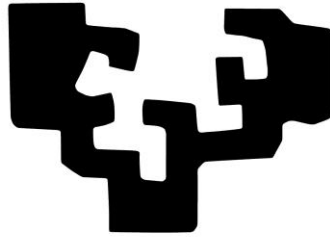


eman ta zabal zazu



Universidad
del País Vasco

Euskal Herriko
Unibertsitatea

Facultad de Informática / Informatika Fakultatea

**Aplicación nativa en Android del Festival de cine de
San Sebastián – Donostia Zinemaldia**

Alumno: D. Ramón Velasco López

Director: D. José Miguel Blanco Arbe

Proyecto Fin de Carrera, julio 2013

Agradecimientos

Quiero agradecer en primer lugar a mi familia al completo, su apoyo y sus ánimos incondicionales. Mis padres Ramón y Carmen, mi hermano Pablo, y Marina, va por ellos.

También tengo que agradecer al Festival de Cine la gran oportunidad que me han brindado, a todo el equipo y sobre todo agradecer la ayuda y consejos que me han prestado Mikel y Joseba del departamento de Sistemas. Un honor trabajar con vosotros.

No menos importante la ayuda que me ha prestado mi director de proyecto José Miguel, al cual agradezco enormemente todos sus consejos y sabiduría.

Y sobre todo, gracias a todos los que habéis compartido a lo largo de estos meses esta idea conmigo, expresando vuestras opiniones, ideas, locuras, genialidades, etc.

Resumen

Este proyecto describe la creación de una aplicación para el Festival de Cine de San Sebastián. Lo dispuesto a continuación explica y expone el diseño, y desarrollo desde cero, de la aplicación y relata los contenidos que dispone, las características que tiene y los requisitos que debe cumplir como aplicación destinada a una empresa.

El lector encontrará una serie de capítulos que divulgan los acontecimientos que han surgido en este proyecto.

Entre otros, encontrará cómo se ha realizado la planificación, el diseño y desarrollo, o la implementación, así como las pruebas y las conclusiones obtenidas con ellas.

Índice

1. Introducción	1
2. Antecedentes	2
3. Objetivos del proyecto	4
3.1. Motivación	4
3.2. Objetivos	4
3.3. Alcance	4
3.4. Estructura de Desglose del Trabajo	6
3.5. Relaciones con empresa	7
3.5.1. Requisitos	7
3.5.2. Modificaciones	7
3.5.3. Exigencias de diseño	7
3.5.3.1. Estilos, menús, texto, color	7
3.5.3.2. Logotipos e imagen comercial	8
3.5.3.3. Contenidos y funcionalidades	9
3.5.4. Promoción y marketing	9
4. Estudio de alternativas y tecnologías	11
4.1. Formación Android, historia de Android, Android frente a otros SO	11
4.2. Android Development Tool ADT	16
4.2.1. Interfaz java	16
4.2.2. Actividades	18
4.2.3. Uso de las herramientas gráficas	20
4.3. Estudio tecnologías php, json	21
4.4. Eclipse, Dispositivos, versiones	22
4.5. Firma de la aplicación, publicación, Google Play	24
4.6. Restricciones y decisiones de plataforma	25
5. Funcionalidades aplicación	26
5.1. Objetivo de la aplicación	26
5.2. Requisitos y navegación	26
5.3. Estructuración. Arquitectura de la aplicación	28
5.4. Diseño e implementación	28
5.4.1. Menús	29
5.4.2. Grafismo	29
5.4.3. Características de arranque, logo	29
5.4.4. Diseño e implementación de las funcionalidades	30
5.4.4.1. Noticias y redes sociales	30
5.4.4.2. Galería fotos y vídeos	32
5.4.4.3. Programación	34
5.4.4.4. Secciones	39
5.4.4.5. Premios y premiados	40
5.4.4.6. Entradas	41
5.4.4.7. Mapa	41
5.4.4.8. Configuración	42
5.4.4.9. Multilingüismo	43
5.4.4.10. Otros objetos	43
5.5. Validación	44
5.5.1. Descripción, test	44

5.5.2. Resultado	45
6. Datos	46
6.1. Leer/obtener datos	46
6.1.1. Conexión base datos	46
6.1.2. PHP, Json	46
6.1.3. Consultas SQL	47
6.2. Crear datos	47
6.3. Protección de datos	47
7. Interfaz usuario, Interacción persona computador	48
7.1. Usabilidad	48
7.2. Accesibilidad	48
7.3. Pruebas	48
7.4. Características de la interfaz	48
7.5. Visualización y desarrollo	49
8. Gestión del Proyecto	52
8.1. Estimación de Tiempo	52
8.2. Hitos	55
8.3. Análisis de Riesgos	56
8.4. Análisis de Costes	57
8.5. Análisis de Adquisiciones	57
8.6. Análisis de Comunicación	59
8.7. Análisis de Cambios	59
8.8. Otros análisis	60
8.9. Control y seguimiento	60
9. Conclusiones y líneas futuras de trabajo	62
9.1. Concordancia entre resultados y objetivos	62
9.2. Líneas futuras. Creación de nuevas características. I+D	62
9.3. Capacidad de aumento	63
9.4. Lecciones aprendidas	63
9.5. Conclusiones	63
10. Bibliografía	65
Anexo I. Manual del desarrollador	67
Anexo II. Manual del usuario	70

Índice de figuras

Fig. 1: Menú principal de las aplicaciones no nativas hechas en iBuildApp	3
Fig. 2: Esquema de desglose de trabajo	6
Fig. 3: Análisis de colores	8
Fig. 4: Logo del Festival de cine de San Sebastián	8
Fig. 5: Cabecera del menú principal portrait	9
Fig. 6: Cabecera del menú principal landscape	9
Fig. 7: Uso de las APIs de Android.	13
Fig. 8: Volumen de tráfico de datos	13
Fig. 9: Estimación de venta de Smartphones.	14
Fig. 10: Gestión de ciclo de vida de la interfaz	19
Fig. 11: Gráfico de ventas de los dispositivos Android	23
Fig. 12: Gráfico del uso de las diferentes versiones de Android.	23
Fig. 13: Menú principal	29
Fig. 14: Logotipo de la aplicación	30
Fig. 15: Pantallas de navegación en Noticias	32
Fig. 16: Pantallas de navegación en Multimedia	34
Fig. 17: Pantallas de navegación en Programación	38
Fig. 18: Pantalla de selección y visualización de la programación	39
Fig. 19: Diseño de secciones y películas	39
Fig. 20: Pantallas de navegación en Premios y jurados	40
Fig. 21: Pantalla de entradas	41
Fig. 22: Pantalla del mapa	42
Fig. 23: Pantalla de configuración	42
Fig. 24: Slider	44
Fig. 25: Pantallas portrait y landscape del menú principal	51

Índice de tablas

Tabla 1: Comparativa de SO Móviles	11
Tabla 2: Estimación de tiempos y tiempo necesario real	54
Tabla 3: Resumen de tiempos totales	54
Tabla 4: Cronograma del proyecto	55
Tabla 5: Hitos principales del proyecto	55
Tabla 6: Cuantificación de riesgos	56

1. Introducción

En este documento se describe el Proyecto Fin de Carrera realizado por Ramón Velasco en el Festival de Cine de San Sebastián bajo la supervisión de Mikel Ormaetxea y dirigido por el profesor de la UPV/EHU José Miguel Blanco. El proyecto consiste en la realización de la aplicación del Festival de Cine para dispositivos móviles Android.

La memoria tiene la estructura que se detalla a continuación:

El segundo capítulo muestra los antecedentes, ideas y trabajo que se realizó antes de empezar dicho proyecto.

El tercer capítulo muestra los objetivos del proyecto en el que se presenta la motivación que ha llevado a cabo la realización de este proyecto, la descripción completa del proyecto y sus objetivos así como los aspectos referentes a las relaciones empresa-alumno muy importantes en este proyecto dado que es un producto destinado al uso por parte de una empresa y del público general.

En el cuarto capítulo se analizan las tecnologías y herramientas de desarrollo disponibles. También se hace un balance del estado actual del mercado de los Smartphones y sus sistemas operativos, mediante el cual se aportan los razonamientos oportunos para las elecciones de las distintas herramientas y programas.

En el quinto capítulo se describe cómo ha sido el diseño e implementación de la aplicación. En él se exponen los objetivos que deben cumplir la aplicación, las restricciones de desarrollo, diseño, estructuración de la aplicación, características gráficas, diseño e implementación de cada funcionalidad y la validación de todo el desarrollo.

El sexto capítulo describe todo lo referente a los datos del servidor; seguridad y protección de los datos y escritura y lectura de los mismos.

En el séptimo capítulo se presenta un análisis de la capacidad de la interfaz de usuario y se tratan aspectos tan importantes como la usabilidad, la correcta navegación, la accesibilidad, capacidad de mejora del interfaz, así como pruebas con diferentes grupos de usuarios. Además, se tratan aspectos de visualización y desarrollo de interfaces gráficas.

En el octavo capítulo se hace una descripción de la planificación y la gestión del proyecto en todos sus ámbitos como pueden ser, tiempos, riesgos, comunicaciones, etc. Asimismo, se muestra el control y seguimiento de las actividades llevadas a cabo.

En el noveno capítulo, se analizan las conclusiones del trabajo realizado, líneas de mejora, aumento de las capacidades de la aplicación, etc.

En el capítulo diez, se muestra un listado de la bibliografía utilizada a la hora de crear y desarrollar este proyecto fin de carrera

Por último se incluyen como anexos los manuales del desarrollador y el usuario.

2. Antecedentes

La idea de una aplicación móvil para el Festival de Cine de San Sebastián, surge durante la realización de unas prácticas en el Departamento de Informática del propio Festival durante el verano 2012. En ellas me doy cuenta de que la empresa no dispone de una aplicación para dispositivos móviles que agilice la obtención por parte de los usuarios de la información principal del Festival. Es entonces cuando propongo al responsable de Informática del Festival, Mikel Ormaetxea, crear una aplicación que ponga fin a esta carencia.

En ese momento comienza la recopilación de los requisitos que tendrá que cumplir la aplicación y los requerimientos que se necesitan para llevarla a cabo.

La idea principal de la aplicación era proporcionar información inmediata al usuario, información que durante todo el año y en concreto durante el Festival es cambiante y modificable. También interesaba proporcionar diversos servicios relacionados entre los que destacamos:

- Noticias del Festival, última hora.
- Programación del Festival, horario, lugar, día y sección correspondiente a la película.
- Fotografías y vídeos del festival.
- Secciones con las películas y premios.
- Información de entradas, taquilla virtual.
- Enlaces a redes sociales.
- Contacto con el festival.

Junto con el responsable de Informática del Festival, defino varios requisitos que debería cumplir la aplicación para que sea útil, sencilla y que ofrezca todo lo necesario para un usuario corriente.

Tras una primera valoración de cómo debería ser la aplicación y los contenidos que debe poseer, valoramos la plataforma para la cual desarrollarla. En un principio se pretende realizar la aplicación para dispositivos ANDROID y IOS. Viendo el poco tiempo del que disponía para aprender Objective-C, desistí de hacerla para IOS y centrarme en desarrollar un primer borrador de la aplicación en JAVA para dispositivos ANDROID. Con lo que me puse a investigar cómo hacerlo, y cómo desarrollar de manera simple la primera aplicación.

En un primer momento, localicé varias páginas web que disponían de asistentes para la creación de aplicaciones multi-plataforma de manera sencilla, con algunas funcionalidades pre-diseñadas, y plantillas para diferentes tipos de aplicaciones. Con ello realicé un primer borrador en una de ellas, iBuildApp.com, véase Fig. 1



Fig. 1: Menú principal de las aplicaciones no nativas hechas en iBuildApp

Lo cual era una solución bastante aceptable. Pero no permitía trabajar con el código y dotar a la aplicación de un diseño más personalizado y con funcionalidades más completas, cosa que requiere el Festival.

Lamentablemente, el tiempo no me permitió continuar y no pudimos avanzar en nuestro objetivo de lanzar una aplicación nativa en la edición 60 del Festival (2012), con lo que decidí posponer el desarrollo de la aplicación para el futuro, sin olvidar que podría ser un posible proyecto fin de carrera para el próximo año.

De vuelta en enero de 2013, me pongo en contacto con el responsable de Informática del Festival para retomar el proyecto, haciéndolo además como Proyecto Fin de Carrera, a lo cual me da su aprobación. No sin antes sugerirme que esta vez intente desarrollar el proyecto en las dos plataformas que inicialmente propusimos (ANDROID y IOS).

Después de hablar con mi director de proyecto y profesor de la Facultad de Informática, José Miguel Blanco, observamos que el alcance del proyecto propuesto podía suponer un riesgo de no finalización y poner en peligro la terminación de la carrera. Dicho esto, y después de hablar nuevamente con el responsable de Informática del Festival, valoramos el desarrollo de la aplicación de manera paralela en ambas plataformas (Android e iOS) pero en lenguaje HTML5, o hacerla de forma nativa en JAVA pero sin ofrecer un número suficiente de servicios.

Finalmente me decanto por la tercera opción; desarrollar la aplicación de forma nativa sólo para el sistema operativo de dispositivos móviles Android.

3. Objetivos del proyecto

A continuación se expone la motivación que llevó a realizar el proyecto, así como las características principales del proyecto, entre los que destacan los objetivos y alcance que marcaron el guion del mismo, siempre condicionado por los requisitos del Festival de cine de San Sebastián.

Este capítulo relata los objetivos del proyecto y sus propiedades.

3.1. Motivación

La aplicación nació con la idea de ofrecer un aporte de información adicional al de la web del Festival dentro de las plataformas móviles, y en concreto para el sistema operativo móvil Android.

La posibilidad de ofrecer información variada del Festival de manera inmediata en una aplicación de un dispositivo era y es algo necesario y útil para cualquier usuario o espectador del Festival.

Además, la oportunidad de hacerla destinada a darle un uso real, en conjunto con una institución tan exigente como el Festival de Cine de San Sebastián y enmarcada en el proyecto final de carrera, eran grandes motivadores para la realización del mismo.

3.2. Objetivos

El principal objetivo de este proyecto era la realización de una aplicación nativa del Festival de Cine de San Sebastián, en Java para dispositivos móviles con sistema operativo Android.

Era un objetivo que aportase información variada del Festival, tal como imágenes, noticias, películas, etc. y que dichos contenidos se actualizasen automáticamente. También era un objetivo que dispusiera de traducción a varios idiomas.

El principal objetivo de este proyecto era la realización de una aplicación nativa del Festival de Cine de San Sebastián, en Java para dispositivos móviles con sistema operativo Android.

Además, para la consecución de este objetivo se debían cumplir otros objetivos formativos que se describen a continuación:

- Conocer las herramientas de desarrollo a utilizar, en concreto la plataforma Android y sus posibilidades.
- Adquirir los conocimientos de PHP, JSON, XML, SQL necesarios para las distintas partes de una aplicación.
- Estudio de la herramienta Eclipse, juntos con las librerías de Android necesarias y sus posibilidades de desarrollo.

3.3. Alcance

Recopilados algunos requisitos del Festival de cine de San Sebastián y del Proyecto, a continuación se expone el alcance del proyecto y una descripción del producto a crear.

El alcance del proyecto se resume en el diseño, implementación y realización de una fase de pruebas, de una aplicación nativa desarrollada en el lenguaje JAVA para sistemas operativos de dispositivos móviles Android versión 2.3.3 o superiores.

Esta aplicación, debía disponer de un menú principal con el logotipo del Festival de Cine cuyas características sobre formato, color, tipos y demás elementos del diseño gráfico vendrían dadas y resueltas por el Departamento de Producción del Festival de cine de San Sebastián.

Dentro del menú principal debíamos encontrar diferentes botones que tendrían un diseño intuitivo acorde a la funcionalidad que ofrecía cada uno de ellos. Cada botón representaba una funcionalidad de la aplicación, y a continuación se detalla cada una de las funcionalidades de cada botón:

- **Enlace Noticias:** al pulsar el botón Noticias, se debía acceder a una lista de todas las noticias del Festival que se hallasen en la página web, básicamente debía ser un visor de noticias.
- **Galería Fotográfica:** mediante esta funcionalidad podríamos acceder a otro menú que ofrece la posibilidad de seleccionar diferentes álbumes de fotos del festival.
- **Videoteca:** este servicio debía acceder al álbum de vídeos del Festival y al llamado "San Sebastián Festival TV" que era un servicio que ofrecía la posibilidad de visualizar los vídeo del Festival, galas, y streaming en directo de lo que sucedía.
- **Sección de Contacto:** accediendo a esta sección, se mostraban los principales contactos del festival, direcciones, teléfonos y correo electrónico. Los teléfonos y los e-mails disponían de enlace que ofrecía en el propio dispositivo la llamada telefónica automática a dichos números o la creación de un nuevo correo electrónico en blanco a la dirección clicada.
- **Premios y Premiados:** Sección que daba información sobre los principales premios que se dan en el Festival, así como premiados de años anteriores y premiados actuales. Fotografías y textos del tema.
- **Secciones y categorías cinematográficas:** Este enlace proporcionaba información sobre las secciones cinematográficas del festival, también se mostraban las retrospectivas de la edición en curso y breve explicación de todos los ítems.
- **Buscador de la Programación:** Este servicio proporcionaba un discriminador de búsqueda de la programación mediante variables de fecha, hora, lugar o sección. Mostraría las películas y galas disponibles para los criterios escogidos y daba la posibilidad de acceder a la compra de entradas para dichos actos.
- **Enlace al sistema de tele-venta de entradas:** Era un enlace al servicio web de tele-venta de entradas mediante el propio navegador del dispositivo.
- **Enlace a Twitter:** Enlace directo a la cuenta Twitter del Festival.
- **Enlace a Facebook:** Enlace directo a la cuenta Facebook del Festival

- Enlace a Flickr: Enlace directo a la cuenta Flickr del Festival y sus fotos.
- Mapa interactivo con la señalización de los principales lugares de interés en Donostia-San Sebastián relacionados con el Festival (Kursaal, Teatro Victoria Eugenia, hoteles, etc.).
- Aplicación en los idiomas oficiales del Festival: castellano, euskera e inglés.

Tras el diseño e implementación de la aplicación, se procedió a la realización de diversas pruebas, en las que se analizaron la usabilidad y utilidad de la aplicación, su fiabilidad, respuesta a cambios en las fuentes de información y respuesta de los usuarios frente al uso que daban de ella, además de otras pruebas específicas para cada funcionalidad.

Se atendió especial atención a la Calidad del producto y se realizó un seguimiento completo de las tareas realizadas. También se realizó una planificación detallada de las tareas a realizar, actividades, duración, posibles riesgos, costes y adquisiciones, recursos, comunicación entre las partes y demás aspectos necesarios analizar y planificar para lograr el mejor producto posible optimizando las posibilidades.

3.4. Estructura de Desglose del Trabajo

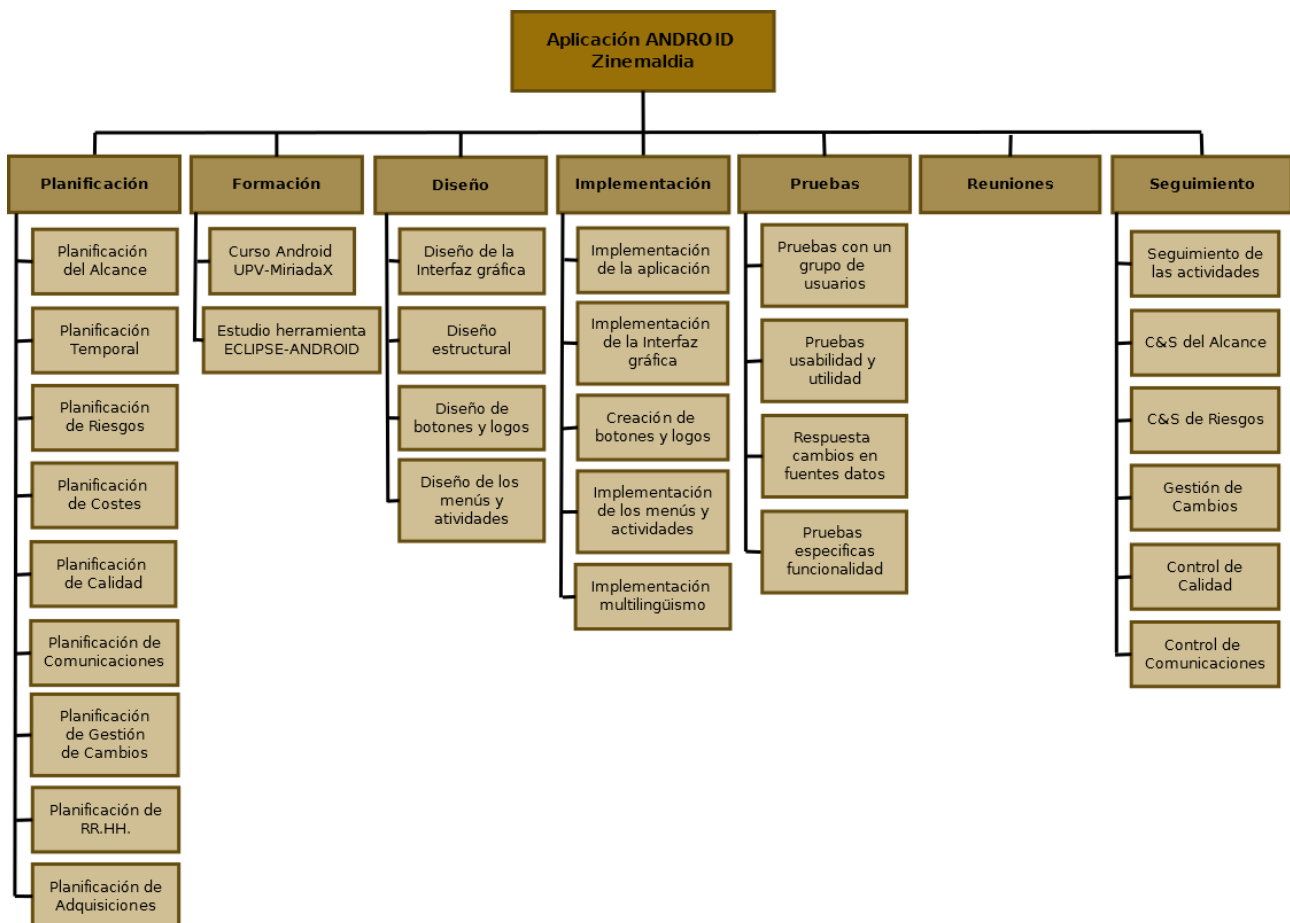


Fig. 2: Esquema de desglose de trabajo

3.5. Relaciones con empresa

Dada la naturaleza del proyecto, las relaciones con la empresa Festival de Cine de San Sebastián-Donostia Zinemaldia ocuparon una parte de los objetivos del proyecto y condicionaron notablemente el camino a tomar dentro del mismo.

