I.E.S. “AL-ÁNDALUS”

CICLO FORMATIVO DE GRADO SUPERIOR
DESARROLLO DE APLICACIONES MULTIPLATAFORMA

Proyecto Integrado

Desarrollo de una aplicación android, un sitio web y un Web Service.

Adrián Muñoz Molina

Almería, 2014

DESARROLLO DE UNA APLICACIÓN PARA DISPOSITIVOS MÓVILES EN EL SISTEMA OPERATIVO ANDROID, BASADA EN ARQUITECTURA CLIENTE-SERVIDOR QUE PERMITA LA REPRODUCCION ON-LINE DE AUDIOLIBROS Y GESTIONAR NUESTROS FAVORITOS ACCEDIENDO A UN WEBSERVICE, ASI COMO UN SITIO WEB Y LOS SERVICIOS WEB CORRESPONDIENTES.

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE TÉCNICO SUPERIOR EN DESARROLLO DE APLICACIONES MULTIPLATAFORMA.

Realizado por
Adrián Muñoz Molina

Dirigido por
Abel Ruiz Micó

ALMERÍA, Marzo de 2014

AGRADECIMIENTOS

A toda mi familia por la confianza que depositaron en mí, sobre todo a mis padres y hermanos.

A todos mis profesores, por todo lo que he aprendido gracias a vosotros.

Por último y no menos importante gracias a mi tutor Abel por su orientación y sus consejos durante todo el proyecto. Su disponibilidad y el buen trato tenido siempre conmigo han sido de gran valor para mí.

Gracias a todos.

Índice de contenidos

1. INTRODUCCIÓN. 5

1.1. DESCRIPCIÓN DE LA APLICACIÓN. 5
1.2. RESTRICCIONES GENERALES 6
1.3. REQUISITOS DE EFICIENCIA 7

2. ANÁLISIS DEL ENTORNO. 7

2.1. DESCRIPCIÓN DE LAS OPORTUNIDADES DE NEGOCIO. 7
2.2. OBLIGACIONES FISCALES, LABORALES Y DE PREVENCIÓN DE RIESGOS DEL PROYECTO. 8
2.3. AYUDAS O SUBVENCIONES. 8
2.3.1. Ayudas a Autónomos/as de nueva creación: 8
2.3.2. Línea de Ayudas del Instituto Andaluz de la Juventud: 8
2.3.3. MicroBank 9

3. ANÁLISIS DEL SISTEMA ACTUAL. 11

4. SOLUCIÓN PROPUESTA. 11

4.1 DIFERENTES TECNOLOGÍAS EXISTENTES PARA EL DESARROLLO Y PUESTA A PUNTO DE LA APLICACIÓN QUE SE DESARROLLARÁ. 11
4.1.1. CLIENTE 11
4.1.1.1. Entorno iOS (4.2) 12
4.1.1.2. Entorno Android (3.0) 12
4.1.1.3. Entorno BlackBerry OS (6.0) 12
4.1.1.4. Entorno Symbian Pt (Q4) 12
4.1.1.5. Entorno Windows Phone 7 12
4.2.2. SERVIDOR / WEB 13
4.2.2.1. Plataformas disponibles para servidores 13
4.2.2.2. Paneles de Control 13
4.2.2.3. Servidores Web 13
4.2.2.4. Lenguaje del lado del cliente 14
4.2.2.5. Lenguaje del lado del servidor 14
4.2.2.6. Sistemas Gestores de BBDD 15
4.2 ESTUDIO ECONÓMICO. VALORACIÓN DE PRESUPUESTOS. 16
4.3 ELECCIÓN Y JUSTIFICACIÓN DE LAS TECNOLOGÍAS SELECCIONADAS. 17
4.3.1 Cliente 17
4.3.2 Servidor / Web 18
4.4 RECURSOS HUMANOS Y MATERIALES NECESARIOS PARA DESARROLLAR EL PROYECTO. 20

5. PLANIFICACIÓN TEMPORAL DEL DESARROLLO DEL PROYECTO. 20

5.1 METODOLOGÍA Y PLAN DE TRABAJO 20
5.2 LISTA DE MODIFICACIONES O AMPLIACIONES FUTURAS QUE SE PODRÍAN APLICAR A LA APLICACIÓN PROYECTADA. 21

6. ESTUDIO DE LA VIABILIDAD DEL PROYECTO. 23

7. DOCUMENTACIÓN DEL DISEÑO E IMPLEMENTACIÓN DE LA SOLUCIÓN ADOPTADA. 27

7.1 PROTOTIPO DE LA APLICACIÓN. DISEÑO DE INTERFACES. 28
7.1.0. Arquitectura 28

7.1.1 Cliente. *28*
7.1.2 Servidor. *33*
7.2 DISEÑO LÓGICO DE LA APLICACIÓN A DESARROLLAR, DIAGRAMACIÓN UML O DE FLUJO DE DATOS. 39
7.2.1 Casos de uso *39*
7.2.1.1 Cliente *39*
7.2.1.2 Servidor. Web-Service *40*
7.2.1.3 Servidor. Sitio Web. *41*
7.2.2 Modelo de clases *42*
7.2.2.1. Web-Service *42*
7.2.2.2. Sitio Web. *43*
7.2.2.3. Cliente. AudioBooks. *44*
7.3 DESCRIPCIÓN MODULAR DEL SOFTWARE A DESARROLLAR 45
7.3.1 Servidor - Web Service. *45*
7.3.2. Servidor - Sitio Web. *45*
7.3.3. Cliente - AudioBooks. *47*
7.4 DISEÑO DE LA/S BASES DE DATOS, CON SUS DIAGRAMAS CORRESPONDIENTES. 49
7.4.1 Cliente *49*
7.4.2 Servidor / Web *50*
7.5 OTRAS ESTRUCTURAS DE DATOS QUE SE UTILIZAN EN LA APLICACIÓN. 51
Ficheros: *51*
SharedPreferences: *51*
***JSON** (JavaScript Object Notation)* *52*
7.6 ESTUDIO DE LA SEGURIDAD DE LA APLICACIÓN. 53
7.7 PUBLICACIÓN. 54
7.7.1 Publicar fuera de Google Play *54*
7.7.2 Publicar en Google Play *54*

8. CÓDIGO FUENTE DOCUMENTADO. 56

- *Anexo A.* *56*
- *Anexo B.* *56*

9. MANUAL DE CONFIGURACIÓN Y FUNCIONAMIENTO DE LA APLICACIÓN. 56

9.1 CLIENTE. 56
9.2 SERVIDOR 58

10. MANUAL DE USUARIO 63

10.1 INICIO DE LA APLICACIÓN 63
10.2 MENÚ PRINCIPAL. 63
10.3 REGISTRO. 65
10.4 FAVORITOS. 66
10.5 CATÁLOGO. 68
10.6 CONFIGURAR. 70
10.7. ACERCA DE… 72
10.8. SALIR. 72
10.9. MENSAJES DE LA APLICACIÓN 73

11. PLAN DE FORMACIÓN A LOS USUARIOS DE LA APLICACIÓN. 75

12. CONCLUSIONES. 75

BIBLIOGRAFÍA Y FUENTES DE INFORMACIÓN 76

LIBROS: 76
PÁGINAS WEBS: 76

GLOSARIO DE TÉRMINOS 78

Android *78*
API. *78*
Aplicación *78*
App. *78*
Avatar. *78*
Cookie. *78*
Cpanel. *78*
Cron. *78*
Eclipse *78*
Google Play *78*
Hosting *78*
HTML *78*
HTTP. *78*
JavaScript. *78*
JSON *79*
JVM. *79*
MP3. ..... *79*
MySQL. *79*
PHP *79*
PhpMyAdmin. *79*
SDK. *79*
Servicio Web. *79*
Smartphone. *79*
OAuth *79*
SQLite. *79*
UML. *79*
Vista *79*
Wi-Fi. *79*

ANEXO A: CÓDIGO FUENTE SITIO WEB Y WEB SERVICE 80

WEB SERVICE 81
Config.php *81*
connectbd.php *81*
crea_json.php *82*
db_descrip.php *83*
registro.php *84*
valida_User.php *85*
funciones_bd.php *86*
SITIO WEB 88
conec.php *88*
funciones.php *89*
index.html *95*
catalogo.html *99*
descargas.html *101*
mp3.html *104*

datos.html.php *107*
datos.php *111*
logeo.php *115*
olvido.html *119*
olvido.php *121*
registro.html *124*
registro.php *128*
restringida.htm.php *132*
style.css *137*

ANEXO B: CÓDIGO FUENTE APLICACIÓN "AUDIOBOOKS" ANDROID 139

AndroidManifest.xml *139*
SplashScreenActivity.java *140*
Principal.java *143*
Registro.java *154*
MusicPlayerActivity.java *161*
Utils.java *174*
AcercaDe.java *194*
Catalogo.java *195*
Configuracion.java *212*
DataBase.java *218*
Libros.java *222*
Favoritos.java *225*
ImageAdapter.java *233*
JSONParser.java *236*
Libroviewer.java *239*
Archivos.xml *245*

LICENCIA. 246

1. Introducción.

1.1. Descripción de la aplicación.

El proyecto se basa en una arquitectura cliente/servidor.

El cliente consistirá en una aplicación destinada a dispositivos móviles Android.

Dentro de las características que implementa la aplicación, el usuario podrá realizar diferentes operaciones:

- La reproducción on-line de audiolibros alojados en un servidor externo.
- Gestión de nuestros favoritos.
- Uso de la base de datos SQLite.
- Obtención del catálogo de audiolibros disponibles.
- Filtrado por título, autor o tema.
- Añadir a favoritos
- Registro del servicio.
- Configuración general.

El servidor, consistente en la realización de un *Sitio Web* www.audiolibros.hol.es en internet y un *Web Service* para servicio de la aplicación cliente.

El sitio Web permite a los usuarios que la visiten:

- Navegación por las distintas páginas del sitio.
- Visualizar el catálogo de títulos disponibles.
- Área para usuarios registrados.
- Registro y mantenimiento de usuarios.
- Control de sesiones y cookies.
- Descarga de la aplicación android.
- Descarga y reproducción de audiolibros.

¿Que es un web service y para que nos sirve?

La mayoría de los sitios webs grandes (Google, Bancos, Facebook, Microsoft, etc.) usan aplicaciones que utilizan servicios webs (web services). Un *web service* es un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Así que distintas aplicaciones de software desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet, esto significa que los ws aportan interoperabilidad.

La interoperabilidad se consigue mediante la adopción de estándares abiertos.

De manera más clara, un ws es un conjunto de métodos que se pueden invocar por alguna aplicación para realizar una tarea compleja, desarrollada en cualquier lenguaje y corriendo sobre diferentes plataformas. Para que la aplicación desarrollada pueda realizar muchas actividades con tan solo consumir los servicios y si es necesario enviar solo algunos parámetros para ejecutar operaciones complejas.

Nuestro Web Service realiza las siguientes funciones:

- Registro de usuarios desde la apps.
- Generar catálogo actualizado.
- Control de acceso de usuarios registrados.
- Facilita datos a la aplicación android.

Así pues, mediante la aplicación de dispositivos móviles y la parte web se pretende conseguir un mayor número de usuarios y difusión, permitiendo que el acceso a los datos pueda realizarse desde un ordenador, sin necesidad de disponer de un dispositivo móvil.

El desarrollo de la aplicación se puede dividir en dos partes, por un lado la aplicación Android destinada a dispositivos móviles y por otro lado, la parte del servidor en el que se incluye la base de datos y las diferentes funciones que la aplicación Android utiliza para obtener la información necesaria.

Para la aplicación Android se emplea el SDK Android utilizando el lenguaje de programación Java.

En cuanto al lado del servidor, se utilizará el servidor Apache y una base de datos MySQL debido a la alta compatibilidad entre estos dos sistemas y para la realización de las funciones que empleará la aplicación Android se empleará PHP para su desarrollo. Por otro lado el sitio web utiliza los lenguajes de programación HTML, CSS, Javascript y Php.

En cuanto a la aplicación Android en sí, el objetivo es desarrollar una aplicación de fácil acceso y aprendizaje, mostrando todas las opciones disponibles de la forma más sencilla. La presentación de la interfaz debe mostrarse igual independientemente de la resolución del terminal usado, por otro lado también deberá de ser compatible con el mayor número de versiones posibles del sistema operativo Android.

1.2. Restricciones generales

En cuanto a las restricciones que tiene el proyecto se puede mencionar la necesidad de trabajar con una base de datos con una estructura preestablecida. En caso de modificarse dicha estructura o eliminarse parcialmente, la aplicación funcionaría de manera incorrecta y no deseada o incluso podría dejar de ser funcional. En caso de tener que modificar la estructura de la base de datos, sería necesario realizar correcciones en la aplicación y/o en las funciones utilizadas para el acceso a la base de datos.

A la hora de emplear la aplicación hay que tener en cuenta que es necesario disponer de acceso a internet. Esto puede realizarse mediante conexión Wi-Fi o mediante conexiones móviles 3G. Hay que tener en cuenta que la velocidad con la que la aplicación responda a las peticiones dependerá tanto de la velocidad de la conexión como del rendimiento del servidor, así pues, siempre será mejor disponer de una conexión de alta velocidad y un buen servidor.

En cuanto a las dependencias cabe mencionar que la aplicación funciona únicamente en dispositivos móviles Android. En cuanto a la base de datos y las funciones utilizadas para acceder a ella son independientes del servidor, por lo que podría alojarse en un servidor con

cualquier sistema operativo siempre y cuando sea posible ejecutar Apache, interpretar PHP y alojar una base de datos MySQL.

1.3. Requisitos de eficiencia

El uso de la aplicación se realiza de forma individual, usando cada usuario su dispositivo móvil pero esto no evita que puedan existir varios usuarios accediendo de forma concurrente a los datos almacenados. Este servicio lo proporciona el propio servidor, de forma que se permita que varios usuarios accedan al servidor y por tanto a la base de datos. Se puede decir entonces que la eficiencia de la aplicación en el lado del servidor dependerá de éste.

Por otro lado, puesto que la aplicación se ejecuta en un dispositivo móvil la eficiencia de la propia aplicación dependerá de los requisitos de éste, a pesar de esto, cualquier terminal capaz de disponer del sistema operativo Android debería ser capaz de ejecutar la aplicación de forma eficiente.

2. Análisis del entorno.

2.1. Descripción de las oportunidades de negocio.

Según el artículo aparecido en diversas publicaciones digitales, entre ellas el http://www.eldiario.es/cultura/libros/Audiolibros-negocio-frustrado-Espana_0_208229793.html, comenta lo siguiente:

Audiolibros: el negocio frustrado en España

La venta de este formato en nuestro país sólo supone el 0,1% del total del mercado, según los datos de 2012 de la Federación del Gremio de Editores

Los grandes editores no apuestan por los libros para escuchar por sus altos costes de producción, los altos porcentajes de la distribución, los diferentes hábitos culturales con respecto a otros países y la negociación de los derechos de autor

El futuro, según los grandes sellos, son los libros multimedia que posean audio e imágenes

A la vista de la noticia anterior, las oportunidades de negocio que se presentan en la comercialización de audiolibros son poco esperanzadoras.

La distribución de audiolibros de contenidos sujetos a derechos de autor es por tanto prohibitivo.

Firmas como Canal Audiolibro o escuchalibros.com y audiolibro.es, que se encargan de producir y vender por su cuenta libros que no tienen derechos de autor (en su mayoría, clásicos como *La Celestina*), tienen muy difícil su supervivencia.

En España este formato se vende a través de la plataforma de Apple, iTunes. Audible.com no ha llegado a nuestro país, aunque sí tiene una parte en iTunes.

La oportunidad de negocio habría que buscarla en la recopilación y especialización temática de contenidos no sujetos a derechos de autor y apoyado en la publicidad.

2.2. Obligaciones fiscales, laborales y de prevención de riesgos del proyecto.

2.3. Ayudas o subvenciones.

2.3.1. Ayudas a Autónomos/as de nueva creación:

Cuantías desde 5000 €.

Concedidas por el Servicio Andaluz de Empleo. Consejería de Economía, Innovación Ciencia y Empleo.

Requisitos: Haber estado en situación de desempleo y estar dado/a de alta en el Régimen Especial de Trabajadores/as Autónomos/as durante los 6 últimos meses. Estas ayudas están destinadas fundamentalmente a la creación de empresas en nuevos yacimientos de empleo.

2.3.2. Línea de Ayudas del Instituto Andaluz de la Juventud:

Ámbito: Andaluz

Subvenciones a Jóvenes Andaluces o residentes en Andalucía hasta 35 años, para desarrollar proyectos empresariales en Andalucía (INNOVACTIVA 6000).

Subvenciones en régimen de concurrencia competitiva a jóvenes andaluces o residentes en Andalucía hasta 35 años, para poner en funcionamiento proyectos empresariales en Andalucía, a través de la constitución de empresas de nueva creación, que cuenten al menos con dos o más socios y cuyo domicilio social radique en Andalucía.

Requisitos de los aspirantes:

Jóvenes andaluces o residentes en Andalucía hasta 35 años, inclusive, con formación académica universitaria o de ciclo formativo de grado superior y cuyo proyecto esté avalado por un informe de viabilidad realizado por la Fundación Pública Andaluza "Andalucía Emprende», la Asociación de Jóvenes Empresarios (AJE), Cámaras de Comercio, la Confederación Empresarial Española de Empresas de Economía Social (CEPES), la Federación Empresarial Andaluza de Sociedades Laborales (FEANSAL), la Federación Andaluza de Empresas Cooperativas de Trabajo (FAECTA) o demás Entidades Públicas que certifiquen la viabilidad del proyecto.

Los proyectos empresariales que se constituyan deberán adoptar alguna de las siguientes formas jurídicas societarias: Sociedad Anónima, Sociedad de Responsabilidad Limitada, Sociedad de Responsabilidad Limitada Nueva Empresa, Sociedad Laboral, Sociedad Cooperativa.

Las empresas deben contar al menos con dos o más socios y el domicilio social radicar en Andalucía.

Dotación: La cuantía máxima de la subvención será de 6.000 euros.

2.3.3. MicroBank

Esta línea de financiación mediante microcréditos tiene un doble objetivo:

- El primero de ellos está enfocado a potenciar y fomentar el autoempleo, así como la actividad emprendedora, favoreciendo de esta manera, la igualdad de oportunidades de hombres y mujeres, jóvenes, nuevos residentes, personas con discapacidad y atendiendo con especial atención a los colectivos vulnerables.
- El segundo es financiar el establecimiento, la consolidación o la ampliación de microempresas, negocios de autónomos y proyectos de autoempleo con el objetivo de contribuir a desarrollar el tejido productivo, la creación y consolidación de empleo y la contribución al progreso social

Los interesados pueden acudir a cualquiera de los Centros de Apoyo al Desarrollo Empresarial (CADE) de Andalucía Emprende, Fundación Andalucía Emprende, donde se ofrecerá información y asesoramiento personalizado sobre esta línea de financiación ayudando en la preparación de las solicitudes de financiación y la documentación a presentar.

El microcrédito es una herramienta financiera que persigue reducir el desempleo y aumentar los ingresos productivos, estimulando el consumo y el crecimiento, permitiendo a su vez, reducir la exclusión financiera.

Destinatarios

- Personas con dificultades para acceder al sistema crediticio tradicional y con un proyecto empresarial viable de autoempleo.
- Autónomos con una renta anual de hasta 60.000 €.
- Personas jurídicas (Microempresas de hasta 10 trabajadores y con una facturación anual inferior a 500.000 €).

Es importante destacar que los destinatarios pueden ser personas emprendedoras, autónomos o autónomas ya dados de alta y personas jurídicas.

Características el crédito

El importe financiable es el 95% del proyecto con un importe máximo de 25.000 euros.

Tipo de interés: el 8,25 %.

Plazo de devolución: El plazo de devolución será de 60 meses. En este plazo se incluye un periodo de carencia opcional de 6 meses.

Garantías: Sin garantía real, ni condición de aval, salvo cuando el titular sea una persona jurídica, que se requerirá aval necesariamente.

Para mayor información puede acudir o ponerse en contacto con el CADE que tenga más cerca visitando el siguiente enlace http://www.andaluciaemprende.es/es/cades.

3. Análisis del sistema actual.

Dado que es una aplicación lúdica y no se integrará o complementará a ninguna aplicación existente, no se realiza análisis del sistema actual al no existir éste.

4. Solución propuesta.

4.1 Diferentes tecnologías existentes para el desarrollo y puesta a punto de la aplicación que se desarrollará.

4.1.1. Cliente

Tecnologías disponibles para dispositivos móviles.

Plataformas disponibles: Android, IOS, Windows, Symbian y otras.

Cuando se trata de analizar plataformas, una vez más Gartner no encuentra muchas sorpresas. Vuelve a ser Android el que domina la industria de los dispositivos móviles, gracias a la amplia variedad de smarpthones que corren con su sistema operativo, durante dos años más continuarán en la cresta de la ola. Este año se enviarán en total 866 millones de unidades Android, como hemos dicho antes, un tercio de los dispositivos totales enviados. Se calcula que Android llegará a los 1.060 millones de unidades vendidas en 2014, lo que equivale a casi la mitad de los dispositivos que se venderán durante ese año.

Table 2

Worldwide Devices Shipments by Operating System (Thousands of Units)

Operating System	2012	2013	2014
Android	505,509	866,781	1,061,270
Windows	346,464	339,545	378,142
iOS/MacOS	212,878	296,356	354,849
RIM	34,584	25,224	22,291
Others	1,118,004	820,592	689,877
Total	**2,217,440**	**2,348,497**	**2,506,429**

Source: Gartner (June 2013)

Global Smartphone UB by OS : % of Total	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017
Android	0.0%	0.3%	2.2%	13.0%	33.7%	49.8%	56.4%	57.5%	56.3%	54.2%	51.9%
Apple iOS	1.5%	5.1%	8.9%	11.8%	15.1%	18.3%	20.5%	21.1%	21.1%	20.8%	20.4%
Bada	0.0%	0.0%	0.0%	0.6%	1.4%	1.5%	1.4%	1.4%	1.4%	1.3%	1.3%
Blackberry OS	9.0%	11.2%	13.9%	13.7%	10.8%	6.8%	4.9%	4.2%	4.0%	3.9%	3.8%
Firefox OS	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.2%	0.6%	1.0%	1.4%	1.7%
LiMo	0.0%	2.2%	2.5%	1.9%	1.1%	0.5%	0.2%	0.1%	0.1%	0.0%	0.0%
MeeGo	0.0%	0.0%	0.0%	0.0%	0.1%	0.1%	0.1%	0.0%	0.0%	0.0%	0.0%
Microsoft	12.7%	12.7%	10.9%	7.2%	4.0%	3.9%	5.2%	7.8%	10.3%	12.5%	14.2%
Palm OS / webOS	1.6%	1.5%	1.1%	0.7%	0.2%	0.1%	0.0%	0.0%	0.0%	0.0%	0.0%
Symbian	64.8%	59.3%	54.7%	47.8%	32.0%	18.2%	10.2%	5.9%	3.6%	2.3%	1.5%
Tizen	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.3%	1.0%	1.7%	2.4%	2.9%
Others	10.5%	7.8%	5.7%	3.3%	1.5%	0.7%	0.4%	0.4%	0.7%	1.2%	2.3%
Total	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%

Así mismo cada plataforma dispone de su correspondiente lenguaje/s de programación y entorno de desarrollo.

4.1.1.1. Entorno iOS (4.2)

Fabricante: Apple
Tienda: App Store (+350.000 apps / 60.000 iPad)
Lenguajes: Objective-C, C, C++, HTML
Herramientas: Xcode + Interface Builder
Dispositivos: iPhone, iPod touch e iPad
Coste de desarrollo: SDK y simulador gratuito, licencia para publicar = 79€/año

4.1.1.2. Entorno Android (3.0)

Fabricante: Google
Tienda: Android Market (100.000 apps)
Lenguajes: Java, C, C++, HTML
Herramientas: Eclipse + Android Developer Tools (ADT)
Dispositivos: Nexus One, HTC Desire, Samsung Galaxy Tab
Coste de desarrollo: SDK y simulador gratuito, licencia para publicar = 25$/año

4.1.1.3. Entorno BlackBerry OS (6.0)

Fabricante: RIM (Research In Motion)
Tienda: App World (+ 10.000 apps)
Lenguajes: Java, HTML
Herramientas: Eclipse + Blackberry Java Plug-in
Dispositivos: Blackberry
Coste de desarrollo: SDK y simulador gratuito, licencia para publicar = 200$/10 apps

4.1.1.4. Entorno Symbian Pt (Q4)

Fabricante: Nokia
Tienda: Ovi Store (+ 6.000 apps)
Lenguajes: C++, Java, HTML
Herramientas: Nokia Qt SDK (C++), Eclipse + plug-ins (Java)
Dispositivos: Nokia X6, N8
Coste de desarrollo: SDK y simuladores gratuitos, licencia para publicar = 1€

4.1.1.5. Entorno Windows Phone 7

Fabricante: Windows
Tienda: Windows Phone Marketplace
Lenguajes: Silverlight, .NET(C#, Visual Basic.NET), XNA(C#)
Herramientas: Microsoft Visual Studio + Expression Blend
Dispositivos: HTC Surround, Samsung Focus
Coste de desarrollo: SDK y simuladores gratuitos, licencia para publicar = 99$/año (5 apps)

4.2.2. Servidor / Web

4.2.2.1. Plataformas disponibles para servidores

Fundamentalmente de dividen en dos: Unix y Windows.

Unix, disponible con diferentes distribuciones Linux: CentOS, Debian, Ubuntu, Red Hat, BSD, etc…

Windows, con sus diferentes versiones de Windows Server.

4.2.2.2. Paneles de Control

cPanel, es un panel de control comercial para la administración de servidores y web hosting de sitios web, cuentas de correo, base de datos, entre otros. cPanel es la interfaz gráfica para clientes y WHM la interface de administrador.

Plesk permite gestionar el alojamiento de tus dominios en el servidor. Plesk es un panel intuitivo y fácil de usar, evita tener conocimientos avanzados.

ISPConfig, es un panel de control para la administración de servidores Linux y web hosting en general. Le otorga un entorno grafico de administración similar a cPanel pero es completamente gratuito y de código libre.

4.2.2.3. Servidores Web

Apache, es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras.

Cherokee, es un servidor web multiplataforma. Su objetivo es ser rápido y completamente funcional, sin dejar de ser liviano comparado con otros servidores web.

Tomcat, es un servidor web con soporte de servlets y JSPs. Tomcat no es un servidor de aplicaciones. Incluye el compilador Jasper, que compila JSPs convirtiéndolas en servlets.

Internet Information Services o IIS es un servidor web y un conjunto de servicios para el sistema operativo Microsoft Windows.

4.2.2.4. Lenguaje del lado del cliente

HTML, siglas de HyperText Markup Language («lenguaje de marcas de hipertexto»), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que, en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, etc.

JavaScript, (abreviado comunmente "JS") es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

Las hojas de estilo en cascada o (Cascading Style Sheets, o sus siglas CSS) hacen referencia a un lenguaje de hojas de estilos usado para describir la presentación semántica (el aspecto y formato) de un documento escrito en lenguaje de marcas. Su aplicación más común es dar estilo a páginas webs escritas en lenguaje HTML y XHTML, pero también puede ser aplicado a cualquier tipo de documentos XML, incluyendo SVG y XUL.

4.2.2.5. Lenguaje del lado del servidor

Una aplicación del lado del servidor es cualquier programa o conjunto de instrucciones diseñadas con la finalidad de que un Servidor Web las procese para realizar alguna acción. Las aplicaciones del lado del servidor están escritas mediante algún lenguaje de programación, entre los que destacan:

Lenguaje	Fecha de primera versión estable	Sistema operativo	Última versión estable
PHP	1995	Multiplataforma	5.3.5
ASP.Net	1998	Windows (Algunas versiones)	4.0
Perl	1987	Multiplataforma	5.12.3
Python	1991	Multiplataforma	3.2.0
Ruby	1995	Multiplataforma	1.9.3-p125

El 75% de las aplicaciones del lado del servidor están escritas en PHP, seguido de ASP y las demás opciones usadas de forma alternativa y muy casual.

4.2.2.6. Sistemas Gestores de BBDD

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

PostgreSQL, es un SGBD relacional orientado a objetos y libre, publicado bajo la licencia BSD.

Oracle Database, es un sistema de gestión de base de datos objeto-relacional (u ORDBMS por el acrónimo en inglés de Object-Relational Data Base Management System), desarrollado por Oracle Corporation. Se considera a Oracle Database como uno de los sistemas de bases de datos más completos,

Microsoft SQL Server, es un sistema para la gestión de bases de datos producido por Microsoft basado en el modelo relacional. Sus lenguajes para consultas son T-SQL y ANSI SQL. Microsoft SQL Server constituye la alternativa de Microsoft a otros potentes sistemas gestores de bases de datos como son Oracle, PostgreSQL o MySQL.

4.2 Estudio económico. Valoración de presupuestos.

La compra de un dominio propio puede oscilar entre los 10€/año para un .es a los 12€/año para un .com, .net, .org, etc..

Hosting Web Empresarial a partir de 5,63€/mes.

Alta como desarrollador en Google Play. El registro como desarrollador Android en Google Play tiene un coste de 25 dólares en un sólo pago, no existe cuota anual.

Alta como desarrollador en App Store. El registro en el programa iOS Developer para poder publicar tus propias aplicaciones tiene un coste de 99 dólares anuales.

Otro asunto es el relativo a los audiolibros, si vamos a distribuir libros que no hayan prescrito sus derechos de autor habrá que tener en cuenta lo estipulado en la Ley de la Protección Intelectual,

Artículo 17. Derecho exclusivo de explotación y sus modalidades.

Corresponde al autor el ejercicio exclusivo de los derechos de explotación de su obra en cualquier forma y, en especial, los derechos de reproducción, distribución, comunicación pública y transformación, que no podrán ser realizadas sin su autorización, salvo en los casos previstos en la presente Ley.

Así mismo, el artículo 9.1 del Convenio de Berna reconoce a los autores el derecho de reproducción, y el 9.3 dispone que toda grabación sonora o visual será considerada como una reproducción.

Por lo que habría que negociar las cantidades a pagar a los autores por los mismos. Ya hemos visto anteriormente que en España empresas de primer nivel lo han intentado y no les ha sido rentable.

El software utilizado es libre o con licencias para educación, por lo que no representa ningún coste.

Rentabilidad: Desde el punto de vista económico la programación de aplicaciones para el sistema Android no cumple las expectativas. El usuario de estos teléfonos, cómo se vio posteriormente, está acostumbrado a recibir información (buscadores de Google) y dispositivos de forma gratuita o a muy bajo precio. La mayoría de aplicaciones descargadas y ofrecidas son versiones gratuitas. Esto es una situación muy diferente al caso de los usuarios del iPhone. Sólo pocos programadores pueden vivir de la programación para dispositivos Android.

4.3 Elección y justificación de las tecnologías seleccionadas.

4.3.1 Cliente

El JDK (Java Development Kit) es un software con herramientas de desarrollo para programar programas en Java. Este JDK es distribuido bajo la licencia GNU (General Public Licence/licencia para software libre).

Para Java existen diferentes entornos de trabajo. Algunos de ellos son de libre distribución (Open Source) mientras que otros son de licencia. Entre los entornos libres destacan Eclipse de Eclipse Foundation y NetBeans de Sun. Para la programación de éste proyecto se utilizó el entorno de programación de Eclipse.

Para la programación de aplicaciones para móviles Android hace falta además la instalación de un plugin llamado Android Development Tools (ADT). Este plugin permite una configuración rápida de proyectos, la creación fácil del User Interfaces (interfaces de usuarios), depuración mediante herramientas del Android SDK (Android Software Development Kit), exportación de ficheros de instalación, etc.

JDK (Java Development Kit): en primer lugar es necesario tener instalado Java, la versión mínima del JDK debe ser la 5. Existen versiones para 32 y 64 bits.

Android SDK: mientras que el JDK nos proporciona lo necesario para utilizar las clases de Java, el SDK nos dará acceso a las clases necesarias para construir una aplicación *Android*. Para ello lo descargamos de la página oficial de *Android* en el apartado comentado anteriormente, seleccionando la versión correspondiente a la máquina en la que se vaya a trabajar (*Mac*, *Windows* o *Linux*). Esto nos permite guardar un archivo comprimido, el cual únicamente hay que descomprimir y guardar la carpeta resultante en el directorio en el que queramos tener el SDK.

Android de la empresa Google como sistema operativo bajo el cual la aplicación se ejecuta. En la actualidad, junto con iOS, Android es el sistema operativo móvil más utilizado y con un mayor número de aplicaciones disponibles y desarrolladores, lo que lo convierte en un sistema operativo con un amplio número de usuarios.

Eclipse IDE: es un entorno de desarrollo para compilar y ejecutar aplicaciones.

AVD Manager, sirve para crear dispositivos virtuales.

La justificación principal está condicionada por ser software libre y haber sido el utilizado durante el curso, por lo que se conoce sus posibilidades y funcionamiento. Además el resto de sistemas y software al ser desconocidos implicaría de más tiempo para obtener una curva de aprendizaje hasta alcanzar el nivel de los sistemas elegidos para implementar el proyecto elegido.

4.3.2 Servidor / Web

El software para el Servidor (Web y Web Service) vendrá condicionado por el que se encuentre disponible para las distintas modalidades de hosting elegido o contratado para montar el servicio.

En mi caso me he decidido por una cuenta de alojamiento gratuito de la empresa *"Hostinger"* que aunque no totalmente funcional con respecto a las cuentas de pago sí que mantiene unas buenas características, cubriendo las necesidades mínimas para poner en funcionamiento el proyecto planteado. Ofrece subdominios gratuitos.

Características:

Espacio de Disco	2000 MB
Tráfico de Datos	100 GB
Número de Sitios	¡Ilimitado!
Sencillo Creador de Sitios	Si
Instalador Automático	50 Scripts
Copias de Seguridad de Datos	Limitado
Garantía de Tiempo en Línea	99%
Soporte PHP 5.2, 5.3, 5.4	Si
Bases de Datos MySQL	2
Soporte de Base de Datos InnoDB	No
Conexiones MySQL Remotas	No
Herramienta phpMyAdmin	Si
Acceso FTP	Si
Usuarios FTP	2
Consola Web SSH	Si
Tareas Programadas	Si
Curl y Curl SSL	Si
Streaming de Audio/Video	Si
Dominios Aparcados	2
Subdominios	2
Casillas de Correo	2
Acceso a Webmail	Si
Acceso POP3 / IMAP	Si
Acceso SMTP	No
Filtros de Spam	Si
Forwarding de Correo	Si
Autorespondedores de Email	Si

Asistencia y Soporte Básico	Si
Velocidad de la Red del Servidor	10 mbps
RAM del Servidor	8 GB
Procesador del Servidor	Xeon E3-1230
Configuración de Disco Duro	RAID-1
Sistema Operativo	Centos 6.2
Administrador de Cuentas de Correo	Si
Copia de Seguridad/Restaurar Base de Datos	Si
Copia de Seguridad/Restaurar Sitio	Si
Editor de Zona DNS	Si
Consola Web SSH	Si
Estadísticas del Sitio	Si
2 Administradores de Archivos	Si
Dominios Aparcados	Si
Dominios Adicionales	Si
Redirecciones de Dominio	Si
Páginas de Error Personalizadas	Si
Directorios Protegidos por Contraseña	Si
Administrador de Bloqueo de IP	Si
Publicidad	No

- Servidor con S.O. Linux Centos 6.2
- Servidor Web Apache 2.2
- Panel de Control cPanel.
- Lenguaje de Programación de servidor PHP 5.3
- Base de Datos MySQL 5.1.61
- Administrador de BBDD PhpMyAdmin 4.1.3
- Lenguajes HTML5, CSS y JavaScript
- Cliente FTP Filezilla 3.7.3

4.4 Recursos humanos y materiales necesarios para desarrollar el proyecto.

Los recursos humanos están compuestos por:

- Un programador, analista y diseñador (titular del proyecto).

Los recursos materiales empleados son:

- Un ordenador portátil (Intel Dual Core 2.1 GHz, 3GB RAM, 320GB HD).
- Un ordenador de sobremesa (Intel I3, 8GB RAM, 500GB HD).
- Conexión a Internet ADSL.
- Dominio y hosting (Gratuito).
- Plataforma Windows 7 Professional. Licencia MSDN.
- Entorno de desarrollo Eclipse.
- ADT de Android y emulador AVDs (Android Virtual Devices).
- Diverso software de apoyo (Cliente FTP, retoque imagen, etc…).
- Smartphone HTC con Android 2.1.

5. Planificación temporal del desarrollo del proyecto.

5.1 Metodología y Plan de Trabajo

En este primer apartado se van a entrar a detallar las pautas que se han seguido para hacer el Proyecto que se presenta. Para ello se indican tanto la metodología y el modelo que se han usado a la hora de construir la aplicación, como el orden que se ha seguido en el plan de trabajo a lo largo de la realización del proyecto. Por último detallamos al final de cada tapa una estimación del tiempo que se ha necesitado para realizar las tareas que se proponen.

• *Fase de documentación y estudio*. Lo primero que necesitamos es introducirnos en el mundo de aplicaciones Web, Servicios Web que se necesitan utilizar, API de Google que vamos a utilizar y diseño de la solución a adoptar. Tiempo estimado: 4 semanas.

• *Creación del Sitio Web.* Entre el diseño y la implementación del código y pruebas necesarias para su puesta en funcionamiento en internet. Tiempo estimado: 8 semanas.

• *Creación de la base de datos*. En este segundo ciclo se tratarán de guardar en la base de datos información de los datos que vamos a necesitar para el funcionamiento del proyecto. Tiempo estimado: 1 semana.

• *Consumir Servicios Web*. En este ciclo trabajaremos sobre los servicios Web. El objetivo es que tanto la aplicación Web como la aplicación de Android puedan consumir los servicios Web para consultar, modificar o borrar datos de la base de datos. Tiempo estimado: 2 semanas.

• *Aplicación cliente*. Programación de los diferentes módulos que componen la aplicación, diseño de interfaces y realizar la depuración correspondiente. Tiempo estimado: 8 semanas.

• *Memoria del proyecto*. Documentación de los diferentes puntos que componen el guión de la memoria explicativa del proyecto. Tiempo estimado: 8 semanas.

5.2 Lista de modificaciones o ampliaciones futuras que se podrían aplicar a la aplicación proyectada.

Esta aplicación está preparada para futuras ampliaciones, las cuales pueden ser tanto de mejora de la interfaz como de incorporación de nuevas funcionalidades.

Permitir la descarga y almacenamiento local de los audiolibros favoritos para la reproducción sin conexión.

Respecto a la seguridad de la conexión, se puede mejorar en dos aspectos básicos:

- Por un lado, que el servidor permita conexión mediante el protocolo *HTTPS.*
- Por otro lado, que se pueda establecer el método de autenticación *OAUTH*, dado que como se ha comentado *Basic* es muy básico a nivel de seguridad. Con el uso de este protocolo se evitaría tener que estar enviando usuario y contraseña.

Plantearse la posibilidad de implementar la aplicación para otros Sistemas Operativos.

Crear un panel de control para el Sitio Web para actualizar y modificar los contenidos, independiente de phpMyAdmin.

Dado que utilizamos, entre otros, los audiolibros disponible en leemp3.com, crear un script para actualizar nuestra base de datos con la última versión disponible para posteriormente generar nuestro catálogo.

Permitir a los usuarios crear y subir contenidos al servidor, supervisado por el administrador, para su publicación.

ACTIVIDAD	OCT	NOV	DIC	ENE	FEB	MAR
Diseño de la solución a adoptar.						
Desarrollo de la aplicación.						
Puesta a punto de la aplicación.						
Fase de pruebas.						
Confección de la Memoria del Proyecto y Presentación						
Formación de los usuarios finales.						
Lista de modificaciones o ampliaciones futuras						
Plan de mantenimiento de la aplicación						

Cronograma del desarrollo del proyecto.

6. Estudio de la viabilidad del proyecto.

Dado que el fundamento de la aplicación es la reproducción de “audiolibros” habrá que ver en qué forma afecta en este ámbito el tema de los derechos de autor y reproducción.

Derecho de autor

El símbolo del *copyright* "©" se usa para indicar que una obra está sujeta al derecho de autor.

El **derecho de autor** es un conjunto de normas jurídicas y principios que regulan los derechos morales y patrimoniales que la ley concede a los autores (los *derechos de autor*), por el solo hecho de la creación de una obra literaria, artística, musical, científica o didáctica, esté publicada o inédita.

Está reconocido como uno de los derechos humanos fundamentales en la Declaración Universal de los Derechos Humanos.

En el derecho anglosajón se utiliza la noción de ***copyright*** (traducido literalmente como "derecho de copia") que —por lo general— comprende la parte patrimonial de los derechos de autor (derechos patrimoniales).

Una obra pasa al dominio público cuando los derechos patrimoniales han expirado. Esto sucede habitualmente trascurrido un plazo desde la muerte del autor (*post mortem auctoris*). El plazo mínimo, a nivel mundial, es de 50 años y está establecido en el Convenio de Berna. Muchos países han extendido ese plazo ampliamente. Por ejemplo, en el Derecho europeo, son 70 años desde la muerte del autor. Una vez pasado ese tiempo, dicha obra entonces puede ser utilizada en forma libre, respetando los derechos morales.

En España se conoce como Ley de la Propiedad Intelectual a lo que los ordenamientos jurídicos denominan derecho de autor. La vigente Ley de Propiedad Intelectual (LPI) data de 11 de noviembre de 1987. Tras algunas reformas y la aprobación de varias leyes especiales, en 1996 se llevó a cabo, mediante el Real Decreto Legislativo 1/1996, una refundición que ya ha sido objeto de modificaciones posteriores por la Ley 5/1998 y las Leyes 19 y 23/2006. Una característica fundamental de dicho ordenamiento es que se configura al derecho de autor como único, pero integrado por varias facultades; así: de reproducción, comunicación, distribución, transformación... Una propiedad esencial del derecho de autor en España es que tiene por objeto un bien inmaterial: la obra.

En la actualidad, y tal como establece la LPI, puede decirse de modo general que, en el caso más simple y frecuente de un solo autor, los derechos de explotación de la obra duran toda la vida del autor y 70 años después de su muerte o declaración del fallecimiento. En caso de obras con varios autores ("obras en colaboración"), los 70 años cuentan a partir de la muerte del autor que muera el último. En los casos de obras con varios autores pero editadas y

divulgadas bajo un único nombre ("obras colectivas"), obras seudónimas y obras anónimas, los 70 años cuentan desde la fecha de publicación.

Sin embargo, hay que considerar que la ley de 1879 establecía un plazo de protección de las obras de 80 años a partir de la muerte del autor, lo que ha sido respetado en la LPI de 1987 mediante varias disposiciones transitorias. Esto hace que el plazo efectivo de la mayoría de las obras cercanas a la expiración sea de 80 años (será así hasta 2057).

Situación actual.

El audiolibro no existe en España. No hay apenas oferta ni inversión por parte de las editoriales, agentes, autores y distribuidores. Y, por tanto, las ventas son irrisorias. Según el informe de Comercio Interior de la Federación del Gremio de Editores (FGEE) de 2012, los sellos facturaron 214,70 millones de euros en otros soportes diferentes al libro en papel; de ellos, los audiolibros representaron el 0,1%, en torno a los 210.000 euros. Muy lejos incluso de ese 3% de facturación que ya suponen los libros electrónicos.

Los grandes grupos editoriales como Planeta, Penguin Random House –antiguo Mondadori- y Santillana no se han querido embarcar en estos proyectos y a día de hoy no tienen ni un solo título en su catálogo. Sólo ha habido una mínima apuesta por parte de Planeta con algunos bestsellers, pero como asegura a eldiario.es su director de área digital y Mass-Market, Santos Palazzi, "no ha funcionado. Las ventas son residuales y, además, nos hemos gastado lo que no está escrito. Es más bien un experimento de ensayo-error".

Tampoco les va muy bien a las pequeñas empresas creadas por emprendedores que pensaron que el audiolibro explotaría con la llegada de la tecnología digital. A pesar de la existencia de los podcast y de un consumo mucho más flexible de la escucha, firmas como la ya citada Canal Audiolibro o escuchalibros.com y audiolibro.es, que se encargan de producir y vender por su cuenta libros que no tienen derechos de autor (en su mayoría, clásicos como *La Celestina*), tienen muy difícil su supervivencia.

Las razones del fracaso

El interrogante que se plantea es por qué nunca ha triunfado este formato en España, por qué no se ha apostado y por qué tampoco parece que se vaya a arriesgar por él en el futuro. Las razones son de todo tipo:

Hábitos culturales: Según comentan Carmen Ospina, Iría Álvarez y Santos Palazzi, responsables de las áreas digitales de Penguin, Santillana y Planeta, respectivamente, en España no se hacen largos trayectos en coche que permitan ir escuchando un libro mientras conducen. "En EEUU tardan horas en llegar al trabajo por las largas distancias y eso hace que uno pueda ir escuchando tranquilamente un audiolibro", confirma Palazzi. Con respecto al norte de Europa, el razonamiento tiene que ver con las horas de luz. "Si ya desde el mediodía es de noche, tienen que ocupar su tiempo en algo mientras hacen otras tareas en casa", añade este directivo.

El coste de producción: Crear un audiolibro no es barato. Necesitas uno o dos locutores como mínimo, un estudio de grabación, un técnico de sonido y un equipo para recrear los efectos especiales. *"La hora de grabación cuesta unos mil euros, por lo que si un audiolibro sale a unas diez horas, tienes que gastarte 10.000 euros. Si luego lo vas a vender a cinco euros, los márgenes de beneficios no llegan"*, sostiene Rafael Martínez. En Planeta ponen los costes en 30.000 euros por audiolibro.

La distribución: En España este formato se vende a través de la plataforma de Apple, iTunes. Audible.com no ha llegado a nuestro país, aunque sí tiene una parte en iTunes. No obstante como señalan productores y editores, los porcentajes que maneja la firma comprada por Jeff Bezos son "insultantes". *"Son unas condiciones salvajes, ellos se quedan con el 80% y sólo el 15% va a la editorial.* Con esos porcentajes es imposible", denuncia Victoria Mesas. A día de hoy sería posible comprar en audible.com, pero solamente poseen un 1% de audiolibros en español y funciona mediante una suscripción de 14,95 dólares al mes.

Los derechos de autor: Para crear un audiolibro es necesario tener los derechos del libro en audio. "Y cuando lo quieres hacer tienes que comprarlos, y no es fácil. *Por eso la gente hace audiolibros pero con clásicos que están en dominio público, y eso el lector no es lo que busca"*, manifiesta Gabi Morales, cuya firma está más enfocada a los audiolibros de empresa.

El idioma: Es un problema que no resulta baladí. Si bien no hay obstáculos en leer un libro en español pero con vocabulario mexicano, colombiano o argentino, la escucha del acento ya es otra cosa. *"Tienes que pensar muy bien en qué idioma narras. ¿Un español va a aguantar el acento mexicano, por ejemplo?* Y, además, creo que aquí faltan escuelas de locutores para audiolibros porque tienes que saber dramatizar", señala Carmen Ospina, de Penguin. También Rafael Martínez considera esta cuestión un impedimento notable: "Nosotros siempre indicamos que son audiolibros con un acento latino, porque sabemos que mucha gente se va a sentir extrañada".

Visto lo anteriormente expuesto, la obtención de los recursos realmente "atractivos" para el usuario no serían rentables lo que haría inviable el proyecto.

No obstante, el funcionamiento de la aplicación en sí, es independiente de este tipo de recursos aunque orientada a la reproducción de audiolibros se puede utilizar para cualquier otro tipo de recursos cómo podcast, música y otro cualquiera siempre que sea distribuido y soportado desde el servidor y exentos de derechos de autor y reproducción.

Habría, quizás, el derecho de copia con reserva de derecho de autor por la interpretación oral de la misma. En el caso de ser nosotros o un empleado nuestro el locutor quedaría exento.

Hay que mencionar que cada distribuidora ofrece sus propias aplicaciones para reproducción.

Utilizándolo de la forma antes expuesta, existen múltiples audiolibros o audios exentos de derechos de autor aunque según lo antes mencionado sin mucho atractivo para el usuario en general.

Generalizando, se implementará la aplicación que permite la gestión de favoritos y su reproducción de audiolibros alojados en un servidor que contiene los enlaces a los recursos sin derechos de autor.

7. Documentación del diseño e implementación de la solución adoptada.

En este capítulo se explica paso a paso el desarrollo completo de *AudioBooks*, una aplicación cuyo objetivo es la reproducción on-line de audiolibros en formato mp3 alojados en un servidor y la gestión de nuestros favoritos. Así cómo la creación de un sitio Web y el Web Service.

Mediante este desarrollo se busca ilustrar de una forma más práctica las características principales que ofrece Android.

Algunos de los aspectos más interesantes de Android y que se explican con esta aplicación son los siguientes:

- Uso de componentes *Activity*.
- Solicitudes a través de *Intents*.
- Uso de componentes *Mediaplayer*.
- Comunicaciones por HTTP.
- Uso de la base de datos SQLite.
- Composición del archivo "AndroidManifest.xml".
- Acceso a información sobre el propio dispositivo.
- Composición de interfaces de usuario, tanto con código como con XML.
- Declaración y uso de recursos externos.
- Composición gráfica de elementos en pantalla.
- Gestión de opciones de menú.
- Control de la conexión Wi-Fi.

El sitio web *www.auliolibros.hol.es* desarrollado en un hosting remoto contiene las diversas páginas web disponibles así como el Web Service que dará soporte a la app android.

Los aspectos y características más destacadas utilizadas en su desarrollo son:

- Creación de cuenta en Hosting remoto y administración del entorno.
- Programación en Html, Css, Php y JavaScript.
- Uso de Base de Datos Mysql.
- Uso del Panel de control del Hosting mediante CPanel.
- Uso del administrador de Bases de Datos Mysql mediante PhpMyAdmin.
- Automatización de procesos mediante tareas Cron.
- Utilización de Sesiones de usuario y Cookies.
- Gestión de usuarios registrados.
- Intercambio de información.
- Validación de usuarios.

