



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO EN INFORMÁTICA

Título del proyecto:

DESARROLLO DE UNA APLICACIÓN MÓVIL ORIENTADA
A LA GEOLOCALIZACIÓN DE RECURSOS EN ANDROID

Andrés Uriz Labiano

Eduardo Alfaro Larragueta

Pamplona, 15 de Noviembre de 2011

Índice

1.	INTRODUCCIÓN	8
1.1.	Introducción	9
1.2.	Objetivos	9
1.3.	Descripción del problema	10
1.4.	Requisitos	10
1.4.1.	Requisitos funcionales.....	10
1.4.2.	Requisitos de interfaz.....	12
1.4.3.	Requisitos de sistema.....	12
1.4.4.	Requisitos de distribución	12
1.5.	Solución propuesta.....	13
1.6.	Plataforma, herramientas y lenguajes de programación	13
1.6.1.	Plataforma Android	13
1.6.2.	Herramientas.....	15
1.6.3.	Lenguajes de programación	16
1.7.	Metodología de trabajo: Scrum	17
1.7.1.	¿Qué es Scrum?	17
1.7.2.	Principales Actividades.....	17
1.7.3.	Componentes de Scrum	18
1.7.4.	Roles en Scrum	20
1.7.5.	Planificación Scrum para Disfrutando San Fermín	20
2.	ANÁLISIS DEL PROBLEMA	23
2.1.	Diagrama de casos de uso: general.....	24
2.1.1.	Caso de uso: Geolocalización	25
2.1.2.	Caso de uso: Consultar programa	28
2.1.3.	Caso de uso: Consultar eventos activos	33
2.1.4.	Caso de uso: Buscador de eventos.....	38
2.1.5.	Caso de uso: Consultar servicios	40
2.1.6.	Caso de uso: Consultar teléfonos de interés.....	47
2.1.7.	Caso de uso: Consultar recorrido encierro.....	47
2.1.8.	Caso de uso: Consultar twits San Fermín	48
2.1.9.	Caso de uso: Cambiar idioma	49
3.	DISEÑO	50
3.1.	Introducción	51

3.2.	Diagrama de clases.....	52
3.2.1.	Clases de sistema	52
3.2.2.	Clases de interfaz	56
4.	IMPLEMENTACIÓN, PRUEBAS Y DESPLIEGUE.....	57
4.1.	Implementación	58
4.1.1.	Implementación	58
4.1.2.	Metodología Scrum	72
4.2.	Pruebas.....	74
4.3.	Despliegue	77
5.	CONCLUSIONES Y LÍNEAS FUTURAS	78
5.1.	Conclusiones.....	79
5.2.	Líneas Futuras	80
6.	BIBLIOGRAFÍA.....	81
6.1.	Libros y Documentos.....	82
6.2.	Internet.....	82
	ANEXO 1: MANUAL DE USUARIO	83
A1.1.	Descarga e Instalación	84
A1.2.	Pasos Previos.....	85
A1.3.	Funcionalidades	88
A1.3.1.	Menú Configuración.....	88
A1.3.2.	Programa.....	89
A1.3.3.	¿Qué hacer ahora?.....	92
A1.3.4.	Buscador.....	94
A1.3.5.	Servicios	95
A1.3.6.	Teléfonos de interés	97
A1.3.7.	Encierro	98
A1.3.8.	Twitter.....	99
	ANEXO 2: DESARROLLO DE APLICACIONES ANDROID CON ECLIPSE.....	100
A2.1.	Fundamentos de Android	101
A2.1.1.	Componentes de Aplicación	101
A2.1.2.	Lanzamiento de nuevos componentes	102
A2.1.3.	Android Manifest	102
A2.1.4.	Recursos de la aplicación	104
A2.1.5.	Interfaz de usuario	104

A2.2. Android con Eclipse.....	108
A2.2.1. Instalación de Eclipse	108
A2.2.2. Descarga Android SDK.....	108
A2.2.3. Instalar Android Developer Tools (ADT)	108
A2.2.4. Configuración del plugin ADT.....	110
A2.2.5. Creación de Android Virtual Devices (AVDs).....	110
A2.2.6. Creación de un proyecto Android	112

Índice de Figuras

Figura 1. Arquitectura de Android.....	14
Figura 2. Tablón Scrum	19
Figura 3. Diagrama general de Casos de Uso	24
Figura 4. Caso de Uso Geolocalización.....	26
Figura 5. Caso de Uso Consultar Programa	29
Figura 6. Caso de Uso Ficha Evento.....	31
Figura 7. Caso de uso Consultar Eventos Activos.....	34
Figura 8. Caso de uso Buscador de Eventos	38
Figura 9. Caso de Uso Consultar Servicios.....	41
Figura 10. Caso de uso Ficha de Servicio.....	44
Figura 11. Diagrama de clases de sistema	52
Figura 12. Clase eventos.....	52
Figura 13. Clase Categorías	53
Figura 14. Clase Servicios	53
Figura 15. Clase Teléfonos.....	54
Figura 16. Clase SanferminAplication.....	54
Figura 17. Diagrama de clases de interfaz.....	56
Figura 18. Pantalla inicial.....	58
Figura 19. Pantalla GPS	59
Figura 20. Pantalla Geolocalización.....	59
Figura 21. Pantalla seleccionar dirección	60
Figura 22. Pantalla Principal.....	61
Figura 23. Pantalla Seleccionar Día	61
Figura 24. Pantalla Eventos del Día	62
Figura 25. Pantalla Ficha de Evento	62
Figura 26. Pantalla Agenda.....	63
Figura 27. Pantalla StreetView	63
Figura 28. Pantalla Cómo Llegar.....	63
Figura 29. Pantalla Lista Eventos Activos	63
Figura 31. Realidad aumentada eventos activos	64
Figura 30. Pantalla Mapa Eventos Activos	64
Figura 32. Pantalla Buscador Filtro.....	65
Figura 33. Pantalla Buscador Resultados	66
Figura 34. Pantalla Filtro Servicios	66
Figura 35. Pantalla Servicios Lista	67
Figura 36. Pantalla Servicios Mapa	67
Figura 37. Pantalla Ficha de Servicio	68
Figura 38. Pantalla Teléfonos de interés	69
Figura 39. Pantalla Encierro Recorrido.....	69
Figura 40. Pantalla Encierro Consejos.....	70
Figura 41. Pantalla Encierro Prohibiciones.....	70
Figura 42. Pantalla Twitter	70
Figura 43. Pantalla Opciones.....	71

Figura 44. Android Market	77
Figura 45. Pantalla Descarga 1	84
Figura 46. Pantalla Descarga 2	84
Figura 47. Pantalla Descarga 3	85
Figura 48. Manual GPS	85
Figura 49. Manual Geoposicionamiento	86
Figura 50. Manual Buscando posición.....	86
Figura 51. Manual seleccionar posición	87
Figura 52. Manual menú principal	88
Figura 53. Manual Configuración	88
Figura 54. Manual selección día	89
Figura 55. Manual Eventos del día	89
Figura 56. Manual Ficha de Evento	90
Figura 57. Manual Cómo llegar	90
Figura 58. Manual StreetView	91
Figura 59. Manual Agenda	91
Figura 60. Manual Qué hacer ahora lista	92
Figura 61. Manual Qué hacer ahora mapa.....	92
Figura 62. Manual Wikitude	93
Figura 63. Manual Realidad aumentada	93
Figura 64. Manual Buscador.....	94
Figura 65. Manual Buscador Resultados	94
Figura 66. Manual Servicios Selección	95
Figura 67. Manual Servicios Lista	95
Figura 68. Manual Servicios Mapa	96
Figura 69. Manual Ficha de Servicio.....	96
Figura 70. Manual Teléfonos Interés	97
Figura 71. Manual llamar teléfono.....	97
Figura 72. Manual Encierro recorrido	98
Figura 73. Manual Encierro Consejos.....	98
Figura 74. Manual Encierro Prohibiciones	99
Figura 75. Manual Twitter	99
Figura 76. Jerarquía de Views Android.....	106
Figura 77. Descarga ADT 1	109
Figura 78. Descarga ADT 2.....	109
Figura 79. Descarga ADT 3.....	109
Figura 80. Configuración ADT Eclipse.....	110
Figura 81. Configuración AVD	111
Figura 82. Configuración AVD 2	111
Figura 83. Creación Proyecto Android	112

Índice de Tablas

Tabla 1. División en Historias de Usuario	21
Tabla 2. División en Tareas.....	22
Tabla 3. Geolocalización.....	25
Tabla 4. Geolocalización: Posición actual.....	27
Tabla 5. Geolocalización: Utilizar dirección.....	27
Tabla 6. Geolocalización: Utilizar última posición.....	28
Tabla 7. Consultar programa	29
Tabla 8. Ver eventos programa	30
Tabla 9. Consultar ficha evento.....	31
Tabla 10. Cómo llegar.....	32
Tabla 11. StreetView	32
Tabla 12. Añadir a agenda.....	33
Tabla 13. Consultar eventos activos.....	34
Tabla 14. Consultar lista de eventos activos	35
Tabla 15. Consultar mapa de eventos activos.....	36
Tabla 16. Realidad aumentada de eventos activos	37
Tabla 17 Consultar ficha evento.....	37
Tabla 18. Buscador de eventos	38
Tabla 19. Filtrar eventos.....	39
Tabla 20 Consultar ficha evento.....	40
Tabla 21. Consultar servicios.....	40
Tabla 22. Filtrar listado de servicios.....	42
Tabla 23. Mapa de servicios	42
Tabla 24. Realidad aumentada de servicios	43
Tabla 25. Consultar ficha servicio.....	44
Tabla 26 Cómo llegar.....	45
Tabla 27. StreetView	46
Tabla 28. Llamar por teléfono a servicio	46
Tabla 29. Consultar teléfonos de interés	47
Tabla 30. Consultar recorrido encierro	48
Tabla 31. Consultar twits San Fermín.....	48
Tabla 32. Cambiar idioma.....	49
Tabla 33. Comparación Estimado - Real.....	73
Tabla 34. Prueba posición actual	74
Tabla 35 Prueba utilizar dirección sin red	74
Tabla 36. Prueba utilizar dirección no válida	74
Tabla 37. Prueba última posición	75
Tabla 38. Prueba Cómo Llegar	75
Tabla 39. Prueba StreetView	75
Tabla 40. Prueba Eventos Activos	76
Tabla 41. Prueba Realidad aumentada	76

1. INTRODUCCIÓN

En este primer capítulo se realizará una pequeña introducción en la que se mostrarán conceptos generales sobre la empresa en la que se ha realizado el proyecto, motivaciones para su realización, descripción del problema, solución propuesta y herramientas y lenguajes utilizados para su implementación.

1.1. Introducción

Este proyecto se ha realizado en la empresa Alestis Consulting por Andrés Uriz Labiano durante una beca de la fundación Universidad-Sociedad de la Universidad Pública de Navarra.

Alestis Consulting es una empresa que elabora soluciones técnicas innovadoras y de alto valor añadido mediante la integración de tecnologías del ámbito de la telemática. El carácter innovador de la empresa se proyecta a la propia cadena de procesos internos y en el esfuerzo por una mejora continua. El fin, mejorar día a día su capacidad para gestionar proyectos de éxito y obtener la máxima satisfacción de los clientes.

En los últimos tiempos, con los rápidos avances en el campo de los dispositivos móviles tanto a nivel hardware como software, la empresa ha ido incluyendo en su modelo de negocio el desarrollo de aplicaciones móviles, ya que ofrecen grandes posibilidades para el desarrollo de nuevas soluciones muy útiles y fáciles de utilizar que el usuario tendrá siempre a mano.

Por este motivo la empresa centra gran parte de sus esfuerzos en la realización de aplicaciones móviles de calidad tanto para Android como para iPhone, las dos grandes plataformas móviles del mercado en la actualidad.

La motivación para la realización de la aplicación en que se centra este proyecto: **"Disfrutando San Fermín"** es que San Fermín es probablemente la mejor plataforma de publicidad que exista en la comunidad Foral, por lo que se decide no desaprovechar esa gran oportunidad de promoción y de ganar experiencia en el mercado de las aplicaciones móviles.

1.2. Objetivos

El objetivo de este proyecto es la realización de una aplicación para smartphones con sistema operativo Android donde el usuario encontrará toda la información necesaria sobre las fiestas de San Fermín de una manera sencilla y rápida.

En la actualidad en las guías de fiestas entra en juego el factor temporal, es decir, se especifica en qué horario se realizará una actividad, y como mucho se indica el lugar donde se realizará, obligando al usuario a conocer dónde está ese lugar si quiere disfrutar de ese evento. Se observa esto como un gran problema sobre todo en San Fermín, ya que la gran mayoría de los asistentes a la fiesta no conocen la ciudad.

En la actualidad los SmartPhones ofrecen grandes prestaciones en cuanto a la geolocalización del terminal y de recursos a su alrededor, por lo que se piensa en realizar una aplicación que no sólo ofrezca información de cuando se realiza un evento sino que además muestre dónde se realiza, cuáles son los eventos más cercanos y que indique cómo llegar en caso de que se esté interesado en asistir, además de dar otra información de utilidad al usuario.

A su vez, otro de los objetivos de este proyecto es que sea útil a la mayor cantidad de gente posible, por lo que se implementará en inglés y en castellano aprovechando el soporte multidioma de Android.

1.3. Descripción del problema

La aplicación a realizar deberá permitir al usuario la realización de las siguientes acciones:

- Consultar el programa completo de las fiestas de San Fermín 2011.
- Consultar los eventos que están teniendo lugar en este momento.
- Buscar eventos por categorías.
- Acceder a información sobre los servicios que se ofrecen en las fiestas.
- Acceder a los números de teléfono más relevantes y útiles.
- Acceder a toda la información sobre los encierros.
- Mantenerse al tanto de las últimas novedades en Twitter sobre San Fermín.
- Seleccionar el idioma en que desea utilizar la aplicación.

1.4. Requisitos

En este apartado se mostrarán los requisitos que debe cumplir la aplicación y mediante los cuales se ha realizado la división en historias de usuario, y a partir de ahí, el análisis, diseño e implementación del software. En primer lugar se explicarán los requisitos funcionales, y posteriormente los de interfaz y de sistema.

1.4.1. Requisitos funcionales

- La aplicación deberá ser bilingüe, el software se adaptará automáticamente al idioma en el que el usuario tenga configurado su terminal, castellano si su teléfono está en castellano o inglés en otro caso, dando la posibilidad al usuario de cambiarlo de manera sencilla.
- El usuario podrá elegir entre tres formas de localización para resolver la posición que se empleará en la aplicación:

- 1) Utilizar la posición actual.
 - 2) Utilizar una dirección postal.
 - 3) Utilizar la última posición empleada en la aplicación.
- La aplicación deberá presentar el programa completo de las fiestas de San Fermín 2011, desde el cual se podrá acceder a la ficha de información de cada evento.
- Para cada evento, además de mostrar la información asociada al mismo (nombre, lugar, día, horario y descripción), se tendrá la opción de elegir:
 - 1) Cómo llegar desde la posición seleccionada al iniciar la aplicación.
 - 2) Ver la localización concreta de evento en StreetView.
 - 3) Añadir el evento a la agenda del teléfono.
- Los eventos estarán agrupados por categoría.
- El sistema deberá ofrecer información de los eventos que actualmente están activos y de los que van a tener lugar próximamente. Dichos eventos deberán mostrarse en función de la posición seleccionada al iniciar la aplicación en tres modos:
 - 1) Vista de lista de eventos ordenados por hora de inicio.
 - 2) Vista de mapa donde se indicará la posición del usuario y los eventos se marcarán con el ícono de la categoría correspondiente.
 - 3) Vista de realidad aumentada, donde a través de la cámara del dispositivo se visualizará la posición real de los eventos.
- Se dispondrá de un buscador de eventos para poder localizar los eventos que se deseé con mayor eficacia filtrando por categoría.
- Se podrá acceder a la información y localización de los servicios que ofrece la ciudad para San Fermín que puedan ser de utilidad para el usuario. Tras seleccionar los servicios que se desea mostrar, podrán visualizarse los mismos tres modos que los eventos activos.
- Se tendrá acceso a los números de teléfono de mayor interés para las fiestas, a los que se podrá llamar desde la misma aplicación.
- Se ofrecerá información sobre el encierro: recorrido, consejos y prohibiciones.
- Se podrá acceder a la información de más actualidad sobre San Fermín a través de las redes sociales.
- La aplicación deberá seguir las pautas de navegación de la plataforma Android: botón de volver, botón de menú...

1.4.2. Requisitos de interfaz

- En la aplicación deberán prevalecer los colores característicos de las fiestas de San Fermín, el blanco y el rojo.
- Todas las pantallas de la aplicación deberán seguir una misma línea estética.
- En la pantalla principal se tendrá acceso a todas las secciones de la aplicación de manera sencilla y con un solo clic.
- Los iconos de las categorías deberán seguir una misma línea, cambiará el color y el dibujo interior.
- Los botones de la aplicación deberán ser del estilo Android.
- El icono general de la aplicación deberá representar a la misma.
- La aplicación deberá visualizarse correctamente en cualquier pantalla de un dispositivo con versión de Android 2.1 o superior.

1.4.3. Requisitos de sistema

- La aplicación deberá ser soportada por dispositivos con versión de Android 2.1 o superior, tratando a su vez que funcione en la medida de lo posible en versiones anteriores del sistema operativo.
- Se deberá minimizar en la medida de lo posible la dependencia de la aplicación con el acceso a internet.
- La aplicación no deberá seguir corriendo en background consumiendo recursos una vez se finalice.
- Se deberá optimizar al máximo el proceso de localización del usuario en caso de que éste seleccione “utilizar posición actual”.

1.4.4. Requisitos de distribución

- La aplicación deberá estar disponible en el Android Market de manera gratuita antes del día 1 de Julio de 2011.

1.5. Solución propuesta

La solución que se ha propuesto para que este proyecto cumpla los requisitos descritos en el apartado anterior, consiste en la realización de una aplicación para Android en la que:

- Nada más entrar pide que se seleccione la posición eligiendo unas de las 3 opciones descritas en los requisitos.
- Tiene una pantalla principal donde, a través de iconos claramente visibles colocados en forma de cuadrícula, se puede acceder a las 7 secciones principales de la aplicación: Programa, ¿Qué hacer ahora?, Buscador, Servicios, Teléfonos de interés, Encierro y Twitter.
- Se configurará automáticamente el idioma según el idioma en que se tenga configurado el teléfono. Éste se podrá configurar también desde la pantalla principal.
- La información estática de la aplicación (eventos, servicios, teléfonos y categorías), se almacenará en ficheros XML internos para minimizar el acceso a la red.
- Esta información se cargará en la memoria de la aplicación en el inicio de la aplicación, teniéndola así accesible siempre que se necesite sin necesidad de tener que recargarla.
- Cada sección de la aplicación se implementará siguiendo las pautas de los requisitos.
- Se cuidará la estética para que dé imagen de uniformidad y se identifique claramente con las fiestas de San Fermín.

1.6. Plataforma, herramientas y lenguajes de programación

1.6.1. Plataforma Android

Android es un sistema operativo basado en GNU/Linux diseñado originalmente para dispositivos móviles, pero que posteriormente ha ido introduciéndose en otros dispositivos.

Fue implementado por Android Inc. y posteriormente adquirido por el gigante Google en 2005. Actualmente, Android se apoya en una gran comunidad de desarrolladores que realizan aplicaciones para mejorar la funcionalidad de los Smartphones. A la fecha, se han sobrepasado las 500.000 aplicaciones disponibles para la tienda de aplicaciones oficial de Android: Android Market.

La base del sistema operativo Android se compone de aplicaciones que se ejecutan en un framework Java de aplicaciones orientadas a objetos sobre el núcleo de las bibliotecas de Java en una máquina virtual Dalvik con compilación en tiempo de ejecución.

Android, contrariamente a otros sistemas operativos móviles como iOS o Windows Phone, se desarrolla de forma abierta y se puede acceder tanto al código fuente como al listado de incidencias donde se pueden ver problemas aún no resueltos y reportar problemas nuevos.

La arquitectura del sistema operativo es la siguiente:

- **Aplicaciones:** Todas las aplicaciones están escritas en lenguaje de programación Java.
- **Marco de trabajo de aplicaciones:** los programadores tienen total acceso a los mismos APIs del framework utilizados por las aplicaciones base. La arquitectura está diseñada para simplificar la reutilización de componentes.
- **Librerías:** Android incluye un conjunto de bibliotecas de C/C++ empleadas por varios componentes del sistema y que los desarrolladores podrá utilizar en sus aplicaciones.
- **Runtime de Android:** Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik.
- **Kernel Linux:** Android utiliza Linux para los servicios básicos de seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores, a su vez, también se ocupa de la coordinación entre el hardware y software.

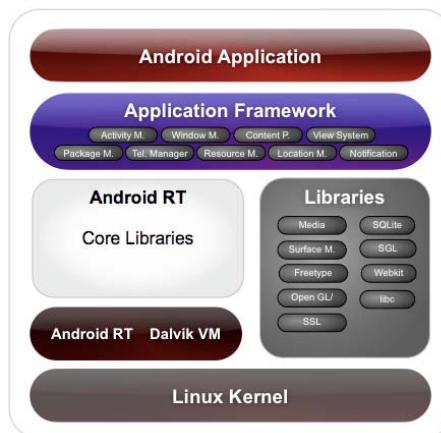


Figura 1. Arquitectura de Android

1.6.2. Herramientas

Para la realización de este proyecto, ha sido necesario: el entorno de desarrollo Eclipse con el plugin ADT (Android Developer Tools) para la programación de la aplicación y Adobe Photoshop CS5 para el diseño de las interfaces.

1.6.2.1. Eclipse IDE + ADT plugin + SDK android

Para la implementación de la aplicación se va a emplear el Framework Eclipse, al que se le configurará el plugin ADT para poder trabajar con aplicaciones de Android, con el cual, Eclipse se convierte en una herramienta muy eficaz para el desarrollo de aplicaciones para Android.

Android Development Tools (ADT) es un plugin para Eclipse IDE que se ha diseñado para proporcionar un entorno poderoso e integrado para el desarrollo de aplicaciones Android.

ADT extiende las funcionalidades de Eclipse para ayudar en el desarrollo de aplicaciones Android, permitiendo crear interfaces de usuario de manera sencilla, añadir componentes basados en el API de Android, depurar las aplicaciones utilizando el SDK de Android o exportar ficheros .apk, firmados o sin firmar, para distribuir tu aplicación.

Para el desarrollo de una aplicación móvil en Android se emplea una arquitectura multicapa, cada una de las capas una funcionalidad concreta y bien definida.

- **Capa interfaz:** La capa de interfaz o de presentación de la aplicación está compuesta por las vistas definidas en ficheros XML. Es muy importante en la aplicación, ya que es la que se encarga de la presentación de la aplicación al usuario.
- **Capa de modelo de negocio:** Esta capa es la encargada de manejar los datos enviados por la capa de interfaz, es decir, en función de los eventos recibidos por la capa de presentación deberá tomar decisiones de modificar el modelo y mostrar los resultados. Aquí se implementa toda la lógica de negocio. En la aplicación, esta capa se implementa en Java.
- **Capa de datos:** Es la encargada de acceder a los datos según reciba las solicitudes de la capa anterior. En el caso concreto de esta aplicación se cargarán los datos desde el inicio en unas clases globales que serán accesibles en cualquier contexto.