Dado que las relaciones se iniciaron hace un año, paso a denotar los requisitos, solicitudes y forma de trabajar que se adoptaron a lo largo de todo el tiempo.

3.5.1. Requisitos

Además de los requisitos mencionados anteriormente, se requirió hacer la aplicación nativa en lenguaje JAVA y disponible en los idiomas principales del Festival de cine de San Sebastián, que son castellano, euskera e inglés. También se requirió que la aplicación estuviera desarrollada para versiones de Android 2.3.3 y superiores.

Asimismo, se requirieron ciertos conocimientos de lenguaje JAVA para desarrollo de aplicaciones Android y conocimientos sobre la obtención de los datos a partir de la página web del Festival de cine de San Sebastián, tales como conocimientos de MySQL y SQL en general, PHP, HTML y demás programas de edición de video e imágenes.

Se requiso realizar la aplicación acorde a las exigencias del Festival en cuanto a diseños, grafismo, estructuración, logotipos, contenidos y demás características.

Se requiso que la aplicación tuviera el tamaño (bytes) mínimo posible y funcionara lo más rápido posible a la hora de enlazar con el servidor y obtener los datos.

3.5.2. Modificaciones

Con regularidad se realizaron modificaciones acordes a lo que solicitó el Festival de cine de San Sebastián, y se tuvo especial atención en cubrir sus exigencias y expectativas, aunque también se tuvieron en cuenta las modificaciones que podían comprometer el cumplimiento del alcance en los plazos y bajo los requisitos inicialmente establecidos.

Las modificaciones se realizaron a medida que con cierta regularidad, fue presentándose el trabajo realizado a los responsables oportunos de Producción e Informática y estos aportaron sus opiniones y requerimientos.

3.5.3. Exigencias de diseño

Existen ciertas exigencias por parte del Festival que se debieron cumplir. Estas exigencias, además de las propias de la aplicación, hacían referencia al estilo, color, imágenes, y otros objetos gráficos que conformaban el diseño gráfico.

En el siguiente apartado se exponen los aspectos del diseño más significativos.

3.5.3.1. Estilos, menús, texto, color

La denominación de la aplicación debía ser la de: Festival de San Sebastián, Donostia Zinemaldia, International Film Festival.

El estilo debía ser sobrio, Android dispone de tipografías que el propio usuario del

dispositivo puede modificar y por tanto se siguió manteniendo, de tal manera que la tipografía propia del dispositivo, fue la de la aplicación.

Los menús fueron breves y a ser posible organizados por agrupación de contenidos en menús como Multimedia, Noticias y social, Secciones y Programación, Premios.

Colores oficiales del festival, a continuación se muestran en la Fig. 3:

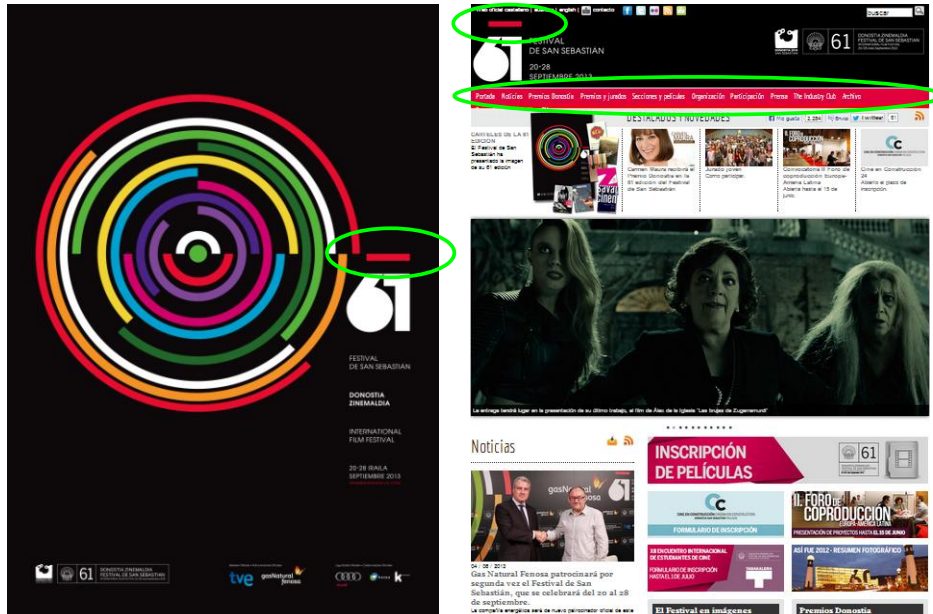
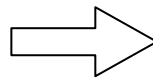


Fig. 3: Análisis de colores

Dado que el grafismo de este año estaba basado en el cartel oficial, el color elegido fue un rojo-rosáceo que está sobre el 61 y en ciertas barras de tareas de la propia página web.

HEX: **#DE093B**
 RGB: r-22 g-9 b-59
 HSV: 346°, 96%, 87%



Los botones y seleccionables ha sido de libre distribución editados y personalizados para el Festival.

3.5.3.2. Logotipos e imagen comercial

El festival centró su logotipo de aplicación en la imagen siguiente:



Fig. 4: Logo del Festival de cine de San Sebastián

Ha sido un botón redondo con la imagen de la concha oficial.

Para el menú principal se disponía de una imagen que daba forma y estructura a la aplicación y hacía las veces de barra introducción. Era personalizada para la próxima 61 edición del Festival de cine.



Fig. 5: Cabecera del menú principal portrait

Para dispositivos en orientación *portrait*.

Para *landscape* disponíamos de la siguiente imagen:

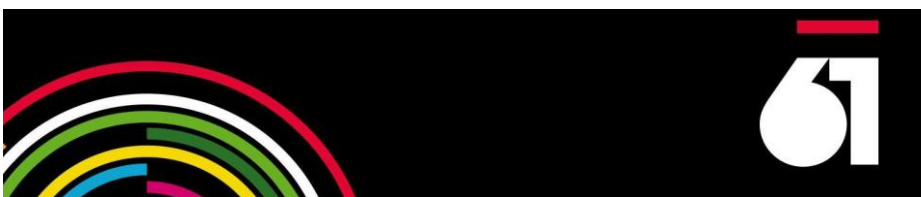


Fig. 6: Cabecera del menú principal landscape

La inclusión de otros logotipos y demás imágenes fueron las oficiales del Festival y siempre con el visto bueno de la persona correspondiente.

La resolución debía tener suficiente calidad como para poder ser correctamente visualizada en cualquier dispositivo.

El formato era .jpg y .png.

3.5.3.3. Contenidos y funcionalidades

La aplicación debía disponer de diferentes funcionalidades que aportasen ciertos contenidos que debían tener un diseño acorde a lo previamente dicho. Se procuró que todos los submenús tuvieran títulos, con barras de inicio, y estilos iguales al resto. Y todo contenido quedase enmarcado.

El comienzo de cada funcionalidad tenía un título con fondo del color del Festival de cine. Y la barra de menú inicial de la 61 edición.

3.5.5. Promoción y marketing

El Festival se reservaba el derecho a colocar cualquier logotipo de patrocinador o marca ajena al proyecto de aplicación y exigió su colocación en lugares acordados.

El Festival podía incluir aspectos que considerase necesarios para mejorar el marketing, mostrar publicidad e imágenes de patrocinadores y colaboradores.

Además, la publicación final, la promoción y publicidad, y uso de su imagen corporativa en la aplicación quedaba a decisión por parte de los responsables del Festival.

Tras la publicación su coste es gratuito.

4. Estudio de alternativas y tecnologías

En el siguiente capítulo, se aborda un estudio de las alternativas que existen actualmente para la consecución de un proyecto como el presente, y se exponen las tecnologías más actuales que hicieron falta a la hora de desarrollar la aplicación.

Además, se proponen las mejores soluciones para cada una de las inquietudes o restricciones que surgieron a lo largo del desarrollo del proyecto.

4.1. Formación Android, historia de Android, uso de Android frente a otros SO

Uno de los objetivos del proyecto era adquirir conocimientos sobre el sistema operativo Android, y destacaba el hecho de conocer las herramientas para desarrolladores que Android nos ofrecía.

En el próximo sub-apartado, se hace una valoración y estudio de la herramienta para desarrollo ADT que se integra dentro de Eclipse.

Ahora en cambio, se realiza una valoración de Android, del mercado de los dispositivos móviles y del porqué de la elección.

Android es un sistema operativo basado en Linux, diseñado principalmente para dispositivos móviles con pantalla táctil como teléfonos inteligentes o tabletas inicialmente desarrollados por Android, Inc., que Google respaldó económicamente y más tarde compró en 2005. Android fue presentado en 2007 junto la fundación del Open Handset Alliance: un consorcio de compañías de hardware, software y telecomunicaciones para avanzar en los estándares abiertos de los dispositivos móviles.

Las unidades vendidas de teléfonos inteligentes con Android se ubican en el primer puesto en los Estados Unidos, en el segundo y tercer trimestres de 2010, con una cuota de mercado de 43,6% en el tercer trimestre. A nivel mundial alcanzó una cuota de mercado del 50,9% durante el cuarto trimestre de 2011, más del doble que el segundo sistema operativo (iOS de Apple, Inc.) con más cuota.

Tiene una gran comunidad de desarrolladores escribiendo aplicaciones para extender la funcionalidad de los dispositivos. A la fecha, se han sobrepasado las 700.000 aplicaciones (de las cuales, dos tercios son gratuitas) disponibles para la tienda de aplicaciones oficial de Android: Google Play.

Fuente: Elgin, Ben (17 de agosto de 2005). «Google Buys Android for Its Mobile Arsenal». Bloomberg Businessweek. Bloomberg y Open Handset Alliance (5 de noviembre de 2007). «Industry Leaders Announce Open Platform for Mobile Devices».

En la siguiente tabla (Tabla 1) se muestra una comparativa entre los principales Sistemas operativos móviles del mercado, y sus características. Entre los aspectos a destacar de Android, se remarca el bajo coste de publicación, las multiplataformas de desarrollo, y la muy alta variedad de dispositivos, pero por encima de todo, el hecho de que es software libre y abierto a todos.

PFC:



	Apple iOS 6	Android 4.2	Windows Phone 7	BlackBerry OS 7	Symbian 9.5
Compañía	Apple	Open Handset Alliance	Windows	RIM	Symbian Foundation
Núcleo del SO	Mac OS X	Linux	Windows CE	Mobile OS	Mobile OS
Familia CPU soportada	ARM	ARM, MIPS, Power, x86	ARM	ARM	ARM
Lenguaje de programación	Objective-C, C++	Java, C++	C#, muchos	Java	C++
Licencia de software	propietaria	software libre y abierto	Propietaria	propietaria	software libre
Año de lanzamiento	2007	2008	2010	2003	1997
Motor del navegador web	WebKit	WebKit	Pocket Internet Explorer	WebKit	WebKit
Soporte Flash	No	Sí	No	Si	Sí
HTML5	Sí	Sí	Sí	Sí	No
Tienda de aplicaciones	App Store	Google Play	Windows Marketplace	BlackBerry App World	Ovi Store
Número de aplicaciones	400.000	300.000	50.000	30.000	50.000
Coste publicar	\$99 / año	\$25 una vez	\$99 / año	sin coste	\$1 una vez
Plataforma de desarrollo	Mac	Windows, Mac, Linux	Windows	Windows, Mac	Windows, Mac, Linux
Actualizaciones automáticas del S.O.	Sí	depende del fabricante	depende del fabricante	Sí	Sí
Soporte memoria externa	No	Sí	No	Sí	Sí
Fabricante único	Sí	No	No	Sí	No
Variedad de dispositivos	modelo único	muy alta	baja	baja	muy alta
Tipo de pantalla	capacitativa	capacitativa /resistiva	capacitativa	/resistiva capacitativa	capacitativa /resistiva
Aplicaciones nativas	Sí	Sí	No	No	Sí

Tabla 1: Comparativa de SO Móviles
Fuente: Curso de android Miriada X.

En la Figura 7 se puede observar el uso de cada una de las versiones de Android y su cuota de mercado. Se puede apreciar claramente el gran número de dispositivos que tienen versiones de Android iguales o superiores a Gingerbread que es la versión 2.3.X, solo Eclair y Froyo son anteriores por lo que el podemos destacar que el 94,5% de los dispositivos Android soportan nuestra aplicación ya que esta aplicación se ha realizado para versiones 2.3.3 en adelante.

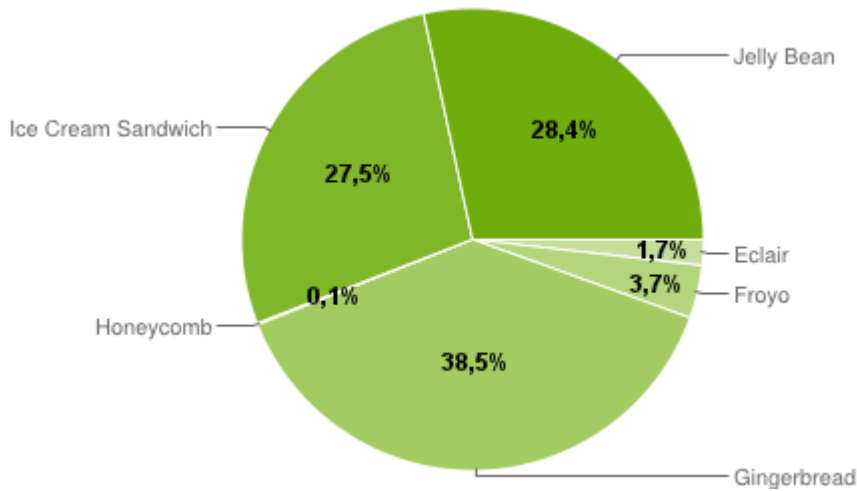


Fig. 7: Uso de las APIs de Android.
 Fuente: Android (<http://developer.android.com/about/dashboards/index.html>)
 Figura editada por Ramón Velasco López (2 de junio de 2013)

En el siguiente gráfico (Fig. 8) se muestra el volumen de tráfico de datos que ha soportado cada sistema operativo móvil durante los últimos cinco años y sus respectivos crecimientos y decrecimientos. Se observa claramente el aumento considerable de Android en contraposición a la bajada de iOS, Symbian y BB OS.

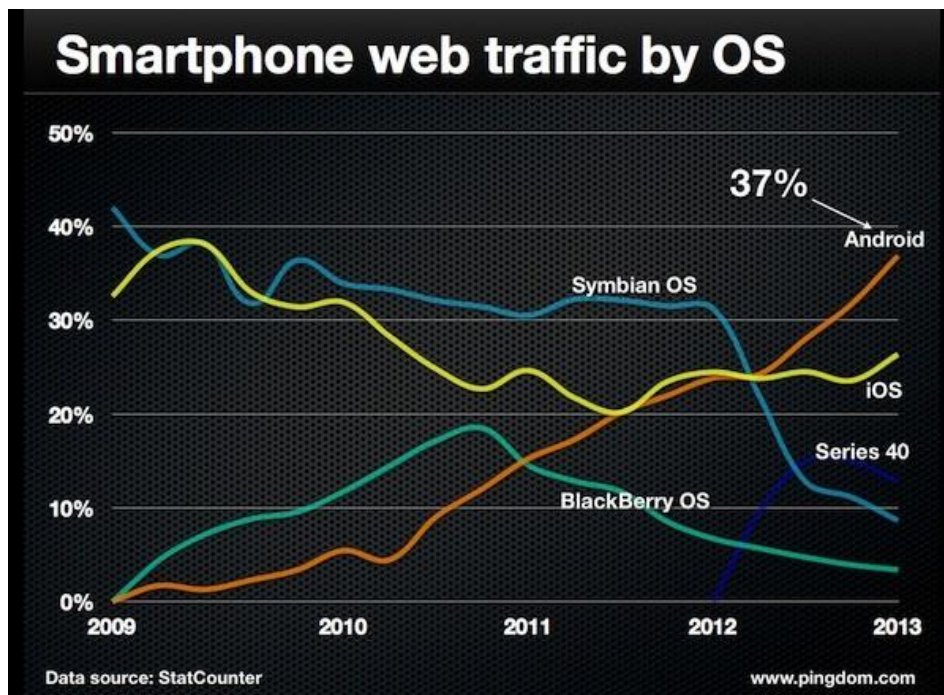


Fig 8: Volumen de tráfico de datos. Fuente: StatCounter (25 de febrero de 2013)

Además, el futuro de Android es prometedor, ya que se espera un aumento de las ventas muy considerable, y convertirse en el sistema operativo por excelencia dentro del mercado de los Smartphones, Tablet y personal computer.

Tal y como se refleja en el gráfico siguiente (Fig. 9), hay una estimación de venta de dispositivos Android de hasta 1500 millones de unidades para el año 2017 en contraste con las ventas de hasta 600 millones de PCs, tablets y teléfonos con (Windows OS y Windows phone OS) y los 500 millones de ordenadores Mac, iPhone y iPad (con MacOS e iOS).

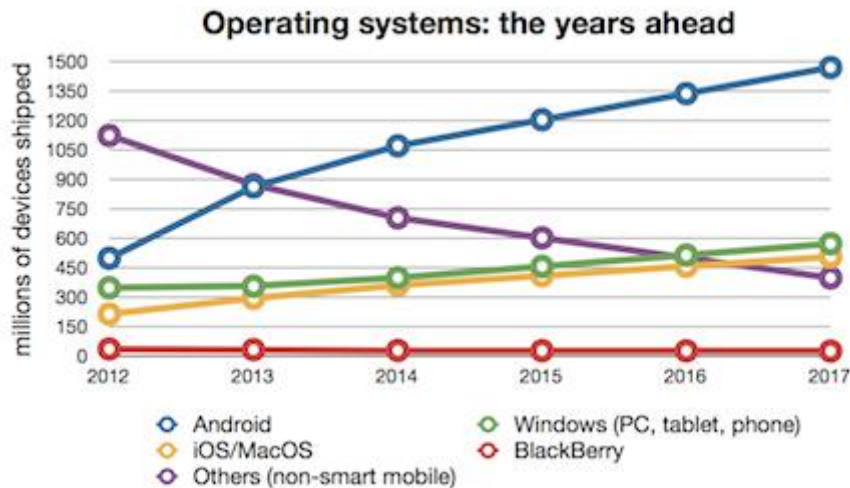


Fig. 9: Estimación de venta de Smartphones. Fuente: The Guardian (4 de abril de 2013)

También se puede observar claramente que los dispositivos no inteligentes están desapareciendo progresivamente aunque seguirán siendo un gran número en 2017(400 millones).

Características de Android

Fuente: Wikipedia (<http://es.wikipedia.org/wiki/Android>). 20/06/2013

- **Diseño de dispositivo:** La plataforma es adaptable a pantallas de mayor resolución, VGA, biblioteca de gráficos 2D, biblioteca de gráficos 3D basada en las especificaciones de la OpenGL ES 2.0 y diseño de teléfonos tradicionales.
- **Almacenamiento:** SQLite, una base de datos liviana, que es usada para propósitos de almacenamiento de datos.
- **Conectividad:** Android soporta las siguientes tecnologías de conectividad: GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, Bluetooth, Wifi, LTE, HSDPA, HSPA+ y WiMAX.
- **Mensajería:** SMS y MMS son formas de mensajería, incluyendo mensajería de texto y ahora la Android Cloud to Device Messaging Framework (C2DM) es parte del servicio de Push Messaging de Android.
- **Navegador web:** El navegador web incluido en Android está basado en el motor de renderizado de código abierto WebKit, emparejado con el motor JavaScript V8 de Google Chrome. El navegador por defecto de Ice Cream Sandwich obtiene una puntuación de 100/100 en el test Acid3.
- **Soporte de Java:** Aunque la mayoría de las aplicaciones están escritas en Java, no hay una máquina virtual Java en la plataforma. El bytecode Java no es ejecutado,

sino que primero se compila en un ejecutable Dalvik y corre en la Máquina Virtual Dalvik. Dalvik es una máquina virtual especializada, diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados. El soporte para J2ME puede ser agregado mediante aplicaciones de terceros como el J2ME MIDP Runner.⁴⁵

- Soporte multimedia: Android soporta los siguientes formatos multimedia: WebM, H.263, H.264 (en 3GP o MP4), MPEG-4 SP, AMR, AMR-WB (en un contenedor 3GP), AAC, HE-AAC (en contenedores MP4 o 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, PNG, GIF y BMP.
- Soporte para streaming: Streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML (HTML5 <video> tag). Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player. Se planea el soporte de Microsoft Smooth Streaming con el port de Silverlight a Android. Adobe Flash HTTP Dynamic Streaming estará disponible mediante una actualización de Adobe Flash Player.
- Soporte para hardware adicional: Android soporta cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, sensores de luz, gamepad, termómetro, aceleración por GPU 2D y 3D.
- Entorno de desarrollo: Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software. El entorno de desarrollo integrado es Eclipse usando el plugin de Herramientas de Desarrollo de Android.
- Google Play: Google Play es un catálogo de aplicaciones gratuitas o de pago en el que pueden ser descargadas e instaladas en dispositivos Android sin la necesidad de un PC.
- Multi-táctil: Android tiene soporte nativo para pantallas capacitivas con soporte multi-táctil que inicialmente hicieron su aparición en dispositivos como el HTC Hero. La funcionalidad fue originalmente desactivada a nivel de kernel (posiblemente para evitar infringir patentes de otras compañías). Más tarde, Google publicó una actualización para el Nexus One y el Motorola Droid que activa el soporte multi-táctil de forma nativa.
- Bluetooth: El soporte para A2DP y AVRCP fue agregado en la versión 1.5; el envío de archivos (OPP) y la exploración del directorio telefónico fueron agregados en la versión 2.0; y el marcado por voz junto con el envío de contactos entre teléfonos lo fueron en la versión 2.2.
- Vídeo-llamada: Android soporta vídeo-llamada a través de Hangouts desde su versión HoneyComb.
- Multitarea: La multitarea real de aplicaciones está disponible, es decir, las aplicaciones que no estén ejecutándose en primer plano reciben ciclos de reloj, a diferencia de otros sistemas de la competencia en la que la multitarea es congelada (Como por ejemplo iOS, en el que la multitarea se limita a servicios internos del sistema y no a aplicaciones externas).