7.1 Prototipo de la aplicación. Diseño de interfaces.

7.1.0. Arquitectura

A continuación se ofrece un diagrama con la arquitectura general del sistema completo:

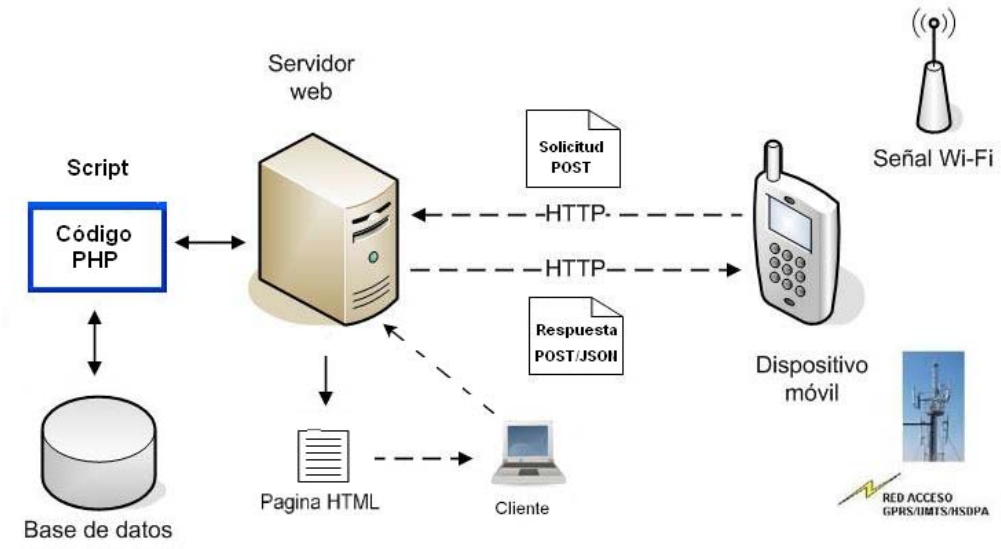


Figura: Arquitectura del sistema completo.

Se pueden apreciar los siguientes elementos:

Dispositivo móvil: dispositivo con el sistema Android corriendo y donde está instalada la aplicación *AudioBooks*. A través de la conexión Wi-Fi realiza intercambios de información con el Web Service.
Cliente / PC: Cualquier cliente conectado a internet y accede al sitio Web de audiolibros.
Datos JSON: el dispositivo móvil realiza intercambios de datos con el servidor utilizando el protocolo HTTP.
Servidor web: permanece siempre a la espera de conexiones por parte de los diferentes usuarios de *AudioBooks*. Aloja el Sitio Web y el Web Service que procesa las peticiones y respuestas Post / Json.
Script: pequeño programa en Php que recibe peticiones, consulta la base de datos, y construye y envía las respuestas de vuelta al dispositivo móvil.
Base de datos: la base de datos almacena toda la información de audiolibros y de los usuarios de *AudioBooks.*

7.1.1 Cliente.

Apps Android *AudioBooks 1.0* cuyo objetivo básico es la reproducción on-line de audiolibros almacenados en un servidor remoto accedido mediante un Web Service, así como la gestión de favoritos. Para ello utiliza fundamentalmente tres características:

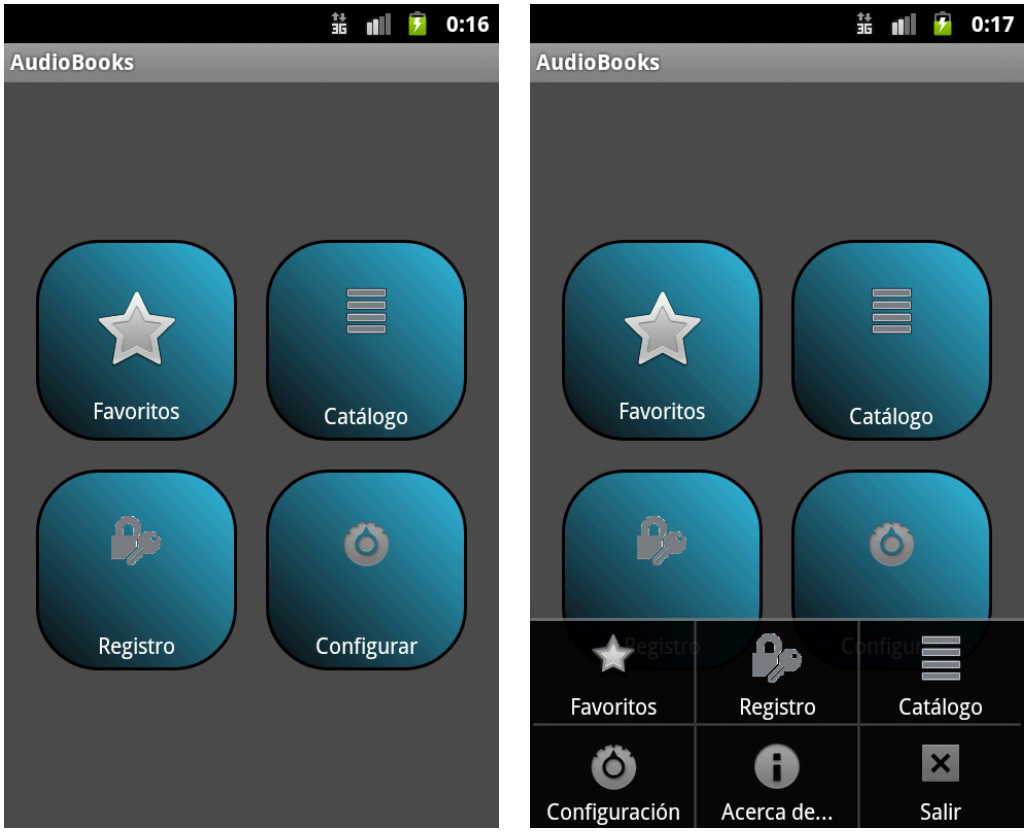
Almacenamiento de datos en el dispositivo mediante la base de datos *SQLITE*.

Uso de *MediaPlayer* para implementar un reproductor para los audiolibros en formato mp3.

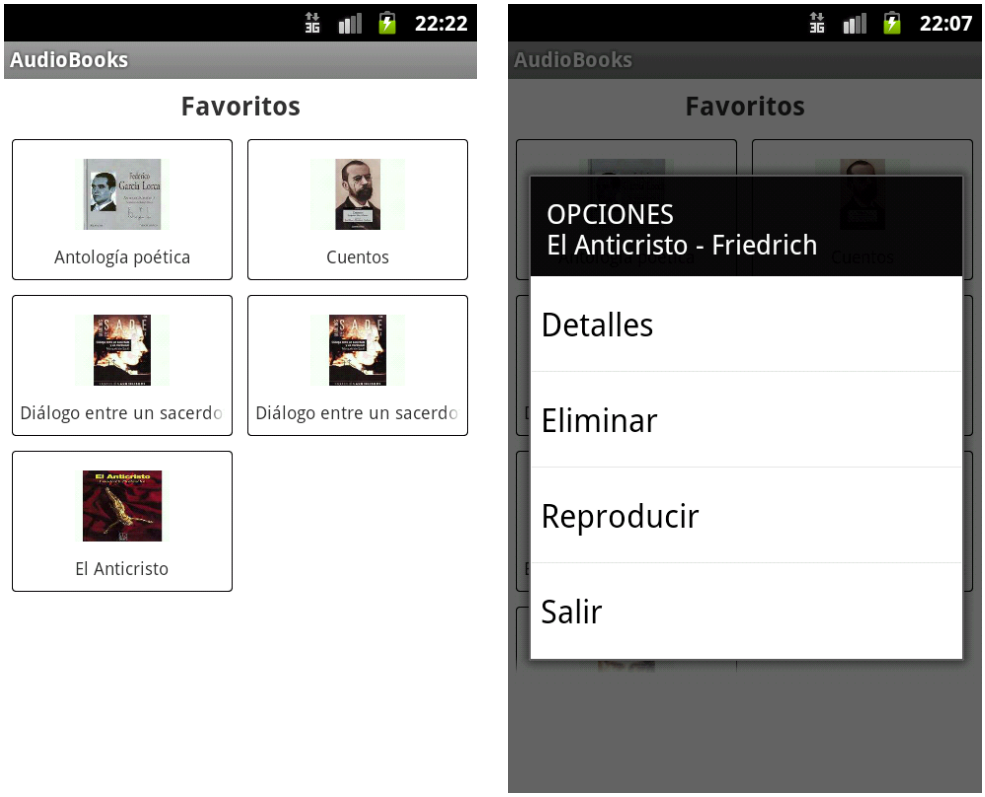
Por otro lado, mediante el establecimiento de una **conexión a Internet** a través de 3G o Wi-Fi, la aplicación *AudioBooks* realiza intercambios de información. Este intercambio se realiza sobre el Web Service ubicado en un servidor remoto utilizando el estándar JSON y sobre el protocolo HTTP.

Implementa el registro del usuario con el servidor, la gestión de favoritos, la consulta del catálogo de audiolibros disponibles y la reproducción de los mismos.

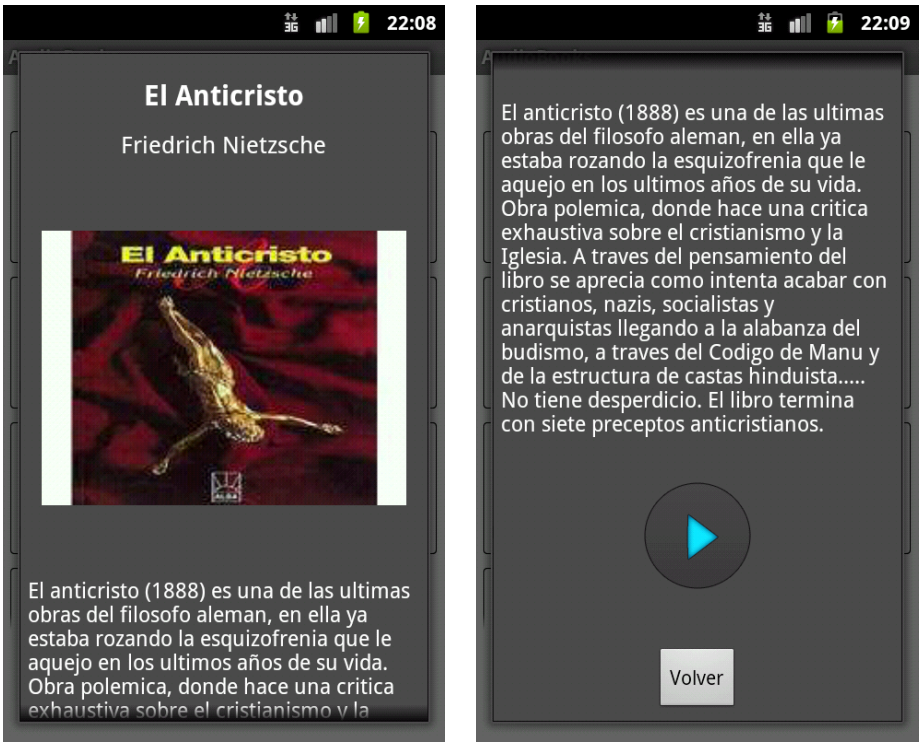
A continuación se muestran las distintas interfaces de que dispone la aplicación en sus diferentes opciones. Se analizan detalladamente en el manual de usuario descrito en el punto 10 de esta memoria.



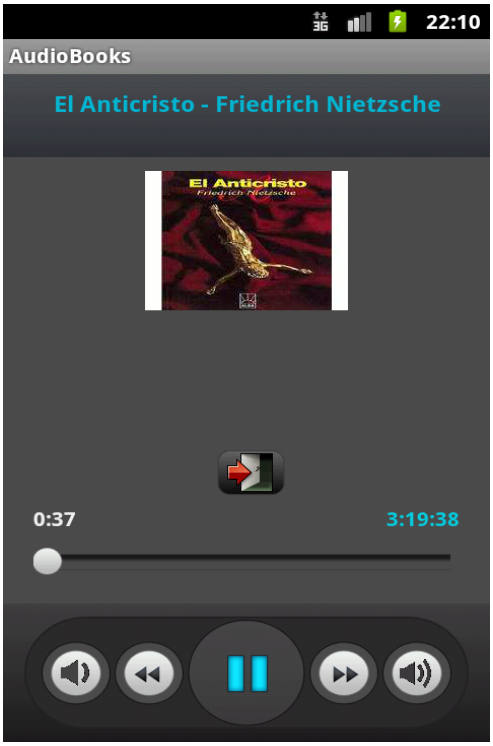
Interfaces: Menús.

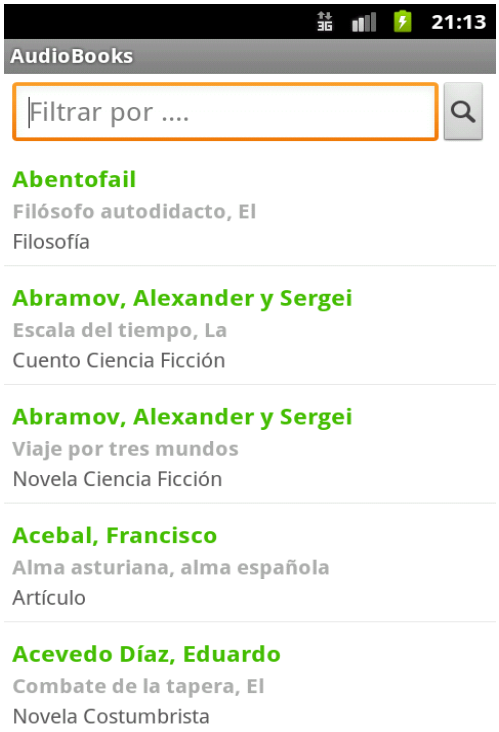


Interfaces: Favoritos y menú contextual de opciones.

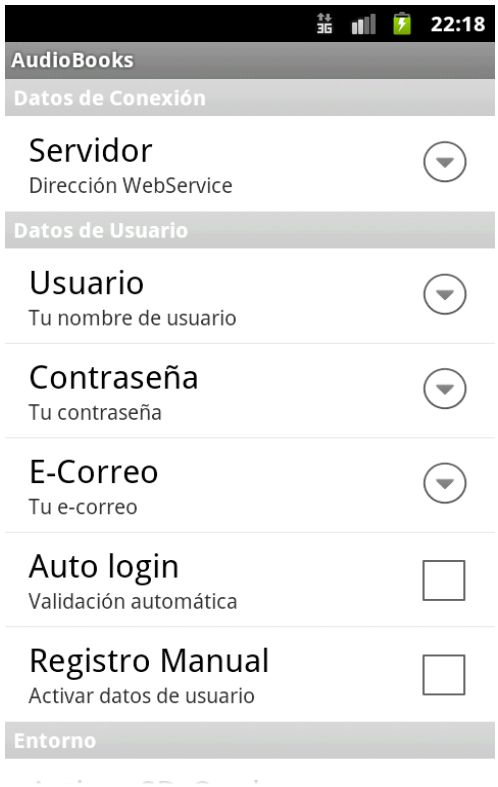


Interfaces: Detalle de audiolibros.





Interfaces: Reproductor de audio y lista de catálogo.



Interfaces: Registro y Configuración.

Estructura de carpetas:

La estructura de carpetas del código fuente del proyecto es la siguiente:

es.hol.audiolibros (carpeta del proyecto).
/assets (aquí se almacenan los recursos que utiliza la aplicación).
/bin (clases compiladas).
/libs (librerías utilizadas).
/res (es el directorio principal de recursos, aquí guardamos las imágenes que utiliza la aplicación).
/drawable-Xdpi (recursos gráficos que utilizará la aplicación en función de la resolución de la pantalla del dispositivo en el que se esté utilizando).
/layout (ficheros xml que definen las vistas que utilizará la aplicación).
/values (ficheros xml que definen valores de la aplicación, como los textos, los colores o las *url* de los servicios Web).
/src (aquí se almacenan los archivos de código fuente Java).
AndroidManifest.xml (archivo donde se define lo que puede hacer la aplicación).
default.properties (fichero de proyecto de Eclipse).

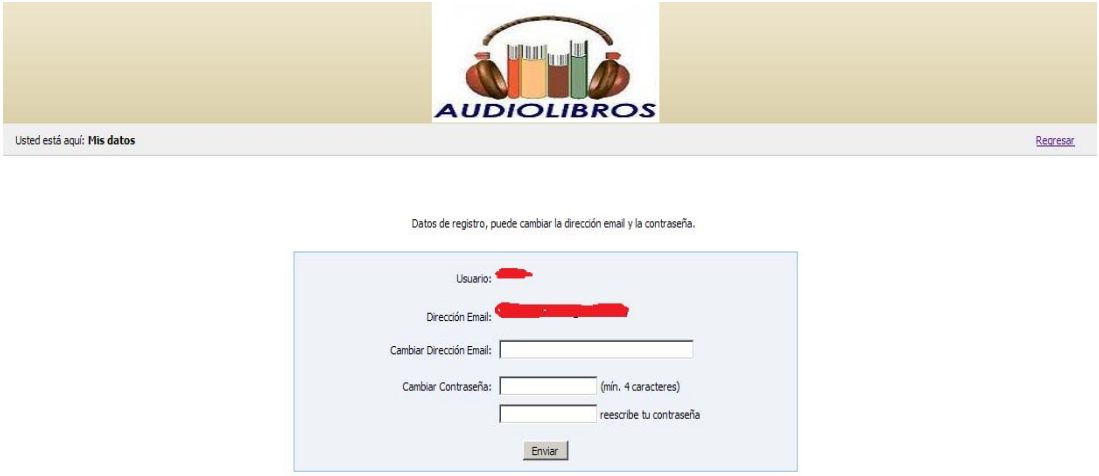
7.1.2 Servidor.

Diseño de un Sitio Web programado en HTML, CSS, PHP y JAVASCRIPT, usa una base de datos MySql con varia tablas. El sitio permite la visita y consulta del catálogo de libros disponibles, además dispone de un área para usuarios registrados desde donde tenemos acceso a la descarga y reproducción de audiolibros y la descarga de la App android. Permite la gestión del perfil del usuario y hace uso de sesiones y cookies.

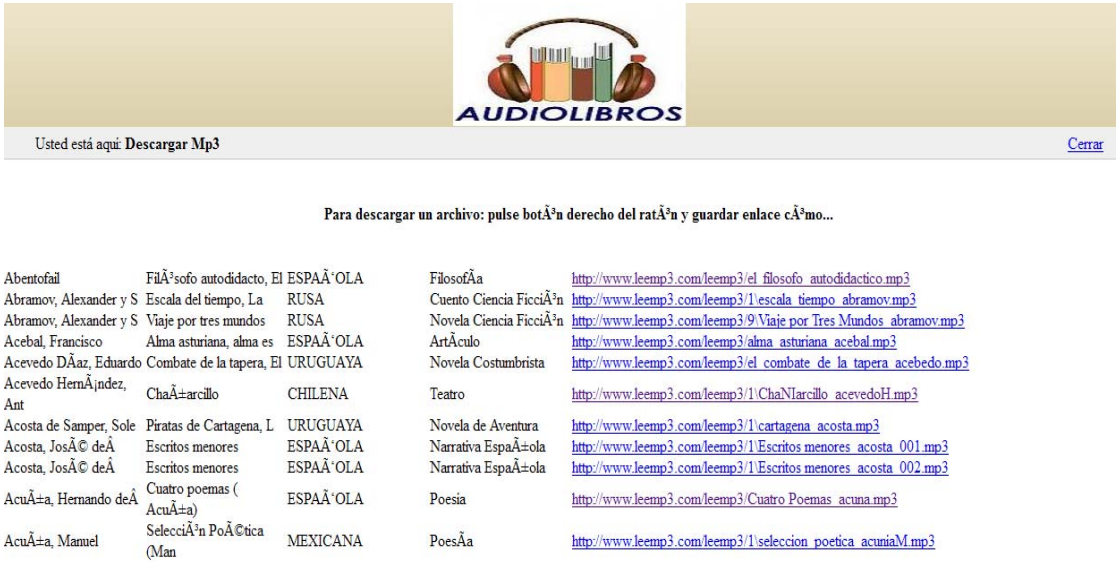
Página principal del Sitio Web.

Interfaces: Opción “Entrar”.

Área de usuarios registrados.

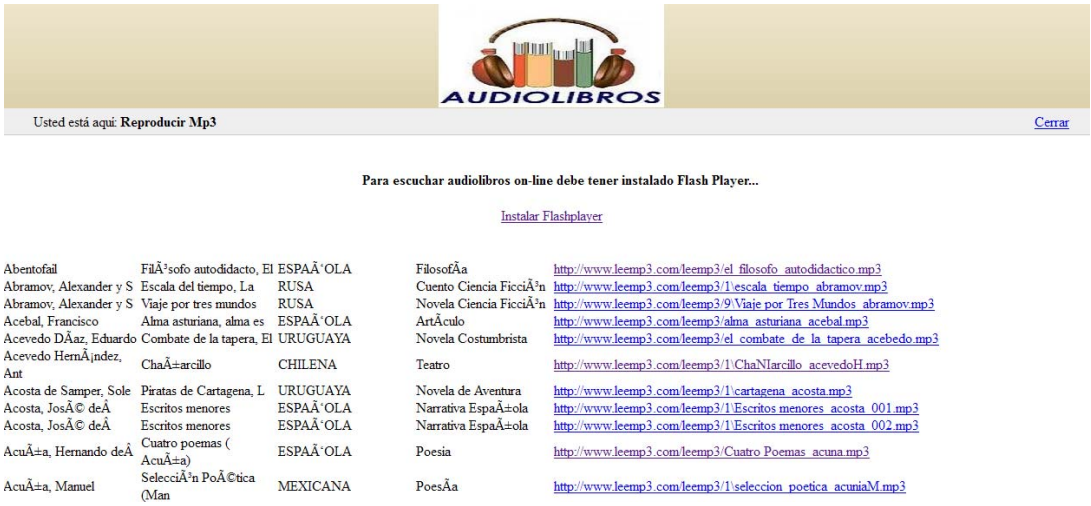


Interfaces: Datos de Usuario.



Descarga de audiolibros.

Descargar App.



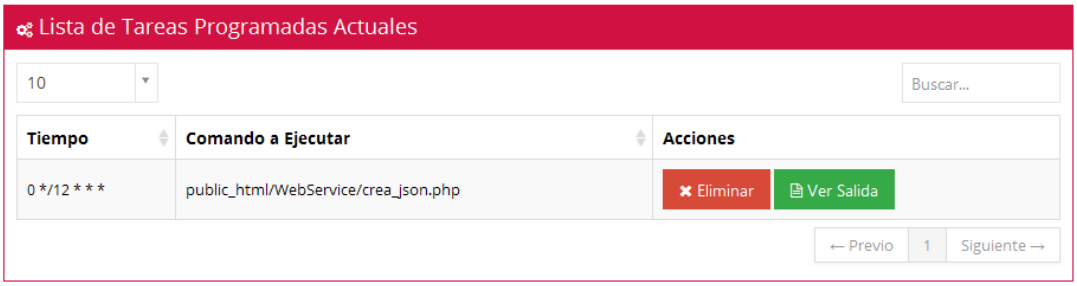
Reproducir audiolibros.

Interfaces: Opción “Registro”.

Opción “Acerca de”.

Crear el Web Service que dará soporte a la aplicación android, programado en PHP accediendo a una base de datos MySQL y que suministra información mediante JSON además de funciones de control. No dispone de interfaces.

En el servidor, mediante una tarea programada, se actualiza el catálogo periódicamente ejecutando un script del Web Service. Generándolo de esta forma siempre existe una versión actualizada del catálogo en el servidor, que posteriormente será descargada por el usuario en su dispositivo ganando tiempo.



Estructura de carpetas:

La estructura de carpetas en el servidor es la siguiente:

/fotolibro, contine las imágenes de cada audiolibro.
/images, contiene las imágenes del sitio Web.
/libros, contiene los archivos .mp3 de los audiolibros.
/WebService, contiene los scripts en php de los servicios.

Ficheros Html, css y PHP del sitio Web.

7.2 Diseño lógico de la aplicación a desarrollar, diagramación UML o de Flujo de datos.

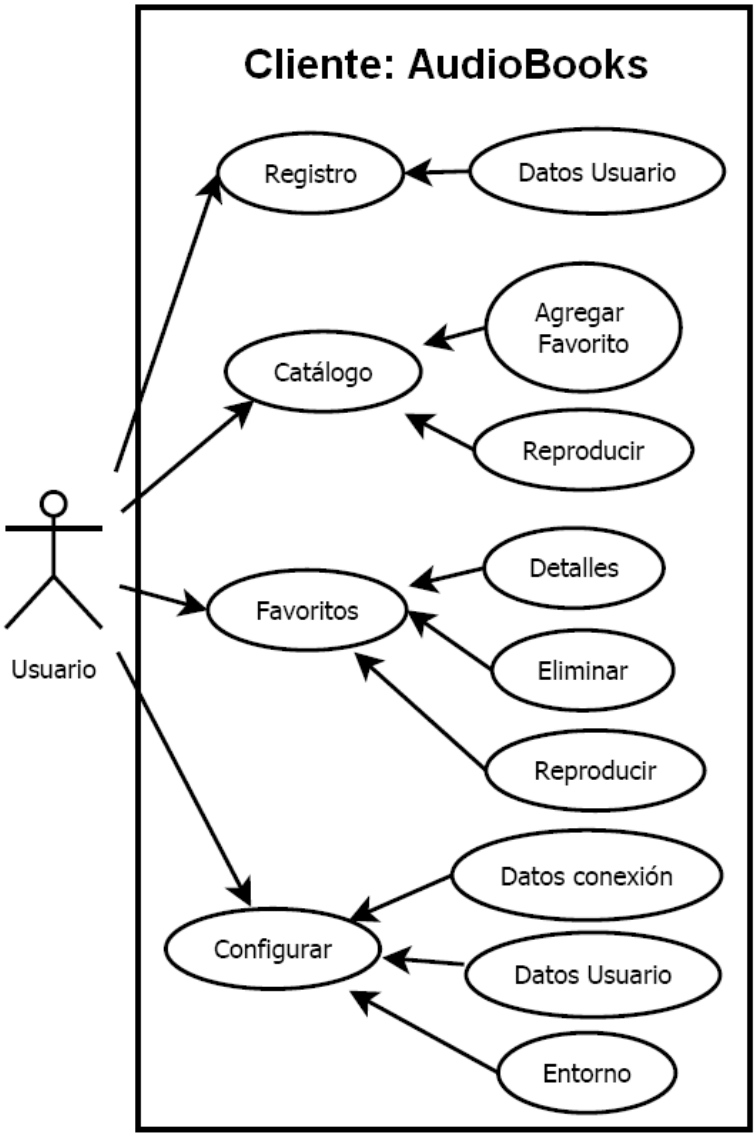
A continuación se detallan los documentos y datos analizados y utilizados en el desarrollo de la aplicación.

7.2.1 Casos de uso

En la etapa de análisis se hace uso de la metodología UML. El primer punto es la realización de los casos de uso. En este caso, como característica o rasgo, podemos destacar que al existir únicamente un usuario todas las acciones los lleva a cabo el mismo, por lo tanto solo es necesario un diagrama de caso de uso para especificar toda la funcionalidad de la aplicación.

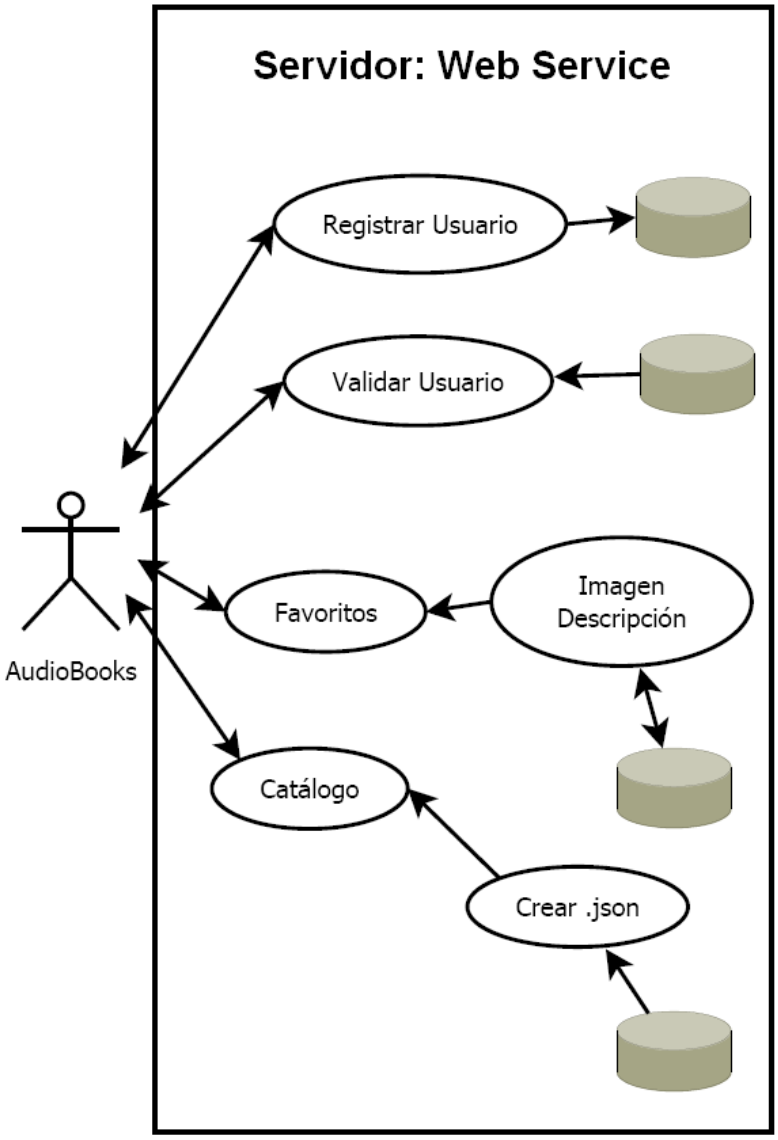
7.2.1.1 Cliente

A continuación se muestra un diagrama con los casos de uso asociados a la aplicación cliente *AudioBooks,* utilizando el estándar UML 2.0.

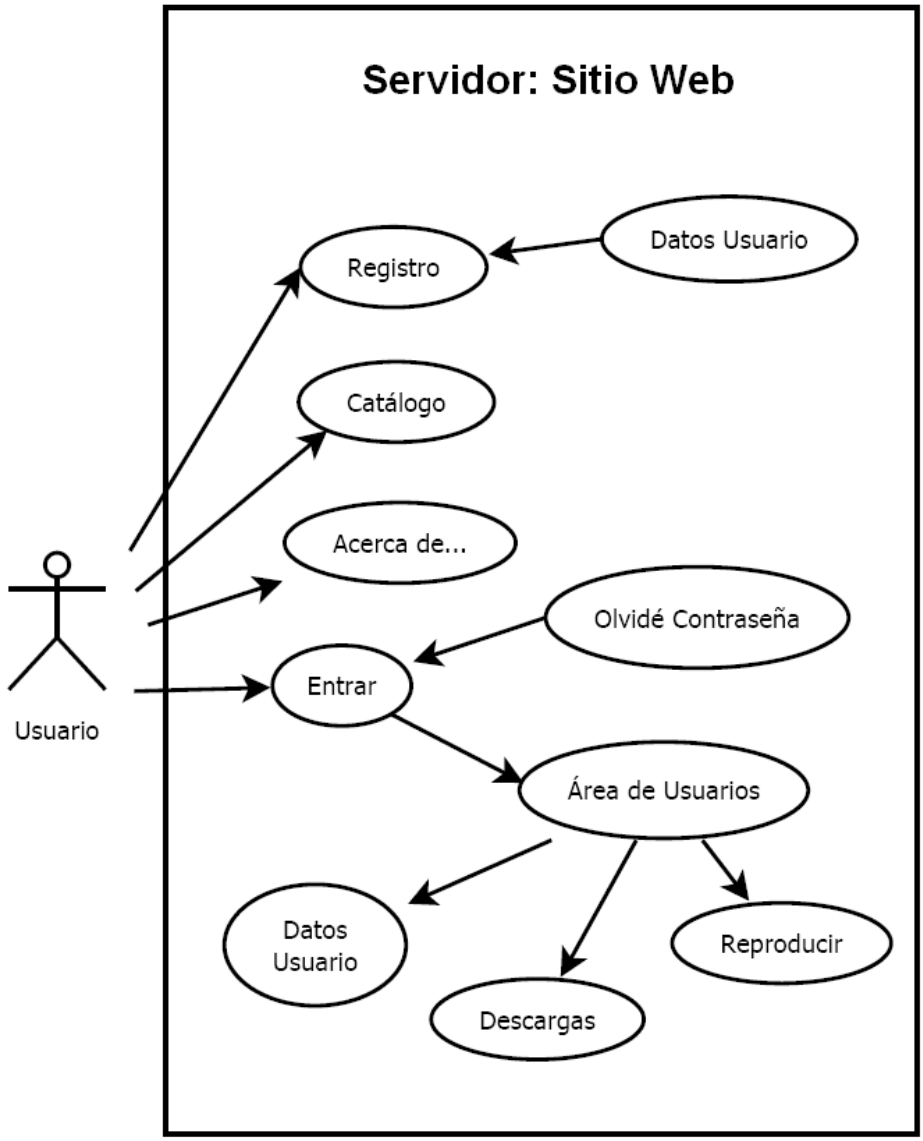


7.2.1.2 Servidor. Web-Service

Siguiendo el mismo criterio, el siguiente diagrama expresa los casos de uso asociados al servidor "Web Service" al que se conecta la aplicación AudioBooks para obtener la información:



7.2.1.3 Servidor. Sitio Web.



Es importante recalcar que los casos de uso simplemente expresan el punto de vista del usuario sobre cómo debe funcionar la aplicación y qué puede realizar a través de ella, y son una forma de facilitar su comprensión. No tiene porqué existir ninguna correspondencia entre los casos de uso y la clases finalmente implementadas, más allá de que las clases en su conjunto, como sistema completo, realizan aquello que los casos de uso expresan.

7.2.2 Modelo de clases

Las clases utilizadas para resolver un determinado problema, sus atributos y métodos, la visibilidad que de estos tienen las demás clases, así como las relaciones que existen entre ellas y sus colaboraciones, constituyen el modelo de clases. Mediante este tipo de modelos se expresa, con mayor o menor nivel de detalle, la futura implementación del sistema, así como permite dar una idea bastante cercana a la forma en la que se ha abordado el problema.

En el presente apartado se ofrece al lector una breve descripción de cuál es el modelo de clases utilizado en la aplicación *AudioBooks,* así como el sitio Web y Web-Service. El objetivo principal es ayudar a comprender cómo se soluciona el problema planteado. Por todo ello, el modelo de clases ofrecido no incluye más que clases, atributos, métodos, y la visibilidad de estos, con vistas a no perder al lector con aspectos demasiado complejos.

En la figura siguiente se enseña el diagrama correspondiente al modelo de clases utilizado, siguiendo el estándar UML 2.0.

7.2.2.1. Web-Service

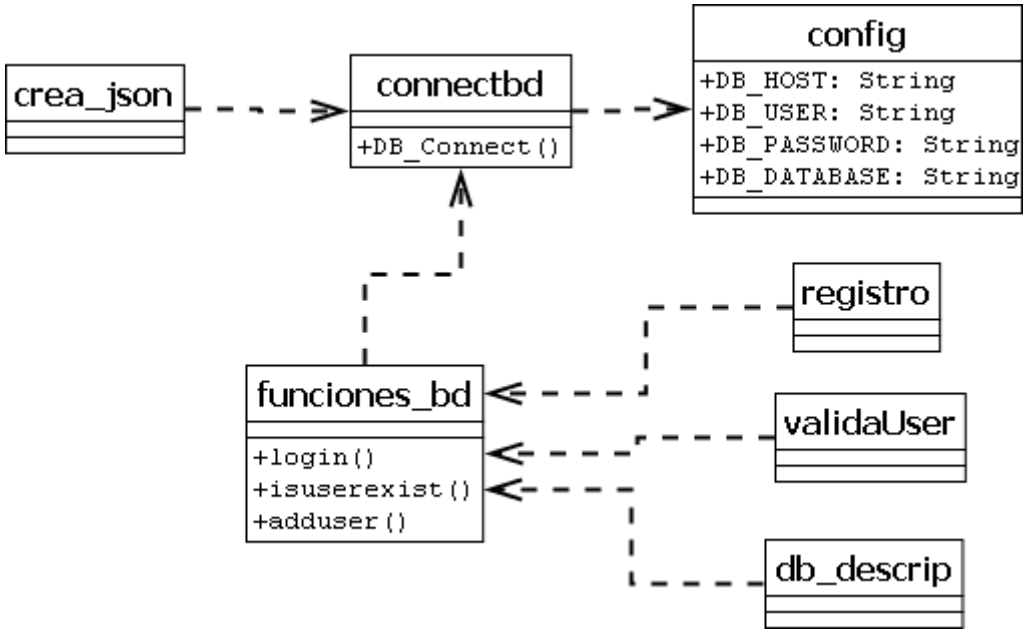


Diagrama de clases Web-Service.

7.2.2.2. Sitio Web.

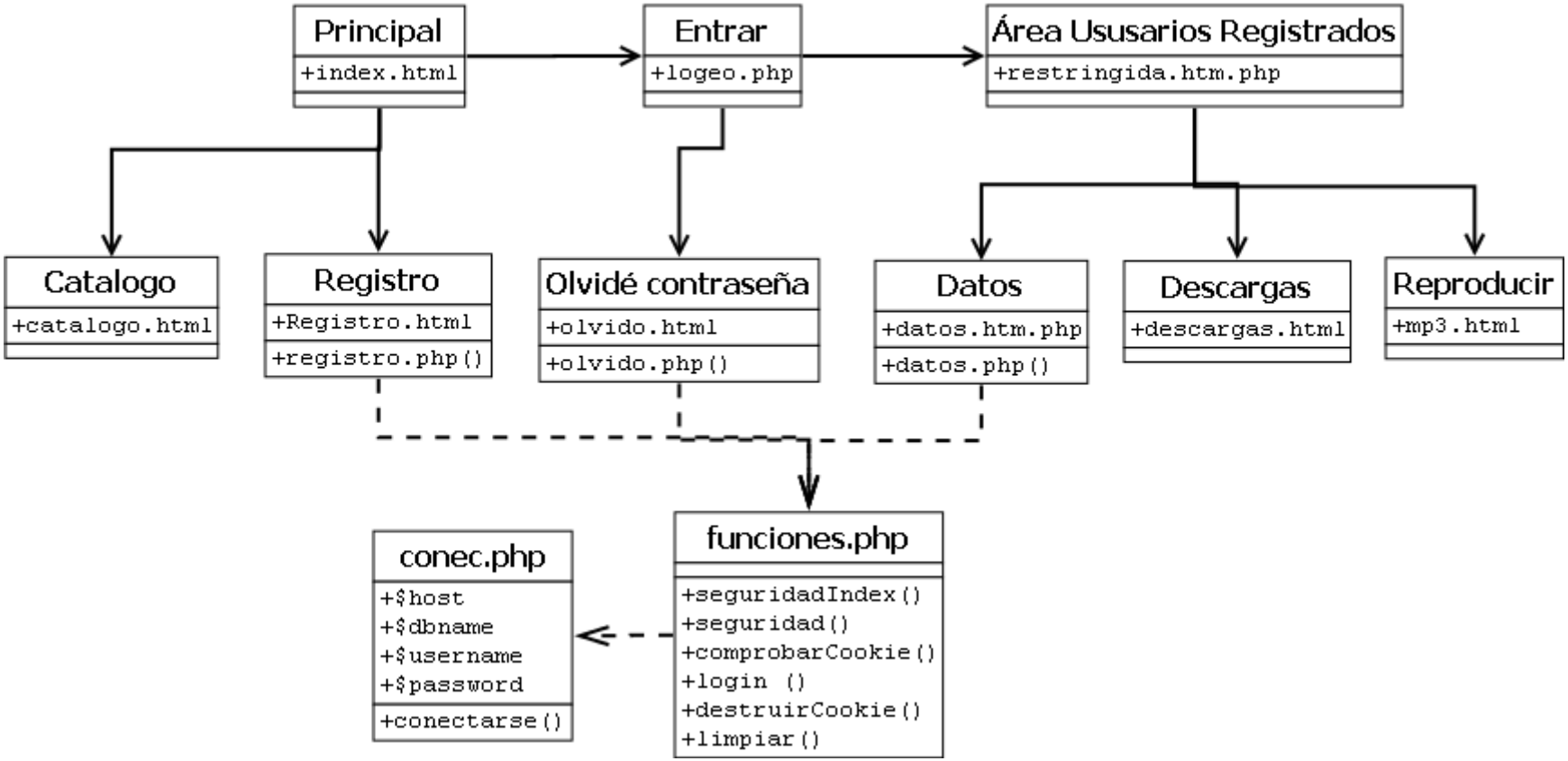


Diagrama de clases Sitio Web.

7.2.2.3. Cliente. AudioBooks.

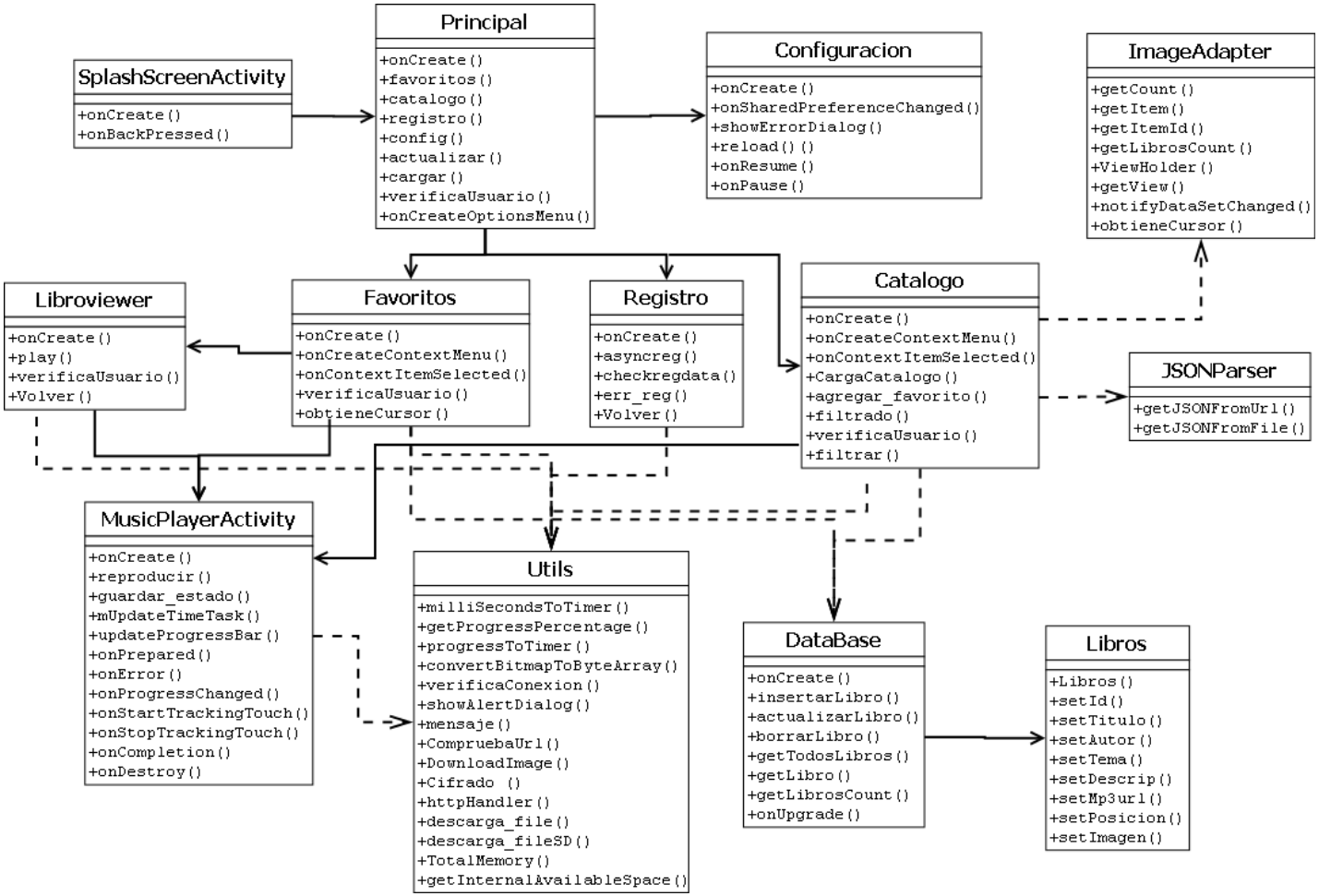


Diagrama de clases AudioBooks.

7.3 Descripción modular del software a desarrollar

7.3.1 Servidor - Web Service.

- **config.php,** contiene la definición de las constantes para la cadena de conexión con la Base de Datos.
- **connectbd.php,** se encarga de conectarse a la Base de Datos.
- **crea_json.php,** función que genera el archivo .jon con la información de audiolibros disponibles para el catálogo.
- **registro.php,** realiza el registro en la base de datos de usuarios con los datos recibidos, realizando las validaciones correspondientes.
- **validaUser.php,** valida que los accesos sean de usuario registrados.
- **db_descrip.php,** devuelve datos complementarios para la ficha de favoritos.
- **funciones_db.php,** contiene funciones de uso general.
    - **login,** verifica que los datos de usuario y contraseña sean correctos.
    - **isuserexist,** comprueba que el nombre de usuario no exista.
    - **adduser,** añade un nuevo usuario a la base de datos con los datos de registro.

7.3.2. Servidor - Sitio Web.

- **index.html,** página principal del sitio web.
- **catalogo.html,** muestra el catálogo de audiolibros disponibles.
- **registro.html,** página web que muestra el formulario de datos de registro.
- **registro.php,** código asociado al formulario anterior que comprueba y valida los datos, si son correctos realiza el registro y envía un correo.
- **logeo.php,** muestra el formulario para entrar al área de usuarios registrados, realiza las validaciones.
- **olvido.html,** muestra un formulario para solicitar una nueva contraseña en caso de olvido.

- **olvido.php,** código asociado al formulario anterior que se encarga de validar usuario y enviar por correo una nueva contraseña generada por el sistema.
- **restringida.htm.php,** página principal para el área de usuarios registrados. Hace uso de sesiones de usuario y cookies.
- **datos.htm.php,** muestra el formulario con los datos actuales del usuario y los que puede modificar.
- **datos.php,** código asociado al formulario anterior que comprueba, valida y actualiza los datos de usuario en la base de datos.
- **descargas.html,** muestra el catálogo de audiolibros disponibles para poder descargarlos.
- **mp3.html,** muestra el catálogo de audiolibros disponibles para reproducirlos, es necesario tener instalado flash player (se indica el enlace).
- **conec.php,** función que define la cadena de conexión a la base de datos.
- **funciones.php,** contiene diversas funciones de uso general.
    - **seguridadIndex,** Comprueba que exista una sesión o una cookie en la página de login.
    - **seguridad,** comprueba que exista una sesión o una cookie, sino redirige al login.
    - **comprobarCookie,** comprueba si existen cookies válidas según los datos de la base de datos.
    - **login,** valida los datos de usuario y contraseña ante el sistema, en caso de querer ser recordado en el pc, crea la cookie.
    - **destruirCookie,** destruye la cookie actualizando los datos en la base de datos.
    - **limpiar,** limpia los datos pasados de cadenas extrañas.

7.3.3. Cliente - AudioBooks.

- **SplashScreenActivity,** esta clase implementa la pantalla de presentación de la aplicación mostrando una imagen, nombre y dirección aplicándole transiciones hasta cargar el menú principal de la aplicación. Se aprovecha el tiempo para inicializar y configurar parámetros de la aplicación.

- **Principal,** muestra el menú principal de la aplicación representado por cuatro botones con las opciones principales y otras que se muestran en menú secundario.

- **Favoritos,** esta clase permite gestionar los audiolibros favoritos, dispone de un menú de opciones (Detalles, Reproducir, Borrar, Salir). Muestra los elementos en un GridView.

- **Registro,** esta clase realiza el registro en el servidor con los datos introducidos por el usuario.

- **Configuracion,** sirve para modificar la configuración de parámetros de la aplicación. Esta clase extiende de PreferenceActivity.

- **Catalogo,** clase que se encarga cargar y convertir los datos del archivo .json descargado del servidor o que se encuentra en el dispositivo. Muestra los datos en un ListView y dispone de un menú de opciones (Agregar a favoritos, Reproducir, Quitar filtro, Salir). Además dispone de la posibilidad de aplicar filtros (Autor, Titulo, Tema).

- **Libroviewer,** depende de la clase Favoritos y muestra información detallada del audiolibro seleccionado. Permite reproducir y salir.

- **ImageAdapter,** es el adaptador para el Listview utilizado en la clase Catalogo.

- **JSONParser,** se encarga de traducir el archivo .json a la estructura de datos necesaria para su representación.

- **DataBase,** esta clase se encarga de abrir la base de datos si existe o de crearla si no existe. Incluso de actualizar la versión si decidimos crear una nueva estructura de la base de datos. Extiende de SQLiteOpenHelper e implementa los métodos necesarios para insertar, actualizar, modificar, obtener y contar los libros.

- **Libros,** define el tipo Libros que contiene la estructura de campos definida en la BD.

- **MusicPlayerActivity,** es el reproductor de audio utilizado por las clases Favoritos, Libroviewer y Catalogo. Implementa las funciones de reproducir, pausar, avanzar, retroceder, volumen +-, barra de progreso desplazable y salir. Nos da la posibilidad de guardar el punto de reproducción cuando salimos del reproductor.

- **Utils,** clase que contiene métodos de uso general utilizados por la mayoría de las otras clases.
    - **httpHandler,** se encarga de realizar la petición http al servidor enviándole la información y obteniendo la respuesta.
    - **CompruebaUrl,** comprueba si existe un recurso especificado devolviendo el código de respuesta y el tamaño del recurso.
    - **DownloadImage,** establece una conexión para descargar la imagen y devuelve un Bitmap.
    - **verificaConexion,** comprueba si existe conexión a internet devolviendo un booleano.
    - **showAlertDialog,** muestra un diálogo de alerta con mensaje.
    - **mensaje,** muestra un mensaje centrado en la pantalla.
    - **Cifrado,** se utiliza para cifrar contenido utilizando diferentes algoritmos cómo MD5, SHA-1, SHA-256 y otros.
    - **milliSecondsToTimer,** convierte el tiempo en milisegundos al formato de horas, minutos y segundos. Se utiliza en la clase MusicPlayerActivity.
    - **getProgressPercentage,** calcula el porcentaje de avance de la barra de progreso. Se utiliza en la clase MusicPlayerActivity.
    - **progressToTimer,** devuelve la duración actual en milisegundos. Se utiliza en la clase MusicPlayerActivity.
    - **convertBitmapToByteArray,** convierte un bitmap a formato de bytearray.
    - **Descarga_file,** se utiliza para descargar, en el dispositivo, el catálogo en la ruta especificada.
    - **Descarga_fileSD,** se utiliza para descargar, en la SD, el recurso especificado.
    - **TotalMemory,** devuelve la cantidad de memoria del dispositivo.
    - **getInternalAvailableSpace,** devuelve la memoria libre del dispositivo.

