

Universidad Autónoma Metropolitana

División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

Reporte de Proyecto Terminal

Sistema de gestión escolar en Android[®]

Sampere Romero Edgar Rodrigo
207300380

Becerril Rodríguez Edgar
208300171

130
Diciembre 2013

Asesor: Risto F. Rangel Kuoppa
Profesor Titular "B" - Tiempo Completo
No. Económico: 27499

Indice

I.	Introducción	3
II.	Resumen	3
III.	Planteamiento	4
	• Objetivos Específicos	
	• Objetivos Generales	
IV.	Justificación	4
V.	Desarrollo del proyecto	5
	• Pantalla de Inicio	
	• Menú Principal	
	• Avisos	
	• Calendario	
	• Horario	
	• Información Académica	
	• Información Personal	
	• Kardex	
	• Seriación	
VI.	Seguimiento a futuro	16
VII.	Bibliografía	17
VIII.	Conclusiones	17
IX.	Apéndice	18

Introducción

La manera de consultar información, tanto académica como personal, dentro de la Universidad, se maneja a través del Sistema de Administración Escolar (SAE), el cual también sirve para que los alumnos puedan reinscribir sus UEA's o renunciar a ellas.

Sin embargo, con el crecimiento de las tecnologías móviles y el alcance que han tenido los dispositivos de móviles, ahora es viable ingresar a este tipo de plataformas Web desde cualquier lado; esto a su vez, representa un replanteamiento de los esquemas como los conocemos de las tecnologías de la información, para que puedan ser utilizables de manera óptima y eficaz en comparación con la idea primera que se tenía sobre estos nuevos dispositivos cuando empezaron a ser comerciales.

El proyecto realizado implementa un Sistema basado en la plataforma en línea SIIUAM (Sistema de Integración e Información UAM) de forma que los alumnos de la institución puedan acceder a dicho sistema aprovechando eficientemente las tecnologías actuales.

Resumen

El presente documento trata sobre la forma en que se desarrollo el Sistema de gestión escolar, las tecnologías que se utilizaron para su realización y un pequeño tutorial de cómo utilizar cada módulo dentro de la aplicación

Dado que el objetivo final de la realización de este proyecto es conceder al alumno una manera más sencilla de interactuar con el sistema escolar y con su trayectoria académica, se realizó esta aplicación pensando en la facilidad de uso de forma que el usuario pudiese consultar cualquier información que necesitase tanto desde dentro como fuera de la Universidad.

Así bien, en el apartado de apéndice se presentan las secciones de código mas relevantes que conforman tanto la aplicación en Android (formato apk) como del servidor Web y del servidor de la base de datos, en donde reside la información de los alumnos a consultar.

Dado que el sistema, como proyecto, tiene aún varios aspectos que pudiesen ser mejorados, o bien, módulos que pudieran ser incorporados, se expone en el apartado de "Seguimiento a futuro" la forma en cómo creemos pertinente que se manejen estos *a posteriori*.

Planteamiento

✓ Objetivos Generales.-

Diseñar e implementar una aplicación que maneje y automatice las funciones del Sistema de Administración Escolar de la UAM para teléfonos inteligentes con Sistema Operativo Android.

✓ Objetivos Específicos.-

- Diseñar e implementar un módulo para la visualización y sincronización del calendario escolar, los avisos de la universidad y la distribución de matrículas de forma dinámica
- Diseñar e implementar un módulo para la visualización de calificaciones del alumno
- Diseñar e implementar un módulo para la consulta de horarios de las UEA's y el horario personal del alumno
- Diseñar e implementar un módulo para la consulta de datos académicos y personales del alumno
- Diseñar e implementar un módulo para la generación de la línea de captura
- Diseñar e implementar un módulo para inscripción
- Diseñar e implementar un módulo para renunciaciones
- Diseñar e implementar la interfaz que utilizará los módulos anteriores

Justificación

En vista del alcance que los dispositivos móviles han tenido en la vida cotidiana y como una forma de ayudar a las actividades tanto académicas como personales (a nivel alumno), planteamos la posibilidad de facilitar el uso de los trámites escolares usando los teléfonos celulares con Sistema Operativo Android.

La intención de este proyecto es proporcionar otra manera, fácil e intuitiva, de realizar los trámites comunes de un alumno de la UAM, así como facilitar el acceso a sus datos académicos en cualquier ubicación, sin la necesidad de tener una computadora enfrente.

Cabe aclarar que esta interfaz no pretende reemplazar ni replicar en ningún sentido al sistema actual; por el contrario, la idea es un uso conjunto de ambos para mejorar los procesos tanto administrativos como de consulta.

El sistema móvil SAE realizará todas las funciones que el sistema web actual, sumando a sus funciones aquellos cambios presentados en este documento.

Desarrollo del proyecto

El primer enfoque que se planteó para la realización de este proyecto fue la creación de la aplicación móvil de forma que pudiera conectarse de forma directa al sistema existente SIIUAM (Sistema de Integración e Información UAM).

Sin embargo, debido a las políticas de seguridad de la escuela fue imposible llevar a cabo una integración similar debido a que no tuvimos forma de conocer la estructura de dicho sistema, menos aún, de saber cómo realizaba su funcionamiento interno.

Luego, tras un replanteamiento de la idea inicial decidimos cambiar las tecnologías a usar. En seguida se muestra un resumen de lo siguiente:

	Original	Nuevo
Plataforma	Android	Windows Phone
Lenguaje	Java	C#
Entorno de Desarrollo	Eclipse	Visual Studio

Asimismo, para obtener una forma de conectar la aplicación al sistema existente se propuso la idea de obtener el código fuente del sistema, el cual por cierto, es un sitio web. De esta forma y utilizando las tecnologías antes mencionadas elaboramos lo que era el comienzo del proyecto, el primer módulo, el cual se encargaría de obtener el código fuente de la página para que sucesivamente, pudiésemos trabajar con él.

Por motivos de simpleza, ese código se omite en este documento por dos razones principales:

1. El código es muy corto.
2. No tiene relevancia con respecto al producto final.

Ahora bien, tras varios esfuerzos, fútiles cabe decir, no se consiguió obtener un código significativo con el que trabajar. Debido a como está integrado el SIIUAM, lo poco de código que se obtuvo del sistema web estaba ofuscado, es decir, encriptado de forma tal que un usuario externo, en este caso nosotros, no pudiera analizar dicho código de manera sencilla.

Existen diversos procedimientos con los cuáles se puede romper esta seguridad y llegar a la estructura de la página, sin embargo, la finalidad de este proyecto no es "hackear" el SIIUAM, ni mucho menos probar su nivel de seguridad actual, por ende, decidimos cambiar de nuevo de enfoque.

Dado que el sistema a implementar es de tipo cliente/servidor, y como la aplicación a realizar debe conectarse al servidor para obtener la información que necesita para trabajar,

se propuso emular el servidor, es decir, el SIIUAM.

Debido a que el proyecto a realizar era el cliente, en realidad no es necesario conocer a fondo la estructura interna y funcionamiento del sistema web, sólo es necesario tener los datos de entrada para obtener la información de salida.

Utilizando las herramientas originalmente propuestas, aunadas a la idea de crear un servidor para probar el cliente, al final se diseñaron y crearon tanto el servidor como el cliente.

A continuación se presentan las tecnologías utilizadas para cada uno, así como las partes en las que se subdividen:

✓ Servidor de la Base de Datos

- Gestor de Base de Datos MariaDB 5.5.34
- Entorno de trabajo MySQL Workbench 6
- Lenguaje Java y Entorno de Programación Eclipse

✓ Servidor de peticiones Web

- Servidor HTTP Apache Tomcat 7
- Motor de servicios Web Axis2 1.6.2

✓ Cliente Android

- Biblioteca ActionBarSherlock
(<https://github.com/JakeWharton/ActionBarSherlock>)
- Android SDK API 17

En los documentos entregables de este documento se anexa todo lo correspondiente al diagrama de la base de datos utilizada para probar la aplicación, así como también el código fuente del servidor en Java y el código del servidor Web. De igual manera, dado que la finalidad de este proyecto es realizar, por decirle de alguna manera, el SIIUAM móvil, no creemos pertinente explicarlos a detalle dado que como solo es una aproximación, pudiese ser que el diseño propuesto no refleje el verdadero diseño del sistema en funcionamiento.

Dicho esto, se presenta a continuación un pequeño manual sobre la aplicación, así como la descripción de cada uno de los módulos que lo conforman.-

Pantalla de Inicio

En cualquier aplicación, ya sea de escritorio, móvil o web, que trabaje con datos de personas, siempre de existir cierto nivel de seguridad. Es por eso que la primera pantalla que se muestra al abrir la aplicación pide al usuario tanto su matrícula como su contraseña del sistema de forma que se pueda confirmar su existencia dentro del mismo.



Figura 1

Una vez que los datos ingresados se confirman en la Base de Datos del sistema, se puede proceder al menú principal de la aplicación. Sin embargo, si los datos son incorrectos, el usuario es rechazado y se le pide comprobar sus datos e ingresarlos de nuevo.

Menú Principal

Cumpliendo con los objetivos del proyecto, se creó una interfaz intuitiva, sin menús ocultos ni configuraciones adicionales, de forma que el usuario en turno no quedara abrumado con tantas opciones.

Así es que llegamos a la interfaz que abajo se muestra. Consiste de dos partes, una pantalla principal donde se desplegarán los componentes de cada módulo y una barra desplegable del lado izquierdo en donde se muestran los módulos que el usuario puede seleccionar.