- Características basadas en voz: La búsqueda en Google a través de voz está disponible como "Entrada de Búsqueda" desde la versión inicial del sistema.
- Tethering: Android soporta *tethering*, que permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico (todos los teléfonos desde la versión 2.2, no oficial en teléfonos con versión 1.6 o inferiores mediante aplicaciones disponibles en Google Play (por ejemplo *PdaNet*). Para permitir a un PC usar la conexión de datos del móvil Android se podría requerir la instalación de software adicional.

4.2. Android Development Tools ADT y SDK

Fuente: Wikipedia (http://es.wikipedia.org/wiki/Desarrollo_de_Programas_para_Android). 25/06/2013

El SDK (Software Development Kit) de Android, incluye un conjunto de herramientas de desarrollo. Comprende un depurador de código, biblioteca, un simulador de teléfono, documentación, ejemplos de código y tutoriales. Las plataformas de desarrollo soportadas incluyen , Max OS X 10.4.9 o posterior, y Windows XP o posterior. La plataforma integral de desarrollo (IDE, Integrated Development Environment) soportada oficialmente es Eclipse junto con el complemento ADT (Android Development Tools). Además, pueden controlarse dispositivos Android que estén conectados.

Las Actualizaciones del SDK están coordinadas con el desarrollo general de Android. El SDK soporta también versiones antiguas de Android, por si los programadores necesitan instalar aplicaciones en dispositivos ya obsoletos o más antiguos. Las herramientas de desarrollo son componentes descargables, de modo que una vez instalada la última versión, pueden instalarse versiones anteriores y hacer pruebas de compatibilidad.

Una aplicación Android está compuesta por un conjunto de ficheros empaquetados en formato .apk y guardada en el directorio /data/app del sistema operativo Android. Un paquete APK incluye ficheros .dex (ejecutables *Dalvik*, un código intermedio compilado), recursos, etc.

Como comentario, señalar que recientemente ha salido al mercado el Android Studio, una herramienta similar a eclipse con el plugin ADT. Es un entorno de desarrollo basado en *intelliJ IDEA*, que permite hacer lo mismo que antes en Eclipse pero integrando todos los contenidos y herramientas en un programa propio de Android. Está en proceso de implementación e implantación.

4.2.1. Interfaz java UI

El desarrollo de la interfaz de usuario, UI, dentro del entorno de desarrollo Eclipse se hace en lenguaje XML, que permite crear la visualización de tu aplicación dentro del dispositivo.

XML es extensible: Después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna. El analizador es un componente estándar, no necesita un analizador específico para cada versión y esto permite acelerar el desarrollo de aplicaciones y evitar bugs. Por otro lado, Es sencillo entender la estructura de XML y procesarla. Mejora la compatibilidad entre aplicaciones. Podemos comunicar aplicaciones de distintas plataformas, sin que importe el origen de los datos.

Los objetos disponibles de serie además de los aportados por nuevas librerías permiten el desarrollo de UI intuitivas y usables con el usuario. El framework UI de Android proporciona tanto herramientas de dibujo, con las que construir la UI, como una rica colección de elementos ya creados.

En Android utilizamos archivos XML para la declaración de layouts y otros elementos de los que hará uso nuestra aplicación para su correcto funcionamiento. De esta forma, características como la instanciación dinámica se reserva para escenarios más complejos en dónde los elementos de UI no se conocen en tiempo de compilación (por ejemplo seleccionar una opción de un combo box que se ha llenado de acuerdo a datos consultados en Internet).

Cada archivo XML está formado por un árbol de elementos que especifican la manera en la que los elementos de la UI y contenedores se acomodarán para definir la parte visual de un objeto *View*. Los atributos de cada elemento en el XML se llaman propiedades y describen cómo es que deben verse los elementos y cómo debe comportarse un contenedor.

Los nodos, llamados *Container Views* o contenedores, son aquellos que pueden tener otros componentes como hijos y son derivados de la clase *android.view.ViewGroup*.

Un *ViewGroup* normalmente apenas realiza ningún dibujo y es responsable de situar a sus hijos en la pantalla, manteniéndolos en su sitio mientras que el dispositivo cambia de forma o de orientación.

Android nos ofrece la posibilidad de crear la “fachada” de la aplicación mediante ficheros XML. De esta forma, nos podemos concentrar en el aspecto funcional de la aplicación (código Java) o puramente estético (XML).

En la carpeta “res”, se almacenan todos los recursos del sistema. Dentro de estos recursos están nuestros archivos XML que usaremos para crear las vistas de nuestra aplicación.

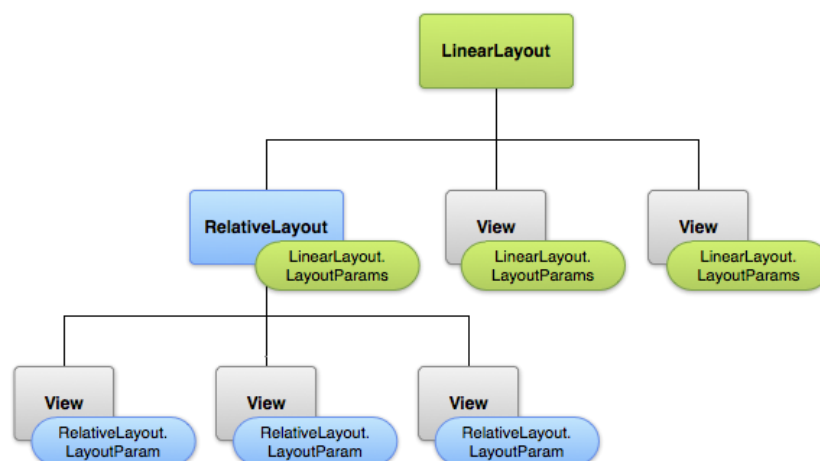


Fig 9: Herencia de objetos en la vista. Fuente: Android Developer (<http://developer.android.com/guide/topics/ui/declaring-layout.html>)

4.2.2. Actividades

Fuente: Estructura de una aplicación Android (<http://ganaclin.blogspot.com.es/2012/11/estructura-y-creacion-de-una-aplicacion.html>)

Una actividad o *Activity* es un componente de aplicación que proporciona una pantalla con la que los usuarios pueden interactuar con el fin de hacer algo, como marcar un teléfono, tomar una foto, enviar un correo electrónico, o ver un mapa. Cada actividad genera una ventana en la que se muestra su interfaz de usuario. La ventana normalmente llena la pantalla, pero puede ser menor que la pantalla y flotan en la parte superior de otras ventanas.

Una aplicación por lo general se compone de múltiples actividades que están libremente delimitadas unas de las otras. Por lo general, una actividad en una aplicación se especifica como la actividad "principal", que se presenta al usuario al iniciar la aplicación por primera vez. Cada actividad puede iniciar otra actividad con el fin de realizar diferentes acciones. Cada vez que se inicia una nueva actividad, la actividad anterior se detiene, pero el sistema conserva la actividad en una pila ("*back stack*"). Cuando se inicia una nueva actividad, se inserta en la pila de nuevo y se lleva a la atención al usuario (UI). El *back stack* tiene el funcionamiento básico de pila LIFO "*last in, first out*", por lo que, cuando el usuario presiona el botón Atrás en la actividad actual, se extrae de la pila la actividad y se destruye, y la actividad anterior se visualiza.

Cuando una actividad se detiene porque se inicia otra actividad, este cambio en el estado se notificará a través de métodos de devolución de llamada (*lifecycle callback*) de ciclo de vida de la actividad. Existen varios métodos de devolución de llamada que una actividad pueda recibir, debido a un cambio en su estado (este creándolo el sistema, deteniéndolo reanudándolo, o destruyéndolo) y cada devolución de llamada te ofrece la oportunidad de realizar un trabajo específico apropiado para el cambio de estado. Por ejemplo, si se detiene, su actividad lanzará objetos pesados, tales como conexiones de red o base de datos. Cuando la actividad se reanude, se puede volver a adquirir los recursos necesarios y reanudar las acciones que fueron interrumpidas. Estas transiciones de estado son parte del ciclo de vida de la actividad.

El resto de este documento analiza los conceptos básicos de cómo crear y utilizar una actividad, incluyendo una explicación completa de cómo funciona el ciclo de vida de la actividad, para que pueda gestionar adecuadamente la transición entre diferentes estados de actividad.

Para crear una actividad, debe crear una subclase de *Activity* (o una subclase existente de la misma). En la subclase, es necesario implementar métodos de devolución de llamada que el sistema llama cuando la actividad hace una transición entre los distintos estados de su ciclo de vida, como cuando se está creando la actividad, se detiene, se reanuda o se destruye. Los dos métodos de devolución de llamada más importantes son:

onCreate ()

Se debe implementar este método. El sistema llama a este método a la hora de crear la actividad. Dentro de la aplicación, se deben inicializar los componentes esenciales de la actividad. Lo más importante es dónde hacer la llamada al *setContentView()* para definir el layout del interfaz de usuario de la actividad.

onPause ()

El sistema llama a este método como la primera indicación de que el usuario abandona la actividad (aunque no siempre significa que la actividad está siendo destruida). Esto es por lo general donde se debe acometer cualquier cambio que debe persistir más allá de la sesión de usuario actual (ya que el usuario podría no volver).

Manifest

Se debe declarar toda actividad en el archivo *manifest* con el fin de que sea accesible para el sistema.

Hay muchas otras características que se pueden incluir en este documento, para definir propiedades tales como la etiqueta de la actividad, un icono para la actividad, o un estilo de UI de la actividad.

Lifecycle

La gestión del ciclo de vida (lifecycle) de sus actividades mediante la aplicación de métodos de devolución de llamada es crucial para el desarrollo de una aplicación fuerte y flexible. El ciclo de vida de una actividad se ve afectado directamente por su asociación con otras actividades, su trabajo y el de la pila.

La siguiente figura muestra los bucles y los caminos que una actividad podría tomar entre estados. Los rectángulos representan los métodos de devolución de llamada que se pueden implementar para llevar a cabo operaciones en las que las transiciones entre los estados de actividad.

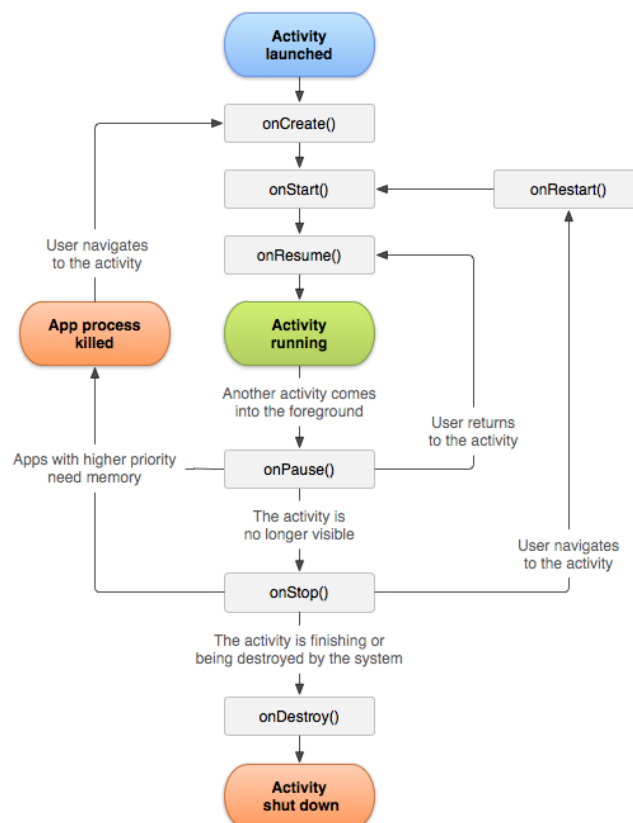


Fig. 10: Gestión de ciclo de vida de la interfaz.

Fuente: <http://developer.android.com/reference/android/app/Activity.html>

4.2.3. Uso de las herramientas gráficas

Eclipse, y en concreto el plugin ADT, proporciona al desarrollador ciertas herramientas que pueden ser utilizadas para crear la interfaz gráfica.

ADT ofrece muchas características que permiten diseñar y construir la interfaz de usuario de la aplicación. Muchas de estas características se encuentran en el editor de diseño gráfico, que se puede acceder mediante la apertura de uno de los archivos de formato XML de la aplicación en Eclipse.

El editor de diseño gráfico es la pantalla principal que se utiliza para diseñar visualmente y construir el interfaz de usuario. Se divide en las siguientes partes:

Canvas. Lienzo

En el medio del editor está el lienzo. Éste proporciona la vista renderizada de su diseño y soporta acciones como arrastrar y soltar UI *widgets* directamente desde la paleta. Se puede seleccionar la versión de la plataforma utilizada para representar los elementos dentro del lienzo. Cada versión de la plataforma tiene su propio aspecto e impresión, que podría ser similar o totalmente diferente a otra versión de la plataforma. El lienzo representa el aspecto y la impresión adecuados para la versión de la plataforma seleccionada. Esta versión de la plataforma no tiene por qué ser la misma que la versión objetivo de la aplicación.

El lienzo también ofrece acciones contextuales en la barra de acciones de diseño, tales como el ajuste de los márgenes y la orientación de diseño. La barra de diseño de acciones muestra las acciones disponibles en función del elemento de interfaz de usuario seleccionado en el lienzo.

Perfilado detalle. Outline

En el lado derecho del editor tenemos una vista en esquema. Aquí se muestra una vista jerárquica de su presentación en la que se pueden hacer cosas como cambiar el orden de los puntos de vista. La vista en esquema expone una funcionalidad similar a la lona, pero muestra la presentación de una lista ordenada en lugar de una vista previa renderizada.

Paleta. Palette

En el lado izquierdo encontramos la paleta. Proporciona un conjunto de *widgets* que se pueden arrastrar al lienzo. La paleta muestra pre-visualizaciones procesadas de los *widgets* para facilitar la búsqueda de los UI *widgets* que desee.

Selector de configuración

La parte superior del editor contiene el selector de configuración. Ofrece opciones para cambiar el modo de representación de un diseño o tipo de pantalla.

Además de estas herramientas gráficas orientadas al diseño de la interfaz de usuario, existen otras herramientas que nos permiten controlar en tiempo real la aplicación, y su funcionamiento. Entre otras:

- Traceview: Le permite perfilar la ejecución de su programa (Window > Open Perspective > Traceview).

- Android: Proporciona acceso al Android SDK Manager y Director de AVD. Otras características de Android, como la creación o actualización de los proyectos (aplicación y biblioteca) están integrados en el IDE de Eclipse.
- Visor de Jerarquía: Le permite visualizar jerarquía de vistas de la aplicación para encontrar ineficiencias (Window > Open Perspective > Hierarchy Viewer).
- Pixel Perfect: Le permite examinar de cerca la interfaz de usuario para ayudar en el diseño y construcción. (Window > Open Perspective > Pixel Perfect).
- DDMS: Proporciona funciones de depuración, incluyendo: captura de pantalla, el hilo y el montón de información y Logcat (Window > Open Perspective > DDMS).
- adb: Proporciona acceso a un dispositivo del sistema de desarrollo. Algunas características de adb se integran en ADT como la instalación del proyecto (menú de ejecución Eclipse), transferencia de archivos, enumeración de dispositivos y Logcat (DDMS).
- ProGuard: Permite la ofuscación la reducción y optimización de código. ADT integra ProGuard como parte de la construcción, si lo activa.

4.3. Estudio tecnologías PHP, JSON

PHP, en su versión 5 más concretamente, es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. PHP puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.

Dado que PHP Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite, y ya que PHP está orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos nos interesa mucho puesto que nuestro servidor es MySQL.

Además PHP es libre y abierto a todos.

JSON, acrónimo de *JavaScript Object Notation*, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

JSON se emplea habitualmente en entornos donde el tamaño del flujo de datos entre cliente y servidor es de vital importancia (de aquí su uso por Yahoo, Google, etc., que atienden a millones de usuarios) cuando la fuente de datos es explícitamente de fiar y donde no es importante el no disponer de procesamiento XSLT para manipular los datos en el cliente.

A partir de la versión, 5.2, PHP soporta paquetes JSON de manera nativa y eso nos permite crear objetos JSON legibles en Android. Estos objetos pueden ser creados mediante cierta consulta SQL a la base de datos concreta, y generando un array de tipo JSON en el propio programa PHP con dichos resultados de la consulta. Este array JSON será la respuesta del programa PHP en cuestión y será recibido mediante una llamada dentro de un *Activity* sin mayor problema.

4.4. Eclipse, Dispositivos, Versiones

Eclipse

Fuente: Wikipedia ([http://es.wikipedia.org/wiki/Eclipse_\(software\)](http://es.wikipedia.org/wiki/Eclipse_(software))) (13 de junio de 2013)

Eclipse es un programa informático compuesto por un conjunto de herramientas de programación de código abierto multiplataforma para desarrollar lo que el proyecto llama "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores.

Eclipse fue desarrollado originalmente por IBM como el sucesor de su familia de herramientas para VisualAge. Eclipse es ahora desarrollado por la Fundación Eclipse, una organización independiente sin ánimo de lucro que fomenta una comunidad de código abierto y un conjunto de productos complementarios, capacidades y servicios.

Junto con el plugin ADT de Android, esta plataforma permite diseñar, desarrollar y testear aplicaciones nativas en Android, gestionando todos sus periféricos y dispositivos hardware y dando soporte gráfico para el diseño e implementación del UI. También proporciona actualizaciones, debugging, puente con dispositivos reales y la posibilidad de ejecutar emuladores de las diferentes versiones de Android entre otras muchas posibilidades.

El principal lenguaje de programación que soporta eclipse es JAVA con un 93% de líneas de código generadas. También otros como ANSI C, C++, JSP, etc.

Dispositivos

Tal como se mencionaba anteriormente, el número de dispositivos con sistema operativo Android en el mercado actual es enorme, y su crecimiento es exponencial.

En este proyecto se intentara dar soporte a los principales dispositivos del mercado que muestro a continuación en un gráfico de ventas (Fig. 11):

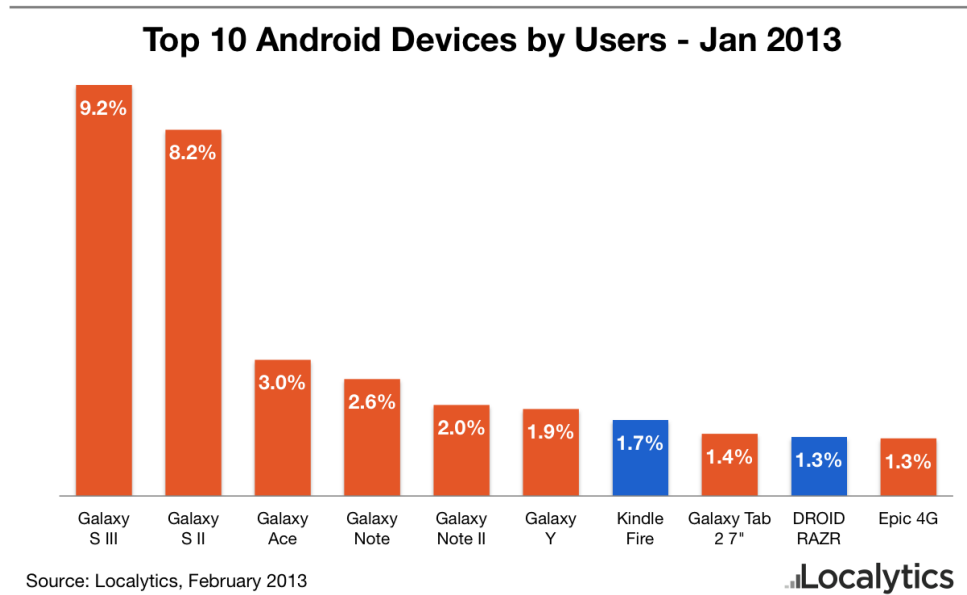


Fig 11: Gráfico de ventas de los dispositivos Android
Fuente: Localytics (11 de febrero de 2013)

El proyecto se centró en testear y validar la aplicación en los principales dispositivos del mercado. Y otros dispositivos tales como HTC, Kindle, Huawei, iPro, iBold, y otros.