7.4 Diseño de la/s bases de datos, con sus diagramas correspondientes.

Bases de datos: cuando los datos a almacenar son abundantes o siguen una estructura más compleja se cuenta con la opción de construir una base de datos *SQL* a través de la librería *SQLite*.

Tanto la parte cliente cómo la de servidor, cuentan para el almacenamiento de la información para su funcionamiento de Bases de Datos.

7.4.1 Cliente

Dispone de una Base de datos SQLite, con la estructura siguiente:

- Tabla libros, almacena los datos de nuestros favoritos.

⊟ libros	table	
_id	field	integer PRIMARY KEY AUTOINCREMENT
titulo	field	text NOT NULL
autor	field	text NOT NULL
tema	field	text NOT NULL
descrip	field	text NOT NULL
mp3url	field	text NOT NULL
posicion	field	integer
imagen	field	BLOB

Estructura de tabla para la tabla "libros"

```
CREATE TABLE libros (_id integer PRIMARY KEY AUTOINCREMENT,
`titulo` text NOT NULL,
 `autor` text NOT NULL,
`tema` text NOT NULL,
 `descrip` text NOT NULL,
`mp3url` text NOT NULL,
`posicion` integer,
 `imagen` BLOB)
```

7.4.2 Servidor / Web

Dispone de una Base de datos MySql con dos tablas y la siguiente estructura:

- Tabla audiolibros, que contiene toda la información relativa a los audiolibros disponibles.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
1	**id**	int(11)			No	*Ninguna*	AUTO_INCREMENT
2	**titulo**	text	utf8_spanish_ci		No	*Ninguna*	
3	**autor**	varchar(50)	utf8_spanish_ci		No	*Ninguna*	
4	**tema**	text	utf8_spanish_ci		No	*Ninguna*	
5	**descripcion**	text	utf8_spanish_ci		No	*Ninguna*	
6	**imagen1**	mediumblob			Sí	*NULL*	
7	**imagen2**	mediumblob			Sí	*NULL*	
8	**mp3**	tinytext	utf8_spanish_ci		No	*Ninguna*	

Estructura de tabla para la tabla "audiolibros"

```
CREATE TABLE IF NOT EXISTS `audiolibros` (
`id` int(11) NOT NULL AUTO_INCREMENT,
`titulo` text COLLATE utf8_spanish_ci NOT NULL,
`autor` varchar(50) COLLATE utf8_spanish_ci NOT NULL,
`descripcion` text COLLATE utf8_spanish_ci NOT NULL,
`descripcion2` text COLLATE utf8_spanish_ci NOT NULL,
`imagen1` mediumblob,
`imagen2` mediumblob,
`mp3` tinytext COLLATE utf8_spanish_ci NOT NULL,
PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8
COLLATE=utf8_spanish_ci AUTO_INCREMENT=1 ;
```

- Tabla usuario, almacena los datos de los usuarios registrados.

#	Nombre	Tipo	Cotejamiento	Atributos	Nulo	Predeterminado	Extra
1	**idUsuario**	int(11)			No	*Ninguna*	AUTO_INCREMENT
2	**login**	varchar(255)	utf8_general_ci		No	*Ninguna*	
3	**pass**	varchar(255)	utf8_general_ci		No	*Ninguna*	
4	**email**	tinytext	utf8_general_ci		No	*Ninguna*	
5	**cookie**	varchar(255)	utf8_general_ci		Sí	*NULL*	
6	**validez**	datetime			Sí	*NULL*	

Estructura de tabla para la tabla "usuario"

```
CREATE TABLE IF NOT EXISTS `usuario` (
`idUsuario` int(11) NOT NULL AUTO_INCREMENT,
 `login` varchar(255) NOT NULL,
 `pass` varchar(255) NOT NULL,
`email` tinytext NOT NULL,
 `cookie` varchar(255) DEFAULT NULL,
 `validez` datetime DEFAULT NULL,
 PRIMARY KEY (`idUsuario`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=5 ;
```

7.5 Otras estructuras de datos que se utilizan en la aplicación.

Ficheros: permite la escritura y lectura de ficheros, tanto en el directorio en el que se encuentra instalada la aplicación, como en la tarjeta de memoria. Utilizaremos dos directorios locales a la aplicación que nos servirá para almacenar los datos descargados:

/files/Cat_Audio (para el catálogo)
/files/Images_Audio (para las imágenes)

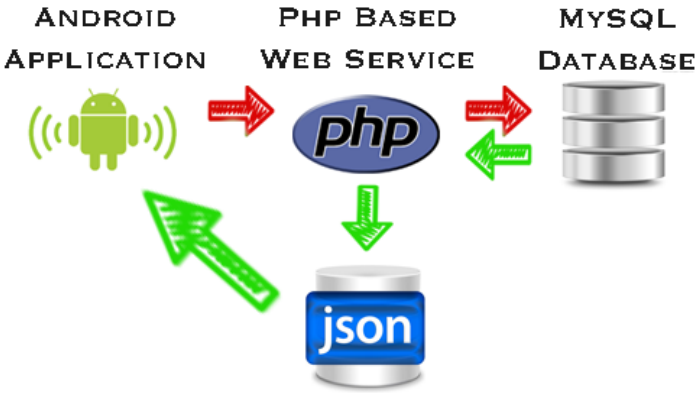
SharedPreferences: permite crear archivos en los que almacenan datos en la forma atributo/valor. Almacena información de configuración y de uso general.

Datos que se pueden ver y configurar por el usuario en la opción correspondiente del menú principal:

- Datos de Conexión, url del server que presta el servicio.
- Datos de Usuario, nombre, contraseña y correo electrónico de la cuenta de usuario.
- Datos de Entorno, valores que afectan al funcionamiento de la aplicación.

Datos transparentes al usuario y que utiliza la aplicación para diversos usos:

- CATALOGO (boolean), indica si existe un catálogo descargado en el dispositivo.
- TAMANO (integer), tamaño del catálogo que se utiliza para compararlo con el disponible en el servidor si hay variación es que hay actualizaciones.
- REGISTRO (boolean), nos indica que existe una cuenta de usuario registrado.



Para mejoras en el tiempo de carga de los datos, el intercambio de datos se hace mediante notación *JSON*, ya que reduce mucho la cantidad de información no útil a enviar con respecto a *XML*. Por muy avanzados que estén actualmente los dispositivos móviles, sigue siendo crítica su capacidad de procesamiento con respecto a los tiempos de respuesta a los que un usuario está acostumbrado con su *PC*. Para llevar a cabo esto, es necesario tener servicios web que devuelvan los datos en esta notación. El código es altamente independiente de la forma de procesar y mostrar los datos, con lo que sólo sería necesario hacer clases lectoras de *JSON*.

JSON *(JavaScript Object Notation)* es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML. La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML. También es frecuente el uso de JSON y XML en la misma aplicación.

La especificación de este formato se puede consultar en la web www.json.org

La mejor manera de describirlo es quizá mediante un ejemplo:

```
{
   "Nombre" : "Juan",
   "Edad": 28,
   "Aficiones": ["Música", "Cine", "Tenis"],
   "Residencia": "Madrid"
}
```

Como vemos, en JSON se pueden representar dos tipos de estructuras:

- Un conjunto de pares (clave, valor) encerrado entre los caracteres "{" y "}", separando la clave del valor por el símbolo ":", y separando cada par del siguiente con el carácter " ,"

- Un conjunto ordenado de valores encerrado entre los caracteres “[“ y “]“, y separando cada valor del siguiente con el carácter “,”.

Por otra parte, el valor de un par (clave, valor) puede ser un elemento simple (cadena de caracteres o número) o bien una estructura de datos. Esto permite representar estructuras de una complejidad arbitraria.

Un punto importante de la especificación es que las cadenas de caracteres deben estar codificadas en Unicode (por ejemplo, UTF-8).

7.6 Estudio de la seguridad de la aplicación.

Nuestra aplicación los únicos datos "sensibles" que maneja son el nombre de usuario y la contraseña necesarios para utilizar los servicios web. Datos que se envían cifrados y codificados.

La codificación empleada es Base64, es un sistema de numeración posicional que usa 64 como base. Es la mayor potencia de dos que puede ser representada usando únicamente los caracteres imprimibles de ASCII. Esto ha propiciado su uso para codificación de correos electrónicos, PGP y otras aplicaciones.

Para cifrar la contraseña se utilizan dos algoritmos de encriptación:

- MD5, (abreviatura de *Message-Digest Algorithm 5*, Algoritmo de Resumen del Mensaje 5) es un algoritmo de reducción criptográfico de 128 bits ampliamente usado.

- SHA1, (*Secure Hash Algorithm*, Algoritmo de *Hash* Seguro) es un sistema de funciones *hash* criptográficas relacionadas de la Agencia de Seguridad Nacional de los Estados Unidos y publicadas por el *National Institute of Standards and Technology* (NIST).

Además se le añade una "máscara" de caracteres extraños, lo que hace más complicada la cadena a codificar.

Respecto a las opciones sobre la URL del servidor, no está implementada la conexión segura ya que su finalidad es conectar mediante el protocolo HTTPS, pero actualmente el servidor en el que se realiza la conexión no soporta este tipo de peticiones para el envío de datos cifrados, con lo cual esto es una actualización para el futuro.

7.7 Publicación.

Tras el desarrollo de una aplicación, el siguiente paso es su publicación, ya sea en Android o en cualquier otro entorno. Si desarrollamos una aplicación para Android, tenemos dos opciones: publicarla dentro de Google Play o fuera.

Google Play (antes llamado Android Market) es una tienda de software en línea desarrollada por Google para los dispositivos Android, pero su uso no es obligatorio. Ahora pasaremos a explicar las dos posibilidades.

7.7.1 Publicar fuera de Google Play

Se trata de la forma convencional de dar a conocer tu aplicación, y además es fácil y barato. De esta forma depende del propio desarrollador la distribución de la aplicación, incluyendo tanto ponerla a disposición del usuario, publicitarla y adoptar un canal de comunicación con el usuario final, de forma que exista un intercambio de información sobre fallos, consejos, ayuda y actualizaciones, entre otras cosas.

Consiste principalmente en compilar la aplicación y subir el archivo .apk resultante a alguna web para que pueda ser descargada.

La promoción puede llevarse a cabo en redes sociales, foros, o mediante una página específica donde, además, se puede proporcionar información acerca de novedades sobre la aplicación.

Esta compilación debe realizarse en Modo Release. Para conseguirlo, basta seleccionar \Export Unsigned Aplication Package" dentro del menú \Android Tools", pulsando con el botón derecho sobre la carpeta principal de nuestro proyecto en Eclipse. El archivo generado es que el debe ponerse a disposición del usuario para su descarga. Sin embargo, el usuario debe configurar su terminal para aceptar aplicaciones que no hayan sido descargadas de Google Play, por lo que puede suponer un inconveniente.

7.7.2 Publicar en Google Play

Google Play es la tienda oficial de aplicaciones Android. Desde Internet podemos acceder a través de su página web y desde todos los terminales que tengan el sistema operativo Android podemos acceder directamente mediante el icono correspondiente. Allí podemos encontrar todas las aplicaciones que los que se han registrado como desarrolladores de Android han implementado (aplicaciones de todo tipo, juegos...). Ellos mismos eligen si sus aplicaciones pueden ser descargadas de forma gratuita, o si por el contrario hay que pagar alguna cantidad de dinero.

Este método, a pesar de tener un coste de 25$, permitirá vender nuestra aplicación, insertarle publicidad, y tener acceso a una red de distribución mucho más extensa de lo que

seguramente podríamos lograr mediante la autopromoción, dado que es una herramienta existente en todos los terminales Android.

El primer paso será crear una cuenta de desarrollador para Google Play accediendo a su web con una cuenta Google e introducir la información requerida, incluyendo los datos de pago. Una vez confirmado podemos empezar a publicar aplicaciones.

Hay una pequeña diferencia con el caso anterior: en el Market las aplicaciones deben ser firmadas, por lo que al crear el .apk habrá que elegir \Export Signed Aplication Package".

Una vez completados estos pasos, la aplicación ya puede ser publicada en Google Play.

Para posteriores actualizaciones, es importante hacer dos cambios en el Manifest de la aplicación, de lo contrario Google Play no aceptará la subida.

Los parámetros son los siguientes:

android:versionCode="1". Hay que incrementar una unidad cada vez que se realice una actualización.

android:versionName=”1.0". Realmente no es obligatorio, pero se recomienda cambiar la numeración, de forma que sea orientativo para el usuario.

8. Código fuente documentado.

Detallado según los anexos siguientes:

- *Anexo A*: Código fuente Sitio Web y Web Service.
- *Anexo B*: Código fuente aplicación android.

9. Manual de configuración y funcionamiento de la aplicación.

9.1 Cliente

Una vez finalizada la implementación de la aplicación y habiendo probado su correcto funcionamiento, es necesario redactar un manual de uso, de forma que, aún teniendo un uso intuitivo, cualquier usuario sea capaz de familiarizarse fácilmente con la interfaz y su correcto funcionamiento.

AudioBooks es una sencilla aplicación destinada a la reproducción on-line y gestión de favoritos de audiolibros alojados en un servidor remoto.

Es necesaria una conexión a Internet ya sea mediante una red Wi-Fi o a través de la conexión de datos ofrecida por el operador de telefonía móvil. Si no dispone de conexión se mostrará una advertencia cuando sea necesaria la conexión para realizar alguna acción.

Las aplicaciones móviles para Android tienen la extensión .apk que es una variante del formato JAR de Java y se usa para distribuir e instalar componentes empaquetados para la plataforma Android para móviles.

Hay dos maneras de instalar aplicaciones apk en Android

- Buscar e instalarlas desde el Android Market de nuestro teléfono. Lo más simple y rápido.
- Lo más complicado, buscar en Internet aplicaciones para Android y descargar el paquete apk. Para poder instalarlas en nuestro teléfono debemos activar lo siguiente:

Ajustes < Aplicaciones < Orígenes desconocidos

Y simplemente debemos copiar las aplicaciones a la tarjeta de memoria y usar un programa explorador de archivos, tal como *Astro* Administrador de archivos, para poder instalarla: simplemente buscamos la aplicación y pulsamos para instalarla.

Todas estas aplicaciones, al no estar instaladas desde el Market, se tienen que desinstalar desde:

Ajustes > Aplicaciones > Administrar aplicaciones.

INSTALAR APLICACION DESDE TARJETA SD

1. Descargar el archivo .APK a tu PC o directamente a la tarjeta SD desde tu celular.
2. En caso de descargarlo a la PC, conectar el teléfono a través de USB.
3. Asegúrate de que la configuración de conexión del teléfono sea correcta. Para comprobar esto, ve a AJUSTES -> Conectar a PC -> Conexión estándar -> Unidad de almacenamiento masivo.
4. Arrastrar el archivo .APK que descargaste en tu PC a la tarjeta SD del teléfono. Se recomienda crear una nueva carpeta en la tarjeta SD llamada, por ejemplo, “Aplicaciones”.
5. Desconectar de forma segura el teléfono.
6. Abrir un explorador de archivos como AstroFile, FileBro o File Manager (puedes encontrar cualquiera de estos exploradores de archivos en el Android Market).
7. Buscar el archivo APK que has puesto en tu tarjeta SD, haz clic en él y luego presiona Instalar.
8. Leer los permisos e instalar la aplicación.

9.2 Servidor

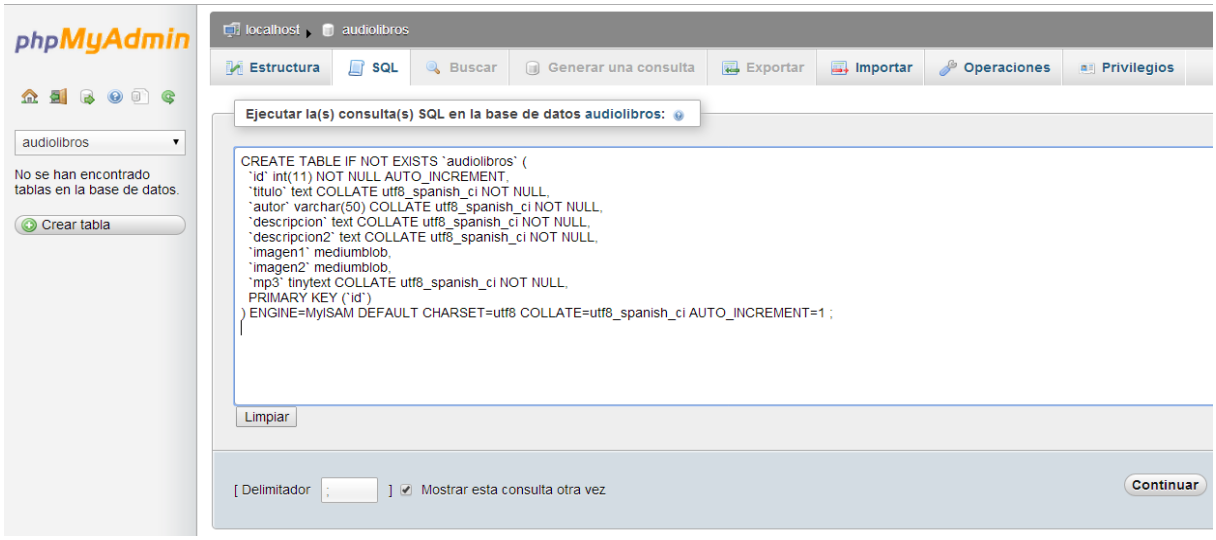
Partiendo de la premisa de que disponemos de una cuenta en un hosting y del dominio correspondiente.

Dado que el hosting contratado dispone de toda la infraestructura necesaria, sólo deberemos configurar la Base de Datos a utilizar y subir todos los archivos que componen el Sitio Web y el Web Service.

Para crear la BD "audiolibros" utilizamos el administrador phpMyAdmin disponible desde el hosting.

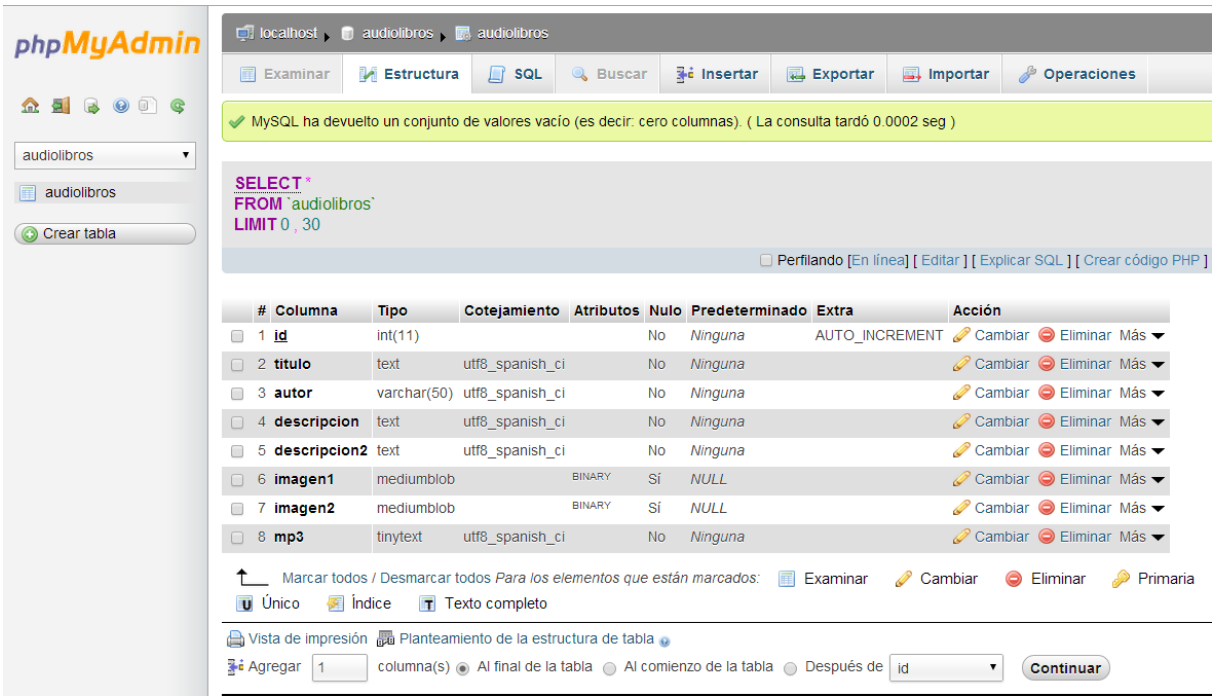
Posteriormente creamos las tablas "audiolibros" y "usuario" que almacenan los datos de libros y usuarios respectivamente.

Las crearemos ejecutando el código siguiente desde la opción SQL del administrador y pulsamos continuar,



```
CREATE TABLE IF NOT EXISTS `audiolibros` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `titulo` text COLLATE utf8_spanish_ci NOT NULL,
  `autor` varchar(50) COLLATE utf8_spanish_ci NOT NULL,
  `descripcion` text COLLATE utf8_spanish_ci NOT NULL,
  `descripcion2` text COLLATE utf8_spanish_ci NOT NULL,
  `imagen1` mediumblob,
  `imagen2` mediumblob,
  `mp3` tinytext COLLATE utf8_spanish_ci NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_spanish_ci
AUTO_INCREMENT=1 ;
```

Si todo es correcto podemos ver la tabla creada,



De igual forma procederemos con la otra tabla,

```
CREATE TABLE IF NOT EXISTS `usuario` (
  `idUsuario` int(11) NOT NULL AUTO_INCREMENT,
  `login` varchar(255) NOT NULL,
  `pass` varchar(255) NOT NULL,
  `email` tinytext NOT NULL,
  `cookie` varchar(255) DEFAULT NULL,
  `validez` datetime DEFAULT NULL,
  PRIMARY KEY (`idUsuario`)
) ENGINE=MyISAM  DEFAULT CHARSET=utf8 AUTO_INCREMENT=1 ;
```

Ya tendríamos creada la Base de datos con sus dos tablas, sólo queda introducir la información de libros y en usuario según se registren estos,

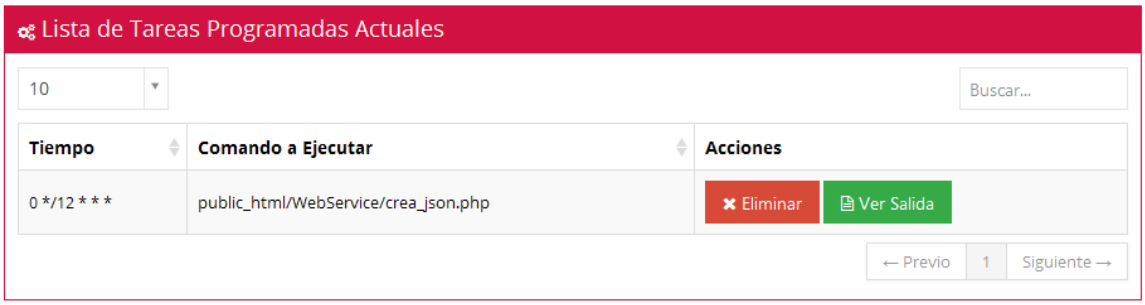
Aunque la carga de datos se haría por parte de la administración del Servicio, inicialmente los datos almacenados en la BD de audiolibros se han obtenido de varias formas:

- Carga manual, desde el administrador de BD phpMyAdmin y obtenidos desde páginas cómo www.ivoox.com y otras.

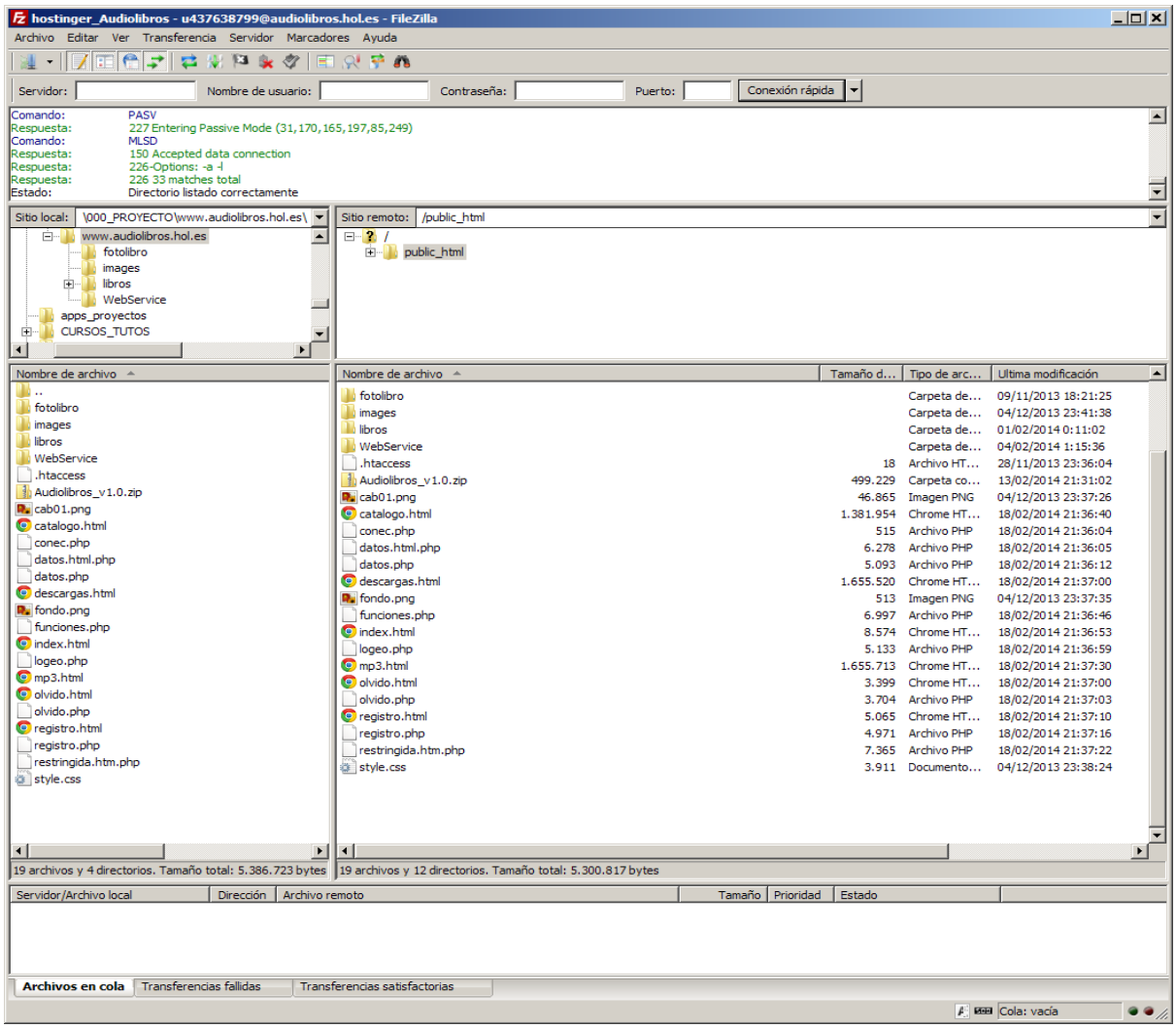
- Carga automática, utilizando el archivo en formato Excel, convertido a csv, con los datos de los audiolibros disponibles en www.leemp3.com facilitado desde la misma página. (Hay que editar enlaces erróneos).

Para implementar el proyecto se hace uso de los audiolibros disponibles en la página www.leemp3.com, biblioteca hispana de libros audibles. 3200 Libros de audio mp3 gratis de dominio público. Sólo se utilizan estos enlaces a modo de ejemplo, se trataría de crear nuestros propios recursos o compartir otros previa autorización.

También configuramos desde el administrador del hosting una tarea programada que nos servirá para ir actualizando el catálogo de libros de forma automática. Esta tarea ejecuta un script php que genera el fichero .json que almacena el catálogo y que será descargado por la aplicación android.



Subiremos al hosting, utilizando un cliente FTP o el administrador de archivos del hosting, el resto de archivos y directorios necesarios para el funcionamiento del Sitio Web y del Web Service.



Y con esto estaría configurado el lado del servidor.

10. Manual de usuario

10.1 Inicio de la aplicación

Una vez la aplicación ha sido instalada el usuario podrá acceder a ella desde el menú de aplicaciones de su dispositivo. La primera ventana que verá siempre es la ventana de inicio de la aplicación, la cual muestra el nombre y página web de la empresa. Esta ventana, además de ser informativa, sirve como tiempo de espera puesto que, durante su visualización, se leen los datos de configuración y se comprueba la disponibilidad de conexión a Internet.

Figura 1: Pantalla de inicio.

10.2 Menú Principal

Inicialmente, dependiendo de la configuración y si no se ha realizado el registro del servicio, podrá aparecer un mensaje de aviso indicando:

- ✓ No se dispone de conexión a Internet.
- ✓ No hay nombre de usuario y contraseña almacenados.

Esta pantalla despliega une menú que representado por cuatro botones que al pulsarlos nos permiten realizar acciones en relación a la opción seleccionada.

También podemos acceder al menú principal desde cualquier pantalla de la aplicación, pulsando sobre el botón de menú del dispositivo y seleccionando la entrada correspondiente.

La siguiente ventana, mostrará las cuatro opciones principales de la aplicación:

- Registro.
- Favoritos.
- Catálogo.
- Configurar.

Además existen:

- Acerca de…
- Salir.

Figura 2: Menú Principal.

10.3 Registro.

Esta ventana muestra un formulario para introducir los datos de registro del servicio, nombre de usuario, contraseña y correo electrónico, cuando la información introducida es errónea se indicará mediante mensajes y además vibrará el dispositivo.

Los mensajes nos indicarán si el usuario ya existe, si la contraseña es válida y si el correo es correcto.

En caso de validar correctamente los datos introducidos, se enviará un mensaje a la dirección de correo especificada con los datos de registro. También se registran los datos en la configuración de la aplicación.

Dispone de dos botones:

- Registrar.
  Envía los datos introducidos al servidor.
- Volver.
  Cierra la ventana de registro sin enviar datos al servidor.

Figura 3: Ventana de registro.

10.4 Favoritos.

Nos permite ver los audiolibros añadidos a nuestros favoritos, sobre los mismos podemos realizar las opciones siguientes:

- Detalles.
  Nos muestra información ampliada sobre el audiolibro seleccionado. Además nos permite reproducirlo.
- Eliminar.
  Borra de Favoritos el audiolibro seleccionado y la información relacionada.
- Reproducir.
  Reproduce el audiolibro seleccionado mostrando un reproductor. Es necesario que exista conexión a internet. Muestra los botones característicos a cualquier reproductor (Reproducir, Pausar, Avanzar, Retroceder, Vol + - y Salir).
- Salir.
  Cierra el menú contextual.

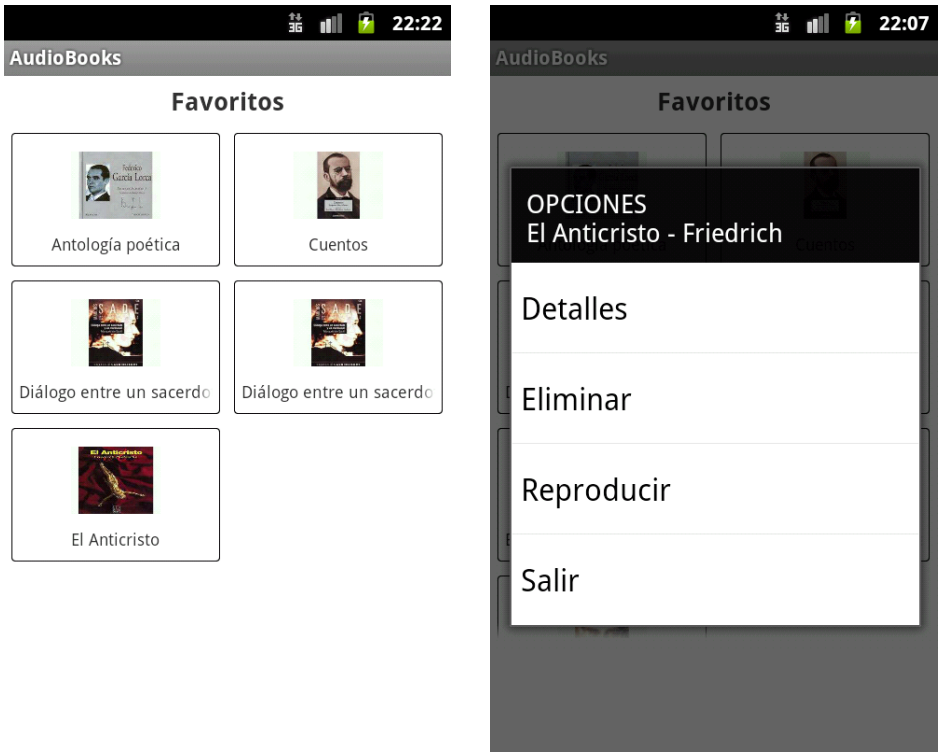


Figura 4: Favoritos y menú de opciones.

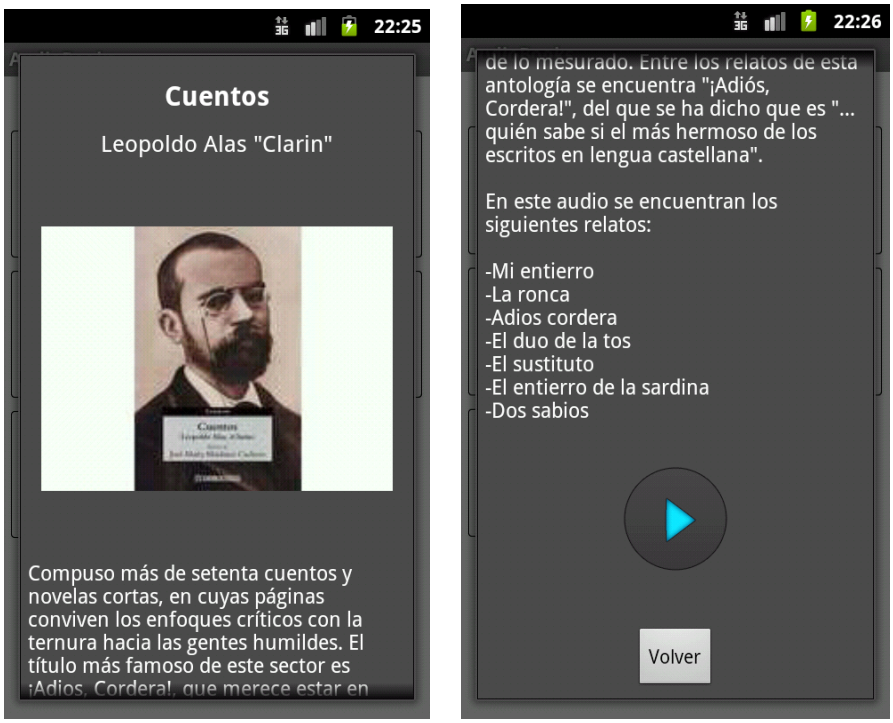


Figura 4.1: Detalles.

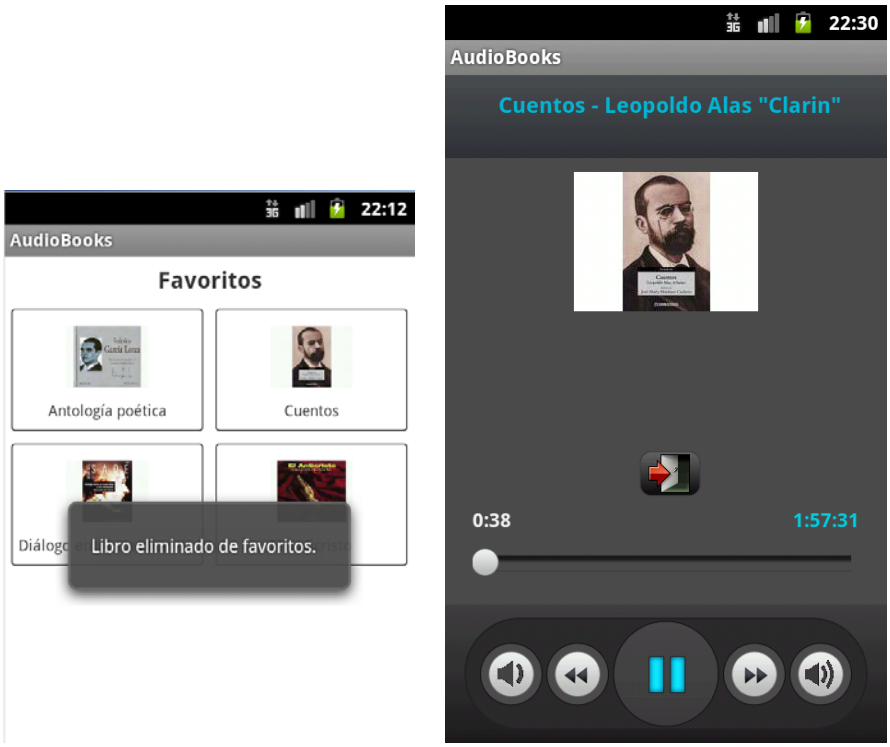


Figura 4.2: Mensaje de libro eliminado y Reproductor.

Al pulsar el botón "Salir" del reproductor, se nos solicitará si queremos guardar el estado de reproducción, lo que permite que la próxima vez se reproduzca desde el último punto guardado.

Figura 4.3: Diálogo para guardar el punto de reproducción.

10.5 Catálogo.

Esta opción permite descargar el catálogo de audiolibros desde el Servidor, lo almacena en el dispositivo y posteriormente lo visualiza.

Para veces posteriores comprobará si existen actualizaciones, pudiendo el usuario decidir si la descarga.

Sobre los audiolibros disponibles mostrados por el catálogo, podemos realizar las opciones siguientes:

- Agregar a Favoritos.
  Incorpora el audiolibro seleccionado y su información correspondiente a nuestros favoritos. Es necesario que exista conexión a internet.
- Reproducir.
  Reproduce el audiolibro seleccionado mostrando un reproductor. Es necesario que exista conexión a internet.
- Salir.
  Cierra el menú contextual.

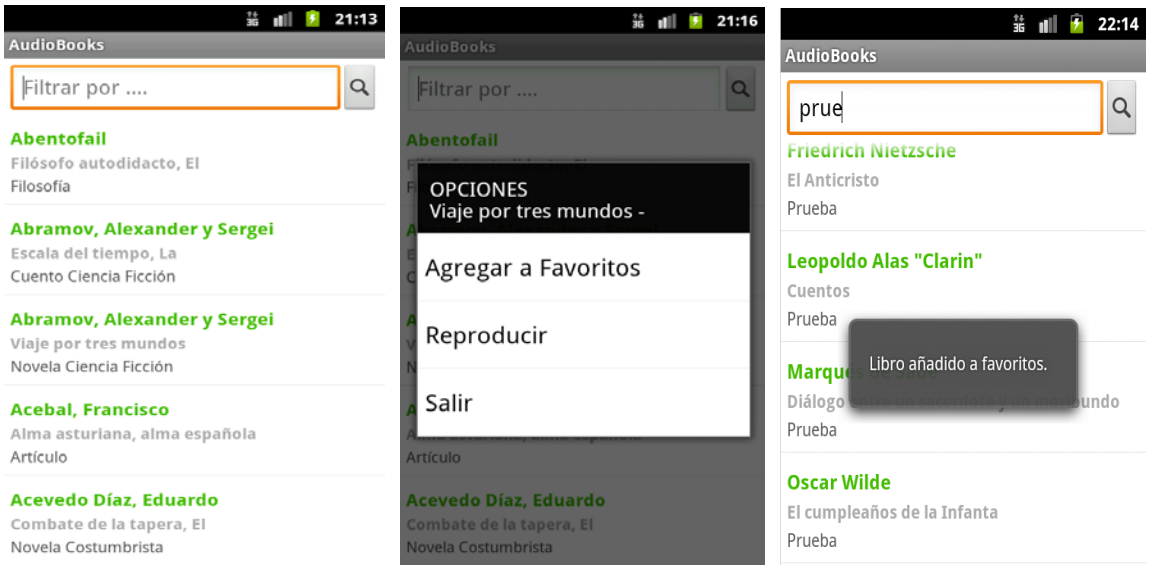


Figura 5.1: Catálogo, Menú de Opciones y mensaje de añadido a favoritos.

Está disponible una ventana de búsqueda que nos permite introducir el texto sobre el que aplicar el filtro sobre el catálogo, al pulsar el botón de "lupa" se abre el menú de opciones al objeto de aplicar el filtro sobre el Autor, Título o Tema, además de poder eliminar el filtro activo.

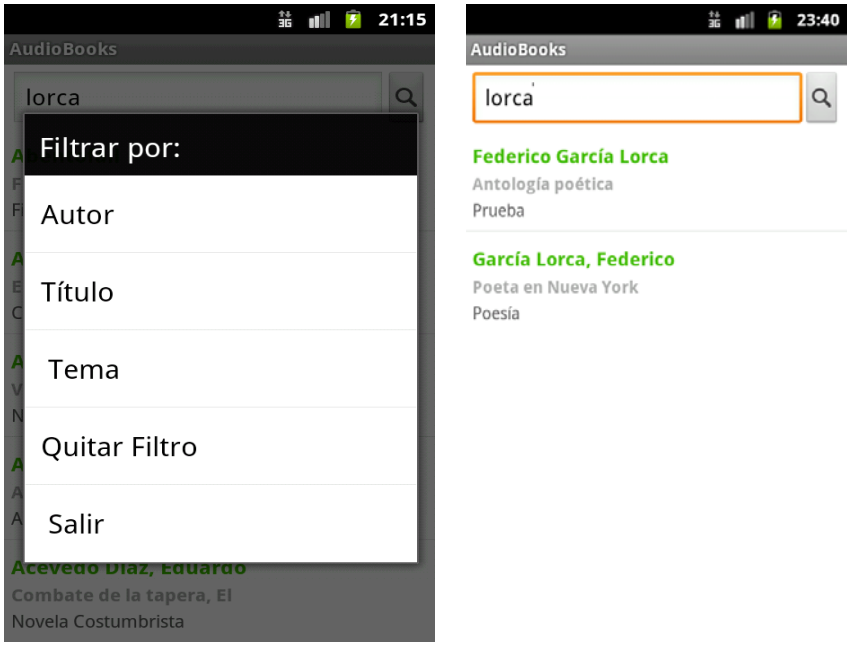


Figura 5.2: Opciones de Filtros disponibles y ejemplo de filtro activo por autor.

10.6 Configurar.

Permite configurar los diferentes parámetros necesarios para el buen funcionamiento de la aplicación. Se encuentran agrupados en tres bloques:

- Datos de Conexión.
  Muestra por defecto el nombre del Servidor que presta los servicios a la aplicación, en nuestro caso www.audiolibros.hol.es.
  En caso de modificarlo, se nos avisa con un mensaje de la posibilidad de mal funcionamiento si la dirección no es correcta.
- Datos de Usuario.
  Almacena los datos de registro introducidos y validados en la opción “Registro”, necesarios para validar diversas acciones ante el servidor.
  Se pueden modificar, pero como en el caso anterior si no son correctos la aplicación no funcionará correctamente.
  Auto login. Activar login automático.
  Registro Manual. Habilita la opción de registro manual para datos no introducidos mediante la opción “Registro”.
- Entorno.
  Activar SD-Card. Permite cambiar donde se almacenan los datos.
  Reiniciar Entorno. Reinicia los datos de información interna para casos de mal funcionamiento.

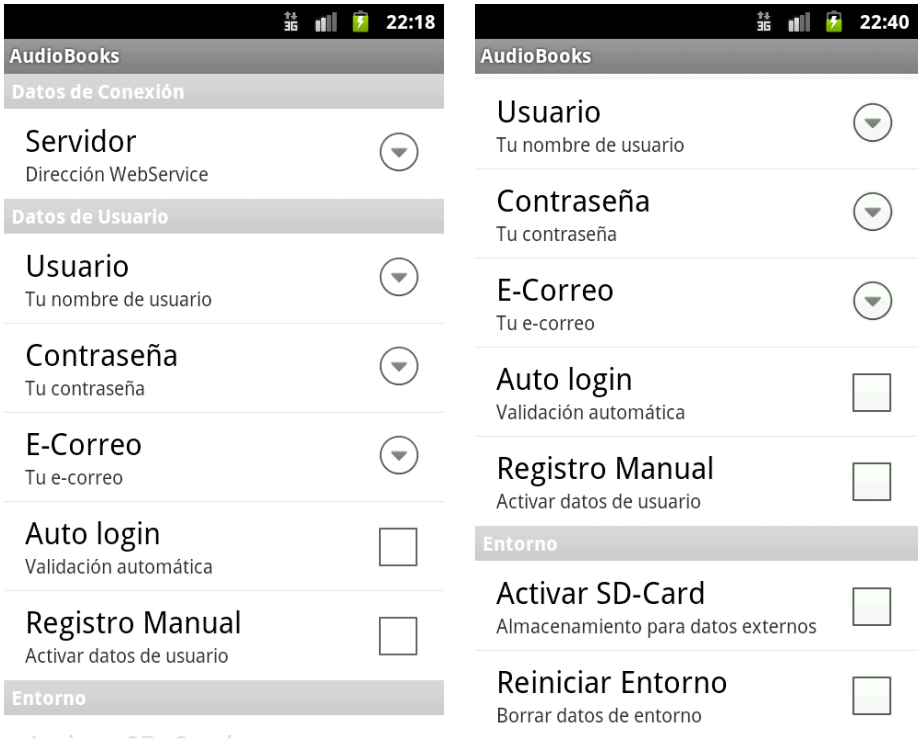


Figura 6: Pantalla de configuración de la aplicación.

Cómo se aprecia en la imagen, la apariencia de esta ventana es similar a cualquiera de las pantallas de configuración de un dispositivo *Android*. En este caso las opciones se encuentran divididas en: Datos de Conexión, de Usuario y Entorno.

Al pulsar en “Servidor de conexión” se muestra una ventana de diálogo con la URL actual (predefinida). Debido a que, si la nueva URL es incorrecta, no podrá accederse a ninguno de los datos, cuando el usuario pulsa en “Aceptar” se muestra un mensaje de aviso.

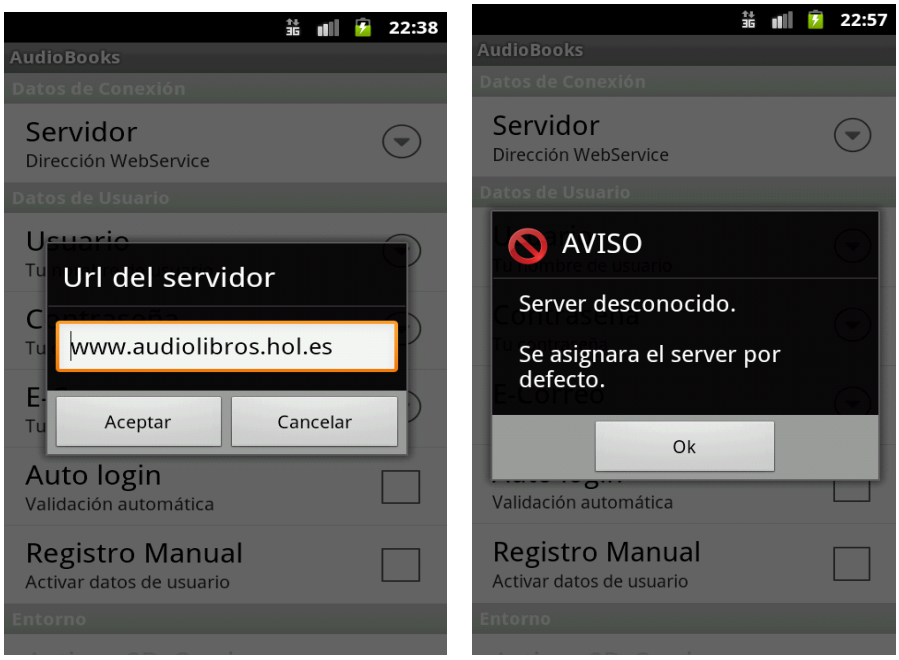
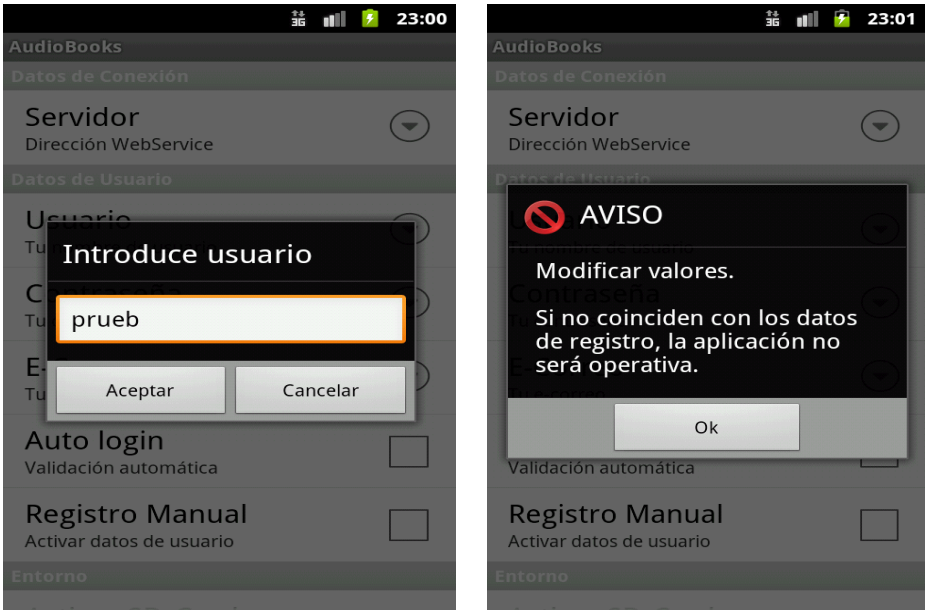


Figura 6.1: Datos de Conexión.

Para los Datos de Usuario, el funcionamiento es similar.



10.7. Acerca de…

Muestra una ventana con los créditos, nombre, versión y derechos de autor de la aplicación.