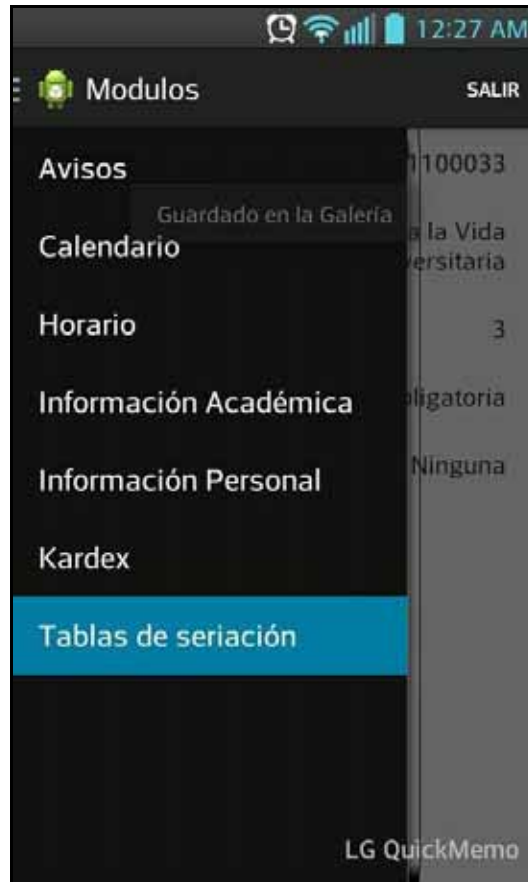


Figura 2

De esta forma basta con que el usuario “jale” la barra desplegable y así puede seleccionar fácilmente la información a consultar.

En la parte derecha superior de la aplicación se diseñó el botón de “Salir”, el cuál, termina con la sesión actual del alumno, desde el módulo en que se encuentre en ese momento, llevándolo de nuevo a la pantalla de inicio.

Avisos

En el módulo de avisos, el alumno puede consultar los avisos pertinentes al trimestre en curso, tales como avisos de Reinscripción, Baja de materias, Inscripción a exámenes de Recuperación, entre otros.

Esto con la intención de que siempre pueda estar informado de eventos importantes correspondientes a su trayectoria trimestral y así la probabilidad de que algún suceso pase desapercibido se disminuya.

Cabe mencionar que cada módulo solo se conecta al servidor al momento de ser seleccionado, de forma que no sature ni a la Base de Datos ni la memoria del teléfono en donde reside la aplicación.

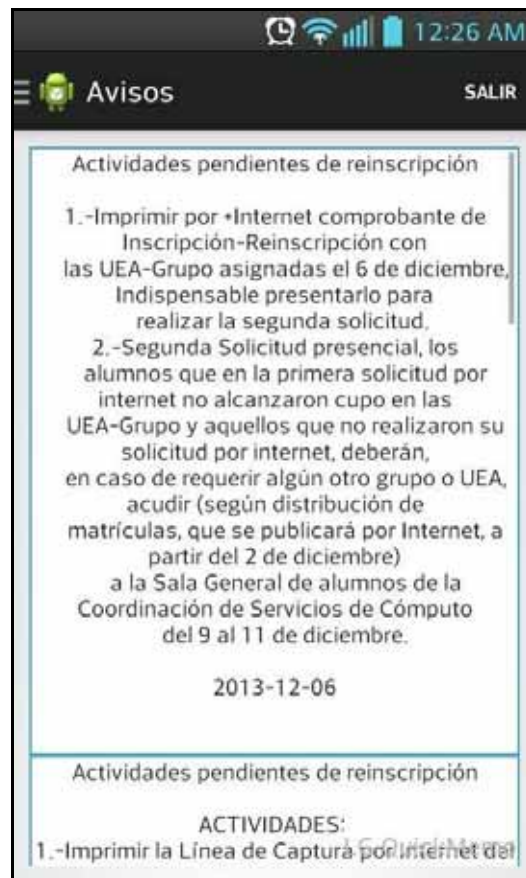


Figura 3

Calendario

Aunque en este módulo la idea original era realizar un calendario dinámico de forma que el alumno pudiese agregar y sincronizar eventos con el calendario de su teléfono, en el producto final no fue posible realizar esta funcionalidad.

Esto debido a que en la actualidad, el sistema no cuenta con esa capacidad *ipso facto*, es decir, la forma en cómo se visualiza el calendario es por medio de un hipervínculo a un archivo pdf con la imagen del calendario escolar.

Debido a que la información de los eventos en el SIUAM no está vinculada de ninguna manera a el calendario, el crear estas condiciones y el diseño que conlleva, hubieran salido dentro del alcance del proyecto actual, sin embargo, podría ser esto un punto a considerar para darle seguimiento, en caso de que se quiera continuar, a la aplicación.

Calendario SALIR

UNIVERSIDAD AUTÓNOMA METROPOLITANA
Casa abierta al tiempo

CALENDARIO ESCOLAR 2013-2014

Trimestre 13-O
Del 26 de agosto al 19 de noviembre de 2013

Trimestre 14-I
Del 6 de enero al 28 de marzo de 2014

Trimestre 14-P
Del 21 de abril al 11 de julio de 2014

AGOSTO 2013

LEN	MAR	MIE	JUE	VIÉ	SAB	DOM	SEM
			1	2	3	4	
5	6	7	8	9	10	11	
12	13	14	15	16●●	17	18	IR/ER
19●●▲	20●●▲	21●▲	22●	23	24	25	IR/ER/EAR
26>	27	28	29	30	31		1

SEPTIEMBRE

LEN	MAR	MIE	JUE	VIÉ	SAB	DOM	SEM
						1	1
2	3	4	5	6	7	8	2
9	10	11	12	13	14	15	3
16	17	18	19	20	21	22	4
23	24	25	26	27	28	29	5

Figura 4

Debido a los motivos previamente expuestos, decidimos ponerlo acorde al sistema actual, es decir, que se pudiera visualizar dentro de la misma aplicación el calendario en formato pdf.

Horario

En este módulo el alumno puede consultar de forma rápida y sencilla el horario correspondiente al trimestre en curso. Dentro de la pantalla principal se visualizará un resumen de las materias divididas en días de la semana donde cada día, al seleccionarlo, muestra en forma de lista las materias correspondientes al día seleccionado, junto con la información pertinente de cada materia, como son: el nombre de la materia, el horario, el salón y el profesor que la imparte.

De forma que se aprovechara el espacio de la pantalla sin tener que estar haciendo acercamientos innecesarios, cada día se despliega de forma que si no se quiere seguir viendo, se puede volver a contraer, de tal forma que solo queden los días de la semana sin mostrar ningún otro dato.

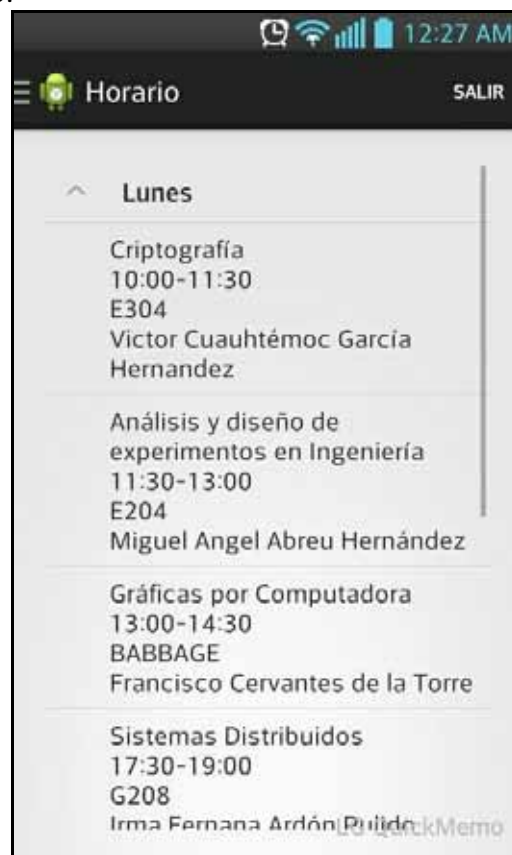


Figura 5

Algo que hay que observar es que se mantiene la plantilla principal dentro de cualquier parte de la aplicación. Se desactivó incluso el posible botón de regresar para evitar que el usuario saliera del flujo de pantallas.


Información Académica

La pantalla de este módulo, como podemos observar abajo, consta de dos subdivisiones. Debido a que ambas pertenecen a la trayectoria académica del alumno, estas se muestran dentro del mismo módulo.

Para desplazarse entre ambas se puede deslizar de izquierda a derecha, o viceversa, de una pantalla a otra; o bien, se puede seleccionar cada subdivisión para acceder a los datos correspondientes.

La pantalla de autorizaciones solo muestra aquellas materias para las cuáles el alumno debe de solicitar un permiso especial para inscribirlas, como es este el caso del Proyecto Terminal, el cuál nos compete en el presente documento.

Dentro de la parte de Datos Académicos el alumno encontrará la información relacionada a su referencia actual con respecto a la carrera.



Datos Academicos	
Unidad:	Azcapotzalco
División:	CBI
Carrera:	Computación
Créditos máximos:	577
Créditos mínimos:	491
Trimestre de ingreso:	070
Último trimestre inscrito:	130
Créditos acumulados:	465
Número de conversiones:	5

Figura 6

Información Personal

Similar al módulo de información académica se encuentra información personal. Dentro de este, el alumno podrá consultar aquella información que la Universidad tiene de él así como también podrá encontrar en la subdivisión de seguro facultativo, su información pertinente al seguro social, si es que cuenta con ella.

Tanto en información personal como en información académica no se le permite al usuario modificar estos datos desde la aplicación debido a que solo la Universidad está capacitada para llevar a cabo esta tarea.



The screenshot shows a mobile application interface for 'Información Personal'. At the top, there is a status bar with the time 12:27 AM and various icons. Below the title bar, there are two tabs: 'DATOS PERSONALES' (selected) and 'SEGURO FACULTATIVO'. The 'Datos Personales' section contains the following information:

Datos Personales	
Nombre:	Edgar Rodrigo Sampere Romero
Nacionalidad:	Mexicano
Fecha:	1988-05-23
CURP:	SARE880523HDFMMD08
Correo:	al20730380@alumnos.azc.uam.mx

The 'Domicilio:' section contains the following information:

Domicilio:	
Calle:	Midas 39
Colonia:	Ensueños
C.P.:	54740
Entidad:	Distrito Federal
Teléfono:	58734073

At the bottom right of the screen, the text 'LG QuickMemo' is visible.

