



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

CAMPUS D'ALCOI

Aplicación Android para Grecia: El juego de rol

MEMORIA PRESENTADA POR:

Juan Carlos Morelló Gandía

GRADO DE INGENIERÍA INFORMÁTICA

Convocatoria de defensa: Noviembre de 2015

Introducción

En este primer capítulo se introducen los aspectos básicos tratados a lo largo de la memoria y las motivaciones que han animado al planteamiento y desarrollo del proyecto, así como un pequeño resume de sus partes para situar al lector y permitirle una mejor comprensión del escrito.

Contexto

Hoy en día, más de la mitad de la población española de más de 18 años dispone de un Smartphone, Tablet o ambos. En este marco hay que destacar el sistema operativo Android, presente en el 90% de los Smartphone vendidos en el mes de julio de 2015 [1].

Motivación

Con este desarrollo se busca la adquisición de conocimiento acerca del sistema operativo Android, una plataforma que desde sus inicios se presentaba tan innovadora como arriesgada ya que por aquel entonces iOS estaba dando sus primeros pasos y Symbian gozaba de bastante éxito.

Una de las motivaciones principales de programar para este sistema operativo es que se trata de un proyecto de código abierto, concretamente se llama AOSP (Android Open Source Project) un proyecto en el que colabora la comunidad de usuarios y que ofrece una alternativa a la plataforma como tal, aunque la propia Google es la principal responsable de la gestión y desarrollo del proyecto. AOSP podría considerarse como una plataforma abierta en sí misma, ya que además del sistema operativo móvil ofrece algunas aplicaciones nativas que permiten por ejemplo reproducir música, realizar y recibir llamadas o sacar fotos con el móvil o Tablet que estamos manejando.

El lenguaje de programación utilizado

Otra motivación que decanta la balanza hacia la elección de este proyecto es la importancia de java, uno de los lenguajes de programación más extendidos en el mundo y que actualmente se encuentra bajo la licencia de GPL de GNU [2].

Objetivos

El objetivo principal de este TFG es el desarrollo de una aplicación Android para Smartphone, que sirva como referencia y ayuda para el juego de rol Grecia, esto conlleva una serie de metas planteadas y explicadas a continuación:

- Debido a que este trabajo se centra en el desarrollo de una aplicación para Android, el primer objetivo que se persigue es obtener un conocimiento global sobre este sistema operativo. Para ello se estudiarán las distintas plataformas de desarrollo, sus fundamentos de programación, como ejecuta las aplicaciones y las herramientas disponibles y necesarias para su desarrollo.
- Otro objetivo es conocer la plataforma Google Play, que será desde donde se podrá descargar esta aplicación. Cuáles son sus condiciones, servicios, etc...
- Como último objetivo a mencionar en este apartado, se encuentra la práctica de la creación completa de una aplicación desde sus inicios hasta sus etapas finales, aplicando todo el saber útil aprendido a lo largo del grado.

Tabla de contenido

Introducción	1
Contexto	2
Motivación.....	2
Objetivos	2
Tabla de contenido.....	3
Estado del arte	6
Contexto actual del desarrollo de aplicaciones	7
Aplicaciones similares.....	8
Envenenado	8
Grimorios Aquelarre	9
Android	11
Introducción	12
Los orígenes.....	12
Características	12
Arquitectura	14
Núcleo Linux.....	15
Runtime de Android.....	15
Bibliotecas nativas	16
Entorno de aplicación	16
Aplicaciones	16
Elementos de un proyecto de Android	16
Componentes de una aplicación.....	17
View	17
Layout	17
Activity	17
Service.....	17
Intent	17
Librería Zxing.....	18
Concepto	19
Utilidad	19
Contenido de un código QR	19
Motivación.....	20
Grecia: el juego de rol	21

Introducción	22
Qué es un juego de rol.....	22
Libros	22
Grecia: el juego de rol – Libro básico	23
El Inframundo	24
Objetos Divinos	25
Vademécum	26
Motivación.....	27
Herramientas de trabajo	28
Herramientas de diseño	29
Herramientas de desarrollo.....	29
Android studio	29
Bluestacks	29
Adobe Photoshop	30
Interfaz gráfica de usuario.....	32
Contexto	33
Pantalla principal	33
Menú lateral	34
Pantalla de bendiciones.....	34
Pantalla de técnicas de combate	35
Pantalla de ventajas y desventajas	35
Pantalla de Lanzar dados	36
Pantalla de códigos QR	38
PANTALLA DEL ORÁCULO	39
Recursos multimedia	40
Recursos gráficos	40
Recursos de audio	40
Implementación de funcionalidades	42
Navigation Drawer.....	43
Código extraído de Principal.java :	43
Código extraído de activity_principal.xml :.....	44
LISTVIEW y LISTVIEW ADAPTER	45
Código extraído de activity_TECNICAS.xml	45
Código extraído de items_layout_tecnicas.xml	45
Código extraído de subitems_layout_tecnicas.xml	46

codigo extraido de ListViewAdapter.java	47
Código extraído de AdapterTecnicas.java	48
Código extraído de Tecnicas.java	50
Lector QR	51
Código referente al paso 4, extraído de activity_qrlector.xml	52
Código referente al paso 5, extraído de QRlector.java	53
RANDOM	57
Vamos a ver su implementación en Delfos.Java (Generador del oraculo).....	57
GOOGLE PLAY.....	59
Que es Google Play	60
Convertirse en Desarrollador.....	60
Creando una nueva aplicación.....	61
Clasificación del contenido.....	62
Precio y distribución.....	62
Pestañas de estadísticas.....	63
Motivación.....	64
Evaluación de la aplicación.....	65
Cuestionario SUS	66
Respuestas del cuestionario	67
Creo que me gustará utilizar esta aplicación con frecuencia	68
Encontré la aplicación innecesariamente compleja	68
Pensé que la aplicación sería más fácil de usar	69
Creo que voy a necesitar el apoyo de un experto para utilizar la aplicación.....	69
Me pareció que las diversas funciones de esta aplicación se integran bien.....	70
Pienso que hay muchos aspectos mejorables en esta aplicación	70
La mayoría de la gente podrá utilizar esta aplicación sin problemas	71
Encontré la aplicación difícil de usar	71
Me sentí seguro utilizando la aplicación	72
Necesito aprender muchas cosas antes de manejarme bien con la aplicación	72
Conclusiones y propuestas	73
Conclusiones.....	74
Trabajo futuro y posibles ampliaciones	75
Manual de instalación en Android	79
Instalación desde Google Play.....	79
Instalación desde descarga Directa - APK.....	80

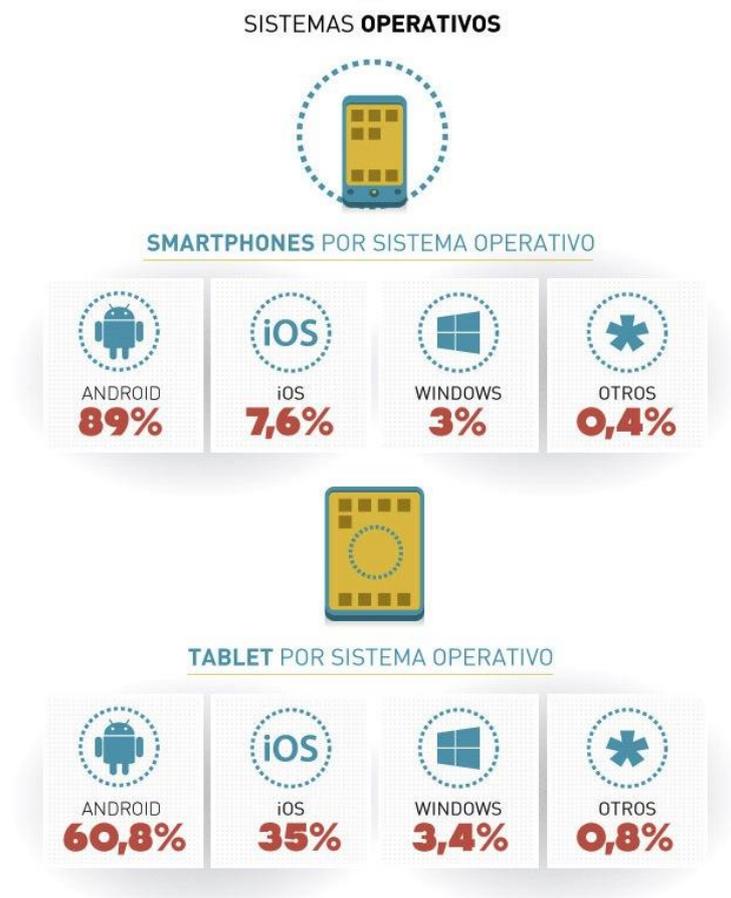
Estado del arte

En este segundo capítulo se realizará un análisis sobre el contexto actual en que nos encontramos al inicio de este proyecto, desde el punto de vista de desarrolladores. Además se expondrán aplicaciones similares existentes para Android que han servido como ejemplo para el desarrollo de la aplicación que deseamos realizar. El objetivo es ver las similitudes de las mismas con nuestra aplicación en cuestión, para así aprovechar sus bondades y solucionar sus posibles debilidades (si las tuviera).

Contexto actual del desarrollo de aplicaciones

Para decantarse sobre la plataforma en la cual se desarrollará el proyecto hay que realizar un análisis de mercado previo, para ver el éxito y la popularidad de cada sistema operativo, ya que una de las principales motivaciones para la distribución de una app debe de ser el máximo número de clientes que podamos tener.

Para eso nos vamos a fijar en el informe sobre el uso de apps en España previamente analizado en una página web. [3]



En esta infografía se puede observar que Android es la plataforma que tiene más usuarios en Smartphones, concretamente con el 89%, y en Tablets, aunque con menor diferencia con sus competidores.

En conclusión, atendiendo a los datos reflejados en este estudio hemos decidido apostar por Android como plataforma destino de nuestro producto, debido a que este sistema operativo está presente en la mayoría de dispositivos móviles, por lo que podemos llegar a una mayor cantidad de usuarios.

También es importante añadir que soy propietario de un Smartphone y una Tablet, ambos con el sistema operativo Android.

Aplicaciones similares

Actualmente existen varias aplicaciones que sirven como apoyo para los juegos de rol, incluyendo en la mayoría de ellas información resumida para los jugadores y el director de juego, información que se puede encontrar en libros, muchas veces de gran tamaño y dispersa (incluso en varios libros).

Estas aplicaciones funcionan como referencia rápida en las mesas de juego, ya que habitualmente se dispone tan solo de un libro y es probable que todos los jugadores necesiten consultarlo en algún momento de la partida. Para agilizar este proceso se crean estas aplicaciones de consulta, para no entorpecer el desarrollo del juego.

En los juegos de rol existe siempre un componente de azar bastante importante, reflejado habitualmente con tiradas de dados de varias caras, que se puede implementar fácilmente en una app con elementos aleatorios que nos proporcionan los lenguajes de programación.

Envenenado

Envenenado es una aplicación para el juego de rol del mismo nombre, editado por Conbarba y creador por Vincent D. Baker [4].



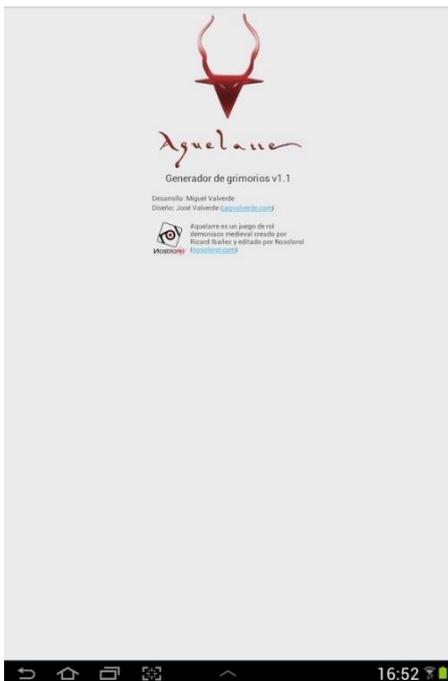
Esta aplicación sirve para crear contenido para el juego, como personajes o enemigos, de forma rápida e intuitiva indicando las opciones que queremos en un formulario de aspecto amigable. Cuando tengamos seleccionadas todas las opciones generaremos el contenido y se nos permite la opción de convertir el resultado a formato PDF.

APLICACIÓN ANDROID PARA GRECIA: EL JUEGO DE ROL



Grimorios Aquelarre

Grimorios Aquelarre es una aplicación para Android basada en el primer juego de rol español editado en nuestro país [5].



Esta aplicación permite generar un grimorio (libro de hechizos) aleatorio, basado en unas pocas indicaciones introducidas por el usuario, el contenido del grimorio se genera aleatoriamente.

APLICACIÓN ANDROID PARA GRECIA: EL JUEGO DE ROL



Android

Puesto que esta aplicación ha sido desarrollada para el sistema operativo Android, se debe realizar una especial mención al mismo, describiéndolo y explicando en profundidad en sus características, funcionalidades y arquitectura.

Introducción

Android es un sistema operativo basado en el núcleo Linux. Fue diseñado principalmente para dispositivos móviles con pantalla táctil, como teléfonos inteligentes o tabletas; y también para relojes inteligentes, televisores y automóviles

Los orígenes

En octubre de 2003, en la localidad de Palo Alto, Andy Rubin, Rich Miner, Chris White y Nick Sears fundan Android Inc. con el objetivo de desarrollar un sistema operativo para móviles basado en Linux. En julio de 2005, la multinacional Google compra Android Inc. Android fue presentado en 2007 junto la fundación del Open Handset Alliance (un consorcio de compañías de hardware, software y telecomunicaciones) para avanzar en los estándares abiertos de los dispositivos móviles. El primer móvil con el sistema operativo Android fue el HTC Dream y se vendió en octubre de 2008.

Características

Las características y especificaciones de la versión Lollipop 5.1 son [6]:

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

APLICACIÓN ANDROID PARA GRECIA: EL JUEGO DE ROL

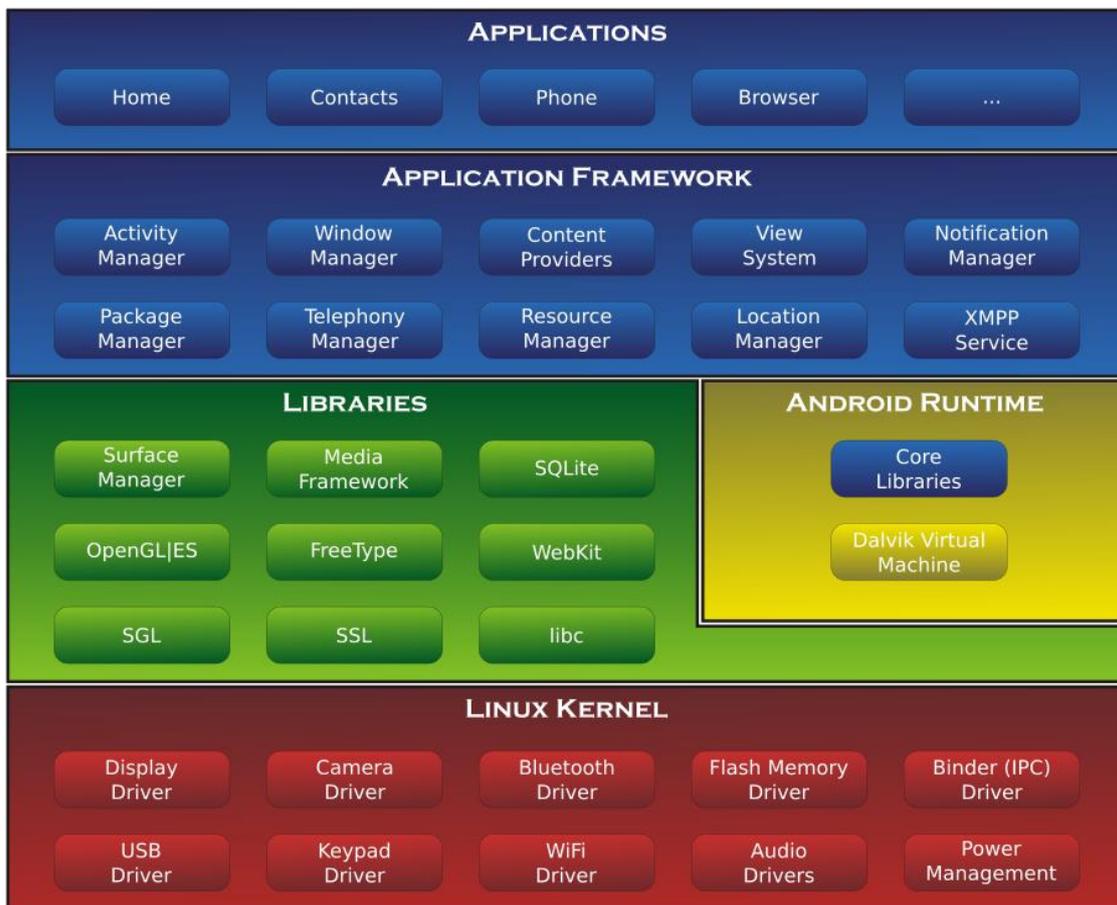
	diseñada específicamente para Android y optimizada para dispositivos móviles que funcionan con batería y que tienen memoria y procesador limitados. El soporte para J2ME puede ser agregado mediante aplicaciones de terceros como el J2ME MIDP Runner. ⁵⁰
Soporte multimedia	Android soporta los siguientes formatos multimedia: WebM , H.263 , H.264 (en 3GP o MP4), MPEG-4 SP , AMR , AMR-WB (en un contenedor 3GP), AAC , HE-AAC (en contenedores MP4 o 3GP), MP3 , MIDI , Ogg Vorbis , WAV , JPEG , PNG , GIF y BMP . ⁴⁹
Soporte para streaming	Streaming RTP/RTSP (3GPP PSS, ISMA), descarga progresiva de HTML (HTML5 <video> tag). Adobe Flash Streaming (RTMP) es soportado mediante el Adobe Flash Player. Se planea el soporte de Microsoft Smooth Streaming con el port de Silverlight a Android. Adobe Flash HTTP Dynamic Streaming estará disponible mediante una actualización de Adobe Flash Player.
Soporte para hardware adicional	Android soporta cámaras de fotos, de vídeo, pantallas táctiles, GPS, acelerómetros, giroscopios, magnetómetros, sensores de proximidad y de presión, sensores de luz, gamepad, termómetro, aceleración por GPU 2D y 3D.
Entorno de desarrollo	Incluye un emulador de dispositivos, herramientas para depuración de memoria y análisis del rendimiento del software. Inicialmente el entorno de desarrollo integrado (IDE) utilizado era Eclipse con el plugin de Herramientas de Desarrollo de Android (ADT). Ahora se considera como entorno oficial Android Studio , descargable desde la página oficial de desarrolladores de Android.
Google Play	Google Play es un catálogo de aplicaciones gratuitas o de pago en el que pueden ser descargadas e instaladas en dispositivos Android sin la necesidad de un PC.
Multi-táctil	Android tiene soporte nativo para pantallas capacitivas con soporte Multi-táctil que inicialmente hicieron su aparición en dispositivos como el HTC Hero. La funcionalidad fue originalmente desactivada a nivel de Kernel (posiblemente para evitar infringir patentes de otras compañías). ⁵¹ Más tarde, Google publicó una actualización para el Nexus One y el Motorola Droid que activa el soporte multi-táctil de forma nativa. ⁵²
Bluetooth	El soporte para A2DP y AVRCP fue agregado en la versión 1.5; el envío de archivos (OPP) y la exploración del directorio telefónico fueron agregados en la versión 2.0; ⁵⁴ y el marcado por voz junto con el envío de contactos