Para obtener más información acerca de la plataforma Android y cómo programar sobre ella en el entorno de desarrollo Eclipse en el **ANEXO 2: DESARROLLO DE APLICACIONES ANDROID CON ECLIPSE** se presenta de manera mucho más detallada este tema.

1.6.2.2. *Adobe Photoshop CS5*

Para el diseño de las interfaces de usuario de la aplicación será de gran ayuda esta herramienta ya que es muy potente y fácil de utilizar. Se empleará para el diseño de iconos, cabeceras, botones y demás elementos estéticos de la aplicación.

1.6.3. Lenguajes de programación

Los lenguajes de programación que se van a utilizar para el desarrollo de este proyecto son: Java y XML. A continuación se introducirá brevemente cada uno:

1.6.3.1. *Java*

Java es un lenguaje de programación orientado a objetos que permite realizar todo tipo de aplicaciones, el cual es de uso libre y gratuito, y está enfocado a cubrir las necesidades tecnológicas más punteras. Actualmente es propiedad de la empresa Oracle, la cual adquirió a Sun Microsystems.

Una de las principales características por las que Java se ha hecho muy famoso es que es un lenguaje independiente de la plataforma, dado que es un lenguaje interpretado, diseñado para ser ejecutado por medio de un intérprete, en contraste con los lenguajes de compilado, lo que posibilita que un código interpretado pueda ser ejecutado en cualquier otro tipo de sistema operativo, si se cuenta con un intérprete codificado para ese sistema operativo.

Las aplicaciones para Android se escriben en este lenguaje, con la única diferencia que al SDK de Java se añaden las funciones propias del Android SDK que le permiten adaptarse y controlar dispositivos con sistema operativo Android.

1.6.3.2. *XML*

El lenguaje XML estará muy presente en el desarrollo de la aplicación, ya que se utilizará tanto para el diseño de las interfaces de usuario de Android, como para el almacenamiento de los datos estáticos de la aplicación.

No será necesaria ninguna herramienta adicional para el trabajo con los XML ya que Eclipse también da soporte en el trabajo con este formato de ficheros.

1.7. Metodología de trabajo: Scrum

En la empresa en la que se ha desarrollado este proyecto se utiliza la metodología de trabajo Scrum, por lo que este proyecto también ha seguido esta metodología. En este apartado se darán unas nociones básicas en las que se pretenderá presentar el funcionamiento de esta metodología.

1.7.1. ¿Qué es Scrum?

Scrum es una metodología ágil de desarrollo de software, no es un proceso detallado, sólo provee una serie de pautas a seguir a partir de las que cada equipo de desarrollo deberá sacar su propia manera de trabajar.

Un equipo de trabajo que utiliza Scrum se caracteriza por ser auto-gestionable y multifuncional. Es auto-gestionado ya que no hay un líder que decide qué tarea realiza cada uno y cómo la resolverá, esto se decidirá por el equipo. Este equipo debe ser a su vez multifuncional, ya que cada miembro debe estar preparado para realizar cualquiera de las tareas.

Estos equipos de desarrollo ágil se apoyan en dos roles específicos: el **Scrum Master** y el **Product Owner**. El Scrum Master es el que organiza al equipo y cada día toma nota de los avances o problemas en la hoja Scrum. El Product Owner representa al cliente y es el que establecerá los requisitos y evaluará el producto final.

Los proyectos Scrum se organizan en **Sprints**, que son iteraciones acotadas en el tiempo de que no deberían durar más de un mes. Al inicio del sprint los miembros del equipo se reúnen a deliberar qué desarrollos se incluirán en el **Product backlog**. Y al final del sprint se evaluará que estos desarrollos cumplan con la funcionalidad definida por el product owner, en la reunión de revisión. En esta reunión además se realizarán otros procesos que servirán de “feedback” para siguientes sprints.

1.7.2. Principales Actividades

El **sprint** en sí es la principal actividad de un proyecto Scrum, ya que éste se define como un proceso iterativo e incremental de series consecutivas de sprints. Normalmente cada sprint se separa del siguiente más o menos una semana, esta semana servirá para evaluar el sprint anterior y preparar el siguiente. Durante este sprint, el equipo de desarrollo hace todo lo posible para llevar a cabo los desarrollos que se le han encomendado.

La tarea inicial del sprint es la **reunión de planificación**. En ella el Product Owner y el equipo discuten de las tareas más prioritarias del product backlog. Los miembros del equipo

estiman cuantos items pueden acometer para crear el product backlog, que define la lista de historias de usuario que se realizarán en el sprint y sus niveles de prioridad.

Cada día del sprint el equipo de trabajo y el Scrum master, también miembro del equipo, se reunirán en el **daily Scrum**. En esta reunión, no más larga de 15 minutos, los miembros del equipo compartirán sus trabajos en el día anterior y lo que van a hacer en ese día, así como posibles impedimentos a su progreso si los hubiera. Estas reuniones servirán para sincronizar el trabajo del equipo mientras ponen en común conocimientos y problemas para buscar posibles soluciones.

Al final del sprint, se lleva a cabo la llamada **reunión de revisión**. En ella, el equipo muestra las funcionalidades desarrolladas durante el sprint al product owner para comprobar que se cumplen todos los requisitos establecidos en la reunión de planificación y tomar medidas para el siguiente sprint.

Otra actividad que se realizará al final de cada sprint es la **reunión de retrospectiva**. En ella, todos los miembros del equipo y el product owner, expresan sus opiniones acerca del sprint que acaba de terminar en búsqueda de tomar medidas de mejora para el siguiente sprint.

1.7.3. Componentes de Scrum

El principal componente de un proyecto Scrum son los propios desarrollos. Se espera que el equipo implemente unos desarrollos que cumplan los requisitos establecidos al comienzo del sprint.

El **product backlog** es la lista completa de funcionalidades a desarrollar. Los items del product backlog son priorizados por el product owner para que el equipo las vaya desarrollando en ese orden. La manera más utilizada y eficiente de llenar un product backlog es con **historias de usuario**, que son pequeñas descripciones de las funcionalidades que se desea conseguir desde el punto de vista del consumidor, a la que se le añaden unos criterios de validación que representan lo que debe cumplir el producto para que el desarrollo sea válido.

El primer día del sprint, en la reunión de planificación, los miembros del equipo junto con el product owner dividirán en tareas lo más pequeñas posibles cada historia de usuario de forma que puedan ser estimadas por separado de una manera más eficaz. Una vez se tenga estimada cada historia de usuario y atendiendo al tiempo de duración del sprint y las prioridades de las historias de usuario, se creará el **sprint backlog**, que contendrá la lista priorizada de las historias de usuario a realizar en ese sprint.

Otros componente muy importante de un proyecto Scrum es el **burndown chart**. Se trata de una gráfica en la que se muestra la evolución del trabajo restante para un sprint a lo largo de los días. Cada día el Scrum master, en la daily Scrum actualizará el tiempo restante para cada historia de usuario del sprint según lo que le digan los miembros del equipo que

queda de las tareas que habían estado realizando. Estas gráficas son muy útiles a la hora de determinar cómo va el sprint, si por encima o por debajo de lo planificado e intentar tomar medidas correctivas lo antes posible.

Por último y no menos importante estaría el **tablón Scrum**. Se trata de un corcho o similar que se construye al principio del sprint en el que se plasmarían las tareas del sprint agrupadas por historias de usuario. Cada tarea se representa por una tarjeta donde estará el nombre de cada tarea.

Estas tareas pueden encontrarse en cuatro estados representados como cuatro columnas en el tablón: tareas sin comenzar, tareas en ejecución, tareas en testing y tareas finalizadas. De esta manera, cada vez que un miembro del equipo comience una tarea le pondrá su nombre y la moverá por las columnas según su estado. Así, las tareas irán moviéndose hacia la derecha según vayan terminándose. Un tablón Scrum representa de una manera muy gráfica el estado del sprint ya que las tareas van viajando de izquierda a derecha y se puede observar de una manera muy clara si se va bien o mal en dicho sprint.

Story	To Do		In Process	To Verify	Done
As a user, I... 8 points	Code the... 9	Test the... 8	Code the... DC 4	Test the... SC 6	Code the... D Test the... SC 8 Test the... SC Test the... SC 6
As a user, I... 5 points	Code the... 2	Code the... 8	Test the... SC 8		Test the... SC Test the... SC Test the... SC 6

Figura 2. Tablón Scrum

1.7.4. Roles en Scrum

El primer rol que se va a analizar es el del **Scrum Master**. El Scrum master es el coordinador del equipo de trabajo que tratará de que cada miembro desarrolle su máximo nivel en el sprint. El Scrum Master difiere del gestor del proyecto tradicional, ya que no provee seguimiento día a día ni asigna tareas a los miembros del equipo, es un miembro más del equipo, que trabaja como los demás, pero que a sus tareas como miembro del equipo debe sumar tareas de coordinación del equipo y organización del sprint. Este rol puede incluso variar entre sprints según el equipo de trabajo.

Mientras que el Scrum Master se centra en que el equipo dé lo máximo, el **product owner** trabaja para dirigir al equipo en la dirección correcta y cumplir los objetivos del sprint. El product owner es el encargado de crear la visión del producto final así como los requisitos que debe cumplir a través del product backlog.

El tercer y último rol en un proyecto Scrum es el **equipo** en sí. Aunque los miembros del equipo sean muy heterogéneos en cuanto a experiencia, títulos..., mientras estén en el equipo son todos iguales. Cada persona contribuye de la mejor manera posible para completar el trabajo al finalizar el sprint. Del mismo modo, el equipo deberá ser lo suficientemente inteligente para que cada miembro del equipo realice las tareas que mejor se adapten a sus conocimientos y cualidades.

Una metáfora muy útil para plasmar lo que es un proyecto Scrum es la de un coche de carreras, en ella, el equipo sería el propio coche, preparado para rodar a todo gas en cualquier dirección que se le dirija, el product owner sería el conductor, el encargado de dirigir al coche en la dirección correcta, y el Scrum master sería el mecánico, manteniendo el coche a punto para realizar un buen trabajo.

1.7.5. Planificación Scrum para Disfrutando San Fermín

Para el proyecto “Disfrutando San Fermín”, como en todos los proyectos que se desarrollan en la empresa Alestis Consulting se ha utilizado como metodología de trabajo la metodología ágil Scrum.

La forma de aplicar la metodología Scrum en la empresa es la siguiente:

- Los sprints tienen una duración aproximada de un mes.
- El grupo de trabajo lo componen cinco personas.
- Los sprints pueden estar formados por historias de usuario de varios proyectos.
- La primera historia de usuario de un proyecto, siempre que se considere necesario, es la de realizar el análisis y diseño del mismo de una manera ágil,

en este caso el análisis y diseño realizado se plasma en los capítulos de análisis y diseño de este documento.

- Por lo demás se sigue la metodología de trabajo explicada en el capítulo de introducción.

El primer paso para introducir un proyecto nuevo en un sprint Scrum es dividirlo en historias de usuario, que son pequeñas descripciones de las funcionalidades que se desea conseguir desde el punto de vista del cliente.

Dentro de la planificación Scrum llevada a cabo en Alestis Consulting, y dependiendo del proyecto del que se trate, se estudia la posibilidad de reservar una historia de usuario para el análisis y el diseño del proyecto. En este caso se ha creído necesaria, por lo que se ha incluido en el Team backlog. Este es el listado de historias de usuario en las que se divide el proyecto “Disfrutando San Fermín”:

Story ID	Proyecto	Historia de Usuario	Prioridad
1	Disfrutando San Fermín	Análisis y Diseño	100
2	Disfrutando San Fermín	Diseño interfaces	100
3	Disfrutando San Fermín	Geolocalización	100
4	Disfrutando San Fermín	Pantalla carga y principal	100
5	Disfrutando San Fermín	Programa	90
6	Disfrutando San Fermín	¿Qué hacer ahora?	90
7	Disfrutando San Fermín	Buscador	90
8	Disfrutando San Fermín	Servicios	80
9	Disfrutando San Fermín	Teléfonos de Interés	80
10	Disfrutando San Fermín	Encierro	80
11	Disfrutando San Fermín	Twitter	80
12	Disfrutando San Fermín	Análisis plataforma Multi-idioma	70
13	Disfrutando San Fermín	Introducción datos	70
14	Disfrutando San Fermín	Adaptación todas pantallas	70

Tabla 1. División en Historias de Usuario

Una vez definidas las historias de usuario por el product owner, el siguiente paso es dividirlas en tareas más pequeñas de forma que sea más fácil estimarlas. Así pues, en la reunión de planificación del día 4 de Mayo de 2011, el equipo de trabajo junto con el product owner, definieron el listado de tareas del proyecto a partir de las historias de usuario así como sus estimaciones iniciales.

Hay que decir que este proyecto no tenía una prioridad alta, antes habría que realizar otras actividades de otros proyectos. El proyecto se comenzaría el día 23 de Mayo de 2011.

Story ID	Task ID	Prioridad	Descripción	Estimación Inicial
1	1	100	Análisis y Diseño	1,5
2	2	100	Diseño pantallas	2
2	3	100	Diseño iconos	1
3	4	100	Geolocalización: Posición Actual	2
3	5	100	Geolocalización: Buscar Dirección	0,5
3	6	100	Geolocalización: Última posición	0,5
4	7	100	Pantalla de carga	1
4	8	100	Pantalla inicial	1
5	9	90	Programa: Listado días	0,5
5	10	90	Programa: Tabs eventos	1
5	11	90	Ficha Evento + Cómo llegar + StreetView + Agenda	2
6	12	90	Cálculo eventos activos y próximos	1
6	13	90	¿Qué hacer ahora?:Vista modo lista	0,5
6	14	90	¿Qué hacer ahora?:Vista modo mapa	2
6	15	90	¿Qué hacer ahora?:Vista modo realidad aumentada	2
6	16	90	¿Qué hacer ahora?:Mostrar ficha evento	0,5
7	17	90	Buscador: filtro por clase	1
7	18	90	Buscador: mostrar resultados	0,5
8	19	80	Servicios: Filtro por tipo de servicio	0,5
8	20	80	Servicios: resultados modo lista	1
8	21	80	Servicios: resultados modo mapa	0,5
8	22	80	Servicios: resultados realidad aumentada	1
8	23	80	Servicios: Ficha Servicio	0,5
9	24	80	Sección teléfonos de interés	1
10	25	80	Encierro: recorrido sobre mapa con fotos	1,5
10	26	80	Encierro: recomendaciones y prohibiciones	0,5
11	27	80	Análisis e implementación sección Twitter	1
12	28	70	Desarrollo plataforma multi idioma	1
12	29	70	Réplica textos	0,5
13	30	70	Introducción de datos	4
14	31	70	Adaptación a todas pantallas	2
14	32	70	Adaptación a todas las resoluciones	2

Tabla 2. División en Tareas

En los siguientes capítulos se realizará el análisis y diseño de este proyecto, actividad correspondiente con la primera tarea del proyecto.

2. ANÁLISIS DEL PROBLEMA

Una vez presentado el problema de manera rápida en el capítulo anterior, en este segundo capítulo se pasará a analizarlo más en profundidad. Para ello se presentará el problema en forma de diagrama de casos de uso donde se esquematizarán todos los retos que se presentan.

2.1. Diagrama de casos de uso: general

Una vez definidos los requisitos en el apartado de introducción, se van a plasmar estos requisitos en los siguientes casos de uso, y de ellos se va a realizar una tabla con más detalle, dividiendo en subdiagramas cuando sea necesario.

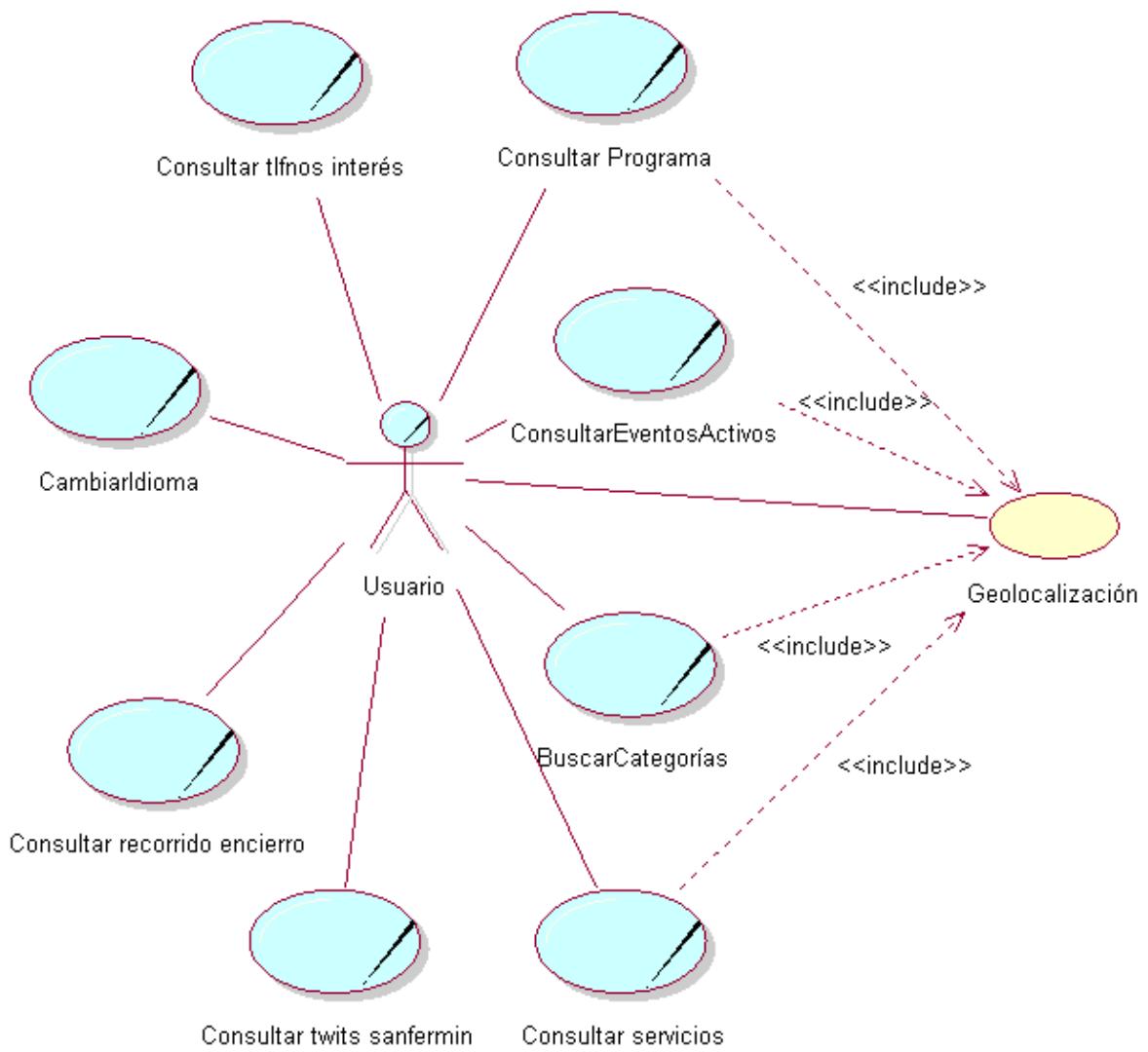


Figura 3. Diagrama general de Casos de Uso

2.1.1. Caso de uso: Geolocalización

Nombre:	Geolocalización.
Descripción:	Asocia al usuario una posición, según tres posibles opciones, para hacer uso de los datos georreferenciados de la aplicación.
Actores:	Usuario.
Precondición:	El dispositivo deberá hacer uso de la red o del GPS para la localización. Si no dispone de ninguna de estas funcionalidades la localización no será posible.
Flujo normal:	<ol style="list-style-type: none">1. El usuario selecciona las opciones deseadas para su localización (posición actual, por dirección o última posición).2. La aplicación posiciona al usuario según la opción elegida.
Flujo alternativo:	<ol style="list-style-type: none">1. Si no se selecciona ninguna de las opciones o se produce un error de localización el dispositivo no queda localizado y alguna de las funcionalidades de la aplicación no estarán completamente operativas.
Poscondiciones:	La posición del usuario queda registrada en el sistema.

Tabla 3. Geolocalización

Este caso de uso se desglosará a continuación:

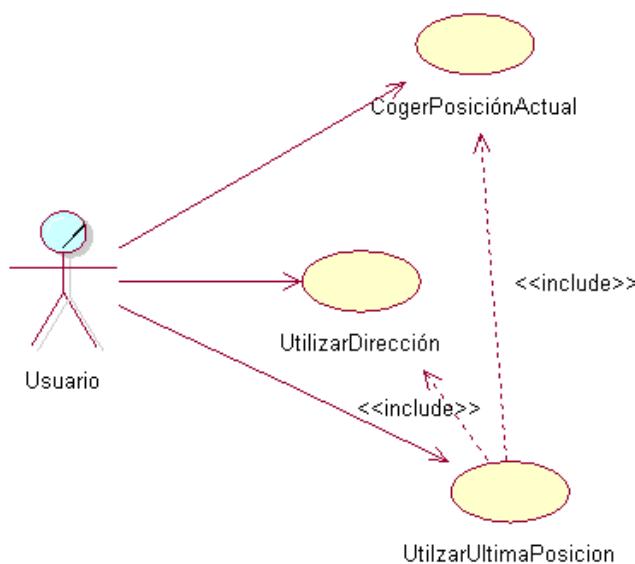


Figura 4. Caso de Uso Geolocalización

2.1.1.1. Caso de uso: *Coger posición actual*

Nombre:	Geolocalización: Coger posición actual.
Descripción:	
Se detecta la posición actual del usuario mediante la red de datos, wifi o Gps, y se registra la posición en el sistema.	
Actores:	
Usuario.	
Precondición:	
El dispositivo deberá hacer uso de la red o del GPS para la localización. Si no dispone de ninguna de estas funcionalidades la localización no será posible.	
Flujo normal:	
<ol style="list-style-type: none"> 1. El usuario selecciona la opción de utilizar la posición actual. 2. Si hay conexión de datos se coge una posición aproximada del dispositivo a través de la red de datos. 3. Se trata de buscar la posición GPS, más precisa, del dispositivo sin ralentizar la aplicación, utilizando mientras tanto la posición aproximada. 4. La posición queda registrada, ya sea por el primero o el segundo método. 	

Flujo alternativo:

1. Si no se dispone de red de datos ni GPS, será imposible localizar al dispositivo.

Poscondiciones:

La posición del dispositivo queda registrada en el sistema y es correcta.

Tabla 4. Geolocalización: Posición actual

2.1.1.2. Caso de uso: Utilizar dirección

Nombre:	Geolocalización: utilizar dirección.
Descripción:	Se registrará como posición a utilizar en el sistema la de una dirección postal introducida por el usuario, que se traducirá a coordenadas con ayuda de Google Geocoder.
Actores:	Usuario.
Precondición:	El dispositivo deberá hacer uso de la red, si no dispone de conexión a red la traducción de la dirección no será posible.
Flujo normal:	<ol style="list-style-type: none">1. El usuario selecciona la opción de utilizar dirección postal.2. El sistema hace una consulta a Google Geocoder para traducirla a coordenadas.3. La posición queda registrada.
Flujo alternativo:	<ol style="list-style-type: none">1. La posición no se puede traducir porque no existe.2. La posición no se puede traducir porque no hay conexión de red.
Poscondiciones:	La posición de la dirección buscada queda registrada en el sistema y es la correcta.