Figura 7

Kardex

Este módulo se utiliza para consultar todas las UEA's(Unidad de Enseñanza Aprendizaje) que el alumno ha cursado a lo largo de su carrera. Aquí se muestra información de cada materia como el nombre, tipo de evaluación (global o recuperación) y la calificación, entre otros.

Para que los datos pudieran estar organizados se implementó la opción de mostrar dichas UEA's agrupadas por trimestre, de forma que, según el trimestre que el alumno elija solo se muestren las materias correspondientes al mismo.

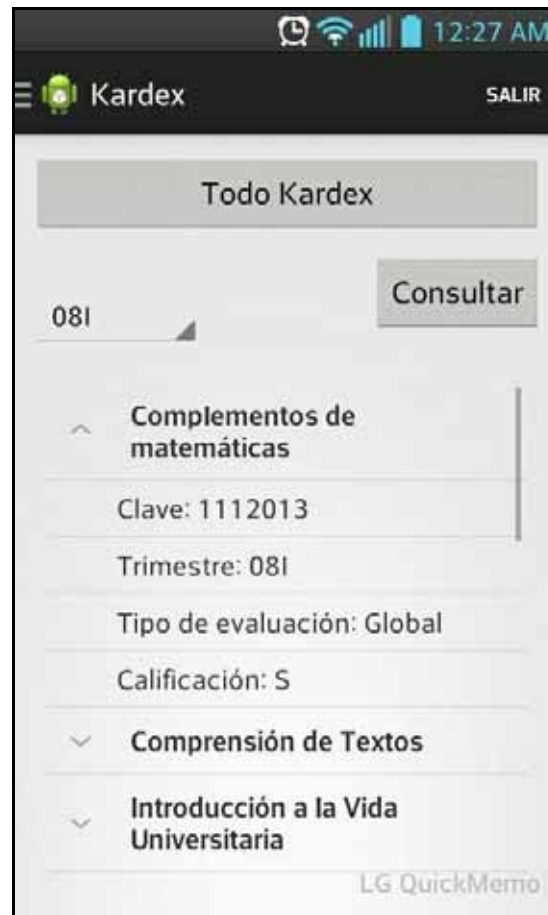


Figura 8

El campo del lado izquierdo donde se muestran los trimestres a seleccionar es un objeto de tipo “menú desplegable”. De esta forma se evita que el usuario pueda introducir un dato incorrecto y así evitar tanto errores como peticiones al servidor innecesarias.

Seriación

Por último tenemos el módulo de seriación, que como el nombre indica, el alumno puede consultar todas las materias que correspondan a sus carreras, tanto aquellas que ha cursado como aquellas que no.

Para poder cursar una materia de un trimestre en específico, hay que cursar las materias previas a esta. Esto se conoce como seriación de materias o dependencias.

En la pantalla de este módulo se muestran las materias que el alumno puede llevar a lo largo de su carrera de forma similar a como se muestra el horario, con la particularidad de que, en lugar de mostrar todos los datos de la materia en la pantalla principal, al seleccionar una materia, esta lleva al usuario a una nueva pantalla donde puede apreciar de manera más concisa los datos de la UEA, así como las materias que necesita cursar para poder inscribirla.

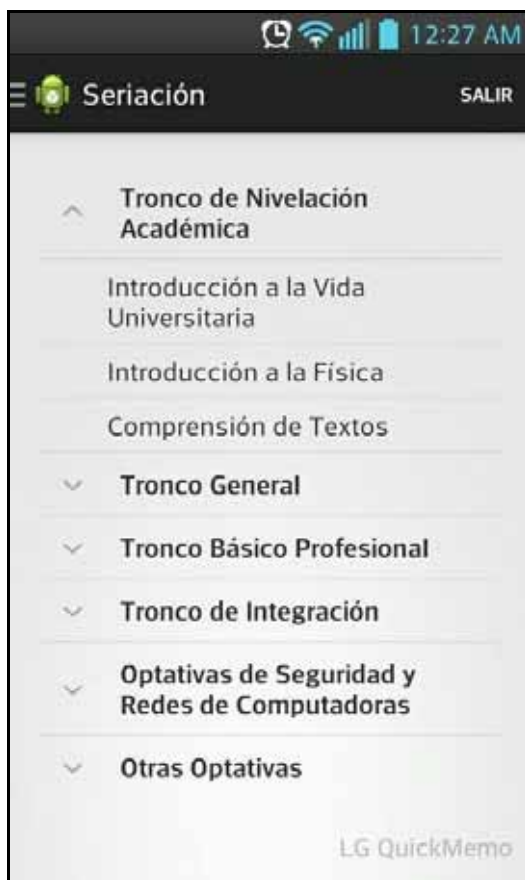


Figura 9

Aquí podemos apreciar cómo las materias aparecen ordenadas según el área a la que pertenezcan. Cuando se selecciona un área, las materias correspondientes se despliegan hacia abajo.

Una vez que selecciona una materia para revisar, se accede a la siguiente pantalla.

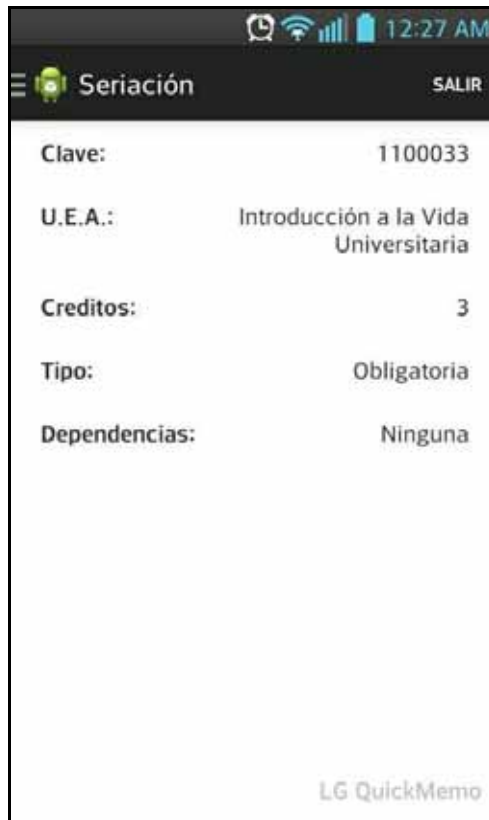


Figura 10

De esta manera se puede visualizar de una mejor manera los datos específicos de una materia. Para hacer la transición entre esta y la pantalla previa se realiza un gesto como el que se lleva haciendo con todos los módulos; sólo es necesario deslizar el dedo de izquierda a derecha para regresar a la pantalla de seriación.

Como puede apreciarse en las figuras anteriores, el sistema está diseñado de tal forma que sea apreciable a una distancia que resulte cómoda entre el usuario y su teléfono, poniendo especial énfasis en la sencillez de uso.

Seguimiento a futuro

En lo que respecta al proyecto y a los objetivos del mismo, quedan por diseñar e implementar dos módulos; el módulo de reinscripción que se encarga de todas las altas, bajas y cambios de UEA's que los alumnos puedan hacer durante el trimestre, y el módulo de renuncia, que inverso del anterior, sirve para darse de baja de una materia inscrita, si el alumno así lo requiere.

Estos módulos no fueron diseñados por la cercanía que mantienen con el sistema original, es decir, para implementarlos sería necesario, ya sea, conocer bien el SIIUAM, o bien, trabajar con el diseño conjunto del sistema desde un inicio.

Por ello, se insta a aquellas personas que quieran dar seguimiento a este proyecto a realizar dichos módulos, así como también, tomar en consideración la implementación de:

- Búsqueda de UEA's por clave
- De ser posible, reestructurar el calendario para poder manejar eventos
- Alarma para el recordatorio de los avisos
- Selección de UEA's dentro del módulo de seriación como propuestas a inscribir

Conclusiones

A pesar de las dificultades que se nos presentaron para la determinación del diseño inicial, y posteriormente, para la consulta de la información, al término del proyecto podemos afirmar que el objetivo principal de este mismo fue alcanzado, si bien, no en su totalidad, de una manera estética y funcionalmente hablando, bastante satisfactoria.

Bibliografía

- [1] Ramírez R. Rodrigo, “Aplicación móvil para la sincronización de archivos”, proyecto terminal, División de CBI, Universidad Autónoma Metropolitana Azcapotzalco, México, 2010
- [2] Ortigosa Flores, José Luis, “Software colaborativo para dispositivos móviles con comunicación Bluetooth” , proyecto terminal, División de CBI, Universidad Autónoma Metropolitana Azcapotzalco, México, 2011
- [3] Velázquez Canales, Alberto Manuel, “Aplicación web de administración de horarios para estudiantes” , proyecto terminal, División de CBI, Universidad Autónoma Metropolitana Azcapotzalco, México, 2007
- [4] Universidad Nacional Autónoma de México, En línea, Disponible: <http://asesorias.cuautitlan2.unam.mx/idiomas/inscripciones/>
- [5] Instituto Politécnico Nacional, En línea, Disponible: www.saes.esimez.ipn.mx
- [6] James Steele, Nelson To, “The Android Developer’s Cookbook”, Ed. Addison-Wesley, 2010
- [7] Clive Sargeant(2011, Agosto 10). “Android, User Interface made easy”, Disponible: http://www.amazon.com/Android-User-Interface-made-ebook/dp/B005GVEH1G/ref=sr_1_3?s=digital-text&ie=UTF8&qid=1331187481&sr=1-3
- [8] Sayed Y.Hashimi, Satya Komatineni, “Pro Android”, Ed. Apress, 2009

