

Memoria PFC



SimplyRssReader (Lector RSS)

Programación de aplicaciones para móviles usando HTML5

Juan Lillo de la Fuente

PFC – Ingeniería Informática

Consultor: Carles Sanchez Rosa

Octubre 2013

Índice

1.- Definición del Proyecto	5
1.1.- Descripción del proyecto	5
1.2.- Objetivos	6
1.3.- Metodología	8
1.4.- Plan de Trabajo	9
1.5.- Diagrama Gantt	11
2.- Estado del arte	12
2.1.- Feedly	12
2.2.- Pocket	13
2.3.- Flipboard	14
3.- Entorno de trabajo	15
3.1.- Eclipse	15
3.2.- Android SDK Tools	16
3.3.- ADT	16
3.4.- Evolus Pencil	17
4.- Tecnología ha usar	18
4.1.- HTML5	18
4.2.- CSS3	18
4.3.-JavaScript	19
4.3.1.- jQuery Mobile	19
4.4.- PhoneGap	20
4.5.- Google TTS	21

Índice

5.- Análisis y Diseño	22
5.1.- Historias de Usuarios	22
5.1.1.- Añadir un feed	22
5.1.2.- Modificar un feed	23
5.1.3.- Eliminar un feed	23
5.1.4.- Listar los feeds	24
5.1.5.- Listar las noticias de un feed	24
5.1.6.- Abrir / Ver un feed	25
5.1.7.- Abrir / Ver una noticia	25
5.2.- Diseño del Prototipo	26
5.2.1.- Pantalla de inicio	26
5.2.2.- Pantalla de añadir feed	27
5.2.3.- Pantalla principal o pantalla de listado noticias	28
5.2.4.- Pantalla contenido de la noticia	29
5.3.- Flujo de Pantallas	30
6.- Implementación	31
6.1.- Estructura de las páginas	31
6.2.- Páginas dinámicas	34
6.3.- Implementación lector Rss	34
6.4.- Implementación Aplicación Android	36
6.5.- Implementación Accesibilidad	40
7.- Manual de Usuario	41

Índice

8.- Conclusiones	44
8.1.- Conclusiones	44
8.2.- Líneas de futuro	46
9.- Entregables PFC	47
Anexo I - Glosario	48
Anexo II – Referencias y Fuentes Consultadas	50
Anexo III – Índice de Figuras	53

1.- Definición del Proyecto

1.1.- Descripción del proyecto

El Proyecto Final de Carrera (PFC), llamado “SimplyRssReader”, es un proyecto dentro del área de las aplicaciones móviles desarrollado con tecnología HTML5.

El PFC tiene como finalidad la creación de una aplicación para dispositivos móviles donde poder leer las diferentes noticias de los lugares donde estamos suscritos, en los diversos formatos existentes (RSS, Atom, XML, etc).

Los usuarios han de poder visualizar las diferentes noticias, agrupadas por lugares web y/o categorías obtenidas de los diferentes feeds. Además, se ha de poder buscar la opción de poder añadir nuevos feeds a los existentes y compartirlos por las diferentes redes sociales.

El público al que va destinada esta aplicación es un público en general, es decir, todo usuario que quiere estar informado en todo momento de lo que le interesa y que usa dispositivos móviles.

1.- Definición del Proyecto

1.2.- Objetivos

Objetivos generales:

El principal objetivo de este proyecto es la creación de una aplicación, la cual permite mantener informado a los usuarios de los temas que más le interesan. Además, esta aplicación tiene que permitir a los usuarios interactuar con esta permitiendo añadir nuevos feeds, etc.

Otro de los objetivos principales de este proyecto es la creación de una aplicación que sea accesible para todo el mundo (personas ciegas y con deficiencia visual incluidas).

En conclusión, el objetivo final se podría resumir en: crear una aplicación simple que permita mantener informado a todos los usuarios y sacar el máximo provecho de la aplicación que tiene en su dispositivo.

1.- Definición del Proyecto

Objetivos específicos:

Los objetivos específicos de la aplicación son los siguientes:

- Poner en práctica los conocimientos sobre gestión de proyectos, los métodos y procesos de la creación de programas y las técnicas de ingeniería actuales disponibles para el desarrollo de aplicaciones móviles.
- Desarrollar una aplicación de código abierto (licencia GPL, Creative Commons).
- Crear una aplicación, en primer lugar para Android, y en futuros proyectos en el resto de plataformas, utilizando los lenguajes HTML5, CSS3 y javascript.

1.- Definición del Proyecto

1.3.- Metodología

La metodología que utilizaremos para realizar el proyecto tiene que ser una metodología ágil, como por ejemplo Kanban o SCRUM. Esta metodología nos permite visualizar el trabajo y las diferentes hitos y/o fases del ciclo de producción, medir el tiempo de cada tarea, etc.

La elección de esta metodología es debido, principalmente a la duración del proyecto es reducida, unos 3,4 meses aproximadamente, y las diferentes entregas existentes en el proyecto (5 entregables).

Por consiguiente la metodología que se utilizará para realizar el proyecto es Kanban. Esta metodología nos permitirá visualizar el trabajo y las fases del ciclo de producción, determinar el límite del “trabajo en curso” y medir el tiempo completo de una tarea.

Básicamente lo que se hace es dividir el trabajo en tareas, y cada tarea se encuentra en uno de los tres estados posibles (por realizar, realizando y realizado).



Figura 1.1: Ejemplo metodología Kanban

1.- Definición del Proyecto

1.4.- Plan de Trabajo

El trabajo ha realizar es el siguiente:

FASE I: Elaboración del plan de trabajo y preparación del entorno de trabajo

Del 18 de septiembre al 9 de octubre de 2013

- Concepción de la idea.
 - Estudio de frameworks HTML 5, API's, metodologías, etc.
 - Preparación del entorno de trabajo.
 - Elaboración del plan de trabajo.
-

FASE II: Análisis y diseño del proyecto

Del 10 de octubre al 30 de octubre de 2013

- Estudio HTML5, CSS3 i javascript.
- Pruebas iniciales del frameworks.
- Análisis y diseño de la aplicación.
- Prototipo Interfaz de usuario.
- Actualizar memoria.

1.- Definición del Proyecto

FASE III: Implementación

Del 31 de octubre al 27 de noviembre de 2013

- Implementación de la aplicación.
- Gestión de contenidos.
- Actualizar memoria.

FASE IV: Pruebas, memoria y presentación

Del 28 de noviembre al 8 de enero de 2014

- Pruebas de la aplicación.
- Creación Manual de uso.
- Creación y presentación virtual.
- Finalización Memoria.

