Proyecto de Sistemas Informáticos
Curso 2009-2010

Proyecto iCartelera

Alumnos del grupo:
José Ramón Arranz Sanz
Julián del Campo Montejo
Héctor Pierna Sánchez

Directores del proyecto:
Jose Ignacio Gómez Pérez
Christian Tenllado van der Reijden

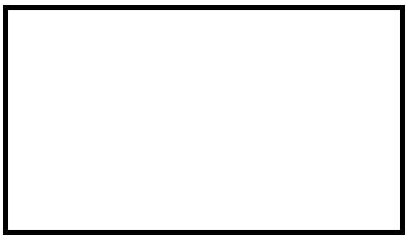
Facultad de Informática
Universidad Complutense de Madrid

Autorización

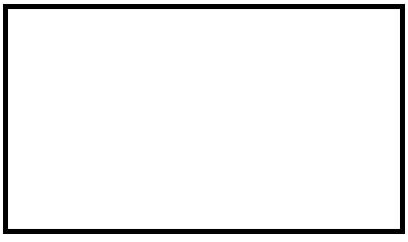
Autorizamos a la Universidad Complutense de Madrid a difundir y utilizar con fines académicos, no comerciales, y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación y/o el prototipo desarrollado.

José Ramón Arranz Sanz

Julián del Campo Montejo



Héctor Pierna Sánchez



A nuestras familias, novias, amigos, en definitiva,
a toda la gente que nos ha hecho ser tal como somos
y por lo tanto conseguir llevar a buen término
este proyecto.

Resumen

Vivimos en una Sociedad de la Información, la cual se involucra en mayor o menor medida en muchos de los aspectos de nuestro día a día. Uno de ellos es el ocio y, más concretamente, el cine.

A menudo tenemos la intención de obtener diferente información que nos ayude a elegir o descartar una película, o elegir un cine cercano donde visionarla, con la consecuente búsqueda de calle, ruta...

Por otro lado, paralelamente, el auge de los dispositivos móviles es cada vez más vertiginoso y a su vez es uno de los sistemas más empleados en nuestro tiempo libre, ya sea para comunicarnos o navegar por Internet.

El principal objetivo de iCartelera es aunar toda ese información, y proporcionarla de una manera independiente de dispositivo, pero a su vez enfocada a los SmartPhones actuales para aprovechar su potencia y sus diversas ventajas.

En el prototipo presentado, se vislumbra una arquitectura cliente-servidor centrada en iPhone, que recoge desde críticas de la película deseada y su trailer en castellano, hasta rutas personalizadas para llegar al cine deseado desde el lugar de la petición, acorde a un rango y una disponibilidad de cartelera dados.

Palabras clave: iPhone, Cartelera, Cine, Pelicula, GPS, SmartPhones, ObjectiveC, GoogleMaps.

Abstract

We are all living in an Information Society, which gets involved in a greater or lesser extent in many aspects of our day to day. One of them is entertainment and, more specifically, films and cinemas.

We often intend to get different information in order to help us to elect or dismiss a movie, or to choose a nearby cinema where we can watch it, with the consequent searching for the street, itinerary...

On the other hand, in parallel, the rise of mobile devices is increasingly rapid and in fact is one of the most used systems in our spare time, either to communicate between us or to surf the Net.

iCartelera's main goal is to combine all this information, and provide it in a device independent way, but also focused on the current smartphones, looking forward to benefit from their computational performance and their various advantages.

In the presented prototype, we can see a client-server architecture centered on iPhone, which manages to show from reviews of the desired movie and its trailer in spanish, to customized paths to reach the desired cinema from the place of the request, according to a rank and a billboard availability given.

Key words: iPhone, Billboard, Cinema, Film, GPS, SmartPhones, ObjectiveC, GoogleMaps.

Índice general

Capítulo 1

Introducción

En la sociedad actual los terminales de telecomunicaciones son cada vez más complejos y proporcionan cada vez más servicios a los usuarios. Esta creciente demanda de prestaciones implica una competitividad entre los diferentes fabricantes de móviles y una puja para ver quién puede ofertar una mayor versatilidad con sus productos.

En este aspecto, se ha extendedido a nivel mundial el término teléfonos móviles 3G. Pero, ¿qué significa exactamente tener un móvil 3G a tu disposición? Para empezar, hay que aclarar que el significado de 3G es 3^a Generación y por ello incluye diferentes servicios que en las generaciones previas no existían. Para entender el significado de 3^a Generación es necesario conocer los pasos que se dieron para su implantación a nivel mundial.

Los servicios asociados con la tercera generación proporcionan la posibilidad de transferir tanto voz y datos (una llamada telefónica) y datos no-voz (como la descarga de programas, intercambio de email, y mensajería instantánea).

Inicialmente la instalación de redes 3G fue demasiado lenta. Esto se debió a que los operadores requieren adquirir una licencia adicional para un espectro de frecuencias diferente al que era utilizado por las tecnologías anteriores 2G. El primer país en implementar una red comercial 3G a gran escala fue Japón. En la actualidad, existen 164 redes comerciales en 73 países usando la tecnología WCDMA.

Estas diferencias supusieron un gran problema para Vodafone Japón cuando su sucursal británica quiso que la subsidiaria japonesa usara sus teléfonos estándar. Los consumidores japoneses estaban acostumbrados a teléfonos más pequeños y se vieron obligados a cambiar a los de estándar europeo, que eran más gruesos y considerados fuera de moda por los japoneses. Durante esta migración, Vodafone Japón perdió 6 consumidores por cada 4 que migró al 3G. Poco después, Vodafone vendió esta subsidiaria (conocida ahora como Softbank Mobile). La tendencia general de tener móviles cada vez más pequeños parece haberse pausado, tal vez incluso dado un giro, ahora que los teléfonos con pantallas grandes ofrecen un mejor uso de Internet, videos y juegos en las redes 3G de telefonia movil.

La International Telecommunication Union (ITU) definió las demandas de redes 3G con el estándar IMT-2000. Una organización llamada 3rd Generation Partnership Project (3GPP) ha continuado ese trabajo mediante la definición de un sistema móvil que cumple con dicho estándar. Este sistema se llama Universal Mobile Telecommunications System (UMTS).

A diferencia de GSM, UMTS se basa en servicios por capas. En la cima está la capa de servicios, que provee un despliegue de servicios rápido y una localización centralizada. En el medio está la capa de control, que ayuda a mejorar procedimientos y permite que la capacidad de la red sea dinámica. En la parte baja está la capa de conectividad donde cualquier tecnología de transmisión puede usarse y el tráfico de voz podrá transmitirse mediante ATM/AAL2 o IP/RTP.

Las redes 3G ofrecen mayor grado de seguridad en comparación con sus predecesoras 2G. Al permitir a la UE autentificar la red a la que se está conectando, el usuario puede asegurarse de que la red es la intencionada y no una imitación. Las redes 3G usan el cifrado por bloques KASUMI en vez del anterior cifrador de flujo A5/1. Aún así, se han identificado algunas debilidades en el código KASUMI.

Además de la infraestructura de seguridad de las redes 3G, se ofrece seguridad de un extremo al otro cuando se accede a aplicaciones framework como IMS, aunque esto no es algo que sólo se haga en el 3G.

1.1. Evolución al 3G

El primer gran paso en la evolución al 2G ocurrió con la entrada del Servicio General de Paquetes vía Radio (GPRS - General Packet Radio Service). Los servicios de los móviles relacionados con el GPRS se conviertieron en 2.5G.

El GPRS podía dar velocidad de datos desde 56 kbit/s hasta 114 kbit/s. Puede usarse para servicios como el acceso al protocolo de aplicaciones inalámbricas (WAP - Wireless Application Protocol), servicio de mensajes cortos (SMS - Short Messaging Service), sistema de mensajería multimedia (MMS - Multimedia Messaging Service), y para servicios de comunicación por Internet como el email y el acceso a la web. La transmisión de datos GPRS es normalmente cobrada por cada megabyte transferido, mientras que la comunicación de datos vía conmutación de circuitos tradicional es facturada por minuto de tiempo de conexión, independientemente de si el usuario está realmente usando la capacidad o si está parado.

El GPRS es una gran opción para el servicio de intercambio de paquetes, al contrario que el intercambio de circuitos, donde una cierta calidad de servicio (QoS) está garantizada durante la conexión para los no usuarios de móvil. Proporciona cierta velocidad en la transferencia de datos, mediante el uso de canales no usados del acceso múltiple por división de tiempo (TDMA). Al principio se pensó en extender el GPRS para que diera cobertura a otros estándares, pero en vez de eso esas redes están convirtiéndose para usar el estándar GSM, de manera que el GSM es el único tipo de red en la que se usa GPRS. El GPRS está integrado en el lanzamiento GSM 97 y en nuevos lanzamientos. Originariamente fue estandarizado por el Instituto Europeo de Normas de Telecomunicaciones (ETSI), pero ahora lo está por el 3GPP.

Las tecnologías de 3G son la respuesta a la especificación IMT-2000 de la Unión Internacional de Telecomunicaciones. En Europa y Japón, se seleccionó el estándar UMTS (Universal Mobile Telephone System), basado en la tecnología W-CDMA. UMTS está gestionado por la organización 3GPP, también responsable de GSM, GPRS y EDGE.

En 3G también está prevista la evolución de redes 2G y 2.5G. GSM y TDMA IS-136 son reemplazadas por UMTS, las redes cdmaOne evolucionan a IS-95.

La estandarización de la evolución del 3G está funcionando tanto en 3GPP como 3GPP2. Las especificaciones correspondientes a las evoluciones del 3GPP y 3GPP2 se llaman LTE y UMB, respectivamente. Desarrollo en UMB ha sido cancelado por Qualcomm a fecha de noviembre del 2008. La evolución del 3G usa en parte tecnologías más allá del 3G para aumentar el rendimiento y para conseguir una migración sin problemas.

Hay 7 caminos diferentes para pasar de 2G a 3G. En Europa el camino principal comienza en GSM cuando se añade GPRS a un sistema. De ahí en adelante es posible ir a un sistema UMTS. En Norteamérica la evolución de sistema comenzará desde el Time division multiple access (TDMA), cambiará a Enhanced Data Rates for GSM Evolution (EDGE) y después a UMTS.

En Japón, se utilizan dos estándares 3G: W-CDMA usado por NTT DoCoMo (FOMA, compatible con UMTS) y SoftBank Mobile (UMTS), y CDMA2000, usados por KDDI. La transición por razones de mercado al 3G se completó en Japón durante el 2006.

La primera introducción de la tecnología 3G en el Caribe (2008) se hizo por América Móvil que era anteriormente MIPHONE en Jamaica. La fase de implementación de esta red fue llevada a cabo por Huawei en conjunto con otras subcontratadas como TSF de Canadá.

Con todo lo mencionado anteriormente se puede deducir que el paso de la tecnología 2G a la 3G en sistemas móviles ha supuesto un gran avance tecnológico tanto a nivel de infraestructura como a nivel de software en el mundo de las telecomunicaciones.

1.2. Comparativa entre diferentes móviles

El primer aspecto a tener en cuenta de cara al desarrollo de la aplicación fue elegir una plataforma para diseñar e implementar la misma. Para ello fue necesario realizar una comparativa entre las características de los sistemas

operativos de los móviles de última generación de las principales compañías del mercado. Se valoraron los siguientes rasgos de cada móvil:

- Aspectos básicos
- Interfaz de usuario
- SDK para el desarrollo de aplicaciones
- Tipo de usuario final mayoritario del móvil

Los sistemas operativos elegidos para la comparativa fueros los siguientes:

- Android
- BlackBerry OS 4.7
- iPhone OS 3.0
- S60 5th Edition
- Palm Web OS
- Windows Mobile 6.5

A continuación se presenta un resumen para cada caso de estudio definido anteriormente.

1.2.1. Aspectos básicos

El Kernel de un sistema operativo es el núcleo del mismo, el software responsable de facilitar a los distintos programas acceso seguro al ordenador o, en forma más básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema. Por ello, es importante conocer qué núcleo utiliza cada uno de los sistemas operativos de esta comparativa.

Tanto Android como Palm están basados en Linux. BlackBerry está basado en un kernel propietario. iPhone se basa en OS X que es una variante de

UNIX. S60 se basa en Symbian y Windows Mobile en Windows CE. La principal diferencia entre un kernel de libre distribución y uno propietario radica en que los de libre distribución como Linux cuentan con una amplia y experimentada comunidad de desarrolladores, gracias a los cuales se detectan rápidamente agujeros de seguridad, fallos, etc. y se realizan mejoras tanto para solucionar estos problemas como para adaptarse a los nuevos tiempos. En los sistemas cerrados o propietarios, es más costoso encontrar errores y mejorarlos, ya que deben ser los propios desarrolladores del sistema los que detecten y realicen las mejoras, por lo que deben dedicarse más recursos a investigación en estos sistemas, con el consiguiente aumento del coste del mismo.

Otro aspecto importante relacionado con el anterior es la adaptabilidad de la plataforma, la capacidad o facilidad para poder adaptarlo a diferentes terminales o en diferentes máquinas. En este sentido, Android es el que mayor adaptabilidad presenta, ya que como se comenta en este artículo, cada vez se está empleando en más dispositivos, no sólo teléfonos móviles, sino también en netbooks y como sistema empotrado. En cambio el resto de sistemas operativos tienen una adaptabilidad algo menor y más complicada.

También mencionaremos la conectividad de los sistemas, ya que hoy en día, para poder sacar el máximo partido a todas las funcionalidades que ofrecen cada uno de ellos es indispensable contar con acceso a Internet. En este sentido, se valora enormemente el hecho de que cuenten con acceso WiFi a Internet, así como conectividad 3G que permitan conectarse a Internet desde cualquier lugar.

	Android	**BlackBerry OS 4.7**	**iPhone OS 3.0**	**S60 5th Edition**	**Palm WebOS**	**Windows Mobile 6.5**
Kernel	Linux con máquina virtual Dalvik	Propietario	OS X	Symbian	Linux	Windows CE
Conectividad	3G, WiFi, GSM, GPRS	3G, GSM, CDMA, WiFi	3G, GSM, WiFi	3G. GSM, WiFi	3G, GSM, CDMA, WiFi	3G, GSM, CDMA, WiFi

Figura 1.1: Aspectos básicos

1.2.2. Interfaz de usuario

El kernel es un aspecto importante del sistema operativo, sin embargo, el usuario lo que más aprecia es la interfaz, que al fin y al cabo es con lo que se tiene un contacto directo y en definitiva es con lo que se trabaja.

En este sentido, iPhone ha marcado estilo con sus iconos y su fácil acceso a las aplicaciones, además de la pantalla táctil que no requiere el uso de ningún puntero, ya que cuenta con gran precisión al utilizarla con los dedos. Android ha seguido esta línea y tiene una interfaz realmente sencilla, intuitiva y amigable con una precisión realmente extraordinaria. Windows Mobile, en cambio, siempre ha requerido el uso de punteros y la interfaz es completamente distinta a la de iPhone y Android. No obstante, la última versión de Windows Mobile, la 6.5, se parece mucho más a estas dos plataformas.

También es importante el grado de personalización de las interfaces de usuario. iPhone no permite cambiar ni el tamaño de las letras. Android permite personalizar completamente el escritorio, organizando los iconos incluso por carpetas. Tiene tres escritorios deslizables que también ayudan a organizar mejor nuestras aplicaciones. Además podemos crear accesos directos a aquellas que más se utilicen. Windows Mobile también permite personalizar los iconos que aparecen en el escritorio, etc. BlackBerry permite únicamente personalizar el tipo y tamaño de letra y en lo que a las aplicaciones se refiere, permite ocultar iconos de acceso directo a las aplicaciones que utilizamos menos. No obstante, no permite desinstalar ninguna de las aplicaciones instaladas de origen en el dispositivo.

