

Índice

1.	Presentación del proyecto	4
1.1.	Análisis de la competencia y objetivo	4
2.	Planificación temporal	5
3.	Diseño conceptual	7
3.1.	Diagrama de clases	7
3.2.	Diagrama de casos de uso	9
4.	Prototipado de la aplicación	10
4.1.	Descripción general	10
5.	La aplicación: series addicts	12
5.1.	Funcionalidades	12
5.2.	Descripción por pantallas de la aplicación	12
5.3.	Feedback de la aplicación	17
6.	Desarrollo de la aplicación	19
6.1.	Elementos en el diseño y usabilidad de la app	19
6.2.	Especificaciones técnicas:	23
6.2.1.	Herramientas utilizadas	23
6.2.2.	Tecnologías utilizadas	25
6.2.2.1.	Html5	25
6.2.2.2.	Css3	26
6.2.2.3.	Php	26
6.2.2.4.	Mysql	26

6.2.2.5. Jquery	26
6.2.2.6. Json	27
6.2.3. Herramientas online utilizadas para el desarrollo	27
6.2.3.1. Firebug	27
6.2.3.2. Jsfiddle	28
6.3. Programación de la aplicación	30
6.3.1. Específico para android	30
6.3.1.1. Configuración de eclipse para phonegap	30
6.3.1.2. Configuración de los archivos de android	30
6.3.2. Estructura de la aplicación	34
6.3.2.1. Base de datos	34
6.3.2.2. Programación del lado del servidor	35
6.3.2.3. Programación del lado de cliente	37
6.3.3. Validación de la aplicación	40
6.4. Usabilidad de la aplicación	40
6.5. Problemas encontrados durante el desarrollo	40
6.5.1. Funcionalidades no implementadas	44
6.6. Mejoras de la aplicación para futuras versiones	45
7. Conclusiones	46
8. Bibliografía	48
8.1. Guías de estilo:	48
8.2. Guías de programación:	48

1. Presentación del proyecto

Series Addicts está basada en la necesidad de una aplicación rápida y ligera con la que tener el control de todas las series que seguimos, para poder acceder a las funcionalidades mientras estamos disfrutando de nuestra serie favorita.

Desde hace unos años, en mi entorno, he detectado una creciente tendencia a visualizar una mayor cantidad de series, principalmente por internet. Una de las razones principales es la falta de una televisión pública de calidad, ya que los usuarios tienen que adaptarse al contenido de la programación televisiva en vez de lo contrario.

Hoy en día con Internet existe la posibilidad de tener un mejor aprovechamiento del tiempo de ocio sin tener que soportar cambios de programación o cortes publicitarios,. Por todo ello, creo que hay una necesidad de un buen sistema de gestión de los contenidos que visualizamos por internet, una especie de teletexto ampliado y personalizado del siglo XXI y eso es lo que se propone con esta aplicación.

Aclaraciones:

En este proyecto no se va a implementar la descarga de las series ni su visionado en streaming. Se trata de una de las especificaciones más demandadas (y por tanto más ofertadas), pero no está muy clara su legalidad. Más adelante se podría estudiar la forma de hacerlo, pero la funcionalidad principal de la aplicación será la gestión de nuestras series.

1.1. Análisis de la competencia y objetivo

Son muchas las aplicaciones que existen en el mercado relacionadas con las series, pero la mayoría de ellas sólo ofrecen el visionado de series online y lo hacen de una forma un tanto descuidada, como hechas sólo con la intención de obtener dinero con la publicidad, un ejemplo de éstas podría ser SeriesDroid N. Ésta aplicación no tendrá ese fallo ya que se va a publicar con una licencia abierta y no se va a realizar con ánimo de lucro.

En cambio, existen varias páginas web, como IMDb o Series.ly que disponen de sus propias aplicaciones móviles, en las que me voy a fijar para realizar mi aplicación, puesto que tienen aplicaciones personalizadas, cuidadas y rápidas. Conseguir una aplicación similar con los mismos estándares de calidad va a ser el objetivo de este Trabajo Fin de Carrera.

2. Planificación temporal

Se ha planteado un diagrama de Gantt para mostrar de una forma gráfica y por tanto más visible la planificación temporal del proyecto.

En él se proponen una serie de tareas y a cada una de ellas se le ha asignado los recursos que han colaborado o que van a realizarla. Estos recursos han sido asignados para semejarse lo más posible a una empresa, aunque evidentemente todo el trabajo lo realizaré yo (excepto el del consultor, por supuesto).

Este equipo consta de:

- Jefe de proyecto. Quien realizará la gestión del proyecto.
- Equipo de desarrollo. Se trata de un equipo de especialistas subdividido en dos tipos:
 - Diseñadores.
 - Programadores.
- Testers o grupo de evaluación. Personal ajeno al proyecto que prueben la aplicación para dar su opinión y sacar a la luz los principales defectos de ésta.
- Consultor. En este caso coincide con el consultor de la UOC, Albert Grau.

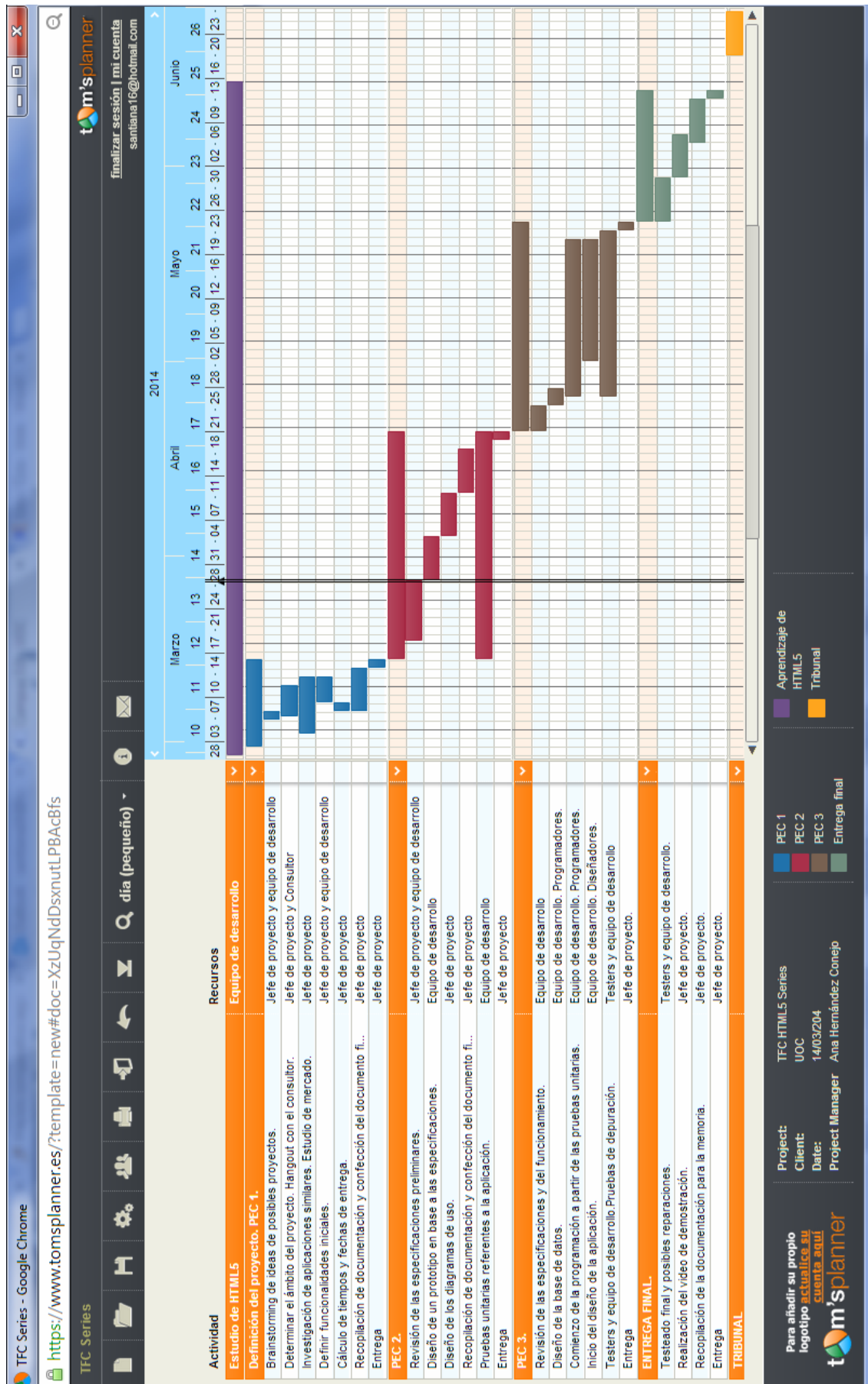


Figura 1. Planificación temporal del proyecto

3. Diseño conceptual

A continuación se muestran los diagramas de clases y de casos de uso de la aplicación con todas sus funcionalidades. Aunque la versión actual no dispone de todas las funcionalidades indicadas, se harán cambios en un futuro próximo para que así sea.

3.1. Diagrama de clases

A continuación se explican las relaciones principales entre las clases:

Tenemos relación de asociación entre:

- Usuario y series: Un usuario puede no tener series o tener varias, y lo mismo con las series, pueden no tener ningún usuario o varios en cada una de ellas.
- Series y alarmas: Para crear una alarma se le debe de asignar una serie, pero las series no tienen por qué tener alarmas.

Vemos relaciones de agregación por valor:

- Los usuarios poseen Alarmas y Comentarios, y es una relación de agregación puesto que cuando se borra el usuario, no tiene sentido guardar el objeto.
- Una serie posee curiosidades (0 o varias), estaríamos en el mismo caso que antes, al desaparecer la serie no tiene sentido guardar las curiosidades.
- Series y los capítulos: Una serie puede tener 0 (si todavía no se ha estrenado) o varios capítulos, pero un capítulo sólo pertenece a una serie. Al desaparecer la serie, desaparecen los capítulos.

Serie y capítulos tienen una relación de asociación, puesto que una serie instancia varios capítulos (la creación del Objeto capítulos esta condicionado a la instanciación proveniente desde el objeto serie).

Persona es una clase abstracta, dos subclases la implementan: usuario y administrador.

Los métodos entre otros son:

- De usuario:
 - Votar
 - Comentar
 - SetAlarma
 - MarcarCapitulo (como visto)
 - SeguirSerie
- Del administrador
 - UpdateBBDD

El método update del administrador, en este caso se refiere a la actualización manual de las series y los capítulos en la base de datos de la aplicación, puesto que en un principio no va a estar conectada a ninguna página como series.ly. Ésto sería un requisito muy deseable para que la aplicación fuera autónoma y por tanto realmente útil y por tanto será estudiado en futuras versiones.