1.- Definición del Proyecto

1.5.- Diagrama de Gantt

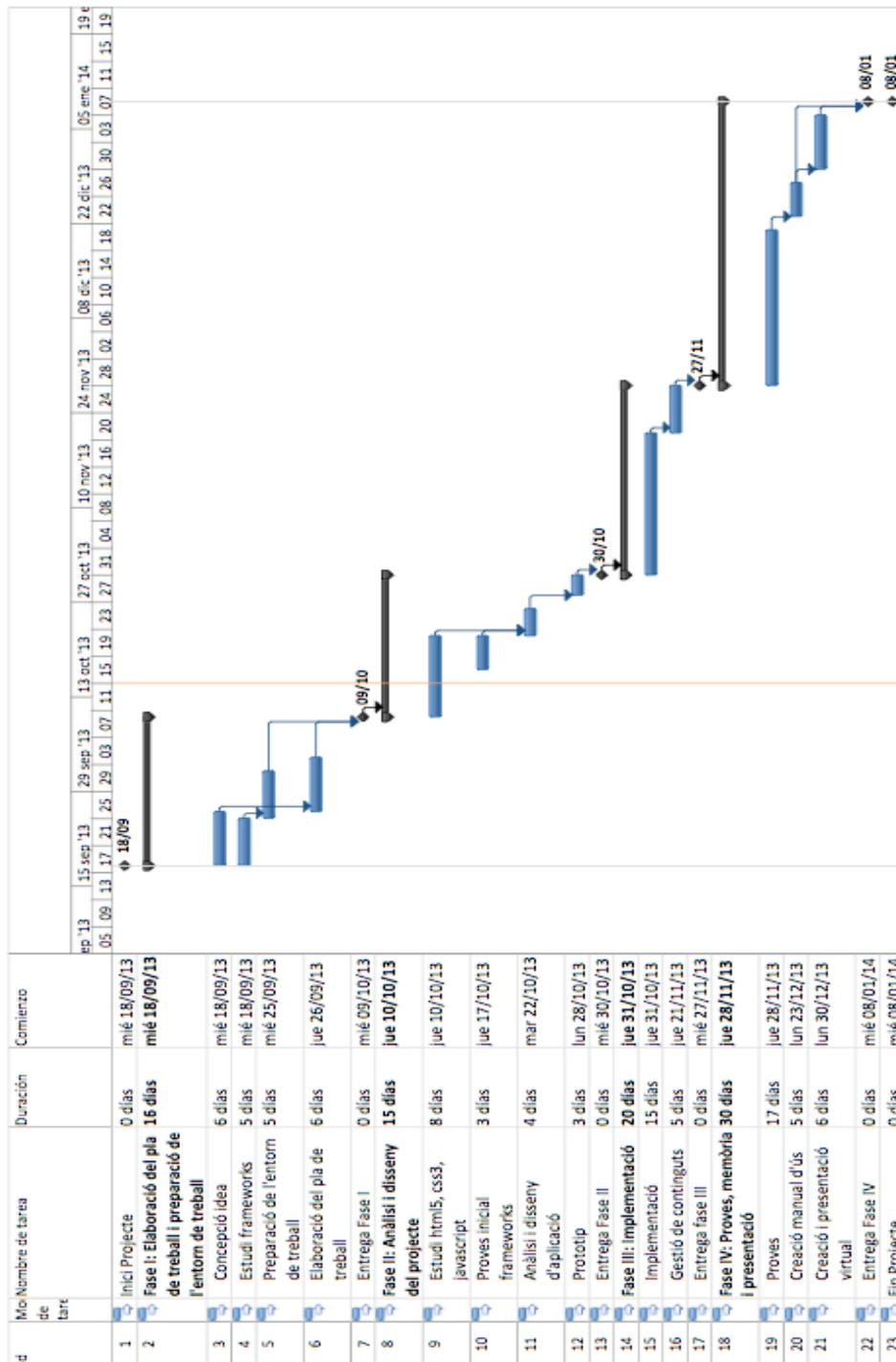


Figura 1.2: Diagrama de Gantt de la planificació i tasques del projecte

2.- Estado del Arte

Entendemos “Estado del arte” como todos los desarrollos realizados con tecnologías similares y enfocado al mismo público que el de este proyecto. Las aplicaciones encontradas, son de las más aceptadas y usadas dentro del sistema Android. Se trata de un estudio muy básico del mercado, sobre las diferentes aplicaciones similares existentes durante la realización de este proyecto, con el único objetivo de valorar las semejanzas y diferencias entre las diferentes aplicaciones.

2.1.- Feedly

Feedly es la aplicación creada por DEvHD en 2008. Es un gestor de noticias para varios navegadores webs, dispositivos móviles, que podemos encontrar tanto en Android como en iOS, y recientemente como un servicio basado en la nube.

Es una aplicación que podemos encontrar totalmente gratuita en las diferentes stores de los dispositivos móviles.

Lo podemos encontrar en Google Play:

<https://play.google.com/store/apps/details?id=com.d.evhd.feedly>



Figura 2.1: Imagen aplicación Feedly

2.- Estado del Arte

2.2.- Pocket

Pocket es un gestor de noticias que se usa en muchas redes sociales. Inicialmente se llamó Read it Later se creó en agosto de 2007.

La aplicación te permite guardar un artículo para leerlo después. El artículo se guarda en una lista, que se puede sincronizar con todos tus dispositivos móviles.

Pocket esta disponible tanto para Android como para iOS.

Es una aplicación que podemos encontrar totalmente gratuita en las diferentes stores de los dispositivos móviles.

Lo podemos encontrar en Google Play:

<https://play.google.com/store/apps/details?id=com.ideashower.readitlater.pro>



Figura 2.2: Imagen aplicación Pocket

2.- Estado del Arte

2.3.- Flipboard

Flipboard es una aplicación creada por Flipboard Inc. en el año 2010.

La aplicación añade los diferentes feeds de las páginas webs y los muestra en forma de revista. Inicialmente fue creada para el sistema operativo de iOS, pero debido a su gran acogida se amplió al resto de sistemas operativos móviles.

Es una aplicación que podemos encontrar totalmente gratuita en las diferentes stores de los dispositivos móviles.

Lo podemos encontrar en Google Play:

<https://play.google.com/store/apps/details?id=flipboard.app>

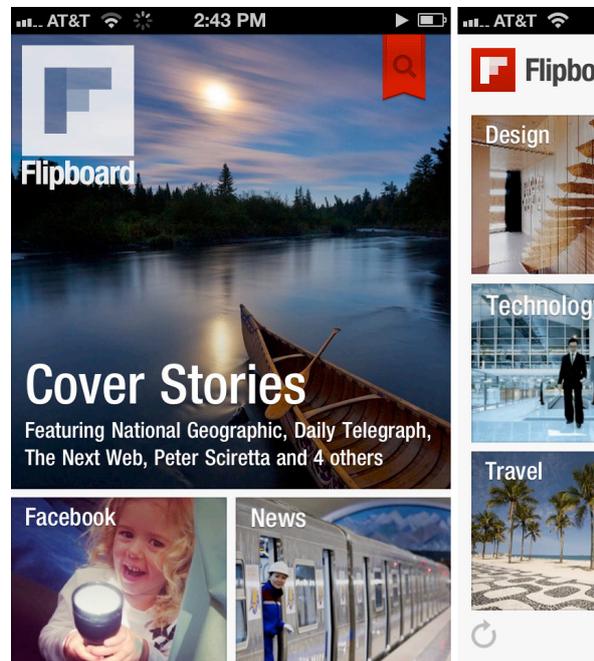


Figura 2.3: Imagen aplicación Flipboard

3.- Entorno de Trabajo

Para realizar el proyecto, he optado por utilizar los siguientes entornos de trabajo.

3.1.- Eclipse

Eclipse es un entorno de desarrollo integrado (IDE),

multiplataforma, de código abierto, modular y preparado para programar aplicaciones en diversos lenguajes, incluyendo HTML5, CSS y javascript.

El principal uso que le vamos a dar a este IDE es, por un lado, trabajar con PhoneGap para el empaquetado de la aplicación y por otro, para implementar y testear la aplicación con el simulador de Android.