Apéndice

✓ Login.java

```
package com.pt.main;

import com.actionbarsherlock.app.SherlockFragmentActivity;
import com.example.pt_fragment.R;
import com.pt.tools.WebService;

import android.content.Context;
import android.content.Intent;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;
import android.widget.Toast;

public class Login extends SherlockFragmentActivity
{
    boolean correcto = false;
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
    }

    public void entrar(View v)
    {
        EditText etUsuario = (EditText)findViewById(R.id.Cuenta);
        String usuario = etUsuario.getText().toString();
        etUsuario.setText("");

        EditText etPassword =
        (EditText)findViewById(R.id.Contrasenia);
        String password = etPassword.getText().toString();
        etPassword.setText("");

        if(usuario.isEmpty() && password.isEmpty())
        {
            mostrarMensaje("Ingreso sus datos");
        }
        else
        {
            AsyncCallL task = new AsyncCallL();
            task.execute(usuario, password);
        }
    }

    public void mostrarMensaje(String mensaje)
    {
        Context context = getApplicationContext();
    }
}
```

```

        int duracion = Toast.LENGTH_SHORT;
        Toast.makeText(context, mensaje, duracion).show();
    }

    private class AsyncCallL extends AsyncTask<String, Void, Void> {
        @Override
        protected Void doInBackground(String... params) {
            correcto =
WebService.invocaServicioWebL("consultaLoginAlumno", "Login",
params[0],params[1]);
            return null;
        }

        @Override
        protected void onPostExecute(Void result)
        {
            if(correcto)
            {
                Intent i = new Intent(getApplicationContext(),
SAE.class);
                startActivity(i);
            }
            else
                mostrarMensaje("Cuenta o constraseña incorrecta");
        }

        @Override
        protected void onPreExecute() {
        }

        @Override
        protected void onProgressUpdate(Void... values) {
        }
    }
}

```

✓ SAE.java

```

package com.pt.main;

import android.content.Intent;
import android.content.res.Configuration;
import android.os.Bundle;
import android.support.v4.app.ActionBarDrawerToggle;
import android.support.v4.app.FragmentTransaction;
import android.support.v4.view.GravityCompat;
import android.support.v4.widget.DrawerLayout;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;

import com.actionbarsherlock.app.ActionBar;
import com.actionbarsherlock.app.SherlockFragment;

```

```

import com.actionbarsherlock.app.SherlockFragmentActivity;
import com.actionbarsherlock.view.Menu;
import com.actionbarsherlock.view.MenuInflater;
import com.actionbarsherlock.view.MenuItem;
import com.example.pt_fragment.R;
import com.pt.modulos.*;

public class SAE extends SherlockFragmentActivity
{
    private String[] Modulos;
    private DrawerLayout myDrawerLayout;
    private ListView myDrawerList;
    private ActionBarDrawerToggle myDrawerToggle;
    private String moduloActual = new String();

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sae);
        init();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu)
    {
        MenuInflater inflater = getSupportMenuInflater();
        inflater.inflate(R.menu.logout, menu);
        return true;
    }

    @Override
    public void onBackPressed()
    {
        if(getSupportFragmentManager().getBackStackEntryCount() > 0)
            super.onBackPressed();
    }

    private void init()
    {
        Modulos = getResources().getStringArray(R.array.modulos);
        myDrawerLayout =
(DrawerLayout)findViewById(R.id.drawer_layout);
        myDrawerList = (ListView)findViewById(R.id.left_drawer);
        myDrawerList.setAdapter(new ArrayAdapter<String>(this,
R.drawable.drawer_list_item, Modulos));
        ActionBar actionBar = getSupportActionBar();
        AvisosFragment fragment = new AvisosFragment();
        FragmentTransaction ft =
getSupportFragmentManager().beginTransaction();

        myDrawerToggle = new ActionBarDrawerToggle(this,
myDrawerLayout,
R.drawable.ic_drawer, R.string.drawer_open,
R.string.drawer_close)
        {
            public void onDrawerClosed(View view)
            {

```

```

        if(moduloActual.equals("Información Personal") ||
moduloActual.equals("Información Académica"))

        getActionBar().setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);
        getActionBar().setTitle(moduloActual);
    }

    public void onDrawerOpened(View drawerView)
    {
        if(moduloActual.equals("Información Personal") ||
moduloActual.equals("Información Académica"))

        getActionBar().setNavigationMode(ActionBar.DISPLAY_SHOW_TITLE);
        getActionBar().setTitle("Modulos");
    }
};

myDrawerLayout.setDrawerListener(myDrawerToggle);
myDrawerLayout.setDrawerShadow(R.drawable.drawer_shadow,
GravityCompat.START);
myDrawerList.setOnItemClickListener(new
DrawerItemClickListener());

        actionBar.setNavigationMode(ActionBar.DISPLAY_SHOW_TITLE);
        actionBar.setTitle("Avisos");
        moduloActual = "Avisos";

        actionBar.setDisplayHomeAsUpEnabled(true);
        actionBar.setHomeButtonEnabled(true);

        ft.replace(R.id.contenedor_modulos, fragment);
        ft.commit();
    }

    public void selectItem(int position)
    {
        FragmentTransaction ft =
getSupportFragmentManager().beginTransaction();
        SherlockFragment fragment;
        ActionBar actionbar = getSupportActionBar();

        myDrawerList.setItemChecked(position, true);
        myDrawerLayout.closeDrawer(myDrawerList);

        switch(position)
        {
            case 0:
                actionbar.setTitle("Avisos");

                moduloActual = "Avisos";

                if(getSupportFragmentManager().findFragmentById(R.id.contenedor_mod
ulos).getClass() != AvisosFragment.class)
                {
                    actionbar.removeAllTabs();
                    fragment = new AvisosFragment();
                    break;
                }
            }
    }

```

```

        }
        else
            return;
    case 1:
        getSupportActionBar().setTitle("Calendario");
        moduloActual = "Calendario";

        if (getSupportFragmentManager().findFragmentById(R.id.contenedor_modulos).getClass() != CalendarioFragment.class)
        {
            actionBar.removeAllTabs();
            fragment = new CalendarioFragment();
            break;
        }
        else
            return;
    case 2:
        getSupportActionBar().setTitle("Horario");
        moduloActual = "Horario";

        if (getSupportFragmentManager().findFragmentById(R.id.contenedor_modulos).getClass() != HorarioFragment.class)
        {
            actionBar.removeAllTabs();
            fragment = new HorarioFragment();
            break;
        }
        else
            return;
    case 3:
        getSupportActionBar().setTitle("Información Académica");
        moduloActual = "Información Académica";

        if (getSupportFragmentManager().findFragmentById(R.id.contenedor_modulos).getClass() != InformacionAcademicaFragment.class)
        {
            actionBar.removeAllTabs();
            fragment = new InformacionAcademicaFragment();
            break;
        }
        else
        {
            getActionBar().setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);
            return;
        }
    case 4:
        getSupportActionBar().setTitle("Información Personal");
        moduloActual = "Información Personal";

        if (getSupportFragmentManager().findFragmentById(R.id.contenedor_modulos).getClass() != InformacionPersonalFragment.class)
        {
            actionBar.removeAllTabs();
            fragment = new InformacionPersonalFragment();
            break;
        }

```

```

        }
        else
        {

getActionBar().setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);
            return;
        }
        case 5:
            getSupportActionBar().setTitle("Kardex");
            moduloActual = "Kardex";

            if(getSupportFragmentManager().findFragmentById(R.id.contenedor_modulos).getClass() != KardexFragment.class)
            {
                actionbar.removeAllTabs();
                fragment = new KardexFragment();
                break;
            }
            else
                return;
        default:
            getSupportActionBar().setTitle("Seriación");
            moduloActual = "Seriación";

            if(getSupportFragmentManager().findFragmentById(R.id.contenedor_modulos).getClass() != SeriacionFragment.class)
            {
                actionbar.removeAllTabs();
                fragment = new SeriacionFragment();
                break;
            }
            else
                return;
        }
        ft.replace(R.id.contenedor_modulos, fragment);
        ft.commit();
    }

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

    @Override
    public boolean onOptionsItemSelected(MenuItem item)
    {
        switch (item.getItemId())
        {
            case R.id.logout:
                Intent intent;
                intent = new Intent(getApplicationContext(),
Login.class);

                intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
                startActivity(intent);
                break;
            case android.R.id.home:

```

```

        if (myDrawerLayout.isDrawerOpen(myDrawerList))
            myDrawerLayout.closeDrawer(myDrawerList);
        else
            myDrawerLayout.openDrawer(myDrawerList);
    }
    return super.onOptionsItemSelected(item);
}

@Override
public void onConfigurationChanged(Configuration newConfig)
{
    super.onConfigurationChanged(newConfig);
    myDrawerToggle.onConfigurationChanged(newConfig);
}

private class DrawerItemClickListener implements
ListView.OnItemClickListener
{
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int
position,
                            long id)
    {
        selectItem(position);
    }
}
}

```

✓ AvisosFragment.java

```

package com.pt.modulos;

import java.util.ArrayList;

import android.content.Context;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.view.ViewGroup.LayoutParams;
import android.widget.TextView;

import com.example.pt_fragment.*;
import com.pt.tools.WebService;

import com.actionbarsherlock.app.SherlockFragment;

public class AvisosFragment extends SherlockFragment
{
    private ArrayList<String> resultado;
    private Context context;
}

