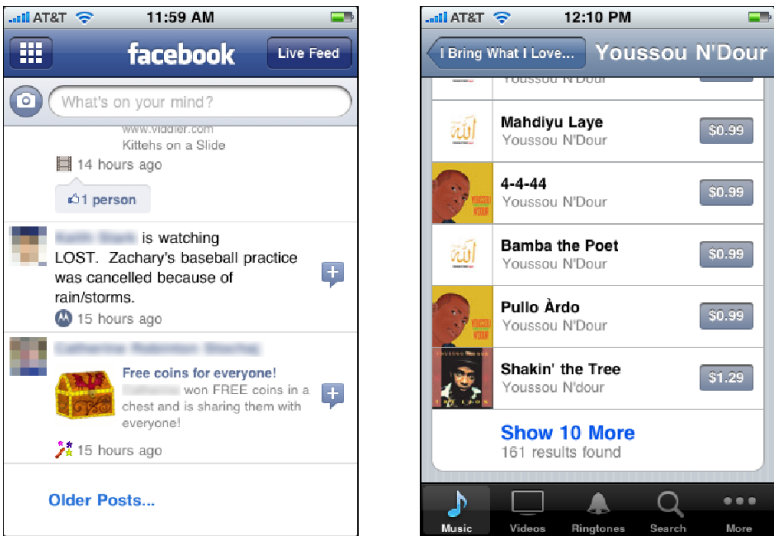

Figura B.3.: Patrón Infinit List [44]

Generous Border En dispositivos móviles con pantallas táctiles, se deben usar márgenes amplios y espacios en blanco alrededor de botones, enlaces y otros contro-

les. Se usa cuando se necesita usar botones con etiquetas, listas de items o enlaces.

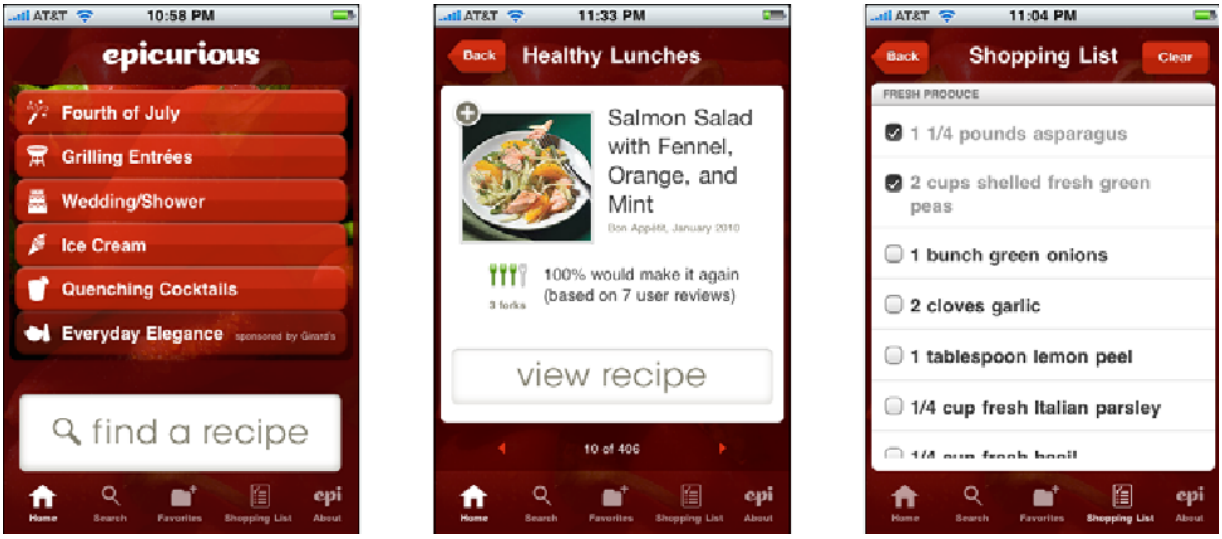

Figura B.4.: Patrón Generous Border[44]

Text Clear Button Limpia un campo de texto al hacer clic en un botón. Se usan cuando en la interfaz del móvil existe un campo de texto. Principalmente si el campo permite muchos caracteres.

