UNIVERSIDAD NACIONAL DE INGENIERIA

FACULTAD DE CIENCIAS

PROGRAMACION EN DISPOSITIVOS MOVILES

CIENCIAS DE LA COMPUTACION

PROYECTO

MMPP - Intranet

Integrantes:

Augusto Pecho Chávez (*)

Carlos Munaylla Ciprián

Felipe Moreno Vera (*)

Kevin Polo Ruiz

Marzo 2016

API EXTRA

*Slash Screen de Bienvenida

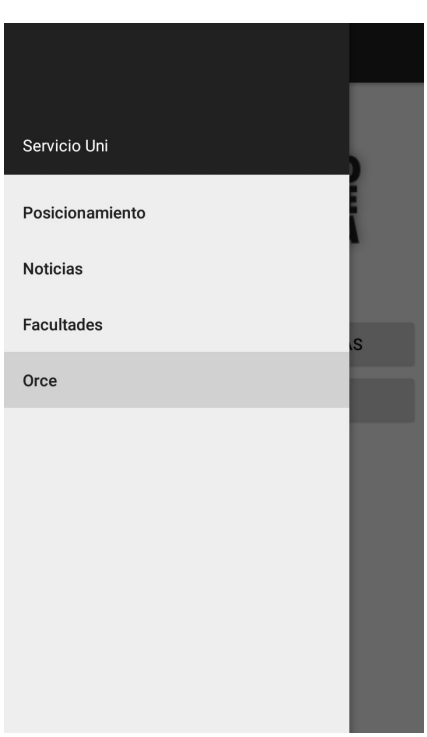
En esta pantalla se muestra una animación de bienvenida. El layout a utilizar es el RelativeLayout el cual contendrá un textView para mostrar el mensaje de “bienvenidos” y un ImageView que será el logo en el medio que va a rotar. Para ello se creó una animación “rotate.xml” en el cual incluimos el tiempo de duración, el grado que va a girar, etc. Una vez que la animación termine nos mandará a la pantalla de bienvenida.

*Aplicacion General (Activity)

Al ingresar a la aplicación tras la correspondiente pantalla de bienvenida nos encontramos en la aplicación. Esta aplicación cuenta con DrawerLayout para tener compatibilidad con NavigationView.
Dado que en nuestra aplicación queremos evitar la pérdida del NavigationView en nuestra aplicación, esta activity contara con un CoodinatorLayout y RelativeLayout, especialmente dicho RelativeLayout se usara para la manipulación de la aplicación mediante el manejo de fragments.

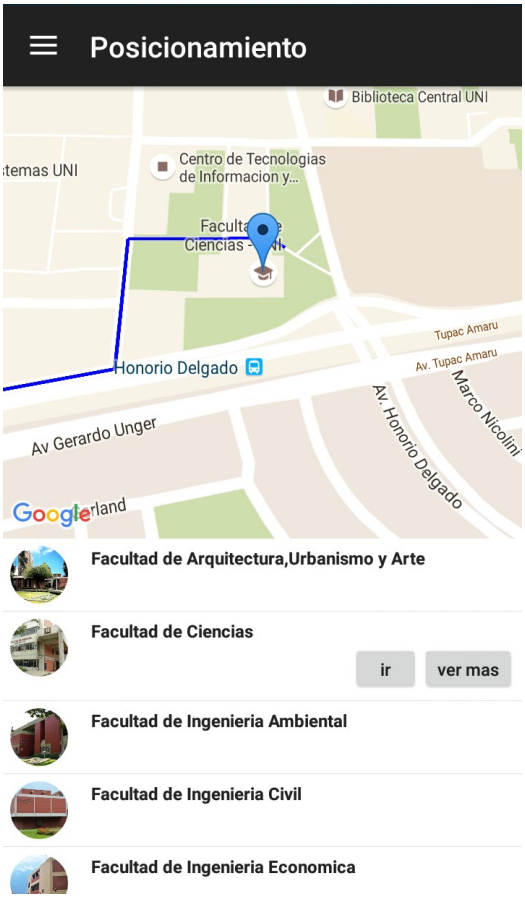
*Bienvenida

El primer Fragment que se nos muestra será el de bienvenida. Un CoordinatorLayout para el manejo de AppBarLayout y LinearLayout.
Dentro de este LinearLayout se insertara una Imagen y cuatro botones los cuales contaran con evento que cambiaran el fragment a los fragments que se comentaran más adelante.