Versiones

Anteriormente se ha mostrado el panorama actual de las versiones de Android en el mercado, existen diferentes versiones y diferentes usos. En concreto véase el siguiente gráfico que muestra el porcentaje de cada una de las versiones sobre el total de dispositivos con algún sistema operativo Android.

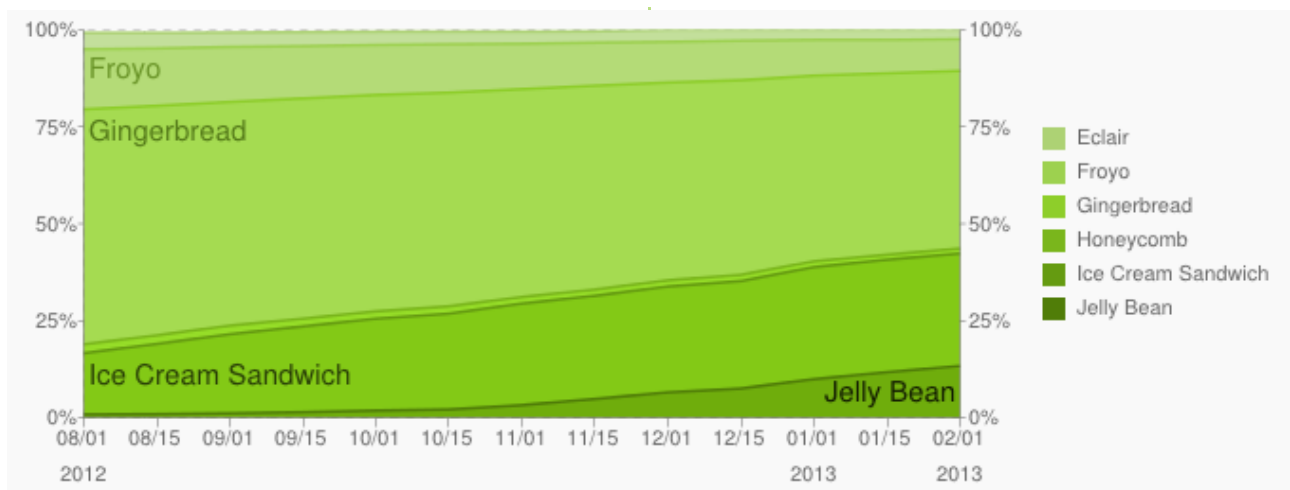


Fig 12: Gráfico del uso de las diferentes versiones de Android.
Fuente: Android (<http://developer.android.com/about/dashboards/index.html>)

Se ve claramente que a medida que disponemos de nuevas versiones de Android, estas se hacen más hueco en el mercado y encontramos un gran número de dispositivos que instalan estas versiones de sistema operativo. Pero también hay que considerar aquellos terminales que disponen de versiones más antiguas y que nuestra aplicación es perfectamente accesible a ellas.

En concreto se observa que las versiones Eclair y Froyo están quedando obsoletas y el nivel de uso actual es inferior al 10% del total de dispositivos Android. Por otro lado, la versión de Gingerbread, hace un año era muy numerosa pero también está perdiendo usuarios poco a poco como era de esperar. Aquí marqué mi frontera de desarrollo, ya que el número de sistemas operativos Android en dispositivos con las versiones Eclair y Froyo era tan baja, deseché de entre mis objetivos el desarrollar mi aplicación enfocada a estas versiones. Sin embargo Gingerbread, (2.3.x) fue una versión muy numerosa y actualmente existen muchos dispositivos que contemplan esta versión. Por lo que mi punto de partida se marcó en esta versión y posteriores. Abarcando así hasta un 94% del mapa de Android.

4.5. Firma de la aplicación, publicación, Google Play

Firma

El fin del desarrollo de la aplicación es publicar la creación en Google Play y hacer que varios usuarios empiecen a utilizarla. Para ello, uno de los requisitos que se debe cubrir es que la aplicación deberá estar firmada y a continuación se muestra cómo.

Motivos por los que se debe firmar la aplicación:

- Se debe firmar la aplicación como medida de seguridad y como requisito de garantía.
- Se debe firmar la aplicación para poder distribuir e instalar la aplicación sin problemas.
- Se debe firmar la aplicación para que de esta forma sólo el desarrollador pueda modificar y actualizar la aplicación.
- Se debe firmar la aplicación porque es un requisito que pide Google Play para subir una aplicación.

Hay ciertos pasos a seguir que vienen determinados por varios tutoriales.

Publicación, Google Play

Una vez registrado en Google Play, es decir, habiendo registrado tu nombre y pagado los 25\$ exigidos, nos permiten subir nuestras aplicaciones firmadas (archivo .apk) indicando ciertas especificaciones que hay que cumplimentar. El proceso de validación de la aplicación y final publicación puede durar entre un día y una semana dependiendo de la limpieza del código, correcto funcionamiento, etc.

Google Play (antes Android Market) es una tienda de software en línea desarrollada por Google para los dispositivos con sistema operativo Android. Es una aplicación que está preinstalada en la mayoría de los dispositivos Android y que permite a los usuarios buscar, obtener información y descargar aplicaciones publicadas por desarrolladores terceros.

La publicación final es un objetivo del proyecto pero no obstante, dado que la decisión final de publicación es por parte del responsable correspondiente del Festival, no se puede incluir como objetivo final indispensable ya que es una decisión en la que no toma parte el alumno, no obstante, se trata como objetivo, y el desarrollo se realiza con la mente puesta en la publicación en Google Play.

4.6. Restricciones y decisiones de plataforma. Herramientas

Anteriormente se han expuesto las posibles tecnologías a utilizar y múltiples razones por las cuales se han tomado ciertas decisiones.

Finalmente se exponen a continuación las herramientas que se han utilizado para el desarrollo del proyecto, las restricciones que generan y otras herramientas útiles.

- Plataforma de desarrollo Eclipse con el plugin ADT ya instalado: ADT Versión JUNO
- Lenguaje de programación JAVA SE 7.
Oracle
- Adobe Dreamweaver: Editor de texto web, edición de PHP y HTML.
ADOBE cs5
- Adobe Photoshop: Editor de imágenes e iconos.
ADOBE cs5
- Navicat premium: Desarrollo de consultas y prueba de los SQLquery en la base de datos del Festival
- CuteFTP: Transferencia de archivos, PHP y demás al servidor en cuestión.
- VideoPad: Editor de vídeos y conversor de formatos.
- La plataforma de desarrollo se centra en la versión 2.3.x de Android que corresponde con la versión API 9 Gingerbread y posteriores.
- Base de Datos MySQL, con acceso cifrado.

5. Funcionalidades aplicación

En el presente capítulo se describen detalladamente las funcionalidades especificadas en el alcance del proyecto y que contiene esta aplicación. Estas funcionalidades aportan cierta información que se debe obtener por medio de internet en comunicación con el servidor del Festival en el que se aloja toda la información requerida.

También se aborda la forma de diseño, de creación y de implementación de las funcionalidades. Una a una se ha ido desarrollando para luego integrar todas las Actividades y funcionalidades en una misma aplicación. Por ello, está dividido en diferentes paquetes por cada una de las funcionalidades principales.

5.1. Objetivo de la aplicación

El objetivo principal de la aplicación era proporcionar información inmediata a todos los usuarios que necesitasen información del Festival de Cine de San Sebastián. Como objetivo concreto se quería proporcionar información útil que un espectador, persona de la industria cinematográfica, o relacionada pudiera requerir llegado el momento. Concretamente debería poder conocer las noticias, imágenes y vídeos de última hora, premios, premiados y jurados, secciones y películas, poder disponer de un buscador de programación, entradas y enlace a redes sociales entre otros.

Era un objetivo importante que la aplicación pudiera ser visualizada en los tres idiomas oficiales del Festival de Cine (castellano, euskera e inglés).

Era un objetivo de la aplicación que nos mostrase toda la información de una manera visualmente atractiva, sobria y sencilla. Debía funcionar sin errores, con la agilidad y velocidad más alta que nos permitieran los accesos a Internet y al servidor y debía poseer un grado de calidad acorde al nivel de exigencia de un festival de cine como el Festival de Cine de San Sebastián.

5.2. Requisitos y navegación

En este apartado se habla de los requisitos que deben cumplir la aplicación y sus funcionalidades en cuanto a calidad, utilidad del servicio, eficacia, contenidos y agilidad.

La aplicación debe responder a las demandas de información del usuario en cualquier lugar del territorio, información móvil y dinámica. La aplicación debe contener un mínimo de calidad de información y agilidad para actualizarla rápidamente.

El servicio debe ser interactivo, intuitiva y de fácil aprendizaje, simple y eficaz. La información debe estar bien estructurada, grandes aportes de datos pero dispuestos con simpleza y mediante acceso rápido.

La aplicación debe tener un manejo sencillo para un desarrollador, su diseño debe estar enfocado a una rápida manipulación y personalización, y disponer de un estilo de desarrollo que pueda ser adaptado a otras plataformas móviles, es decir, que su migración a iOS o Windows Phone OS no sea demasiado costoso y complicado.

La aplicación debe aportar información multi-idioma y debe estar disponible en los principales idiomas del Festival.

Navegación en la aplicación

La aplicación comienza con un vídeo cortinilla del Festival editado por mí mismo. Esta cortinilla se podrá eliminar en próximos accesos a la aplicación para mayor comodidad del usuario. Una vez pasado el vídeo inicial, accedemos al menú inicio de la aplicación.

Dentro del menú inicial, encontramos una cabecera imagen con el logotipo de la próxima edición 61.

Debajo de esta cabecera, hay un slider de imágenes, que es un visor de las tres últimas noticias junto con el último álbum de imágenes añadido y el último vídeo. Se muestran las imágenes que acompañan a cada noticia y el título, así como la imagen principal del álbum y del vídeo y sus respectivos títulos.

Tras la imagen cabecera y el slider, hay unos botones que muestran las principales funcionalidades de la aplicación.

Estos botones son: Noticias, multimedia, premios y jurados, secciones y películas, programación y entradas.

También se mostrará una barra inferior con cuatro botones de acceso rápido.

Cada uno de los botones anteriores muestra una actividad diferente que detallo a continuación.

- **Noticias:** Esta funcionalidad, está diseñada de manera que abarque tres funcionalidades. Separados por selectores superiores encontramos las noticias del festival, otras noticias de interés, y redes sociales. Al seleccionar una de las sub-funcionalidades, mostrará una lista de noticias con el título, ante-título y foto. Al seleccionar una de ellas se mostrará a pantalla completa la noticia. El menú de redes sociales dispondrá de tres botones de acceso al navegador del dispositivo y concretamente a la página de la red social correspondiente.
- **Multimedia:** Al igual que la funcionalidad anterior, ésta agrupa la visualización de imágenes y vídeos del festival, cada uno de los elementos separados en dos pestañas. Dentro de cada funcionalidad, tendremos una lista de álbumes o vídeos. Haciendo clic en los primeros mostrará una nueva lista con todas las fotos del álbum correspondiente. Haciendo clic en el vídeo se abrirá el enlace YouTube correspondiente de cada vídeo.
- **Premios y jurados:** También dispone de un selector dentro de la funcionalidad para dividir entre premios del festival y premio Donostia. Este último separado del resto dada su repercusión e importancia. Cada funcionalidad mostrará listas de premios. Haciendo clic en ellos se muestra el premio en concreto, los jurados y toda la información relacionada.
- **Secciones y películas:** Se muestra una lista de las secciones principales del festival, así como las películas que componen dichas secciones. Al hacer clic en una de las secciones se muestra la información relacionada.

- **Programación:** Esta funcionalidad, muestra cuatro desplegables con la información seleccionable respecto a lugar de proyección, fecha, película y sección cinematográfica.
Al seleccionar los campos deseados, y pulsar el botón de búsqueda, se muestran los pases de películas correspondientes en una lista.
- **Entradas:** Muestra el enlace web al servicio de venta de entradas del Festival.
- **Barra inferior:** Dispone de cuatro botones:
 - **Inicio:** Al pulsar este botón accedemos directamente al menú principal.
 - **Mapa:** Muestra el mapa de localización del festival, hoteles, cines, lugares de interés.
 - **Contacto:** Muestra los principales contactos del festival, teléfonos, e-mails, etc.
 - **Más:** Muestra una lista para funcionalidades adicionales:
 - **Configuración:** Muestra la configuración disponible de la aplicación, selección de idioma y eliminación del vídeo inicial.
 - **About:** Muestra los créditos e información relevante sobre la aplicación.

5.3. Estructuración. Arquitectura de la aplicación

La aplicación del festival de cine es una aplicación centralizada cuya estructura es de tres niveles, tipo cliente-servidor.

En el primer nivel de la arquitectura encontramos el nivel de usuario o nivel aplicativo al usuario que engloba la interfaz gráfica de la aplicación, y se comunica visualmente con el usuario. Este nivel, abarca los layouts de la aplicación y en resumen es la capa que ve el usuario. Con los diseños, botones, características visuales y otros elementos gráficos se compone la interfaz gráfica de usuario o UI de la aplicación del Festival de cine.

Bajo esta capa o nivel, tenemos la lógica de negocio de la aplicación. Este nivel engloba y administra las actividades (activity) y procesos de la aplicación. Es aquí donde la aplicación gestiona sus recursos, hace una obtención de los paquetes de datos, hace un tratamiento de los mismos y los adapta al nivel de usuario. En definitiva es el nivel en el que se implementa el funcionamiento de la aplicación y se gestiona la aplicación. Este nivel está oculto al usuario.

Por ultimo encontramos el nivel de base de datos o de administración de los datos. Este nivel gestiona la conexión con la base de datos, genera paquetes de información con un determinado formato y los ofrece al nivel de lógica de negocio. Aquí se trata la seguridad de la propia base de datos y en definitiva, de la aplicación.

5.4. Diseño e implementación

El diseño de la aplicación se ha realizado acorde a los estándares requeridos por el propio Festival, colores, diseño de botones, estructura de la aplicación,
El diseño general tiene un estilo sobrio, simple y poco recargado. Los colores son adecuados al tema, al igual que los iconos y los seleccionables.

5.4.1. Menús

Los menús de la aplicación son acordes a las características establecidas en el diseño y requisitos por parte del Festival de cine de San Sebastián.

El menú principal es sobre fondo negro y con los colores requeridos. Se ha incluido una barra inferior en los menús principales de la aplicación y de cada funcionalidad para aportar un acceso rápido e intuitivo al menú principal.

También da acceso al resto de funcionalidades incluidas en esta barra de selección, que se consideran de acceso habitual y necesario. En la Fig. 13 se observa el diseño del menú principal:



Fig. 13: Menú principal

5.4.2. Grafismo

Letras: La letra oficial del Festival es *Avant Garde* pero se toma la tipografía propia de cada dispositivo a elección del propio usuario dentro del menú Ajustes de Android.

En cuanto a la composición gráfica, viene dada por los requisitos del Festival de cine en relación a estilos y colores.

5.4.3. Características de arranque, logotipo.

Tras instalar la aplicación en el terminal o dispositivo, aparecerá el icono oficial del Festival, este icono viene dado por los requisitos interpuestos por el Festival y tras pulsarlo, accederemos directamente a la aplicación, vídeo inicial y después menú inicial. A no ser que hayamos eliminado el vídeo inicial dado lo cual accederemos directamente al menú principal.



Fig. 14: Logotipo de la aplicación

5.4.4. Diseño e implementación de las funcionalidades

A continuación se muestra un listado de las funcionalidades de cada aplicación con su diseño e implementación necesarios para llevarlas a cabo.

5.4.4.1. Noticias y redes sociales

Esta funcionalidad muestra las noticias extraídas del servidor propio del Festival de cine. Para la obtención de las noticias disponemos de un servicio web implementado en PHP que se muestra en el capítulo 6.

La actividad que implementa esta funcionalidad es la siguiente:

```
public class MainActivity extends Activity {
    int ini=0;
    String dir;

    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        String val=Locale.getDefault().getLanguage();
        // Seleccionamos el servicio web correspondiente al idioma
        if(val.contentEquals("es")){
            dir=("http://www.sansebastianfestival.com/android/contenedor_es.php?ap=1&inicio=");
        }
        else if (val.contentEquals("eu"))
        {
            dir=("http://www.sansebastianfestival.com/android/contenedor_eu.php?ap=1&inicio=");
        }
        else {
            dir=("http://www.sansebastianfestival.com/android/contenedor_in.php?ap=1&inicio=");
        }
        ListView lvPdonostia = (ListView) findViewById(R.id.lv_noticias);

        ArrayList<Noticia> Notic = new ArrayList<Noticia>();

        try {
            // Llamamos al servicio web para recuperar los datos
            HttpClient httpClient = new DefaultHttpClient();
           HttpGet httpGet = new HttpGet(dir+"0");
            HttpResponse response = (HttpResponse)httpClient.execute(httpGet);
            HttpEntity entity = response.getEntity();
            BufferedHttpEntity buffer = new BufferedHttpEntity(entity);
            InputStream iStream = buffer.getContent();

            String aux = "";

            BufferedReader r = new BufferedReader(new InputStreamReader(iStream));
            String line;
            while ((line = r.readLine()) != null) {
                aux += line;
            }

            // Parseamos la respuesta obtenida del servidor a un objeto JSON
            JSONObject jsonObject = new JSONObject(aux);
            JSONArray Noticias= jsonObject.getJSONArray("contenedor");
        }
    }
}
```

```

// Recorremos el array con los elementos noticia
for(int i = 0; i < Noticias.length(); i++) {
    JSONObject noticia = Noticias.getJSONObject(i);

    // Creamos el objeto Noticia
    Noticia c = new Noticia(noticia.getInt("id"), noticia.getString("titulo"));
    c.setExtracto(noticia.getString("extracto"));
    c.setPhoto(noticia.getString("photo"));

    // Almacenamos el objeto en el array que hemos creado anteriormente
    Notic.add(c);
}
}
catch(Exception e) {
    Log.e("WebService", e.getMessage());
}

// Creamos el objeto NoticiaAdapter y lo asignamos al ListView
TextView vista = (TextView) findViewById(R.id.notiVistas);
vista.setText(" 0.5 ");
final NoticiaAdapter noticiaAdapter = new NoticiaAdapter(this, Notic);
lvPdonostia.setAdapter(noticiaAdapter);
lvPdonostia.setOnItemClickListener(new OnItemClickListener() {
    public void onItemClick(AdapterView<?> parent, View v, int posicion, long id) {
        Intent textIntent = new Intent(MainActivity.this, Noticiatexto.class);
        long pos= noticiaAdapter.getItemId(posicion) ;
        int i= (int)(long)pos;
// Pasamos el identificador de la noticia seleccionada
        textIntent.putExtra("pos", i);
        startActivity(textIntent);
    }
});
}
});

```

De esta manera creamos un *ListView* con los títulos, ante-títulos, e imágenes de cada noticia obtenidos con servicio web que está dentro del propio servidor del Festival de cine, llamado: *contenedor_es.php?ap=1&inicio=0*. Mediante el *adapter*, se muestran correctamente en el dispositivo.

Las noticias están paginadas en páginas de cinco noticias para reducir el tráfico de datos y mejorar la interacción con el usuario. Dos botones permiten navegar adelante y atrás y se muestra el rango de noticias visualizado.

Una vez seleccionada una noticia, pasamos a un nuevo servicio web en PHP, el identificador de la noticia correspondiente de manera que llamamos al PHP *texto.php*. Este nos devuelve los datos de la noticia y se muestran en el dispositivo.

Dependiendo del idioma seleccionamos uno u otro servicio PHP aunque todos son iguales diferenciando los datos que traen en cada idioma.

El resultado final dispone de un *Tabhost* con tres pestañas. La primera visualiza las noticias propias del Festival y la segunda de la misma manera carga noticias relacionadas con el mundo del cine.

Por otro lado, la tercera pestaña muestra un menú con tres botones y sus correspondientes iconos que permiten acceder a cada una de las tres redes sociales que dispone el Festival; Facebook, Twitter y Flickr. Estos accesos son *intents* que abren el browser del dispositivo en cada página web específica de cada red social.