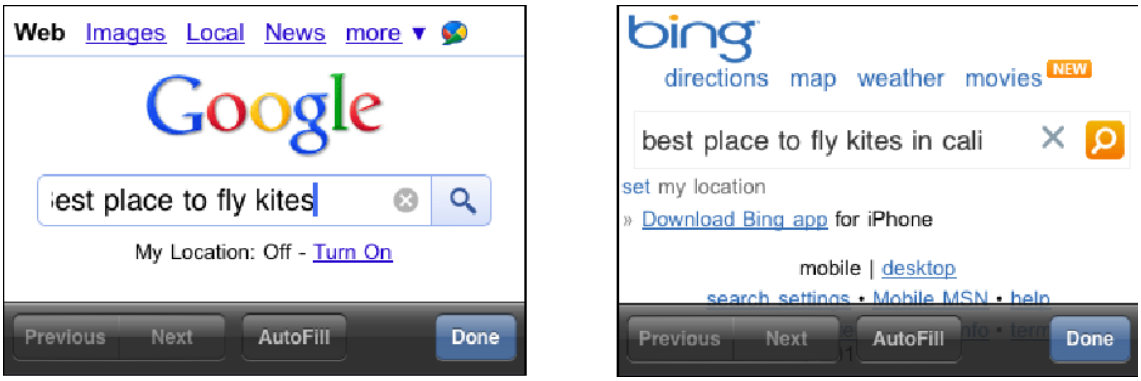

Figura B.5.: Patrón Text Clear Button[44]

C. Algoritmo de sincronización

Algoritmo C.1 Pseudocódigo de sincronización para el cliente

```
lastSyncRevision :: integer //Revisión del servidor, identifica la última sincronización. Se
almacena como una variable global en un SharedPreferences.
//Algoritmo implementado por el SyncAdapter para realizar sincronización.
performSync() {
    //Obtiene los objetos creados o modificados localmente.
    cambiosLocales := obtenerCambiosLocales();
    //Envía los cambios locales al servidor junto con la revisión.
    syncResult := Servidor->performSync(cambiosLocales,lastSyncRevision);
    //Obtiene de la respuesta los cambios recibidos del servidor, estos cambios incluyen los
id del lado del servidor, de los objetos creados en el cliente.
    cambiosEnElServidor := syncResult->getCambiosEnElServidor();
    //Actualiza los datos locales con los recibidos del servidor
    aplicarCambios(cambiosEnElServidor);
    //Almacena la revisión retornada por el servidor
    lastSyncRevision:=result->getSyncRevision();
}
```

Algoritmo C.2 Pseudocódigo de sincronización para el servidor

```
headRevision :: integer; //Revisión del último cambio aplicado en el servidor. (Es la mayor de todas las revisiones de todos los objetos)
//Algoritmo implementado del lado del servidor para la sincronización.
SyncResult :: performSync(cambiosEnElCliente, clientLastSyncRevision){
   objetosModificados :: new Array;
   foreach cambio in cambiosEnElCliente do {
       //Obtengo el objeto, enviado por el móvil de la base del servidor para aplicarle los cambios.
       objetoAModificar := obtenerObjetoCambiado(cambio);
       if(objetoAModificar==null || objetoAModificar->getRevision() <= clientLastSyncRevision){
           // El objeto no fue modificado en una revisión posterior a la indicada por el cliente, entonces se puede aplicar el cambio.
           //Modifica el objeto y aplica los cambios en la base
           aplicarCambio(objetoAModificar, cambio);
           //Se agrega el objeto a la lista de objetos modificados
           objetosModificados.add(objetoAModificar);
       } else{
               //El objeto cambió en una revisión posterior a la enviada por el cliente entonces hay un conflicto.
               //Lo ignoramos, lo que haya en el servidor tiene preferencia.
       }
   }
    cambiosAEnviar := new SyncResult
   //Los cambios a retornar son los que ocurrieron después de la revisión indicada por el cliente.
   cambiosAEnviar.cambiosEnElServidor :=
       obtenerObjetosAPartirDe(clientLastSyncRevision);
   //Se incrementa la revisión, a los objetos que fueron modificados se les asocia el nuevo valor de revisión.
   if(objetosModificados.size>0){
       headRevision ++;
       foreach objeto in objetosModificados do{
       objeto.setRevision(headRevision);
       }
   }
   // Se retorna el valor
   cambiosAEnviar.syncRevision := headRevision;
   return cambiosAEnviar.
}
```

Bibliografía

[1] 37 Signals. http://37signals.com/, 24 November 2012.

[2] Adobe PhoneGap Build. http://build.phonegap.com/, 10 November 2012.