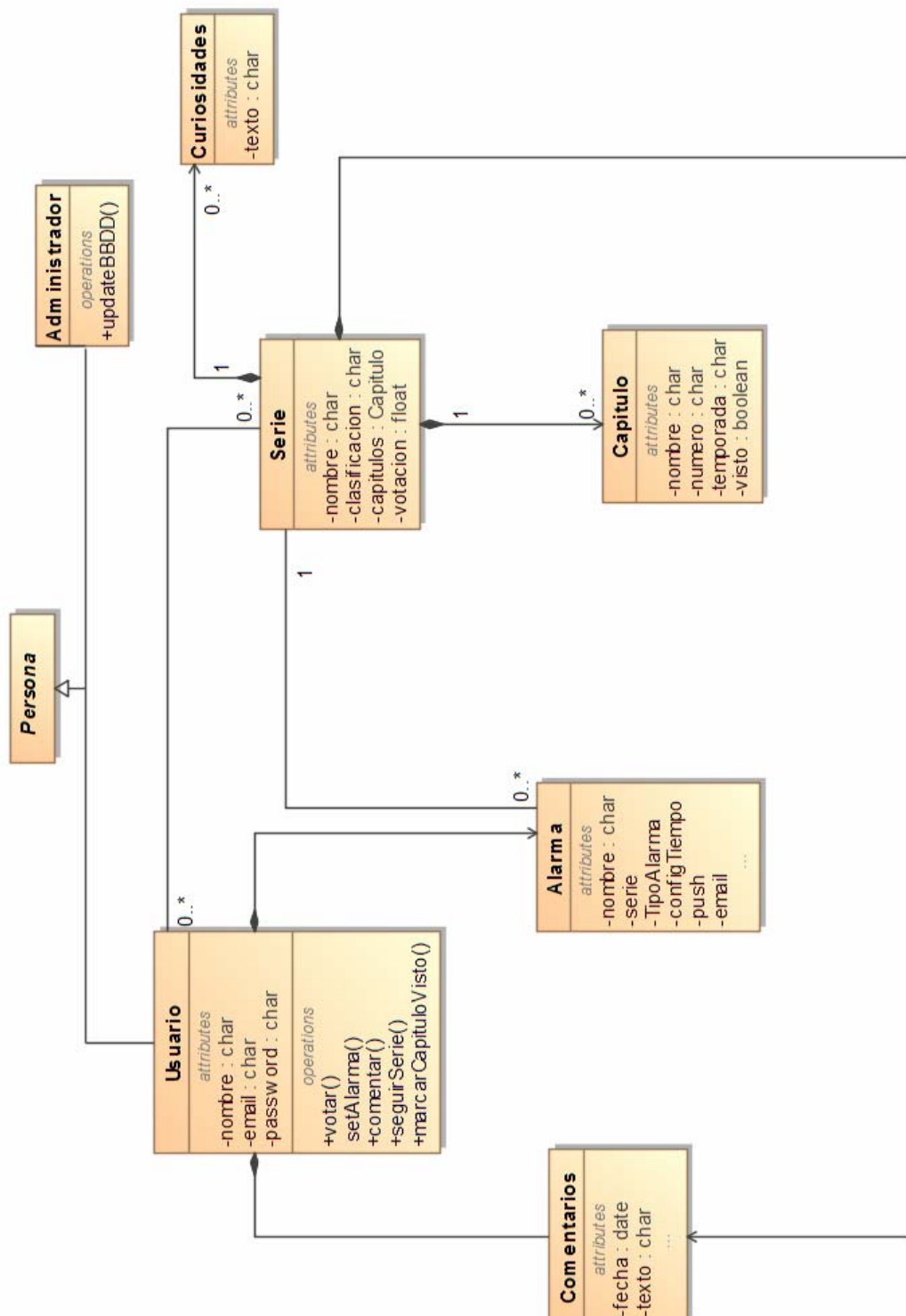


Figura 2. Diagrama de clases

3.2. Diagrama de casos de uso

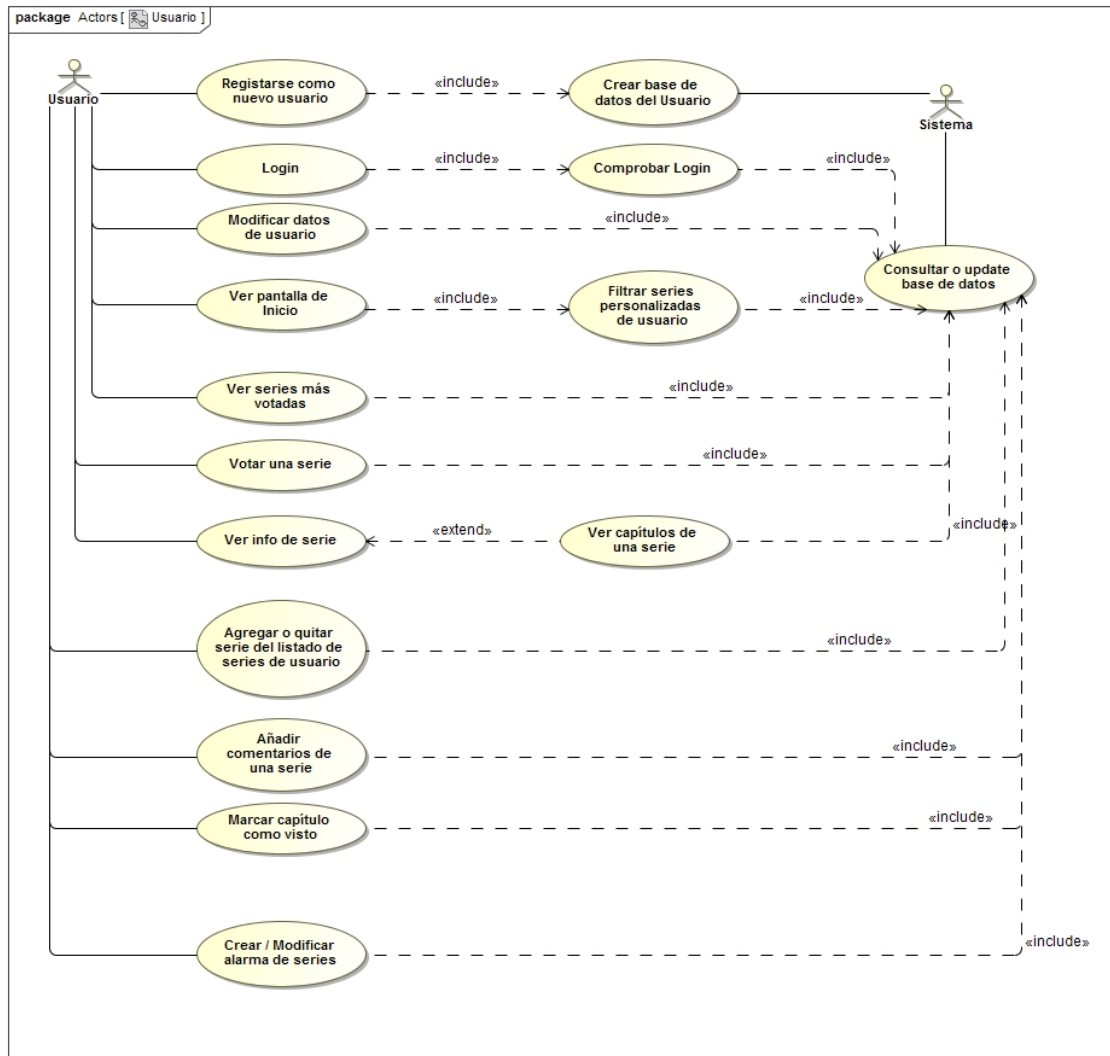


Figura 3. Diagrama de casos de uso

Los actores del diagrama son usuario y sistema. El actor usuario es el que va a interactuar con la aplicación navegando entre las pantallas y el sistema es el que se va a encargar de guardar los cambios y mostrar los resultados de la base de datos.

En cuanto a los casos de uso vemos claramente las funcionalidades del rol sistema, que realiza los cambios de la base de datos.

Todos los casos de uso que tiene asignados el rol usuario terminan haciendo referencia a algún caso de uso del rol sistema puesto que cualquiera de las acciones que puede realizar son consultar las distintas páginas principales de la aplicación, y estas están personalizadas para cada usuario y por tanto se trata de consultas y actualizaciones de la base de datos.

4. Prototipado de la aplicación

4.1. Descripción general

Para el desarrollo del mockup o prototipo se ha utilizado una aplicación muy completa llamada fluid ui, que permite la simulación de una aplicación en un dispositivo android (entre otros) con su diseño característico y en el que además se puede hacer una aproximación de cómo sería la aplicación puesto que se puede programar el paso a unas pantallas u otras como si fuera una aplicación real.

El prototipo puede probarse desde este link:

https://www.fluidui.com/editor/live/preview/p_FuqQRL8eI2DFsWPh3Z7qg9uT6EmbQO6p.1398110727739

En el se puede ir navegando entre las diversas pantallas o mediante el código QR:



Figura 4. Código QR para la visualización del prototipo

El cual puede probarse en cualquier móvil escaneándolo con un lector de códigos QR o con la propia app de fluid ui (fluid ui player), que permite una visualización más realista puesto que muestra exclusivamente el mockup.

Esta aplicación web en su versión gratuita permite el diseño de un máximo de 10 pantallas por proyecto, pero es muy completo puesto que entre otras cosas dispone de control de versiones propio. Es una aplicación que volveré a utilizar para próximos desarrollos.

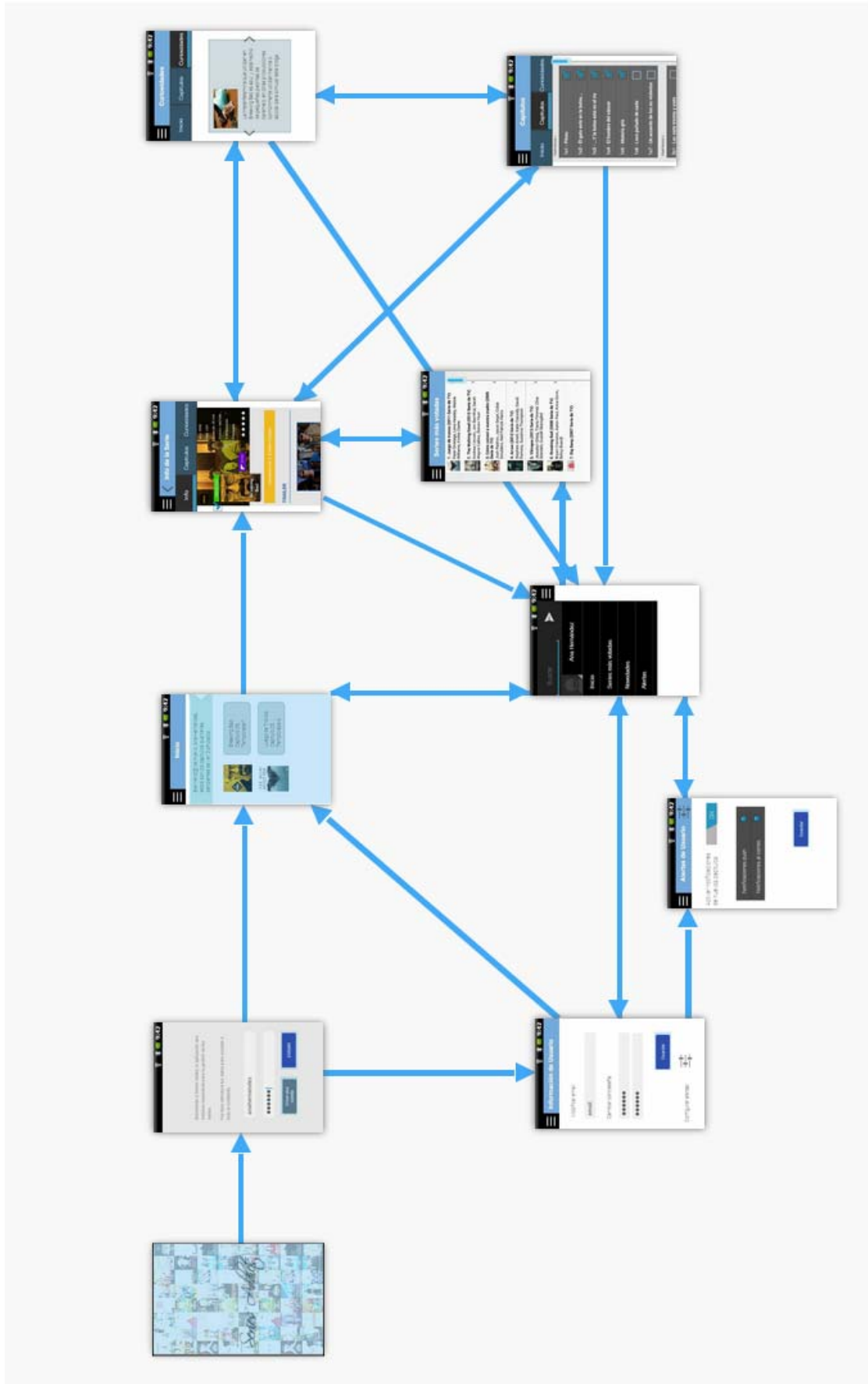


Figura 5. Mapa general del Mockup

5. La aplicación: Series Addicts

5.1. Funcionalidades

La aplicación dispone de las siguientes funcionalidades:

Reconocimiento de usuario en la pantalla inicial. Una vez logeado aparecerán las series que el usuario está siguiendo actualmente.