El diseño final queda de la siguiente manera:

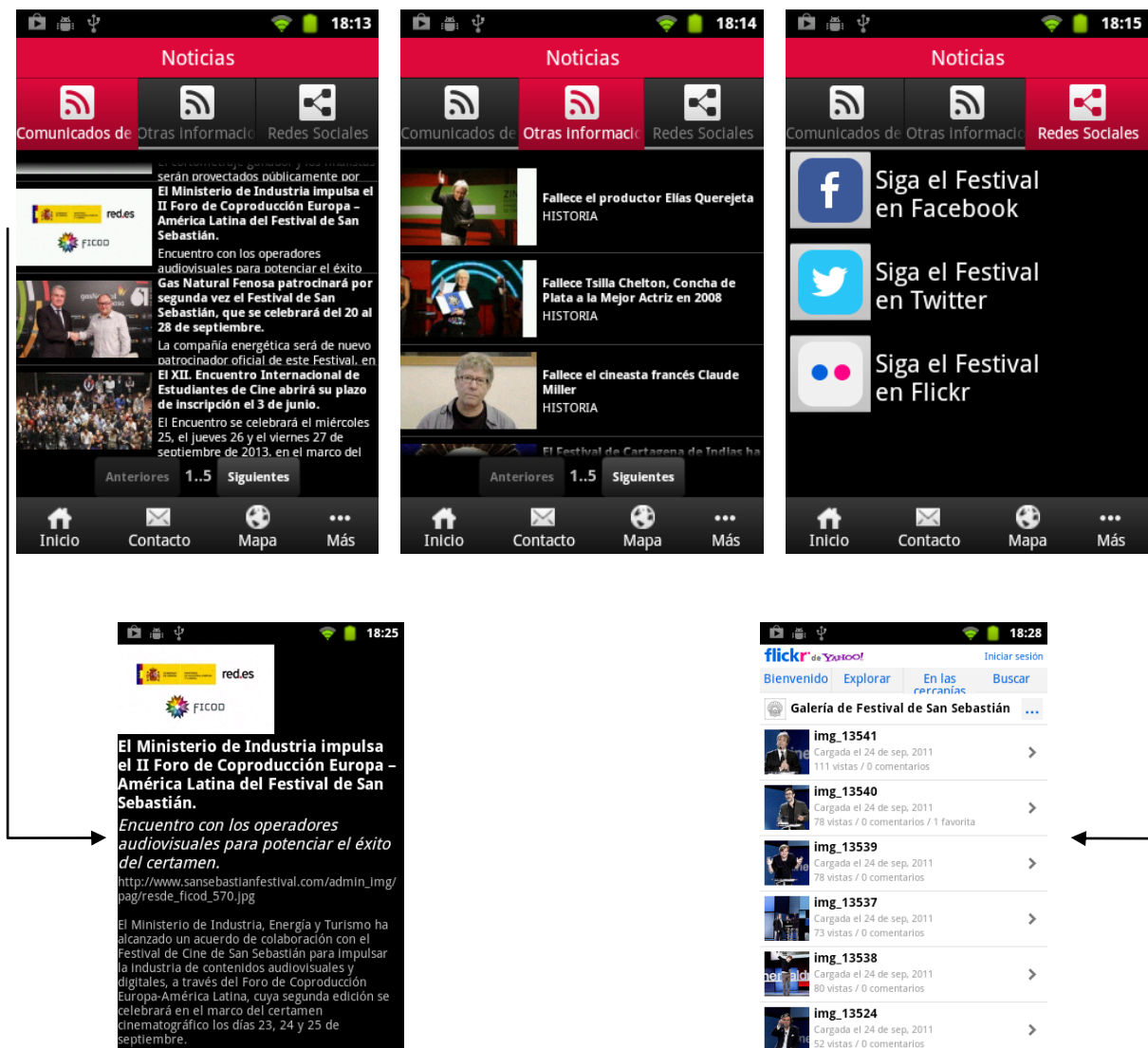


Fig. 15: Pantallas de navegación en Noticias

5.4.4.2. Galería de fotos y vídeos

Este servicio, muestra las imágenes y vídeos del Festival. Para ello, se muestran en dos listados para cada tipo de objetos.

El *Activity* que lo compone es similar al anterior con ciertas variaciones:

```
public class MainActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main_galeria);

        ListView lvGaleria = (ListView) findViewById(R.id.lv_galeria);
        ArrayList<Galeria> Galer = new ArrayList<Galeria>();

        try {
            // Llamamos al servicio web para recuperar los datos
            HttpGet httpGet = new HttpGet("http://www.sansebastianfestival.com/android/galerias.php");
            HttpClient httpClient = new DefaultHttpClient();
            HttpResponse response = (HttpResponse)httpClient.execute(httpGet);
            HttpEntity entity = response.getEntity();
            BufferedHttpEntity buffer = new BufferedHttpEntity(entity);
            InputStream iStream = buffer.getContent();
        }
    }
}
```

```

String aux = "";

BufferedReader r = new BufferedReader(new InputStreamReader(iStream));
//
StringBuilder total = new StringBuilder();
String line;
while ((line = r.readLine()) != null) {
    aux += line;
}

// Parseamos la respuesta obtenida del servidor a un objeto JSON
JSONObject jsonObject = new JSONObject(aux);
JSONArray Agalerias = jsonObject.getJSONArray("galerias");

// Recorremos el array con los elementos galeria
for(int i = 0; i < Agalerias.length(); i++) {
    JSONObject galery = Agalerias.getJSONObject(i);

    // Creamos el objeto Galeria
    Galeria c = new Galeria(galery.getInt("id"), galery.getString("titulo_es"));
    c.setTituloEu(galery.getString("titulo_eu"));
    c.setTituloIn(galery.getString("titulo_in"));
    c.setPhoto(galery.getString("photo"));
    c.setNfotos(galery.getInt("nfotos"));
    c.setGaleria(galery.getString("galeria"));

    // Almacenamos el objeto en el array que hemos creado anteriormente
    Galer.add(c);
}
}
catch(Exception e) {
    Log.e("WebService", e.getMessage());
}

// Creamos el objeto GaleriaAdapter y lo asignamos al ListView
final GaleriaAdapter galeriaAdapter = new GaleriaAdapter(this, Galer);
lvGaleria.setAdapter(galeriaAdapter);
lvGaleria.setOnItemClickListener(new OnItemClickListener() {
    public void onItemClick(AdapterView<?> parent, View v, int posicion, long id) {
        Intent textIntent = new Intent(MainActivity.this, Galeriatexto.class);
        int pos= galeriaAdapter.getItem(posicion).getId();
        textIntent.putExtra("pos", pos);
        startActivity(textIntent);
    }
});
}
}
}

```

Al igual que en la actividad anterior, llamamos a un servicio web en el servidor específicamente diseñado para traer las fotos o los vídeo del festival. El *Activity* mostrado es para las imágenes y mostraría un *listview* de los álbumes del Festival.

La estructura de esta funcionalidad quedaría dividida en dos *tabs* o pestañas de un *Tabhost* para cada tipo multimedia, imágenes por un lado y vídeos por el otro.

Una vez clicado en uno de los álbumes, traemos nueva información sobre todas las fotos de cada álbum y se muestra un listado de ellas. Pulsando en ellas traemos la imagen del servidor con mayor resolución y se amplía.

En el caso de los vídeos se muestra un listado de vídeos acompañados con su foto índice al igual que en las imágenes. Pulsando en uno de los vídeos se muestra el vídeo en la página de YouTube del *browser* del dispositivo o directamente se abre la aplicación de YouTube con el vídeo.

El diseño quedaría de la siguiente manera:

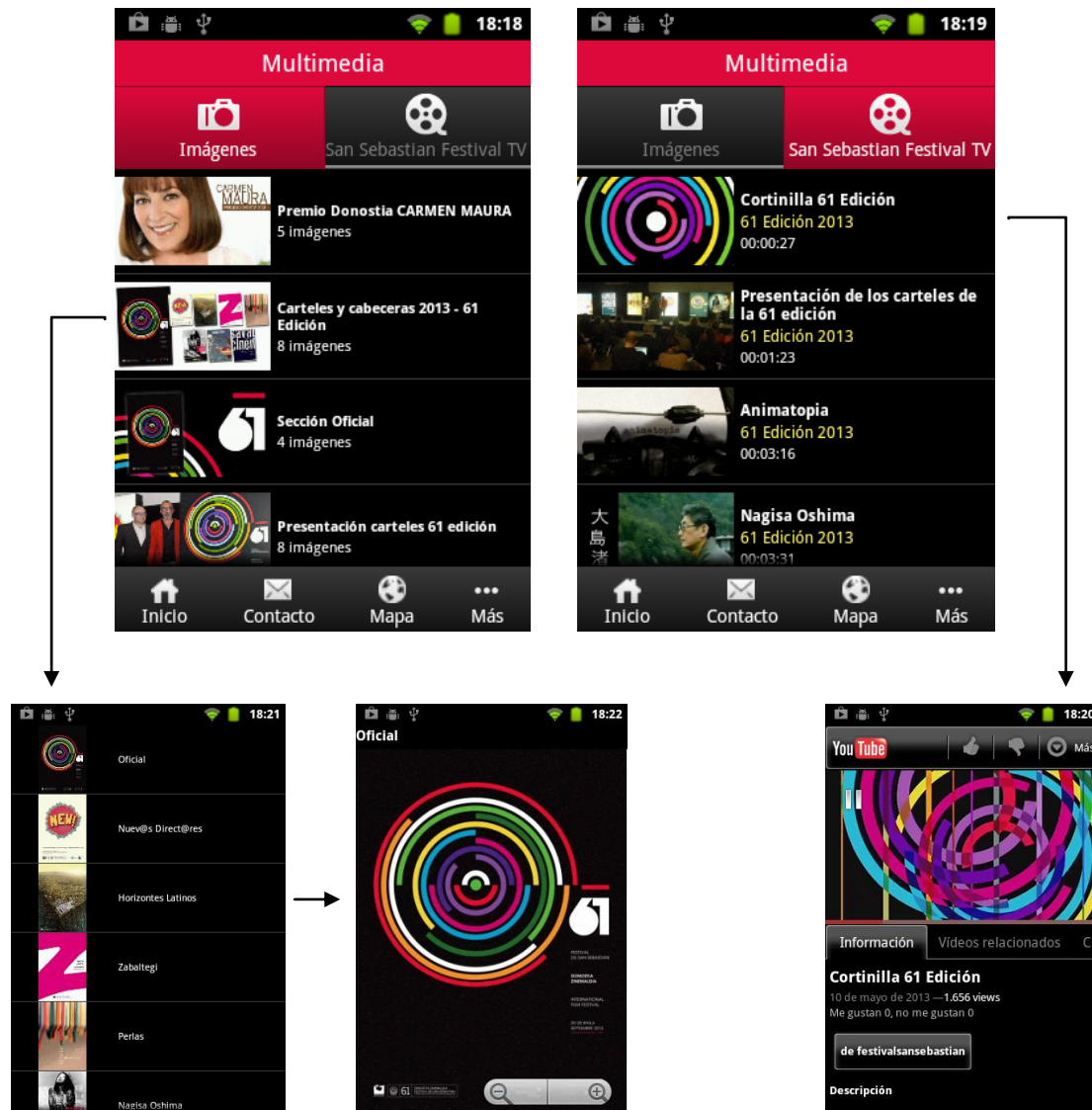


Fig. 16: Pantallas de navegación en Multimedia

5.4.4.3. Programación

La funcionalidad que tratamos a continuación es un buscador de la programación. Este buscador, es un discriminador de consultas, es decir, el usuario introducirá ciertos datos discriminantes por tema (sección, lugar, día, película) y se hace una consulta al servidor con las condiciones dadas. Después se muestra en pantalla un listado de todos los pases de películas que están dentro de las condiciones marcadas por el usuario. Este listado contiene información como el título, duración, lugar, sección y otros datos relevantes.

Los condicionantes, o los valores de discriminación dentro de los desplegables que se muestran inicialmente también son información obtenida desde el servidor y actualizable de manera automática:

El *Activity* siguiente muestra la forma de pasar los datos de búsqueda:

```
public class ProgramacionActivity extends Activity {
    TextView lbopcion, lbopcion1, lbopcion2;
    Spinner spOpciones, spOpciones1, spOpciones2, spOpciones3;
    String[] datosSeccionES, datosSeccionEU, datosSeccionIN, datosLugar, datosFecha, datosPelis;
    int[] intLugar, intSeccion, intPelis;
    int se, lu, d, pe;
}
```

```

    ArrayAdapter<String> adaptador, adaptador1, adaptador2, adaptador3;
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.programacion);
        String val=Locale.getDefault().getLanguage();
        ;

        try {
            // Llamamos al servicio web para recuperar los datos
            HttpGet httpGet = new
                HttpGet("http://www.sansebastianfestival.com/android/desplegables.php");
            HttpClient httpClient = new DefaultHttpClient();
            HttpResponse response = (HttpResponse)httpClient.execute(httpGet);
            HttpEntity entity = response.getEntity();
            BufferedHttpEntity buffer = new BufferedHttpEntity(entity);
            InputStream iStream = buffer.getContent();

            String aux = "";

            BufferedReader r = new BufferedReader(new InputStreamReader(iStream));
            String line;
            while ((line = r.readLine()) != null) {
                aux += line;
            }

            // Parseamos la respuesta obtenida del servidor a un objeto JSON
            JSONObject jsonObject = new JSONObject(aux);
            JSONArray Alugares= jsonObject.getJSONArray("lugares");

            JSONObject jsonObject1 = new JSONObject(aux);
            JSONArray Asecciones= jsonObject1.getJSONArray("secciones");

            JSONObject jsonObject2 = new JSONObject(aux);
            JSONArray Adias= jsonObject2.getJSONArray("dias");

            JSONObject jsonObject3 = new JSONObject(aux);
            JSONArray Apelis= jsonObject3.getJSONArray("peliculas");

            // Recorremos el array con los elementos correspondientes

            datosLugar=new String[Alugares.length()+1];
            intLugar = new int[Alugares.length()+1];
            intLugar[0]=0;
            if(val.contentEquals("es")){
                datosLugar[0]="Lugar";
            }
            else if(val.contentEquals("eu")){
                datosLugar[0]="Gunea";
            }
            else{
                datosLugar[0]="Venue";
            }
            for(int i = 1; i < Alugares.length()+1; i++) {
                JSONObject lug = Alugares.getJSONObject(i-1);

                Lugares c = new Lugares(lug.getInt("sala_id"), lug.getString("sala_cine"));
                datosLugar[i]=c.getCine();
                intLugar[i]=c.getId();
            }

            datosSeccionES=new String[Asecciones.length()+1];
            datosSeccionEU=new String[Asecciones.length()+1];
            datosSeccionIN=new String[Asecciones.length()+1];
            intSeccion = new int[Asecciones.length()+1];
            intSeccion[0]=0;
            if(val.contentEquals("es")){
                datosSeccionES[0]="Sección";
            }
            else if(val.contentEquals("eu")){
                datosSeccionEU[0]="Saila";
            }
            else{
                datosSeccionIN[0]="Section";
            }
            for(int i = 1; i < Asecciones.length()+1; i++) {

```

```

        JSONObject sec = Asecciones.getJSONObject(i-1);
        Secciones c = new Secciones(sec.getInt("cicloKode"), sec.getString("sec_es"),
        sec.getString("sec_eu"), sec.getString("sec_in"));
        datosSeccionES[i]=c.getSeccionEs();
        datosSeccionEU[i]=c.getSeccionEu();
        datosSeccionIN[i]=c.getSeccionIn();
        intSeccion[i]=c.getId();
    }
    datosFecha = new String[Adias.length()+1];

    if(val.contentEquals("es")){
        datosFecha[0]="Fecha";
    }
    else if(val.contentEquals("eu")){
        datosFecha[0]="Eguna";
    }
    else{
        datosFecha[0]="Date";
    }

    for(int i = 1; i < Adias.length()+1; i++) {
        JSONObject di = Adias.getJSONObject(i-1);

        Dias c = new Dias(di.getInt("pase_dia"));
        datosFecha[i]= Integer.toString(c.getId());
    }
    datosPelis=new String[Apelis.length()+1];
    intPelis = new int[Apelis.length()+1];
    intPelis[0]=0;
    if(val.contentEquals("es")){
        datosPelis[0]="Peliculas";
    }
    else if(val.contentEquals("eu")){
        datosPelis[0]="Filmak";
    }
    else{
        datosPelis[0]="Films";
    }

    for(int i = 1; i < Apelis.length()+1; i++) {
        JSONObject pel = Apelis.getJSONObject(i-1);

        Pelis c = new Pelis(pel.getInt("pase_codigo"), pel.getString("pase_titulo"));
        datosPelis[i]=c.getTitulo();
        intPelis[i]=c.getId();
    }
}
catch(Exception e) {
    Log.e("WebService", e.getMessage());
}

if(val.contentEquals("es")){
    adaptador=new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item,
    datosSeccionES);
}
else if(val.contentEquals("eu")){
    adaptador=new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item,
    datosSeccionEU);
}
else{
    adaptador=new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item,
    datosSeccionIN);
}

adaptador1=new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item, datosLugar);
adaptador2=new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item, datosFecha);
adaptador3=new ArrayAdapter<String>(this, android.R.layout.simple_spinner_item, datosPelis);

spOpciones=(Spinner) findViewById(R.id.spOpciones);
spOpciones1=(Spinner) findViewById(R.id.spOpciones1);
spOpciones2=(Spinner) findViewById(R.id.spOpciones2);
spOpciones3=(Spinner) findViewById(R.id.spOpciones3);

adaptador.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spOpciones.setAdapter(adaptador);
adaptador1.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);

```



```

spOpciones1.setAdapter(adaptador1);
adaptador2.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spOpciones2.setAdapter(adaptador2);
adaptador3.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
spOpciones3.setAdapter(adaptador3);

spOpciones.setOnItemSelectedListener(new OnItemSelectedListener() {
    public void onItemSelected(AdapterView<?> parent, View v,int position, long id) {
        se= intSeccion[position];
    }
    public void onNothingSelected(AdapterView<?> arg0) {
        se=0;
    }
});
spOpciones1.setOnItemSelectedListener(new OnItemSelectedListener() {
    public void onItemSelected(AdapterView<?> parent, View v,int position, long id) {
        lu=intLugar[position];
    }
    public void onNothingSelected(AdapterView<?> arg0) {
        lu=0;
    }
});
spOpciones2.setOnItemSelectedListener(new OnItemSelectedListener() {
    public void onItemSelected(AdapterView<?> parent, View v,int position, long id) {
        if((datosFecha[position]).contentEquals("Fecha")||
        (datosFecha[position]).contentEquals("Eguna")||
        (datosFecha[position]).contentEquals("Date")){
            d=0;
        }
        else{
            d=Integer.parseInt(datosFecha[position]);
        }
    }
    public void onNothingSelected(AdapterView<?> arg0) {
        d=0;
    }
});
spOpciones3.setOnItemSelectedListener(new OnItemSelectedListener() {
    public void onItemSelected(AdapterView<?> parent, View v,int position, long id) {
        pe=intPelis[position];
    }
    public void onNothingSelected(AdapterView<?> arg0) {
        pe=0;
    }
});

Button startBuscar= (Button) findViewById(R.id.button1);
startBuscar.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        Intent progIntent = new Intent(ProgramacionActivity.this, ProgramacionDisplay.class);
        progIntent.putExtra("se", se);
        progIntent.putExtra("lu", lu);
        progIntent.putExtra("d", d);
        progIntent.putExtra("pe", pe);
        startActivity(progIntent);
    }
});
}
}
}

```

Una vez obtenidos los datos de los desplegables se muestran de la siguiente forma:

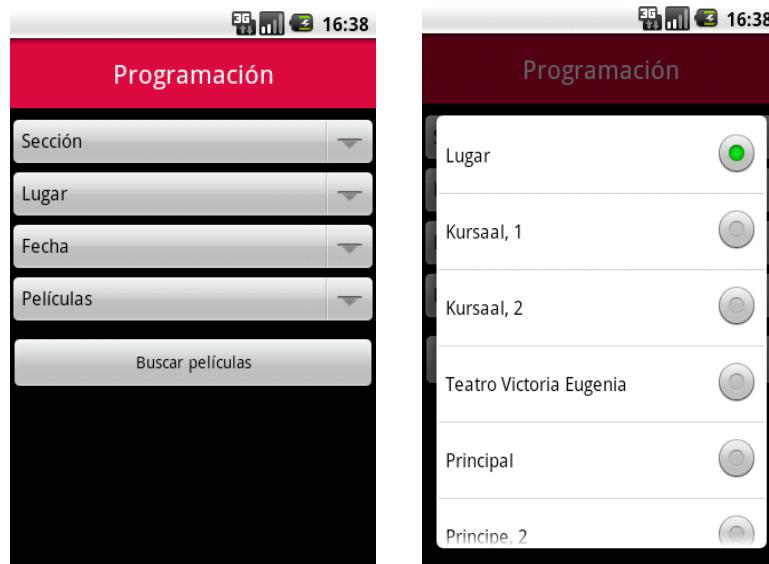


Fig. 17: Pantallas de navegación en Programación

Después de que el usuario introduzca las opciones de búsqueda necesarias y pulsando el botón de búsqueda, se pasan los datos al servidor y se ejecuta un nuevo servicio web con los datos de la consulta necesarios, de manera que se ha diseñado un *Activity* como el siguiente:

```
public class ProgramacionDisplay extends Activity{
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.programacion_disp);
        Bundle bundle = getIntent().getExtras();
        final int se= bundle.getInt("se");
        final int lu= bundle.getInt("lu");
        final int d= bundle.getInt("d");
        final int pe= bundle.getInt("pe");
        ListView lvProg = (ListView) findViewById(R.id.Lv_Prog);
        ArrayList<Programacion> Prog = new ArrayList<Programacion>();
    try {
        // Llamamos al servicio web para recuperar los datos
        HttpGet httpGet = new HttpGet("http://www.sansebastianfestival.com/android/programacion.php?
            se="+Integer.toString(se)+"&lu="+Integer.toString(lu)+"&d="+Integer.toString(d)
            +"&pe="+Integer.toString(pe));
        // Le pasamos los datos de cada campo seleccionado
        HttpClient httpClient = new DefaultHttpClient();
        HttpResponse response = (HttpResponse)httpClient.execute(httpGet);
        HttpEntity entity = response.getEntity();
        BufferedHttpEntity buffer = new BufferedHttpEntity(entity);
        InputStream iStream = buffer.getContent();

        String aux = "";
        BufferedReader r = new BufferedReader(new InputStreamReader(iStream));
        String line;
        while ((line = r.readLine()) != null) {
            aux += line;
        }
        // Parseamos la respuesta obtenida del servidor a un objeto JSON
        JSONObject jsonObject = new JSONObject(aux);
        JSONArray Aprog = jsonObject.getJSONArray("programacion");
        // Recorremos el array con los elementos galeria
        for(int i = 0; i < Aprog.length(); i++) {
            JSONObject prog = Aprog.getJSONObject(i);
            // Creamos el objeto Galeria
            Programacion c = new Programacion(prog.getInt("pase_id"), prog.getString("pase_dia"),
            prog.getString("pase_hora"),prog.getString("pase_titulo"),prog.getString("pase_subtitulos_es"),
            prog.getString("pase_subtitulos_eu"),prog.getString("pase_subtitulos_in"),prog.getString("duracion"),
            prog.getString("ticketing"),prog.getString("sec_es"),prog.getString("sec_eu"),
            prog.getString("sec_in"), prog.getString("sala_cine"));
            // Almacenamos el objeto en el array que hemos creado anteriormente
        }
    }
}
```

```

        Prog.add(c);
    }
}
catch(Exception e) {
    Log.e("WebService", e.getMessage());
}
// Creamos el objeto GaleriaAdapter y lo asignamos al ListView
final ProgramacionAdapter programacionAdapter = new ProgramacionAdapter(this, Prog);
lvProg.setAdapter(programacionAdapter);
}
}

```

El resultado de la búsqueda sería una lista con *scroll* como la siguiente:



Fig. 18: Pantalla de selección y visualización de la programación

5.4.4.4. Secciones

Las secciones cinematográficas del Festival de cine de San Sebastián se muestran en esta funcionalidad. Al igual que las anteriores, la información de estas secciones se obtiene del servidor y se muestra en una lista con las diferentes secciones disponibles junto con una imagen de la sección. Al pulsar en una de las secciones llamamos a un nuevo servicio web PHP que trae la información relativa a la sección en cuestión.