	entre teléfonos lo fueron en la versión 2.2.
Videollamada	Android soporta videollamada a través de Hangouts (ex- Google Talk) desde su versión HoneyComb.
Multitarea	Multitarea real de aplicaciones está disponible, es decir, las aplicaciones que no estén ejecutándose en primer plano reciben ciclos de reloj.
Características basadas en voz	La búsqueda en Google a través de voz está disponible como "Entrada de Búsqueda" desde la versión inicial del sistema. ⁵⁶
Tethering	Android soporta tethering, que permite al teléfono ser usado como un punto de acceso alámbrico o inalámbrico (todos los teléfonos desde la versión 2.2, no oficial en teléfonos con versión 1.6 o inferiores mediante aplicaciones disponibles en Google Play (por ejemplo PdaNet). Para permitir a un PC usar la conexión de datos del móvil Android se podría requerir la instalación de software adicional.

Arquitectura

En el siguiente gráfico vamos a mostrar las diferentes capas de las que se compone la arquitectura. Una peculiaridad de las mismas es que están basadas íntegramente en software libre.

APLICACIÓN ANDROID PARA GRECIA: EL JUEGO DE ROL



Núcleo Linux

Android depende de Linux para los servicios base del sistema como seguridad, gestión de memoria, gestión de procesos, pila de red y modelo de controladores. El núcleo también actúa como una capa de abstracción entre el hardware y el resto de la pila de software.

Runtime de Android

En esta capa está incluido un set de bibliotecas base que proporcionan la mayor parte de las funciones disponibles en las bibliotecas base del lenguaje Java. Cada aplicación Android corre su propio proceso, con su propia instancia de la máquina virtual Dalvik. Dalvik ha sido escrito de forma que un dispositivo puede correr múltiples máquinas virtuales de forma eficiente. Dalvik ejecuta archivos en el formato Dalvik Ejecutable (.dex), el cual está optimizado para memoria mínima. La Máquina Virtual está basada en registros y corre clases compiladas por el compilador de Java que han sido transformadas al formato.dex por la herramienta incluida "dx".

Bibliotecas nativas

Android incluye un conjunto de bibliotecas de C/C++ usadas por varios componentes del sistema. Estas características se exponen a los desarrolladores a través del marco de trabajo de aplicaciones de Android; algunas son: System C library (implementación biblioteca C estándar), bibliotecas de medios, bibliotecas de gráficos, 3D y SQLite, entre otras.

Entorno de aplicación

Gracias a esta capa, los desarrolladores tienen acceso a los APIs que usan las aplicaciones base. Su diseño permite que unas aplicaciones puedan utilizar componentes de otras.

Aplicaciones

Las aplicaciones base incluyen un cliente de correo electrónico, programa de SMS, calendario, mapas, navegador, contactos y otros. Todas las aplicaciones están escritas en lenguaje de programación Java.

Para el desarrollo de esta aplicación se ha optado por utilizar Java con Android Studio.

Elementos de un proyecto de Android

Un proyecto Android está formado por un descriptor de aplicación (AndroidManifest.xml), el código fuente en Java y unos ficheros de recursos. Cada elemento se organiza en un directorio específico:

Src	<i>Carpeta que contiene el código fuente Java.</i>
Gen	<i>Carpeta que contiene el código generado automáticamente por el SDK. Aquí se encuentra el fichero R.java, una clase que asocia los recursos de la aplicación con identificadores, para que dichos recursos puedan ser accesibles desde el código Java.</i>
Android Dependencies	<i>Librerías asociadas al proyecto.</i>
Bin	<i>Directorio donde se compila el código y se genera el .apk listo para instalar en un dispositivo.</i>
Res	<i>Contiene los recursos usados por la aplicación.</i>
Drawable	<i>Almacena ficheros de imágenes y descriptores de imágenes en xml.</i>

Layout	<i>Contiene los ficheros xml que definen la disposición de la interfaz de usuario de cada pantalla visual (activity o fragment).</i>
Menú	<i>Contiene ficheros xml con los menús de cada activity.</i>
Values	<i>Permite especificar valores de string, color o estilo. Además permite traducir la aplicación a otro idioma.</i>
Anim	<i>Contiene ficheros xml con animaciones Tween.</i>
Xml	<i>Otros ficheros xml requeridos por la aplicación.</i>

Componentes de una aplicación

En este último apartado del capítulo se describen de forma breve los componentes de una aplicación Android, los cuales aportan diferentes funcionalidades al desarrollo.

View

Son los elementos que componen la interfaz gráfica de una aplicación, por ejemplo: un botón, una etiqueta o una caja de texto. Pueden definirse en código Java o en un fichero XML.

Layout

Se trata del agrupamiento de un conjunto de vistas para organizarlas y construir una interfaz clara y agradable.

Activity

Son lo que coloquialmente se conoce como “pantalla de una aplicación”. Constituyen la interfaz de usuario, la cual suele tener varias “Activities” independientes, pero que trabajan para un objetivo común.

Service

Son procesos que se ejecutan en segundo plano (similar a los “demonios” de UNIX) sin necesidad de interacción con el usuario.

Intent

Se refiere a la voluntad de realizar alguna acción, como por ejemplo lanzar una “Activity” o un “Service”.

Librería Zxing

En esta aplicación se ha utilizado una librería externa llamada Zxing, creo conveniente explicar a grandes rasgos en que consiste y los motivos por los cuales ha sido necesaria en este proyecto.

Concepto

ZXing es una librería procesadora de imágenes multi-formato en 1D/2D y de código abierto. Actualmente es capaz de reconocer los formatos UPC-A, UPC-E, EAN-8, EAN-13, Códigos 39, 93, 128, ITF, Codabar, RSS-14 (en todas sus variantes), Matriz de datos (Data Matrix), Aztec, PDF 417 y por supuesto los populares códigos QR, muy usados desde que la tecnología móvil está presente en nuestras vidas.

Utilidad

La librería Zxing permite leer **códigos QR**, que son un módulo para almacenar información en una matriz de puntos o en un código de barras bidimensional.



Esto permite tan solo con una captura desde nuestra cámara integrada en el móvil, leer volúmenes de datos largos y engorrosos como podría ser una página web o una ubicación de google maps.

Aunque el desarrollo inicial de los Códigos QR tenía como objetivo principal su utilización en la industria de la automoción, hoy por hoy la posibilidad de leer códigos QR desde teléfonos y dispositivos móviles permite el uso de códigos QR en un sinnúmero de aplicaciones completamente diferentes de las que originales como pueden ser:

- Publicidad
- Campañas de marketing
- Merchandising
- Diseño Gráfico
- Papelería corporativa (tarjetas de visita, catálogos)
- Internet, Webs, blogs. Estos códigos se leen mediante un programa, habitualmente el *Barcode scanner* que se integra con la cámara de nuestro dispositivo Android y pueden contener información de varios tipos: enlaces a páginas web, contactos de teléfono, texto, o números

Contenido de un código QR

Un Código QR puede contener información variada y es una manera sencilla de interactuar con un dispositivo móvil y permitir realizar acciones automáticamente con el terminal como por ejemplo:

- Abrir la URL de una página Web o perfil social
- Leer un Texto
- Enviar un email
- Enviar un SMS
- Realizar un llamada telefónica
- Guardar un evento en la agenda
- Ubicar una posición geográfica en un google maps

Motivación

Zxing se integra perfectamente en Android Studio y se puede introducir información oculta en cualquier documento.

Como Grecia es un juego de rol publicado en papel y en pdf, la utilización de estos códigos QR en los libros, permite que los jugadores interactúen con ellos directamente desde su aplicación, obteniendo diferentes resultados en cada lectura, gracias a un generador aleatorio de contenido basado en una cadena de texto leída en el propio código.

Grecia: el juego de rol

Como esta aplicación está basada en el juego de rol Grecia, se ha decidido añadir un apartado explicando el juego y la necesidad de crear esta aplicación para mejorar sus partidas en mesa.

Introducción

Grecia: el juego de rol, es un juego de rol basado en la Grecia mitológica de licencia libre, auto editado y publicado por el mismo autor de este TFG, que se puede descargar gratuitamente desde el blog oficial [8] pero también existe la posibilidad de comprar los libros físicamente desde lulu.com o Amazon.

Qué es un juego de rol

Un juego de rol es un juego interpretativo-narrativo en el que los jugadores asumen el «rol» de personajes imaginarios a lo largo de una historia o trama en la que interpretan sus diálogos y describen sus acciones. No hay un guion a seguir, ya que el desarrollo de la historia queda por completo sujeto a las decisiones de los jugadores. Por esta razón, la imaginación, la narración oral, la originalidad y el ingenio son primordiales para el adecuado desarrollo de esta forma dramática. En cierto modo los juegos de rol son la versión adulta de los juegos de fantasía infantiles, como «policías y ladrones», «mamá y papá» o «indios y vaqueros», en los que los niños se imaginan ser un personaje que en realidad no son. Efectivamente, en los juegos de rol cada jugador interpreta un personaje ficticio, con una serie de características propias que lo definen. La interpretación del personaje no debe ser tan rigurosa como en una obra de teatro. En el rol no hay guiones por los cuales regirse. Cada jugador definirá el carácter de su personaje según sus propios criterios y, durante una partida, responderá a las diversas situaciones que le puedan surgir decidiendo en el momento las acciones de este personaje, es decir, improvisando. Así pues, una partida de rol no sigue un guion prefijado, sino que la «historia» se va creando con el transcurso de la partida. De forma similar al juego infantil, cuando un jugador anuncia «Ahora yo te disparo», el otro puede responder: «Y yo te lo esquivo». Corresponde al director de juego el decidir hasta qué punto debe quedar la partida en manos del azar, pudiendo intervenir en cualquier momento para reconducir la trama en una u otra dirección.

Otro aspecto que diferencia a los juegos de rol de otros juegos es que cada jugador interpreta a un personaje único y diferente, con personalidad y características distintas, dependiendo del deseo del jugador a la hora de crear su personaje, o de los requisitos del director de juego en el momento de repartirlos. El concepto básico es que los jugadores persiguen un fin común, y deben cooperar entre ellos (aunque a veces esta regla no se cumple), y los personajes pueden ser complementarios; unos destacan en habilidades y capacidades físicas, otros en intelectuales o sociales, y, si la temática del juego lo permite, otros pueden tener habilidades místicas (magia, milagros, etc.). Todas esas características se indican en una hoja de personaje, que varía según el sistema de juego [7].

Libros

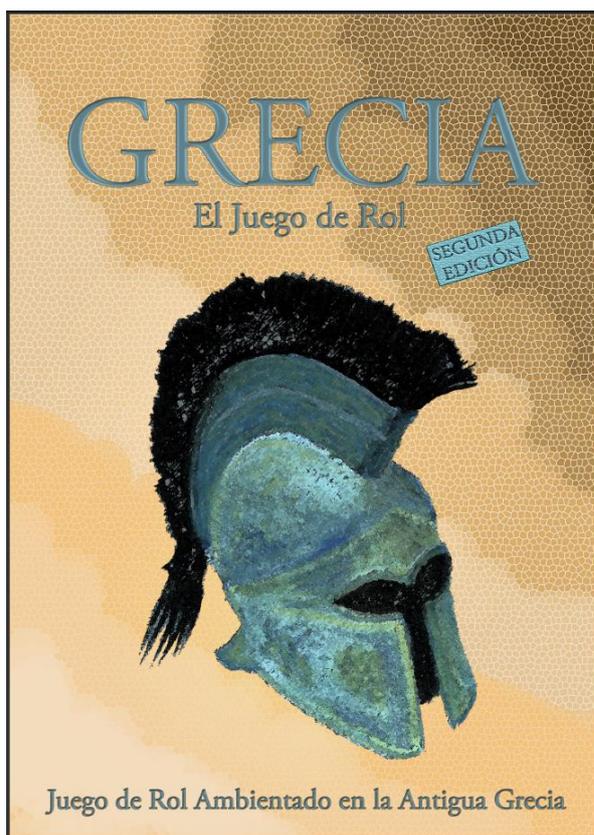
Actualmente Grecia: el juego de rol se encuentra en su segunda edición y tiene, además, una serie de suplementos publicados en pdf y físico, que se pueden obtener con la misma licencia y de la misma manera que su libro básico.

Grecia: el juego de rol – Libro básico

Grecia es un juego de rol donde los jugadores se meterán en el papel de héroes de la Grecia mitológica que se encomendaban a dioses como Zeus, Poseidón o Apolo, para alcanzar la fama e inmortalidad.

En su interior encontraremos además del reglamento del juego, reglas para crear héroes (representados por los jugadores), una descripción del mundo, equipo, bestiario y dos aventuras listas para jugar.

Publicado el 16 de julio de 2012.



Está editado en dos formatos, tapa blanda y rústica y contiene 127 páginas en blanco y negro.

Tapa blanda: <http://www.lulu.com/shop/juan-carlos-morell%C3%B3-gand%C3%ADa/grecia-el-juego-de-rol-tapa-blanda/paperback/product-20609979.html>

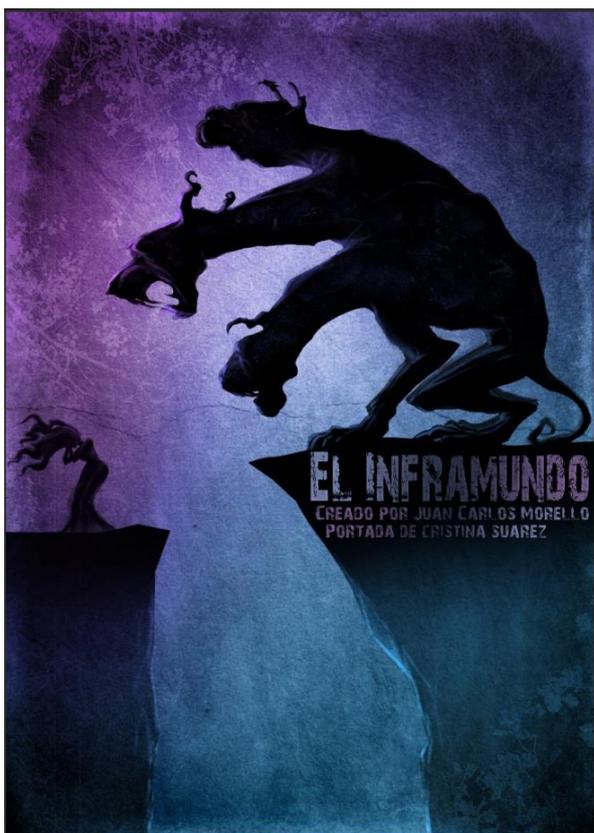
Tapa dura: <http://www.lulu.com/shop/juan-carlos-morell%C3%B3-gand%C3%ADa/grecia-el-juego-de-rol-tapa-dura/hardcover/product-20271209.html>

El Inframundo

El inframundo es la primera ampliación para la segunda edición del juego de rol Grecia. En este suplemento está ambientado en los dominios del dios oscuro Hades, que mantiene desde su fortaleza en equilibrio entre muertos y vivos. Además de Hades, en el inframundo lo habitan otras personalidades importantes como su mujer Perséfone, los tres jueces: Minos, Radamantis y Éaco, el barquero Caronte y muchos seres peligrosos como Cerbero.