Cada serie dispondrá de una funcionalidad que muestre la posición en la que se encuentra en el ranking de la aplicación y que a su vez permita votar cada serie mediante un sistema visual como por ejemplo 5 estrellas.

Dichas series se podrán clicar y llevarán a una página con información de la serie y con un listado de los capítulos pertenecientes a la temporada que esté visualizando actualmente el usuario. Aunque se podrá elegir el resto de temporadas mediante un menú, posiblemente superior.

Los capítulos ya visualizados tendrán una marca, para poder tener un control de los contenidos vistos.


Sólo se ha programado con PhoneGap para Android, pero se modificará más adelante para probarlo en el resto de plataformas.

5.2. Descripción por pantallas de la aplicación

Podemos hacer una diferenciación de la aplicación en tres partes:

- Pantalla de login
- Parte propia del usuario
- Listado de todas las series ordenadas de mayor a menor puntuación.

Se detallan a continuación:



Bienvenido a Series Addicts!

Acceder

Usuario

Contraseña

Login

Figura 6

Esta pantalla es la pantalla de inicio, la primera que se ve al entrar en la aplicación. En esta pantalla se realizará el log In del usuario, sólo hay que introducir los datos como en cualquier aplicación. Si los datos de acceso no son correctos nos avisará con un mensaje.

Aunque he detectado que a algunos usuarios no les gusta en logueado en las aplicaciones, este en concreto, es necesario, ya que la aplicación va a ser utilizada por varios usuarios y cada uno de ellos dispone de sus propias series personalizadas. En el apartado de mejoras se propone una posible solución ante esta respuesta de los testers.

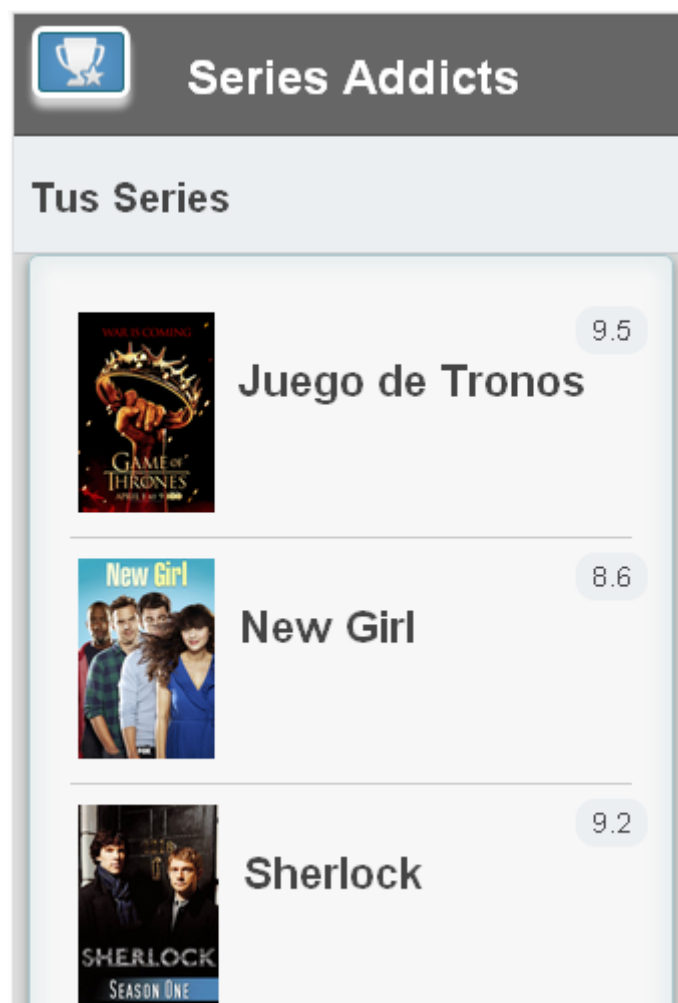


Figura 7

Esta pantalla está configurada para mostrar las series favoritas de cada usuario y la puntuación que tiene cada una de ellas, están ordenadas de mayor a menor puntuación.

Al hacer click en cada una de ellas, la aplicación nos lleva a una nueva página con información detallada de la serie.



Figura 8

Esta es la parte de dedicada a la información de cada una de las series. En esta primera pestaña podemos ver que además de la información, hay un checkbox. Este, dependiendo de si está marcado o no, nos indica si la serie está entre nuestras favoritas o no. Si lo pulsamos, dejamos de seguir la serie y desaparece de nuestra pantalla de inicio.



Figura 9

La pantalla que vemos en la figura es la de los capítulos de una serie en concreto. Podemos marcar el capítulo como visto o no visto mediante el checkbox situado al lado del nombre del mismo.

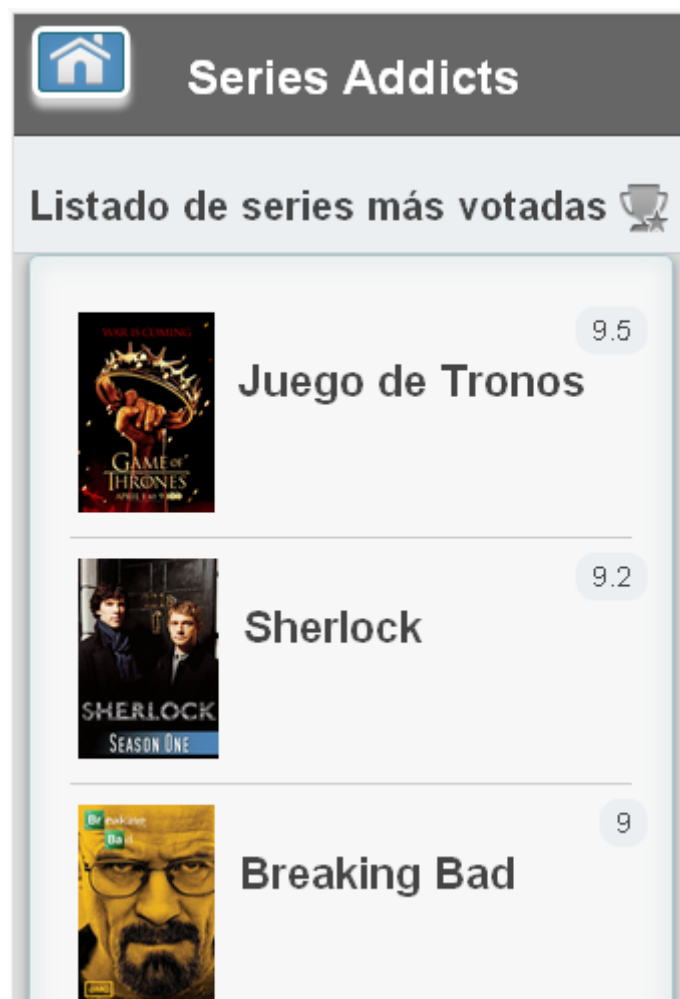


Figura 10

Esta es la pantalla del ranking de las series. Si queremos seguir una nueva serie, sólo tenemos que acceder a ella pulsando en el botón del ranking de la pantalla de inicio. Se trata de un listado con todas las series que existen en la base de datos, ordenadas de mayor a menor puntuación.

Si pulsamos en cualquiera de ellas nos lleva a la misma pantalla de información de la serie, que hemos visto antes. Podemos pulsar el checkbox para seguirla.

5.3. Feedback de la aplicación

Se ha hecho una demostración de la aplicación final a un grupo de gente que coincide con el perfil del usuario final de esta aplicación, es decir personas mayores de edad aficionadas a las series. Y el resultado se expone a continuación:

- Añadir más funcionalidades a la aplicación, como por ejemplo el trailer de la serie, o más información de los actores, de los estrenos, etc.

- Quitar la parte del login de usuario. Por lo visto es algo que no le gusta a los usuarios, porque entorpece un poco la rapidez.
- Introducir alguna forma de contacto con el administrador de la aplicación para informar de posibles averías o para posibles contactos laborales.
- Añadir una forma de salir de la aplicación.

Estos comentarios y otros con fallos de la aplicación ya subsanados han sido de gran ayuda para obtener el mejor acabado posible.

6. Desarrollo de la aplicación

6.1. Elementos en el diseño y usabilidad de la app

La usabilidad en las aplicaciones móviles es uno de los elementos fundamentales para el éxito de las mismas. Para ello una aplicación debe de ser efectiva y eficiente. Estos son los aspectos que se han tenido en cuenta a la hora de programar la aplicación para conseguirlo.

El diseño de la aplicación es uno de los aspectos más importantes en cuanto a la usabilidad, puesto que un mal diseño hace que el usuario no quiera volver a utilizar la aplicación nunca más y procederá a su desinstalación.

Para una buena experiencia de usuario hay que tener en cuenta cuáles son los usos más extendidos, por ejemplo, un smartphone se suele utilizar de forma vertical y una tablet de forma horizontal (aunque no siempre es así, pero es lo más usual). Además tenemos que diseñar nuestra aplicación para que sea cómoda para nuestros usuarios, puesto que hay zonas de las pantallas donde es más fácil llegar, como podemos ver en la figura inferior, donde se situarán las funcionalidades más usadas.



Figura 11. Tipos de accesibilidad en dispositivos móviles.

Fuente: <http://www.lukew.com/ff/entry.asp?1649>

Una de las principales características de una buena aplicación es que su diseño sea responsive. Esto quiere decir que la aplicación se adapte a distintos dispositivos desde una misma vista mediante el uso de CSS. Una aplicación que se pueda ver en

cualquier dispositivo tanto en horizontal como en vertical hace que nuestro producto pueda llegar a más usuarios.

En el caso de Series Addicts se ha programado para que sea así, y podemos comprobar que nuestra programación responsive se está realizando de una forma satisfactoria gracias a las herramientas para desarrolladores que tiene firefox. En mi caso he utilizado “Responsive Layouts”, funcionalidad que se puede utilizar muy fácilmente haciendo clic con el botón derecho del ratón en nuestra aplicación, seleccionamos Web Developer, Resize y finalmente View Responsive Layouts.