Tabla 5. Geolocalización: Utilizar dirección

2.1.1.3. Caso de uso: Utilizar última posición

Nombre:	Geolocalización: Utilizar última posición.
Descripción:	Se registrará como posición a utilizar en el sistema la última posición utilizada en la aplicación.
Actores:	Usuario.
Precondición:	El usuario deberá haber utilizado alguna posición anteriormente.
Flujo normal:	<ol style="list-style-type: none">1. El usuario selecciona la opción de utilizar última posición.2. Se busca en el sistema la última posición utilizada.3. La posición queda registrada.
Flujo alternativo:	<ol style="list-style-type: none">1. No hay ninguna posición anterior, se muestra un mensaje de error.
Poscondiciones:	La posición queda registrada en el sistema y es la correcta.

Tabla 6. Geolocalización: Utilizar última posición

2.1.2. Caso de uso: Consultar programa

Nombre:	Consultar programa.
Descripción:	Permite al usuario consultar la información de los eventos de San Fermín 2011, agrupados por días y por horario, pudiendo acceder a la ficha de cada evento por separado.
Actores:	Usuario.
Precondición:	Si no se ha recogido la posición del usuario habrá opciones no operativas.

Flujo normal:

1. El usuario selecciona el ícono de programa desde la pantalla principal.
2. El usuario selecciona el día del que desea ver los eventos programados.
3. El usuario selecciona el evento deseado moviéndose por la lista o por las pestañas de mañana, tarde y noche para ver la información del mismo.

Flujo alternativo:**Poscondiciones:**

Se mostrará la ficha de información del evento.

Tabla 7. Consultar programa

Este caso de uso se detallará a continuación:

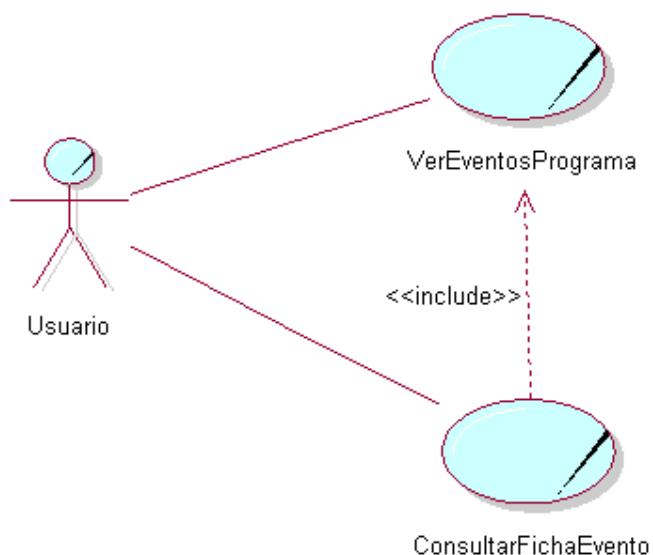


Figura 5. Caso de Uso Consultar Programa

2.1.2.1. Caso de uso: Ver eventos programa

Nombre:	Ver eventos programa.
Descripción:	Permite al usuario consultar los eventos de San Fermín 2011, agrupados por días y por horario.
Actores:	Usuario.
Precondición:	
Flujo normal:	<ol style="list-style-type: none">1. El usuario selecciona el ícono de programa desde la pantalla principal.2. El usuario selecciona el día del que desea ver los eventos programados.3. El usuario selecciona el evento deseado moviéndose por la lista o por las pestañas de mañana, tarde y noche para ver la información del mismo.
Flujo alternativo:	
Poscondiciones:	Se podrán ver los eventos que hay en el programa de manera intuitiva.

Tabla 8. Ver eventos programa

2.1.2.2. Caso de uso: Consultar ficha evento

Nombre:	Consultar ficha evento.
Descripción:	El usuario encontrará la información del evento: nombre, lugar, día, hora y descripción, además de poder obtener información de cómo llegar, una vista real (StreetView) de la localización del evento y añadirlo a la agenda de su dispositivo.
Actores:	Usuario.
Precondición:	El usuario ha seleccionado un evento en cualquiera de los casos de uso anteriores.

Flujo normal:

1. El usuario llega a la ficha del evento a través de cualquiera de los casos de usos anteriores: Consultar Programa, Consultar eventos activos o Buscador.
2. Se muestra la información del evento además de poder acceder a la información de cómo llegar, una vista real (StreetView) de la localización del evento y añadirlo a la agenda de su dispositivo.

Flujo alternativo:**Poscondiciones:**

El usuario visualiza correctamente la ficha con la información del evento.

Tabla 9. Consultar ficha evento

Este caso de uso se detallará a continuación:

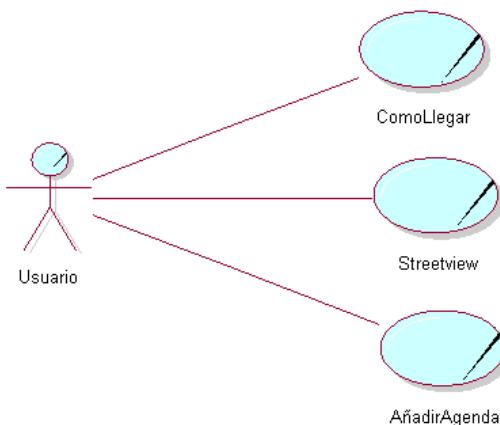


Figura 6. Caso de Uso Ficha Evento

2.1.2.2.1 Caso de uso: Cómo llegar

Nombre:	Cómo llegar.
Descripción:	
Permite al usuario obtener indicaciones para llegar a la posición donde está localizado el evento a partir de la posición que ha registrado en el sistema. Estas indicaciones serán proporcionadas por Google Maps.	
Actores:	
	Usuario.

Precondición:
Existe una posición registrada y hay conexión a la red.
Flujo normal:
<ol style="list-style-type: none"> 1. El usuario selecciona el botón de cómo llegar en la ficha del evento. 2. Aparece un mapa de Google con las indicaciones para ir a pie al evento seleccionado.
Flujo alternativo:
<ol style="list-style-type: none"> 1. Si no hay posición registrada o no se tiene acceso a la red, no será posible obtener indicaciones.
Poscondiciones:
Se muestra al usuario las indicaciones para llegar al evento deseado.

Tabla 10. Cómo llegar

2.1.2.2.2. Caso de uso: StreetView

Nombre:	StreetView.
Descripción:	
Permite al usuario obtener una visión fotográfica de 360º(StreetView) de la ubicación del evento para hacer más fácil su localización, para ello se hará uso de la utilidad StreetView de Google.	
Actores:	
Usuario.	
Precondición:	
El dispositivo tiene acceso a la red.	
Flujo normal:	
<ol style="list-style-type: none"> 1. El usuario selecciona el botón StreetView en la ficha del evento. 2. Se muestra la localización en 3 dimensiones de la localización del evento. 	
Flujo alternativo:	
<ol style="list-style-type: none"> 1. Si el dispositivo no tiene acceso a la red no se obtendrá ningún resultado. 	
Poscondiciones:	
Se obtendrá la vista fotográfica de 360º de la localización del evento de manera satisfactoria.	

Tabla 11. StreetView

2.1.2.2.3. Caso de uso: Añadir a agenda

Nombre:	Añadir a agenda.
Descripción:	Permite al usuario introducir en la agenda de su dispositivo los datos del evento deseado para que se lo recuerde más adelante.
Actores:	Usuario.
Precondición:	
Flujo normal:	<ol style="list-style-type: none">1. El usuario selecciona el botón Agenda de la ficha del evento.2. El evento se añade a la agenda del dispositivo.
Flujo alternativo:	
Poscondiciones:	El evento se añade a la agenda del dispositivo de manera satisfactoria.

Tabla 12. Añadir a agenda

2.1.3. Caso de uso: Consultar eventos activos

Nombre:	Consultar eventos activos.
Descripción:	Permite al usuario consultar los eventos que se están desarrollando en ese instante, así como los que van a comenzar en las próximas 3 horas. Estos eventos se pueden mostrar de tres maneras: En forma de lista, en vista de mapa o en realidad aumentada.
Actores:	Usuario.
Precondición:	Debe estar registrada la posición del usuario y debe ser San Fermín.

Flujo normal:

1. El usuario selecciona el ícono de ¿Qué hacer ahora? desde la pantalla principal.
2. Se obtiene una lista de eventos ordenados cronológicamente y divididos en eventos activos y eventos próximos.
3. El usuario podrá ver los eventos activos en las tres formas de presentación.

Flujo alternativo:

1. No es San Fermín por lo que no se muestra ningún recurso.

Poscondiciones:

Se observarán correctamente los eventos activos en los tres modos de visualización.

Tabla 13. Consultar eventos activos

Este caso de uso se detallará a continuación:

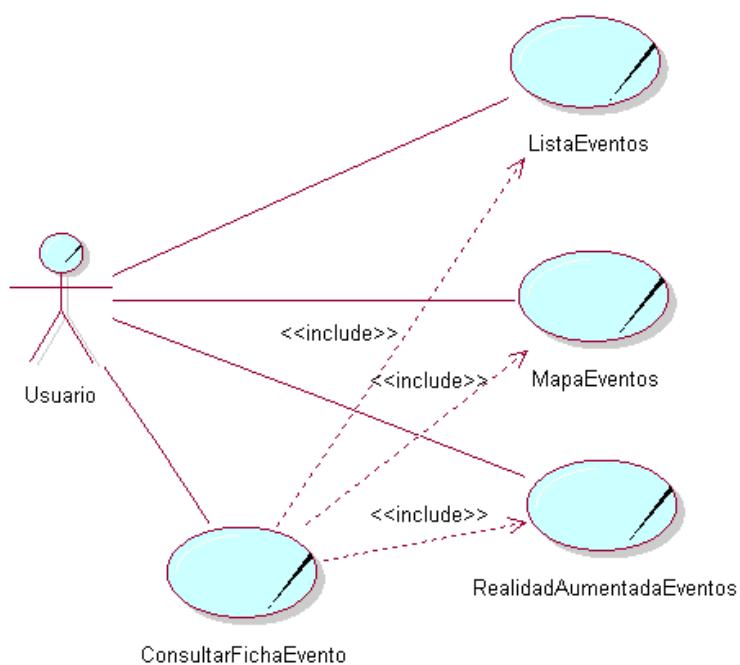


Figura 7. Caso de uso Consultar Eventos Activos

2.1.3.1. Caso de uso: Consultar lista de eventos activos

Nombre:	Consultar lista de eventos activos.
Descripción:	Permite al usuario consultar el listado de eventos que se están desarrollando en ese instante, así como los que van a comenzar en las próximas 3 horas. Si seleccionamos un evento, se accederá a la ficha de dicho evento.
Actores:	Usuario.
Precondición:	Debe estar registrada la posición del usuario y debe ser San Fermín.
Flujo normal:	<ol style="list-style-type: none"> El usuario selecciona el ícono de ¿Qué hacer ahora? desde la pantalla principal. Se obtiene una lista de eventos ordenados cronológicamente y divididos en eventos activos y eventos próximos.
Flujo alternativo:	<ol style="list-style-type: none"> No es San Fermín por lo que no se muestra ningún recurso.
Poscondiciones:	Se observarán correctamente los eventos activos y los próximos en la lista.

Tabla 14. Consultar lista de eventos activos

2.1.3.2. Caso de uso: Consultar mapa de eventos activos

Nombre:	Consultar mapa de eventos activos.
Descripción:	Permite al usuario observar la posición de los eventos activos respecto de la posición registrada. La posición registrada se presentará mediante un muñeco vestido de San Fermín y los demás evento mediante el ícono correspondiente. Si se selecciona un recurso se accederá a la ficha del evento correspondiente.
Actores:	Usuario.

Precondición:
Existe una posición registrada y hay conexión a red.
Flujo normal:
<ol style="list-style-type: none"> 1. El usuario accede a la vista de mapa ya sea desde el menú de eventos activos. 2. El usuario obtiene los eventos pintados en el mapa además de su posición. 3. El usuario selecciona un recurso para ir a la ficha del evento.
Flujo alternativo:
<ol style="list-style-type: none"> 1. Si no hay conexión a red o la posición no ha sido registrada los resultados no serán válidos.
Poscondiciones:
Se visualiza correctamente el mapa y los eventos sobre él.

Tabla 15. Consultar mapa de eventos activos

2.1.3.3. Caso de uso: Realidad aumentada de eventos activos

Nombre:	Realidad aumentada de eventos activos.
Descripción:	
Permite al usuario el visualizar la localización de los eventos activos a su alrededor desde una perspectiva real, marcando la posición de los mismos a través de la cámara del teléfono. Si se selecciona un evento se accederá a la ficha correspondiente.	
Actores:	
Usuario.	
Precondición:	
Hay conexión a la red y se encuentra instalada la aplicación Wikitude en el dispositivo.	
Flujo normal:	
<ol style="list-style-type: none"> 1. El usuario accede a la vista de Realidad Aumentada desde el menú de eventos activos. 2. Se activará la cámara del dispositivo y el usuario podrá observar la localización de los eventos activos a través de ella. 3. Si se selecciona un evento se accederá a la ficha correspondiente. 	
Flujo alternativo:	
<ol style="list-style-type: none"> 1. Si no se encuentra instalada la aplicación Wikitude en el dispositivo, se sugerirá al usuario que se la descargue de manera gratuita desde el Android Market para disfrutar de esta funcionalidad. 	

Poscondiciones:

El usuario visualiza correctamente los recursos a través de su cámara.

Tabla 16. Realidad aumentada de eventos activos

2.1.3.4. Caso de uso: Consultar ficha evento

Nombre:	Consultar ficha evento.
Descripción:	
El usuario encontrará la información del evento: nombre, lugar, día, hora y descripción, además de poder obtener información de cómo llegar, una vista real (StreetView) de la localización del evento y añadirlo a la agenda de su dispositivo.	
Actores:	
Usuario.	
Precondición:	
El usuario ha seleccionado un evento en cualquiera de los casos de uso anteriores.	
Flujo normal:	
<ol style="list-style-type: none">3. El usuario llega a la ficha del evento a través de cualquiera de los casos de usos anteriores: Consultar Programa, Consultar eventos activos o Buscador.4. Se muestra la información del evento además de poder acceder a la información de cómo llegar, una vista real (StreetView) de la localización del evento y añadirlo a la agenda de su dispositivo.	
Flujo alternativo:	
Poscondiciones:	
El usuario visualiza correctamente la ficha con la información del evento.	

Tabla 17 Consultar ficha evento

2.1.4. Caso de uso: Buscador de eventos

Nombre:	Buscador de eventos.
Descripción:	Permite al usuario filtrar los eventos por categoría para encontrar de una manera más fácil lo que busca. El resultado será una lista de eventos ordenados cronológicamente. Si se selecciona un evento de la lista nos mostrará la información del mismo.
Actores:	Usuario.
Precondición:	Si no se ha recogido la posición del usuario habrá opciones no operativas.
Flujo normal:	<ol style="list-style-type: none"> 1. El usuario selecciona el ícono de Buscador desde la pantalla principal. 2. El usuario marca las categorías de las que interesa obtener resultados. 3. El usuario obtiene una lista de eventos ordenados cronológicamente, si selecciona uno de ellos irá a la información de dicho evento.
Flujo alternativo:	
Poscondiciones:	Se mostrará la ficha de información del evento.

Tabla 18. Buscador de eventos

Este caso de uso se detallará a continuación:

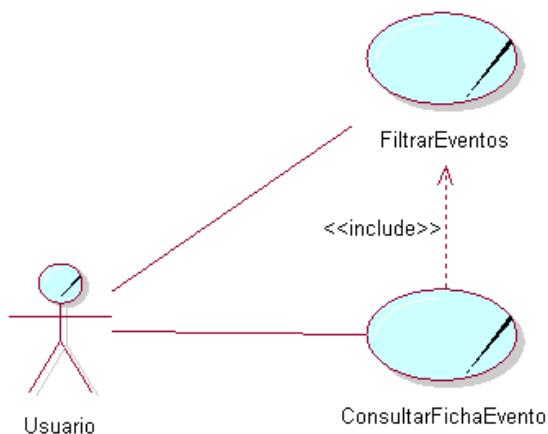


Figura 8. Caso de uso Buscador de Eventos

2.1.4.1. Caso de uso: Filtrar eventos

Nombre:	Filtrar eventos.
Descripción:	
Permite al usuario filtrar los eventos por categoría para encontrar de una manera más fácil lo que busca. El resultado será una lista de eventos ordenados cronológicamente.	
Actores:	
Usuario.	
Precondición:	
Si no se ha recogido la posición del usuario habrá opciones no operativas.	
Flujo normal:	
1. El usuario selecciona el ícono de Buscador desde la pantalla principal. 2. El usuario marca las categorías de las que interesa obtener resultados. 3. El usuario obtiene una lista de eventos ordenados cronológicamente.	
Flujo alternativo:	
Poscondiciones:	

Tabla 19. Filtrar eventos

2.1.4.2. Caso de uso: Consultar ficha evento

Nombre:	Consultar ficha evento.
Descripción:	
El usuario encontrará la información del evento: nombre, lugar, día, hora y descripción, además de poder obtener información de cómo llegar, una vista real (StreetView) de la localización del evento y añadirlo a la agenda de su dispositivo.	
Actores:	
Usuario.	
Precondición:	
El usuario ha seleccionado un evento en cualquiera de los casos de uso anteriores.	

Flujo normal:

1. El usuario llega a la ficha del evento a través de cualquiera de los casos de usos anteriores: Consultar Programa, Consultar eventos activos o Buscador.
2. Se muestra la información del evento además de poder acceder a la información de cómo llegar, una vista real (StreetView) de la localización del evento y añadirlo a la agenda de su dispositivo.

Flujo alternativo:**Poscondiciones:**

El usuario visualiza correctamente la ficha con la información del evento.

Tabla 20 Consultar ficha evento

2.1.5. Caso de uso: Consultar servicios

Nombre:	Consultar servicios.
Descripción:	
Permite al usuario acceder a la información de los servicios de las fiestas de San Fermín, pudiendo filtrar los que desea y obteniendo los resultados georeferenciados en tres formas de presentación: lista, mapa o realidad aumentada.	
Actores:	
Usuario.	
Precondición:	
Debe estar registrada la posición del usuario.	
Flujo normal:	
<ol style="list-style-type: none">1. El usuario selecciona el ícono de Servicios desde la pantalla principal.2. El usuario marca los servicios de los que desea obtener resultados.3. El usuario obtiene los resultados en tres formas de presentación: lista, mapa o realidad aumentada.	
Flujo alternativo:	
1. Si no hay posición registrada no se obtendrán datos válidos.	
Poscondiciones:	
Se observarán correctamente los servicios en los tres modos de visualización.	

Tabla 21. Consultar servicios

Este caso de uso se detallará a continuación:

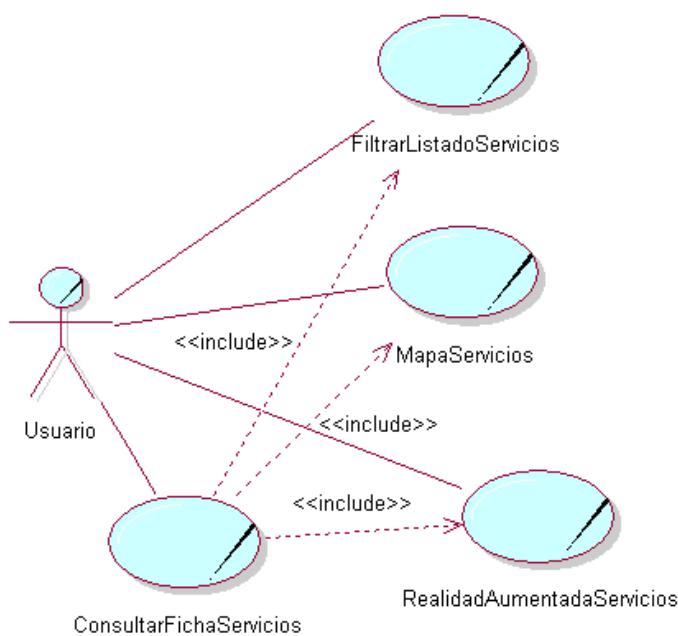


Figura 9. Caso de Uso Consultar Servicios

2.1.5.1. Caso de uso: *Filtrar listado de servicios*

Nombre:	Filtrar listado de servicios.
Descripción:	
Permite al usuario filtrar los servicios por categoría para encontrar de una manera más fácil lo que busca. El resultado será una lista de servicios de esas categorías.	
Actores:	
Usuario.	
Precondición:	
Si no se ha recogido la posición del usuario habrá opciones no operativas.	
Flujo normal:	
<ol style="list-style-type: none"> 1. El usuario selecciona el ícono de Servicios desde la pantalla principal. 2. El usuario marca las categorías servicios de las que interesa obtener resultados. 3. El usuario obtiene una lista de servicios. 	

Flujo alternativo:
Poscondiciones:
Se obtendrá un listado con los servicios ordenados por distancia a la posición registrada.

Tabla 22. Filtrar listado de servicios

2.1.5.2. Caso de uso: Mapa de servicios

Nombre:	Mapa de servicios.
Descripción:	Permite al usuario observar la posición de los servicios respecto de la posición registrada. La posición registrada se presentará mediante un muñeco vestido de San Fermín y los demás servicios mediante el icono correspondiente. Si se selecciona un recurso se accederá a la ficha del servicio correspondiente.
Actores:	Usuario.
Precondición:	Existe una posición registrada y hay conexión a red.
Flujo normal:	<ol style="list-style-type: none"> 1. El usuario accede a la vista de mapa desde el menú de servicios. 2. El usuario obtiene los servicios pintados en el mapa además de su posición. 3. El usuario selecciona un recurso para ir a la ficha correspondiente.
Flujo alternativo:	<ol style="list-style-type: none"> 1. Si no hay conexión a red o la posición no ha sido registrada los resultados no serán válidos.
Poscondiciones:	Se visualiza correctamente el mapa y los recursos sobre él.

Tabla 23. Mapa de servicios

2.1.5.3. Caso de uso: Realidad aumentada de servicios

Nombre:	Realidad aumentada de servicios.
Descripción:	Permite al usuario el visualizar la localización de los servicios a su alrededor desde una perspectiva real, marcando la posición de los mismos a través de la cámara del teléfono. Si se selecciona un recurso se accederá a la ficha correspondiente.
Actores:	Usuario.
Precondición:	Hay conexión a la red y se encuentra instalada la aplicación Wikitude en el dispositivo.
Flujo normal:	<ol style="list-style-type: none">1. El usuario accede a la vista de Realidad Aumentada desde el menú de servicios.2. Se activará la cámara del dispositivo y el usuario podrá observar la localización de los recursos a través de ella.3. Si se selecciona un recurso se accederá a la ficha correspondiente.
Flujo alternativo:	<ol style="list-style-type: none">1. Si no se encuentra instalada la aplicación Wikitude en el dispositivo, se sugerirá al usuario que se la descargue de manera gratuita desde el Android Market.
Poscondiciones:	El usuario visualiza correctamente los recursos a través de su cámara.

Tabla 24. Realidad aumentada de servicios

2.1.5.4. Caso de uso: Consultar ficha servicio

Nombre:	Consultar ficha servicio.
Descripción:	El usuario encontrará la información del servicio: nombre, teléfono y descripción, además de poder obtener información de cómo llegar, una vista real (StreetView) de la localización del servicio y llamar por teléfono si se conoce el número.