Ejemplos del diseño final de la funcionalidad:



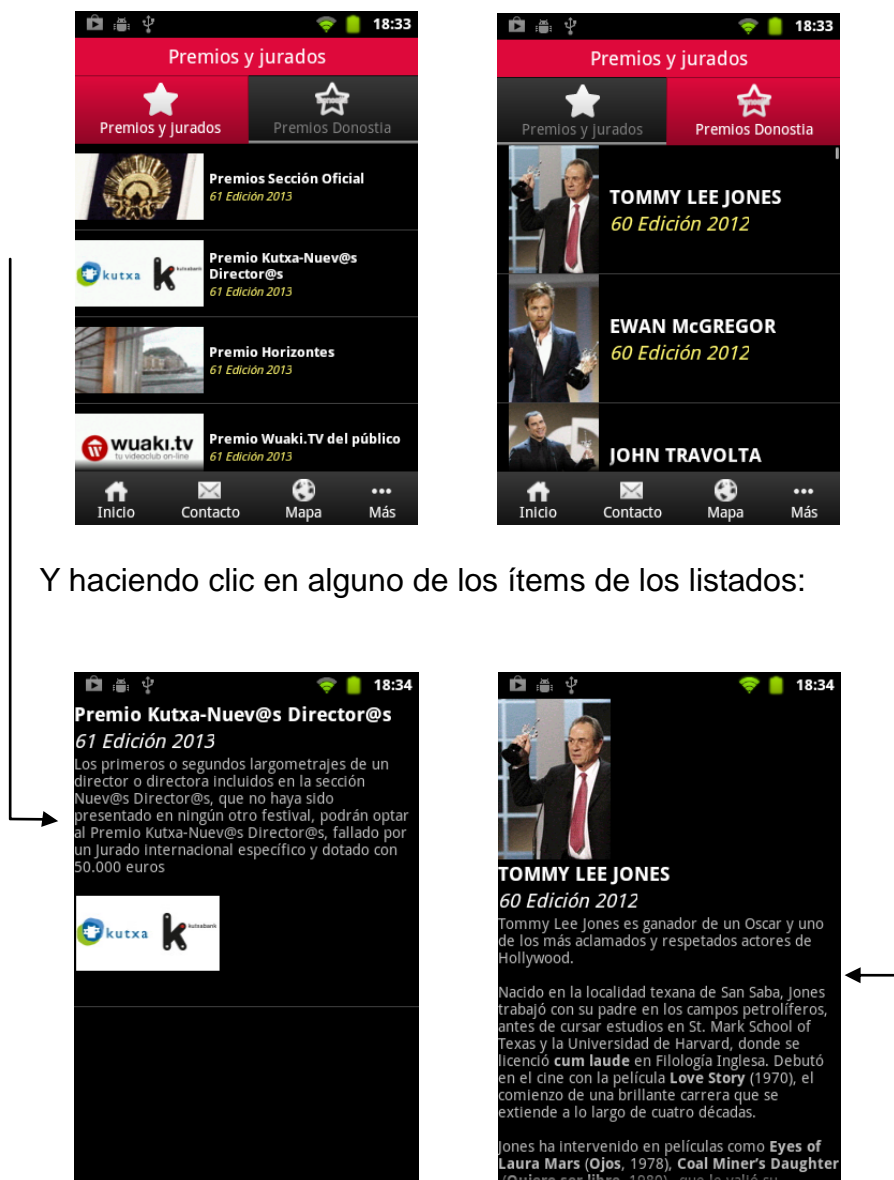
Fig. 19: Diseño de secciones y películas

5.4.4.5. Premios y premiados

Esta funcionalidad, informa sobre los premios y premiados del Festival de cine de San Sebastián y es similar a las anteriores. Hacemos una llamada a un servicio web concreto que trae la información sobre los premios. En esta ocasión, se divide en dos pestañas dentro de un *Tabhost*. La primera pestaña corresponde a los premios generales del Festival y la segunda a los premios Donostia.

En esta funcionalidad se muestra un listado de los premios del Festival de cine de San Sebastián y al hacer clic en alguno de ellos se muestra información más detallada de cada elemento, sean premiados con el Premio Donostia o premios y jurados en general.

El diseño que se obtiene es el siguiente:



Y haciendo clic en alguno de los ítems de los listados:

Fig. 20: Pantallas de navegación en Premios y jurados

5.4.4.6. Entradas

La funcionalidad que lleva al usuario a las entradas es un *Intent* que lanza el browser del dispositivo con la dirección web correspondiente a las entradas.

El diseño es simple y solo se muestra el apartado de entradas propio de la página web del festival para dispositivos móviles.



Fig. 21: Pantalla de entradas

5.4.4.7. Mapa

La funcionalidad del mapa viene condicionada por varios aspectos: En primer lugar se diseñó un mapa con la API de Google Maps v2 para que la ejecución del mapa fuera de forma nativa en la aplicación, sin embargo al disponer de un mapa propio del Festival alojado en la web y con los iconos y señalizaciones concretas del Festival, se decidió apartar el mapa previamente diseñado y hacerlo más simple mediante un *WebView* en el que se muestra la dirección en la que está alojado el mapa que hay en la propia página web del Festival de cine mostrando todos los iconos y el diseño establecido.

Además y en segundo lugar, el mapa ya diseñado en la página web no se podía trasladar al API de google maps y eso restaba calidad y complicaba mucho la funcionalidad, por lo que finalmente se decidió tomar la segunda opción y dejar la primera como posibles mejoras en un futuro.

Una captura de pantalla muestra cómo queda la funcionalidad:

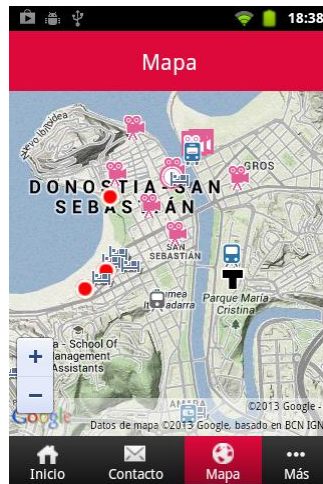


Fig. 22: Pantalla del Mapa

5.4.4.8. Configuración

Esta funcionalidad surge a raíz de dos problemas de la aplicación:

En un primer momento se decidió hacer la aplicación en los tres idiomas del Festival de cine de San Sebastián (Inglés, castellano y euskera) pero como bien es sabido y lamentablemente, no existen sistemas operativos Android con el idioma oficial en euskera por tanto se debía hacer la elección por software en vez de utilizar la detección automática.

Para ello, dentro del menú de configuración disponemos de botones para la elección del idioma junto con unas banderas que aportan un grado más de accesibilidad.

Por otro lado, a modo de presentación, se muestra un vídeo cortinilla del festival nada más iniciar la aplicación, y era interesante poder eliminar este vídeo para posibles futuros accesos a la aplicación; es por eso que se ha creado un botón que permite eliminar este vídeo, así como restaurarlo.

El diseño que se ha establecido es el siguiente:



Fig. 23: Pantalla de configuración

5.4.4.9. Multilingüismo

Tal y como se refleja en la funcionalidad anterior, la aplicación está diseñada de manera que el usuario pueda interactuar en los tres idiomas oficiales del Festival.

La detección automática del idioma funciona para el castellano en caso de tener el sistema operativo en castellano, e inglés para el resto. De manera que las personas que se descarguen la aplicación dispondrán del idioma automáticamente.

En el caso del euskera, se ha dispuesto de la aplicación en dicho idioma pero para seleccionarlo se debe acceder al menú configuración ya que no existen sistemas operativo en euskera y por tanto no se detecta ese idioma, aunque si es seleccionable.

Una vez elegido uno u otro idioma, la aplicación recuerda el lenguaje seleccionado y lo usa para futuras ejecuciones de la aplicación.

5.4.4.10. Otros objetos

Inicialmente disponemos de un vídeo cortinilla que hacen más atractiva la aplicación y le da un aspecto de mayor calidad. Este vídeo dura ciertos segundo y después automáticamente se accede al menú principal de la aplicación. Este vídeo se puede eliminar o restaurar mediante un botón en el menú de configuración.

Se han añadido varios elementos como forma de aumentar la calidad y de proporcionar mayor confort a la hora de navegar por la aplicación:

Bottom Bar: se ha añadido una barra inferior con cuatro botones seleccionables de acceso rápido para que el usuario pueda acceder al menú principal sin demoras y para poder visualizar el mapa y los contactos del Festival rápidamente ya que son funcionalidades que el usuario necesita acceder con más frecuencia.

Ejemplo:



Fig 23: Bottom Bar

Slider: se ha añadido un slider interactivo en el menú principal de la aplicación que muestra la imagen y el título de las tres últimas noticias, el último álbum de fotos y el último vídeo. Hace las funciones de portada interactiva y al seleccionar alguna de sus fotos o contenidos se muestra la noticia, álbum o vídeo en cuestión.



Fig. 24: Slider

5.5. Validación

Una vez implementada la aplicación así como durante su desarrollo se han hecho multitud de pruebas tanto con los propios emuladores de Android que ejecuta Eclipse, como con multitud de dispositivos.

5.5.1. Descripción, test

A continuación describo las pruebas realizadas para la validación de la aplicación:

En concreto se han hecho pruebas con dispositivos Samsung Galaxy, modelos SIII, SII, Ace, y Note. Y otros dispositivos como HTC, Huawei, iPro y iBold.

En la mayoría de los dispositivos utilizados para validación se encontraron ciertos aspectos a corregir pero no se encontraron grandes errores.

Cabe mencionar que en un primer momento solo se hizo validación con emuladores y esto produjo ciertos errores a la hora de tratar la aplicación en dispositivos. El más importante fue un problema con los paquetes de datos que traía desde el servidor y con algunas conexiones a la base de datos. Los principales problemas eran que los paquetes de datos eran demasiado grandes en el caso de las noticias, y tuve que limitar las noticias y traerlas por páginas de 5 en 5. Esto hacía que a la hora de utilizar el dispositivo en redes WIFI y sobre todo en redes 3G se optimizase el tiempo de espera y que la aplicación apenas tarde en traer la información.

Además, algunas redes wifi redirigen a un login por página web después de conectarse a ellas y esto hace que de error ya que a la hora de hacer llamadas al servicio web se atasca en dicho paso previo. Una vez solucionado ese paso previo propio de la red wifi, el funcionamiento es normal.

Por otro lado, se han realizado pruebas con un grupo de personas entre los que están, usuarios corrientes de Android, usuarios trabajadores del Festival, y usuarios no familiarizados con Android pero sí con otros sistemas operativos para Smartphones.

Las pruebas y test realizados tienen como objetivo conocer la conformidad del usuario con la información aportada, saber su opinión sobre usabilidad y accesibilidad, conocer las facilidades de navegación y otros aspectos destacables a mejorar.

5.5.2. Resultado

Después de realizar ciertas pruebas con los diferentes grupos de usuarios se obtienen los siguientes resultados:

G1: Usuarios corrientes de Android: una vez realizadas las pruebas de navegación se observa que la mayoría de usuarios entienden como se navega y creen que la información es bastante clara aunque coinciden que en ciertos aspectos se podría mejorar la interfaz en cuanto a aspectos gráficos, sin embargo se muestran muy cómodos utilizando la aplicación y creen que es muy usable.

En un principio opinaban que se debía simplificar el menú principal y aportar más contenidos gráficos.

También querían un menor número de clics a la hora de acceder al menú principal desde un submenú o acceso rápido a información necesaria como contactos, y mapa de una manera directa. Aspectos que en cierta manera se han solucionado.

G2: Usuarios de Android del Festival de cine de San Sebastián: como era de esperar, los usuarios que trabajan en el Festival de cine de San Sebastián siempre pedían más, querían saber más información y tener más detalles aunque destacaban que lo que se mostraba era bastante conciso y completo. También se mostraban muy cómodos a la hora de navegar por la aplicación ya que se ha orientado al estilo de la página web y a sus contenidos y se les hacía conocido.

G3: Usuarios de otros sistemas operativos móviles: Este grupo concreto estaba familiarizado con iOS y Windows Phone, y a la hora de interactuar con la aplicación en Android destacaron su comodidad y usabilidad y echaron en falta un feedback a la hora de cargar datos y una barra de botones inferior (bottom bar). Pero creyeron que los contenidos eran muy completos, que era sencilla pero eficaz y que aportaba un servicio útil y necesario.

6. Datos

La manipulación de datos en la aplicación se ha hecho con especial cuidado. Cabe destacar que la decisión de utilizar servicios web para la obtención de los datos aporta un grado de seguridad añadido, simpleza y facilidad.

6.1. Leer/obtener datos

El lenguaje PHP nos permite realizar consultas SQL sobre una base de datos MySQL alojada en el servidor mismo de la página web, cosa que es interesante ya que todos los datos de la web del Festival se van actualizando y generando en función del tiempo. Este lenguaje nos permite diseñar consultas cuyos resultados pueden ser almacenados en arrays legibles en java. Además, Java permite hacer llamadas a estas rutinas en PHP desde las actividades y obtener los arrays de respuesta.

6.1.1. Conexión con la base datos

La conexión con la base de datos se hace directamente desde el programa en PHP correspondiente. Este está alojado en el servidor de la página web y lo llamamos por dirección URL desde nuestro *Activity* concreto.

Un ejemplo de llamada y obtención de datos en el *Activity*:

```
try {
    // Llamamos al servicio web PHP para obtener los datos
    HttpGet httpGet = new HttpGet("http://www.sansebastianfestival.com/android/ejemplo.php");
    HttpClient httpClient = new DefaultHttpClient();
    HttpResponse response = (HttpResponse)httpClient.execute(httpGet);
    HttpEntity entity = response.getEntity();
    BufferedHttpEntity buffer = new BufferedHttpEntity(entity);
    InputStream iStream = buffer.getContent();

    String aux = "";
    BufferedReader r = new BufferedReader(new InputStreamReader(iStream));
    String line;
    while ((line = r.readLine()) != null) {
        aux += line;
    }
    // Parseamos la respuesta obtenida del servidor a un objeto JSON
    JSONObject jsonObject = new JSONObject(aux);
    JSONArray Aejem = jsonObject.getJSONArray("ejemplo"); //Nombre del Array JSON
    // Recorremos el Array
    for(int i = 0; i < Aejem.length(); i++) {
        JSONObject ejemplo = Aejem.getJSONObject(i);
        // Creamos el objeto con los elementos del Array en un tipo de datos predefinido
        // Almacenamos el objeto en un Array que hemos creado anteriormente
    }
}
catch(Exception e) {
    Log.e("WebService", e.getMessage());
}
```

6.1.2. PHP, Json

La aplicación utiliza Json para crear arrays que puedan ser interpretados en Java por los *Activity*. Estos arrays son creados dentro de un programa en lenguaje PHP que estando alojado en el servidor, conecta con una de las bases de datos del mismo y genera una consulta SQL. Una vez hecha la consulta sobre la base de datos, el programa en PHP recupera los datos y genera un array tipo Json con dicha información, a continuación se muestran varios ejemplos de los programas PHP desarrollados:

6.1.3. Consultas SQL

Las consultas SQL que realiza el servicio web en la base de datos del Festival, son las siguientes:

Ejemplo de consulta SQL realizada dentro de los servicios web PHP para la obtención de los datos:

```
<?php
...
```

```
$res = mysql_query
("SELECT n_paginas.id, n_paginas.fecha, n_paginas.apartado, n_paginas.antetitulo_es, n_paginas.titulo_es,
n_paginas.mostrar_es, n_paginas.foto_indice, n_apartados.apartado_id, n_apartados.apartado_es,
n_apartados.cabecera_es, n_apartados.apartado_menu, n_apartados.apartado_texto_es, n_ediciones.ano,
n_ediciones.edicion_es, n_paginas.edicion
FROM n_paginas INNER JOIN n_apartados ON n_paginas.apartado = n_apartados.apartado_id INNER JOIN n_ediciones
ON n_paginas.edicion = n_ediciones.ano
WHERE apartado = 3 AND mostrar_es = 1 AND n_paginas.apartado NOT IN ('2') AND n_ediciones.ano='2013' AND
n_paginas.orden BETWEEN 1 AND 12
ORDER BY n_paginas.orden");
```

```
...
?>
```

6.2. Crear datos

La creación de datos está descartada en esta aplicación ya que no se necesita memorizar información del usuario, y no necesitamos ninguna funcionalidad que escriba en la base de datos. Por otra parte, se observa el hecho de que en un futuro, la aplicación necesite realizar operaciones como un Login, acceso a una sección solo para personas de la industria, acceso a personal acreditado, etc., por ello la aplicación, estructura, lógica de negocio y diseño dejan la puerta abierta a estas hipotéticas funcionalidades.

6.3. Protección de datos

La aplicación carece de información personal, y tampoco la solicita. Los datos e información relativos al Festival y la protección de los mismos está regulado por la seguridad interna del propio servidor ya que las consultas que se realizan, y la parte de conexión segura con la base de datos está dentro del programa PHP y este a su vez dentro del servidor.

En resumen, la aplicación simplemente llama al programa y recibe un Array con los datos solicitados.

Cabecera del servicio web en PHP que hace la conexión con la base de datos:

```
<?php
$con = mysql_connect('Server_location', 'DB_name', 'DB_password');
mysql_query("SET CHARACTER SET utf8");
mysql_query("SET NAMES utf8");

if( $con )
```

7. Interfaz usuario, Interacción persona computador

7.1. Usabilidad

La aplicación contiene ciertos aspectos que hacen de ella una aplicación usable entre los que destacan algunos parámetros de la aplicación como pueden ser:

Fácil de aprender: la aplicación se ha diseñado para que su aprendizaje se haga con facilidad; nuevos usuarios desarrollan una interacción efectiva con el producto. Es predecible, y es familiar, es decir, la navegación es conocida y común en otras aplicaciones.

Fácil de usar: Es una aplicación que tiene facilidades de uso, tiene menús simples, se requieren pocos toques/clics, y es eficaz.

Flexible: La aplicación tiene variedad y las tareas que se realizan son similares entre sí.

Robusta: Se apoya al usuario a alcanzar sus objetivos en cuanto a la aplicación, orientada a su observación y la información recogida.

7.2. Accesibilidad

Entre los dispositivos que dispone Android para dotar de cierta accesibilidad en sus aplicaciones encontramos los lectores y magnificadores de pantalla o alto contraste para personas ciegas o con resto visual. La aplicación a intentado seguir un diseño visual de líneas y colores orientadas a aportar una pequeña ayuda a las personas que necesitan de estas funcionalidades.

7.3. Pruebas

Dentro de los test de validación realizada con varios grupos de personas, se observan ciertos aspectos de mejora de la interfaz de usuario, en cuanto a propiedades tan importantes como la usabilidad y accesibilidad. Se observó que el usuario necesitaba fluidez, y por ellos se acortaron el número de funcionalidades en el menú principal, integrando varias en una misma, y además se redujo el número de toques para acceder a la información sin perder eficacia ni involucran un incremento del aprendizaje, también se aportó un grado más de información incluyendo el slider de cinco elementos diseñado a modo de portada de novedades y que permitían acceder a los últimos contenidos de manera veloz.

7.4. Capacidad de mejora

Las posibilidades de mejora en estos términos de la interfaz de usuario de la aplicación, pasan por hacer un mejor estudio de uso en diferentes grupos y analizar las carencias o problemas que pudiera tener la aplicación.

Como idea general, se observa que la principal mejora que puede tener la aplicación sería en el menú principal, agregando otros mecanismos de menú más novedosos como *TabHost* deslizantes, o haciendo más interactivo y con más contenidos multimedia la presentación del menú principal.