Además, mencionaremos aspectos tales como el modo de recibir las notificaciones, cómo se administran los contactos, si se permite o no realizar más de una tarea al mismo tiempo, etc.

	Android	BlackBerry OS 4.7	iPhone OS 3.0	S60 5th Edition	Palm WebOS	Windows Mobile 6.5
Interfaz intuitiva	Sí	Sí	Sí	Menos	Sí	Sí
Instalación de nuevas aplicaciones	Sencilla (Android Market)	Sencilla	Sencilla (App Store)	Compleja	Sencilla	Costosa
Notificación	Bandeja	Pop-up, fondo	Pop-up	Pop-up	Bandeja	Bandeja, pop-up
Administración de contactos	Google. Posibilidad de sincronización con otros servicios.	BES, BIS	Exchange, ActiveSync, Mac OS Address Book	Exchange, Domino, BlackBerry, iSync	Synergy	Exchange, Domino, BlackBerry, ActiveSync
Multitaskin	Sí	Sí	No	Sí	Sí	Sí
Copiar/Pegar	Sí	Sí	Sí	Sí	Sí	Sí
Ecosistema/Soporte multimedia	Amazon	iTunes sin DRM	iTunes	Ovi	Amazon	Windows Media Player
Actualización del firmware	OTA	Tethered, OTA	Tethered	Tethered, OTA	¿?	Tethered, OTA
Motor navegador	Webkit	Propietario	Webkit	Webkit	Webkit	Internet Explorer
Thethering	Sí	Sí	Sí	Sí	Sí	Sí
Bluetooth estéreo	Sí	Sí	Sí	Sí	Sí	Sí

FIGURA 1.2: Interfaz de usuario

1.2.3. SDK para desarrollo de aplicaciones

Para poder incluir nuevas funcionalidades y nuevas aplicaciones, es importante que la plataforma admita desarrollo de terceros. En este sentido todas las plataformas analizadas en esta comparativa ponen a disposición de los desarrolladores el SDK que permite desarrollar aplicaciones para la plataforma en cuestión.

Todas estas aplicaciones desarrolladas por terceros deben dejarse disponibles en algún lugar de la red para que los usuarios puedan descargárselas a sus terminales. iPhone fue pionero en crear un mercado virtual, conocido como AppStore, en el que los desarrolladores colgasen sus aplicaciones y los usuarios pudiesen descargarlas. En este mercado hay ya más de 40.000 aplicaciones, tanto gratuitas como de pago. A iPhone le siguió Android con su Android Market, que en apenas 6 meses de su lanzamiento ya cuenta con más de 2300 aplicaciones. Viendo el éxito de estos dos mercados, Windows y Palm han seguido los mismos pasos y ya han lazado mercados similares. Lo mismo harán próximamente BlackBerry y Symbian.

Además de las aplicaciones de terceras personas, también es importante que la plataforma cuente con aplicaciones nativas propias de la plataforma. Tanto iPhone como Android, Symbian y Windows Mobile cuentan con aplicaciones de este tipo.

	Android	BlackBerry OS 4.7	iPhone OS 3.0	S60 5th Edition	Palm WebOS	Windows Mobile 6.5
Disponibilidad del SDK	Sí	Sí	Sí	Sí	Sí	Sí
Tienda de aplicaciones	Sí	Próximamente	Sí	Próximamente	Sí	Sí
Disponibilidad de aplicaciones	Alta	Mediana	Alta	Mediana	Baja	Alta
Aplicaciones nativas	Sí	No	Sí	Sí	No	Sí
Administración local de aplicaciones	Excelente	Buena	Excelente	Buena	Excelente	Buena

FIGURA 1.3: SDK para desarrollo de aplicaciones

1.2.4. Tipo de usuario final

Android es ante todo un avanzado sistema preparado para cualquier tipo de terreno, como entretenimiento o medio de comunicación, pero también, y cómo no, como herramienta de trabajo de profesionales que necesiten llevar a cualquier parte un dispositivo que cumpla el papel que en la actualidad pueden estar realizando ordenadores portátiles, blocs de notas, libretas, PDA's o la colección de las tarjetas de visitas de nuestros clientes o proveedores.

Android es el perfecto compañero para todos aquellos que necesiten llevar consigo siempre la lista de contactos que ofrece una larga lista de opciones y posibilidades como añadir múltiples teléfonos, direcciones, notas, imágenes, etc. y facilitar así el duro trabajo diario que supone mantener actualizada la agenda. Además, también podría utilizarse para llevar fotos y vídeos sobre sus productos.

La sincronización es constante y gratuita evitando, si es nuestro deseo, el uso de servidores Exchange o servicios de pago como MobileMe. Es decir,

cualquier cambio que realicemos en el móvil (como nuevos contactos, editar contactos existentes, añadir nueva tarea en el calendario público o laboral, etc.), serán reflejados en las aplicaciones web, y viceversa. Con esto nos ahorraremos el problema de que, hasta llegar a la oficina o a casa, tengamos que estar pendientes de sincronizar el móvil con el ordenador para que a su vez, este lo sincronice con los servicios web. Por supuesto, es totalmente compatible con una cuenta de $GoogleApps^{TM}$.

La movilidad es tan amplia como lo es la cobertura de nuestra operadora pudiendo contar con sus redes en todo momento, tengan disponibilidad 3G o GPRS. Otro aspecto que podrá solucionarnos es conocer dónde pueden encontrarse nuestros empleados o compañeros y guiarlos o ponernos en contacto con ellos gracias al servicio de $GoogleLatitude^{TM}$ o conocer con todo detalle cualquier ciudad, aunque sea desconocida para nosotros con $GoogleMaps^{TM}$ y a través de él saber los negocios de determinados sectores que puedan encontrarse más próximos a nosotros o a lo largo de toda una ciudad

Por todos es conocida la afinidad de BlackBerry con el entorno empresarial, con una infraestructura exitosa e implantada, asimilan la nueva Storm como una BlackBerry más, un mercado por otra parte donde el iPhone no se siente tan cómodo. Esta continuidad no implica que hayan descuidado sus funcionalidades multimedia, con un hardware adecuado, una pantalla propicia, incluyendo incluso la sincronización con iTunes, la BlackBerry es una seria oportunidad de unir el mundo de los negocios con las necesidades personales.

La capacidad de trabajar multitarea con eficiencia es algo que se va a empezar a valorar, muchos usuarios van a demandar la productividad que se consigue utilizando varias aplicaciones al mismo tiempo.

Si bien BlackBerry aporta una bonita interfaz de usuario, su sistema operativo sigue siendo el tradicional BlackBerry O / S. Un sistema con interesantes cualidades multitarea, como poder ejecutar aplicaciones de mensajería instantánea de fondo, mientras navegamos por Internet o redactamos un correo electrónico.

Este tampoco es el fuerte del iPhone, como mucho puedes llamar o repro-

ducir con iTunes mientras tenemos una aplicación abierta. Con BlackBerry no sólo conseguimos una productividad efectiva, sino que el sistema se vuelve más dinámico al realizar notificaciones y operaciones en tiempo real mientras realizamos otras tareas.

El gran fallo de la nueva BlackBerry es la falta de conectividad WiFi, que poco a poco se va convirtiendo en una importante red de acceso a Internet.

A pesar de la cobertura de los medios, en cuestión de búsquedas por parte de los usuarios, BlackBerry está teniendo bastante más interés que G1 (Android), que desde que ha sido presentado ha disminuido su presencia en las búsquedas.

El diseño innovador y un sistema operativo altamente intuitivo ha conseguido posicionar rápidamente al iPhone como el preferido entre los empresarios; o como mínimo esta es la conclusión a la que se llega en el estudio de J.D. Power & Asociates sobre la satisfacción de los empresarios con sus smartphones.

Se trata de un estudio en el que se evalúan los móviles en una escala del 0 al 1000 en la que utilizan 5 parámetros para valorar las virtudes de las terminales: facilidad de trabajo, sistema operativo, diseño físico, características del aparato y batería.

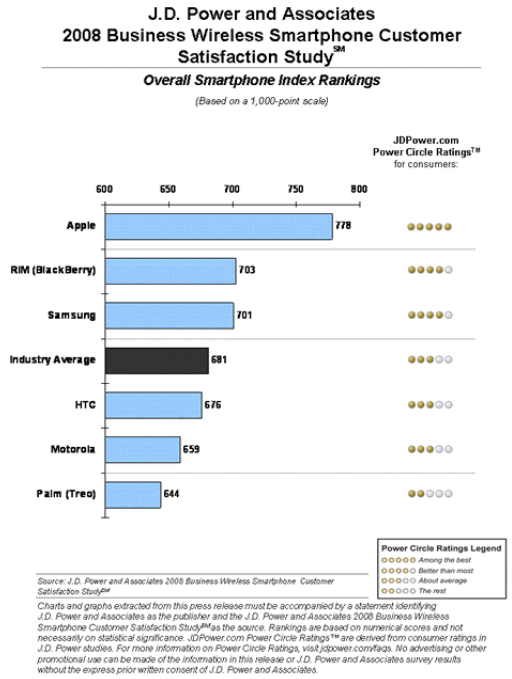


FIGURA 1.4: Comparativa entre móviles

Se trata de unos resultados muy significativos ya que hasta ahora Blackberry dominaba flagrantemente el mercado empresarial.

Entre las características que este tipo de usuarios valoran más en sus móviles encontramos: el acceso a Internet y al email, el diseño, el Bluetooth y el formato del teclado.

Oficialmente no se pueden instalar programas en el iPhone que no hayan sido firmados por Apple, para lo cuál hace falta pagar para entrar a formar parte del iPhone Developer Program (descargar el SDK, por otro lado, si es gratuito). Es posible, no obstante, desarrollar aplicaciones web para Safari o instalar aplicaciones de terceros mediante jailbreaking a través de los programas PwnageTool y WinPwn, que también permiten liberar el iPhone de primera generación.

Se trata de un buen sistema operativo, con una interfaz muy interesante, como nos tiene acostumbrados Apple, aunque es una lástima el hardware sobre el que corre el sistema, que tiene muchas carencias, y las restricciones

auto impuestas:

- Restricciones sobre el hardware en el que correrá el sistema
- Restricciones sobre el software que puede ejecutar el sistema
- Restricciones sobre las aplicaciones que se pueden ejecutar en segundo plano, o lo que es lo mismo, restricciones en la multitarea (que se pueden subsanar de nuevo con PwnageTool y WinPwn)
- Y restricciones sobre los operadores con los que poder utilizar el teléfono

Windows Mobile presentó la versión 6.5. Más de lo mismo, con algunas novedades pero en el fondo sigue siendo el Windows de siempre.

Habrá que ver si, con la versión 7, Microsoft llega con la lección aprendida, aunque siempre puede sobrevivir oculta en las interfaces más amigables creadas por los fabricantes de móviles.

Pese a que puede parecer chocante en una época de abundancia de recursos hardware como la nuestra en el sentido que los teléfonos móviles cada día disponen de un hardware más potente, el nuevo Symbian de Nokia economiza en el uso de esos recursos: memoria RAM, batería, uso del procesador... El objetivo es permitir dotar de la máxima potencia que permite el sistema incluso a los terminales más simples y con menos recursos.

Symbian 9.5 incluye un desfragmentador de memoria RAM y un optimizador para sacar el máximo partido a la memoria que tengamos instalada en nuestro dispositivo y mejorar el rendimiento de varias aplicaciones ejecutándose simultáneamente.

Una mejor sincronización con equipos de sobremesa es lo que se necesita casi en cualquier nueva versión de Symbian para batir a Pocket PC/Windows CE, su rival de Microsoft que encaja como un guante (como no podía ser de otra forma al ser del mismo fabricante) en Windows.

1.2.5. Conclusión

Teniendo todo lo expuesto anteriormente en consideración, se decidió elegir al iPhone como plataforma de desarrollo para la aplicación. Esto fue así debido a que ofrece una alta gama de posibilidades de cara al SDK que utiliza además de que el elevado número de aplicaciones existentes para el iPhone indican una gran versatilidad del mismo. Por último, también se eligió el iPhone por la disponibilidad real de un terminal propio de uno de los integrantes del grupo, lo que nos permitiría en un futuro probar la aplicación en un sistema real.

1.3. Aplicación iCartelera

En vista al estudio presentado en la sección anterior, se decidió crear una aplicación para las plataformas móviles de última generación preparada para soportar las demandas sociales de dichas aplicaciones. De esta manera, se acordó entre los miembros del grupo que dicha aplicación debía estar orientada hacia el ocio, ya que la mayoría de aplicaciones para plataformas móviles sirven para facilitar el acceso al ocio por parte del usuario.

Con fin de conseguir un sistema actual y a la vez novedoso en este campo se propuso la creación de una aplicación capaz de facilitar al usuario la localización de los cines más cercanos a su posición en los que se proyecta una película de su elección. Por una parte se garantizó que fuera un proyecto de actualidad, ya que debido a la gran repercusión y extensión del cine en el mundo se asegura que la aplicación no caiga en desuso o que se quede desfasada. Por otra parte, en cuanto a la novedad del proyecto, al realizar una búsqueda de aplicaciones similares, se encontraron bastantes aplicaciones para móviles que permitían consultar la cartelera de los cines mediante internet o mensaje de texto. Sin embargo, el paso innovador que se dio fue incorporar el GPS del terminal para poder mostrar los cines que estuvieran cercanos al usuario dentro de un rango de su elección.

Además también se planteó la posibilidad de incorporar un sistema de

reconocimiento de imágenes para que mediante una foto hecha por el usuario del cartel de una película se identificara una película para poder mostrar los cines cercanos que la proyectaran.

1.4. Memoria

La presente memoria está estructurada en los siguientes capítulos:

- Arquitectura del proyecto: En este capítulo se explican los pasos seguidos para desarrollar la arquitectura cliente-servidor empleada así como sus diferentes elementos.
- Arquitectura del servidor y BBDD: Este capítulo se centra en describir la estructura del servidor así como la de la BBDD instalada en el mismo.
- Arquitectura del cliente (iPhone): En este capítulo se explica en profundidad la estructura de la aplicación cliente instalada en el iPhone.
- Profiling: En este capítulo se explican los datos de rendimiento que hemos obtenido en las sesiones de prueba del sistema así como una comparativa entre 2 versiones del proyecto.
- Manual de usuario: En este capítulo se explica un ejemplo de caso de uso de funcionamiento con capturas de la aplicación que pueden servir como manual de usuario.

Además en las diferentes secciones habrá títulos correspondientes a las 2 versiones del proyecto: la Versión basada en el reconocimiento de un código QR para extraer el título de la película y la Versión basada en el reconocimiento real del cartel de la película.

Capítulo 2

Diseño de la aplicación

El proyecto se basa en una arquitectura Cliente-Servidor.Esta arquitectura consiste básicamente en un cliente que realiza peticiones a otro programa (el servidor) que le da respuesta. Aunque esta idea se puede aplicar a programas que se ejecutan sobre una sola computadora es más ventajosa en un sistema operativo multiusuario distribuido a través de una red de computadoras.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde el servidor no se ejecuta necesariamente sobre una sola máquina ni es necesariamente un sólo programa. Los tipos específicos de servidores incluyen los servidores web, los servidores de archivo, los servidores del correo, etc. Mientras que sus propósitos varían de unos servicios a otros, la arquitectura básica seguirá siendo la misma.

Una disposición muy común son los sistemas multicapa en los que el servidor se descompone en diferentes programas que pueden ser ejecutados por diferentes computadoras aumentando así el grado de distribución del sistema. La siguiente figura muestra un ejemplo de esta arquitectura.