Actores:
Usuario.
Precondición:
El usuario ha seleccionado un servicio en cualquiera de los casos de uso anteriores.
Flujo normal:
<ol style="list-style-type: none"> 1. El usuario llega a la ficha del servicio a través de cualquiera de las formas de visualización del resultado del filtrado de los servicios. 2. Se muestra la información del servicio además de poder acceder a la información de cómo llegar, una vista real (StreetView) de la localización del servicio y llamar por teléfono si se conoce el número.
Flujo alternativo:
Poscondiciones:
El usuario visualiza correctamente la ficha con la información del servicio.

Tabla 25. Consultar ficha servicio

Este caso de uso se detallará a continuación:

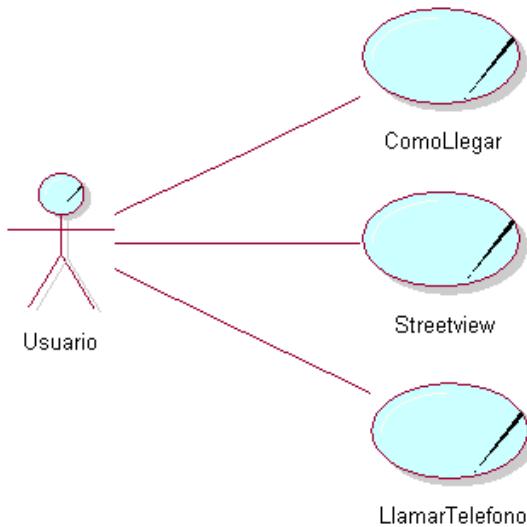


Figura 10. Caso de uso Ficha de Servicio

2.1.5.4.1. Caso de uso: Cómo llegar

Nombre:	Cómo llegar.
Descripción:	Permite al usuario obtener indicaciones para llegar a la posición donde está localizado el servicio a partir de la posición que ha registrado en el sistema. Estas indicaciones serán proporcionadas por Google Maps.
Actores:	Usuario.
Precondición:	Existe una posición registrada y hay conexión a la red.
Flujo normal:	<ol style="list-style-type: none"> 1. El usuario selecciona el botón de cómo llegar en la ficha del servicio. 2. Aparece un mapa de Google con las indicaciones para ir a pie al evento seleccionado.
Flujo alternativo:	<ol style="list-style-type: none"> 1. Si no hay posición registrada o no se tiene acceso a la red, no será posible obtener indicaciones.
Poscondiciones:	Se muestra al usuario las indicaciones para llegar al servicio deseado.

Tabla 26 Cómo llegar

2.1.5.4.2. Caso de uso: StreetView

Nombre:	StreetView.
Descripción:	Permite al usuario obtener una visión fotográfica de 360º (StreetView) de la ubicación del servicio para hacer más fácil su localización, para ello se hará uso de la utilidad StreetView de Google.
Actores:	Usuario.
Precondición:	El dispositivo tiene acceso a la red.

Flujo normal:
1. El usuario selecciona el botón StreetView en la ficha del servicio.
2. Se muestra la localización en 3 dimensiones de la localización del servicio.
Flujo alternativo:
1. Si el dispositivo no tiene acceso a la red no se obtendrá ningún resultado.
Poscondiciones:
Se obtendrá la vista fotográfica de 360º de la localización del servicio de manera satisfactoria.

Tabla 27. StreetView

2.1.5.4.3. Caso de uso: Llamar por teléfono a servicio

Nombre:	Llamar por teléfono a servicio.
Descripción:	
Permite al usuario llamar por teléfono al servicio seleccionado si se dispone de su número de teléfono.	
Actores:	
Usuario.	
Precondición:	
El dispositivo debe poder realizar llamadas telefónicas.	
Flujo normal:	
1. El usuario selecciona el botón Llamar de la ficha del servicio. 2. El teléfono comienza una llamada a dicho número.	
Flujo alternativo:	
Poscondiciones:	
Se realiza la llamada satisfactoriamente.	

Tabla 28. Llamar por teléfono a servicio

2.1.6. Caso de uso: Consultar teléfonos de interés

Nombre:	Consultar teléfonos de interés.
Descripción:	El usuario tendrá acceso a un listado de números de teléfono que le pueden ser muy útiles en las fiestas de San Fermín. Podrá llamar directamente si lo necesita sin necesidad de salir de la aplicación.
Actores:	Usuario.
Precondición:	
Flujo normal:	<ol style="list-style-type: none">1. El usuario selecciona el ícono de Teléfonos de interés desde la pantalla principal.2. Se obtiene un listado de números de teléfono de interés.
Flujo alternativo:	
Poscondiciones:	El listado de teléfonos de interés se visualiza correctamente.

Tabla 29. Consultar teléfonos de interés

2.1.7. Caso de uso: Consultar recorrido encierro

Nombre:	Consultar recorrido encierro.
Descripción:	El usuario obtiene una representación del recorrido del encierro sobre Google maps, en la que se intercalan imágenes representativas de cada tramo. Además el usuario podrá acceder a la pantalla “Consejos útiles para correr el encierro” y “No se puede hacer en el encierro...”.
Actores:	Usuario.
Precondición:	Debe tener conexión a red.

Flujo normal:

1. El usuario selecciona el ícono de Encierro desde la pantalla principal.
2. Se obtiene la vista del recorrido con las imágenes.
3. Hay dos botones para acceder a consejos y prohibiciones para el encierro.
4. Si se selecciona uno de ellos se accede a una pantalla en modo texto en cada sección.

Flujo alternativo:

1. Si no hay conexión a red, el mapa de Google no se cargará.

Poscondiciones:

Se visualizan correctamente todas las pantallas de la sección.

Tabla 30. Consultar recorrido encierro

2.1.8. Caso de uso: Consultar twits San Fermín

Nombre:	Consultar twits San Fermín.
Descripción:	
El usuario accederá a una vista de los twits más importantes de San Fermín.	
Actores:	
Usuario.	
Precondición:	
El usuario debe tener conexión a la red.	
Flujo normal:	
<ol style="list-style-type: none">1. El usuario selecciona la opción de Twitter en la pantalla principal.2. Se obtiene una lista de twits.	
Flujo alternativo:	
<ol style="list-style-type: none">1. Si no dispone de red no será posible obtener este listado.	
Poscondiciones:	
La pantalla con los twits se carga de manera satisfactoria.	

Tabla 31. Consultar twits San Fermín

2.1.9. Caso de uso: Cambiar idioma

Nombre:	Cambiar idioma.
Descripción:	El usuario podrá seleccionar el idioma de la aplicación desde la pantalla principal, pudiendo elegir entre inglés y castellano.
Actores:	Usuario.
Precondición:	
Flujo normal:	<ol style="list-style-type: none">1. El usuario selecciona la opción de configuración en la pantalla principal.2. El usuario selecciona entre inglés y castellano.3. La aplicación se recarga en el idioma deseado.
Flujo alternativo:	
Poscondiciones:	La aplicación se carga con el idioma deseado por el usuario.

Tabla 32. Cambiar idioma

3. DISEÑO

En este capítulo, se pasará a diseñar tanto las estructuras de los datos que se van a utilizar en la aplicación, como las clases en las que se plasmarán los casos de uso desarrollados en el capítulo anterior.

3.1. Introducción

Como bien se ha definido anteriormente en el apartado de herramientas (1.6.2), la arquitectura que se utiliza para el desarrollo de una aplicación móvil en Android es multicapa, cada una de las capas una funcionalidad concreta y bien definida:

- **Capa interfaz:** La capa de interfaz o de presentación de la aplicación está compuesta por las vistas definidas en ficheros XML. Es muy importante en la aplicación, ya que es la que se encarga de la presentación de la aplicación al usuario.
- **Capa de modelo de negocio:** Esta capa es la encargada de manejar los datos enviados por la capa de interfaz, es decir, en función de los eventos recibidos por la capa de presentación deberá tomar decisiones de modificar el modelo y mostrar los resultados. Aquí se implementa toda la lógica de negocio. En la aplicación, esta capa se implementa en Java. Estas clases se denominarán a partir de ahora como clases de interfaz.
- **Capa de datos:** Es la encargada de acceder a los datos según reciba las solicitudes de la capa anterior. En el caso concreto de esta aplicación se cargarán los datos desde el inicio en unas clases globales que serán accesibles en cualquier contexto. Estas clases se denominarán a partir de ahora como Clases de sistema.

En esta aplicación, como prácticamente todas las funcionalidades necesitan de la intervención del usuario para que se inicien, el grueso de las funcionalidades estarán en las clases de interfaz, dejando las clases de sistema prácticamente como un mero almacén de los datos necesarios de la aplicación, es decir, las clases de interfaz se encargarán además de gestionar las interfaces y sus eventos, de mostrar y guardar datos desde las clases de sistema.

En este apartado se van a definir más en profundidad las clases de sistema y se realizará un pequeño esbozo de la estructura de las clases de interfaz. Ya que será en el siguiente capítulo, **Implementación**, donde se profundizará en ellas así como en los XMLs que definirán las vistas de cada pantalla.

3.2. Diagrama de clases

3.2.1. Clases de sistema

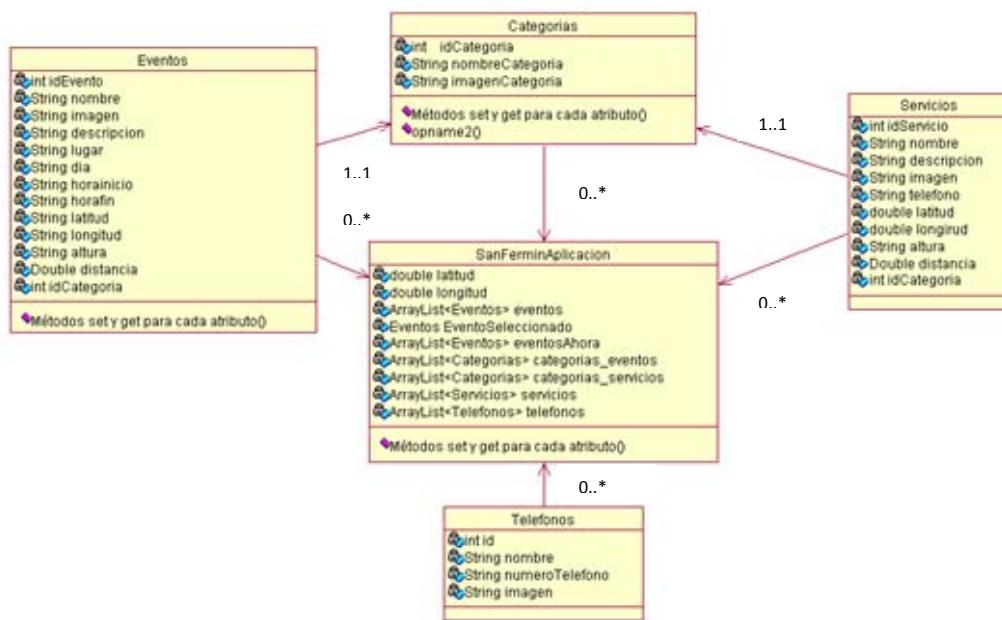


Figura 11. Diagrama de clases de sistema

En este diagrama se muestra la estructura de las clases de sistema de esta aplicación. En los siguientes apartados se presentará cada una por separado.

3.2.1.1. Clase Eventos

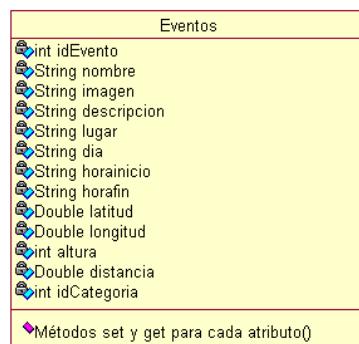


Figura 12. Clase eventos

Esta clase representa a un evento de San Fermín. Además de almacenar los datos informativos sobre el evento: nombre, imagen, lugar, día, hora..., también guarda datos de su georreferenciación mediante la latitud, la longitud y la altitud. Se introduce el atributo idCategoría para relacionarlo con una categoría, clase que se explicará a continuación.

Los métodos de esta clase son los get() y set() para cada uno de los atributos, de forma que puedan ser accedidos o modificados desde clases externas de la misma aplicación.

3.2.1.2. Clase Categorías

Categorias
• int idCategoria
• String nombreCategoria
• String imagenCategoria
♦ Métodos set y get para cada atributo()

Figura 13. Clase Categorías

Esta clase representa una categoría de eventos o de servicios. Guarda los datos del idCategoria, nombreCategoria e imagenCategoria, que será el ícono que utilizarán los objetos de esa categoría en la aplicación.

Al igual que la clase anterior, los métodos de esta clase son los get() y set() para cada uno de los atributos.

3.2.1.3. Clase Servicios

Servicios
• int idServicio
• String nombre
• String descripcion
• String imagen
• String telefono
• double latitud
• double longitud
• String altura
• Double distancia
• int idCategoria
♦ Métodos get() y set() para cada atributo()

Figura 14. Clase Servicios

Esta clase representa a un servicio que se ofrecerá en San Fermín. Al igual que la clase eventos, almacenará información del servicio: nombre, descripción, teléfono..., así como

información para la georreferenciación del servicio. Se le asociará una categoría introduciendo también el idCategoría.

En cuanto a los métodos de esta clase, no serán necesarios más que los métodos get() y set() para cada atributo.

3.2.1.4. Clase Teléfonos

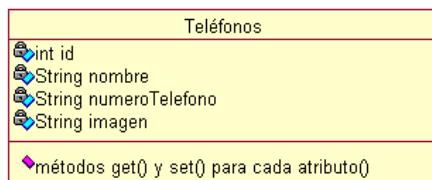


Figura 15. Clase Teléfonos

Esta clase representa a un teléfono de interés para San Fermín. Se almacenará del mismo su nombre, el número de teléfono y el ícono que se utilizará en la aplicación. En cuanto a los métodos, no será necesario ningún método adicional a los get() y set() para cada atributo.

3.2.1.5. Clase SanferminAplicacion



Figura 16. Clase SanferminAplicacion

Esta clase es una clase especial en un proyecto Android. Es la clase de aplicación, debe extender del objeto Application, y en ella se guardarán las variables que se quiere que sean globales y accesibles para todas las clases de la aplicación, estos datos formarán parte de la memoria de la aplicación y permanecerán cargadas en memoria hasta que la misma sea finalizada.

Por todo esto se almacenarán en esta clase todos los datos que se vayan a utilizar en varias pantallas:

- La latitud y la longitud representan la posición que se utilizará como posición del usuario en la aplicación. Se guardará una vez y se podrá utilizar en cualquier pantalla.
- El ArrayList de eventos representa al listado completo de objetos de la clase Eventos, todos los eventos de San Fermín, que servirá de base para todas las secciones relacionadas con eventos.
- El ArrayList de categorías de eventos representa el listado de posibles categorías de eventos.
- El ArrayList de categorías de servicios representa el listado de posibles categorías de servicios.
- El ArrayList de eventos ahora representa los eventos que se están desarrollando actualmente o que comenzarán en las próximas 3 horas. Es un filtrado de la lista completa de eventos, pero se almacena para solo tener que calcularlo una vez.
- El ArrayList de servicios es una lista de objetos de la clase Servicios que representa al listado completo de servicios que se ofrecen en SanFermin. Se cargarán al comenzar la aplicación.
- El ArrayList de servicios seleccionados es una lista de objetos de la clase Servicios que representa al listado de servicios obtenidos al realizar un filtrado por categoría en la sección servicios. Se guardan en memoria para no tener que filtrar de nuevo al cambiar entre las formas de visualización lista, mapa y realidad aumentada de la sección servicios.
- El ArrayList de teléfonos de interés. Es una lista de objetos de la clase Teléfonos que representan a los teléfonos de interés que se utilizarán en la sección teléfonos de interés. También se cargarán al iniciar la aplicación.

3.2.2. Clases de interfaz

Las clases de interfaz de un proyecto Android, extenderán del objeto Activity de Android y serán las encargadas de gestionar la interfaz y los eventos que sobre ella se produzcan.

3.2.2.1. Diagrama de clases

En este diagrama se mostrará la secuencia de las clases de interfaz en la aplicación:

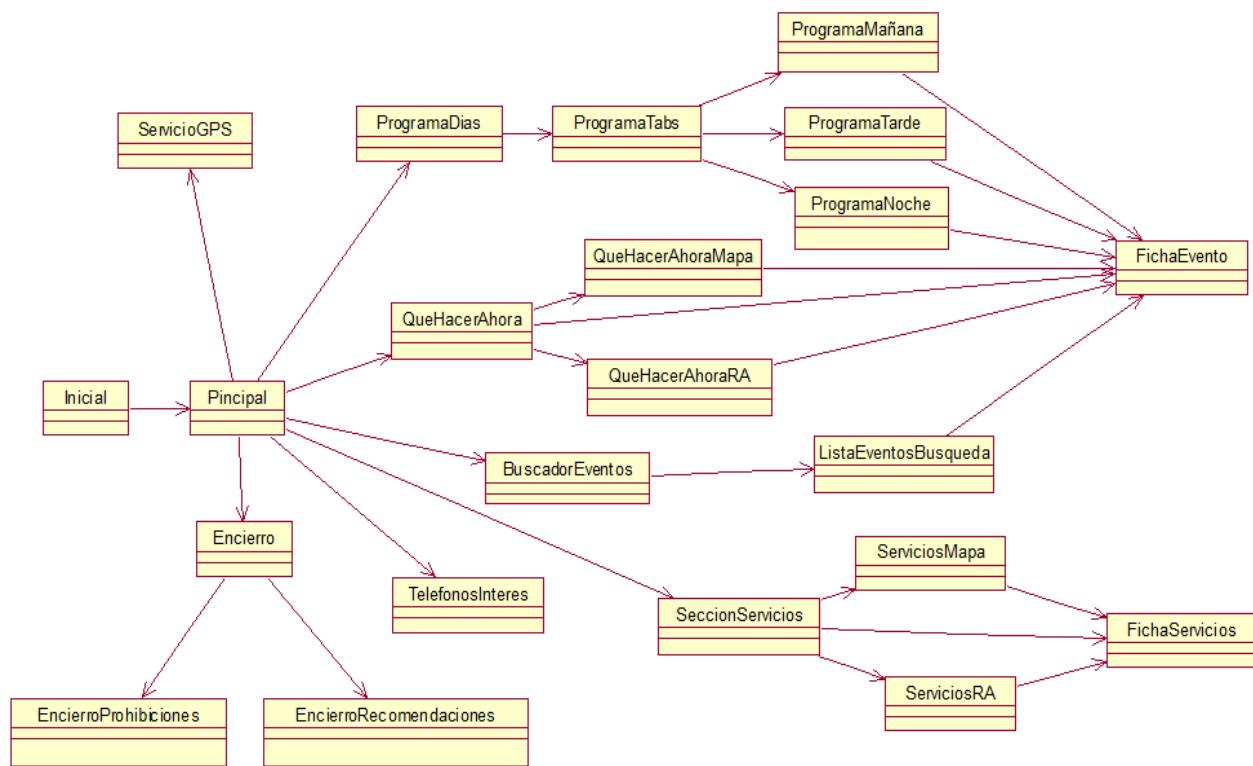


Figura 17. Diagrama de clases de interfaz

Debido a que son clases especiales, que cada una controla una pantalla y que se van a explicar en el próximo capítulo no se ve necesario realizar un desarrollo de las mismas en este capítulo.

4. IMPLEMENTACIÓN, PRUEBAS Y DESPLIEGUE

Una vez realizado el diseño de las clases y del modelo de datos se va a pasar en este capítulo a explicar cómo se ha implementado dicho diseño. Para ello se realizará una explicación detallada de cada pantalla de la aplicación, su interfaz y funcionalidad.

4.1. Implementación

Una vez realizado el diseño de las clases y del modelo de datos se va a pasar en este capítulo a explicar cómo se ha implementado dicho diseño, tanto técnica como metodológicamente. En este caso, como se trata de un proyecto realizado en empresa, no se mostrará código, sino que se explicará de manera detallada qué se hace en cada una de las pantallas.

4.1.1. Implementación

Debido a que en Android la implementación de la interfaz de usuario y de la pantalla se realiza por separado, para realizar esta explicación también se expondrán por separado.

4.1.1.1. Pantalla inicial de carga



Interfaz: En este caso la implementación de la interfaz es muy sencilla, sólo se trata de un ImageView a toda pantalla donde se mostrará el cartel de San Fermín 2011.

Funcionalidad: En esta pantalla de introducción se aprovecha para, con la excusa de introducir la aplicación, coger los datos de las diferentes secciones (Eventos, categorías de eventos, Servicios, Categorías de Servicios y Teléfonos de interés) de los XML, y guardarlos en las variables reservadas para estos datos en la clase aplicación, para poder utilizarlas posteriormente sin tener necesidad de cargarlas cuando se necesiten.

Figura 18. Pantalla inicial

4.1.1.2. Pantalla de Geoposicionamiento

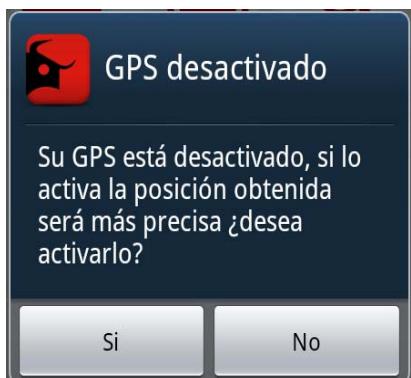


Figura 19. Pantalla GPS

Interfaz: No será necesario realizar ninguna interfaz para esta pantalla, ya que será un AlertDialog que se generará por código.

Funcionalidad: Lo primero que se comprueba en esta pantalla es si se tiene activado el GPS, ya que la posición obtenida mediante el GPS será más precisa, y en caso negativo se muestra un AlertDialog preguntando al usuario a ver si lo quiere activar. En caso afirmativo se lanza una petición al sistema (Intent), para que lo active.



Figura 20. Pantalla Geolocalización

El siguiente paso es mostrar el cuadro de dialogo con las tres opciones posibles de localización. Para ello se crea un AlertDialog con dichas opciones y se les asocia un evento onClick a cada una para hacer algo diferente en cada caso.

- Utilizar posición actual

En caso de elegir la opción de utilizar la posición actual, lo primero que realiza el sistema es comprobar qué medios de localización tiene, habiendo 3 posibilidades:

- 1) Si sólo dispone de conexión a red, ya sea por 3G o por WIFI, es decir, no tiene activado el GPS, intenta localizar al usuario por la conexión de red. En este caso la posición obtenida será poco precisa pero será un proceso rápido. Se pondrá un límite de 20 segundos intentando localizar para no bloquear la aplicación.
- 2) Si sólo dispone de GPS, no dispone de conexión a red, se intentará localizarlo por GPS. La posición obtenida será muy precisa, pero el proceso es más lento, sobre todo si es la primera vez que se usa el GPS en un tiempo. Se pondrá un límite de espera de 60 segundos para conseguir localizarlo.
- 3) Si tiene ambas formas de localización, red y GPS, se tratará de coger la posición por red, proceso mucho más rápido, y se dejará un Servicio Android, proceso que corre en background, escuchando posiciones GPS. Cuando le llegue la posición por GPS modificará la obtenida por red, ya que esta última será más precisa. En todos los casos se instanciará el objeto **LocationListener** para buscar posiciones, sólo que se llamará con diferentes parámetros.

Una vez obtenidas las coordenadas de posición se guardan en la clase aplicación para poder usarlas más adelante en cualquier clase de la aplicación.

