



**(2997: APLICACIÓ ANDROID PER ESTUDIANTS UNIVERSITARIS)**

Memòria del Projecte Fi de Carrera  
d'Enginyeria en Informàtica  
realitzat per  
IRENA LOJIC VUKOVIC  
i dirigit per  
RAMON GRAU SALA  
Bellaterra, 20 de JUNY de 2011



El sotasignat Ramon Grau Sala Professor/a de l'Escola Tècnica Superior d'Enginyeria de la UAB,

**CERTIFICA:**

Que el treball a què correspon aquesta memòria ha estat realitzat sota la seva direcció per en Irena Lojic Vukovic

I per tal que consti firma la present.

Signat:  .....

Bellaterra, 20 de juny de 2011

# Índice

---

<b>Capítulo 1- Introducción .....</b>	<b>7</b>
Visión general .....	8
Objetivos y motivaciones .....	8
Estado del Arte .....	9
<b>Capítulo 2 - Análisis de requisitos .....</b>	<b>10</b>
Casos de uso .....	13
<b>Capítulo 3 – Diseño .....</b>	<b>14</b>
<b>Capítulo 4 – Implementación .....</b>	<b>19</b>
Conceptos teóricos.....	20
Anatomía de una aplicación Android .....	20
Activity.....	20
Intent.....	20
Listeners .....	21
Services.....	21
Content Providers .....	21
Interfaz de usuario .....	22
Diseño.....	22
Jerarquía.....	22
Layouts .....	23
Widgets .....	26
Android Manifest .....	26
Aplicación PACK_UAB.....	27
Pantalla Bienvenida.....	28
Pantalla Menú .....	29
Pantalla Login .....	30
Pantalla Expediente.....	32
Pantalla Laboratorios .....	33
Identificación Transparente Expediente antiguo .....	34

Parseo expediente antiguo .....	36
Resultados Expediente antiguo.....	37
Cambio expediente .....	38
Identificación Transparente Expediente Nuevo.....	39
Parseo Expediente Nuevo .....	40
Resultados Expediente Nuevo.....	43
Identificación Transparente Laboratorios.....	44
Parseo Laboratorios .....	45
Resultados Laboratorios.....	47
<b>Capítulo 5 - Puesta en Marcha.....</b>	<b>49</b>
Dispositivo Real .....	50
Firma.....	50
Resultados .....	50
Android Market .....	51
<b>Capítulo 6 – Planificación y presupuesto.....</b>	<b>52</b>
Planificación temporal .....	53
Presupuesto .....	55
Coste de personal.....	55
Coste de software y licencias .....	56
Coste de Hardware.....	56
Coste de material .....	56
Resumen de costes.....	57
<b>Capítulo 7 - Propuesta de futuro .....</b>	<b>58</b>
Implementación .....	59
Resultados de la maqueta .....	60
<b>Capítulo 8 – Conclusiones .....</b>	<b>63</b>
<b>Capítulo 9 – Bibliografía.....</b>	<b>65</b>
Links.....	66
Libros .....	67
<b>Capítulo 10 - ANNEXOS .....</b>	<b>68</b>
ANNEXO I : Casos de Uso .....	69
Escenario I: Uso del resto del móvil .....	69

Escenario II: Aplicación PACK_UAB .....	69
Escenario III: Opción Expediente.....	69
Escenario IV: Opción Laboratorios .....	70
Escenario V: Login .....	70
Escenario VI: Consulta de Expediente.....	71
Escenario VII: Consulta/Inscripción de Laboratorios .....	71
Escenario VIII: Opción Pizarra .....	72
Escenario IX: LoginPizarra .....	72
Escenario X: Publicar Noticias .....	73
Escenario XI: Consultar noticias .....	73
ANNEXO II: Herramientas de desarrollo .....	74
Entorno de desarrollo .....	74
Android SDK .....	74
ANNEXO III: Juego de pruebas .....	75
ANNEXO IV: Tabla de figuras.....	78

*Este proyecto está dedicado a mis padres por estar siempre ahí*

*y a mi novio, por haber sido mi mayor apoyo y mi*

*compañero de viaje durante todos estos años.*

*Mil gracias.*

## Capítulo 1- Introducción

---

*En este capítulo introductorio se presenta la idea general del proyecto, el por qué se ha realizado y si en la actualidad existe algo parecido.*

## Visión general

Los dispositivos móviles hoy en día son mucho más que un terminal con servicios de telefonía, nos proporcionan multitud de servicios y funciones que a través de aplicaciones podemos explotar de diferentes maneras.

El uso de la tecnología Android en el proyecto fue una de las opciones debido a su carácter de libre distribución y al crecimiento que ha sufrido en un breve espacio de tiempo, convirtiéndose en un fuerte competidor en el mercado de los sistemas operativos móviles.

## Objetivos y motivaciones

Mi aplicación, llamada PACK\_UAB, consiste en mejorar la comodidad de los estudiantes de nuestra universidad, ofreciéndoles una aplicación gratuita para teléfonos Android, desde la cual puedan acceder a su expediente académico para consultar sus notas y a la página del neptu para apuntarse o consultar los grupos de prácticas.

Se quería conseguir algo novedoso, pero a la vez útil para los universitarios.

La idea surgió partiendo del “boom” del Internet en el móvil, causa de la popularización de las tarifas planas que se asocian a los smartphones, que gracias a su bajo precio hoy en día son accesibles a cualquier tipo de usuario.

Por mi experiencia y la de mis compañeros como estudiantes de la *ETSE*, a lo largo de todos estos años, nos hemos encontrado con una serie de problemas que en algunas ocasiones nos han complicado un poco la vida y es lo que pretendo solucionar.

¿Cuántas veces nos hemos ido de viaje o simplemente no hemos tenido un ordenador a mano, y ha salido alguna nota y durante horas hemos estado con los nervios de saber ese tan esperado resultado?

¿Cuántas veces nos hemos tenido que apuntar a un grupo de prácticas y hemos tenido que cambiar nuestros planes para estar cerca de un ordenador justamente a la hora que abren los grupos, para poder apuntarnos al grupo que nos va bien y que resulta que es precisamente el que todo el mundo quiere?



Pues ahora, gracias a mi aplicación, este problema dejara de existir, ya que desde nuestro móvil Android podremos consultar las notas desde cualquier sitio y sobre todo, podremos apuntarnos a las prácticas también desde cualquier sitio, sin preocupaciones, únicamente apretando un botón e identificándonos.

Además de esto, para facilitar todavía más las cosas, la idea es que tanto la página del expediente, como la de los laboratorios estén adaptadas para el teléfono, parseadas de tal forma que no tengamos que irnos desplazando por la pantalla para conseguir ver algo y para que solamente nos aparezca en la página lo que realmente nos interesa.

La identificación será a través de la interfaz del dispositivo, nos saltaremos la identificación de la página y las miles de paginitas a las que hay que acceder para conseguir llegar a lo que nos interesa.

En resumen, gracias a esta aplicación podremos ahorrar tiempo y conseguiremos no depender del espacio.

## Estado del Arte

Cuando tuve la idea de hacer el *PACK\_UAB* no existía ninguna aplicación igual, que te permitiera consultar el expediente académico de la universidad a través del móvil. Cuando estaba ya en la fase de implementación me enteré de que una empresa estaba creando una aplicación para Android con la misma idea, pero para la universidad de Madrid. Hace cosa de un mes, encontré en *youtube* el video publicitario de la aplicación, porque ya la habían terminado y lanzado al mercado. La idea es la misma, pero no siguen la misma metodología ni filosofía que yo para llegar a la solución. Es curioso que alguien tuviera mi misma idea, por una parte es algo que me reconforta, porque si una empresa lo ha hecho es porque la idea tampoco es tan mala.

## Capítulo 2 - Análisis de requisitos

---

*En esta fase se describen los requisitos de los usuarios para esta aplicación, es decir, que funcionalidades tiene que ofrecer y que objetivos debe alcanzar. Se detallan tanto los requisitos funcionales como los no funcionales, es decir, los que tienen que ver con el funcionamiento de la aplicación y los que tiene que ver con las propiedades del sistema, más a nivel de hardware. También se presenta el diagrama de casos de uso de la aplicación, detallando cada una de las opciones de las que se dispone.*

Los estudiantes de nuestra universidad tienen la necesidad de disponer de una aplicación mediante la cual puedan consultar su expediente académico y los grupos de prácticas (con la opción de apuntarse a ellos) a través de su teléfono móvil. La aplicación dispondrá de un menú para poder elegir la opción que se desea. Seguidamente el estudiante se tendrá que identificar como tal, porque no dejan ser páginas propias de nuestra universidad, por lo tanto solo podrán acceder a ella los estudiantes que se logeen correctamente con su niu y password.

La idea es que la aplicación haga de intermediario entre el usuario y la página, de tal forma que a partir del momento en que le das tu usuario y contraseña, haga todo el trabajo de manera transparente y que el usuario solamente vea el resultado final, que es lo que le interesa. Se entiende como resultado final la página (expediente o laboratorios), parseada, de tal manera que solo salga lo necesario.

Los usuarios serán los actores de la aplicación. Tienen que ser estudiantes de nuestra universidad para poder acceder a las diferentes opciones (tienen que tener NIU y PASSWD). Además de esto, para acceder al neptu (laboratorios) tienen que ser alumnos de la ETSE, ya que únicamente estos utilizan esta plataforma de gestión de los grupos de prácticas.

El escenario será el siguiente:



Fig.1.

A continuació se ilustra una tabla resumen con los requisitos funcionales y no funcionales de la aplicación:

Requisitos funcionales	
1-	El sistema debe permitir acceso ÚNICAMENTE a los usuarios identificados correctamente.
2-	La identificación debe de ser transparente al usuario, es decir, este introducirá el Niu y Password a través de la interfaz del teléfono y no lo hará a través del login de la página.
3-	El sistema debe permitir consultar el expediente académico a los alumnos identificados correctamente.
4-	El sistema debe permitir consultar los laboratorios de los alumnos identificados correctamente y en caso de que el grupo de prácticas este abierto, apuntarse a un grupo o bien salir de un grupo.
5-	El sistema debe recoger la página del expediente devuelta y únicamente debe de mostrar las notas del año actual y los datos personales del alumno. El menú con las otras opciones NO se debe mostrar. Además, la tabla principal de las notas, no tiene que mostrar el GRUPO, el CURSO, etc...
6-	El sistema debe recoger la pagina de los laboratorios y parsearla de tal forma que directamente salga el menú con todas las asignaturas para que el alumno pueda elegir. El menú derecho de las diferentes opciones no debe de aparecer. Además, el contenido debe de estar adaptado a la pantalla.

Fig.2.

Requisitos no funcionales	
1-	La aplicación tiene que funcionar en todos los teléfonos Android, siempre y cuando tengan acceso a Internet.

Fig.3.

## Casos de uso

En la siguiente figura se muestra el diagrama de casos de uso de la aplicación:

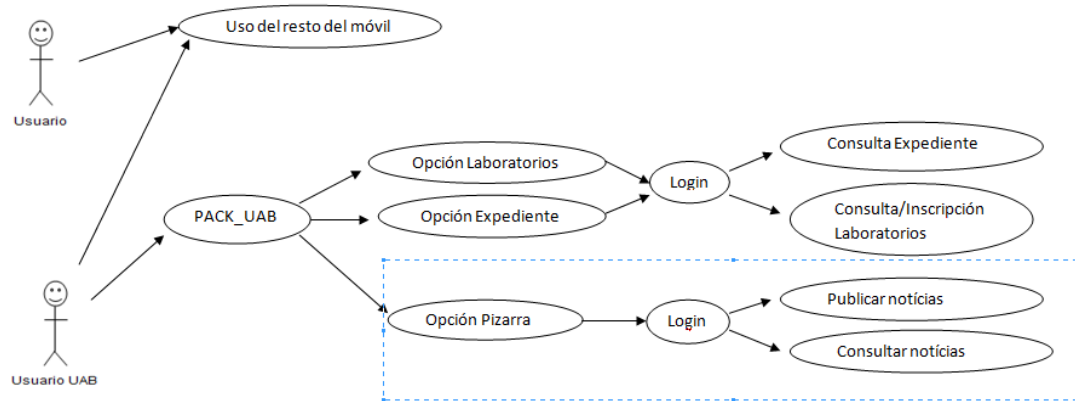


Fig.4.

Nota: La parte Opción Pizarra de explicara en el apartado “Propuesta de Futuro”

## Capítulo 3 – Diseño

---

*En esta parte se diseña la arquitectura y estructura de la aplicación para facilitar su posterior implementación. Esto se hace mediante patrones de diseño que en este caso nos ayudaran a ver como quedara estéticamente cada una de las pantallas de mi aplicación.*

Una vez se entienden razonablemente los requisitos de la aplicación, ya podemos pasar a la fase de diseño. En esta fase nos dedicaremos a crear una arquitectura para su posterior implementación.

Como para cualquier otro tipo de aplicación, en esta fase utilizaremos patrones de diseño, pero en este caso serán patrones especiales para Android. Android Patterns recopila una serie de patrones de diseño que son recomendables de cara a la construcción de una aplicación consistente y usable. Estos patrones están categorizados en función del contexto y nos ayudan a responder una serie de preguntas que nos podrían surgir en la fase de diseño.

Nuestra aplicación tendrá la siguiente estructura, para poder moverse por las diferentes pantallas:



Fig.5.

- 1- Cada pantalla tendrá un botón Next y un botón Previous, para poder desplazarse a la siguiente pantalla o a la anterior, exceptuando la primera pantalla que solamente tendrá el botón next.
- 2- Cuando el usuario pulsa el botón Next...
- 3- ..La siguiente pantalla es cargada...
- 4- ..y el botón Atrás se activa para poder ser pulsado.

En nuestro caso, en la primera pantalla nos aparecerá una pequeña explicación de nuestra aplicación y tendremos activo el botón next, que una vez pulsado nos llevará a la pantalla del menú, para poder escoger una opción (Expediente/Laboratorios).

El menú será de este estilo:

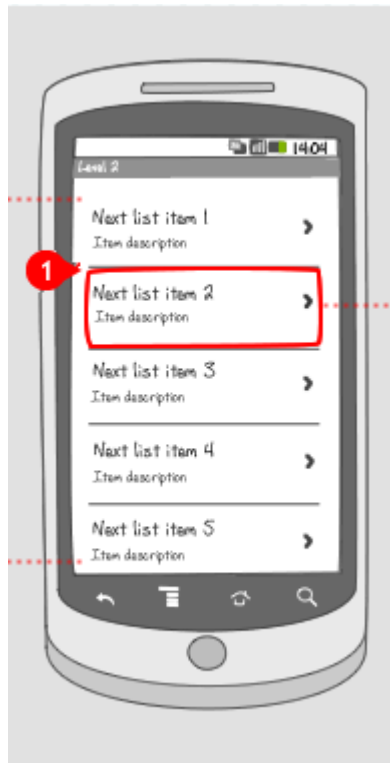


Fig.6.

- 1- Tendremos las diferentes opciones y cuando se pulse sobre alguna, nos aparecerá la pantalla de login..