7.5. Visualización y desarrollo

El desarrollo de la interfaz de usuario ha supuesto varios pasos que se detallan a continuación:

- Creación de los menús principales
- Estructuración de los Layouts
- Diseño de iconos y botones
- Aplicación del color y estilos

A continuación muestro un ejemplo del layout del menú principal:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:android1="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android1:layout_gravity="center_vertical"
    android:baselineAligned="true"
    android:orientation="vertical"
    android1:gravity="center_horizontal" >
    <ImageView
        android1:id="@+id/imageView2"
        android1:layout_width="wrap_content"
        android1:layout_height="wrap_content"
        android1:adjustViewBounds="true"
        android1:src="@drawable/edicion61largo_sin" />
    <ScrollView
        android1:id="@+id/scrollView1"
        android1:layout_width="match_parent"
        android1:layout_height="wrap_content"
        android:overScrollMode="always"
        android:isScrollContainer="true"
        android:scrollbarAlwaysDrawVerticalTrack="true"
        android:scrollbarStyle="outsideInset"
        android:scrollbars="vertical">
        <LinearLayout
            android1:layout_width="match_parent"
            android1:layout_height="match_parent"
            android1:orientation="vertical" >
            <LinearLayout
                android1:id="@+id/LayoutSlider"
                android1:layout_width="match_parent"
                android1:layout_height="wrap_content"
                android1:layout_marginBottom="6dp"
                android1:layout_marginLeft="2dp"
                android1:layout_marginRight="2dp"
                android1:layout_marginTop="4dp"
                android1:background="@drawable/red_button_style"
                android1:orientation="vertical" >
                <ImageView
                    android1:id="@+id/imageView3"
                    android1:layout_width="match_parent"
                    android1:layout_height="192dp"
                    android1:layout_weight="1"
                    android1:clickable="true" />
                <TextView
                    android1:id="@+id/textView1"
                    android1:layout_width="fill_parent"
                    android1:layout_height="wrap_content"
                    android1:layout_gravity="center|center_horizontal"
                    android1:layout_marginBottom="2dp"
                    android1:layout_marginLeft="4dp"
                    android1:layout_marginRight="2dp"
                    android1:layout_weight="1"
                    android1:clickable="true"
                    android1:text="TextView"
                    android1:textColor="#FFFFFF"
                    android1:textStyle="bold" />
            </LinearLayout>
        </ScrollView>
    </LinearLayout>
```

```

<TableLayout
    android1:layout_width="match_parent"
    android1:layout_height="match_parent"
    android1:orientation="vertical" >

    <TableRow
        android1:layout_width="fill_parent"
        android1:layout_height="wrap_content"
        android1:layout_marginBottom="6dp"
        android1:layout_marginLeft="2dp"
        android1:layout_marginRight="2dp"
        android1:gravity="center_horizontal" >
        <Button
            android1:id="@+id/start_noticias"
            android1:layout_width="0dp"
            android1:layout_height="match_parent"
            android1:layout_marginRight="4dp"
            android1:layout_weight="1"
            android1:background="@drawable/red_button_style"
            android1:text="@string/noticias"
            android1:textColor="#FFFFFF"
            android1:textStyle="bold" />
        <Button
            android1:id="@+id/start_media"
            android1:layout_width="0dp"
            android1:layout_height="match_parent"
            android1:layout_marginLeft="4dp"
            android1:layout_weight="1"
            android1:background="@drawable/red_button_style"
            android1:text="@string/media"
            android1:textColor="#FFFFFF"
            android1:textStyle="bold" />
    </TableRow>

    <TableRow
        android1:layout_width="fill_parent"
        android1:layout_height="wrap_content"
        android1:layout_marginBottom="6dp"
        android1:layout_marginLeft="2dp"
        android1:layout_marginRight="2dp"
        android1:gravity="center_horizontal" >
        <Button
            android1:id="@+id/start_premios"
            android1:layout_width="0dp"
            android1:layout_height="match_parent"
            android1:layout_marginRight="4dp"
            android1:layout_weight="1"
            android1:background="@drawable/red_button_style"
            android1:text="@string/premios"
            android1:textColor="#FFFFFF"
            android1:textStyle="bold" />
        <Button
            android1:id="@+id/start_secciones"
            android1:layout_width="0dp"
            android1:layout_height="match_parent"
            android1:layout_marginLeft="4dp"
            android1:layout_weight="1"
            android1:background="@drawable/red_button_style"
            android1:text="@string/secciones"
            android1:textColor="#FFFFFF"
            android1:textStyle="bold" />
    </TableRow>

    <TableRow
        android1:layout_width="fill_parent"
        android1:layout_height="wrap_content"
        android1:layout_marginBottom="6dp"
        android1:layout_marginLeft="2dp"
        android1:layout_marginRight="2dp"
        android1:gravity="center_horizontal" >
        <Button
            android1:id="@+id/start_programacion"
            android1:layout_width="0dp"
            android1:layout_height="match_parent"
            android1:layout_marginRight="4dp"

```

```

        android1:layout_weight="1"
        android1:background="@drawable/red_button_style"
        android1:text="@string/programacion"
        android1:textColor="#FFFFFF"
        android1:textStyle="bold" />
    <Button
        android1:id="@+id/start_entradas"
        android1:layout_width="0dp"
        android1:layout_height="match_parent"
        android1:layout_marginLeft="4dp"
        android1:layout_weight="1"
        android1:background="@drawable/red_button_style"
        android1:text="@string/entradas"
        android1:textColor="#FFFFFF"
        android1:textStyle="bold" />