Por otro lado, también se guardará la posición como preferencia de la aplicación, con el objeto SharedPreferences, para poder utilizarla en una próxima ejecución si se selecciona la opción de utilizar última posición.

- *Seleccionar una dirección*

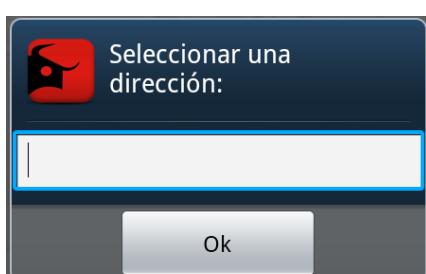


Figura 21. Pantalla seleccionar dirección

En caso de elegir la opción de seleccionar una dirección, se muestra otro AlertDialog donde el usuario podrá introducir la dirección que desea utilizar.

Una vez se tiene la dirección a resolver se realiza una petición a Google Geocoder:

`(geocoder.getFromLocationName("dirección"))`

y éste devolverá un listado de posibles coordenadas. Se guardará la primera porque es la más adecuada según Google Geocoder.

Una vez se tenga disponible la posición se guardará en la clase Aplicación y en las SharedPreferences, de la misma manera que antes.

- *Utilizar última posición*

Este caso es el más sencillo, sólo se tiene que recuperar la posición guardada en las preferencias de la aplicación (SharedPreferences), por alguno de los dos métodos anteriores, y guardarla en la clase Aplicación como posición a utilizar. Si no hay posiciones previas guardadas se mostrará un mensaje de error y se volverá a presentar el menú de geolocalización.

4.1.1.3. Pantalla Principal



Figura 22. Pantalla Principal

Interfaz: Al contrario que las pantallas anteriores, en esta pantalla la mayor complejidad se encuentra en la realización de la interfaz de usuario. No sólo por la combinación de iconos, backgrounds..., sino por la dificultad en cuanto al diseño de los mismos y al combinarlos de manera que se adapten y se muestren correctamente en cualquier tipo de pantalla Android.

Para conseguir que la pantalla se adapte automáticamente a la pantalla del dispositivo en que se ejecuta hay que trabajar con medidas relativas (dip) en vez absolutas (px) y jugar con los pesos de cada componente.

Funcionalidad: La funcionalidad de esta pantalla, es relativamente sencilla, lo que se hace es coger cada elemento al que se desea dar funcionalidad por separado y asociarle un evento onClick para que llame a la pantalla que se haya seleccionado mediante Intents.

4.1.1.4. Seleccionar Día Programa



Figura 23. Pantalla Seleccionar Día

Interfaz: Esta es una pantalla relativamente sencilla en cuanto a interfaz, se trata de dos LinearLayout, uno que servirá de cabecera donde se insertará un ImageView para volver a la principal y el texto de título de sección, y otra que hará de footer donde se insertará el eslogan de la aplicación. En el medio se insertará un ListView que será una lista de elementos que habrá que llenar en la clase de interfaz, para ello, habrá que crear otro XML de interfaz para cada fila de la lista.

Funcionalidad: Lo más complicado de esta pantalla es llenar la lista, es decir, adaptar el Array de elementos de la clase Dia al ListView que se ha creado en el interfaz. Para ello se utiliza un Custom Adapter en el que se mapea a cada elemento del interfaz su valor. Además hay que asociar un evento onClick a cada fila de la lista para que llame con un Intent a la pantalla de los eventos de ese día. Para ello, se pasará como variable al intent el día para que se resuelva en la siguiente pantalla.

4.1.1.5. Consultar Eventos del día



Figura 24. Pantalla Eventos del Día

Interfaz: En esta pantalla además de la cabecera y el footer ya explicados anteriormente, se introduce un TabWidget para trabajar con pestañas. Estas pestañas se gestionarán desde el código.

Funcionalidad: Lo primero que se hace en esta pantalla es comprobar qué día hay que cargar comprobando la variable que viene con el Intent. Tras esto hay que crear las 3 pestañas y asociarles un evento a cada una (Mañana, tarde, noche).

En este evento se consultarán los eventos de la clase aplicación que corresponden a dicho día y dichas horas y se mostrarán en un ListView creado por código. A cada evento se le asociará un evento onClick qué lanzará un Intent a la ficha de ese evento con el id del evento como parámetro extra.

4.1.1.6. Consultar Ficha Evento



Figura 25. Pantalla Ficha de Evento

Interfaz: En esta pantalla, además de la cabecera y footer, hay que crear un LinearLayout para el nombre del evento, otro para todos los datos del evento (imagen, lugar, día, hora y descripción), y otro adicional que en el que se situarán tres botones para las funcionalidades adicionales de la ficha.

Funcionalidad: La funcionalidad de esta pantalla es la de llenar los campos que se han creado en la interfaz, para ello, el primer paso es encontrar el evento que tenga el id que se ha pasado en el Intent para obtener sus datos. Una vez se hayan recogido, se mapeará cada campo con el valor de dicho evento para ese atributo.

Adicionalmente, habrá que asociar los eventos a cada botón de la parte inferior. En este caso las tres llamadas harán referencia a aplicaciones externas, por lo que bastará con realizar la llamada, utilizando la

posición del usuario y la del evento en los dos primeros casos, y Android se encargará de resolver estas llamadas con la aplicación correspondiente (Cómo llegar, Street View, Agenda).



Figura 28. Pantalla Cómo Llegar



Figura 27. Pantalla StreetView



Figura 26. Pantalla Agenda

4.1.1.7. Listado de Eventos Activos



Figura 29. Pantalla Lista Eventos Activos

Interfaz: Esta pantalla se compone de la cabecera y footer habituales, a la que se añade un ListView para llenar con eventos activos y un grupo de botones en un LinearLayout en la parte inferior para visualizar los eventos de tres maneras diferentes.

Funcionalidad: El primer paso es obtener de la lista total de eventos de la clase Aplicación los eventos que están activos actualmente o los que van a comenzar en las próximas tres horas. Para ello habrá que coger el día y la hora del sistema y compararlos con las horas de inicio y de fin, y las fechas de los eventos. Una vez obtenidos se guardan como variable de aplicación para poder utilizarlos en las otras formas de visualización.

Una vez se tienen los eventos en dos Arrays: activos y próximos es cuestión de llenar el ListView con un

CustomAdapter introduciendo cabeceras para separarlos. Además habrá que dar funcionalidad a los botones de "Mapa" y "R.A." cada uno llamará a un Intent. Así como a los eventos para ir a la ficha.

4.1.1.8. Mapa de eventos Activos

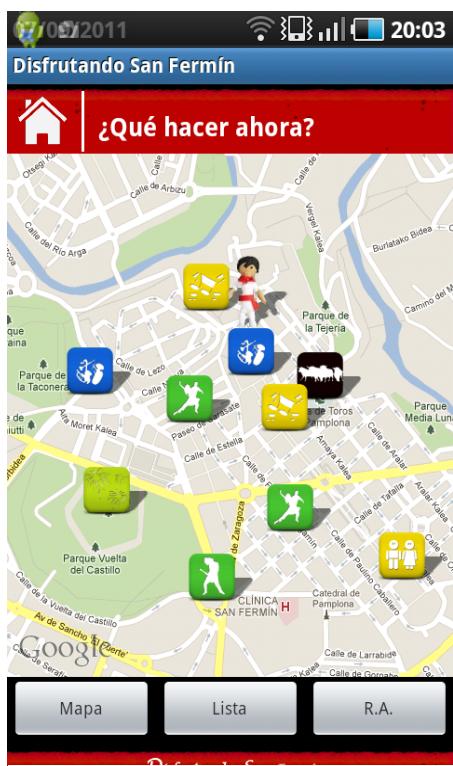


Figura 30. Pantalla Mapa Eventos Activos

Interfaz: Esta pantalla es idéntica a la anterior, sólo que se cambia el ListView por un MapView, donde se podrá colocar un mapa de Google e insertar Markers.

Funcionalidad: Lo primero que hay que hacer en esta pantalla es coger la posición registrada de la clase aplicación así como la lista de eventos activos calculados en la pantalla anterior.

Una vez se tienen todos los datos necesarios hay que inicializar el mapa de Google, decirle el tipo, el zoom y el centro, en este caso se centrará en la posición del usuario. Posteriormente habrá que cargar los markers. Por un lado se cargará la posición del usuario con el icono correspondiente y por otro lado habrá que recorrer el array de eventos activos cargando cada marker con su icono y dándole la funcionalidad para que vaya a la ficha de dicho evento onTap.

4.1.1.9. Realidad aumentada de Eventos Activos



Figura 31. Realidad aumentada eventos activos

Interfaz: No será necesario realizar interfaces para esta funcionalidad ya que se utiliza una llamada al API de la aplicación de realidad aumentada Wikitude para ofrecer esta forma de visualización de eventos.

Funcionalidad: Wikitude ha desarrollado un API muy sencilla e intuitiva para poder introducir la realidad aumentada en cualquier aplicación, tanto para Android como para iPhone.

Sólo habrá que coger la lista de eventos activos calculados anteriormente, rellenar con ellos la lista de puntos de Wikitude, dándoles funcionalidad en caso de que se seleccione uno de ellos e inicializar un Intent del API de Wikitude. A partir de ahí, wikitude se encarga de todo, volviendo a la pantalla en la que se encontraba el usuario al volver de Wikitude.

4.1.1.10. Buscador de Eventos: Filtro Categoría



Interfaz: Esta interfaz consta de la cabecera y el footer, un ListView para llenar con categorías de eventos y un botón para efectuar la acción de la búsqueda. Cabe destacar que en la interfaz de la fila de la lista se introduce el elemento CheckBox.

Funcionalidad: En esta pantalla lo que se realiza es el llenado de las categorías de eventos en la lista, para ello habrá que cogerlas previamente de la clase Aplicación. Para hacer la adaptación se utiliza un CustomAdapter. Cada vez que se seleccione o se deseccione una categoría esta se introduce o se borra de un Array de Categorías seleccionadas

De esta forma, en el evento onClick del botón de buscar sólo habrá que pasar como parámetro al Intent los ids de las categorías seleccionadas para que sea la siguiente pantalla la que realice el filtrado.

Figura 32. Pantalla Buscador Filtro

4.1.1.11. Buscador de Eventos: Resultados Búsqueda



Figura 33. Pantalla Buscador Resultados

4.1.1.12. Consultar Servicios: Filtrar por Categoría



Figura 34. Pantalla Filtro Servicios

Interfaz: Se trata de una interfaz muy sencilla, solamente se compone por la cabecera y el footer, y un ListView simple para mostrar los resultados de la búsqueda de eventos.

Funcionalidad: El primer paso es recibir las categorías de eventos que se han pasado de la pantalla anterior con el intent. Una vez se tienen se recorre toda la lista de eventos de la clase Aplicación y se comprueba si su categoría coincide con las seleccionadas. De esta forma se construye un Array de eventos.

De la misma manera que en muchas de las pantallas anteriores, se mapeará ese array de eventos resultante en el ListView creado en el interfaz mediante un CustomAdapter, añadiendo a cada fila un evento onClick para que al seleccionarlo, lleve al usuario a la ficha de dicho evento.

Interfaz: Esta interfaz es idéntica a la del filtro por categorías de eventos (4.1.10), sólo cambia el título, la estructura es la misma, se rellenará por código de manera diferente.

Funcionalidad: En esta pantalla se introducen las categorías de servicios en la lista, para ello habrá que cogerlas previamente de la clase Aplicación. Para mapearlas en el ListView se utiliza un CustomAdapter. Cada vez que se seleccione o se deseccione una categoría esta se introduce o se borra de un Array de Categorías seleccionadas.

En el evento onClick del botón de buscar sólo habrá que pasar como parámetro al Intent los ids de las categorías de servicios seleccionadas para que sea la siguiente pantalla la que realice el filtrado.

4.1.1.13. Consultar Servicios: Resultados Modo Lista



Figura 35. Pantalla Servicios Lista

Interfaz: a la cabecera y footer habituales, se añade un ListView para llenar con servicios de las categorías seleccionadas y un grupo de botones en la parte inferior para visualizar los eventos de tres maneras diferentes.

Funcionalidad: El primer paso es obtener de la lista total de servicios seleccionados una vez realizados los filtros por categoría. Para ello habrá recorrerlos comprobando si son de las categorías seleccionadas. Una vez obtenidos, se calcula la distancia entre la posición del usuario y el servicio, se guardan como variable de aplicación para poder utilizarlos en las otras formas de visualización y se rellena el ListView con un CustomAdapter.

A su vez habrá que dar funcionalidad a los botones de "Mapa" y "R.A.", asociándoles un evento onClick, cada uno llamará a un Intent.

4.1.1.14. Consultar Servicios: Resultados Modo Mapa

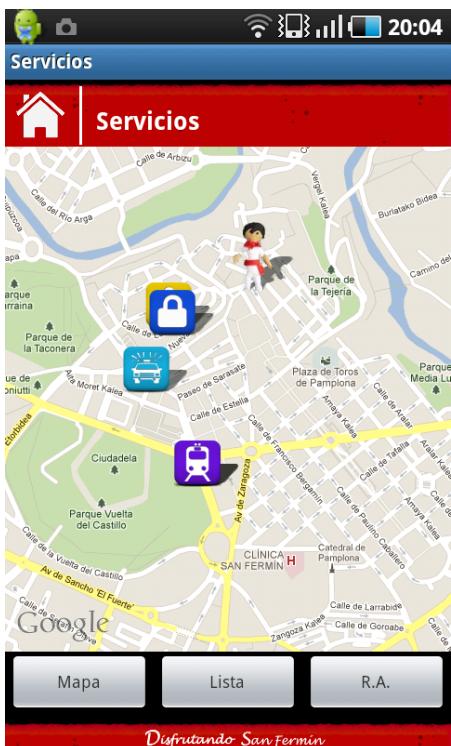


Figura 36. Pantalla Servicios Mapa

Interfaz: Esta pantalla es la de mostrar los eventos activos en modo mapa, sólo que se llenará con los servicios en vez de con los eventos activos.

Funcionalidad: El primer paso es coger la posición registrada así como los servicios a presentar de la clase aplicación.

Tras esto habrá que inicializar el mapa de Google, decirle el tipo, el zoom y el centro, en este caso se centrará en la posición del usuario.

Por último habrá que cargar los servicios. Por un lado se cargará la posición del usuario con el ícono correspondiente y por otro lado habrá que recorrer el array de servicios seleccionados cargando cada servicio con su ícono de categoría correspondiente y darle la funcionalidad para que vaya a la ficha de dicho servicio onTap.

4.1.1.15. Consultar Servicios: Resultados modo realidad aumentada

Interfaz: No será necesario realizar interfaces para esta funcionalidad ya que se utiliza una llamada al API de la aplicación de realidad aumentada Wikitude para ofrecer esta forma de visualización de eventos.

Funcionalidad: Sólo habrá que coger la lista de servicios seleccionados, rellenar con ellos la lista de puntos de Wikitude, dándoles funcionalidad en caso de que se seleccione uno de ellos e inicializar un Intent del API de Wikitude.

4.1.1.16. Consultar Ficha Servicio



Figura 37. Pantalla Ficha de Servicio

Interfaz: Esta pantalla tiene una estructura muy similar a la de la ficha de Evento, además de la cabecera y footer, hay que crear un LinearLayout para el nombre del evento, otro para todos los datos del servicio (imagen, teléfono y descripción), y otro adicional que en el que se situarán tres botones para las funcionalidades adicionales de la ficha.

Funcionalidad: La funcionalidad de esta pantalla es la de llenar los campos que se han creado en la interfaz, para ello, el primer paso es encontrar el Servicio que tenga el id que se ha pasado en el Intent para obtener sus datos. Una vez se hayan recogido, se mapeará cada campo con el valor de dicho servicio para ese atributo.

A su vez, habrá que dar funcionalidad a los botones de la parte inferior. En este caso las tres llamadas harán referencia a aplicaciones externas a la nuestra por lo que bastará con realizar la llamada, utilizando la posición del usuario y la del evento en los dos primeros casos, y Android se encargará de resolver estas llamadas con la aplicación correspondiente (Cómo llegar, Street View, Realizar llamada).

4.1.1.17. Teléfonos de Interés



Interfaz: Se trata de la misma pantalla utilizada anteriormente con solamente un ListView con cabecera y footer, lo que cambia es el interfaz de cada fila, a la que hay que añadir un ImageView a la derecha.

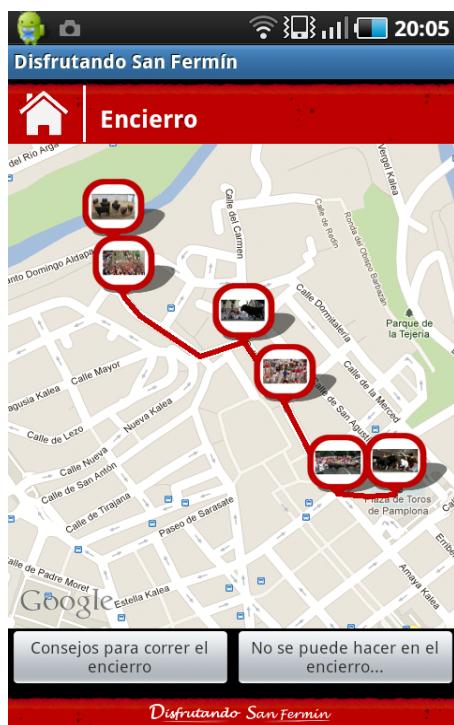
Funcionalidad: Lo primero que hay que hacer es coger la lista de objetos Teléfono previamente cargados en la clase aplicación.

Una vez se tiene la lista, se recorre y se va mapeando en el ListView con un CustomAdapter. Para cada fila se hace un evento onClick en el que se lanzará un AlertDialog para que el usuario confirme que desea llamar. En caso afirmativo se lanzará un Intent al sistema (Intent.ACTION_CALL), para que se llame al teléfono seleccionado.



Figura 38. Pantalla Teléfonos de interés

4.1.1.18. Encierro



Interfaz: Esta interfaz se compone por el header y el footer ya conocidos, delimitando a un MapView y un LinearLayout con dos botones para ir a las demás pantallas de la sección.

Funcionalidad: En esta pantalla se marca el recorrido del encierro en un mapa de Google con fotos de las zonas más importantes. Para ello el primer paso es inicializar el mapa de Google: centro, zoom, tipo... y posteriormente se marcará la línea de recorrido definida por un Array de coordenadas.

Una vez marcado el recorrido se pasará a poner los markers con las imágenes del encierro. Para ello se irá uno por uno seleccionando su posición y su marker. A cada marker habrá que asociarle un evento onTap para que al hacer click se amplíe la imagen que contiene mostrándola en un AlertDialog.

Figura 39. Pantalla Encierro Recorrido

En cuanto a los botones de la parte inferior, ambos lanzan un Intent a unas pantallas que no realizarán funcionalidad. Solamente se cargará la interfaz estática correspondiente que dará información. Esta pantalla consta del header y el footer con un ScrollView donde estará el texto informativo y los botones de la pantalla anterior para moverse por la sección. Estas son las pantallas:

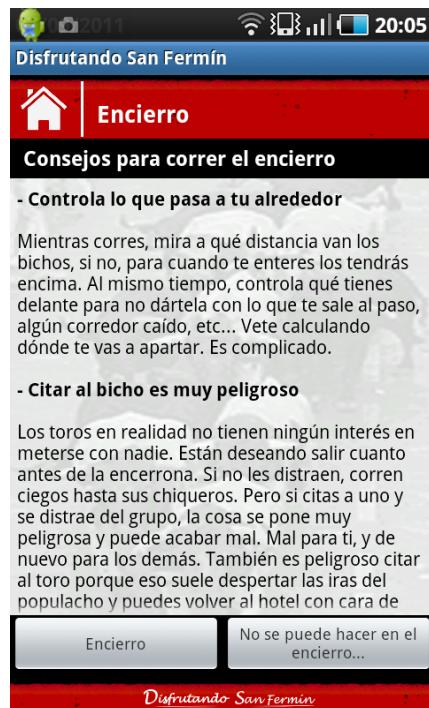


Figura 40. Pantalla Encierro Consejos

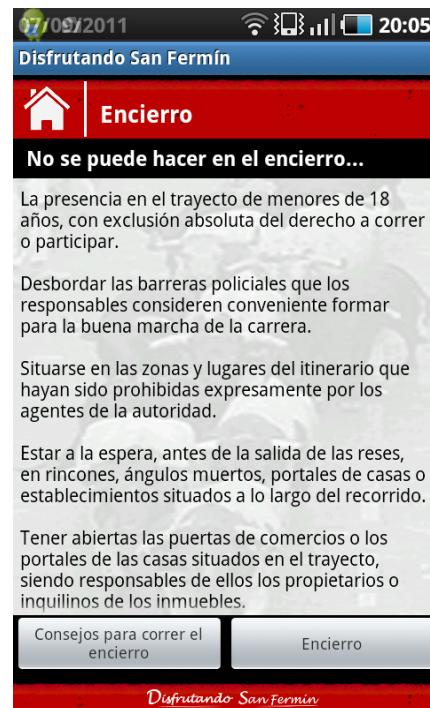


Figura 41. Pantalla Encierro Prohibiciones

4.1.1.19. Twitter



Figura 42. Pantalla Twitter

Para esta pantalla no será necesario desarrollar ninguna interfaz ni clase Java ya que sólo será necesario que al seleccionar el ícono de Twitter en la pantalla inicial se lance un Intent con la url del Twitter con el HashTag deseado y se abrirá un navegador que se hará cargo de la ejecución, volviendo a la aplicación al cerrar dicho navegador.

4.1.1.20. Pantalla Opciones

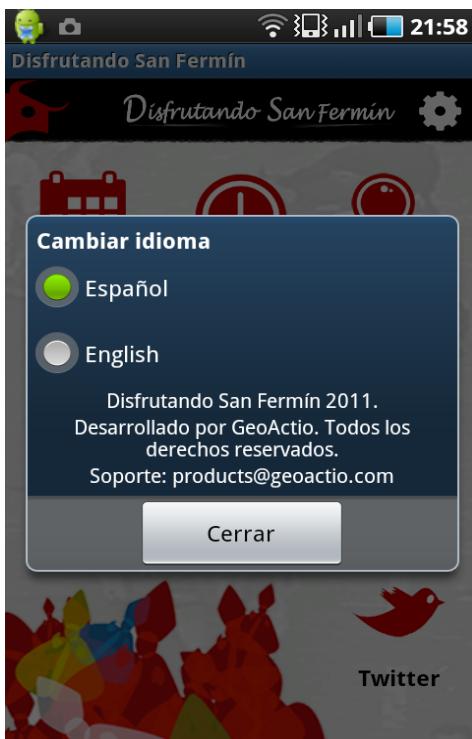


Figura 43. Pantalla Opciones

Interfaz: para esta pantalla no hay que realizar interfaz alguna ya que se trata de un AlertDialog que se rellenará por código.

Funcionalidad: Para el menú opciones se lanzará un AlertDialog que habrá que llenar por código. Se introducirán dos RadioButton para seleccionar el idioma y el texto informativo de la aplicación en la parte inferior.

La funcionalidad en los RadioButtons es la siguiente. Cuando se pulse en alguno de ellos, se cambiará el parámetro Locale de la clase interna de Android “Configuration” y se reiniciará la aplicación para que se haga efectivo el cambio de idioma.