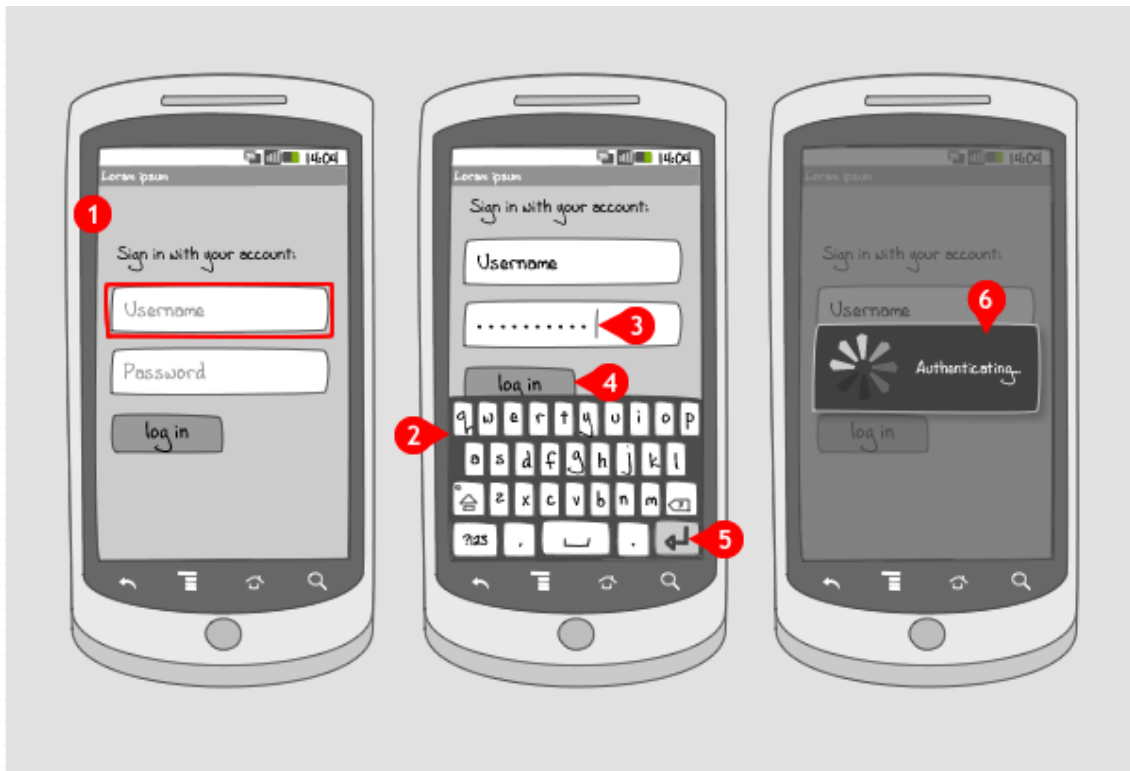


Fig.7.

- 1- La pantalla de login mostrara dos celdas de texto, una para el usuario y la otra para la contraseña y un botón dedicado para hacer el logeo. Cuando el primer campo se ha rellenado...
- 2- Un teclado saldrá en la pantalla para que el usuario pueda introducir su contraseña.
- 3- La contraseña estará oculta, porque es un dato privado.
- 4- El botón de login es visible en parte, los usuarios podrán escoger si clicar sobre el botón o bien..
- 5- Si clicar al Intro del teclado para que este desaparezca y nos quede visible totalmente el botón de login.
- 6- Cuando se clique en el botón de login, aparecerá una barra de carga, para notificar al usuario de que su usuario y password se están autenticando. Si la combinación es correcta...



Fig.8.

- 1- Nos aparecerá una página en blanco con una barra de progreso, mostrándole al usuario que el contenido de su expediente o de los laboratorios de está cargando.
- 2- Cuando acabe la carga el contenido será mostrado.

Este es el diseño de mi aplicación y a continuación, en la fase de Implementación se detallara a nivel de programación como se ha hecho cada cosa.

## Capítulo 4 – Implementación

---

*Esta fase describe paso a paso como se ha estructurado el proyecto y por cada bloque se detalla su implementación. Además también se muestran los resultados de cada una de las partes.*

Una vez tenemos los requisitos y el diseño de la aplicación a punto, podemos pasar a su desarrollo. Crearemos un nuevo proyecto en el Eclipse y comenzaremos a crear una clase por cada pantalla del diseño.

## Conceptos teóricos

### Anatomía de una aplicación Android

Una aplicación Android, está compuesta por cinco componentes principales: *Activity*, *Listeners*, *Intents*, *Servicios* y *Content Provider*. No todas las aplicaciones deben utilizar los cinco, pero si, una combinación de ellos.

#### *Activity*

Las actividades son el componente más genérico y más común de la plataforma. Una *Activity* es generalmente una pantalla de la aplicación con su propia interfaz de usuario. Para hacer uso de ella necesitamos que nuestra clase herede de la superclase *Activity*. Cada una de nuestras clases (actividades) podrá mostrar una interfaz de usuario compuesta por Vistas (*Views*) y podrá reaccionar a los eventos del usuario (clicar un botón, escribir un campo de texto, etc). Cada vez que se cambie de Vista, se cambiará de Actividad. La navegación entre pantallas se hará mediante el arranque de una nueva *Activity* (generalmente mediante el método *StartActivity(myintent)*) y mediante una clase especial llamada *Intent*.

Por ejemplo, una aplicación de correo electrónico puede tener una actividad que muestra una lista de correos electrónicos nuevos, otra actividad para componer un correo electrónico, así como otra actividad para la lectura de mensajes de correo electrónico.

Cuando una pantalla es abierta, la previa es puesta en pausa y agregada al *History Stack*. Posteriormente, el usuario puede navegar hacia previas pantallas invocando las pantallas almacenadas en el *History Stack*. Las pantallas también pueden ser borradas de éste cuando resulta inapropiado su almacenamiento. Android mantiene un "History Stack" por cada una de las aplicaciones que son activadas desde la *Home Screen*.

#### *Intent*

Android usa una clase especial llamada *Intent* para moverse de una pantalla a otra. Un *Intent* describe lo que una aplicación desea hacer. Las dos partes más importantes de la estructura de datos de un *Intent* son la acción y los datos sobre los cuales se actuará.

### ***Listeners***

Los *Listeners* se utilizan para reaccionar a eventos externos (por ejemplo, una llamada). No tienen interfaz de usuario, pero pueden utilizar el servicio *NotificationManager* para avisar al usuario. Para lanzar un aviso no hace falta que la aplicación se esté ejecutando, en caso necesario, *Android* la iniciará si se activa el *Listeners* por algún evento.

### ***Services***

Un servicio es código que se ejecuta en segundo plano (in background) y no necesita de una interfaz de usuario en su ciclo de vida.

En caso de que haya múltiples servicios a la vez, se les puede indicar diferentes prioridades según las necesidades.

Por ejemplo, un gestor de descarga en el cual se indican los contenidos a descargar y posteriormente el usuario puede acceder a una nueva Vista sin que el gestor se interrumpa. Un servicio también puede reproducir música en segundo plano mientras el usuario está en otra aplicación.

### ***Content Providers***

Este componente es usado para compartir datos entre diferentes aplicaciones. Es una clase que implementa un conjunto estándar de métodos que permite a otras aplicaciones guardar y obtener la información que maneja dicho *Content Provider*.

Se pueden utilizar también para leer o escribir datos privados de una aplicación.

Por ejemplo, el Note Pad es una aplicación que utiliza un *Content Provider* para guardar las notas.

## Interfaz de usuario

### *Diseño*

La interfaz de usuario se puede hacer de dos formas diferentes:

Procedual: como tradicionalmente se han implementado las interfaces en Java.

Declarativa: usando el lenguaje de marcas XML.

Diseñar una pantalla usando Java puede resultar complejo y poco eficiente, por eso la manera más común es la declarativa, con archivos de formato XML del directorio *res/layout*. Cada pantalla tendrá un código XML diferente. XML ofrece una estructura legible para el diseño, al igual que HTML. Cada fichero describe un *layout* (una pantalla) y cada *layout* a su vez puede contener otros elementos.

### *Jerarquía*

En la plataforma Android, se define la interfaz de usuario de una actividad, usando una jerarquía de nodos *View* y *ViewGroup*, como se muestra en el siguiente diagrama.

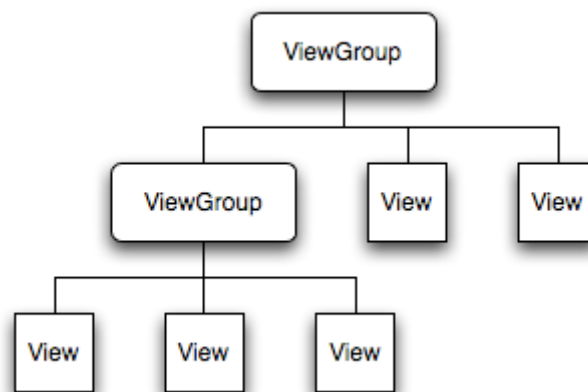


Fig.9.

Este árbol de jerarquía puede ser tan simple o tan complejo como sea necesario, y se puede construir utilizando el conjunto de *widgets* de Android y diseños predefinidos, o con vistas personalizadas.

Los *Views* y *Viewgroups* deben estar contenidos en los *layouts*, los cuales contienen otros elementos presentes en una vista. Dentro de cada *layout* podemos poner todos los elementos necesarios, incluidos otros *layouts*. Así conseguiremos estructurar la pantalla de la manera deseada y definir el ordenamiento que tendrán los componentes que queremos pintar.

## Layouts

Existen una gran variedad de *layouts*, en función de su posicionamiento en la pantalla y se describen a continuación algunos de ellos:

**LinearLayout**: Se les conoce como contenedores y sirven para reorganizar los elementos de nuestra aplicación. Sus hijos son los *Views*, *Viewgroup* u otros *layouts*. Nos permitirán alinear sus hijos en una única dirección, ya sea horizontal o vertical, agregándolos uno por uno en el orden que se van definiendo. La orientación predeterminada es horizontal.

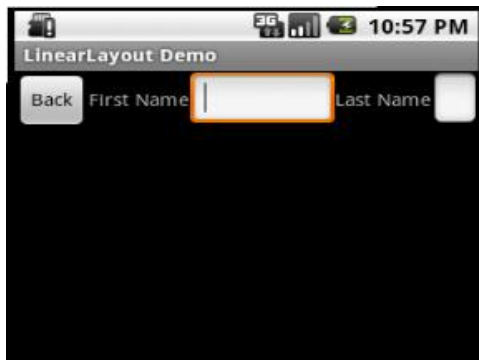


Fig.10.

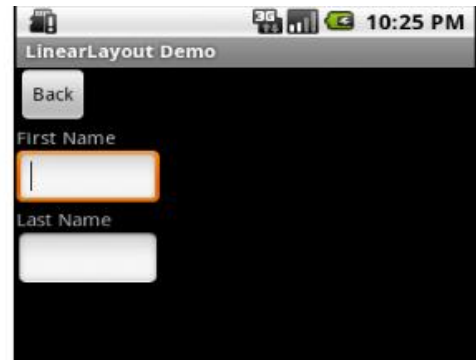


Fig.11.

**RelativeLayout**: Este layout permite agregar elementos respecto a uno previo. Esto significa que el posicionamiento se administra según la posición del elemento anterior.

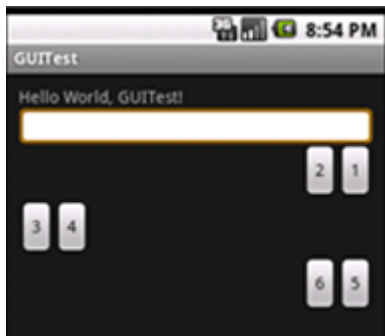


Fig.12.

**Absolute layout:** Coloca los elementos en posiciones absolutas en la pantalla, teniendo en cuenta que la posición (0,0) es el extremo superior izquierdo de la pantalla. A continuación se puede ver un ejemplo. El *Absolute layout* especifica la ubicación exacta de sus hijos.

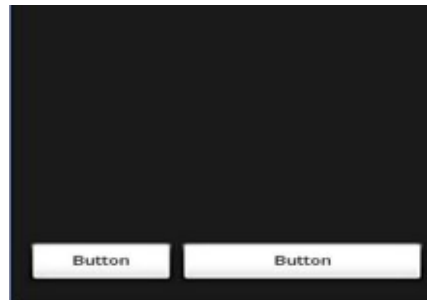


Fig.13.

**TableLayout:** permite colocar los elementos en forma de tabla, en filas y columnas.

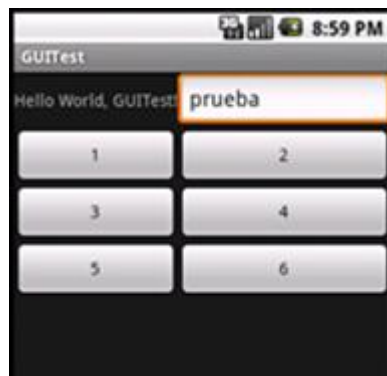


Fig.14.



**FrameLayout:** Es el tipo más simple de layout que permite agregar un solo elemento en un espacio en blanco definido. Se pueden agregar múltiples puntos de vista a un *FrameLayout* pero cada uno se acumulará en la parte superior izquierda de la anterior.

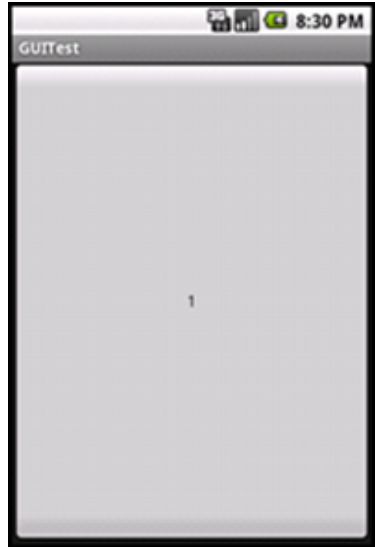


Fig.15.

**ScrollView:** Es un tipo especial de *FrameLayout* ya que permite a los usuarios desplazarse por una serie de puntos de vista que ocupan más espacio que el despliegue físico. El *scrollView* solo puede contener un *ViewGroup* y suele ser *LinearLayout*.



Fig.16.

### ***Widgets***

Un *widget* es un objeto View que sirve como interfaz para la interacción con el usuario. Android proporciona un conjunto de *widgets* totalmente implementado, como botones, casillas de verificación, y campos de entrada de texto, para poder construir la interfaz de usuario. Algunos *widgets* proporcionados por Android son más complejos, como un selector de fechas, un reloj, y controles de zoom. Pero, además, también se puede hacer algo más personalizado mediante la definición de su propio objeto View o mediante la ampliación y la combinación de *widgets* existentes.

Una vez se han añadido algunos Views/widgets a la interfaz de usuario, para que el usuario pueda interactuar con ellos, se deben realizar acciones. Para ellos se debe definir un detector de eventos, que define de qué manera se van a escuchar los eventos. La clase View tiene una colección de interfaces para ello.

### **Android Manifest**

En Android, existe un archivo XML llamado `AndroidManifest.xml`, que aunque no forme parte del código principal de la aplicación, es necesario para su correcto funcionamiento. Antes de que una aplicación pueda ser ejecutada por Android, el sistema tiene que conocer que componentes utiliza, mirando el *AndroidManifest.xml*. Este archivo es el fichero de control que le dice al sistema que tiene que hacer con todos los componentes que intervienen en la aplicación.

También le ofrece otra información como por ejemplo los permisos de usuario que la aplicación requiere, como Internet o el mínimo nivel de API que requiere la aplicación, etc.

### Aplicación PACK\_UAB

Mi aplicación está compuesta por varias clases, que vienen a ser todas las actividades que la forman. Cada clase o actividad tiene asociada una interfaz grafica, que es un archivo xml. A parte de esto, también está el archivo Manifest.xml que es donde se reflejan todos los permisos de la aplicación y todos los componentes y el archivo strings.xml que sirve para los textos, ya que en las interfaces ponemos identificadores y en este archivo le asociamos el texto.