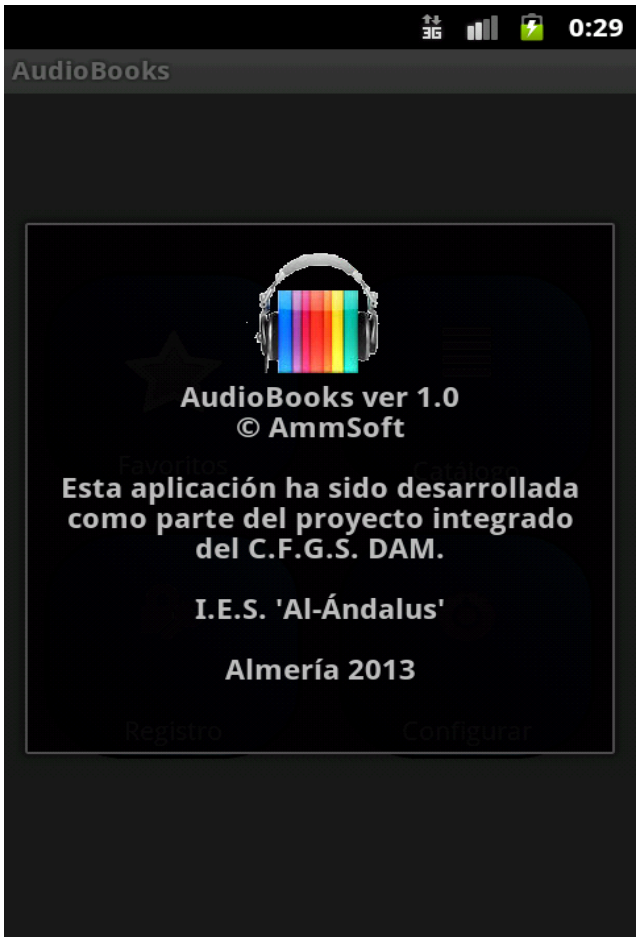


Figura 7: Acerca de...

10.8. Salir.

Cierra la aplicación.

Figura 8: Salir de la aplicación.

10.9. Mensajes de la aplicación

A continuación se muestran una serie de mensajes que pueden aparecer en cualquiera momento durante la ejecución de la aplicación.

El primero y más importante es el mensaje de error de conexión. Este aparecerá siempre que se haya perdido la conexión a Internet y se intente realizar una acción que la necesite.

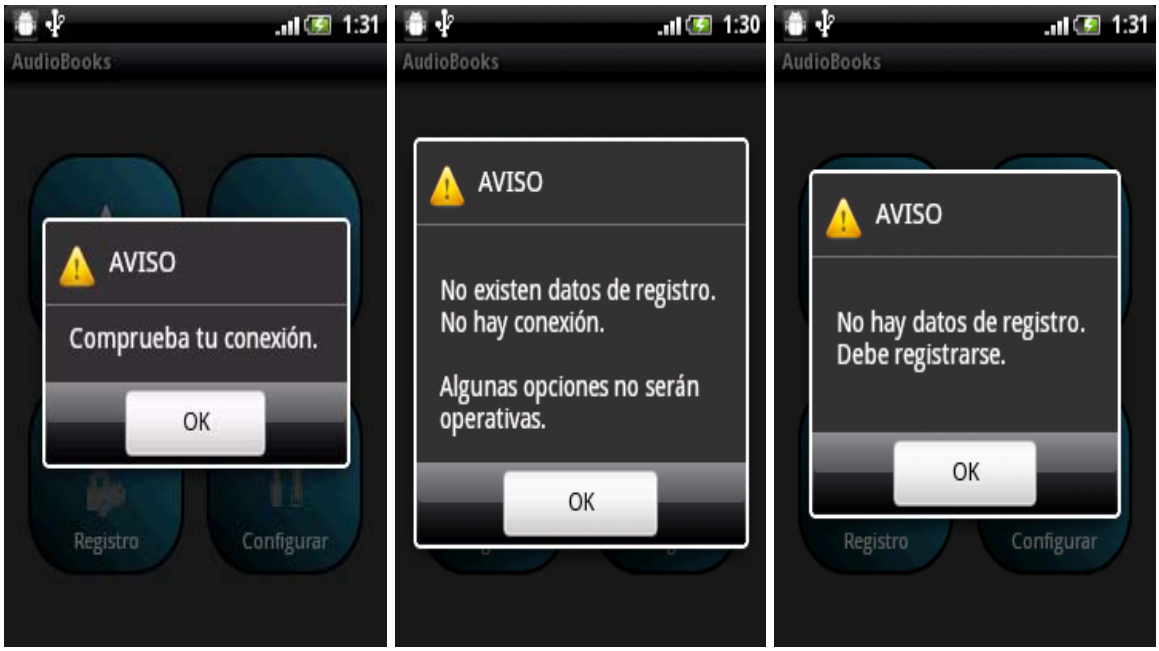
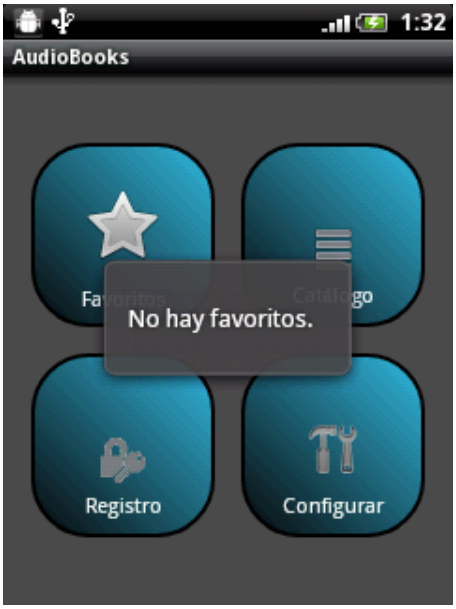


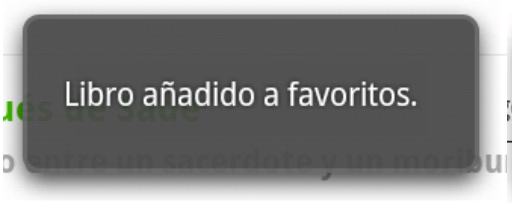
Figura: Mensaje de Error de Conexión y Datos de Registro.

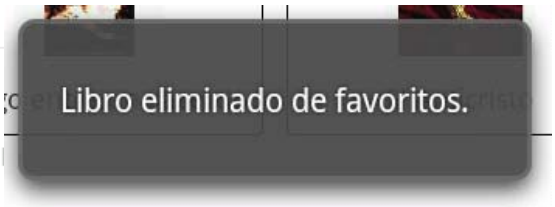
Otro mensaje que puede aparecer en cualquier momento es la pantalla de espera. Ésta aparecerá en las transiciones entre pantallas cuando sea necesario leer datos de Internet y la operación pueda tardar unos segundos. Aparecerá una ventana similar pero con mensaje de texto cuando, dentro de una ventana, se vaya a realizar una operación que pueda tardar un tiempo considerable. Ambas bloquean cualquier otra acción que se pudiera realizar para no ralentizar más el proceso.

Figura: Ventana de espera dentro de una pantalla.

Otras ventanas de diálogo y mensajes muestran un mensaje de aviso en espera de una confirmación o una cancelación, ante las cuales se realizará la acción asociada o no se hará nada.







Además podrán aparecer otros mensajes informativos informando de errores ocurridos en el acceso al servidor o durante el intercambio de datos.

No se realiza un manual de usuario del sitio Web debido a la sencillez de manejo y uso intuitivo del mismo.

11. Plan de Formación a los Usuarios de la Aplicación.

Debido a la simplicidad de uso de la aplicación y su manejo intuitivo no es necesario realizar ningún plan de formación específico.

A nivel técnico, la instalación del servidor y servicios necesita ser administrados por personal con formación específica. Apartado éste transparente al usuario final.

12. Conclusiones.

El último apartado a tratar es el relacionado con las conclusiones del proyecto que se ha desarrollado. Cabe mencionar que el proyecto, a lo largo de todo su desarrollo, ha logrado cumplir con los objetivos y motivaciones que se habían marcado al iniciar el proyecto.

He podido realizar el desarrollo de una aplicación destinada a dispositivos móviles desde cero hasta lograr una aplicación que funciona de la manera esperada. De esta forma, he adquirido conocimientos relacionados con el desarrollo de aplicaciones para dispositivos móviles, en este caso la programación destinada a Android así como de la arquitectura cliente-servidor.

El proceso de programación ha sido interesante al haber sido posible aprovechar algunos de los conocimientos obtenidos durante el Ciclo formativo. Aun así fue necesario aprender muchas cosas nuevas específicas del lenguaje de programación Java y de la programación de dispositivos móviles, conocimientos que no se adquieren durante el Ciclo.

Por otro lado, el hecho de tener que desarrollar la aplicación de forma individual, sin un código inicial o referencia ha hecho que consultar diferentes fuentes tales como libros o páginas web haya cobrado gran relevancia a la hora de abordar y solventar diversos problemas o cuestiones surgidos a lo largo del desarrollo.

Para funciones básicas de programación existe abundante información documentada y códigos ejemplo en Internet, mientras que otros aspectos más difíciles, como algunos usados en esta aplicación, no están lo suficientemente documentado. Muchos apartados de la página web oficial para desarrolladores de Android podrían estar mejor documentados.

Por otro lado, la creación de las interfaces es dificultosa debido a que la herramienta empleada no está debidamente desarrollada por lo que es costoso ajustar los distintos componentes de la interfaz. Cabe también mencionar la poca optimización a la hora de emplear un dispositivo virtual, lo que lleva a que las pruebas sean un proceso lento debido a la poca fluidez que presenta esta característica del SDK.

Como conclusión mencionar que el proyecto me ha servido para aprender el uso de tecnologías antes desconocidas como es la programación Cliente-Servidor para dispositivos Android, así como para poner en práctica conocimientos adquiridos durante la realización del Ciclo Formativo.

Bibliografía y fuentes de información

Libros:

JESÚS TOMÁS GRIONÉS (2011), *El gran libro de Android*. Barcelona: Marcombo.

ED BURNETTE (2011), *Programación Android.* Madrid: Anaya Multimedia.

FRANK ABLESON, CHARLIE COLLINS y ROBISEN (2010), *Android, Guía para desarrolladores*. Madrid: Anaya Multimedia.

JAVIER CABALLOS (2001), *El lenguaje de programación Java*. Madrid: Ra-Ma.

MARKO GARGENTA (2011), *Learning Android.* O'Reilly.

RETO MEIER (2010), *Professional Android 2 Application Development.* Wrox.

Páginas webs:

Web oficial para desarrolladores Android:
http://developer.android.com

Blog con una serie de artículos sobre programación Android:
http://www.sgoliver.net/blog/?page_id=2935

Android-Spa, foro de desarrolladores Android:
http://www.android-spa.com

Googlegroups Desarrolladores Android:
http://groups.google.com/group/desarrolladores-android

Wikipedia, la enciclopedia libre:
http://www.wikipedia.org/

Web oficial de API Maps for Android:
http://code.google.com/intl/es-ES/android/add-ons/google-apis/reference/index.html

StackOverFlow, foro de consulta para desarrolladores en general:
http://stackoverflow.com/

Eclipse:
http://www.eclipse.org/

Google Play:
https://play.google.com/store

Web para crear una cuenta de desarrollador en Google Play:
https://play.google.com/apps/publish/signup

Web para descargar el ADT para Eclipse:
https://dl-ssl.google.com/android/eclipse

Introducción a *JSON* (JavaScript Object Notation)
http://www.**json**.org/**json**-es.html

Ley de Propiedad Intelectual
https://www.boe.es/buscar/act.php?id=BOE-A-1996-8930
http://www.eoi.es/wiki/index.php/Los_derechos_de_explotaci%C3%B3n._en_Propiedad_intelectual

Hosting
http://www.hostinger.es/

PHP: Hypertext Preprocessor
http://www.php.net/

Glosario de términos

Android. Sistema operativo móvil basado en Linux.

API. Siglas de Application Programming Interface, en español Interfaz de Programación de Aplicaciones. Consiste en un conjunto de llamadas que ofrecen acceso a funciones y procedimientos, representando una capa de abstracción para el desarrollador.

Aplicación. Programa informático diseñado como herramienta para realizar uno o varios trabajos determinados.

App. Denominación genérica para aplicaciones para dispositivos móviles.

Avatar. Imagen asociada a un usuario para su identificación.

Cookie. (**galleta informática**) es una pequeña información enviada por un sitio web y almacenada en el navegador del usuario, de manera que el sitio web puede consultar la actividad previa del usuario.

Cpanel. (Acrónimo de control Panel) es una herramienta de administración basado en tecnologías web para administrar sitios de manera fácil, con una interfaz limpia.

Cron. En el sistema operativo Unix, es un administrador regular de procesos en segundo plano (*demonio*) que ejecuta procesos o guiones a intervalos regulares (por ejemplo, cada minuto, día, semana o mes). Los procesos que deben ejecutarse y la hora en la que deben hacerlo se especifican en el fichero crontab. El nombre *cron* viene del griego *chronos* (χρόνος) que significa "tiempo".

Eclipse. Entorno de desarrollo integrado de código abierto.

Google Play. Tienda de aplicaciones Android de Google.

Hosting. El alojamiento web (en inglés *web hosting*) es el servicio que provee a los usuarios de Internet un sistema para poder almacenar información, imágenes, vídeo, o cualquier contenido accesible vía web.

HTML. Siglas de *HyperText Markup Language* («lenguaje de marcas de hipertexto»), hace referencia al lenguaje de marcado para la elaboración de páginas web.

HTTP. Siglas de *HyperText Transfer Protocol*, en español Protocolo de Transferencia de Hipertexto. Constituye el protocolo utilizado para la transmisión de documentos a través de la Web entre un cliente y servidor.

JavaScript. Es un lenguaje de programación interpretado, se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

JSON. *Javascript Object Notation*, es un formato ligero de intercambio de datos, que no requiere el uso de XML.

JVM. Siglas de *Java Virtual Machine*, en español Máquina Virtual de Java. Constituye un elemento software de la tecnología Java, encargado de transformar el código intermedio universal o *bytecode* en código máquina específico del hardware donde está instalado.

MP3. formato de compresión de audio digital cuyo nombre completo es MPEG-1Audio Layer 3. También utilizado, por extensión, para nombrar al dispositivo móvil reproductor de audio digital en el que se almacena, organiza o reproduce archivos de audio digital.

MySQL. Sistema de gestión de bases de datos relacional, multihilo y multiusuario.

PHP. Lenguaje de programación de uso general de *script* del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico.

PhpMyAdmin. Es una herramienta escrita en PHP con la intención de manejar la administración de MySql través de páginas web, utilizando Internet.

SDK. Siglas de Software Development Kit, en español Kit de Desarrollo de Software. Constituye un conjunto de herramientas que permiten a un desarrollador crear aplicaciones para una determinada plataforma o lenguaje.

Servicio Web. (Web Service) Tecnología que utiliza un conjunto de protocolos y estándares que sirven para compartir datos entre aplicaciones.

Smartphone. Dispositivo móvil que representa una evolución de los teléfonos móviles, con la inclusión de pantalla táctil, teclado, conexión Wi-Fi, aplicaciones de usuario como navegador web o cliente de correo, entre otros.

OAuth (*Open Authorization*). Es un protocolo abierto, propuesto por Blaine Cook y Chris Messina, que permite autorización segura de una API de modo estándar y simple para aplicaciones de escritorio, móviles y web.

SQLite. Sistema de gestión de bases de datos relacional compatible con ACID, contenida en una relativamente pequeña (~275 KB) biblioteca escrita en C.

UML. Lenguaje Unificado de Modelado (LUM o UML, por sus siglas en inglés, *Unified Modeling Language*) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad.

Vista. Objeto que define el interfaz de usuario de una aplicación Android.

Wi-Fi. Acrónimo de *Wíreless Fidelity*, estándar de envío de datos que utiliza ondas de radio en lugar de cables.

ANEXO A: Código fuente Sitio Web y Web Service

Estructura de carpetas y archivos de www.audiolibros.hol.es

fotolibro	Carpeta de archivos	
images	Carpeta de archivos	
libros	Carpeta de archivos	
WebService	Carpeta de archivos	
.htaccess	Archivo HTACCESS	1 KB
Audiolibros_v1.0.zip	Carpeta comprimi...	488 KB
cab01.png	Imagen PNG	46 KB
catalogo.html	Chrome HTML Do...	1.377 KB
conec.php	Archivo PHP	1 KB
datos.html.php	Archivo PHP	7 KB
datos.php	Archivo PHP	6 KB
descargas.html	Chrome HTML Do...	1.645 KB
fondo.png	Imagen PNG	1 KB
funciones.php	Archivo PHP	8 KB
index.html	Chrome HTML Do...	9 KB
logeo.php	Archivo PHP	6 KB
mp3.html	Chrome HTML Do...	1.645 KB
olvido.html	Chrome HTML Do...	4 KB
olvido.php	Archivo PHP	4 KB
registro.html	Chrome HTML Do...	6 KB
registro.php	Archivo PHP	6 KB
restringida.htm.php	Archivo PHP	8 KB
style.css	Archivo CSS	4 KB

Carpeta WebService

catalogo.json	Archivo JSON	639 KB
config.php	Archivo PHP	1 KB
connectbd.php	Archivo PHP	1 KB
crea_json.php	Archivo PHP	1 KB
db_descrip.php	Archivo PHP	1 KB
funciones_bd.php	Archivo PHP	2 KB
registro.php	Archivo PHP	3 KB
validaUser.php	Archivo PHP	1 KB

Web Service

Config.php

```php
<?php
	/**
	 * Database config variables
	 */
	// Nombre del host que alberga la BD.
	define("DB_HOST", "mysql.hostinger.es");
	// nombre de usuario definido en la configuracion de la BD.
	define("DB_USER", "u437638799_audio");
	// password elegido.
	define("DB_PASSWORD", "xxxxxxxxxx");
	// Nombre de la base de datos.
	define("DB_DATABASE", "u437638799_login");
?>
```

connectbd.php

```php
<?php

	class DB_Connect {

		// constructor
		function __construct() {
		}

		// destructor
		function __destruct() {
			// $this->close();
		}

		// Conectar la BD
		public function connect() {
			// Importar constantes
			require_once 'config.php';
			// conectar mysql
			$con = mysql_connect(DB_HOST, DB_USER, DB_PASSWORD);
			// seleccionar BD
			mysql_select_db(DB_DATABASE);

			// devuelve el manejador de la BD
			return $con;
		}
```

```
        // Cerrar la conexión de la BD
        public function close() {
            mysql_close();
        }

    }

?>
```

crea_json.php

```
<?php
        // Importar constantes
        require_once 'config.php';
        // Conectar y seleccionar BD
        mysql_connect(DB_HOST,DB_USER,DB_PASSWORD);
        mysql_select_db(DB_DATABASE);
        // Obtener resultados según selección en BD
        $sql=mysql_query("SELECT id, titulo, autor, tema, descripcion, mp3 FROM audiolibros ORDER BY
                        autor, titulo ASC");
        // Recorrer array obtenido y codificar en json
        while($row=mysql_fetch_assoc($sql))
                $output[]=$row;
                $jsonencoded = json_encode($output);

        // Hay que indicar la ruta para crear desde cronjob
        $filename = "/home/u437638799/public_html/WebService/catalogo.json";
        // Guardar en un fichero el resultado convertido en json
        $fh = fopen($filename, 'w');
        fwrite($fh, $jsonencoded);
        fclose($fh);
        print($jsonencoded);

        // Cerrar conexión a la BD
        mysql_close();
?>
```

db_descrip.php

```
<?php

	// Nos devolvera la descripcion del registro especificado
	// Esto lo devuelve el servidor y es lo que leera la app de android.

	// Obtener los datos
	$_id = $_POST['_id'];

	// Incorporar funciones
	require_once 'funciones_bd.php';
	$db = new funciones_BD();

	// Obtener la descripcion del registro especificado
	$sql = mysql_query("SELECT descripcion from audiolibros WHERE id = '$_id'");

	//numero de filas retornadas
	$num_rows = mysql_num_rows($sql);

	// Obtener valor obtenido
	if ($num_rows > 0) {
		// el registro existe
		$row = mysql_fetch_array($sql);
		$resultado = $row["descripcion"];
	} else {
		// no existe
		$resultado = null;
	}

	//echo json_encode($resultado);
	// Devolver valor obtenido
	echo $resultado;

?>
```

registro.php

```
<?php

	// Nos devolvera un valor indicando si el registro fue correcto o no.
	// 0  —> registro invalido –> el usuario ya existe
	// 1  —> registro invalido –> el correo no es valido
	// 2  —> registro invalido –> error de sistema
	// 3  —> registro valido
	// Esto lo devuelve el servidor y es lo que leera la app de android.

	// Obtener los datos
	$usuario = $_POST['usuario'];
	$passw   = $_POST['password'];
	$mail    = $_POST['correo'];
	$clave   = $_POST['clave'];

	// Decodificar valores en base64
	$usuario = base64_decode($usuario);
	$passw = base64_decode($passw);
	$mail = base64_decode($mail);
	$clave = base64_decode($clave);

	// Incorporar funciones
	require_once 'funciones_bd.php';
	$db = new funciones_BD();

	// Comprobar si el usuario existe
	if($db->isuserexist($usuario,$passw)){
		//echo(" Este usuario ya existe ingrese otro diferente!");
		// asignar resultado
		$resultado="0";

	}else{

		// verificar si el mail existe (hasta cierto punto), verificamos que el dominio existe
		// checkdnsrr — Comprueba registros DNS correspondientes a un nombre de host de Internet
		dado o   dirección IP
		$dominio = explode('@',$mail); // devuelve un array
		if(!checkdnsrr($dominio[1])){
			// asignar resultado
			$resultado="1";
		}
		else{
		  if($db->adduser($usuario,$passw,$mail)){
			//echo(" El usuario fue agregado a la Base de Datos correctamente.");

			// preparar y enviar mensaje
			$para=$mail;
```

```
                    $de='admin@audiolibros.hol.es';
                    $asunto='AVISO: Registro efectuado';
                    $mensaje="Bienvenido a audiolibros en español.\n".
                     "Los datos de registro son:\n\tUSUARIO: ".$usuario."\n\tCLAVE: ".$clave.
                     "\n\n http://www.audiolibros.hol.es";
                    $cabecera="FROM: ".$de;
                    //La direccion de correo desde donde supuestamente se envió
                    $cabeceras = "From: admin@audiolibros.hol.es\n";
                    //La direccion de correo a donde se responderá
                    $cabeceras .= "Reply-To: admin@audiolibros.hol.es\n";

                    // Enviar correo
                    mail($para, $asunto, $mensaje, $cabeceras);

                    // asignar resultado
                    $resultado="3";

                  }else{
                            //echo(" ha ocurrido un error.");
                            // asignar resultado
                            $resultado="2";
                  }
            }
      }

      //echo json_encode($resultado);
      // Devolver resultado
      echo $resultado;

?>
```

valida_User.php

```
<?php

      // Nos devolvera un valor indicando si la validacion fue correcto o no.
      // 0  —> validacion incorrecta –> el usuario o contraseña no valida
      // 1  —> validacion correcta   –> el correo no es valido
      // Esto lo devuelve el servidor y es lo que leera la app de android.

      // Obtener los datos
      $usuario = $_POST['usuario'];
      $passw   = $_POST['password'];

      // Decodificar valores en base64
      $usuario = base64_decode($usuario);
      $passw = base64_decode($passw);
```

```
        // insertar funciones
        require_once 'funciones_bd.php';
        $db = new funciones_BD();

        if($db->login($usuario,$passw)){
                //echo(" usuario y contraseña válidas ");
                // asignar resultado
                $resultado="1";
        }else{
                //$resultado="0"." user: ".$usuario." pass: ".$passw;
                // asignar resultado
                $resultado="0";
        }

        //echo json_encode($resultado);
        // Devolver resultado
        echo $resultado;

?>
```

funciones_bd.php

```
<?php

class funciones_BD {

        private $db;

        // constructor
        function __construct() {
                // insertar conexion
                require_once 'connectbd.php';
                // conectar a la BD
                $this->db = new DB_Connect();
                $this->db->connect();

        }

        // destructor
        function __destruct() {
        }

         /**
        * agregar nuevo usuario
        */
        public function adduser($username, $password, $mail) {

                // Realizar inserción en la BD
```

```
        $result = mysql_query("INSERT INTO usuario(login,pass,email) VALUES('$username',
                                        '$password', '$mail')");

        // comprobar si la inserccion es correcta y retornar resultado
        if ($result) {
                return true;

        } else {
                return false;
        }

}


/**
 * Verificar si el usuario ya existe por el username
*/
public function isuserexist($username) {

        // Consulta sobre la BD
        $result = mysql_query("SELECT login FROM usuario WHERE login = '$username'");
        //numero de filas retornadas
        $num_rows = mysql_num_rows($result);

        // retornar resultado
        if ($num_rows > 0) {
                // el usuario existe
                return true;
        } else {
                // no existe
                return false;
        }
}
```

```
	public function login($user,$passw){

		// Consulta sobre la BD
		$result=mysql_query("SELECT COUNT(*) FROM usuario WHERE login='$user' AND
								pass='$passw' ");
		// Devuelve en numero de filas
		$count = mysql_fetch_row($result);

		/*como el usuario debe ser unico cuenta el numero de ocurrencias con esos datos*/
		if ($count[0]==1){
			return true;
		}else{
			return false;
		}
	}

} //  funciones_BD

?>
```

Sitio Web

conec.php

```
<?php

	function conectarse()
	{
		// Datos para la conexion
		$host = "mysql.hostinger.es";
		$dbname = "u437638799_login";
		$username = "u437638799_audio";
		$password = "";

		if(!($link=mysql_connect($host,$username,$password)))
		{
			//echo "ERROR EN EL SERVIDOR";
			 exit();
		}
		if(!mysql_select_db($dbname,$link))
		{	//echo "ERROR EN LA BASE DE DATOS";
			exit();
		}

		//echo "CONEXION EXITOSA A LA BASE DE DATOS";
		return $link;
	}

?>
```

funciones.php

```
<?php
	// iniciar sesion
	session_start();

	// mascaras para añadir a la contraseña y cookies
	$salt = "|#€7`¬23ads4ook12";
	$saltCookie = "|@#57e+ç´|@#d";

	// insertar conexion
	require ("conec.php");


/**
 * Comprueba que exista una sesion o una cookie en la página de login
 */
function seguridadIndex()
{

   if (isset($_SESSION['usuario']))
   {
		//calculamos el tiempo transcurrido
		$fechaGuardada = $_SESSION["ultimoAcceso"];
		$ahora = date("Y-n-j H:i:s");
		$tiempo_transcurrido = (strtotime($ahora)-strtotime($fechaGuardada));

		//comparamos el tiempo transcurrido
		if($tiempo_transcurrido >= 900)
		{
			//si pasaron 15 minutos o más destruyo la sesión
			//guardar el nombre de la sessión para luego borrar las cookies
			$session_name = session_name();
			session_destroy();

			// Para borrar las cookies asociadas a la sesión
			// Es necesario hacer una petición http para que el navegador las elimine
			if ( isset( $_COOKIE[ $session_name ] ) ) {
				setcookie($session_name, '', time()-3600, '/');
			}
			if(isset($_COOKIE['identificado'])){
				setcookie('identificado', '', time()-3600, '/');
			}
			if(isset($_COOKIE['iduser'])){
				setcookie('iduser', '', time()-3600, '/');
			}

			// petición http
```

```
                        header("Location: index.html");

                }
                else {
                    //sino, actualizo la fecha de la sesión
                        $_SESSION["ultimoAcceso"] = $ahora;
                        header("Location: restringida.htm.php");
                        exit();
                }
    }
    else if( isset($_COOKIE['identificado']))
    {
        $cookie = limpiar($_COOKIE['identificado']);
        $idusuario = comprobarCookie($cookie);

        // si usuario no es nulo es que tiene una cookie válida
        if($idusuario <> "")
        {
            header("Location: restringida.htm.php");
            exit();
        }

    }

}


/**
 * Comprueba que exista una sesion o una cookie, sino redirige al login
 *
 * @return int estado
 */
function seguridad(){

    if (isset($_SESSION['usuario']))
    {
                //calculamos el tiempo transcurrido
                $fechaGuardada = $_SESSION["ultimoAcceso"];
                $ahora = date("Y-n-j H:i:s");
                $tiempo_transcurrido = (strtotime($ahora)-strtotime($fechaGuardada));

                //comparamos el tiempo transcurrido
                if($tiempo_transcurrido >= 900)
                {
                        //si pasaron 15 minutos o más destruyo la sesión
                        //guardar el nombre de la sessión para luego borrar las cookies
                        $session_name = session_name();
                        session_destroy();
```

```
                    // Para borrar las cookies asociadas a la sesión
                    // Es necesario hacer una petición http para que el navegador las elimine
                    if ( isset( $_COOKIE[ $session_name ] ) ) {
                            setcookie($session_name, '', time()-3600, '/');
                    }
                    if(isset($_COOKIE['identificado'])){
                            setcookie('identificado', '', time()-3600, '/');
                    }
                    if(isset($_COOKIE['iduser'])){
                            setcookie('iduser', '', time()-3600, '/');
                    }
                            //envío al usuario a la pag. de inicio
                    ?>
                      <script type="text/javascript">
                              alert("La sesión ha expirado. \n \n"+
                                "Tiene que entrar de nuevo al área de usuarios registrados. \n \n");
                              document.location.href='index.html'
                      </script>
                    <?php
            }
            else {
                    //sino, actualizo la fecha de la sesión
                    $_SESSION["ultimoAcceso"] = $ahora;
                    return;
            }
  }
  else if( isset($_COOKIE['identificado']))
  {
    $cookie = limpiar($_COOKIE['identificado']);
    $idusuario = comprobarCookie($cookie);
    if(!$idusuario)
    {
      echo "<script language='javascript'>
                document.location.href='restringida.htm.php'
            </script>";
      exit();
    }
  }
  else
  {
    echo "<script language='javascript'>
              document.location.href='index.html'
          </script>";
    exit();
  }

}
```

```
/**
 * Comprueba que la cookie sea validad en nuestra BD
 *
 * @param string $cookie
 * @return int idUsuario
 */
function comprobarCookie($cookie)
{

        $conexion=conectarse();

        // establecer codificacion BD
        mysql_query("SET NAMES 'utf8'");

        // Realizar consulta sobre la BD
         $sql = "select idUsuario from usuario where cookie='".mysql_escape_string($cookie)."' and validez
                        >'".date("Y-m-d h:i:s")."'";

        // Obtener resultados
        $result = mysql_query($sql,$conexion);

        // Comprobar resultados
        if(!$result || mysql_affected_rows()<1) return false;
        else
        {
                $row = mysql_fetch_array($result);
                $_SESSION['usuario']=$row['idUsuario'];
                return $row['idUsuario'];
        }
}


/**
 * Comprueba y el user y pass son correcto. En caso de querer ser recordado en el pc, crea la cookie
 *
 * @global string $salt
 * @global string $saltCookie
 * @param string $user
 * @param string $pass
 * @param bool $recordarme
 * @return int estado
 */
function login ($user,$pass,$recordarme)
{
   // limpiar contenido variables
   $user = mysql_escape_string($user);
   $pass = mysql_escape_string($pass);

   // Comprobar longitud
```

```
    if(strlen($user)<4 || strlen($pass)<4) return -3;

    global $salt;
    $pass = sha1($salt.md5($pass));

    // Realizar conexión y establecer codificacion
    $conexion=conectarse();
    mysql_query("SET NAMES 'utf8'");

    // Realizar consulta sobre la BD
    $sql = "select idUsuario, login from usuario where UPPER(login)='".strtoupper($user)."' and
pass='".$pass."'";

    // Obtener resultado
    $result = mysql_query($sql,$conexion);
     // Comprobar resultado
     if(mysql_affected_rows()<=0 || !$result) return -1; //user no existe

    $row = mysql_fetch_array($result);
    $idUsuario = $row['idUsuario'];

    $nameuser = $row['login'];

    //definir usuario y la fecha y hora de inicio de sesión en formato aaaa-mm-dd hh:mm:ss
    $_SESSION['usuario']=$idUsuario;
    $_SESSION["ultimoAcceso"]= date("Y-n-j H:i:s");

    // cookie con nombre de usuario
    setCookie("iduser",$nameuser,time()+900,'/'); //cookie 15min


    if($recordarme){
        global $saltCookie;

        $cookie = sha1($saltCookie.md5($idUsuario.date("Y-d-m h:i:s")));

        $sql2 = "update usuario set cookie='".$cookie."',validez=DATE_ADD(now(),INTERVAL 15 MINUTE)
where                `idUsuario`='".$idUsuario."'";
        $result2 = mysql_query($sql2,$conexion);

        setCookie("identificado",$cookie,time()+900,'/'); //cookie 15min

    }

    return true;
}
```

```
function destruirCookie($cookie)
{
   if(!isset($_SESSION['usuario'])) return;
   else $idusuario = $_SESSION['usuario'];

   $conexion=conectarse();
   mysql_query("SET NAMES 'utf8'");

   $sql = "update usuario set validez=DATE_SUB(now(),INTERVAL 6 MINUTE) where
`idUsuario`='".$idusuario."'";
   $result = mysql_query($sql,$conexion);

   if(mysql_affected_rows()>0)
            return true; //cookie puesta invalida
   else
           return false;

}


           /**
           * @param string $valor
           * @return string string limpiado de fallos de seguridad
           */
           function limpiar($valor)
           {
                    $valor = strip_tags($valor);
                    $valor = stripslashes($valor);
                    $valor = htmlentities($valor);
                    return $valor;
           }

?>
```

index.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>
        <title>Audiolibros en Español</title>

        <meta content="text/html;" http-equiv="content-type" charset="utf-8"/>
        <link href="style.css" rel="stylesheet" type="text/css" />

        <script type="text/javascript">

        // abre una ventana con las caracteristicas especificadas
        function AbrirCentrado(Url,NombreVentana,width,height,extras) {
                var largo = width;
                var altura = height;
                var adicionales= extras;
                var top = (screen.height-altura)/2;
                var izquierda = (screen.width-largo)/2;
                nuevaVentana=window.open("+ Url + ","+ NombreVentana + ",'width=' + largo + ',height=' +
                                altura +  ',top=' + top + ',left=' + izquierda + ',features=' + adicionales + ");
                nuevaVentana.focus();
        }

        // muestra el DIV especificado
        function muestra(id){
                 var elDiv = document.getElementById(id); //se define la variable "elDiv" igual a nuestro div
                elDiv.style.display='block';            //damos un atributo display:block que  el div
        }

        // oculta el DIV especificado
        function oculta(id){
                var elDiv = document.getElementById(id); //se define la variable "elDiv" igual a nuestro div
                elDiv.style.display='none';            //damos un atributo display:none que oculta el div
        }


        function valida_log(form1)
        {
                if (form1.txtUsuario.value==""){
                        alert("USUARIO esta vacío...");
                        form1.txtUsuario.focus();
                        return;
                }
                if (form1.txtClave.value==""){
```

```
                    alert("CLAVE está vacía...");
                    form1.txtClave.focus();
                    return;
                }

                form1.submit();
                return;
        }

        </script>

        <style>
                .fieldarea {FONT-SIZE: 14px; FONT-FAMILY:  tahoma, arial; font-weight: bold;
                TEXT-ALIGN: center}
        .pequeña{FONT-SIZE: 11px; FONT-FAMILY: Tahoma, arial}
        .login1 {display:none; position: absolute; z-index: 2; left: 30px; top: 155px; width: 125px}
        .rotulo1 {position: absolute; z-index: 2; left: 245px; top: 20px}
        .acercade1 {display:none; position: absolute; left: 5px; top: 353px;
background:url(images/leftbg.gif)                    repeat-x;  width:745px;  padding:10px 10px
10px 20px;}

        </style>

</head>

<body>

        <div id="container">

          <div id="header">

                <ul>
                <li id="navtop"><br />
                </li>
                <li><a class="link" href="#" onClick="oculta('acercade');">INICIO</a></li>
                <li><a class="link" href="logeo.php" onClick="oculta('acercade');">ENTRAR </a></li>

                <li><a class="link" href="registro.html"
onClick="oculta('acercade');">REGISTRO</a></li>          <li><a class="link" href="#"
onClick="muestra('acercade');">ACERCA DE <br /></a></li>
                <li id="navbottom"><br />
                </li>
                </ul>

          <div id="rotulo" class="rotulo1">
                <p  style= "text-align: center;  font-family: Comic Sans MS,cursive; font-size: 40px;
color:                    rgb(156, 31, 156);">
                        <strong>Audiolibros en español</strong>
                </p>
```

```
    </div>
    </div>

<div class="inner_copy" />

</div>

<div id="content">

        <div id="left">
                <h3 align=center>QUE OFRECEMOS</h3>
                <p>Audiolibros en español.</p>
                <p><br /></p>
                <p>Reproducción on-line.</p>
                <p><br /></p>
                 <p>Escucha tus audiolibros favoritos desde tu android con nuestra App.</p>
                <p><br /></p>
                <p><br /></p>
                <p><br /><br /><br /><br /><br /><br /><br /></p>
                <p align=center></p>
                <img id="djset" src="images/movil.jpg" alt="movil" />
        </div>

        <div id="middle">
                <h3>AUDIOLIBROS EN ESPAÑOL</h3>
                <p>Registrate y disfruta de nuestro amplio catálogo de audiolibros en
español.<br>                <a href="registro.html">Registrarme.</a>                </p>
                <br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br                <br />
                <img id="photos" src="images/photos.jpg" alt="libros" />
        </div>

        <div id="right">
                <img style="border: 0px solid ; width: 265px; height: 244px;"
                        src="images/audiolibros.JPG" alt="audiolibros en español" />

                <p></p>
                <p align=center>Visita nuestro <a href="catalogo.html" TARGET="_blank">
                        Catálogo</a></p>
        </div>

        <div class="clear" >
        </div>

        <div id="footer">
                <p class="lf" >&copy; Copyright 2013 AmmSoft.</a></p>
                <ul class="fmenu">
                  <li ><a href="#" onClick="oculta('acercade');">Inicio</a></li>
                  <li><a href="logeo.php" onClick="oculta('acercade'); ">Entrar</a></li>
```

```
                    <li><a href="registro.html" onclick="oculta('acercade');">Registro</a></li>
                    <li><a href="#" onClick="muestra('acercade');">Acerca de</a></li>
                  </ul>
                  <div class="clr"></div>
            </div>

            <div class="fclear" >
            </div>
      </div>

      <div id="acercade" class="acercade1" >
            <h3 align=center>Audiolibros.hol.es</h3>
            <p>  Disponemos de un amplio catálogo compuesto por más de 3000 títulos, con la más
variada                  temática.</p>
            <p><br /></p>
            <p>Los audiolibros son grabaciones de los contenidos de un libro, leídos en voz alta por
un                              narrador.
                  Las grabaciones se presentan en formato digital a través de descargas o para
                              reproducirlos on-line.</p>
            <p><br /></p>
            <p>Los usuarios registrados podrán acceder a la descarga o reproducción de los
audiolibros a                   partir del menú de acceso a este servicio. El sistema les pedirá un login y
un password.
                  El servicio dará acceso al catálogo de audiolibros. </p>
            <p><br /></p>
            <p>Además si te registras, descarga nuestra App y escucha tus audiolibros favoritos
desde tu                        android.</p>
            <p><br /></p>

            <table width="95%">
                  <tr>
                        <td align="center">
                              <img src="images/lector.jpg" alt="audiolibro" />
                        </td>
                  </tr>
                  <tr>
                        <td align="center">
                              <a href="" onClick="oculta('acercade');">Volver</a>
                        </td>
                  </tr>
            </table>
      </div>

</body>

</html>
```

catalogo.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
        "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">


<HTML>
<HEAD>
        <TITLE>Audiolibros. Catálogo de libros audibles. 3200 Libros de audio mp3 gratis.</TITLE>
        <meta content="text/html; charset=windows-1252" http-equiv=Content-Type>
        <!-- <meta content="text/html;" http-equiv="content-type" charset="utf-8"/> -->

        <style>

        BODY {BACKGROUND: url(fondo.png) #ffffff repeat-x; MARGIN: 0px}

        </style>

</HEAD>

<BODY>

        <table align="center" cellpadding="0" cellspacing="0">
                <tr>
                        <td width="100%" height="122" style="background:url('fondo.png') repeat-
x;">
                                <img src="cab01.png" alt="cab">
                        </td>
                </tr>
        </table>

        <table width="100%" cellpadding="0" cellspacing="0" style="padding-left:35px; padding-
                right:35px;border-bottom:1px solid #CCCCCC; background:#EFEFEF;">
                <tr>
                        <td height="30">
                                Usted est&aacute; aqu&iacute;: <b>Catálogo</b>
                                </td>

                                <td style="text-align:right;">
                                        <a href="#" onclick="javascript:window.close();">Cerrar</a>
                                </td>
                </tr>
        </table>

        <br>
        <TABLE align="center" cellSpacing=1 cellPadding=0 border=0>
                <TR>
                  <td valign="top" style=" padding-left:20px;padding-bottom:20px;">
                        <br>
```

```
                    <h3><strong>Listado de títulos disponibles.</strong></h3>
                  </td>
                </TR>
        </TABLE>

<TABLE style="WIDTH: 30345pt; BORDER-COLLAPSE: collapse; TABLE-LAYOUT: fixed"
        border=0 cellSpacing=0 cellPadding=0 width=40545>
        <COLGROUP>
        <COL style="WIDTH: 119pt; mso-width-source: userset; mso-width-alt: 5814"  span=255
width=159>
 <TBODY >
 <TR style="HEIGHT: 12pt; mso-height-source: userset" height=16>
  <TD style="HEIGHT: 12pt" class=xl67 height=16>Abentofail</TD>
  <TD class=xl68>Filósofo autodidacto, El</TD>
  <TD class=xl67>ESPAÑOLA</TD>
  <TD class=xl67>Filosofía</TD>
  <TD style="mso-ignore: colspan" class=xl85 colSpan=2></TD>
  <TD class=xl67></TD>
  <TD style="mso-ignore: colspan" colSpan=248></TD></TR>
 <TR style="HEIGHT: 12pt; mso-height-source: userset" height=16>
  <TD style="HEIGHT: 12pt" class=xl67 height=16>Abramov, Alexander y S<SPAN
   style="DISPLAY: none">ergei</SPAN></TD>
  <TD class=xl67>Escala del tiempo, La</TD>
  <TD class=xl67>RUSA</TD>
  <TD class=xl67>Cuento Ciencia Ficción</TD>
  <TD style="mso-ignore: colspan" class=xl85 colSpan=2></TD>
  <TD class=xl67></TD>
  <TD style="mso-ignore: colspan" colSpan=248></TD></TR>
 <TR style="HEIGHT: 12pt; mso-height-source: userset" height=16>
  <TD style="HEIGHT: 12pt" class=xl67 height=16>Abramov, Alexander y S<SPAN
   style="DISPLAY: none">ergei</SPAN></TD>
  <TD class=xl67>Viaje por tres mundos</TD>
  <TD class=xl67>RUSA</TD>
  <TD class=xl67>Novela Ciencia Ficción</TD>
  <TD style="mso-ignore: colspan" class=xl86 colSpan=2></TD>
  <TD class=xl67></TD>
  <TD style="mso-ignore: colspan" colSpan=248></TD></TR>
 <TR style="HEIGHT: 12pt; mso-height-source: userset" height=16>
  <TD style="HEIGHT: 12pt" class=xl67 height=16>Acebal, Francisco</TD>
  <TD class=xl68>Alma asturiana, alma es<SPAN
   style="DISPLAY: none">pañola</SPAN></TD>
  <TD class=xl67>ESPAÑOLA</TD>
  <TD class=xl67>Artículo</TD>
  <TD style="mso-ignore: colspan" class=xl85 colSpan=2></TD>
  <TD class=xl67></TD>
        .
        .
        .
</TBODY></TABLE>
```

```
        <table width="100%" cellpadding="0" cellspacing="0" style="padding-left:35px; padding-
                        right:35px;border-        bottom:1px solid #CCCCCC;
background:#EFEFEF;">
                <tr>
                        <td height="30">
                                Usted est&aacute; aqu&iacute;: <b>Catálogo</b>
                        </td>

                        <td style="text-align:right;">
                                <a href="#" onclick="window.close();">Cerrar</a>
                        </td>
                </tr>
        </table>

        <br> <br>

 </BODY>
</HTML>
```

descargas.html

```
<?php

        require ("funciones.php");

        //comprobamos sesion valida
        seguridad();
?>

<HTML>
<HEAD>
        <TITLE>Audiolibros. Catálogo de libros audibles. 3200 Libros de audio mp3 gratis.</TITLE>
        <meta content="text/html; charset=windows-1252" http-equiv=Content-Type>
        <!-- <meta content="text/html;" http-equiv="content-type" charset="utf-8"/> -->

        <style>
                BODY {BACKGROUND: url(fondo.png) #ffffff repeat-x; MARGIN: 0px}
        </style>

</HEAD>

<BODY>

        <table align="center" cellpadding="0" cellspacing="0">
                <tr>
                        <td width="100%" height="122" style="background:url('fondo.png') repeat-
x;">
```

```
                                <img src="cab01.png" alt="cab">
                        </td>
                </tr>
        </table>

        <table width="100%" cellpadding="0" cellspacing="0" style="padding-left:35px; padding-
                        right:35px;border-bottom:1px solid #CCCCCC; background:#EFEFEF;">
                <tr>
                        <td height="30">
                                        Usted est&aacute; aqu&iacute;: <b>Descargar Mp3</b>
                                </td>

                                <td style="text-align:right;">
                                        <a href="#" onclick="javascript:window.close();">Cerrar</a>
                                </td>
                </tr>
        </table>

        <TABLE align="center" cellSpacing=1 cellPadding=0 border=0>
                <TR>
                    <td valign="top" style=" padding-left:20px;padding-bottom:20px;">
                        <br><br>
                        <strong>Para descargar un archivo: pulse botón derecho del ratón y guardar
enlace                          cómo...</strong>
                    </TD>
                </TR>
        </TABLE>

        <br>

        <table x:str border=0 cellpadding=0 cellspacing=0 width=1272 style='border-collapse: collapse;table-
                        layout:fixed;width:952pt'>
                <col width=159 span=256 style='mso-width-source:userset;mso-width-alt:5814;
width:119pt'>
                 <tr class=xl27 height=16 style='mso-height-source:userset;height:12.0pt'>
                        <td height=16 class=xl27 width=159
style='height:12.0pt;width:119pt'>Abentofail</td>
                        <td class=xl26 width=159 style='width:119pt'>Filósofo autodidacto, El</td>
                        <td class=xl27 width=159 style='width:119pt'>ESPAÑOLA</td>
                        <td class=xl27 width=159 style='width:119pt'>Filosofía</td>
                        <td class=xl28 colspan=3 width=477 style='mso-
ignore:colspan;width:357pt'><href="http://www.leemp3.com/leemp3/el_filosofo_autodidactico.mp3">http
://www.leemp3.com/leemp3/el_filosofo_autodidactico.mp3</a></td>
                        <td class=xl27 width=159 style='width:119pt'></td>
                </tr>
                <tr class=xl27 height=16 style='mso-height-source:userset;height:12.0pt'>
                        <td height=16 class=xl27 style='height:12.0pt'>Abramov, Alexander y S<span
                                style='display:none'>ergei</span></td>
                        <td class=xl27>Escala del tiempo, La</td>
```

```
                        <td class=xl27>RUSA</td>
                        <td class=xl27>Cuento Ciencia Ficción</td>
                        <td class=xl28 colspan=3 style='mso-ignore:colspan'><a

href="http://www.leemp3.com/leemp3/1/escala_tiempo_abramov.mp3">http://www.leemp3.com/leemp3/1
\escala_tiempo_abramov.mp3</a></td>
                        <td class=xl27></td>
                </tr>
        .
        .
        .
        </table>

        <br> <br>

        <table width="100%" cellpadding="0" cellspacing="0" style="padding-left:35px; padding-
                        right:35px;border-        bottom:1px solid #CCCCCC;
background:#EFEFEF;">
                <tr>
                        <td height="30">
                                Usted est&aacute; aqu&iacute;: <b>Descargar Mp3</b>
                        </td>

                        <td style="text-align:right;">
                                <a href="#" onclick="window.close();">Cerrar</a>
                        </td>
                </tr>
        </table>

</body>
</html>
```

mp3.html

```
<?php

	require ("funciones.php");

	seguridad(); //comprobamos sesion valida
?>

<HTML>
<HEAD>
	<TITLE>Audiolibros. Catálogo de libros audibles. 3200 Libros de audio mp3 gratis.</TITLE>
	<meta content="text/html; charset=windows-1252" http-equiv=Content-Type>
	<!-- <meta content="text/html;" http-equiv="content-type" charset="utf-8"/> -->

	<style>

	BODY {BACKGROUND: url(fondo.png) #ffffff repeat-x; MARGIN: 0px}

	</style>

</HEAD>

<BODY>

	<table align="center" cellpadding="0" cellspacing="0">
		<tr>
			<td width="100%" height="122" style="background:url('fondo.png') repeat-
x;">
				<img src="cab01.png" alt="cab">
			</td>
		</tr>
	</table>

	<table width="100%" cellpadding="0" cellspacing="0" style="padding-left:35px; padding-
			right:35px;border-bottom:1px solid #CCCCCC; background:#EFEFEF;">
		<tr>
			<td height="30">
				Usted est&aacute; aqu&iacute;: <b>Reproducir Mp3</b>
			</td>

			<td style="text-align:right;">
				<a href="#" onclick="javascript:window.close();">Cerrar</a>
			</td>
		</tr>
	</table>

	<TABLE align="center" cellSpacing=1 cellPadding=0 border=0>
```

```
                <TR>
                    <td valign="top" style=" padding-left:20px;padding-bottom:20px;">
                        <br><br>
                        <strong>Para escuchar audiolibros on-line debe tener instalado Flash
Player...</strong>
                    </TD>
                </TR>
                <TR>
                    <td class="pequeña" align="center" >
                        <a href="http://get.adobe.com/es/flashplayer/otherversions/"
                                    TARGET="_blank">Instalar Flashplayer</a>
                            <br> <br>
                    </TD>
                </TR>
        </TABLE>

        <br>

        <table x:str border=0 cellpadding=0 cellspacing=0 width=1272 style='border-collapse:
                collapse;table-layout:fixed;width:952pt'>
                <col width=159 span=256 style='mso-width-source:userset;mso-width-alt:5814;
width:119pt'>
        <tr class=xl27 height=16 style='mso-height-source:userset;height:12.0pt'>
                 <td height=16 class=xl27 width=159 style='height:12.0pt;width:119pt'>Abentofail</td>
                <td class=xl26 width=159 style='width:119pt'>Filósofo autodidacto, El</td>
                <td class=xl27 width=159 style='width:119pt'>ESPAÑOLA</td>
                <td class=xl27 width=159 style='width:119pt'>Filosofía</td>
                <td class=xl28 colspan=3 width=477 style='mso-ignore:colspan;width:357pt'><a

href="http://www.leemp3.com/leemp3/el_filosofo_autodidactico.mp3">http://www.leemp3.com/leemp3/el
_filosofo_autodidactico.mp3</a></td>
                <td class=xl27 width=159 style='width:119pt'></td>
        </tr>
        <tr class=xl27 height=16 style='mso-height-source:userset;height:12.0pt'>
                <td height=16 class=xl27 style='height:12.0pt'>Abramov, Alexander y S<span
                        style='display:none'>ergei</span></td>
                <td class=xl27>Escala del tiempo, La</td>
                <td class=xl27>RUSA</td>
                <td class=xl27>Cuento Ciencia Ficción</td>
                <td class=xl28 colspan=3 style='mso-ignore:colspan'><a

href="http://www.leemp3.com/leemp3/1/escala_tiempo_abramov.mp3">http://www.leemp3.com/leemp3/1
\escala_tiempo_abramov.mp3</a></td>
                <td class=xl27></td>
        </tr>
        .
        .
        .
</table>
```

```
        <br> <br>

 <table width="100%" cellpadding="0" cellspacing="0" style="padding-left:35px; padding-
right:35px;border-        bottom:1px solid #CCCCCC; background:#EFEFEF;">
        <tr>
                <td height="30">
                        Usted est&aacute; aqu&iacute;: <b>Descargar Mp3</b>
                </td>

                <td style="text-align:right;">
                        <a href="#" onclick="window.close();">Cerrar</a>
                </td>
        </tr>
</table>


</body>
</html>
```