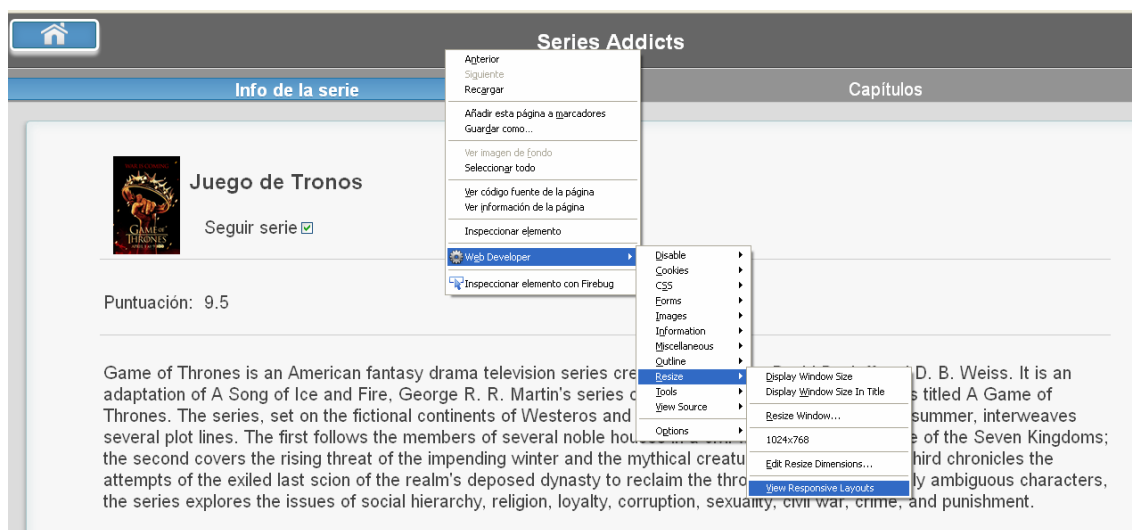


Figura 12. Acceso a Responsive Layouts de firefox

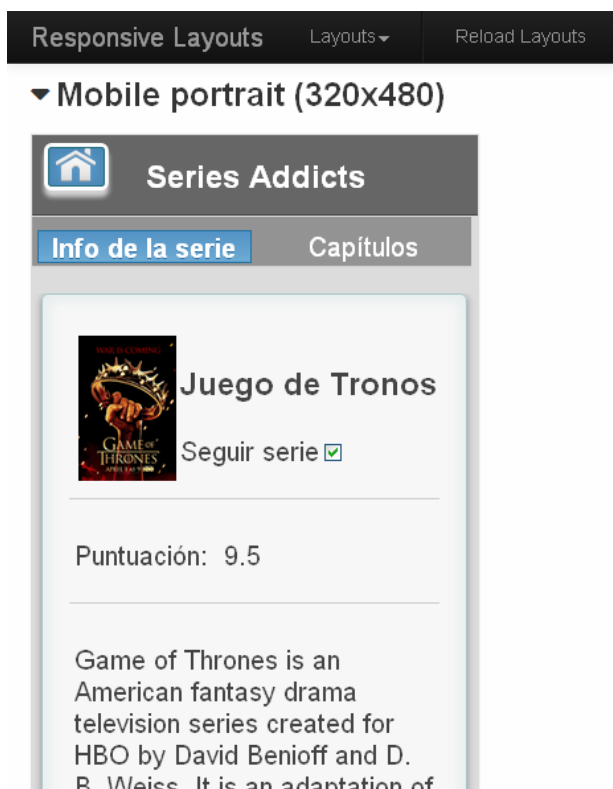


Figura 13. Responsive Layouts de firefox. Vista de pantalla de 320x480 en vertical.

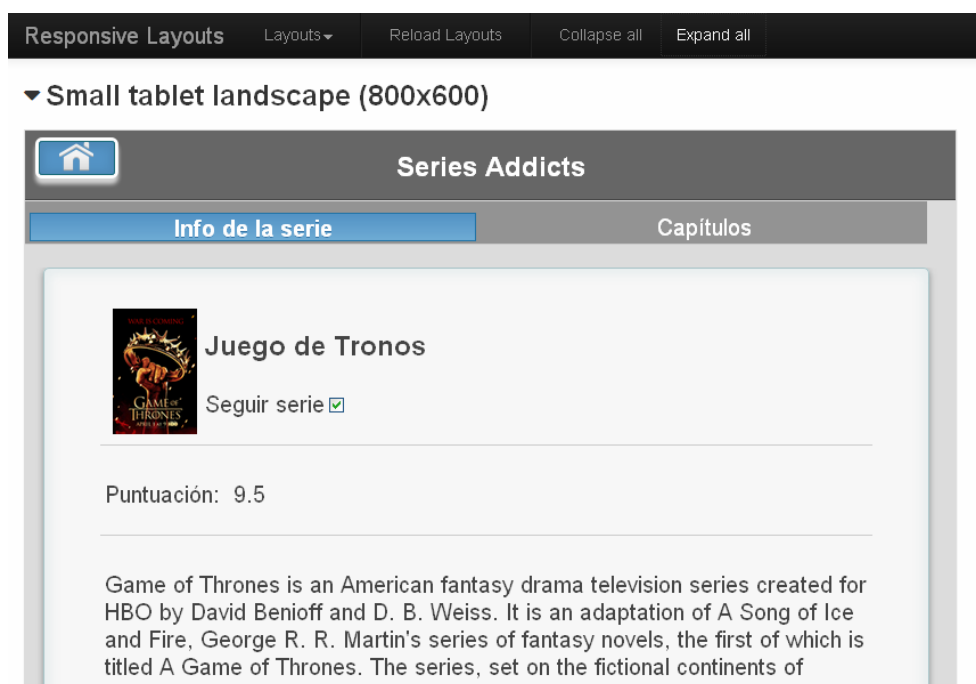


Figura 14. Responsive Layouts de firefox.
Vista de pantalla de 800x600 en horizontal.

Otro elemento que se ha realizado durante el desarrollo de la aplicación es la creación de iconos personalizados para cada resolución. Cada uno de ellos se ha guardado en su carpeta correspondiente en res (drawable-hdpi, drawable-mdpi, drawable-xhdpi y drawable-xxhdpi). Y para cada dispositivo se instalará el icono más adecuado a su tamaño.



Figura 15. Icono para resolución xhdpi.

Se ha buscado que sea fácil moverse dentro de la aplicación, mediante una navegación con botones, que contienen imágenes descriptivas de dónde nos dirigimos.

Estos botones se han diseñado para que sean adaptables y se redimensionen automáticamente dependiendo del tipo de pantalla:



Figura 16. Muestra de los botones de navegación adaptables.
Pantallas de 320x480 y de 1024x768

El área de navegación resultante deja espacio de sobra al área de contenidos, puesto que se han situado en el header de la aplicación.

Se ha conseguido que el funcionamiento y carga de la aplicación se haga de forma suave y fluida gracias a la carga de datos dinámica mediante JSON, lo que hace que sea más cómoda de utilizar.

En el apartado de tecnologías utilizadas se hace una exposición más detallada de la carga de datos.

Siguiendo estos aspectos, en mi opinión se ha obtenido una aplicación intuitiva, de fácil manejo y por tanto de buena calidad.

6.2. Especificaciones técnicas:

La aplicación se ha desarrollado en HTML5, pero en principio sólo se ha testado en un dispositivo Android, puesto que es el terminal del que dispongo. En un futuro el proyecto se adaptará para el resto de plataformas.

6.2.1. Herramientas utilizadas

Elementos software:

- Entorno de desarrollo: Eclipse + ADT plugin. Paquete que se descarga directamente desde la página: <http://developer.android.com/sdk/index.html>. Este paquete contiene:
 - o Eclipse (Version: 4.3.1.v20130911-1000) y plugin ADT (Build: v22.6.2-1085508)
 - o Herramientas Android SDK
 - o Última plataforma desarrollada de Android
 - o El último modelo desarrollado de emulador. (En este caso no se ha utilizado, puesto que es muy lento y se ha ido comprobando la aplicación en los navegadores firefox y chrome)
- PhoneGap. Framework para el desarrollo de aplicaciones móviles que permite a los programadores desarrollar aplicaciones para dispositivos móviles utilizando herramientas genéricas tales como JavaScript, HTML5 y CSS3. Las aplicaciones resultantes son híbridas, es decir que no son realmente aplicaciones nativas al dispositivo (ya que el renderizado se realiza mediante vistas web y no con interfaces gráficas específicas de cada sistema), pero no se tratan tampoco de aplicaciones web (teniendo en cuenta que son aplicaciones que son empaquetadas para poder ser desplegadas en el dispositivo incluso trabajando con el API del sistema nativo).
- Notepad ++
- Filezilla

Elementos hardware:

- Samsung Galaxy S3
- Portátil Samsung Intel Core 2 Duo

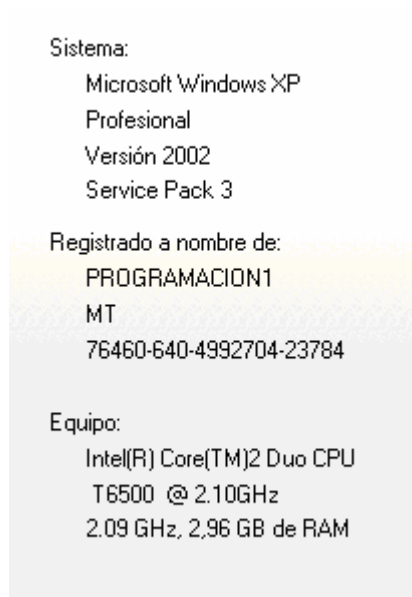


Figura 17. Características del equipo en el que se ha programado la aplicación.

Software utilizado durante la realización del video de presentación:

- Prezi. Software utilizado para la presentación en vez de power point. Me ha sorprendido su facilidad de uso y el acabado tan profesional que tienen.
- Camtasia. Utilizado para la grabación de la presentación en video. Es una aplicación muy completa y fácil de utilizar. Y aunque no es de uso libre, creo que merece la pena pagar la licencia por sus grandes prestaciones.

6.2.2. Tecnologías utilizadas

6.2.2.1. HTML5

HTML5 es la quinta revisión importante del lenguaje básico HTML. Todavía se encuentra en modo experimental, no es un estándar, pero ya es utilizada por una gran parte de los desarrolladores web por sus mejoras y ventajas.

HTML5 nos permite una mayor interacción entre nuestras páginas web y contenido media

Una de las mejoras ha sido la inclusión de nuevas etiquetas. Para el diseño de estas etiquetas han seguido las costumbres utilizadas por los desarrolladores web desde hace años, y consiste en cambiar las clásicas etiquetas `<div>` (con la clase header, footer, nav, etc) por una etiqueta con ese mismo nombre que haga la misma tarea de forma más rápida y eficiente.