En las carpetas drawable se pondrán las imágenes que queramos que se muestren en la interfaz. Como podemos observar hay tres carpetas, es para poner la imagen en tres dimensiones diferentes para que así la imagen se ajuste a todos los móviles y en todas las posiciones de la pantalla (horizontal/vertical). Una vez tenemos las imágenes guardadas se tendrán que vincular con el archivo .xml dónde se quiere que aparezcan utilizando `<ImageView>`.

En la siguiente figura se muestra todo lo que compone el proyecto:

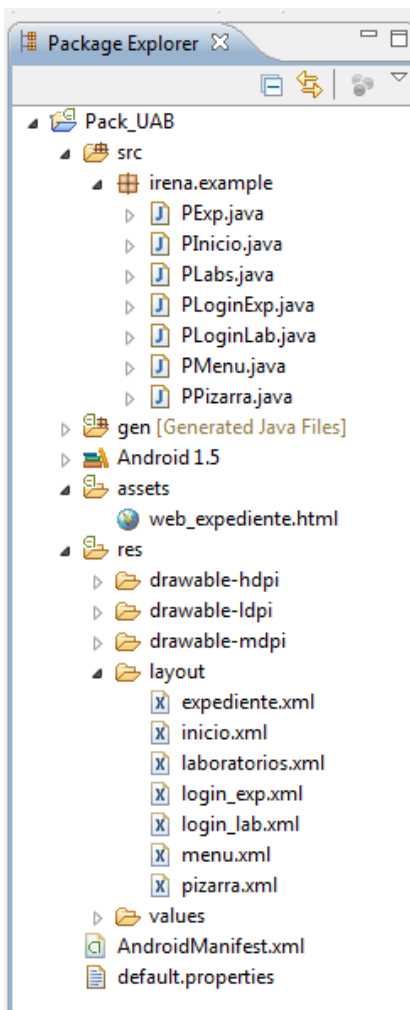


Fig.17.

### ***Pantalla Bienvenida***

Se trata de una activity muy sencilla, que nos muestra una pequeña explicación de lo que es la aplicación. La interfaz está compuesta por los widgets *TextView* y un *Button*. El *TextView* permite mostrar un campo de texto y *Button* nos dibuja sobre la pantalla un botón al que habrá que asociarle una determinada acción. Por otra parte, todo esto está dentro de un *LinearLayout*, que es como una caja en la que se ponen los otros componentes y se puede decidir cómo se mostraran, horizontalmente o verticalmente, entre muchas otras cosas.

En la activity (archivo .java), lo primero que hacemos es iniciarla con el método `onCreate()` i dentro le asociamos el archivo xml de la interfaz.

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

Después declaramos un objeto tipo *Button*, con la finalidad de asociarle una acción. Utilizamos un Listener que escuche eventos y en el momento que el botón ha sido pulsado, lo que hace es saltar a la siguiente activity, que es la del menú de opciones.

```
next.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View view) {
        Intent intent =new Intent();
        intent.setClass (PInicio.this, PMenu.class);
        startActivity(intent);
        finish();
    }

});
```

**Nota:** le corresponden los archivos PInicio.java y inicio.xml



Fig.18.

### ***Pantalla Menú***

En esta actividad es en la que se muestra el menú con las opciones que tiene nuestra aplicación. Finalmente hay 3 opciones, que son el Expediente, los Laboratorios y una Pizarra Virtual que se explica en el apartado “Propuesta de futuro”.

En la interfaz se define el *LinearLayout* para decir cómo se colocaran los widgets por la pantalla y dentro se declaran 3 botones, uno para cada opción. Finalmente, se define otro *LinearLayout*, en este caso con orientación horizontal, con un botón dentro, para tener la opción ir hacia atrás y volver a la pantalla de bienvenida.

En el archivo .java, lo que hacemos es iniciar la actividad, asociarle el archivo xml de la interfaz y declarar los cuatro objetos tipo *Button* (Expediente, Laboratorios, Pizarra y Atrás). Por cada uno de ellos definimos un Listener a la escucha que suceda algún evento, en todos los casos el *onClick()*, que se activa al pulsar el botón. En caso de pulsar el Expediente, nos iremos a la activity *PLoginExp.java*, en caso de pulsar los Laboratorios nos iremos a *PLoginLab.java*, en caso de pulsar la Pizarra nos iremos a *PPizarra.java* y finalmente si pulsamos Atrás, nos iremos a la activity anterior que es la pantalla de Bienvenida, *Plnicio.java*.

**Nota:** le corresponden los archivos *PMenu.java* y *menu.xml*



Fig.19.

### ***Pantalla Login***

Esta actividad nos sirve para la identificación. La interfaz consta de dos cajas de texto, *AutoCompleteTextView*, una para introducir el niu y otra para introducir la contraseña. La última tiene activada la propiedad de password, para que ésta quede oculta. También tenemos un *Button*, para que el usuario una vez ha introducido sus datos pueda darle a Aceptar y se inicie el proceso de validación y otro para volver Atrás. Los *TextView* los utilizamos para los títulos. Todo esto está incluido en los *LinearLayouts* pertinentes, para que quede todo bien organizado por la pantalla .

En esta *Activity* teníamos una problemática y era que estos datos (niu/contraseña), al pulsar el botón, de alguna manera se tenían que pasar a los campos niu/contraseña de la página del Expediente o Laboratorios, y el botón aceptar de dichas paginas se tenía que pulsar “solo”, para que directamente nos cargara la pagina pertinente.

Esto se ha solucionado utilizando un objeto *Bundle*, el cual nos permite pasar datos a través de las diferentes actividades, es decir, compartir datos. El *Bundle* se define de la siguiente manera:

```
//Creamos un bundle para poder pasar las variables a la actividad 4
Bundle bundle = new Bundle();

bundle.putString("niu_n",niu.getText().toString());
bundle.putString("passwd_n",passwd.getText().toString());

//Añadimos nuestro bundle
intent.putExtras(bundle);
|
```

Podemos observar que primero se crea un *Bundle* vacío, seguidamente se le asocian los datos que queremos compartir y finalmente se lanza.

La *Activity* que quiere recibir los datos lo hace de la siguiente forma:

```
// Capturamos el intent que hemos lanzado en la actividad principal
// con el bundle que lleva asociado
Bundle bundle = this.getIntent().getExtras();

if ( bundle != null ) {

    // Recogemos los valores de los parametros pasados en la
    // actividad principal

    niu = bundle.getString("niu_n");
    passwd = bundle.getString("passwd_n");

}
```

En las variables niu y passwd ya tenemos los valores compartidos.

Además de esto, en el archivo .java, se inicia la actividad con el método onCreate(), se vincula el archivo xml de la interfaz y por cada botón se crea un objeto Button y se hace un Listener a la espera de que suceda algún evento, que será el onClick() conforme se ha clicado el botón. Una vez pulsado el botón Aceptar, es donde definiremos el Bundle y nos iremos a la actividad PExp.java, en caso del expediente, y a la PLabs.java en caso de los Laboratorios. Al pulsar el botón Atrás volveremos a la Actividad anterior, que es la del menú, PMenu.java.

**Nota:** le corresponden los archivos PLoginExp.java y login\_exp.xml o PLoginLab.java y login\_lab.xml



Fig.20.



Fig.21.

### ***Pantalla Expediente***

Esta actividad es la que nos mostrará la página del expediente en caso de que el alumno se haya identificado correctamente.

Para ello, se utiliza un *WebView*, que es una clase que permite mostrar páginas web como parte de nuestra actividad. No incluye todas las características de un navegador, como los controles de navegación o una barra de direcciones, lo único que hace es limitarse a mostrar una página web. Además, interpreta el javascript de las páginas, la cual cosa nos vendrá muy bien a la hora de parsear la página y permite crear tus propias páginas html.

Si la web a la que nos vamos a conectar utiliza JavaScript, para que todo funcione correctamente, tendremos que habitarlo. Esto se hace en el archivo .java de la siguiente manera:

```
mWebView.getSettings().setJavaScriptEnabled(true);
```

Para que todo esto funcione, la aplicación deberá de tener acceso a Internet. Para ello, deberemos solicitar el permiso INTERNET en el archivo AndroidManifest.xml.

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

En la interfaz se define el *WebView* dentro de un *LinearLayout* y se indica que ocupe toda la pantalla. Además también se define dentro un *Button* que será el que permitirá volver al menú principal.

Por otro lado, en el archivo .java, se define el objeto *WebView*, se inicia la actividad con el *OnCreate()* y dentro se define el objeto *Bundle*, explicado anteriormente, para poder recoger los datos pasados por la actividad Login. Esto es necesario para poder hacer la identificación Transparente al usuario (Esto se explica detalladamente en el apartado *Identificación Transparente*). También se asocia el archivo xml correspondiente a la interfaz y se define el *Listener* que esperará a que se pulse el botón Menu, que nos permitirá volver al menú principal. Finalmente, utilizando el método *loadUrl* del *WebView* se carga la página de Login del expediente en el *WebView*, a la que le pasaran los datos del *Bundle*.

**Nota:** le corresponden los archivos PExp.java y expediente.xml



### ***Pantalla Laboratorios***

Esta actividad es la que nos mostrará la página del neptu en caso de que el alumno se haya identificado correctamente. A través de esta página podremos apuntarnos a los grupos de prácticas en caso de que estén abiertos o en su defecto consultarlos.

Se utiliza un *WebView*, igual que en el caso del expediente académico.

Tendremos que habilitar el Javascript en el archivo .java igual que en el caso del expediente.

```
mWebView.getSettings().setJavaScriptEnabled(true);
```

También deberemos solicitar el permiso INTERNET en el archivo AndroidManifest.xml.

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

En la interfaz se define el *WebView* dentro de un *LinearLayout* y se indica que ocupe toda la pantalla. Además también se define dentro un *Button* que será el que permitirá volver al menú principal.

Por otro lado, en el archivo .java, se define el objeto *WebView*, se inicia la actividad con el *OnCreate()* y dentro se define el objeto *Bundle*, explicado anteriormente, para poder recoger los datos pasados por la actividad Login pensada para los laboratorios. Esto es necesario para poder hacer la identificación Transparente al usuario. También se asocia el archivo xml correspondiente a la interfaz y se define el *Listener* que esperará a que se pulse el botón Menu, que nos permitirá volver al menú principal. Finalmente, utilizando el método *loadUrl* del *WebView* se carga la página de Login del neptu en el *WebView*, a la que le pasaran los datos del *Bundle*.

**Nota:** le corresponden los archivos PLabs.java y laboratorios.xml

### Identificación Transparente Expediente antiguo

Lo que se quiere conseguir es saltarse la identificación de la página web, de tal forma que cuando el usuario introduzca su niu y contraseña en la actividad del Login, mostrada anteriormente, de manera automática se pongan los datos en las casillas de identificación de la página y se pulse el botón. Todo esto sin que el usuario lo vea. El usuario lo único que tiene que ver al pulsar el botón aceptar, de la actividad de Login, es como carga la página final.

Además de esto, para llegar a ver el expediente por la vía normal, se tienen que seguir una serie de pasos que también se quieren evitar. Estos pasos son los que se muestran a continuación:

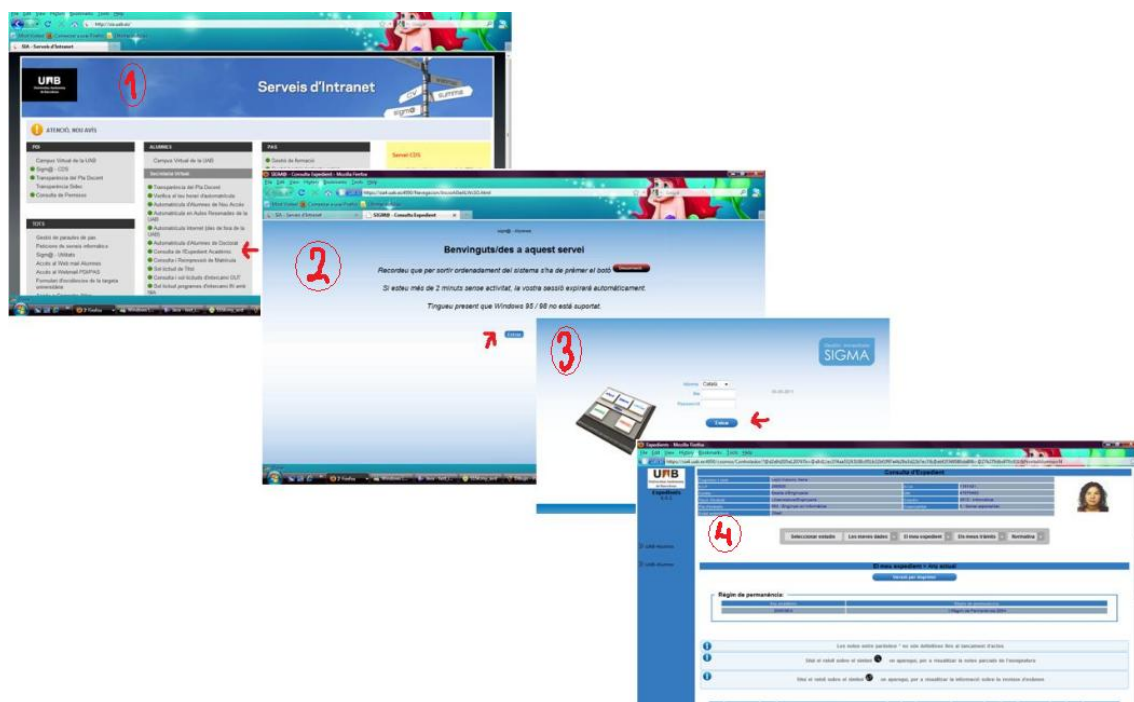


Fig.22.

Por lo tanto, al pulsar el botón Aceptar de la actividad de Login, queremos que directamente nos aparezca el paso 4 en caso de que el alumno se haya identificado correctamente. Para ello también nos aprovecharemos de las propiedades del Javascript.

Primeramente, utilizando el método `loadUrl` de `WebView` explicado anteriormente, nos saltaremos el paso 1, redireccionando directamente a la página del paso 2.

Lo que se podría haber hecho es de la misma manera redireccionar directamente a la página 3, pero esto no era óptimo, ya que la URL de la página 3, por motivos de seguridad periódicamente cambia, la cual cosa nos hacía la aplicación inestable.

Una vez situados en el paso 2 y una vez cargada la página, se ha creado una función en Javascript para que el botón de entrar se pulse solo.

Después de esto, nos encontramos en el paso 3, que también tenemos que esquivar y también lo haremos con Javascript. Gracias al *Bundle* explicado anteriormente, podemos pasar datos de una actividad a otra. Lo que haremos será pasar los datos de identificación que ha introducido el usuario, de la actividad del Login a la actividad siguiente, que es la que nos muestra el expediente. Como ya dispondremos de los datos almacenados en variables, los introduciremos en las casillas correspondientes, mediante javascript. La función es la siguiente:

```
mWebView.setVisibility(View.INVISIBLE);  
mWebView.loadUrl("javascript: document.forms[0].idUserario.value='"+niu+"'; " +  
                  "document.forms[0].password.value='"+passwd+"'; nolanzar();" +  
                  "document.forms[0].submit();" );
```

Como podemos ver, al campo con identificador *idUserario* del primer formulario del documento, le metemos el valor del *niu* que ha introducido el usuario y que se ha pasado mediante el *Bundle*. Para la contraseña hacemos exactamente lo mismo. Finalmente llamamos la función *nolanzar()* usada por la página y llamamos a la función *submit()* que se encargara de pulsar el botón de aceptar.