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

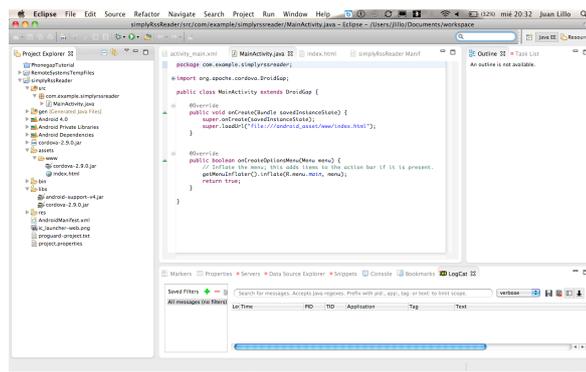


Figura 3.1: Eclipse con proyecto SimplyRssReader

3.- Entorno de Trabajo

3.2.- Android SDK Tools

Android SDK Tools es el conjunto de desarrollo de aplicaciones para Android. De entre las diferentes herramientas que trae, podemos destacar el SDK Manager y el Virtual Device Manager o gestor de dispositivos.

Android SDK Tools:

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

3.3.- ADT (Android Developer Tools)

ADT o Android Developer Tools es un plugin para Eclipse que ofrece las herramientas de desarrollo necesarias para realizar aplicaciones para el sistema operativo Android.

ADT:

<http://developer.android.com/intl/es/tools/sdk/eclipse-adt.html>

3.- Entorno de Trabajo

3.4.- Evolus Pencil

Evolus Pencil es una Evolus Pencil:

herramienta para la creación de prototipos GUI, que esta disponible en todas las plataformas.

<http://pencil.evolus.vn/>

Es una herramienta libre y de código abierto.

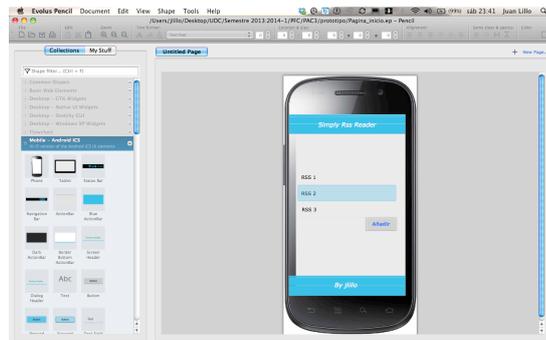


Figura 3.2: Evolus Pencil con proyecto SimplyRssReader

4.- Tecnología ha utilizar

La tecnología utilizada para realizar este proyecto es el siguiente:

4.1.- HTML5

HTML5 es la quinta revisión del lenguaje clásico de la WWW, el HTML (HyperText Markup Language). Es el lenguaje que predomina en la creación de páginas web.

En esta nueva versión se definen nuevos estándares de desarrollo web y dispositivos móviles, añadiendo nuevas características.



Figura 4.1: Icono HTML5

4.2.- CSS3

CSS3 es la tercera revisión/versión del lenguaje de estilo de hojas (CSS = Cascading Style Sheets). Se usa para describir el aspecto y formato de un documento escrito en lenguaje de marcas o HTML.



Figura 4.2: icono CSS3

4.- Tecnología ha utilizar

4.3.- JavaScript

JavaScript es un lenguaje de programación interpretado. Se define como un lenguaje orientado a objetos. Se utiliza principalmente en la parte del cliente, implementado como parte del navegador permitiendo mejoras en la interfaz de usuario y en las páginas webs dinámicas, etc.

4.3.1.- jQuery Mobile 1.3.2

jQuery Mobile es un Framework de interfaz unificada, con el único objetivo de hacer compatible jQuery con los navegadores de los diferentes dispositivos móviles.

jQuery Mobile ofrece una API actualizada, a parte de una completa librería de componentes para la creación de interfaces de usuario.

Su principal característica es que se consigue buenos resultados con poco código y además, permite una customización de las aplicaciones de manera muy sencilla.

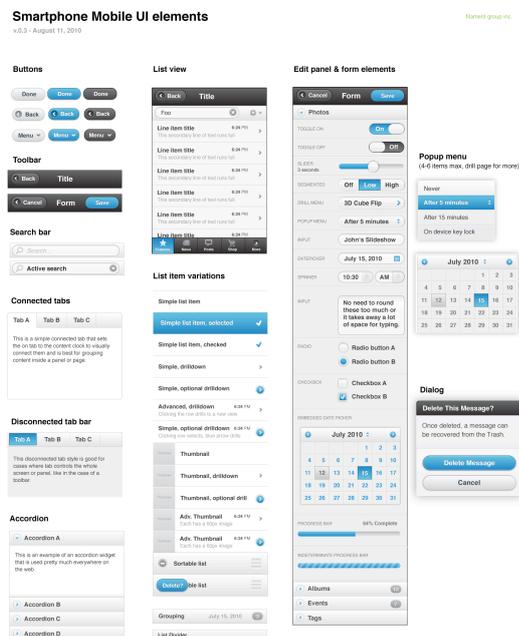


Figura 4.3: jQuery Mobile UI elements

4.- Tecnología ha utilizar

4.4.- PhoneGap 2.9

PhoneGap permite la programación de aplicaciones móviles para móviles con HTML5, CSS3 y JavaScript.

Las aplicaciones resultantes son híbridas, es decir, son una mezcla entre aplicaciones nativas y aplicaciones web. PhoneGap nos aporta APIs que nos permiten usar prácticamente todos los elementos del móvil (cámara. Geoposicionamiento, notificaciones, etc).

PhoneGap es de código abierto y de licencia abierta con el objetivo principal de crear aplicaciones multiplataforma con HTML5, CSS3 y JavaScript.

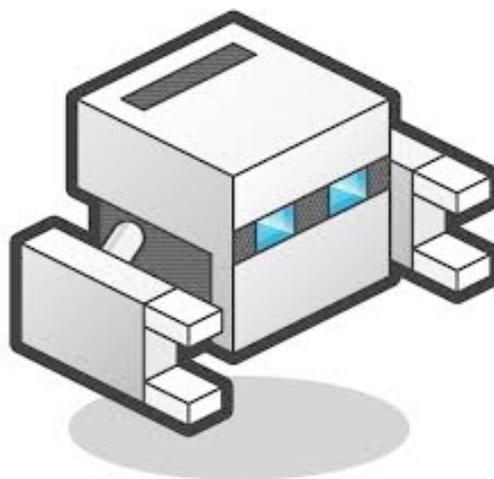


Figura 4.4: Icono PhoneGap

4.- Tecnología ha utilizar

4.5.- Google Text To Speech (TTS)

Google Text To Speech (TTS) es un servicio que ofrece Google traductor conjuntamente con su sistema de traducción de textos y de páginas web.

Google TTS dispone de una API para la transformación de texto en voz. Además permite la opción de elegir entre varios idiomas.

Existen muchos servicios TTS, como por ejemplo IVONA, pero estos son de pago.

5.- Análisis y Diseño

5.1.- Historias de usuario

Las historias de usuarios son una representación de los requerimientos de la aplicación escritos de forma clara utilizando el lenguaje común del usuario. Las historias de usuario son utilizados en las metodologías de desarrollo ágiles para la especificación de los requerimientos.

Para cada historia de usuario, valoramos el esfuerzo y su prioridad. Una vez finalizada la implementación, cada historia deberá cumplir sus pruebas de aceptación.