Este suplemento añade esta nueva localización al mundo de Grecia ampliando todos sus apartados (Dioses, Equipo, Bestiario...) y además contiene una aventura ambientada en los dominios de Hades.

Publicado el 8 de diciembre de 2012.



Está editado en tapa blanda y contiene 72 páginas en blanco y negro.

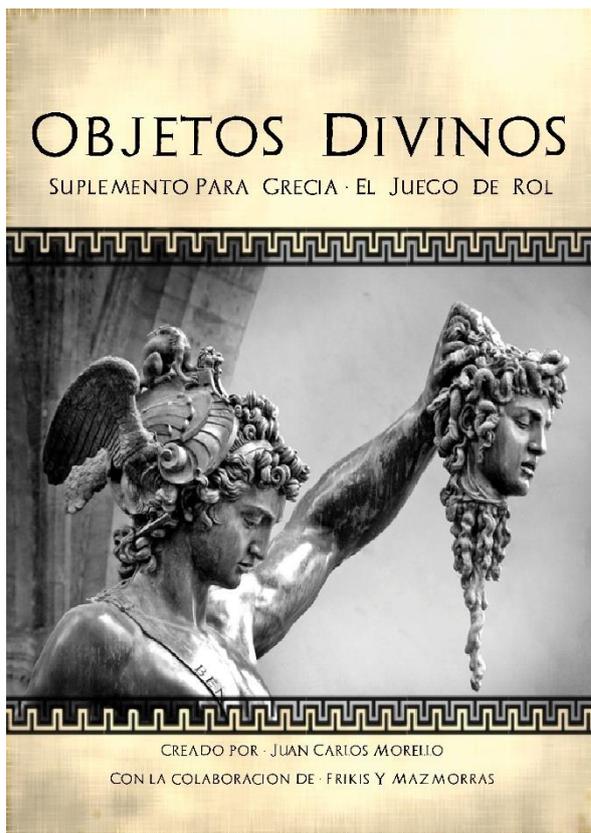
El Inframundo: <http://www.lulu.com/shop/juan-carlos-morell%C3%B3-gand%C3%ADa/el-inframundo/paperback/product-20562687.html>

Objetos Divinos

Objetos Divinos es la segunda ampliación para Grecia: El Juego de Rol segunda edición. En este suplemento encontrarás una gran cantidad de objetos, dotados de gran poder por los dioses. Algunos de ellos fueron usados por grandes héroes como Odiseo, Jasón o Perseo.

Dentro de este suplemento encontraremos leyendas sobre las deidades y los héroes clásicos y una descripción detallada de algunos de los objetos que solían utilizar, entre ellos el casco de Hades, el cuerno de la abundancia o la flauta de Pan.

Publicado el 20 de enero de 2014.



Está editado en tapa blanda y contiene 84 páginas en blanco y negro.

Objetos Divinos: <http://www.lulu.com/shop/juan-carlos-morell%C3%B3-gand%C3%ADa/objetos-divinos/paperback/product-21408251.html>

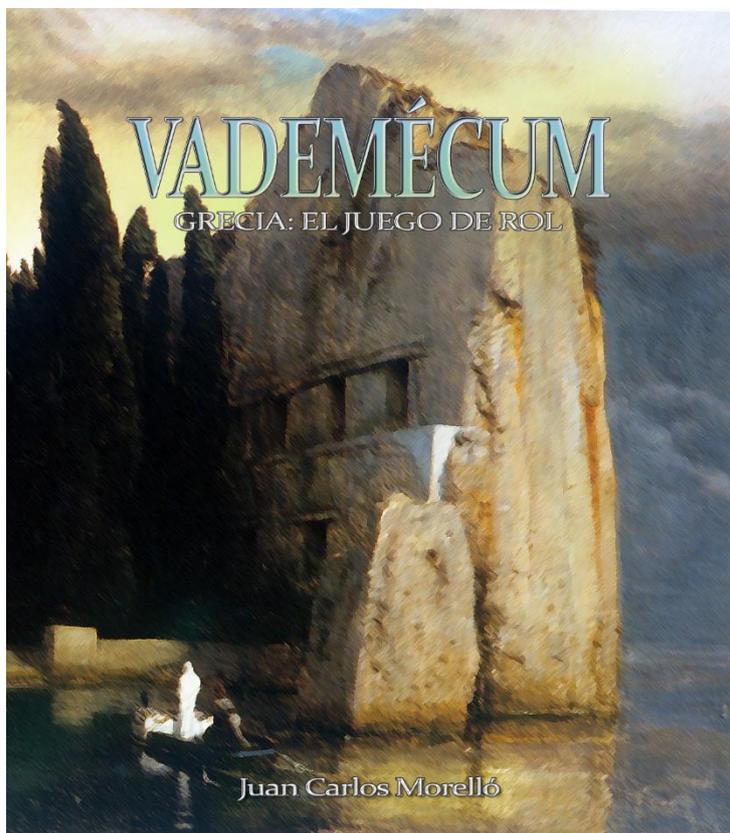
Vademécum

El Vademécum es la tercera ampliación para Grecia: el juego de rol y a diferencia de las anteriores no se dedica en expandir o profundizar un tema en concreto. El propósito de este suplemento es recoger pequeñas ayudas publicadas en el blog y junto a nuevas ideas y reglas unirlas todas en un mismo suplemento.

Como punto fuerte de este suplemento se encuentran las reglas navales y el catálogo de embarcaciones que ocupan más de la mitad de sus páginas, pero también encontraremos reglas adicionales para Grecia, y los suplementos independientes Quimeras y Gigantes de Grecia mejorados y ampliados para la ocasión.

Además en este suplemento se encuentran los códigos QR que se podrán leer desde la aplicación para móviles/tabletas de Grecia.

Pendiente de publicación.



Está editado en tapa blanda y contiene 84 páginas en blanco y negro.

Motivación

La tecnología está cambiando el mundo de la cultura y las obras escritas llevan unos años compitiendo directamente con libros electrónicos y otras variantes, como los lectores de cómics o libros para tabletas o portátiles. En los juegos de rol se está viendo también esta corriente, publicándose cada vez más en formato electrónico a la par que físicamente. Durante una partida de rol, los libros quedan como consulta encima de la mesa, la flexibilidad e inmediatez que nos proporciona en ese momento una búsqueda en internet o directamente en un pdf, es algo con lo que muchos directores de juego ya se han familiarizado, incluyendo en sus mesas de juego dispositivos electrónicos para agilizar la partida o ambientarla con, música, imágenes o incluso vídeos.

La idea de crear esta aplicación para el móvil, es precisamente esa, servir como ayuda a los jugadores y al director de juego, clasificando toda la información de varios libros de una manera amigable y rápida, para que las partidas fluyan de una manera más fluida sin entorpecer el juego.

También se crearán funcionalidades que no son posibles de implementar en un libro físico, como generadores automáticos de objetos y aventuras o un lanzador de dados, intuitivo y adaptado al juego de Grecia.

Herramientas de trabajo

En este capítulo vamos a ver con que herramientas se ha trabajado para crear esta aplicación y las motivaciones que me han llevado a su elección.

Herramientas de diseño

Para el desarrollo de esta aplicación no se han necesitado herramientas de diseño adicionales, ya que toda la información requerida estaba ya estructurada y bien definida en los libros del juego.

Herramientas de desarrollo

La herramienta principal para el desarrollo de esta aplicación ha sido Android Studio, que proporciona casi todos los recursos para afrontar el desarrollo de cualquier aplicación para esta plataforma.

Android studio

Android Studio es un entorno de desarrollo integrado para la plataforma Android. Fue anunciado el 16 de mayo de 2013 en la conferencia Google I/O, y reemplazó a Eclipse como el IDE oficial para el desarrollo de aplicaciones para Android. La primera versión estable fue publicada en diciembre de 2014.

Está basado en el software IntelliJ IDEA de JetBrains, y es publicado de forma gratuita a través de la Licencia Apache 2.0. Está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux.

Características

- Renderización en tiempo real
- Consola de desarrollador: consejos de optimización, ayuda para la traducción, estadísticas de uso.
- Soporte para construcción basada en Gradle.
- Refactorización específica de Android y arreglos rápidos.
- Herramientas Lint para detectar problemas de rendimiento, usabilidad, compatibilidad de versiones, y otros problemas.
- Plantillas para crear diseños comunes de Android y otros componentes.
- Soporte para programar aplicaciones para Android Wear.

Motivación

Android Studio fue el entorno de desarrollo sugerido por el tutor de esta práctica y tras unos primeros programas de prueba, me dejó un buen sabor de boca con lo que decidí que esta sería la plataforma donde se desarrollaría esta aplicación.

Bluestacks

Bluestacks es una aplicación que funciona en un entorno de Windows (o MAC) y es un emulador de un entorno de Android que permite instalar aplicaciones .apk o directamente conectarse a los servidores de Google Play, previa creación (o logeo) de una cuenta de Gmail, para descargar todas estas aplicaciones.

APLICACIÓN ANDROID PARA GRECIA: EL JUEGO DE ROL

Es un emulador sólido que permite lanzar las aplicaciones directamente desde Android Studio a Bluestacks haciendo que la carga de datos entre las dos aplicaciones se reparta.

Bluestacks se puede descargar gratuitamente desde su página web [9] aunque también existe una versión de pago sin publicidad. Está orientada a juego de móviles, ya que la reciente explosión de estos juegos hace que la gente quiera jugarlos, aunque no dispongan de móviles con la memoria o requisitos suficientes para ejecutarlos.

Características

- Tamaño de la descarga: 185 Mb
- Integración de la cámara
- Prueba Developer Support
- Soporte técnico de Windows-Nativo Gráficos
- Multi-touch Soporte
- Sensores Integrados
- Ejecuta aplicaciones basados en x86
- Ejecuta aplicaciones basados en ARM
- Mover archivos entre Windows + Android
- Integración Micrófono
- Integración del ratón del teclado
- Móvil / Desktop Sync
- Haga doble click APK abierto desde el escritorio
- Android-on-TV Capacidad

Motivación

Ya era usuario de Bluestacks antes de embarcarme en la realización del TFG, pero después de probar con el Android studio y los emuladores de teléfonos que me ofrecía el programa funcionaban de una forma un poco lenta y en ocasiones inestable, así que decidí utilizar esta herramienta que además se integra muy bien, ya que después de la compilación me permite instalar y lanzar el programa en un segundo plano en Bluestacks, hasta que me informa que he instalado correctamente.

Adobe Photoshop

Adobe Photoshop es un editor de gráficos desarrollado por Adobe Systems Incorporated. Usado principalmente para el retoque de fotografías y gráficos.

El uso de esta aplicación ha sido exclusivamente para retocar y dar formato a diversas imágenes y títulos que aparecen en la aplicación.

Es el único software utilizado en el desarrollo de este proyecto que no es gratuito, pero sí que tiene disponible en descarga una versión de prueba de 30 días desde su página web [10].

Motivación

Como ya tengo mucha experiencia con el paquete de Adobe, sobre todo con las herramientas Photoshop e InDesign, Grecia, está maquetado íntegramente con esta aplicación. He decidido utilizar el editor de gráficos para darle un aspecto más atractivo a las imágenes que aparecen en la aplicación.

Interfaz gráfica de usuario

La interfaz gráfica es una de las partes más importantes en las aplicaciones de Android, porque representa la aplicación a ojos del usuario ya que es lo que éste puede visualizar y manipular para interactuar con la misma.

Contexto

En este proyecto la interfaz gráfica no es la parte más importante, sería en contenido y funcionalidades, pero aun así necesita que muestre claramente la información y que aporte a la aplicación una funcionalidad clara e intuitiva.

Para diseñar una buena interfaz gráfica hay que tener en cuenta varios factores:

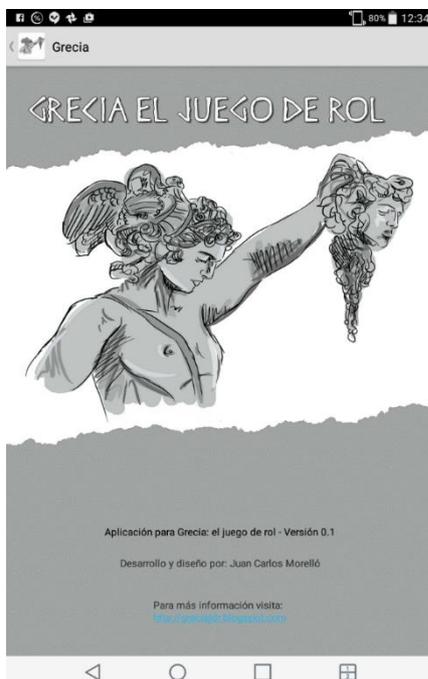
- Usabilidad: debemos implementar una aplicación fácil de usar y aprender.
- Accesibilidad: tenemos que asegurarnos de que las personas sean capaces de utilizar nuestro producto.
- Solidez: los usuarios tienen que poder realizar sus objetivos sin problemas.
- Consistencia: se trata de una de las características más importantes y está ligada íntimamente a la usabilidad. Consiste en que dentro de la aplicación, los mecanismos deben ser usados de la misma manera proporcionando la misma respuesta ante una misma acción, ofreciendo una apariencia estética común.

Pantalla principal

La pantalla principal es la primera vista que el usuario obtendrá nada más abrir la aplicación, por lo que es muy importante cuidar su estética y aportar sólo la información necesaria de una forma organizada y clara.

En esta aplicación se ha decidido que la pantalla principal incluirá información sobre la aplicación sin incluir ninguna funcionalidad más allá de abrir el menú lateral con las diferentes opciones.

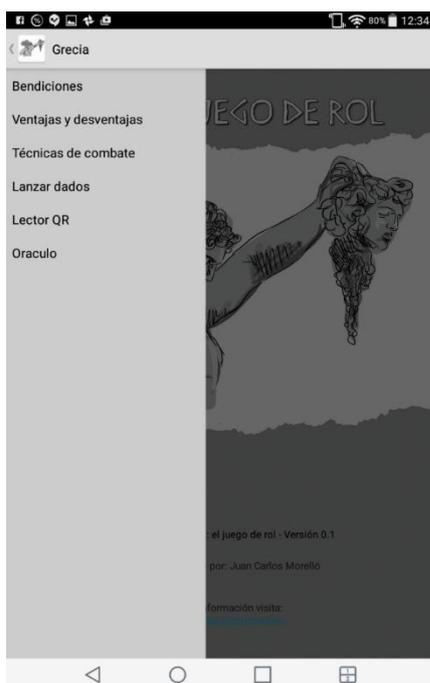
Al ser la pantalla principal, se ha cuidado un poco más el aspecto gráfico con una ilustración en tonos grises basada en la portada del suplemento *Objetos Divinos*, realizada por Aida Sancho.



Menú lateral

Para la navegación en la aplicación se ha decidido emplear un Navigation Drawer, se trata de un estándar que Google ha adoptado en todas sus aplicaciones recientemente para facilitar el uso y la navegación por las distintas opciones. Actualmente, no sólo Google está renovando sus productos con esta nueva interfaz, sino que otros muchos desarrolladores importantes ya han hecho la implementación de sus aplicaciones con esta nueva filosofía gráfica.

La función de este panel es actuar como un contenedor de opciones para navegar por las distintas ventanas (activities) de la aplicación. Este panel está disponible desde todas las ventanas de la aplicación.



Para mejorar la accesibilidad a este panel, además de poder abrirse utilizando un gesto deslizando, también se puede acceder al mismo tocando el icono de la aplicación.

Pantalla de bendiciones

En esta pantalla aparecen los nombres de los 13 dioses que se pueden encontrar en el libro básico y el suplemento de El Infamundo.

Al tocar cada deidad se abre el listview y se muestran las bendiciones asignadas a cada dios, al hacer tocar cada una de estas bendiciones, se muestran los efectos de la bendición en concreto.

Pantalla de técnicas de combate

La pantalla de Técnicas de combate es muy similar a la de Bendiciones, pero en esta ocasión nos muestra las 6 escuelas que se presentan entre el libro y suplementos.

Al tocar cada Escuela se abre el listview para mostrarnos cada técnica de combate y al hacer click sobre cada una de ellas aparece la descripción y sus efectos.