Nota: el javascript se tiene que poner dentro del *WebView*, usando el método *loadUrl*, para que cuando la página ha cargado se apliquen los cambios.

Una vez hecho esto, ya nos cargará la página del expediente (paso 4), con lo cual ya nos habremos saltado todos los pasos y habremos hecho la identificación de forma transparente al usuario. Para que todavía quede más pulido, lo que se ha hecho ha sido jugar con la propiedad que tiene el *WebView* de la visibilidad, y en ciertos puntos hemos puesto que sea invisible, con lo cual, cuando cargue la página de Login y pasen los datos, el usuario no lo verá.

```
mWebView.setVisibility(View.INVISIBLE);
```

Una vez cargada la página del expediente, tendremos que parsearla.

*Parseo expediente antiguo*

Para poder parsear una página web, lo primero que se tiene que hacer es analizar con mucho detalle su código html.

Se ha decidido utilizar el lenguaje Javascript, ya que principalmente se utiliza en el lado del cliente y permite mejoras en la interfaz de usuario y en páginas web dinámicas. Por su definición podemos observar que es ideal para la tarea del parseo, ya que podremos ocultar fácilmente los tags del html que no queremos que se muestren y de esta manera podremos obtener los resultados que buscamos, el parseo.

La página del expediente está formada por tres frames como podemos ver en la siguiente figura:

Any Acadèm.	Coef.	Descripció	Cr.	Gr.	Convocatòria	Qualificació	Q.M.	Cvl.	Tipus assig.	Cic.	Cur.	Observació
2010/11-0	20384	Compiladors I	6.0	1	febrer	Aprobat	6.4	N	Troncal	2	XIX	
2010/11-0	20381	Bioinformàtica	6.0	51	febrer	Notable	7.5	N	Optativa	2	XIX	
2010/11-0	20385	Arquitectura avançada	6.0	51	febrer	*No Presentat		N	Lliure Elecció	1	3	
2010/11-0	20391	Visió per computador	6.0	1	juny			N	Lliure Elecció	1	3	
2010/11-0	22593	Projecte d'Enginyeria en Informàtica	15.0	2	juny			N	Obligatòria	2	XIX	
2010/11-0	25002	Compiladors II	4.5	1	juny			N	Troncal	2	XIX	
2010/11-0	25814	Comunicació en llengua francesa I	6.0	5	febrer	Notable	7.3	N	Lliure Elecció	1	3	
2010/11-0	25853	Introducció a les finances	6.0	5	febrer	Notable	8.0	N	Lliure Elecció	2	XIX	
2010/11-0	29270	Envellir bé: de la biologia mol·lecular als hàbits de vida	6.0		juny			N	Lliure Elecció	2	XIX	

Crèdits	Superats				Matriculats i No superats			
	Primer cicle	Segon cicle	Aïtres	Total	Primer cicle	Segon cicle	Aïtres	Total
Troncal	0.0	6.0	0.0	6.0	0.0	4.5	0.0	4.5
Obligatòria	0.0	0.0	0.0	0.0	0.0	15.0	0.0	15.0
Optativa	0.0	6.0	0.0	6.0	0.0	0.0	0.0	0.0
Lliure Elecció	6.0	6.0	0.0	12.0	12.0	6.0	0.0	18.0
Complement de Formació	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Sense determinar	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Total	6.0	18.0	0.0	24.0	12.0	25.5	0.0	37.5

Fig.23.

Lo primero que quiere hacer es ocultar los frames 1 y 2, de tal forma que solo nos quede el frame 3. Para ellos, se ha reducido al máximo el ancho del frame 1 y la altura del frame 2, accediendo a sus propiedades y cambiándolas. Se ha hecho una función javascript que consiste en cambiar los atributos de los frames.

```
mWebView.setVisibility(Webview.VISIBLE);
```

```
String javascript = "javascript: window.onload = selection (); ";
javascript = javascript + " function selection () { ";
javascript = javascript + " var hijos = document.getElementsByTagName('frameset'); ";
javascript = javascript + " hijos[0].setAttribute('rows','1*,0'); ";
javascript = javascript + " hijos[1].setAttribute('cols','0,1*'); ";
javascript = javascript + " hijos[2].setAttribute('rows','0,0'); ";
```

Esta vez hemos tenido que cambiar la visibilidad a **VISIBLE**, ya que como en el paso anterior estaba **INVISIBLE**, no se veía nada.

Una vez hecho esto, únicamente se verá el frame 3, que son los datos del alumno y la tabla de las notas. Se quiere acceder a este frame para ir accediendo a los diferentes tags e irlos ocultando. Esto no se ha conseguido hacer, ya que me han cambiado la página del expediente por otra y todavía no lo tenía hecho.

Mirar finalmente la implementación del parseo en el apartado *Parseo expediente nuevo*.

### ***Resultados Expediente antiguo***



Fig.24.

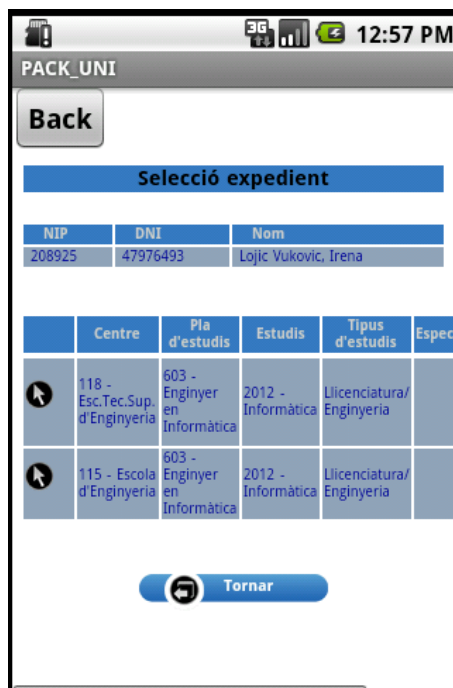


Fig.25.



Fig.26.

Podemos apreciar que los resultados son los esperados, ya que el frame izquierdo y el de abajo han desaparecido, la cual cosa implica que la página quede adaptada a lo ancho de la pantalla. Lo que no se ha conseguido hacer es quitar información irrelevante como por ejemplo el botón de imprimir, el menú y algunos campos de los datos del alumno, así como algunas columnas de la tabla de notas. Esto no se ha conseguido porque se ha cambiado la página del expediente por sorpresa y la parte del parseo interno no la tenía hecha. No obstante, se ha adaptado lo hecho al expediente nuevo y es el que se ha parseado (Mirar apartado *Parseo expediente nuevo*).

### ***Cambio expediente***

Antes de entregar el proyecto han cambiado la página del expediente, ya que han hecho una migración del entrono de la BBDD y de paso han cambiado todo el aspecto de la página. He tenido la mala suerte de que han tocado parte del código y la aplicación ya no me funcionaba. He decidido investigar que partes han modificado para intentar adaptar la aplicación a la versión nueva con la finalidad de tener las dos.

En términos generales, he visto que el expediente antiguo estaba compuesto por framsets anidados, los cuales dentro contienen frames, mientras el expediente nuevo usa iframes anidados solamente. Esto es una buena noticia, ya que el problema que tenía para acceder a los frames que estaban dentro de los framesets desaparecerá.

Seguidamente se explican los cambios que se han tenido que hacer para la adaptación de cada fase.

### *Identificación Transparente Expediente Nuevo*

Ahora para acceder al expediente por la vía normal, tenemos que seguir los siguientes pasos:

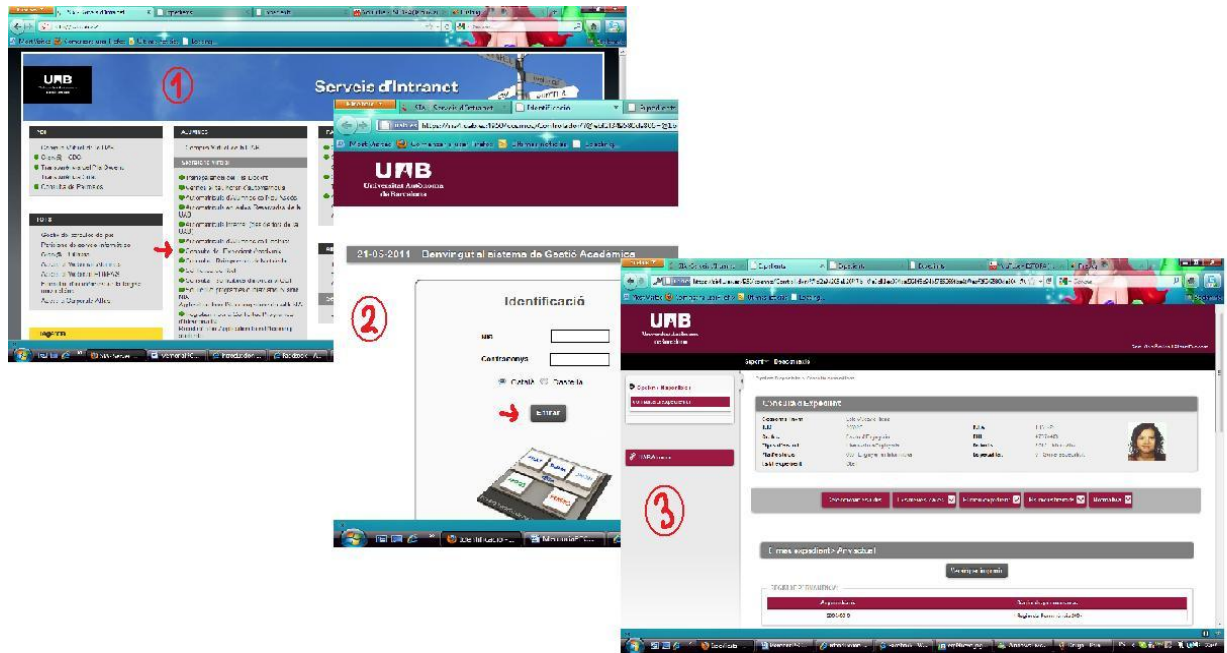


Fig.27.

Como podemos observar, ahora para acceder al expediente hay tres pasos en vez de cuatro. En este caso, tendremos que acceder directamente al paso 2, dónde se pasara la identificación introducida por la actividad de login de la aplicación. Pero, esto no nos influye, ya que antes accedíamos a la página que ya no está, pulsábamos el botón aceptar y llegábamos al login. Ahora tenemos que hacer que directamente aparezca el login.

La identificación transparente se puede reaprovechar ya que no ha cambiado. Lo único que se ha tenido que modificar ha sido la URL de la página de entrada ya que es otra. También hemos tenido que quitar el código javascript que hacía que se pulsara el botón de la página que ya no está y hacer que directamente se re direcciona a la del login.

*Parseo Expediente Nuevo*

En la siguiente figura se muestra el aspecto de la nueva página del expediente académico:

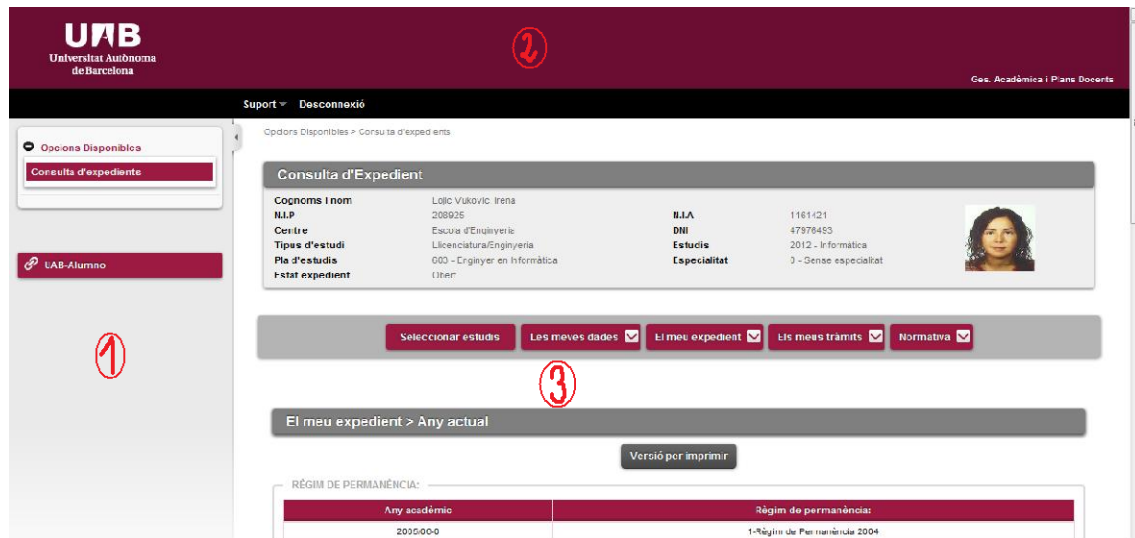


Fig.28.

Primeramente, se quiere quitar la barra izquierda (1) y la barra superior (2). Analizando el código hemos podido observar que cada una de las barras está dentro de sus correspondientes <div> y no dentro de frames como en el caso anterior. Además, cada uno de los divs tiene un identificador único. Por lo tanto, se ha accedido a cada uno de estos divs por su identificador y se han ocultado utilizando las propiedades de Javascript, de la siguiente manera:

```
mWebView.loadUrl("javascript:document.getElementById('NavBarraIzqda').style.display='none';");

mWebView.loadUrl("javascript:document.getElementById('NavBotonOcultarBarraIzqda').style.display='none';");
mWebView.loadUrl("javascript:document.getElementById('NavBotonMostrarBarraIzqda').style.display='none';");

mWebView.loadUrl("javascript:document.getElementById('NavZonaCabecera').style.display='none';");
mWebView.loadUrl("javascript:document.getElementById('NavZonaOpciones').style.display='none';");

mWebView.loadUrl("javascript:document.getElementById('NavZonaLogo').style.display='none';");
mWebView.loadUrl("javascript:document.getElementById('caminoAriadna').style.display='none';");
```



## Aplicació Android per Estudiants Universitaris

Haciendo esto solamente nos queda visible la parte interior (3) que es la siguiente:

Consulta d'Expedient

Cognoms i nom

Lojic Vukovic, Irena

N.I.P

208925

Centre

Escola d'Enginyeria

Tipus d'estudi

Llicenciatura/Enginyeria

Pla d'estudis

603 - Enginyer en Informàtica

Estat expedient

Obert

N.I.A

1161421

DNI


47976493

Estudis

2012 - Informàtica

Especialitat

0 - Sense especialitat



Seleccionar estudis

Les meves dades

El meu expedient

Els meus tràmits

Normativa

El meu expedient > Any actual

Versió per imprimir

RÈGIM DE PERMANÈNCIA:

Any acadèmic

2005/06-0

Règim de permanència:

1-Règim de Permanència 2004

Les notes entre parèntesi \* no són definitives fins al tancament d'actes

Situi el ratolí sobre el símbol  on aparegui, per a visualitzar la notes parcials de l'assignatura

Situi el ratolí sobre el símbol  on aparegui, per a visualitzar la informació sobre la revisió d'exàmen

	Any Acadèmic	Codi	Descripció	Cr.	Gr.	Convocatòria	Qualificació	Q.M.	Cv.L.	Tipus assign.	Cic.	Cur.	Observacions
	2010/11-0	20384	Compiladors I	6.0	1	febrer	Aprovat	6.4	N	Troncal	2	XX	
	2010/11-0	20381	Bioinformàtica	6.0	51	febrer	Notable	7.5	N	Optativa	2	XX	
	2010/11-0	20385	Arquitectura avançada	6.0	51	febrer	*No Presentat		N	Lliure Elecció	1	3	
	2010/11-0	20385	Arquitectura avançada	6.0	51	juny	( ) 		N	Lliure Elecció	1	3	
	2010/11-0	20391	Visió per computador	6.0	1	juny	( ) 		N	Lliure Elecció	1	3	
	2010/11-0	22593	Projecte d'Enginyeria en Informàtica: Projecte d'Enginyeria en Informàtica: "Confeció del portal de l'estudiant universitari"	15.0	2	juny	( ) 		N	Obligatòria	2	XX	
	2010/11-0	25002	Compiladors II	4.5	1	juny	( ) 		N	Troncal	2	XX	
	2010/11-0	25814	Comunicació en llengua francesa I	6.0	5	febrer	Notable	7.3	N	Lliure Elecció	1	3	
	2010/11-0	25853	Introducció a les finances	6.0	5	febrer	Notable	8.0	N	Lliure Elecció	2	XX	
	2010/11-0	29270	Envellir bé: de la biologia mol·lecular als hàbits de vida	6.0	5	juny	( ) 		N	Lliure Elecció	2	XX	

Crèdits	Superats				Matriculats i No superats			
	Primer cicle	Segon cicle	Altres	Total	Primer cicle	Segon cicle	Altres	Total
Troncal	0,0	6,0	0,0	6,0	0,0	4,5	0,0	4,5
Obligatòria	0,0	0,0	0,0	0,0	0,0	15,0	0,0	15,0
Optativa	0,0	6,0	0,0	6,0	0,0	0,0	0,0	0,0
Lliure Elecció	6,0	6,0	0,0	12,0	12,0	6,0	0,0	18,0
Complement de Formació	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
Sense determinar	0,0	0,0	0,0	0,0	0,0	0,0	0,0	0,0
Total	6,0	18,0	0,0	24,0	12,0	25,5	0,0	37,5

Fig.29.