Las historias de usuario son las siguientes:

5.1.1.- Añadir un feed

Código:	Nombre: Añadir un feed
Historia: Como usuario, quiero disponer de la opción de poder añadir nuevos feeds.	
Pruebas de aceptación: <ul style="list-style-type: none">○ Validar que el nuevo feed se muestre por pantalla.○ Validar que el contenido del feed se muestre correctamente.	
Estimación: 4 días	Prioridad: Media

5.- Análisis y Diseño

5.1.2.- Modificar un feed

Código: U2	Nombre: Modificar Feed
Historia: Como usuario, quiero poder modificar un feed.	
Pruebas de aceptación: <ul style="list-style-type: none">○ Validar que las modificaciones se realicen sobre el feed correctamente	
Estimación: 4 días	Prioridad: Media

5.1.3.- Eliminar un feed

Código: U3	Nombre: Eliminar un feed
Historia: Como usuario, quiero poder eliminar, dar de baja un feed.	
Pruebas de aceptación: <ul style="list-style-type: none">○ Validar que el feed dado de baja ya no se muestra en pantalla.	
Estimación: 4 días	Prioridad: Media

5.- Análisis y Diseño

5.1.4.- Listar los feeds

Código: U4	Nombre: Listar los feeds
Historia: Como usuario, puedo acceder al listado de feeds existentes en la aplicación.	
Pruebas de aceptación: <ul style="list-style-type: none">○ Validar que el usuario puede listar los feeds.○ Validar que los datos de la lista son correctos.	
Estimación: 4 días	Prioridad: Alta

5.1.5.- Listar las noticias de un feed

Código: U5	Nombre: Listar los feeds
Historia: Como usuario, puedo acceder al listado de las noticias de un feed existentes en la aplicación.	
Pruebas de aceptación: <ul style="list-style-type: none">○ Validar que el usuario puede listar las noticias de un feeds.○ Validar que los datos de la lista son correctos.	
Estimación: 4 días	Prioridad: Alta

5.- Análisis y Diseño

5.1.6.- Abrir/ Ver un feed

Código: U6	Nombre: Abrir/ Ver un feed
Historia: La aplicación puede abrir un feed y cargar los elementos del feed.	
Pruebas de aceptación: <ul style="list-style-type: none">○ Validar que la aplicación carga el feed y que muestra el contenido de este.○ Validar que el contenido se muestra correctamente.	
Estimación: 3 días	Prioridad: Media

5.1.7.- Abrir/Ver una noticia

Código: U7	Nombre: Abrir/ Ver una noticia
Historia: La aplicación puede abrir el contenido de una noticia.	
Pruebas de aceptación: <ul style="list-style-type: none">○ Validar que la aplicación carga el contenido.○ Validar que el contenido se muestra correctamente.	
Estimación: 3 días	Prioridad: Media

5.- Análisis y Diseño

5.2.- Diseño del Prototipo

Un prototipo es una representación de la aplicación, que nos permite comunicar decisiones de diseño y evaluarlo antes de desarrollar la aplicación final. Su versatilidad hace que sea posible poder realizar modificaciones en su diseño.

5.2.1.- Pantalla de inicio



Figura 5.1: Pantalla de inicio

La pantalla de inicio consta de una interfaz sencilla donde se muestra un listado de los diferentes feeds existentes en la aplicación.

Debajo del listado nos encontramos un botón que nos permite añadir feeds nuevos (nos lleva a la pantalla de nuevo feed).

En la zona superior podemos encontrar el nombre de la aplicación y en la parte inferior el autor de la aplicación.

5.- Análisis y Diseño

5.2.2.- Pantalla de añadir feed



Figura 5.2: Pantalla de añadir feed

La pantalla de inicio consta de una interfaz sencilla donde se muestra un campo de texto, donde podemos introducir la nueva dirección del feed.

En la zona superior podemos encontrar el nombre de la aplicación y en la parte inferior el autor de la aplicación.

5.- Análisis y Diseño

5.2.3.- Pantalla principal o pantalla listado noticias

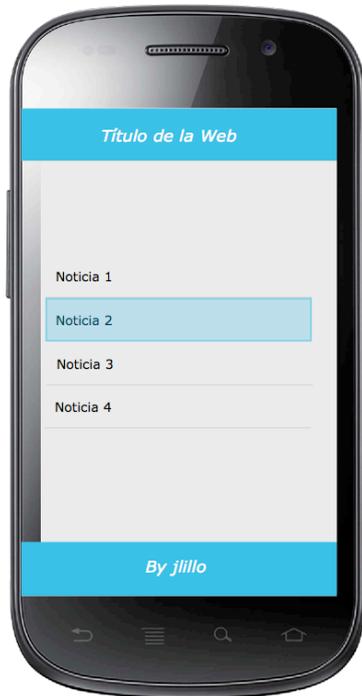


Figura 5.3: Pantalla listado noticias

La pantalla del listado de noticias es una pantalla sencilla con un listado de las noticias existentes de ese feed.

En la zona superior podemos encontrar el nombre del feed o de la web y en la parte inferior el autor de la aplicación.

5.- Análisis y Diseño

5.2.4.- Pantalla contenido de la noticia



Figura 5.4: Pantalla contenido noticia

La pantalla del contenido de la noticia, es una pantalla sencilla que nos mostrará el contenido de la noticia seleccionada en la pantalla del listado de noticias.

En la zona superior podemos encontrar el nombre de la noticia y un botón que nos permite volver a la pantalla del listado de noticias y en la parte inferior el autor de la aplicación.

5.- Análisis y Diseño

5.3.- Flujo de pantallas

La aplicación sigue el flujo de pantalla que se puede observar en la siguiente imagen.

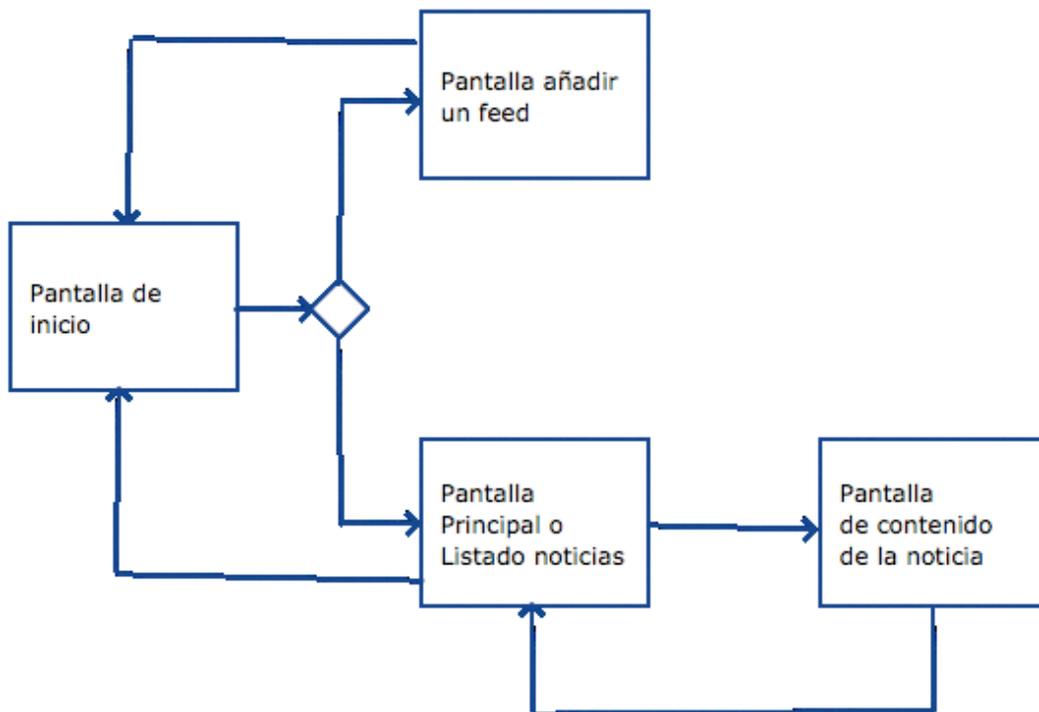


Figura 5.5: Diagrama flujo de pantallas

6.- Implementación

6.1.- Estructura de la página HTML

Como ya se ha especificado en el proyecto, la aplicación móvil se desarrolla mediante el Framework jQuery Mobile, que se basa en las tecnologías HTML5, CSS3 y JavaScript y es la que establecerá la estructura de las páginas.