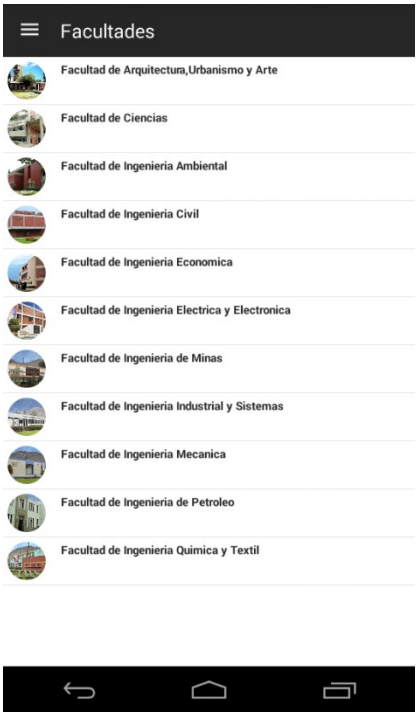
*Posicionamiento

En este fragment nos encontraremos con un CoordinatorLayout que tendrá un AppBarLayout y un LinearLayout que contara con un pequeño layout que es donde se cargará SupportMapFragment para visualizar un mapa,y un ListView que mostrará facultades de la Universidad Nacional de Ingenieria.
Este ListView contara con opciones IR, que mostrará la ruta a caminar para llegar al edificio escogido en el ListView,esta opcion solo se mostrar si se encontró su posición mediante GPS, y la opción Ver Más que cambiará el fragment que contiene la información de dicha facultad.



*Facultades

En la pantalla de facultades se muestra el listado de todas las facultades de la UNI.
En esta vista se utilizó un CoordinatorLayout para poder realizar interacciones entre views, acá lo usaremos para crear interacciones entre la AppBar y views con scrolling. Dentro de ella se incluye un LinearLayout, el cual contendrá un FrameLayout y un ListView para mostrar el listado de las facultades. Dentro del coordinatorLayout tenemos a un AppBarLayout que siempre debe envolver a la Toolbar y los demás componentes pertenecientes a ella.



Para poder ver el listado de las facultades, se utilizó un RelativeLayout, en cual se incluyó un LinearLayout, dentro del cual se colocó un CircleImageView, que permite ver las imágenes de cada facultad en círculo, como se muestra en el pantallazo anterior; además dentro de otro LinearLayout, se incluyó un textView para mostrar el nombre de cada facultad.

Facultades (información de cada facultad)

En esta pantalla se muestra la información de la facultad seleccionada anteriormente. En la vista se incluye un CoordinatorLayout, dentro de ella se hace uso del NestedScrollView y un AppBarLayout. Dentro del primero, se incluye un LinearLayout, en el cual vamos a incluir varios CardView. El primero será para mostrar la información de la facultad y su nombre en un TextView (estos incluidos dentro de un LinearLayout); el segundo será para mostrar las distintas escuelas profesionales de cada facultad. Cabe mencionar que dentro de ese LinearLayout se permitirá mostrar la imagen y el nombre de la respectiva escuela; además al presionar en dicha escuela, esta nos llevará a otra pantalla para mostrar un poco la información de dicha escuela. Por último, en el tercer CardView, será para mostrar contactos y enlaces, en el cual se muestran los teléfonos de dicha facultad y un Button que al presionarlo, nos llevará a la página web de la respectiva facultad y obtener mayor información de dicha facultad.

Por otro lado, también se incluye un AppBarLayout en esta vista, en la cual se implementa el CollapsingToolbarLayout, un Layout especial que envuelve a la Toolbar para controlar las reacciones de expansión y contracción de los elementos que se encuentran dentro del AppBarLayout, como es en este caso, es decir, se refiere a la variación de las dimensiones y desapariciones de la escena (la imagen de cada facultad). Dentro de ese CollapsingToolBarLayout vamos a tener un ImageView, que en este caso sería el de la facultad, y también tendremos un Toolbar.

