



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

BANDEVENT

Proyecto Final de Carrera

Ingeniería Informática

Autor: Fco. José Guerra Aroca

Directores: David de Andrés Martínez

Santiago Sánchez Alepuz

24-9-2012

Resumen

Con la realización de este proyecto vamos a ver la importancia de las nuevas tecnologías móviles en el día a día, así como los esfuerzos que realizan las empresas actualmente en proporcionar un servicio a sus clientes ó usuarios. El proyecto a realizar se centra concretamente en las necesidades de la plataforma musical Bandness en proporcionar un nuevo servicio a sus usuarios a través de los dispositivos móviles, más allá de los de su propia web.

Para la realización de la aplicación móvil, utilizaremos lenguajes web como HTML5, CSS3, PHP y JS. Esto quiere decir que será una aplicación web desarrollada para dispositivos móviles, lo cual permitirá su ejecución con independencia del dispositivo, sin tener que desarrollar una aplicación distinta dependiendo el sistema operativo del terminal.

Personalmente, es una gran oportunidad para tener un acercamiento a dichos lenguajes web, dada la poca presencia de ellos a lo largo de la carrera y su importancia hoy en día. Lo cual convierte este proyecto en una gran motivación tanto de aprendizaje, como de superación para llegar a realizar la aplicación prevista.

Palabras clave: Desarrollo móvil multiplataforma, Bandness, Smartphone, Phonegap



Tabla de contenidos

1. Introducción.....	6
2. Contexto de la aplicación.....	10
2.1. Evolución de la tecnología móvil.....	10
2.2. Sistemas operativos para Smartphone.....	15
2.3. Aplicaciones nativas Vs. Aplicaciones Web.....	19
2.4. Frameworks de desarrollo de aplicaciones híbridas.....	21
2.5. Conclusiones.....	23
3. Bandevent multiplataforma: Especificación.....	24
3.1. Modelo de negocio.....	25
3.1.1. Características de la aplicación.....	25
3.2. Diseño.....	27
3.2.1. Interfaz gráfica de usuario.....	28
3.2.2. Diagrama flujo entre pantallas.....	29
3.2.3. Servicios Públicos Aplicación.....	30
3.2.3.1. Menú principal.....	30
3.2.3.2. Bandas.....	33
3.2.3.3. Banda.....	33
3.2.3.4. Evento.....	34
3.2.3.5. Login.....	36
3.2.4. Servicios Privados Aplicación.....	37
3.2.4.1. Menú usuario.....	37
3.2.4.2. Conciertos.....	38
3.2.4.3. Diario de conciertos.....	39
3.2.4.4. Configuración.....	39
3.3. Modelo de datos.....	40
4. Implementación.....	41

4.1. Primeros pasos con Phonegap en Android.....	41
4.2. Lenguajes de programación.....	46
4.3. Peticiones servidor.....	47
4.4. Establecimiento conexión: Usuario - Servidor.....	48
4.5. Servicio de búsqueda por parámetros.....	49
4.6. Muestra de los resultados de las consultas.....	50
4.6.1. Listados.....	51
4.6.2. Rellenado de Campos.....	52
4.7. Autenticación.....	53
4.8. Almacenamiento de datos en memoria local.....	55
4.9. Tratamiento de eventos.....	55
4.9.1. Confirmar asistencia.....	56
4.9.2. Cancelar asistencia.....	57
5. Distribución y mercado.....	58
6. Conclusiones.....	60
7. Referencias.....	63



1. Introducción

Hoy en día es muy habitual encontrarse cada día con una gran cantidad de personas conectadas a las últimas tecnologías. No es nada raro, tal como podría ser hace algunos años, encontrar alguien escuchando música por la calle gracias a su MP3 o su iPod, o comprobar cómo la gente lee libros en el metro mediante los libros electrónicos (e-books), y así un sinnúmero de acciones que ahora mismo nos parecen totalmente corrientes y que ni prestamos atención pero que hasta hace algunos años no podíamos tan siquiera imaginar. Pero si hay algo de lo que se podría decir que es ya prácticamente de nuestro día a día, y que tenemos una necesidad o una utilización muy habitual, son los teléfonos móviles, pero no como los conocíamos hace unos años, que nos servían únicamente para llamar o enviar SMS, ahora estamos antes la generación siguiente, la de los Smartphone. Los conocidos como teléfonos inteligentes, construidos sobre una plataforma informática móvil, con mayores prestaciones que las de un móvil convencional [1]. Los podríamos considerar como pequeños ordenadores personales de bolsillo, con múltiples funcionalidades y aplicaciones para todos los gustos, público y al alcance de cualquiera, que se han convertido en algo del día a día para sus usuarios, como podrían ser los mensajes directos vía Internet entre dispositivos mediante Whatsapp, o la toma de fotografías y posterior publicación en redes sociales.

Según un estudio realizado por Tomi T. Ahonen Consulting en Diciembre de 2012, en España un 46 % de la población tiene un Smartphone [2]. Eso quiere decir entre otras cosas, que prácticamente la mitad de la población española a finales del año pasado poseía un Smartphone. Teniendo en cuenta el notable crecimiento y éxito de estos terminales no sería raro pensar que actualmente ese porcentaje es mucho mayor, lo cual hace ver a las claras como de arraigado esta ya hoy en día su uso.

Y no es algo de un solo sector de la población o grupo de edad si no que hasta entre lo más jóvenes está en su día a día. Tal es así que según un estudio realizado entre junio y agosto de 2011 sobre jóvenes españoles de 10 a 16 años, se comprueba que la edad en la que entran al mundo de los Smartphone es de tan solo 13 años [3]. Lo que por una parte se puede considerar un gran éxito en cuanto, por ejemplo, a la facilidad de su utilización que permite que hasta jóvenes en tan prontas edades puedan sacarle el mayor partido, por otra acerca a jóvenes de edades muy tempranas a un mundo en el cual son completamente libres. Y esto es así porque lo que esta detrás de los Smartphone, en lo que se basan, y lo que es su pilar central, no es otra cosa y no podemos nunca olvidarlo, Internet. Lo que hasta hace nada pensábamos que no podría salir de nuestros hogares o empresas, ahora incluso nuestro hijo/a de 12 ó 13 años puede utilizar en todas partes.

El tiempo avanza y con ello las tecnologías. Ahora ya no se necesita ningún cable para poder conectarse a Internet, sino que simplemente mediante una red Wi-Fi o las redes 3G cualquiera puede estar conectado. Se podría decir que a día de hoy es más difícil encontrar alguien que esté no conectado, que a alguien que lo esté. Esta libertad de conexión puede abrir múltiples discusiones sobre cómo o quién, o a qué edad, debería utilizar este tipo de tecnología, pero los tiempos cambian y no hay otra cosa que hacer que adaptarse a ellos, y en este caso, sacarles el máximo partido y todavía no se ha llegado ni mucho menos al máximo de esta.

Existe multitud de dispositivos en el mercado, cada vez más potentes, más ligeros, más rápidos, más cómodos, o simplemente con funciones más exclusivas, con su propio mercado, y usuarios. Contamos con una multitud de marcas que, para hacernos una idea de la importancia de este mercado en la actualidad todas ellas podrían considerarse grandes potencias o empresas del siglo XXI. Podríamos decir nombres como Nokia, Apple, RIM, Samsung, HTC, Motorola, SonyEricsson, LG, etc. y probablemente todo el mundo haya oído hablar de ellas. Y una vez ya dentro de cada una de las empresas podríamos encontrar un sinfín de dispositivos todos ellos con sus características que hacen que, como muestra la Figura 1, prácticamente no sea posible encontrar dos iguales por mucho que a simple vista pudiera parecerlo.



Figura 1: Diferentes Smartphone del mercado actual

Como es lógico al ser pequeños ordenadores, se debe contar con un sistema operativo el cual nos haga factible su utilización, siendo los más famosos: Android, iOS, Symbian OS, Blackberry OS y Windows Phone. Esto hace que un móvil no solo sea tenido en cuenta por sus prestaciones técnicas, sino también por el sistema operativo (S.O.) o que lo gestiona. Este es un detalle muy importante hoy en día ya que como todo, no cabe olvidar que esta tecnología para las empresas que los desarrollan es un negocio, y como tal, intentan siempre estar por encima de sus competidores, y sobretodo diferenciarse de ellos. Es muy importante destacar que las aplicaciones desarrolladas para terminales que posean un determinado sistema operativo, como Android, no podrían ser ejecutadas en terminales que dispongan de otro sistema operativo, como iOS.

Si nos paramos en este punto se podría pensar que desde el punto de vista de alguien que quisiera hacer una aplicación para todos y cada uno de los sistemas operativos, se tendría que crear una versión por cada uno de ellos lo cual sería un gran problema ya que habría que hacer una gran cantidad aplicaciones, cada una de ellas en un lenguaje y con unas especificaciones técnicas distintas. Por suerte esto no es así. Existe un lenguaje comúnmente conocido que es el que nos ayudará en este caso a ser capaces de crear una aplicación que funcione independientemente de su S.O. ya que como se ha comentado, los Smartphone tienen su pilar central en Internet y, por tanto, el lenguaje al que nos referimos no es otro que el HTML [4]. Todos y cada uno de los terminales deben ser capaces de leer este lenguaje de creación de páginas web, ya que si no, serían incapaces de leer correctamente las páginas, y eso sí que es algo que ninguno de ellos puede permitirse.

En el caso del proyecto a realizar, está centrado en las necesidades y ámbito de negocio de Bandness [5], una empresa de reciente creación que busca abrirse paso como plataforma

musical referente. Entre sus usuarios actualmente hay una gran cantidad de bandas independientes, muchas de ellos que están empezando o llevan una corta trayectoria, a las cuales mediante la plataforma se les ofrece la posibilidad de darse a conocer al público, encontrar lugares en los cuales poder tocar, o incluso encontrar nuevos miembros para la banda. Pero también es una plataforma para que los usuarios tengan una interacción directa con sus bandas, sin intermediarios, que tenga la posibilidad de escuchar nueva música que les sería prácticamente imposible encontrar de otra manera, y hasta entablar relaciones personales como en cualquier otra plataforma, en este caso a través de un factor común, la música. Es por ello que uno de los factores claves de dicha plataforma y que más se persigue desarrolla y mejorar para hacer más cómoda la relación entre ambos tipos de usuarios, es la interacción y conexión entre los mismos. De esta forma, aparece la necesidad de seguir incrementando estas comodidades y una de las maneras más directas y que mayor aceptación tienen hoy en día es el uso de las aplicaciones móviles, pero si se busca comodidad y darle facilidades al usuario, debe ser una aplicación en la cual no importe el dispositivo móvil que este tenga, y que pueda ser lanzada en Smartphone sin importar, si es un Android, o es un iOS. Otra cosa que no hay que olvidar es que gracias a las aplicaciones móviles, se ofrece la posibilidad de interconectar a sus usuarios en cualquier momento y lugar con sus Smartphone, lo cual es una comodidad añadida que da aún más motivos para iniciar el desarrollo de una aplicación móvil.

Partiendo de esta idea de realizar algo al servicio de los usuarios, a la cual puedan acceder todos y cada uno de ellos, lo que se busca es la creación de una aplicación que pueda ser ejecutada con independencia del Smartphone, lo que se conoce como una aplicación multiplataforma, utilizando todos los recursos disponibles en los lenguajes de desarrollo (HTML5, CSS3) para obtener una calidad en la aplicación, sobretodo visual y en cuanto a la carga de datos, propia de una aplicación nativa del S.O., de esta manera si se logra crear algo con una calidad notable, es posible que la propia aplicación sea la que capte nuevos usuarios para la plataforma, lo cual hace que aparte de dar comodidades a los usuarios de la plataforma, la aplicación tenga también el objetivo de expansión de la misma y publicitarla, ya que si la aplicación capta la atención de un cliente que se la ha descargado, es muy posible que lo siguiente que haga sea acceder a la propia plataforma y porque no, registrarse en ella.

Pero además de las necesidades de la propia empresa donde se desarrolla la aplicación hay una parte muy importante de mi propia necesidad y voluntad de aprendizaje en el desarrollo de aplicaciones móviles, por el grado de interés propio en la materia, en su desarrollo y ser capaz de crear aplicaciones que finalmente sean publicados en los distintos markets de las compañías y las cuales puedan disfrutar usuarios de todas partes del mundo. A lo largo del curso he podido disfrutar y realizar un aprendizaje en la asignatura de ADM, de la programación la plataforma Android, la cual fue mi primera toma de contacto con dicho mundo, y que me ha hecho decidirme en que sean las aplicaciones y el desarrollo móvil lo que desee hacer y dedicarme en el futuro. Esta aplicación por su parte, es un punto de partida en el desarrollo de aplicaciones multiplataforma de donde espero que salgan a posteriori grandes cosas dentro del mismo ámbito pues a medida que he ido tomando contacto con el desarrollo multiplataforma, he podido observar las grandes posibilidades que tiene, así como la gran cantidad de margen de mejora y desarrollo que existe en él.

En resumen, si unimos el enorme por venir de la empresa responsable del proyecto, Bandness, la cual día a día no hace más que avanzar en sus objetivos y lograrse un nombre dentro del complicado mundo de la música. Todo ello con trabajo duro, y aprovechando al máximo lo que las nuevas tecnologías ofrecen, tratando siempre de ofrecer lo mejor a sus

usuarios. Porque al final los usuarios son la clave del éxito, si no tienes gente que te siga, a la que le guste tu proyecto y que esté interesada en él, por muy bueno que sea, no logrará llegar a puerto, de tal forma que si por algo hay que trabajar, y desde Bandness lo tienen claro, es por los usuarios.

Y que mejor manera de trabajar para ellos, que dándoles un servicio tan práctico y cómodo como el de Bandevent en sus dispositivos móviles. Que tengan presente que siempre se está para ayudarles y tratar cada vez de hacerlo mejor. Si además del trabajo duro y con un objetivo claro, tenemos en cuenta el enorme potencial de este tipo de programación en cuanto a los resultados que se pueden obtener, así como las múltiples funcionalidades que puede ofrecer en prácticamente cualquier Smartphone, obtenemos un elemento muy potente.

Esto es solo el principio, y desde este punto, se irán proporcionando una serie de servicios nuevos, que den al usuario prácticamente todo aquello que desea de una plataforma musical al alcance de la mano. Este es el objetivo de Bandness crecer día a día, pero siempre teniendo claro, que lo más importante aquí, son las personas a las que va dirigida, los músicos, y los amantes de la música que prácticamente, somos todos.



2. Contexto de la aplicación

Antes de empezar a explicar el proyecto que vamos a realizar, con su diseño, implementación, funcionalidades, y todo aquello que lo describe, hay que ponerse en situación y ver como se ha llegado hasta tener la opción de poder realizar lo que finalmente mostraremos y se ha creado. En primer lugar, de donde ha surgido esta gran tecnología de la que se dispone hoy en día, cuál fue su origen y como ha ido evolucionando a lo largo de estos años hasta lo que nos encontramos ahora. También es importante que analicemos el mercado actual de la telefonía móvil, su situación actual, las grandes empresas, y los S.O. que se utilizan en cada uno de los terminales para hacernos una idea de donde sería más conveniente programar, en cual alcanzaríamos a un mayor público y cómo afectaría en general la elección de un S.O. a la aplicación que tenemos prevista desarrollar y las grandes diferencias que se presentan entre unos y otros. Una vez visto esto, veremos también que no todo son aplicaciones exclusivas para cada uno de los S.O., si no que existen otro tipo de ellas, que pueden ser usadas todos y cada uno de ellos por igual, lo que se conoce como aplicaciones multiplataforma, y nos introduciremos en el framework Phonegap, para conocer como son y sus ventajas e inconvenientes frente a las aplicaciones nativas.

2.1. Evolución de la tecnología móvil

Podríamos decir que la telefonía móvil, empezó en los años 70 de la mano de Martin Cooper, que introdujo en 1973 el primer radioteléfono mientras trabajaba en la empresa Motorola (ver Figura 2) conocido como el pionero de esta tecnología que hoy, más de 40 años después aún está en plena evolución y con un amplio horizonte por delante. Pero sería justo decir que no fue hasta prácticamente 1979 cuando aparecería el primer sistema comercial en Tokio (Japón), por la compañía NTT [6]. Tras esto la telefonía móvil se fue extendiendo, primero en los países nórdicos, y luego dentro de los propios Estados Unidos. Y desde entonces esta tecnología no ha hecho más que evolucionar y extenderse, gracias a su gran demanda y aceptación en el mercado hasta lo que vemos hoy en día.



Figura 2. MicroTAC 9800X de Motorola, considerado el primer teléfono móvil

La evolución de la telefonía móvil podríamos separarlas en distintas etapas, o como se las conoce comúnmente, generaciones.

La primera generación 1G: Los primeros teléfonos móviles (1979-1989)

En esta primera generación la señal está basada en sistemas de transmisión analógicos, y los dispositivos suelen ser más ligeros (aunque todavía su tamaño es considerable, debido a las baterías que llevaban acopladas para poder funcionar de manera inalámbrica) y menos costosos que sus antecesores. Además durante esta etapa se implementan estándares como el Sistema Telefónico Móvil Avanzado (Advanced Mobile Phone System), o el Sistema de Comunicación de Acceso Total (Total Access Communications System). Durante esta época este mercado creció entorno a un 30 y 50 % anualmente [7].



Figura 3. Ericsson Hotline del año 1988

La segunda generación 2G: Redes GSM y GPRS (1989 – 2002)

Su principal diferencia es que entramos en el mundo digital. Se crean protocolos más sofisticados que son los utilizados aún hoy en día, entre los más destacados encontramos:

- El GSM (Global System for Mobile Communications), donde se utiliza la modulación digital para mejorar la calidad de la transmisión de voz.
- El IS-136, es un sistema dual de comunicaciones que permite definir en un solo slot de tiempo tanto la frecuencia como el canal de voz, lo cual hace que aumente el número de llamadas que se pueden realizar al compartir una frecuencia de transmisión [8].
- El CDMA (Code division multiple access), un estándar en el cual para diferenciar a los distintos usuarios de la red en vez de utilizar frecuencias separadas se utilizan números digitales únicos [9].
- Y finalmente podríamos destacar el PDC (Personal Digital Cellular), una tecnología que divide cada canal en ranuras de tiempo individuales para incrementar la cantidad de datos que pueden ser transportados [10].