Todo lo que está en verde es lo que se pretende ocultar y para ello hemos tenido que analizar y entrar en los iframes para ocultar los correspondientes tags.

Esta parte está compuesta por dos iframes anidados. Todo lo que está encima de “El meu expedient” está dentro del iframe externo y todo lo demás está en el iframe interno. Con lo cual, para ocultar los campos señalados de los datos personales y el menú de las diferentes opciones, habrá que acceder solamente al iframe externo. Por otro lado, para ocultar todo lo demás señalado habrá que acceder al iframe interno.

El acceso al iframe externo se hace de la siguiente forma:

```
javascript = javascript +"document.getElementById('iFrameAplZonaAplicacion')." +  
                        "contentWindow.document.getElementById('X')." +  
                        "style.display='none';";
```

Dónde, X será el identificador del tag que queramos ocultar.

El acceso al iframe interno se hace de la siguiente forma:

```
javascript = javascript +"document.getElementById('iFrameAplZonaAplicacion')." +  
                        "contentWindow.document.getElementById('contenido')." +  
                        "contentWindow.document.getElementById('Y')." +  
                        "style.display='none';";
```

Dónde, Y será el identificador del tag que queramos ocultar.

Sabiendo cómo hacer los accesos, parsear la página es muy sencillo, ya que se trata de ir accediendo a los tags por su identificador e irlos ocultando. Cuando se trate de una tabla habrá que acceder por su fila y por su columna para poder parsear campo a campo.

Para información más detallada, mirar el código fuente.

### Resultados Expediente Nuevo

Fig.30.

NIP	DNI	Nom
208925	47976493	Lojic Vukovic, Irena

Centre	Estudis	Tipus d'estudis	Estat/Motiu
118 - Esc.Tec.Sup. d'Enginyeria	2012 - Informàtica	Llicenciatura/ Enginyeria	Tancat per segregació de centres
115 - Escola d'Enginyeria	2012 - Informàtica	Llicenciatura/ Enginyeria	Obert

Fig.31.

Fig.32.

Haciendo el scroll, podemos observar como la tabla de notas queda parseada solamente con las columnas que nos interesan.

Descripció	Convocatoria	Qualificació	Q.N.	Tipus assig.
Compiladors I	febrer	Aprovat	6.4	Troncal
Bioinformàtica	febrer	Notable	7.5	Optativa
Arquitectura avançada	febrer	*No Presentat		Lliure Elecció
Arquitectura avançada	juny	( )		Lliure Elecció
Visió per computador	juny	( )		Lliure Elecció
Projecte d'Enginyeria en Informàtica: Projecte d'Enginyeria en Informàtica: "Confecció del portal de l'estudiant universitari"	juny	( )		Obligatòria
Compiladors II	juny	( )		Troncal

Fig.33.

Comunicació en llengua francesa I	febrer	Notable	7.3	Lliure Elecció
Introducció a les finances	febrer	Notable	8.0	Lliure Elecció
Envellir bé: de la biologia mol·lecular als hàbits de vida	juny	( )		Lliure Elecció

Crèdits	Superats	No superats
	Total	Total
Troncal	6,0	4,5
Obligatòria	0,0	15,0
Optativa	6,0	0,0
Lliure Elecció	12,0	18,0
Complement de Formació	0,0	0,0
Sense determinar	0,0	0,0
Total	24,0	37,5

Fig.34.

Los resultados son los esperados ya que la pagina queda totalmente adaptada a la pantalla del teléfono y solo salen las columnas más relevantes.

### Identificación Transparente Laboratorios

Como en el caso del expediente, queremos saltarnos la identificación de la página web, y hacerla a través de la interfaz del dispositivo. Para ello, aprovecharemos las propiedades del JavaScript, como en el caso del expediente.

Primero, utilizando el método loadUrl de WebView nos conectaremos con la página de login del neptu:

```
mWebView.loadUrl("http://neptu.uab.es/PSG/Login.do");
```

Seguidamente, gracias al Bundle definido en la actividad de Login del dispositivo, podremos obtener el niu y contraseña introducidos por el usuario, llamándolo en esta actividad. Una vez tenemos los datos en variables, utilizaremos una función echada en JavaScript para introducir estos datos en las casillas correspondientes del logeo de la página.

```
mWebView.loadUrl("javascript: document.forms[0].userName.value='"+niu2+"' ; " +
    "document.forms[0].password.value='"+passwd2+"' ;" +
    "document.forms[0].submit();" );
```

Como podemos ver, en el campo con identificador *userName*, del primer formulario del documento, introducimos el valor del niu pasado por el *Bundle*. Hacemos lo mismo para la contraseña y llamando a la función *submit()* se pulsa el botón aceptar. Para dejarlo todavía más pulido, jugaremos con la propiedad de visibilidad del *WebView* y la pondremos a *INVISIBLE*, para que de esta forma el usuario no vea nada de este proceso.

```
mWebView.setVisibility(View.INVISIBLE);
```

Con esto, la identificación transparente de los laboratorios queda finalizada.

### Parseo Laboratorios

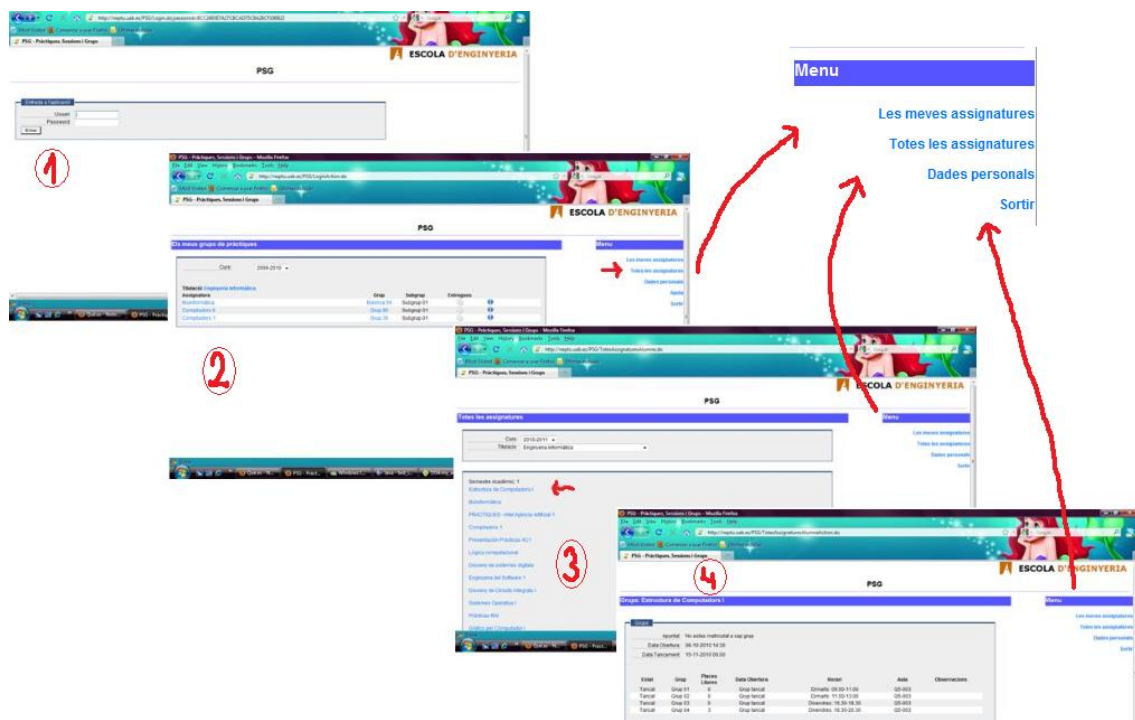


Fig.35.

Como podemos ver en la siguiente figura, una vez identificados, nos situamos en el paso dos. Lo que realmente nos interesa, es poder saltarnos el paso 2 y que directamente al logearnos a través de la interfaz del dispositivo, directamente nos apareciera el paso 3, para que el usuario pueda escoger una asignatura entre todas las que está matriculado. El paso 2, simplemente contiene las asignaturas a las que se ha apuntado a alguna práctica, con lo cual esto no nos interesa. Para poder acceder al paso 3, se tiene que pulsar el link "Totes les Asignatures" del menú de la derecha. Esto queremos que se haga solo y transparentemente, una vez nos hemos

identificado, para que directamente nos salga el paso 3. El menú de la derecha también lo queremos ocultar, porque no lo vamos a necesitar.

Esto lo haremos con JavaScript de la siguiente forma:

```
mWebView.loadUrl("javascript: location.href='/PSG/TotesAssignaturesAlumne.do'");
mWebView.loadUrl("javascript: document.getElementById('menu_dreta').style.display='none';")
```

Con la primera línea estamos diciendo que el enlace “TotesAssignatures” de la página, se pulse. En la segunda línea estamos ocultando en menú de la derecha, cogiendo del documento el id del menú y ocultándolo. Por último, tendremos que poner la página INVISIBLE, para que este proceso no se vea

```
mWebView.setVisibility(View.INVISIBLE);
```

Una vez hecho esto, estaremos en el paso 3 y el usuario podrá escoger la carrera que desea entre las posibles y cuando le aparezcan las asignaturas, deberá de escoger alguna. En el paso 4, también queremos quitar el menú de la derecha y de la tabla tampoco nos interesa la columna de observaciones ni la del aula, ya que la finalidad es apuntarse a los grupos de prácticas y saber la fecha. Estas columnas se han quitado para que la página quedara ajustada a la pantalla del teléfono. Se han cogido estas porque se han considerado las menos necesarias.

Después de estudiar detalladamente cómo está organizada la tabla en la página, para eliminar la columna de observaciones lo hemos hecho de la siguiente forma:

```
mWebView.loadUrl("javascript: document.getElementById('menu_dreta').style.display='none';")
mWebView.loadUrl("javascript: var nt= document.getElementsByTagName('table');");
mWebView.loadUrl("javascript: var ntr= document.getElementsByTagName('tr');");
mWebView.loadUrl("javascript: nt[1].getElementsByTagName('td')[7].style.display='none';");
mWebView.loadUrl("javascript: nt[1].getElementsByTagName('td')[15].style.display='none';");
mWebView.loadUrl("javascript: nt[1].getElementsByTagName('td')[23].style.display='none';");
mWebView.loadUrl("javascript: nt[1].getElementsByTagName('td')[31].style.display='none';");
mWebView.loadUrl("javascript: nt[1].getElementsByTagName('td')[39].style.display='none';");
mWebView.loadUrl("javascript: nt[1].getElementsByTagName('td')[47].style.display='none';");
mWebView.loadUrl("javascript: nt[1].getElementsByTagName('td')[55].style.display='none';");
```

Y así para el máximo de filas (buscar entre todas las asignaturas la que tiene el máximo de filas).

Para eliminar la columna del aula, hemos hecho lo siguiente:

```
mWebView.loadUrl("javascript: nt[1].getElementsByTagName('td')[6].style.display='none';");
mWebView.loadUrl("javascript: nt[1].getElementsByTagName('td')[14].style.display='none';");
mWebView.loadUrl("javascript: nt[1].getElementsByTagName('td')[22].style.display='none';");
mWebView.loadUrl("javascript: nt[1].getElementsByTagName('td')[30].style.display='none';");
mWebView.loadUrl("javascript: nt[1].getElementsByTagName('td')[38].style.display='none';");
mWebView.loadUrl("javascript: nt[1].getElementsByTagName('td')[46].style.display='none';");
mWebView.loadUrl("javascript: nt[1].getElementsByTagName('td')[54].style.display='none';");
mWebView.loadUrl("javascript: nt[1].getElementsByTagName('td')[62].style.display='none';");
```

En este caso, eso también lo hemos hecho para el máximo de filas.

En resumen, lo que hacemos es del documento recogemos en un vector todas las tablas y nos quedamos con la segunda y es de la que vamos ocultando las filas que le vamos diciendo.

### Resultados Laboratorios



Fig.36.



Fig.37.

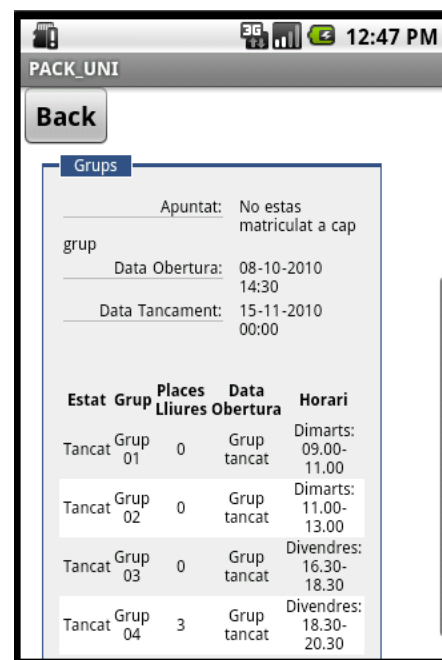


Fig.38.

Como podemos observar, hemos obtenido los resultados deseados, ya que un usuario, al introducir su usuario y contraseña y darle al botón aceptar, directamente se le cargan todas las asignaturas. Cuando selecciona alguna, nos aparece la tabla con la información, sin las observaciones y sin el aula, tal y como queríamos. Además, los menús de la derecha tampoco aparecen. También podemos apreciar, que la pagina está perfectamente adaptada al dispositivo respecto a lo ancho, que es lo importante.



## Capítulo 5 - Puesta en Marcha

---

*En este capítulo se explican los pasos a seguir para distribuir una aplicación hecha para Android y también como crear el ejecutable para poder testarlo sobre un dispositivo real.*

## Dispositivo Real

### Firma

Una vez finalizada la aplicación Android, ésta se tiene que exportar en formato .apk para poder instalarse en los teléfonos móviles.