Escuelas Profesionales

En esta pantalla veremos una breve información de las distintas escuelas profesionales de la UNI.

Para la vista se hace uso de un RelativeLayout en el cual vamos a incluir un CoordinatorLayout, que ya se explicó anteriormente su funcionalidad. Dentro de él se incluye un LinearLayout, dentro de él otro LinearLayout y dentro de este se incluirá un ImageView para colocar la imagen respectiva para cada escuela. Para ver la información de dicha escuela, se hace uso de un TextView en un LinearLayout, quedando la vista como se verá.

*Noticias

Para realizar la sección de noticias se hizo uso de un lector RSS, que son archivos xml. Además en esta sección, se mostrará primero un splash screen o lo que vendría a ser una pantalla de bienvenida, con un spinner y un background como fondo de pantalla; luego a partir del feed devuelto por el analizador, mostrará la imagen en miniatura, asi como el título y la fecha en un custom List View; y finalmente al hacer click en el “list ítem”, se mostrará la descripción en una actividad diferente. Para llevar a cabo el análisis de HTML a través de la descripción que se obtendrá después de analizar el archivo xml, se utiliza una biblioteca externa llamada ‘jsoup’, además se usan algunas clases externas tales como: ‘FileCache.java’, ‘ImageLoader.java’, MemoryCache.java’ y ‘Utils.java’; que será para cargar las imágenes en el ListView.

Splash Sreen

En el xml, se usó un RelativeLayout para mostrar una imagen y cargar el spinner con un texto. Dentro de ese RelativeLayout se encuentra un ImageView, un LinearLayout que contendrá un ProgressBar y un TextView para mostrar el mensaje de cargando.

Antes de comenzar con el proceso de análisis, es decir, cargar el rss, se necesita comprobar la conectividad a internet. Si estamos conectados, entonces procederemos con el análisis y si no hay conectividad, mostramos un alert para comprobar la conectividad.
El análisis se realiza en un hilo separado, esto lo logramos mediante el uso de la clase 'AsyncTask'. La razón por la que se usa es ver si se inicia el análisis sintáctico en el hilo principal o en el hilo de la interfaz de usuario. Asi se crea la clase "AsyncLoadXMLFeed" que extiende de AsyncTask, que tiene los métodos doInBackground(), en donde se inicia el proceso de análisis y una vez hecho se va al metodo onPostExecute(), en el que podemos realizar la acción que se tiene que hacer después del análisis, mostrar la lista de actividades.

Lista de Noticias

-Antes de pasar al análisis sintáctico, se crean dos clases llamadas 'RSSItem', que contiene las variables de cadena en donde se almacenan los datos necesarios de cada elemento en el XML. También tenemos algunos setters y getters para establecer y obtener los datos de la página (noticias de la uni); y la otra clase 'RSSFeed' que contiene todos los elementos de RSSItem que se han analizado; esta clase contiene un método llamado getItemCount() para obtener el número total de elementos (es decir, las noticias).
Luego se crea una clase analizadora que tomará la URL y realizará el análisis sintáctico, esta clase se llama 'DOMParser', que será para

analizar los datos XML, se crea un método del tipo RSSFeed que tendrá como parámetro un string.

EL XML
Para mostrar los elementos de la fuente (del rss noticias), se creó una lista personalizada con cada elemento de la lista que contiene una imagen, título y fecha. Para realizar dicha lista se crearon los siguientes XML: “lista_noticia.xml” que tendrá un CoordinatorLayout, dentro de ella se define un LinearLayout, el cual a su vez tendrá un FrameLayout y un ListView para mostrar las noticias. Después de ello, se muestra un AppBarLayout que incluye a un Toolbar. “vista_noticia.xml” con un RelativeLayout en donde se incluyen un CircleImageView para ver la imagen en círculo en el listado de noticias, un RelativeLayout, en donde se encuentran dos textView: uno para el título de la noticia y el otro para la fecha. Finalmente se colocó un ImageView en el cual se puso una flecha a la derecha de todo, como se podrá observar en el pantallazo.