4.1.2. Metodología Scrum

El proyecto “Disfrutando San Fermín” entró en el product backlog de los sprints de Mayo y Junio de 2011. En estos Sprints, las historias de usuario de este proyecto se mezclaban con historias de usuario de otros proyectos.

En cuanto al reparto de tareas en estos sprints, un servidor estuvo dedicado al 100% a este proyecto, recibiendo apoyo de otros miembros del equipo cuando era necesario, sobre todo en tareas de diseño e inserción de datos.

Debido a que en los sprints se mezclaban las tareas de este proyecto con las de otros proyectos, de los que no corresponde dar información, no se mostrarán ni la evolución día a día de las tareas ni los burndown charts, ya que no tienen sentido sólo para un proyecto de los varios del sprint al no estar asociado un número fijo de personas al mismo.

Las tareas definidas en la tabla anterior se repartieron entre los sprints de Mayo y Junio, terminando la última de las tareas el día 28 de Junio. Antes de la fecha límite de finalización. Estas es la tabla de comparación de los días persona estimados con los días persona reales para cada tarea.

Story ID	Task ID	Descripción	Estimado	Real
1	1	Análisis y Diseño	1,5	1,5
2	2	Diseño pantallas	2	3
2	3	Diseño iconos	1	1
3	4	Geolocalización: Posición Actual	2	2
3	5	Geolocalización: Buscar Dirección	0,5	0,5
3	6	Geolocalización: Última posición	0,5	0,5
4	7	Pantalla de carga	1	1,5
4	8	Pantalla inicial	1	1,5
5	9	Programa: Listado días	0,5	0,5
5	10	Programa: Tabs eventos	1	1,5
5	11	Ficha Evento + Cómo llegar + StreetView + Agenda	2	2
6	12	Cálculo eventos activos y próximos	1	0,5
6	13	¿Qué hacer ahora?:Vista modo lista	0,5	0,5
6	14	¿Qué hacer ahora?:Vista modo mapa	2	1,5
6	15	¿Qué hacer ahora?:Vista modo realidad aumentada	2	1,5
6	16	¿Qué hacer ahora?:Mostrar ficha evento	0,5	0,5
7	17	Buscador: filtro por clase	1	1
7	18	Buscador: mostrar resultados	0,5	0,5
8	19	Servicios: Filtro por tipo de servicio	0,5	0,5
8	20	Servicios: resultados modo lista	1	0,5
8	21	Servicios: resultados modo mapa	0,5	0,5
8	22	Servicios: resultados realidad aumentada	1	1
8	23	Servicios: Ficha Servicio	0,5	1
9	24	Sección teléfonos de interés	1	1
10	25	Encierro: recorrido sobre mapa con fotos	1,5	2

10	26	Encierro: recomendaciones y prohibiciones	0,5	0,5
11	27	Análisis e implementación sección Twitter	1	0,25
12	28	Desarrollo plataforma multi idioma	1	0,5
12	29	Réplica textos	0,5	0,5
13	30	Introducción de datos	4	5
14	31	Adaptación a todas pantallas	2	2,5
14	32	Adaptación a todas las resoluciones	2	1,5

Total Real		38,75
Total Estimado		37,50

Tabla 33. Comparación Estimado - Real

Se observa que los días persona que se ha tardado en realidad en realizar el proyecto superan a los días que se planificaron para su realización, aunque en un margen muy pequeño, lo que ha permitido entregar a tiempo el proyecto.

4.2. Pruebas

En este apartado se va a comprobar el correcto funcionamiento de la aplicación mediante un conjunto de pruebas. Estas pruebas se centrarán sobre todo en vigilar cómo se comporta la aplicación en escenarios no ideales. Para ello se estudiarán los posibles flujos alternativos de los casos de uso para analizar el comportamiento de la aplicación en ellos.

Caso de Uso:	CogerPosicionActual
Caso de Prueba:	Coger Posición Actual sin red ni GPS
Condiciones:	Se pretende utilizar la posición actual sin tener acceso a red ni a GPS
Acción:	El usuario selecciona utilizar posición actual en el dialogo de geoposicionamiento.
Resultado:	La aplicación muestra un error de geoposicionamiento y vuelve a mostrar el dialogo de geolocalización.

Tabla 34. Prueba posición actual

Caso de Uso:	UtilizarDirección
Caso de Prueba:	Utilizar dirección para geoposicionamiento si no hay red.
Condiciones:	Se pretende utilizar una dirección sin tener acceso a red.
Acción:	El usuario selecciona utilizar dirección en el dialogo de geoposicionamiento.
Resultado:	La aplicación muestra un error de imposible resolver dirección y vuelve a mostrar el dialogo de geolocalización.

Tabla 35 Prueba utilizar dirección sin red

Caso de Uso:	UtilizarDirección
Caso de Prueba:	Utilizar dirección no válida para el geoposicionamiento.
Condiciones:	Se introduce una dirección no válida en el campo de dirección.
Acción:	El usuario selecciona utilizar dirección en el dialogo de geoposicionamiento e introduce una dirección no válida.
Resultado:	La aplicación muestra un error de imposible resolver dirección y vuelve a mostrar el dialogo de geolocalización.

Tabla 36. Prueba utilizar dirección no válida

Caso de Uso:	UtilizarUltimaPosicion
Caso de Prueba:	Utilizar ultima dirección cuando no hay todavía.
Condiciones:	Se opta por utilizar última posición cuando no existe ninguna posición previa guardada.
Acción:	El usuario selecciona utilizar última posición en el dialogo de geoposicionamiento.
Resultado:	La aplicación muestra un error de que no hay posiciones previas guardadas y muestra de nuevo el dialogo de geoposicionamiento.

Tabla 37. Prueba última posición

Caso de Uso:	ComoLlegar
Caso de Prueba:	Utilizar cómo llegar sin posición registrada.
Condiciones:	Se selecciona cómo llegar en la ficha del evento o del servicio cuando no hay posición del usuario registrada.
Acción:	El usuario selecciona cómo llegar en la ficha del evento o del servicio.
Resultado:	Se muestra un mensaje de que no es posible mostrar indicaciones.

Tabla 38. Prueba Cómo Llegar

Caso de Uso:	StreetView
Caso de Prueba:	Utilizar StreetView donde no lo hay.
Condiciones:	Se selecciona Streetview en la ficha del evento o del servicio con una posición de evento o de servicio donde no está disponible esta funcionalidad de Google.
Acción:	El usuario selecciona StreetView en la ficha del evento o del servicio.
Resultado:	Se muestra un mensaje de que no es posible mostrar vista 360º de la ubicación del recurso.

Tabla 39. Prueba StreetView

Caso de Uso:	ConsultarEventosActivos
Caso de Prueba:	Intentar consultar eventos activos cuando no hay.
Condiciones:	Se selecciona consultar eventos activos cuando no hay ninguno, ya sea porque no es San Fermín o porque no hay ninguno en ese momento.
Acción:	El usuario selecciona “Qué hacer ahora” desde la pantalla principal.
Resultado:	Se muestra el listado de eventos vacío y los botones de vista de mapa y de realidad aumentada están deshabilitados.

Tabla 40. Prueba Eventos Activos

Caso de Uso:	RealidadAumentadaEventos/RealidadAumentadaServicios
Caso de Prueba:	Realidad aumentada sin Wikitude.
Condiciones:	Se selecciona en vista de eventos activos o de servicios como modo de visualización la Realidad aumentada y no se tiene instalada previamente la aplicación Wikitude World Browser.
Acción:	El usuario selecciona el modo de visualización la Realidad aumentada en eventos activos o servicios.
Resultado:	Se muestra un dialogo informando que sin Wikitude no se puede disfrutar de esta funcionalidad y se sugiere al usuario que se lo descargue desde el market de manera gratuita incluyendo un enlace al mismo.

Tabla 41. Prueba Realidad aumentada

Hay que decir que a pesar de los esfuerzos que se han realizado para que esta aplicación sea lo menos dependiente posible de la red, hay funcionalidades para la que es necesario acceder a red: descargar mapa Google, cómo llegar y StreetView. Para estas funcionalidades se presupone que el usuario tiene acceso a la red, en otro caso no podrá disfrutar de ellas.

4.3. Despliegue

En este caso el despliegue del sistema consistirá en la subida de la aplicación finalizada al Android Market, desde donde estará accesible a cualquier SmartPhone Android del mundo de forma gratuita.

El Android Market es una tienda de software en línea desarrollada por Google para los dispositivos Android. Viene pre instalado en la mayoría de los dispositivos Android y permite a los usuarios buscar y descargar aplicaciones publicadas por terceros.

Para poder subir la aplicación al Android Market el primer paso es crearse una cuenta de desarrollador de Android. Se realiza de una manera sencilla y sólo cuesta 25\$ (<https://market.android.com/publish>). Con esta licencia se podrán subir cuantas aplicaciones como se desee al market sin tener que abonar nada nunca más. También se podrá cobrar por la descarga de las aplicaciones, en este caso Google se queda con un 30% de los beneficios de las ventas.

Una vez se ha creado la cuenta de desarrollador, el siguiente paso es compilar la aplicación y firmarla. Para firmarla hay que obtener una clave privada. Para ello habrá que acceder al directorio donde se tiene instalado Java del ordenador y ejecutar el siguiente comando:

```
$ keytool -genkey -v -keystore my-release-key.keystore  
-alias alias_name -keyalg RSA -keysize 2048 -validity 10000
```

El fichero keystore que se obtenga servirá para firmar la aplicación al exportarla. Una se haya creado el .apk firmado, habrá que acceder a la cuenta anteriormente creada desde <https://market.android.com/publish/Home>. Habrá que pulsar sobre “Subir aplicación” y llenar todos los campos requeridos.

Tras este proceso la aplicación ya estará disponible para la descarga de todo el que lo deseé en el Android Market de manera gratuita.



Figura 44. Android Market

5. CONCLUSIONES Y LÍNEAS FUTURAS

En este capítulo se analizarán los resultados obtenidos en este proyecto y se comprobará si se han logrado los objetivos propuestos al inicio del mismo, analizando a su vez posibles mejoras futuras para la aplicación.

5.1. Conclusiones

Una vez finalizado San Fermín llega el momento de sacar las conclusiones necesarias de este proyecto, para ello lo primero es recordar los objetivos propuestos al inicio del proyecto:

- Por un lado el primer objetivo era el desarrollo de una aplicación robusta y atractiva en la que el usuario pudiera encontrar todo lo necesario para disfrutar de las fiestas de San Fermín de una manera rápida y sencilla.
- Por otro lado un segundo objetivo era el de darse publicidad llegando al mayor número posible de personas.

Analizando a posteriori el primero de los objetivos, se observa que la aplicación cumple con todos los requisitos tanto funcionales, como de interfaz y de sistema, además de haber superado con éxito todas las pruebas a las que ha sido sometido.

En lo que respecta al segundo objetivo, llegar al mayor número de gente posible, se considera que se ha conseguido llegar a un buen número de personas, ya que analizando los datos del Android Market, se observa que los resultados a fecha de 18 de Julio de 2011, han sido de 3621 descargas, cifra que se considera buena debido a la poca publicidad que se le ha dado a la aplicación y al número reducido de días en que esta aplicación ha estado disponible en el Android Market, apenas 3 semanas.

Además de cumplir estos dos objetivos también hay que considerar que, durante la realización de este proyecto, se ha producido un avance en los conocimientos del sistema Android y su programación, habiendo adquirido conocimientos que pueden ser muy útiles para el futuro tanto a nivel personal como para la empresa.

En cuanto a la planificación del proyecto, se sacan conclusiones muy positivas, ya que a pesar de que se ha tardado algo más de lo planificado en realizarlo, no ha sido un impedimento para su entrega a tiempo, ya que esta demora ha sido mínima. En cuanto a la planificación Scrum, cada vez el equipo va ganando más experiencia en esta forma de trabajar, ganando en independencia, versatilidad y eficiencia.

Analizando estos datos, se concluye que este proyecto ha sido un éxito ya que ha cumplido en plazo con los objetivos que se establecieron en su lanzamiento.

5.2. Líneas Futuras

Una vez se ha observado que este desarrollo ha sido un éxito, se puede comenzar a pensar en evoluciones del mismo en un futuro.

El primero de los objetivos para el futuro debe ser el de ofrecer la aplicación para una mayor gama de sistemas operativos móviles: iPhone, Windows Phone7, BlackBerry... Éste será un tema de estudio para las próximas versiones de esta aplicación.

Por otro lado, en el mundo de la tecnología y sobre todo en el mundo de las aplicaciones móviles, todo avanza muy rápido, por lo que habrá que pensar en nuevas mejoras tanto en funcionalidad como en funcionamiento de la aplicación si se quiere que esta aplicación sea una aplicación de referencia.

Una de estas mejoras podría ser la de permitir actualizaciones de los datos de la aplicación sin necesidad de subir una nueva versión al Android Market. Un posible método sería el siguiente: en el instalable de la aplicación no iría ningún dato fijo. Cuando la aplicación se instala en el terminal se conecta a un Web Service mediante el que se descarga los datos a un sistema de almacenamiento local (BBDD, XML...). Periódicamente la aplicación comprobará si ha habido cambios en los datos y se descargarán las novedades. Mediante este método se mantendría la aplicación actualizada en todo momento rebajando además el peso de la misma al no llevar los datos en el propio instalable.

Por otro lado la sección de Twitter debería ganar más peso en las siguientes versiones ya que Twitter es en la actualidad una tecnología en auge. Estas mejoras podrían ir en torno a posicionar twits georreferenciados en la aplicación, permitir login del usuario... , en definitiva, trabajar con el API de Twitter.

En cuanto a nuevas funcionalidades, para nuevas versiones habría que potenciar la información en tiempo real y la interacción entre los usuarios, ya que en cuanto a información estática esta aplicación es bastante completa. Unas posibles funcionalidades nuevas serían las siguientes: comentarios en eventos, puntuaciones de los mismos, información de los encierros de cada día...

6. BIBLIOGRAFÍA

En este capítulo se mostrarán los libros, documentos y sitios Web que han sido de mucha utilidad para la realización de este proyecto.

6.1. Libros y Documentos

- **Android Essentials;** *Chris Haseman*; Apress
- **Mobile Design and Development;** *Brian Fling*; O'REILLY;
- **The Busy Coders Guide to Android Development;** *Mark Lawrence Murphy*;
- **Scrum y XP desde las trincheras,** Henrik Kniberg, InfoQ
- **Beginning Android,** Mark Murphy

6.2. Internet

- <http://developer.android.com/index.html>
- <http://android-developers.blogspot.com/>
- <http://code.google.com/intl/es/android/>
- www.android-spa.com/
- www.sanfermin.com/
- www.google.com/mobile/maps/
- www.google.com/mobile/android/
- <http://www.wikitude.com/developer/getting-started>
- http://www.wikitude.com/category/04_developer/wikitude-api
- <http://market.android.com/publish>
- <http://www.google.com/support/androidmarket/developer/bin/answer.py?hl=en&answer=113469>

ANEXO 1:

MANUAL DE USUARIO

En este anexo se mostrará de manera detallada el funcionamiento de la aplicación desde el punto de vista del usuario, mostrando el funcionamiento de cada pantalla por separado y mostrando las opciones que tendrá en cada una de ellas.

A1.1. Descarga e Instalación

La aplicación se puede descargar del Android Market. La instalación de la aplicación es muy sencilla, y similar a cualquier aplicación que se encuentra en el Android Market. Para realizar la descarga se buscará la aplicación “Disfrutando San Fermín” en el Android Market y una vez en la página de la aplicación se pulsará en el botón de instalación gratuita, ya que la aplicación es totalmente gratuita.



Figura 45. Pantalla Descarga 1

En la siguiente pantalla se puede comprobar cómo la aplicación esta descargándose:



Figura 46. Pantalla Descarga 2

Una vez la descarga se haya realizado correctamente, aparecerá una pantalla como la siguiente, la aplicación ya está instalada en el dispositivo y pulsando en abrir ya se podría comenzar a utilizar “Disfrutando San Fermín”.



Actualizaciones
Mantén esta aplicación actualizada automáticamente.

Actualizaciones automáticas

Mi comentario
Comparte tu opinión sobre la aplicación con

Figura 47. Pantalla Descarga 3

A1.2. Pasos Previos

Al ejecutar el ícono de la aplicación y antes de comenzar a utilizar la aplicación propiamente dicha, y sólo en el caso de que el GPS del dispositivo esté desactivado, aparecerá un menú como el siguiente, preguntando al usuario si quiere activar el GPS ya que así podrá obtener una localización más precisa.

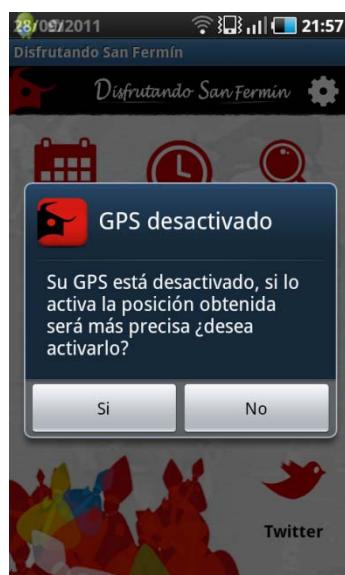


Figura 48. Manual GPS

A continuación aparecerá otro menú en el que se deberá seleccionar la opción deseada para la geolocalización del dispositivo, si no se selecciona ninguna de estas opciones el dispositivo no estará geolocalizado y no estarán disponibles algunas de las opciones de la aplicación.



Figura 49. Manual Geoposicionamiento

Si se pulsa la opción utilizar posición actual la aplicación buscará automáticamente la posición del dispositivo, mientras se esté realizando esta operación la aplicación permanecerá como se muestra a continuación.



Figura 50. Manual Buscando posición

Si el usuario decide introducir él mismo la dirección en la que se encuentra, deberá teclear la dirección y pulsar en aceptar.



Figura 51. Manual seleccionar posición

Finalmente, si el usuario selecciona la tercera opción del menú, utilizar última posición, el dispositivo tomará como posición la última utilizada en la aplicación.

Una vez seleccionada la forma de geoposicionamiento por parte del usuario, este accederá al menú general de la aplicación desde el que podrá acceder a las funcionalidades que se muestran a continuación.

A1.3. Funcionalidades

Una vez se hayan realizado los pasos previos de geolocalización del dispositivo, aparecerá el menú principal de la aplicación como podemos ver en la siguiente imagen:



Figura 52. Manual menú principal

A1.3.1. Menú Configuración

El usuario puede acceder al menú de configuración pulsando en el ícono de la parte superior derecha de la pantalla, en él, podrá modificar el idioma de la aplicación pudiendo elegir entre castellano e inglés, como se ve en la siguiente pantalla. Además accederá a los datos de la empresa desarrolladora.



Figura 53. Manual Configuración

A1.3.2. Programa

Si el usuario entra en la sección programa, en la siguiente pantalla se encontrará con una lista de los días de San Fermín. Para visualizar los datos de los eventos del programa deberá pulsar sobre el día deseado:



Figura 54. Manual selección día

El programa de cada día quedará dividido por secciones: las actividades que se realizan por la mañana, por la tarde y por la noche, cada grupo de actividades estará en una pestaña diferente. Para ver con detalle la deseada, se pulsa sobre ella. Además se podrá cambiar de día del programa pulsando sobre las flechitas que aparecen en la parte inferior de la pantalla.



Figura 55. Manual Eventos del día

Al seleccionar un evento en la pantalla anterior, se accederá a la ficha de dicho evento, donde se mostrará el evento con todos los detalles, además en la parte inferior de la pantalla se mostrará un menú con tres opciones: Cómo llegar, StreetView y Agenda.



Figura 56. Manual Ficha de Evento

Si se selecciona Cómo llegar, se accederá a las indicaciones de Google Maps para llegar desde la posición del dispositivo a la posición donde se va a realizar el evento seleccionado. Se obtendrá una pantalla como la siguiente:



Figura 57. Manual Cómo llegar

Si se selecciona la opción de Street View, se proporcionará al usuario, a través de Google Street View, la vista fotográfica 360º del lugar en el que se desarrolla el evento seleccionado.



Figura 58. Manual StreetView

Para finalizar, la última opción de este menú es Agenda. Si se selecciona se tendrá la posibilidad de guardar el evento en la agenda del dispositivo, para ello se accederá a la pantalla de creación de eventos del dispositivo donde ya se habrán llenado los campos con los datos de dicho evento, de forma que sólo deberá pulsar guardar para tener este evento en su agenda.



Figura 59. Manual Agenda

A1.3.3. ¿Qué hacer ahora?

La opción ¿Qué hacer ahora? Permitirá al usuario ver de una forma rápida y cómoda los eventos que están activos en ese momento y los que van a dar comienzo en las próximas tres horas. Se podrán ver de tres formas diferentes: localizando los eventos en el mapa, en formato de lista o en realidad aumentada, según se elija en el menú inferior de la pantalla. A continuación se muestra el formato de lista, que es el que se mostrará por defecto al pulsar en la opción ¿Qué hacer ahora? del menú principal.



Figura 60. Manual Qué hacer ahora lista

La pantalla que mostrará dichos eventos en modo mapa tendrá esta forma:



Figura 61. Manual Qué hacer ahora mapa

Por último, si se selecciona la vista de realidad aumentada se obtendrá la vista de la ubicación real de los eventos a través de la cámara del dispositivo. Para que esta opción de visualización funcione correctamente deberá estar instalado en el dispositivo la aplicación gratuita Wikitude Word Browser, si el programa no está instalado en el dispositivo el programa alertará con un mensaje como el siguiente, donde si se pulsa sobre descargar se accederá directamente a la página de descarga de la aplicación en el Android Market.



Figura 62. Manual Wikitude

En caso de que esté instalada, se mostrará una pantalla como la siguiente:



Figura 63. Manual Realidad aumentada

Si se selecciona un evento de los que aparecerán en la lista, mapa o realidad aumentada para ver más detalles se mostrará una pantalla de ficha de evento idéntica a la que ya se ha mostrado para la sección programa, y en la que también estarán disponibles las opciones Cómo llegar, Street View y Agenda.

A1.3.4. Buscador

Al acceder a la sección buscador, la aplicación mostrará una pantalla como la siguiente. En ella se permitirá al usuario buscar eventos de las categorías deseadas. Para realizar la búsqueda se seleccionará la o las categorías deseadas y se pulsará en el botón de buscar.



Figura 64. Manual Buscador

A continuación, aparecerá una lista con los eventos pertenecientes a las categorías seleccionadas en la búsqueda. Si se selecciona un evento de la para obtener más información del mismo, se mostrará la ficha del evento que ya se ha mostrado en programa o en ¿Qué hacer ahora?, y en la que se encontrarán también las opciones Cómo llegar, Street View y Agenda para el evento seleccionado.



Figura 65. Manual Buscador Resultados

A1.3.5. Servicios

Al acceder a la sección servicios se mostrará una lista con las categorías de algunos de los servicios que le pueden ser de gran utilidad al usuario en San Fermín. Se podrá seleccionar una o varias categorías de servicios de las que mostrar los servicios.



Figura 66. Manual Servicios Selección

Como resultado de la búsqueda se mostrarán los servicios resultantes en una pantalla como la siguiente en la que aparecerá además del servicio, los metros a los que se encuentra de la posición del usuario.



Figura 67. Manual Servicios Lista

Además de mostrar los servicios en formato lista, la aplicación dará la posibilidad al usuario de localizarlos en el mapa, de manera similar a la sección de ¿Qué hacer ahora?