[3] Android. http://developer.android.com, 20 July 2012.

[4] Android Patterns. http://www.androidpatterns.com/, 24 November 2012.

[5] Appcelerator. http://www.appcelerator.com, 24 November 2012.

[6] Apple iOS. http://developer.apple.com, 24 November 2012.

[7] AppWorld BlackBerry Web Store. http://appworld.blackberry.com/webstore/?, 24 November 2012.

[8] BlackBerry Curve 9380. http://es.blackberry.com/devices/blackberrycurve9380/features.jsp, 23 June 2012.

[9] BlackBerry Developers. http://us.blackberry.com/developers/, 24 November 2012.

[10] BlackBerry Playbook. http://us.blackberry.com/playbook-tablet.html, 23 June 2012.

[11] Central Desktop. http://www.centraldesktop.com/, 24 November 2012.

[12] GeoTrust SSL Trial Certificate. http://www.geotrust.com, 11 October 2012.

[13] html5test.com. http://html5test.com, 27 November 2012.

[14] iOS App Store. http://itunes.apple.com/gb/app/apple-store/id375380948?mt=8, 24 November 2012.

[15] jQTouch. http://www.jqtouch.com/, 24 November 2012.

[16] JQueryMobile. http://jquerymobile.com, 24 November 2012.

[17] MotherApp. http://www.motherapp.com/, 24 November 2012.

[18] Nokia Lumia 900. http://www.nokia.com/es-es/productos/moviles/lumia900/, 23 June 2012.

[19] PhoneGap. http://www.phonegap.com/, 24 November 2012.

[20] Pure Android. http://developer.android.com/design/patterns/pure-android.html, 25 June 2012.

[21] Rhomobile. http://www.motorola.com/Business/US-EN/Business+Product+and+Services/Software+and+Applications/RhoMobile+Suite, 24 November 2012.

[22] Samsung Galaxy S3. http://www.samsung.com/global/galaxys3/, 23 June 2012.

[23] Samsung Galaxy Tab 10.1. http://www.samsung.com/global/microsite/galaxytab/10.1/index.html, 23 June 2012.

[24] Sencha Touch. http://www.sencha.com/, 24 November 2012.

[25] Vision Mobile. http://www.visionmobile.com/product/developer-economics-2012/, 5 July 2012.

[26] Zoho. http://www.zoho.com/, 24 November 2012.

[27] Android Developers. HttpsURLConnection. http://developer.android.com/reference/javax/net/ssl/HttpsURLConnection.html, 22 October 2012.

[28] Apache Software Foundation. Apache Cordova. http://incubator.apache.org/cordova/, 10 January 2012.

[29] Apple. iPad. http://www.apple.com/ipad/, 3 July 2012.

[30] Apple. iPhone 4S. http://www.apple.com/iphone/features/, 23 June 2012.

[31] Brian Fling. *Mobile Design and Development.* O'Reilly Media Inc., August 2009.

[32] Bryan Rieger. Effective Design for Multiple Screen Sizes. http://mobiforge.com/designing/story/effective-design-multiple-screen-sizes, 19 June 2012.

[33] Infographic by GO-Gulf.com Web Design Company. Smartphone Users around the world - Statistics and facts. http://www.go-gulf.com/blog/smartphone, 24 November 2012.

[34] Daniel A. Tauber. What's J2ME? http://onjava.com/pub/a/onjava/2001/03/08/J2ME.html, 3 August 2011.

[35] doc.orable.com/javase. keytool - Key and Certificate Management Tool. http://docs.oracle.com/javase/6/docs/technotes/tools/windows/keytool.html, 11 October 2012.

[36] Feng Office Team. Feng Office Mobile! http://forums.fengoffice.com/index.php?action=printpage;topic=14164.0, 7 July 2011.

[37] The Apache Software Foundation. Apache Module mod_ssl. http://httpd.apache.org/docs/2.2/mod/mod_ssl.html, 11 October 2012.

[38] Google. Google Apps. http://www.google.com/intl/es/mobile/, 24 November 2012.

[39] IDC. IDC's Worldwide Quarterly Mobile Phone Trackers. http://www.idc.com/getdoc.jsp?containerId=prUS23299912, 6 February 2012.

[40] why has Java ME overtaken it? If Android is so hot. Philip Elmer-DeWitt. http://tech.fortune.cnn.com/2012/01/01/if-android-is-so-hot-why-has-java-me-overtaken-it/, 7 November 2012.