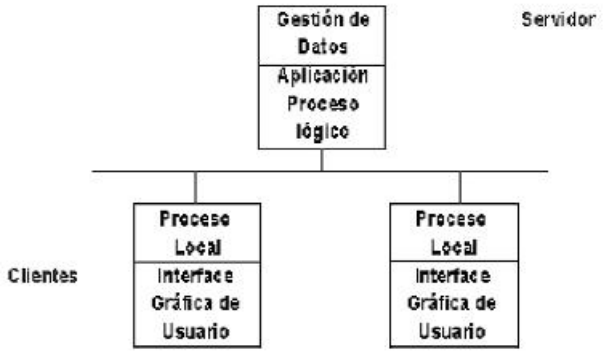


FIGURA 2.1: Arquitectura Cliente-Servidor

Así mismo, en el proyecto, el cliente será la aplicación en el terminal móvil del usuario y se encargará de transmitir al servidor los datos referentes a la fotografía del cartel, título de la película y posición GPS del usuario. En cuanto al servidor, será el equipo donde esté desplegada la parte de la aplicación servidor del proyecto y la base de datos y se encargará de recibir los datos que le envía el cliente, procesarlos y devolver al cliente la información de la película, cines, horarios, trailer y críticas para su visualización en el mismo. Por lo tanto, el proyecto está compuesto por dos aplicaciones: una que necesitará ser instalada en el terminal móvil y otra que será desplegada en una máquina servidor.

Otro dato a tener en cuenta es que en ambos extremos de la comunicación (cliente-servidor) es necesaria la conexión a internet dado que ese es el medio elegido para poder establecer el diálogo entre los extremos de la comunicación. Esto no supone ninguna dificultad añadida ya que el acceso a internet está garantizado tanto desde terminales móviles de última generación como desde equipos informáticos en la amplia mayoría de los casos.

2.1. Restricciones del diseño

Al utilizar una arquitectura Cliente-Servidor a través de internet se plantea una nueva restricción referente al tiempo de respuesta y en consecuencia de ejecución de la aplicación.

En el momento de plantear la arquitectura se tuvo en cuenta esta restricción y se han utilizado diferentes métodos para solventarla. Uno de ellos es minimizar el número de intercambios de información entre cliente y servidor ya que cuanto menor sea el número de comunicaciones entre los dos menor es el tiempo de espera por ambas partes.

Otra medida tomada para mejorar el tiempo de respuesta es distribuir con la aplicación a instalar en el terminal móvil todos los estilos y plantillas utilizados en el HTML que se presenta al usuario con los resultados de la consulta.

Por último, se ha desarrollado e incluido en la Versión Final del prototipo una interfaz con métodos nativos de la plataforma móvil escogida para la implementación que agilizan el procesamiento interno de la imagen en la misma así como su presentación en formato web.

2.2. Alternativas de diseño

Para comenzar el desarrollo de la aplicación se plantearon diversas alternativas a la hora de extraer la información para obtener los datos de la película en el terminal móvil y enviar la información resultante del procesamiento del servidor de vuelta al terminal móvil en cuanto al protocolo de comunicación.

2.2.1. QR Code

La siguiente alternativa se refiere a la obtención del título de la película por parte del terminal móvil. Un código QR (Quick Response Barcode) es un sistema para almacenar información en una matriz de puntos o un código de barras bidimensional creado por la compañía japonesa Denso-Wave en 1994; se caracterizan por los tres cuadrados que se encuentran en las esquinas y que permiten detectar la posición del código al lector. La sigla “QR” se derivó de la frase inglesa “Quick Response” pues el creador aspiraba a que el código permitiera que su contenido se leyera a alta velocidad. Los códigos QR son

muy comunes en Japón y de hecho son el código bidimensional más popular en ese país.

Aunque inicialmente se usó para registrar repuestos en el área de la fabricación de vehículos, hoy, los códigos QR se usan para administración de inventarios en una gran variedad de industrias. Recientemente, la inclusión de software que lee códigos QR en teléfonos móviles japoneses, ha permitido nuevos usos orientados al consumidor, que se manifiestan en comodidades como el dejar de tener que introducir datos de forma manual en los teléfonos. Las direcciones y los URLs se están volviendo cada vez más comunes en revistas y anuncios japoneses. El agregado de códigos QR en tarjetas de presentación también se está haciendo común, simplificando en gran medida la tarea de introducir detalles individuales de un nuevo cliente en la agenda de un teléfono móvil.

Los consumidores que cuenten con dispositivos y programas de captura, en combinación con un PC con interfaz RS-232C pueden usar un escáner para leer los datos.

El estándar japonés para códigos QR ([JIS]] X 0510) fué publicado en enero de 1999 y su correspondiente estándar internacional ISO (ISO/IEC18004) fue aprobado en junio de 2000.

Un detalle muy importante sobre el código QR es que su código es abierto y que sus derechos de patente (propiedad de Denso Wave) no son ejercidos. A continuación se expone un ejemplo de este código:

FIGURA 2.2: QR Code para “Proyecto iCartelera Sistemas Informáticos”

Este código permite almacenar datos numéricos de hasta 7.089 carácteres o datos alfanuméricos de hasta 4.296 caracteres en un espacio muy reducido. Aprovechando estas posibilidades de almacenamiento y el hecho de que es código abierto se eligió como una buena alternativa para que el título de cada película estuviera contenido en un QR Code. Tomando ventaja de la inclusión de aplicaciones que leen códigos QR en teléfonos móviles, se obtuvo una de dichas aplicaciones de código abierto para el iPhone en particular que al hacer una fotografía busca y reconoce el QR Code así como los datos almacenados y a partir de ahí se fue modificando para que se adaptara a nuestro sistema mediante ingeniería inversa.

La única desventaja de esta alternativa es que a pesar de estar muy extendido el uso del QR Code en Japón todavía no se han explotado comercialmente sus posibilidades en Europa, por lo que sería necesario incluir el código QR en los carteles de las películas para que mediante la foto al QR Code del cartel se pudiera reconocer la película en la que está interesado el usuario.

2.2.2. Reconocimiento de imágen

La siguiente alternativa se refiere a la obtención del título de la película por parte del terminal móvil. Consiste en la extracción de la información

del propio cartel de la película mediante un sistema de reconocimiento de imágenes. Para este fin, mediante las recomendaciones de los directores de proyecto se buscó información sobre las librerías OpenCV para comprobar su adaptabilidad al proyecto.

OpenCV es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Desde que apareció su primera versión alfa en el mes de enero de 1999, se ha utilizado en infinidad de aplicaciones. Desde sistemas de seguridad con detección de movimiento, hasta aplicativos de control de procesos donde se requiere reconocimiento de objetos. Esto se debe a que su publicación se da bajo licencia BSD, que permite que sea usada libremente para propósitos comerciales y de investigación con las condiciones en ella expresadas.

OpenCV es multiplataforma, Existiendo versiones para Linux, Mac OS X y Windows. Contiene mas de 500 funciones que abarcan una gran gama de areas en el proceso de Visión, como reconocimiento de objetos (reconocimiento facial), calibración de camaras, vision estereo y visión robótica.

Como metas el proyecto pretende proveer un "Tool-Kit" o Marco de desarrollo fácil de utilizar y altamente eficiente. Esto se ha logrado, realizando su programación en código c y c++ optimizados, aprovechando además las capacidades que proveen los procesadores multi nucleo. Open CV puede además utilizar el sistema las Primitivas de Rendimiento Integradas de Intel. Que es un conjunto de rutinas de bajo nivel específicas para procesadores Intel.

Esta alternativa posee un gran potencial, pero debido a que la integración de esta alternativa al proyecto supondría realizar una investigación en profundidad paralela al desarrollo del propio proyecto sobre el tratamiento de imágenes, no se llegó a implementar para la versión final del proyecto por no disponer de los recursos tanto de tiempo como de personal necesarios para llevarla a cabo. Sin embargo, está prevista su incorporación en versiones futuras de la aplicación.

2.2.3. Introducción del título por el usuario

La siguiente alternativa se refiere a la obtención del título de la película por parte del terminal móvil. Esta alternativa se propuso como sistema complementario a alguno de los anteriores en caso de que debido a una mala calidad de la fotografía no se reconociera la película o en caso de no disponer de un cartel en las cercanías para realizar dicha fotografía. Se basa en proporcional al usuario una opción del menú en la que mediante el teclado del terminal móvil pueda introducir manualmente el título de la película de la que quiere recibir la información.

Este método requiere la intervención por parte del usuario, por lo que un objetivo a conseguir es la sencillez y usabilidad máxima que el propio terminal móvil ofrece. En el caso del iPhone esta alternativa queda implementada con una interfaz sencilla e intuitiva para el usuario.

2.2.4. Envío del resultado en formato HTML

La siguiente alternativa se refiere al formato de la información que devuelve el servidor al terminal móvil. Esta alternativa se elaboró para que la aplicación fuera compatible con todos los móviles que tuvieran a su disposición un navegador web. Se basa en la creación y envío por parte del servidor de un archivo de texto con extensión HTML con los datos necesarios para su correcta interpretación por un navegador web así como la información requerida por el usuario en su consulta al servidor prepara para su visualización.

Al enviar la información en dicho formato se puede acceder a ella sin necesidad de procesamiento por parte del terminal móvil y se muestra al usuario directamente. Además en caso de que no se pudiera realizar la transferencia del archivo por problemas relativos a los permisos de acceso de un determinado modelo de terminal móvil, se podría colgar el archivo HTML generado en un servidor web y acceder mediante el navegador web del terminal móvil mediante URL a dicha página.

2.2.5. Envío del resultado en formato XML

La siguiente alternativa se refiere al formato de la información que devuelve el servidor al terminal móvil. Dado que los terminales móviles de última generación y en consecuencia sus SDK ofrecen un amplio repertorio de posibilidades, se decidió aprovechar esta característica para poder estandarizar la aplicación y hacerla compatible a la vez con un número mayor de modelos y marcas de terminales móviles.

El resultado que envía el servidor en esta alternativa es un fichero XML que contiene la información de los objetos de tipo Cines, Criticas, Trailer y los datos de la película mediante el lenguaje de marcado XML. De esta manera se dota al terminal móvil de una mayor autonomía de cara al procesamiento de los datos ya que cada modelo de terminal puede organizar la información resultado según su propio criterio.

Estructura del archivo XML

El archivo XML tiene como nombre el título de la película y extensión XML. Su contenido está distribuido mediante marcadores y no se han introducido separadores ni saltos de línea en el archivo. Para una mayor claridad de la explicación aquí se mostrará el contenido separado por bloques y con saltos de línea.

La cabecera contine los datos del formato del archivo y no se debe modificar:

<?xml version=”1.0” encoding=”UTF-8” ?><infocines>

Posteriormente es introducida la información de la película:

<PELICULA>

<TITULO>Titulo</TITULO>

<ANO>Año</ANO>

<DUR>Duracion</DUR>

<PAIS>Pais</PAIS>

<CARTEL>URL del Cartel</CARTEL>

<TRAILER>URL del Trailer</TRAILER>

<PUNT>Puntuacion de la pelicula</PUNT>

<DIRECTOR>Director</DIRECTOR>

</PELICULA>

A continuación se introduce la información de los cines siguiendo el siguiente patrón para cada uno y encerrándolos todos dentro de las etiquetas <CINES>y </CINES>:

<CINES>

<CINE>

<NAME>Nombre</NAME>

<HOR>Horarios del cine</HOR>

<DIST>Distancia al cine</DIST>

<POS>Latitud y Longitud del cine</POS>

</CINE>

...

<CINE>

...

</CINE>

</CINES>

La siguiente parte del archivo son las críticas sobre la película y de la misma manera que en los cines hay un bloque para cada una y van encerradas dentro del bloque <CRITICAS>y </CRITICAS>:

<CRITICAS>

<CRITICA>

<SITIO>Nombre del sitio</SITIO>

<AUTOR>Autor de la crítica</AUTOR>

<TEXT>Texto de la crítica</TEXT>

<URL>URL de la crítica</URL>

</CRITICA>

...

<CRITICA>

...

</CRITICA>

</CRITICAS>

Por último se añade la etiqueta de cierre de todo el archivo </infocines>. A continuación se incluye un ejemplo de archivo XML generado por la aplicación:


```
<?xml version="1.0" encoding="UTF-8" ?>
- <infocines>
  - <PELICULA>
      <TITULO>The Blind Side (Un sueño posible)</TITULO>
      <ANO>2009</ANO>
      <DUR>128</DUR>
      <PAIS>EE.UU.</PAIS>
      <CARTEL>http://www.ecartelera.com/carteles/3500/3527/001-th.jpg</CARTEL>
      <TRAILER />
      <PUNT>3stars</PUNT>
      <DIRECTOR>John Lee Hancock</DIRECTOR>
    </PELICULA>
  - <CINES>
    - <CINE>
        <NAME>Cinesa La Moraleja 3D</NAME>
        <HOR>12:15 - 16:15 - 19:15 - 22:15</HOR>
        <DIST>21779.812</DIST>
        <POS>40.5194,-3.6571</POS>
      </CINE>
    - <CINE>
        <NAME>Yelmo Cineplex Tres Aguas</NAME>
        <HOR>12:00 - 15:30 - 17:45 - 20:05 - 22:30</HOR>
        <DIST>5318.472</DIST>
        <POS>40.3575,-3.83676</POS>
      </CINE>
    - <CINE>
        <NAME>Cinesa Xanadú 3D</NAME>
        <HOR>12:15 - 16:15 - 19:15 - 22:00</HOR>
        <DIST>9211.294</DIST>
        <POS>40.3001,-3.92568</POS>
      </CINE>
    - <CINE>
        <NAME>Yelmo Cineplex Planetocio Villalba</NAME>
        <HOR>15:30 - 17:50 - 20:10 - 22:30</HOR>
        <DIST>37674.273</DIST>
        <POS>40.6362,-4.01593</POS>
      </CINE>
    </CINES>
  - <CRITICAS>
    - <CRITICA>
        <SITIO>FILMAFFINITY</SITIO>
        <AUTOR>Desconocido</AUTOR>
        <TEXT>Un inesperado y enorme éxito de taquilla, el film costó 29 millones de dólares, y en pocos meses superó los 255 millones de dólares en el Box
          Office USA, convirtiendo a Sandra Bullock en la primera actriz en superar los 200 millones por una película protagonizada por una mujer.</TEXT>
      </CRITICA>
    - <CRITICA>
        <SITIO>The Washington Post</SITIO>
        <AUTOR>Ann Hornaday</AUTOR>
        <TEXT>Fiel a su directa y desarmante realidad de su experiencia, la película tiene una obvia ausencia de sentimentalismo barato, que la salva de ser
          sensiblera o dulzona. (...) Puntuación: *** (sobre 4).</TEXT>
        <URL>http://www.washingtonpost.com/gog/movies/the-blind-side,1158777/critic-review.html#reviewNum1</URL>
      </CRITICA>
    - <CRITICA>
        <SITIO>Variety</SITIO>
        <AUTOR>Joe Leydon</AUTOR>
        <TEXT>Otro estimulante y placentero drama de deportes basado en hechos reales. (...) el film, atractivamente presentado, ocasionalmente corre el
          riesgo de forzar su credibilidad -o, peor, de invitar al escepticismo-.</TEXT>
        <URL>http://www.variety.com/review/VE1117941608.html?categoryid=31&cs=1</URL>
      </CRITICA>
    - <CRITICA>
        <SITIO>USA Today</SITIO>
        <AUTOR>Claudia Puig</AUTOR>
        <TEXT>Tiene buenas actuaciones y emocionantes escenas de fútbol. Pero su superficialidad la aparta de la conmovedora historia que podría haber sido.
          (...) Puntuación: **1/2 (sobre 4)</TEXT>
        <URL>http://www.usatoday.com/life/movies/reviews/2009-11-20-blindside20_ST_N.htm</URL>
      </CRITICA>
    - <CRITICA>
        <SITIO>Diario El País</SITIO>
        <AUTOR>Javier Ocaña</AUTOR>
        <TEXT>Que semejante pastel, mentiroso, melifluo y fariseo formara parte de la noche de los Oscar demuestra que Hollywood se dirige hacia su suicidio
          artístico, industrial y ético</TEXT>
        <URL>http://www.elpais.com/articulo/cine/desliz/Oscar/elpepicin/20100618elpepicin_7/Tes/</URL>
      </CRITICA>
    </CRITICAS>
</infocines>
```


Figura 2.3: Ejemplo de archivo XML

2.3. Implantación del sistema

A lo largo del desarrollo de la aplicación se implantaron dos versiones prototipo claramente diferenciadas combinando las diferentes alternativas de diseño mencionadas en la sección 2.2.