Antes de nada, la aplicación se tendrá que firmar. Firmar una aplicación es una medida de seguridad y de garantía. Firmaremos una aplicación Android para poder distribuirla en el Android Market o en cualquier otro medio y poder instalarla en nuestros dispositivos sin ningún problema. También nos garantiza que solo nosotros podamos modificar y actualizar la aplicación.

En este caso, la firma la haremos con el Eclipse. Necesitamos un Keystore (almacén de claves dónde se encuentran todos los certificados ya convalidados).

Como no tenemos inicialmente ninguna Keystore vamos a crear uno y lo guardaremos en nuestro ordenador. Seguidamente rellenamos un formulario, ponemos la clave de nuestro certificado y ponemos la ruta dónde se guardará nuestra aplicación firmada.

Para las próximas veces no hará falta crear un nuevo certificado, sino solamente ponemos la contraseña anterior y podremos exportar directamente.

Una vez hecho esto, ya tendremos nuestro archivo .apk firmado y podremos instalarlo en cualquier dispositivo Android

Nota: en número de años que queremos que dure el keystore, pondremos un número grande, para no tener problemas a la larga, ya que si el keystore caduca no podremos modificar ni actualizar nuestras aplicaciones (25 años).

### Resultados

Tras la instalación de PACK\_UAB en un dispositivo móvil, los resultados se pueden ver en el *Annexo III* (Juego de Pruebas).

## Android Market

Para poder subir aplicación en el Android Market, lo primero que hay que hacer es crear una cuenta en Google Checkout y abonar 25\$ para toda la vida. Después del pago ya tendremos acceso a las pantallas que nos permitirán subir aplicaciones.

Deberemos rellenar un formulario que se encuentra actualmente dividido en tres secciones:

La parte de **assets** que es dónde se sube el .apk, las imágenes promocionales, el icono, un vídeo para youtube, etc.

La parte de **listing details**, con la descripción de la aplicación en diferentes idiomas.

La parte de **publishing options**, con la protección y clasificación por edades, además de la información de contacto y aceptaciones correspondientes.

Una vez publicada la aplicación, en breves minutos ya se encontrará en el Market. A partir de aquí comenzaremos a recibir feedback del número de descargas, de comentarios y de errores.

## Capítulo 6 – Planificación y presupuesto

---

*En este capítulo se refleja la planificación temporal del proyecto, concretamente la inicial y se hace una reflexión sobre si se han cumplido o no los plazos. Además, también se detallan todos los costes que se deberían de asumir si este proyecto fuera realizado por una empresa (coste de personal, coste de software y coste de otros materiales).*

## Planificación temporal

Inicialmente, se ha pensado trabajar en el proyecto 3 horas diarias de Lunes a Viernes desde el 15/11/2010 hasta el 30/05/2011, con flexibilidad de horas y de días. En total, semanalmente, se han de invertir 15 horas. En época de exámenes, se ha tenido en cuenta y se ha considerado una semana de vacaciones para poder estudiar, tanto en febrero como en junio.

Planificación inicial:

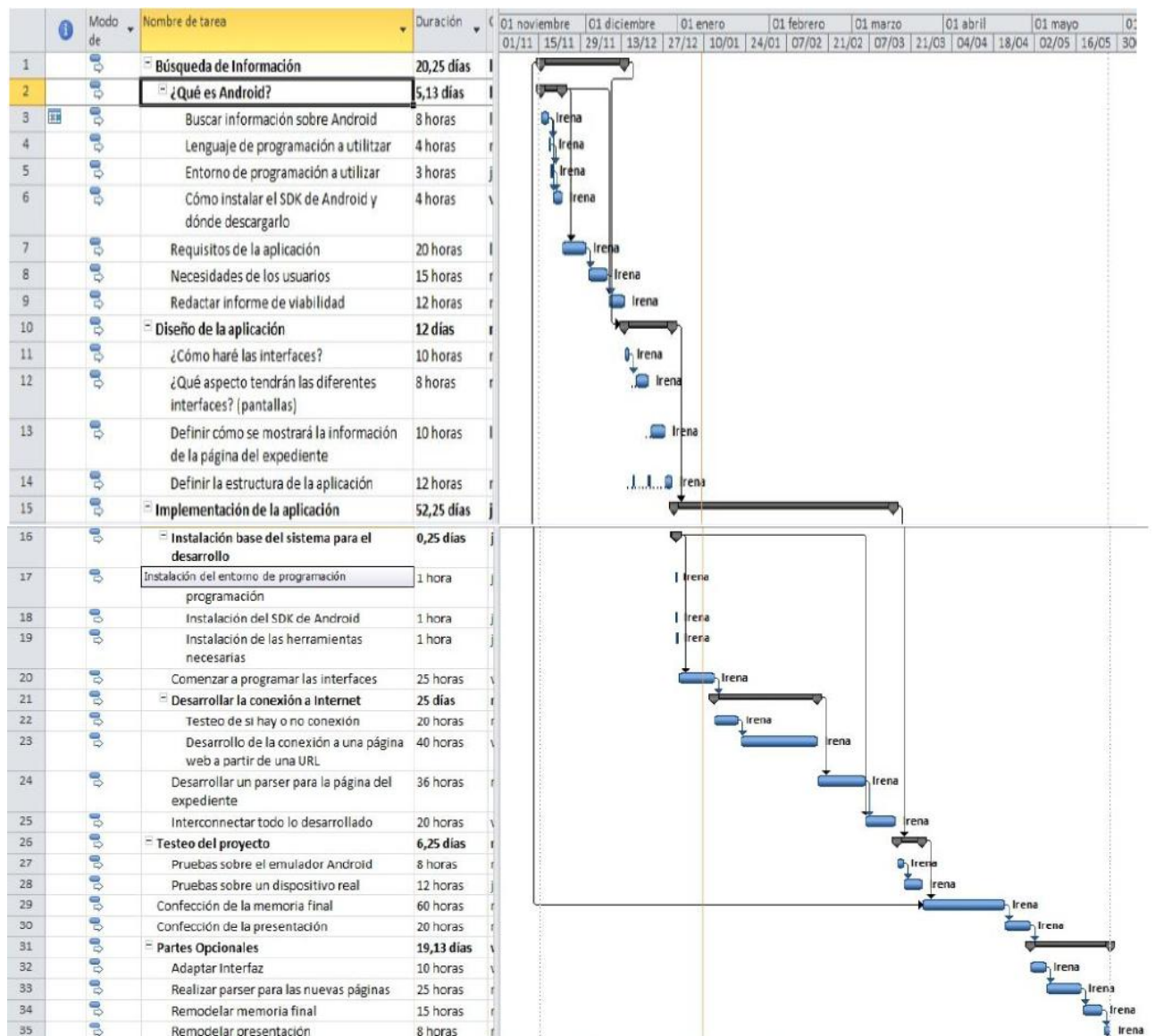


Fig.61.

Realmente, como era de esperar, la planificación no se ha cumplido al 100%.

Se ha tardado menos en cosas que se había previsto mucho más tiempo y a la inversa. Por ejemplo, para realizar la conexión a Internet no se ha tardado tanto como se esperaba. Realmente la mayoría del tiempo invertido ha sido investigando como hacerla, pero cuando tope con el WebView de Android y entendí cómo funcionaba, realizar la conexión con una URL se hizo en un momento. Si que se tardó tiempo en encontrar que existía el WebView, pero en total, el tiempo utilizado ha sido bastante menos que lo que ponía.

Por otro lado, en el parser del expediente se ha tardado bastante más de lo previsto, porque me he encontrado con problemas graves con el acceso a los iframes anidados, que me ha costado mucho solucionar. Lo he solucionado en el último momento, después de probar muchísimas cosas. Esto ha pasado, porque cuando hice la planificación inicial, todavía no me había mirado el código de la página con lo cual es normal que no estipulara bien el tiempo que tardaría. En cambio, en el parser de los laboratorios más o menos he tardado lo que puse ya que este no tenía iframes.

También hay que destacar que a medio proyecto cambió la página del expediente, con lo cual tuve que perder más tiempo mirando si habían realizado muchos cambios y cuales, para poder aprovechar lo que tenía y adaptar el código al nuevo expediente.

También ha habido épocas en las que he trabajado más horas semanales de las estipuladas, concretamente durante los 20 días de junio y las semanas antes de las reuniones.

También he invertido horas haciendo la maqueta de la pizarra virtual que es la propuesta de futuro.

El resto de las partes del proyecto más o menos si que ha cumplido la planificación inicial.

Por todo esto, no he cumplido en cuanto a fechas, ya que en la planificación inicial se había estipulado finalizar el proyecto el 30 de mayo y finalmente se ha finalizado el 20 de Junio, la cual cosa implica que se han realizado más horas de las estipuladas.

## Presupuesto

En este apartado se detalla el presupuesto aproximado del proyecto especificando el gasto de personal, del software, del material, entre otros, como si de una empresa real se tratara.

Se ha considerado que se trabajaba durante 3 horas semanales desde el 15/11/2010 hasta el 20/06/2011.

Primeramente calcularemos el total de horas invertidas en el proyecto (por fases), consultando los datos de la planificación temporal.

Fase de análisis: 20 días \* 3 horas/día = 60 horas

Fase diseño: 12 días \* 3 horas/día = 36 horas

Fase implementación: 53 días \* 3 horas/día = 159 horas

Fase pruebas: 6 días \* 3 horas/día = 18 horas

Partes opcionales: 20 días \* 3 horas/día = 60 horas

Fase documentación: 80 horas

### Coste de personal

	Nº de horas	Coste hora	Total (€)
ANALISTA	60 h	33 €	1980 €
DISEÑADOR	36 h	33 €	1188 €
PROGRAMADOR	159h + 60 h	25 €	5475 €
RESPONSABLE DE DOCUMENTACIÓN	80 h	15 €	1200 €
<b>TOTAL</b>			<b>9843 €</b>

### Coste de software y licencias

Durante el proyecto, se han utilizado diferentes aplicaciones y herramientas para generar tanto el código de la aplicación como toda la documentación. Por ello, se incluyen las licencias de dichas aplicaciones dentro del presupuesto.

Descripción	Coste imputable (€)
Microsoft Office 2007 Professional	289 €
Microsoft Office Project 2007	119 €
Microsoft Office PowerPoint 2007	119 €
Eclipse	0 €
SDK Android	0 €
<b>TOTAL</b>	<b>527 €</b>

### Coste de Hardware

He tenido que comprar un dispositivo Android para poder hacer las pruebas y ver si realmente funciona.

Descripción	Coste imputable (€)
HTC Wildfire	100 €

### Coste de material

Se ha utilizado varios cartuchos de tinta para imprimir documentación, además de diverso material de oficina. También se tiene que tener en cuenta la electricidad utilizada.



Descripción	Coste (€)
Recambios impresora	200 €
Material de oficina	50 €
Electricidad	70 €
<b>TOTAL</b>	<b>320 €</b>

### Resumen de costes

Finalmente mostraremos una tabla resumen con los costes totales aproximados con y sin IVA que nos supondría llevar a cabo este proyecto.

Descripción	Costes totales (€)
Personal	9843 €
Costes de software	527 €
Costes de hardware	100 €
Costes de materiales	320 €
<b>TOTAL (sin IVA)</b>	<b>10790 €</b>
<b>TOTAL (con 18% IVA)</b>	<b>12732,2 €</b>

Hay que destacar que este presupuesto es aproximado.

## Capítulo 7 - Propuesta de futuro

---

*Como todo proyecto siempre hay mejoras y ampliaciones por hacer. En este apartado propongo una ampliación muy útil para mi aplicación que si en un futuro se consiguiera hacer tendría mucho éxito. Además, muestro resultados de la maqueta que he implementado para poder tener una mejor idea de lo que propongo.*

Se pretende añadir una funcionalidad nueva a la aplicación, llamada Pizarra Virtual. Se quiere poder establecer una comunicación instantánea entre profesor/alumno. Esta funcionalidad está relacionada con los grupos de prácticas de tal forma que un profesor pueda enviar un mensaje a todos los alumnos apuntados en su grupo de prácticas de una determinada asignatura.

Esta propuesta se plantea con la finalidad de evitar tener que escribir correos electrónicos a la hora de tener que informar sobre algo, ya que no son cómodos y tienes que estar consultándolos para saber si hay nuevos.

Se ha hecho una maqueta con la finalidad de mostrar lo que se pretende conseguir, pero realmente el desarrollo queda pendiente para un futuro, ya que es complicado y laborioso y me falta tiempo.

La maqueta se ha hecho jugando con los tags <div> de HTML y con la propiedad de JavaScript que nos permite mostrar u ocultar tags.

## Implementación

Utilizaremos SQLite para la creación de la base de datos. Creamos una tabla para los profesores, otra para los alumnos y otra para las asignaturas. De esta manera podremos vincular a un profesor con una asignatura y los correspondientes alumnos.

En este caso tendremos las siguientes actividades:

- Identificación (alumno/profesor)
- Menú de asignaturas profesor
- Menú de asignaturas alumno
- Por cada asignatura posibilidad de redactar un mensaje
- Por cada asignatura posibilidad de consultar los mensajes.

Tendremos que crear una interfaz de usuario para cada actividad, igual que hemos hecho con el expediente académico y con los laboratorios. Utilizaremos xml y podremos los widgets que le correspondan (botones, áreas de texto, etc..)

A su vez, por cada archivo xml deberemos crear un archivo .java, para darle funcionalidad.

Cuando un profesor quiere escribir un mensaje se identifica, selecciona la asignatura y redacta el mensaje. Cuando lo publica, se hace una consulta en la base de datos, para obtener todos los alumnos asociados a ese profesor. Una vez se saben los alumnos aparece el mensaje en su

pizarra. El caso inverso, cuando un alumno quiere consultar sus mensajes, se identifica, selecciona la asignatura y automáticamente le aparecen. La idea es que cuando un alumno tiene un mensaje nuevo, en el menú de asignaturas le aparezca una redondita roja en esa asignatura, para que no tenga que consultar tontamente.

## Resultados de la maqueta

Parte profesores:



Fig.39.



Fig.40.

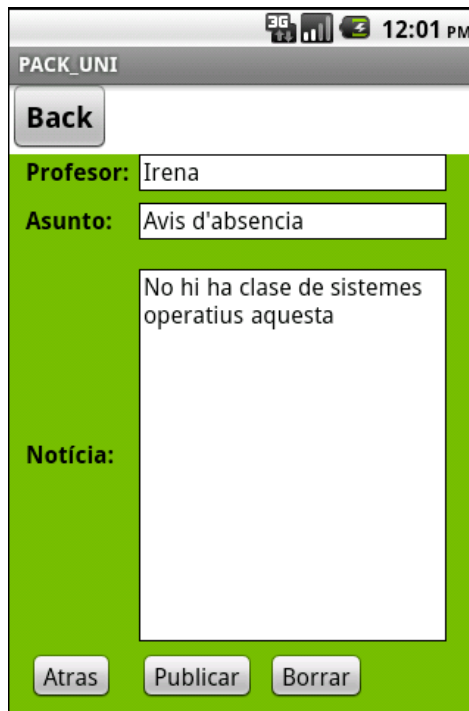


Fig.41.