[41] Java. Java2ME. http://www.oracle.com/technetwork/java/javame/index.html, 23 June 2012.

[42] Java-MIDP. MIDP. http://www.oracle.com/us/technologies/java/midp-139954.html, 24 November 2012.

[43] JCAnez. El Windows Phone no es lo mismo que el Windows Mobile. http://conversaciones.nokia.com/2012/01/12/el-windows-phone-no-es-lo-mismo-que-el-windows-mobile/, 12 January 2012.

[44] Jenifer Tidwell. *Designign Interfaces*. O'Reilly, 2 edition, 2011.

[45] joapp.com. Jo - The simple app framework for HTML5. http://joapp.com/, 14 November 2012.

[46] Jorge Franganillo. Html5: el nuevo estándar básico de la Web. In *Anuario ThinkEPI*, 6 September 2010.

[47] jquery.com. jQuery: The Write Less, Do More, JavaScript Library. http://jquery.com/, 14 November 2012.

[48] json.org. Introducción a JSON. http://www.json.org/json-es.html, 28 October 2012.

[49] Mark Pilgrim. Dive Into HTML5. http://diveintohtml5.info/storage.html, 13 November 2012.

[50] Microsoft. Windows Mobile. http://msdn.microsoft.com/en-us/windowsmobile/bb264318.aspx, 24 November 2012.

[51] Microsoft. Windows Phone. https://dev.windowsphone.com//en-us, 24 November 2012.

[52] Microsoft. Windows Phone Market Place. http://www.windowsphone.com/es-es/store, 24 November 2012.

[53] mobincube.com. Mobincube website. http://www.mobincube.com/es/, 10 November 2012.

[54] Motorola. Motorola XOOM. http://www.motorola.com, 27 June 2012.

[55] Javier Naranjo Sanhermelando. Entornos de desarrollo multiplataforma para clientes móviles de aprendizaje ubicuo. Master's thesis, Universidad Politécnica de Valencia, September 2011.

[56] Yahoo Developer Network. About the Pattern Library - Design Pattern Library. http://developer.yahoo.com/ypatterns/about/, 24 November 2012.

[57] node.js. Node Js website. http://nodejs.org/, 12 October 2012.

[58] Nokia. Nokia Developers. http://www.developer.nokia.com/, 24 November 2012.

[59] Nokia. Ovi - Symbian App Store. http://store.ovi.com/, 24 November 2012.

[60] Feng Office. Feng Office API Developer's Guide - Version 1. http://www.fengoffice.com/web/wiki/doku.php/api_documentation, 13 October 2012.

[61] patternry.com. Dynamic Input Hints. http://patternry.com/p=dynamic-input-hints/, 21 May 2012.

[62] OurMobile Planet. Smartphones en América Latina. http://www.thinkwithgoogle.com/mobileplanet/es/, 24 November 2012.

[63] quantcast. May Mobile OS Share - North America. http://www.quantcast.com/inside-quantcast/quantcast/2010/06/may-mobile-os-share-north-america.html, 14 June 2012.

[64] 37 Signals. 37 Signals Basecamp. http://basecamp.com/, 24 November 2012.

[65] Sitio web de Genexus. Genexus for Smart Devices. http://www.genexus.com/sd, 10 November 2012.

[66] Symantec. Symantec SSL Certificates. https://www.symantec.com/theme.jsp?themeid=verisign-ssl-certificates, 11 October 2012.

[67] The World Wide Web Consortium. About W3C. http://www.w3.org/Consortium/, 11 November 2012.

[68] Theresa Neil. *Mobile Design Pattern Gallery.* O'Reilly Media, Inc, 1 March 2012.

[69] ui patterns.com. Inline Help Box. http://ui-patterns.com/patterns/InlineHelpBox, 21 May 2012.

[70] w3schools. HTML5 Introduction. http://www.w3schools.com/html/html5_intro.asp, 11 November 2012.

[71] WHATWG. Web Hypertext Application Technology Working Group. http://www.whatwg.org/, 11 November 2012.

[72] www.akadia.com. How to create a self-signed SSL Certificate. http://www.akadia.com/services/ssh_test_certificate.html, 11 October 2012.

[73] Yahoo! Developer Network. AutoComplete. http://developer.yahoo.com/ypatterns/selection/autocomplete.html, 20 May 2012.