El primer paso es crear la estructura del fichero HTML, con la declaración de los tags necesarios, así como incluir las librerías y hojas de estilos necesarias:

Declaración Doctype HTML5:

```
<!DOCTYPE HTML>  
<html>
```

Librerías jQuery Mobile:

```
<script src="js/jquery-1.7.min.js"></script>  
<script src="js/jquery.mobile-1.3.2.min.js"></script>  
<script src="js/lectorRss.js"></script>
```

Hoja de estilo CSS3:

```
<link rel="stylesheet" href="css/simpleRssReader.css" />  
<link rel="stylesheet" href="css/jquery.mobile.icons.min.css" />  
<link rel="stylesheet" href="js/jquery.mobile-1.3.2.min.css" />
```

Y por último, añadimos un tag recomendable, el meta “*viewport*”, que con los atributos necesarios permite que la aplicación se adapte a las dimensiones de la pantalla de cada dispositivo.

```
<meta name="viewport" content="width=device-width, initial-scale=1">
```

Para crear la aplicación, se usa la estructura multi-página, es decir, que toda la aplicación se desarrolla en un solo HTML, más concretamente **index.html**. Esto no es más que la unión de diversas páginas básicas en un mismo fichero, y jQuery se encargará de mostrarlo como si fueran páginas independientes.

6.- Implementación

Para crear esta estructura es necesario incluir en el cuerpo de la página diferentes bloques con el atributo *data-role="page"*. Cada uno de estos bloques representa una página. Estas páginas también contienen unas partes que configuran sus estructuras., que son header, content y footer.

En la sección header, nos encontramos con el nombre de la aplicación, el título del artículo, etc. y el botón atrás.

En el pie de página, o footer, nos encontramos con el nombre del autor, el año de creación de la aplicación y el copyright.

En el content es donde se carga, en cada caso, el contenido correspondiente de cada página.

La estructura básica de la página HTML es la siguiente:

```
<div data-role="page" id="page">
  <div data-role="header" data-theme="b">
    </div>

  <div data-role="content">
    </div>

  <div data-role="footer" data-theme="b">
    </div>
</div>
```

La estructura de la página HTML del proyecto es la siguiente (sección body):

```
<!-- Pantalla inicial donde se muestra el listado de Rss -->
  <div data-role="page" id="pageRss">
    <div data-role="header" data-position="fixed" data-
theme="b">
      <h1>SimplyRssReader</h1>
    </div>
    <div data-role="contentRss">
      <br><br>
      <ul id="linksList1" data-role="listview" data-
inset="true">
```

6.- Implementación

```
<li><a href="#mainPage" title="Blog Personal Juan Lillo" >Blog
Personal Juan Lillo</a></li>
    </ul>
    <br><br>
    <a class="botonAnyadir" href="#infoBoton" data-
rel="popup">Añadir nuevo Rss</a>
        <div data-role="popup" id="infoBoton">
            <p>Opción disponible en futuras
versiones<p>
                </div>
            <br><br>
        </div>
        <div data-role="footer" data-position="fixed" data-
theme="b">
            <h4>Lector RSS by jlillo - &copy; 2013</h4>
        </div>
    </div>

    <!-- Pantalla principal donde se muestra el listado de
noticias -->
    <div data-role="page" id="mainPage">
        <div data-role="header" data-position="fixed" data-
theme="b">
            <a href="#pageRss" data-rel="back">ATRAS</a>
            <h1></h1>
        </div>
        <div data-role="content">
            <div id="status"></div>
            <ul id="linksList" data-role="listview" data-
inset="true"></ul>
        </div>
        <div data-role="footer" data-position="fixed" data-
theme="b">
            <h4>Lector RSS by jlillo - &copy; 2013</h4>
        </div>
    </div>

    <!-- Pantalla donde se muestra el contenido de la noticia -
->
    <div data-role="page" id="contentPage">
        <div data-role="header" data-position="fixed" data-
theme="b">
            <a href="#mainPage" data-rel="volverRss">ATRAS</a>
```

6.- Implementación

```
        <h1></h1>
    </div>
    <div data-role="content" id="entryText">
    </div>
<div data-role="footer" data-position="fixed" data-theme="b">
    <h4>Lector RSS by jlillo - &copy; 2013</h4>
    </div>
</div>
```

6.2.- Páginas dinámicas

En la estructura de la página HTML **index.html** se ha podido comprobar como en muchas de las páginas no había contenido aparente. Sólo se muestra los bloques de estructura básica de la página:

```
<!-- Pantalla principal donde se muestra el listado de noticias
-->
    <div data-role="page" id="mainPage">
        <div data-role="header" data-position="fixed" data-
theme="b">
            <a href="#pageRss" data-rel="back"
title="Atras">ATRAS</a>
            <h1></h1>
        </div>
        <div data-role="content">
            <div id="status"></div>
            <ul id="linksList" data-role="listview" data-
inset="true"></ul>
        </div>
        <div data-role="footer" data-position="fixed" data-
theme="b">
            <h4>Lector RSS by jlillo - &copy; 2013</h4>
        </div>
    </div>
```

Esto es debido a que se inyectará el contenido dinámicamente de estas páginas dependiendo de las llamadas que se realicen a los diferentes servicios. Para realizar este proceso, se ha creado un fichero **.js (lectorRss.js)** donde se especifica que información se ha de cargar según la acción que realicemos.

6.- Implementación

6.3.- Implementación lector rss

Como se ha definido anteriormente, el fichero **lectorRss.js** es el encargado de cargar toda la información referente a los contenido del feed.

El funcionamiento es el siguiente, una vez que el evento *pageinit* ha finalizado, procedemos a recorrer el xml del feed en busca de los datos que nos interesa mostrar en la aplicación:

```
$("#h1", this).text(titulo);
$.get(RSS, {}, function(res, code) {
    var xml = $(res);
    var items = xml.find("item");
    $.each(items, function(i, v) {
        entry = {
            title:$(v).find("title").text(),
            link:$(v).find("link").text(),
            //description:$.trim($(v).find("description").te
            xt()),
            description:$.trim($(v).find("encoded").text()),
            category:$.trim($(v).find("category").text()),
            date:$(v).find("pubDate").text().substr(0,16),
            autor:$(v).find("creator").text()
        };
        entries.push(entry);
    });
});
```

Una vez guardada la información que necesitamos en la variable *entries*, y utilizando parte de esta (título, autor y fecha), se crea el listado de contenidos del feed y se carga en la sección correspondiente, *#linksList*:

```
var s = '';
$.each(entries, function(i, v) {
    s += '<li><a href="#contentPage" class="contentLink" data-
    entryid="'+i+'">' + v.title + '<br><i>' + v.autor + ' - ' +
    v.date + '</i></a></li>';
});
$("#linksList").append(s);
$("#linksList").listview("refresh");
```

6.- Implementación

Si se pulsa sobre alguna noticia del listado anterior, se ejecuta el evento *pageshow*, que nos mostrará el contenido de la noticia elegida:

```
$("#h1", this).text(entries[selectedEntry].title);
var contentHTML = "";
contentHTML += entries[selectedEntry].description;
contentHTML += '<br><br> Category: ' +
entries[selectedEntry].category;
contentHTML += '<br> Fecha publicacion: ' +
entries[selectedEntry].date;
contentHTML += '<br> <i>Autor: ' +
entries[selectedEntry].autor + '</i>';
contentHTML += '<br><br> <a
href="' + entries[selectedEntry].link + '">Leer via web</a>';
$("#entryText", this).html(contentHTML);
```

6.4.- Implementación Aplicación Android

Para implementar la aplicación en Android, se hará uso de la tecnología PhoneGap, que permite crear aplicaciones nativas con tecnología web y el entorno de programación Eclipse.