2.3.1. Versión Inicial

La aplicación, se ha basado en esta arquitectura para poder transmitir los datos necesarios entre el terminal móvil y el equipo servidor dada la variedad heterogénea en los sistemas operativos de los terminales móviles según se ha explicado en la sección 1.2 y su consecuente variación en el lenguaje de programación que utiliza cada uno.

Para esta versión se han utilizado las alternativas de diseño expuestas en las secciones 2.2.1 y 2.2.3 para la obtención del título de la película y la alternativa expuesta en la sección 2.2.4 para el envío de la información desde el servidor al terminal móvil.

Puesto que en el terminal sobre el que se ha trabajado usa código en Objective-C para programar la interfaz y las conexiones con el servidor y en el servidor se usa Java para programar la conexión con la BBDD, se estableció un protocolo de comunicación para la correcta transmisión de los datos.

El protocolo viene explicado en la siguiente figura:

FIGURA 2.4: Protocolo Cliente-Servidor Versión Inicial

Como se observa en la figura, en primer lugar se envía un objeto de tipo String que contiene el título de la película, el municipio, las coordenadas GPS del terminal móvil y el radio de búsqueda de cines al servidor, que tras una consulta a la BBDD de cines obtenemos los cines cercanos en los que proyectan la película. Esta información, junto con el tráiler, los datos de la película y las críticas es introducida en un fichero HTML que posteriormente es transmitido al terminal móvil para su visualización.

2.3.2. Versión Final

En esta versión se ha utilizado la alternativa de diseño expuesta en la sección 2.2.3 para la extracción del título de la película y la alternativa expuesta en la sección 2.2.5 para el envío de la información desde el servidor al terminal móvil con lo que el esquema seguido es el mismo pero hay un ligero cambio en el protocolo. En este prototipo se ha decidido eliminar la alternativa basada en el QR Code debido a que todavía no está implantado en Europa el uso de esta técnica. Además se ha eliminado del String transmitido

al servidor el rango introducido por el usuario dado que ahora se devuelve al terminal móvil todos los cines de la ciudad en la que se encuentra y la presentación de los cines en rango se hace dinámicamente sobre la aplicación del terminal reduciendo así el tiempo perdido en la transmisión de datos. En la siguiente figura se muestra un diagrama del protocolo:

FIGURA 2.5: Protocolo Cliente-Servidor Versión Final

En este caso, lo que se transmite por parte del terminal móvil es el título de la película junto con las coordenadas GPS y la ciudad en la que está ubicado el usuario. Después, tras hacer la consulta web a la página de eCartelera y obtener los cines de la ciudad donde proyectan la película, se envían al móvil todos los datos recopilados pero esta vez en formato XML para que el terminal los presente con la disposición que elija. Esto es así para que la portabilidad a otras plataformas móviles sea más sencilla y para facilitar el diseño modular del proyecto.

Capítulo 3

Servidor y Base de Datos

El servidor es el encargado de recibir las peticiones del cliente (terminal móvil), recuperar información de la BBDD y otras fuentes externas, procesar toda la información y generar el HTML o XML resultado listo para enviarlo de vuelta al cliente.

Para la programación del servidor se ha usado Java dado que es un lenguaje orientado a objetos muy versátil y extendido por la comunidad informática.

El servidor se ha diseñado con un sistema de hilos (threads) de manera que cada petición de un cliente diferente cree un nuevo hilo y sea tratada independientemente para así aprovechar el procesamiento en paralelo de la máquina.

En cuanto a la BBDD, se ha usado MySQL debido a que es una herramienta fiable a pesar de su carácter gratuito y porque ya se tenían buenas experiencias previas al haber trabajado en anteriores proyectos con esta herramienta.

Para conectar la aplicación del servidor hecha en Java y la base de datos de MySQL se ha usado un conector previamente diseñado pero configurado por los integrantes del grupo.

Ejemplo de ejecución

En una ejecución normal se sigue el siguiente diagrama de secuencia:

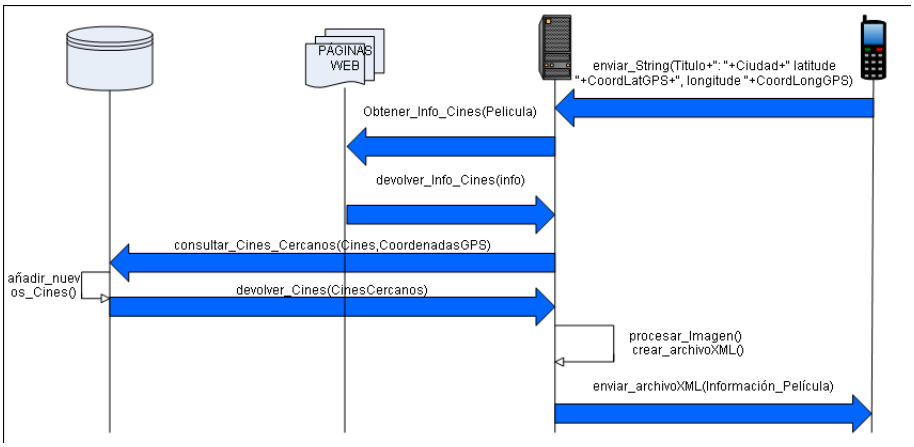


FIGURA 3.1: Diagrama de Secuencia de una comunicación

En primer lugar el servidor está a la escucha para recibir las conexiones de los clientes. Una vez que un cliente se ha conectado al servidor, este crea un nuevo hilo mediante la clase ThreadCines y continúa escuchando en el puerto.

Posteriormente, al ejecutar el hilo, se llama a la clase ParserCines con los datos extraídos de la conexión con el iPhone. A partir de aquí se usan las clases Peticiones y GoogleMaps para hacer las consultas a la página de eCartelera y obtener los datos de la película, los cines y los horarios. Una vez obtenida la información de los cines en los que proyectan la película, se hace una consulta a la BBDD mediante la clase ConectorBD para hallar la intersección entre los cines que proyectan la película y los cines que están dentro del radio que ha indicado el usuario.

Después de haber calculado toda la información anterior se llama a las clases Críticas y Trailer para que extraigan de Youtube y de las páginas de críticas el tráiler y las críticas y se invoca a la clase Cartel para que obtenga el poster de la película de la página de eCartelera.

Por último, se genera el fichero HTML o XML con los resultados obtenidos y calculados en los pasos anteriores y se remite a la clase ThreadCines para

que lo envíe de vuelta al iPhone.

3.1. Servidor

Haciendo una descripción a alto nivel el servidor realiza las siguientes funciones de cara al funcionamiento del sistema:

- Aceptar conexiones de los clientes: Consiste en abrir un socket en la máquina servidor en el puerto 7013 y mantenerlo escuchando para que cada vez que se conecte un cliente crear un hilo de ejecución en la máquina servidor que atienda las peticiones de ese cliente.

- Obtener la información de la película: Para conseguir los datos de título, director, país, duración y año el servidor obtiene el código fuente de la página de eCartelera correspondiente a la película y parsea dichos datos.

- Obtener la información de los cines: En este caso, el servidor hace una consulta mediante el título de la película y la fecha actual a la página de eCartelera y obtiene las salas de España en las que se proyecta. A partir de ahí se filtran en función de la ciudad que ha enviado el cliente. Además también se obtiene de la página de información de cada cine la dirección del mismo para su consulta a GoogleMaps y poder así obtener sus coordenadas GPS para posteriormente hallar la distancia a la que se encuentra del usuario.

- Obtener el cartel de la película: Con un nuevo parseo de la página de eCartelera se obtiene la URL del cartel de la película. Además se descarga y almacena en el servidor para futuras ampliaciones como por ejemplo la explicada en la sección 2.2.2 de comparación entre imágenes.

- Obtener la información de las críticas: El servidor se conecta a la página web de “FilmAffinity” y mediante el título y el año de la película obtiene las correspondientes críticas previo parseo del código fuente de la

página. También se obtiene de la página web de “FilmAffinity” la valoración de la película que posteriormente se convierte en la puntuación en estrellas que aparece en el resultado.

- Obtener el trailer de la película: Mediante la búsqueda del título de la película en “Youtube” junto con “trailer” y “español” se obtiene la URL del trailer tras el correspondiente parseo de la página de resultados.

- Producción del fichero resultado (HTML o XML): El servidor crea el correspondiente fichero resultado recopilando todos los datos enunciados anteriormente y los distribuye en sus respectivas posiciones en caso de ser un fichero HTML o los asigna las debidas etiquetas en caso de ser un fichero XML.

- Transmisión del fichero resultado al cliente: Se envía el fichero resultado por el socket abierto anteriormente de vuelta al cliente y se finaliza el hilo de ejecución para ese cliente.

A más bajo nivel la aplicación servidor se compone de los siguientes paquetes y clases según indica el diagrama:

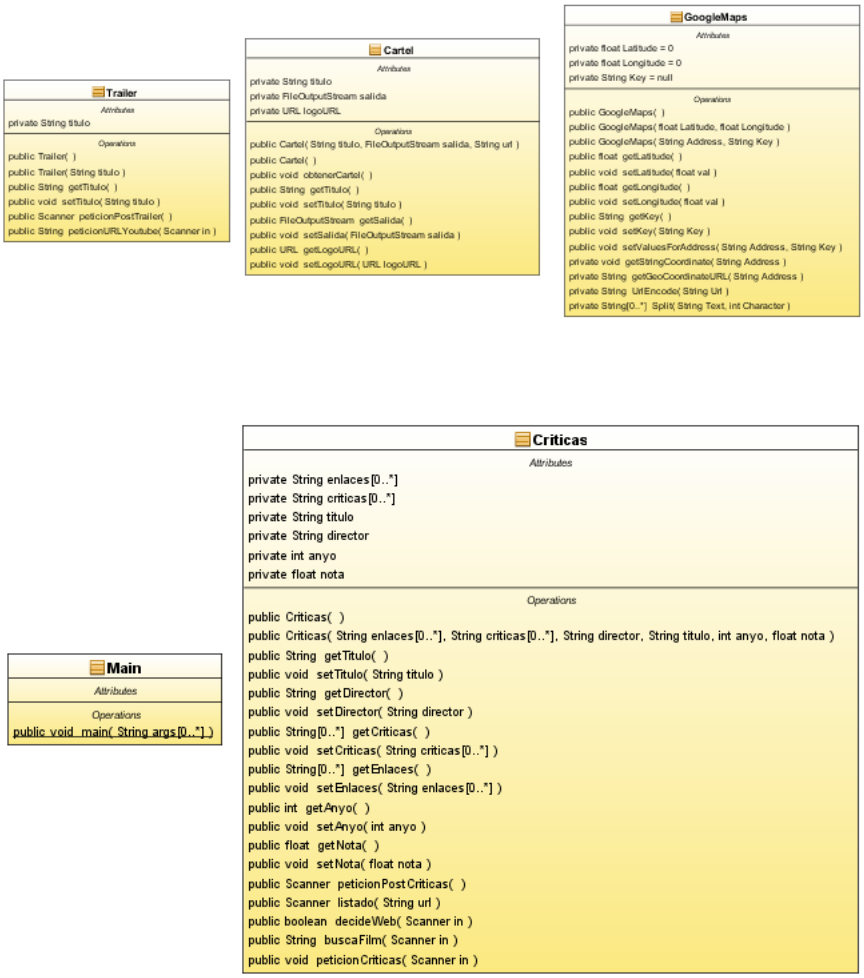


FIGURA 3.2: Diagrama de Clases Servidor (1)

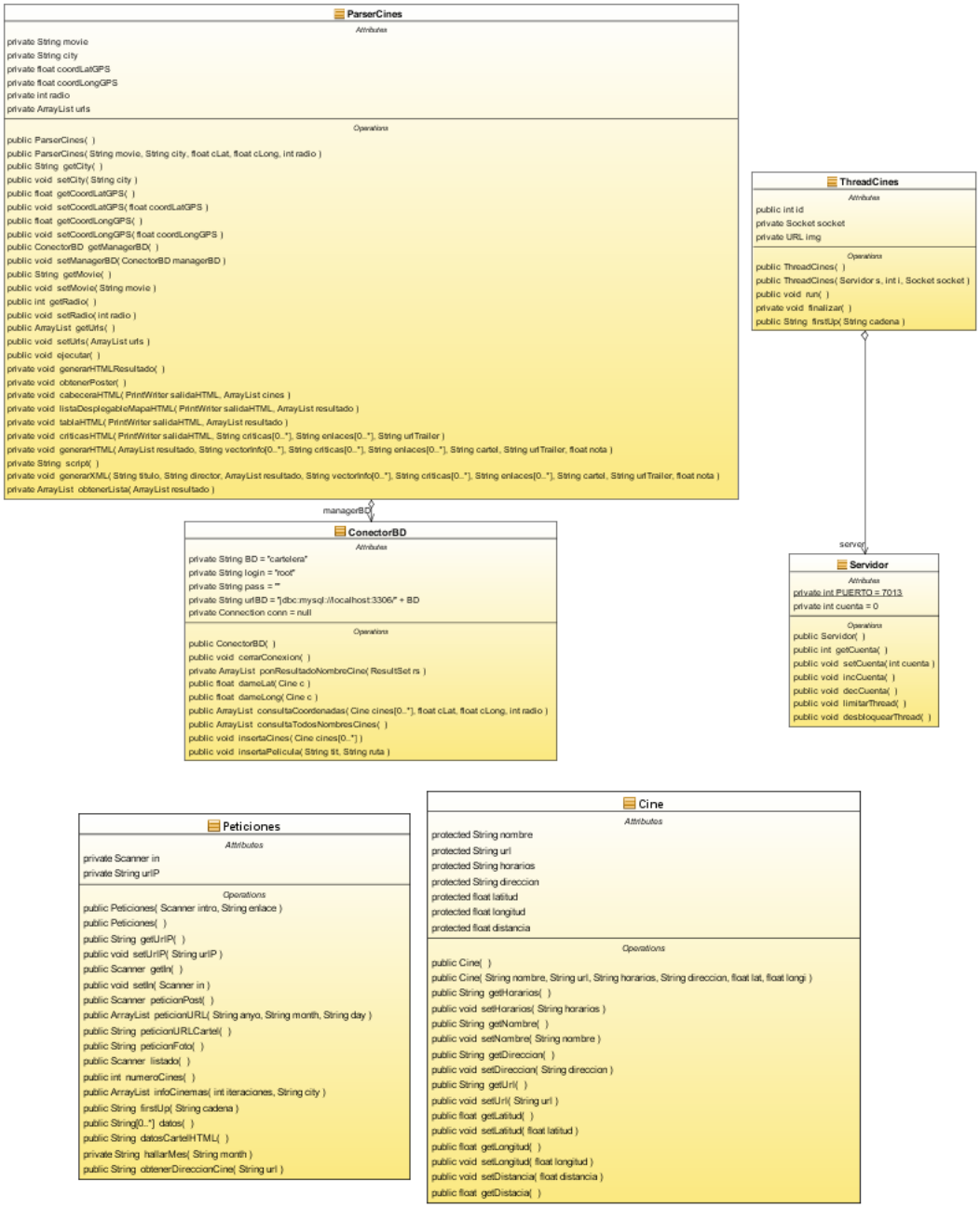


FIGURA 3.3: Diagrama de Clases Servidor (2)