Pantalla de ventajas y desventajas

Ventajas y Desventajas funciona de una manera similar a las pantallas anteriores de bendiciones y Técnicas de combate, mostrándonos todas las ventajas y desventajas que aparecen en el libro básico y sus suplementos. Dada su especial condición (son totalmente independientes unas de otras), estas opciones no están agrupadas, están ordenadas alfabéticamente, para que su búsqueda sea mucho más natural.

A la izquierda de cada Ventaja/Desventaja, se encuentra un círculo de color verde o rojo, indicándonos si se trata de una ventaja (verde) o desventaja (rojo). Al hacer click sobre cada una de estas opciones, se nos muestra la descripción, los efectos que tiene y su coste en puntos.

APLICACIÓN ANDROID PARA GRECIA: EL JUEGO DE ROL



Pantalla de Lanzar dados

La pantalla de lanzar dados está creada para que los jugadores puedan sentarse a jugar a Grecia sin necesidad de dados especiales, en este caso dados de 6, de 10 o de 20 caras.

Se muestran una serie de dados sin número, y al tocarlos aparece un valor, de uno hasta la máxima capacidad del dado. Al hacer click sucesivamente, vuelve a aparecer otro valor en el dado, porque aunque en Grecia las tiradas suelen hacerse de uno en uno, las tiradas pueden sucederse con rapidez, así que es mejor no crear pantallas por las que moverse para realizar una tirada.

APLICACIÓN ANDROID PARA GRECIA: EL JUEGO DE ROL

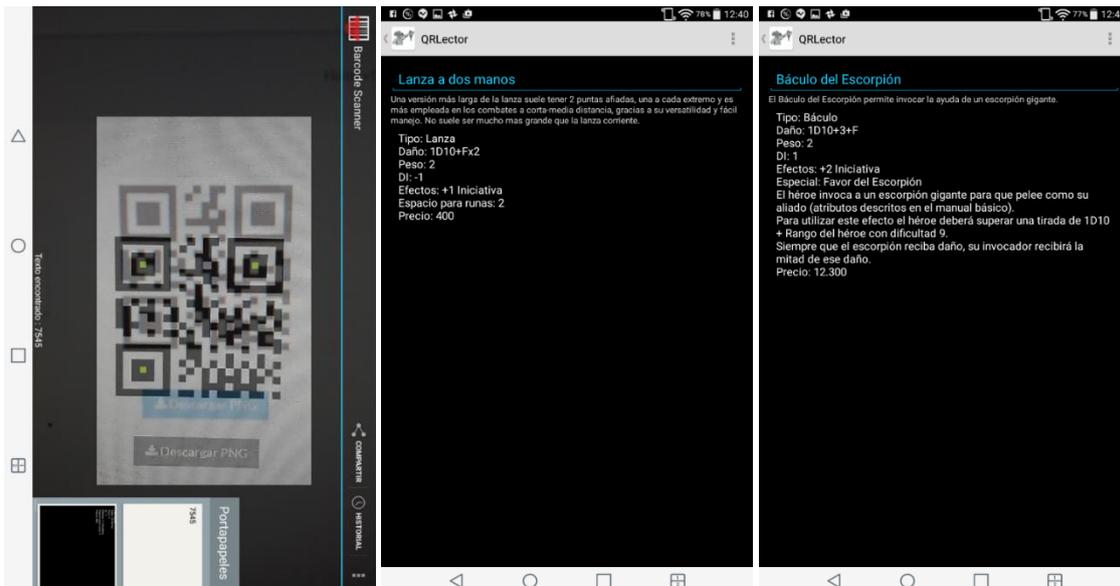


Pantalla de códigos QR

En esta pantalla se muestra el dibujo de un cofre, para representar que el jugador va a encontrar un tesoro pronto.



Al hacer click en la imagen, se nos abre la aplicación Barcode Scanner para capturar el contenido de un código QR. Después de leer el código proporcionado por los libros, se ejecuta una función aleatoria para mostrar el contenido del cofre, cuya rareza puede variar, dependiendo del código que leamos.



PANTALLA DEL ORÁCULO

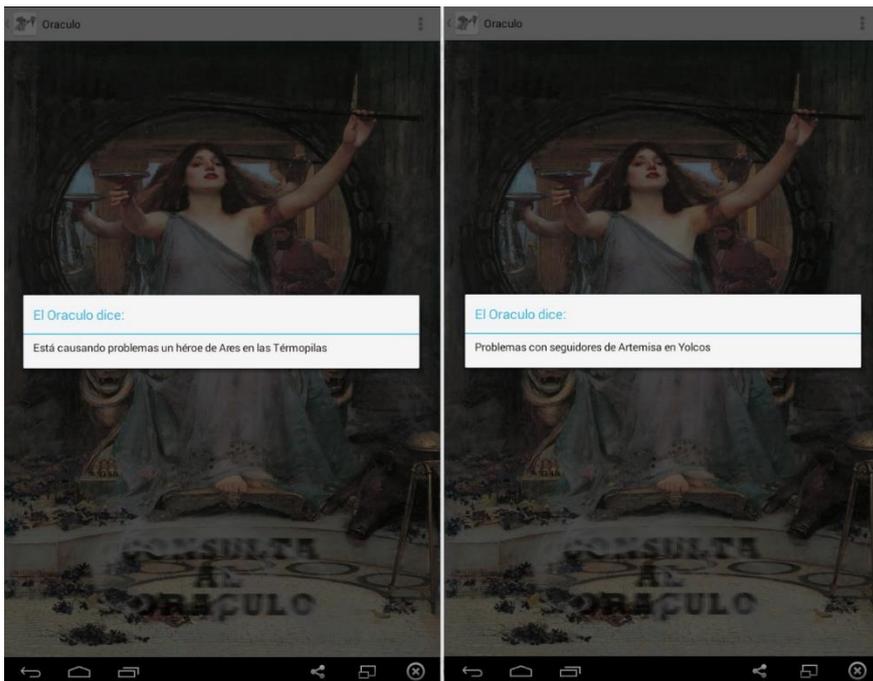
En esta pantalla se muestra una imagen de fondo de una sacerdotisa griega y un mensaje en letras fantasmales en el que se puede leer “consulta al oráculo”. Esta pantalla está pensada para generar semillas de aventura de manera aleatoria y con un cierto grado de coherencia. A partir de aquí el director puede empezar a crear una serie de actos y situaciones para que los jugadores construyan la historia.



Las directrices en las que se basa este sistema de creación de historia son tres: Que, quien y donde.

- Que, representa lo que acaba de suceder o está a punto de ocurrir en el mundo, puede tratarse de una guerra, un secuestro o tratarse del ataque de una bestia.
- Quien, representa siempre a una deidad, puede que sea la responsable de lo que está sucediendo o ser la deidad que necesita la ayuda de los héroes, este aspecto queda en manos del director de juego.
- Donde, es el lugar donde se ha desarrollado el evento anterior o donde se prevé que ocurrirá. En algunas ocasiones esta localización será concreta, por ejemplo Atenas o Tebas, pero en otras se nombrará la región u otras extensiones de tierra (o mar) y el director de juego podrá ubicarla más fácilmente.

Al hacer click en la imagen se mostrará una ventana que representa el mensaje entregado por el oráculo a los jugadores, este mensaje corto se genera de forma aleatoria y totalmente invisible para ellos, sin posibilidad de modificación alguna.



Recursos multimedia

Esta aplicación no utiliza un número muy elevado de recursos multimedia, ya que prima el aspecto informativo frente al estético.

Recursos gráficos

Los recursos gráficos han sido creados para el propio juego o son de licencia libre.

Para su utilización en la aplicación deben ser almacenados en la carpeta "res/drawable/".

La imagen de portada representa a Perseo cortando la cabeza de medusa y es obra de Aida Sancho, para el blog del juego de rol de Grecia.

La imagen del cofre que aparece en la pantalla del lector de QR está obtenida de un repositorio de imágenes libres de internet y está modificada con Photoshop por el autor de esta aplicación.

La imagen de la pantalla de Oráculo es un cuadro famoso llamado "Circe ofreciendo la copa a Odiseo" de John William Waterhouse.

Recursos de audio

Los recursos de audio representan los sonidos que se reproducen al activar ciertos efectos en determinadas pantallas, estos sonidos han sido extraídos de diferentes canciones.

El sonido que se reproduce al ejecutar el lector de códigos QR es un extracto de la canción "Wilderness" de Kow Otani.

El sonido que se reproduce al obtener una profecía en la pantalla del oráculo es un extracto de la canción “Lakeside” de Kow Otani.

Implementación de la reproducción de los sonidos en una aplicación Android

Como primer paso estos sonidos han sido almacenados en una carpeta que hemos creado y llamado raw, esta carpeta la ubicaremos dentro de los recursos, es decir en “res/raw”.

Para su implementación deberemos crear una clase nueva, a la que hemos llamado sonido.

```
public void sonido () {  
    MediaPlayer mp = MediaPlayer.create(this,  
    R.raw.sonidodelfos);  
    mp.start();  
}
```

Esta clase es llamada desde la función donde se ejecuta el evento relacionado.

```
sonido();
```

Implementación de funcionalidades

En esta sección veremos la aplicación en profundidad, con fragmentos de código que explicarán sus funcionalidades más destacadas.

Navigation Drawer

Como se dijo en la sección anterior, desde todas las pantallas de la aplicación se dispondrá de acceso al menú deslizante lateral denominado "Navigation Drawer".

Para llevar a cabo su implementación se ha consultado la documentación pertinente en la página oficial de desarrolladores Android.

El código se ha implementado en cada una de las "activities" visibles del programa y también en su correspondiente archivo .java.

Código extraído de `Principal.java`:

Declaración:

```
private DrawerLayout mDrawer;
private ListView mDrawerOptions;
private static final String[] values = {"Bendiciones",
    "Ventajas y desventajas", "Técnicas de combate", "Lanzar
    dados", "Lector QR"};
```

Código dentro de la función `OnCreate`:

```
getActionBar().setDisplayHomeAsUpEnabled(true);
mDrawerOptions = (ListView)
    findViewById(R.id.left_drawer);
mDrawer = (DrawerLayout)
    findViewById(R.id.drawer_layout);
mDrawerOptions.setAdapter(new ArrayAdapter<String>(this,
    android.R.layout.simple_list_item_1, android.R.id.text1,
    values));
mDrawerOptions.setOnItemClickListener(this);
```

Código incluido después de la función `OnCreate`:

```
@Override
public void onItemClick(AdapterView<?> adapterView, View
    view, int i, long l) {

    switch (i){
        case 0://BENDICIONES
            startActivity(new Intent(Principal.this,
                Bendiciones.class));
            break;
        case 1://VENTAJAS
            startActivity(new Intent(Principal.this,
                Ventajas.class));
            break;
        case 2://Tecnicas
            startActivity(new Intent(Principal.this,
                Tecnicas.class));
            break;
        case 3://DADOS
```

APLICACIÓN ANDROID PARA GRECIA: EL JUEGO DE ROL

```

        startActivity(new Intent(Principal.this,
LanzarDados.class));
        break;
        case 4://QR
            startActivity(new Intent(Principal.this,
QRlector.class));
            break;
    }
    mDrawer.closeDrawers();
}

```

Para añadir la funcionalidad de abrir el menú cada vez que se pulsa el botón superior izquierdo, el que da título a la pantalla en la que nos encontramos, hay que añadir una funcionalidad al método “public boolean onOptionsItemSelected()” en caso de que exista, si no existe se crea de la siguiente forma:

```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case android.R.id.home:
            if (mDrawer.isDrawerOpen(mDrawerOptions)) {
                mDrawer.closeDrawers();
            } else {
                mDrawer.openDrawer(mDrawerOptions);
            }
            return true;
    }

    return super.onOptionsItemSelected(item);
}

```

Código extraído de `activity_principal.xml`:

```

<android.support.v4.widget.DrawerLayout
xmlns:android="http://schemas.android.com/apk/res/android
"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".Principal">

<!-- Aquí vendría el resto del código xml -->

<ListView
    android:id="@+id/left_drawer"
    android:layout_width="280dp"
    android:layout_height="match_parent"
    android:layout_gravity="start"
    android:choiceMode="singleChoice"
    android:divider="@android:color/transparent"

```

```

    android:dividerHeight="0dp"
    android:background="#CCC">
</ListView>

```

```
</android.support.v4.widget.DrawerLayout>
```

LISTVIEW y LISTVIEW ADAPTER

Android tiene un tipo de elemento llamado Listview, este elemento fue creado específicamente para desplegar listas y contiene todo lo que necesitas para representar los datos de una aplicación, contenidos por ejemplo en una array.

En esta aplicación existen varias pantallas que aprovechan esta funcionalidad para mostrar información, ya sea como una listview simple (Ventajas y Desventajas) o con una listview en árbol con dos niveles (Bendiciones y Técnicas de combate).

Vamos a ver cómo queda implementado en las técnicas de combate.

Código extraído de activity_TECNICAS.xml

Primeramente se crea un listview en la pantalla, pero sin introducir los ítems (contenido en árbol). Después vamos a necesitar dos layouts adicionales, uno para los ítems, y otro para los subitem.

```

<ListView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/listViewTC"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true" />

```

Código extraído de items_layout_tecnicas.xml

Este layout contendrá los ítems, que en este caso serán los nombres de las escuelas de las diferentes técnicas de combate.

```

<CheckedTextView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/textView1"
    android:layout_width="wrap_content"

    android:layout_height="?android:attr/listPreferredItemHeight"
    android:layout_marginLeft="8dp"
    android:gravity="left"
    android:paddingLeft="32dp"
    android:paddingTop="8dp"
    android:paddingBottom="8dp"
    android:text="Test"
    android:textSize="18sp"

```

```

    android:textAlignment="textEnd"
    android:textStyle="bold" />

```

Código extraído de `subitems_layout_tecnicas.xml`

Para finalizar, se crea otro código XML que contendrá el formato y los ítems a mostrar de cada apartado del listview.

El `LinearLayout`, es donde se contienen los elementos de la pantalla y deberá tener la propiedad `clickable:true`.

Para representar los elementos de las técnicas de combate se necesitan dos `TextView`, uno para el nombre y otro para la descripción. Adicionalmente hemos creado un pequeño `view` de un `1dp` de altura para representar una línea de color de separación.

<`LinearLayout`

```

    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="40dp"
    android:clickable="true"
    android:orientation="vertical"
    android:paddingLeft="40dp"
    tools:context=".MainActivity" >

```

<`TextView`

```

    android:id="@+id/nombreTC"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

```

```

    android:textAppearance="?android:attr/textAppearanceMedium"
    android:drawablePadding="200dp"
    android:layout_marginTop="10dp"
    android:layout_marginBottom="10dp"
    android:gravity="center_vertical"
    android:text="nombreTC"
    android:textSize="16sp" />

```

<`TextView`

```

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

```

```

    android:textAppearance="?android:attr/textAppearanceSmall"
    android:text="descripcionTC"
    android:layout_marginBottom="10dp"
    android:id="@+id/descripcionTC" />

```

<`View`

```

    android:layout_width="match_parent"
    android:layout_height="1dp"
    android:background="@android:color/black" />

```

</`LinearLayout`>

codigo extraido de [ListViewAdapter.java](#)

Este código, nos servirá para los dos ListView en árbol que se utilizan en la aplicación, Bendiciones y Técnicas de combate. Por lo que no es necesario crear dos clases diferentes.

```
public class ListViewAdapter extends BaseAdapter {

    Context context;
    String[] titulos;
    String[] descripcion;
    LayoutInflater inflater;

    public ListViewAdapter(Context context, String[] titulos,
String[] descripcion) {
        this.context = context;
        this.titulos = titulos;
        this.descripcion = descripcion;
    }

    @Override
    public int getCount() {
        return titulos.length;
    }

    @Override
    public Object getItem(int position) {
        return null;
    }

    @Override
    public long getItemId(int position) {
        return 0;
    }

    public View getView(int position, View convertView,
ViewGroup parent) {

        TextView txtTitle;
        ImageView imgImg;
        TextView txtDesc;

        inflater = (LayoutInflater)

context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);

        View itemView = inflater.inflate(R.layout.list_row,
parent, false);

        txtTitle = (TextView)
itemView.findViewById(R.id.list_row_title);
        txtDesc = (TextView)
itemView.findViewById(R.id.list_row_descripcion);
```

```

        txtTitle.setText(titulos[position]);
        txtDesc.setText(descripcion[position]);

        return itemView;
    }
}