Además por primera vez se ofrecen servicios nuevos además de la transmisión de voz, como son el de fax o el SMS (Short Message Service). Dentro de esta generación podríamos incluir la 2.5 donde aparecería el GPRS (General Packet Radio System), una técnica de conmutación de paquetes integrable con el GSM que aumentó considerablemente las velocidades de transmisión, además es capaz de aprovechar diversos canales de y utilizar los huecos disponibles para las transmisiones [11]. En este caso fue Motorola la que en 1989 lanzó el teléfono móvil más pequeño hasta la fecha, iniciando así la era del 2G, se trataba del MicroTAC [12].



Figura 4. Motorola MicroTAC del año 1989

Tres años después sería Nokia la que en este caso sacaría el 1011, el primer móvil que dispuso de GSM.



Figura 5. Nokia 1011 del año 1992

En 1997, Nokia también lanzaría el primer Smartphone, con una CPU Intel 386 y 8MB de RAM el cual se llamó Nokia 9000i. Permitía el envío de fax, SMS, y emails, pudiendo acceder a Internet a través de mensajes SMS.



Figura 6. El primer Smartphone de Nokia, impulsado por una CPU Intel 386

En el año 1999 Internet móvil ya era una realidad, y con ella las conexiones WAP (Wireless Application Protocol) que aparecería por primera vez en el Nokia 710.



Figura 7. Nokia 710, el primero con protocolo WAP para el acceso inalámbrico

Años después, en 2002 aparecería el primer modelo de un móvil muy conocido en la actualidad, sobretodo entre los más jóvenes gracias a su cómodo diseño y utilización, no es otro que las BlackBerry. La compañía que lanzo estos terminales al mercados es RIM. En este primer modelo ya se aprecia el teclado tan característico de estos terminales, además de disponer de funciones de agenda persona y soporte de email push.

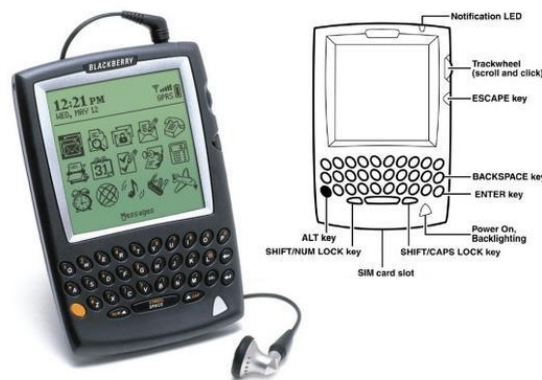


Figura 8. Primera BlackBerry del mercado, creada en 2002.

En esos años de la generación 2G también fueron surgiendo infinidad de dispositivos que incluyeron por ejemplo cámara integrada, el diseño *clamshell* con una gran pantalla para el móvil, los teclados planos, y un largo etcétera.

La tercera generación 3G: Internet en el móvil (2002 – Presente)

En esta generación es donde por fin a los teléfonos móviles se les da acceso inalámbrico a internet con una elevada tasa de transmisión de datos, aparecen aplicaciones multimedia (audio, imágenes y video) y aumentan considerablemente las transmisiones de datos, lo que se conoce como UMTS (Universal Mobile Telecommunications System). Se busca estandarizar las redes en un único protocolo global dejando atrás los utilizados en distintos lugares como Europa, Estados Unidos, etc. Esto es lo que se conoce como IMT-2000, que no es otra cosa que un estándar mundial para redes inalámbricas que permite conectar diversos sistemas de redes terrestres y/o por satélite [13]. En un primer momento, bajo condiciones óptimas esta red podría llegar a ofrecer velocidades de hasta 2Mbps mediante la tecnología W-CDMA. Actualmente esta velocidad ha conseguido aumentar hasta los 14Mbps, que se conoce como tecnología HSDPA (High-Speed Downlink Packet Access), 3.5G o 3G+. En el tiempo presente esta generación 3.5G es en la que actualmente nos encontramos en nuestros Smartphone la mayoría de los usuarios. Pese a que esta generación empezó en 2002, no fue hasta el año 2007 donde en España empezaron a aparecer los primeros móviles que marcaron un antes y un después entre el 2G y el 3G (estando más bien anteriormente en una etapa 2.5G). El Smartphone que fue pionero de esta generación en España es el ya famoso iPhone de Apple, con él llegó la pantalla táctil a los teléfonos (o multitáctil en este caso), la sencillez y comodidad del acceso a Internet a través del dispositivo y una gran capacidad multimedia.



Figura 9. El primer iPhone, una autentica revolución, en 2007

A partir de entonces los dispositivos no han hecho más que evolucionar un nivel increíble. Gran parte de esa evolución se debe también a la última gran compañía en entrar a este mundo de las tecnologías móviles, estamos hablando de Google, que en 2008 sacaría el HTC Dream con su sistema Android.



Figura 10. HTC Dream, el primer móvil con sistema Android del mercado.

La cuarta generación 4G: El futuro inmediato

Con esta cuarta generación se pretenden alcanzar tasas de transmisión de hasta 20 Mbps gracias al uso de las características de Calidad de Servicio (Quality of Service), lo que permitirá ajustar el tráfico de datos y priorizar dependientemente del tipo de aplicación ajustándose en tiempo real. Se pretende que esté basado en el protocolo IP (Internet Protocol) y donde la transmisión de paquetes sea enteramente por conmutación IP. Se busca alcanzar velocidades de hasta 100Mbps en cualquier lugar. Esto es algo que será de gran ayuda en mejorar las video conferencias (o streaming), dado que las velocidades actuales no permiten un funcionamiento perfecto al no disponerse de ancho de banda suficiente.

Tal y como se ha podido ver, la evolución de las tecnologías de comunicación hace posible disfrutar de Internet móvil en los dispositivos actuales, en cualquier momento y en cualquier lugar. Esto ha abierto la puerta a una amplia variedad de aplicaciones, inimaginables años atrás, que aprovechan estas capacidades de comunicación para proporcionar información en tiempo real a los usuarios, mensajería instantánea y un largo etcétera donde el único límite lo fija la imaginación de los desarrolladores y no ya la tecnología disponible. Así, esta evolución nos permitirá aceptar el desafío de desarrollar una aplicación para poder interconectar usuarios en distintas partes del país o del mundo, en una misma plataforma gracias a nuestra aplicación. Persona que hasta hace años estaban separadas por miles de kilómetros, y que no tenían opción de comunicarse o tener relación entre ellas ni a nadie se le hubiera pasado por la cabeza, ahora gracias a los avances de las tecnologías móviles es algo al alcance de la mano y que trataremos de conseguir. Sin embargo, la amplia variedad de dispositivos, fabricantes, y sistemas operativos, provocan una gran fragmentación, tanto en el mercado de dispositivos como en el desarrollo de aplicaciones. Por ello, es muy importante, determinar qué sistemas operativos existen actualmente y sus características principales, a fin de seleccionar el idóneo para la aplicación que deseamos desarrollar o, en su caso, determinar posibles alternativas para el desarrollo. La siguiente sección aborda esta temática.

2.2. Sistemas operativos para Smartphone

Como se ha comentado, los Smartphone nacieron a partir de la generación 2G aunque no será hasta la 3G donde comenzarán su gran expansión en el mercado, así como también una

gran cantidad de aplicaciones y utilidades. Empezaremos a acostumbrarnos a términos como widgets, que no son otra cosa que pequeñas aplicaciones o programas cuyo fin es el dar facilidad de acceso a funciones frecuentemente usadas y proveernos de una información visual de las mismas, o a los markets, que son las tiendas virtuales que cada uno de los S.O. tiene para comercializar sus aplicaciones.

Los Smartphone, pese a sus pocos años en el mercado, ya son mundialmente utilizados existiendo multitud de dispositivos según la compañía en la cual nos fijemos. Pero detrás de cada Smartphone hay un Sistema Operativo, la clave del funcionamiento del dispositivo, lo que marca la diferencia entre unos y otros, más allá de sus características técnicas, su peso, diseño, etc. Y bien lo saben las compañías, pues estos sistemas operativos están en constante desarrollo y evolución sacando actualizaciones y mejoras constantemente. Aunque hay muchos en el mercado, cinco de ellos son los predominantes dentro de la telefonía móvil hoy en día:

Android OS:

El sistema operativo de Google, que pese a prácticamente ser el último en aparecer en el mercado allá por 2008, hoy en día es el que posee una mayor cuota de mercado con cerca de un 68,1 % [13]. Su éxito es debido en gran medida a deberse a un sistema de distribución libre y código abierto. Para muestra de su gran éxito, en el primer trimestre de 2012 se vendieron cerca de 89.9 millones de terminales con dicho S.O. en todo el mundo [14]. Sus características técnicas principales podrían ser su pantalla con soporte multitáctil, además de adaptabilidad a gráficos 2D y 3D. Utiliza tecnologías de conectividad 2G 3G y 4G, que permiten a los usuarios una total integración de su Smartphone con los servicios de Google (tales como GMail, GTalk, Google play, etc.). Su interfaz es moderna y cómoda, permitiendo una gran personalización gracias a la gran cantidad de widgets y su market (ó store) es probablemente el más grande en cuanto a aplicaciones y herramientas para descargar.

Para el desarrollo de aplicaciones se utiliza lo que se conoce como Android SDK (Android Software Development Kit), siendo su lenguaje de programación predominante JAVA. El hecho de ser un S.O. de código abierto también hace que el propio S.O. sea usualmente modificado por usuarios de manera propia personalizándolo aún más de lo que los métodos estándar provistos por el fabricante de cada terminal móvil permiten. Existen dos acciones muy usuales dentro de los usuarios de estos dispositivos. La primera de ellas se conoce como *rootear*, que no es otra cosa que la liberalización total del S.O. al usuario del móvil obteniendo él mismo, permisos de administrador en el sistema con los cuales tener total libertad para modificar cualquier tipo de parámetro existente en el móvil. Una vez el usuario tiene totales permisos sobre el S.O. es capaz incluso de cambiar el propio S.O. por otra versión más personalizada del mismo, esta segunda acción es lo que se conoce como *modificar la ROM*.

Esta facilidad de modificación de código y libertad en el mismo hacen que existan una grandísima cantidad de desarrolladores, ya que toda la información es libre lo cual permite que cualquier persona con la dedicación adecuada pueda llegar a ser un total experto en la materia y un gran desarrollador. Otra cosa muy a tener en cuenta, es que una gran parte de sus aplicaciones resultan de carácter gratuito y al alcance de todo el mundo sin pagar nada a cambio por ellas. Esto es sustentado principalmente porque muchas de ellas optan por el método de ganar dinero a través de la publicidad que se inserta en ellas, pero no por el hecho de hacer pagar al usuario que las descarga. Pero no todo son ventajas y virtudes en este sistema, existe un problema principal conocido como *Fragmentación*. Cada dispositivo posee unas características

técnicas específicas, y funcionan con una versión concreta del S.O. El problema se origina ya que no todas las versiones del S.O., funcionan en todos y cada uno de los dispositivos, esto quiere decir que probablemente una versión recién sacada al mercado no vaya a funcionar o no sea posible de instalar en móviles cuyas prestaciones sean altas o que lleven más de 2 años en el mercado. Esto hace que alguien con un terminal de hace 3 años, se quede estancado en una versión ya obsoleta del S.O. y sin poder actualizar a una versión superior ya que no le es permitido por las características técnicas de su dispositivo [16]. De esta manera solo los teléfonos más actuales son actualizados mientras que lo más antiguos quedan olvidados de estas actualizaciones del sistema. Y esto se convierte también en un gran problema a la hora del desarrollo de aplicaciones, puesto que cada actualización introduce nuevos elementos o mejoras en el sistema y, si una aplicación se desarrolla con características propias de las últimas versiones, no podrá ser ejecutado en terminales con una versión inferior para la cual se ha realizado la aplicación. Así, muchos teléfonos no podrá disfrutar de dicha aplicación, aunque quizás si que pudieran ejecutarla técnicamente, su versión no se lo permite. Pese a ello, algo muy común en la programación para este S.O. es el desarrollo para la versión mas inferior posible en la cual sea posible ejecutar la aplicación, de tal manera que el número de versiones que pueda ejecutar la aplicación sea lo más elevado posible extendiendo así los terminales que podrían disfrutar de ella.

iOS:

Si Android es el S.O. libre por excelencia, iOS es todo lo contrario, muy centrado en la seguridad que consigna a sus usuarios y con un código cerrado. Desarrollado por Apple, la empresa pionera en los Smartphone gracias al iPhone que llegó a España en 2007, posee una cuota de mercado del 16.9 %. Es el utilizado en los exclusivos y apreciados iPad, iPod y el ya nombrado iPhone, y en el primer cuatrimestre del año vendió 35.1 millones de equipos. Un número inferior al de Android cierto es, pero al ser modelos más caros y sus aplicaciones ser una gran cantidad de ellas de pago, hacen que este sistema mueva mucha más cantidad de dinero que Android, pese a tener una cuota y una cantidad de dispositivos mucho menor. Es probablemente algo superior en cuanto a rapidez que el sistema Android y con una interfaz gráfica excelente y también superior a la de su rival. También cuenta con aplicaciones propias del sistema operativo como son Safari, iTunes, iBooks, Game Center y etc. Está previsto que en próximas fechas salga la versión 6.0 del sistema operativo.

El Kit de desarrollo ó SDK fue liberado en 2007 aunque al público general no llegaría hasta 2008. Pero como se ha comentado iOS mueve mucho más dinero que Android y para muestra un dato y es que para poder llegar a comercializar una aplicación realizada para este sistema, el usuario debe pagar una cuota de asociación mínima, de 99 \$ al año si quiere poder formar parte de lo que se conoce como el iOS Developer Program [17].

BlackBerry OS:

El sistema operativo de la empresa canadiense RIM, y usado en todos los teléfonos BlackBerry es el tercer sistema operativo en cuanto a cuota de mercado con un 6.8 %. Dado que este S.O. es único y exclusivo de dichos teléfonos una cuota de mercado inferior implica a su vez, una caída del número de ventas de dispositivos BlackBerry. No obstante todavía cuenta con la predilección de un sector que aprecia su seguridad, no en vano este S.O. se apoya en los sistemas de encriptado múltiple para reafirmar esa confianza en la seguridad. Su última versión tiene la ventaja sobre otros S.O. de ser compatible con todas y cada una de las ediciones pasadas



del sistema. En cuanto a sus puntos fuertes cabe destacar los aspectos de comunicación cuya punta de lanza sería el BlackBerry Messenger como aplicación de mensajería instantánea entre dispositivos BlackBerry. Además el diseño del propio dispositivo esta adaptado para hacer aún más cómodo este servicio de mensajería ya que el teléfono cuenta con un teclado físico al contrario que la gran mayoría de terminales nuevos del mercado que se decantan por una pantalla táctil sin teclado físico. Esta opción le hace conservar una gran cantidad de usuarios sobre todo entre los jóvenes, que lo ven como el Smartphone perfecto para conversar con sus amistades en cualquier lugar de forma cómoda y sencilla. De hecho en España casi el 30 % de los usuarios son jóvenes de menos de 24 años, y el 59 % se encuentra por debajo de los 34 años [19].

En el ámbito de las aplicaciones BlackBerry no está a la altura en cuanto a cantidad de ellas como las que encontramos en Android o iOS. Pese a ello también existen métodos y Kits de desarrollo de aplicaciones que se realizan, al igual que en Android, en lenguaje JAVA, además cuenta con servicios de ayuda al usuario y herramientas para el desarrollo de las aplicaciones [20].

Symbian OS:

El gran afectado por el incremento de Android y iOS en los últimos años, y cuya cuota de mercado pese a todavía ser la cuarta con un 6.4% no hace más que descender en los últimos meses. Se sigue utilizando en teléfonos de compañías punteras como Nokia, Sony Ericsson, Samsung, Lenovo, LG, Motorola, Siemens y Sharp. Pese a que no puede competir en el terreno de las aplicaciones con respecto a Android y iOS, sí que es cierto que su capacidad de ahorro de memoria así como procesador, hacen que la duración de su batería sea superior, un punto muy importante ya que si de algo carecen todos los Smartphone es de baterías de larga duración como en épocas posteriores. Pese a no tener la amplitud que tienen los demás S.O., tiene una calidad excelente en las funciones de telefonía y multimedia, permite múltiples conexiones y es muy seguro en el manejo de datos.

Al igual que Android, Symbian es un sistema de código abierto, y a su vez, Symbian también tiene un problema similar al de la fragmentación en Android, y es que cada dispositivo con Symbian presenta unas características específicas principalmente en lo que se refiere a la interfaz de usuario. De hecho el propio Symbian define una serie de plataformas de interfaz de usuario para definir familias de móviles según S.O. y sus características de IU. Por lo general cada versión consiste en un conjunto de APIs que dan acceso a las funciones de las que dispone el móvil, por lo cual hay que escoger la correcta que necesitemos en cada uno de los casos para lo que deseamos desarrollar [18].

Windows Phone:

Mientras que tanto Symbian como BlackBerry bajan su cuota de mercado, en este caso nos encontramos con que Windows Phone consigue aumentar con respecto a tiempos pasado y va aumentando hasta llegar al 3.5 % (en el primer trimestre su cuota creció 1.3 puntos pasando del 2.2 % al 3.5 % de este trimestre). No hay que olvidar que detrás de este S.O. se encuentra probablemente el mayor gigante de la informática hoy en día junto a Apple en equipos informáticos, que no es otro que Microsoft. Ni que decir tiene que dicho S.O. es compatible con todos los programas y productos de Microsoft, además ofrece una interfaz de usuario muy atractiva así como también para organizar los temas de trabajo. Microsoft está haciendo un gran esfuerzo en dicho S.O. dotándolo de mejoras constantes para intentar llegar a la par de sus

competidores más importantes, Android y iOS, aunque todavía le quede un largo camino que recorrer sobretodo en cuanto a cuota de mercado se refiere.