En primer lugar, se hará es la creación de un proyecto nuevo de tipo **Android Application Project**.

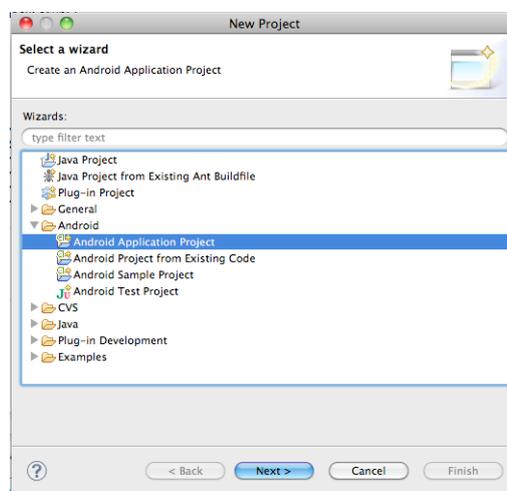


Figura 6.1: Creación de un proyecto Android

6.- Implementación

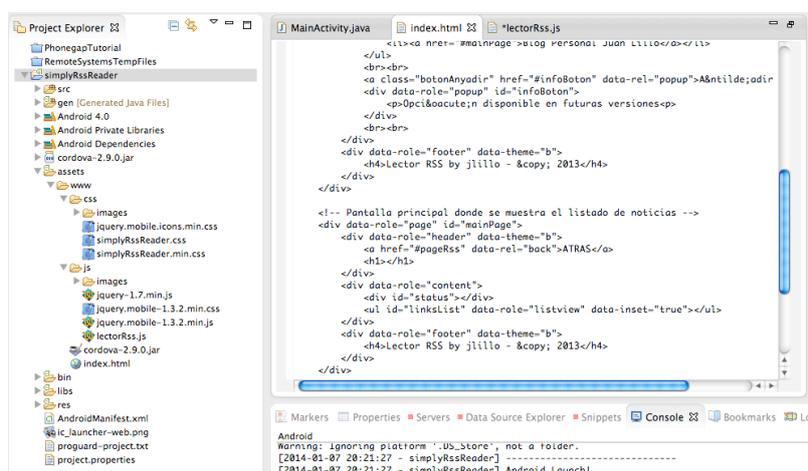


Figura 6.2: Directorios Proyecto “simplyRssReader” Android

En el directorio **assets**, se creará un nuevo directorio llamado **www**, donde incluiremos nuestros archivos para la aplicación. En este directorio, también incluiremos el archivo/librería JS del PhoneGap, llamada Cordova. Se añadirá al archivo **index.html** la instrucción necesaria para poder usar la librería nueva:

```
<script type="text/javascript" charset="utf-8" src="cordova-2.9.0.js"></script>
```

Lo siguiente es copiar los fichero **cordova-2.9.0.jar** al directoria **libs** y la carpeta **xml** al directorio **res**. Seguidamente añadimos PhoneGap a nuestro proyecto la librería jar de cordova al **Build Path**.

Al seguir estos pasos, ya tenemos el entorno preparado y ahora será necesario modificar la clase principal del proyecto. En primer lugar, cambiamos la clase base de **Activity** a **DroidGap**, después, reemplazamos la llamada **setContentView()** para que cargue el archivo **index.html** de la aplicación y por último, añadimos la sentencia **import org.apache.cordova.***.

6.- Implementación

```
import org.apache.cordova.DroidGap;

import android.os.Bundle;

public class MainActivity extends DroidGap {

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

Por último es necesario modificar los metadatos del proyecto. En el fichero **AndroidManifest.xml** se añaden los siguientes nodos:

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

<uses-permission android:name="android.permission.VIBRATE" />
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION"
/>
<uses-permission
android:name="android.permission.ACCESS_LOCATION_EXTRA_COMMANDS"
/>
<uses-permission
android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.RECEIVE_SMS"
/>
<uses-permission android:name="android.permission.RECORD_AUDIO"
/>
<uses-permission
android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
```

6.- Implementación

```
<uses-permission android:name="android.permission.READ_CONTACTS"
/>
<uses-permission
android:name="android.permission.WRITE_CONTACTS" />
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission
android:name="android.permission.ACCESS_NETWORK_STATE"
/>
<uses-permission android:name="android.permission.GET_ACCOUNTS"
/>
<uses-permission
android:name="android.permission.BROADCAST_STICKY" />
```

Y por último incorporamos un nodo nuevo en **<Activity>**:

```
<activity
    android:name="org.apache.cordova.DroidGap"
    android:label="@string/app_name"
    android:configChanges="orientation|keyboardHidden">
    <intent-filter></intent-filter>
</activity>
```

Una vez realizado todos los pasos anteriores, se prueba la aplicación con el emulador o un dispositivo compatible con las especificaciones elegidas.

6.- Implementación

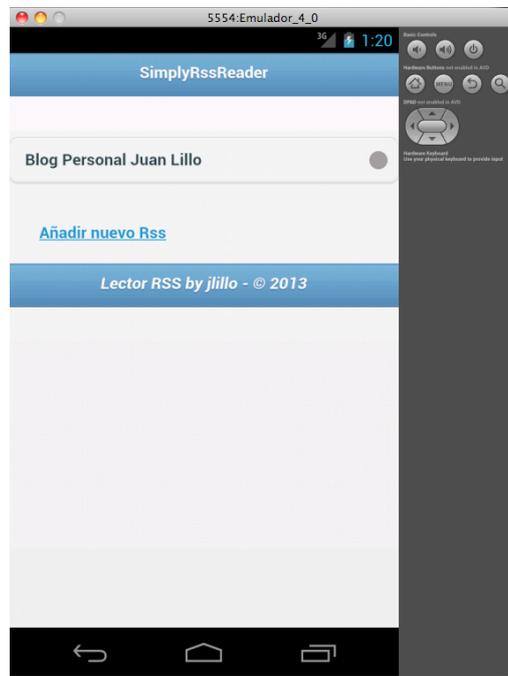


Figura 6.3: Emulador Android

6.5.- Implementación Accesibilidad

En un principio, se iba a utilizar el servicio de Google TTS, pero después de investigar mucho sobre el tema, hoy en día el tema de la accesibilidad en los dispositivos esta muy avanzada y no ha sido necesario usar un servicio externo para poder realizarlo. El propio S.O. Android lleva integrado un lector de pantallas, llamado talkbak, que funciona bien.

Sólo ha sido necesario utilizar la etiqueta title en determinadas partes del código para hacerlas accesibles:

```
<a href="#mainPage" data-rel="back" title="Atras">ATRAS</a>
```

7.- Manual de uso

Los requisitos mínimos necesarios para poder ejecutar la aplicación son los siguientes;

- Un dispositivo compatible con Android.
- Android 2.3.3, recomendado Android 4.0 Jelly Bean.

Para la instalación de la aplicación, lo que hay que hacer es, lo primero de todo, bajarse la apk de la siguiente dirección: <http://goo.gl/34q86m> y ejecutarla. Durante la instalación, aceptamos los permisos necesarios para su funcionamiento y ya estamos listo para usarla.