```



```

private ViewGroup layout;

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup
container,
    Bundle savedInstanceState)
{
    View layoutView = inflater.inflate(R.layout.fragment_avisos,
container, false);
    layout = (ViewGroup)layoutView.findViewById(R.id.AvisosLayout);
    context = layoutView.getContext();

    AsyncCallA task = new AsyncCallA();
    task.execute();

    return layoutView;
}

private class AsyncCallA extends AsyncTask<String, Void, Void> {
    @Override
    protected Void doInBackground(String... params) {
        resultado =
WebService.invocaServicioWeb("consultaAviso", "Avisos", "207300380");
        return null;
    }

    @Override
    protected void onPostExecute(Void result) {
        int t = resultado.size()/3;
        int j = 0;
        for(int i = 0;i<t;i++)
        {
            String aux = "";
            aux += resultado.get(j)+"\n\n";
            aux += resultado.get(++j)+"\n\n";
            aux += resultado.get(++j)+"\n\n";
            j++;
            TextView tv = new TextView(context);
            tv.setLayoutParams(new
LayoutParams(LayoutParams.WRAP_CONTENT, LayoutParams.WRAP_CONTENT));
            tv.setSingleLine(false);
            tv.setGravity(Gravity.CENTER_HORIZONTAL);
            tv.setBackgroundResource(R.drawable.borde);
            tv.setText(aux);
            layout.addView(tv);
        }
    }

    @Override
    protected void onPreExecute() {
    }

    @Override
    protected void onProgressUpdate(Void... values) {
    }
}

```

```
}  
}
```

✓ CalendarioFragment.java

```
package com.pt.modulos;  
  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.webkit.WebView;  
  
import com.example.pt_fragment.*;  
  
import com.actionbarsherlock.app.SherlockFragment;  
  
public class CalendarioFragment extends SherlockFragment  
{  
    WebView myWebView;  
  
    @Override  
    public View onCreateView(LayoutInflater inflater, ViewGroup  
container,  
        Bundle savedInstanceState)  
    {  
        View layoutView = inflater.inflate(R.layout.fragment_calendario,  
container, false);  
  
        myWebView =  
(WebView)layoutView.findViewById(R.id.CalendarioEscolar);  
        myWebView.loadUrl("http://www.uam.mx/calendario/index.html");  
        myWebView.getSettings().setBuiltInZoomControls(true);  
        return layoutView;  
    }  
}
```

✓ HorarioFragment.java

```
package com.pt.modulos;  
  
import java.util.ArrayList;  
import java.util.HashMap;  
import java.util.List;  
  
import android.content.Context;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.view.LayoutInflater;  
import android.view.View;  
import android.view.ViewGroup;  
import android.widget.ExpandableListView;  
  
import com.example.pt_fragment.*;
```

```

import com.pt.tools.ExpandableListAdapter;
import com.pt.tools.WebService;

import com.actionbarsherlock.app.SherlockFragment;

public class HorarioFragment extends SherlockFragment
{
    private ExpandableListView exLv;
    private ExpandableListAdapter exLAdapter;
    private List<String> titulos;
    private HashMap<String, List<String>> hijos;
    private ArrayList<String> resultado;
    private Context context;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
        Bundle savedInstanceState)
    {
        View layoutView = inflater.inflate(R.layout.fragment_horario,
container, false);

        exLv =
(ExpandableListView)layoutView.findViewById(R.id.Horarios);
        titulos = new ArrayList<String>();
        hijos = new HashMap<String, List<String>>();
        context = layoutView.getContext();

        titulos.add("Lunes");
        titulos.add("Martes");
        titulos.add("Miercoles");
        titulos.add("Jueves");
        titulos.add("Viernes");

        AsyncCallH task = new AsyncCallH();
        task.execute();

        return layoutView;
    }

    private class AsyncCallH extends AsyncTask<String, Void, Void> {
        @Override
        protected Void doInBackground(String... params) {
            resultado = WebService.invocaServicioWeb("consultaHorario",
"Horario", "207300380");
            return null;
        }

        @Override
        protected void onPostExecute(Void result)
        {
            int t = resultado.size()/5;
            int j = 0;
            List<String> aux = new ArrayList<String>();
            List<String> aux2 = new ArrayList<String>();
            for(int i = 0;i<t;i++)
            {

```

```

        aux.add(resultado.get(j+1)+"\n"+resultado.get(j+2)+"\n"+resultado.g
et(j+3)+"\n"
                +resultado.get(j+4));

        switch(Integer.parseInt(resultado.get(j)))
        {
        case 1:
            hijos.put(titulos.get(0), aux);
            hijos.put(titulos.get(2), aux);
            hijos.put(titulos.get(4), aux);
            break;
        case 2:
            hijos.put(titulos.get(1), aux);
            hijos.put(titulos.get(3), aux);
            break;
        case 3:
            hijos.put(titulos.get(0), aux);
            hijos.put(titulos.get(2), aux);
            hijos.put(titulos.get(3), aux);
            break;
        case 4:
            hijos.put(titulos.get(1), aux);
            hijos.put(titulos.get(3), aux);
            hijos.put(titulos.get(4), aux);
            break;
        default:
            hijos.put(titulos.get(0), aux);
            hijos.put(titulos.get(3), aux);
            break;
        }
        j += 5;
    }
    for(int i=0;i<titulos.size();i++)
        if(hijos.get(titulos.get(i)) == null)
            hijos.put(titulos.get(i), aux2);

    exLAdapter = new ExpandableListAdapter(context, titulos,
hijos);
    exLv.setAdapter(exLAdapter);
}

@Override
protected void onPreExecute() {
}

@Override
protected void onProgressUpdate(Void... values) {
}
}
}

```

✓ InformacionAcademicaFragment.java

```
package com.pt.modulos;

import android.os.Bundle;
import android.support.v4.view.ViewPager;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import com.example.pt_fragment.*;
import com.pt.informacion_academica.Autorizaciones;
import com.pt.informacion_academica.DatosAcademicos;
import com.pt.tools.MyAdapter;

import com.actionbarsherlock.app.ActionBar;
import com.actionbarsherlock.app.SherlockFragment;

public class InformacionAcademicaFragment extends SherlockFragment
{
    private ViewPager myViewPager;
    private MyAdapter myAdapter;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
        Bundle savedInstanceState)
    {
        View layoutView =
inflater.inflate(R.layout.fragment_informacion_academica, container,
false);
        init(layoutView);
        return layoutView;
    }

    public void init(View v)
    {
        myViewPager =
(ViewPager)v.findViewById(R.id.pager_informacion_academica);
        ActionBar actionBar =
getSherlockActivity().getSupportActionBar();

        actionBar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);

        myAdapter = new MyAdapter(getSherlockActivity(), myViewPager);
        myAdapter.addTab(actionBar.newTab().setText("Datos Academicos"),
DatosAcademicos.class, null);
        myAdapter.addTab(actionBar.newTab().setText("Autorizaciones"),
Autorizaciones.class, null);
    }
}
```

✓ InformacionPersonalFragment.java

```
package com.pt.modulos;

import android.os.Bundle;
import android.support.v4.view.ViewPager;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import com.example.pt_fragment.*;
import com.pt.informacion_personal.*;
import com.pt.tools.*;

import com.actionbarsherlock.app.ActionBar;
import com.actionbarsherlock.app.SherlockFragment;

public class InformacionPersonalFragment extends SherlockFragment
{
    private ViewPager myViewPager;
    private MyAdapter myAdapter;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
        Bundle savedInstanceState)
    {
        View layoutView =
inflater.inflate(R.layout.fragment_informacion_personal, container,
false);
        init(layoutView);
        return layoutView;
    }

    public void init(View v)
    {
        myViewPager =
(ViewPager)v.findViewById(R.id.pager_informacion_personal);
        ActionBar actionBar =
getSherlockActivity().getSupportActionBar();

        actionBar.setNavigationMode(ActionBar.NAVIGATION_MODE_TABS);

        myAdapter = new MyAdapter(getSherlockActivity(), myViewPager);
        myAdapter.addTab(actionBar.newTab().setText("Datos
Personales"), DatosPersonales.class, null);
        myAdapter.addTab(actionBar.newTab().setText("Seguro
Facultativo"), SeguroFacultativo.class, null);
    }
}
```

✓ KardexFragment.java

```
package com.pt.modulos;
```

```

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

import android.content.Context;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ExpandableListView;
import android.widget.Spinner;

import com.example.pt_fragment.*;
import com.pt.tools.ExpandableListAdapter;
import com.pt.tools.WebService;

import com.actionbarsherlock.app.SherlockFragment;

public class KardexFragment extends SherlockFragment
{
    private ExpandableListView exLv;
    private ExpandableListAdapter exLAdapter;

    private List<String> trimestres;
    private List<String> titulos;
    private ArrayList<String> resultado;
    private HashMap<String, List<String>> hijos;

    private Spinner trim;
    private Context context;

    private boolean inicio = false;
    private boolean todoKardex = false;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
        Bundle savedInstanceState)
    {
        View layoutView = inflater.inflate(R.layout.fragment_kardex,
container, false);

        exLv =
(ExpandableListView)layoutView.findViewById(R.id.Calificaciones);
        trim = (Spinner)layoutView.findViewById(R.id.KTrimestre);
        titulos = new ArrayList<String>();
        hijos = new HashMap<String, List<String>>();
        context = layoutView.getContext();

        Button bTodoKardex =
(Button)layoutView.findViewById(R.id.KardexCompleto);
        bTodoKardex.setOnClickListener(todoListener);

```

```

        Button bTrimKardex =
(Button)layoutView.findViewById(R.id.BTrimestre);
        bTrimKardex.setOnClickListener(trimListener);

        AsyncCallK task = new AsyncCallK();
            task.execute("trimestres");

        //mostrarTodoKardex();

        return layoutView;
    }

    private OnClickListener todoListener = new OnClickListener()
    {
        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            mostrarTodoKardex();
        }
    };

    private OnClickListener trimListener = new OnClickListener()
    {
        @Override
        public void onClick(View v) {
            // TODO Auto-generated method stub
            mostrarKardex();
        }
    };

    public void llenaTrimestres()
    {
        ArrayAdapter<String> adapter = new
ArrayAdapter<String>(context, android.R.layout.simple_spinner_item, trimest
res);

        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dro
pdown_item);
        trim.setAdapter(adapter);
    }

    public void mostrarTodoKardex()
    {
        if(!todoKardex)
        {
            AsyncCallK task = new AsyncCallK();
            task.execute("todo");
            todoKardex = true;
        }
    }

    public void mostrarKardex()
    {
        AsyncCallK task = new AsyncCallK();
        task.execute(trim.getSelectedItem().toString());
        todoKardex = false;
    }
}