datos.html.php

```
<?php
	// Si la sesión no existía, esta función crea la sesión y le asocia un identificador de sesión único.
	// Si la sesión ya existía, esta función permite que la página tenga acceso a la información vinculada a la
sesión.
	require ("funciones.php");
	//comprobamos sesion valida
	seguridad();
?>


<html>

<head>
	<meta content="text/html;" http-equiv="content-type" charset="utf-8"/>
	<title>AUDIOLIBROS - Mis Datos </title>

<style>

	BODY {BACKGROUND: url(fondo.png) #ffffff repeat-x; MARGIN: 0px  }
	TABLE.wrapper {WIDTH: 760px}
	TD {FONT-SIZE: 11px; FONT-FAMILY: Tahoma}
	FORM {MARGIN: 0px}
	INPUT {FONT-SIZE: 11px; FONT-FAMILY: Tahoma}
	.navbar {BACKGROUND: #8fbce9}
	.navbox {PADDING-RIGHT: 5px; PADDING-LEFT: 5px; PADDING-BOTTOM: 5px;
PADDING-TOP:		5px; BACKGROUND-COLOR: #eff2f9}
	.navinput {BORDER-RIGHT: #8fbce9 1px solid; BORDER-TOP: #8fbce9 1px solid; BORDER-
LEFT:		#8fbce9 1px solid; WIDTH: 100px; BORDER-BOTTOM: #8fbce9 1px solid}
	.submitbutton {BORDER-RIGHT: #8fbce9 1px solid; BORDER-TOP: #8fbce9 1px solid; FONT-
		WEIGHT: bold; BORDER-LEFT: #8fbce9 1px solid; CURSOR: hand; COLOR:
#ffffff;		BORDER-BOTTOM: #8fbce9 1px solid; BACKGROUND-COLOR: #8fbce9}
	.fieldarea {BACKGROUND-COLOR: #eff2f9; TEXT-ALIGN: right}

</style>

<script type="text/javascript">

	function valida(form1)
	{

		if(form1.txtMail.value.length==0){
			//alert("E-MAIL es obligatorio...");
			//form1.txtMail.focus();
			//return;

		}
		else {
```

```
                valor = document.getElementById("txtMail").value;
                expr = /^([a-zA-Z0-9_\.\-])+\@(([a-zA-Z0-9\-])+\.)+([a-zA-Z0-9]{2,4})+$/;

                if( !(expr.test(valor)) ) {
                        alert("Error: La dirección de correo: \n\n"+valor+"\n\nes incorrecta.");
                        form1.txtMail.focus();
                        return;
                }
                else {
                        // validar si existe
                        //alert("Correo correcto");
                }

        }

        if(form1.txtClave.value==""){
                //alert("Contraseña es obligatoria...");
                //form1.txtClave.focus();
                //return;
        }
        else {
                if(form1.txtClave.value.length < 4){
                        alert("CONTRASEÑA debe tener cuatro caracteres como mínimo...");
                        form1.txtClave.focus();
                        return;
                }
        }

        if(form1.txtClave.value!=form1.txtClave2.value){
                alert("Contraseña no coincide...");
                form1.txtClave.value="";
                form1.txtClave2.value="";
                form1.txtClave.focus();
                return;
        }

        form1.txtMail.value=form1.txtMail.value.toLowerCase();
        form1.submit();

        return;

    }

</script>

</head>

<body>
```

```
<table align="center" cellpadding="0" cellspacing="0">
        <tr>
                <td width="100%" height="122" style="background:url('fondo.png') repeat-
x;">
                        <img src="cab01.png" alt="cab">
                </td>
        </tr>
</table>

<table width="100%" cellpadding="0" cellspacing="0" style="padding-left:35px; padding-
                right:35px;border-bottom:1px solid #CCCCCC; background:#EFEFEF;">
        <tr>
                <td height="30">
                        Usted est&aacute; aqu&iacute;: <b>Mis datos </b>
                </td>

                <td style="text-align:right;"><a href="restringida.htm.php">Regresar </a>
                </td>
        </tr>
</table>

<br>

<TABLE align="center" cellSpacing=1 cellPadding=0 border=0>
        <TBODY>
                <TR>
                  <td valign="top" style=" padding-left:20px;padding-bottom:20px;">
                        <br><br>
                        <p>Datos de registro, puede cambiar la dirección email y la
contraseña.</p>
                </td>
                </TR>
                <TABLE class=navbar align="center" cellSpacing=1 cellPadding=2 width="45%"
                        border=0>
                        <TR>
                                <?php
                                        $conexion=conectarse();

                                        $idusuario = $_SESSION['usuario'];

                                        // recuperar datos según idusuario
                                        $registro=mysql_query("SELECT login,pass,email FROM
usuario                                 WHERE idusuario=$idusuario",$conexion);
                                        if ($reg=mysql_fetch_array($registro))
                                        {
                                ?>
                                td width="100%" class=navbox valign="top" align="center" >
```

```
                                        <form id="form1" name="form1"  method="post"
                                                    action="datos.php" >
                                        <table cellspacing="0" cellpadding="0" >
                                        <tr><td>
                                          <table width="100%" cellpadding="3">
                                                <tr><td class="fieldarea">Usuario:</td>
                                                <td><h3><?php echo $reg['login']; ?></h3></td></tr>
                                                <tr><td class="fieldarea">Direcci&oacute;n Email:</td>
                                                <td><h3><?php echo $reg['email']; ?></h3></td></tr>
                                                <tr><td class="fieldarea">Cambiar Direcci&oacute;n
                        Email:</td><td><input type="text" name="txtMail" id="txtMail" size=40
value=""> </td></tr>
                                                <script>
                                                        document.forms[0].txtMail.focus();
                                                </script>
                                                <tr><td class="fieldarea"></td><td></td></tr>
                                                <tr><td class="fieldarea">Cambiar
                        Contrase&ntilde;a:</td><td><input name="txtClave" id="txtClave" size="18"
maxlength="15"                  type="password"> (max. 15 caracteres) </td></tr>
                                                <tr><td class="fieldarea">></td><td><input
name="txtClave2"                class="formulario_peque&ntilde;o" id="txtClave2" size="18"
maxlength="15"                  type="password">reescribe tu contrase&ntilde;a
</td></tr>
                                                <tr><td class="fieldarea">></td><td></td></tr>
                                                <tr>
                                                <td colspan="2" align="center"
                                                        <input type="hidden" name="txtLogin"
id="txtLogin"                                           value="<?php echo $reg['login']; ?>"/>
                                                                <input type="hidden"
name="txtId" id="txtId"                                                         value="<?php
echo $idusuario; ?>"/>
                                                        <input type="button" name="Button"
value="Enviar"                                          onclick="javascript:valida(form1)" />
                                                </td>
                                                </tr>


                                        </table>
                                        </td></tr>
                                        </table>
                                        </form>
                                </td>
                        <?php
                        }
                        else {
                                //echo 'No existe un artículo con dicho código';
                                ?>
                                        <script type="text/javascript">
                                        alert("Error, sesión no válida. \n \n");
```

```
                                        </script>
                                <?php
                        }

                        mysql_close($conexion);

                ?>

                </TR>
                </table>
        </TBODY>
        </TABLE>
        <br>

</body>
</html>
```

datos.php

```
<?php

                // asignar variables pasadas desde formulsrio
                $usuario=$_POST["txtLogin"];
                $clave=$_POST["txtClave"];
                $mail=$_POST["txtMail"];
                $idusuario=$_POST["txtId"];

                if ($mail<>"")
                {
                        // verificar si el mail existe (hasta cierto punto), verificamos que el dominio existe
                        $dominio = explode('@',$mail); // devuelve un array
                        if(!checkdnsrr($dominio[1])){
                                ?>
                                 <script type="text/javascript">
                                        alert("La dirección de correo no es válida. \n \n"+
                                                "No es posible registrarse. \n \n");
                                        history.back();
                                 </script>
                                <?php
                                exit();
                        }
                }

                if ($clave<>"") {

                        // escapar cadenas para que sea segura ponerla en query
                        $pass = mysql_escape_string($clave);
```

```
                // codificar la contraseña junto con la máscara añadida
                $salt = "|#€7`¬23ads4ook12";
                $pass = sha1($salt.md5($pass));
        }

        if ($mail=="" && $clave="")
        {
                ?>
                 <script type="text/javascript">
                        alert("No ha introducido ningún dato. \n \n");
                        history.back();
                 </script>
                <?php
                exit();
        }

        include("conec.php");
        $conexion=conectarse();

        if (!$conexion) {

                ?>
                        <script type="text/javascript">
                                alert("No ha sido posible realizar la conexión. \n \n"+
                                        "Hay problemas con la conexión, inténtelo en otro
momento \n                                      \n");
                                window.close();
                        </script>
                <?php

        }

        // establecer codificación
        mysql_query("SET NAMES 'utf8'");

        // Realizar consulta
        if ($mail<>"" && $clave<>""){
                $sql = "UPDATE usuario SET  email='".$mail."',pass='".$pass."' WHERE
`idUsuario`='".                         $idusuario."'";
        }
        else if ($mail <> ""){
                $sql = "UPDATE usuario SET  email='".$mail."' WHERE
`idUsuario`='".$idusuario."'";
        }
        else if ($clave <> ""){
                // recuperamos mail
                if ($mail=="") {
                        $sql="SELECT * FROM usuario WHERE idUsuario='".$idusuario."'";
                        $result= mysql_query($sql,$conexion);
```

```
                        while($row=mysql_fetch_array($result)){
                                $mail=$row["email"];
                                //echo "mail: ".$mail."<br>";

                        }
                }
                $sql = "UPDATE usuario SET  pass='".$pass."' WHERE
`idUsuario`='".$idusuario."'";
        }

        $result = mysql_query($sql,$conexion);

                // Comprobar resultado
                // Muestra la consulta enviada a MySQL, y el error. Uso para depuración.
                if (!$result) {
                        $message  = 'Consulta inválida: ' . mysql_error() . "\n";
                        $message .= 'Consulta enviada: ' . $sql;

                        echo $message;

                        ?>
                         <script type="text/javascript">
                                alert("No ha sido posible realizar la validacion. \n \n"+
                                        "Hay problemas con la conexión, inténtelo en otro
momento \n                                        \n");
                                history.back();
                         </script>
                        <?php

                        die($message);
                }
                else{
                        // Verificamos que existe servidor de correo en el hosting
                        $domain = "audiolibros.hol.es";
                        if (checkdnsrr($domain, "MX")) {
                                echo "Online host verification Test: PASSED <br>";
                                echo "Email Status: VALID <br>";

                                // preparar y enviar mensaje

                                $para=$mail;
                                $de='admin@audiolibros.hol.es';
                                $asunto='TU CUENTA HA SIDO ACTUALIZADA';
                                $mensaje="Los datos de registro son:\nUSUARIO:
".$usuario."\nCLAVE:                                        ".$clave."\n\n
http://www.audiolibros.hol.es";
                                $cabecera="FROM: ".$de;
                                //La direccion de correo desde donde se envió
                                $cabeceras = "From: admin@audiolibros.hol.es\n";
```

```
                                        //La direccion de correo a donde se responderá
                                        $cabeceras .= "Reply-To: admin@audiolibros.hol.es\n";


                                        mail($para, $asunto, $mensaje, $cabeceras);

                                        ?>
                                                <script type="text/javascript">
                                                alert("CAMBIO EFECTUADO
                                                        SATISFACTORIAMENTE \n \n"+
                                                        "Se ha enviado un mensaje a su dirección de
                                                                correo \n \n"+
                                                         "con los datos actualizados. \n \n");
                                                location.href='restringida.htm.php';
                                                </script>
                                        <?php

                                } else {
                                        echo "Online host verification Test: FAILED <br>";
                                        echo "Email Status: INVALID";

                                        ?>
                                                <script type="text/javascript">
                                                        alert("CAMBIO EFECTUADO
                                                                        SATISFACTORIAMENTE \n
\n");
                                                        location.href='restringida.htm.php';
                                                </script>
                                        <?php
                                }
                        }
                mysql_close($conexion);

        /**
         * @param string $valor
         * @return string string limpiado de fallos de seguridad
         */
        function limpiar($valor)
        {
                // quitar etiquetas a HTML y PHP
                $valor = strip_tags($valor);
                // Quita las barras de un string con comillas escapadas.
                $valor = stripslashes($valor);
                // esta función convierte todos los caracteres aplicables a entidades HTML.
                $valor = htmlentities($valor);
                return $valor;
        }

?>
```

logeo.php

```
<?php
        require ("funciones.php");

        seguridadIndex();

        $error = 0;

        if(isset($_POST['login']))
        {
          $recordarme=0;
          if(isset($_POST['recordarme']))
                        $recordarme=1;

                $error = login(limpiar($_POST['user']), $_POST['pass'],$recordarme);

          if($error>0)
          {
            header("Location: restringida.htm.php");
            exit();
          }

        }
?>

<html>

<head>

        <meta content="text/html;" http-equiv="content-type" charset="utf-8"/>

        <title>AUDIOLIBROS - Área Registrada </title>


        <style>

                BODY {BACKGROUND: url(fondo.png) #ffffff repeat-x; MARGIN: 0px  }
                TABLE.wrapper {WIDTH: 760px}
                TD {FONT-SIZE: 11px; FONT-FAMILY: Tahoma}
                FORM {MARGIN: 0px}
                INPUT {FONT-SIZE: 11px; FONT-FAMILY: Tahoma}
                .navbar {BACKGROUND: #8fbce9}
                .navbox {PADDING-RIGHT: 5px; PADDING-LEFT: 5px; PADDING-BOTTOM: 5px;
                                PADDING-TOP: 5px; BACKGROUND-COLOR: #eff2f9 }
                .navinput {BORDER-RIGHT: #8fbce9 1px solid; BORDER-TOP: #8fbce9 1px solid;
BORDER-                         LEFT: #8fbce9 1px solid; WIDTH: 100px; BORDER-BOTTOM:
#8fbce9 1px solid          }
```

```
                .submitbutton { BORDER-RIGHT: #8fbce9 1px solid; BORDER-TOP: #8fbce9 1px
solid;                          FONT-WEIGHT: bold; BORDER-LEFT: #8fbce9 1px solid; CURSOR:
hand; COLOR:                    #ffffff; BORDER-BOTTOM: #8fbce9 1px solid; BACKGROUND-
COLOR: #8fbce9}
                .fieldarea {BACKGROUND-COLOR: #eff2f9; TEXT-ALIGN: right}
                .pequeña{FONT-SIZE: 11px; FONT-FAMILY: Tahoma, arial}

        </style>

</head>

<body>

        <table align="center" cellpadding="0" cellspacing="0">
                <tr>
                        <td width="100%" height="122" style="background:url('fondo.png') repeat-
x;">
                                <img src="cab01.png" alt="cab">
                        </td>
                </tr>
        </table>

        <table width="100%" cellpadding="0" cellspacing="0" style="padding-left:35px; padding-
                        right:35px;border-bottom:1px solid #CCCCCC; background:#EFEFEF;">
                <tr>
                        <td height="30">
                                Usted est&aacute; aqu&iacute;: <b>Entrar </b>
                        </td>

                        <td style="text-align:right;">
                                <a href="index.html">Regresar </a>
                        </td>
                </tr>
        </table>

        <br>

        <TABLE align="center" cellSpacing=1 cellPadding=0 border=0>
                <TR>
                    <td valign="top" style=" padding-left:20px;padding-bottom:20px;">
                        <br><br>
                        <p>Acceso al área de usuarios registrados.</p>
                    </TD>
                </TR>
        </TABLE>

        <TABLE class=navbar align="center" cellSpacing=1 cellPadding=2 width="45%" border=0>

                <TR>
```

```
                <td width="100%" class=navbox valign="top" >
                    <form name="login" method="post" action="">
                        <table width="100%" cellpadding="3" >
                            <tr><td class="fieldarea"></td><td></td></tr>
                            <tr>
                            <td class="fieldarea"><label for="usuario">Usuario:
</label></td>
                            <td><input type="text" id="user" name="user" <?php
if($error>0) echo                                    'value="'.limpiar($_POST['user']).'"'; ?>/>
                                        <script>
                                            document.forms[0].user.focus();
                                        </script>
                            </td>
                            </tr>
                            <tr><td class="fieldarea"></td><td></td></tr>
                            <tr>
                                <td class="fieldarea"> <label for="usuario"> Clave:
                                                    </label></td>
                                <td><input type="password" id="pass" name="pass"
/></td>
                            </tr>
                            <tr><td class="fieldarea"></td><td></td></tr>
                            <tr>
                                <td class="fieldarea"><label
for="recordarme">Recordarme:
        </label></td>
                                <td><input type="checkbox" id="recordarme"
                                        name="recordarme" /></td>
                            </tr>
                            <tr><td class="fieldarea"></td><td></td></tr>
                            <tr>
                                <td colspan="2" align="center">
                                    <input type="submit" name="login" id="login"
                                            value="Login" /></td>
                            </tr>
                            <tr>
                                <td class="pequeña" colspan="2" align="center">
                                    <br>
                                    <a href="olvido.html"
                                    >¿Olvidé mi Contraseña?</a>
                                </td>
                            </tr>
                        </table>

                        <?php
                        switch ($error) {
                            case -1://login
                                //echo '<br/><strong>Usuario o clave incorrecta</strong>';
                                ?>
```

```
                                        <script type="text/javascript">
                                                alert("Usuario o clave incorrecta. \n \n");
                                        </script>
                                        <?php

                                        break;
                                case -2://registro
                                        echo '<br/><strong>Error al registrarse. Usuario ya
                                                        existente.</strong>';
                                        break;
                                case -3://registro
                                        //echo '<br/><strong>El usuario y la contraseña deben tener
como                                            mínimo 4 carácteres.</strong>';
                                        ?>
                                        <script type="text/javascript">
                                        alert("El usuario y la contraseña deben tener como mínimo
4                                               carácteres. \n \n");
                                        </script>
                                        <?php
                                        break;
                                default:
                                        break;
                        }
                        ?>

                </form>
            </td>
            </TR>

        </table>
        <br>

</body>
</html>
```

olvido.html

```
<html>

<head>
	<meta content="text/html;" http-equiv="content-type" charset="utf-8"/>
	<title>AUDIOLIBROS - Solicitar Contraseña</title>

	<style>

	BODY {BACKGROUND: url(fondo.png) #ffffff repeat-x; MARGIN: 0px}
	TABLE.wrapper {WIDTH: 760px}
	TD {FONT-SIZE: 11px; FONT-FAMILY: Tahoma}
	FORM {MARGIN: 0px}
	INPUT {FONT-SIZE: 11px; FONT-FAMILY: Tahoma}
	.navbar {BACKGROUND: #8fbce9}
	.navbox {PADDING-RIGHT: 5px; PADDING-LEFT: 5px; PADDING-BOTTOM: 5px;
PADDING-TOP:		5px; BACKGROUND-COLOR: #eff2f9}
	.navinput {BORDER-RIGHT: #8fbce9 1px solid; BORDER-TOP: #8fbce9 1px solid; BORDER-
LEFT:		#8fbce9 1px solid; WIDTH: 100px; BORDER-BOTTOM: #8fbce9 1px solid}
	.submitbutton {BORDER-RIGHT: #8fbce9 1px solid; BORDER-TOP: #8fbce9 1px solid; FONT-
			WEIGHT: bold; BORDER-LEFT: #8fbce9 1px solid; CURSOR: hand; COLOR:
#ffffff;			BORDER-BOTTOM: #8fbce9 1px solid; BACKGROUND-COLOR: #8fbce9}

	</style>

	<script type="text/javascript">

	function valida_olv(form1)
	{
		if (form1.txtUsuario.value==""){
			alert("USUARIO esta vacío...");
			form1.txtUsuario.focus();
			return;
		}
		if (form1.txtUsuario.value.length < 4){
			alert("USUARIO debe tener cuatro caracteres como mínimo...");
			form1.txtUsuario.focus();
			return;
		}

		form1.submit();
		return;
	}

	</script>

</head>

<body>
```

```
<table align="center" cellpadding="0" cellspacing="0">
        <tr>
                <td width="100%" height="122" style="background:url('fondo.png') repeat-
x;">
                        <img src="cab01.png" alt="cab">
                </td>
        </tr>
</table>

<table width="100%" cellpadding="0" cellspacing="0" style="padding-left:35px; padding-
                right:35px;border-bottom:1px solid #CCCCCC; background:#EFEFEF;">
        <tr>
        <td height="30">
                Usted est&aacute; aqu&iacute;: <b>Solicitar Contraseña</b>
        </td>
        <td style="text-align:right;">
                <a href="index.html">Regresar </a>
        </td>
        </tr>
</table>

<br>

<TABLE align="center" cellSpacing=1 cellPadding=0 border=0>
        <TR>
            <td valign="top" style=" padding-left:20px;padding-bottom:20px;">
                        <br><br>
                        <p>Solicitud de envío de contraseña.</p>
                </TD>
        </TR>
</TABLE>

<TABLE class=navbar align="center" cellSpacing=1 cellPadding=2 width="45%" border=0>
  <TBODY>
    <TR>
        <TD  width="100%" class=navbox valign="top">
                <FORM id="form1" name="form1" action=olvido.php method=post>
                        <table width="100%" cellpadding="3" >

                                <BR><BR>
                                <TABLE cellSpacing=0 cellPadding=1 border=0 align="center">
                                <TBODY>
                                        <TR>
                                        <TD align=right>
                                                <SPAN class=smalltext>Usuario:</SPAN>
                                        </TD>
                                        <TD>
```

```
                                                <INPUT class=navinput name="txtUsuario"
                                                        type="text" size="18"
maxlength="15">
                                                <script>
                                                        document.forms[0].txtUsuario.focus();
                                                </script>
                                        </TD>
                                        </TR>
                                        <tr><td> <br></td><td></td></tr>
                                        <tr>
                                        <td colspan="2" align="center">
                                                <input type="button" name="Button"
value="Enviar"
        Onclick="javascript:valida_olv(form1)"  />
                                        </td>
                                        </tr>

                                </TBODY>
                                </TABLE>
                                <br>
                        </table>
                </FORM>
        </TD>
        </TR>
        </TBODY>
        </TABLE>

</body>
</html>
```

olvido.php

```
<html>
<head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />

        <script type="text/javascript">

        function envio()
        {
                window.alert("Se ha enviado un mensaje a su dirección de correo."+"\n \n"+
                        "con tu clave de acceso. Por favor verifícalo ... "+"\n");
                return;
        }

        function error()
        {
                window.alert("\nUSUARIO NO EXISTE O INCORRECTO\n");
                return;
```

```
        }

        function error2()
        {
                window.alert("\nNo se puede enviar un mensaje en este monento\n\n"+
                        "Inténtelo más tarde. \n" );
                return;
        }


        </script>


</head>

<body bgcolor="#eff2f9">

        <?php
        if($_POST){

                include("conec.php");
                $link=conectarse();

                // asignar variable
                $usuario=$_POST["txtUsuario"];

                // codificar la contraseña junto con la máscara añadida
                // dado que no podemos desencriptarla, asignamos una nueva
                // contraseña generica y actualizamos la BD
                // número aleatorio entre 1 y 9000, incluyendo estos dos valores entre los posibles.
                $num_aleatorio = rand(1,9000);
                $pass = "pass45_".$num_aleatorio;
                $salt = "|#€7`¬23ads4ook12";
                $cpass = sha1($salt.md5($pass));

                // Realizar consultas, primero actualizamos
                $sql = "UPDATE usuario SET  pass='".$cpass."' WHERE login='$usuario';";
                $result= mysql_query($sql,$link);

                // Recuperamos registro
                $sql="SELECT * FROM usuario WHERE login='$usuario';";
                $result= mysql_query($sql,$link);

                 //numero de filas retornadas. Si 0 no existe, 1 existe
                $num_rows = mysql_num_rows($result);

                 //echo $result."\n";echo $num_rows;

                while($row=mysql_fetch_array($result)){
```

```
                $para=$row["email"];
                //$clave=$row["pass"];
        }

        // si filas devueltas es 0 no existe usuario
        if ($num_rows == 0) {
                $message  = 'Invalid query: ' . mysql_error() . "\n";
                $message .= 'Whole query: ' . $sql;
                ?>
                        <script>
                                error();
                                history.back();
                                document.forms[0].txtUsuario.focus();
                        </script>
                <?php

                die($message);
        }
        else {
        // comprobamos el servidor de correo
                $domain = "audiolibros.hol.es";
                if (checkdnsrr($domain, "MX")) {
                        echo "Online host verification Test: PASSED <br>";
                        echo "Email Status: VALID";

                        // componemos el mensaje
                        $de='admin@audiolibros.hol.es';
                        $asunto='TU CLAVE OLVIDADA';
                        $mensaje="Los datos de registro son:\n USUARIO: ".$usuario."\n CLAVE:
".$pass.
                                "\n\t Esta contraseña se ha generado de forma aleatoria, debe
cambiarla.".
                                "\n\n http://www.audiolibros.hol.es";
                        $cabecera="FROM: ".$de;
                        //La direccion de correo desde donde supuestamente se envió
                        $cabeceras = "From: admin@audiolibros.hol.es\n";
                        //La direccion de correo a donde se responderá
                        $cabeceras .= "Reply-To: admin@audiolibros.hol.es\n";

                        // enviamos el correo generado
                        mail($para, $asunto, $mensaje, $cabeceras);

                        ?>      <script>
                                        envio();
                                        location.href='index.html';
                                </script>
                        <?php

                        } else {
```

```
                                        echo "Online host verification Test: FAILED <br>";
                                        echo "Email Status: INVALID";

                                        ?>      <script>
                                                        error2();
                                                        location.href='index.html';
                                                </script>
                                        <?php
                                }

                        }
                }
                ?>

</body>

</html>
```

registro.html

```
<html>

<head>
        <meta content="text/html;" http-equiv="content-type" charset="utf-8"/>
        <title>AUDIOLIBROS - Solicitud de Registro </title>

        <style>

        BODY {BACKGROUND: url(fondo.png) #ffffff repeat-x; MARGIN: 0px  }
        TABLE.wrapper {WIDTH: 760px}
        TD {FONT-SIZE: 11px; FONT-FAMILY: Tahoma}
        FORM {MARGIN: 0px}
        INPUT {FONT-SIZE: 11px; FONT-FAMILY: Tahoma}
        .navbar {BACKGROUND: #8fbce9}
        .navbox {PADDING-RIGHT: 5px; PADDING-LEFT: 5px; PADDING-BOTTOM: 5px;
PADDING-TOP:                    5px; BACKGROUND-COLOR: #eff2f9}
        .navinput {BORDER-RIGHT: #8fbce9 1px solid; BORDER-TOP: #8fbce9 1px solid; BORDER-
LEFT:           #8fbce9 1px solid; WIDTH: 100px; BORDER-BOTTOM: #8fbce9 1px solid}
        .submitbutton { BORDER-RIGHT: #8fbce9 1px solid; BORDER-TOP: #8fbce9 1px solid;
FONT-                   WEIGHT: bold; BORDER-LEFT: #8fbce9 1px solid; CURSOR: hand; COLOR:
#ffffff;                BORDER-BOTTOM: #8fbce9 1px solid; BACKGROUND-COLOR: #8fbce9}
        .fieldarea {BACKGROUND-COLOR: #eff2f9; TEXT-ALIGN: right}

        </style>

        <script type="text/javascript">

        function valida(form1)
```

```
{
        if(form1.txtMail.value.length==0){
                alert("E-MAIL es obligatorio...");
                form1.txtMail.focus();
                return;
        }
        else {
                valor = document.getElementById("txtMail").value;
                expr = /^([a-zA-Z0-9_\.\-])+\@(([a-zA-Z0-9\-])+\.)+([a-zA-Z0-9]{2,4})+$/;

                if( !(expr.test(valor)) ) {
                        alert("Error: La dirección de correo: \n\n"+valor+"\n\nes incorrecta.");
                        form1.txtMail.focus();
                        return;
                }
                else {
                        // validar si existe
                        //alert("Correo correcto");
                }
        }

        if(form1.txtUsuario.value==""){
                alert("USUARIO es obligatorio...");
                form1.txtUsuario.focus();
                return;
        }
        if(form1.txtUsuario.value.length < 4){
                        alert("USUARIO debe tener cuatro caracteres como mínimo...");
                        form1.txtUsuario.focus();
                        return;
        }

        if(form1.txtClave.value==""){
                alert("Contraseña es obligatoria...");
                form1.txtClave.focus();
                return;
        }
        if(form1.txtClave.value.length < 4){
                alert("CONTRASEÑA debe tener cuatro caracteres como mínimo...");
                form1.txtClave.focus();
                return;
        }

        if(form1.txtClave.value!=form1.txtClave2.value){
                alert("Contraseña no coincide...");
                form1.txtClave.value="";
                form1.txtClave2.value="";
                form1.txtClave.focus();
                return;
```

```
                }

                form1.txtUsuario.value=form1.txtUsuario.value.toLowerCase();
                form1.txtMail.value=form1.txtMail.value.toLowerCase();
                form1.submit();

                return;

        }

        </script>

</head>

<body>

        <table align="center" cellpadding="0" cellspacing="0">
                <tr>
                        <td width="100%" height="122" style="background:url('fondo.png') repeat-
x;">
                                <img src="cab01.png" alt="cab">
                        </td>
                </tr>
        </table>

        <table width="100%" cellpadding="0" cellspacing="0" style="padding-left:35px; padding-
                        right:35px;border-bottom:1px solid #CCCCCC; background:#EFEFEF;">
                <tr>
                        <td height="30">
                                Usted est&aacute; aqu&iacute;: <b>Registrar </b>
                        </td>

                        <td style="text-align:right;"><a href="index.html">Regresar </a>
                        </td>
                </tr>
        </table>

        <br>

        <TABLE align="center" cellSpacing=1 cellPadding=0 border=0>
        <TBODY>
                <TR>
                        <td valign="top" style=" padding-left:20px;padding-bottom:20px;">
                                <br><br>
                                <p>Por favor rellene los campos para registrar una nueva cuenta. Los
campos                                  marcados con un * son obligatorios.</p>
                        </TD>
                </TR>
```

```
                <TABLE class=navbar align="center" cellSpacing=1 cellPadding=2 width="45%"
border=0>
                        <TR>
                         <td width="100%" class=navbox valign="top" align="center" >
                                <form id="form1" name="form1"  method="post" action="registro.php"
>
                                 <table cellspacing="0" cellpadding="0" >
                                        <tr><td>
                                        <table width="100%" cellpadding="3">
                                                <tr><td class="fieldarea">Direcci&oacute;n
                                                        Email:</td><td><input type="text"
name="txtMail"                                                  id="txtMail" size=40
value=""> *</td></tr>
                                                <script>
                                                        document.forms[0].txtMail.focus();
                                                </script>
                                                <tr><td class="fieldarea"></td><td></td></tr>
                                                <tr><td class="fieldarea">Usuario:</td><td><input
                                                        name="txtUsuario" id="txtUsuario"
size="18"                                               maxlength="15" type="text">
(max. 15 caracteres)                                    *</td></tr>
                                                <tr><td
class="fieldarea">Contrase&ntilde;a:</td><td><input
        name="txtClave" id="txtClave" size="18"
                maxlength="15" type="password">
                                                (mín. 4 caracteres) *</td></tr>
                                                <tr><td class="fieldarea">></td><td><input
name="txtClave2"
        class="formulario_peque&ntilde;o" id="txtClave2"
                size="18" maxlength="15" type="password">
                                                reescribe tu contrase&ntilde;a *</td></tr>
                                                <tr><td class="fieldarea">></td><td></td></tr>
                                                <tr>
                                                <td colspan="2" align="center">
                                                <input type="button" name="Button"
value="Enviar"                                          onclick="javascript:valida(form1)" />
                                                </td>
                                                </tr>

                                        </table>
                                         </td></tr>
                                 </table>
                                </form>
                        </td>
                        </TR>
                </table>
                </TBODY>
        </TABLE>
        <br>
```

```
</body>

</html>
```

registro.php

```
<?php

        // asignar variables
        $usuario=limpiar($_POST['txtUsuario']);
        $clave=$_POST["txtClave"];
        $mail=$_POST["txtMail"];
        //$administrador=0;
        //$vacio="";

        // verificar si el mail existe (hasta cierto punto), verificamos que el dominio existe
        $dominio = explode('@',$mail); // devuelve un array
        if(!checkdnsrr($dominio[1])){
                ?>
                        <script type="text/javascript">
                                alert("La dirección de correo no es válida. \n \n"+
                                        "No es posible registrarse. \n \n");
                                history.back();
                        </script>
                <?php
                exit();
        }


        include("conec.php");
        $conexion=conectarse();


        if (!$conexion) {

                ?>
                        <script type="text/javascript">
                                alert("No ha sido posible realizar la conexión. \n \n"+
                                        "Hay problemas con la conexión, inténtelo en otro
momento \n                              \n");
                                window.close();
                        </script>
                <?php

        }
```

```
// escapar cadenas para que sea segura ponerla en query
$user = mysql_escape_string($usuario);
$pass = mysql_escape_string($clave);

// codificar la contraseña junto con la máscara añadida
$salt = "|#€7`¬23ads4ook12";
$pass = sha1($salt.md5($pass));

// establecer codificación
mysql_query("SET NAMES 'utf8'");

// Realizar consulta
$sql = "SELECT idUsuario FROM usuario WHERE UPPER(login)='".strtoupper($user)."'";
$result = mysql_query($sql,$conexion);

//numero de filas retornadas. Si 0 no existe, 1 existe
$num_rows = mysql_num_rows($result);

if ($num_rows == 0) {

        //echo "no existe";

        // Añadir nuevo usuario
        $sql = "INSERT INTO usuario (login,pass,email) VALUES ('".$user."','".$pass."','".
                        $mail."')";
        $result = mysql_query($sql,$conexion);

        // Comprobar resultado
        // Muestra la consulta enviada a MySQL, y el error. Uso para depuración.
        if (!$result) {
                $message  = 'Consulta inválida: ' . mysql_error() . "\n";
                $message .= 'Consulta enviada: ' . $sql;

                echo $message;

                ?>
                 <script type="text/javascript">
                        alert("No ha sido posible realizar la validacion. \n \n"+
                                "Hay problemas con la conexión, inténtelo en otro
momento \n                                \n");
                        history.back();
                 </script>
                <?php

                die($message);
        }
        else{
                // Verificamos que existe servidor de correo en el hosting
                $domain = "audiolibros.hol.es";
```

```
                    if (checkdnsrr($domain, "MX")) {
                        echo "Online host verification Test: PASSED <br>";
                        echo "Email Status: VALID";

                        // preparar y enviar mensaje

                        $para=$mail;
                        $de='admin@audiolibros.hol.es';
                        $asunto='TU CUENTA HA SIDO ACTIVADA';
                        $mensaje="Los datos de registro son:\nUSUARIO:
".$usuario."\nCLAVE:                                    ".$clave. "\n\n
http://www.audiolibros.hol.es";

                            $cabecera="FROM: ".$de;
                        //La direccion de correo desde donde supuestamente se envió
                        $cabeceras = "From: admin@audiolibros.hol.es\n";
                        //La direccion de correo a donde se responderá
                        $cabeceras .= "Reply-To: admin@audiolibros.hol.es\n";

                        mail($para, $asunto, $mensaje, $cabeceras);

                        ?>
                        <script type="text/javascript">
                            alert("  ¡ ¡ ¡ ¡   BIENVENIDO   ! ! ! !   \n \n"+
                             "REGISTRO EFECTUADO SATISFACTORIAMENTE
\n \n"+

                             "Se ha enviado un mensaje a su dirección de correo \n \n"+
                             "con los datos de registro. \n \n"+
                            "Ya puede acceder al área registrada \n \n");
                            location.href='logeo.php';
                        </script>
                        <?php

                    } else {
                        echo "Online host verification Test: FAILED <br>";
                        echo "Email Status: INVALID";

                        ?>
                        <script type="text/javascript">
                            alert("  ¡ ¡ ¡ ¡   BIENVENIDO   ! ! ! !   \n \n"+
                             "REGISTRO EFECTUADO SATISFACTORIAMENTE
\n \n                         "Ya puede acceder al área registrada \n \n");
                            location.href='logeo.php';
                        </script>
                        <?php
                    }

                }
            }
            else{
```

```
                ?>
                <script type="text/javascript">
                        alert("LO SENTIMOS... ESE NOMBRE DE USUARIO YA EXISTE \n \n"+
                                "ELIGE OTRO NOMBRE DE USUARIO \n \n");
                        history.back();
                </script>
                ?php

        }


/**
 *
 * @param string $valor
 * @return string string limpiado de fallos de seguridad
 */
function limpiar($valor)
{
        // quitar etiquetas a HTML y PHP
        $valor = strip_tags($valor);
        // Quita las barras de un string con comillas escapadas.
        $valor = stripslashes($valor);
        // esta función convierte todos los caracteres aplicables a entidades HTML.
        $valor = htmlentities($valor);
        return $valor;
}

?>
```

restringida.htm.php

```
<?php

	require ("funciones.php");
	$error = 0;

	seguridad(); //comprobamos que se esté logueado

	// Cerrar sesion y eliminar cookie
	if(isset($_POST['salir']))
	{
		// eliminar cookies en BD
		destruirCookie($_COOKIE['identificado']);

		$_SESSION = array();

		//guardar el nombre de la sesión para luego borrar las cookies
		$session_name = session_name();

		//Para destruir una variable en específico
		unset($_SESSION['usuario']);

		// Finalmente, destruye la sesión
		session_destroy();

		// Para borrar las cookies asociadas a la sesión
		// Es necesario hacer una petición http para que el navegador las elimine
		if ( isset( $_COOKIE[ $session_name ] ) ) {
			setcookie($session_name, '', time()-3600, '/');
		}
		if(isset($_COOKIE['identificado'])){
			setcookie('identificado', '', time()-3600, '/');
		}
		if(isset($_COOKIE['iduser'])){
			setcookie('iduser', '', time()-3600, '/');
		}
		// petición http
		header("Location: index.html");
		exit();

	}

?>


<html>

<head>
```

```
<meta content="text/html;" http-equiv="content-type" charset="utf-8"/>
<link href="style.css" rel="stylesheet" type="text/css" /><title>Audiolibros en Español</title>

<script type="text/javascript">

        function AbrirCentrado(Url,NombreVentana,width,height,extras) {
                var largo = width;
                var altura = height;
                var adicionales= extras;
                var top = (screen.height-altura)/2;
                var izquierda = (screen.width-largo)/2;
                nuevaVentana=window.open("+ Url + ","+ NombreVentana + ",'width=' + largo +
',height=' +                    altura + ',top=' + top + ',left=' + izquierda + ',features=' + adicionales
+ ");
                nuevaVentana.focus();
        }

        function muestra(id){
                var elDiv = document.getElementById(id); //se define la variable "elDiv" igual a
nuestro div
                elDiv.style.display='block';          //damos un atributo display:block que  el div
        }

        function oculta(id){
                 var elDiv = document.getElementById(id); //se define la variable "elDiv" igual a
nuestro div
                 elDiv.style.display='none';          //damos un atributo display:none que oculta el div

        }

        function cerrar_sesion()
        {
                alert("GRACIAS, ".$nickUsualioL+"\n \n"+
                        "por utilizar nuestros servicios"+"\n \n");
                return;
        }

</script>

<style>

        .pequeña{FONT-SIZE: 11px; FONT-FAMILY: Tahoma, arial}
        .rotulo1 {position: absolute; z-index: 2; left: 245px; top: 20px}
        .salir {position: absolute; z-index: 2; left: 45px; top: 100px}
        .login1 {position: absolute; z-index: 2; text-align: center; left: 20px; top: 140px; width:
125px}
        .fieldarea {FONT-SIZE: 12px; FONT-FAMILY:  tahoma, arial;TEXT-ALIGN: center;
                        color: rgb(156, 31, 156)}
```

```
                .misdatos1 {display:none; position: absolute; left: 253px; top: 380px;
                        background:url(images/leftbg.gif) repeat-x;  width:745px; padding:10px
10px 10px                20px;}
        </style>

</head>

<body>

        <div id="container">

                <div id="header">

                        <ul>
                                <li id="navtop"><br />
                                </li>
                                <li><a class="link" href="datos.html.php" >Mis datos</a></li>
                                <li id="navbottom"><br />
                                </li>
                        </ul>

                        <div id="salir" class="salir" align="center">
                                <form name="login" method="post" action="">
                                        <table>
                                        <tr>
                                                <td align="right">
                                                        <input type="submit" name="salir" id="salir"
value="                                                         Cerrar y salir " />
                                                </td>
                                        </tr>
                                        </table>

                                </form>
                        </div>

                        <div id="rotulo" class="rotulo1">
                        <p  style= "text-align: center;  font-family: Comic Sans MS,cursive; font-size:
40px;                   color: rgb(156, 31, 156);">
                                <strong>Audiolibros en español</strong>
                        </p>
                        </div>

                        <div id="log" class="login1">
                                <p  class="fieldarea" align="center">
                                Hola,
                                <strong>
                                <?php
                                                if(isset($_COOKIE['iduser']))
                                        echo $_COOKIE['iduser'];
```

```
                        ?>
                        </strong>
                        </p>
                    </div>

                </div>

                <div class="inner_copy" />
                </div>
                <div id="content">
                    <div id="left">
                        <h3 align=center>DESCARGAR MP3</h3>
                        <p>Audiolibros en español.</p>
                        <p><br /></p>
                        <p>Más de 3200 títulos disponibles.</p>
                        <p><br /></p>
                        <p>Descarga de forma gratuita tus audiolibros favoritos.</p>
                        <p><br /></p>
                        <p><br /></p>
                        <p><br /><br /><br /><br /><br /><br /><br /></p>
                        <p align=center></p>
                        <a href="descargas.html" TARGET="_blank">
                        <img id="djset" src="images/mp3.jpg" alt="Descargar mp3" />
                        </a>
                    </div>
                    <div id="middle">
                        <h3 align=center>DESCARGAR App</h3>
                        <p>Con nuestra App para android disfruta escuchando on-line, en
cualquier                                   lugar, tus títulos favoritos.</p>
                        <br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br /><br
/><br /><br                                 /><br />
                        <a href="Audiolibros_v1.0.zip"><img id="photos" src="images/app.jpg"
                                alt="Descargar Apps" /></a>

                    </div>

                    <div id="right2">
                        <h3 align=center>REPRODUCIR AUDIOLIBROS</h3>
                        <p>Selecciona el título y reproduce on-line, nuestro catálogo de
audiolibros en                              español.</p>
                        <br /><br />
                        <a href="mp3.html" TARGET="_blank"><img id="photos2"
                                src="images/audio.jpg" alt="Reproducir mp3" /></a>
                    </div>

                    <div class="clear" >
                    </div>

                    <div id="footer">
```

```
                        <p class="lf" >&copy; Copyright 2013 AmmSoft.</a></p>

                        <div class="clr"></div>
                        </div>

                        <div class="fclear" >
                        </div>
                    </div>

                </div>

                <div id="misdatos" class="misdatos1" >

                    <h3 align=center>Audiolibros.hol.es</h3>
                    <p>  Disponemos de un amplio catálogo compuesto por más de 3000 títulos, con
la más                  variada temática.</p>
                    <p><br /></p>
                    <p>Los audiolibros son grabaciones de los contenidos de un libro, leídos en voz
alta por                    un narrador.
                            Las grabaciones se presentan en formato digital a través de descargas o
                            reproducirlos on-line.</p>
                    <p><br /></p>
                    <p>Los usuarios registrados podrán acceder a la descarga o reproducción de los
                            audiolibros a partir del
                            menú de acceso a este servicio. El sistema les pedirá un login y un
password.
                            El servicio dará acceso al catálogo de audiolibros. </p>
                    <p><br /></p>
                    <p>Además si te registras, descarga nuestra App y escucha tus audiolibros
favoritos                   desde tu android.</p>
                    <p><br /></p>
                    <table width="95%">
                        <tr>
                            <td align="center">
                                <img src="images/lector.jpg" alt="audiolibro" />
                            </td>
                        </tr>
                        <tr>
                            <td align="center">
                                <a href="" onClick="oculta('acercade');">Volver</a>
                            </td>
                        </tr>
                    </table>
                </div>

</body>

</html>
```

style.css

```
* {margin:0;padding:0}
body {background:#020104 url(images/bodybg.gif) repeat-x;color:#7F7F7F;font-family:tahoma,
arial;font-size:10px;text-align:center}
.fleft {float:left}
.fright {float:right}
.fclear {clear:both}.pequeña{FONT-SIZE: 11px; FONT-FAMILY: Tahoma, arial          }
#header ul {margin:20px 25px 25px
20px;padding:0;display:block;color:#c5b16b;width:auto;width:110px;position:absolute;background:#f00
}
#header li {background:#cebc7d;width:136px;text-align:left;padding:0 0 1px 10px;list-style-type:none}
.link,#last {font-size:14px;font-weight:900;text-decoration:none;padding:0;margin:0;text-
transform:uppercase;width:120px;display:block;color:#7a6b3e;font-size:14px;font-
weight:bold;position:relative;text-align:center}
.inner_copy {border:0;color:#f00;float:left;width:50% !important;margin:-100% 0 0
0;overflow:hidden;line-height:0;padding:0;font-size:11px}
.link:hover, #last:hover {color:#4c5b38}
.link {border-bottom:1px dashed #9c8f5e}
#last {border-bottom:0 none #9c8f5e}
#navtop {background:url(images/navtop.gif);background-repeat:no-
repeat;display:block;width:146px;height:20px}
#navbottom {background:url(images/navbottom.gif);display:block;width:146px;height:19px;background-
repeat:no-repeat;background-position:left bottom}
p a:link {color:#7a6b3e;font-size:14px;font-weight:bold;text-decoration:underline}
p a:visited {color:#7a6b3e;font-size:14px;font-weight:bold;text-decoration:underline}
p a:hover, a:active {color:#7a6b3e;font-size:14px;font-weight:bold;text-decoration:none}
p {font-family:Arial,Helvetica,sans-serif;font-size:14px;font-style:normal;font-
weight:normal;color:#5e573b}
#container {margin:30px auto;text-align:left;width:785px;display:block;position:relative}
#header
{width:785px;height:351px;margin:0;display:block;background:url(images/header.jpg);background-
repeat:no-repeat;position:relative}
#content {background:url(images/contentbg.gif) repeat-y;width:785px}
h3 {font-family:Arial,Helvetica,sans-serif;font-size:14px;font-weight:bolder;font-variant:normal;text-
transform:uppercase;color:#4c5b38;text-decoration:none;margin-bottom:20px}
#left {position:relative;margin-left:5px;float:left;padding:10px 10px 10px
20px;display:inline;background:url(images/leftbg.gif) repeat-x;width:250px}
#djset {position:absolute;bottom:0;right:0;border:0}
#middle {padding:10px 10px 10px
20px;position:relative;float:left;display:inline;background:url(images/middlebg.gif) repeat-
x;width:200px}
#photos {position:absolute;bottom:0;left:0;border:0 none
#fff}#photos2{position:relative;bottom:0;left:42;border:0 none #fff}
#right {float:left;display:inline;background:url(images/rightbg.gif) repeat-x;width:265px}
#right p {padding:10px 10px 10px 20px;font-family:Arial,Helvetica,sans-serif;font-size:14px;font-
style:normal;font-weight:normal;color:#e5d7a7}
#right p a {font-family:Arial,Helvetica,sans-serif;font-size:14px;font-style:normal;font-
weight:normal;color:#e5d7a7}
```

```
#right img {border:0 none #fff}
#right2 {padding:10px 10px 10px
20px;position:relative;float:left;display:inline;background:url(images/rightbg.gif) repeat-x;width:235px}
.clear {clear:both}
#footer {margin-left:5px;display:block;background:url(images/footerbg.gif) repeat-
x;width:775px;height:67px}
#footer div {padding:0 15px}
#footer p {font-family:Arial,Helvetica,sans-serif;font-size:12px;font-weight:normal;font-
variant:normal;text-transform:none;color:#f1e6c0;text-decoration:none;text-align:center;padding-
top:20px}
ul.fmenu { margin:0; padding:8px 0; list-style:none; float:right; width:auto;}
ul.fmenu li { margin:0; padding:0 12px; float:left;}
ul.fmenu li a { color:#5f5f5f; text-decoration:none; font-weight:bold;}
ul.fmenu li a:hover, ul.fmenu li.active a { color:#9c8f5e;}
ul.fmenu li a:hover { text-decoration:underline;}
.footer p.lf { margin:0; padding:0; float:right; width:auto; line-height:1.5em;}
.clr { clear:both; padding:0; margin:0; width:100%; font-size:0px; line-height:0px;}
```

ANEXO B: Código fuente Aplicación "AudioBooks" Android

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
   package="es.hol.audiolibros"
   android:versionCode="1"
   android:versionName="1.0" >

   <uses-sdk
      android:minSdkVersion="8"
      android:targetSdkVersion="16" />

   <uses-permission android:name="android.permission.INTERNET"/>
   <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
   <uses-permission android:name="android.permission.VIBRATE"/>
   <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

   <application
      android:allowBackup="true"
      android:icon="@drawable/icono_app"
      android:label="@string/app_name"
      android:theme="@style/AppTheme" >

      <!--  android:theme="@style/AppTheme"> -->
      <!-- android:theme="@android:style/Theme.Holo" -->

      <activity
         android:name="es.hol.audiolibros.SplashScreenActivity"
         android:label="@string/app_name" >
         <intent-filter>
            <action android:name="android.intent.action.MAIN" />

            <category android:name="android.intent.category.LAUNCHER" />
         </intent-filter>
      </activity>

      <activity
         android:name=".Principal"
         android:label="@string/app_name" />

      <activity
         android:name=".MusicPlayerActivity" />

      <activity
         android:name=".Configuracion" />

       <activity
         android:name=".Registro"
         android:theme="@android:style/Theme.Dialog" />

      <activity
```

```
        android:name=".Favoritos" />

    <activity
        android:name=".Libroviewer"
        android:theme="@android:style/Theme.Dialog"/>

    <activity
        android:name=".Catalogo" />

    <activity
        android:name=".AcercaDe"
        android:theme="@android:style/Theme.Dialog"/>

  </application>

</manifest>
```

SplashScreenActivity.java

```
package es.hol.audiolibros;

import java.util.Timer;
import java.util.TimerTask;
import android.app.Activity;
import android.content.Intent;
import android.content.pm.ActivityInfo;
import android.os.Bundle;
import android.os.Handler;
import android.view.Window;

public class SplashScreenActivity extends Activity {

    // establecer la duracion de la presentacion
    private static final long SPLASH_SCREEN_DELAY = 3000;

    private static final int TIME = 4 * 1000;// 4 seconds

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // En primer lugar configuramos la pantalla en vertical y a pantalla completa
        // (ocultando la barra de título o la action bar dependiendo de la versión de Android que estemos usando)
        // utilizando los métodos setRequestedOrientation(int) y requestWindowFeature(int).