Figura 68. Manual Servicios Mapa

También habrá la opción de visualizar los servicios en realidad aumentada, de manera idéntica a la vista de realidad aumentada de los eventos activos.

Si se pulsa en uno de los servicios en cualquiera de las vistas se accederán a los detalles del servicio y se tendrá la posibilidad de llamar por teléfono directamente, o informarse sobre ¿Cómo llegar? O utilizar la opción de Google Street View para dicho servicio.



Figura 69. Manual Ficha de Servicio

A1.3.6. Teléfonos de interés

Al pulsar en el menú principal el icono de teléfonos de interés aparecerá una lista de teléfonos que pueden ser útiles para el usuario en Pamplona y San Fermín.



Figura 70. Manual Teléfonos Interés

Al pulsar en sobre el teléfono deseado aparecerá un dialogo para que el usuario confirme que realmente desea realizar la llamada.



Figura 71. Manual llamar teléfono

A1.3.7. Encierro

Al acceder a la sección encierro, la aplicación muestra el recorrido del encierro sobre un mapa de Google. En el mapa aparecerán fotos en los puntos más destacados del encierro. Al pulsar sobre ellas se mostrarán a pantalla completa.



Figura 72. Manual Encierro recorrido

A su vez, desde el menú de la parte inferior de la pantalla se podrá acceder a los consejos para correr el encierro así como a lo que no se debe hacer si se corre el encierro. Esta sería la pantalla de consejos para correr en el encierro.

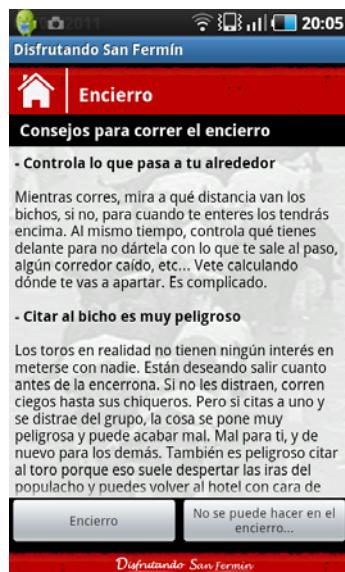


Figura 73. Manual Encierro Consejos

La pantalla de qué no se puede hacer en el encierro sería la siguiente:



Figura 74. Manual Encierro Prohibiciones

A1.3.8. Twitter

Por último, al pulsar sobre Twitter la aplicación hará una búsqueda automática en la red social Twitter sobre los últimos comentarios sobre San Fermín. Twitter es una red social que permite mandar mensajes de texto plano de bajo tamaño con un máximo de 140 caracteres y que por ello permite transmitir de forma rápida información y comentarios de los usuarios.



Figura 75. Manual Twitter

ANEXO 2: DESARROLLO DE APLICACIONES ANDROID CON ECLIPSE

En este apartado se introduce el desarrollo de aplicaciones Android en el entorno Eclipse, mostrando a su vez para ello los fundamentos de la programación en Android.

A2.1. Fundamentos de Android

En este apartado se introducirán los conceptos básicos de la programación en Android.

A2.1.1. Componentes de Aplicación

Una aplicación de Android se compone de bloques o componentes de aplicación. Hay cuatro tipos diferentes de componentes de aplicación, cada uno con un propósito y función diferente en la aplicación:

- **Actividades**: Una actividad representa a una única pantalla con una interfaz de usuario. Aunque las actividades trabajan juntas para formar una aplicación cada una es independiente de las demás. Una aplicación puede comenzar con cualquiera de estas actividades. Una actividad se implementa como una subclase de *Activity*.
- **Servicios**: Un servicio es un componente que corre en el background del sistema operativo Android para realizar por ejemplo una tarea de larga duración. Un servicio no provee interfaz de usuario. Un servicio se implementa como una subclase de *Service*.
- **Content providers**: Un Content Provider gestiona un conjunto de datos de aplicación. A través de content providers, otras aplicaciones pueden gestionar datos de una aplicación. Por ejemplo, Android provee un content provider que gestiona la agenda, de manera que se puede acceder a la información de los contactos desde cualquier aplicación que tenga permiso. Un content provider se implementa como subclase de *ContentProvider*.
- **Broadcast receivers**: Un broadcast receiver es un componente que responde a una petición broadcast del sistema. Muchos broadcast se inician por parte del sistema, por ejemplo una llamada, pero también pueden ser iniciados desde las aplicaciones. Aunque no proveen interfaz de usuario pueden crear notificaciones para alertar al usuario o pueden servir de disparador para el lanzamiento de nuevos componentes de aplicación. Un broadcast receiver extiende de *BroadcastReceiver*.

Una característica muy importante de Android es que cualquier componente de aplicación puede iniciar otro componente de aplicación. Por ejemplo, si se quiere sacar una foto con la cámara existe otra aplicación a la que llamar para que sea ella la que se encargue de esta tarea, sin necesidad de que se tenga que desarrollar desde cero o incluir librerías en el código, mientras que el usuario tiene la sensación que la gestión de la cámara de fotos tiene lugar en la propia aplicación.

Con motivo de que el sistema ejecuta los procesos de cada aplicación por separado, una aplicación no podrá activar directamente un componente de otra aplicación, será el sistema el que lo haga, para ello la aplicación tendrá que realizar una llamada al sistema en la que se especifica qué **Intent** se desea iniciar.

A2.1.2. Lanzamiento de nuevos componentes

Tres de los cuatro tipos de componentes, Actividades, Servicios y Broadcast Receivers, se activan mediante mensajes asíncronos llamados **Intents**. Los Intents unen componentes atómicos en tiempo de ejecución. Un intent se instancia con el objeto **Intent**, que especifica el mensaje al que deberán responder los Broadcast Receivers (activación explícita) o el componente en concreto que se deseará activar (activación implícita).

Por otro lado, los Content Providers no son activados por Intents. En vez de eso se activan cuando son llamados desde un ContentResolver. Un Content Resolver maneja todas las transacciones con el content provider, de forma que se abstrae el content provider del componente que lo llama, así se gana en seguridad.

En cuanto a los métodos para activar cada componente. Existen diferentes formas de activación para cada uno:

- Para activar una actividad hay dos métodos diferentes: `startActivity(Intent)`, que iniciará el intent deseado, o `startActivityForResult(Intent)`, que iniciará la actividad y esperará que la misma le devuelva un resultado para tratarlo como se deseé.
- Para iniciar un servicio o comunicarse con uno que está en ejecución se puede iniciar un Intent con `startService(Intent)`, o se puede hacer bind del servicio con `bindService(Intent)`.
- Se puede iniciar un broadcast receiver con los métodos: `sendBroadcast()`, `sendOrderedBroadcast()` o `sendStickyBroadcast()`.
- Para realizar una consulta a un content provider hay que ejecutar un método `query()` en un content Resolver.

A2.1.3. Android Manifest

Toda aplicación Android tendrá un archivo llamado "AndroidManifest.xml" en el directorio raíz del proyecto. El "Android Manifest" guarda información esencial sobre la aplicación, información sin la que no se podría ejecutar el proyecto. El contenido del manifest es el siguiente:

- **Nombre del paquete de la Aplicación:** que sirve como un identificador único de la aplicación.
- **Componentes de la aplicación:** Describe los componentes de Aplicación que forman la aplicación. Para cada uno de ellos deberá presentar el nombre y el componente que implementa. De esta forma el sistema Android sabe qué componentes hay y bajo qué condiciones se ejecutarán.

- El Android Manifest también determina qué **procesos contendrán los componentes** de la aplicación.
- Determina qué **permisos** tiene la aplicación para acceder a partes protegidas del API e interactuar con otras aplicaciones.
- También determina qué **permisos tienen otras aplicaciones** para acceder a los componentes de la aplicación.
- Declara el **nivel mínimo del API Android**.
- Lista de las **librerías** que va a utilizar la aplicación.
- **Tipos de pantallas** que soportará la aplicación.

La estructura de un "Android Manifest" es la que se muestra a continuación:

```

<?xml version="1.0" encoding="utf-8"?>
<manifest>

    <uses-permission />
    <permission />
    <permission-tree />
    <permission-group />
    <uses-sdk />
    <uses-configuration />
    <uses-feature />
    <supports-screens />

    <application>

        <activity>
            <intent-filter>
                <action />
                <category />
                <data />
            </intent-filter>
            <meta-data />
        </activity>
        <receiver>
            <intent-filter> . . . </intent-filter>
            <meta-data />
        </receiver>
        <uses-library />

    </application>

</manifest>

```

A2.1.4. Recursos de la aplicación

Además de código Java, una aplicación Android tiene otros componentes: imágenes, ficheros de datos, audio, vídeo... y además, ficheros XML que definen la estructura de las pantallas de la aplicación. La utilización de estos recursos facilita la realización de la aplicación ya que para ciertos ámbitos, generación de interfaces de usuario, multi-idioma, adaptación a diferentes pantallas..., es mucho más eficiente que hacerlo todo por código.

Para cada recurso que se introduce en el proyecto, el SDK le asocia un identificador único, mediante el que se podrá referenciar al recurso desde el código. Estos identificadores se almacenarán en la clase R de Java automáticamente, así por ejemplo, para hacer referencia a la imagen ‘casa.png’ de la carpeta ‘res/drawable/’ se utilizará el identificador que se encuentra en la clase R accediendo mediante R.drawable.casa.

Uno de las funcionalidades más importantes que ofrece el uso de recursos es la de separar la fuente de datos del código, de manera que se pueden definir fuentes de datos diferentes para diferentes configuraciones sin necesidad de repetir el código. Por ejemplo, mediante el uso de constantes del tipo String para cada idioma, se pueden definir diferentes frases para cada idioma y desde el código se utilizará la que corresponda al idioma que se está empleando en ese momento en la aplicación.

Android ofrece diferentes tags para tus recursos. Este tag es una pequeña cadena de texto que se incluye en el nombre del directorio de recursos para definir la configuración a la que responderán los recursos que hay en ese directorio. Por ejemplo en el caso de los XML que definen las pantallas de la aplicación, se guardarán en la carpeta layout las interfaces que definirán las pantallas en vertical y en la carpeta layout-land las que definen las interfaces que en horizontal. Si se guardan XMLs con el mismo nombre en ellas y se utiliza este interfaz en alguna actividad de la aplicación, ésta cargará la pantalla de vertical o de horizontal según sea la posición del teléfono.

A2.1.5. Interfaz de usuario

En una aplicación Android, la interfaces de usuario se construyen usando Views y ViewGroups.

Los objetos View son las unidades básicas de la interfaz de usuario en la plataforma Android. La clase View es la clase base de la que extienden los llamados Widgets, que son objetos completamente implementados como campos de texto o botones. Un ViewGroup sirve como base de la que extenderán las clases llamadas Layouts, que ofrecen diferentes formas de estructurar los Views. Este es el listado de Widgets y Layouts definidos en la plataforma Android:

Widgets

- Button
- ImageButton
- EditText
- TextView
- CheckBox
- Radio Button
- Gallery
- ImageSwitcher
- List
- Grid
- DatePicker
- TimePicker
- Spinner
- AutoComplete

Layouts

- LinearLayout: Permitirán alinear sus hijos en una única dirección, ya sea horizontal o vertical. La orientación predeterminada es horizontal.
- RelativeLayout: En este caso todos los elementos van colocados en una posición relativa a otro. Aquí se puede jugar con las distancias entre elementos en la pantalla.
- Absolute layout: Coloca los elementos en posiciones absolutas en la pantalla, teniendo en cuenta que la posición (0,0) es el extremo superior izquierdo de la pantalla.
- TableLayout: permite colocar los elementos en forma de tabla. Se utiliza el elemento <TableRow> para designar a una fila de la tabla. Cada fila puede tener uno o más puntos de vista. Cada vista se coloca dentro de una fila en forma de celda.
- FrameLayout: Es un marcador de posición que puede usarse para mostrar una única vista. Se pueden agregar múltiples puntos de vista a un FrameLayout pero cada uno se acumulará en la parte superior izquierda de la anterior.
- ScrollView: Es un tipo especial de FrameLayout ya que permite a los usuarios desplazarse por una serie de puntos de vista que ocupan más espacio que el despliegue físico. El scrollView solo puede contener un viewGroup y suele ser LinearLayout.

A2.1.5.1. Jerarquía de Views

En la plataforma Android, se definen las interfaces de usuario de las actividades (clases Activity), mediante una jerarquía de Views y ViewGroups. Esta jerarquía puede ser tan simple o compleja como se necesite y se utilizaran Views y GroupViews ya definidos en Android o se podrán crear objetos propios que se podrán utilizar sin problemas.

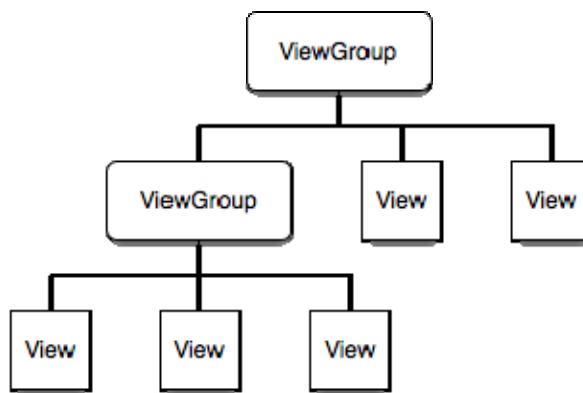


Figura 76. Jerarquía de Views Android

Para asociar una jerarquía de vistas a una actividad, esta debe de llamar a la función `setContentView()` y pasarle como parámetro el nodo raíz de dicha jerarquía. De esta forma la jerarquía de Views seleccionada se mostrará por pantalla.

A2.1.5.2. Layouts

La manera más sencilla de crear una jerarquía de Views es mediante la creación de un fichero XML de Layout. En este fichero se establecerán nodos XML que definirán los Views y ViewGroups que se quieren representar de manera jerárquica y asociándoles unos parámetros de configuración: altura, anchura, fondo...