```



```

private class AsyncCallK extends AsyncTask<String, Void, Void> {
    @Override
    protected Void doInBackground(String... params) {
        if(params[0].equalsIgnoreCase("trimestres"))
            trimestres =
WebService.invocaServicioWeb("consultaTrimestres", "Kardex");
        else if(params[0].equalsIgnoreCase("todo"))
            resultado =
WebService.invocaServicioWeb("consultaKardexCompleto", "Kardex",
"207300380");
        else
            resultado =
WebService.invocaServicioWeb("consultaKardexTrimestral", "Kardex",
"207300380", params[0].toString());
        return null;
    }

    @Override
    protected void onPostExecute(Void result) {
        if(!inicio)
        {
            llenaTrimestres();
            inicio = true;
        }
        else
        {
            if(!titulos.isEmpty())
                titulos.clear();
            if(!hijos.isEmpty())
                titulos.clear();

            int t = resultado.size()/5;
            int j = 0;
            for(int i = 0;i<t;i++)
            {
                titulos.add(resultado.get(j+1));
                List<String> aux = new ArrayList<String>();
                aux.add("Clave: "+resultado.get(j));
                aux.add("Trimestre: "+resultado.get(j+2));
                if(resultado.get(j+3).equals("1"))
                    aux.add("Tipo de evaluación: Global");
                else
                    aux.add("Tipo de evaluación: Recuperación");
                aux.add("Calificación: "+resultado.get(j+4));

                hijos.put(titulos.get(i), aux);
                j += 5;
            }

            exLAdapter = new ExpandableListAdapter(context, titulos,
hijos);
            exLv.setAdapter(exLAdapter);
        }
    }

    @Override

```

```

        protected void onPreExecute() {
        }

        @Override
        protected void onProgressUpdate(Void... values) {
        }
    }
}

```

✓ SeriacionFragment.java

```

package com.pt.modulos;

import java.util.ArrayList;
import java.util.HashMap;
import java.util.List;

import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentTransaction;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ExpandableListView;
import android.widget.ExpandableListView.OnChildClickListener;

import com.example.pt_fragment.*;
import com.pt.tools.DetailFragment;
import com.pt.tools.ExpandableListAdapter;
import com.pt.tools.WebService;

import com.actionbarsherlock.app.SherlockFragment;

public class SeriacionFragment extends SherlockFragment
{
    private ExpandableListView exLv;
    private ExpandableListAdapter exLAdapter;
    private List<String> titulos;
    private HashMap<String, List<String>> hijos;

    private ArrayList<String> resultado;
    private List<String> TNA;
    private List<String> TG;
    private List<String> TBP;
    private List<String> TI;
    private List<String> OSRC;
    private List<String> OO;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container,
        Bundle savedInstanceState)
    {

```

```

View layoutView = inflater.inflate(R.layout.fragment_seriacion,
container, false);

exLv = (ExpandableListView)layoutView.findViewById(R.id.Areas);
titulos = new ArrayList<String>();
hijos = new HashMap<String, List<String>>();

TNA = new ArrayList<String>();
TG = new ArrayList<String>();
TBP = new ArrayList<String>();
TI = new ArrayList<String>();
OSRC = new ArrayList<String>();
OO = new ArrayList<String>();

AsyncCalls task = new AsyncCalls();
    task.execute();

hijos.put(titulos.get(0), TNA);
hijos.put(titulos.get(1), TG);
hijos.put(titulos.get(2), TBP);
hijos.put(titulos.get(3), TI);
hijos.put(titulos.get(4), OSRC);
hijos.put(titulos.get(5), OO);

exLAdapter = new ExpandableListAdapter(layoutView.getContext(),
titulos, hijos);
exLv.setOnChildClickListener(new OnChildClickListener() {

    @Override
    public boolean onChildClick(ExpandableListView parent,
View v,
                                int groupPosition, int childPosition, long
id) {

        // TODO Auto-generated method stub
        SherlockFragment newFragment = new
DetailFragment();

        FragmentManager fm =
SeriacionFragment.this.getSherlockActivity().getSupportFragmentManager();
        Bundle b = new Bundle();

        b.putString("UEA",hijos.get(titulos.get(groupPosition)).get(childPo
sition));

        newFragment.setArguments(b);

        FragmentTransaction ft = fm.beginTransaction();
        ft.replace(R.id.contenedor_modulos, newFragment);

        ft.setTransition(FragmentTransaction.TRANSIT_FRAGMENT_FADE);
        ft.addToBackStack(null);
        ft.commit();
        fm.executePendingTransactions();

        return false;
    }
});

```

```

        exLv.setAdapter(exLAdapter);

        return layoutView;
    }

    private class AsyncCalls extends AsyncTask<String, Void, Void> {
        @Override
        protected Void doInBackground(String... params) {
            resultado =
WebService.invocaServicioWeb("consultaSeriacion","Seriacion");
            return null;
        }

        @Override
        protected void onPostExecute(Void result) {
            int t = resultado.size()/2;
            int j = 0;
            for(int i = 0;i<t;i++)
            {
                if(resultado.get(j+1).equalsIgnoreCase("TNA"))
                    TNA.add(resultado.get(j));
                else if(resultado.get(j+1).equalsIgnoreCase("TG"))
                    TG.add(resultado.get(j));
                else if(resultado.get(j+1).equalsIgnoreCase("TBP"))
                    TBP.add(resultado.get(j));
                else if(resultado.get(j+1).equalsIgnoreCase("TI"))
                    TI.add(resultado.get(j));
                else if(resultado.get(j+1).equalsIgnoreCase("OSRC"))
                    OSRC.add(resultado.get(j));
                else
                    OO.add(resultado.get(j));
                j+=2;
            }
        }

        @Override
        protected void onPreExecute() {
            titulos.add("Tronco de Nivelación Académica");
            titulos.add("Tronco General");
            titulos.add("Tronco Básico Profesional");
            titulos.add("Tronco de Integración");
            titulos.add("Optativas de Seguridad y Redes de
Computadoras");
            titulos.add("Otras Optativas");
        }

        @Override
        protected void onProgressUpdate(Void... values) {
        }
    }
}

```

✓ Autorizaciones.java

```

package com.pt.informacion_academica;

import java.util.ArrayList;

import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import com.example.pt_fragment.R;
import com.pt.tools.WebService;

public class Autorizaciones extends Fragment
{
    private ArrayList<String> resultado;
    private TextView Clave;
    private TextView UEA;
    private TextView Trimestre;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState)
    {
        View layoutView =
inflater.inflate(R.layout.fragment_autorizaciones, container, false);

        //layout =
(ViewGroup)layoutView.findViewById(R.id.AutorizacionesLayout);
        //context = layoutView.getContext();

        Clave = (TextView)layoutView.findViewById(R.id.Clave);
        UEA = (TextView)layoutView.findViewById(R.id.NombreUEA);
        Trimestre = (TextView)layoutView.findViewById(R.id.Trimestre);

        AsyncCallAut task = new AsyncCallAut();
        task.execute();

        return layoutView;
    }

    private class AsyncCallAut extends AsyncTask<String, Void, Void> {
        @Override
        protected Void doInBackground(String... params) {
            resultado =
WebService.invocaServicioWeb("consultaAutorizaciones","InformacionAcademi
ca", "207300380");
            return null;
        }

        @Override
        protected void onPostExecute(Void result) {
            Clave.setText(resultado.get(0));
            UEA.setText(resultado.get(1));
            Trimestre.setText(resultado.get(2));
        }
    }
}

```

```

    }
    @Override
    protected void onPreExecute() {
    }

    @Override
    protected void onProgressUpdate(Void... values) {
    }
}
}
}

```

✓ DatosAcademicos.java

```

package com.pt.informacion_academica;

import java.util.ArrayList;

import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import com.example.pt_fragment.R;
import com.pt.tools.WebService;

public class DatosAcademicos extends Fragment
{
    private ArrayList<String> resultado;
    private TextView Unidad;
    private TextView Division;
    private TextView Carrera;
    private TextView CreditosMax;
    private TextView CreditosMin;
    private TextView TrimIngreso;
    private TextView TrimInscrito;
    private TextView CreditosAcumulados;
    private TextView Conversiones;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState)
    {
        View layoutView =
inflater.inflate(R.layout.fragment_datos_academicos, container, false);

        Unidad = (TextView) layoutView.findViewById(R.id.Unidad);
        Division = (TextView) layoutView.findViewById(R.id.Division);
        Carrera = (TextView) layoutView.findViewById(R.id.Carrera);
        CreditosMax = (TextView)
layoutView.findViewById(R.id.CreditosMax);

```

```

        CreditosMin = (TextView)
layoutView.findViewById(R.id.CreditosMin);
        TrimIngreso = (TextView)
layoutView.findViewById(R.id.TrimestreIngreso);
        TrimInscrito = (TextView)
layoutView.findViewById(R.id.TrimestreInscrito);
        CreditosAcumulados = (TextView)
layoutView.findViewById(R.id.CreditosAcumulados);
        Conversiones = (TextView)
layoutView.findViewById(R.id.Conversiones);

        AsyncCallDA task = new AsyncCallDA();
        task.execute();

        return layoutView;
    }

    private class AsyncCallDA extends AsyncTask<String, Void, Void> {
        @Override
        protected Void doInBackground(String... params) {
            resultado =
WebService.invocaServicioWeb("consultaDatosAcademicos", "InformacionAcadem
ica", "207300380");
            return null;
        }

        @Override
        protected void onPostExecute(Void result) {
            Unidad.setText(resultado.get(0));
            Division.setText(resultado.get(1));
            Carrera.setText(resultado.get(2));
            CreditosMax.setText(resultado.get(3));
            CreditosMin.setText(resultado.get(4));
            TrimIngreso.setText(resultado.get(5));
            TrimInscrito.setText(resultado.get(6));
            CreditosAcumulados.setText(resultado.get(7));
            Conversiones.setText(resultado.get(8));
        }

        @Override
        protected void onPreExecute() {
        }

        @Override
        protected void onProgressUpdate(Void... values) {
        }
    }
}