        // Fijar orientacion vertical
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

        // Ocultar barra de titulo
        requestWindowFeature(Window.FEATURE_NO_TITLE);

        setContentView(R.layout.splash_screen);
```

```
/*
    // A continuación definimos una cuenta atrás usando las clases Timer y TimerTask,
    // mientras se está ejecutando esta cuenta atrás se mostrará el splash screen,
    // una vez se haya cumplido la cuenta atrás se lanzará la actividad principal y
    // el splash screen dejará de ser visible.

    // Tarea a realizar cuando finalice el temporizador
    TimerTask task = new TimerTask() {
        @Override
        public void run() {

            // Lanzar la nueva actividad
            Intent mainIntent = new Intent().setClass(
                    SplashScreenActivity.this, Principal.class);
            startActivity(mainIntent);

            // Finalizar la activity
            finish();
        }
    };

    // Crear el temporizador
    Timer timer = new Timer();

    // Iniciar temporizador
    timer.schedule(task, SPLASH_SCREEN_DELAY);

*/

    new Handler().postDelayed(new Runnable() {

                @Override
                public void run() {

                        Intent intent = new Intent(SplashScreenActivity.this,Principal.class);
                        startActivity(intent);

                        SplashScreenActivity.this.finish();

                        // Manejar transicions animadas entre actividades
                        overridePendingTransition(R.anim.fade_in, R.anim.fade_out);

                }
            }, TIME);

            new Handler().postDelayed(new Runnable() {
                @Override
                public void run() {
                    }
                }, TIME);


}
```

```
    @Override
        public void onBackPressed() {
            this.finish();
            super.onBackPressed();
        }

}
```

Principal.java

```
package es.hol.audiolibros;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.concurrent.ExecutionException;
import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import es.hol.audiolibros.Registro.asyncreg;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.ContentValues;
import android.content.Context;
import android.content.ContextWrapper;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.ActivityInfo;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.Bitmap;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.SystemClock;
import android.preference.PreferenceManager;
import android.util.Base64;
import android.util.Log;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.Toast;
import android.graphics.BitmapFactory;


public class Principal extends Activity {

        ImageView image, image1;
        Button imgbtn1, imgbtn2, imgbtn3, imgbtn4;
        Bitmap bitmap;

        private static final String NOM_BDD = "dbfavoritos";
        private DataBase  maBaseSQLite;
         private SQLiteDatabase db;
```

```
 private SharedPreferences prefs;
private String server="", url = "";
private boolean catalogo = false;
private int tamano = 0;
private boolean registrado = false;
private boolean actualiza = false;



@Override
  public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    // fijar de manera permanente la orientación en modo vertical
    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

    setContentView(R.layout.principal);


    //Obtiene el objeto de preferencias de la aplicacion
    //SharedPreferences sharedPreferences = getApplicationContext().getSharedPreferences("settings", 0);
    prefs = PreferenceManager.getDefaultSharedPreferences(Principal.this);

    // Obtiene valores almacenados en las preferencias.
    // El segundo parametro indica el valor a devolver si no lo encuentra, en este caso, falso.
    registrado = prefs.getBoolean("REGISTRO", false);
    catalogo   = prefs.getBoolean("CATALOGO", false);
    tamano     = prefs.getInt("TAMANO", 0);
                 server     = prefs.getString("server", "www.audiolibros.hol.es");


                 Log.i("Principal"," espace: "+Utils.getInternalAvailableSpace());

                 // Crear estructura de directorios
                 ContextWrapper context=this;
                 File myDir = context.getFilesDir();

            String carpeta = "Images_Audio";
            File Carpeta = new File(myDir, carpeta);
            Carpeta.mkdirs();

            carpeta = "Cat_Audio";
            Carpeta = new File(myDir, carpeta);
            Carpeta.mkdirs();


    // variables de uso general
    String mensa = "";
    boolean errores = false;

    // comprobar registro
    if(!registrado)
```

```
        {
           //Codigo que queramos que se ejecute mientras no hay datos de registro

              // crear mensaje y actualizar error
               mensa =  "\nNo existen datos de registro.";
               errores = true;

        }

        // comprobar conexion
        if(!Utils.verificaConexion(this)){

              // crear mensaje y actualizar error
              mensa = mensa + "\nNo hay conexión.";
              errores = true;
        }


        // mostrar aviso por errores
        if (errores){

              mensa = mensa +"\n\nAlgunas opciones no serán operativas.";
              // Mostrar dialogo
              Utils.showAlertDialog(Principal.this, "AVISO", mensa, false);

        }


       // Referenciar objetos
        imgbtn1 = (Button) findViewById(R.id.imgbtn1);
        imgbtn2 = (Button) findViewById(R.id.imgbtn2);
        imgbtn3 = (Button) findViewById(R.id.imgbtn3);
        imgbtn4 = (Button) findViewById(R.id.imgbtn4);


    } // onCreate



    /* Métodos que controlan la pulsación de los botones
     * favoritos()
     * catalogo() incluye actualizar() y cargar()
     * registro()
     * config()
     */

   // ************************** F A V O R I T O S ****************************************

    public void favoritos(View v)
    {
               // obtener si hay favoritos en la BD
               DataBase db = new DataBase(this);
               int datos = db.getLibrosCount();
              db.close();
```

```
            // Si hay los mostramos en una nueva activity o mostrar mensaje
            if (datos > 0){
                        Intent intent = new Intent(Principal.this,Favoritos.class);
                                    startActivity(intent);
            }
            else{
                        Utils.mensaje(getApplicationContext(),"\nNo hay favoritos.\n");
            }

      }

      // ****************************** C A T A L O G O ******************************************

      public void catalogo(View v)
      {

            boolean errores  = false;

            // recargar preferencias y obtener tamaño de catalogo desde preferencias
            prefs = PreferenceManager.getDefaultSharedPreferences(Principal.this);
            registrado = prefs.getBoolean("REGISTRO", false);
            catalogo = prefs.getBoolean("CATALOGO", false);
            tamano   = prefs.getInt("TAMANO", 0);

            Log.i("CATALOGO",catalogo +" tam: "+tamano);

            // Si existe un catalogo descargado verificar actualizaciones
            if (catalogo == true){

                        // Si hay conexion a Internet
                        if (Utils.verificaConexion(this)) {

                                    // Construir url
                                    url = "http://"+server+"/WebService/catalogo.json";

                                    // comprobar si el tamaño es diferente, indicar que existe una actualizacion
                                    int result[] = Utils.CompruebaUrl(getBaseContext(),url);

                                    if (result[0] == 200){


                                                if (result[1] != tamano){

                                                            Log.i("CATALOGO","code: "+result[0]+"  tam: "+result[1]);

                                                            AlertDialog.Builder ad = new AlertDialog.Builder(this);
                                                            ad.setTitle("CATALOGO\nActualización disponible.");
                                                            // asignar icono
                                                            boolean status = true;
                                                            ad.setIcon((status) ? R.drawable.success : R.drawable.fail);
                                                            ad.setMessage("¿Descargar Actualización?");
                                                            ad.setCancelable(false);
                                                            ad.setPositiveButton("Confirmar",new
DialogInterface.OnClickListener() {
```

```
                                        public void onClick(DialogInterface dialog, int arg1) {

                                                actualizar();
                                        }
                                    });

                                    ad.setNegativeButton("Cancelar", new
DialogInterface.OnClickListener() {
                                public void onClick(DialogInterface dialogo1, int id) {
                                    cargar();
                                }
                            });

                                    ad.show();


                                }else {
                                        //errores = true;
                                        Utils.mensaje(getApplicationContext(), "No hay
actualizaciones.");
                                        cargar();
                                }

                            }else {
                                    errores = true;
                            }

                        }else {
                            errores = true;
                        }

                    } else {
                        // Si no existe, descargar catalogo
                        cargar();
                    }

                    // Si hubo errores al comprobar actualizaciones
                    // cargar catalogo existente
                    if (errores){

                        Utils.mensaje(getApplicationContext(), "Imposible comprobar actualizaciones.");
                        cargar();
                    }

                }

                public void actualizar() {

                    // forzar descargar catalogo
                    catalogo = false;

                    actualiza = true;

                    /*
                    // activar editor de preferencias
```

```
                SharedPreferences.Editor prefEditor = prefs.edit();

                // Actualizar valor
                prefEditor.putBoolean("CATALOGO", false);
prefEditor.commit();
*/

cargar();

    }

    public void cargar() {

                    // variables de uso general
                    String mensa = "";
                    boolean errores = false;


                // Si catalogo = true es que existe un catalogo descargado en el movil
                // por lo que podemos lanzar la actividad
                if (catalogo){

                    // Ejecutar catalogo
            Intent intent = new Intent(Principal.this,Catalogo.class);
                    startActivity(intent);

                }
                // Si no hay catalogo verificar conexión y credenciales
                // para obtenerlo y lanzar actividad
                else {

                    // Si hay datos de registro verificar con el WebService
                    if (registrado){

                        // contiene datos de registro
                        ArrayList<NameValuePair>  datos;

                        // Obtener datos desde preferencias
                        String usuario = prefs.getString("user", "");
                    String clave   = prefs.getString("pass", "");



                    // ruta de la función del Web Service
                    String url = "http://"+server+"/WebService/validaUser.php";

                    //Rellenar datos de registro
                        //datos = new String[] {usuario.trim(), cpass};
        datos = new ArrayList<NameValuePair>();
                        datos.add(new BasicNameValuePair("usuario", usuario));
                        datos.add(new BasicNameValuePair("password", clave));
                        datos.add(new BasicNameValuePair("url", url));


                        // Si hay conexion a Internet
```

```
                    if (Utils.verificaConexion(this)) {

                        // ejecutar tarea asincrona de validar usuario
                        verificaUsuario validar = new verificaUsuario(this,datos);
                        validar.execute();

                    // Si no hay conexion mostrar aviso
                    }else {

                // crear mensaje y actualizar errores
                mensa = mensa + "\nComprueba tu conexión.";
                errores = true;
            }

        // Si no hay registro mostrar aviso
                }else {

                // crear mensaje y actualizar errores
                mensa = mensa + "\nNo hay datos de registro.\nDebe registrarse.";
                errores = true;
            }

        }

        // mostrar aviso por errores
    if (errores){

        mensa = mensa +"\n";
        // Mostrar dialogo
      Utils.showAlertDialog(Principal.this, "AVISO", mensa, false);

    } else {



    }
}



/*
         * Autentifica al usuario mediante funcion WebService
         *
         * Parámetro: context, contexto de la aplicación
         * Parámetro: datos, ArrayList con los datos
         * Retorno: resp, string con el valor devuelto por el WebService
        */

        // En nuestra tarea no necesitamos entrada,
        // no usaremos información de progreso y devolveremos un String.
        class verificaUsuario extends AsyncTask< Void, Void, String > {

        ProgressDialog Dialog;
        String cpass;
```

```
                    Context context;
                    ArrayList<NameValuePair> datos;
                    public verificaUsuario(Context context, ArrayList<NameValuePair> datos){
                      this.context=context;
                      this.datos = datos;
                    }

            protected void onPreExecute() {

              //para el progress dialog
            Log.i("VerificaUsuario", "onPreExecute "+context);
            Log.i("VerificaUsuario", "onPre clave: "+datos.get(1).getValue().trim());

                              //Dialog = ProgressDialog.show(context,"VALIDACION", "Comprobando datos de
usuario");

            Dialog = new ProgressDialog(Principal.this);
            Dialog.setMessage("Validando....");
            Dialog.setIndeterminate(false);
            Dialog.setCancelable(false);
            Dialog.show();

                              // ciframos la contraseña aplicandole una máscara (la misma que aplica el
WebService)
                              String masc = "|#€7`¬23ads4ook12";
                    cpass = masc + Utils.Cifrado.cifrar(datos.get(1).getValue().trim(), "MD5");
                    cpass = Utils.Cifrado.cifrar(cpass, "SHA-1");
                    // datos.remove(1);
                    datos.set(1, new BasicNameValuePair("password", cpass));

                    Log.i("VerificaUsuario", "onPre cclave: "+datos.get(1).getValue().trim());

                    // Codificar en base64
                    byte[] byteArray;
                    try {
                                        byteArray = datos.get(0).getValue().trim().getBytes("UTF-8");
                                        datos.set(0, new BasicNameValuePair("usuario",
Base64.encodeToString(byteArray,Base64.DEFAULT)));
                                  byteArray = datos.get(1).getValue().trim().getBytes("UTF-8");
                                  datos.set(1, new BasicNameValuePair("password",
Base64.encodeToString(byteArray,Base64.DEFAULT)));
                              } catch (UnsupportedEncodingException e) {
                                        // TODO Auto-generated catch block
                                        Log.i("VerificaUsuario", "Error: "+ e);
                                        //e.printStackTrace();
                              }


                    Log.i("VerificaUsuario", "onPre b64clave: "+datos.get(1).getValue().trim());


            }


                    protected String doInBackground(Void... params) {
```

```
                    String url  = datos.get(2).getValue().trim();

                    // elimimar url de datos
                    datos.set(2, new BasicNameValuePair("url", ""));

                    // usamos una clase httpHandler para enviar una petición al servidor web.
            Utils.httpHandler handler = new Utils.httpHandler();
            String resp = handler.post(url, datos);

            Log.i("VericaUsuario","resp: "+ resp+"url: "+url);

            // valor devuelto para onPostExecute
                return resp;

                }

                /*Una vez terminado doInBackground comprobar lo que halla ocurrido
                 * recibe resp
                */
protected void onPostExecute(String resp) {

  Log.i("onPostExecute=","resp: "+resp);

                //validamos el resultado obtenido
                    if (resp.trim().contains("0")){

                        // Mostrar mensaje
                        Log.i("verifica0 ", "invalido");
                        Utils.mensaje(context, " Usuario o contaseña incorrectos ");

                    }
                    else if (resp.trim().contains("1")){

                        // Mostrar mensaje
                        Log.i("verifica1 ", "valido");
                        //Utils.mensaje(context, " Usuario validado ");

                        if (actualiza){
                            // Cambiar preferencia para forzar descargar catalogo
                            // en la actividad catalogo

                            // activar editor de preferencias
                            SharedPreferences.Editor prefEditor = prefs.edit();

                            // Actualizar valor
                            prefEditor.putBoolean("CATALOGO", false);
                        prefEditor.commit();
                        }

                        // Ejecutar catalogo
                    Intent intent = new Intent(Principal.this,Catalogo.class);
                        startActivity(intent);

                    }
```

```
                                        else{

                                                // Mostrar mensaje
                                                Log.e("registerstatus ", "Error de sistema");
                                                Utils.mensaje(context, " Error de sistema. \n Vuelva a intentarlo.
");
                                        }

                                Dialog.dismiss();//ocultamos progess dialog.

        } // Onpostexecute


    } // verificaUsuario



// ************************** R E G I S T R O ****************************************

public void registro(View v)
{
        if (Utils.verificaConexion(this)) {

                Intent intent = new Intent(Principal.this,Registro.class);
                startActivity(intent);

        } else {
      // No hay conexion a Internet
        // Mostrar dialogo
      Utils.showAlertDialog(Principal.this, "AVISO",
          "Comprueba tu conexión.", false);
  }

}

// ************************* C O N F I G ***********************************************

public void config(View v)
{
                Intent intent = new Intent(Principal.this,Configuracion.class);
                startActivity(intent);
}



// ****************************** M E N U ***********************************************


// MENU
// Tendremos que implementar el evento onCreateOptionsMenu() de la actividad que queremos que lo muestre
// En este evento deberemos “inflar” el menú. Primero obtendremos una referencia al inflater
// mediante el método getMenuInflater() y posteriormente generaremos la estructura del menú llamando a su
```

```
        // método infate() pasándole como parámetro el ID del menu definido en XML, que mi nuestro caso será
R.menu.activity_main.
        // Por último devolveremos el valor true para confirmar que debe mostrarse el menú.

        @Override
        public boolean onCreateOptionsMenu(Menu menu) {

            super.onCreateOptionsMenu(menu);

            MenuInflater inflater = getMenuInflater();

            inflater.inflate(R.menu.menu, menu);

            return true; /** true -> el menú ya está visible */

          }


 // Construido el menú, la implementación de cada una de las opciones se incluirá en el evento onOptionsItemSelected()
// de la actividad que mostrará el menú. Este evento recibe como parámetro el item de menú que ha sido pulsado por el
usuario,
 // cuyo ID podemos recuperar con el método getItemId(). Según este ID podremos saber qué opción ha sido pulsada y
ejecutar
 // unas acciones u otras.

          @Override
          public boolean onOptionsItemSelected(MenuItem item) {

                        Intent intent;

              switch (item.getItemId()) {

                case R.id.favoritos:

                        favoritos(imgbtn1);

                  break;

                case R.id.config:

                        intent = new Intent(Principal.this,Configuracion.class);
                        startActivity(intent);

                  break;

                case R.id.registro:

                        intent = new Intent(Principal.this,Registro.class);
                        startActivity(intent);

                  break;

                case R.id.catalogo:

                        catalogo(imgbtn2);
```

```
                    break;

                case R.id.acercaDe:

                        // lanzar AcercaDe
                        intent = new Intent(Principal.this,AcercaDe.class);
                        startActivity(intent);

                    break;

                case R.id.salir:

                        finish();

                    break;

            }

            return true; /** true -> consumimos el item, no se propaga*/

        }


}
```

Registro.java

```
package es.hol.audiolibros;


import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.SharedPreferences;
import android.content.pm.ActivityInfo;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Vibrator;
import android.preference.PreferenceManager;
import android.util.Base64;
import android.util.Log;
import android.view.View;
import android.view.Window;
import android.view.WindowManager;
import android.widget.Button;
import android.widget.EditText;
```

```
public class Registro extends Activity {

   // Variables objetos View
   EditText user;
   EditText pass;
   EditText mail;
   Button btn_regis;

   // Variables de uso general
   String IP_Server="";                          // Ip o dominio del server
   String URL_connect="";                        // ruta del WebService
   String mensa = null;                          // mensajes para toast
   Context context;                              // contexto
   ArrayList<NameValuePair>  datos;              // contiene datos de registro

   boolean result_back;
   private ProgressDialog pDialog;

   @Override
   public void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);

      // fijar de manera permanente la orientación en modo vertical
      setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

      // Ocultar barra de titulo
      requestWindowFeature(Window.FEATURE_NO_TITLE);

      setContentView(R.layout.registro);

      // Obtener el contexto de la aplicacion
      context = getApplicationContext();

      // Referenciar objetos View
      user= (EditText) findViewById(R.id.edusuario);
      pass= (EditText) findViewById(R.id.edpassword);
      mail= (EditText) findViewById(R.id.edcorreo);

      btn_regis = (Button) findViewById(R.id.btn_regis);


      // Obtenemos la lista de preferencias mediante el método getDefaultSharedPreferences() y posteriormente
    // utilizamos los distintos métodos get() para recuperar el valor de cada opción dependiendo de su tipo.
    SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences(Registro.this);

            IP_Server     = prefs.getString("server", "www.audiolibros.hol.es");
            URL_connect = "http://"+IP_Server+"/WebService/registro.php";//ruta en donde estan nuestros archivos

            // Asignamos los valores de prefs a los objetos
            user.setText(prefs.getString("user", ""));
            pass.setText(prefs.getString("pass", ""));
            mail.setText(prefs.getString("mail", ""));
```

```
// Ocultamos el teclado virtual
// Aparece al tener el foco un EditText
getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_STATE_ALWAYS_HIDDEN);

 // Escuchador del boton registrar
btn_regis.setOnClickListener(new View.OnClickListener(){
public void onClick(View view){

        // Verificar conexion
        if (!Utils.verificaConexion(getApplicationContext())){
                // No hay conexion a Internet
                // Mostrar dialogo
                Utils.showAlertDialog(Registro.this, "Internet", "Comprueba tu conexión", false);

                return;
        }

        //Extraemos datos de los EditText
        String usuario = user.getText().toString().trim();
        String passw   = pass.getText().toString().trim();
        String correo  = mail.getText().toString().trim();

        //verificamos si no estan en blanco
        if(checkregdata(usuario, passw, correo) == true){

                // ciframos la contraseña aplicandole una máscara (la misma que aplica el WebService)
                String masc = "|#€7`¬23ads4ook12";
                String cpass = masc + Utils.Cifrado.cifrar(passw, "MD5");
                 cpass = Utils.Cifrado.cifrar(cpass, "SHA-1");

                //Rellenar datos de registro
                datos = new ArrayList<NameValuePair>();
                datos.add(new BasicNameValuePair("usuario", usuario));
                datos.add(new BasicNameValuePair("password", cpass));
                datos.add(new BasicNameValuePair("correo", correo));
                datos.add(new BasicNameValuePair("clave", passw));

                // Codificar en base64
                 byte[] byteArray;
                 try {
                        byteArray = datos.get(0).getValue().trim().getBytes("UTF-8");
                        datos.set(0, new BasicNameValuePair("usuario",

Base64.encodeToString(byteArray,Base64.DEFAULT)));
                          byteArray = datos.get(1).getValue().trim().getBytes("UTF-8");
                          datos.set(1, new BasicNameValuePair("password",

Base64.encodeToString(byteArray,Base64.DEFAULT)));
                           byteArray = datos.get(2).getValue().trim().getBytes("UTF-8");
                           datos.set(2, new BasicNameValuePair("correo",

Base64.encodeToString(byteArray,Base64.DEFAULT)));
                           byteArray = datos.get(3).getValue().trim().getBytes("UTF-8");
```

```
                                datos.set(3, new BasicNameValuePair("clave",

        Base64.encodeToString(byteArray,Base64.DEFAULT)));
                        } catch (UnsupportedEncodingException e) {
                                // TODO Auto-generated catch block
                                Log.i("Registro", "Error: "+ e);
                                //e.printStackTrace();
                        }

                        // ejecutar tarea asincrona de registro
                        new asyncreg().execute();

                }else{

                        Log.e("registerstatus ", "Validar campos");

                }// Check

        } // onclick

    }); // listener



} // onCreate

// Cierra la actividad
public void Volver(View v){
        finish();
}



//vibra y muestra un Toast
public void err_reg(Context context, String mensa, boolean vip){

        if (vip){
                // Activar vibracion
                Vibrator vibrator =(Vibrator) getSystemService(Context.VIBRATOR_SERVICE);
                vibrator.vibrate(200);
        }

           // Mostrar mensaje
           Utils.mensaje(context,mensa);

}


//validamos los campos
public boolean checkregdata(String username, String password, String mail  ){

        String mensa = "";
        boolean result = true;
```

```
        if      (username.equals("")) {
                Log.e("Register ui", "registerdata user");
                mensa = " Usuario vacio \n";
                result = false;
        }

        if (password.equals("") || password.length() < 4 ){
                Log.e("Register ui", "registerdata pass");
                mensa = mensa + " Contraseña vacia o longitud incorrecta \n";
                result = false;
        }

        if (mail.equals("") || !mail.contains("@")){
                Log.e("Register ui", "registerdata mail error");
                mensa = mensa + " Correo vacio o incorrecto \n";
                result = false;
        }

        if (!result){
                // Mostrar mensaje
                        err_reg(context, mensa, true);

                return false;
        }
        else{

                return true;
        }

}

        /*                      CLASE ASYNCTASK
         *
         * usaremos esta para poder mostrar el dialogo de progreso mientras enviamos y obtenemos los datos
         * podria hacerse lo mismo sin usar esto pero si el tiempo de respuesta es demasiado lo que podria ocurrir
         * si la conexion es lenta o el servidor tarda en responder la aplicacion sera inestable.
         * ademas observariamos el mensaje de que la app no responde.(ANR)
         */

// las peticiones http deben ejecutarse de una manera asíncrona, el motivo es porque a partir de la
// versión 4 de android dará un error si estas peticiones se ejecutan en el hilo principal,
// y esto tiene sentido, pues este tipo de sentencias pueden ralentizar la aplicación, aún así,
// en las versiones 2.x de android funcionará si lo ejecutamos en el hilo principal.
// Para evitar errores usaremos un proceso Asynctask para comunicarnos con el servidor.

class asyncreg extends AsyncTask< String, String, String > {

        //String user, pass, mail;

    protected void onPreExecute() {
        //para el progress dialog
      pDialog = new ProgressDialog(Registro.this);
      pDialog.setMessage("Registrando....");
      pDialog.setIndeterminate(false);
      pDialog.setCancelable(false);
```

```
    pDialog.show();
}

                protected String doInBackground(String... params) {

                        //String usuario = params[0];
                        //String passw   = params[1];
                        //String correo  = params[2];

                        // usamos una clase httpHandler para enviar una petición al servidor web.
            Utils.httpHandler handler = new Utils.httpHandler();
            String resp = handler.post(URL_connect, datos);

            Log.e("resp: ", resp+" url: "+URL_connect);

            // valor devuelto para onPostExecute
                return resp;

                }

                /*Una vez terminado doInBackground comprobar lo que halla ocurrido
                 * recibe resp
                */
protected void onPostExecute(String resp) {

  pDialog.dismiss();//ocultamos progess dialog.
  Log.e("onPostExecute=","""+resp);


                //validamos el resultado obtenido
                        if (resp.trim().contains("0")){

                                // Mostrar mensaje
                                Log.e("registerstatus0 ", "invalido");
                                err_reg(getBaseContext(),"  Usuario ya existe ", true);
                        }
                        else if (resp.trim().contains("1")){

                                // Mostrar mensaje
                                Log.e("registerstatus1 ", "invalido");
                                err_reg(getBaseContext(),"  Correo no válido ", true);
                        }
                        else if (resp.trim().contains("2")){

                                // Mostrar mensaje
                                Log.e("registerstatus2 ", "invalido");
                                err_reg(getBaseContext(),"  Error en BD ", true);
                        }
                        else if (resp.trim().contains("3")){

                                // Mostrar mensaje
                                Utils.mensaje(getBaseContext(),
                                        " REGISTRO EFECTUADO \n \n"+
                                                "Se ha enviado un mensaje a su dirección \n "+
                                                "de correo con los datos de registro. \n \n"+
```

```
                                                "Ya puede disfrutar de audiolibros \n \n");
                                Log.e("registerstatus3 ", "valido");

                                // Actualizar preferencias
                                SharedPreferences prefs =
PreferenceManager.getDefaultSharedPreferences(context);

                                SharedPreferences.Editor editor = prefs.edit();
                                editor.putString("user", user.getText().toString().trim());
                                editor.putString("pass", pass.getText().toString().trim());  // contraseña sin cifrar
                                editor.putString("mail", mail.getText().toString().trim());
                                editor.putBoolean("REGISTRO", true);
                                editor.putBoolean("reg_man", false);
                                editor.commit();

                                // Cerrar actividad
                                finish();

                        }
                                else{

                                        // Mostrar mensaje
                                         err_reg(getBaseContext(," Error de sistema. \n Vuelva a intentarlo. ",
true);

                                         Log.e("registerstatus ", "Error de sistema");

                                }

      } // Onpostexecute

   } // asyncreg

} //activity
```

MusicPlayerActivity.java

```
package es.hol.audiolibros;


import java.io.IOException;
import java.util.ArrayList;
import java.util.HashMap;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.ContentValues;
import android.content.Context;
import android.content.DialogInterface;
import android.content.pm.ActivityInfo;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.media.AudioManager;
import android.media.MediaPlayer;
import android.media.MediaPlayer.OnCompletionListener;
import android.media.MediaPlayer.OnErrorListener;
import android.media.MediaPlayer.OnPreparedListener;
import android.os.Bundle;
import android.os.Handler;
import android.util.Base64;
import android.util.Log;
import android.view.View;
import android.widget.ImageButton;
import android.widget.ImageView;
import android.widget.SeekBar;
import android.widget.TextView;


// La aplicación extiende la clase Activity.
// Además implementamos varios interfaces que corresponden a los escuchadores de eventos.
// OnCompletionListener será invocado cuando la reproducción llegue al final.
// OnSeekBarChangeListener será invocado cuando la barra de progreso cambie.
// OnPreparedListener será invocado cuando el audio esté preparado.
// OnErrorListener será invocado cuando hay un error.

//public class AndroidBuildingMusicPlayerActivity extends Activity implements OnCompletionListener,
SeekBar.OnSeekBarChangeListener {
public class MusicPlayerActivity extends Activity implements  OnPreparedListener, OnErrorListener,
OnCompletionListener, SeekBar.OnSeekBarChangeListener {


        // declaración de variables.
        private ImageButton btnVolbajar;
        private ImageButton btnBackward;
        private ImageButton btnPlay;
        private ImageButton btnForward;
        private ImageButton btnVolsubir;
        private ImageButton btnSalir;
```

```
private SeekBar songProgressBar;
private TextView songTitleLabel;
private TextView songCurrentDurationLabel;
private TextView songTotalDurationLabel;
private ImageView img_player;
private ProgressDialog pDialog;
private AudioManager audioManager = null;

// Media Player
private  MediaPlayer mp;

// Handler para actualizar timer, progress bar etc,.
private Handler mHandler;// = new Handler();

private Utils utils;
private int seekForwardTime = 5000; // 5000 milliseconds
private int seekBackwardTime = 5000; // 5000 milliseconds

// definimos una variable de tipo Bundle para los parametros pasados por Intent
// contiene desde, id, titulo_autor, url, pos
private Bundle bundle;

// almacena la posicion de reproduccion del audiolibro
private int pos_actual = 0;


@Override
public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // Fijar orientacion vertical
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

        setContentView(R.layout.player);

        setVolumeControlStream(AudioManager.STREAM_MUSIC);

        // Referenciar objetos View
        // componenetes del reproductor
        btnPlay = (ImageButton) findViewById(R.id.btnPlay);
        btnForward = (ImageButton) findViewById(R.id.btnForward);
        btnBackward = (ImageButton) findViewById(R.id.btnBackward);
        btnVolsubir = (ImageButton) findViewById(R.id.btnVolsubir);
        btnVolbajar = (ImageButton) findViewById(R.id.btnVolbajar);
        btnSalir = (ImageButton) findViewById(R.id.btnSalir);
        songProgressBar = (SeekBar) findViewById(R.id.songProgressBar);
        songTitleLabel = (TextView) findViewById(R.id.songTitle);
        songCurrentDurationLabel = (TextView) findViewById(R.id.songCurrentDurationLabel);
        songTotalDurationLabel = (TextView) findViewById(R.id.songTotalDurationLabel);
        img_player = (ImageView) findViewById(R.id.player_img);

        //Obtener el manejador del servicio de audio.
        audioManager = (AudioManager) getSystemService(Context.AUDIO_SERVICE);

        // Mediaplayer
```

```
        mp = new MediaPlayer();

        //mp = new MediaPlayer();
        mHandler = new Handler();

//      songManager = new SongsManager();
        utils = new Utils();


        // Escuchadores / Listeners
        // Indicamos mediante el método setOnSeekBarChangeListener la referencia de la clase
        // que será informada cuando cambie la barra de progreso.
        songProgressBar.setOnSeekBarChangeListener(this); // Important

        // Indicamos mediante el método setOnCompletionListener la referencia de la clase
        // que será informada cuando el audio finalice.
        mp.setOnCompletionListener(this);

        // Indicamos mediante el método setOnPreparedListener la referencia de la clase
        // que será informada cuando el audio esté preparado.
        mp.setOnPreparedListener(this);

        // Getting all songs list
        //songsList = songManager.getPlayList();

        // By default play first song
        //playSong(0);

        // inicializamos llamando al método getExtras() de la clase Intent (para recuperar el o los parámetros
        // que envió la otra actividad
        bundle = getIntent().getExtras();

        if (bundle.getString("desde").contains("favoritos")){
                // asignar punto de reproduccion
                pos_actual = Integer.parseInt(bundle.getString("pos"));

                // Obtener y convertir imagen de base64 a bitmap
                byte[] bytarray = Base64.decode(bundle.getString("img"), Base64.DEFAULT);
Bitmap bmimage = BitmapFactory.decodeByteArray(bytarray, 0, bytarray.length);

img_player.setImageBitmap(bmimage);

        }

        // asignar url
        String url = bundle.getString("url");

        // Pasar a reproducir el audio-stream
        reproducir(url);


        /**
         * Evento de pulsacion del boton Play
         * reproduce el audio y cambia el boton a pause
         * pausa el audio y cambia el boton a play
```

```
 * */
btnPlay.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {
                // comprobar si está reproduciendo
                if(mp.isPlaying()){
                        if(mp!=null){
                                mp.pause();
                                // cambiar botón a play
                                btnPlay.setImageResource(R.drawable.btn_play);
                        }
                }else{
                        // Reanudar audio
                        if(mp!=null){
                                mp.start();
                                // Cambiar boton a pause
                                btnPlay.setImageResource(R.drawable.btn_pause);
                        }

                }

        }
});

/**
 * Boton avanzar
 * Avanza los segundo especificados
 * */
btnForward.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {
                // obtener la posicion actual
                int currentPosition = mp.getCurrentPosition();
                // comprobar que el tiempo a avanzar es menor que la duracion del audio
                if(currentPosition + seekForwardTime <= mp.getDuration()){
                        // avanzar audio
                        mp.seekTo(currentPosition + seekForwardTime);
                }else{
                        // avanzar hasta el final
                        mp.seekTo(mp.getDuration());
                }
        }
});

/**
 * Boton retroceder
 * retrocede los segundos especificados
 * */
btnBackward.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {
                // obtener la posicion actual
```

```
                int currentPosition = mp.getCurrentPosition();
                // comprobar que el tiempo a retroceder es mayor que 0 sec
                if(currentPosition - seekBackwardTime >= 0){
                        // retroceder audio
                        mp.seekTo(currentPosition - seekBackwardTime);
                }else{
                        // retroceder a posicion inicial
                        mp.seekTo(0);
                }

        }
});

/**
 * Boton subir volumen
 *
 * */
btnVolsubir.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {

                        audioManager.adjustStreamVolume(AudioManager.STREAM_MUSIC,
                        AudioManager.ADJUST_RAISE, AudioManager.FLAG_SHOW_UI);

        }
});

/**
 * Boton bajar volumen
 *
 * */
btnVolbajar.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {

                audioManager.adjustStreamVolume(AudioManager.STREAM_MUSIC,
                        AudioManager.ADJUST_LOWER, AudioManager.FLAG_SHOW_UI);

        }
});


/**
 * Boton salir
 *
 * */
btnSalir.setOnClickListener(new View.OnClickListener() {

        @Override
        public void onClick(View arg0) {

                mp.pause();
```

```
                    // Recuperamos el valor "desde" pasado por en el intent
                    String desde = bundle.getString("desde");

                    if (desde.equals("catalogo"))
                    {

                        // Cerrar reproductor
                        finish();


                    }
                    else {   // es un favorito

                        // Creamos un objeto de la clase AlertDialog a través de la clase Builder
                        // Configuramos el mensaje y título del diálogo
                        // Evitamos que el diálogo sea saltado por cualquier medio distinto a
presionar                                   // alguno de los dos botones
                        // Llamamos al método setPositiveButton y setNegativeButton indicando el
texto a                 // mostrar en el botón y la clase
                        // anónima que capturará el evento clic del botón

                        AlertDialog.Builder dialogo = new
AlertDialog.Builder(MusicPlayerActivity.this);
                        dialogo.setMessage("¿Desea guardar estado de reproducción?")
                            .setTitle("   Salir   ")
                            .setCancelable(false)
                            .setNegativeButton("Salir",
                                new DialogInterface.OnClickListener() {
                                  public void onClick(DialogInterface dialog, int id) {
                                        finish();
                                    //dialog.cancel();
                                  }
                                })
                            .setPositiveButton("Guardar",
                                new DialogInterface.OnClickListener() {
                                  public void onClick(DialogInterface dialog, int id) {


                                        pos_actual = mp.getCurrentPosition();
                                        int _id = Integer.parseInt(bundle.getString("id"));

                                        guardar_estado(pos_actual, _id);

                                        finish();
                                  }
                                });

                        // Creamos el dialogo definido y lo mostramos
                        AlertDialog alert = dialogo.create();
                        alert.show();

                    } // if
                }
            });
```

```
} //onCreate


/**
 * reproducir
 * Param: url
 * */
public void  reproducir(String url){

        //String url ="http://audiolibros.hol.es/libros/gatonegro-audiolibros-ivoox1666676.mp3";
        //mp.release();
        //mp = new MediaPlayer();
        //mHandler = new Handler();
        //mp.setOnPreparedListener(this);
        mp.setAudioStreamType(AudioManager.STREAM_MUSIC);

        /*
        Toast toast = Toast.makeText(getApplicationContext(),
  "url: "+url, Toast.LENGTH_LONG);
  toast.setGravity(Gravity.TOP, 25, 300);
  toast.show();
        */

        try {
                mp.reset();
                mp.setOnErrorListener(this);
                mp.setOnPreparedListener(this);
                mp.setDataSource(url);
                //Log.d("URL2:",url);
                mp.prepareAsync();

        } catch (IllegalArgumentException e) {
                e.printStackTrace();
                Log.d("MP","Error1");
        } catch (SecurityException e) {
                e.printStackTrace();
                Log.d("MP","Error2");
        } catch (IllegalStateException e) {
                e.printStackTrace();
                Log.d("MP","Error3");
        } catch (IOException e) {
                e.printStackTrace();
                Log.d("MP","Error4");
        }



        pDialog = new ProgressDialog(this); //AndroidBuildingMusicPlayerActivity

        //pDialog = ProgressDialog.show(this, "", "Cargando...", true);
        pDialog.setMessage("Conectando...");
        pDialog.setIndeterminate(false);
        pDialog.setCancelable(true);
        pDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
```

```
                pDialog.show();


        }


        public void onPrepared(MediaPlayer mp) {

                // eliminar diálogo de progreso
                pDialog.dismiss();


                // Obtener el titulo + autor pasado por el Intent
                // Alargamos la cadena para poder hacer scroll
                String cancion = bundle.getString("titulo_autor");
                int lcancion=cancion.length();
                String relleno = " ";

                Log.d("Cancion",cancion+"/");

                if (lcancion < 42)
                {
                        for(int i=0; i < 60 - lcancion; i++)
                        {
                          cancion = cancion + relleno;
                        }

                        // añadimos un punto al final de la cadena para que los espacios añadidos tengan efecto
                        cancion = cancion + "·";
                }

                Log.d("Cancion",cancion+"/");

                // Asignamos el título y forzamos la seleccion del textview para activar scroll horizontal
                // El TextView empezara a ser scrollable cuando tenga el foco y/o haya sido seteada como selected

                songTitleLabel.setText(cancion);
                songTitleLabel.setSelected(true);

                // comenzar reproducción
    mp.start();
    mp.seekTo(pos_actual); // si es favorito

    // Cambiar la imagen del boton de play a pause
        btnPlay.setImageResource(R.drawable.btn_pause);

        // valores de la barra de progreso
        songProgressBar.setProgress(0);
        songProgressBar.setMax(100);

        // actualizar barra de progreso
        updateProgressBar();
}
```

```
public boolean onError(MediaPlayer arg0, int arg1, int arg2) {


        // eliminar diálogo de progreso
        if (pDialog.isShowing())
                pDialog.dismiss();

        // detener tarea en segundo plano
        mHandler.removeCallbacks(mUpdateTimeTask);

        // detener, obtener posicion y liberar reproductor
        mp.stop();
        pos_actual = mp.getCurrentPosition();
        mp.reset();

        // Recuperamos el valor "desde" pasado por en el intent
        String desde = bundle.getString("desde");

        if (desde.equals("catalogo"))
        {
                // Creamos un objeto de la clase AlertDialog a través de la clase Builder
                // Configuramos el mensaje y título del diálogo
                // Evitamos que el diálogo sea saltado por cualquier medio distinto a presionar alguno de los dos
                        // botones
                // Llamamos al método setPositiveButton y setNegativeButton indicando el texto a mostrar en
el botón                // y la clase
                // anónima que capturará el evento clic del botón

                AlertDialog.Builder dialogo = new AlertDialog.Builder(MusicPlayerActivity.this);
                dialogo.setMessage("Se ha producido un error o no hay conexión.")
                    .setTitle("   Salir   ")
                    .setCancelable(false)
                    .setNegativeButton("Salir",
                        new DialogInterface.OnClickListener() {
                          public void onClick(DialogInterface dialog, int id) {
                                finish();
                            //dialog.cancel();
                          }
                        });


                // Creamos el dialogo definido y lo mostramos
                AlertDialog alert = dialogo.create();
                alert.show();

        }
        else {  // es un favorito

                // Creamos un objeto de la clase AlertDialog a través de la clase Builder
                // Configuramos el mensaje y título del diálogo
                // Evitamos que el diálogo sea saltado por cualquier medio distinto a presionar alguno de los dos
                        // botones
                // Llamamos al método setPositiveButton y setNegativeButton indicando el texto a mostrar en
el botón                // y la clase
                // anónima que capturará el evento clic del botón
```

```
                    AlertDialog.Builder dialogo = new AlertDialog.Builder(MusicPlayerActivity.this);
                    dialogo.setMessage("¿Desea guardar estado de reproducción?")
                        .setTitle("   Salir   ")
                        .setCancelable(false)
                        .setNegativeButton("Salir",
                            new DialogInterface.OnClickListener() {
                              public void onClick(DialogInterface dialog, int id) {
                                    finish();
                                //dialog.cancel();
                              }
                            })
                        .setPositiveButton("Guardar",
                            new DialogInterface.OnClickListener() {
                              public void onClick(DialogInterface dialog, int id) {

                                    pos_actual = mp.getCurrentPosition();

                                    int _id = Integer.parseInt(bundle.getString("id"));

                                    guardar_estado(pos_actual, _id);

                                    finish();

                              }
                            });

                    // Creamos el dialogo definido y lo mostramos
                    AlertDialog alert = dialogo.create();
                    alert.show();

            }

    return false;

}


/**
 * Actualizar barra de progreso
 * */
public void updateProgressBar() {
            mHandler.postDelayed(mUpdateTimeTask, 100);
 }

/**
 * Hilo secundario
 * */
private Runnable mUpdateTimeTask = new Runnable() {
              public void run() {
                        long totalDuration = mp.getDuration();
                        long currentDuration = mp.getCurrentPosition();

                        // Muestra la duracion total
                        songTotalDurationLabel.setText(""+utils.milliSecondsToTimer(totalDuration));
```

```
                    // Muestra la posicion actual
                    songCurrentDurationLabel.setText(""+utils.milliSecondsToTimer(currentDuration));

                    // Actualiza barra de progreso
                    int progress = (int)(utils.getProgressPercentage(currentDuration, totalDuration));
                    //Log.d("Progress", ""+progress);
                    songProgressBar.setProgress(progress);

                    // Ejecutar el proceso cada 100 milliseconds
                mHandler.postDelayed(this, 100);
            }
        };

    /**
     *
     * */
    @Override
    public void onProgressChanged(SeekBar seekBar, int progress, boolean fromTouch) {

    }

    /**
     * Cuando el usuario mueve la barra de progreso
     * */
    @Override
    public void onStartTrackingTouch(SeekBar seekBar) {

        // detenemos el proceso en segundo plano
        mHandler.removeCallbacks(mUpdateTimeTask);
}

    /**
     * Cuando el usuario para el movimiento de la barra de progreso
     * */
    @Override
public void onStopTrackingTouch(SeekBar seekBar) {

        // detenemos el proceso en segundo plano
        mHandler.removeCallbacks(mUpdateTimeTask);

        int totalDuration = mp.getDuration();
        int currentPosition = utils.progressToTimer(seekBar.getProgress(), totalDuration);

        // avanzar o retroceder
        mp.seekTo(currentPosition);

        // actualizar barra de progreso
        updateProgressBar();
}

    /**
     * Cuando el audio se completa
     *
     * */
    @Override
```

```
public void onCompletion(MediaPlayer arg0) {

                    // detenemos el proceso en segundo plano runnable
                    //mHandler.removeCallbacks(mUpdateTimeTask);

                     // cambiar botón a play
                    btnPlay.setImageResource(R.drawable.btn_play);
                    mp.pause();
                    mp.seekTo(0);

                    // asignar valores a barra de progreso
                    songProgressBar.setProgress(0);


                    // Actualizar barra de progreso
                    updateProgressBar();

}

public void guardar_estado(int pos, int id){

          // Definir constantes
          final String KEY_ROWID    = "_id";
          final String KEY_POSICION = "posicion";
          final String DATABASE_TABLE = "libros";

          // Asignar objetos
          DataBase  db;                          // Clase SQLiteOpenHelper
          SQLiteDatabase bsSql;        // Clase para manipular los datos

          db = new DataBase(getApplicationContext());
          // Abre una conexion a la BD para lectura
          bsSql = db.getWritableDatabase();

          ContentValues args = new ContentValues();
          args.put(KEY_POSICION, pos);

          //manda una sentencia UPDATE a la BD para modificar el contacto identifiado por id
          bsSql.update(DATABASE_TABLE, args,KEY_ROWID + "=" + id, null) ;

          bsSql.close();
          db.close();

}

@Override
 public void onDestroy(){
 super.onDestroy();

 if (bundle.getString("desde").contains("favoritos")){

// btnSalir.performClick();

 }
```

```
        // detenemos el proceso en segundo plano runnable
                mHandler.removeCallbacks(mUpdateTimeTask);

                if(mp !=null){
                    if (mp.isPlaying()){
                        mp.stop();
                        pos_actual = mp.getCurrentPosition();
                    }
                    // liberar reproductor
                    mp.reset();
                    mp.release();
                }



        }

}
```

Utils.java

```
package es.hol.audiolibros;

import java.io.BufferedInputStream;
import java.io.BufferedReader;
import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileNotFoundException;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.HttpURLConnection;
import java.net.MalformedURLException;
import java.net.URL;
import java.net.URLConnection;
import java.net.UnknownHostException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.ArrayList;
import java.util.List;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.HttpStatus;
import org.apache.http.NameValuePair;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.protocol.HTTP;
import org.apache.http.util.ByteArrayBuffer;
import org.apache.http.util.EntityUtils;
import org.json.JSONArray;
import org.json.JSONException;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.SharedPreferences;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.graphics.Bitmap.CompressFormat;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Environment;
import android.os.StatFs;
import android.preference.PreferenceManager;
import android.util.Log;
```

```
import android.view.Gravity;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.ImageView;
import android.widget.Toast;


public class Utils {

	/**
	 * Convierte el tiempo en milisegundos
	 * al formato de tiempo
	 * HH:MM:SS
	 * */
	public String milliSecondsToTimer(long milliseconds){
		String finalTimerString = "";
		String secondsString = "";

		// Convertir la duracion total
		  int hours = (int)( milliseconds / (1000*60*60));
		  int minutes = (int)(milliseconds % (1000*60*60)) / (1000*60);
		  int seconds = (int) ((milliseconds % (1000*60*60)) % (1000*60) / 1000);

		  // Añadir horas
		  if(hours > 0){
			finalTimerString = hours + ":";
		  }

		  // colocar 0 a los segundos si son de un digito
		  if(seconds < 10){
			secondsString = "0" + seconds;
		  }else{
			secondsString = "" + seconds;}

		  finalTimerString = finalTimerString + minutes + ":" + secondsString;

		// devolver la cadena de tiempo
		return finalTimerString;
	}

	/**
	 * Funcion para obtener el porcentaje de progreso
	 * @param currentDuration
	 * @param totalDuration
	 * */
	public int getProgressPercentage(long currentDuration, long totalDuration){
		Double percentage = (double) 0;

		long currentSeconds = (int) (currentDuration / 1000);
		long totalSeconds = (int) (totalDuration / 1000);

		// calculando porcentaje
```

```
		percentage =(((double)currentSeconds)/totalSeconds)*100;

		// devolviendo porcentaje
		return percentage.intValue();
}

/**
 * Funcion para cambiar el progreso a escala de tiempo
 * @param progress -
 * @param totalDuration
 * returns current duration in milliseconds
 * */
public int progressToTimer(int progress, int totalDuration) {
		int currentDuration = 0;
		totalDuration = (int) (totalDuration / 1000);
		currentDuration = (int) ((((double)progress) / 100) * totalDuration);

		// devolver la duracion actual en milisegundos
		return currentDuration * 1000;
}


/**
 * Funcion para convertir bitmap en array de byte
 * @param  bitmap image
 *
 * returns array de byte []
 * */

public static byte[] convertBitmapToByteArray(Bitmap image) {

		if (image == null) {
			return null;
		} else {
			byte[] b = null;
			try {
				ByteArrayOutputStream stream = new ByteArrayOutputStream();
				image.compress(CompressFormat.JPEG, 100, stream);
				b = stream.toByteArray();
			} catch (Exception e) {
				e.printStackTrace();
			}
			return b;
		}
 }



/*
 * Autentifica al usuario mediante funcion WebService usando clase asincrona
 *
 * Parámetro: context, contexto de la aplicación
 * Parámetro: datos, ArrayList con los datos
 * Retorno: resp, string con el valor devuelto por el WebService
*/
```

```
// En nuestra tarea no necesitamos entrada,
// no usaremos información de progreso y devolveremos un String.
static class verificaUsuario extends AsyncTask< Void, Void, String > {


ProgressDialog Dialog;
String cpass;

        Context context;
        ArrayList<NameValuePair> datos;
        public verificaUsuario(Context context, ArrayList<NameValuePair> datos){
          this.context=context;
          this.datos = datos;
        }

protected void onPreExecute() {

//para el progress dialog
Log.i("VerificaUsuario", "onPreExecute "+context);
Log.i("VerificaUsuario", "onPre clave: "+datos.get(1).getValue().trim());

Dialog = ProgressDialog.show(context,"VALIDACION", "Comprobando datos de usuario");

// ciframos la contraseña aplicandole una máscara (la misma que aplica el WebService)
String masc = "|#€7`¬23ads4ook12";
cpass = masc + Utils.Cifrado.cifrar(datos.get(1).getValue().trim(), "MD5");
cpass = Utils.Cifrado.cifrar(cpass, "SHA-1");
datos.set(1, new BasicNameValuePair("password", cpass));

Log.i("VerificaUsuario", "onPre cclave: "+datos.get(1).getValue().trim());


}

protected String doInBackground(Void... params) {

                //String usuario = params[0];
                //String passw   = params[1];
                //String url  = params[2];

        String url  = datos.get(2).getValue().trim();

        // usamos una clase httpHandler para enviar una petición al servidor web.
        Utils.httpHandler handler = new Utils.httpHandler();
        String resp = handler.post(url, datos);

         Log.i("VericaUsuario","resp: "+ resp+"url: "+url);

        // valor devuelto para onPostExecute
        return resp;

}

/*Una vez terminado doInBackground comprobar lo que halla ocurrido
```

```
 * recibe resp
*/
protected void onPostExecute(String resp) {

        Dialog.dismiss();//ocultamos progess dialog.