-A partir de ello se creó la clase ListActivity, en donde en la clase ‘CustomListAdapter’ que extiende de BaseAdapter, se utilizó la clase ‘ImageLoader’, que es externa, junto con otras clases de apoyo. En el onItemClickListener(), se pasa la información con la posición de cada elemento y al hacer click pasamos a ver la noticia detalladamente .

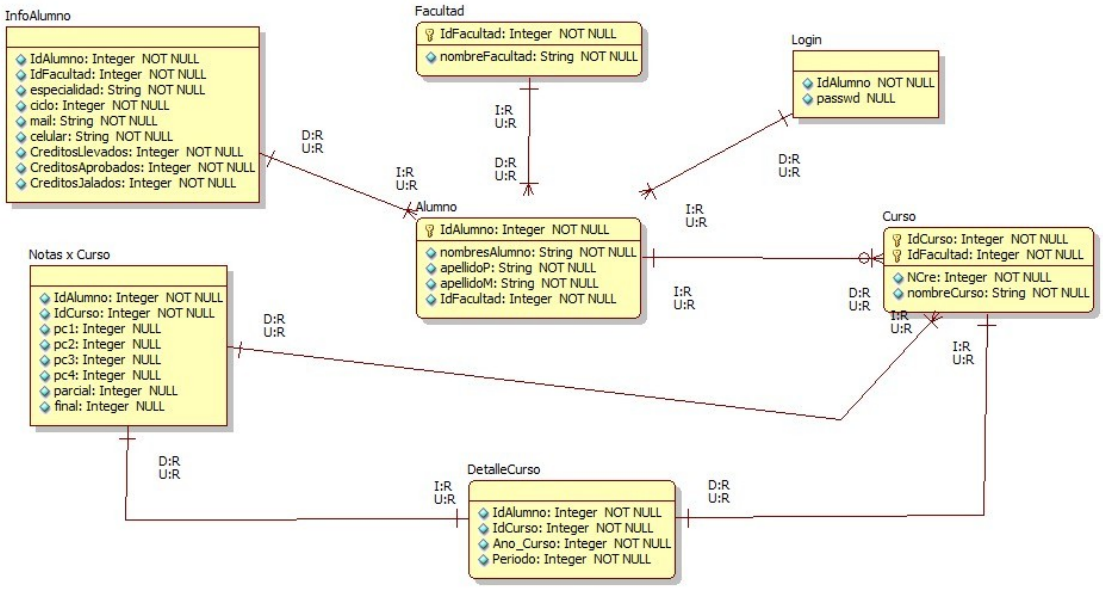
Detalle de Noticias

Para obtener la vista de cada noticia, lo primero es crear un “detail.xml”, que será un RelativeLayout el cual contendrá a un ViewPager, que nos va a permitir desplazar cada noticia de derecha a izquierda y viceversa, en vez de regresar y estar seleccionando la noticia que querramos. Este ViewPager lleva un adaptador de donde se parte de todos los puntos de vista. Lo segundo será crear un “detail_fragment.xml”, el cual será un fragment, en donde se mantiene la informacion detallada. En ese detail_fragment tenemos un ScrollView el cual contendrá un WebView, que nos permitirá mostrar la página web de la noticia seleccionada.

En un principio hubieron algunos problemas para mostrar la noticia detalladamente, ya que ésta se mostraba en tamaño grande y la visibilidad era dificultosa, pero luego se hizo solucionó dicho problema haciendo uso del método ‘loadUrl()’ que pudo permitir “jalar” la información de la página web de las noticias.

Login con la base de datos virtual

Al inicio de la aplicación nos muestra un menú principal en un Navigation View al lado izquierdo y a su vez unos 4 botones con funccionalidades similares a los que se encuentran en el Navigation View.
En primer lugar antes de tocar este tema, hablemos de la Base de Datos.



Se tuvo que modificar muestra base de datos, pues teníamos un problema para almacenar más variables conforme aparecía la necesidad, quedando este como diseño final.