El nombre del elemento XML es el mismo que la clase Java a la que representa. Un `<Button>` representará a un objeto de la clase Button. Cuando se carga un XML de Layout como interfaz para una actividad, Android inicializa todos estos objetos declarados en el XML en tiempo de ejecución. Este es un ejemplo de XML de Layout:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:orientation="vertical" >
    <TextView android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a TextView" />
    <Button android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello, I am a Button" />
</LinearLayout>
```

A2.2. Android con Eclipse

A2.2.1. Instalación de Eclipse

El primer paso es descargarse la última versión estable de Eclipse IDE de la fundación Eclipse, se trata de una carpeta comprimida. Se extraerá el contenido en el lugar deseado de la máquina.

Para iniciar Eclipse solo habrá que hacer doble click en el ejecutable de la carpeta de instalación de eclipse (eclipse.exe). Al iniciar el programa preguntará por el directorio donde se desea colocar el ‘workspace’, se puede elegir el que se quiera. En este tutorial se ha dejado el directorio por defecto. Una vez se haya instalado correctamente Eclipse, el siguiente paso será obtener el Android SDK.

A2.2.2. Descarga Android SDK

Existen versiones disponibles del SDK de Android para Windows, Mac Os X y Linux. Para obtener las fuentes habrá que accede a <http://developer.android.com/sdk/index.html> y descargarse la última versión del SDK para la plataforma que se vaya a utilizar.

El Android SDK es un fichero comprimido, se descargará y se descomprimirá en el directorio que se desee.

Una vez se tenga instalado Eclipse y descargado el SDK es el momento de instalar el plugin en Eclipse para desarrollar en Android.

A2.2.3. Instalar Android Developer Tools (ADT)

Los pasos a seguir para instalar el plugin ADT en Eclipse son los siguientes:

1. Una vez iniciado Eclipse, se deberá acceder a **Help > Install New Software** y se accederá una pantalla como esta:

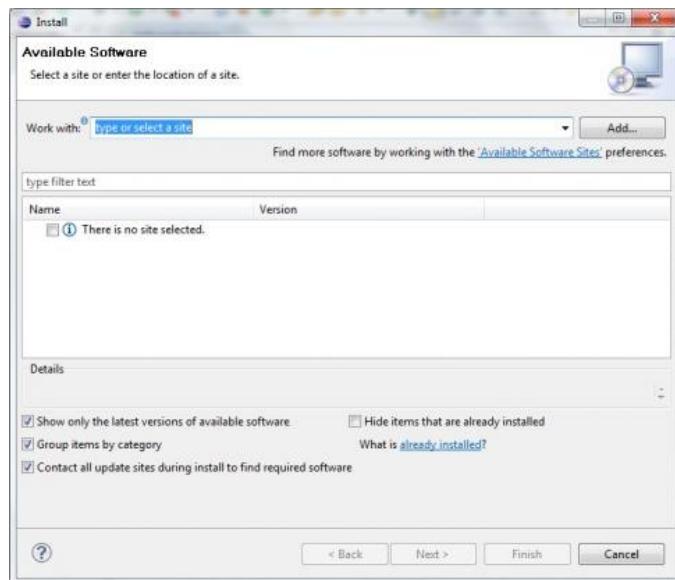


Figura 77. Descarga ADT 1

2. Se pulsará a añadir (Add) y se obtendrá una pantalla como la que se mostrará a continuación. En ella habrá que llenar el nombre, donde se pondrá lo que se desee, y en el campo location habrá que seleccionar la url que habilita Google para la descarga del plugin (<https://dl-ssl.google.com/android/eclipse>).

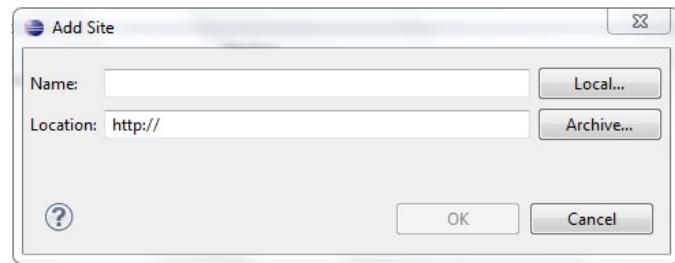


Figura 78. Descarga ADT 2

3. Una vez seleccionada la url del plugin se pulsará en siguiente y en finalizar. Entonces la descarga del plugin comenzará.

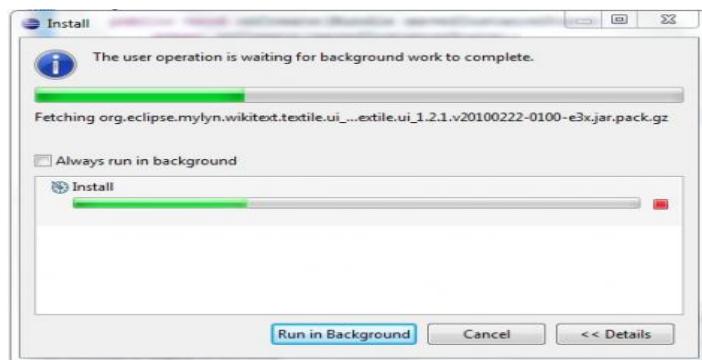


Figura 79. Descarga ADT 3

4. Cuando termine se pedirá reiniciar Eclipse. El plugin ya estará descargado.

A2.2.4. Configuración del plugin ADT

Una vez reiniciado Eclipse, habrá llegado la hora de conectar el plugin al SDK que se ha descargado anteriormente.

Para ello habrá que seleccionar en Eclipse el menú **Windows > preferences**. Se obtendrá la siguiente pantalla:

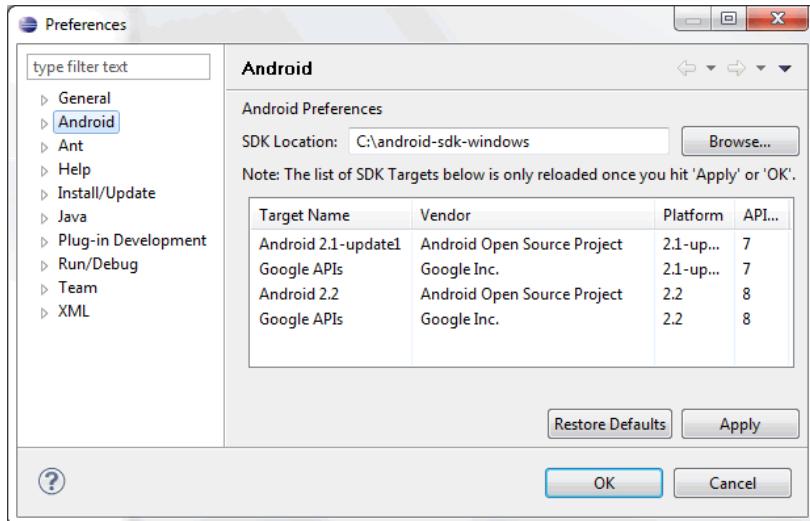


Figura 80. Configuración ADT Eclipse

En esta pantalla se seleccionará el menú “Android” de la parte izquierda de la pantalla y en el campo de SDK Location habrá que introducir el directorio donde se ha descomprimido el SDK anteriormente descargado. Por último se aceptarán los cambios pulsando OK.

Una vez se haya especificado la ubicación del SDK habrá otras acciones necesarias para tener correctamente configurado el plugin:

- En la sección build habrá una opción para que los recursos se recompilen automáticamente, habrá que comprobar que esta opción esté activada.
- DDMS (Dalvik Debug Monitor Service), que se utiliza para hacer debug de las aplicaciones. Existen varios niveles de log y de debug, con los valores por defecto debería valer pero se pueden cambiar si se considera necesario.

A2.2.5. Creación de Android Virtual Devices (AVDs)

A su vez para poder emular las aplicaciones que se vayan desarrollando, ADT dispone de una herramienta de creación de emuladores para el testeo de estas aplicaciones.

Para crear un nuevo AVD habrá que acceder a la opción “Android SDK and AVD Manager” de Eclipse. Una vez allí se seleccionará la opción Virtual Devices y en la pantalla que aparece se seleccionará “New...” para crear uno nuevo.

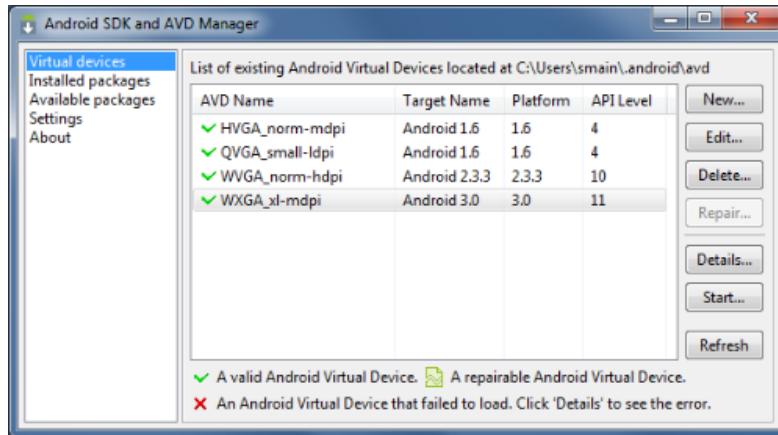


Figura 81. Configuración AVD

En la siguiente pantalla se configurarán los parámetros del nuevo AVD: nombre, versión del sistema operativo, especificaciones hardware y densidad de pantalla. Si se juega con estos parámetros se podrá emular cualquier dispositivo Android, herramienta muy útil para probar aplicaciones en diferentes dispositivos.

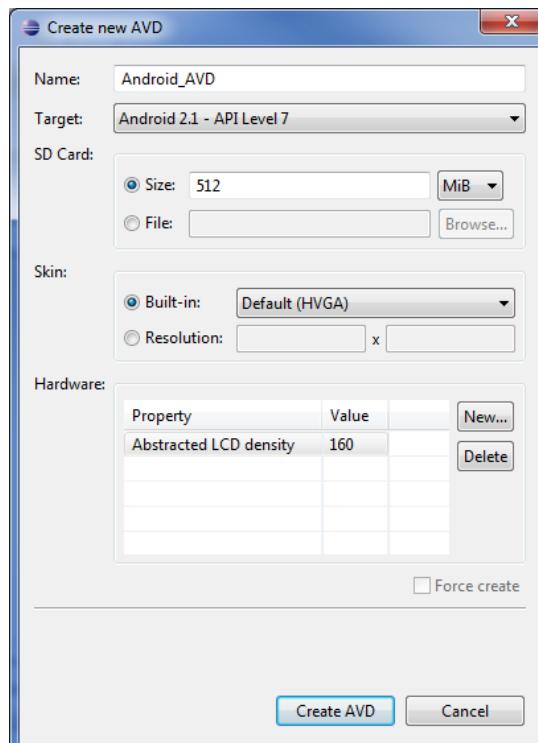


Figura 82. Configuración AVD 2

A2.2.6. Creación de un proyecto Android

Tras instalar todo lo necesario en el entorno de desarrollo y habiendo introducido los conceptos básicos de la programación Android ya se puede comenzar con un proyecto.

El ADT (Android Development Tools) provee un wizard (una ayuda) para crear proyectos Android de una manera rápida y sencilla, para poder crear el primer proyecto se deberán seguir los siguientes pasos: **File>New>Project>Projects...>Android > Android Project**. Tras estos pasos se obtendrá una pantalla como esta. Donde los campos a llenar serán los siguientes:

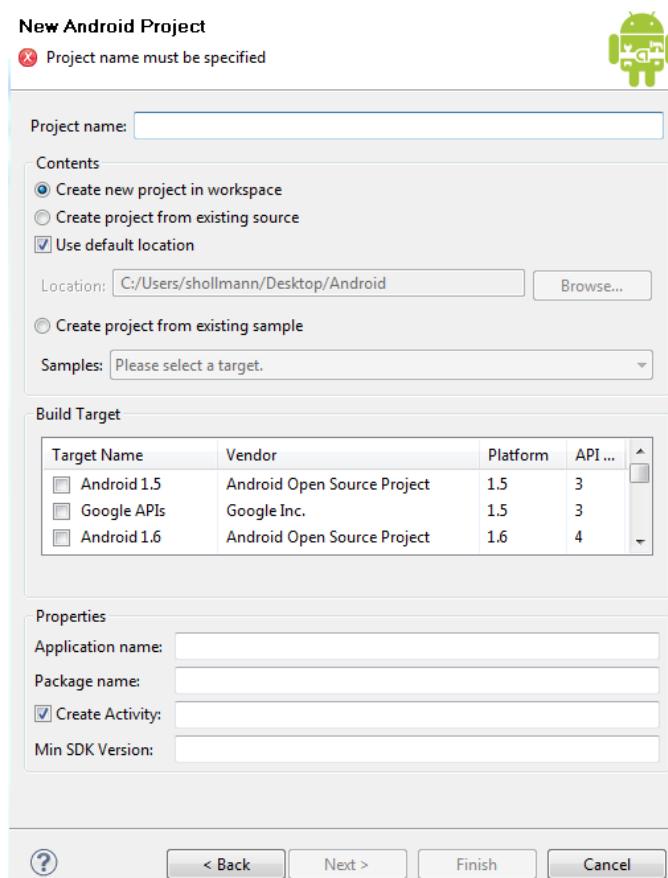


Figura 83. Creación Proyecto Android

- **Project Name:** Será el nombre del proyecto. Por ejemplo, Mi Proyecto Android.
- **Build Target:** Es la versión de Android para la cual se va a crear la aplicación (es decir, en la que se basará el testing y la aceptación del programa). Es recomendable marcar siempre la última versión de Android disponible.
- **Application Name:** Es el nombre de la aplicación, el que aparecerá en el dispositivo en que se ejecute. Por ejemplo: Hola Mundo App.

- **Package Name:** Es el nombre que recibirá el paquete de la aplicación (un paquete a grandes rasgos, es un conjunto (o “paquete”) de archivos). Por convención se utiliza la URL invertida de la empresa/persona que desarrolla la aplicación (exista o no en la realidad). Por ejemplo: com.empresa.holamundoapp.
- **Create Activity:** Es el nombre que recibirá la Activity principal de la aplicación. Por convención al final del nombre se coloca la palabra Activity. Por ejemplo: PantallaPrincipalActivity.
- **Min SDK Version:** Se debe ingresar un valor numérico que representa a la versión de Android a partir de la cual la aplicación será compatible.

Una vez se hayan llenado todos los campos, se presionará el botón Finish y ya se podrá disfrutar de su primer proyecto Android. Al crear un proyecto, se crea por default un Hello World!, por lo que sobre esta base se podrá comenzar a construir la nueva aplicación.

DESARROLLO DE UNA APLICACIÓN MÓVIL ORIENTADA A LA GEOLOCALIZACIÓN DE RECURSOS EN ANDROID

“DISFRUTANDO SAN FERMÍN”

Ingeniería Informática
Andrés Uriz Labiano
Eduardo Alfaro Larragueta
Pamplona,
15 de Noviembre de 2011



ÍNDICE DE CONTENIDOS

1. INTRODUCCIÓN:

- ESTADO DEL ARTE
- EMPRESA

2. OBJETIVOS

3. DESCRIPCIÓN DEL PROBLEMA

4. METODOLOGÍA DE TRABAJO: SCRUM

5. IMPLEMENTACIÓN:

- FUNDAMENTOS DE ANDROID
- PANTALLA POR PANTALLA

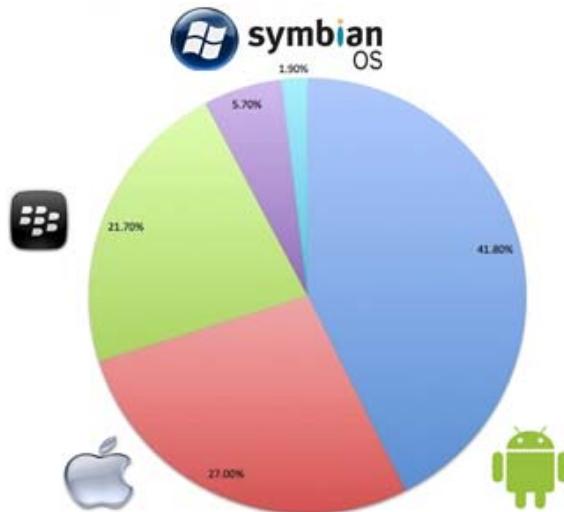
6. CONCLUSIONES

7. LÍNEAS FUTURAS

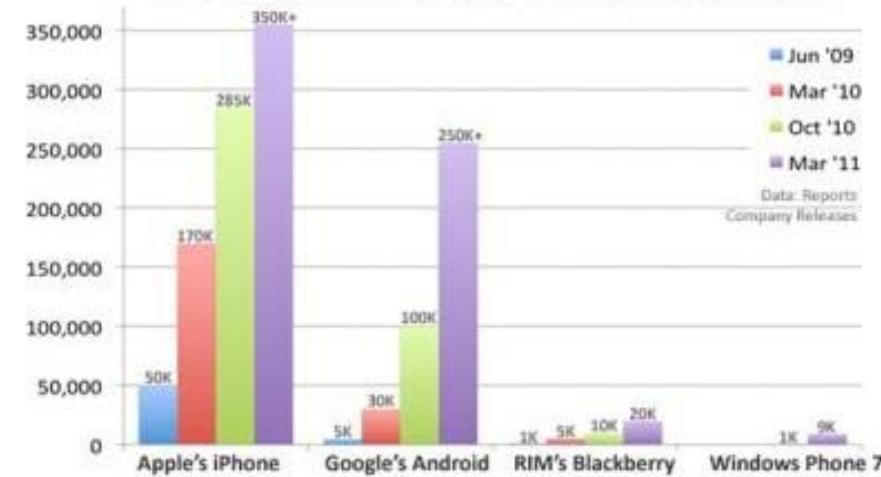


INTRODUCCIÓN: ESTADO DEL ARTE

- Últimos años, gran desarrollo plataformas móviles.
 - Amplio mercado de aplicaciones para SmartPhones.
 - Nuevas prestaciones: Acceso a red, geolocalización...
 - Nuevas posibilidades de ofrecer soluciones útiles.



upna Distribución mercado smartphones
Universidad Pública de Navarra
Universitatea Publikoa
Todas los derechos reservados
Eskubide guztiek erresalbatu dira



Evolución markets de aplicaciones móviles

INTRODUCCIÓN: EMPRESA

- **Alestis Consulting** es una empresa que elabora soluciones técnicas innovadoras en las que:
 - Integración de tecnologías del ámbito de la telemática.
 - Metodologías ágiles para la planificación.
 - Desarrollos Web, Google Maps, Soluciones Móvil...



- Necesidad de darse a conocer en el creciente sector de las aplicaciones móviles, San Fermín: gran escaparate.

OBJETIVOS

1. Realización de una aplicación Android que:
 - Permita acceder a toda la información de San Fermín 2011 de forma fácil y rápida.
 - Ofrezca información georreferenciada, aprovechar nuevas funcionalidades de los smartphones.
 - Sea útil al mayor número de usuarios posible, multi idioma.
2. Promocionar a la empresa en el sector de las aplicaciones móviles.
3. Ganar experiencia en el sector y las plataformas.



DESCRIPCIÓN DEL PROBLEMA

- La aplicación a realizar deberá permitir al usuario la realización de las siguientes acciones:
 - Consultar el programa completo de las fiestas de San Fermín 2011.
 - Consultar los eventos que están teniendo lugar en este momento.
 - Buscar eventos por categorías.
 - Acceder a información sobre los servicios que se ofrecen en las fiestas.
 - Acceder a los números de teléfono más relevantes y útiles.
 - Acceder a toda la información sobre los encierros.
 - Mantenerse al tanto de las últimas novedades en Twitter sobre San Fermín.
 - Seleccionar el idioma en que desea utilizar la aplicación.



METODOLOGÍA DE TRABAJO: SCRUM

- Este proyecto se ha incluido en la metodología ágil Scrum llevada a cabo en la empresa.
- Componentes:
 - Equipo de trabajo: autogestionado y multifuncional.
 - Scrum master: organiza al equipo.
 - Product Owner: dirige al equipo en la dirección correcta.
- Los proyectos se organizan en iteraciones acotadas en el tiempo llamadas Sprints.
- Actividades:
 - Reunión de planificación.
 - Daily Scrum.
 - Reunión de revisión.
 - Reunión de retrospectiva.



METODOLOGÍA DE TRABAJO: SCRUM

HISTORIAS DE USUARIO

Story ID	Proyecto	Historia de Usuario	Prioridad
1	Disfrutando San Fermín	Análisis y Diseño	100
2	Disfrutando San Fermín	Diseño interfaces	100
3	Disfrutando San Fermín	Geolocalización	100
4	Disfrutando San Fermín	Pantalla carga y principal	100
5	Disfrutando San Fermín	Programa	90
6	Disfrutando San Fermín	¿Qué hacer ahora?	90
7	Disfrutando San Fermín	Buscador	90
8	Disfrutando San Fermín	Servicios	80
9	Disfrutando San Fermín	Teléfonos de Interés	80
10	Disfrutando San Fermín	Encierro	80
11	Disfrutando San Fermín	Twitter	80
12	Disfrutando San Fermín	Análisis plataforma Multi-idioma	70
13	Disfrutando San Fermín	Introducción datos	70
14	Disfrutando San Fermín	Adaptación todas pantallas	70

Días persona estimados: **37.5**



IMPLEMENTACIÓN: FUNDAMENTOS ANDROID

- Programación multicapa:
 - Capa de modelo de negocio:
 - Clases Activity: Controlan una pantalla.
 - Clases Service: Corren en Background.
 - Clase Application: Clase Global.
 - Capa de interfaz: XML que definen UI.
 - Layouts: LinearLayout, RelativeLayout, ScrollView...
 - Widgets: Button, EditText, TextView, RadioButton...
 - Capa de datos: Clases Java.
- Aplicación: Conjunto de componentes atómicos unidos por Intents en tiempo de ejecución.



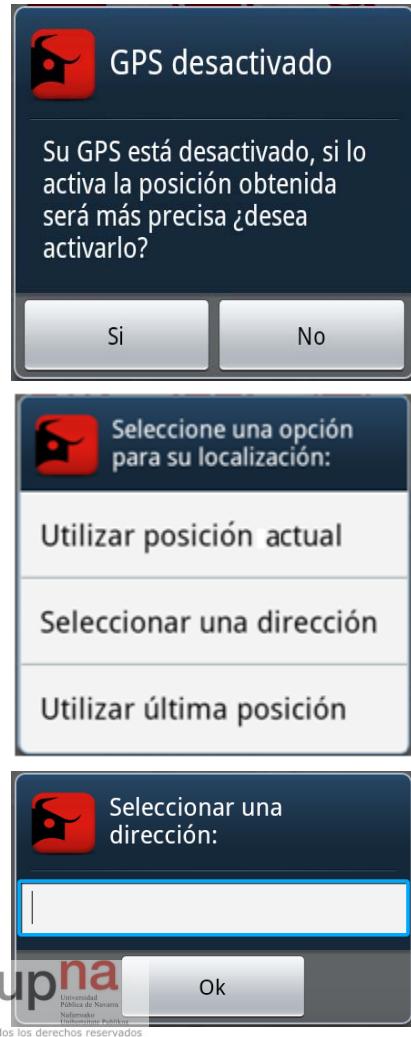
IMPLEMENTACIÓN: PANTALLA INICIAL DE CARGA



- Presentación aplicación.
- Carga de datos en clase Aplicación.
- Desde XML locales para minimizar acceso a red.



IMPLEMENTACIÓN: GEOPOSICIONAMIENTO



- Dialogo Activación GPS.
- Dialogo opciones localización:
 - Posición actual:
 - Análisis medios disponibles (GPS, Red).
 - Búsqueda posición con LocationListener.
 - Utilizar dirección:
 - Dialogo para introducir posición.
 - Petición a Google Geocoder.
 - Utilizar última posición:
 - Coger posición de las preferencias de la aplicación.
- Guardar la posición en la clase Aplicación.

IMPLEMENTACIÓN: PANTALLA PRINCIPAL



- Menú principal desde al que acceder a todas las secciones.
- Dificultad en el diseño de la interfaz: diferentes pantallas, efectos...
- Dar funcionalidad a todos los iconos para lanzar un Intent a la pantalla correspondiente.



IMPLEMENTACIÓN: PROGRAMA

Programa: selección de día



Intent(int Dia)



Programa: selección de evento



Intent(int idEvento)



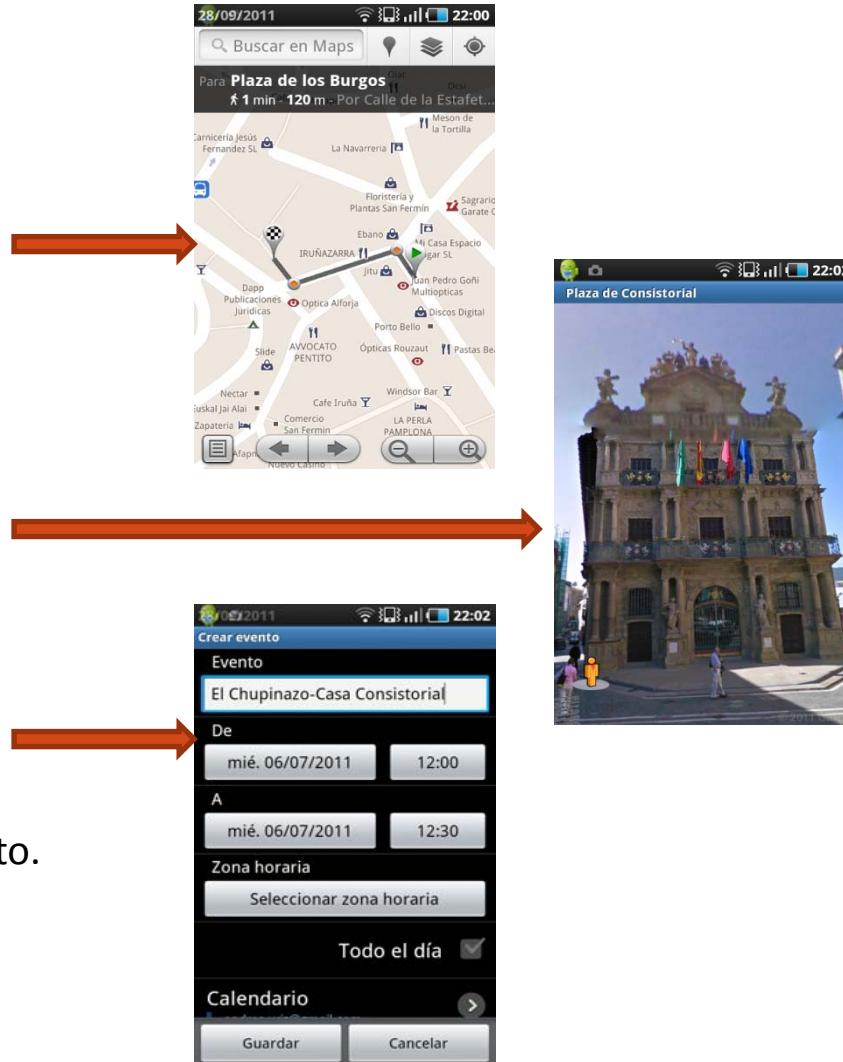
- Rellenar ListView(CustomAdapter).
- Asociar evento onClick a cada fila.

- Coger eventos del día clase aplicación.
- Separados en mañana, tarde y noche.
- Una Actividad por pestaña.
- Asociar evento onClick a cada fila.

IMPLEMENTACIÓN: PROGRAMA

Programa: ficha de evento

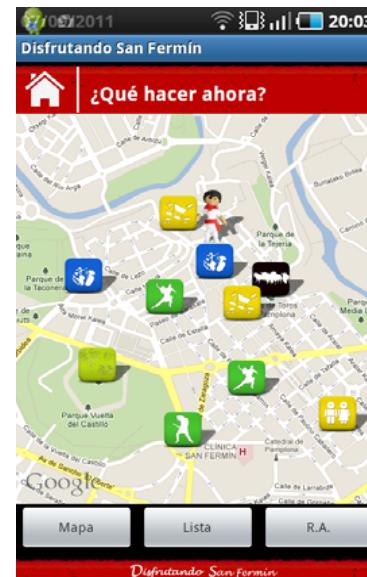
Intent(int idEvento)



- Coger y mostrar información del evento.
- Dar funcionalidad a los botones.
- Llamadas al sistema (Intents).

IMPLEMENTACIÓN: ¿QUÉ HACER AHORA?

Lista de eventos activos



- Calcular eventos activos.
- Mostrar en la lista.
- Guardar en clase aplicación.

Mapa de eventos activos

- Coger Eventos calculados anteriormente.
- Mostrar en el mapa.
- Evento onTap markers.

R.A. de eventos activos

- Coger Eventos calculados anteriormente.
- Llamar a wikitude utilizando su API.



IMPLEMENTACIÓN: BUSCADOR DE EVENTOS

Buscador: selección de categoría



Intent(String cats)

Buscador: resultados



- Coger Categorías de eventos.
- Rellenar ListView(CustomAdapter).
- Evento del botón, pasar String idsCategorías en el Intent.

- Coger idsCategorías del Intent.
- Buscar los eventos de esas categorías.
- Rellenar ListView(CustomAdapter).
- Asociar evento onClick a cada fila.

IMPLEMENTACIÓN: SECCIÓN SERVICIOS

Servicios: selección de categoría



Intent(String cats)

Servicios: lista de resultados



Ir a Mapa

Ir a R.A.

- Coger Categorías de servicios.
- Rellenar ListView(CustomAdapter).
- Evento del botón, pasar String idsCategorías en el Intent.

- Coger idsCategorías del Intent.
- Buscar los servicios de esas categorías.
- Calcular distancias.
- Rellenar ListView(CustomAdapter).
- Eventos de los botones.



IMPLEMENTACIÓN: SECCIÓN SERVICIOS

Servicios: mapa



Ir a Mapa



Servicios: ficha de servicio



Intent(intent idServicio)



Servicios: R.A.



Ir a R.A.



- Coger y mostrar información del servicio.
- Dar funcionalidad a los botones.
- Llamadas al sistema (Intents).



IMPLEMENTACIÓN: TELÉFONOS DE INTERÉS

Teléfonos de interés: Lista



Teléfonos de interés: Confirmación

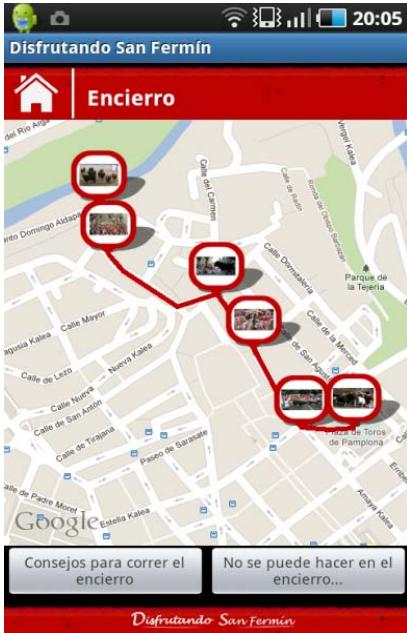


- Coger Teléfonos de clase Aplicación.
- Rellenar ListView(CustomAdapter).
- Evento onClick a cada fila.
- Lanzar AlertDialog.
- Llamar: Intent Call al sistema.



IMPLEMENTACIÓN: ENCIERRO

Encierro: recorrido



Encierro: Consejos y Prohibiciones



- Pintar Array de puntos en mapa.
 - Situar markers con fotos.
 - Evento marker: ampliar foto.
- Eventos botones.**

- Cargar texto html desde XML.

IMPLEMENTACIÓN: TWITTER Y CONFIGURACIÓN

Twitter



Configuración



- Llamada a la url de búsqueda del hashtag #SanFermin en Twitter.

- Mostrar AlertDialog.
- Rellenar Datos.
- Eventos onClick radio button.
- Reinicio con nuevo idioma.



CONCLUSIONES

- Resultados:
 - Planificación Scrum: Estimado 37.5, Real 38.75.
 - Cumple requisitos funcionales, de interfaz y de sistema.
 - Superadas pruebas de sistema y validaciones en la reunión de revisión.
 - En el Android Market antes de la fecha límite.
 - 3621 descargas a fecha de 18 de Julio.
- Desarrollo exitoso, superó todas las pruebas.
- Terminada en plazo.
- Número importante de descargas.
- Conocimientos adquiridos.



LÍNEAS FUTURAS

- Nuevos sistemas operativos móviles: iPhone, Windows Phone7, BlackBerry...
- Nuevas mejoras tanto en funcionalidad como en funcionamiento:
 - Permitir actualizaciones de los datos de la aplicación.
 - Twitter más peso: twits georreferenciados, permitir login, utilizar API twitter.
 - Potenciar la información en tiempo real y la interacción entre los usuarios: comentarios y puntuaciones de eventos, información de los encierros...