```

Código extraído de AdapterTecnicas.java

A continuación deberemos crear el adaptador para el **contenido** de las técnicas de combate.

```

public class AdaptadorTecnicas extends BaseExpandableListAdapter
{
    private final SparseArray<GrupoDeItems> grupos;
    public LayoutInflater inflater;
    public Activity activity;

    // Constructor
    public AdaptadorTecnicas(Activity act,
SparseArray<GrupoDeItems> grupos) {
        activity = act;
        this.grupos = grupos;
        inflater = act.getLayoutInflater();
    }

    // Nos devuelve los datos asociados a un subitem en base
    // a la posicion
    @Override
    public Object getChild(int groupPosition, int childPosition)
    {
        return
grupos.get(groupPosition).children.get(childPosition);
    }

    // Devuelve el id de un item o subitem en base a la
    // posicion de item y subitem
    @Override
    public long getChildId(int groupPosition, int childPosition)
    {
        return 0;
    }

    // En base a la posicion del item y de subitem nos devuelve
    // el objeto view correspondiente y el layout para los
    subitems
    @Override
    public View getChildView(final int groupPosition, final int
childPosition,
        boolean isLastChild, View convertView, ViewGroup
parent) {
        final String children = (String) getChild(groupPosition,
childPosition);

```

APLICACIÓN ANDROID PARA GRECIA: EL JUEGO DE ROL

```

    TextView textvw = null;
    if (convertView == null) {
        convertView =
inflater.inflate(R.layout.subitems_layout_tecnicas, null);
    }
    textvw = (TextView)
convertView.findViewById(R.id.textView1);
    textvw.setText(children);

    final TextView textdsc;
    textdsc = (TextView)
convertView.findViewById(R.id.textView2);
    textdsc.setVisibility(View.GONE);

    switch (groupPosition){
    case 0://Luz
        switch (childPosition){
            case 0://Cegar
                textdsc.setText("La espada...");
                break;
            case 1://Ataque de doble filo
                textdsc.setText("El héroe...");
                break;
            case 2://Defensa perfecta
                textdsc.setText("El héroe...");
                break;
            case 3://Estocada de Aquiles
                textdsc.setText("El héroe...");
                break;
            case 4://Ataque doble
                textdsc.setText("El héroe...");
                break;
            case 5://Espada ardiente

                textdsc.setText("El héroe...");
                break;
            case 6://El resurgir del Fénix
                textdsc.setText("El héroe...");
                break;
        }

//Aquí seguiremos introduciendo la informacion de los items.

        break;
    }
convertView.setOnClickListener(new OnClickListener() {

    //Mostramos o escondemos la información.
    @Override
    public void onClick(View v) {
        if(textdsc.getVisibility()==View.VISIBLE)

```

```

        textdsc.setVisibility(View.GONE);
    }
    else
        textdsc.setVisibility(View.VISIBLE);
    });

    return convertView;
}

//Obtenemos el layout para los items
@Override
public View getView(int groupPosition, boolean
isExpanded,
    View convertView, ViewGroup parent) {
    if (convertView == null) {
        convertView =
inflater.inflate(R.layout.items_layout_tecnicas, null);
    }
    GrupoDeItems grupo = (GrupoDeItems)
    getGroup(groupPosition);
    ((CheckedTextView) convertView).setText(grupo.string);
    ((CheckedTextView) convertView).setChecked(isExpanded);
    return convertView;
}

@Override
public boolean hasStableIds() {
    return false;
}

//Nos informa si es seleccionable o no un item o subitem
@Override
public boolean isChildSelectable(int groupPosition, int
childPosition) {
    return false;
}
}
}

```

Código extraído de Tecnicas.java

Finalmente modificamos el java de Técnicas de combate para incluir los ítems y subítems en los que divide nuestro árbol, es decir el nombre de las escuelas y el nombre de cada técnica de combate.

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_tecnicas);

    crearDatos();
    ExpandableListView listView = (ExpandableListView)
    findViewById(R.id.listViewexp);
    AdaptadorTecnicas adapter = new AdaptadorTecnicas(this,
grupos);
}

```

```
        listView.setAdapter(adapter);
    }

    public void crearDatos() {

        GrupoDeItems grupo0 = new GrupoDeItems("Luz");
        grupo0.children.add("Cegar");
        grupo0.children.add("Ataque de doble filo");
        grupo0.children.add("Defensa perfecta");
        grupo0.children.add("Estocada de Aquiles");
        grupo0.children.add("Ataque doble");
        grupo0.children.add("Espada ardiente");
        grupo0.children.add("El resurgir del Fenix");
        grupos.append(0, grupo0);

        //Aquí seguimos añadiendo items del mismo modo.

    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.menu_tecnicas, menu);
        return true;
    }
}
```

Lector QR

El lector QR de esta aplicación, se ha programado ayudándose de Zxing, una librería externa y opensource para Java que se utiliza para poder leer los códigos QR con la cámara de nuestro dispositivo móvil o Tablet.

Para integrar Zxing en nuestra aplicación deberemos seguir una serie de pasos.

1. Crear un nuevo Package llamado: com.google.zxing.integration.android
2. Crear dentro de este Package dos clases nuevas llamadas IntentIntegrator y IntentResult
3. Copiar y pegar el código de estas clases, este código se puede encontrar en la página oficial de Zxing
<https://github.com/zxing/zxing/blob/master/android-integration/src/main/java/com/google/zxing/integration/android/IntentIntegrator.java>
<https://github.com/zxing/zxing/blob/master/android-integration/src/main/java/com/google/zxing/integration/android/IntentResult.java>
4. Crear el código de la vista, es decir del xml donde se vaya a ejecutar el lector.
5. Crear la llamada a la aplicación para realizar el scanner.

Código referente al paso 4, extraído de `activity_qrlector.xml`

```
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:textAppearance="?android:attr/textAppearanceLarge"
    android:id="@+id/nombreObjeto"
    android:layout_alignParentTop="true"
    android:layout_alignParentLeft="true"
    android:text="titulo"
    android:editable="false"
    android:textIsSelectable="false"
    android:visibility="invisible"
    android:textColor="#23d0ff" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

    android:textAppearance="?android:attr/textAppearanceSmall"
    android:id="@+id/descripcionObjeto"
    android:text="descripcion corta"
    android:layout_below="@+id/nombreObjeto"
    android:visibility="invisible"
    android:textColor="#d6d8d4" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/reglasObjeto"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_below="@+id/descripcionObjeto"
    android:text="reglas"
    android:visibility="invisible"
    android:editable="false"
    android:textColor="#ffffff" />

<ImageView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:id="@+id/imageObjeto"
    android:src="@drawable/cofre"
    android:longClickable="true"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true" />
```

Este código contiene una `ImageView` que es el cofre mostrado que se trata como un botón para iniciar la llamada al lector de QR.

El resultado es leído y después de tratar esta información internamente, se muestra un resultado aleatorio que se escribe en los distintos campos `EditText`, que hasta el momento han

permanecido invisibles, la información se vuelve visible y el cofre se esconde para evitar más iteraciones.

Código referente al paso 5, extraído de QRlector.java

```

final ImageView boton3 = (ImageView)
findViewById(R.id.imageObjeto);
boton3.setOnClickListener(new View.OnClickListener() {

    public void onClick(View view) {

        //Intentamos abrir el Barcode Scanner
        try {
            Intent intent = new
Intent("com.google.zxing.client.android.SCAN");

            intent.putExtra("com.google.zxing.client.android.SCAN.SCAN_
MODE", "QR_CODE_MODE");
            startActivityForResult(intent, 0);
        } catch (ActivityNotFoundException e) {

            Toast.makeText("Error:",
Toast.LENGTH_SHORT).show();
            IntentIntegrator integrator = new
IntentIntegrator(fragment.getActivity());
            integrator.initiateScan();
        }
    }
});

```

Al principio declaramos la variable boton3, para leer las iteraciones que hagamos con la ImageView del código xml, es decir cuando hagamos click en el cofre.

Después se efectuará un try/catch para realizar la llamada a la aplicación lectora de QR externa, en caso de no efectuarse muestra un mensaje de error. En caso de que no haya ningún error leeremos el código QR correspondiente y trataremos la información conseguida.

El código QR que lee la aplicación es un número entero que mediante una serie de operaciones es tratado para que actué la aplicación de una forma o de otra.

```

public void onActivityResult(int requestCode, int resultCode, Intent
data) {
    String contents = null;
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == 0) {
        if (resultCode == RESULT_OK) {
            contents = data.getStringExtra("SCAN_RESULT");
            String format = data.getStringExtra("SCAN_RESULT_FORMAT");
            final TextView nombreO = (TextView)
findViewById(R.id.nombreObjeto);
            final TextView descripcionO = (TextView)
findViewById(R.id.descripcionObjeto);
            final TextView reglaO = (TextView)

```

APLICACIÓN ANDROID PARA GRECIA: EL JUEGO DE ROL

```

findViewById(R.id.reglasObjeto);
    final ImageView cofreO = (ImageView)
findViewById(R.id.imageObjeto);

    long resultadoQR = 0;
    int valorSwitch = 0;

    try {
        resultadoQR = Long.parseLong(contents);
    } catch (Exception e) {
        AlertDialog alertDialog;
        alertDialog = new
AlertDialog.Builder(QRlector.this).create();
        alertDialog.setTitle("Error");
        alertDialog.setMessage("Código QR no valido para esta
aplicación.");
        alertDialog.show();
        // No se ha registrado correctamente el escaneado
startActivity(new Intent(QRlector.this,
QRlector.class));
    }

    resultadoQR = resultadoQR % 100000000;

    /*
    AlertDialog alertDialog;
    alertDialog = new
AlertDialog.Builder(QRlector.this).create();
    alertDialog.setTitle("Detectado:");
    */

    if (resultadoQR < 10) {
        valorSwitch = 1; //Objeto básico
        alertDialog.setMessage("Objeto básico " +
resultadoQR);
    } else if (resultadoQR < 100) {
        valorSwitch = 2; //Aleatorio
        alertDialog.setMessage("Objeto aleatorio " +
resultadoQR);
    } else if (resultadoQR < 1000) {
        valorSwitch = 3; //Objeto Divino
        alertDialog.setMessage("Objeto divino < 1000 " +
resultadoQR);

        //Código Oculto (Easter Eggs)

        } else {
            valorSwitch = 15; //Valor grande
            alertDialog.setMessage("Objeto divino aleatorio " +
resultadoQR);
        }
        //alertDialog.show();

        int valorQR = 0;
        Random r = new Random();

        nombreO.setVisibility(View.VISIBLE);
        descripcionO.setVisibility(View.VISIBLE);
        reglaO.setVisibility(View.VISIBLE);

```

```
switch (valorSwitch) {  
  
case 1:  
    cofreO.setVisibility(View.GONE);  
    valorQR = r.nextInt(19);  
  
    nombreO.setText (getResources ().getStringArray (R.array.objeto_corriente_nombre) [valorQR]);  
  
    descripcionO.setText (getResources ().getStringArray (R.array.objeto_corriente_desc) [valorQR]);  
  
    reglaO.setText (getResources ().getStringArray (R.array.objeto_corriente_reglas) [valorQR]);  
  
break;  
  
case 2: //Probabilidad dependiente del resultadoQR, menor que 100.  
    cofreO.setVisibility(View.GONE);  
    valorQR = r.nextInt(100);  
    if (valorQR > resultadoQR) {  
        valorQR = r.nextInt(19);  
    }  
  
    nombreO.setText (getResources ().getStringArray (R.array.objeto_corriente_nombre) [valorQR]);  
  
    descripcionO.setText (getResources ().getStringArray (R.array.objeto_corriente_desc) [valorQR]);  
  
    reglaO.setText (getResources ().getStringArray (R.array.objeto_corriente_reglas) [valorQR]);  
    } else if (valorQR >= 99) {  
        valorQR = r.nextInt(50);  
    }  
  
    nombreO.setText (getResources ().getStringArray (R.array.objeto_divino_nombre) [valorQR]);  
  
    descripcionO.setText (getResources ().getStringArray (R.array.objeto_divino_desc) [valorQR]);  
  
    reglaO.setText (getResources ().getStringArray (R.array.objeto_divino_reglas) [valorQR]);  
    } else {  
        valorQR = r.nextInt(12);  
    }  
  
    nombreO.setText (getResources ().getStringArray (R.array.objeto_zodiaco_nombre) [valorQR]);  
  
    descripcionO.setText (getResources ().getStringArray (R.array.objeto_zodiaco_desc) [valorQR]);  
  
    reglaO.setText (getResources ().getStringArray (R.array.objeto_zodiaco_reglas) [valorQR]);  
    }  
break;
```

APLICACIÓN ANDROID PARA GRECIA: EL JUEGO DE ROL

```

case 3:
cofreO.setVisibility(View.GONE);
    valorQR = r.nextInt(12);

nombreO.setText(getResources().getStringArray(R.array.objeto_zodiaco_nombre) [valorQR]);

descripcionO.setText(getResources().getStringArray(R.array.objeto_zodiaco_desc) [valorQR]);

reglaO.setText(getResources().getStringArray(R.array.objeto_zodiaco_reglas) [valorQR]);
break;

case 15:
if (resultadoQR>95000000) {
    valorQR = (int) resultadoQR%12;

nombreO.setText(getResources().getStringArray(R.array.objeto_zodiaco_nombre) [valorQR]);

descripcionO.setText(getResources().getStringArray(R.array.objeto_zodiaco_desc) [valorQR]);

reglaO.setText(getResources().getStringArray(R.array.objeto_zodiaco_reglas) [valorQR]);

} else if (resultadoQR>80000000) {
    valorQR = (int) resultadoQR%50;

nombreO.setText(getResources().getStringArray(R.array.objeto_divino_nombre) [valorQR]);

descripcionO.setText(getResources().getStringArray(R.array.objeto_divino_desc) [valorQR]);

reglaO.setText(getResources().getStringArray(R.array.objeto_divino_reglas) [valorQR]);
} else {
    valorQR = (int) resultadoQR%19;

nombreO.setText(getResources().getStringArray(R.array.objeto_corriente_nombre) [valorQR]);

descripcionO.setText(getResources().getStringArray(R.array.objeto_corriente_desc) [valorQR]);

reglaO.setText(getResources().getStringArray(R.array.objeto_corriente_reglas) [valorQR]);
    }
    cofreO.setVisibility(View.GONE);

break;
default:
    nombreO.setText("Reintentar");

break;
}}}
```

En este código se trata el resultado obtenido por el código QR y se realizan distintas operaciones aleatorias (random), dependiendo del resultado (switch). Se han omitido los valores de los Strings para ahorrar espacio, estos contienen los nombres y la información de los objetos.

Ciertos códigos de barras, se han tratado de manera diferente, con unos huevos de pascua, para provocar un efecto directo, sin pasar por el aleatorio, en algunos casos los objetos de han creado a propósito para la aplicación.

Esto está pensado como una broma o sorpresa para los usuarios que lean los códigos de barras de diferentes juegos o productos.

RANDOM

La clase Random proporciona un generador de números aleatorios que es más flexible que la función estática random de la clase Math. Se ha utilizado en esta aplicación constantemente, dada la naturaleza aleatoria de esta clase de juegos, muchas veces hay que echar mano del azar para obtener resultados aleatorios.

Para crear una secuencia de números aleatorios tenemos que seguir los siguientes pasos:

1. Proporcionar a nuestro programa información acerca de la clase Random. Mediante un import que será: java.util.Random.
2. Crear un objeto de la clase Random.
3. Llamar a una de las funciones miembro que generan un número aleatorio.
4. Usar el número aleatorio.

Vamos a ver su implementación en [Delfos.Java](#) (Generador del oraculo)

1. Importamos la clase random

```
import java.util.Random;
```

2. Creamos un objeto de la clase Random

```
Random r = new Random();
```

3. Declaramos algunas variables y llamando a la función nextInt les asignamos un valor aleatorio.

```
int RandomQue;  
int RandomQuien;  
int RandomDonde;
```

```
RandomQue = r.nextInt(8);  
RandomQuien = r.nextInt(15);  
RandomDonde = r.nextInt(30);
```

4. Utilizamos este valor.