La función de cada clase es la siguiente:

- Cartel: Se encarga de obtener el cartel de la película de una url y almacenarlo en un fichero de salida.
- Cine: Contiene las características de un determinado cine, el nombre, su url, sus coordenadas GPS y los horarios.
- ConectorBD: Se encarga de los accesos a la BBDD así como de su correcta puesta en marcha y desconexión. Contiene los métodos para añadir cines a la BBDD, consultar las coordenadas de los cines, insertar películas y hallar los cines cercanos a un punto dado un radio de distancia.
- Criticas: Su función es buscar las críticas en las respectivas páginas expertas, parsearlas para extraer el comentario y la puntuación asignada y hacer las consultas a esas páginas con el título de la película.
- GoogleMaps: Contiene el código necesario para hacer una petición a Google Maps con la dirección de un cine y así obtener sus coordenadas GPS para añadirlas a la BBDD y compararlas con las del iPhone.
- Main: Realiza una llamada a la constructora de Servidor para poner a la escucha el mismo.
- ParserCines: En esta clase es donde se almacenan los datos que envía el iPhone introducidos por el usuario y se genera el HTML resultado que es después enviado de vuelta al iPhone. Para ello esta clase utiliza instancias de las clases Peticiones, Criticas y Trailer. Contiene los métodos para recuperar el cartel y los cines en los que proyectan la película, generar la cabecera del HTML y generar el cuerpo del HTML.
- Peticiones: Se encarga de hacer las peticiones POST a la página de eCartelera con el título de la película, la fecha actual para obtener los cines y los carteles de la película. También se encarga de parsear y extraer la información relevante de todas las páginas que ha consultado para pasárselo a la clase ParserCines.

- Servidor: Su misión es mantener escuchando y a la espera siempre a nuestra aplicación además de crear los diferentes hilos cada vez que un cliente se conecta.
- ThreadCines: El objetivo de esta clase es extraer la información que transmite el iPhone del socket para poder pasársela a la clase ParserCines. También se encarga de enviar el fichero HTML al iPhone por el socket una vez creado.
- Trailer: Se encarga de hacer la petición POST a Youtube para obtener el tráiler de la película y almacenarlo.

3.1.1. Ampliaciones

Actualmente, solamente se usa la página web de eCartelera para obtener la información de los cines en los que proyectan las películas, los horarios de las mismas y sus datos. Para evitar depender de una sola página se llegó a la conslusión de que se podía establecer una jerarquía de varias páginas web fuentes de las que extraer dicha información.

Es decir, tener un listado de varias páginas web de donde se obtiene toda la información de cines, películas y horarios con el fin de que si en algún momento la página principal sufre algún inconveniente (problemas de conexión, servidor "offline"...) se pueda pasar a la siguiente página de la lista para buscar los datos antes mencionados y así sucesivamente.

Para ello, es necesario diseñar e implementar una clase nueva para cada página web de la lista de fuentes. Esto es debido a que para cada página web los métodos de parseo de la información son completamente diferentes y no se puede estandarizar un patrón a reconocer para todas. El resultado sería una clase similar a la clase Peticiones pero con otros patrones de reconocimiento.Con esta ampliación lo que se busca es proporcionar a la aplicación una mayor robustez y una mayor tolerancia a fallos.

También se ha planteado la posibilidad de portar la aplicación a otros terminales móviles además del iPhone. Con este fin se ha tratado de estandarizar

tanto el protocolo de envío de información como la propia información enviada. En la segunda versión del prototipo basado en reconocimiento de imágen, se cambió la información de respuesta del servidor para que en vez de enviar una página web HTML se envíe un fichero en XML para que sea el propio terminal el que presente dicha información según las directrices de cada compañía. Con esta mejora se pretende conseguir una aplicación compatible con la mayoría de sistemas móviles del mercado.

3.2. Base de datos

Según se ha comentado anteriormente, la base de datos utilizada en el proyecto ha sido MySQL, pero se podría sustituir por cualquier otra fácilmente realizando los cambios pertinentes en la clase ConectorBD.

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB -desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009- desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y el copyright del código está en poder del autor individual, MySQL es propietario y está patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet. MySQL AB fue fundado por David Axmark, Allan Larsson y Michael Widenius.

Existen varias APIs que permiten, a aplicaciones escritas en diversos

lenguajes de programación, acceder a las bases de datos MySQL, incluyendo C, C++, Pascal, Delphi (via dbExpress), Eiffel, Smalltalk, Java (con una implementación nativa del driver de Java), Lisp, Perl, PHP, Python, Ruby,Gambas, REALbasic (Mac y Linux), (x)Harbour (Eagle1), FreeBASIC, y Tcl; cada uno de estos utiliza una API específica. También existe un interfaz ODBC, llamado MyODBC que permite a cualquier lenguaje de programación que soporte ODBC comunicarse con las bases de datos MySQL. También se puede acceder desde el sistema SAP, lenguaje ABAP.

MySQL es muy utilizado en aplicaciones web, como Drupal o phpBB, en plataformas, y por herramientas de seguimiento de errores como Bugzilla. Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL. MySQL es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional MyISAM, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones. Sea cual sea el entorno en el que va a utilizar MySQL, es importante adelantar monitoreos sobre el desempeño para detectar y corregir errores tanto de SQL como de programacion.

Por las razones expuestas anteriormente y por el alto nivel de integración para el lenguaje de programación JAVA se eligió MySQL como base de datos para la aplicación.La estructura de la tablas de la BBDD viene explicada en el siguiente diagrama:

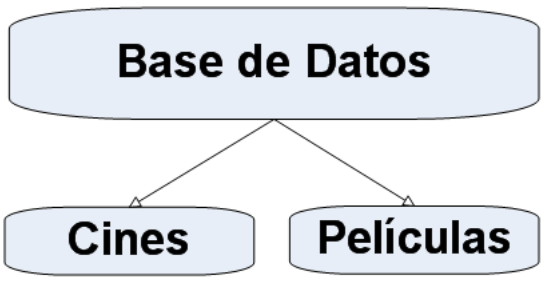


Figura 3.4: Diagrama Estructura BBDD

Como se puede observar en el diagrama anterior, la BBDD se basa en dos tablas: Cines y Peliculas. La BBDD se construye mediante un sistema de inclusión de información automático basado en que en cada consulta de un cliente se añaden a la BBDD aquellos cines o películas que previamente a la consulta no estuvieran incluidos.

3.2.1. Tabla Cines

En esta tabla se recoge la información de las coordenadas de latitud y longitud de los cines que se han obtenido como resultado en alguna consulta de un cliente. Sus atributos son:

- Nombre (Varchar(50)): Es la clave primaria de la tabla y contiene el nombre del cine.
- CoordLat (float): Contiene la coordenada de latitud donde está ubicado el cine.
- CoordLong (float): Contiene la coordenada de longitud donde está ubicado el cine.

3.2.2. Tabla Películas

En esta tabla se recoge la información de las películas así como la ruta donde está almacenado el cartel de la misma con el cual se va a hacer la comparación con la imagen enviada desde el iPhone. Sus atributos son:

- Titulo (Varchar(50)): Es la clave primaria de la tabla y contiene el título de la película.
- PathImagen (Varchar(100)): Contiene la ruta donde está almacenado el cartel de la película.

3.2.3. Conector BBDD

Con el fin de conectar la aplicación desarrollada en Java con la BBDD de MySQL se ha utilizado un conector prediseñado que se ha añadido como librería al proyecto denominado "mysql-connector-java-5.1.6.jar".

Mediante la configuración del mismo proporcionándole los datos del servidor donde está ubicada la BBDD así como los datos de login ha permitido enlazar la aplicación con la BBDD. El conector consta de 5 componentes según se muestra en la siguiente figura:

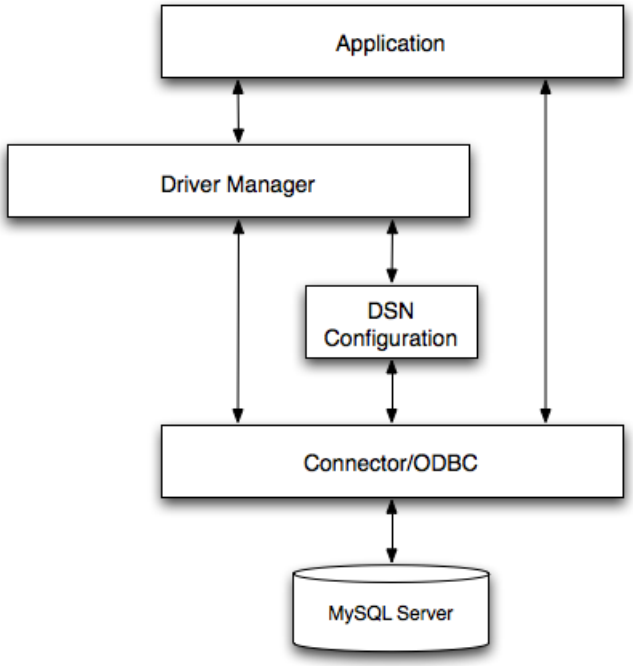


Figura 3.5: Arquitectura Conector BBDD

- Aplicación: Usa el API de ODBC para acceder a la información del servidor de MySQL, por lo tanto, el API de ODBC sirve de intermediario entre la aplicación y el “Driver Manager”. La aplicación se comunica con el “Driver Manager” usando llamadas de ODBC estándar y sin preocuparse de la localización o modo de almacenamiento de la información a la que quiere acceder. La aplicación solo necesita saber el Nombre del Origen de Datos (DSN).

- "Driver Manager": Es una librería que controla la comunicación entre la aplicación y los dispositivos. Se encarga de tareas como la resolución de los Nombres de Origen de Datos (DSN) que identifican a un controlador de una base de datos, una base de datos, un servidor y una información de "login" para acceder a la base de datos; la carga y descarga del controlador necesario para acceder a una determinada base de datos; el procesamiento de de las llamadas a ODBC o su delegación al controlador para procesarlas.
- "Connector/ODBC Driver": Es una librería que implementa las funciones soportadas por el API de ODBC. Procesa las llamadas a funciones de ODBC, hace las consultas SQL al servidor de MySQL y devuelve el resultado a la aplicación. Además, si es necesario, modifica las peticiones de la aplicación para que sean compatibles con la sintaxis de MySQL.
- "DSN Configuration": Almacena el controlador y la información de la base de datos requerido para conectar al servidor. Se usa por el "Driver Manager" para determinar qué controlador ha de ser cargado en función de la base de datos con la que se vaya a conectar.
- "MySQL Server": Es el la base de datos donde está almacenada la información. Aquí es de donde se extrae la información que necesita la aplicación o donde se guarda la información que la aplicación crea oportuna.

3.2.4. Ampliaciones

En un principio se estudió la posibilidad de almacenar los horarios de las películas en la BBDD para poder tener un acceso más rápido a ellos. Sin embargo, esta posibilidad se descartó debido a la falta de coherencia que podría suponer si un cine elimina o añade algún horario en su cartelera. Además, para poder mantener la coherencia entre la información mostrada en las páginas web y la BBDD sería necesario utilizar algún tipo de robot que periódicamente actualizara la información de la BBDD.

Una posible ampliación es la implementación de dicho robot para poder así mantener la coherencia de la BBDD con las páginas web y poder acceder a la información de las películas de un modo mucho más eficiente y con un incremento en el rendimiento de la aplicación ya que la mayor pérdida de tiempo se produce en las consultas a las páginas web.

Capítulo 4

Cliente iPhone

El cliente de la aplicación es la parte más importante, pues es con lo que trabajará el usuario. Desde el cliente, por cualquiera de los medios citados en la sección sobre las alternativas de diseño 2.2 , el usuario podrá solicitar la información de una determinada película. Además de la petición de la película en sí, se incluyen los datos GPS del terminal, es decir la posición y la comunidad autónoma donde nos encontramos, con el objetivo de que el servidor pueda devolver los cines cercanos donde emitan dicha película. Además es el cliente quien muestra los datos formateados para la mejor compresión del usuario a partir del archivo XML recibido, por tanto es importante crear un cliente visualmente atractivo y sencillo de usar.

Por ello se eligió el iPhone como primer terminal donde desarrollar la aplicación. El iPhone se trata del dispositivo móvil más usado de internet, lo cual amplia el espectro de posibles usuarios. Además presenta una interfaz sencilla y eficaz, donde podemos mostrar toda la información que necesitamos de manera que el usuario no se sienta invadido por un exceso de información y que sepa donde encontrar lo que necesita en la aplicación. Además la curva de aprendizaje para el uso del dispositivo es prácticamente inexistente, sus intuitivos interfaces hacen que desde el primer momento se le pueda sacar el máximo partido. Por último otro factor importante para la elección de este modelo ha sido el entorno de desarrollo, Apple ofrece un entorno integrado, basado en el editor de código por defecto de MacOS X, Xcode. Desde él

podemos realizar todos los paso de la creación de la aplicación, desde el código hasta los interfaces, incluso podemos medir el rendimiento o el consumo de memoria de la aplicación. Lo que facilita mucho el proceso de desarrollo en sí. A continuación se describe con más detalle la implementación del cliente.

4.1. ¿Qué es el iPhone?

Actualmente el iPhone es el teléfono móvil más avanzado del mercado, y no ya solo por su software (iOS4), común en las últimas 3 generaciones, también por su hardware, recientemente presentado y conocido como "iPhone 4". Esta última generación cuenta con un procesador desarrollado por la propia Apple, en colaboración con Samsung, un A4, siendo una versión de menos consumo y dimensiones que la que se encuentra en el iPad, pero con las mismas prestaciones, además esta versión ya cuenta con 512MB de memoria RAM, lo que convierte este iPhone en uno de los dispositivos más potentes del mercado.

Pero este iPhone no solo ha mejorado con el procesador, ha cambiado por completo. El uso de un chip más pequeño ha permitido aumentar el tamaño de la batería, con lo que sonseguimos un notable aumento de autonomía, algo realmente necesario en este tipo de dispositivos. Además se ha mejorado la cámara de fotos, no solo en megapíxeles, si no que también el sensor, siendo uno de los más avanzados del mercado, también se ha incorporado un flash de LED. Aprovechando la mejora de la capacidad de proceso el iPhone es ahora capaz de grabar, reproducir y tratar vídeo en alta definición.

La tercera mejora más importante es la pantalla. Esta nueva versión incorpora un nuevo tipo de panel bautizado como "retina display", su característica principal es que tiene una densidad de pixeles mayor que la que es capaz de apreciar el ojo humano medio a unos 30 cm de distancia. Consiguiendo una resolución de 960x640 pixeles en una pantalla de 3,5'. De esta forma consiguen que los textos tengan la misma definición que un texto escrito, haciendo la navegación web, la lectura de eBooks y demás tareas mucho más cómodas, además de ofrecer una calidad para la visualización de fotos y libros a la que

pocos dispositivos pueden aspirar. Además de estas tres destacables características, el iPhone ahora incorpora giroscopios, que unidos al chip GPS, los acelerómetros y la brújula, lo convierten en un dispositivo perfecto para ser usado con aplicaciones de realidad aumentada.