Los Querys se presentan en el documento Anexo_Querys.docx.

Se describen las tablas a continuación:

a) *Alumno*: Esta tabla contiene el id del alumno, sus nombres y apellidos y el id de la facultad a la que pertenece.

b) *InfoAlumno*: Aquí se guarda la información personal del alumno: id del alumno, id de la facultad del alumno, especialidad, ciclo, mail, celular y la cantidad de créditos llevados, aprobados y desaprobados.

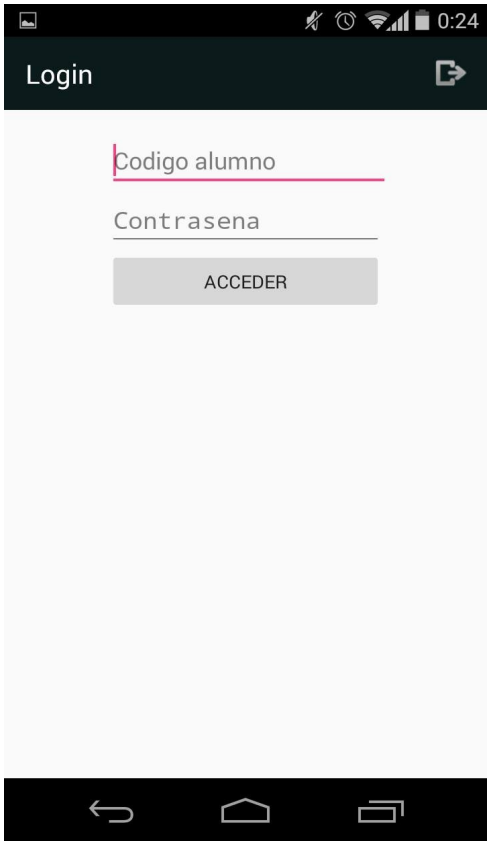
c) *Facultad*: En esta tabla se guarda un id que identifica a cada facultad de la universidad.

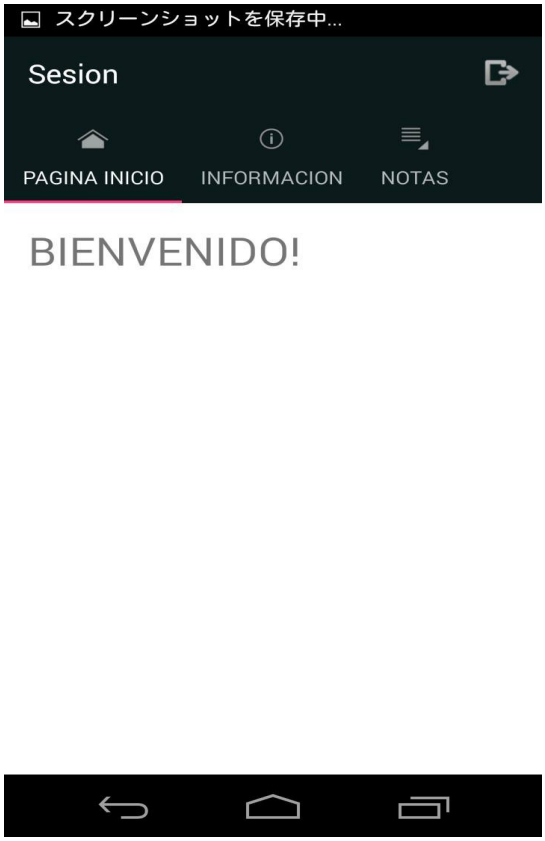
d) *Curso*: Almacena el nombre de los cursos disponibles por facultad y la cantidad de créditos que les corresponde. Cada curso tiene un id único y un id de facultad relacionado.

e) *DetalleCurso (D_Curso)*: Contiene la información del ciclo y periodo de los cursos que llevó un alumno. Se identifica con el id del alumno y el id de los cursos que le corresponden.

f) *NotasCurso*: Guarda la información de las prácticas correspondientes a cada curso (pc1, pc2, pc3, pc4, examen parcial y examen final). Se relaciona mediante el id del alumno y el id del curso.