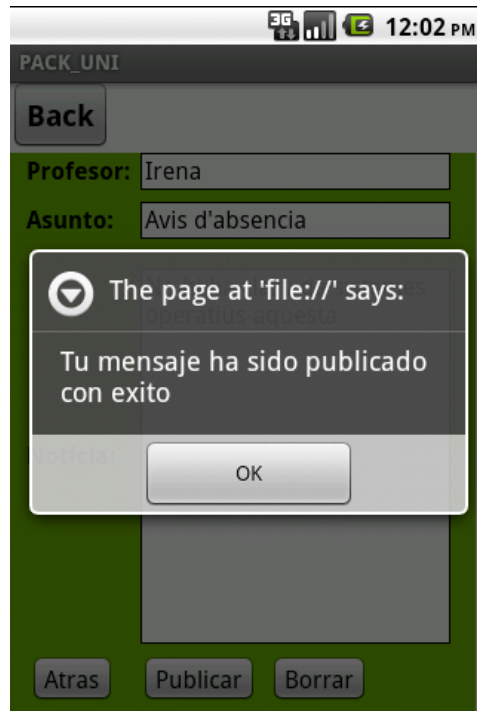


Fig.42.

Parte alumnos:



Fig.43.

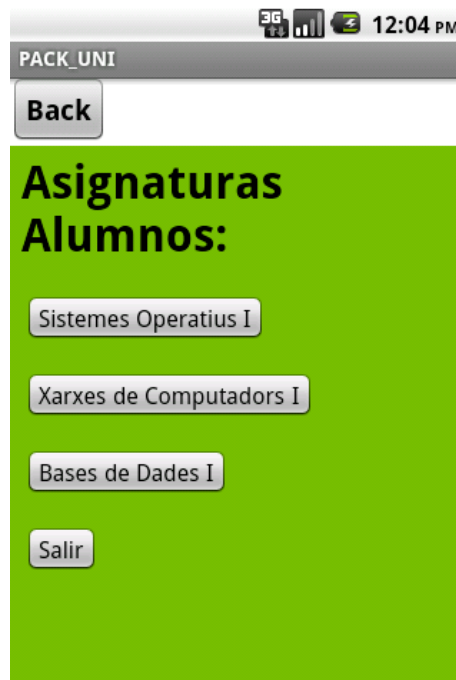


Fig.44.

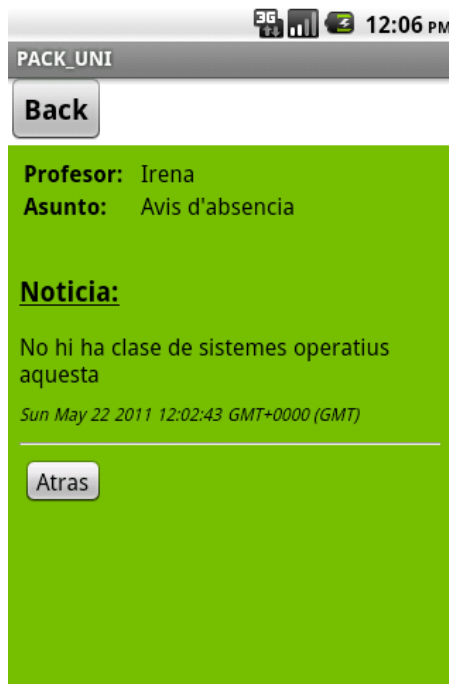


Fig.45.

Nota: En el caso de publicar más noticias de la misma asignatura, aparecerían todas ordenadas de más nueva a más antigua.

Ejemplo:

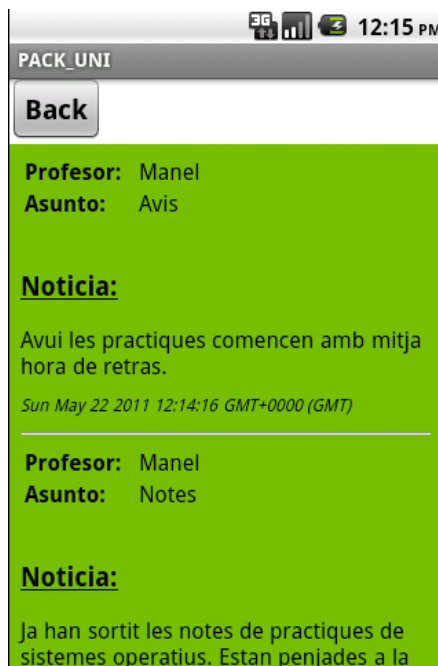


Fig.46.

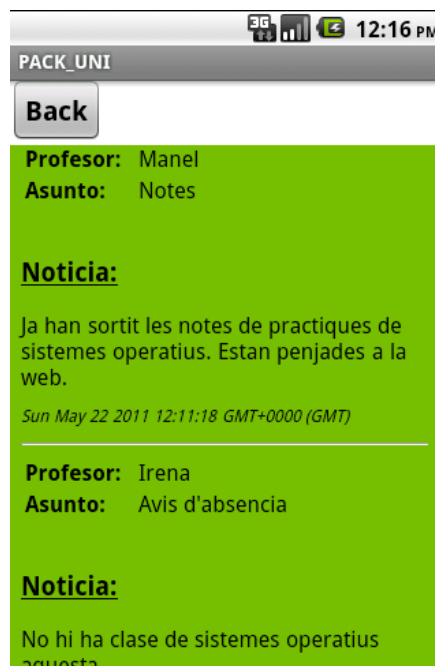


Fig.47

## Capítulo 8 – Conclusiones

---

*En este capítulo se mira atrás y hace una reflexión sobre cómo se ha desenvuelto el proyecto centrándonos en cada una de sus fases y se obtiene una valoración.*

Después de estar todo un curso académico dedicándome a este proyecto, puedo decir que he cumplido todos los objetivos que me propuse. He conseguido hacer una aplicación muy útil que permite a los estudiantes consultar su expediente académico y consultar o apuntarse a los grupos de prácticas, desde su teléfono Android. Lo mejor de todo es que ambas páginas están parseadas para que únicamente salga la información importante y además están adaptadas a la pantalla. Se puede decir que es la versión para “móvil”.

Estoy orgullosa de haber cumplido todos los objetivos, ya que no estaba segura de que lo conseguiría porque me he ido encontrando con varias dificultades que con paciencia y esfuerzo he conseguido solventar.

A lo largo del proyecto he adquirido un amplio conocimiento de la plataforma Android, de su funcionamiento interno y también he aprendido a desarrollar aplicaciones para dicha plataforma. Todo esto ha sido posible gracias a la extensa documentación que Google pone a disposición de los desarrolladores, aunque también ha sido muy importante el trabajo de investigación que he realizado al respecto ya que sin ello, esto no hubiese sido posible.

Por otro lado, gracias a este proyecto he podido poner en práctica diferentes aspectos aprendidos durante la carrera. Por un lado, he aplicado los conocimientos de la rama de Ingeniería del Software, siendo capaz de definir objetivos, requisitos y de realizar un buen diseño de la aplicación. También he demostrado los conocimientos adquiridos en cuanto a lenguajes y herramientas de desarrollo adquiridos durante la carrera y he demostrado la capacidad de aprender nuevos lenguajes y/o herramientas de desarrollo no utilizadas anteriormente, ya que yo la tecnología Android no la conocía ya que es muy reciente. También he puesto en práctica las capacidades de gestión de proyectos aprendidas a lo largo de estos años, sabiendo planificar y gestionar un proyecto desde la idea inicial hasta su implementación e puesta en marcha, pasando por cada una de las fases intermedias. Esto último es lo que realmente me reconforta, ya que me gustaría poder llegar a ser jefa de proyectos algún día. Realmente, me he sorprendido a mi misma al ver que toda esta gestión la he conseguido hacer sin problemas.

Puedo asegurar que este proyecto es totalmente viable, ya que he conseguido acabarlo en el tiempo previsto y además he podido probarlo sobre un terminal Android y ver que funciona correctamente.

He llegado a la conclusión de que con esfuerzo, ganas y un poco de ingenio, todo es posible, ya que yo la tecnología Android no la conocía y finalmente he sido capaz de llevar al cabo un proyecto con esta tecnología, con todo lo que ello representa.

Por último, quiero dar las gracias mi director, Ramon Grau, por haberme guiado durante el proyecto y por dedicarme parte de su tiempo.



## Capítulo 9 – Bibliografía

---

*En este capítulo se citan todos los links de las páginas web que nos han sido útiles durante el desarrollo de este proyecto, además de los libros o otras fuentes consultadas.*

## Links

- [1] Documentación para desarrolladores.  
<http://developer.android.com/index.html>
- [2] Documentación para desarrolladores (WebView).  
<http://developer.android.com/guide/webapps/index.html>
- [3] Web oficial de la plataforma Eclipse.  
<http://www.eclipse.org/>
- [4] SDK Android  
[http://developer.android.com/sdk/1.5\\_r2/index.html](http://developer.android.com/sdk/1.5_r2/index.html)
- [5] Wikipedia  
<http://es.wikipedia.org/>
- [6] Blog del departamento de desarrollo del sistema operativo Android.  
<http://android-developers.blogspot.com/>
- [7] Foro de programación en Android  
<http://www.anddev.org/>
- [8] Paleta de colores Android  
<http://www.colorhunter.com/tag/android/1>
- [9] Otros  
<http://www.wetere.de.com/2010/10/estructura-de-un-proyecto-android-en-eclipse/>  
<http://www.android-spa.com/info.php>  
<http://www.sgoliver.net/blog/?p=1316>

## Libros

- **Hello, Android : introducing Google's mobile development platform**  
Burnette, Ed  
Raleigh, N.C. : Pragmatic Bookshelf, cop. 2009  
2nd ed.
- **Pro Android 2**  
Hashimi, Sayed Y.  
New York : Apress : Distributed to the book trade worldwide by Springer-Verlag, cop.  
2010
- **JavaScript [Recurs electrònic] : the good parts**  
Crockford, Douglas  
Sebastopol, Calif. : O'Reilly Media, c2008  
1st ed

## Capítulo 10 - ANNEXOS

---

*En este capítulo se adjuntan otras informaciones que puedan ser de interés sobre el proyecto, como por ejemplo, la planificación temporal, los casos de uso con más detalle, las herramientas utilizadas para su desarrollo, entre otros.*

## ANNEXO I : Casos de Uso

### Escenario I: Uso del resto del móvil

Nombre	Uso del resto de móvil
<b>Descripción</b>	Uso de las funcionalidades del móvil que no sean del PACK_UAB, como llamadas, sms, juegos, ...
<b>Actores</b>	Usuario
<b>Precondiciones</b>	Ninguna
<b>Flujo normal</b>	-Encender el móvil
<b>Excepciones</b>	Ninguna
<b>Postcondiciones</b>	Ninguna

Fig.50.

### Escenario II: Aplicación PACK\_UAB

Nombre	Aplicación PACK-UAB
<b>Descripción</b>	Ejecuta la aplicación PACK_UAB para poder utilizarla
<b>Actores</b>	Usuario
<b>Precondiciones</b>	Tener la aplicación instalada correctamente
<b>Flujo normal</b>	-Clickar sobre el icono de la aplicación
<b>Excepciones</b>	Ninguna
<b>Postcondiciones</b>	Ninguna

Fig.51.

### Escenario III: Opción Expediente

Nombre	Opción Expediente
<b>Descripción</b>	Opción para poder consultar el expediente académico
<b>Actores</b>	Usuario
<b>Precondiciones</b>	Ninguna
<b>Flujo normal</b>	-Saltarte la pantalla de bienvenida apretando el botón next

	-Seleccionar la opción Expediente del menú
<b>Excepciones</b>	Ninguna
<b>Postcondiciones</b>	Identificarse como alumno

Fig.52.

**Escenario IV: Opción Laboratorios**

<b>Nombre</b>	<b>Opción Laboratorios</b>
<b>Descripción</b>	Opción para poder consultar la página de los laboratorios
<b>Actores</b>	Usuario
<b>Precondiciones</b>	Ninguna
<b>Flujo normal</b>	-Saltarte la pantalla de bienvenida apretando el botón next -Seleccionar la opción Laboratorios del menú
<b>Excepciones</b>	Ninguna
<b>Postcondiciones</b>	Identificarse como alumno

Fig.53.

**Escenario V: Login**

<b>Nombre</b>	<b>Login</b>
<b>Descripción</b>	Garantizar el acceso a un usuario a la aplicación, es decir, garantizar que es alumno de nuestra universidad.
<b>Actores</b>	Usuario UAB
<b>Precondiciones</b>	Ser estudiante de la UAB (tener un niu y un password dados por la universidad)
<b>Flujo normal</b>	-Rellenar los campos de Niu y Password -Clickar el botón aceptar
<b>Excepciones</b>	-Niu inexistente -Niu existente, pero combinación niu-password incorrecta
<b>Postcondiciones</b>	El usuario se identifica y accede a la página solicitada

Fig.54.

**Escenario VI: Consulta de Expediente**

Nombre	Consulta de Expediente
<b>Descripción</b>	Acceder a la página personal del expediente
<b>Actores</b>	Usuario UAB
<b>Precondiciones</b>	Haberse identificado correctamente
<b>Flujo normal</b>	-Clicar botón aceptar después de haber metido niu/passwd  -Después de la carga aparecerá nuestro expediente
<b>Excepciones</b>	Fallo en la conexión a Internet
<b>Postcondiciones</b>	Ninguna

Fig.55.

**Escenario VII: Consulta/Inscripción de Laboratorios**

Nombre	Consulta/Inscripción de Laboratorios
<b>Descripción</b>	Acceder a la página personal de laboratorios, donde podremos consultar por asignaturas los grupos de prácticas donde nos hemos apuntado y en caso de que estén las inscripciones abiertas podremos inscribirnos.
<b>Actores</b>	Usuario UAB
<b>Precondiciones</b>	Haberse identificado correctamente
<b>Flujo normal</b>	-Clicar al botón aceptar después de haber metido niu/passwd  -Después de la carga aparecerán nuestras asignaturas
<b>Excepciones</b>	Fallo en la conexión a Internet
<b>Postcondiciones</b>	Ninguna

Fig.56.

**Escenario VIII: Opción Pizarra**

Nombre	Opción Pizarra
<b>Descripción</b>	Opción para poder acceder a la pizarra virtual y como profesor publicar noticias y como alumno consultarlas.
<b>Actores</b>	Usuario
<b>Precondiciones</b>	Ninguna
<b>Flujo normal</b>	-Saltarte la pantalla de bienvenida apretando el botón next  -Seleccionar la opción Pizarra del menú
<b>Excepciones</b>	Ninguna
<b>Postcondiciones</b>	Identificarse como alumno o como profesor

Fig.57.

**Escenario IX: LoginPizarra**

Nombre	LoginPizarra
<b>Descripción</b>	Identificarse como alumno o como profesor de la universidad
<b>Actores</b>	Usuario UAB
<b>Precondiciones</b>	Ser alumno o profesor de la universidad
<b>Flujo normal</b>	-Introducir niu y password  -Clickar el botón aceptar
<b>Excepciones</b>	-Niu inexistente  -Niu existente, pero combinación niu/passwd erronea  No tener conexión a Internet
<b>Postcondiciones</b>	Ninguna

Fig.58.



**Escenario X: Publicar Noticias**

Nombre	Publicar Noticias
<b>Descripción</b>	Permite a un profesor dejar una noticia a sus alumnos de prácticas.
<b>Actores</b>	Usuario UAB
<b>Precondiciones</b>	Haberse identificado correctamente como profesor
<b>Flujo normal</b>	<ul style="list-style-type: none"> <li>- Clicar al botón aceptar después de haber metido niu/passwd</li> <li>-Seleccionar una asignatura entre las disponibles</li> <li>-Escribir el nombre, asunto y mensaje</li> <li>-Clicar el botón Publicar</li> </ul>
<b>Excepciones</b>	Fallo de la conexión a Internet
<b>Postcondiciones</b>	Ninguna

Fig.59.