</TableRow>
</TableLayout>
</LinearLayout>
</ScrollView>
</LinearLayout>

```



Fig. 25: Pantallas portrait y landscape del Menú principal

8. Gestión del Proyecto

A continuación se detallan los aspectos más significativos de la gestión del presente proyecto, primeramente se hizo una estimación del tiempo que se debería haber empleado para la consecución de los objetivos. Así como un análisis de los riesgos, costes, etc. que podían ayudar a la correcta marcha y finalización del proyecto.

Seguidamente se ha hecho un control y seguimiento de las tareas realizadas para acercarse sin problemas a los objetivos del proyecto.

8.1. Estimación de Tiempo

A continuación se muestra una tabla con las actividades programadas y el tiempo que fue estimado inicialmente para cada una de ellas así como el tiempo empleado finalmente en su realización:

Actividades	Tiempo estimado	Tiempo dedicado	Desvío
1. Inicio			
1.1 Inicio de gestiones PFC	1	2	+1
2. Planificación			
2.1. Alcance	6	5	-1
2.2. Tiempos	8	9	+1
2.3. Costes	1	1	0
2.4. Riesgos	6	4	-2
2.5. Calidad	6	6	0
2.6. Comunicaciones	4	4	0
2.7. Gestión de Cambios	4	4	0
2.8. Recursos Humanos	1	1	0
2.9. Adquisiciones	3	4	+1
2.10. Gestión de la Planificación	3	3	0
3. Formación			
3.1. Curso Android UPV- MiriadaX	120	140	+20
3.2. Estudio herramienta ECLIPSE	4	2	-2
3.3. Gestión de proyectos	2	2	0
3.4. Gestión de la Formación	1	1	0
4. Diseño			
4.1. Interfaz Gráfica	3	6	+3
4.2. Estructura	3	3	0
4.3. Botones y Logotipos	4	6	+2
4.4. Menús y Actividades	5	8	+3

4.5. Gestión Diseño	1	1	0
5. Implementación			
5.1. Aplicación Android			
5.1.1. Noticias del Festival	16	20	+4
5.1.2. Galería fotográfica con álbumes	8	7	-1
5.1.3. Videoteca y TV	10	11	+1
5.1.4. Buscador Programación	10	15	+5
5.1.5. Secciones cinematográficas del Festival	8	8	0
5.1.6. Premios y Premiados	10	9	-1
5.1.7. Enlace a Twitter	2	1	-1
5.1.8. Enlace a Facebook	2	1	-1
5.1.9. Enlace a Flickr	2	1	-1
5.1.10. Enlace al sistema de tele-venta de entradas	2	1	-1
5.1.11. Sección de contactos con hipervínculos	4	4	0
5.1.12. Mapa de localización de puntos de interés	4	3	-1
5.2. Interfaz gráfica			
5.2.1. Estructuración	5	10	+5
5.2.2. Ubicación de contenidos	5	6	+1
5.3. Botones y Logotipos	10	12	+2
5.4. Menús y actividades	12	12	0
5.5. Multilingüismo	8	15	+7
5.6. Gestión de la Implementación	3	3	0
6. Pruebas			
6.1. Pruebas con un grupo de usuarios	10	12	+2
6.2. Pruebas usabilidad y utilidad	4	4	0
6.3. Pruebas de respuesta frente a cambios	6	5	-1
6.4. Pruebas específicas funcionalidad	6	4	-2
6.5. Gestión de Pruebas	3	1	-2
7. Reuniones			
7.1. Reuniones con las partes	12	12	0
8. Control y Seguimiento			
8.1. Alcance	10	5	-5
8.2. Tiempos	8	8	0
8.3. Riesgos	8	6	-2
8.4. Cambios	8	6	-2
8.5. Calidad	8	10	+2
8.6. Comunicaciones	8	5	-3

8.7. Costes	2	1	-1
8.8. Adquisiciones	4	4	0
8.9. Recursos Humanos	2	1	-1
8.10. Gestión del Seguimiento y Control	4	4	0
9. Cierre			
9.1. Creación presentaciones	1	3	+2
9.2. Presentación al consejo del Festival	1	1	0
9.3. Presentación de defensa en la Universidad	1	1	0
Tiempo total	413	444	+31

Tabla 2: Estimación de tiempos y tiempo necesario real.

Tiempo dedicado total

A continuación se muestra una tabla con las fases principales y el tiempo estimado de dedicación para cada una de ellas:

Actividades	Tiempo estimado	Tiempo dedicado	Desvío
1. Inicio	1	2	+1
2. Planificación	42	41	-1
3. Formación	127	145	+18
4. Diseño	16	24	+8
5. Implementación	121	124	+3
6. Pruebas	29	26	-3
7. Reuniones	12	12	0
8. Control y Seguimiento	62	50	-12
9. Cierre	3	5	+2
Tiempo total	413	444	+31 (+8,07%)

Tabla 3: Resumen de tiempos totales

Se puede observar que el proyecto ha sufrido un incremento en los tiempos estimados aunque individualmente hay actividades en las que se estimó un tiempo que después no ha sido necesario. En resumen, la desviación no ha sido muy elevada ya que no supera el 10 % y puedo decir que el proyecto ha seguido la estimación inicial, con un incremento lógico de no más de un 10% del tiempo estimado, lo cual se puede considerar muy aceptable.

Cronograma del proyecto

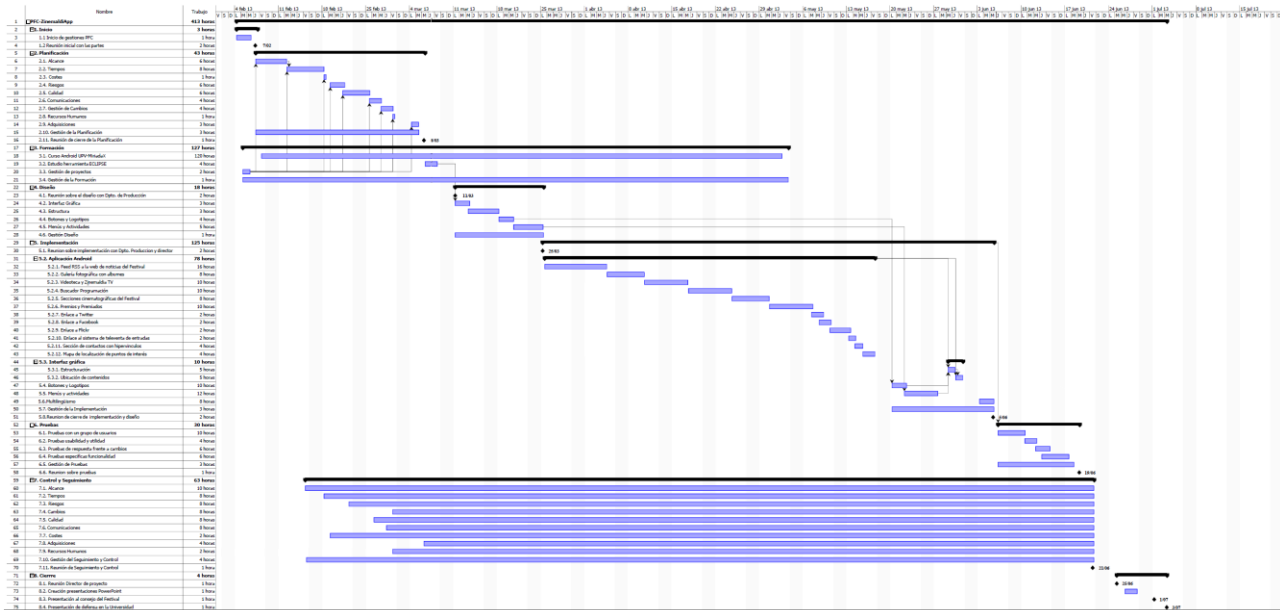


Tabla 4: Cronograma del proyecto

8.2. Hitos

Nombre	Trabajo	feb '13	mar '13	abr '13	may '13	jun '13	jul '13	
EPFC-ZinemaldiaApp	413 horas	[Barra de actividad principal]						
E1. Inicio	3 horas	[Barra de inicio]						
1.2 Reunión inicial con las partes	2 horas	[Barra de reunión]						
E2. Planificación	43 horas	[Barra de planificación]						
2.1.1. Reunión de cierre de la Planificación	1 hora	[Barra de reunión]						
E4. Diseño	18 horas	[Barra de diseño]						
4.1. Reunión sobre el diseño con Dpto. de Producción	2 horas	[Barra de reunión]						
E5. Implementación	125 horas	[Barra de implementación]						
5.1. Reunión sobre implementación con Dpto. Producción y director	2 horas	[Barra de reunión]						
5.8. Reunión de cierre de implementación y diseño	2 horas	[Barra de reunión]						
E6. Pruebas	30 horas	[Barra de pruebas]						
6.6. Reunión sobre pruebas	1 hora	[Barra de reunión]						
E7. Control y Seguimiento	63 horas	[Barra de control]						
7.1.1. Reunión de Seguimiento y Control	1 hora	[Barra de reunión]						
E8. Cierre	4 horas	[Barra de cierre]						
8.1. Reunión Director de proyecto	1 hora	[Barra de reunión]						
8.3. Presentación a consejo de Festival	1 hora	[Barra de presentación]						
8.4. Presentación de defensa en la Universidad	1 hora	[Barra de presentación]						

Tabla 5: Hitos principales del proyecto.

En la tabla 5 se muestran los hitos principales del proyecto. Se han cumplido los hitos establecidos y se han conseguido los objetivos marcados en cada uno de ellos.

Hay que destacar que se señalaron unos plazos de control con el director del proyecto para el primer día de cada mes. En lo referente a estos plazos establecidos, se han cumplido los plazos marcados en el proyecto pero hay que destacar que hubo un retraso sin importancia en la entrega del 2 de mayo debido a problemas leves con la

implementación de la aplicación. En resumen se han cumplido los plazos establecidos como objetivos.

8.3. Análisis de Riesgos

En este apartado se estudiaron los diferentes riesgos que podían encontrarse y se encontraban en el proyecto así como el plan de acción frente a estos riesgos. La gestión de riesgos duró toda la realización del proyecto, y se ha llevado un seguimiento y control de los mismos.

Riesgos:

- Indisponibilidad de alguna de las partes: la ausencia del alumno o del tutor de la empresa obligada por enfermedades u otros motivos podía ocasionar graves retrasos o incluso la cancelación del proyecto.
- Pérdida de información: La pérdida de datos o cualquier otro tipo de información debida a diferentes motivos, como averías en el equipo o borrados accidentales, que podían ocasionar graves retrasos.
- Problemas de desarrollo: falta de experiencia por parte del alumno en la utilización de alguna de las herramientas y tecnologías aplicadas en el proyecto.
- Estimación errónea: la inexperiencia en la planificación y gestión de proyectos, podía provocar una estimación de incorrecta que afectase a la duración prevista del proyecto.

Cuantificación de riesgos

En la siguiente tabla tenemos una enumeración de los riesgos junto a una estimación de la probabilidad de que ocurra. También se valoró el impacto que podían tener en la realización del proyecto en caso de que alguno de ellos sucediera.

Riesgo	Probabilidad	Impacto	Valoración
Indisponibilidad de alguna de las partes	Baja	Alto	Medio
Pérdida de información	Baja	Alto	Alto
Problemas en el desarrollo	Alta	Medio	Medio
Estimación errónea	Media	Bajo	Medio

Tabla 6: Cuantificación de riesgos

Plan de contingencia y actuación.

Indisponibilidad de alguna de las partes: Si el alumno sufría una indisponibilidad moderada, éste debía recuperar las horas perdidas.

En caso de que la indisponibilidad se alargara en el tiempo demasiado había que revisar la planificación de la fecha de entrega.

Si la indisponibilidad era causada por alguno de los tutores durante un largo periodo de tiempo, se procedería a buscar un sustituto.

Pérdida de información: Para tratar de evitar esta situación se llevaría a cabo una política de copias de seguridad consistente en:

- Realizar copias diarias de las aplicaciones y de la memoria.
- Dichas copias se almacenarían en un pendrive, en el PC en casa y en un servidor privado del Festival de Cine.

Problemas en el desarrollo: Para evitar este riesgo se trataría de reunir previamente la mayor información posible orientada a cumplir los objetivos de este proyecto. Si esta información no era suficiente durante el desarrollo, se procedería a buscar la información necesaria durante la fase de desarrollo o tomar otras alternativas.

Estimación errónea: si se producía esta situación, se replanificaría la fase en la que se produjo y las posteriores fases del proyecto.

8.4. Análisis de Costes

Los costes del proyecto han sido mínimos y a destacar los siguientes gastos que ha tenido el proyecto:

- PC para trabajar: 500 horas x 0,09€/hora* = 47,30 €
- Dispositivo Android iPro: 155,20 €
- Tablet Fnac: 199,90 €
- Desplazamientos: aprox. 50 €**
- Costes de material de oficina e impresiones: 20 €

TOTAL= 517,30 €

*La amortización de los ordenadores y Hardware utilizándose ha *realizado de la siguiente manera:*

Coste medio de un ordenador: 1.000,00 €

Tiempo de utilización medio es de cinco años, equivalentes a: 8 horas lab. x 22 días lab. x 12 meses x 5 años= 10.560 horas

Por todo ello el coste por hora de utilización es de: 1.000€/10.560h = 0,09€/hora

***El coste de desplazamientos es ponderado y teniendo en cuenta desplazamientos en bicicleta y transporte público.*

Plan de contingencia a problemas presupuestarios.

En el caso de que haber tenido problemas con el presupuesto, fuera limitado o lo hubiera sido superado antes del cierre del proyecto, hubiera seguido adelante con aquellos recursos posibles, a precio mínimo.

8.5. Análisis de Adquisiciones

Las adquisiciones necesarias para este proyecto fueron varias y siempre cumpliendo unas cualidades:

Las principales cualidades fueron la adquisición de elementos al mínimo precio o gratis, como los iconos que eran objetos de uso libre.

Adquirir y utilizar dispositivos propios de la gente, así como un dispositivo propio para pruebas de desarrollo.

Adquirir programas y código libre y aportes de información de libre uso y distribución.

Planificación:

- OpenProj(Cronograma) [Licencia GPLv2 (software libre)]
Coste: 0,00€
Duración: Ilimitada
<http://www.openproject.org/>
- Dia (diagramas) [Licencia GPLv2 (software libre)]
Coste: 0,00€
Duración: Ilimitada
<http://live.gnome.org/Dia>
- OpenOffice (Redacción de documentos) [GNU Lesser General Public License v3]
Coste: 0,00€
Duración: Ilimitada
http://www.openoffice.org/licenses/lgpl_license.html
- DropBox (Intercambio de ficheros) [Cerrada (Clientes de Windows & Mac clients y de Linux dropbox daemon), Abierta (nautilus-dropbox)]
Coste: 0,00€
Duración: Ilimitada
En cualquier caso la versión utilizada de Dropbox es gratuita, no infringe licencia alguna con su uso
<https://www.dropbox.com/dmca>
- Gmail (cuentas para comunicaciones internas) [Licencia de Software libre]
Coste: 0,00€
Duración: Ilimitada
<http://www.google.com/accounts/TOS?loc=ES&hl=es>

Formación:

- Curso on-line de aplicaciones móviles Android MiriadaX. Universidad politécnica de Valencia.
Coste: 0,00€
Duración 150 horas.
<http://www.miriadax.net>

Desarrollo:

- Soporte Youtube
Coste: 0,00€
Duración: Ilimitada
<http://www.youtube.com/t/terms>

- Eclipse (ADT free distribution)
Coste: 0,00€
Duración: Ilimitada
<http://developer.android.com/tools/sdk/eclipse-adt.html>
- Adobe Dreamweaver y Photoshop.ADOBE cs5.
Coste: 0,00€ (Licencia de la empresa)
Duración: Ilimitada
<http://www.adobe.com/products/eulas>
- Navicat premium:
Coste: 0,00€ (Licencia de la empresa)
Duración: Ilimitada
<http://www.navicat.com/products/navicat-premium>
- CuteFTP 9.0:
Coste: 0,00€
Duración: Ilimitada
<http://www.cuteftp.com/cuteftp>
- VideoPad: (free license)
Coste: 0,00€
Duración: ilimitada
<http://www.nchsoftware.com>

8.6. Análisis de Comunicación

El plan de comunicaciones de las partes se resume en dos partes:

Por un lado teníamos la comunicación con el director del proyecto. Esta comunicación se hizo vía mail corporativo de la facultad y el intercambio de documentos o archivos se realizó aportando enlaces de descarga en dropbox, google docs o similares.

Por otro lado teníamos la comunicación con el responsable de informática del Festival, el contacto con él era mayor dado que se trabajó en el propio Festival de manera conjunta, sin embargo la comunicación digital se realizó mediante mail del propio Festival y personal. El intercambio de archivo o datos se realizó mediante la red interna del Festival de Cine de San Sebastián.

Contacto:

- Ramón Velasco López: rvelasco008@ikasle.ehu.es, ramonvelasco0@gmail.com
- José Miguel Blanco: josemiguel.blanco@ehu.es
- Mikel Ormaetxea Isasi: sistemas@sansebastianfestival.com

Reuniones: Se realizaron reuniones con cada uno de los responsables cada cierto tiempo, y como mínimo una vez al mes que fue a primeros de mes para mostrar el trabajo realizado y concretar las líneas de trabajo futuras.

8.7. Análisis de Cambios

Diferenciamos básicamente tres tipos de solicitudes de cambio, las provenientes del Festival de Cine sobre el producto, las provocadas por el cumplimiento de algún riesgo y

la ejecución de su plan de contingencia asociado y otros tipos de cambios.

En el caso de los cambios a realizar por parte del Festival de cine, se realizaron todos aquellos cambios que solicitaron siempre y cuando no supusieran una modificación muy elevada del planteamiento general del alcance de la aplicación y contribuyeran a la no finalización del proyecto.

En el tema de los riesgos, la forma de actuar fue analizar el riesgo acontecido y ejecutar la solución y cambio oportuno contemplado en su plan de contingencia.

En el caso de otros cambios no destaca ninguno salvo aquellos que requirieron un tiempo elevado o moderado y tuvieron que ser acometidos indispensablemente. En estos casos, los cambios se hicieron lo más brevemente posible y se dejó constancia.

8.8. Otros análisis

Como recursos humanos disponía de mí mismo, para cualquier ayuda consulta o sugerencia acudí al director de proyecto, y al tutor de la empresa y compañeros del Festival.

8.9. Control y seguimiento

En este informe de seguimiento se pretende mostrar el progreso del proyecto a lo largo de su ciclo de vida: esclarecer qué aspectos han cambiado; a qué se han debido estos cambios, si han sido fruto de la activación de un riesgo o simplemente a una decisión rutinaria para reconducir el proyecto y facilitar la consecución de los objetivos de este.

Por supuesto se comprobará que los objetivos se han cumplido: cómo ha variado el alcance y si se han superado los requisitos mínimos que este imponía.

El proyecto llevado a cabo es un proyecto que ha ido creciendo a lo largo del tiempo de una manera bastante fluida sin demasiados sobresaltos, aunque por supuesto con momentos claves que han requerido de toma de decisiones críticas para una buena continuación.

A día de hoy no se puede decir que el proyecto haya estado en peligro en ningún momento, ni siquiera en los momentos en que aspectos ajenos a este han activado riesgos que han amenazado ciertas fases de su ciclo de vida.

Estado general del proyecto

El objetivo final del proyecto era la elaboración de una aplicación móvil Android para el Festival de Cine de San Sebastián, con unos requisitos mínimos establecidos a petición del Festival. Estos requisitos fueron los que junto con una serie de condiciones, restricciones y asunciones compusieron el alcance del proyecto. Este ha ido creciendo a medida que lo ha hecho el proyecto, añadiéndose nuevos aspectos surgidos con el desarrollo del proyecto.

Como se ha dicho anteriormente los requisitos del alcance se convirtieron en obligaciones contractuales con el Festival, pero además de esto, se propusieron unos requisitos de calidad mínimos de los que también se asumió la obligación de cumplir, para asegurar la consecución de no sólo un producto bueno sino también de un proyecto exitoso

internamente.

Si bien no se va a profundizar, es digno de destacar que el 100% de los requisitos del Festival y el 100% de los requisitos de calidad del proyecto se han cumplido y que por tanto se ha superado la línea base establecida y se han logrado los objetivos mínimos propuestos.

Grado de consecución de los resultados del proyecto

Para analizar el grado de cumplimiento o consecución de los objetivos en base a los resultados obtenidos se ha medido el porcentaje de requisitos superados, tanto los denominados requisitos mínimos, como los denominados requisitos de calidad como las líneas de mejora. Consecución de las actividades propuestas, etc.

De acuerdo a los diferentes cambios provocados por riesgos o por otros motivos, se produjeron modificaciones en las actividades planificadas inicialmente y se modificaron algunos tiempos.

A lo largo del proyecto se han produjeron cambios de mayor o menor envergadura que fueron registrados para así conocer donde existían las modificaciones y replanteamientos más importantes.

El seguimiento de la calidad del proyecto se fue midiendo a través de periódicas revisiones de calidad, verificando que los objetivos se iban alcanzando.

El seguimiento del resto de planificaciones ha sido constante y como conclusión decir que se han completado los objetivos marcados al principio en un 100%.

9. Conclusiones y líneas futuras de trabajo

En este capítulo se expone un resumen general de lo que ha sido el diseño, desarrollo, y creación de todos los contenidos y de la aplicación en sí.

Se exponen los resultados, las ideas de futuro como nuevas funcionalidades, las capacidades de aumento de la aplicación y las conclusiones obtenidas en el desarrollo del presente proyecto.

9.1. Concordancia entre resultados y objetivos

Los objetivos marcados en un principio fueron los de crear un aplicación destinada al público general que pudiera aportar un extra de información sobre el Festival de Cine, además, la aplicación debía actualizarse automáticamente y debía disponer de un grado de calidad suficiente para representar a una entidad tal como el Festival de Cine.

Analizando los resultados a la terminación del proyecto, considero haber alcanzado los objetivos marcados en un principio, ya que la aplicación aporta información bien estructurada y de manera sencilla y eficaz. Por otro lado, sus contenidos se actualizan automáticamente y el diseño y desarrollo de la aplicación está orientado a que el mantenimiento a realizar sea mínimo.

Finalmente, la aplicación, su diseño, interfaz, y tanto las características visuales como la lógica de negocio, disponen de una gran calidad y es perfectamente utilizable de manera empresarial.

9.2. Líneas futuras. Creación de nuevas características. *I+D*

Dentro de las ideas que surgen durante el desarrollo de la aplicación, junto con las demandas de compañeros del Festival, opiniones propias y ajenas, se observan ciertas líneas de trabajo para un futuro junto con proyectos de nuevas características.

Las principales han sido las siguientes:

Creación de un apartado personal para el sector de prensa, con acceso a demanda de acreditaciones, buscador de pases películas privados, solo para prensa o información relativa al público concreto del mundo de la prensa escrita y prensa gráfica.

También se posibilita la creación de un login para los accesos privados que pueden requerir algunas de las nuevas funcionalidades mencionadas.

El Festival también dispone de una agenda de contactos privada con gente de la Industria cinematográfica solo accesible mediante login y sería interesante poder aportar este servicio.

Dentro de las líneas de mejora, se encuentran otras más difíciles de llevar a cabo y que requieren de cierta investigación y desarrollo exhaustivo. Una de ellas podría ser la de enviar notificaciones con los nuevos contenidos, noticias, imágenes, etc.

También puede ser la creación de un sistema de regalo de entradas a última hora por posicionamiento GPS cerca de las salas de proyección. Este sistema notificaría al usuario cercano al lugar que dispone de cinco minutos justo antes de empezar el pase para coger una entrada gratuita e ir al cine.

Estas últimas ideas requieren de cierta investigación adicional y son menos prioritarias.

9.3. Capacidad de aumento

La principal vía de aumento que se observa durante el desarrollo de la aplicación y al final del mismo es una mejora de la interfaz, de la visualización de contenidos y del aumento del tráfico de datos.

Se observó a esto último que el servidor del Festival podría soportar un tráfico añadido elevado a medida que aumentara la descarga de la aplicación. Se vería como actuar con el tiempo y se trataría de aumentar el ancho de banda del acceso al servidor y a la base de datos.

9.4. Lecciones aprendidas

Durante el desarrollo de la aplicación, se observa que he cometido errores y fallos de principiantes y cuestiones que merece la pena reflejar para no cometer en un futuro.

Las principales lecciones aprendidas son varias:

1.- Utilizar siempre un dispositivo para probar la aplicación y no solo el emulador. La ejecución de una aplicación en un terminal es muy diferente a ejecutarla en el emulador de eclipse. Algunas funcionalidades de Android no funcionan correctamente o no se pueden realizar dentro de un emulador y viceversa. Además la ejecución de una aplicación en el emulador tiene ciertas ventajas, como el ancho de banda de Internet frente al 3G de los móviles, ejecuciones más rápidas, etc.

2.- Realizar la traducción de la aplicación desde el principio mediante las carpetas de Strings: simplifica el trabajo siguiente y además permite crear nuevos idiomas fácilmente, si no lo haces desde el principio luego el trabajo de sustitución es muy tedioso.

3.- Uso de servicios web en contraposición a la conexión directa con el servidor. Es mejor realizar servicios web para traer información concreta del servidor y de la base de datos que abrir una conexión directa desde la aplicación con la base de datos y exponerse a problemas de seguridad, pérdida de información, etc.

4.- Programar las funcionalidades como proyectos independientes para no “ensuciar” el código: Es recomendable por propia experiencia separar las funcionalidades y después unirlos, ya que así permite validar cada funcionalidad de manera independiente y no contaminamos el código que ya funciona correctamente, además, sabiendo que a los informáticos nos gusta el método del prueba y error, es recomendable separar funciones.

5.- No intentar cosas que están muy fuera de tu alcance o de tus conocimientos. Hay que ser ambicioso pero no hay que intentar realizar funciones y servicios que vemos en otras aplicaciones y nos inspiran pero que son demasiado laboriosas o requieren cierto grado de experiencia. Es mejor realizar cosas que sabes e ir poco a poco.

9.5. Conclusiones

El presente proyecto ha sido todo un reto para mí, empecé con él hace casi un año y desde entonces he sido capaz de buscar mis propios métodos, de crear una aplicación completa, eficaz y útil partiendo de cero, y además he aprendido muchos aspectos sobre la programación de aplicaciones en general y para sistemas operativos Android en particular.

Debo reconocer que había momentos de verdadera frustración, desconocía como afrontar un proyecto de esta magnitud y tenía mis momentos de debilidad. Se generaban dudas y las funcionalidades se resistían, además de tener que ser autodidacta en la materia Android (salvo lo aprendido en un curso online) y tener que responder mis dudas sin grandes ayudas.

Por otro lado, al final las cosas van saliendo, y se van desarrollando y la satisfacción de ver que el trabajo da sus frutos y las cosas salen adelante es magnífica.

Estas son las conclusiones en cuanto a mis sensaciones en la realización del proyecto. Sin embargo las conclusiones referentes al propio proyecto son distintas. Creo que he conseguido realizar un buen proyecto y una buena aplicación, creo que he alcanzado todos mis objetivos y he superado el alcance inicial que establecí.

También creo que el desarrollo ha sido limpio sin grandes complicaciones y el producto final muestra una aplicación útil y bien desarrollada que perfectamente se puede publicar en los mercados actuales.

10. Bibliografía

- Wikipedia: Artículo sobre Android.
<http://es.wikipedia.org/wiki/Android>
- Gráfico del tráfico de datos por SO Móvil.
<http://www.pingdom.com>
- Interfaz Gráfica de Android UI.
<http://www.terminalsandroid.com/montando-la-interfaz-grafica-en-android-parte-i>
- Usando layouts basadas en XML con Android.
<http://androideity.com/2011/07/11/usando-layouts-basadas-en-xml-con-android/>
- Android develop tools guide.
<http://developer.android.com/tools/help/>
- PHP
<http://es.wikipedia.org/wiki/PHP>
- JSON
<http://es.wikipedia.org/wiki/Json>
- Ejemplo JSON:
<http://www.json.org/example.html>
- Introducing JSON
<http://www.json.org/>
- Tutorial de PHP y MySQL
<http://paginaweb2.com/cat-php-y-mysql-6>
- Gráfico de ventas de dispositivos: Fuente: Localytics.
<http://static.elandroidelibre.com/wp-content/uploads/2013/02/top-10-android-devices-january-2013.png>
- Firma de aplicaciones Android:
<http://androideity.com/2011/08/25/%C2%BFcomo-firmar-aplicaciones-android/>
- Guia del PMBOK: Fundamentos para la Dirección de proyectos: 4ª edición
- Android: JSON: Obtener y enviar datos JSON mediante JSONObject
<http://www.inforjmr.es/?p=497>
- Patrones y expresiones regulares
<http://developer.android.com/reference/java/util/regex/Pattern.html>
- Implementing Video Playback on Android using the VideoView and MediaController Classes
http://www.techotopia.com/index.php/Implementing_Video_Playback_on_Android_using_the_VideoView_and_MediaController_Classes

- Introducción a los Layouts para Android
http://www.programacion.com/articulo/introduccion_a_los_layouts_para_android_400

Anexo I: Manual del desarrollador:

Este sencillo manual se ha creado con el objeto de aportar al programador que desee modificar contenidos, una guía sobre los elementos que debe modificar dependiendo de sus necesidades, enfocando el manual a cambios de una edición del Festival a otra.

Todo desarrollador que desee modificar la edición del Festival, la información y contenidos debe tener en cuenta el cambio de edición y en concreto los siguientes elementos en orden de funcionalidad:

1. **Cortinilla:** Este vídeo está alojado en la carpeta Res del proyecto en eclipse. Para más información el vídeo debe estar en los siguientes formatos:

<u>Type</u>	<u>Format / Codec</u>	<u>Encoder</u>	<u>Decoder</u>	<u>Details</u>	<u>Supported File Type(s) / Container Formats</u>
Video	H.263	•	•		<ul style="list-style-type: none"> • 3GPP (.3gp) • MPEG-4 (.mp4)
	H.264 AVC	• (Android 3.0+)	•	Baseline Profile (BP)	<ul style="list-style-type: none"> • 3GPP (.3gp) • MPEG-4 (.mp4) • MPEG-TS (.ts, AAC audio only, not seekable, Android 3.0+)
	MPEG-4 SP		•		3GPP (.3gp)
	VP8		• (Android 2.3.3+)	Streamable only in Android 4.0 and above	<ul style="list-style-type: none"> • WebM (.webm) • Matroska (.mkv, Android 4.0+)

Fuente: Android supported media Formats (<http://developer.android.com/guide/appendix/media-formats.html>)

Un cambio de edición supondría modificar el vídeo a uno nuevo y aplicarle formato específico.

En caso de no poner nuevos vídeos se debe tener en cuenta que el menú de configuración dispone de un selector en el caso que se desee eliminar dicho vídeo y se debería inhabilitar esta función.

2. Dentro del **menú principal**, y en el resto de submenús existe una imagen cabecera que debería modificarse con un cambio de edición.

Es una imagen con formato .png y en ella se contiene el logo o parte del logo del Festival en la edición que acontece y el número de edición aunque estos parámetros son variables. Las imágenes se alojan en la carpeta drawable.

3. Los **colores** oficiales de la aplicación, en cuanto a fondos de vistas de imágenes, slider, texto o cualquier otro elemento deben modificarse de una edición a otra. Este valor se selecciona en las características de cada una de las vistas contenidas en los layouts de la aplicación y está en formato hexadecimal.

Los layouts también contienen las imágenes de iconos que en caso de modificar debemos acudir a la carpeta drawable.

4. Un cambio en algunos **enlaces** como los de Facebook, Twitter, Flickr, adición de redes sociales o tele-venta de entradas deben tenerse en cuenta y actualizarse en el caso de que tengamos cambios en las URL de enlace.

En cuanto a los contenidos, la mayoría se actualizan solos, aunque hay que considerar los siguientes casos:

5. En el caso de que las **consultas SQL** dispongan un discriminador por fecha, se deberá cambiar este valor de un año para otro dentro de la carpeta en la que se alojan los servicios web y más concretamente donde se detallan las consultas SQL.

En el caso de las noticias, premios, secciones, programación, imágenes y vídeos, etc. Todos ellos obtienen la información desde la base de datos y requieren una pequeña modificación en el año de selección.

Actividad: Noticias, Imágenes, Vídeos, Premios, Secciones, Programación.

6. **Algunas imágenes** principales de las secciones y de los premios, no están incluidas en la base de datos que se componen, al no poder modificar la base de datos, hay algunas fotos que han sido introducidas directamente en el propio servicio web (.php). Estas fotos también se modifican si hay un cambio de edición y es importante actualizar las rutas relativas por las cuales obtienen esas fotos.

Actividad: Premios y Secciones

Por otro lado observamos que:

7. En cuanto al **acceso a la Base de datos** comentar que un cambio en el nombre de usuario y contraseña para acceder a la base de datos tiene que ser actualizado ya que los servicios web requieren estos datos correctos para un buen funcionamiento y acceso a la base de datos.

8. Los **idiomas** de castellano, euskera e inglés están contemplados en la aplicación, sin embargo existen algunos pormenores para el desarrollador:

En primer lugar si quisiéramos modificar traducciones o ampliar el contenido, solo tendríamos que acceder a la carpeta strings del proyecto en eclipse y modificar los valores o añadirlos, en las tres carpetas que existen para cada uno de los idiomas todo con su correspondiente extensión (ej: string-eu, string-es, string-en)

Por otro lado, si quisiéramos incluir otro idioma, sería más complicado ya que algunos de los títulos y string que se ven en la aplicación se adquieren por programa y no automáticamente. El caso es de las noticias que dependiendo de un idioma u otro lee un servicio web u otro.

En el menú de configuración existe un selector de idioma que también habría que contemplar en este caso.

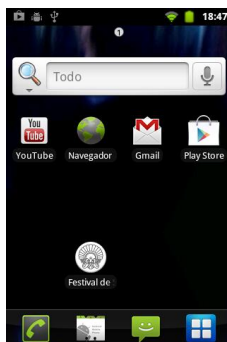
Anexo II. Manual del usuario:

A continuación se muestra una sencilla guía de navegación para el usuario que detalla los movimientos y las interacciones que puede realizar con la aplicación.

Está dividida en los siguientes pasos:

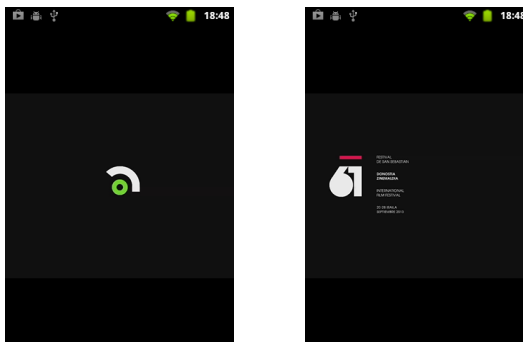
1. Acceso inicial

Tras instalar la aplicación, el usuario dispone de un icono con el logotipo del Festival de cine en el menú de aplicaciones de Android. Tras pulsarlo comienza la aplicación.



2. Cortinilla

La aplicación comienza con un vídeo inicial de cinco segundos que accede directamente al menú principal. Este vídeo se puede eliminar para futuros accesos pero siempre aparecerá en el primer acceso del usuario.



3. Menú principal

Tras el vídeo le sigue el menú principal, está compuesto por tres elementos de navegación: *Slider*, *botones* y *bottom bar*.



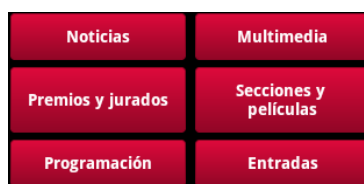
3.1. Slider

Esta funcionalidad muestra cinco imágenes con título de las tres últimas noticias, el último álbum de fotos y el último vídeo introducidos. Cada tres segundos cambia la visualización del elemento y pulsando sobre la vista, accedemos directamente al contenido seleccionado en cuestión sean noticias, fotos o un vídeo



3.2. Botones

Compuesto por seis botones. Las funcionalidades se enmarcan en la vista inferior del menú principal mediante selección de estos botones.



3.2.1. Noticias

Una vez accedemos a noticias disponemos de tres submenús separados por *tabs* o pestañas, cada una de ellas representa un servicio concreto: Noticias del Festival, noticias en general y enlaces a redes sociales. Seleccionando las diferentes pestañas navegaremos entre las posibilidades de las noticias y la prensa del Festival.

Cuando usuario seleccione una de las noticias, se abrirá la noticia a pantalla completa y dispondrá de toda la información.

Las noticias están paginadas, es decir, están divididas en páginas de cinco noticias y evitan un tráfico denso.

3.2.2. Multimedia:

En esta sección encontramos los álbumes de fotos del Festival y una colección de los últimos vídeos subidos por el Festival separados por dos pestañas.

Seleccionando las imágenes, el usuario accederá al álbum concreto q a su vez contiene las fotos pertenecientes a dicho álbum, si selecciona la foto, se abrirá a pantalla completa.

Al seleccionar un vídeo, accederemos directamente a YouTube o al navegador propio del dispositivo y en concreto a la página del vídeo en YouTube

3.2.3. Secciones y películas

El usuario encontrará las principales secciones cinematográficas del Festival de cine y las películas que hay en cada sección.

Una vez que el usuario seleccione una de las secciones, se mostrará a pantalla completa la información de la sección y todas sus películas.

3.2.4. Premios y premiados

Premios Donostia y resto de premios oficiales del Festival de Cine. Dos pestañas dividen cada submenú.

Al seleccionar uno de los premios en cada apartado, se abrirá un informe detallado del premio en cuestión, jurados o información sobre la personalidad premiada (premios Donostia)

3.2.5. Programación

Esta sección dispone de un buscador de la programación en el que el usuario puede introducir con cuatro campos distintos, una búsqueda de los pases de películas.

Lugar, fecha, sección y película, estos son los criterios de búsqueda. Se permiten dejar en blanco.

3.2.6. Entradas

Esta función es un acceso directo a la página de venta de entradas del Festival.

3.3. Bottom bar.

Compuesto por cuatro botones, este servicio está disponible en el menú principal y en los principales submenús de cada funcionalidad y permite al usuario acceder a los servicios más comunes en cualquier momento.



Lo componen cuatro botones con una funcionalidad cada uno:

3.3.1. Home

Pulsando este botón, accedemos al menú principal desde cualquier instancia.

3.3.2. Contacto

Si el usuario pulsa este botón, accederá a un menú con toda la información de contacto relativa al Festival

3.3.3. Mapa

Si el usuario pulsa el botón del Mapa, accederá a una visualización del mapa de San Sebastián con los principales lugares relacionado con el Festival marcados con señales concretas.

3.3.4. Más

Este acceso visualiza una lista de funcionalidades, actualmente el menú de configuración y el menú about:

3.3.4.1. Configuración:

Muestra un menú en el que es posible seleccionar el idioma de la aplicación y eliminar el vídeo inicial para que no vuelva a salir cuando accedamos.

3.3.4.2. About:

Muestra el creador de la aplicación, créditos e información sobre la aplicación.