En este caso, tal como nos pasaba en iOS nos volvemos a encontrar con un sistema de código cerrado, por lo cual sus modificaciones son inviables para los usuarios. En el desarrollo de aplicaciones se utiliza el lenguaje de programación propio de Microsoft, el C#, y se cuenta también con un gran KIT de desarrollo como en las demás compañías así como una web con todo lo necesario, herramientas, manuales, etc. para los que se quieran iniciar y animar a crear sus propias aplicaciones [21]. Aunque al igual que Symbian y BlackBerry, el número de aplicaciones existentes y el grado de implicación de los desarrolladores en cuanto a crear nuevas para estos tres S.O., todavía dista mucho de llegar a Android ó iOS, pese a ello día a día se va incrementado en este caso, y quizás con la próxima actualización del S.O., el Windows phone 8, la diferencia entre ellos se reduzca.

Como vemos hay una gran diferencia en cuanto a las facilidades que nos dan las compañías a la hora de desarrollar en sus S.O., mientras que algunas como Android ofrecen al desarrollador un código abierto con el cual poder trabajar y hacer lo que desee hasta modificar el propio sistema y personalizarlo o hasta mejorarlo, otras como iOS o Blackberry no son tan permisivas con ello, aunque eso mismo haga por otra parte que la seguridad de sus S.O. sea de gran nivel y atraiga a su vez muchos usuarios por la confianza que depara. Aunque aquí se habla de los más importantes existen otros S.O., como por ejemplo Linux que también son utilizados en terminales móviles, pese a que su cuota de mercado es muy inferior. Prácticamente esto hace ver que una misma aplicación debería ser desarrollada 6 ó 7 veces, una prácticamente para cada S.O., para poder llegar a ocupar el 100 % de la cuota de mercado existente. Pero como vamos a ver a continuación hay otra manera de poder abarcar prácticamente ese 100 %, sin tener que hacer distintas versiones para una misma aplicación.

2.3. Aplicaciones nativas Vs. Aplicaciones Web

Cada uno de los S.O. tiene unos métodos de desarrollo y habitualmente un lenguaje o metodología distinta a la hora de programar aplicaciones, lo que podría servir para un sistema iOS no funcionará en un Symbian. Este tipo de aplicaciones generadas específicamente para cada uno de los sistemas es lo que se denominan: *Aplicaciones nativas*. Sus características principales serían:

- Permiten usar la plena capacidad de los elementos presentes en el dispositivo móvil a través del S.O. como puede ser la cámara, el acelerómetro, GPS, audio, video, imágenes, etc.
- Son distribuidas y comercializadas a través de los “Stores” de cada una de las compañías, dando unas grandes posibilidades de marketing y branding, así como poderse realizar actualizaciones de las mismas aplicaciones a través de la Store.
- No necesitan conexión a Internet para ser ejecutadas, ya que no se ejecutan en el navegador del dispositivo.



- Permiten el envío de Push Notifications, para mantener al usuario de la aplicación informado de noticias o información sobre posibles actualizaciones a través de la Store.
- Las aplicaciones puede ser comercializadas de manera gratuita o mediante pago por uso dentro de las stores.
- La realización de compras dentro de una aplicación se realiza a través de la cuenta del usuario asociada al dispositivo donde se ejecuta la aplicación de tal manera que no es necesario facilitar tarjeta de crédito al realizar una compra.
- La sincronización o el almacenamiento de datos en la cache del móvil, al funcionar off-line permite una ejecución veloz de la interfaz de la aplicación, superior a la de una aplicación Web, cuya interfaz también necesita conexión a Internet para ser cargada.

Por otra parte tenemos las *Aplicaciones Web*, las cuales son multiplataforma y que permiten ser ejecutadas generalmente en cualquier dispositivo. Sus características generales serían:

- Se utilizan lenguajes de programación como HTML5, JS o CSS, mucho más sencillos que otros lenguajes como podrían ser JAVA o C#, lo que permite que usuarios menos expertos en la programación puedan realizar sus propias aplicaciones, de esta manera este tipo de programación esta más extendida.
- No utilizan características nativas del teléfono como puedan ser la cámara, el GPS, acelerómetro, etc.
- No pueden ser vendidas en las stores de las compañías, por lo cual se necesita una campaña externa de marketing si se desea promocionar y que lleguen a ser conocidas para los usuarios. Además se necesitan recursos para el propio sistema de distribución de la aplicación como servidores donde el usuario pueda acceder para descargar la aplicación.
- No están sometidos a los controles de los canales de distribución de las operadoras y fabricantes, de tal manera que no hay revisión sobre las actualizaciones y versiones de las aplicaciones y llegan de manera automática a todos los usuarios y plataformas a la vez.
- Al no estar sometidos a políticas de comercialización en las Stores, existe una mayor libertad de precios sobre la distribución de contenidos desde la propia aplicación (in-App), evitando así los rangos fijos establecidos por algunas compañías. Así como también modificar el método de pago que se utilizará para compras in-App evitando así las tasa de hasta un 30 % establecidas en las Stores.

Existe también lo que se denominan *Aplicaciones Híbridas*, que mezclan elementos de las aplicaciones Web con otros de las nativas. Suelen ser aplicaciones nativas con HTML incrustado o también aplicaciones generadas mediante un framework de desarrollo común,

donde utilizando tecnologías web (como HTML, JavaScript y CSS), se genera una aplicación multiplataforma y que además permite utilizar las funciones del teléfono (uno de los grandes inconvenientes de las aplicaciones Web, que no permiten la utilización de dichos elementos). Dichas aplicaciones pueden ser comprimidas como nativas de tal manera que pueden ser comercializadas dentro de los markets de cada una de las compañías, teniendo un código base prácticamente común entre todas ellas. Aunque está claro al utilizar elementos Web habrá secciones determinadas que serán cargadas desde servidores lo cual hará que sean algo más lentas que las aplicaciones propiamente nativas.

Se ve a las claras, que si lo que buscamos es un resultado de gran calidad técnica, nuestra opción sería hacer una aplicación nativa para un S.O. y luego ir desarrollando la misma idea en los distintos S.O. restantes. Pero hay que plantearse, si lo que deseamos es un producto ideal con un alto coste de recursos y de trabajo a sus espaldas, o un producto notable, cuyo lenguaje de desarrollo no deja de evolucionar y tener cada vez más recursos y mayor calidad que además nos permite centrar todos nuestros esfuerzos en un único código que además vamos a poder hacer llegar a todos los usuarios. Tanto Bandness, como yo mismo, tenemos claro, que de igual manera que dicha plataforma apuesta por el futuro de la música, en este caso nosotros apostaremos por el que creemos será el método de desarrollo en un futuro no tan lejano, que no es otro que el de las aplicaciones híbridas.

Obviamente, detrás de esta decisión tiene que haber un soporte en el cual apoyarse, en el cual trabajar, y que nos de plena confianza en que la aplicación que vamos a realizar, tiene los resultados deseados. Para ello, pese a haber varias opciones en el mercado actualmente, todas ellas de gran calidad, cada una con sus ventajas, hemos elegido utilizar *Phonegap*.

2.4. Frameworks de desarrollo de aplicaciones híbridas

Existen en el mercado tecnologías como Titanium Mobile [24], Adobe Air [25], Corona SDK [26] ó Phonegap [22], para desarrollar aplicaciones móviles nativas usando JavaScript y lenguajes Web como HTML/CSS. Por ejemplo Titanium Mobile presenta un ambiente y un resultado más nativo que las otras, Adobe Air a la vez que permite la creación de aplicaciones para móviles también permite de la creación de aplicaciones para escritorio (o de propósito general), y Corona SDK utiliza el lenguaje de programación lua, un lenguaje imperativo, estructurado, bastante ligero y que es suficientemente compacto como para usarse en diferentes plataformas, esto es que el mismo código o modificados mínimamente, podría ser ejecutado tanto por ejemplo en un terminal Android, como en una plataforma de videoconsolas.

Así que, si Titanium da un aspecto más nativo de la aplicación, Adobe Air permite la creación de aplicaciones para más aspectos que la telefonía móvil, y corona SDK es un lenguaje simple y que sin modificación alguna funciona en muchos dispositivos, por qué elegir Phonegap. La respuesta es que Phonegap es el framework que tiene el mayor soporte de plataformas móviles en el mercado, o lo que es lo mismo tenemos la seguridad de que un código generado mediante Phonegap a día de hoy, funcionará en cualquier dispositivo iOS, Android, Symbian, BlackBerry ó Windows Phone solamente con el propio hardware del dispositivo móvil permita la ejecución de las acciones que solicite la aplicación, si por ejemplo la aplicación utiliza la cámara y el móvil no tuviera, la aplicación no podría instalarse.



Lo que se propuso desde un principio es que la aplicación a generar alcanzara al mayor mercado posible y eso es precisamente lo que Phonegap nos proporciona, además pese a que la calidad del producto no sea la más nativa de todas las alternativas, la relación de cuota de mercado que podemos alcanzar compensa, y si a ello sumamos las enormes posibilidades que se abren gracias a HTML5 / CSS3 para crear una aplicación de gran nivel y que no tenga nada que envidiar a la calidad de una aplicación nativa, podemos estar seguros de haber tomado la decisión correcta respecto al framework a utilizar.

Phonegap no es otra cosa que un framework OpenSource, basado en JavaScript, que permite crear aplicaciones mediante HTML5 y CSS3, capaz de ser ejecutado nativamente en los diferentes sistemas operativos móviles [22]. En la actualidad es soportada por los sistemas Android, iOS, BlackBerry, Windows Phone y Symbian. Además permite el acceso a los elementos hardware propio del teléfono tales como:

<i>API</i>	<i>Descripción</i>
Accelerometer	Lecturas del sensor de movimiento
Camera	Capturar imágenes a través de la cámara de fotos.
Compass	Obtener la dirección en la que apunta el dispositivo.
Contacts	Gestión de contactos.
Device	Información específica del dispositivo.
Events	Capturar eventos nativos.
File	Gestor de ficheros.
Geolocation	Servicios de localización.
Media	Control multimedia de video y audio.
Network	Comprobaciones del estado de la conexión a la red.
Notification	Gestión de notificaciones a nivel de sonidos y vibraciones.
Storage	Control de almacenamiento.

Figura 11. Tabla elementos hardware disponibles utilizando Phonegap [23]

Dos de los aspectos fundamentales y más ventajosos de Phonegap son por una parte que podemos ejecutar las aplicaciones en nuestro navegador web, sin necesidad de depender de un

dispositivo físico o de un emulador del mismo en el que lanza la aplicación, y segundo, la posibilidad de incluir en nuestro código funciones sobre frameworks como JQuery Mobile.

En definitiva, pese a que todas las opciones actuales tienen una gran calidad y cada una de ellas presenta algún que otro servicio exclusivo, de todas ellas el servicio que más interesa en este momento es que sus resultados sean ejecutables en la gran mayoría de S.O. existentes. Lo que es lo mismo, la cuota de mercado a alcanzar con dichos resultados sea la mayor de todas, y en este caso es Phonegap la que presenta los mejores en ese aspecto, y por lo tanto la elegida. Pese a que esta decisión se toma en información tomada de la propia plataforma, una vez trabajemos con ella, vemos que además, su utilización y funcionamiento se caracterizan por su facilidad y sencillez, siendo lo verdaderamente importante el código de la aplicación. Pero nada más explicativo que esto que demostrar cómo es posible instalar Phonegap en cualquier ordenador, y así poder trabajar con él.

2.5. Conclusiones

Tras este breve recorrido, hemos descubierto donde y cuanto hace que nació lo que se conoce como tecnología móvil. Como ha evolucionado a lo largo de estos años, desde aquellos grandes aparatos con enorme baterías a los terminales de bolsillo que hoy mismo abundan y que prácticamente todos tenemos. Más importante todavía, se ha visto las nuevas tecnologías que han hecho posible los medios de comunicación que tenemos hoy en día, tanto como el propio servicio de teléfono, como de Internet a través del móvil. Sin estos avances ahora mismo sería imposible hacer lo que nos hemos propuesto, así que hay que dar gracias a todas aquellas personas que colaboraron y lograron lo que ahora tenemos, y esta en nuestra mano poder hacer uso de ello.

Estos avances y la gran velocidad a la que se producen son también debidos a la gran competencia entre marcas y sistemas que hay en la actualidad, todas ellas quieren ser la mejor, innovar, buscar nuevas soluciones a los problemas, nuevos horizontes. Esa competencia es la que hace que se superen a sí mismas, dando a los usuarios servicios cada vez más increíbles y potentes. No es fácil decir cuál de ellas es la mejor, probablemente sea imposible, cada uno de los S.O. tienen unas virtudes y unos defectos, y cada tipo de usuario busca una cosa distinta a la hora de elegir. Algunos buscan simple comodidad, algo fácil, otros que sea seguro, otros poder personalizarlo, mil y un motivos distintos. Entonces si no podemos decir cual es mejor, como vamos a desarrollar algo específicamente para uno solo, y dejar a los demás sin el servicio que deseamos prestar, sería injusto, y por ello hemos decidido evitarlo.

Con nuestra aplicación híbrida, utilizando un framework para este tipo de aplicaciones conocido como *Phonegap*, conseguimos nuestro propósito principal, dar servicio a todos y cada uno de nuestros usuarios, sean de la compañía que sean. Estamos al servicio de ellos, pues ellos son los que valoran, y a ellos es a los que va dirigida la aplicación. No podemos cerrarnos solo a unos pocos elegidos, esto es para todo aquel que quiera disfrutarlo, sin excepciones, y ese es nuestro objetivo.



3. Bandevent multiplataforma: Especificación

Bandevent se enmarca dentro de las necesidades de la plataforma musical Bandness de expansión de mercado así como para facilitar a sus usuarios la posibilidad de seguir conectados e informados mediante las nuevas tecnologías móviles. Bandness nació hace un año de la mano de alumnos y también antiguos alumnos de la UPV aunque no fue hasta hace pocos meses cuando el proyecto realizó oficialmente su lanzamiento [31]. Por un lado, el objetivo de Bandness, es crear una plataforma donde tanto músicos como fans pudieran interrelacionarse, las bandas puedan buscar nuevos componentes para su grupo, difundir información sobre sus próximas actuaciones, música, gustos propios, buscar locales de ensayo, participar en concursos promocionados por Bandness, etc. y por su parte los usuarios de a pie podría escuchar música de bandas que por razones comerciales o marketing no son tan mediáticas o conocidas en dicho momento, dando así una oportunidad magnífica a bandas que de otra manera tendrían muy complicado llegar a un público tan grande de una manera tan sencilla como la que ofrece dicha plataforma. Pese a que existen otro tipo de plataformas similares en el mercado actual, Bandness intenta ofrece lo mejor de cada una de ellas. Además, al ser un proyecto joven, sus creadores y colaboradores se esfuerzan día a día en superarse y mejorar la experiencia del usuario. Es ahí donde nace la necesidad buscar una aplicación que permita a los usuarios, no solo vivir la experiencia a través de su ordenador, si no también de proveerles una nueva experiencia a través de sus Smartphone.

Dado que es muy difícil especificar la cuota de mercado de cada una de las plataformas de telefonía móvil dentro de los usuarios de Bandness, la elección de crear una aplicación puramente nativa para cada una de las plataformas se antoja complicada a la vez que inviable. En primer lugar, por el largo trabajo que conllevaría, ya que habría que partir de cero en cada uno de los códigos para generar la aplicación en cada uno de los S.O. y, al ser una empresa joven. En segundo lugar, se busca algo con lo que poder abarcar la mayor cuota de mercado posible, pero con una buena calidad de servicio, y que en un futuro pueda ser mejorada o adecuada a nuevas necesidades si así fuera necesario. Y dada la ventana de posibilidades que ofrece el nuevo HTML5 y CSS3, esta elección no es para nada desacertada.

Es por ello que, mediante el framework Phonegap, nos hemos apoyado en lenguajes HTML5 para elaborar cada una de las interfaces mediante un diseño Web, mediante CSS se define la presentación y el estilo de cada una de interfaces, con JavaScript controlamos las funciones a realizar dentro de la aplicación y que son lo que le dará funcionalidad a la misma y, finalmente, utilizamos PHP para establecer una conexión de datos entre el terminal y el servidor para obtener los datos que necesitamos a la hora de mostrar información que el usuario solicite.

La aplicación se podría dividir en tres partes. Por una parte una primera donde planteamos que queremos hacer en la aplicación y que características deseamos que tenga y que se pueda realizar con ella. Otra parte, que sería la de hacer el diseño de la aplicación, que sea interesante y acorde a las características y los servicios que deseamos prestar. Finalmente definir un modelo de datos en el cual se sustentará la aplicación y que le dará consistencia.

3.1. Modelo de negocio

Esta es la primera parte de todo lo que abarca la aplicación en sí, que no solamente es programarla y ya está. En primer lugar tenemos que tener muy claro, para no ir dando pasos sin rumbo, que es lo que queremos hacer en la aplicación, que servicios queremos ofrecer, como queremos ofrecerlos, y delimitar claramente lo que nuestra aplicación puede y no puede hacer.

3.1.1. Características de la aplicación

Estamos buscando algo que sea una experiencia rápida para el usuario, y que le ofrezca información que pueda ser de gran utilidad en ese momento para él mismo. De esta manera se plantean muchas opciones sobre qué queremos proporcionar. Dado que esta aplicación va dirigida a los teléfonos móviles, es muy probable que los usuarios de la misma la puedan utilizar no solo en sus hogares si no cuando están por la calle, y les pueda proporcionar una información que en ese momento les pueda ser útil. De esta manera, pensando en ello, surge el convencimiento que se puede proveer de un servicio para darle al usuario algo que hacer en ese momento, como por ejemplo un evento al que acudir. Es ahí, en ese punto, donde gira gran parte de la usabilidad de la aplicación, pero no solo eso, ya se puede dar que en ese mismo momento no haya nada interesante para realizar, pero si en un tiempo futuro no muy lejano, de tal manera que también se provee al usuario de un servicio con el cual poder recordar aquellos eventos que le han interesado y podría asistir cuando se realicen, como si fuera una agenda.