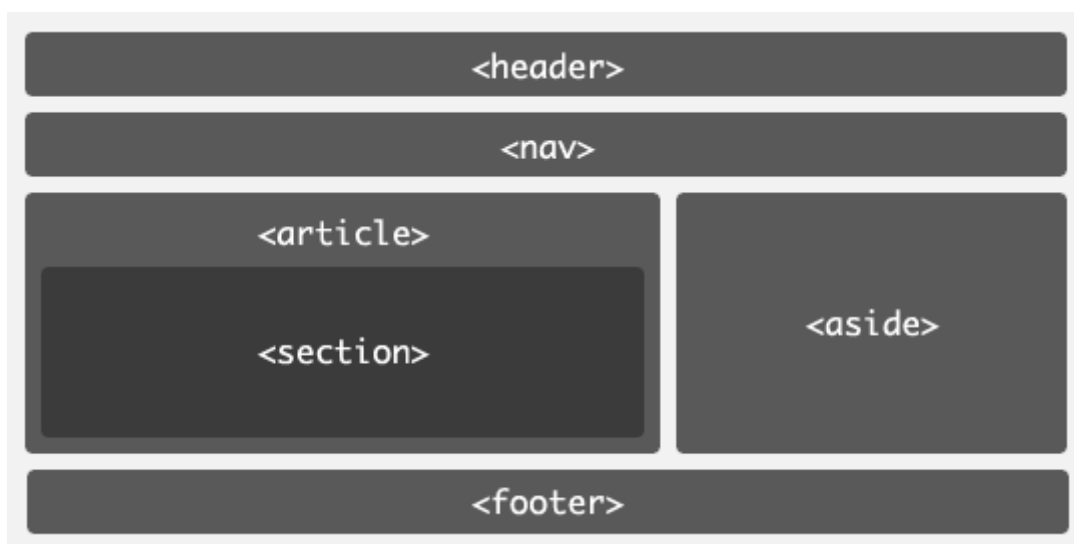


Figura 18. Nuevas etiquetas en HTML5.

6.2.2.2. CSS3

CSS es un lenguaje creado para definir el estilo de las páginas web escritas con HTML. Se creó para separar el contenido de la forma consiguiendo así mantener un control mucho más preciso y facilitando aún más el mantenimiento de los proyectos web.

La mayor ventaja de CSS es que se puede modificar completamente la apariencia de una página web sin tocar nada del código .html.

En este proyecto ha sido fundamental el uso de las hojas de estilo, para el diseño de toda la aplicación, puesto que he tenido problemas al utilizar jQueryMobile (framework que pensaba utilizar para dar ese acabado tan profesional de las aplicaciones móviles en muy poco tiempo, como su propio logo indica “write less, do more”).

6.2.2.3. PHP

PHP es el acrónimo de Hipertext Preprocesor. Es un lenguaje de programación del lado del servidor gratuito e independiente de plataforma, rápido, con una gran librería de funciones y mucha documentación.

PHP se ha utilizado a lo largo de la aplicación de una forma recursiva, ya que todas las interacciones con la base de datos se han realizado a través de archivos .php ubicados en un servidor externo.

6.2.2.4. MySQL

MySQL se ha utilizado para las sentencias con las que nos hemos dirigido a la base de datos. Siempre se ha utilizado la combinación php+mysql, puesto que

6.2.2.5. jQuery

jQuery es un framework de JavaScript open-source, que trabaja en el lado del cliente, (puesto que está basado en JavaScript), que funciona en múltiples navegadores, y que es compatible con CSS3. Su objetivo principal es hacer la programación de scripts mucho más fácil y rápida (como buen framework que es).

Con jQuery se pueden producir páginas dinámicas así como animaciones en poco tiempo y lo que es más importante, sin importar el navegador.

Durante este proyecto ha sido fundamental el uso de jQuery para el acceso y la modificación de los elementos del DOM.

6.2.2.6. JSON

JSON (JavaScript Object Notation) es un formato para el intercambios de datos, básicamente JSON describe los datos con una sintaxis dedicada que se usa para identificar y gestionar los datos y nació como una alternativa a XML. Las ventajas que tiene el uso de JSON es que puede ser leído por cualquier lenguaje de programación. Por lo tanto, puede ser usado para el intercambio de información entre distintas tecnologías.



Figura 19. Logotipo de JSON.

Los tipos de valores que podemos encontrar en JSON son los siguientes:

- Un número (entero o float)
- Un string (entre comillas simples)
- Un booleano (true o false)
- Un array (entre corchetes [])
- Un objeto (entre llaves { })
- Null

En este proyecto el papel del JSON ha sido fundamental y se ha utilizado en numerosas ocasiones. Más adelante analizaremos una parte de la aplicación realizada con JSON.

6.2.3. Herramientas online utilizadas para el desarrollo

6.2.3.1. Firebug

Firebug es una extensión del navegador Firefox que nos permite editar webs y hojas de estilo, monitorizar tiempos de carga, depurar javascript y ver los errores y procesos de carga en la página además de explorar el DOM (acrónimo de “Modelo en Objetos para la representación de Documentos”).

Esta aplicación ha sido fundamental en el desarrollo de la aplicación y también la utilizo constantemente en mi trabajo. Con ella podemos modificar tanto los CSS como ver tiempos de carga de la aplicación, como visualizar si la respuesta de la aplicación se está haciendo de forma correcta desde el apartado de la consola.

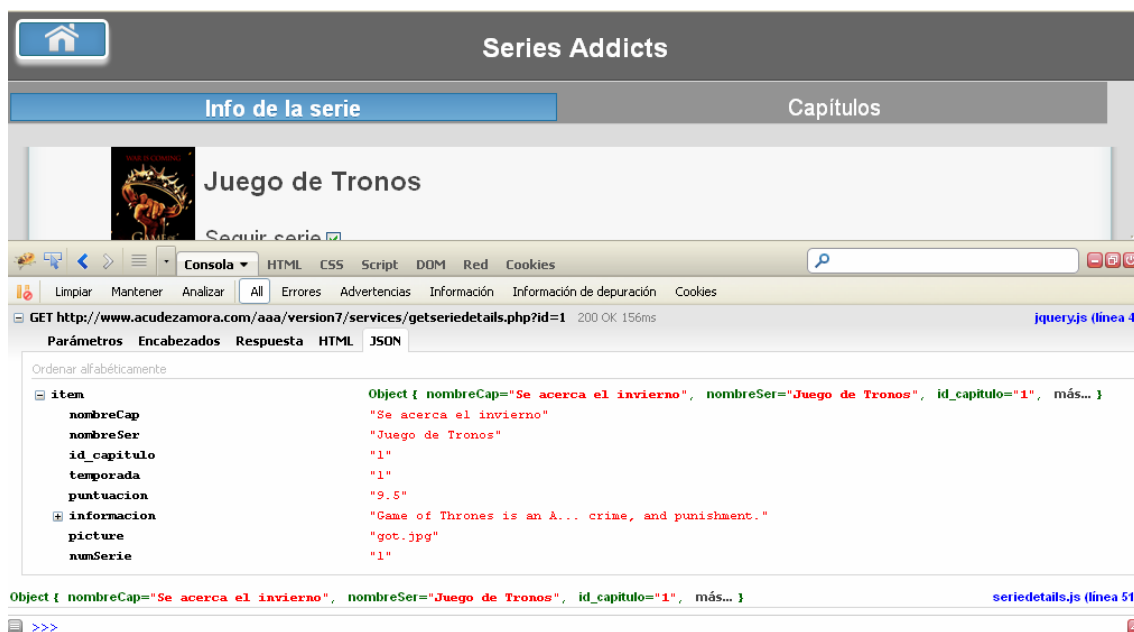


Figura 20. Visualización del JSON de un capítulo con Firebug

6.2.3.2. JSFIDDLE

Una parte fundamental del trabajo la he realizado con la aplicación online jsfiddle, que me ha servido para probar de manera rápida las funcionalidades de la aplicación.

JSFIDDLE es un entorno de desarrollo en línea, totalmente gratuito y de máxima utilidad para los desarrolladores de tecnologías web. Sirve para la codificación de HTML, CSS y Javascript. Como dato curioso, la palabra *fiddle* se traduciría como *trapicheo*, o sea que sirve para trapichear con la codificación.

Lo que JSFiddle hace es extremadamente sencillo, pero altamente poderoso, en particular para los que programan estos frameworks con regularidad. Se trata de un editor para programar fácilmente en JavaScript usando librerías muy potentes de este lenguaje: jQuery, Mootools, Prototype, ExtJS, etc.

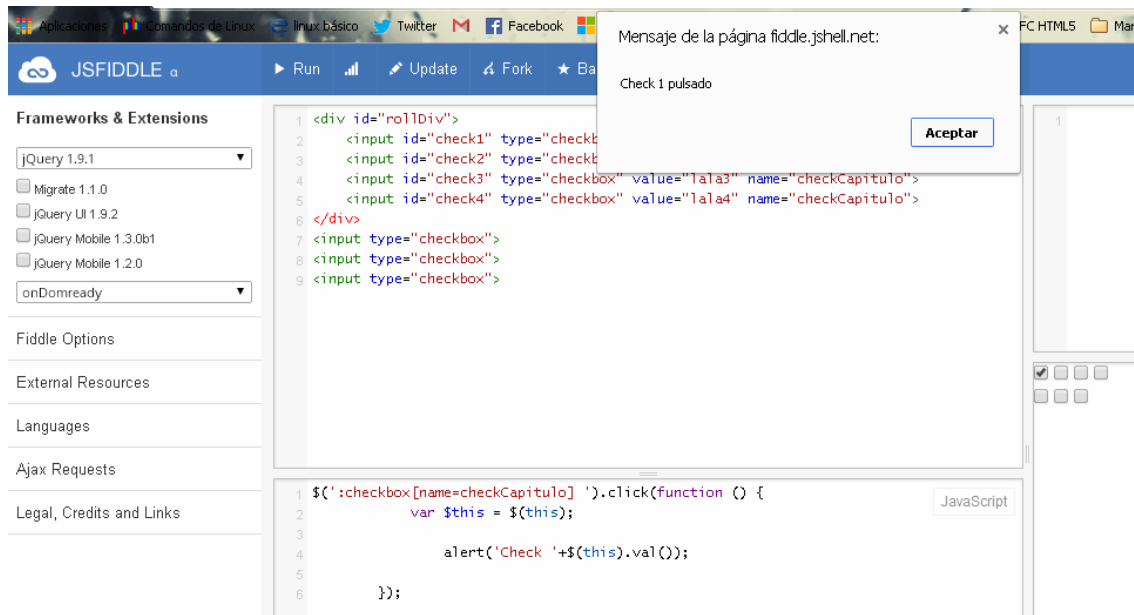


Figura 21. Prueba de programación en JSFIDDLE

6.3. Programación de la aplicación

Se ha realizado la carga de elementos en los documentos mediante JSON para no realizar una recarga de la página cada vez que se señale un capítulo como visto. Así se consigue un ahorro en tiempo y en ancho de banda, dos de las cosas más importantes para que el usuario obtenga una buena experiencia con la aplicación.

6.3.1. Específico para Android

Como ya se ha comentado anteriormente, aunque se ha desarrollado en html5 y éste sirve para desarrollar aplicaciones multiplataforma, me he centrado sólo en hacerla compatible con la plataforma Android.

A continuación se muestran los pasos seguidos para su funcionamiento.

6.3.1.1. Configuración de Eclipse para PhoneGap

A grandes rasgos, los pasos seguidos para el correcto funcionamiento de phonegap han sido:

- Copiar el archivo cordova-2.9.0.jar en libs
- Copiar la carpeta xml en libs\Android
- Añadir la librería anterior al build Path
- En MainActivity.java añadir `public class MainActivity extends DroidGap {...}`

6.3.1.2. Configuración de los archivos de Android

La programación se ha realizado para móviles Android desde la versión 8, aunque la versión para trabajar adecuadamente con la misma sería de la 17 en adelante. Esto se ha configurado en el archivo AndroidManifest.xml, que está situado en la carpeta de assets.