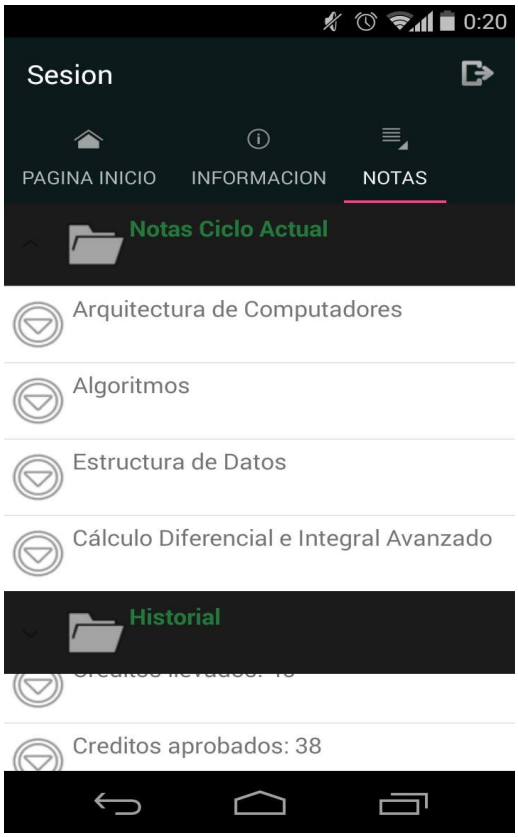
El login consta de 2 activitys, una del login y otra de la sesion





Para hacer conexión de nuestro login que recibe parámetros como código alumno y contraseña, usamos la base de datos creada en el servidor web hostinger, que administra la base de datos con PhpMyAdmin. Mediante una clase llamada ConnexionTask que extiende de AsynTask, es la que se conecta al servidor mediante la url y el método (en este caso, de tipo POST) el cual contiene un script en php que se conecta a la base de datos y obtiene data mediante un query, y dicha data es retornada a la aplicación en formato de un string JSON, la cual podemos desencriptar mediante las clases JSONArray o JSONObject. Y así, verificamos si el login es correcto o no.
Una vez se encuentra que es correcto, abre la sesión. La cual muestra 3 fragments que son manipulados por una clase creada llamada MiFragmentAdapter que extiende de FragmentPageAdapter. El cual solo administra los fragments por pestaña mediante su función Fragment getItem, muy aparte se tiene 3 fragments que son supervisados por nuestro MiFragmentAdapter, HomeFragment, InfoFragment y GradeFragment, donde HomeFragment muestra un mensaje de bienvenida.
Una vez ingresada sesión y durante el OnCreateView, desde la activity Login se hace un Intent hacia la activity Sesion, pasando el dato de ID_Alumno que servirá para hacer consultas explicadas a continuación:

InfoFragment, es el que una vez ingresado a la sesión, hace una nueva consulta al servidor obteniendo información del alumno según el código de alumno, obteniendo nombre, apellidos, ciclo, etc ... , también obtiene su foto y según el ciclo que tiene, se presenta un progressbar coloreando su progreso.



Y también el GradeFragment hace otra consulta también usando el código alumno, obteniendo información sobre créditos llevados, aprobados y jalados hasta ahora, junto con los cursos llevados actualmente y también una lista sobre los periodos académicos cursados.
Todos los casos de Login y los fragments de InfoFragment y GradeFragment de la activity Sesion, usan la clase ConnexionTask que es llamada como AsynTask, para que no detenga ni haga más lento el funcionamiento de la app.

EQUIPO DE TRABAJO

El equipo de trabajo se distribuyó de la manera siguiente:
Responsables del desarrollo de la aplicación Servicio Uni:

- Kevin Polo Ruiz.
- Carlos Munaylla Ciprián.

Responsables del desarrollo de la intranet móvil:

- Augusto Pecho Chávez.
- Felipe Moreno Vera.

Link del Proyecto

https://github.com/Jenazad/PDM/tree/master/Proyecto_Moviles

En este link, se encuentra hasta la versión anterior a la integración de ambas partes.