Pero estos aspectos son muy genéricos, así que vamos a intentar enumerarlos de una manera clara e iremos viendo como tratamos de cumplir los objetivos propuestos dentro de la aplicación. Para ello realizaremos una lista de requerimientos de la aplicación y los clasificaremos mediante el *estándar MoSCoW* [32]. Dicho método clasifica los requerimientos de un producto en 4 categorías: Must, Should, Could y Would.

- **Must:** Son los requerimientos esenciales que deben ser incluidos para que el producto o servicio funcione.
- **Should:** Son aquellos que pueden ser integrados para dar mayor beneficio.
- **Could:** Son los que si existe tiempo se podrían integrar, pero su ausencia no afecta significativamente al funcionamiento del producto o servicio.
- **Would:** Son aquellos que se tiene previsto integrar en un futuro, ya sea debido a que no se han podido integrar en este momento por falta de recursos económicos, tiempo, o la tecnología necesaria. Son lo que se llamaría un extra.

Ahora que ya sabemos como funciona cada uno de los cuatro apartados, podemos poner los requerimientos que queremos cumpla nuestra aplicación, por categorías.

MUST:

- **Servicio Público:** Al que pueda acceder cualquier persona con simplemente descargar la aplicación en su móvil. Dicho servicio debe contar con una serie de funciones:

- Buscador de eventos: Esta función como su propio nombre indica debe servir para mostrar al usuario los eventos que se realizarán próximamente. Dicha búsqueda puede ser realizada en tres opciones distintas:
 - Ciudad: El usuario selecciona la ciudad en la que buscar eventos.
 - Estilo musical: El usuario realiza una búsqueda por los estilos musicales que desee, pudiendo buscar por tres estilos al mismo tiempo.
 - Proximidad (Geolocalización): Mediante coordenadas GPS obtenidas de su posición a través del dispositivo móvil, se le mostrarán los eventos que estén en una posición próxima a su situación en ese momento.
- Información eventos: Información más detallada de los eventos que se realizan.
- Información bandas: Información más detallada de las bandas que se encuentran dentro de la plataforma Bandness.
- Servicio Privado: Es necesario que el usuario se identifique en la plataforma para poder acceder a contenidos privados de la misma. Contará con características añadidas:
 - Registro de eventos: Control de los eventos a los cuales se tiene previsto acudir o son de interés, y aquellos también que ya han pasado, a modo de diario.
 - Eventos de amigos: Conocer los eventos a los cuales asistirán tus amigos y poder también confirmar la asistencia a los mismos.

SHOULD:

Probablemente, la parte que sea más necesaria de ampliar sería la parte privada, para dotar al usuario de un mayor interés o una mayor motivación para registrarse en la plataforma Bandness, de esta manera podría ser interesante incluir algunos servicios más en la parte privada:

- Compartir en redes sociales: Poder publicar un enlace o una publicación en distintas redes de tal manera que el usuario pudiera comunicar que va a asistir a dicho evento a través de otras redes sociales para así hacer una mayor difusión del evento, o simplemente para hacérselo saber a sus amigos por si quieren unirse al mismo.
- Avisos de cercanía a un evento: Que el usuario reciba en su dispositivo móvil una notificación cada vez que un evento al cual ha marcado que asistirá este próximo a celebrarse a modo de recordatorio del mismo.

COULD:

Dentro de aquellos requerimientos que podrían ser interesantes si diera tiempo a ello, aunque no son prioritarios, podría haber tanto servicios de carácter público, como privado:

- Privados:
 - Notificaciones bandas: Que el usuario pueda marcar o directamente la aplicación tome los datos directamente del perfil del usuario en la

plataforma Bandness, y le indique cuando una banda de las que tiene marcada como favorita va a realizar un evento y la aplicación le avise de ello para que el usuario pueda decidir si quiere o no asistir, sin necesidad de que este tenga que buscarlo.

- Comentarios: Tener la posibilidad de insertar comentarios en un evento, a los que pudieran acceder y escribir, tanto las bandas que tocarán en el, como los usuarios que también asistirán.

- Públicos:

- Buscador de bandas: Implementar, además de la opción de buscar eventos, la posibilidad de realizar la búsqueda por bandas, utilizando las opciones de ciudad y por estilo.

WOULD:

Estos son algunos requerimientos, que dada su complejidad o los recursos que era necesario invertir en ellos, actualmente eran inviables, pero que en un futuro con los medios adecuados serían muy útiles:

- Fotos en los eventos: La posibilidad de que los usuarios subieran fotos de un evento, de tal manera que se creará un álbum del evento gracias a la colaboración de todos los usuarios. Además todos los asistentes del mismo tendría la posibilidad de ver dicho álbum y descargarse en sus dispositivos aquellas fotos que deseen.
- Comprar entradas: Si un evento es de pago, tener la posibilidad de comprar las entradas a dicho evento a través de la aplicación, sin tener que acceder a terceras páginas, o desplazarse físicamente para hacer la compra.
- Escuchar música del evento: Para aquellos usuarios indecisos que no saben si ir o no, porque no conocen los grupos que tocarán en el evento, darles la posibilidad de escuchar algunas canciones de lo mismos y que se hagan una propia opinión.

3.2. Diseño

Antes de ponerse manos a la obra con el desarrollo de la aplicación hay que tener claro qué es lo que se quiere ofrecer, y de qué manera hacerlo. En primer lugar, vamos mostrar las características que tendrá nuestra aplicación una vez finalizado su desarrollo. De esta manera marcaremos unos objetivos claros de trabajo desde un punto inicial hasta cumplir con todas las funcionalidades que deseamos nuestra aplicación sea capaz de realizar. Una vez claras las características de la aplicación procedemos a ver como mostrarlas, y en este caso estamos buscando por un lado que sea cómodas de utilizar, y además útiles, que el usuario vea que tiene sentido y utilidad la aplicación para su uso constante. De esta manera nuestro diseño no va a estar enfocado a la calidad de las texturas, los colores, o las transiciones entre pantallas. Lo que buscamos es una aplicación que destaque por su utilidad, no por su brillantez estética.



3.2.1. Interfaz gráfica de usuario

Con los objetivos ya claros de lo que queremos hacer, ahora es necesario seleccionar la manera adecuada de ofrecer estos servicios al usuario, de tal manera que la aplicación sea funcional, y cuente con una buena usabilidad, así como mostrar los resultados de las búsquedas solicitadas por el usuario también de una manera clara y sencilla, por lo cual se busca una interfaz que más que brillar por su estilo, o detallismo, sea importante por su claridad, por dar al usuario lo que busca directamente, con un tránsito entre las pantallas claro e intuitivo.



Bandness mobile

Figura 12. Diseño preliminar de las interfaces para la aplicación

Los colores predominantes tanto en menús, fondos, o barras de botones son tonalidades de color blanco, gris y negro, esto tiene que ver con los colores predominantes también en el diseño de la propia plataforma, y seguir una línea de diseño común entre ambas. En la pantalla de login predomina el color claro, para que el fondo de la plataforma destaque en gran medida, luego dentro de la aplicación se utiliza un degradado de tonalidades, desde la parte superior más clara hacia la parte inferior de la base más oscura, se puede apreciar claramente en pantallas como el menú de usuario. Luego para los listado como pueden ser tanto las bandas, o los eventos, se utilizan diversas tonalidades de gris para diferenciar entre los resultados mostrados, de tal manera que se vea claramente la diferencia entre el espacio que ocupa un resultado y otro.

3.2.2. Diagrama flujo entre pantallas

Como se puede observar, las interfaces están provistas de elementos cuya funcionalidad esta claramente reconocida por su propio nombre, con sencillez, pero toda aplicación debe tener unos estados entre los cuales transitar, y que estén determinados de una manera clara y siempre la misma, que el usuario sepa que cuando pulse un botón en un menú este le llevará a otra interfaz y que esta será siempre la misma, este conjunto de estados de estados por lo cuales pasa la aplicación durante se ejecución en respuesta a una serie de eventos se conoce como Diagrama de estados.

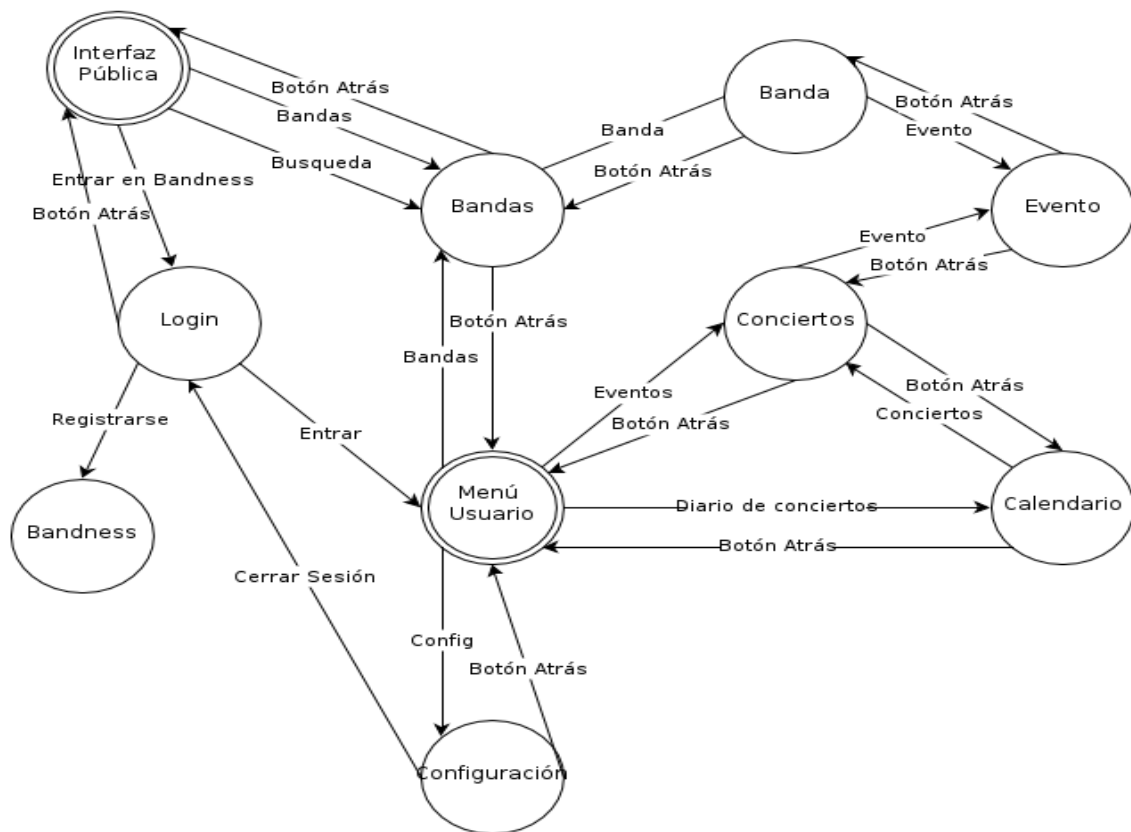


Figura 13. Diagrama estados aplicación

Encontramos dos puntos de entrada a la aplicación. Como usuario no registrado, en un primer momento, vamos a parar a la interfaz pública, desde donde podemos utilizar los servicios públicos de nuestra aplicación sin necesidad de hacer login. Si lo que queremos es ya entrar en los servicios privados, disponibles para aquellos usuarios registrados en Bandness, accedemos mediante la opción de login, y una vez validada nuestra sesión, está permanecerá abierta para posteriores ocasiones, y entraremos automáticamente en el menú de usuario. Siempre tendremos la opción de cerrar sesión en la configuración. Como puede verse existe interacción entre servicios privados y públicos ya que hay servicios como el de bandas que son comunes en ambos casos o el de eventos (salvo algunos detalles). Un detalle importante a comentar, es que pese a que la aplicación es multiplataforma, hay ciertos detalles que no es posible implementar de una misma manera en los diversos sistemas soportados. Esto quiere decir que aunque gran parte del código sea el mismo habrá pequeños detalles que no lo serán o que simplemente no

podrán funcionar a como está previsto en el diagrama de estados. Este diagrama de estados en concreto es aplicable a la mayoría de S.O. como Android, pero no así ocurre con iOS. El motivo es la transición entre estados que responde al evento *Botón Atrás*, esto es así por la sencilla razón que en los terminales con S.O. iOS dicho botón físico no existe, no como en el resto de dispositivos que si que lo tienen. Esto quiere decir que mientras que en la mayoría de los S.O. el evento *Atrás* se lanzaría mediante la pulsación de un botón del propio dispositivo, en iOS habrá que implementar otra manera distinta de realizarlo, o simplemente modificar parte del diseño para que se pueda acceder a las distintas pantallas de la aplicación a través de un submenú.

3.2.3. Servicios públicos de Bandevent

La aplicación, pese a estar dirigida a los usuarios de la plataforma musical Bandness, posee una parte publicada para toda aquella persona que desee conocer un poco más que hay dentro de Bandness, así como también obtener información sobre grupos pertenecientes. De esta forma se establece una conexión de tal manera que cualquier persona en el mundo que posea la aplicación, puede estar informada de los eventos de las bandas que se encuentran dentro de la plataforma, y en tiempo real. Este servicio público da unas grandes opciones para la promoción, y para atraer a nuevos usuarios dentro de la plataforma.

3.2.3.1. Menú principal

Al acceder por primera vez a la aplicación, esta será la ventana que le aparecerá a todo usuario. En ella nos aparecerán en el centro de la misma los eventos (más concretamente su portada o cartel) que por fecha están más próximos a celebrarse, por si el usuario estuviera en búsqueda de algún plan para esa misma noche o un día cercano. Esta es una buena manera de encontrar algo de una manera rápida y con solo dos pasos, el primero abrir la aplicación, y el segundo pulsar sobre la imagen del evento y que nos aparezca su información. En la parte superior se encuentra una barra con hasta tres botones, que nos permiten configurar las opciones de búsqueda sobre las que deseamos realizar nuestra consulta, pudiéndola hacer bien por ciudad donde se celebrará, por estilo musical del evento, o simplemente si queremos que nos diga los eventos que se van a celebrar próximos al lugar en el cual se encuentra el usuario en ese mismo instante mediante geolocalización. Finalmente, en la parte inferior de la misma, tenemos otras dos opciones que podemos ejecutar, que son por una parte el acceso a la parte de login, para así poder entrar a la aplicación si somos usuarios de Bandness, o bien otra opción en la cual nos aparecerá un listado con todas y cada una de las bandas, ordenadas según la fecha del próximo evento en el que participarán. Este método de visualización es bueno para las propias bandas incluso, ya que hace que tengan un interés también en querer que su perfil y su calendario este activo y con movimiento, ya que de esta manera su nombre será de los primero resultados de cualquier usuario que entre a la aplicación.



Figura 14. Menú Principal

Hay algunos aspectos del diseño en esta parte pública que hay que destacar, y que aparecerán también en las demás interfaces cuando el usuario no este identificado, por ejemplo la barra de botones inferior, con un tamaño constante respecto a su altura independientemente del dispositivo donde se visualice. Este es un aspecto que es complicado de implementar de una manera correcta para todos los dispositivos porque cada uno posee unas características de tamaño. La primera opción sería realizar la barra en un tamaño ajustado al porcentaje de pantalla que utiliza. Es decir, que barra inferior ocupará, por ejemplo, un 5 % de la altura total. Es cierto, que parece la más adecuada ya que se ajusta al tamaño y todos los elementos se escalarían en consecuencia, pero probablemente hay un punto de tamaño el cual ya no es necesaria la necesidad de que este siga aumentando. De esta manera, lo que hacemos en nuestra aplicación, es que las barras tanto superior como inferior, tengo un tamaño más o menos adecuado, y que resulten fácil de utilizar (no ser demasiado pequeños), pero que no llegue un punto que sean ridículamente grandes sin motivo alguno, y que lo que aumente en consecuencia al tamaño, sean son otros elementos de la pantalla, como el menú del centro de la aplicación con los eventos, que si que puede interesar sean cuanto más grandes mejor.

Esta barra inferior de menú, es la que aparecerá siempre que el usuario este en una ventana sin estar identificado, dándole siempre un acceso directo al login y a las bandas. Además, como hemos explicado, todos los elementos de menús, y los distintos botones de la interfaz tendrán siempre un tamaño constantes con independencia del dispositivo, lo único que se ampliará de tamaño será la base sobre la que están colocados, y mínimamente su posición para ajustarse al ancho y alto. Por otra parte lo que se tratará de ajustar al máximo son los elementos que más se quieren resaltar, como pueden ser los resultados de una búsqueda o los detalles de un evento, que se tratarán de mostrar siempre con el mayor tamaño posible.

Si pulsamos en los botones de la parte superior de la interfaz, nos aparecerá un nuevo contenedor (div) [33], en el lugar en el que con anterioridad nos aparecían los eventos más recientes que se iban a celebrar. Al pulsar el primero de los botones empezando por la parte izquierda, el de *Estilo*, nos aparecerán 3 *input text*, donde colocar los estilos de música que pretendemos buscar en nuestro evento, pudiéndose realizar la búsqueda independientemente del número de estilos que se introduzcan, sean tres, dos, uno, o ninguno (en este caso aparecerían todos los eventos). Una vez introducidos, para realizar la búsqueda se deberá pulsar el botón *¡Dale Caña!*



Figura 15. Menú Principal – Búsqueda Estilo

Al pulsar el botón *Ciudades*, lo que nos aparecerá en este caso es un único *input text* para introducir la ciudad donde queremos buscar los eventos que se producirán próximamente. Al igual que con la búsqueda por estilos, al pulsar el botón *¡Dale caña!*, la búsqueda se ejecutará.



Figura 16. Menú Principal – Búsqueda Ciudad

Finalmente, existe la posibilidad de búsqueda por geolocalización pulsando el botón *Cercanos*, la cual mostraría los eventos que se van a producir cercanos a nuestra posición actual. Su funcionamiento es que mediante posicionamiento, se obtienen las coordenadas en las cuales el usuario está con su teléfono móvil, a esas coordenadas se les aplica un radio de búsqueda en el cual por ejemplo, buscaría en 10 Km. a la redonda desde su ubicación, y como de cada evento están almacenados en la base de datos las coordenadas del mismo, aquellos que cumplan las condiciones de proximidad serán devueltos como resultados.