El sistema operativo actual del iPhone no está exento de innovaciones. Aunque a priori parece que muchas de las mejoras son ya conocidas en otros sistemas operativos móviles, también forman parte de sus puntos débiles. La más destacada es la multitarea. Mientras en otros sistemas cuando hay más de dos aplicaciones abiertas ya se empiezan a resentir, en iOS 4 han conseguido una manera de gestionar la memoria que evita esas ralentizaciones y una administración de los servicios en segundo plano que garantiza el mantenimiento de recursos, es decir, solo se permiten en segundo plano aplicaciones de música, subida o descarga de archivos y sistemas de geolocalización GPS, de esta manera el resto de aplicaciones se mantienen dormidas hasta que son activadas. Siendo la activación inmediata y el consumo de recursos en espera mínimo. De la misma manera que ocurrió con la incorporación del "copiar y pegar", llegó tras dos actualizaciones de softwate, pero de una manera fácil e intuitiva que no se había conseguido en ninguna otra plataforma. También en esta versión han incorporado la posibilidad de sincronizar el dispositivo con un treclado bluetooth, siendo el iPhone un sistema prácticamente completo

4.2. Objective C y el entorno de desarrollo

Para el desarrollo de aplicaciones para el iPhone se utiliza el lenguaje Objective C. Es una variante propietario de C, orientado a objetos, basado en las librerías "Cocoa", presentes en los sistemas operativos de Apple. Para las operaciones "no orientadas a objetos" su funcionamiento es prácticamente igual que el lenguaje C convencional, sin embargo, cuando tratamos con objetos es donde vemos la diferencia. Basa su funcionamiento en el paso de mensajes, donde su mayor desventaja es que no tiene comprobación de tipos, de esta forma el receptor del mensaje no garantiza que vaya a responder al mensaje, en cuyo caso lanza una excepción. Sin embargo una de las ventajas

de resolver los métodos en tiempo de ejecución es que tras la primera ejecución de un mensaje, que en este caso puede ser hasta 3 veces más lenta que en C++, las siguientes se obtienen de una caché intermedia, haciendo cada ejecución del mensaje un 50 % más rápidas que las llamadas en C++.

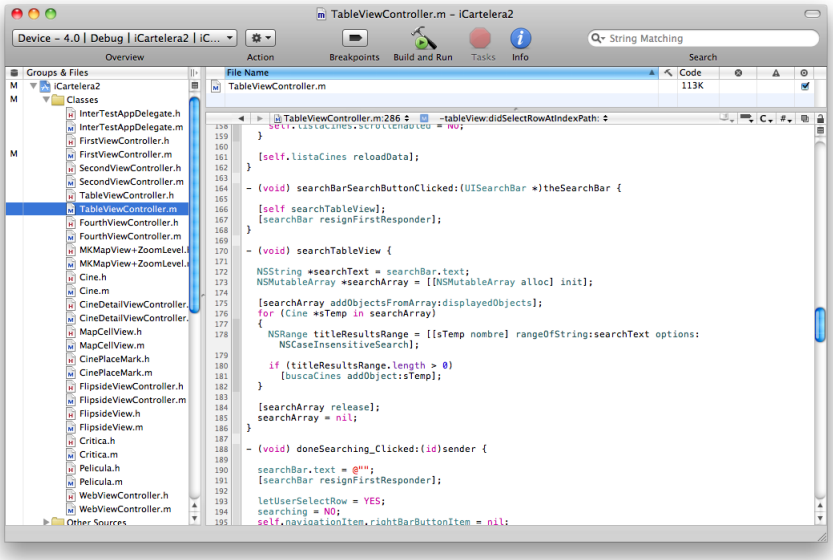


FIGURA 4.1: Vista general del entorno de desarrollo

En general no es un lenguaje muy complejo, comparándolo con C++, pero requiere de un periodo de aprendizaje y adaptación. Otra de sus desventajas, y además en el iPhone se trata de algo muy importante, es la ausencia de un gestor automático de memoria, como el "garbage collector" presente en Java. Debido a las limitaciones de memoria de los dispositivos, sobre todo de los de primera y segunda generación, es primordial liberar todas las reservas de memoria una vez que el objeto es destruido. Es un lenguaje muy enfocado a los interfaces de usuario, y es por ese motivo que su uso conjunto con el entorno de desarrollo, Xcode, hacen de ello una muy potente herramienta de programación.

El Xcode es un entorno de desarrollo, o mejor dicho, una suite de desarrollo, además de incluir el propio programa editor-compilador-debugger, incluye aplicaciones para todas las etapas por las que pasas en la elaboración

de un programa.

La segunda en importancia sería Interface Builder, con ella podemos crear todo tipo de interfaces, tanto a partir de plantillas como desde cero, que con bastante facilidad podremos enlazar con nuestro código. Además nos permite desarrollar interfaces tanto para las dos resoluciones de pantalla del iPhone como para el iPad. Otra de las herramientas más destacadas es Instruments, con ella podemos hacer mediciones del rendimiento de la aplicación en tiempo real, tanto del uso del procesador, como de la memoria y cualquier otro detalle que nos interese observar.

Otra de las facilidades que ofrece el entorno de desarrollo para el iPhone es la gran cantidad de APIs disponibles para el usuario, y la sencillez de uso de las mismas. De esta manera Apple proporciona las librerías para poder sacar partido a todas las funcionalidades del dispositivo, es decir, si queremos usar la geolocalización debemos cargar la librería "Core Location", si además queremos usar la agenda usaremos el "Address UI Kit". De esta manera es muy sencillo añadir características a nuestras aplicaciones.

4.3. Aspectos generales

La versión del cliente para el iPhone tienen como función principal la de tratar y mostrar la información recibida en el archivo XML desde el servidor. Además podemos solicitar la información de una película mediante la introducción del título con el teclado. Como posible mejora consideramos enviar una foto del cartel de la película, ya sea tomada con la cámara, descargada de internet o una foto previamente guardada en el terminal, de tal forma que el servidor reconozca de qué película se trata mediante la comparación con los carteles originales.

En la implementación de este cliente se ha buscado conseguir una alta velocidad de respuesta. Cuando se diseña una aplicación para un dispositivo móvil, como es el iPhone, tenemos que tener en cuenta que esta aplicación se va a usar en cortos periodos de tiempo, y la información ha de aparecer de manera casi instantánea. Por este motivo se optó por almacenar el

archivo XML en el propio iPhone y que sea la aplicación, con cada ejecución, quien actualice los datos referentes a la posición, de esta manera siempre tendremos la información actualizada aunque nos estemos moviendo con el iPhone mientras la usamos. Dentro de la aplicación también hemos diseñado un pequeño navegador web, de tal forma que se puedan consultar las páginas de las críticas sin tener que salir de esta.

La aplicación está basada en un diseño de pestañas, existiendo 4 posibles visualizaciones. Con este diseño se gana en claridad de uso de la aplicación, pues de un primer vistazo se puede ver todas las opciones disponibles. Como pestaña por defecto se selecciona la de información de la película, pues se trata de la más visual y atractiva para el usuario. En las secciones siguientes se detallará el diseño de casa una de estas visualizaciones.

4.4. Información de película

Esta pestaña está dedicada únicamente a la película solicitada, no se muestra aún ningún tipo de información al respecto de los cines. La primera vista incluye la información más básica, se muestra el cartel de la película, el título, el año, el país y el director.

Se reserva un espacio, basado en una pequeña vista ÜIWebview", que es la que nos permite incorporar vistas de internet en el SDK, con el trailer de la película. Y por último, en la parte inferior de la vista mostramos la puntuación, usando un sistema de hasta 5 estrellas y un botón para acceder a la sección de las críticas.

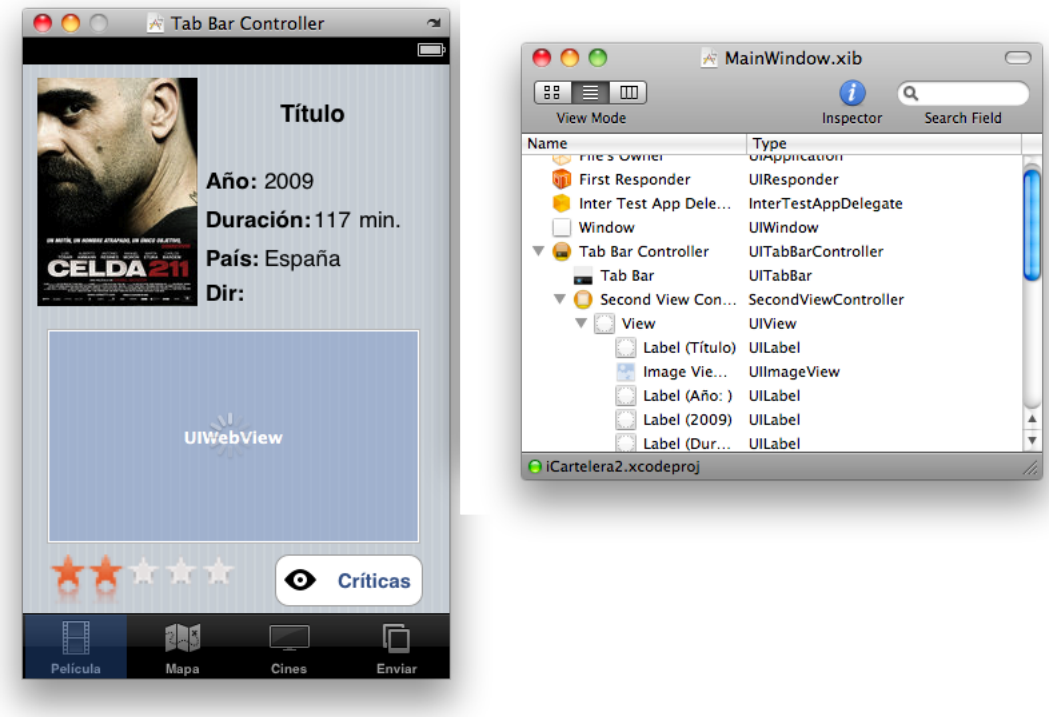


FIGURA 4.2: Información de la película

Para hacer la aplicación más visual hemos elegido una transición de ventana de tipo "flipside", es decir, que la ventana da la vuelta y se muestran las críticas como si estas estuviesen por detrás. Para mostrarlas se ha optado por una vista de tabla por secciones, donde cada sección corresponde a una crítica. El usuario verá la publicación de donde se obtiene la crítica, el autor de la misma, la crítica en sí, en función del tamaño de esta se mostrará un extracto o la crítica entera. Y por último, si está disponible, un enlace al sitio original de la crítica.

Figura 4.3: Críticas de la película

En la visualización de la crítica hemos creado un navegador, donde podremos movernos a la página anterior y siguiente cuando las haya, parar la carga o recargar. En definitiva una manera cómoda y sencilla de navegar por el sitio de la crítica sin necesidad de salir de la aplicación. Aún así se ofrece la posibilidad de abrir la página en Safari, el navegador por defecto en el iPhone, donde ya están habilitadas todas las funciones que admite la navegación en el iPhone. Para mostrar y salir de este navegador se ha optado por una transición de disolución, también con el objetivo de hacer la aplicación visualmente más atractiva.

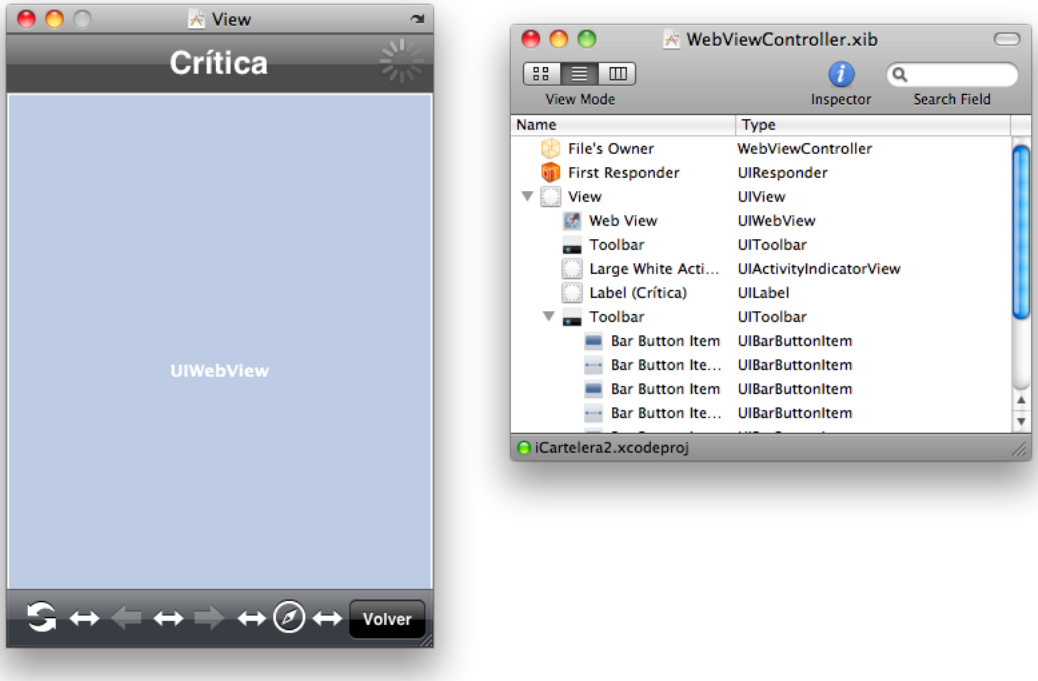


FIGURA 4.4: Navegador integrado

4.5. Mapa

Esta pestaña muestra los cines en un mapa centrado en nuestra ubicación actual. Se da la posibilidad de seleccionar el rango de búsqueda de los cines, adaptando el zoom del mapa, también tenemos la posibilidad de seleccionar entre los tres tipos de mapa disponibles, es decir, con la vista de callejero, la vista satélite o una mezcla de ambas. Por defecto se muestran los cines en un radio de 2000 metros.

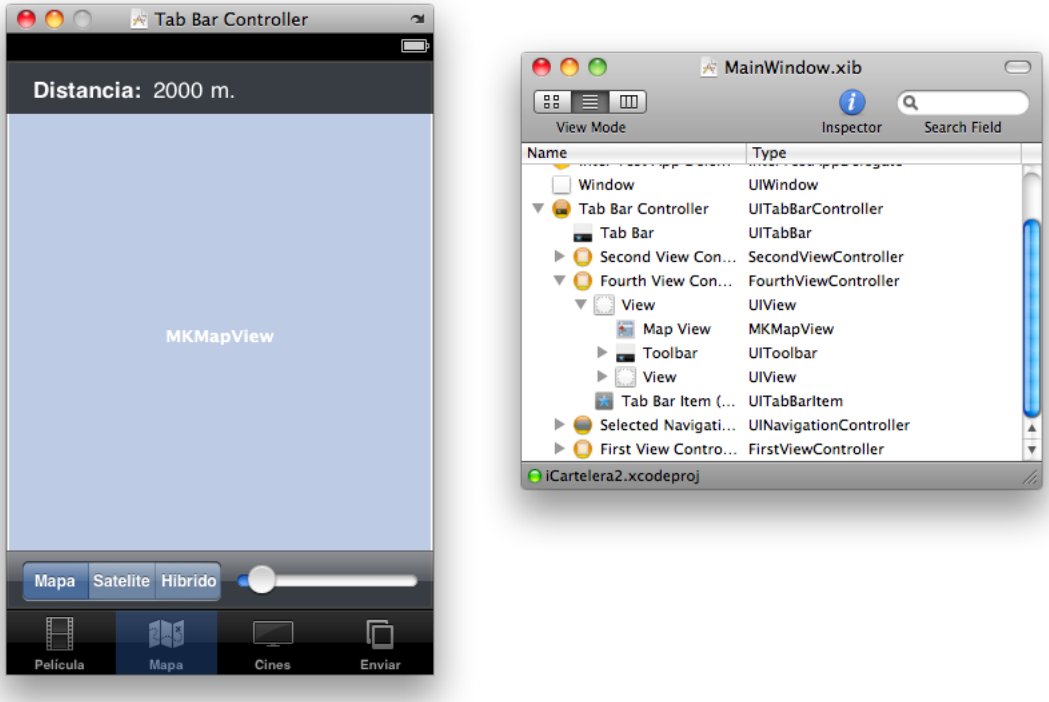


FIGURA 4.5: Mapa de cines

Los cines y la ubicación están representados por chinchetas, la de color rojo representa donde nos encontramos y las azules los cines, cada una de estas chinchetas tiene una pequeña vista de título, donde se muestra el nombre del cine y las horas de las sesiones en las que se emite la película, se ofrece también un botón mediante el que acceder a la vista de detalle del cine.