```

✓ DatosPersonales.java

```

package com.pt.informacion_personal;

import java.util.ArrayList;

```

```

import com.example.pt_fragment.*;
import com.pt.tools.WebService;

import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

public class DatosPersonales extends Fragment
{
    private ArrayList<String> resultado;
    private TextView Nombre;
    private TextView Nacionalidad;
    private TextView Fecha;
    private TextView CURP;
    private TextView Correo;
    private TextView Calle;
    private TextView Colonia;
    private TextView CP;
    private TextView Entidad;
    private TextView Telefono;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
        Bundle savedInstanceState) {
        View layoutView =
inflater.inflate(R.layout.fragment_datos_personales, container, false);

        Nombre = (TextView) layoutView.findViewById(R.id.Nombre);
        Nacionalidad = (TextView)
layoutView.findViewById(R.id.Nacionalidad);
        Fecha = (TextView) layoutView.findViewById(R.id.Fecha);
        CURP = (TextView) layoutView.findViewById(R.id.CURP);
        Correo = (TextView) layoutView.findViewById(R.id.Correo);
        Calle = (TextView) layoutView.findViewById(R.id.Calle);
        Colonia = (TextView) layoutView.findViewById(R.id.Colonia);
        CP = (TextView) layoutView.findViewById(R.id.Codigo);
        Entidad = (TextView) layoutView.findViewById(R.id.Entidad);
        Telefono = (TextView) layoutView.findViewById(R.id.Telefono);

        AsyncCallDP task = new AsyncCallDP();
        task.execute();
        return layoutView;
    }

    private class AsyncCallDP extends AsyncTask<String, Void, Void> {
        @Override
        protected Void doInBackground(String... params) {
            resultado =
WebService.invocaServicioWeb("consultaDatosPersonales", "InformacionPerson
al", "207300380");
            return null;
        }
    }
}

```



```

@Override
protected void onPostExecute(Void result) {
    Nombre.setText(resultado.get(0));
    Nacionalidad.setText(resultado.get(1));
    Fecha.setText(resultado.get(2));
    CURP.setText(resultado.get(3));
    Correo.setText(resultado.get(4));
    Calle.setText(resultado.get(5));
    Colonia.setText(resultado.get(6));
    CP.setText(resultado.get(7));
    Entidad.setText(resultado.get(8));
    Telefono.setText(resultado.get(9));
}
@Override
protected void onPreExecute() {
}

@Override
protected void onProgressUpdate(Void... values) {
}
}
}

```

✓ SeguroFacultativo.java

```

package com.pt.informacion_personal;

import java.util.ArrayList;

import com.example.pt_fragment.*;
import com.pt.tools.WebService;

import android.os.AsyncTask;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

public class SeguroFacultativo extends Fragment
{
    private ArrayList<String> resultado;
    private TextView Numero;
    private TextView Fecha;
    private TextView Nombre;
    private TextView Calle;
    private TextView Colonia;
    private TextView CP;
    private TextView Entidad;
    private TextView Telefono;

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,

```

```

        Bundle savedInstanceState) {
            View layoutView =
inflater.inflate(R.layout.fragment_seguro_facultativo, container, false);

            Numero = (TextView) layoutView.findViewById(R.id.Numero);
            Fecha = (TextView) layoutView.findViewById(R.id.Fecha);
            Nombre = (TextView) layoutView.findViewById(R.id.Nombre);
            Calle = (TextView) layoutView.findViewById(R.id.Calle);
            Colonia = (TextView) layoutView.findViewById(R.id.Colonia);
            CP = (TextView) layoutView.findViewById(R.id.Codigo);
            Entidad = (TextView) layoutView.findViewById(R.id.Entidad);
            Telefono = (TextView) layoutView.findViewById(R.id.Telefono);

            AsyncCallDP task = new AsyncCallDP();
                task.execute();
            return layoutView;
        }

private class AsyncCallDP extends AsyncTask<String, Void, Void> {
    @Override
    protected Void doInBackground(String... params) {
        resultado =
WebService.invocaServicioWeb("consultaSeguroFacultativo","InformacionPers
onal","207300380");
        return null;
    }

    @Override
    protected void onPostExecute(Void result) {
        Numero.setText(resultado.get(0));
        Fecha.setText(resultado.get(1));
        Nombre.setText(resultado.get(2));
        Calle.setText(resultado.get(3));
        Colonia.setText(resultado.get(4));
        CP.setText(resultado.get(5));
        Entidad.setText(resultado.get(6));
        Telefono.setText(resultado.get(7));
    }

    @Override
    protected void onPreExecute() {
    }

    @Override
    protected void onProgressUpdate(Void... values) {
    }
}
}
}

```

✓ DetailFragment.java

```
package com.pt.tools;
```

```

import java.util.ArrayList;

import android.os.AsyncTask;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.TextView;

import com.actionbarsherlock.app.SherlockFragment;
import com.example.pt_fragment.R;

public class DetailFragment extends SherlockFragment
{
    private ArrayList<String> resultado;
    private String Busqueda;
    TextView Clave;
    TextView UEA;
    TextView Creditos;
    TextView Tipo;
    TextView Dependencias;

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

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup
container, Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.detail_fragment,
container, false);
        Bundle b = getArguments();

        Busqueda = b.getString("UEA");
        Clave = (TextView)view.findViewById(R.id.SClaveUEA);
        Clave.setText(Busqueda);
        UEA = (TextView)view.findViewById(R.id.SNombreUEA);
        Creditos = (TextView)view.findViewById(R.id.SCreditos);
        Tipo = (TextView)view.findViewById(R.id.STipo);
        Dependencias =
(TextView)view.findViewById(R.id.SDependencias);

        AsyncCalls task = new AsyncCalls();
        task.execute();

        return view;
    }

    private class AsyncCalls extends AsyncTask<String, Void, Void> {
        @Override
        protected Void doInBackground(String... params) {
            resultado =
WebService.invocaServicioWebS("consultaSeriacionUEA","Seriacion",
Busqueda);
            return null;
        }
    }
}

```

```

    }

    @Override
    protected void onPostExecute(Void result) {
        Clave.setText(resultado.get(0));
        UEA.setText(Busqueda);
        Creditos.setText(resultado.get(1));

        if(resultado.get(2).equalsIgnoreCase("1"))
            Tipo.setText("Obligatoria");
        else
            Tipo.setText("Optativa");

        if(resultado.get(3).equalsIgnoreCase("null") &&
resultado.get(4).equalsIgnoreCase("error ocurred"))
            Dependencias.setText("Ninguna");
        else if(resultado.get(4).equalsIgnoreCase("null"))
            Dependencias.setText(resultado.get(3));
        else
            Dependencias.setText(resultado.get(3)+" y
"+resultado.get(4));
    }
    @Override
    protected void onPreExecute() {
    }

    @Override
    protected void onProgressUpdate(Void... values) {
    }

}
}

```

✓ ExpandableListAdapter.java

```

package com.pt.tools;

import java.util.HashMap;
import java.util.List;

import com.example.pt_fragment.R;

import android.content.Context;
import android.graphics.Typeface;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.BaseExpandableListAdapter;
import android.widget.TextView;

public class ExpandableListAdapter extends BaseExpandableListAdapter {

    private Context context;
    private List<String> titulos;

```

```

private HashMap<String, List<String>> hijos;

public ExpandableListAdapter(Context context, List<String>
listaTitulos, HashMap<String, List<String>> listaHijos)
{
    this.context = context;
    this.titulos = listaTitulos;
    this.hijos = listaHijos;
}

@Override
public Object getChild(int groupPosition, int childPosition) {
    // TODO Auto-generated method stub
    return this.hijos.get(this.titulos.get(groupPosition))
        .get(childPosition);
}

@Override
public long getChildId(int groupPosition, int childPosition) {
    // TODO Auto-generated method stub
    return childPosition;
}

@Override
public View getChildView(int groupPosition, int childPosition,
boolean isLastChild, View convertView, ViewGroup
parent) {
    // TODO Auto-generated method stub
    final String childText = (String) getChild(groupPosition,
childPosition);

    if (convertView == null) {
        LayoutInflater infalInflater = (LayoutInflater) this.context
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        convertView = infalInflater.inflate(R.layout.list_item,
null);
    }

    TextView txtListChild = (TextView) convertView
        .findViewById(R.id.exLObjeto);

    txtListChild.setText(childText);
    return convertView;
}

@Override
public int getChildrenCount(int groupPosition) {
    // TODO Auto-generated method stub
    return this.hijos.get(this.titulos.get(groupPosition))
        .size();
}

@Override
public Object getGroup(int groupPosition) {
    // TODO Auto-generated method stub
    return this.titulos.get(groupPosition);
}

```

```

@Override
public int getGroupCount() {
    // TODO Auto-generated method stub
    return this.titulos.size();
}

@Override
public long getGroupId(int groupPosition) {
    // TODO Auto-generated method stub
    return groupPosition;
}

@Override
public View getGroupView(int groupPosition, boolean isExpanded,
    View convertView, ViewGroup parent) {
    // TODO Auto-generated method stub
    String headerTitle = (String) getGroup(groupPosition);
    if (convertView == null) {
        LayoutInflater infalInflater = (LayoutInflater) this.context
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        convertView = infalInflater.inflate(R.layout.list_group,
null);
    }

    TextView lblListHeader = (TextView) convertView
        .findViewById(R.id.exLTitulo);
    lblListHeader.setTypeface(null, Typeface.BOLD);
    lblListHeader.setText(headerTitle);

    return convertView;
}

@Override
public boolean hasStableIds() {
    // TODO Auto-generated method stub
    return false;
}

@Override
public boolean isChildSelectable(int groupPosition, int
childPosition) {
    // TODO Auto-generated method stub
    return true;
}
}