3.2.3.2. Bandas

Esta es una de las ventanas principales de la aplicación, ya que se entra a ella a través de varios menús o como resultado de una búsqueda. Para acceder a ella, tampoco es necesario ningún tipo de registro aunque una de las maneras mediante las cuales es posible entrar a ella es desde el propio menú de usuario. La primera manera de acceder a ella, es pulsando el botón *Bandas actualizadas*, situado en la parte inferior derecha del menú principal, donde al acceder nos aparecerá el listado con todas las bandas pertenecientes a la plataforma. La segunda será con los resultados obtenidos en las consultas que se puede realizar, también en el menú principal. Y finalmente, también el usuario registrado en su menú principal, tendrá la opción de acceder a listado completo de las bandas pulsado sobre el botón *Bandas*.

En lo que al diseño propio de la interfaz se refiere, cada banda está enmarcada dentro de una línea, unas de fondo más oscuro, otras más claro, que sirve como distintivo entre un resultado y otro. En la parte izquierda del mismo aparecen en primer lugar el nombre de la banda, a continuación la fecha en la que tendrá lugar el próximo evento en el que participan, y finalmente el lugar (sala) donde se llevará a cabo dicho evento. En la parte derecha lo que nos aparecerá es el logo distintivo de la banda. Y así por cada una de las bandas del listado o resultados de la consulta. En el caso que el número de bandas supere el espacio disponible de la pantalla, un scroll vertical permitirá el desplazamiento entre los resultados. La información de cada banda se muestra muy resumida al mostrarse en el listado así que, para ampliar dicha información, solo es necesario pulsar sobre la banda en cuestión y nos aparecerá una nueva ventana con información más detallada de la misma.

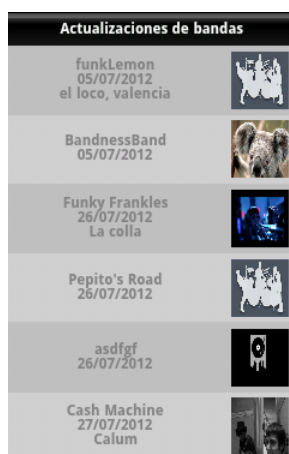


Figura 17. Bandas

3.2.3.3. Banda

Dado que la información proporcionada en el listado de las bandas, si no que es solo un resumen muy básico, en esta interfaz podemos encontrar toda la información importante respecto a ella. En la parte superior tendremos por una parte, en la parte izquierda el logo de la banda, y a continuación con un margen de separación el nombre de la misma. Debajo de este modulo superior encontraremos información de la banda como la ciudad donde se encuentra, los componentes de la misma, su estilo musical, así como el sello discográfico al cual pertenece. A continuación lo que nos aparecerá son las fechas en las que la banda va a celebrar su próximos



eventos, ordenadas de izquierda a derecha en orden ascendente por la fecha en la cual se celebrarán (es decir, el más cercano a la izquierda y así sucesivamente), de esta forma, no solo es posible ver quiénes son, si no también donde va a ser posible verlos. Esta información también es pública, es decir que toda persona puede verla pese a no pertenecer a Bandness, así que volvemos a incluir el menú inferior propio del menú principal por si alguien desea en este punto hacer login simplemente ver el listado completo de bandas.

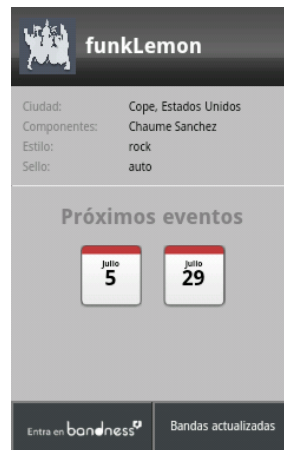


Figura 18. Banda

3.2.3.4. Eventos

Esta es la interfaz que muestra la información referente a los distintos eventos que están almacenados dentro de la base de datos, tanto aquellos pasados, como los que están por suceder. En la parte superior tenemos a la izquierda la fecha en la cual se va a realizar, o se ha realizado el evento, y a continuación a su derecha nos aparece el nombre del mismo. Seguidamente en un menú más central, bajo la barra superior, nos encontramos con detalles del evento, como la hora, la sala, la ciudad, el precio de la entrada, si la hubiera, y una descripción del evento. Tras estos detalles, en la parte inferior, nos encontramos una serie de iconos que veremos siempre que entremos a esta interfaz. Por una parte, a la izquierda aparece el número de usuarios que han confirmado que asistirán al evento, en la parte central aparece un botón llamado *Ver Mapa* el cual pulsando sobre él nos mostrará la localización exacta en el mapa del lugar donde se va a realizar el evento, para que aquellas personas que no sepan donde se encuentra el lugar pueda localizarlo de una manera rápida, y finalmente en la parte derecha nos aparece en primer lugar la opción de *Confirmar asistencia*. Para aquellos usuarios que no estén registrados, si tratan de pulsar sobre dicho botón les saldrá un mensaje de alerta que les informará que deben de registrarse en web de Bandness [5], para poder realizar dicha confirmación. Si en cambio el usuario ya esta registrado y identificado en la aplicación, se modificará el mensaje mostrado por el de *Asistencia confirmada*, el botón cambiará a un color azul claro, y automáticamente el contador de asistencia verá incrementado en uno su valor. Debajo de este botón nos encontramos con la posibilidad de compartir el evento en las redes sociales, tanto en la red social Facebook [33] como twitter [34], de tal manera que sus amigos en dichas redes puedan ver también la información del evento a través del enlace que el usuario les proporciona.

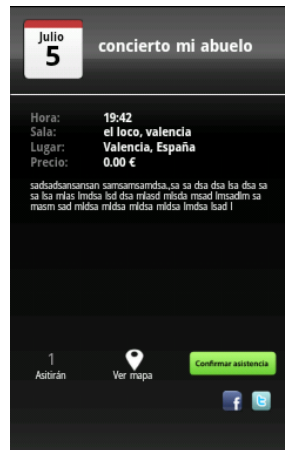


Figura 19. Evento



Figura 20. Localización



Figura 21. Compartir en Facebook

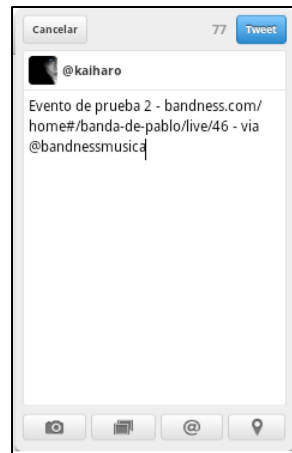


Figura 22. Compartir el Twitter

3.2.3.5. Login

En esta pantalla como su propio nombre indica, es donde el usuario podrá acceder a los servicios privados de la aplicación, una vez se haya logeado correctamente en la aplicación. Tanto su usuario como contraseña serán los mismos que en la web Bandness [5]. De no estar registrado en la web, se le da la opción a través del botón *Registrarme* de entrar directamente a la web donde poder registrarse en la plataforma. El atributo *user* corresponde al correo electrónico del usuario que deberá introducir en el input text que aparece a continuación de dicha palabra. Por otro lado, el atributo *pass* quiere decir el password que aparecerá en formato “pass” mientras se van rellenando los campos del input text correspondiente, de tal forma que en vez del texto correspondiente al password lo que aparecerán serán * por cada una de las letras que lo formen, impidiendo así que alguien que intente espiar la pantalla del usuario de la aplicación pueda obtener su password. Si el usuario o el password son inválidos, una vez pulsado el botón *¡Entrar!*, nos aparecerá un mensaje de error en la parte inferior de cada uno de los atributos, indicándonos que alguno de los dos parámetros es incorrecto, indicando en cada caso si es causa de que el usuario no existe, o que el password introducido no coincide con el de dicho usuario. Si es correcto, accederemos automáticamente al menú del usuario.



Figura 23. Login

3.2.4. Servicios privados de Bandevent

Una vez el usuario ha realizado con éxito el login, y se ha validado correctamente, podrá acceder a los contenidos y funcionalidades privadas de la aplicación. Además, a no ser que el mismo desee cerrar la sesión de manera automática en la configuración de la aplicación, cada vez que vuelva a acceder ésta recordará su sesión para agilizar el proceso y entrará directamente a la parte privada de la misma. Los servicios añadidos de esta parte privada y de los que podrá disfrutar el usuario son:

- Guardar información sobre asistencia a eventos, de tal manera que en primer lugar se aumente por una parte el número de asistentes a dicho evento dentro de la información del mismo y por otra puedan ser enviadas notificaciones al usuario avisándole de la proximidad del evento en fecha cuando sea conveniente.
- Poder acceder a los eventos a los cuales asistirán nuestros amigos, y de igual manera podrá indicar que asistirá a dicho evento a través de la ventana principal del mismo.
- Diario de eventos, tanto eventos pasados y que ya se han realizado, como eventos que están por producirse.

3.2.4.1. Menú usuario

Como usuario registrado se presentan cuatro opciones a poder realizar dentro del menú:

- Conciertos: Se accede a un listado con los eventos a los cuales el usuario tiene previsto asistir próximamente.
- Bandas: Aquí se muestra el listado completo de bandas dentro de la plataforma ordenadas por proximidad de un evento de la banda, es decir que aquellas bandas que tengan un evento que se vaya a realizar en un espacio corto de tiempo, aparecerán por encima que aquellas no tengan ninguno en próximas fechas.
- Diario de conciertos: Se muestra un calendario, partiendo el año actual, con todos los eventos a los que se ha asistido o se asistirá, de tal manera que se pueda acceder a la información de todos ellos y tener un control desde esta ventana.
- Configuración: Muestra los elementos básicos de configuración de la aplicación o de la sesión de usuario que pueden ser modificados.





Figura 24. Menú usuario

3.2.4.2. *Conciertos*

En esta ventana es donde el usuario registrado podrá ver aquellos conciertos a los cuales ha confirmado su asistencia, para acceder a ellos, y consultar su información en cualquier momento que lo desee de una manera rápida. Si pulsa sobre el evento en cuestión se le abrirá la ventana con la información detallada del evento en cuestión. En la parte inferior de la misma, el usuario tiene la opción de acceder a los conciertos que tienen previsto asistir sus amigos, pudiendo acceder a cada uno de ellos y comprobar a donde irán, ya que quizás de dicha información puedan realizar algún plan al coincidir en un mismo concierto o encuentra algún concierto que también pueda resultarle de interés y confirmar el mismo su asistencia.



Figura 25. Conciertos

3.2.4.3. *Diario de conciertos*

Esta funcionalidad de diario, sirve para tener un control, tanto de los eventos a los que se tiene intención de asistir por parte del usuario, como aquellos a los que ya ha asistido, para tener siempre un recuerdo o un control de ello. Los eventos no son mostrados directamente si no que nos aparece un resultado que nos indica para el año en el que estamos (aunque podemos

desplazarnos años atrás o adelante gracias a las flechas situadas en la barra inferior de la pantalla), cuantos eventos tenemos o hemos tenido en los meses de dicho año. De solo haber resultados en ciertos meses del año, solo nos mostrará aquellos que no sean 0. Si queremos acceder a los eventos de dicho mes el usuario pulsará sobre la línea de resultado correspondiente a dicho mes y le llevará hasta una nueva ventana con el formato de *Conciertos* donde le aparecerán los respectivos eventos correspondientes a dicho mes (a los cuales como es obvio también podrá entrar en cada uno de ellos a la respectiva pantalla con la información de cada uno, la de *evento*).

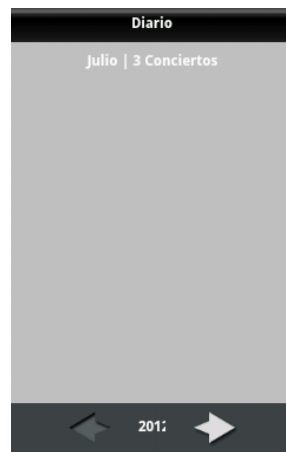


Figura 26. Diario de conciertos

3.2.4.4. Configuración

En la última de las opciones que tendrá el usuario registrado, estará la de modificar ciertos valores de la configuración de la aplicación o de su propia sesión de usuario, son los siguientes:

- Activar geolocalización: Dado que este servicio utiliza el GPS del dispositivo móvil, este es un servicio que consume recursos como la batería del Smartphone, algo muypreciado en estos, la baja duración de sus baterías.
- Activar notificaciones: Este es un servicio propio de los Smartphone y de las aplicaciones nativas que envía mensajes de información al usuario por ejemplo para informarle de una actualización de la aplicación a través del market. En nuestro caso las notificaciones serían enviadas desde el servidor de la plataforma, cuando un evento al que un usuario haya confirmado asistencia vaya a realizarse en un plazo corto de tiempo, haciéndole un recordatorio de ello. Pero dado que es un servicio de alertas, si el usuario no desea recibir ninguna tiene la opción de desactivar dicho servicio.
- Compartir eventos: Si un usuario desea mantener total privacidad sobre los eventos a los que va a asistir, se puede deshabilitar esta opción de tal manera que ningún usuario de la plataforma pueda conocer los eventos a los que dicho usuario va a asistir.

- Cerrar sesión: La sesión dentro de la plataforma, una vez iniciada, permanecerá abierta para posteriores ocasiones en las que el usuario utilice la aplicación. Si desea que no sea así y cerrar dicha sesión de usuario puede hacerlo mediante esta función y cerrar su sesión actual de usuario.

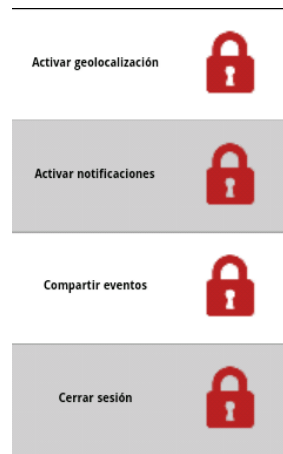


Figura 27. Configuración

3.3. Modelo de datos

Pese a que la plataforma Bandness, cuenta con una extensa base de datos y de tablas, para la aplicación que aquí tenemos solo nos hará falta una parte de ellas. En concreto todas aquellas que tienen que ver con los elementos que definen la aplicación: Los eventos, las bandas, y los propios usuarios. A su vez, existen otras tablas que están encargadas de relacionar dichos elementos, por ejemplo para el control de bandas asistentes a un evento, o para saber si un usuario asistirá a un evento.

Existen otras tablas, que pese a no estar relacionadas directamente con ninguna, nos aportan elementos fundamentales de información, como por ejemplo las referentes a los estilos musicales de las bandas (*music_likes*), los amigos de los usuarios (*friends*), y la localización donde está asentada una banda o se va a celebrar un evento (*location*). La manera en la que estas tablas se conectan con las demás es a través del atributo *referer* que dependiendo de su valor quiere decir que hace referencia a un tipo distinto, como por ejemplo si se trata de un evento o de una banda, y el atributo *ítem_id*, es igual al id del evento o banda dependiendo de lo que se trate en cada caso.

Todo ello podemos definirlo de una manera sencilla mediante tablas de datos, relacionadas entre sí como vemos en la figura siguiente:

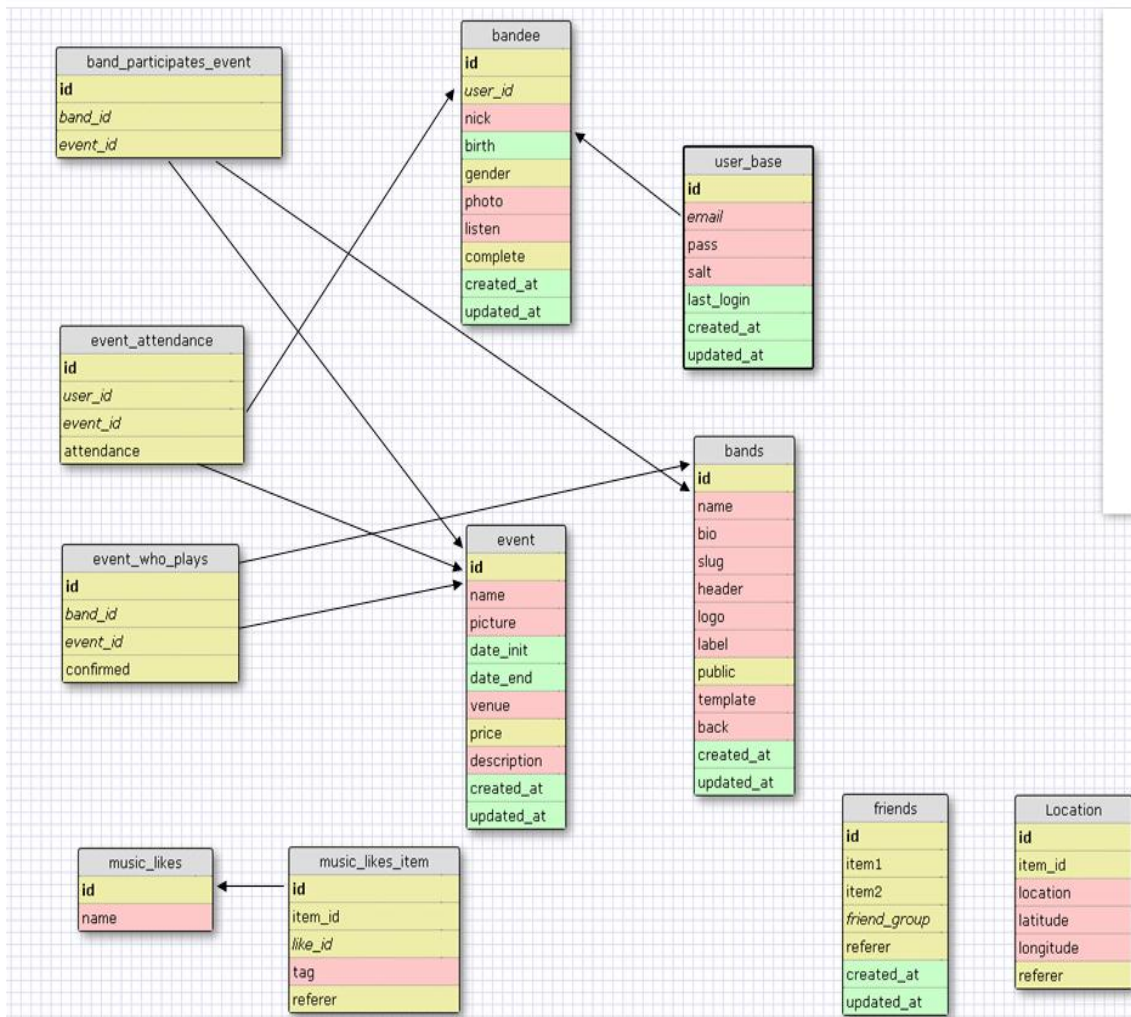


Figura 28. Modelo de datos



4. Implementación

Una vez presentado el diseño que presentará nuestra aplicación así como el modelo de datos en el cual se ejecutará, es hora de mostrar los puntos más destacados de la implementación del código que hacen posible la de la aplicación, y de todas sus funcionalidades preestablecidas. Pero también aquellos puntos que han sido más laboriosos en cuanto a su desarrollo y posterior funcionamiento, ya sea por dificultad de los mismos, o por los problemas surgidos a la hora de programar mediante lenguaje web, en una aplicación para móvil.