```
AlertDialog.setTitle("El Oraculo dice:");  
AlertDialog.setMessage(Que[RandomQue] + Quien[RandomQuien]+  
" " + Donde[RandomDonde]);  
AlertDialog.show();
```

GOOGLE PLAY

Para descargar esta aplicación desde esta plataforma, ha sido necesario registrarse como desarrollador en Google. Vamos a ver un poco su funcionamiento.

Que es Google Play

Google Play Store (anteriormente Android Market) es una plataforma de distribución digital de aplicaciones móviles para los dispositivos con sistema operativo Android, así como una tienda en línea desarrollada y operada por Google. Esta plataforma permite a los usuarios navegar y descargar aplicaciones (desarrolladas mediante Android SDK), juegos, música, libros, revistas y películas. También se pueden adquirir dispositivos móviles como ordenadores Chromebook, teléfonos inteligentes Nexus, Google Chrome, entre otros.

Las aplicaciones se encuentran disponibles de forma gratuita, así como también con costo. Pueden ser descargadas directamente desde un dispositivo con Android a través de la aplicación móvil Google Play, que ya viene incluida en todos los sistemas operativos Android.

Convertirse en Desarrollador

Para convertirnos en desarrollador de aplicaciones para Google Play, hay que tener una cuenta de Google Gmail (Gratuita) y además deberemos hacer un pago de 25 dólares, una única vez, con lo que ya podremos ser desarrolladores que distribuyen apps en Google Play.

Una realizado el pago, podemos entrar en Google Play Developer Console por primera vez, nuestro centro de gestión e información como desarrolladores.

The screenshot shows the Google Play Developer Console interface. At the top, there is a search bar and user information for Juan Carlos Morelló. The main content area is titled 'TUS APLICACIONES' and contains a table with the following data:

NOMBRE DE LA APLICACIÓN	PRECIO	INSTALACIONES ACTUALES/TOTALES	VALORACIÓN MEDIA / TOTAL	ERRORES Y ANRS	ÚLTIMA ACTUALIZACIÓN	ESTADO
Greca APP 1.0	Gratuita	18 / 22	4.50 / 2	1	22/10/2015	Publicada

Below the table, there are sections for 'RECURSOS ÚTILES PARA ANDROID', 'HERRAMIENTAS ÚTILES', and '¿NECESITAS AYUDA?'. The footer contains copyright information for 2015 Google.

En esta web, podremos ver en un lateral:

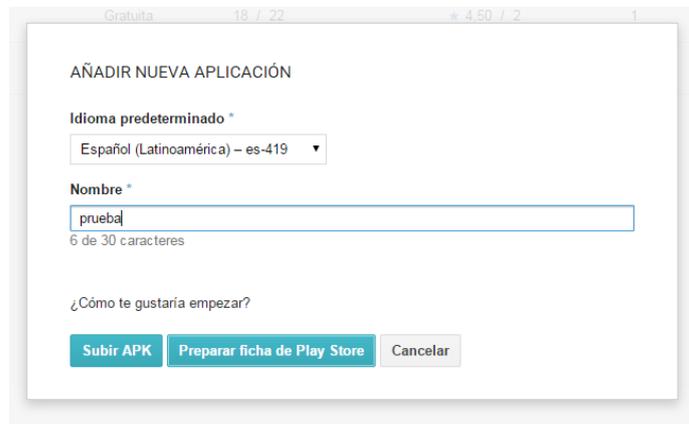
- Listado de nuestras aplicaciones
- Servicios para Google Play Games
- Informes de nuestros beneficios
- Configuración
- Alertas
- Noticias

APLICACIÓN ANDROID PARA GRECIA: EL JUEGO DE ROL

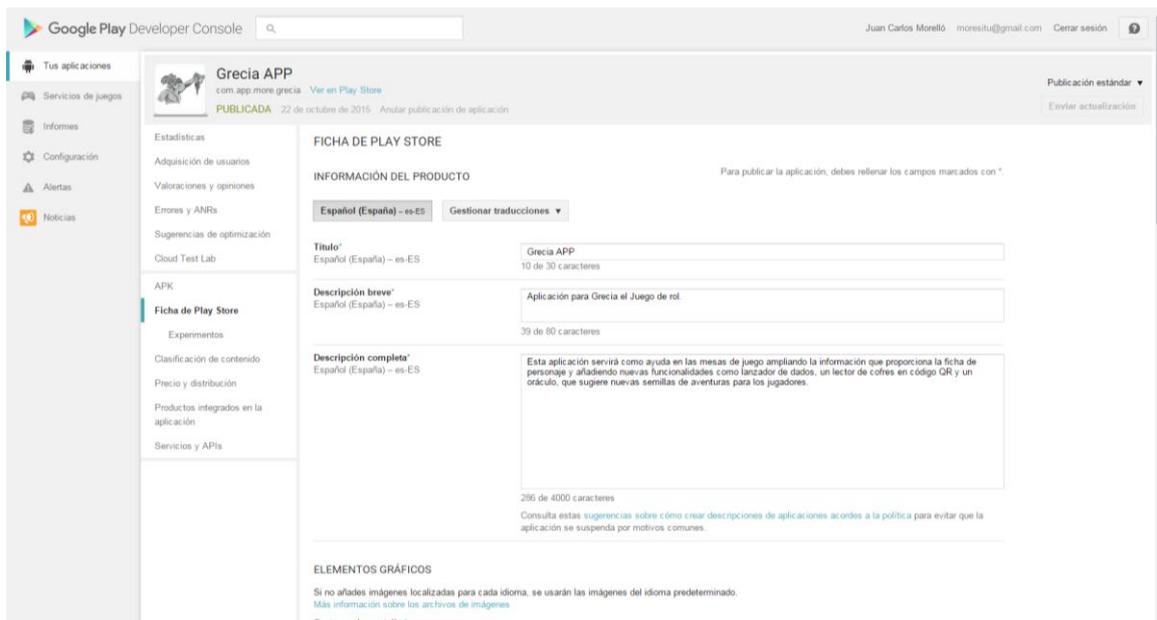
Y en la parte principal nuestras aplicaciones, con diversos parámetros y la opción de añadir nueva aplicación.

Creando una nueva aplicación

Para añadir una nueva aplicación a Google Play, basta con que le demos al botón **Añadir nueva aplicación** y nos llevará a un diálogo donde deberemos escoger el lenguaje por defecto de la aplicación y el título:



Una vez introducidos, se nos abrirá toda la información sobre la aplicación. Con todo el contenido que debemos rellenar para que se encuentre disponible para todos.



En primer lugar, podemos subir el APK de nuestra aplicación. Para ello, basta con que en la sección APK incluyamos el fichero. También podemos incluirlo en una fase de testeo Alpha o Beta, si nos conviene, o directamente a productivo, que es lo que aparecerá en Google Play.

APLICACIÓN ANDROID PARA GRECIA: EL JUEGO DE ROL

Las versiones Alpha y Beta no aparecerán listadas junto a las otras aplicaciones en Google Play, pero podrán ser descargadas por testeadores a los que podremos enviar directamente un correo o enlace para que descarguen la aplicación.

Clasificación del contenido

Antes de publicar deberemos pasar un test sobre la clasificación del contenido, que puede variar según países.

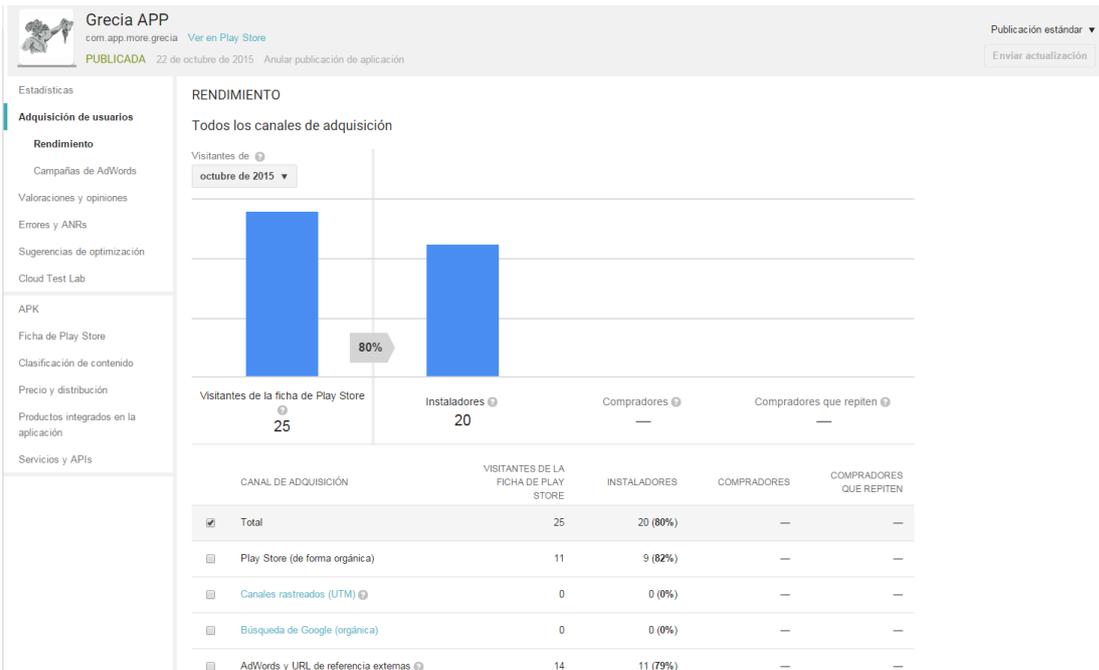
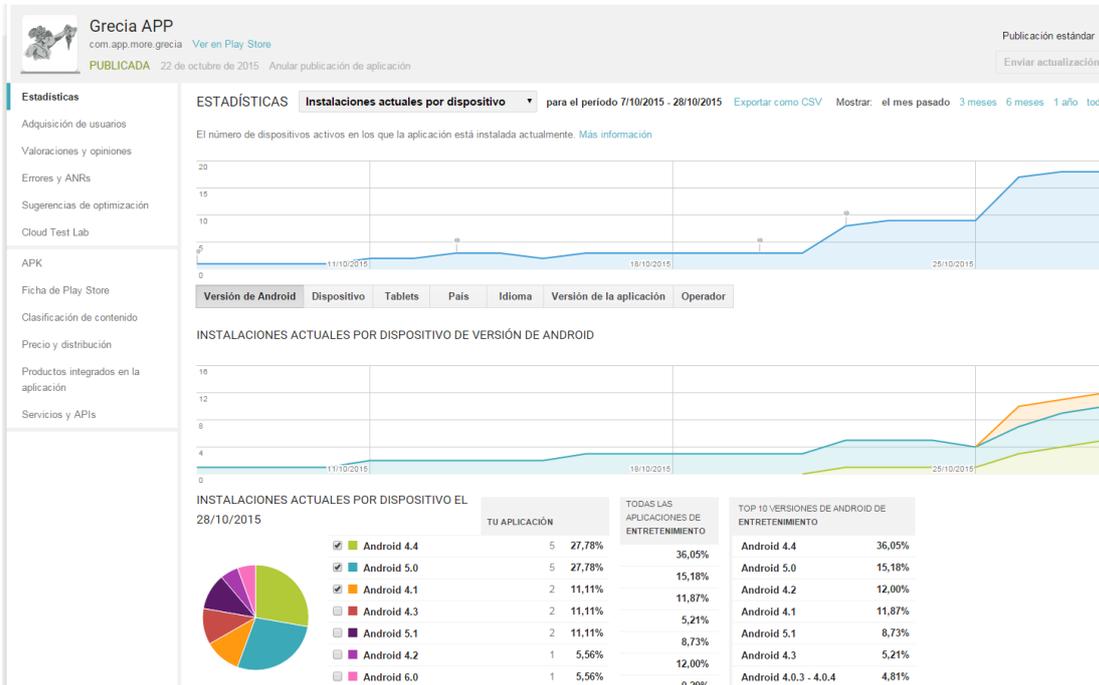
Precio y distribución

En la pestaña de precio y distribución están las condiciones y normas para vender nuestros productos en esta plataforma. Pero como esta app va a ser gratuita y sin publicidad, no se ha investigado en este apartado.

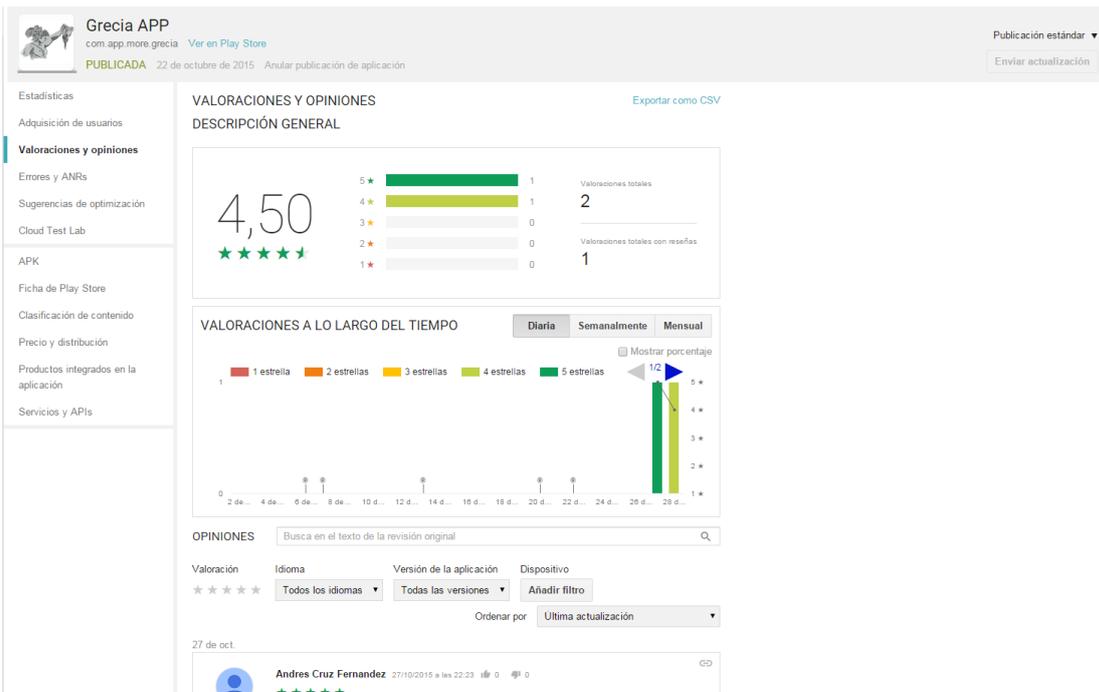
APLICACIÓN ANDROID PARA GRECIA: EL JUEGO DE ROL

Pestañas de estadísticas

Las demás pestañas nos muestran diferentes estadísticas según las descargas, los dispositivos que la están ejecutando y las opiniones que han dado los usuarios en Google Play.



APLICACIÓN ANDROID PARA GRECIA: EL JUEGO DE ROL



Motivación

La elección de convertirme en desarrollador en Google Play y publicar en su tienda online ha sido más una inversión de futuro ya que el pago es único y el mercado de aplicaciones Android está creciendo.

Ahora cada vez que realice alguna aplicación podré publicarla sin coste alguno.

Evaluación de la aplicación

Para obtener una valoración de la aplicación se creó un cuestionario entre los primeros usuarios de la aplicación por Google Play.

Cuestionario SUS

Con motivo de obtener una valoración de la satisfacción de la aplicación se ha decidido planificar una prueba de satisfacción con un cuestionario SUS (*System Usability Scale*) que nos permite recoger opiniones anónimas acerca de varios aspectos de la aplicación. El propósito de este cuestionario es proporcionar un test fácil de responder por los encuestados y que permita obtener unos resultados más o menos objetivos. Tiene una cantidad pequeña de preguntas y éstas se contestan puntuando en un rango del 1 -5. Para llevarlo a cabo se ha utilizado un servicio de Google que permite la creación de formularios de forma rápida y sencilla que además facilita la difusión y recopila los resultados en una hoja de cálculo para su posterior manipulación.

Este cuestionario se realizó mediante **Google Forms**, una funcionalidad gratuita de **Google Drive** que nos permite crear formularios de una manera sencilla y visualmente atractiva, que nos permite su difusión rápidamente mediante listas de correos, círculos de google+ o sencillamente con un enlace.

Este cuestionario sigue abierto actualmente, aunque para la redacción de la memoria se recogieron los datos el día 5/10/2015, se puede acceder a él mediante el siguiente enlace:

<https://goo.gl/forms/X5quSUJunR>

Las preguntas empleadas en el cuestionario son las siguientes:

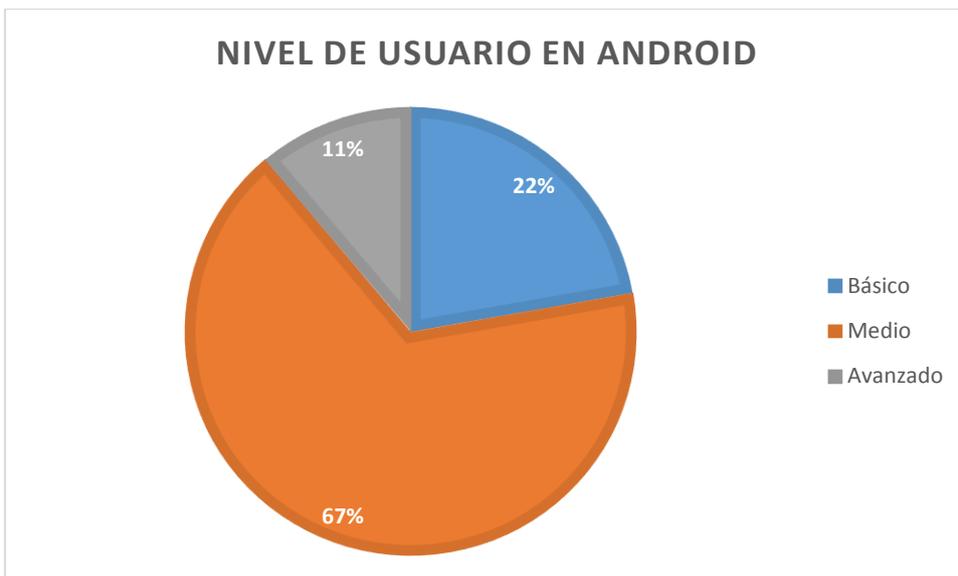
- Creo que me gustará utilizar esta aplicación con frecuencia
- Encontré la aplicación innecesariamente compleja
- Pensé que la aplicación sería más fácil de usar
- Creo que voy a necesitar el apoyo de un experto para utilizar la aplicación
- Me pareció que las diversas funciones de esta aplicación se integran bien
- Pienso que hay muchos aspectos mejorables en esta aplicación
- La mayoría de la gente podrá utilizar esta aplicación sin problemas
- Encontré la aplicación difícil de usar
- Me sentí seguro utilizando la aplicación
- Necesito aprender muchas cosas antes de manejarla bien con la aplicación

El baremo de las respuestas irá desde el 1, en desacuerdo, hasta el 5, totalmente de acuerdo.

Respuestas del cuestionario

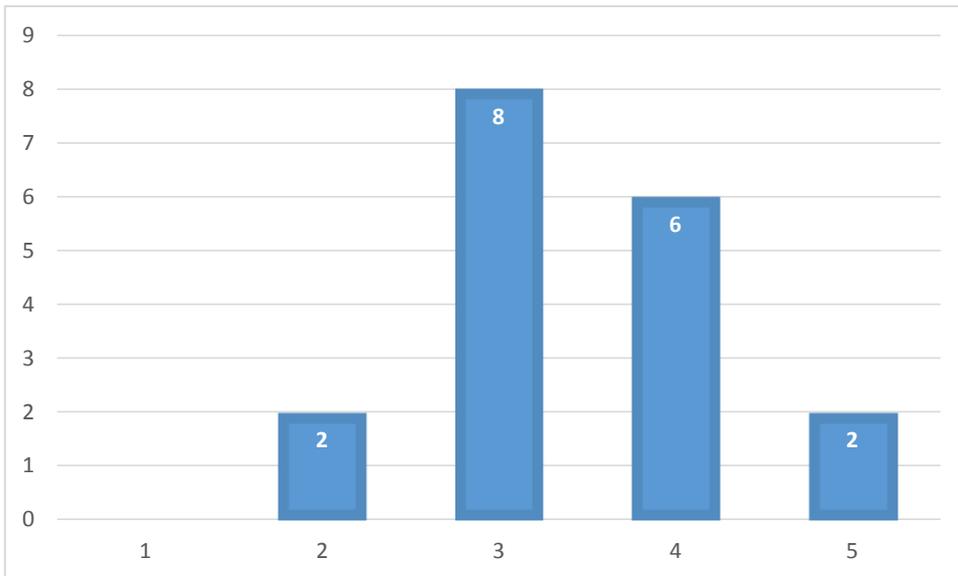


La mayoría de las personas que participaron en la encuesta fueron hombres, esto es bastante lógico, ya que en mis círculos de google+, que es donde se promocionó la encuesta, tengo una gran mayoría de hombres agregados.



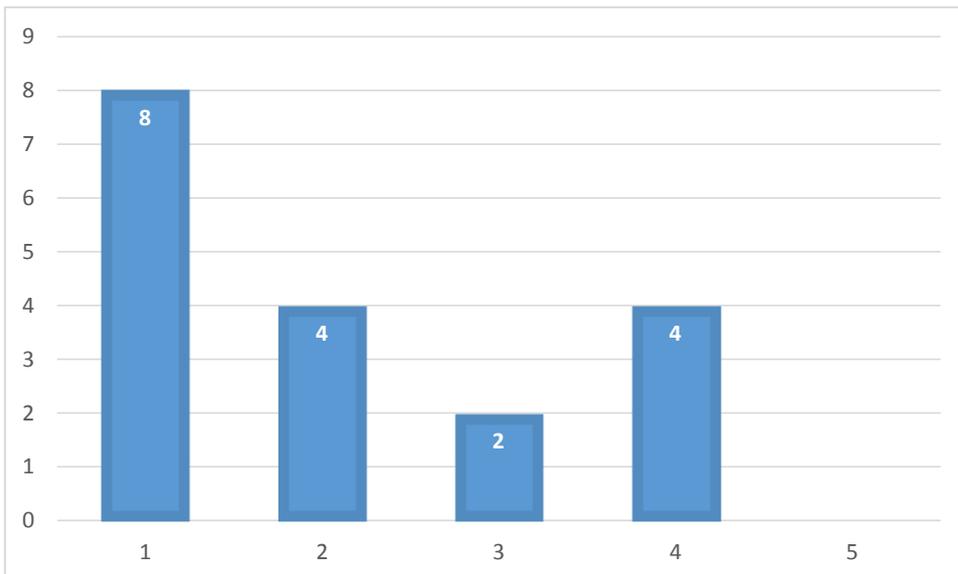
La mayoría de los encuestados son usuarios medios de Android, es decir son usuarios que con soltura, descargan y utilizan distintas aplicaciones de Google Play.