        Log.i("onPostExecute=","resp: "+resp);

         //validamos el resultado obtenido
         if (resp.trim().contains("0")){

                // Mostrar mensaje
                Log.i("verifica0 ", "invalido");
                Utils.mensaje(context, " Usuario o contaseña incorrectos ");
                //err_reg(context," Usuario no existe ", true);
        }
        else if (resp.trim().contains("1")){

                // Mostrar mensaje
                Log.i("verifica1 ", "valido");
                Utils.mensaje(context, " Usuario validado ");
                //err_reg(context," Correo no válido ", true);

        }
        else{

                // Mostrar mensaje
                //err_reg(context," Error de sistema. \n Vuelva a intentarlo. ", true);
                Log.e("registerstatus ", "Error de sistema");
                Utils.mensaje(context, " Error de sistema. \n Vuelva a intentarlo. ");
        }

  } // Onpostexecute


} // verificaUsuario


/*
 * Comprueba si hay conexión de datos
 *
 * Parámetro: context, contexto de la aplicación
 *
 * Retorno: bConectado true o false
*/

public static boolean verificaConexion(Context ctx) {

        boolean bConectado = false;
```

```
                ConnectivityManager connec =

        (ConnectivityManager)ctx.getSystemService(ctx.CONNECTIVITY_SERVICE);

                // Almacenar las conexiones disponibles, no sólo wifi, también GPRS
                 NetworkInfo[] redes = connec.getAllNetworkInfo();

                // bucle para comprobar el estado de las conexiones
                for (int i = 0; i < 2; i++) {
                        // ¿Tenemos conexión? ponemos a true
                        if (redes[i].getState() == NetworkInfo.State.CONNECTED) {
                                bConectado = true;
                        }
                }

          return bConectado;
        }



 /**
 * Funcion que muestra un simple Alert Dialog
 * @param context - application context
 * @param title - alert dialog title
 * @param message - alert message
 * @param status - success/failure (used to set icon)
 * */
static void showAlertDialog(Context context, String title, String message, Boolean status) {

   AlertDialog alertDialog = new AlertDialog.Builder(context).create();

   // Setting Dialog Title
   alertDialog.setTitle(title);

   // Setting Dialog Message
   alertDialog.setMessage(message);

   alertDialog.setCancelable(false);

   // Setting alert dialog icon
   alertDialog.setIcon((status) ? R.drawable.success : R.drawable.aviso);

   // Setting OK Button
   alertDialog.setButton("OK", new DialogInterface.OnClickListener() {
      public void onClick(DialogInterface dialog, int which) {
      }
   });

   // Showing Alert Message
   alertDialog.show();

 }
```

```
/*
 * Muestra un mensaje centrado en la pantalla
 *
 * Parámetro: context, para poder mostrar mensajes
 * Parámetro: mensa, cadena a mostrar
 *
 * Retorno: No
*/

static void mensaje(Context context, String mensa){

            Toast toast = Toast.makeText(context, mensa , Toast.LENGTH_LONG);
             toast.setGravity(Gravity.CENTER_VERTICAL, 0, 0);
             toast.show();
 }


/*
 * Comprueba que existe un recurso
 *
 * Parámetro: context para poder mostrar mensajes
 * Parámetro: url dirección del recurso
 *
 * Retorno: int[] que contine el código de respuesta y el tamaño del recurso
*/


          static int[] CompruebaUrl(Context context, String url){

                    int responseCode = 0;
                    int totalSize = 0;

                    try {

                       //desactivar redirecciones
                      HttpURLConnection.setFollowRedirects(false);

                       //establecemos la conexión con el destino
                       HttpURLConnection con = (HttpURLConnection) new URL(url).openConnection();

                       // Conectamos utilizando el método GET
                       con.setRequestMethod ("GET");

                       // obtener codigo de respuesta
                       responseCode = con.getResponseCode();

                       // obtener tamaño del recurso
                       totalSize = con.getContentLength();

                       // desconectar conexion
                       con.disconnect();

                    } catch (UnknownHostException uhe) {
                       // Handle exceptions as necessary
                              Utils.mensaje(context,"Error: \n"+uhe);
```

```
                } catch (FileNotFoundException fnfe) {
                    // Handle exceptions as necessary
                        Utils.mensaje(context,"No Existe: "+url);
                } catch (Exception e) {
                    // Handle exceptions as necessary
                        Utils.mensaje(context,"Error: \n"+e);
                }

                Log.i("CompUrl","Code: "+responseCode+" Size: "+totalSize);

                // devolver codigo y tamaño
                return new int[] {responseCode, totalSize};

        }

/*
 * Descargar imagen desde url
 *
 * Parámetro: url dirección del recurso
 *
 * Retorno: Bitmap que contiene la imagen
 */

        static class DownloadImage{

                static Bitmap downloadImage(String URL){

                Bitmap bitmap = null;
                InputStream in = null;

                try {
                        in = OpenHttpConnection(URL);
                        bitmap = BitmapFactory.decodeStream(in);
                         in.close();
                } catch (IOException e1) {
                        // TODO Auto-generated catch block
                        e1.printStackTrace();
                }

                return bitmap;

                } // downloadImage

        }


// abrir una conexion HTTP y devolver un objeto InputStream
   private static InputStream OpenHttpConnection(String urlString) throws IOException
   {
      InputStream in = null;
      int response = -1;

      URL url = new URL(urlString);
      URLConnection conn = url.openConnection();
```

```
    if (!(conn instanceof HttpURLConnection))
        throw new IOException("Not an HTTP connection");

    try{
        HttpURLConnection httpConn = (HttpURLConnection) conn;
        httpConn.setAllowUserInteraction(false);
        httpConn.setInstanceFollowRedirects(true);
        httpConn.setRequestMethod("GET");
        httpConn.connect();

        response = httpConn.getResponseCode();
        if (response == HttpURLConnection.HTTP_OK) {
            in = httpConn.getInputStream();
        }
    }
    catch (Exception ex)
    {
        throw new IOException("Error connecting");
    }
    return in;
}

 /*
* Descargar imagen desde url
*
* Parámetro: url dirección del recurso
*
* Retorno: Array de byte que contiene la imagen
*/
 static class  DownloadImage1{

            static byte[] getLogoImage(String url){

                try {
                        URL imageUrl = new URL(url);
                        URLConnection ucon = imageUrl.openConnection();

                        InputStream is = ucon.getInputStream();
                        BufferedInputStream bis = new BufferedInputStream(is);

                        ByteArrayBuffer baf = new ByteArrayBuffer(500);
                        int current = 0;
                        while ((current = bis.read()) != -1) {
                                baf.append((byte) current);
                        }

                        return baf.toByteArray();
                } catch (Exception e) {
                        Log.d("ImageManager", "Error: " + e.toString());
                }

                return null;
            }

}
```

```
/*
 * Descargar imagen desde url usando clase asíncrona
 *
 * Parámetro: url dirección del recurso
 *
 * Retorno: Bitmap que contine la imagen
*/


        static class ImageDownloader3 extends AsyncTask<String, Void, Bitmap > {

                ProgressDialog Dialog;

                Context context;
                String url;
                public ImageDownloader3(Context context, String url){
                  this.context=context;
                  this.url = url;
                }

                        @Override
                        protected Bitmap doInBackground(String... param) {
                                // TODO Auto-generated method stub
                                //return downloadBitmap(param[0]);
                                return downloadBitmap(url);
                        }

                        @Override
                        protected void onPreExecute() {
                                super.onPreExecute();
                                Log.i("Async-Example", "onPreExecute Called "+context);
                                Dialog = ProgressDialog.show(context,"Wait", "Downloading Image");

                        }

                        @Override
                        protected void onPostExecute(Bitmap result) {
                                Log.i("Async-Example", "onPostExecute Called");
                                // downloadedImg.setImageBitmap(result);
                                Dialog.dismiss();
                                Toast.makeText(context, url, Toast.LENGTH_LONG).show();

                        }

                        private Bitmap downloadBitmap(String url) {
                                // initilize the default HTTP client object
                                final DefaultHttpClient client = new DefaultHttpClient();

                                //forming a HttoGet request
                                final HttpGet getRequest = new HttpGet(url);
                                try {

                                        HttpResponse response = client.execute(getRequest);
```

```
                    //check 200 OK for success
                    final int statusCode = response.getStatusLine().getStatusCode();

                    if (statusCode != HttpStatus.SC_OK) {
                        Log.w("ImageDownloader", "Error " + statusCode + " while
                                retrieving bitmap from " + url);
                        return null;
                    }

                    final HttpEntity entity = response.getEntity();
                    if (entity != null) {
                        InputStream inputStream = null;
                        try {
                            // getting contents from the stream
                            inputStream = entity.getContent();

                            // decoding stream data back into image Bitmap that
                                android understands
                            final Bitmap bitmap =

BitmapFactory.decodeStream(inputStream);
                            return bitmap;
                        } finally {
                            if (inputStream != null) {
                                inputStream.close();
                            }
                            entity.consumeContent();
                        }
                    }
                } catch (Exception e) {
                    // You Could provide a more explicit error message for IOException
                    getRequest.abort();
                    Log.e("ImageDownloader", "Something went wrong while retrieving
                            bitmap from " + url + e.toString());
                }

                return null;
            }

    }


/*
 * Descargar imagen desde url usando clase asíncrona
 *
 * Parámetro: url dirección del recurso
 *
 * Retorno: Bitmap que contine la imagen
*/
```

```
        // DownloadImage AsyncTask
static class DownloadImage0 extends AsyncTask<String, Void, Bitmap> {

    ProgressDialog mProgressDialog;

            Context context;
            String URL;
            public  DownloadImage0(Context context, String URL){
              this.context=context;
              this.URL = URL;
            }

  @Override
  protected void onPreExecute() {
    super.onPreExecute();
    // Create a progressdialog
    mProgressDialog = new ProgressDialog(context);
    // Set progressdialog title
    mProgressDialog.setTitle("Download Image Tutorial");
    // Set progressdialog message
    mProgressDialog.setMessage("Loading...");
    mProgressDialog.setIndeterminate(false);
    // Show progressdialog
    mProgressDialog.show();
  }

  @Override
  protected Bitmap doInBackground(String... param) {

    String imageURL = URL;

    Bitmap bitmap = null;
    try {
      // Download Image from URL
      InputStream input = new java.net.URL(imageURL).openStream();
      // Decode Bitmap
      bitmap = BitmapFactory.decodeStream(input);
    } catch (Exception e) {
      e.printStackTrace();
    }

    return bitmap;
  }

  @Override
  protected void onPostExecute(Bitmap result) {
    // Set the bitmap into ImageView
    //image.setImageBitmap(result);
    // Close progressdialog
    mProgressDialog.dismiss();
  }
}
```

```
/*
* Descargar imagen desde url usando clase asíncrona
*
* Parámetro: cadena de typo byte[]
*
* Retorno: string cifrado
*/


    static class Cifrado {

        //algoritmos
        public static String MD2 = "MD2";
        public static String MD5 = "MD5";
        public static String SHA1 = "SHA-1";
        public static String SHA256 = "SHA-256";
        public static String SHA384 = "SHA-384";
        public static String SHA512 = "SHA-512";

        /***
         * Convierte un arreglo de bytes a String usando valores hexadecimales
         * @param digest arreglo de bytes a convertir
         * @return String creado a partir de <code>digest</code>
         */
        private static String toHexadecimal(byte[] digest){
            String hash = "";
            for(byte aux : digest) {
                int b = aux & 0xff;
                if (Integer.toHexString(b).length() == 1) hash += "0";
                hash += Integer.toHexString(b);
            }
            return hash;
        }

        /***
         * Encripta un mensaje de texto mediante algoritmo de resumen de mensaje.
         * @param message texto a encriptar
         * @param algorithm algoritmo de encriptacion, puede ser: MD2, MD5, SHA-1, SHA-256, SHA-384, SHA-512
         * @return mensaje encriptado
         */
        public static String cifrar(String message, String algorithm){
            byte[] digest = null;
            byte[] buffer = message.getBytes();
            try {
                MessageDigest messageDigest = MessageDigest.getInstance(algorithm);
                messageDigest.reset();
                messageDigest.update(buffer);
                digest = messageDigest.digest();
            } catch (NoSuchAlgorithmException ex) {
                System.out.println("Error creando Digest");
            }
            return toHexadecimal(digest);
        }
    }
```

```
/*CLASE AUXILIAR PARA EL ENVIO DE PETICIONES A NUESTRO SISTEMA
 * Y MANEJO DE RESPUESTA.
 *
*/
public class Httppostaux{

        InputStream is = null;
        String result = "";

        public JSONArray getserverdata(ArrayList<NameValuePair> parameters, String urlwebserver ){

                //conecta via http y envia un post.
                httppostconnect(parameters,urlwebserver);

                if (is!=null){//si obtuvo una respuesta

                        getpostresponse();

                       return getjsonarray();

                }else{

                   return null;

                }

        }


    //peticion HTTP
    private void httppostconnect(ArrayList<NameValuePair> parametros, String urlwebserver){

      //
      try{
            HttpClient httpclient = new DefaultHttpClient();
            HttpPost httppost = new HttpPost(urlwebserver);
            httppost.setEntity(new UrlEncodedFormEntity(parametros));
            //ejecuto peticion enviando datos por POST
            HttpResponse response = httpclient.execute(httppost);
            HttpEntity entity = response.getEntity();
            is = entity.getContent();

      }catch(Exception e){
            Log.e("log_tag", "Error in http connection "+e.toString());
      }

   }

   public void getpostresponse(){

      //Convierte respuesta a String
      try{
            BufferedReader reader = new BufferedReader(new InputStreamReader(is,"iso-8859-1"),8);
            StringBuilder sb = new StringBuilder();
```

```
            String line = null;
            while ((line = reader.readLine()) != null) {
                sb.append(line + "\n");
            }
            is.close();

            result=sb.toString();
            Log.e("getpostresponse"," result= "+sb.toString());
        }catch(Exception e){
            Log.e("log_tag", "Error converting result "+e.toString());
        }
    }

    public JSONArray getjsonarray(){
        //parse json data
        try{
          JSONArray jArray = new JSONArray(result);

          return jArray;
        }
        catch(JSONException e){
            Log.e("log_tag", "Error parsing data "+e.toString());
            return null;
        }

    }


    }


// En esta clase lo que hacemos es implementar la comunicación, aqui estas todo lo que se necesita
// para enviar la petición a PHP junto con información.

public static class httpHandler {

        String post(String posturl, ArrayList<NameValuePair>  params){

        InputStream is = null;
        String text = "";

        Log.i("Handler ", posturl);

        // Creamos el objeto de HttpClient que nos permitira conectarnos mediante peticiones http
        HttpClient httpclient = new DefaultHttpClient();

        // El objeto HttpPost permite que enviemos una peticion de tipo POST a una URL especificada
        HttpPost httppost = new HttpPost(posturl);

                try {

                        // anexamos los parametros al objeto para que al enviarse al servidor
                        // envien los datos que hemos añadido
                        httppost.setEntity(new UrlEncodedFormEntity(params));
```

```
                    // Finalmente ejecutamos enviando la info al server
                     HttpResponse resp = httpclient.execute(httppost);

                     // y obtenemos una respuesta
                     HttpEntity ent = resp.getEntity();

                    is = ent.getContent();

                    //String valor = EntityUtils.toString(resp.getEntity(), HTTP.UTF_8);

                    //Log.i("Handler: ", "Valor: " + valor);

                     if (is != null) {

                            //Convierte respuesta a String
                            try{
                    BufferedReader reader = new BufferedReader(new InputStreamReader(is,"UTF-8"),8);
                    StringBuilder sb = new StringBuilder();

                                    String line = null;
                                    while ((line = reader.readLine()) != null) {
                                            sb.append(line + "\n");
                                    }

                                    is.close();

                                    text=sb.toString();
                                    Log.i("Handler: ", "Result: " + is);
                                    Log.i("Handler: ", "Result= "+sb.toString());
                                    Log.i("Handler: ", "Result: " +text);

                            }catch(Exception e){
                                    Log.e("log_tag", "Error converting result "+e.toString());
                            }
                     }
                     //String text = EntityUtils.toString(ent);

            }
            catch(Exception e) {
                     Log.e("Handler ", "Error al enviar al server");
                     e.printStackTrace();
                     return "error";
            }

            return text;
    }

}
```

```
/*
 * Descargar fichero desde url y guardarlo en el dispositivo
 *
 * Parámetro: ruta dirección del recurso
 *
 * Retorno: nada
*/

    public static void descarga_file (Context context,String ruta){

            try {
              //indico URL del archivo
              String DownloadUrl = ruta;

              //nombre que tendrá el archivo
              String fileName="catalogo.json";

              // Crear drectorio si no existe
              String path = context.getFilesDir() + "/Cat_Audio/";
                File storagePath = new File(path);  // + File.separator
                storagePath.mkdirs();

                Log.i("Descarga", "Ruta: "+storagePath);

                // Definir ruta como URL
              URL url = new URL(DownloadUrl);
              //File file = new File(storagePath, fileName);

              long startTime = System.currentTimeMillis();
              Log.d("Descarga", "descarga comenzada");
              Log.d("Descarga", "descarga url:" + url);
              Log.d("Descarga", "nombre archivo:" + fileName);

              // Abrir una conexion a URL. y obtener el tamaño del recurso
              URLConnection ucon = url.openConnection();
              int totalSize = ucon.getContentLength();

              Log.d("Descarga", "CONEXION CORRECTA tam: "+totalSize);

              // Defino InputStreams para leer desde la URLConnection.
              InputStream is = ucon.getInputStream();
              // Definir buffer de lectura con tamaño inicial
              BufferedInputStream bis = new BufferedInputStream(is, 1024 * 5);

              Log.d("Descarga", "INPUT STREAM");

             // File file = new File(context.getDir("Cat_Audio", Context.MODE_PRIVATE) + "/"+ fileName);

              //Creamos un objeto del tipo de fichero
              // donde descargaremos nuestro fichero
              File file = new File(storagePath, fileName);

              // Si existe lo borramos
              if (file.exists())
              {
```

```
            file.delete();
          }

         // boolean resp = file.createNewFile();

         // Log.i("Descarga", "resp: "+ resp);


          // creamos un buffer con un tamaño inicial y una variable para ir almacenando el
          // tamaño temporal de este
          ByteArrayBuffer baf = new ByteArrayBuffer(5 * 1024);
          int current = 0;

          // Leer y añado bytes para el Buffer hasta que no queden mas para leer(-1)
          while ((current = bis.read()) != -1) {
            baf.append((byte) current);
          }

          Log.d("Descarga", "path: "+path+fileName);

          // utilizaremos un objeto del tipo fileoutputstream
          // para escribir el archivo que descargamos en el nuevo
          FileOutputStream fos = new FileOutputStream(file);

         //  byte[] buff = new byte[5 * 1024];

         //  int len;
         //  while ((len = bis.read(buff)) != -1)
         //  {
         ///      fos.write(buff, 0, len);
         //  }

          // Guardar el contenido del buffer en disco
          fos.write(baf.toByteArray());
          fos.flush();

          // Cerrar y limpiar
          fos.close();
          bis.close();
          baf.clear();

          Log.d("Descarga", "descarga lista en "  + ((System.currentTimeMillis() - startTime) / 1000) + "
                    segundos");

        } catch (IOException e) {
            Log.d("Descarga", "Error: " + e);
        }

    }

/*
 * Descargar fichero desde url y guardarlo en la SD del dispositivo
 *
 * Parámetro: ruta dirección del recurso
 *
```

```
 * Retorno: nada
 */

    public static void descarga_fileSD (String ruta){

        try {

                //indico URL al archivo
                String DownloadUrl = ruta;

                //nombre que tendrá el archivo
                String fileName="catalogo.json";

                // Crear drectorio si no existe
                File storagePath = new File(Environment.getExternalStorageDirectory(),"Cat_Audio");
                                        storagePath.mkdirs();

                // Definir variables
                    URL url = new URL(DownloadUrl);
                     File file = new File(storagePath, fileName);

                      long startTime = System.currentTimeMillis();
                      Log.d("Descarga", "descarga comenzada");
                      Log.d("Descarga", "descarga url:" + url);
                      Log.d("Descarga", "nombre archivo:" + fileName);

                      /* abro una conexion a URL. */
                      URLConnection ucon = url.openConnection();
                      int totalSize = ucon.getContentLength();
                      Log.d("Descarga", "CONEXION CORRECTA tam: "+totalSize);


                      // Defino InputStreams y buffer para leer desde la URLConnection
                      InputStream is = ucon.getInputStream();
                      BufferedInputStream bis = new BufferedInputStream(is);
                      Log.d("Descarga", "INPUT STREAM");


                      // Definir buffer con capacidad inicial
                      ByteArrayBuffer baf = new ByteArrayBuffer(5 * 1024);
                      int current = 0;

                      // leo y añado bytes para el Buffer hasta que no queden mas para leer(-1)
                      while ((current = bis.read()) != -1) {
                        baf.append((byte) current);
                      }

                      // Guardar datos leidos
                      FileOutputStream fos = new FileOutputStream(file);
                      fos.write(baf.toByteArray());
                      fos.flush();
                      fos.close();

                      Log.d("Descarga", "descarga lista en "  + ((System.currentTimeMillis() - startTime) / 1000) + "
                                segundos");
```

```
                    } catch (IOException e) {
                        Log.d("Descarga", "Error: " + e);
                    }

        }


/*
 * Funciones para control del espacio libre de memoria
 *
 * Memoria Total, Libre y ocupada.
 *
 *
 */

        /*****************************************************************************************
        Returns size in MegaBytes.

        If you need calculate external memory, change this:
          StatFs statFs = new StatFs(Environment.getRootDirectory().getAbsolutePath());
        to this:
          StatFs statFs = new StatFs(Environment.getExternalStorageDirectory().getAbsolutePath());
        *****************************************************************************************/
        public static long TotalMemory()
          {
            StatFs statFs = new StatFs(Environment.getRootDirectory().getAbsolutePath());
            long Total = ( (long) statFs.getBlockCount() *  (long) statFs.getBlockSize()) / 1048576;
            return Total;
          }

          public static long FreeMemory()
          {
            StatFs statFs = new StatFs(Environment.getRootDirectory().getAbsolutePath());
            long Free  = (statFs.getAvailableBlocks() *  (long) statFs.getBlockSize()) / 1048576;
            return Free;
          }

          public static long BusyMemory()
          {
            StatFs statFs = new StatFs(Environment.getRootDirectory().getAbsolutePath());
            long Total = ( (long) statFs.getBlockCount() * (long) statFs.getBlockSize()) / 1048576;
            long Free  = (statFs.getAvailableBlocks() *  (long) statFs.getBlockSize()) / 1048576;
            long Busy  = Total - Free;
            return Busy;
          }


         /**
          * @return Numero de  bytes disponibles en almacenamiento interno
          */
         public static long getInternalAvailableSpace() {
           long availableSpace = -1L;
           try {StatFs stat = new StatFs(Environment.getDataDirectory()
                 .getPath());
```

```
            stat.restat(Environment.getDataDirectory().getPath());
            availableSpace = (long) stat.getAvailableBlocks() * (long) stat.getBlockSize();
        } catch (Exception e) {
            e.printStackTrace();
        }

        return availableSpace;
    }

}  // Utils
```

AcercaDe.java

```
package es.hol.audiolibros;

import android.app.Activity;
import android.content.pm.ActivityInfo;
import android.os.Bundle;
import android.view.Window;

public class AcercaDe extends Activity {

    /** Called when the activity is first created. */

    @Override public void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        // Fijar orientacion vertical
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

        // Ocultar barra de titulo
        requestWindowFeature(Window.FEATURE_NO_TITLE);

        setContentView(R.layout.acercade);

    }

}
```

Catalogo.java

```
package es.hol.audiolibros;

import java.io.ByteArrayOutputStream;
import java.io.File;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.HashMap;
import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import es.hol.audiolibros.Libroviewer.verificaUsuario;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Environment;
import android.preference.PreferenceManager;
import android.app.ListActivity;
import android.app.ProgressDialog;
import android.content.ContentValues;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.ActivityInfo;
import android.graphics.Bitmap;
import android.util.Base64;
import android.util.Log;
import android.view.ContextMenu;
import android.view.ContextMenu.ContextMenuInfo;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.WindowManager;
import android.widget.AdapterView;
import android.widget.AdapterView.AdapterContextMenuInfo;
import android.widget.EditText;
import android.widget.ImageButton;
import android.widget.ListAdapter;
import android.widget.ListView;
import android.widget.SimpleAdapter;
import android.widget.Toast;
import android.widget.AdapterView.OnItemClickListener;
```

```
public class Catalogo extends ListActivity {

        // Definir constantes
        private static final String TAG_ID     = "id";
        private static final String TAG_AUTOR  = "autor";
        private static final String TAG_TITULO = "titulo";
        private static final String TAG_TEMA   = "tema";
        private static final String TAG_URL    = "mp3";

        public static final String KEY_ROWID    = "_id";
        public static final String KEY_TITULO   = "titulo";
        public static final String KEY_AUTOR    = "autor";
        public static final String KEY_TEMA     = "tema";
        public static final String KEY_DESCRIP  = "descrip";
        public static final String KEY_MP3URL   = "mp3url";
        public static final String KEY_POSICION = "posicion";
        public static final String KEY_IMAGEN   = "imagen";

        // Variables de uso general

        // Un mapa es una estructura que nos permite almacenar pares clave/valor.
        // De tal manera que para una clave solamente tenemos un valor.
        ArrayList<HashMap<String, String>> jsonlist = new ArrayList<HashMap<String, String>>();
        ArrayList<HashMap<String, String>> array_sort = new ArrayList<HashMap<String, String>>();

        ListAdapter adapter;
        ListView lv ;

        // Objetos para busqueda
        EditText inputSearch;
        ImageButton buscar;

        // Nos indica si se ha utilizado algun filtro
        boolean filtro = false;

        private String server, usuario, clave;
        private String url, titulo, mp3url;

        private int _ID = 0;
        private boolean catalogo = false;

        private SharedPreferences prefs;
        private int result[];

        boolean registrado = false;
        boolean favoritos = false;
        boolean reproducir = false;

        // contiene datos de registro
        ArrayList<NameValuePair>  datos;


        @Override
        protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);

        // Fijar orientacion vertical
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

        setContentView(R.layout.catalogo);

        //Obtenemos las referencias a los controles
        inputSearch = (EditText) findViewById(R.id.inputSearch);
        lv        =  (ListView) findViewById(android.R.id.list);
        buscar     = (ImageButton) findViewById(R.id.ibtn_buscar);

        // Ocultamos el teclado virtual
        // Aparece al tener el foco un EditText
        getWindow().setSoftInputMode(WindowManager.LayoutParams.
                                                SOFT_INPUT_STATE_ALWAYS_HIDDEN);

        //Asociamos el menú contextual al control para el listview y el boton de busqueda
        registerForContextMenu(this.getListView());
        registerForContextMenu(buscar);

        // Obtenemos la lista de preferencias mediante el método getDefaultSharedPreferences() y
        // posteriormente utilizamos los distintos métodos get()
        // para recuperar el valor de cada opción dependiendo de su tipo.
        prefs = PreferenceManager.getDefaultSharedPreferences(Catalogo.this);

        server     = prefs.getString("server", "www.audiolibros.hol.es");
        usuario    = prefs.getString("user", "");
        clave      = prefs.getString("pass", "");
        registrado = prefs.getBoolean("REGISTRO", false);
        catalogo   = prefs.getBoolean("CATALOGO", false);
        boolean autologin  = prefs.getBoolean("autologin", true);

        // Construir url
        url = "http://"+server+"/WebService/catalogo.json";

        // Lanzar tarea asincrona de carga del catalogo
        new CargaCatalogo(Catalogo.this).execute();

        // Escuchador para pulsacion de un elemento de la lista
         lv.setOnItemClickListener(new OnItemClickListener() {
        public void onItemClick(AdapterView<?> parent, View view, int position, long id) {

          // forzamos mostrar el menu contextual
          openContextMenu(view);

        }
        });


    } // onCreate


     // MENU CONTEXTUAL
```

```
@Override
public void onCreateContextMenu(ContextMenu menu, View v, ContextMenuInfo menuInfo)
{
   super.onCreateContextMenu(menu, v, menuInfo);

   //  obtendremos una referencia al inflater mediante el método getMenuInflater()
   MenuInflater inflater = getMenuInflater();


   // para que se construya un menú distinto dependiendo del control asociado.
   // Esto lo haremos obteniendo el ID del control al que se va a asociar el menú contextual,
   // que se recibe en forma de parámetro (View v) en el evento onCreateContextMenu().
   // Utilizaremos para ello una llamada al método getId() de dicho parámetro

   if(v.getId() == android.R.id.list) // Para el listview
   {
         // convertimos el parámetro menuInfo a un objeto de tipo AdapterContextMenuInfo
         // menuInfo. Este parámetro contiene información adicional del control que se ha pulsado
         // para mostrar el menú contextual, y en el caso particular del control ListView contiene
         // la posición del elemento concreto de la lista que se ha pulsado.
      AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo)menuInfo;

      // personalizar el título del menú contextual mediante setHeaderTitle()
      // para que muestre el texto del elemento seleccionado en la lista
      // Obtener el id del item pulsado para usarlo posteriormente

      if (filtro){
         titulo = array_sort.get(info.position).get("titulo")+" - "+array_sort.get(info.position).get("autor");
         mp3url = array_sort.get(info.position).get("mp3");
         _ID = Integer.parseInt(array_sort.get(info.position).get("id").trim());

      }else {
         titulo = jsonlist.get(info.position).get("titulo")+" - "+jsonlist.get(info.position).get("autor");
         mp3url = jsonlist.get(info.position).get("mp3");
         _ID = Integer.parseInt(jsonlist.get(info.position).get("id").trim());

      }

          Log.i("Catalogo","id: "+String.valueOf(_ID));

         menu.setHeaderTitle("OPCIONES\n"+titulo);

         // generamos la estructura del menú llamando a su método infate() pasándole como parámetro
      // el ID del menu definido en XML
      inflater.inflate(R.menu.menu_ctx, menu);
   }

   if(v.getId() == R.id.ibtn_buscar) // para el imagebutton
   {

         // Si existen datos se activa el menu contextual para busquedas
         int tam = jsonlist.size();
                              if ( tam > 0){
                                          menu.setHeaderTitle("Filtrar por: ");
                     inflater.inflate(R.menu.filtrar_ctx, menu);
```

```
                    }
                    else{

            Utils.mensaje(getApplicationContext(),” No hay datos ”);

            finish();
                    }

    }

}


@Override
public boolean onContextItemSelected(MenuItem item) {

        // utilizamos la información del objeto AdapterContextMenuInfo para saber qué elemento
        // de la lista se ha pulsado,
        // lo obtenemos mediante una llamada al método getMenuInfo()
        //de la opción de menú (MenuItem) recibida como parámetro.
    AdapterContextMenuInfo info = (AdapterContextMenuInfo) item.getMenuInfo();

    // obtener datos para las opciones agregar a favoritos o reproducir.
    if((item.getItemId() == R.id.CtxLstOpc1) || (item.getItemId() == R.id.CtxLstOpc2))
    {
                // Si hay datos de registro verificar con el WebService
                if (registrado){

                        // ruta de la función del Web Service
                        String url = "http://"+server+"/WebService/validaUser.php";

                        //Rellenar datos de registro
            datos = new ArrayList<NameValuePair>();
                    datos.add(new BasicNameValuePair("usuario", usuario));
                    datos.add(new BasicNameValuePair("password", clave));
                    datos.add(new BasicNameValuePair("url", url));
                    datos.add(new BasicNameValuePair("pos", String.valueOf(info.position)));

          // Si no hay registro mostrar aviso
                }else {

              // Mostrar dialogo
        Utils.showAlertDialog(Catalogo.this, "AVISO",
                    "\nNo hay datos de registro.\nDebe registrarse.", false);
            }
    }

    switch (item.getItemId()) {

      case R.id.CtxLstOpc1: // Agregar a favoritos

        // Utils.mensaje(getApplicationContext(),"Pendiente de codificar.\n\nHas pulsado "+
        //                                    lv.getAdapter().getItem(info.position));

        favoritos = true;
```

```
            reproducir = false;

            // ejecutar tarea asincrona de validar usuario
                        verificaUsuario validar = new verificaUsuario(this,datos);
                        validar.execute();

            //agregar_favorito(_ID, info.position);

            return true;

        case R.id.CtxLstOpc2: // Reproducir

            // Comprobar conexion
            if (!Utils.verificaConexion(getApplicationContext())) {
                    // No hay conexion a Internet
            // Mostrar dialogo
      Utils.showAlertDialog(Catalogo.this, "Internet",
            "Comprueba tu conexión", false);

      return true;
}

            // algunas url contienen espacios y darían error
            mp3url = mp3url.replaceAll("\\s","%20");

            // comprobar url
                    result = Utils.CompruebaUrl(getApplicationContext(),mp3url);

                    if (result[0] == 200) {

                            reproducir = true;
                            favoritos = false;

                            // ejecutar tarea asincrona de validar usuario
                            verificaUsuario validar2 = new verificaUsuario(this,datos);
                                    validar2.execute();

                    }
                    else {
                       Utils.mensaje(getApplicationContext(),"Code: "+result[0] +
                                            "\nNo se encontró el audiolibro o\nno hay conexión.");
                    }

            return true;

        case R.id.CtxLstOpc3: // Salir

            return true;

        case R.id.Ctx_autor:

            // Obtener los datos filtrados por autor
                    filtrado("autor");

            return true;
```

```
                case R.id.Ctx_titulo:

            // Obtener los datos filtrados por titulo
                    filtrado("titulo");

            return true;

        case R.id.Ctx_tema:

            // Obtener los datos filtrados por tema
                    filtrado("tema");

            return true;

        case R.id.Ctx_qfiltro:

                    // Mostrar listview
                    ListAdapter adapter = new SimpleAdapter(this, jsonlist,R.layout.list_item,
                            new String[] { TAG_AUTOR, TAG_TITULO,TAG_TEMA, TAG_URL },
                            new int[] {R.id.autor, R.id.titulo, R.id.tema,R.id.url });

                            setListAdapter(adapter);

                            lv = getListView();

                            filtro = false;

                            inputSearch.setText("");

                            return true;

        case R.id.Ctx_salir:

            inputSearch.setText("");

            return true;

        default:
            return super.onContextItemSelected(item);
    }
}


            /*
            * Autentifica al usuario mediante funcion WebService
            *
            * Parámetro: context, contexto de la aplicación
            * Parámetro: datos, ArrayList con los datos
            * Retorno: resp, string con el valor devuelto por el WebService
            */

            // En nuestra tarea no necesitamos entrada,
            // no usaremos información de progreso y devolveremos un String.
            class verificaUsuario extends AsyncTask< Void, Void, String > {
```

```
        ProgressDialog Dialog;
        String cpass;

                Context context;
                ArrayList<NameValuePair> datos;
                public verificaUsuario(Context context, ArrayList<NameValuePair> datos){
                  this.context=context;
                  this.datos = datos;
                }

protected void onPreExecute() {

        //para el progress dialog
      Log.i("VerificaUsuario", "onPreExecute "+context);
      Log.i("VerificaUsuario", "onPre clave: "+datos.get(1).getValue().trim());

        Dialog = ProgressDialog.show(context,"", "Validando...");

        // ciframos la contraseña aplicandole una máscara (la misma que aplica el WebService)
                String masc = "|#€7`¬23ads4ook12";
            cpass = masc + Utils.Cifrado.cifrar(datos.get(1).getValue().trim(), "MD5");
            cpass = Utils.Cifrado.cifrar(cpass, "SHA-1");
            datos.set(1, new BasicNameValuePair("password", cpass));

            Log.i("VerificaUsuario", "onPre cclave: "+datos.get(1).getValue().trim());

            // Codificar en base64
            byte[] byteArray;
            try {
                byteArray = datos.get(0).getValue().trim().getBytes("UTF-8");
                datos.set(0, new BasicNameValuePair("usuario",
                                Base64.encodeToString(byteArray,Base64.DEFAULT)));
                byteArray = datos.get(1).getValue().trim().getBytes("UTF-8");
                datos.set(1, new BasicNameValuePair("password",
                                Base64.encodeToString(byteArray,Base64.DEFAULT)));

                } catch (UnsupportedEncodingException e) {
                        // TODO Auto-generated catch block
                        Log.i("VerificaUsuario", "Error: "+ e);
                        //e.printStackTrace();
                }

            Log.i("VerificaUsuario", "onPre b64clave: "+datos.get(1).getValue().trim());

}


        protected String doInBackground(Void... params) {

                String url  = datos.get(2).getValue().trim();

                // elimimar url de datos
                datos.set(2, new BasicNameValuePair("url", ""));
```

```
                // usamos una clase httpHandler para enviar una petición al servidor web.
                 Utils.httpHandler handler = new Utils.httpHandler();
                String resp = handler.post(url, datos);

                 Log.i("VericaUsuario","resp: "+ resp+"url: "+url);


                // valor devuelto para onPostExecute
                return resp;

            }

            /*Una vez terminado doInBackground comprobar lo que halla ocurrido
             * recibe resp
            */
        protected void onPostExecute(String resp) {

            Log.i("onPostExecute=","resp: "+resp);

                //validamos el resultado obtenido
                 if (resp.trim().contains("0")){

                        // Mostrar mensaje
                        Log.i("verifica0 ", "invalido");
                        Utils.mensaje(context, " Usuario o contaseña incorrectos ");
                }
                else if (resp.trim().contains("1")){

                        // Mostrar mensaje
                        Log.i("verifica1 ", "valido");
                        //Utils.mensaje(context, " Usuario validado ");

                        if (favoritos){
                                int pos = Integer.parseInt(datos.get(3).getValue().trim());
                                agregar_favorito(_ID, pos);
                        }
                        else if (reproducir){
                                // Crear un Intent, pasar parametros y lanzar una nueva Activity con el
reproductor
                                Intent i = new Intent(getApplicationContext(), MusicPlayerActivity.class);

                                // llamamos al método putExtra de la clase Intent. Tiene dos parámetros
                                // de tipo String, clave-valor en el primero indicamos el nombre del dato
                                //  y en el segundo el valor del dato:
                                 i.putExtra("titulo_autor", titulo);
                                 i.putExtra("url", mp3url);
                                 i.putExtra("desde", "catalogo");

                                startActivity(i);
                        }

                 }
                 else{
                        // Mostrar mensaje
```

```
                    Log.e("registerstatus ", "Error de sistema");
                    Utils.mensaje(context, " Error de sistema. \n Vuelva a intentarlo. ");
                }

                Dialog.dismiss();//ocultamos progess dialog.

        } // Onpostexecute



    } // verificaUsuario


// Añadir a la BD de favoritos los datos del libro seleccionado del catalogo
public void agregar_favorito(int id, int pos){

        // 1º Descargar imagen segun id y guardarla
        // 2º Comprimir imagen para la BD
        // 3º Asignar resto de datos

        byte[] imageInByte = null;
        FileOutputStream fos;

        // Comprobar que existe el recurso
        String url = "http://"+server.trim()+"/libros/fotolibro/"+String.valueOf(id).trim()+".jpg";
        result = Utils.CompruebaUrl(getBaseContext(),url);

        Log.i("Catalogo","url: "+url);
        Log.i("Catalogo","code: "+result[0]);

        if (result[0] != 200) {
                // No existe imagen, descargar generica
                url = "http://"+server.trim()+"/libros/fotolibro/"+"0.jpg";

                // forzar result
                result[0] = 200;

                Log.i("Catalogo1","url: "+url);
                Log.i("Catalogo1","code: "+result[0]); //String.valueOf(result[0]));
        }

        // Si es correcto
         if (result[0] == 200) {

                String Titulo, Autor, Tema, Descrip, Mp3url;
                Bitmap Imagen = Utils.DownloadImage.downloadImage(url);

                // Extraer el nombre de la imagen
                File file = new File(url.trim());
                String name = file.getName();

                // Definimos una array de Bytes
                ByteArrayOutputStream bytes = new ByteArrayOutputStream();

                // Elegimos el formato de compresion y calidad
```

```
Imagen.compress(Bitmap.CompressFormat.JPEG, 100, bytes);

// Crear drectorio si no existe
File storagePath = new File(Environment.getExternalStorageDirectory(),"Images_Audio");
storagePath.mkdirs();

// Crear el fichero en SD card
file = new File(Environment.getExternalStorageDirectory()+"/Images_Audio/"
        + File.separator + name);

try {
            file.createNewFile();
} catch (IOException e) {
   e.printStackTrace();
}

// Crear un nuevo FileOutputStream y escribir los bytes en un fichero

try {
            fos = new FileOutputStream(file);
   fos.write(bytes.toByteArray());
   fos.close();
   //Toast.makeText(this, "Imagen guardada", Toast.LENGTH_SHORT).show();
} catch (IOException e) {
   e.printStackTrace();
}

if (Imagen != null){
            // convert bitmap to byte and compres
            ByteArrayOutputStream stream = new ByteArrayOutputStream();
            Imagen.compress(Bitmap.CompressFormat.JPEG, 50, stream);
            imageInByte = stream.toByteArray();
}

// Obtener descripcion de la BD del servidor

// ruta de la función del Web Service
url = "http://"+server+"/WebService/db_descrip.php";

//Rellenar datos paara consulta
datos.clear();
  //datos = new ArrayList<NameValuePair>();
datos.add(new BasicNameValuePair("_id", String.valueOf(id)));

// usamos una clase httpHandler para enviar una petición al servidor web.
 Utils.httpHandler handler = new Utils.httpHandler();
String resp = handler.post(url, datos);

Log.i("Favorito","resp: "+ resp+"url: "+url);

Descrip = resp.trim();

if (filtro){
            Titulo  = array_sort.get(pos).get("titulo");
            Autor   = array_sort.get(pos).get("autor");
```

```
                    Tema    = array_sort.get(pos).get("tema");
                    //Descrip = array_sort.get(pos).get("descripcion");
                    Mp3url  = array_sort.get(pos).get("mp3");

            }else {
                    Titulo  = jsonlist.get(pos).get("titulo");
                    Autor   = jsonlist.get(pos).get("autor");
                    Tema    = jsonlist.get(pos).get("tema");
                    //Descrip = jsonlist.get(pos).get("descripcion");
                    Mp3url  = jsonlist.get(pos).get("mp3");
            }

            if (Tema == null){
                    Tema = "No definido";
            }

        if (Descrip == null){
            Descrip = "Sin descripción";
        }

        ContentValues Values = new ContentValues();

            Values.put(KEY_TITULO, Titulo);
            Values.put(KEY_AUTOR, Autor);
            Values.put(KEY_TEMA, Tema);
            Values.put(KEY_DESCRIP, Descrip);
            Values.put(KEY_MP3URL, Mp3url);
            Values.put(KEY_POSICION, 0);
            Values.put(KEY_IMAGEN, imageInByte);

        if (Imagen != null){

            // Agregar el elemento seleccionado a la BD
            DataBase db = new DataBase(this);
            boolean result = db.insertaLibro(Values);
            db.close();

            if (!result){
                    Utils.mensaje(getApplicationContext(),"\nError: No se pudo insertar a favoritos.");

            }
            else{
                    Utils.mensaje(getApplicationContext(),"\nLibro añadido a favoritos.\n");
            }

        }else{
            Utils.mensaje(getApplicationContext(),"\nError: No se pudo insertar a favoritos.");
        }

    }
    else {

        Utils.mensaje(getApplicationContext(),"Code: "+ result[0] +
                    "\nNo se pudo insertar o\nno hay conexión.");
    }
```

```
        }

// Método evento Onclick botones para filtro listview
public void filtrar(View v) {

        // filtro = true;

         switch (v.getId()) {

              case R.id.ibtn_buscar:

                          // forzamos mostrar el menu contextual al pulsar el boton
                    openContextMenu(v);
                    //openOptionsMenu();

                         break;
   } // switch

}



public void filtrado(String valor)
{
         // Obtener los datos filtrados por valor

        int textlength=0;

       textlength = inputSearch.getText().length();
      array_sort.clear();
      for (int i = 0; i < jsonlist.size(); i++)
      {
         if (textlength <= jsonlist.get(i).get(valor).length())
         {
                  // sin distingir may/min
            if(jsonlist.get(i).get(valor).toLowerCase().contains(
                          inputSearch.getText().toString().toLowerCase().trim()))
            {
               array_sort.add(jsonlist.get(i));
            }
          }
      }

      filtro = true;

      // Mostrar nuevo listview con los datos filtrados
         ListAdapter adapter = new SimpleAdapter(this, array_sort,R.layout.list_item,
```

```
                                        new String[] { TAG_AUTOR, TAG_TITULO,TAG_TEMA,
TAG_URL },
                                        new int[] {R.id.autor, R.id.titulo, R.id.tema,R.id.url });

        setListAdapter(adapter);

        lv = getListView();

    }

        // Clase asíncrona para cargar el catalogo en el listview

        private class CargaCatalogo extends AsyncTask<String, Void, Boolean> {

                private ProgressDialog dialog;
                private ListActivity activity;
                private Context context;

                public CargaCatalogo(ListActivity activity) {
                        this.activity = activity;
                        context = activity;
                        dialog = new ProgressDialog(context);
                }

                protected void onPreExecute() {

                        // Asignar titulo y mostrar progressdialog
                        // dependiendo si existe o hay que bajarlo
                        if (catalogo == false){
                                this.dialog.setTitle("Conexión con \nwww.audiolibros.hol.es");
                                this.dialog.setMessage("Obteniendo Datos\nConviritiendo Datos\nPuede durar...");
                                this.dialog.show();
                        }
                        else {
                                //this.dialog.setTitle("CATALOGO");
                                //this.dialog.setMessage("Conviritiendo Datos\nPuede durar...");
                                //this.dialog.show();

                                dialog = ProgressDialog.show(Catalogo.this, "CATALOGO", "Conviritiendo
Datos\nPuede durar...");
                        }

                }

                @Override
                protected void onPostExecute(final Boolean success) {
                        if (dialog.isShowing()) {
                                dialog.dismiss();
                        }

                        // Si no existen datos mostrar mensaje y salir
                        int tam = jsonlist.size();
                        if ( tam == 0){
                                // No hay conexion a Internet
                // Mostrar dialogo
```

```
//Utilities.showAlertDialog(Catalogo.this, "Internet","No existe Catálogo o no hay conexión", false);

Utils.mensaje(getApplicationContext(),"  No existe Catálogo o no hay conexión ");

try {
   Thread.sleep(2000);
} catch(InterruptedException ex) {
   Thread.currentThread().interrupt();
}

finish();
            }

            // Mostrar listview
            ListAdapter adapter = new SimpleAdapter(context, jsonlist,R.layout.list_item,
                                 new String[] { TAG_AUTOR, TAG_TITULO,TAG_TEMA, TAG_URL },
                                 new int[] {R.id.autor, R.id.titulo, R.id.tema,R.id.url });

            setListAdapter(adapter);

            lv = getListView();

  }

  protected Boolean doInBackground(final String... args) {

     // Si catalogo = true es que tenemos descargado el fichero en el movil
     // si false tenemos que descargarlo
     if(catalogo == false) {

                  result = Utils.CompruebaUrl(getBaseContext(),url);

                  // Si es correcto
               if (result[0] == 200) {

                  // descargamos catalogo en la SD
                  Utils.descarga_file(getBaseContext(), url);

                  // Como existe en la SD cambiar la url
                  //url = Environment.getExternalStorageDirectory()+"/Cat_Audio/catalogo.json";
                  // Como existe en la memoria cambiar la url
                  url = context.getFilesDir()+"/Cat_Audio/catalogo.json";
                  Log.d("URL", url);

                  // Verificar que el catalogo existe
                  File file = new File(url);

                  if(file.exists()) {
                        Log.d("File", "Existe");
                              // Actualizar preferencias
                        //Obtenemos el editor de las preferencias.
                  SharedPreferences.Editor editor = prefs.edit();

                  // almacenar un booleano CATALOGO con valor true.
                  // y un int con el tamaño del catalogo
```

```
                editor.putBoolean("CATALOGO", true);
                editor.putInt("TAMANO", result[1]);

                // Tras haber indicado los cambios a realizar (en nuestro caso dos),
                // le indicamos al editor que los almacene en las preferencias.
                editor.commit();
                }
            }

        }
        else {

            // Como existe en la SD cambiar la url
            //url = Environment.getExternalStorageDirectory()+"/Cat_Audio/catalogo.json";
            //url = Environment.getDataDirectory()+"/Cat_Audio/catalogo.json";
            url = context.getFilesDir()+"/Cat_Audio/catalogo.json";
            Log.d("URL", url);

        }

        // Obtener los datos del archivo json y crear la lista
        JSONParser jParser = new JSONParser();

        // Obtener el archivo .json y asignarlo al objeto array
        //JSONArray json = jParser.getJSONFromUrl(url);

        // crear el json desde el archivo de la SD
        JSONArray json = jParser.getJSONFromFile(url);

        if (json == null){

            return null;
        }

        for (int i = 0; i < json.length(); i++) {

            try {
                JSONObject c = json.getJSONObject(i);
                String vid = c.getString(TAG_ID);
                String vautor = c.getString(TAG_AUTOR);
                String vtitulo = c.getString(TAG_TITULO);
                String vtema = c.getString(TAG_TEMA);
                String vurl = c.getString(TAG_URL);