Prácticamente todas y cada una de las funciones establecidas dentro de la aplicación han sido testeadas y comprobadas mediante distintas pruebas para su correcto funcionamiento. Algunas de estas pruebas también las mostraremos para que se vea como se han desarrollado y su resultado positivo. Otra cosa muy a tener en cuenta a la hora de realizar la implementación de nuestro código es el disponer unas buenas herramientas técnicas que nos ayuden en nuestra tarea. Por ejemplo a la hora de la programación, que nos sean un buen medio de testeo o de información acerca del funcionamiento de la aplicación en cada momento del desarrollo, mostrándonos posibles fallos de código o excesos de utilización de memoria.

4.1. Primeros pasos con Phonegap en Android

La elección de trabajo de Android como plataforma nativa sobre la cual lanzar la aplicación generada mediante Phonegap se debe principalmente a disponer de un terminal propio con S.O. Android en el cual poder realizar las pruebas de la aplicación, además de disponer ya de los conocimientos necesarios y tener instalados previamente los elementos requeridos para poder programar aplicaciones para este S.O. Antes de ponerse con la programación específica de la aplicación es necesario configurar correctamente el entorno de trabajo para poder ejecutar a posteriori la aplicación correctamente.

Así que vamos a detallar brevemente esta configuración para todo aquel que desee saber como hacerla por si en algún momento está interesado en esta tecnología y en las oportunidades que nos da en el mundo de las aplicaciones móviles [27].

1. Requisitos:
 - Eclipse 3.4 ó superior
2. Instalar SDK + Cordova (Nombre actual de la tecnología Phonegap):
 - Descarga e instalación de Android SDK [28].
 - Descarga e instalación ADT Plugin para Eclipse [29].
 - Descarga e instalación de la última versión de Cordova [30].
3. Configuración nuevo proyecto:
 - Una vez lanzado Eclipse, seleccionamos el menú ítem -> Project
 - Elegimos Android Application Project

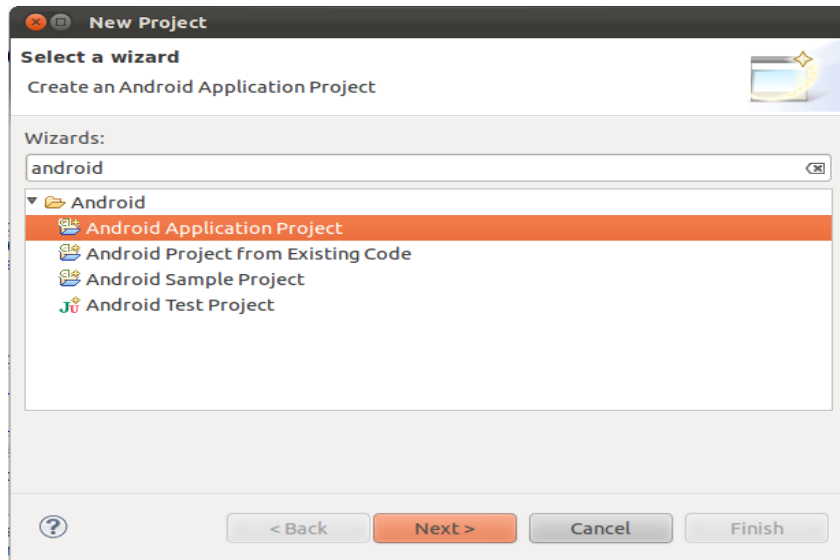


Figura 29. Selección New Project

- Especificamos el nombre de la aplicación, del proyecto y del paquete con el Namespace

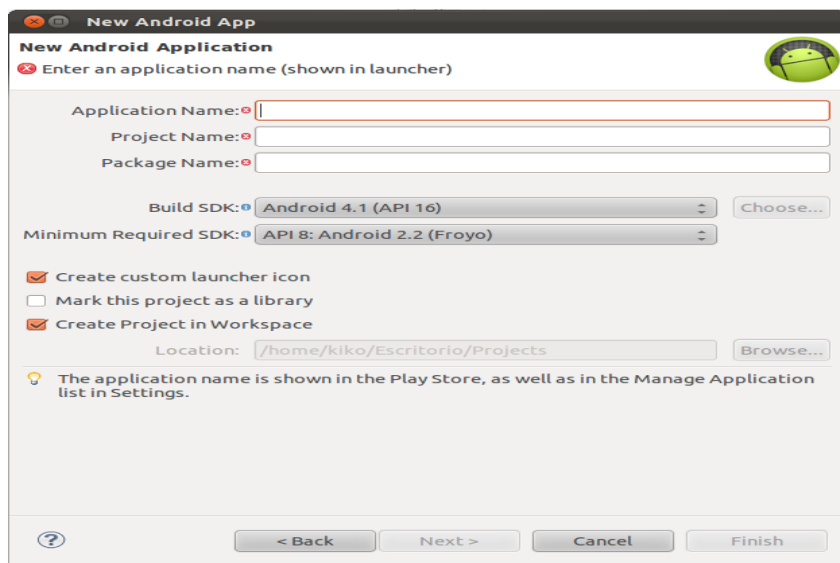


Figura 30. Configuración proyecto

- Seleccionamos un icono para la aplicación

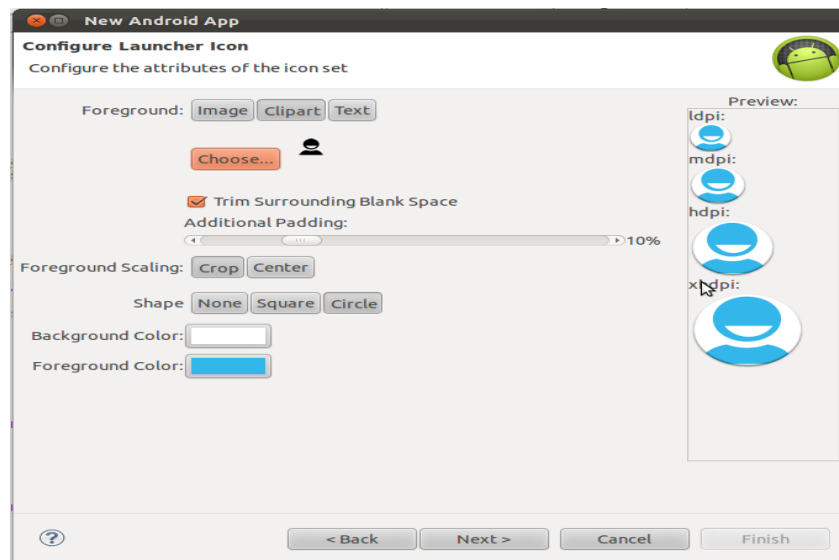


Figura 31. Selección Launch icon

- Creamos una actividad (main) en blanco, le damos un nombre a continuación (siguiente ventana) y con ello se creará el proyecto

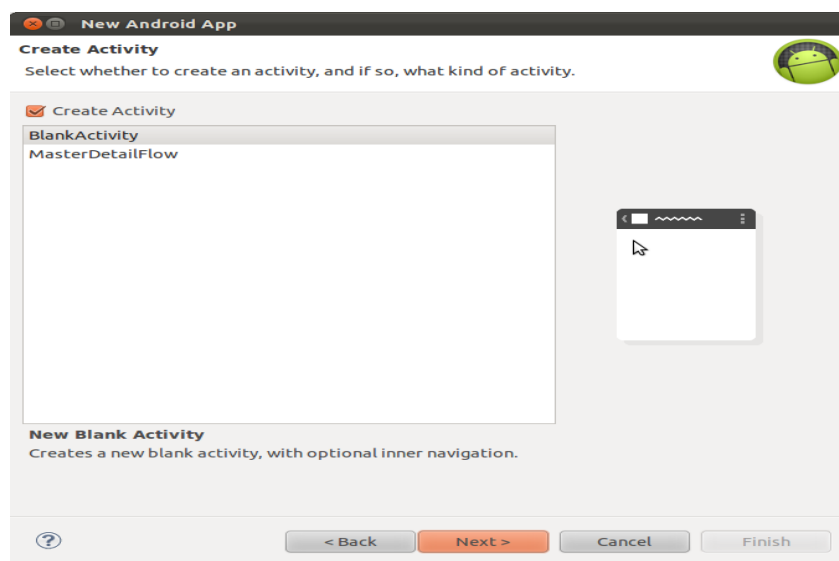


Figura 32. Creación main activity

- Una vez creado el proyecto, en la ruta del mismo creamos dos nuevos directorios:
 - /libs
 - assets/www
- Copiamos el fichero .js de phonegap que nos hemos descargado previamente en assets/www
- Copiamos fichero .jar de phonegap en /libs
- Copiamos el fichero .xml descargado con phonegap en /res

- Comprobamos que el fichero .jar está en el Build path de nuestro proyecto. Para ello entramos en las propiedades del mismo, properties -> Java Build Path -> Libraries

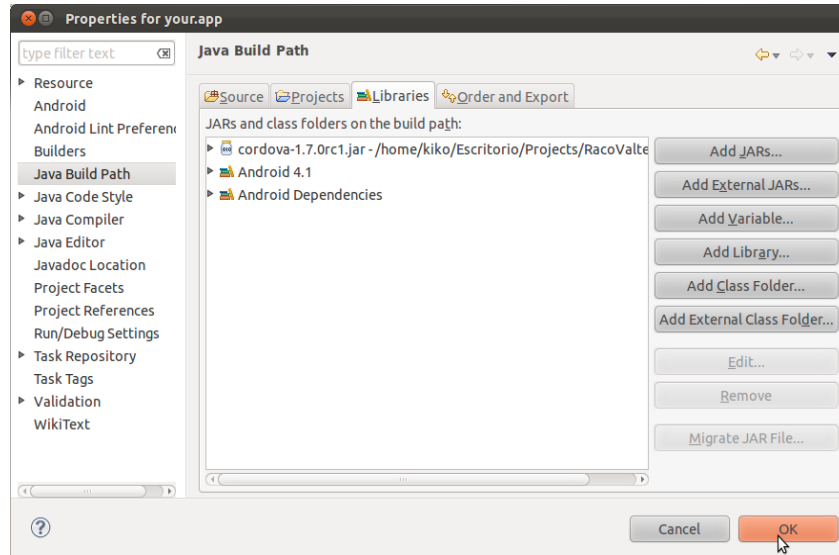


Figura 33. Librerías

- Finalmente modificamos el fichero MainActivity.java para que quede de la siguiente manera:

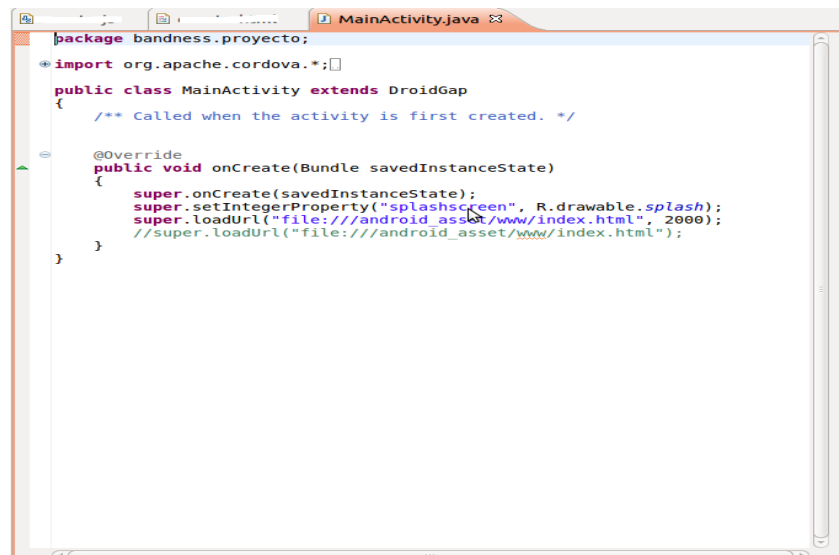



Figura 34. MainActivity.java

- Añadimos: **import org.apache.cordova***
- Cambiamos la clase extend de **Activity** a **DroidGap**
- Remplazamos la línea **setContentView()** por **super.loadUrl("file:///android_asset/www/index.html")**

- Finalmente, si abrimos el **AndroidManifest.xml** -> **Text Editor**, podremos añadir una serie de permisos entre el `<uses-sdk.../d>` y `<application.../>` tags



supports-screens:
Hace que el código a ejecutar funcione en cualquier tipo de pantalla

android.permission:
Permite que la aplicación pueda ejecutar el servicio correspondiente del teléfono, como puede ser **la vibración, el acceso a internet, etc. etc.**

Figura 35. AndroidManifest.xml

- Es importante no añadir más permisos de los que de verdad vayamos a usar, pues esto puede hacer que al subir nuestro código al market, se pueda ejecutar en menos terminales de los que en realidad es posible, al pedir permisos de servicios que ni tan siquiera usamos y que además algunos teléfonos no disponen impidiendo así que pudiera ejecutarse la aplicación.

4.2. Lenguajes de programación

Durante el desarrollo de la aplicación se utilizan distintos tipos de tecnologías a la hora de programarla. Tenemos tanto para establecer conexiones usuario – servidor, como para realizar consultas a una base de datos, pasando por lo que debe hacer la aplicación cuando el usuario pulse un botón de la misma. Vamos a pasar a introducirlas brevemente, y cuál es la función de cada una:

- **PHP (PHP Hypertext Pre-processor):** Es un lenguaje de programación que se utiliza para la gestión de información por parte de un servidor, enviada por un usuario. De esta manera, utilizaremos dicho lenguaje para la gestión de las conexiones entre usuario – servidor de la aplicación [36].
- **AJAX (Asynchronous JavaScript And XML):** Es una tecnología que se encarga en este caso, de gestionar e implementar la parte usuario, en una conexión usuario - servidor [37].
- **MySQL:** Es un sistema de gestión de bases de datos relacional, mediante el cual realizaremos las consultas, insertado y borrado de datos de la aplicación en la base de datos del servidor [38].

- JavaScript: Es un lenguaje de programación interpretado, orientado a objetos, que se utiliza principalmente en la parte cliente de la aplicación. Sirve entre otras cosas para realizar mejoras en la interfaz de usuario, páginas web dinámicas, bases de datos locales del navegador, etc. [39]
- HTML: Es el lenguaje predominante para la elaboración de páginas web, que se utiliza para describir y transcribir la estructura e información del texto, así como complementarla con distintos elementos [40].
- CSS: Conocido como hoja de estilos en cascada, es utilizado para definir la presentación de un documento estructurado elaborado en HTML o XML [41].

4.3. Peticiones servidor

Uno de los aspectos fundamentales de la aplicación, sin la cual sería imposible que esta aplicación pudiera funcionar, es el intercambio de información entre el cliente y servidor (En este caso el servidor de la plataforma Bandness). Este intercambio de información está apoyado en el protocolo HTTP para gestionar las peticiones, usando el cliente métodos como GET y POST, para respectivamente, recibir y enviar información, y códigos de respuesta típicos del propio protocolo [42].

4.4. Establecimiento conexión: Usuario - Servidor

Dado que nuestra aplicación tiene una parte de funcionalidad para la cual no es necesario ningún tipo de registro. En este caso, lo primero que se planteo fue la necesidad de establecer una conexión, usuario – servidor, para búsqueda de datos. Mediante una serie de parámetros que se introducen en la interfaz de la aplicación, el usuario realiza una consulta, a la cual responde el servidor con los resultados obtenidos. En primer lugar, como funcionalidad automática, el servidor devuelve una respuesta con los eventos más próximos en fecha, y se los muestra al usuario nada más abrir esta la aplicación. Esto se realiza mediante una función JavaScript utilizando AJAX. Se establece una conexión con el servidor mediante una serie de parámetros de llamada como la url destino donde está el servidor, el tipo de conexión, o el formato de datos entre otros:

```
$.ajax({
  url: 'http://192.168.1.100/evento_por_fecha.php',
  dataType: 'jsonp',
  jsonp: 'jsoncallback',
  type: 'get',
  timeout: 5000,
  success: function(data/*, status*/){
    $.each(data, function(i,item){
      var celda = document.getElementById(i);
      celda.id=item.id;

      celda.src='http://test.bandness.com/uploads/images/posters/'+item.pictu
      re;
    });
  },
});
```



```

    error: function(){
    }
});

```

Esta función, con ciertas modificaciones, como la url destino, o el trato de los resultados obtenidos dentro del campo *success*, es la que utilizaremos para comunicarnos con el servidor. En dicha función como vemos en la llamada, se utiliza JSONP (JavaScript Object Notation with Padding), un complemento apoyado en el formato de datos JSON, que nos permite la recepción de datos desde un servidor situado en un dominio diferente.

El formato JSON es un formato ligero de intercambio de datos [43]. Esta constituido en dos partes, ambas universales de tal forma que son soportadas por todos los lenguajes de programación:

- Una colección de pares {nombre / valor}. Conocido en otros lenguajes como estructura, registro, diccionario o tabla hash.
- Una lista ordenada de valores.