FIGURA 4.6: Detalle de las chinchetas

La vista de detalle es una pequeña tabla de tres secciones. En la primera mostramos los horarios de las sesiones. La segunda muestra la distancia, en metros o kilómetros, en función de la magnitud de esta, desde la ubicación actual al cine, en línea recta, sin tener en cuenta las calles. Y por último se presenta una pequeña vista de mapa con la ubicación del cine ofreciendo la posibilidad de mostrar la ruta desde la ubicación actual al cine.

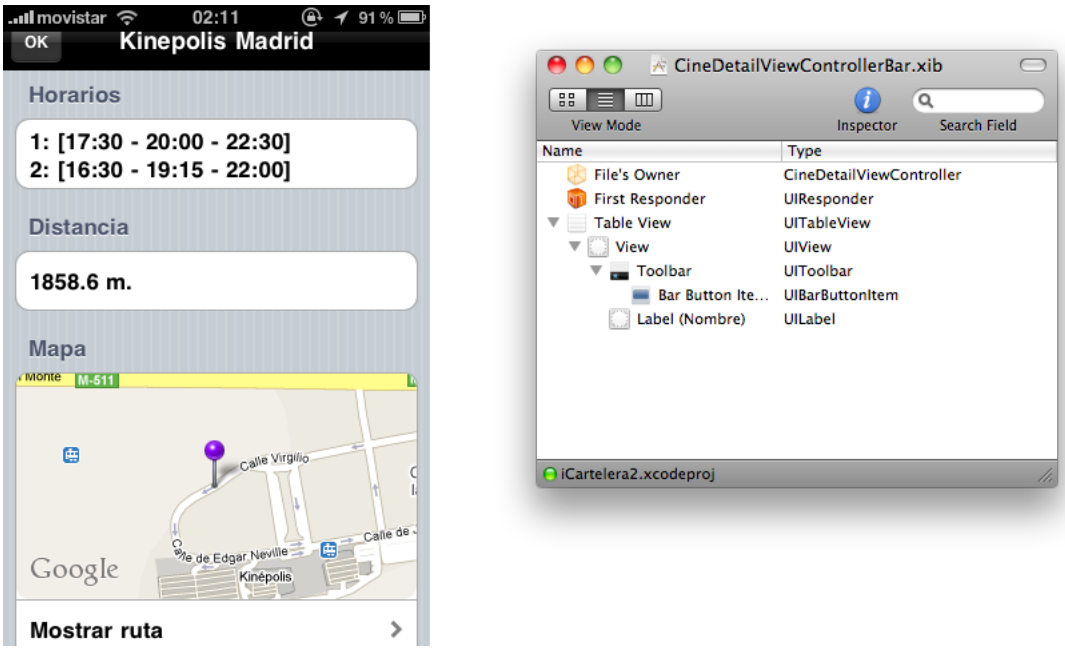


FIGURA 4.7: Detalle de un cine

4.6. Cines

La tercera pestaña consiste en una tabla con plana donde se muestra el nombre de los cines y la distancia. La tabla está ordenada en orden de cercanía, de más cercano a más lejano. En la primer vista de esta pestaña, si antes no hemos pasado por la de mapas, se muestra una lista de todos los cines en los que se emite la película. Pero esta vista está ligada al rango seleccionado en la vista del mapa, es decir, aparecen los mismos cines que en el mapa pero en formato de lista.

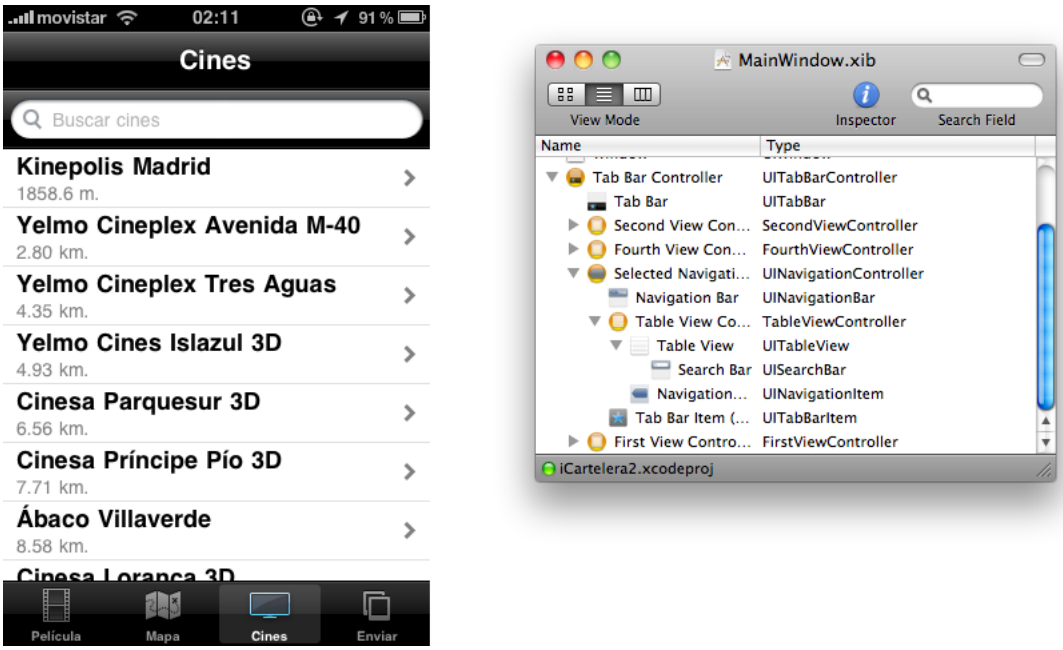


FIGURA 4.8: Tabla de cines

Además en esta pestaña incorpora una función de búsqueda en la tabla, por nombre y en tiempo real con la escritura de la línea de búsqueda, es decir, carácter a carácter actualiza los cines mostrados. Al seleccionar un cine se llega a la vista detallada de este, igual a la mostrada desde la vista de mapa, pudiendo también acceder a la ruta.

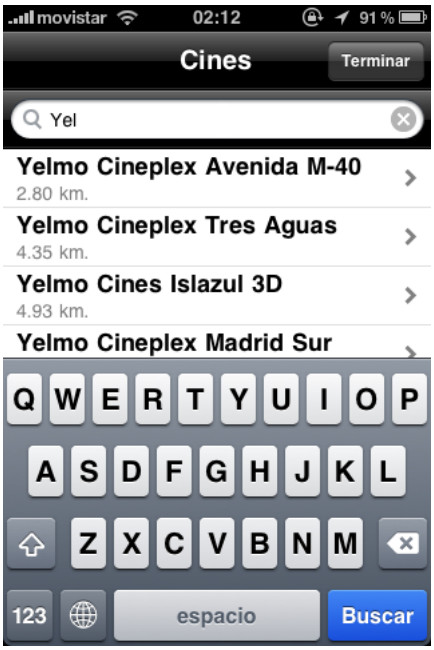


Figura 4.9: Búsqueda de cines

4.7. Envío de la petición

La última pestaña es donde se encuentran todas las alternativas de envío al servidor. Desde aquí se puede seleccionar una foto para enviar, buscando en el álbum o directamente desde la cámara. O mediante una ventana emergente enviar un texto con el título de la película. En esta ventana además se obtiene una previsualización del área seleccionada de la foto, tras hacer zoom y recortar las zonas de la foto que no son necesarias.

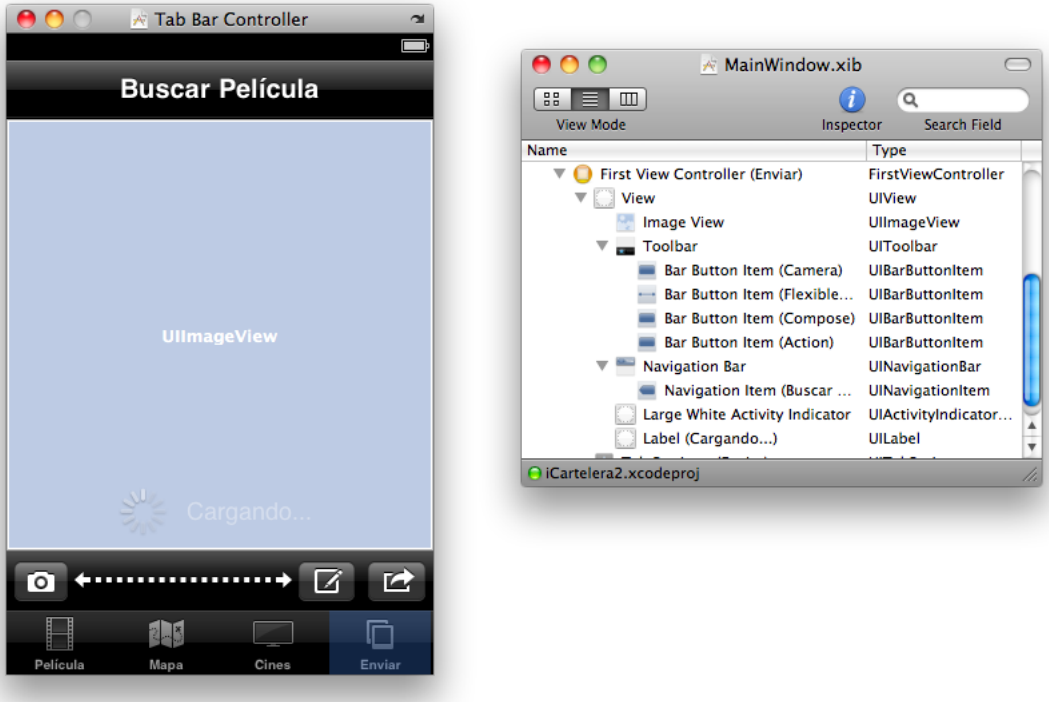


FIGURA 4.10: Envío de datos

Cuando la aplicación se encuentra con esta pestaña activa empiezan a funcionar los modulos “corelocation” y “geocoder” para obtener las coordenadas y la información de la comunidad donde se encuentre el usuario, para enviar toda la información al servidor. Cuando la pestaña deja de estar activa se vuelve a poner el GPS en reposo para conservar la batería en la medida de lo posible.

La aplicación permanece inactiva mientras se produce el intercambio de información con el servidor, para informar de este hecho al usuario se muestra un indicador de actividad animado que desaparece en el momento en que se recibe el archivo XML.

4.8. Instalación en el dispositivo

El mercado de aplicaciones para el iPhone tiene solo una vía de distribución, la "Appstore" de Apple. Para poder distribuir una aplicación hay que estar registrado como desarrollador, con ello se obtiene el derecho a poder probar las aplicaciones hasta en 100 dispositivos. También a distribuirla de forma "ad-hoc", es decir, directamente a cualquier dispositivo, sin pasar por la Appstore.

Figura 4.11: iCartelera en iTunes

Pero para la distribución a nivel mundial hay que enviar la aplicación a Apple. Ellos la testean, comprueban el código, y si cumple las condiciones que ellos especifican, la aprueban. Es entonces cuando se lanza la aplicación en la Store. Una de las ventajas de este modelo de negocio es que el desarrollador no tiene que preocuparse por el hosting o la publicidad, pues Apple será quien gestione todo eso y quien promocione la aplicación si esta tiene éxito.

Finalmente el usuario para instalarla tendrá que comprar la aplicación, desde iTunes o el propio dispositivo, y con la siguiente sincronización la aplicación quedará instalada. Es un proceso sencillo y bastante intuitivo, además de automático. De esta manera se asegura una correcta instalación independientemente del nivel de manejo del usuario.

Capítulo 5

Profiling

5.1. Introducción

Para evaluar el rendimiento de nuestra aplicación se ha hecho un estudio detallado con la herramienta Profiling que proporciona el IDE de Netbeans. También hay que tener en cuenta que debido a la sobrecarga del sistema al entrar en un modo de depuración (profiling) los tiempos son mayores que en una ejecución normal pero sirven a título comparativo.

Debido a ello, se optó por integrar un sistema de Profiling en otro paquete del servidor denominado Profiling y capturar los tiempos de ejecución manualmente en cada método. Este sistema está basado en un patrón Singleton para que todos los hilos introduzcan sus tiempos en el mismo objeto y así poder mostrarlos en un fichero HTML con los tiempos reales de cada método.

Los datos presentados a continuación son los resúmenes del estudio realizado y están basados tanto en la herramienta Profiling del IDE de Netbeans como en el sistema de Profiling incorporado a la aplicación por los integrantes del grupo. Se ha tenido en cuenta tanto el rendimiento de CPU como la carga en memoria del servidor que supone la aplicación.

Además, se ha medido la duración media de tratamiento de una petición del cliente al servidor dando como resultado un tiempo de ejecución medio de

4,45 segundos desde que el usuario envía la información del título al servidor desde el terminal móvil hasta que se le presenta la información resultado en su terminal.

5.2. Rendimiento de CPU

Desde una perspectiva de las funciones realizadas por el servidor se estudiará el tiempo consumido por cada función en relación al tiempo total de ejecución de una petición del cliente. El siguiente gráfico muestra un resumen del estudio:

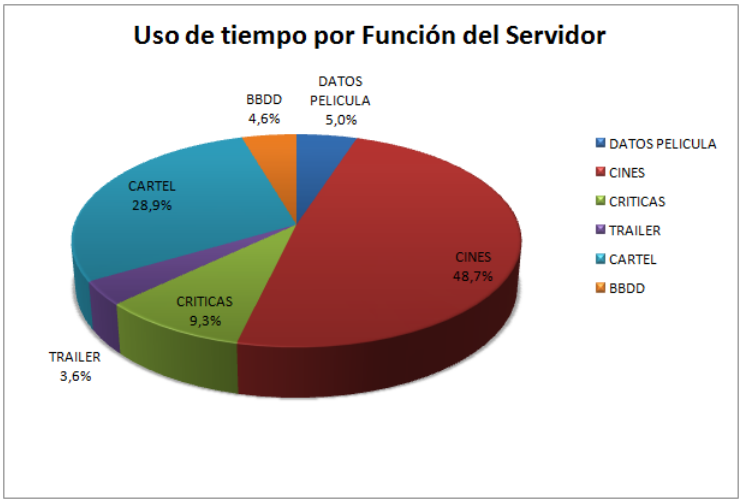


Figura 5.1: Distribución de tiempo por Funciones del Servidor

En la figura se puede ver que casi un 50 % del tiempo se invierte en la obtención de los datos de los cines. Esto es debido a que es necesario consultar la página web de eCartelera en varias ocasiones y recuperar el código fuente de dos páginas web para realizar los correspondientes parseos y extraer la información relativa a los cines y sus horarios. También cabe destacar que se invierte cerca de un 30 % del tiempo en la obtención del cartel de la película. En este caso, se invierte esa cantidad de tiempo porque es necesario descargar la foto desde la página web de eCartelera con su correspondiente pérdida de tiempo en la transferencia.