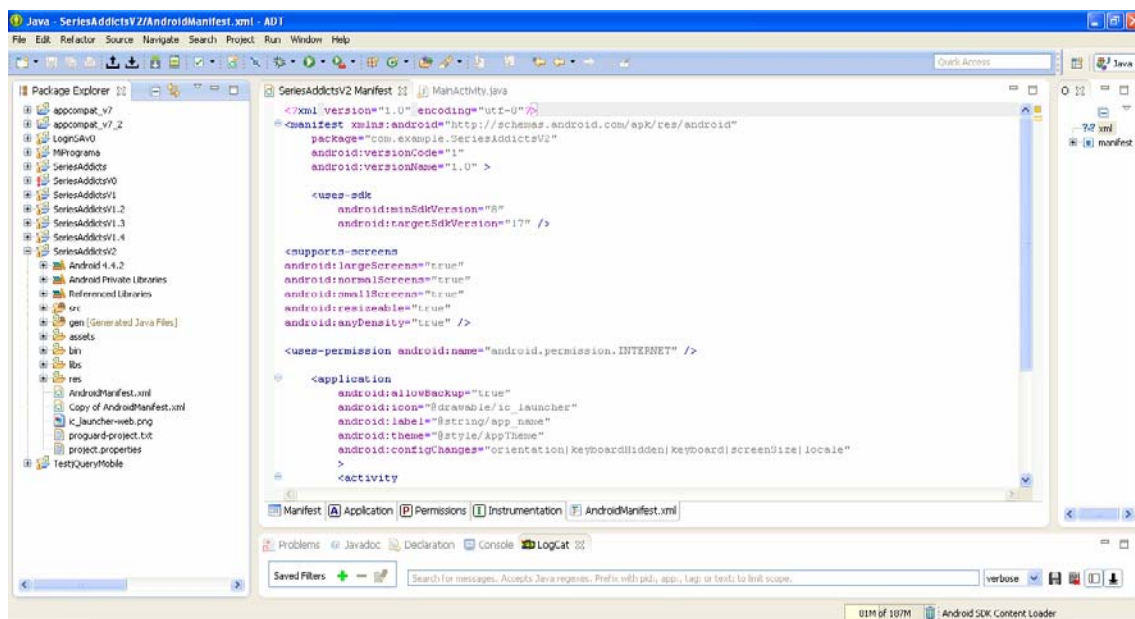


Figura22. Archivo AndroidManifest.xml

Este archivo también sirve para la configuración de los permisos de la aplicación. Uno de los permisos fundamentales para la aplicación es `<uses-permission android:name="android.permission.INTERNET" />` y sirve para que la aplicación pueda tener acceso a Internet, requisito indispensable para que funcione nuestra aplicación,

Estos permisos son los que va a tener que aceptar el usuario a la hora de instalar la aplicación, y es importante configurarlos de forma correcta por dos razones:

- Si la aplicación tiene todos los permisos, puede ser utilizada por otras aplicaciones *malware* como fuente para obtener datos confidenciales y estafar al usuario como por ejemplo suscribiéndolo a servicios SMS Premium, todo por no haber dedicado unas horas a su correcta configuración. Podemos ver un ejemplo en un artículo muy interesante de Chema Alonso en el que habla de la aplicación "linterna molona", <http://www.elladodelmal.com/2014/01/la-estafa-de-la-linterna-molona-que.html>.
- La otra razón para configurar correctamente los permisos es para que aquellos usuarios que se preocupen de leerlos, que son un porcentaje no muy grande, pero si ven que nuestra app necesita más permisos de los normales para una aplicación de este tipo van a recelar y así perderemos una descarga y posiblemente un cliente para nuestras futuras aplicaciones.

A continuación se muestra el archivo AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.SeriesAddictsV2"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <supports-screens
        android:largeScreens="true"
        android:normalScreens="true"
        android:smallScreens="true"
        android:resizeable="true"
        android:anyDensity="true" />

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

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

        android:configChanges="orientation|keyboardHidden|keyboard|screenSize|
        locale"
        >
        <activity
            android:name="com.example.SeriesAddictsV2.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```


Otro de los archivos más importantes que hay en Android es el MainActivity.java. Está situado en /SeriesAddictsV2/src/com/example/SeriesAddictsV2/ y sirve para iniciar una actividad instancia, en este caso nuestro index.html, volcando los métodos específicos que corresponden a las etapas del ciclo de vida. Lo mostramos a continuación:

```
package com.example.SeriesAddictsV2;

import android.os.Bundle;

import org.apache.cordova.*;

import com.example.miprograma.R;

import android.app.Activity;
import android.view.Menu;

public class MainActivity extends DroidGap {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        super.loadUrl("file:///android_asset/www/index.html");
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it
        is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```

6.3.2. Estructura de la aplicación

La aplicación se ha desarrollado tanto para el lado de cliente como para el lado de servidor, puesto que hay ciertos lenguajes que no son aceptados por los dispositivos móviles. A continuación se detallan las bases de la estructura de la aplicación.

6.3.2.1. Base de Datos

Se han utilizado dos formas de acceso a los datos de la aplicación, una externa mediante MySQL y otra interna mediante localStorage. Se detallan a continuación:

El acceso a una base de datos externa:

La BBDD se ha creado en un servidor externo gestionada con MySQL5 y administrada con phpMyAdmin.

Consta de 5 tablas:

- Usuario: Contiene los datos de cada usuario. Nombre, apellidos, usuario y contraseña para el login, email y un id de usuario. Se ha creado una clave primaria con el email para que no haya usuarios repetidos.
- Serie: Esta tabla contiene los principales datos de cada serie, su nombre, la puntuación, la dirección de la imagen, un campo de información y un id_serie único para cada serie.
- Series_usuario: En esta tabla se guarda qué series están siendo seguidas por cada usuario. Tiene el campo id_serie y el id_usuario para que se relacionen con sus respectivas tablas.
- Capitulo: Tabla que contiene los capítulos de cada serie. Contiene los campos id_capitulo, id_serie (para relacionarlo con su serie correspondiente) y temporada.
- capitulos_usuario: En esta tabla se guardan los capítulos que ya han sido vistos por cada usuario. Sus campos son id_usuario, id_serie e id_capitulo, cada uno correspondiente a su propia tabla.

Todas las tablas tienen un cotejamiento utf8_general_ci.

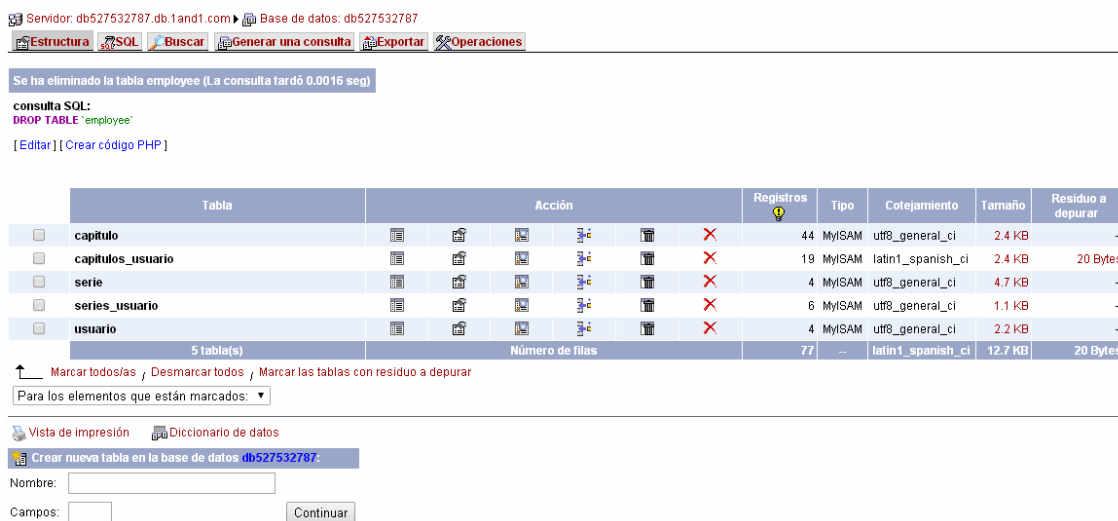


Figura 23. Gestor de BBDD utilizado

localStorage

La persistencia de datos en la aplicación se ha realizado utilizando localStorage[], que es la base de datos local para HTML5, y sirve principalmente para mantener o preservar datos del usuario. A continuación vemos un ejemplo de localStorage en el que hemos guardado la url del servidor donde están situados los archivos .php.

```
localStorage['serviceURL'] ="http://www.acudezamora.com/aaa/version7/services/";
```

El acceso nuevamente a la información guardada es el siguiente:

```
var serviceURL = localStorage['serviceURL'];
```

6.3.2.2. Programación del lado del servidor

Para acceder a los datos de cada tabla de la bbdd, se ha utilizado php, pero como este lenguaje no es aceptado por los móviles se ha desglosado del resto de aplicación y se ha subido cada archivo al servidor.

Todos los archivos .html de este TFC han necesitado de una conexión a la base de datos para obtener información personalizada de las series que sigue cada usuario y los capítulos que ha visualizado, por tanto todos han precisado de un archivo .php.

Estos archivos se han guardado en la carpeta Services, a la misma altura que la carpeta www.

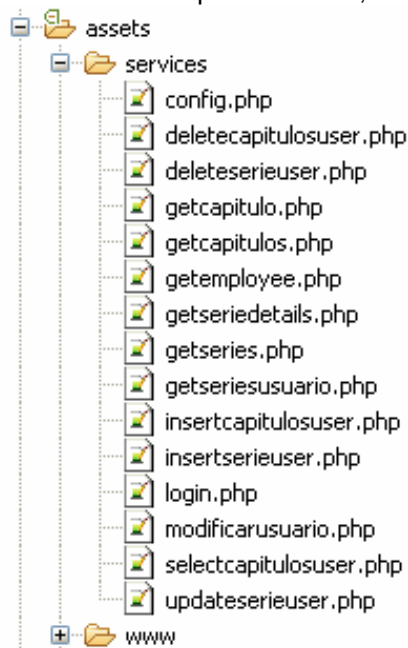


Figura 24. Archivos php

Cada uno de los archivos .php necesita de unos datos de conexión, por lo que se ha creado un archivo común llamado config.php con todos esos datos y después se ha incluido la llamada al mismo en todo los demás archivos .php.

A continuación vamos a mostrar uno de los archivos de interacción con la bbdd:

En la primera línea podemos ver el include que comentábamos anteriormente.

Se trata de una operación insert en el que cuando un usuario pulsa un capítulo como visto, este archivo guarda una nueva línea en la tabla capitulos_usuario con los datos del id de usuario, el id de la serie y el id del capítulo que recoge mediante GET y que se introducen en el sql mediante :id_usu, etc.