                HashMap<String, String> map = new HashMap<String, String>();

                // añadimos cada nodo al HashMap clave => valor
                map.put(TAG_ID, vid);
                map.put(TAG_AUTOR, vautor);
                map.put(TAG_TITULO, vtitulo);
                map.put(TAG_TEMA, vtema);
                map.put(TAG_URL, vurl);

                // añadimos el nodo creado al array jsonlist
                jsonlist.add(map);
```

```
                    } catch (JSONException e) {
                        // TODO Auto-generated catch block
                        e.printStackTrace();
                    }
                }

                return null;
            }
        }

        @Override
        public void onDestroy(){
        super.onDestroy();


        }


}
```

Configuracion.java

```
package es.hol.audiolibros;

import android.app.AlertDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.SharedPreferences;
import android.content.SharedPreferences.OnSharedPreferenceChangeListener;
import android.content.pm.ActivityInfo;
import android.os.Bundle;
import android.preference.PreferenceActivity;
import android.preference.PreferenceManager;
import android.util.Log;
import android.view.WindowManager;


// Las preferencias no son más que datos que una aplicación debe guardar para personalizar la experiencia del usuario,
// por ejemplo información personal, opciones de presentación, etc.
// Cada preferencia se almacenará en forma de clave-valor, es decir, cada una de ellas estará compuesta
// por un identificador único (p.e. "email") y un valor asociado a dicho identificador (p.e. "prueba@email.com").
// Además, los datos se guardan en ficheros XML.
// al extender a PreferenceActivity, se encargará por nosotros de crear la interfaz gráfica de nuestra lista de opciones
// según hemos la definido en el XML y se preocupará por nosotros de mostrar, modificar y guardar las opciones cuando sea
// necesario tras la acción del usuario haciendo uso de la API de preferencias compartidas (Shared Preferences)

public class Configuracion extends PreferenceActivity implements OnSharedPreferenceChangeListener{

        SharedPreferences prefs;
        Context context;

        String servidor, usuario, clave, correo;

        public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    // Fijar orientacion vertical
    setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

    // Ocultamos el teclado virtual
        // Aparece al tener el foco un EditText
        getWindow().setSoftInputMode(WindowManager.LayoutParams.SOFT_INPUT_STATE_ALWAYS_HIDDEN);

    context = this; //getApplicationContext(); getBaseContext();
    prefs = PreferenceManager.getDefaultSharedPreferences(context);

    // obtener valores para verificar cambios
    servidor = prefs.getString("server", "");
    usuario = prefs.getString("user", "");
    clave = prefs.getString("pass", "");
    correo = prefs.getString("mail", "");

    // llamada al método addPreferencesFromResource(), mediante el que indicaremos el fichero XML
    // en el que hemos definido la pantalla de opciones
```

```
    addPreferencesFromResource(R.xml.configuracion);
}

        // Método llamado cada vez que un cambio es confirmado
        // Lo utilizamos para verificar el contenido de los parámetros
        @Override
        public void onSharedPreferenceChanged(SharedPreferences prefs, String key) {

                // activar editor de preferencias
                SharedPreferences.Editor prefEditor = prefs.edit();


                Log.i("Config","key: "+key);


                // Validar server
            if(key.equals("server")){

                String valor = prefs.getString("server", "");

                Log.i("Config","key if: "+key);

                if (valor.equals("")){

                        showErrorDialog(context,"Server vacio.\n\nSe asignara el server por defecto.",false);
                        Log.i("Config","key vacio: "+key+" "+valor);

                        // Asignar valor por defecto
                        prefEditor.putString("server", "www.audiolibros.hol.es");
                  prefEditor.commit();
                  // reload();
                }

                else if (!valor.equals(servidor)){

                                showErrorDialog(context,"Server desconocido.\n\nSe asignara el server por
defecto.",false);
                                Log.i("Config","key vacio: "+key+" "+valor);

                                // Asignar valor por defecto
                                prefEditor.putString("server", "www.audiolibros.hol.es");
                        prefEditor.commit();
                        // reload();
                    }

            }

            // Validar usuario
            if(key.equals("user")){

                String valor = prefs.getString("user", "");

                Log.i("Config","key if: "+key);

                if (valor.equals("") || !valor.equals(usuario)){
```

```
                    showErrorDialog(context,"Modificar valores.\n\nSi no coinciden con los datos de registro, la "
+
                                "aplicación no será operativa.",false);
                    Log.i("Config","key vacio: "+key+" "+valor);

            }

        }


        // Validar contraseña
        if(key.equals("pass")){

            String valor = prefs.getString("pass", "");

            Log.i("Config","key if: "+key);

            if (valor.equals("") || !valor.equals(clave)){

                    showErrorDialog(context,"Modificar valores.\n\nSi no coinciden con los datos de registro, la "
+
                                "aplicación no será operativa.",false);
                    Log.i("Config","key vacio: "+key+" "+valor);

            }

        }

        // Validar correo
        if(key.equals("mail")){

            String valor = prefs.getString("mail", "");

            Log.i("Config","key if: "+key);

            if (valor.equals("") || !valor.equals(correo)){

                    showErrorDialog(context,"Modificar valores.\n\nSi no coinciden con los datos de registro, la "
+
                                "aplicación no será operativa.",false);
                    Log.i("Config","key vacio: "+key+" "+valor);

            }

        }


        // Registro manual
        if(key.equals("reg_man")){

            boolean valor = prefs.getBoolean("reg_man", false);

            Log.i("Config","key if: "+key);
```

```
        if (valor){

                                showErrorDialog(context,"Modificar valores.\n\nSi no coinciden con los datos de
                                registro, la " +"aplicación no será operativa.",false);

                        // Editar valor
                        prefEditor.putBoolean("REGISTRO", true );
          prefEditor.commit();

                        Log.i("Config","key vacio: "+key+" "+valor);

        } else {

            // Editar valor
                        prefEditor.putBoolean("REGISTRO", false );
                    prefEditor.commit();

        }

    }

    // Reiniciar datos entorno
    if(key.equals("entorno")){

        boolean valor = prefs.getBoolean("entorno", false);

        Log.i("Config","key if: "+key);

        if (valor){

                        showErrorDialog(context,"Reiniciar valores.\n\nSe borrarán los datos de entorno.",false);

                        // Editar valor
                        prefEditor.putBoolean("CATALOGO", false );
                         prefEditor.putInt("TAMANO", 0 );
                         prefEditor.commit();

                        Log.i("Config","key vacio: "+key+" "+valor);

        } else {

        }

    }


    return;


}


private void showErrorDialog(Context context,String errorString, boolean status){
  //String okButtonString = context.getString(R.string.ok_name);
```

```
        AlertDialog.Builder ad = new AlertDialog.Builder(context);
        ad.setTitle("AVISO");
        // asignar icono
        ad.setIcon((status) ? R.drawable.success : R.drawable.fail);
        ad.setMessage(errorString);
        ad.setPositiveButton("Ok",new DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int arg1) {
                    reload();
                }
                } );

        //ad.create();
        ad.show();
        return;

    }


    // relanzar la actividad usando el mismo intent
    private void reload(){
        startActivity(getIntent());
        finish();
    }



    // Para obtener el método de notificación, hay que registrar tu actividad como un escuchador válido.
    // La mejor forma es registrarse en onResume y desregistrarse en onPause:
    @Override
    protected void onResume() {
                super.onResume();
            prefs.registerOnSharedPreferenceChangeListener(this);
    }

        @Override
    protected void onPause() {
                super.onPause();
                prefs.unregisterOnSharedPreferenceChangeListener(this);
    }


}



/*

La API para el manejo de estas preferencias es muy sencilla. Toda la gestión se centraliza en la clase SharedPrefences,
que representará a una colección de preferencias. Una aplicación Android puede gestionar varias colecciones de preferencias,
que se diferenciarán mediante un identificador único. Para obtener una referencia a una colección determinada utilizaremos
el método getSharedPrefences() al que pasaremos el identificador de la colección y un modo de acceso.
El modo de acceso indicará qué aplicaciones tendrán acceso a la colección de preferencias y qué operaciones
tendrán permitido realizar sobre ellas. Así, tendremos tres posibilidades principales:
```

MODE_PRIVATE. Sólo nuestra aplicación tiene acceso a estas preferencias.
MODE_WORLD_READABLE. Todas las aplicaciones pueden leer estas preferencias, pero sólo la nuestra puede modificarlas.
MODE_WORLD_WRITABLE. Todas las aplicaciones pueden leer y modificar estas preferencias.

Para actualizar o insertar nuevas preferencias el proceso será igual de sencillo,
con la única diferencia de que la actualización o inserción no la haremos directamente sobre el objeto SharedPreferences,
sino sobre su objeto de edición SharedPreferences.Editor. A este último objeto accedemos mediante el método edit()
de la clase SharedPreferences. Una vez obtenida la referencia al editor, utilizaremos los métodos put correspondientes
al tipo de datos de cada preferencia para actualizar/insertar su valor, por ejemplo putString(clave, valor),
para actualizar una preferencia de tipo String. De forma análoga a los métodos get que ya hemos visto,
tendremos disponibles métodos put para todos los tipos de datos básicos: putInt(), putFloat(), putBoolean(), etc.
Finalmente, una vez actualizados/insertados todos los datos necesarios llamaremos al método commit() para confirmar los cambios.

```
SharedPreferences prefs = getSharedPreferences("MisPreferencias",Context.MODE_PRIVATE);

SharedPreferences.Editor editor = prefs.edit();
editor.putString("email", "modificado@email.com");
editor.putString("nombre", "Prueba");
editor.commit();
```

¿Pero donde se almacenan estas preferencias compartidas?
Como dijimos al comienzo del artículo, las preferencias no se almacenan en ficheros binarios como las bases de datos SQLite,
sino en ficheros XML. Estos ficheros XML se almacenan en una ruta que sigue el siguiente patrón:

/data/data/paquete.java/shared_prefs/nombre_coleccion.xml

*/

DataBase.java

```
package es.hol.audiolibros;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.database.sqlite.SQLiteDatabase.CursorFactory;
import android.util.Log;

// Esta clase se encarga de abrir la base de datos si existe o de crearla si no existe.
// Incluso de actualizar la versión si decidimos crear una nueva estructura de la base de datos.
// Para crear la base de datos crearemosuna  subclasede  SQLiteOpenHelper  y sobreescribiremos el método
// onCreate().  Opcionalmente también se puede implemenar el método  onUpgrade() y  onOpen().
// El método  onCreate()se  utiliza para crear la BD la primera vez. En el proceso de creación usaremos
// sentencias SQL como CREATE TABLE.
// El método onUpgrade() se utiliza para cambiar de versión, y lo normal es realizar un proceso de migración de
// datos, antes de hacer un DROP TABLE.

public class DataBase extends SQLiteOpenHelper{

		// Definir constantes
		public static final String KEY_ROWID    = "_id";
		public static final String KEY_TITULO   = "titulo";
		public static final String KEY_AUTOR    = "autor";
		public static final String KEY_TEMA     = "tema";
		public static final String KEY_DESCRIP  = "descrip";
		public static final String KEY_MP3URL   = "mp3url";
		public static final String KEY_POSICION = "posicion";
		public static final String KEY_IMAGEN   = "imagen";

		private static final String TAG = "AdaptadorBD";

		private static final String DATABASE_NAME    = "dbfavoritos";
		private static final String DATABASE_TABLE   = "libros";
		private static final int    DATABASE_VERSION = 1;

		private static final String CREATE_BDD =
		"create table "+DATABASE_TABLE+
		"("+KEY_ROWID+   " integer primary key autoincrement, "
		  +KEY_TITULO+  " text not null, "
		  +KEY_AUTOR+   " text not null, "
		  +KEY_TEMA+    " text not null, "
		  +KEY_DESCRIP+ " text not null, "
		  +KEY_MP3URL+  " text not null, "
		  +KEY_POSICION+" integer, "
		  +KEY_IMAGEN+  " BLOB);";

	//Array de strings para su uso en los diferentes métodos
	private String[] todasColumnas =new String[] {KEY_ROWID,KEY_TITULO,KEY_AUTOR,KEY_TEMA,
		KEY_DESCRIP,KEY_MP3URL, KEY_POSICION,KEY_IMAGEN};
```

```
        // Definir objeto
        private SQLiteDatabase bsSql;

        // Constructor
        public DataBase (Context context) {
                        super(context, DATABASE_NAME, null, DATABASE_VERSION);
        }



        @Override
        public void onCreate(SQLiteDatabase db) {

                //crear tabla a partir de los datos de la variable CREATE_BDD
                try{
                        //ejecuta la sentencia SQL de creación de la BD
                        db.execSQL(CREATE_BDD);
                }catch(SQLException e){
                                e.printStackTrace();
                }
        }

        @Override
        public void onUpgrade(SQLiteDatabase db, int oldVersion,int newVersion){
                        Log.w(TAG, "Actualizando base de datos de la versión " + oldVersion
                        + " a "
                        + newVersion + ", borraremos todos los datos");
                        //elimina tabla de la BD
                        db.execSQL("DROP TABLE IF EXISTS " + DATABASE_TABLE);
                        //crea la nueva BD
                        onCreate(db);
        }


    /**
    * METODOS PARA GESTIONAR LA BASE DE DATOS
    */

        //-- INSERTAR
        //inserta una fila en la BD a partir de los datos de un libro, mediante el método insert().
        void insertarLibro(String Titulo,String Autor,String Tema,String Descrip,String Mp3url, int Posicion, byte[]
Imagen){

                SQLiteDatabase db = this.getWritableDatabase();

                ContentValues Values = new ContentValues();

                Values.put(KEY_TITULO, Titulo);
                Values.put(KEY_AUTOR, Autor);
                Values.put(KEY_TEMA, Tema);
                Values.put(KEY_DESCRIP, Descrip);
                Values.put(KEY_MP3URL, Mp3url);
                Values.put(KEY_POSICION, Posicion);
```

```
            Values.put(KEY_IMAGEN, Imagen);

            // sentencia INSERT a la BD para insertar una fila con los valores initialValues
            db.insert(DATABASE_TABLE, null, Values);

            //db.close();

        }


        //actualiza los datos del libro identificado por numero, con los nuevos valores
        //pasados como parámetros,mediante el método  update().
        public boolean insertaLibro(ContentValues args ){

            SQLiteDatabase db = this.getWritableDatabase();

            // sentencia INSERT a la BD para insertar una fila con los valores args
            return db.insert(DATABASE_TABLE, null, args) > 0;
        }

        //-- MODIFICAR
        //actualiza los datos del libro identificado por numero, con los nuevos valores
        //pasados como parámetros,mediante el método  update().
        public boolean actualizarLibro(int Id, String Titulo, String Autor, String Tema, String Mp3url, String Descrip,
                                                                                                    int Posicion, byte[]
Imagen){
            SQLiteDatabase db = this.getWritableDatabase();

            ContentValues args = new ContentValues();

            args.put(KEY_TITULO, Titulo);
            args.put(KEY_AUTOR, Autor);
            args.put(KEY_TEMA, Tema);
            args.put(KEY_MP3URL, Mp3url);
            args.put(KEY_DESCRIP, Descrip);
            args.put(KEY_POSICION, Posicion);
            args.put(KEY_IMAGEN, Imagen);

            //manda una sentencia UPDATE a la BD para modificar el contacto identifiado por id
            return db.update(DATABASE_TABLE, args,KEY_ROWID + "=" + Id, null) > 0;
        }

        //actualiza los datos del libro identificado por numero, con los nuevos valores
        //pasados como parámetros,mediante el método  update().
        public boolean actualizaLibro(int Id, ContentValues args ){

            SQLiteDatabase db = this.getWritableDatabase();

            //manda una sentencia UPDATE a la BD para modificar el contacto identifiado por id
            return db.update(DATABASE_TABLE, args,KEY_ROWID + "=" + Id, null) > 0;
        }

        //-- ELIMINAR
        //elimina el libro identificado por numero, mediante el método delete().
        public boolean borrarLibro(long numero){
```

```
        SQLiteDatabase db = this.getWritableDatabase();

        //manda una sentencia DELETE a la BD para eliminar la fila identificada por numero
        return db.delete(DATABASE_TABLE, KEY_ROWID + "=" + numero, null) > 0;
    }


    //--CONSULTAR
    //consulta a la BD para obtener todos los libros usando el método query().
    public Cursor getTodosLibros() {

         SQLiteDatabase db = this.getWritableDatabase();

        return db.query(DATABASE_TABLE, todasColumnas,null,null,null,null,null);
    }

    //consulta de un libro por 'id' (clave primaria)
    public Cursor getLibro(int Id) throws SQLException{

        SQLiteDatabase db = this.getReadableDatabase();

        Cursor mCursor = db.query(true, DATABASE_TABLE, todasColumnas,
                                  KEY_ROWID + "=" + Id,null,null,null,null,null);
        //si hay datos devueltos, apunta al principio
        if (mCursor != null)  mCursor.moveToFirst();

        return mCursor;
    }

    // Obtener cantidad de libros
    public int getLibrosCount() {

        String countQuery = "SELECT * FROM " + DATABASE_TABLE;

        SQLiteDatabase db = this.getReadableDatabase();

        Cursor cursor = db.rawQuery(countQuery, null);
        int cant = cursor.getCount();
        cursor.close();

        // return count
        return cant;
    }

}
```

Libros.java

```
package es.hol.audiolibros;

public class Libros {

	int Id;
	String Titulo;
	String Autor;
	String Tema;
	String Descrip;
	String Mp3url;
	int Posicion;
	byte[] Imagen;

	Libros (int _Id,String _Titulo,String _Autor,String _Tema,String _Descrip,String _Mp3url, int _Posicion, byte[]
_Imagen){

		Id = _Id;
		Titulo = _Titulo;
		Autor = _Autor;
		Tema = _Tema;
		Descrip = _Descrip;
		Imagen = _Imagen;
		Mp3url = _Mp3url;
		Posicion = _Posicion;

	}

	Libros (String _Titulo,String _Autor,String _Tema,String _Descrip,String _Mp3url, int _Posicion, byte[] _Imagen){

		Titulo = _Titulo;
		Autor = _Autor;
		Tema = _Tema;
		Descrip = _Descrip;
		Mp3url = _Mp3url;
		Posicion = _Posicion;
		Imagen = _Imagen;

	}

	Libros (){

		Titulo = null;
		Autor = null;
		Tema = null;
		Descrip = null;
		Mp3url = null;
		Posicion = 0;
		Imagen = null;

	}

	public void setId(int _Id){
```

```
        Id = _Id;
}

public void setTitulo(String _Titulo){
        Titulo = _Titulo;
}

public void setAutor (String _Autor){
        Autor = _Autor;
}

public void setTema (String _Tema){
        Tema = _Tema;
}

public void setDescrip(String _Descrip){
        Descrip = _Descrip;
}

public void setMp3url(String _Mp3url){
        Mp3url = _Mp3url;
}


public void setPosicion(int _Posicion){
        Posicion = _Posicion;
}

public void setImagen(byte[] _Imagen){
        Imagen = _Imagen;
}


public int getId(){
        return Id;
}

public String getTitulo(){
        return Titulo;
}

public String getAutor (){
        return Autor;
}

public String getTema (){
        return Tema;
}

public String getDescrip(){
        return Descrip;
}

public String getMp3url(){
        return Mp3url;
```

```
        }

        public int getPosicion(){
                return Posicion;
        }

        public byte[] getImagen(){
                return Imagen;
        }

}
```

Favoritos.java

```
package es.hol.audiolibros;


import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.ActivityInfo;
import android.database.Cursor;
import android.os.AsyncTask;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.util.Base64;
import android.util.Log;
import android.view.ContextMenu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.view.ContextMenu.ContextMenuInfo;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.GridView;
import android.widget.TextView;


public class Favoritos extends Activity {

	// Definir constantes
	public static final int KEY_ROWID    = 0;
	public static final int KEY_TITULO   = 1;
	public static final int KEY_AUTOR    = 2;
	public static final int KEY_TEMA     = 3;
	public static final int KEY_DESCRIP  = 4;
	public static final int KEY_MP3URL   = 5;
	public static final int KEY_POSICION = 6;
	public static final int KEY_IMAGEN   = 7;

	// Definir variables de uso general
	public Cursor cur;

	private int _ID;                        // id del registro del item pulsado (puede no coincidir)

	GridView gridview;
	ImageAdapter adapter;
	private int result[];
	private String titulo, mp3url;
```

```
private int pos, id;
byte[] imageInByte = null;

/** Called when the activity is first created. */
@Override
 public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        // fijar de manera permanente la orientación en modo vertical
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

         setContentView(R.layout.favoritos);

        // crear el gridview a partir del elemento del xml gridview
        GridView gridview = (GridView) findViewById(R.id.gridview);

         //Asociamos el menú contextual al control para el gridview
         this.registerForContextMenu(gridview);

         // con setAdapter se llena el gridview con datos. en este caso un nuevo objeto
         // de la clase ImageAdapter, que está definida en otro archivo.
         adapter = new ImageAdapter(this);
         gridview.setAdapter(adapter);

        // para que detecte la pulsación se le añade un listener de itemClick
         // que recibe un OniTemClickListener creado con new
         gridview.setOnItemClickListener(new OnItemClickListener() {
        // dentro de este listener difinimos la función que se ejecuta al hacer click en un item.
         // el metodo pertenece a AdapterView, es decir, es AdapterView.OnItemClickListener
        // dentro de este, tenemos el método onItemClick, que es el que se invoca al pulsar
        // un item del AdapterView, esa función recibe el objeto padre, que es un adapterview en el
        // que se ha pulsado, una vista, que es el elemento sobre el que se ha pulsado,
        // una posicion, que es la posicion del elemento dentro del adapter,
         // y un id, que es el id de fila del item que se ha pulsado
        public void onItemClick(AdapterView<?> parent, View v, int position, long id) {

                // Referenciar y obtener el contenido del textview que almacena el id del registro de la BD
                // Es necesario dado que la posicion del item no puede coincider con el id del registro
                _ID = Integer.parseInt(((TextView) v.findViewById(R.id.gv_id)).getText().toString().trim());

                Log.i("Favoritos item/id",position+" "+String.valueOf(_ID));

                // forzamos mostrar el menu contextual
                openContextMenu(v);

                // mostrar una nueva actividad con los datos ampliados
                //Intent intent=new Intent(Favoritos.this,Libroviewer.class);
                //intent.putExtra("ID", _ID);
                //startActivity(intent);

        }
        });

 } // Oncreate
```

```
// MENU CONTEXTUAL

    @Override
    public void onCreateContextMenu(ContextMenu menu, View v, ContextMenuInfo menuInfo)
    {
      super.onCreateContextMenu(menu, v, menuInfo);

          // obtendremos una referencia al inflater mediante el método getMenuInflater()
          MenuInflater inflater = getMenuInflater();

          // convertimos el parámetro menuInfo a un objeto de tipo AdapterContextMenuInfo
          // menuInfo. Este parámetro contiene información adicional del control que se ha pulsado
          // para mostrar el menú contextual, y en el caso particular del control ListView contiene
          // la posición del elemento concreto de la lista que se ha pulsado.
        AdapterView.AdapterContextMenuInfo info = (AdapterView.AdapterContextMenuInfo)menuInfo;

        // Obtener datos necesarios de la BD
          cur = obtieneCursor(_ID);
          titulo = cur.getString(KEY_TITULO).trim()+" - "+cur.getString(KEY_AUTOR).trim();
          cur.close();

        // personalizar el título del menú contextual mediante setHeaderTitle()
        menu.setHeaderTitle("OPCIONES\n"+titulo);

          // generamos la estructura del menú llamando a su método infate() pasándole como parámetro
        // el ID del menu definido en XML
        inflater.inflate(R.menu.favoritos_ctx, menu);

        // Obtener id del elemento pulsado
        // Referenciar y obtener el contenido del textview que almacena el id del registro de la BD
          int _ID2 = Integer.parseInt(((TextView) v.findViewById(R.id.gv_id)).getText().toString().trim());
          Log.d("Favoritos ctxmenu", String.valueOf(_ID2));

          Log.i("Favoritos pos",String.valueOf(info.position));

    }


    @Override
    public boolean onContextItemSelected(MenuItem item) {

          // utilizamos la información del objeto AdapterContextMenuInfo para saber qué elemento de la lista se ha
                  // pulsado, lo obtenemos mediante una llamada al método getMenuInfo() de la opción de menú
(MenuItem)                  // recibida como parámetro.
///     AdapterContextMenuInfo info = (AdapterContextMenuInfo) item.getMenuInfo();


      switch (item.getItemId()) {

          case R.id.CtxGvOpc0: // Detalles
```

```
                // mostrar una nueva actividad con los datos ampliados
    Intent intent=new Intent(Favoritos.this,Libroviewer.class);
    intent.putExtra("ID", _ID);
    startActivity(intent);

                return true;

  case R.id.CtxGvOpc1: // Eliminar

     // Eliminar el elemento seleccionado de la BD
     DataBase db = new DataBase(this);
     db.borrarLibro(_ID);
     db.close();

     // Notificar al adaptador que hay cambios
     adapter.notifyDataSetChanged();

     // Mensaje
    Utils.mensaje(getApplicationContext(),"\nLibro eliminado de favoritos.\n");

    return true;

  case R.id.CtxGvOpc2: // Reproducir

    //Obtiene el objeto de preferencias de la aplicacion
    SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences(Favoritos.this);

    // Obtener datos desde preferencias
    String usuario = prefs.getString("user", "");
       String clave   = prefs.getString("pass", "");
       String server  = prefs.getString("server", "www.audiolibros.hol.es");
    boolean registrado = prefs.getBoolean("REGISTRO", false);

    // Comprobar conexion
    if (!Utils.verificaConexion(this)){
                // No hay conexion a Internet
                // Mostrar dialogo
    Utils.showAlertDialog(Favoritos.this, "Internet", "Comprueba tu conexión", false);

    return true;
    }

    // Obtener datos necesarios de la BD
    cur = obtieneCursor(_ID);
    mp3url          = cur.getString(KEY_MP3URL);
    titulo          = cur.getString(KEY_TITULO).trim()+" - "+cur.getString(KEY_AUTOR).trim();
    pos             = cur.getInt(KEY_POSICION);
    id              = cur.getInt(KEY_ROWID);
    imageInByte = cur.getBlob(KEY_IMAGEN);
    cur.close();

    Log.i("Favoritos play",mp3url);
    Log.i("Favoritos play",titulo);
```

```
        // algunas url contienen espacios y darían error
        mp3url = mp3url.replaceAll("\\s","%20");

        // Comprobar que existe la url
        result = Utils.CompruebaUrl(getBaseContext(),mp3url);

        if (result[0] == 200) {

                // Si hay datos de registro verificar con el WebService
                if (registrado){

                        // contiene datos de registro
                        ArrayList<NameValuePair>  datos;

                          // ruta de la función del Web Service
                          String url = "http://"+server+"/WebService/validaUser.php";

                           //Rellenar datos de registro
                                  datos = new ArrayList<NameValuePair>();
                                  datos.add(new BasicNameValuePair("usuario", usuario));
                                  datos.add(new BasicNameValuePair("password", clave));
                                  datos.add(new BasicNameValuePair("url", url));

                                  // ejecutar tarea asincrona de validar usuario
                                  verificaUsuario validar = new verificaUsuario(this,datos);
                                  validar.execute();


                   // Si no hay registro mostrar aviso
                }else {

                        // Mostrar dialogo
                           Utils.showAlertDialog(Favoritos.this, "AVISO",
                                  "\nNo hay datos de registro.\nDebe registrarse.", false);
                     }

                }
                else {

                        Utils.mensaje(getApplicationContext(),"Code: "+result[0]+
                                        "\nNo se encontró el audiolibro o " + "\nno hay conexión.");

                }

        return true;

      case R.id.CtxGvOpc3: // Salir

        return true;

      default:
        return super.onContextItemSelected(item);
    }
}
```

```
/*
         * Autentifica al usuario mediante funcion WebService
         *
         * Parámetro: context, contexto de la aplicación
         * Parámetro: datos, ArrayList con los datos
         * Retorno: resp, string con el valor devuelto por el WebService
        */

        // En nuestra tarea no necesitamos entrada,
        // no usaremos información de progreso y devolveremos un String.
        class verificaUsuario extends AsyncTask< Void, Void, String > {

                ProgressDialog Dialog;
                String cpass;

                Context context;
                ArrayList<NameValuePair> datos;
                public verificaUsuario(Context context, ArrayList<NameValuePair> datos){
                  this.context=context;
                  this.datos = datos;
        }

    protected void onPreExecute() {

        //para el progress dialog
         Log.i("VerificaUsuario", "onPreExecute "+context);
        Log.i("VerificaUsuario", "onPre clave: "+datos.get(1).getValue().trim());

        //Dialog = ProgressDialog.show(context,"VALIDACION", "Comprobando datos de usuario");

        Dialog = new ProgressDialog(Favoritos.this);
      Dialog.setMessage("Validando....");
      Dialog.setIndeterminate(false);
      Dialog.setCancelable(false);
      Dialog.show();

                // ciframos la contraseña aplicandole una máscara (la misma que aplica el WebService)
                String masc = "|#€7`¬23ads4ook12";
            cpass = masc + Utils.Cifrado.cifrar(datos.get(1).getValue().trim(), "MD5");
            cpass = Utils.Cifrado.cifrar(cpass, "SHA-1");
            datos.set(1, new BasicNameValuePair("password", cpass));

            Log.i("VerificaUsuario", "onPre cclave: "+datos.get(1).getValue().trim());

            // Codificar en base64
            byte[] byteArray;
            try {
                byteArray = datos.get(0).getValue().trim().getBytes("UTF-8");
                datos.set(0, new BasicNameValuePair("usuario",
                                Base64.encodeToString(byteArray,Base64.DEFAULT)));
                byteArray = datos.get(1).getValue().trim().getBytes("UTF-8");
```

```
                    datos.set(1, new BasicNameValuePair("password",
                                    Base64.encodeToString(byteArray,Base64.DEFAULT)));

                    } catch (UnsupportedEncodingException e) {
                            // TODO Auto-generated catch block
                            Log.i("VerificaUsuario", "Error: "+ e);
                            //e.printStackTrace();
                    }


                Log.i("VerificaUsuario", "onPre b64clave: "+datos.get(1).getValue().trim());


        }


    protected String doInBackground(Void... params) {

            String url  = datos.get(2).getValue().trim();

            // elimimar url de datos
            datos.set(2, new BasicNameValuePair("url", ""));

            // usamos una clase httpHandler para enviar una petición al servidor web.
            Utils.httpHandler handler = new Utils.httpHandler();
                String resp = handler.post(url, datos);

                Log.i("VericaUsuario","resp: "+ resp+"url: "+url);

                // valor devuelto para onPostExecute
                    return resp;

    }

    /*Una vez terminado doInBackground comprobar lo que halla ocurrido
    * recibe resp
    */
     protected void onPostExecute(String resp) {

            Log.i("onPostExecute=","resp: "+resp);

            //validamos el resultado obtenido
             if (resp.trim().contains("0")){

                    // Mostrar mensaje
                    Log.i("verifica0 ", "invalido");
                    Utils.mensaje(context, " Usuario o contaseña incorrectos ");

             }
             else if (resp.trim().contains("1")){

                    // Mostrar mensaje
                    Log.i("verifica1 ", "valido");
                    //Utils.mensaje(context, " Usuario validado ");
```

```
                    // Crear un Intent, pasar parametros y lanzar una nueva Activity con el reproductor

                    Intent i = new Intent(getApplicationContext(), MusicPlayerActivity.class);

                    // llamamos al método putExtra de la clase Intent. Tiene dos parámetros de tipo String,
                    // clave-valor en el primero indicamos el nombre del dato y en el segundo el valor del dato:

                    i.putExtra("titulo_autor", titulo);
                    i.putExtra("url", mp3url);
                    i.putExtra("pos", String.valueOf(pos));
                    i.putExtra("id", String.valueOf(id));
                    i.putExtra("img", Base64.encodeToString(imageInByte, Base64.DEFAULT));
                    i.putExtra("desde", "favoritos");

                    startActivity(i);

                }
                else{
                    // Mostrar mensaje
                    Log.e("registerstatus ", "Error de sistema");
                    Utils.mensaje(context, " Error de sistema. \n Vuelva a intentarlo. ");
                }

                    Dialog.dismiss();//ocultamos progess dialog.

            } // Onpostexecute

        } // verificaUsuario


    /* Obtener un cursor con todos los favoritos
     *
    */
    private Cursor obtieneCursor(int id) {

        DataBase db = new DataBase(this);

      cur = db.getLibro(id);
        cur.moveToFirst();

        db.close();

        return cur;
        }

    @Override
        public void onDestroy(){
        super.onDestroy();

            //cur.close();
            // liberar cursor
        }

}
```

ImageAdapter.java

```
package es.hol.audiolibros;


import android.app.Activity;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.BitmapFactory;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseAdapter;
import android.widget.EditText;
import android.widget.GridView;
import android.widget.ImageView;
import android.widget.TextView;

//un BaseAdapter puede usarse para un Adapter en un listview o gridview
//hay que implementar algunos métodos heredados de la clase Adapter,
//porque BaseAdapter es una subclase de Adapter
//estos métodos en este ejemplo son:  getCount(), getItem(), getItemId(), getView()


public class ImageAdapter extends   BaseAdapter {

        // Definir constantes
        private static final String TABLE_IMG ="libros";

        public static final String KEY_ROWID    = "_id";
        public static final String KEY_TITULO   = "titulo";
        public static final String KEY_AUTOR   = "autor";
        public static final String KEY_IMAGEN   = "imagen";

        // Definir variables
        private Context context;
        private Activity activity;

        // Asignar objetos
        private DataBase  db;                           // Clase SQLiteOpenHelper
        private SQLiteDatabase bsSql;           // Clase para manipular los datos
        private Cursor cur;                             // Interface para manipular las filas devueltas por una consulta

        private byte[] bb ;

        // el constructor necesita el contexto de la actividad donde se utiliza el adapter
         public ImageAdapter(Context context) {
                super();
                this.context = context;
                 cur = obtieneCursor();
        }
```

```
// devuelve el número de elementos que se introducen en el adapter
public int getCount() {
        // TODO Auto-generated method stub
        return cur.getCount();
}

// este método debería devolver el objeto que esta en esa posición del
 // adapter. No es necesario en este caso más que devolver un objeto null.
public Object getItem(int position) {
        return null;
 }

// este método debería devolver el id de fila del item que esta en esa
 // posición del adapter. No es necesario en este caso más que devolver 0.
public long getItemId(int position) {
         return 0;
 }

// Obtener cantidad de libros
public int getLibrosCount() {

        String countQuery = "SELECT * FROM " + TABLE_IMG;
        bsSql = db.getReadableDatabase();
        Cursor cursor = bsSql.rawQuery(countQuery, null);
        cursor.close();

        // return count
        return cursor.getCount();
}

// contenedor de los objetos view del item
public static class ViewHolder
{
        public ImageView gv_image;          // imagen
        public TextView gv_text;     // titulo
        public TextView gv_id;              // id del registro (oculto)
}

// crear un nuevo ImageView para cada item referenciado por el Adapter
// este método crea una nueva View para cada elemento añadido al
 // ImageAdapter. Se le pasa el View en el que se ha pulsado, converview
 // si convertview es null, se instancia y configura un ImageView con las
// propiedades deseadas para la presentación de la imagen
 // si converview no es null, el ImageView local es inicializado con este
 // objeto View
 public View getView(int position, View convertView, ViewGroup parent) {

        ViewHolder view;
        // LayoutInflater inflator = activity.getLayoutInflater();
         LayoutInflater inflator = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);


        if (convertView == null) {  // if it's not recycled, initialize some attributes
```

```
                    view = new ViewHolder();
                    convertView = inflator.inflate(R.layout.row_grid, null);

                    view.gv_text = (TextView) convertView.findViewById(R.id.gv_text);
                    view.gv_image = (ImageView) convertView.findViewById(R.id.gv_image);
                    view.gv_id = (TextView) convertView.findViewById(R.id.gv_id);

                    convertView.setTag(view);

            } else {
                    view = (ViewHolder) convertView.getTag();
            }

            //la imagen de este imageView será el elemento de la
            //posicion 'position' del cursor cur, declarado abajo.
            cur.moveToPosition(position);
             bb = cur.getBlob(2);

             // asignar la imagen convertida de byte a bitmap
            // imageView.setImageBitmap(BitmapFactory.decodeByteArray(bb, 0, bb.length));

            view.gv_image.setImageBitmap(BitmapFactory.decodeByteArray(bb, 0, bb.length));
            //gv_text.setText(cur.getString(1));
            view.gv_text.setText(cur.getString(1));
            view.gv_id.setText(cur.getString(0));

             //return imageView;
            return convertView;

}


 @Override
public void notifyDataSetChanged (){
            // Recargamos el cursor
            cur = obtieneCursor();
            super.notifyDataSetChanged();
 }


 /* Obtener un cursor con todos los favoritos
 *
 */
private Cursor obtieneCursor() {

            db = new DataBase(context);
            // Abre una conexion a la BD para lectura
            bsSql = db.getReadableDatabase();

            // Realizar consulta y obtener cursor
            String countQuery = "SELECT _id,titulo,imagen FROM " + TABLE_IMG + " ORDER BY titulo ASC";
             cur = bsSql.rawQuery(countQuery, null);
            cur.moveToFirst();
            //          cur.close();
```

```
                db.close();

        return cur;
        }

}
```

JSONParser.java

```
package es.hol.audiolibros;

import java.io.BufferedReader;
import java.io.FileInputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.UnsupportedEncodingException;
import java.net.SocketTimeoutException;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.StatusLine;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.impl.client.DefaultHttpClient;
import org.json.JSONArray;
import org.json.JSONException;
import org.json.JSONObject;
import android.util.Log;
import android.view.Gravity;
import android.widget.Toast;

public class JSONParser {

        static InputStream is = null;
        static JSONArray jarray = null;
        static String json = "";

        // constructor
        public JSONParser() {

        }

        public JSONArray getJSONFromUrl(String url) {

                StringBuilder builder = new StringBuilder();
                 HttpClient client = new DefaultHttpClient();


                 try {
```

```
                HttpGet httpGet = new HttpGet(url);
                HttpResponse response = client.execute(httpGet);
                StatusLine statusLine = response.getStatusLine();
                int statusCode = statusLine.getStatusCode();

                  if (statusCode == 200) {
                    HttpEntity entity = response.getEntity();
                    InputStream content = entity.getContent();

                    BufferedReader reader = new BufferedReader(new InputStreamReader(content));

                    String line;
                    while ((line = reader.readLine()) != null) {
                      builder.append(line);
                      //builder.append(line+"\n");
                  }
                  } else {
                          Log.e("==>", "Falló la descarga del archivo");

                  }

            } catch (ClientProtocolException e) {
                    e.printStackTrace();
            } catch (SocketTimeoutException e) {
               Log.e("==>", "Error when calling postData", e);

            } catch (IOException e) {
                    e.printStackTrace();
            }

                    // try parse the string to a JSON object
                    try {
                            jarray = new JSONArray( builder.toString());
                    } catch (JSONException e) {
                            Log.e("JSON Parser", "Error parsing data " + e.toString());

                    }

                    // return JSON String
                    return jarray;

}


public JSONArray getJSONFromFile(String ruta) {

            StringBuilder builder = new StringBuilder();

             try {
                    FileInputStream stream = new FileInputStream(ruta);

                     BufferedReader reader = new BufferedReader(new InputStreamReader(stream),1024 * 5);

                        String line;
                        while ((line = reader.readLine()) != null) {
```

```
                    builder.append(line);
                    //builder.append(line+"\n");
                }


            } catch (OutOfMemoryError e){
                Log.e("JSON Parser", "Error parsing data " + e.toString());
            } catch (Exception e) {
                e.printStackTrace();
            }

        // try parse the string to a JSON object
        try {
            jarray = new JSONArray( builder.toString());
        } catch (JSONException e) {
            Log.e("JSON Parser", "Error parsing data " + e.toString());

        }

        // return JSON String
        return jarray;

    }
}
```

Libroviewer.java

```
package es.hol.audiolibros;


import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import org.apache.http.NameValuePair;
import org.apache.http.message.BasicNameValuePair;
import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.content.pm.ActivityInfo;
import android.database.Cursor;
import android.graphics.BitmapFactory;
import android.os.AsyncTask;
import android.os.Bundle;
import android.preference.PreferenceManager;
import android.util.Base64;
import android.util.Log;
import android.view.View;
import android.view.Window;
import android.widget.ImageView;
import android.widget.TextView;


public class Libroviewer extends Activity{

	// Definir constantes
	public static final int KEY_ROWID    = 0;
	public static final int KEY_TITULO   = 1;
	public static final int KEY_AUTOR    = 2;
	public static final int KEY_TEMA     = 3;
	public static final int KEY_DESCRIP  = 4;
	public static final int KEY_MP3URL   = 5;
	public static final int KEY_POSICION = 6;
	public static final int KEY_IMAGEN   = 7;

	private Cursor cur;
	private byte[] bb ;

	private String titulo, mp3url;
	private int pos, id;
	 private int result[];

	public void onCreate(Bundle savedInstanceState) {
				super.onCreate(savedInstanceState);

				// fijar de manera permanente la orientación en modo vertical
		setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_PORTRAIT);

		// Ocultar barra de titulo
```

```
        requestWindowFeature(Window.FEATURE_NO_TITLE);

                setContentView(R.layout.libroviewer);

        // Obtener los valores pasados en el Intent
                Bundle extras = getIntent().getExtras();
                int idpos = extras.getInt("ID");

                // Refenciar objetos View
                ImageView image = (ImageView) findViewById(R.id.imageView1);
        TextView tit = (TextView) findViewById(R.id.txt_tit);
        TextView aut = (TextView) findViewById(R.id.txt_aut);
        TextView desc = (TextView) findViewById(R.id.txt_des);

        // Obtener cursor con el registro especificado en pos
        DataBase db = new DataBase(this);
        cur = db.getLibro(idpos);
        db.close();

        // Asignar datos del cursor a objetos View
        tit.setText(cur.getString(KEY_TITULO));
        tit.setSelected(true);
        aut.setText(cur.getString(KEY_AUTOR));
        desc.setText(cur.getString(KEY_DESCRIP));

        tit.setSelected(true);

        bb = cur.getBlob(KEY_IMAGEN);
        image.setImageBitmap(BitmapFactory.decodeByteArray(bb, 0, bb.length));

        mp3url = cur.getString(KEY_MP3URL);
        pos = cur.getInt(KEY_POSICION);
                titulo = cur.getString(KEY_TITULO)+" - "+cur.getString(KEY_AUTOR);
                id = cur.getInt(KEY_ROWID);

        cur.close();

    } // onCreate


     // Cierra la actividad
public void Volver(View v){
    finish();
}


    // Al pulsar boton play
    public void play(View boton){

        //Utils.mensaje(getApplicationContext(),"Reproducir");

        Log.d("LibroV", mp3url+"-"+titulo);

        // Obtener cursor con el registro especificado en pos
        // puede haber cambiado
```

```
DataBase db = new DataBase(this);
cur = db.getLibro(id);
db.close();

pos = cur.getInt(KEY_POSICION);

cur.close();

//Obtiene el objeto de preferencias de la aplicacion
SharedPreferences prefs = PreferenceManager.getDefaultSharedPreferences(Libroviewer.this);

// Obtener datos desde preferencias
String usuario = prefs.getString("user", "");
String clave   = prefs.getString("pass", "");
String server  = prefs.getString("server", "www.audiolibros.hol.es");
boolean registrado = prefs.getBoolean("REGISTRO", false);

// Comprobar conexion
if (!Utils.verificaConexion(this)) {
        // No hay conexion a Internet
        // Mostrar dialogo
        Utils.showAlertDialog(Libroviewer.this, "Internet",
                "Comprueba tu conexión", false);

 return;
}

// algunas url contienen espacios y darían error
mp3url = mp3url.replaceAll("\\s","%20");

Log.d("LibroV", mp3url);

   // Comprobar que existe la url
   result = Utils.CompruebaUrl(getBaseContext(),mp3url);

   if (result[0] == 200) {
        //toast = Toast.makeText(getApplicationContext(),
   //         "Existe: "+mp3url, Toast.LENGTH_LONG);
   //      toast.setGravity(Gravity.TOP, 25, 300);
   //      toast.show();

        // Si hay datos de registro verificar con el WebService
        if (registrado){

                // contiene datos de registro
                ArrayList<NameValuePair>  datos;

           // ruta de la función del Web Service
           String url = "http://"+server+"/WebService/validaUser.php";

           //Rellenar datos de registro
                datos = new ArrayList<NameValuePair>();
                datos.add(new BasicNameValuePair("usuario", usuario));
                datos.add(new BasicNameValuePair("password", clave));
                datos.add(new BasicNameValuePair("url", url));
```

```
                    // ejecutar tarea asincrona de validar usuario
                    verificaUsuario validar = new verificaUsuario(this,datos);
                    validar.execute();


                // Si no hay registro mostrar aviso
                }else {

                    // Mostrar dialogo
                    Utils.showAlertDialog(Libroviewer.this, "AVISO",
                    "\nNo hay datos de registro.\nDebe registrarse.", false);
            }
        }
        else {

            Utils.mensaje(getApplicationContext(),"Code: "+result[0] +
                        "\nNo se encontró el audiolibro o\nno hay conexión.");
        }

    }



     /*
     * Autentifica al usuario mediante funcion WebService
     *
     * Parámetro: context, contexto de la aplicación
     * Parámetro: datos, ArrayList con los datos
     * Retorno: resp, string con el valor devuelto por el WebService
    */

    // En nuestra tarea no necesitamos entrada,
    // no usaremos información de progreso y devolveremos un String.
    class verificaUsuario extends AsyncTask< Void, Void, String > {

    ProgressDialog Dialog;
    String cpass;

        Context context;
        ArrayList<NameValuePair> datos;
        public verificaUsuario(Context context, ArrayList<NameValuePair> datos){
          this.context=context;
          this.datos = datos;
        }

protected void onPreExecute() {

    //para el progress dialog
  Log.i("VerificaUsuario", "onPreExecute "+context);
  Log.i("VerificaUsuario", "onPre clave: "+datos.get(1).getValue().trim());

                //Dialog = ProgressDialog.show(context,"VALIDACION", "Comprobando datos de usuario");

  Dialog = new ProgressDialog(Libroviewer.this);
```

```
Dialog.setMessage("Validando....");
Dialog.setIndeterminate(false);
Dialog.setCancelable(false);
Dialog.show();

        // ciframos la contraseña aplicandole una máscara (la misma que aplica el WebService)
        String masc = "|#€7`¬23ads4ook12";
    cpass = masc + Utils.Cifrado.cifrar(datos.get(1).getValue().trim(), "MD5");
    cpass = Utils.Cifrado.cifrar(cpass, "SHA-1");
    datos.set(1, new BasicNameValuePair("password", cpass));

    Log.i("VerificaUsuario", "onPre cclave: "+datos.get(1).getValue().trim());

    // Codificar en base64
    byte[] byteArray;
    try {
                    byteArray = datos.get(0).getValue().trim().getBytes("UTF-8");
                    datos.set(0, new BasicNameValuePair("usuario",
Base64.encodeToString(byteArray,Base64.DEFAULT)));
                byteArray = datos.get(1).getValue().trim().getBytes("UTF-8");
                datos.set(1, new BasicNameValuePair("password",
Base64.encodeToString(byteArray,Base64.DEFAULT)));
            } catch (UnsupportedEncodingException e) {
                    // TODO Auto-generated catch block
                    Log.i("VerificaUsuario", "Error: "+ e);
                    //e.printStackTrace();
            }


    Log.i("VerificaUsuario", "onPre b64clave: "+datos.get(1).getValue().trim());


}

        protected String doInBackground(Void... params) {

            String url  = datos.get(2).getValue().trim();

            // elimimar url de datos
            datos.set(2, new BasicNameValuePair("url", ""));

            // usamos una clase httpHandler para enviar una petición al servidor web.
    Utils.httpHandler handler = new Utils.httpHandler();
    String resp = handler.post(url, datos);

    Log.i("VericaUsuario","resp: "+ resp+"url: "+url);

    // valor devuelto para onPostExecute
        return resp;

        }

        /*Una vez terminado doInBackground comprobar lo que halla ocurrido
         * recibe resp
        */
```

```
    protected void onPostExecute(String resp) {

      Log.i("onPostExecute=","resp: "+resp);

                    //validamos el resultado obtenido
                        if (resp.trim().contains("0")){

                            // Mostrar mensaje
                            Log.i("verifica0 ", "invalido");
                            Utils.mensaje(context, " Usuario o contaseña incorrectos ");

                        }
                        else if (resp.trim().contains("1")){

                            // Mostrar mensaje
                            Log.i("verifica1 ", "valido");
                            //Utils.mensaje(context, " Usuario validado ");

                            // Crear un Intent, pasar parametros y lanzar una nueva Activity con el reproductor

                            Intent i = new Intent(getApplicationContext(), MusicPlayerActivity.class);

                            // llamamos al método putExtra de la clase Intent. Tiene dos parámetros de tipo
String,
                        // clave-valor en el primero indicamos el nombre del dato y en el segundo el valor del dato:

                        i.putExtra("titulo_autor", titulo);
                        i.putExtra("url", mp3url);
                        i.putExtra("pos", String.valueOf(pos));
                        i.putExtra("id", String.valueOf(id));
                        i.putExtra("img", Base64.encodeToString(bb, Base64.DEFAULT));
                        i.putExtra("desde", "favoritos");

                                startActivity(i);

                        }
                            else{
                                // Mostrar mensaje
                                Log.e("registerstatus ", "Error de sistema");
                                Utils.mensaje(context, " Error de sistema. \n Vuelva a intentarlo. ");
                            }

                        Dialog.dismiss();//ocultamos progess dialog.

    } // Onpostexecute


  } // verificaUsuario

}
```

Archivos.xml

El resto de archivos se relacionan a continuación y se puede ver su contenido en los archivos de código fuente que se adjuntan a esta memoria.

/ res

/anim

fade_in.xml
fade_out.xml

/layout

acercade.xml
bg_player_footer.xml
bg_player_header.xml
catalogo.xml
corner_button.xml
favoritos.xml
libroviewer.xml
list_item.xml
player.xml
principal.xml
registro.xml
rounded_button.xml
rounded_corner.xml
row_grid.xml
splash_screen.xml

/menu

favoritos_ctx.xml
filtrar_ctx.xml
menu.xml
menu_ctx.xml

/xml

configuracion.xml

Licencia.

Esta obra está bajo una licencia de Creative Commons