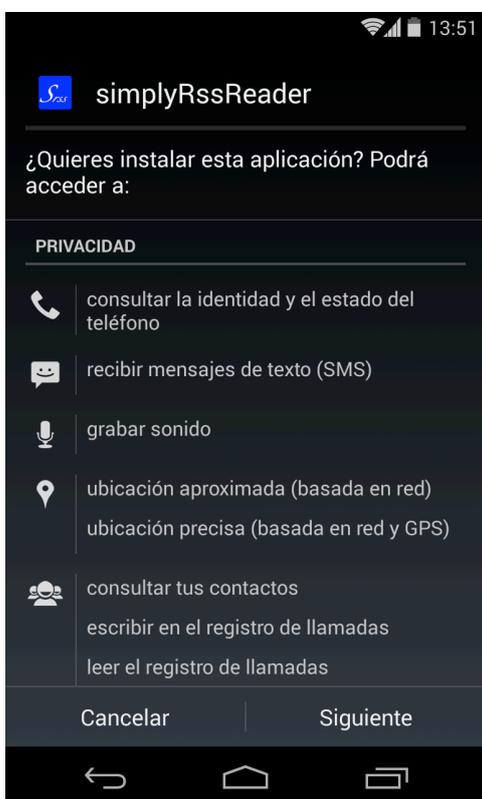


Figura 7.1: Pantalla Aceptación permisos

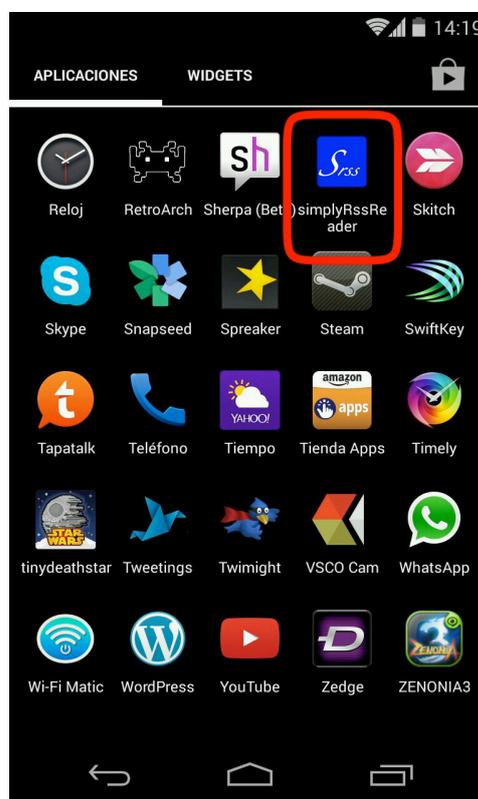


Figura 7.2: Pantalla Aceptación permisos

7.- Manual de uso

El uso de la aplicación es muy intuitivo y fácil de aprender. Al ejecutar la aplicación, lo primero que se muestra es la pantalla inicial del listado de feeds.

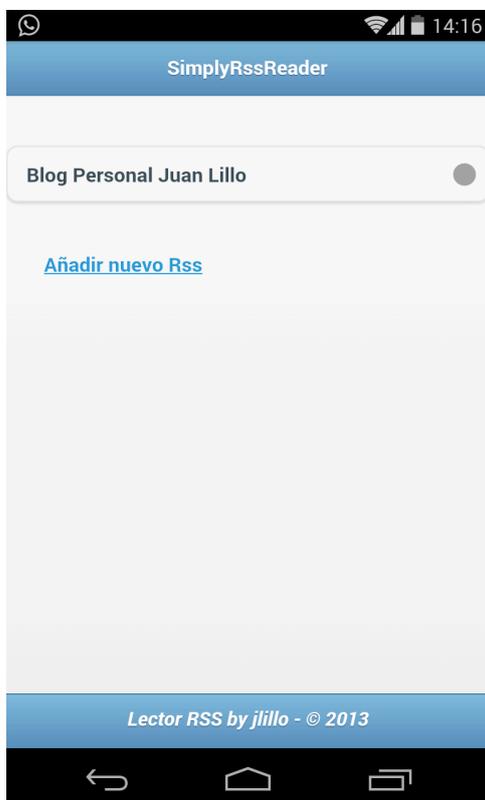


Figura 7.3: Pantalla Listado Feeds



Figura 7.4: Pantalla Listado Noticias

Al pulsar en cualquier opción del listado, nos muestra la pantalla principal o listado de noticias. En esta pantalla, como se observa, se ve las últimas noticias existentes en la web.

Sj pulsamos en una noticia, se abre la pantalla de contenido de la noticia, donde podemos ver, entre otros datos, el contenido de la noticia, el autor, la categoría a la que pertenece y un link para ver, si se quiere, la noticia directamente en la web.

7.- Manual de uso

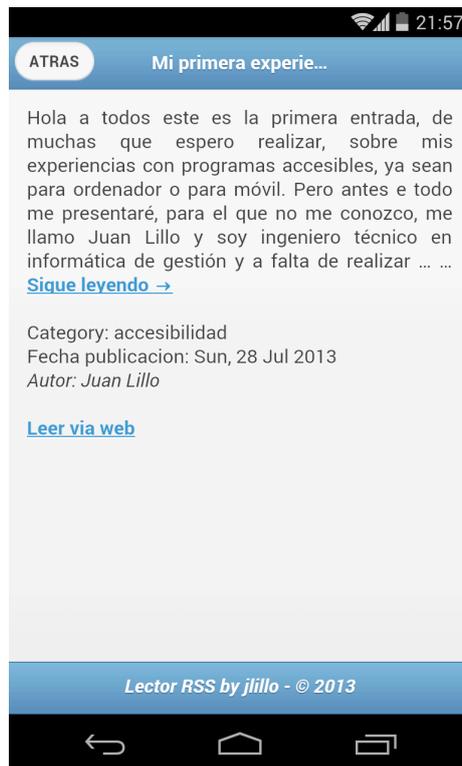


Figura 7.5: Pantalla Contenido

Como se puede observar en la pantalla, existe el botón de atrás, que nos permite volver a la pantalla del listado de noticias.

8.- Conclusiones

8.1.- Conclusiones

Después de la realización del proyecto, se ha llegado a diversas conclusiones en temas relacionados con el desarrollo de la aplicación para dispositivos móviles y con las diferentes tecnologías usadas en el PFC:

Partiendo de la base que era la primera vez que realizaba este tipo de aplicaciones para dispositivos móviles y teniendo en cuenta que el desarrollo se tenía que realizar con tecnología web (HTML5, CSS3 y JS), en un principio dudaba que con el calendario tan apretado que teníamos, obtendría un resultado tan satisfactorio.

La conclusión que he extraído es que es realmente viable la realización de aplicaciones con tecnologías web y que además es una solución ventajosa por diversos motivos. El primero lugar se consigue una aplicación multi-plataforma, y en segundo lugar, es una tecnología más sencilla de usar que la existente en los diversos SO.

Respecto a los objetivos generales, se pueden dar por conseguidos. Ahora disponemos de una aplicación llamada **simplyRss Reader**, una nueva aplicación para poder leer las noticias de las páginas que nos interesan.

En referente al calendario, es un punto en donde he tenido problemas. Teniendo en cuenta que el calendario era ajustado, he tenido problemas en cumplir con algunas de las fechas de entrega. La planificación del proyecto se ha ido siguiendo pero con dificultades, sobretodo por mi situación personal y por problemas técnicos que ha hecho que realice un esfuerzo extra al final del proyecto.

Sobre los objetivos específicos, se puede decir que han sido alcanzados, como por ejemplo hemos obtenido una solución técnica para la creación de aplicaciones.