```

<?php
include 'config.php';

$sql="INSERT INTO capitulos_usuario ( id_usuario , id_serie,
id_capitulo ) VALUES (:id_usu, :id_ser, :id_cap)";

try {
    $dbh = new PDO("mysql:host=$dbhost;dbname=$dbname", $dbuser,
$dbpass);
    $dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $stmt = $dbh->prepare($sql);
    $stmt->bindParam("id_usu", $_GET[id_usu]);
    $stmt->bindParam("id_ser", $_GET[id_ser]);
    $stmt->bindParam("id_cap", $_GET[id_cap]);
    $stmt->execute();
} catch(PDOException $e) {
    echo '{"error":{"text":"' . $e->getMessage() . '}}';
}

?>

```

Vamos a ver otro ejemplo en el que el .php realiza una devolución codificando los ítems de cada fila (obtenidos mediante `fetchObject`) del resultado de salida en un json con `json_encode`.

```

<?php
include 'config.php';

    $sql="SELECT c.nombre as nombreCap, id_capitulo, temporada,
picture, s.id_serie as numSerie
    FROM capitulo as c inner join serie as s where
c.id_serie=s.id_serie and c.id_serie=:id";

try {
    $dbh = new PDO("mysql:host=$dbhost;dbname=$dbname", $dbuser,
$dbpass);
    $dbh->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $stmt = $dbh->prepare($sql);
    $stmt->bindParam("id", $_GET[id]);
    $stmt->execute();
    $cap = $stmt->fetchAll(PDO::FETCH_OBJ);
    $dbh = null;
    echo '{"items":"' . json_encode($cap) . '}';
} catch(PDOException $e) {
    echo '{"error":{"text":"' . $e->getMessage() . '}}';
}

?>

```

6.3.2.3. Programación del lado de cliente

La programación en el lado del cliente es la que se ejecuta en el terminal del usuario. En este TFC se ha realizado mediante HTML5, CSS3 y JavaScript.

Estos archivos se han colocado en la carpeta www, los de estilo en la subcarpeta css y los scripts en la carpeta de JavaScript.

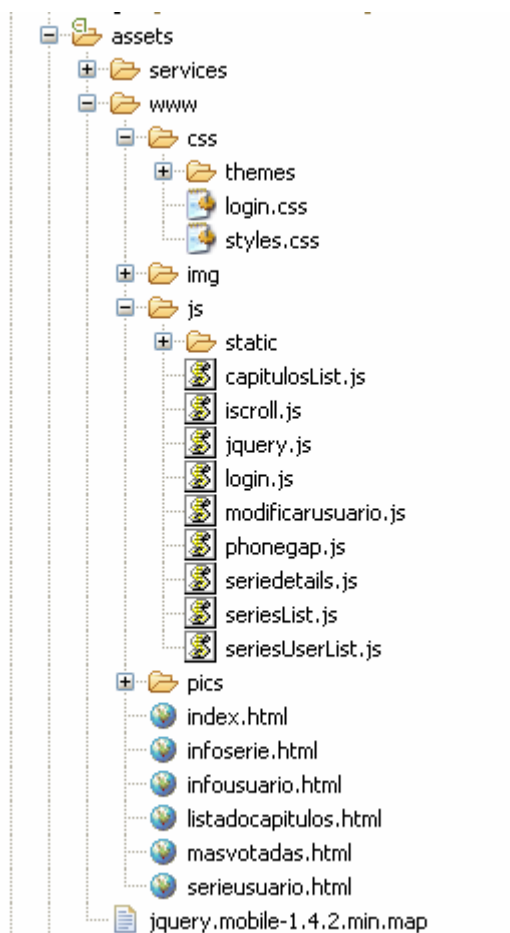


Figura 25. Archivos del lado de cliente

A mi parecer, una de las características más importantes en este proyecto ha sido la utilización de JSON, que como hemos explicado en el capítulo 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. Y por tanto se trata de programación de lado de cliente.

Vamos a analizar por ejemplo su uso en el archivo seriesUserList.js, que junto con serieusuario.html y getserieusuario.php son los encargados de mostrar las series que cada usuario ha decidido seguir.

La carga completa de la página con este método sólo ha tardado 2.91 segundos.

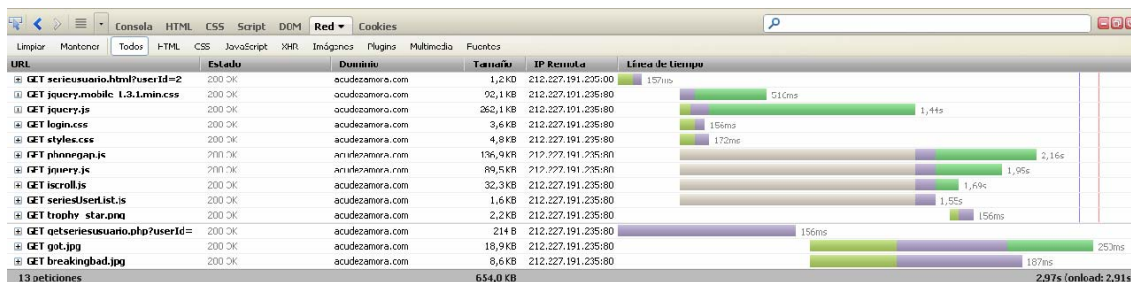


Figura 26. Características del equipo en el que se ha programado la aplicación.

Podemos ver que lo que menos tiempo tarda es la carga del código html inicial:



Figura 27.

Y lo que más tarda es la carga de las imágenes de cada serie. Aunque ahora no es un tiempo realmente significativo, se debe de estudiar un sistema de carga de imágenes más inteligente en futuras versiones.



Figura 28.

La devolución de los datos de la base de datos se ha realizado mediante la función getJSON(), en la que le indicamos la url del archivo php que hace la consulta y el id del usuario actual.



Figura 29. JSON devuelto con getJSON()

Esta forma de devolución se ha programado así tanto para las listas de series como para las listas de capítulos. Adjunto la función para entender mejor lo explicado:

```
function getSeriesList() {
    $('#busy').show();
    $.getJSON(serviceURL + 'getseriesusuario.php?userId='+userId,
function(data) {
    $('#busy').hide();
    $('#seriesList li').remove();
    series = data.items;
    $.each(series, function(index, key) {

        $('#seriesList').append('<li id="lista"><a
for="lista" href="infoserie.html?id=' + key.id_serie + '">' +
        '' +
        '<p class="line1">' + key.nombre + '</p>'
+
        '<span class="bubble">' + key.puntuacion
+ '</span></a></li>');
        localStorage['seriesUsuario']+=key.id_serie+' ';
    });
    setTimeout(function(){
        scroll.refresh();
    });
});
}
```

El JSON devuelto se va recorriendo y se va añadiendo en formato html al elemento lista con id *seriesList*, que se ha definido previamente en el documento .html.

De esta forma se crea una lista dinámica que no se percibe a ojos del usuario, puesto que tarda muy poco tiempo en crearse (157 milisegundos), pero que da una sensación de limpieza y eficiencia muy adecuada para nuestro propósito.

6.3.3. Validación de la aplicación

Se ha programado en todo momento siguiendo las directivas de HTML5 para que no existan problemas de compatibilidades con ningún navegador o con ningún dispositivo en el que se vaya a usar la aplicación. Para ello se han ido validando las páginas con la página de validación del W3C. Aquí se puede ver el resultado de la validación de una de las páginas:

The screenshot shows the W3C Markup Validation Service interface. At the top, it says "W3C® Markup Validation Service" and "Check the markup (HTML, XHTML, ...) of Web documents". Below this, there are links for "Jump To: Notes and Potential Issues" and "Congratulations - Icons". A green banner states "This document was successfully checked as HTML5!". Below the banner, a table shows the validation details:

Result:	Passed, 1 warning(s)
Address:	<input type="text" value="http://www.acudezamora.com/aaa/version7/www/index.html"/>
Encoding:	utf-8 (detect automatically)
Doctype:	HTML5 (detect automatically)
Root Element:	html

Below the table, there are logos for "W3C VALIDATOR Suite" and "mozilla". Text on the page includes: "Try now the W3C Validator Suite™ premium service that checks your entire website and evaluates its conformance with W3C open standards to quickly identify those portions of your website that need your attention." and "The W3C validators are developed with assistance from the Mozilla Foundation, and supported by community donations. Donate and help us build better tools for a better web." There is also a "4880" counter and a "Flattr" button.

Figura 30. Validación de HTML5 con W3C

6.4. Usabilidad de la aplicación

En el prototipo de la aplicación se habló de mostrar una imagen inicial como presentación de la aplicación. Esta pantalla se llama splash screen y en principio parece buena idea, puesto que se muestra mientras la aplicación se carga, pero leyendo en profundidad, se desaconseja su uso por temas de usabilidad de la aplicación y es que puede resultar tedioso ver siempre la misma pantalla durante tres segundos cada vez que se inicia la aplicación.

6.5. Problemas encontrados durante el desarrollo

jQueryMobile

El principal problema que he tenido durante este proyecto ha sido a la hora de juntar las pruebas unitarias que había ido programando, ya que unas las tenía programadas con jQueryMobile (como por ejemplo el login), pero otras no las había conseguido integrar correctamente con el framework y ante la falta de tiempo las realicé con jQuery.

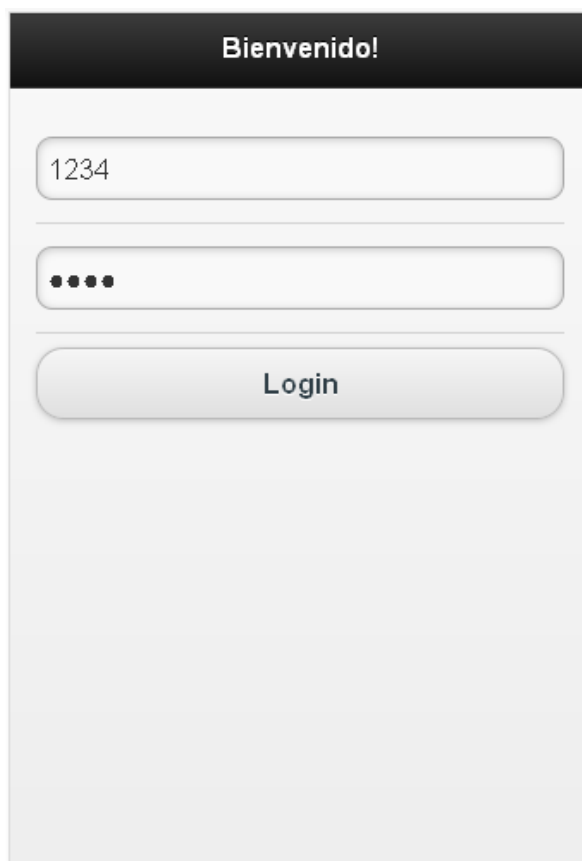


Figura 31. Prueba unitaria de Login con jQueryMobile

El problema que tuve al programar en jQueryMobile fue que al tener las 5 páginas en un solo documento html no supe hacer funcionar la llamada de cada archivo javascript correspondiente a cada una de las páginas, y que se cargase en cada página la información relacionada con el elemento seleccionado en la página anterior. Por lo que decidí separarlos en varios archivos, cada uno con su correspondiente archivo javascript.

Ha sido un problema sobre todo de tiempo, que es la principal carencia de este proyecto, pero tuve que elegir entre la funcionalidad o el diseño que proporciona jQuery, y elegí funcionalidad.

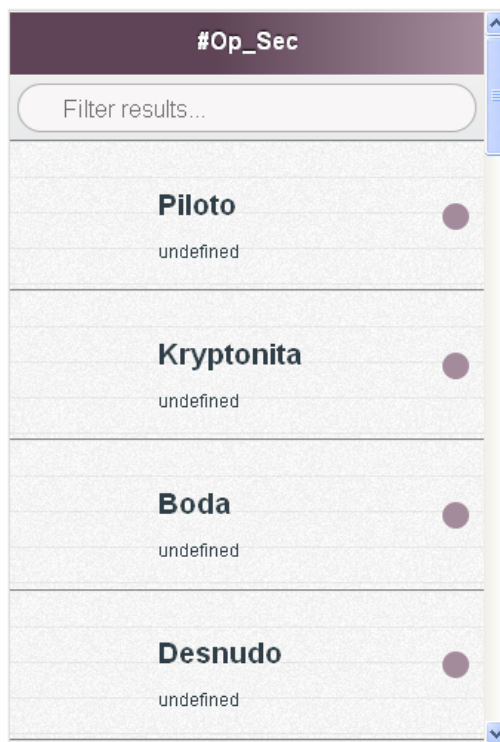



Figura 32. Prueba unitaria del listado de capítulos con jQueryMobile

Problemas de seguridad

En la primera prueba de desarrollo no se conseguía obtener información de la base de datos. Esto era debido a un problema con la seguridad, ya que los navegadores habían detectado el contenido mostrado como no seguro y lo bloqueaban.

La solución se encontró en esta página: <https://support.mozilla.org/es/kb/como-afecta-mi-seguridad-el-contenido-no-seguro#firefox:winxp:fx28>. Sólo hay que hacer clic en el icono del escudo  en la barra de direcciones y escoger *Desactivar protección* en esta página en el menú desplegable.

Pequeños problemas de programación

Durante el desarrollo y testeo de la app, se han encontrado los típicos errores de programación.

Vamos a explicar uno de ellos. El problema ha sido el chequeo de la serie vista en la página de información de la serie (únicamente volviendo de la pestaña de capítulos), puesto que las series vistas se guardan en sesión al cargar la página, pero al chequear una nueva serie para seguirla no se recargan las variables de sesión hasta volver a la pantalla de inicio, durante la navegación por el resto de aplicación no vuelve a calcularlas, por tanto la serie no aparecía como marcada hasta que no se volvía al inicio.

Para subsanarlo se creó esta pequeña función, que vuelve a guardar en local las series correctas después de hacer clic en una de ellas.

//Función para añadir a la variable de sesión de las series seguidas, la que se chequea en este momento.