De esta forma ya tenemos presentado como funcionará nuestro código en cuanto a la parte de JavaScript se refiere, es decir la parte de usuario en este caso, pero ahora hay que ver como interpretará el servidor la consulta realizada. Para ello se utilizará el lenguaje PHP, en el caso planteado como inicial de la devolución de una serie de eventos ordenados por fecha, el código que el servidor interpretará es el siguiente:

- En primer lugar se establecer una conexión con la base de datos del servidor, para tener acceso a la misma y poder así realizar la consulta correspondiente:

```

$database = "Bandness";

$link = mysql_connect('localhost', 'root', 'admin');
$db_selected = mysql_select_db($database, $link);

if (!$link) {
    die('Could not connect: ' . mysql_error());
}

```

- Una vez establecida la conexión procederemos a realizar la propia consulta. Algo que nos ha supuesto ciertos problemas en este caso ha sido la codificación de caracteres de los datos, ya que en ciertas ocasiones las consultas eran fallidas pese a que el código SQL era correcto, así que para evitarnos problemas es necesario introducir esta línea en nuestro código PHP:

```
mysql_query('SET CHARACTER SET utf8');
```

- Ahora ya si, podemos realizar la consulta a la base de datos mediante un código SQL, el lenguaje de acceso a bases de datos, de fácil aprendizaje y utilización, que usaremos para nuestras consultas en todo momento. En primer lugar escribiremos nuestra consulta en la base de datos y se almacenará en una variable, después dicha consulta será ejecutada mediante la función *mysql_query* en la base de datos. Una vez ejecutada, se almacenan los resultados en *\$result* y a posteriori son tratados para devolverlos en pares {nombre / valor}, pudiendo ese término *nombre* por múltiples

elementos (array), dado que las consultas no tienen por qué devolver solo un único atributo de cada uno de los valores:

```
$sql = "SELECT id,picture,date_init FROM event GROUP BY date_init";  
$result = mysql_query($sql) or die("Query error: " . mysql_error());  
  
while ($row = mysql_fetch_assoc($result)) {  
    $records[] = $row;  
}
```

- Finalmente, una vez realizada la consulta y tratados con éxito los datos, devolveremos los datos resultados obtenidos al javascript del usuario, para que trabaje con ellos, y los utilice como corresponda:

```
echo $_GET['jsoncallback'] . '(' . json_encode($records) . ')';  
?>
```

Ahora que ya sabemos la estructura de la conexión entre el usuario y el servidor y como esta funciona, vamos a analizar las diferentes búsquedas y las particularidades que se presentan a lo largo del código, en primer lugar como ya hemos citado antes, la que nos permite la búsqueda de bandas mediante una serie de parámetros indicados por el usuario.

4.5. Servicio búsqueda por parámetros

Dentro de nuestra ventana principal de la aplicación, a la que accedemos al inicio por primera vez de la misma, y que solo cambiará cuando el usuario este ya logeado, se nos presenta el primer punto de interés dentro del código realizado. Mediante la consulta al servidor de unos datos que varían dependiendo de los parámetros de entrada que el usuario indique a cada momento.

Como se ha dicho previamente, existen tres tipos de búsquedas, por estilo, ciudad y geolocalización. Obviaremos esta última ya que su funcionamiento es diferente al de las otras dos, al tomar directamente la posición del teléfono mediante GPS. De esta forma en vez de introducir el usuario el parámetro, es el propio dispositivo quien lo obtiene automáticamente y posteriormente lo envía al servidor para hacer la consulta.

En cada una de las interfaces propias de ambas consultas, tanto estilo, como ciudad, nos aparecen una serie de campos, o uno único, donde introducir los parámetros de la búsqueda. Además existe un parámetro que varía dependiendo de la clase de búsqueda que se quiera realizar llamado *tipo*, que al llegar al servidor diferenciará de si se trata de una consulta u otra, de tal forma que el código a ejecutarse sea distinto dependiendo del tipo de consulta. Esta serie de parámetros son enviados a otro archivo, en la url del mismo, en el cual se procederá a lanzar el JavaScript con la consulta. Este sería el procedimiento de paso de parámetros para una consulta por estilo:

```
function buscarEstilo(){  
    var tipo="ES";  
    valor=document.getElementById("Estilo1").value;
```



```

estilo2=document.getElementById("Estilo2").value;
estilo3=document.getElementById("Estilo3").value;
linkLocation =
"html/bandas.html?Banda="+valor+"&estilo2="+estilo2+"&estilo3="+estilo3+"&tipo="+tipo;

$("body").fadeOut('slow',redirectPage);
}

function redirectPage() {
    window.location = linkLocation;
}

```

Tras ello, los parámetros serían leídos mediante otra función JavaScript, la cual los almacenaría en variables para su posterior utilización. Una vez leídos, se procedería a realizar la llamada mediante AJAX al servidor, en ella en este caso, en el parámetro de la url, irán las variables con los valores recibidos previamente:

```
url:'http://192.168.1.100/localizacion.php?busqueda='+Banda+"&estilo2="+estilo2+"&estilo3="+estilo3+"&tipo="+tipo
```

Los cuales, al ejecutarse el PHP, serán leídos y almacenados como primer paso del código, para luego poder utilizarlos en la/s consulta/s a realizar:

```

$busqueda = $_GET['busqueda'];
$estilo2 = $_GET['estilo2'];
$estilo3 = $_GET['estilo3'];
$tipo = $_GET['tipo'];

```

Como anteriormente, esta consulta devolverá un resultado que la interfaz de la aplicación mostrará en la nueva ventana a la cual hemos accedido al ejecutar nuestra consulta, donde nos aparecerán listados los resultados obtenidos con nuestros parámetros de búsqueda.

4.6. Muestra de los resultados de las consultas

Dentro de la aplicación, pese a que se realizan diferentes búsquedas y de distinto ámbito, sobre distintas tablas, y con todo tipo de parámetros habiendo una gran diversidad de las mismas, el formato en el que esos datos se muestran es más constante. Existen dos maneras de mostrar los datos, principalmente, que son mediante un listado, dado que el número de valores del mismo puede ser superior a uno, aunque no siempre se cumpla, o bien mediante el rellenado de campos.

4.6.1 Listado

Como bien se ha dicho, cuando el número de valores devueltos en una consulta es superior a uno, o bien hay opción de que así sea, los resultados son mostrados en alguno de los diferentes listados que hay en la aplicación, ya sea en el bandas, el de eventos, etc. Para ello, pese a que la consulta en sí no presenta nada distinto a lo que sería una consulta de otro tipo, si

que ocurre con los datos recibidos de ella. Partiendo del ejemplo el cual hemos analizado anteriormente, que nos devolvía una consulta sobre bandas, en este caso, que cumplían una serie de parámetros, estos mismos resultados serán mostrados en un listado de las mismas (como podemos ver en la figura 28)

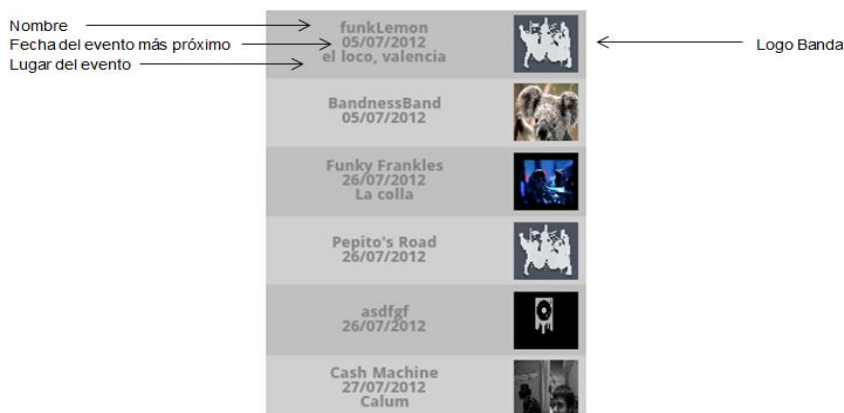


Figura 36. Listado resultados

Para el caso que nos ocupa se toman cuatros parámetros de cada una de las bandas, los cuales se muestran en distintas filas para cada una de ellas. En este caso, dentro del AJAX donde se procede a la consulta con el servidor, al ser esta ejecutada con éxito, se realiza lo siguiente:

```
$.each(data, function(i,item){
    name=item.name;
    id=item.id;

    cabecera="http://test.bandness.com/uploads/images/band/header/"+item.header;
    lugar=item.venue;
    if(item.date_init!=0){
        fecha=item.date_init;
        landmark =
'<h2>'+name+'</h2><BR><h2>'+fecha+'</h2><BR><h2>'+lugar+'</h2>';
    }

    else
        landmark = '<h2>'+name+'</h2><BR>';

    addNewRow();
}
```

El *\$.each* que tenemos quiere indicar que para cada uno de los valores obtenidos como resultados (ítem), se ejecutará el código escrito a continuación (entre { }). Es decir, que en primer lugar como vemos se almacenan en variables las atributos recibidos en la consulta, y que una vez esto completado se procederá a ejecutarse la función *addNewRow()*.

Esta función *addNewRow* tiene como objetivo el crear las filas, con su correspondiente estilo, y la colocación de los atributos de cada valor conforme a lo que se puede ver en la imagen previa. Además en cada línea de resultados se establecerá un hipervínculo mediante el

cual el usuario pulsando en ella, podrá acceder a la siguiente interfaz que en este caso sería la Banda. La función `addNewRow` es la siguiente:

```
function addNewRow()
{
  // obtenemos acceso a la tabla por su ID
  var TABLE = document.getElementById("bandas");
  // obtenemos acceso a la fila maestra por su ID
  var TROW = document.getElementById("fila");
  // tomamos la celda
  var content = TROW.getElementsByTagName("td");
  // creamos una nueva fila
  var newRow = TABLE.insertRow(-1);
  newRow.className = TROW.attributes['class'].value;

  if(par=='1'){
    newRow.style.background = "#D0D0D0";
    par--;
  }

  else
    par++;

  // creamos una nueva celda
  var newCell = newRow.insertCell(newRow.cells.length);
  var newCell2 = newRow.insertCell(newRow.cells.length);

  newCell.className = 'nombres';
  newCell2.className = 'logo';

  // y lo asignamos a la celda
  newCell.innerHTML = landmark;
  newCell2.innerHTML = ""

  newRow.idName=id;
  newRow.onclick=function(){Cargar_Banda(newRow.idName);}
}
```

Esta función será la que utilizaremos prácticamente para todos y cada uno de los listados en la aplicación con algunas modificaciones en los datos que mostrarán ya que variaran dependiendo de si es un evento, o una banda. Pero esto únicamente modificará el contenido de la misma, no su funcionamiento. La siguiente opción que utilizaremos, es principalmente cuando obtenemos una gran cantidad de nombres acerca de un mismo valor, por ejemplo todos los datos de una banda, o los de un evento.

4.6.2. Rellenado de Campos

Cuando lo que realizamos es una consulta más específica, donde sabemos con seguridad que solo va a existir como mucho un único valor posible de resultados, es cuando trabajamos como *id*. Es decir, buscamos un valor en concreto, en algunas ocasiones lo que haremos será pasarle el *id* para obtener los atributos en los que se encuentre dicho valor. En el ejemplo que hemos comentado antes al explicar el listado, hemos visto como cada fila tenía un enlace a la

ventana con el evento correspondiente y sus detalles. Bien, uno de los atributos resultado en la consulta anterior fue el *id* de cada banda. De esta manera lanzaremos la siguiente pantalla, pasándole el *id* del evento para realizar la siguiente consulta en la base de datos:

```
window.location = "banda.html?Banda="+x;
```

Ya en el código correspondiente a la banda, haremos las consultas necesarias a la base de datos, pasándole el id que hemos obtenido anteriormente, para obtener todos los datos que necesitamos para mostrar al usuario de la aplicación:

```
url: 'http://192.168.1.100/banda_evento.php?banda='+id,
```

Una vez hemos realizado todas las consultas necesarias, y tenemos todos los datos necesarios, llamamos a la función *mostrardatos()*, esta función lo que hará será modificar el valor de algunos HTML de tal manera que su valor sea el que hemos obtenido en las consultas, todo ello antes de que el usuario vea el resultado, para que una vez entre, ya tenga los resultados deseados en pantalla y no tenga que esperar un tiempo hasta verlos una vez dentro:

```
function mostrardatos() {  
    $('#BANDA_imagen').prop('src',cabecera);  
    $('#BANDA_nombre').html(var_nombre);  
    $('#BANDA_ciudad').html(var_ciudad);  
    $('#BANDA_estilo').html(var_estilo);  
    $('#BANDA_sello').html(var_sello);  
    $('#BANDA_componentes').html(var_componentes);  
  
    $("##base").fadeOut('1000');  
}
```

Pese a que como es normal, no siempre se obtienen los mismos datos y la consulta no siempre es realizada de una misma manera o con los mismos parámetros, la muestra de resultados para este tipo de elementos, siempre será de una manera muy parecida, y con una metodología similar.

4.7. Autenticación

Para poder acceder a la parte privada de la aplicación, como es lógico, debemos comprobar que el usuario de la aplicación está registrado dentro de la plataforma Bandness. Para comprobarlo en la pantalla de login, le pedimos tanto su correo electrónico con el cual está registrado, como su contraseña.

Lo primero que vamos a validar es su correo, para comprobar que efectivamente existe, todo ello lo realizaremos mediante PHP que realizará consultas al servidor:

```
$sql = "SELECT id FROM user_base WHERE email='$busqueda'";
```

Una vez hemos comprobado que dicho correo existe, de ser así, pasaremos a la comprobación del password (contraseña) que el usuario ha introducido. De ser el correo



directamente erróneo y no existir, no haría falta tan siquiera comprobar el password, y saltaría un mensaje de error alertando de ello. Posteriormente, si el correo es correcto, realizaremos otra consulta donde obtener el *id* asociado al usuario de dicho correo para poder proceder a la comprobación del password del mismo. Esto se debe a que las tablas donde se encuentran el correo y el password de un usuario son distintas, y lo que las relaciones es el *id* del usuario:

```
$sql = "SELECT id FROM bandee WHERE user_id='$user'";
```

A la hora de realizar la comprobación del password hay que tener en cuenta que para dar una mayor seguridad y confianza a los usuarios de la plataforma, este se encuentra cifrado dentro de la base de datos mediante un algoritmo llamado SHA-1, sucesor del SHA (Secure Hash Algorithm) [44]. Este algoritmo es extremadamente fiable en la actualidad ya que no se conoce ningún ataque efectivo contra el mismo que permita descifrarlo. Dicho esto, la base de datos trabaja de la siguiente manera, mediante la consulta que realizamos con el password introducido por el usuario:

```
function checkPass($pass,$var)
{
    return $var["pass"] == sha1($var["salt"].$pass);
}

$id = $_GET['id'];
$pass = $_GET['pass'];

//cadena de conexion
$databse = "Bandness";

-----

mysql_query('SET CHARACTER SET utf8');

$sql = "SELECT * FROM user_base WHERE id='$id'";
$result = mysql_query($sql) or die("Query error: " . mysql_error());
$var=mysql_fetch_array($result);

$resultado[['Resultado']] = checkPass($pass,$var);
```

En primer lugar, lo que hacemos es declarar la función *checkPass*. Esta función lo que hace es que una vez recibidos los atributos del usuario obtenidos del *id* que se ha recibido como parámetro de la consulta, se llama a la función PHP *sha1*. En este caso el valor que nos interesa de la tabla del usuario es el *salt*. Este valor es simplemente una serie de bits aleatorios que forman un string [45], los cuales mediante aplicar una clave, a cual es en este caso el parámetro password que introduce el usuario, dan como resultado el password encriptado que se almacena en el la misma tabla de usuario con el nombre de *pass*.

Si el password introducido por el usuario, una vez realizada la encriptación mediante la función *sha*, no diera como resultado el que se encuentra en *pass* eso querría decir que el password introducido para hacer el login en la aplicación no es el correcto. De ser así tal como pasa como el correo, saldría un mensaje de aviso indicando que pese a que el correo pudiera ser correcto, el password del mismo es incorrecto.

Por el contrario, si todas las comprobaciones son correctas, el usuario entraría al menú de usuario de la aplicación. El valor del *id* del usuario obtenido en la comprobación será de aquí en adelante almacenado ya que será importante a la hora de las distintas consultas que se realizarán porque tendrán como referente dicho usuario. De esta manera, por ejemplo, se podrán modificar valores dentro de la base de datos, introducir nuevos, o borrar, dependiendo de las acciones que el usuario realice.

Como se comento previamente, la aplicación una vez el usuario haya hecho login correctamente, permitirá que la sesión del usuario permanezca abierta a no ser que el mismo la cierre, esto es gracias al almacenamiento local de algunos parámetros, como veremos a continuación, uno de ellos será el *id*, el cual de esta manera no será necesario pasarlo siempre a través de la url, si no simplemente leyendo una variable.

4.8. Almacenamiento de datos en memoria local

Nuestro navegador dispone de una pequeña cache de datos, donde podemos almacenar variables de manera persistente, para así no tener que estar constantemente enviándolas a través de la url. Esto además es poco seguro ya que dichos valores podrían ser robadas, y mientras que a lo mejor el *id* de un evento es un valor que carece de mucha importancia al ser una tabla pública, no así sería por ejemplo el *id* de un usuario, o su *pass*, pues esa información es totalmente privada y debe ser preservada por todos los medios posibles.

Esto es muy parecido a lo que se conoce en los navegadores como *cookies* [46], pequeñas piezas de información de un sitio web, almacenadas en el navegador del usuario. Se podría decir que el *localStorage*, es una versión más avanzada para almacenar los datos, gracias al HTML5.