Creo que me gustará utilizar esta aplicación con frecuencia



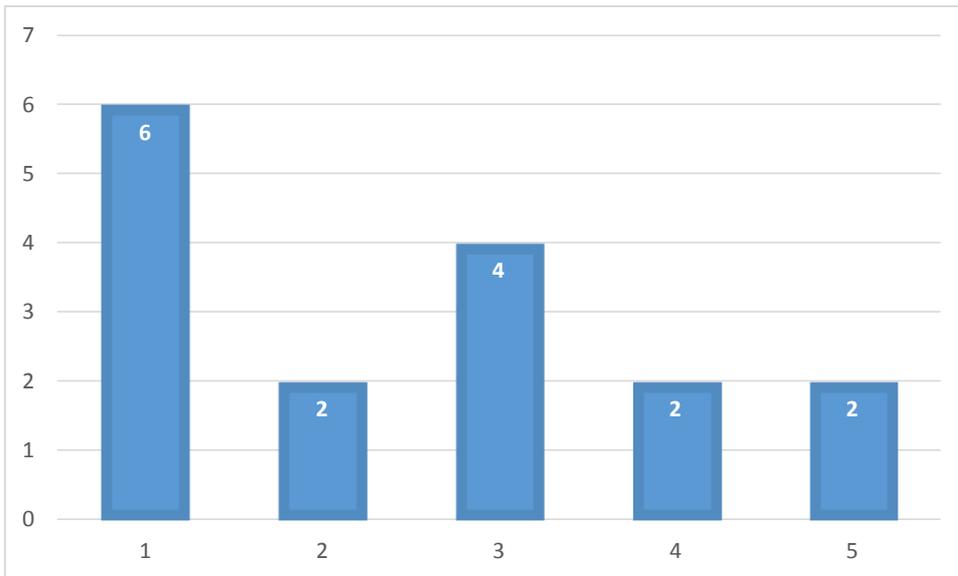
Al analizar el Diagrama de barras nos damos cuenta de que los valores 3 y 4 son los que más se repiten, seguido por el 5 y el 2, por lo que podemos concluir que a los usuarios les gustará utilizar la aplicación.

Encontré la aplicación innecesariamente compleja



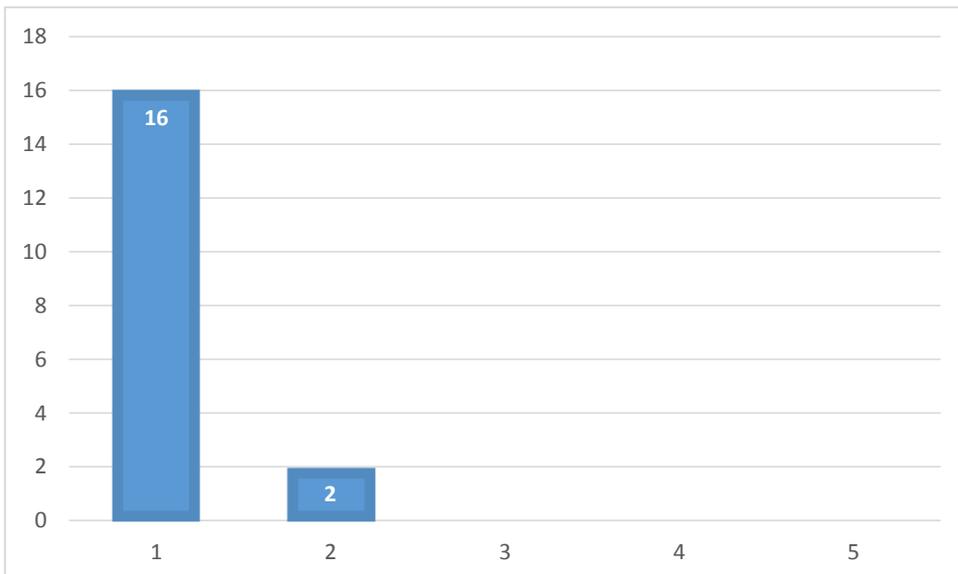
En este gráfico el valor que más se repite es el 1, seguido del 2 y el 4, lo que nos indica que si bien la aplicación no es compleja, sí que puede suponer algo de complicación para algún usuario.

Pensé que la aplicación sería más fácil de usar



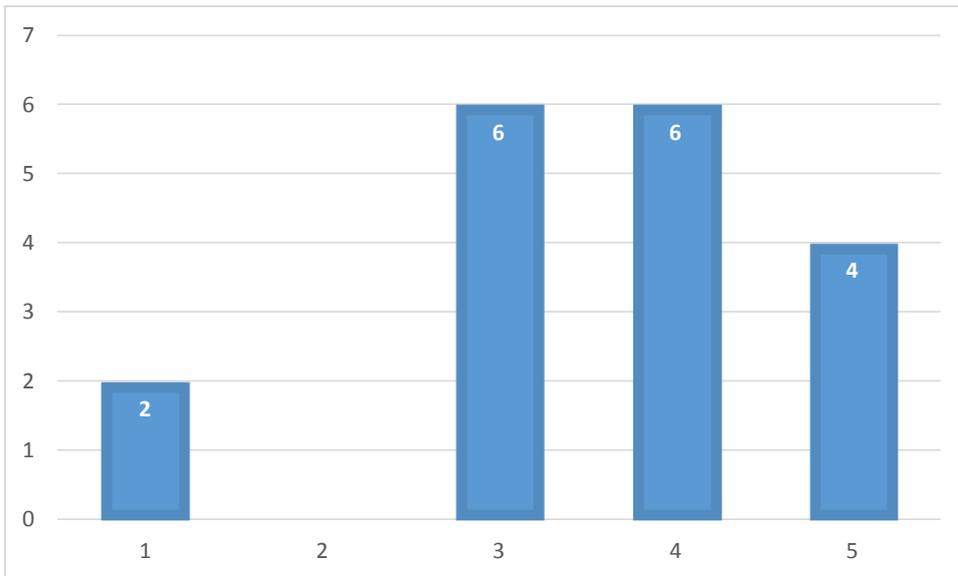
En este grafico vemos que los resultados están un poco más repartidos, aunque el 1, en desacuerdo y el 3 son un poco superiores, lo que nos indica que la aplicación no es demasiado difícil de utilizar, o que no requiere de mucha práctica para emplearla.

Creo que voy a necesitar el apoyo de un experto para utilizar la aplicación



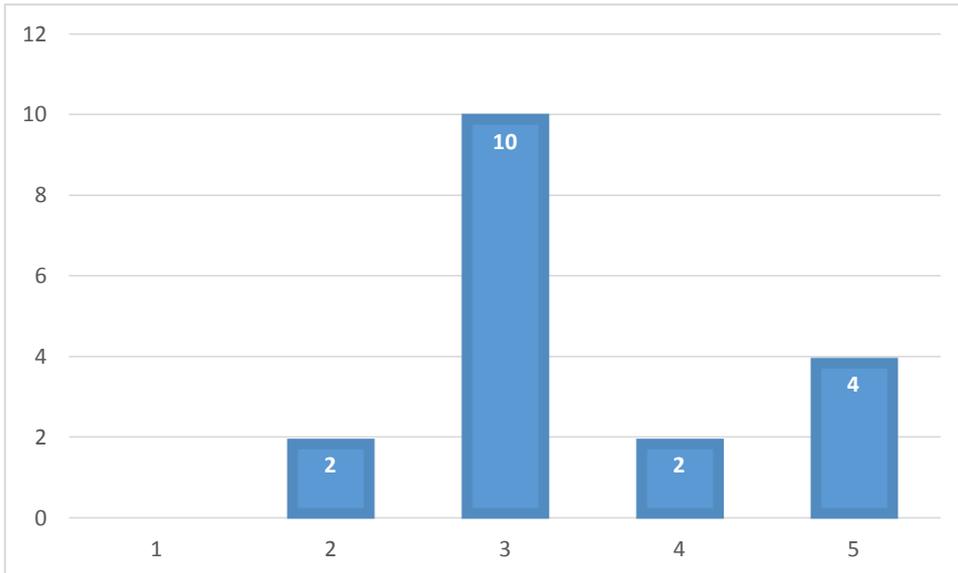
En esta pregunta, se ve claramente que los resultados están totalmente en desacuerdo del apoyo de un experto para utilizar la aplicación.

Me pareció que las diversas funciones de esta aplicación se integran bien



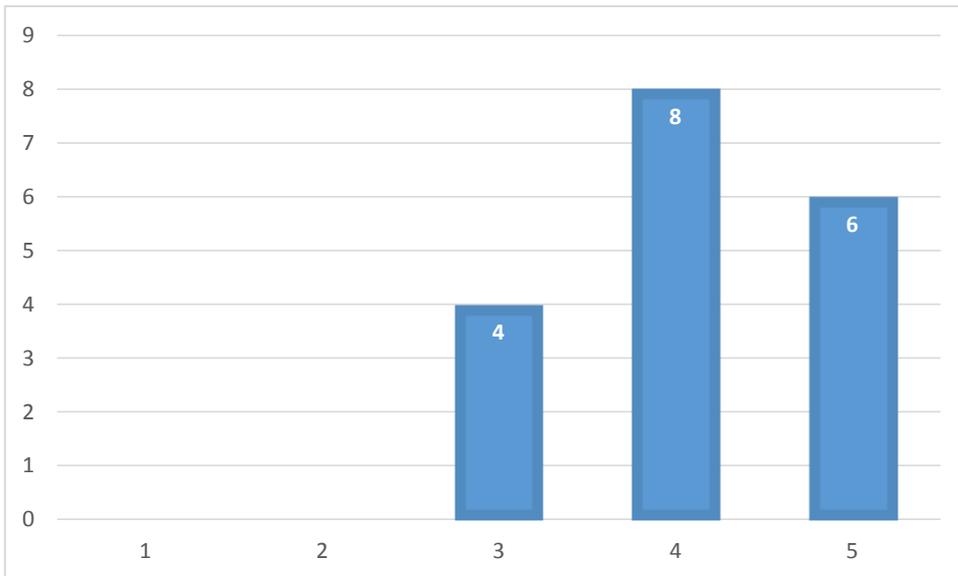
La gran mayoría de los encuestados están de acuerdo o parcialmente de acuerdo con que las funciones de esta aplicación se integran bien, solo unos pocos usuarios opinan que no.

Pienso que hay muchos aspectos mejorables en esta aplicación



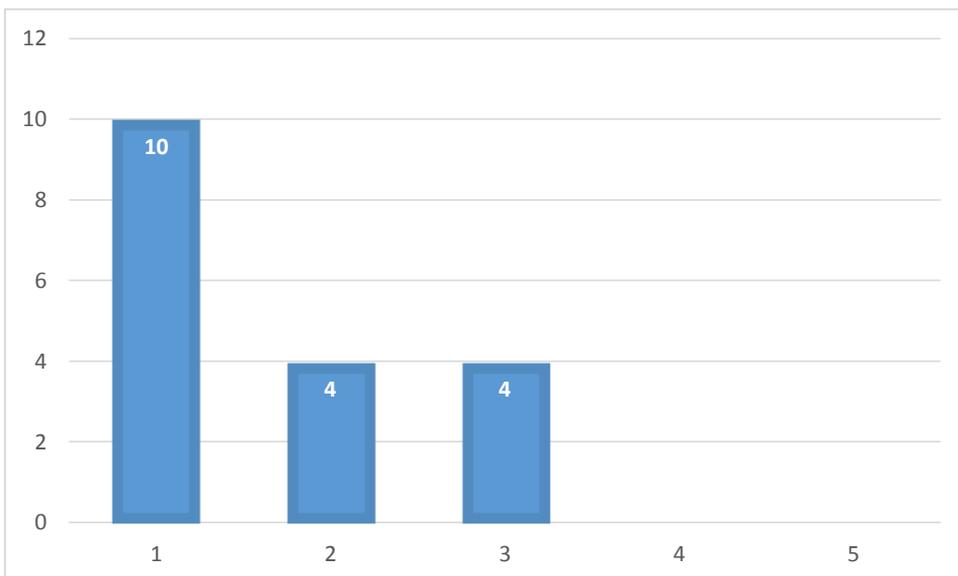
Casi todos los usuarios piensan que la aplicación es mejorable, aunque que la mayoría de ellos hayan votado el 3, parece indicar que hay margen de mejora pero que puede no ser tan grande.

La mayoría de la gente podrá utilizar esta aplicación sin problemas



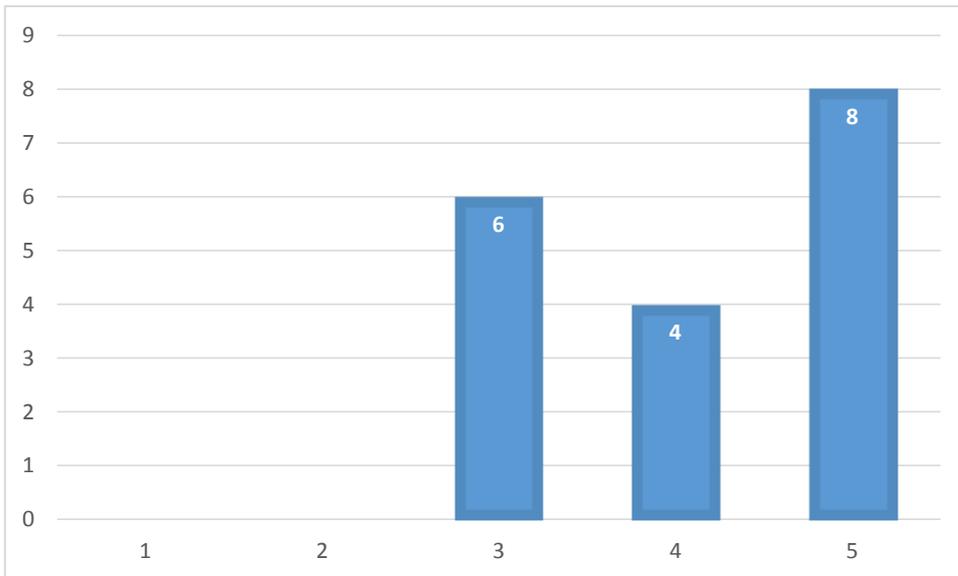
En este gráfico se entiende que todos los usuarios de un teléfono Android o Tablet con este sistema operativo podrán hacerse fácilmente con el manejo de la aplicación, ya que las respuestas 3, 4 y 5 se llevan todos los votos.

Encontré la aplicación difícil de usar



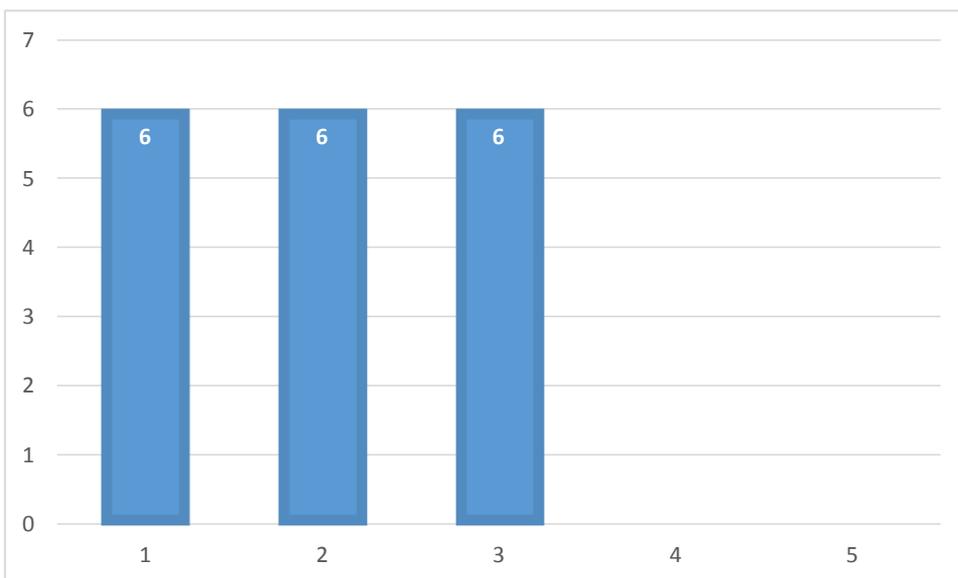
En esta ocasión los usuarios han votado los valores en desacuerdo: 1, 2 y además el 3, indicando que la aplicación no es difícil de utilizar o que la dificultad que pueda tener a priori no es demasiado grande.

Me sentí seguro utilizando la aplicación



En esta gráfica podemos ver una tendencia positiva sobre si los usuarios se sintieron seguros utilizando la aplicación.

Necesito aprender muchas cosas antes de manejarme bien con la aplicación



En esta ocasión los valores de 1, 2 y 3 están empatados indicándonos que si bien no se necesita aprender muchas cosas antes de manejarse con la aplicación, podría haber algún apartado de la misma que requiera de más tiempo para utilizarse mejor.

Conclusiones y propuestas

En este último capítulo se exponen las conclusiones extraídas después de todo el trabajo llevado a cabo para desarrollar la versión presente de la aplicación. Además, en el segundo apartado también se mencionan las posibles nuevas funcionalidades que pueden implementarse como trabajos futuros, para dotar al proyecto de más funciones y obtener una aplicación más completa, sobre todo en caso de expansión del juego.

Conclusiones

Para terminar y una vez acabada tanto la redacción de este documento escrito como el desarrollo de la aplicación objetivo de este trabajo, se pueden extraer varias conclusiones interesantes que ponen de manifiesto el estudio llevado a cabo para alcanzar los objetivos propuestos al inicio y que sin duda ayudarán a construir una perspectiva más precisa a la hora de comenzar otro proyecto de complejidad similar.

El primer propósito que se ha perseguido ha sido conocer en profundidad el sistema operativo sobre el que se ejecuta la aplicación desarrollada. Para ello se ha realizado un análisis de su arquitectura interna, de los elementos que constituyen un proyecto y de los componentes que construyen las aplicaciones.

Además del sistema operativo se ha necesitado conocer el entorno de desarrollo elegido y propuesto por el tutor, Android Studio, del que no sabía ni su existencia. La primera toma de contacto y los primeros programas de prueba me hicieron hacerme con el control de la aplicación en pocas semanas.

Otro objetivo relevante perseguido y alcanzado durante este desarrollo ha sido la construcción de una interfaz siguiendo la filosofía de diseño centrado en el usuario, donde lo importante es crear un producto visual intuitivo que aporte una experiencia satisfactoria a los usuarios, logrando que éstos realicen las tareas con el mínimo esfuerzo posible, sin perderse en laberintos de pantallas. Para conseguir esto se han consultado fuentes de información variadas.

Para hacer que la experiencia de un juego de rol de verdad se trasladase a la aplicación se han utilizado las librerías Random y se ha jugado con ellas matemáticamente para conseguir una aleatoriedad en ocasiones dirigida por medio de los códigos QR.

Por último, se ha realizado un cuestionario SUS donde han participado varios usuarios valorando múltiples aspectos de la aplicación. De esta forma se han podido obtener una serie de resultados que permiten conocer el grado de satisfacción de los usuarios cuando utilizan la aplicación.

Una vez concluido el desarrollo de la presente versión de la aplicación se ha procedido a su inclusión y promoción en Google Play Store (para ello he adquirido una cuenta de desarrollador), donde una vez subida se pone a disposición de millones de usuarios, facilitando enormemente su difusión y aumentando sus probabilidades de éxito.

La cuenta de desarrollador es de pago único, por lo que no requiere renovación ni más gasto, así que ahora puedo subir cualquier otra aplicación que desarrolle sin coste adicional.

Desde mi punto de vista personal he de decir que me ha sido muy placentero el poder desarrollar un proyecto software de estas características ya que me ha permitido poner en práctica muchos de los conceptos adquiridos a lo largo de los diferentes cursos de este grado, además de la importancia de leer páginas de documentación, oficial o no, tanto en inglés como en español. Es en estos momentos cuando uno realmente se da cuenta de la importancia de cada una de las experiencias vividas en la universidad y de las dificultades superadas a lo largo de la elaborada formación de la que se dispone en la EPSA.

Trabajo futuro y posibles ampliaciones

Grecia, es un proyecto que sigue creciendo con nuevos suplementos y ayudas oficiales o de sus aficionados, las cuales son distribuidas con la misma licencia que el juego original.

Cada vez que se amplíe la información contenida en el juego que sea visible para los jugadores, es decir, nuevos dioses, ventajas, desventajas o técnicas de combate, se debería actualizar la aplicación para incluir esta información. Rara vez se deberá actualizar el contenido existente en el juego, la información contenida en los libros es invariable y solo se efectuarían correcciones ortográficas, en caso de que existiesen.

También se podría ampliar las funcionalidades de la aplicación añadiendo más ventanas de información, como descripciones de los dioses, la geografía o la historia de Grecia.