```
function modifySeriesUser(id, flag){
    var nuevaLocalStorage="";
    //alert(flag);
    if(flag==true){
        for (var i = 0; i < seriesUser.length; i++) {
            nuevaLocalStorage+=seriesUser[i];
        }
        nuevaLocalStorage+=id;
    } else if(flag==false){
        for (var i = 0; i < seriesUser.length; i++) {
            if (seriesUser[i]!=id){
                nuevaLocalStorage+=seriesUser[i];
            }
        }
    }
    localStorage['seriesUsuario']=nuevaLocalStorage;
    //alert(nuevaLocalStorage);
}
```

6.5.1. Funcionalidades no implementadas

A continuación se describen las funcionalidades pensadas en el prototipo pero no desarrolladas:

- La pantalla de splash no se ha incluido en la versión final de la aplicación debido a la gran reticencia que he encontrado en varias webs especializadas en desarrollo de apps. Ya que en estas decían que afectaba a la usabilidad y que los usuarios terminaban por desinstalarse las aplicaciones que contenían este tipo de pantallas. Al no ser necesaria para la aplicación, se ha prescindido de ella.
- Menú lateral deslizable:
No fue implementado debido a problemas de compatibilidad de la versión utilizada de jQuery con la versión de jQueryMobile. Se intentó dar una solución durante varios días, pero ante la falta de tiempo se decidió prescindir de él y programar con CSS3.
- Alarmas de nuevos capítulos o series: No se ha implementado también debido a falta de tiempo, tampoco era necesaria para la primera versión de la aplicación, aunque si resultará útil una vez que se haga (ya fuera del ámbito del proyecto).

6.6. Mejoras de la aplicación para futuras versiones

A continuación se expone un listado de las mejoras comentadas a lo largo del TFC:

- Integración de jQueryMobile con la aplicación para obtener un diseño más fluido y elegante.
- Integrar todas las funcionalidades que disponía el mockup y que no se han podido introducir por falta de tiempo.
- Estudiar un sistema de carga de imágenes más inteligente en futuras versiones.
- Adaptación para el resto de plataformas, porque aunque Android sea un porcentaje importante del mercado, estamos perdiendo posibles futuros clientes. Gracias a HTML5 y a PhoneGap esta adaptación se podrá realizar de una forma relativamente sencilla, abriendo así nuevos horizontes en el mercado de las aplicaciones móviles.
- Modificar la parte del login de usuario. Es algo que no le gusta a los usuarios, porque entorpece un poco la fluidez. Se estudiará una forma de logeo único en el que sólo se cambie de usuario cuando se seleccione un botón de logout en vez de cada vez que se inicia la aplicación.
- Hacer que la base de datos sea autogestionable. Actualmente, la bbdd sólo la puede gestionar un administrador, pero es un trabajo que lleva mucho esfuerzo y dedicación sin obtener ninguna recompensa directa por ello. La posible mejora es realizar una funcionalidad en la que los propios usuarios sean los que vayan introduciendo nuevas series y sus capítulos correspondientes. Habría que poner bastante atención a muchos detalles para no

7. Conclusiones

Series Addicts es una aplicación móvil programada en HTML5, CSS3, JavaScript, PHP, MySQL, y JSON. Tecnologías ya comentadas a lo largo de esta memoria. Dicha aplicación se ha programado para que sea adaptable a cualquier tipo de pantalla, y puede utilizarse tanto como en un navegador normal como en un dispositivo móvil.

Se ha desarrollado en un principio para la plataforma Android, pero al estar programada con PhoneGap, puede convertirse en multiplataforma en un periodo muy corto de tiempo, sobre todo comparándolo con lo que llevaría desarrollar la misma aplicación de forma nativa. Esa es una de las grandes ventajas de HTML5 y el motivo de haber seleccionado esta tecnología para desarrollar Series Addicts.

Con este Trabajo Final de Carrera se ha conseguido lograr el objetivo principal propuesto: obtener una aplicación rápida y fiable con la que sus potenciales usuarios puedan controlar las series que siguen actualmente. Sin embargo, esta es sólo una primera versión de Series Addicts. Será necesario implementar las funcionalidades descritas en el capítulo anterior para conseguir una aplicación más completa.

En el aspecto personal, me gustaría destacar que el proyecto desarrollado me ha permitido adquirir conocimientos que he podido aplicar posteriormente en mi vida profesional, como programadora web. En primer lugar, JQuery, que en este proyecto se ha usado constantemente para el acceso y manejo de los elementos del DOM, lo he utilizado en mi trabajo para mejorar la eficiencia de mis scripts. Por otro lado, JSON, utilizado aquí en la devolución de la información de las bases de datos, lo he aplicado en mi trabajo consiguiendo tiempos de carga de las páginas de hasta un 70% menos de tiempo.

Hay un concepto que si bien no es tan importante desde el punto de vista técnico, lo es desde un punto de vista más comercial: la usabilidad. Y este es un tema sobre el que se ha prestado una gran atención en este trabajo. Es evidente, que el primer objetivo de cualquier aplicación desarrollada por un programador es que funcione de forma correcta. Sin embargo, por muy bien programado que esté, si el software desarrollado no es agradable y fácil de usar, el usuario final no lo utilizará, perdiendo así la oportunidad de llegar al mercado.

También hay una serie de puntos débiles que hay que comentar, relacionados con mi planteamiento inicial en el Trabajo Fin de Carrera.

El principal ha sido mi propia falta de experiencia en la materia. Creo que fui demasiado ambiciosa en cuantos a los objetivos propuestos inicialmente, con la intención de desarrollar una aplicación muy completa ya desde su primera versión. Sin embargo, dada la limitación de tiempo existente para la realización del proyecto, habría sido más lógico proponer una aplicación con menos funcionalidades pero mejor desarrolladas y, sobre todo, mejor diseñadas; ya que al no haber podido contar con la ayuda de jQueryMobile, la aplicación ha sido diseñada directamente con CSS3.

Otro error cometido ha sido el de no haber empezado antes con las pruebas unitarias de programación, puesto que si lo hubiera hecho desde el primer día, a la hora de empezar la fase de implementación, ya hubiera tenido una mayor soltura.

Sin embargo, a pesar de la sensación continua durante estos meses de que me hacían falta días de 30 horas, creo que ha sido una experiencia enriquecedora, y sobre todo ha sido agradable ver que la mayoría de las cosas utilizadas tiene una aplicación práctica y que todas las horas dedicadas no van a ser en vano. En un futuro

próximo (aunque ya fuera del ámbito de este proyecto), pretendo completar la aplicación con las nuevas funcionalidades mencionadas anteriormente, con la intención de crear una aplicación que pueda ser utilizada a diario por todos los adictos a las series.

8. Bibliografía

<http://tomsplanner.es>
<http://phonegap.com/>
<http://es.wikipedia.org/wiki/PhoneGap>
<http://jquerymobile.com/>
<http://www.lukew.com/ff/entry.asp?1649>
<http://blog.aplicacionesmovil.com/aplicaciones-celular/html5-y-aplicaciones-moviles/>
<http://www.molecularts.com/blog/2012/04/24/desarrollo-de-aplicaciones-moviles-con-jquery-mobile-android-sdk-y-phonegap-parte-2>
<http://www.redeszone.net/2013/03/14/como-configurar-y-utilizar-git-en-eclipse/>
<http://geekytheory.com/json-i-que-es-y-para-que-sirve-json/>
<http://users.dcc.uchile.cl/~psalinas/uml/modelo.html>
<http://www.seas.es/blog/informatica/tipos-de-relaciones-en-diagramas-de-casos-de-uso-uml/>
http://es.wikipedia.org/wiki/Manifiesto_%C3%A1gil
<http://www.40defiebre.com/recursos-diseno-aplicaciones-moviles/>
<http://todoserries.com/16-cosas-que-no-sabiamos-de-breaking-bad/>
<http://www.imdb.com/>
<http://series.ly/scripts/>
<http://blog.aplicacionesmovil.com/aplicaciones-celular/html5-y-aplicaciones-moviles/>
<http://www.molecularts.com/blog/2012/04/24/desarrollo-de-aplicaciones-moviles-con-jquery-mobile-android-sdk-y-phonegap-parte-2>
http://www.aprenderaprogramar.es/index.php?option=com_content&view=article&id=107:firebug-una-extension-gratuita-de-firefox-para-programadores-y-disenadores-web&catid=57:herramientas-informaticas&Itemid=179
<http://www.nerdcoder.com/tutorial-html5-localstorage-base-de-datos-local-para-todos/webintenta.com/trabajar-con-checkboxes-en-jquery.html>
<https://support.mozilla.org/es/kb/como-afecta-mi-seguridad-el-contenido-no-seguro#firefox.winxp.fx28>

8.1. Guías de estilo:

<http://www.appdesignbook.com/es/contenidos/disenovisual/>
<http://developer.android.com/design/index.html>
https://developer.apple.com/library/ios/design/index.html#//apple_ref/doc/uid/TP40013289
<http://dev.windowsphone.com/en-us/design>
<http://www.mozilla.org/en-US/styleguide/products/firefox-os/>

8.2. Guías de programación:

<http://www.w3schools.com/>