8.- Conclusiones

También ha estado interesante poder aplicar conocimientos obtenidos en otras asignaturas de la ingeniería y poder aplicar los conocimientos obtenidos de la gestión de proyectos, métodos de desarrollo ágiles, etc.

En definitiva, he quedado bastante satisfecho con el resultado obtenido y gracias a este proyecto me he introducido en el mundo del desarrollo de aplicaciones para móviles y espero que esta no sea la primera y última aplicación que realizo.

8.- Conclusiones

8.2.- Líneas de futuro

Con la entrega del proyecto, no termina el ciclo de vida de esta y hemos de continuar mejorando la aplicación. A continuación se presentan posibles futuras actualizaciones:

- Añadir la opción de añadir nuevos feeds.
- Implementar la aplicación en otras plataformas (iOS, Windows phone, etc.).
- Ampliar la gama de dispositivos útiles (sobretudo tablets).
- Añadir integración con redes sociales.
- Notificaciones push.

9.- Entregables PFC

La entrega final de PFC consiste en un fichero zip, que contiene el siguiente contenido:

- Memoria del PFC: memoria_pfc_jlillo.pdf
- Presentación del PFC: presentacion_pfc_jlillo.ppt
- Video de la presentación del PFC: video_presentacion_pfc_jlillo
- Código fuente del PFC: codigo_PFC_jlillo

Annexo I – Glosario

API: Acrónimo en inglés de *Application Programming Interface*, es el conjunto de funciones y procedimientos que ofrecen ciertas bibliotecas para ser utilizadas en otros programas.

Android: Sistema operativo móvil creado por Google.

CSS: Acrónimo del inglés *Cascading Style Sheets*, es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML.

CSS3: versión tercer del CSS.

HTML: Siglas de *HyperText Markup Language*, que hace referencia al lenguaje de marcado que se usa para la creación de páginas webs.

HTML5: versión quinta del HTML.

IDE: Acrónimo del inglés *Integrated Development Environment*. Es un programa informático compuesto por un conjunto de herramientas de programación. Puede ser exclusivo de un lenguaje o de varios.

iOS: Sistema operativo móvil de Apple. Inicialmente fue creado para su dispositivo iPhone, pero ha acabado implantándose en todos los dispositivos móviles de Apple.

JavaScript: Lenguaje de programación interpretado, usado principalmente en las páginas web para permitir mejoras en las interacciones con las páginas web.

META: Etiqueta HTML que se incorpora en el encabezado de una página web y que resulta invisible para un visitante, pero de gran valor para los navegadores.

Annexo I – Glosario

SDK: Acrónimo del inglés *Software Development Kit*. Es un conjunto de herramientas desarrollo de programas que permite al programador crear aplicaciones para un sistema concreto.

MVC: Modelo – Vista – Controlador. Es un modelo o patrón de desarrollo de programas que separa los datos de una aplicación (modelo), la interfaz de usuario (vista) y la lógica de negocio (controlador) en tres componentes diferentes,

SO: Siglas de Sistema Operativo.

Anexo II – Referencias y Fuentes Consultadas

BENEITO, Roser – Presentación de documentos y elaboración de presentaciones (UOC) [En línea]

http://materials.cv.uoc.edu/cdocent/HKZB5UG6XS130_6R5O43.pdf?ajax=true

SÁENZ, Nita i Vidal Rut – Redacción de textos científico-técnicos (UOC) [En línea]

http://materials.cv.uoc.edu/cdocent/8QROP4G6IXT6ND3J1_XE.pdf?ajax=true

SOLIS, Carlos – Tutorial: mejorar el rendimiento del emulador Android [En línea]

<http://revolucion.mobi/2012/04/17/tutorial-mejorar-el-rendimiento-del-emulador-android/>

ROSET, Román – Test Developement Driven (TDD) i Behaviour Development Driven (BDD) [En línea]

http://biblioteca.uoc.edu/prestatgeries/11116/Tests_i_performace.pdf

ROSET, Román – Els enfocis per escollir un Framework [En línea]

[http://biblioteca.uoc.edu/prestatgeries/11116/Escollir_el_framework_JS_\(i_l_est_il\)_de_treball.pdf](http://biblioteca.uoc.edu/prestatgeries/11116/Escollir_el_framework_JS_(i_l_est_il)_de_treball.pdf)

Anexo II – Referencias y Fuentes Consultadas

ROSET, Román – Introducció a les aplicacions per a telefonia mòbil [En línea]

http://biblioteca.uoc.edu/prestatgeries/11116/Introduccio_a_les_aplicacions_per_a_telefonia_mobil.pdf

CLARISÓ, Robert – Orientacions generals per al Treball Final [en línea]

<http://www.slideshare.net/rclariso/orientacions-generals-per-al-treball-final-a-la-uoc>

BATALLER; Alfons, BENEITO, Roser; SÁENZ, Nita; VIDAL, Rut – Treball Final de carrera (UOC) [En línea]

http://materials.cv.uoc.edu/continguts/XW08_19018_00443/

PhoneGap [En línea]

<http://phonegap.com/>

PhoneGap – Getting Started with Android [En línea]

http://docs.phonegap.com/en/2.7.0rc1/guide_getting-started_android_index.md.html

Mike – PhoneGap para Android con Eclipse

<http://www.phonegapspain.com/tutorial/phonegap-para-android-con-eclipse/>

Evolus Penzin

<http://pencil.evolus.vn/>

Anexo II – Referencias y Fuentes Consultadas

jQuery Mobile (I) – Introducción [En línea]

<http://www.franciscojavierpulido.com/2012/07/jquery-mobile-i-introduccion.html>

¿Qué es el método Kankan para la gestión de proyectos? [En línea]

<http://www.javiargarzas.com/2011/11/kanban.html>

Guía breve de Accesibilidad web [En línea]

<http://www.w3c.es/Divulgacion/GuiasBreves/Accesibilidad>

Introducción a la Accesibilidad Web [En línea]

<http://www.w3c.es/Traducciones/es/WAI/intro/accessibility>

HTML Techniques for Web Content Accessibility Guidelines 1.0 [En línea]

<http://www.w3.org/TR/WCAG10-HTML-TECHS/>

Anexo III – Índice de Figuras

Figura 1.1: Ejemplo metodología Kanban	7
Figura 1.2: Diagrama de Gantt	10
Figura 2.1: Imagen aplicación Feedly	11
Figura 2.2: Imagen aplicación Pocket	12
Figura 2.3: Imagen aplicación Flipboard	13
Figura 3.1: Eclipse con proyecto SimplyRssReader	14
Figura 3.2: Evolus Pencil con proyecto SimplyRssReader	16
Figura 4.1: Icono HTML5	17
Figura 4.2: Icono CSS3	17
Figura 4.3: jQuery Mobile UI elements	18
Figura 4.4: Icono PhoneGap	19
Figura 5.1: Pantalla de inicio	25
Figura 5.2: Pantalla de añadir feed	26
Figura 5.3: Pantalla listado noticias	27
Figura 5.4: Pantalla contenido noticia	28
Figura 5.5: Diagrama flujo de pantallas	29
Figura 6.1: Creación de un Proyecto Android	37
Figura 6.2: Directorios Proyecto “simplyRssReader” Android	38
Figura 6.3: Emulador Android	41

Anexo III – Índice de Figuras

Figura 7.1: Pantalla aceptación permisos	42
Figura 7.2: Pantalla Aplicaciones	42
Figura 7.3: Pantalla Listado Feeds	43
Figura 7.4: Pantalla Listado Noticias	43
Figura 7.5: Pantalla Contenido	44