**Escenario XI: Consultar noticias**

Nombre	
<b>Descripción</b>	Permite a los alumnos consultar noticias publicadas por sus profesores de prácticas.
<b>Actores</b>	Usuario UAB
<b>Precondiciones</b>	Haberse identificado correctamente como alumno
<b>Flujo normal</b>	<ul style="list-style-type: none"> <li>-Clicar al botón aceptar después de haber metido niu/password</li> <li>-Seleccionar una asignatura entre las disponibles</li> </ul>
<b>Excepciones</b>	Fallo de la conexión a Internet
<b>Postcondiciones</b>	Ninguna

Fig.60.

## ANNEXO II: Herramientas de desarrollo

Los primeros pasos para comenzar a desarrollar con Android son configurar el entorno de desarrollo y esto lo haremos de la siguiente manera:

### Entorno de desarrollo

Lo primero es descargar e instalar Eclipse que es un entorno de desarrollo gratuito para unos cuantos lenguajes y que cuenta con muchas herramientas y plugins. Utilizaremos la versión para desarrolladores de Java.

Lo siguiente es instalar el plugin Android Development Tools o ADT en el Eclipse, para poder integrar Android con Eclipse.

### Android SDK

Android SDK (o *software development kit*) es el kit de desarrollo de software necesario para crear aplicaciones para Android. Dispone de todas las herramientas y las APIs necesarias para empezar a desarrollar aplicaciones que se puedan ejecutar en dispositivos con esta tecnología: entornos de desarrollo y depuración, librerías, un emulador de móviles, documentación, tutoriales, código de ejemplo, etc. Todo está incluido en el kit para que desde que se descargue e instale ya se pueda comenzar cualquier desarrollo.

Una vez instalado y preparado todo el entorno de desarrollo comentado anteriormente, desde el Eclipse tendremos que vincular la dirección dónde tenemos guardado el SDK con la finalidad de instalarlo. Una vez hecho esto, estamos listos para empezar a crear nuestro propio proyecto Android.

El SDK para Android se puede descargar de forma gratuita en la página oficial. Está soportado en S.O. *Windows, Linux y Mac*.

### ANNEXO III: Juego de pruebas

Se ha probado la aplicación en un dispositivo Android real y se han obtenido los siguientes resultados:



Fig.61.



Fig.62.

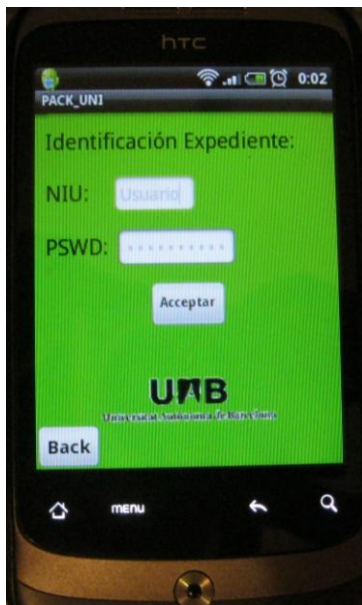


Fig.63.



Fig.64.



Fig.65.

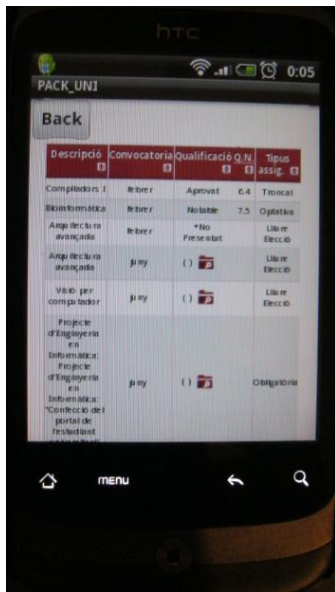


Fig.66.

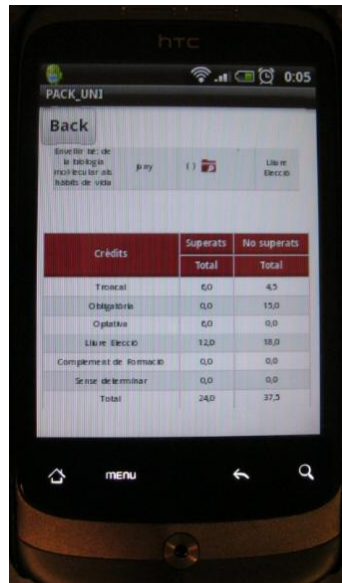


Fig.67.



Fig.68.



Fig.69.

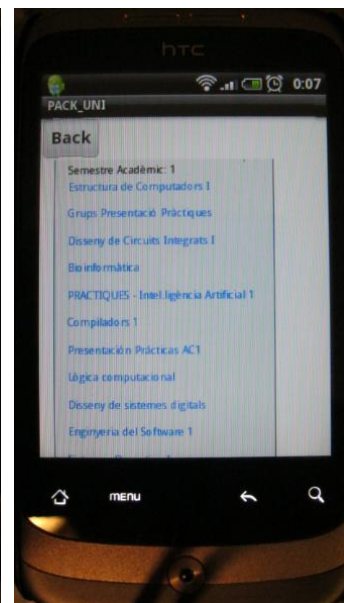


Fig.70.

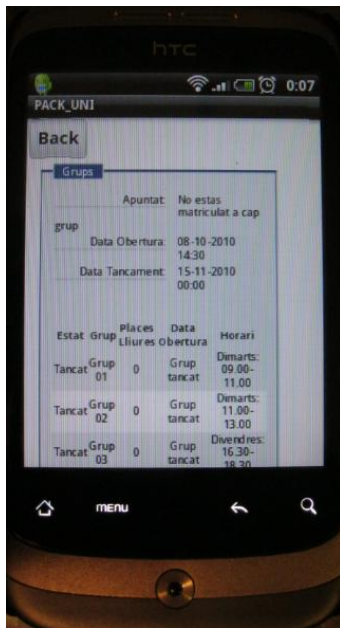


Fig.71.

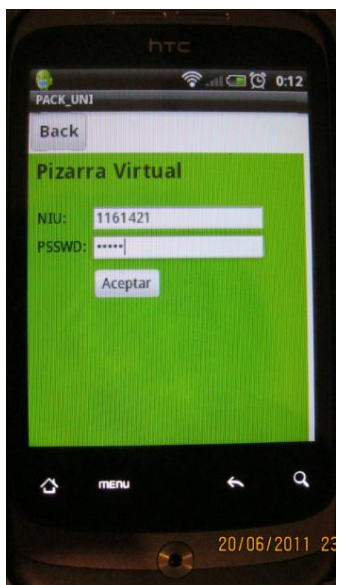


Fig.72.

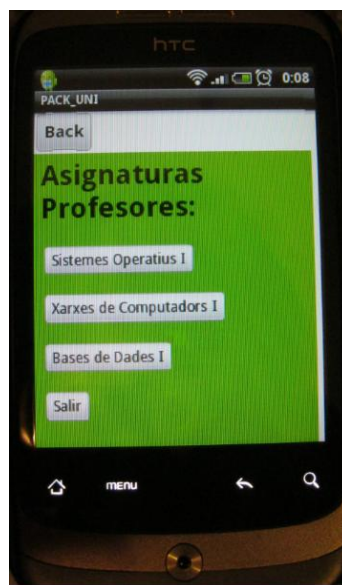


Fig.73.

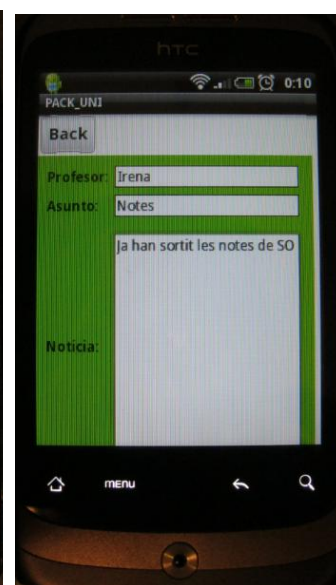


Fig.74.



Fig.75.

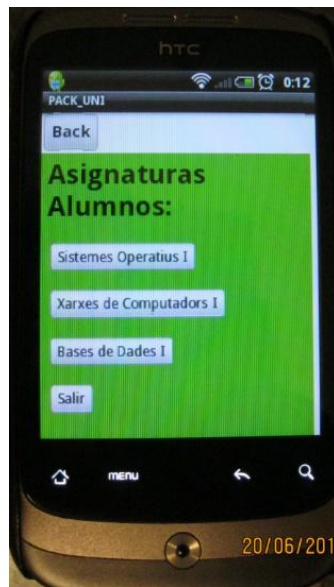


Fig.76.

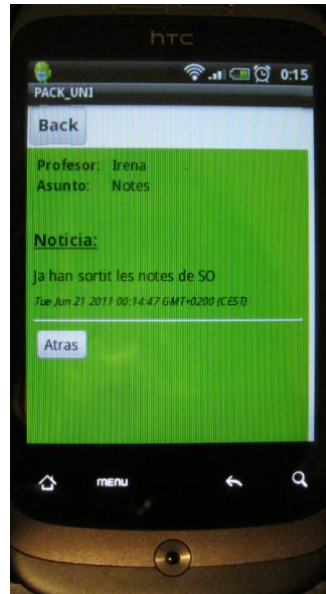


Fig.77.

Los resultados son explicados en la tabla de figuras (*Mirar Anexo IV*).

## ANNEXO IV: Tabla de figuras

Fig.1.	Escenario Cliente/Servidor
Fig.2.	Tabla requisitos funcionales
Fig.3.	Tabla requisitos no funcionales
Fig.4.	Diagrama de casos de uso
Fig.5.	Diseño pantallas con navegación
Fig.6.	Diseño pantalla menú
Fig.7.	Diseño pantalla Login
Fig.8.	Diseño pantalla de carga de la pagina solicitada
Fig.9.	Jerarquía plataforma Android
Fig.10.	LinearLayout horizontal
Fig.11.	LinearLayout vertical
Fig.12.	Relative Layout



Fig.13.	Absolute Layout
Fig.14.	TableLayout
Fig.15.	FrameLayout
Fig.16.	ScrollView
Fig.17.	Estructura y clases del proyecto
Fig.18.	Pantalla Bienvenida
Fig.19.	Pantalla Menú
Fig.20.	Pantalla Login Expediente
Fig.21.	Pantalla Login Laboratorios
Fig.22.	Pasos para llegar al expediente antiguo por la vía normal
Fig.23.	Página del expediente antiguo
Fig.24.	Resultados del expediente antiguo (Login)
Fig.25.	Resultados del expediente antiguo (Selección expediente)
Fig.26.	Resultados del expediente antiguo (Página Expediente)
Fig.27.	Pasos para llegar al expediente nuevo por la vía normal
Fig.28.	Página del expediente nuevo
Fig.29.	Parte interior (frame interior) de la página del expediente nuevo
Fig.30.	Resultados expediente nuevo (Login)
Fig.31.	Resultados expediente nuevo (Selección expediente)
Fig.32.	Resultados expediente nuevo (Página Expediente)
Fig.33.	Resultado de la tabla de notas
Fig.34.	Resultado de la tabla de créditos
Fig.35.	Pasos para entrar en los laboratorios por la vía normal
Fig.36.	Resultados laboratorios (Login)
Fig.37.	Resultados laboratorios (Menú asignaturas)
Fig.38.	Resultados laboratorios (Área privada de una asignatura concreta)
Fig.39.	Maqueta Pizarra (Login)
Fig.40.	Maqueta Pizarra (Menú asignaturas profesores)

Fig.41.	Maqueta Pizarra (Formulario noticia)
Fig.42.	Maqueta Pizarra (Noticia Publicada)
Fig.43.	Maqueta Pizarra (Login)
Fig.44.	Maqueta Pizarra (Menú asignaturas profesores)
Fig.45.	Maqueta Pizarra (Listado de noticias)
Fig.46.	Listado noticias
Fig.47.	Listado noticias
Fig.50.	Caso de uso: Uso del resto del móvil
Fig.51.	Caso de uso: Aplicación PACK_UAB
Fig.52.	Caso de uso: Opción Expediente
Fig.53.	Caso de uso: Opción Laboratorios
Fig.54.	Caso de uso: Login
Fig.55.	Caso de uso: Consulta expediente
Fig.56.	Caso de uso: Consulta / Inscripción Laboratorios
Fig.57.	Caso de uso: Opción Pizarra
Fig.58.	Caso de uso: Login Pizarra
Fig.59.	Caso de uso: Publicar noticias
Fig.60.	Caso de uso: Consultar noticias
Fig.61.	Resultados Pantalla Bienvenida – Dispositivo real
Fig.62.	Resultados Pantalla Menu – Dispositivo real
Fig.63.	Resultados expediente (Login) –Dispositivo real
Fig.64.	Resultados expediente (Selección expediente) – Dispositivo real
Fig.65.	Resultados expediente (Página Expediente) – Dispositivo real
Fig.66.	Resultado de la tabla de notas – Dispositivo real
Fig.67.	Resultado de la tabla de créditos – Dispositivo real
Fig.68.	Resultados laboratorios (Login) – Dispositivo real
Fig.69.	Resultados laboratorios (Menú asignaturas) – Dispositivo real
Fig.70.	Resultados laboratorios (Menú asignaturas) – Dispositivo real



Fig.71.	Resultados laboratorios (Área privada de una asignatura concreta) – Dispositivo real
Fig.72.	Maqueta Pizarra (Login) – Dispositivo real
Fig.73.	Maqueta Pizarra (Menú asignaturas profesores) – Dispositivo real
Fig.74.	Maqueta Pizarra (Formulario Noticias) – Dispositivo real
Fig.75.	Maqueta Pizarra (Noticias publicadas) – Dispositivo real
Fig.76.	Maqueta Pizarra (Menú asignaturas alumnos) – Dispositivo real
Fig.77.	Maqueta Pizarra (Listado de Noticias) – Dispositivo real

**Abstracto**

Mi proyecto es una aplicación Android llamada PACK\_UAB que permite a los estudiantes consultar diferentes páginas de nuestra universidad, como el expediente académico y la página de los laboratorios para la consulta o inscripción a los grupos de prácticas. Estas páginas están parseadas para que solamente aparezca la información importante para el estudiante y además están adaptadas a la pantalla del teléfono. Este proyecto es una solución para los estudiantes porque pueden hacer estas tareas desde cualquier sitio con su teléfono Android si disponen de conexión a Internet y solamente pulsando un botón.

**Abstracta**

El meu projecte es una aplicació Android anomenada PACK\_UAB que permet als estudiants consultar diferents pàgines de la nostra universitat, com l'expedient acadèmic i la pàgina dels laboratoris per consultar o inscriure's als grups de pràctiques. Aquestes pàgines estan parsejades perquè únicament aparegui la informació important per l'estudiant i a més estan adaptades a la pantalla del telèfon. Aquest projecte es una solució pels estudiants perquè poden fer aquestes tasques des de qualsevol lloc amb el seu telèfon Android si disposen de connexió a Internet i únicament prement un botó.

**Abstract**

My project is an Android application called PACK\_UAB that allows consult the different pages of our university as the page of academic marks and the page of groups for practices. This pages are parsed only with the important information for de student and also are adapted to the screen of mobile. This project is one solution for the students because they can do this tasks anywhere with their Android telephone if they have internet and only touching a button.