A continuación se muestra un detalle del tiempo empleado en la obtención de los datos de los cines ya que es la función que tiene una mayor inversión de tiempo en cada petición del cliente:

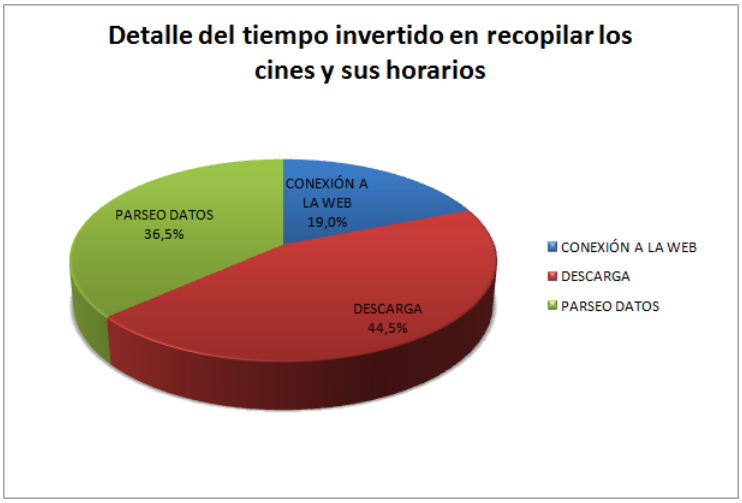


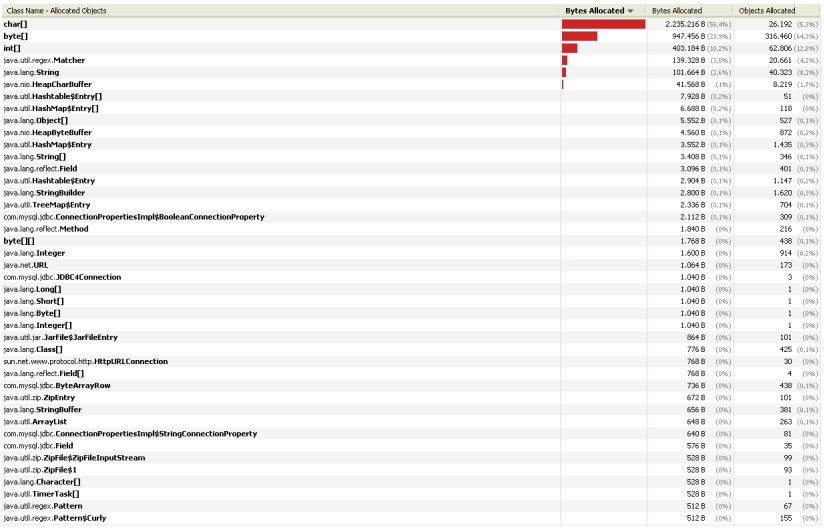
FIGURA 5.2: Detalle del tiempo invertido en obtener los datos de los cines

Se ha optado por agrupar el tiempo empleado en la obtención de los datos de los cines en 3 apartados indicados con porcentajes: tiempo invertido para establecer la conexión entre el servidor y las diferentes páginas fuente, tiempo utilizado para la descarga del código fuente de cada página fuente y tiempo empleado parseando dicho código fuente para extraer la información de los horarios de los cines.

Según muestra el gráfico, la mayor parte del tiempo invertido en la obtención de los datos de los cines está repartida entre la descarga del código fuente de la página y los correpondientes parseos del mismo para poder extraer los horarios de cada cine. Una posible solución para reducir el tiempo empleado en la extracción de los horarios y en la descarga del código fuente de las páginas es mantener almacenada en la BBDD dicha información según se ha explicado en la sección 3.2.4.

5.3. Carga en memoria

En esta sección se va a mostrar lo que ocupa la aplicación en la memoria de la máquina servidor. Se mostrarán en el cuadro el tamaño que ocupa cada objeto o grupo de objetos creados en memoria.En la siguiente figura se muestra la carga en memoria con un nivel de actividad medio de la aplicación:



Class Name - Allocated Objects	Bytes Allocated	Bytes Allocated	Objects Allocated
char[]		2.235.216 B (56,4%)	26.192 (5,3%)
byte[]		947.456 B (23,9%)	316.460 (64,3%)
int[]		403.184 B (10,2%)	62.806 (12,8%)
java.util.regex.Matcher		139.328 B (3,5%)	20.661 (4,2%)
java.lang.String		101.664 B (2,6%)	40.323 (8,2%)
java.nio.HeapCharBuffer		41.568 B (1%)	8.219 (1,7%)
java.util.Hashtable$Entry[]		7.928 B (0,2%)	51 (0%)
java.util.HashMap$Entry[]		6.688 B (0,2%)	118 (0%)
java.lang.Object[]		5.552 B (0,1%)	527 (0,1%)
java.nio.HeapByteBuffer		4.560 B (0,1%)	872 (0,2%)
java.util.HashMap$Entry		3.552 B (0,1%)	1.435 (0,3%)
java.lang.String[]		3.408 B (0,1%)	946 (0,1%)
java.lang.reflect.Field		3.096 B (0,1%)	401 (0,1%)
java.util.Hashtable$Entry		2.904 B (0,1%)	1.147 (0,2%)
java.lang.StringBuilder		2.800 B (0,1%)	1.620 (0,3%)
java.util.TreeMap$Entry		2.336 B (0,1%)	704 (0,1%)
com.mysql.jdbc.ConnectionPropertiesImpl$BooleanConnectionProperty		2.112 B (0,1%)	309 (0,1%)
java.lang.reflect.Method		1.840 B (0%)	216 (0%)
byte[][]		1.768 B (0%)	438 (0,1%)
java.lang.Integer		1.600 B (0%)	914 (0,2%)
java.net.URL		1.064 B (0%)	173 (0%)
com.mysql.jdbc.JDBC4Connection		1.040 B (0%)	3 (0%)
java.lang.Long[]		1.040 B (0%)	1 (0%)
java.lang.Short[]		1.040 B (0%)	1 (0%)
java.lang.Byte[]		1.040 B (0%)	1 (0%)
java.lang.Integer[]		1.040 B (0%)	1 (0%)
java.util.jar.JarFile$JarFileEntry		864 B (0%)	101 (0%)
java.lang.Class[]		776 B (0%)	425 (0,1%)
sun.net.www.protocol.http.HttpURLConnection		768 B (0%)	30 (0%)
java.lang.reflect.Field[]		768 B (0%)	4 (0%)
com.mysql.jdbc.ByteArrayRow		736 B (0%)	438 (0,1%)
java.util.zip.ZipEntry		672 B (0%)	101 (0%)
java.lang.StringBuffer		656 B (0%)	381 (0,1%)
java.util.ArrayList		648 B (0%)	263 (0,1%)
com.mysql.jdbc.ConnectionPropertiesImpl$StringConnectionProperty		640 B (0%)	81 (0%)
com.mysql.jdbc.Field		576 B (0%)	35 (0%)
java.util.zip.ZipFile$ZipFileInputStream		528 B (0%)	99 (0%)
java.util.zip.ZipFile$1		528 B (0%)	93 (0%)
java.lang.Character[]		528 B (0%)	1 (0%)
java.util.TimerTask[]		528 B (0%)	1 (0%)
java.util.regex.Pattern		512 B (0%)	67 (0%)
java.util.regex.Pattern$Curly		512 B (0%)	155 (0%)

FIGURA 5.3: Carga media en memoria

Sumando los tamaños parciales de cada objeto se puede ver que el tamaño ocupado en memoria total es de 3,8 MB y no supone una carga importante para la máquina servidor, por lo que se pueden mantener ejecutándose otras aplicaciones en la misma máquina sin suponer una sobrecarga importante en la memoria de la misma.

Capítulo 6

Manual de Usuario

6.1. Introducción

En este capítulo se va a presentar el manual de usuario de la aplicación mediante un ejemplo de un caso de uso real. En relación a la interfaz se ha buscado que tanto el diseño como la usabilidad estén al nivel de otras aplicaciones en el mercado para el iPhone.

En cuanto al diseño, se ha conseguido una interfaz agradable a la vista, sobria y de fácil visualización de la información. La información se presentará en formato nativo al iPhone debido a su gran versatilidad y posibilidades, además de por ser un recurso proporcionado por el SDK del terminal móvil.

En relación a la usabilidad, la mayor preocupación ha sido proveer al usuario con las mayores posibilidades de acción sin disminuir la presentación de información en pantalla. Para este fin, se han distribuido los menús en las mismas localizaciones de la pantalla que otras aplicaciones para el iPhone, minimizando así el impacto para los nuevos usuarios. Además, dichos menús son de fácil navegación y contienen acciones intuitivas y de fácil comprensión.

Con todo esto, un usuario habitual de iPhone no tendría ningún problema en adaptarse y utilizar esta aplicación sin ningún tipo de formación adicional. Sin embargo, se expondrá a continuación cómo desenvolverse y realizar una consulta en la aplicación a modo de ejemplo.

6.2. Ejemplo de caso de uso

En primer lugar se parte de que la aplicación está instalada ya en el iPhone. Posteriormente en la sección **??** se explica cómo instalar la aplicación en el terminal.

Una vez iniciada la aplicación iCartelera, aparecerá la pantalla de introducción del título. En la siguiente figura tenemos una vista de la misma:

FIGURA 6.1: Introducir el título de la película

Después de haber introducido el título y haber presionado “Aceptar” aparecerá la pantalla de carga:

FIGURA 6.2: Cargando los datos de la película

Tras unos pocos segundos se mostrará la pantalla de información de la película y el trailer en la parte inferior de la misma:

FIGURA 6.3: Información de la película

Si se pulsa en el botón "Críticas" se accederá a la lista de críticas de la película:

FIGURA 6.4: Lista de Críticas

En la pantalla de críticas se puede acceder a la página de origen de cada una simplemente pulsando en "Ver crítica":

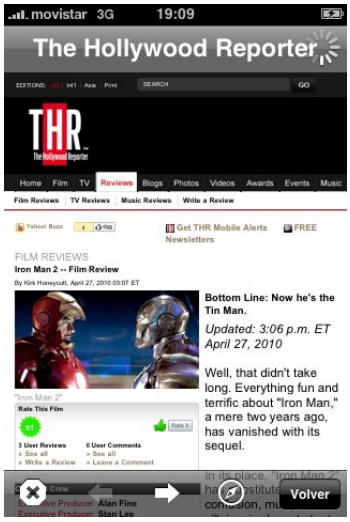


FIGURA 6.5: Página origen de la crítica

En cualquier momento, si se pulsa el botón "Volver" de la parte superior izquierda de la pantalla de críticas, se retrocederá a la página inicial dónde se muestran los datos de la película y el trailer.

Desde la página inicial si se presiona el botón “Cines” se accederá a la lista de cines ordenados por distancias según muestra la figura:

Figura 6.6: Lista de cines

También existe la posibilidad de buscar un cine en la lista introduciendo su nombre segun indica la siguiente figura:

Figura 6.7: Búsqueda de cines

Si se presiona en el nombre de un cine se accederá a la pantalla de descripción de dicho cine que contiene su nombre, los horarios de la película y

un pequeño mapa de su ubicación como se muestra a continuación:

Figura 6.8: Descripción del cine

Presionando el botón ubicado debajo del mapa que pone “Mostrar ruta” se saldrá de la aplicación y se mostrará la ruta al cine elegido mediante la aplicación de GoogleMaps. La siguiente figura lo muestra:

Figura 6.9: Salir de la aplicación para mostrar la ruta al cine

Volviendo a la página inicial si se presiona el botón “Mapa” se accederá al

mapa donde se presentarán los cines que proyectan las películas representados con chinchetas de color morado y la posición del usuario representada como una chincheta de color rojo según se muestra en la siguiente figura:

FIGURA 6.10: Mapa de cines

En la pantalla del mapa se tienen varias opciones en cuanto al tipo de mapa que se muestra, se puede mostrar el mapa de las calles, por satélite y uno híbrido. Además proporciona la posibilidad de ajustar el radio en el cual se quiere mostrar los cines mediante el deslizador de la parte inferior derecha de la pantalla. En la parte superior de la pantalla se muestra el radio actual seleccionado. También se puede desplazar el mapa y hacer zoom.

Si se pulsa en la chincheta de un cine aparecerá un resúmen de la información del mismo igual que si se hubiera seleccionado desde la lista de cines.

Tras estos sencillos pasos, toda la información necesaria para poder elegir un cine donde ver la película, saber cómo llegar a ese cine y la opinión de varios expertos referente a la película estaría a disposición del usuario.

Según se ha ido observando el proceso no entraña ninguna dificultad y requiere una mínima intervención del usuario para poder llevarse a cabo. En cuanto a la calidad de la información obtenida, se cuenta con páginas fuente de renombre tales como eCartelera, GoogleMaps o Youtube y se confía en la fiabilidad que proporcionan.

Bibliografía

[APIa] Api de googlemaps. http://code.google.com/intl/es-ES/apis/maps/.

[APIb] Api de java. http://java.sun.com/javase/6/docs/api/.

[CMP] Comparativa entre varios móviles. http://diarioandroid.com/2009/05/06/comparativa_entre_sistemas_operativos_moviles/.

[COL] Tabla de referencia de colores. http://www.kanzaki.com/docs/colortable-t.html.

[CSJa] Manual de comunicacion con sockets en java 1. http://www.mitecnologico.com/Main/ComunicacionClienteServidorSockets.

[CSJb] Manual de comunicacion con sockets en java 2. http://www.chuidiang.com/clinux/sockets/sockets_simp.php.

[EJG] Ejemplo de googlemaps. http://code.google.com/p/jposition/.

[FIP] Foro de desarrolladores de iphone. http://www.iphonedevforums.com/forum/.

[HIB] Tutorial de hibernate. http://www.laliluna.de/articles/first-hibernate-example-tutorial.html.

[HTMa] Manual de comandos html. http://www.w3.org/TR/2000/WD-DOM-Level-1-20000929/level-one-html.html.

[HTMb] Manual de html con css. http://blixt.org/articles/tabbed-navigation-using-css#section.

[JAV] Manual de objetos en html. http://www.desarrolloweb.com/manuales/20/.

[LATa] Ejemplos de plantillas en latex. http://zoonek.free.fr/LaTeX/LaTeX_samples_title/0.html.

[LATb] Ejemplos de comandos para bibliografía en latex. http://www.cs.arizona.edu/~collberg/Teaching/07.231/BibTeX/bibtex.html.

[MIP] Manuales de desarrollo para iphone. http://developer.apple.com/iphone/index.action.

[MSQ] Manual de mysql. http://netbeans.org/kb/docs/web/mysql-webapp.html.

[OCV] Página oficial del proyecto opencv. http://sourceforge.net/projects/opencv/.

[QRC] Qr code generator. http://qrcode.kaywa.com/.

[SOK] Manual de sockets para java. http://pedrorojas.over-blog.es/pages/Trabajar_Sockets_en_Java-1353613.html.

[SQL] Información sobre odbc. http://dev.mysql.com/doc/refman/5.1/en/connector-odbc-general-information.html.

[TIP] Tutoriales para iphone. http://iphone.zcentric.com/.

[TMP] Plantillas para páginas html. http://templatenavigator.com/template-2-0.html.

Índice de figuras