Las ampliaciones más fáciles de implementar y que más profundidad le aportarían al juego, serían la posibilidad de lectura de diferentes códigos QR, que aparecerían en alguna aventura como pistas, mapas, enemigos o cofres. En este aspecto esta debería ser la funcionalidad que más novedades pueda incluir, ya que admitiría un espectro más amplio de novedades.

BIBLIOGRAFIA Y REFERENCIAS

[1] <http://www.movilzona.es/2015/09/02/android-afianza-su-dominio-en-el-mercado-espanol-mientras-windows-phone-sigue-cayendo/>

[2] https://es.wikipedia.org/wiki/GNU_General_Public_License

[3] <https://www.yeeply.com/blog/informe-sobre-el-uso-de-apps-en-espana-2014/>

[4] <https://www.conbarba.es/efa>

[5] <http://www.nosolorol.com>

[6] <http://developer.android.com/about/versions/lollipop.html>

[7] https://es.wikipedia.org/wiki/Juego_de_rol

[8] <http://greciajdr.blogspot.com>

[9] <http://www.bluestacks.com/>

[10] <https://creative.adobe.com/products/download/photoshop>

<http://developer.android.com/intl/es/index.html>

<http://www.imanolbracero.es/>

<http://www.rubenjarcia.es/>

<http://joomla.probando-cosas.com.ar/>

<http://www.sidar.org/>

<http://www.c-sharpcorner.com/>

GLOSARIO DE TÉRMINOS

A continuación se han incluido una serie de términos referentes a la programación en Android, aunque no todos estos términos se emplean en esta memoria, su comprensión es muy recomendable para entender el funcionamiento de los sistemas operativos Android.

ADB: Puente de depuración Android. Esta herramienta se utiliza para conectar y enviar comandos o archivos entre el teléfono y el ordenador.

ADB Shell: Es un *shell* (terminal) que funciona sobre *adb* y ofrece una línea de comandos que permite entre otras cosas operar con comandos *Unix* compatibles incluidos en el sistema.

Android: Es el nombre del sistema operativo Android.

AOSP (*Android Open Source Project*): quiere decir que Android es un proyecto de código libre y todos lo podemos utilizar.

API (*Application Programming Interface*): Es el conjunto de funciones y procedimientos que ofrece una biblioteca para utilizarse en otro software.

APN (*Access Point Name*): son los datos para poder acceder a Internet desde el móvil que tiene cada operadora de forma única.

APP (*Application*): es cualquier aplicación/es que instalamos.

APK (*Android Package Archive*): es la extensión de los paquetes de los programas que operan en este sistema.

Backup: es una copia de seguridad de nuestra ROM o *Firmware*.

Bootloader: es el gestor de arranque de Android, puede estar bloqueado o desbloqueado.

Bug: es un defecto en el software o un hardware que no ha sido descubierto por los creadores o diseñadores de los mismos durante su fabricación o desarrollo.

ODEX o DEODEX: En Android, todas las aplicaciones vienen empaquetadas en archivos con extensión APK. Junto con las aplicaciones de sistema o que vienen "de serie" en el firmware del fabricante (*system/app/*), se puede encontrar un archivo de extensión ODEX con el mismo nombre que el APK cuya función es la de ahorrar espacio. Estos archivos *.odex* son colecciones de partes de la aplicación correspondiente, que están optimizadas antes del arranque. Lo que se consigue con esto es que se arranque el sistema más rápido ya que con este sistema se hace una pre carga de parte del código de la aplicación.

Es por esto que las aplicaciones de sistema o preinstaladas sean difíciles de modificar, ya que parte del código de la aplicación ya está escrito en dicho archivo *.odex*, Llamo modificación a editar código para añadir o quitar funcionalidades a la aplicación o, simplemente, cambiar el aspecto de las aplicaciones editando la parte visual (*frameworks* y demás... iconos, logos, fondos, etc...).

Firmware: el firmware es una versión de Android modificada para que pueda adaptarse a un determinado terminal. Hay también distintos firmwares para cada versión de Android.

FC (Forzar Cierre): es un error común en las aplicaciones de Android, ocurre por un mal funcionamiento de la aplicación.

GPRS (*General Packet Radio Service*): forma parte de la segunda generación y media (2.5G), permite transmitir datos a velocidades comprendidas entre los 56 Kbps y los 114 Kbps.

GSM: también conocido como 2G

Kernel: es el núcleo de los sistemas Linux, Android al ser un sistema operativo basado en Linux también tiene un Kernel.

Launcher: es la parte encargada de la interfaz de usuario de Android.

RAM: (*Random Access Memory*) es una memoria limitada de nuestro dispositivo conectada directamente con la CPU. Tiene la capacidad de escribir y reescribirse infinitamente para agilizar los procesos de escritura, pero tiene almacenamiento limitado.

Rom: Es una copia original o modificada del sistema operativo en un archivo.

Root: es la acción de cambiar el sistema para poder acceder al mismo con permisos de administrador.

Scheduler: planificador que utiliza el *Kernel* para acceder al disco duro. Los schedulers sirven para Reducir al mínimo la latencia de búsqueda del disco duro, dar prioridad a las operaciones de I/O de algunos procesos, asignar más espacio en disco para los procesos en ejecución y garantizar que ciertas peticiones se ejecutan antes de un tiempo límite.

SDK (*Software Development Kit*): es un conjunto de herramientas de desarrollo que te permite crear aplicaciones para un sistema, como por ejemplo Java SDK o Android studio.

Script: o archivo de órdenes o archivo de procesamiento por lotes es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano. Los script son casi siempre interpretados.

Stock Rom: firmware originales de cualquier empresa.

Swap: es el área de intercambio de cualquier sistema Linux

Upgrade: actualización del software.

Widget: parte de algunas aplicaciones que son colocadas en nuestro escritorio para una mayor comodidad en su uso.

ANEXO

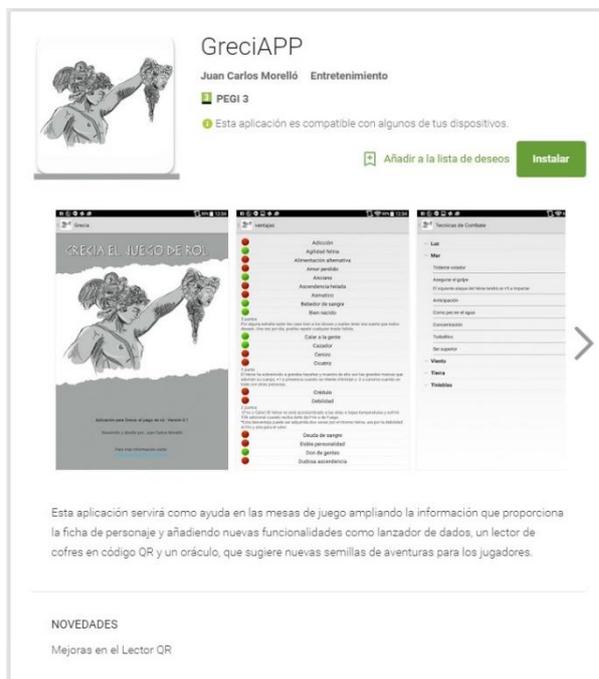
Manual de instalación en Android

Instalación desde Google Play

La mayoría de dispositivos Android llevan instalada de fábrica una versión de la aplicación Google Play, la tienda oficial de aplicaciones Android de Google. Esta aplicación permite descargar e instalar un millar de aplicaciones para este sistema operativo de una forma sencilla y cómoda.

Además el usuario puede instalar directamente la aplicación que desee en su dispositivo Android bien buscándola en la aplicación Google Play o bien en la web oficial de este servicio.

El caso de esta aplicación no es diferente, pudiendo proceder a su descarga mediante la aplicación de Google Play en desde cualquier navegador web, como puede apreciarse en la siguiente ilustración:



O directamente desde la aplicación de Play Store de cualquier dispositivo Android.

APLICACIÓN ANDROID PARA GRECIA: EL JUEGO DE ROL



Instalación desde descarga Directa - APK

Un APK es un archivo que constituye un paquete que se utiliza para distribuir e instalar aplicaciones y componentes para el sistema operativo Android.

Puede obtenerse directamente desde la opción de descargas del blog de Grecia: el juego de rol: <http://grecia.idr.blogspot.com> Y este paquete contiene todo lo necesario para poder instalar y ejecutar la aplicación (clases, recursos, AndroidManifest.xml, etc.).

Este modo de instalación es una alternativa a lo ofrecido por Google Play para quien no tenga acceso a la tienda de Google o tenga algún tipo de problema con ella.

De esta forma, al disponer del archivo .apk en nuestro Smartphone o Tablet Android sólo hay que seleccionarlo para ejecutar el instalador y esperar a que el proceso termine.