Lo primero que hay que hacer es insertar los datos que queremos almacenar en el *localStorage*, con un *setItem*, mediante pares. El par está compuesto por una clave (key) que indicará el lugar (o dicho de otra manera, el nombre tendrá a partir de ahora dicho valor cuando queramos recuperarlo), y por un valor (el dato):

```
localStorage.setItem("user_id",id);
localStorage.setItem("pass",pass);
```

En este caso estamos guardando la *pass* del usuario, así como su *id* dentro de Bandness en nuestro navegador, para en vez de tener que solicitarlos cada vez, tanto al usuario el *pass*, como el *id* al servidor. Una vez almacenadas su recuperación será tan fácil como realizar un *getItem*, indicándole la clave y nos devolverá el valor que lleva almacenado:

```
localStorage.getItem("user_id");
```

4.9. Tratamiento de eventos

La parte más compleja y probablemente más importante de la aplicación es la que se refiere a los eventos. Mientras que la mayoría de interacciones entre el usuario y el servidor son simples consultas, es en este caso donde también existe la posibilidad de insertar nuevos



elementos en la base de datos, modificar o eliminar. Todo ello será posible debido al hecho de que un usuario puede confirmar su asistencia o eliminarla, a un evento, lo cual conlleva una serie de acciones dentro de la base de datos para darle solidez a dicha acción.

Cuando un usuario entra a pantalla del evento, hay dos opciones posibles. Puede haber accedido mediante su propio menú por lo cual es un evento el cual ya tiene con seguridad confirmada su asistencia (si no, no le aparecería dicho evento). El otro caso sería, si ha accedido mediante el buscador o gracias a los eventos a los cuales asistirán sus amigos, de esta manera su asistencia podría no estar confirmada y por tanto, realizarla.

4.9.1. Confirmar asistencia

Pese a que tanto la acción de confirmar asistencia, como la de cancelarla, se realizan de igual manera, las acciones que se llevan a cabo por ello son totalmente distintas. Si ponemos por caso que accedemos a un evento al cual no hemos confirmado nuestra asistencia en un principio, nos aparecerá la imagen que podemos ver a continuación (figura 35):

Figura 37. Botón para confirmar asistencia evento

Este es el botón que el usuario deberá pulsar para que su asistencia al evento sea confirmada en la base de datos. Un usuario no registrado no puede realizar esta acción ya que no está registrado en la base de datos, y por tanto la acción que realizaríamos de incluir dicha asistencia no sería consistente con ella. Si un usuario no registrado tratará de realizar la acción, se mostraría un mensaje de aviso indicándole que debe acceder a la aplicación haciendo login en ella, para poder realizarla.

Si el usuario pulsará dicho botón para realizar la confirmación, se ejecutaría un código que daría como resultado la ejecución de la siguiente instrucción SQL mediante código PHP:

```
$sql = "INSERT INTO event_attendance (id, user_id, event_id, attendance) VALUES (',$user_id','$event_id','1)";
```

Como se puede apreciar, se realiza una inserción en la tabla *event_attendance*, donde el valor *id* se genera de manera automática sin existir dos iguales dentro de la tabla. El valor de *user_id* corresponde al *id* del usuario que confirma su asistencia. El valor de *event_id* es el *id* del evento al cual se confirma asistencia. Finalmente el valor *attendance* tendrá valor constante de 1.

Tras esta acción se producen dos consecuencias inmediatas:

- En cuanto al evento en sí, el valor de la cantidad de usuarios que va a asistir al mismo se incrementa automáticamente en uno. Esto es debido a que dicha cuenta se realiza mirando la tabla *event_attendance*, y obteniendo el número de valores (la cuenta) en los cuales el *event_id* es el mismo que el del evento en cuestión:


```
$sql3 = "SELECT count(*) as cuenta FROM event_attendance WHERE event_id = '$evento';"
```

- Si nos referimos al usuario, la acción realizada se verá reflejada a la hora de mostrar los resultados de los eventos a los cuales él mismo va a asistir tanto en su diario como en su sección de conciertos. De igual manera que en los eventos, se mira la tabla *event_attendance*, para llevar un control de los mismo, en este caso comparando el *user_id* con del usuario:

```
$sql = "SELECT id,picture,name,date_init,venue FROM event WHERE id IN (SELECT event_id FROM event_attendance WHERE user_id = '$usuario)";"
```

Por descontado queda que estas mismas consecuencias, se producirán de eliminar la confirmación de asistencia al evento, aunque ello en este caso se producirá no por la inserción de un nuevo valor en la base de datos, si no por su eliminación.

4.9.2. Cancelar asistencia

Si entramos a la pantalla de un evento, cuya asistencia nosotros previamente hemos confirmado, en vez de la imagen mostrada anteriormente nos encontraremos con esta otra, en el botón de asistencia (figura 30):

Un botón rectangular con un fondo azul claro y un borde azul más oscuro. El texto "Asistencia confirmada" está centrado en el botón en una tipografía blanca.

Figura 38. Botón para cancelar asistencia evento

En consecuencia acción que nosotros realizaremos si pulsamos dicho botón, no será la de confirmar, si no lo contrario, eliminar la confirmación que previamente hemos introducido en la tabla *event_attendance*, lo cual se realizará mediante el siguiente código:

```
$sql = "DELETE FROM event_attendance WHERE user_id = '$valor' AND event_id='$event_id';"
```

De esta manera borraremos la confirmación, donde el *user_id* se corresponda al *id* del usuario en cuestión, y además el *event_id* corresponda al *id* del evento en el cual nos encontramos en ese momento. A continuación y como hemos indicado antes, se verá también modificados tanto el listado de eventos a los que acude el usuario, como la cantidad de asistentes que tendrá este evento en particular.

5. Distribución y mercado

Una vez terminada la aplicación, o bien ya disponemos de una versión que cumple las características fundamentales que habíamos exigido, es cuando tenemos que comenzar su distribución en los distintos markets, tanto de Android, como iOS, Blackberry, etc. Pero cada uno de ellos funciona de una manera distinta, y tanto sus requisitos, como su coste pueden variar considerablemente.

En primer lugar, como se ha indicado a la hora de la implementación, la aplicación se ha desarrollado en un entorno Android, utilizando un dispositivo de este S.O. para los test previos. Esto quiere decir, que en este momento únicamente disponemos de una aplicación ejecutable en dispositivos Android, pero para solventar este, a priori, problema, phonegap presenta una herramienta de gran utilidad llamado *Adobe® PhoneGap™ Build* [47]. Se trata de un servicio en la nube el cual permite compilar la aplicación (subiendo los ficheros de la carpeta *assets*, así como un archivo *config.xml* y los distintos logos de la aplicación por tamaño) generando distintos ejecutables para cada uno de los S.O. existentes.

Lo único que es necesario para obtener dichos ejecutables son unas claves de desarrollo en iOS, Android y Blackberry (todas ellas gratuitas y que puede generar cualquier usuario), y absolutamente nada en los demás S.O., ahora bien, una vez obtenidos estos ejecutables y llegado el momento de subir las aplicaciones a los distintos markets, la cosa cambia.

Por un lado tenemos los markets de Android y iOS, que solicitan una cuota de registro para que el usuario pueda subir sus aplicaciones (19.95 \$ aprox. Android para siempre, y 99 \$ iOS anuales). Como se indicó en un punto previo, ya vemos claramente como iOS deja a las claras que las aplicaciones son un negocio en el cual sacan un gran beneficio económico, es por eso también que son muy estrictos a la hora de aceptar que una aplicación sea distribuida.

Cada una de las aplicaciones que se suben deben pasar unos controles y revisiones que duran aproximadamente una semana hasta que se obtiene un resultado. Si incumple alguna de las condiciones, ya sea de distribución, contenidos, diseño, etc. la aplicación es rechazada. En cambio Android, pese a también ser de pago, sus controles son mucho menores, y solo verifican los contenidos que sean acordes, por lo cual su publicación es mucho más sencilla e inmediata.

Por otro lado, otras compañías como Blackberry, permiten a sus usuarios subir aplicaciones sin coste alguno, aunque también, pasando una serie de controles de calidad que verifiquen la aplicación antes de ser publicada.

Existen maneras alternativa de distribuir la aplicación, por ejemplo mediante códigos QR, un usuario puede gracias a simplemente a la cámara integrada en el móvil, conexión a internet y un programa específico para reconocer dichos códigos, descargar la aplicación e instalarla en su dispositivo.

No hace falta decir, que gran parte del usuario tipo que descargará esta aplicación, serán usuarios de la propia plataforma Bandness, para poder obtener servicios exclusivos, no disponibles en la web. Pero también hay que tener en cuenta muchos jóvenes o incluso gente no tan joven ya, que sin saber que hacer un día, utilicen la aplicación para buscar un plan que realizar o que estén aburridos y busquen escuchar nueva música.

Podríamos indicar, pese a no poder realizar un estudio de mercado con datos de descarga de la aplicación, que la misma va dirigida a personas con edades comprendidas entre los 18 – 45 años de toda clase de gustos musicales y clases sociales ya que con la música no existen excepciones, a todos nos gusta.

6. Conclusiones

Finalmente, podemos decir que la parte principal e imprescindible de la aplicación está terminada y en funcionamiento. Se ha conseguido que tanto la parte buscador de la aplicación así como el almacenamiento de información y recuperación de la misma por parte de un usuario se lleve a cabo con total normalidad, sin problemas de carga y controlando muy bien que la base de datos se comporte de una manera correcta.

Para concretar dicho proyecto y poder finalizarlo con éxito, el trabajo ha estado estructurado de tal manera que primero hemos hablado de lo que nos ha llevado al día de hoy y cómo han evolucionado los terminales para ser capaces de hacer lo que estábamos buscando realizar. Hemos analizado un poco más los S.O. predominantes y vistió la gran ventaja que tiene en nuestro caso el método de desarrollo establecido, así como presentar el framework que nos ha ayudado a realizar dicha aplicación. Finalmente presentamos el caso práctico de la aplicación de BandEvent con el framework Phonegap, una aplicación que muestra perfectamente la utilización de un lenguaje universal dentro de las aplicaciones móviles, y cómo es posible desarrollar con independencia del dispositivo en que se ejecute con un resultado vistoso, elegante y fluido en su ejecución.

Cabe destacar, una vez finalizado su desarrollo principal, que hay puntos de la aplicación que han quedado bien encaminados para ser desarrollados e implementados en un futuro no muy lejano, como por ejemplo las notificaciones al Smartphone avisando de actualizaciones o el buscador de bandas de la plataforma Bandness. Además a medida que el proyecto iba avanzando, aparecían nuevas opciones y posibilidades que incluir en versiones posteriores, y que no estaban previstas en un principio como la posibilidad de subir imágenes de los eventos, o comprar entradas de los mismos. Es decir, que pese a que había una estructura clara de trabajo y de lo que se quería conseguir, la propia aplicación ha ido evolucionando y dando lugar a nuevos servicios que implementar en un futuro.

Otro aspecto destacable es que el diseño de la aplicación es fácilmente editable, lo cual hace que si para la versión final que será publicada, se desea modificar aspectos del mismo o adaptarlos a una nueva imagen corporativa, la forma en la que la aplicación está desarrollada facilita que se pueda llevar a cabo con comodidad.

Durante todo el trabajo de la aplicación ha sido fundamental y de gran importancia, toda la información y ayuda obtenida a través de las propias páginas de documentación de phonegap, ya que me ha permitido la solución de problemas puntuales surgidos durante la misma, sin necesidad de buscar soluciones complejas. Tengo que decir que en lo que se refiere a este aspecto, he tenido una grata sorpresa al ser capaz de solucionar prácticamente por cuenta propia cualquier problema surgido, tanto en el diseño de la aplicación, como en su funcionamiento interno.

Pese a que al principio, la programación resultaba complicada y laboriosa, dado que se tratan aspectos y lenguajes bastante ajenos a la ingeniería informática como tal, y que no son muy estudiados o utilizados en la carrera, los primeros meses de trabajo fueron complicados, ya que fueron más una labor de aprendizaje, que de desarrollo de la aplicación como tal. Una vez superados estos meses, y cogido un ritmo de trabajo, la aplicación fue poco a poco tomando

forma, y la programación de la misma era cada vez más rápida. Errores que al principio parecían no tener sentido, eran fácilmente explicados una vez que mis conocimientos y trabajo iban aumentando.

Puedo decir finalmente, y para concluir, que estoy muy satisfecho del trabajo realizado. En primer lugar por la ayuda que he encontrado de la empresa Bandness, que me ha ayudado tanto a la hora del desarrollo, aportándome los conocimientos necesarios para superar distintos problemas que se planteaban, como por su labor de implicación en la misma. Y también satisfecho conmigo mismo, por ser capaz de aprender nuevos conocimientos, en un mundo nuevo y en el que deseo tener un camino por delante como es el del diseño web y el de las aplicaciones móviles. Con este proyecto he obtenido ambas cosas, aprender cómo realizar una aplicación web, pero además ser capaz de realizar las modificaciones y la programación para que su ejecución estuviera adaptada a un dispositivo móvil.

Así que no me queda otra cosa ya más que añadir, salvo que me alegro mucho de haber sido capaz de cumplir los objetivos conseguidos, y de haber podido tener los medios técnicos y personales de ayuda necesarios cada momento que ha sido necesario.

7. Referencias

- [1] Los Smartphone y una breve explicación, http://es.wikipedia.org/wiki/Tel%C3%A9fono_inteligente
- [2] Estadísticas de la penetración de los Smartphone en la población de los países desarrollados, <http://communities-dominante.blogspot.com/2011/12/smartphone-penetration-rates-by-country-we-have-good-data-finally.html>
- [3] Edad media de adquisición del primer teléfono móvil y Smartphone entre jóvenes de 10 a 16 años, http://www.inteco.es/indicadores/Seguridad/Observatorio/Indicadores/Indicador_INT222
- [4] Guía de html, <http://sestud.uv.es/manual.esp/>
- [5] Bandness – Plataforma de música independiente, <http://bandness.com/>
- [6] La evolución de la telefonía móvil, <http://www.eveliux.com/mx/la-evolucion-de-la-telefonía-movil.php>
- [7] Las generaciones de la telefonía móvil, <http://linkea.do/2011/05/07/evolucion-de-la-tecnología-movil-1g-2g-3g-4g/>
- [8] ¿What is IS-136?, <http://www.home.agilent.com/agilent/editorial.jsp?cc=ES&lc=eng&ckey=1000003136:epsg:faq&nid=-11143.0.00&id=1000003136:epsg:faq>
- [9] CDMA, http://www.upv.es/satelite/trabajos/Grupo3_99.00/GlobalStar8.htm
- [10] Personal Digital Cellular, http://es.wikipedia.org/wiki/Personal_Digital_Cellular
- [11] GPRS, <http://www.gsmSpain.com/glosario/?palabra=GPRS>
- [12] Evolución en los teléfonos, <http://www.muymovil.com/2009/10/10/la-evolucion-de-los-telefonos-moviles>
- [12] IMT-200, <http://es.wikitel.info/wiki/IMT-2000>
- [13] Cuota de Mercado S.O. para móviles, http://www.impulsonegocios.com/contenidos/2012/08/09/Editorial_21252.php
- [14] Sistemas operativos móviles, <http://celulares.about.com/od/Smartphones/a/Sistemas-Operativos-De-Celulares-En-La-Lupa.htm>
- [15] Rom en Android, http://tecnología.facilísimo.com/blogs/general/sabes-lo-que-es-una-rom-en-android_639321.html
- [16] Fragmentación en Android, <http://www.genbeta.com/sistemas-operativos/el-problema-de-la-fragmentacion-de-android-las-actualizaciones>

- [17] iOS Developer Program, <https://developer.apple.com/programs/ios/>
- [18] Desarrollo con Symbian, <http://javiercancela.com/2007/10/17/introduccion-al-desarrollo-de-aplicaciones-para-telefonos-moviles-symbian/>
- [19] BlackBerry en el mercado Español, <http://www.elreferente.es/tecnologia/triunfo-de-rim-en-el-mercado-espanol-19892>
- [20] BlackBerry Developers, <http://es.blackberry.com/developers/javaappdev/>
- [21] Windows Phone Developers, <https://dev.windowsphone.com/en-us>
- [22] Phonegap, <http://www.phonegap.com/>
- [23] Phonegap API Reference, <http://docs.phonegap.com/en/2.0.0/index.html>
- [24] Titanium mobile, <http://www.appcelerator.com/platform/>
- [25] Adobe Air, <http://www.adobe.com/es/products/air.html>
- [26] Corona SDK, <http://www.coronalabs.com/products/corona-sdk/>
- [27] Getting started with Android, http://docs.phonegap.com/en/2.0.0/guide_getting-started_android_index.md.html#Getting%20Started%20with%20Android
- [28] Android SDK, <http://developer.android.com/sdk/index.html>
- [29] ADT Plugin, <http://developer.android.com/sdk/eclipse-adt.html#installing>
- [30] Last version of Cordova (Phonegap), <http://phonegap.com/download>
- [31] Lanzamiento plataforma Bandness, <http://www.upv.es/noticias-upv/noticia-5165-bandness-es.html>
- [32] How the MoSCoW method helps you priotize goals, <http://www.bizengine.com/moscow-method/>
- [33] Tag HTML div, <http://www.htmlquick.com/es/reference/tags/div.html>
- [34] Facebook, www.facebook.com
- [35] Twitter, www.twitter.com
- [36] PHP, <http://www.php.net/>
- [37] jQuery.ajax(), <http://api.jquery.com/jQuery.ajax/>
- [38] Aprender SQL, <http://www.w3schools.com/sql/default.asp>
- [39] JavaScript, <http://www.w3schools.com/js/default.asp>
- [40] HTML, http://www.w3schools.com/html/html_intro.asp
- [41] CSS, <http://www.w3schools.com/css/>
- [42] Protocolo HTTP, <http://acsblog.es/articulos/trunk/LinuxActual/Apache/html/x49.html>



- [43] JSON, <http://www.json.org/json-es.html>
- [44] Secure Hash Standard, <http://www.itl.nist.gov/fipspubs/fip180-1.htm>
- [45] Salt (cryptography), [http://en.wikipedia.org/wiki/Salt_\(cryptography\)](http://en.wikipedia.org/wiki/Salt_(cryptography))
- [46] HTTP_Cookies, http://en.wikipedia.org/wiki/HTTP_cookie
- [47] Adobe® PhoneGap™ Build , <https://build.phonegap.com/>
- [48] Códigos QR, <http://www.codigo-qr.es/>