```

✓ MyAdapter.java

```

package com.pt.tools;

import java.util.ArrayList;

import com.actionbarsherlock.app.ActionBar;

```

```

import com.actionbarsherlock.app.ActionBar.Tab;
import com.actionbarsherlock.app.SherlockFragmentActivity;

import android.content.Context;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;
import android.support.v4.app.FragmentTransaction;
import android.support.v4.view.ViewPager;

public class MyAdapter extends FragmentPagerAdapter implements
ActionBar.TabListener , ViewPager.OnPageChangeListener
{
    private final Context myContext;
    private final ActionBar myActionBar;
    private final ViewPager myViewPager;
    private final ArrayList<TabInfo> myTabs = new ArrayList<TabInfo>();

    static final class TabInfo
    {
        private final Class<?> clss;
        private final Bundle args;

        TabInfo(Class<?> clss, Bundle args)
        {
            this.clss = clss;
            this.args = args;
        }
    }

    public MyAdapter(SherlockFragmentActivity fa, ViewPager pager)
    {
        super(fa.getSupportFragmentManager());
        myContext = fa;
        myActionBar = fa.getSupportActionBar();
        myViewPager = pager;
        myViewPager.setAdapter(this);
        myViewPager.setOnPageChangeListener(this);
    }

    public void addTab(ActionBar.Tab tab, Class<?> clss, Bundle args)
    {
        TabInfo info = new TabInfo(clss, args);
        tab.setTag(info);
        tab.setTabListener(this);
        myTabs.add(info);
        myActionBar.addTab(tab);
        notifyDataSetChanged();
    }

    @Override
    public Fragment getItem(int position)
    {
        TabInfo info = myTabs.get(position);
        return Fragment.instantiate(myContext, info.clss.getName(),
info.args);
    }
}

```

```

@Override
public int getCount()
{
    return myTabs.size();
}

@Override
public void onPageScrollStateChanged(int arg0) {
    // TODO Auto-generated method stub
}

@Override
public void onPageScrolled(int arg0, float arg1, int arg2) {
    // TODO Auto-generated method stub
}

@Override
public void onPageSelected(int position) {
    // TODO Auto-generated method stub
    myActionBar.setSelectedNavigationItem(position);
}

@Override
public void onTabSelected(Tab tab, FragmentTransaction ft) {
    // TODO Auto-generated method stub
    myViewPager.setCurrentItem(tab.getPosition());
    Object tag = tab.getTag();
    for(int i = 0; i < myTabs.size(); i++)
        if(myTabs.get(i) == tag)
            myViewPager.setCurrentItem(i);
}

@Override
public void onTabUnselected(Tab tab, FragmentTransaction ft) {
    // TODO Auto-generated method stub
}

@Override
public void onTabReselected(Tab tab, FragmentTransaction ft) {
    // TODO Auto-generated method stub
}
}

```

✓ Webservice.java

```

package com.pt.tools;
import java.util.ArrayList;

import org.ksoap2.SoapEnvelope;

```



```

import org.ksoap2.serialization.PropertyInfo;
import org.ksoap2.serialization.SoapObject;
import org.ksoap2.serialization.SoapPrimitive;
import org.ksoap2.serialization.SoapSerializationEnvelope;
import org.ksoap2.transport.HttpTransportSE;

public class Webservice {

    private static String NAMESPACE = "http://ws.apache.org/axis2";
    private static String URL =
"http://192.168.43.125:8080/axis2/services/";
    //private static String SOAP_ACTION =
"http://ws.apache.org/axis2/consultaDatosPersonales";
    //private static String METHOD_NAME = "consultaDatosPersonales";

    public static ArrayList<String> invocaServicioWeb(String METHOD_NAME,
String SERVICE_NAME)
    {
        ArrayList<String> resTxt = new ArrayList<String>();
        // Create request
        SoapObject request = new SoapObject(NAMESPACE, METHOD_NAME);

        // Create envelope
        SoapSerializationEnvelope envelope = new
SoapSerializationEnvelope(SoapEnvelope.VER11);
        // Set output SOAP object
        envelope.setOutputSoapObject(request);
        // Create HTTP call object
        HttpTransportSE androidHttpTransport = new
HttpTransportSE(URL+SERVICE_NAME+"?wsdl");

        try
        {
            // Invoke web service
            androidHttpTransport.call(NAMESPACE+"/"+METHOD_NAME,
envelope);

            // Get the response
            SoapObject response = (SoapObject) envelope.bodyIn;
            // Assign it to resTxt variable static variable
            int t = response.getPropertyCount();
            for(int i=0;i<t;i++)
                resTxt.add(response.getProperty(i).toString());
        }
        catch (Exception e)
        {
            //Print error
            e.printStackTrace();
            //Assign error message to resTxt
            resTxt.add("Error ocurred");
        }
        //Return resTxt to calling object
        return resTxt;
    }
}

```

```

    public static ArrayList<String> invocaServicioWeb(String METHOD_NAME,
String SERVICE_NAME, String Matricula)
    {
        ArrayList<String> resTxt = new ArrayList<String>();
        // Create request
        SoapObject request = new SoapObject(NAMESPACE, METHOD_NAME);
        // Property which holds input parameters
        PropertyInfo dpPI = new PropertyInfo();

        // Property
        dpPI.setName("matricula");
        dpPI.setValue(Matricula);
        dpPI.setType(Integer.class);
        // Add the property to request object
        request.addProperty(dpPI);

        // Create envelope
        SoapSerializationEnvelope envelope = new
SoapSerializationEnvelope(SoapEnvelope.VER11);
        // Set output SOAP object
        envelope.setOutputSoapObject(request);
        // Create HTTP call object
        HttpTransportSE androidHttpTransport = new
HttpTransportSE(URL+SERVICE_NAME+"?wsdl");

        try
        {
            // Invoke web service
            androidHttpTransport.call(NAMESPACE+"/"+METHOD_NAME,
envelope);

            // Get the response
            SoapObject response = (SoapObject) envelope.bodyIn;
            // Assign it to resTxt variable static variable
            int t = response.getPropertyCount();
            for(int i=0;i<t;i++)
                resTxt.add(response.getProperty(i).toString());

        }
        catch (Exception e)
        {
            //Print error
            e.printStackTrace();
            //Assign error message to resTxt
            resTxt.add("Error occurred");
        }
        //Return resTxt to calling object
        return resTxt;
    }

    public static ArrayList<String> invocaServicioWeb(String METHOD_NAME,
String SERVICE_NAME, String Matricula, String Trimestre)
    {
        ArrayList<String> resTxt = new ArrayList<String>();
        // Create request
        SoapObject request = new SoapObject(NAMESPACE, METHOD_NAME);
        // Property which holds input parameters

```

```

PropertyInfo dpPI = new PropertyInfo();
PropertyInfo dpPI2 = new PropertyInfo();
// Property
dpPI.setName("matricula");
dpPI.setValue(Matricula);
dpPI.setType(Integer.class);

dpPI2.setName("trimestre");
dpPI2.setValue(Trimestre);
dpPI2.setType(String.class);

// Add the property to request object
request.addProperty(dpPI);
request.addProperty(dpPI2);

// Create envelope
SoapSerializationEnvelope envelope = new
SoapSerializationEnvelope(SoapEnvelope.VER11);
// Set output SOAP object
envelope.setOutputSoapObject(request);
// Create HTTP call object
HttpTransportSE androidHttpTransport = new
HttpTransportSE(URL+SERVICE_NAME+"?wsdl");

try
{
    // Invoke web service
    androidHttpTransport.call(NAMESPACE+"/"+METHOD_NAME,
envelope);

    // Get the response
    SoapObject response = (SoapObject) envelope.bodyIn;
    // Assign it to resTxt variable static variable
    int t = response.getPropertyCount();
    for(int i=0;i<t;i++)
        resTxt.add(response.getProperty(i).toString());
}
catch (Exception e)
{
    //Print error
    e.printStackTrace();
    //Assign error message to resTxt
    resTxt.add("Error ocurred");
}
//Return resTxt to calling object
return resTxt;
}

public static boolean invocaServicioWebL(String METHOD_NAME, String
SERVICE_NAME, String Matricula, String Contraseña)
{
    boolean result = false;
    // Create request
    SoapObject request = new SoapObject(NAMESPACE, METHOD_NAME);
    // Property which holds input parameters
    PropertyInfo dpPI = new PropertyInfo();

```

```

PropertyInfo dpPI2 = new PropertyInfo();
// Property
dpPI.setName("matricula");
dpPI.setValue(Matricula);
dpPI.setType(Integer.class);

dpPI2.setName("contraseña");
dpPI2.setValue(Contraseña);
dpPI2.setType(String.class);

// Add the property to request object
request.addProperty(dpPI);
request.addProperty(dpPI2);

// Create envelope
SoapSerializationEnvelope envelope = new
SoapSerializationEnvelope(SoapEnvelope.VER11);
// Set output SOAP object
envelope.setOutputSoapObject(request);
// Create HTTP call object
HttpTransportSE androidHttpTransport = new
HttpTransportSE(URL+SERVICE_NAME+"?wsdl");

try
{
    // Invoke web service
    androidHttpTransport.call(NAMESPACE+"/"+METHOD_NAME,
envelope);

    // Get the response
    SoapPrimitive response = (SoapPrimitive)
envelope.getResponse();
    // Assign it to resTxt variable static variable
    result = Boolean.valueOf(response.toString());
}
catch (Exception e)
{
    //Print error
    e.printStackTrace();
}
//Return resTxt to calling object
return result;
}

public static ArrayList<String> invocaServicioWebS(String METHOD_NAME,
String SERVICE_NAME, String Nombre)
{
    ArrayList<String> resTxt = new ArrayList<String>();
    // Create request
    SoapObject request = new SoapObject(NAMESPACE, METHOD_NAME);
    // Property which holds input parameters
    PropertyInfo dpPI = new PropertyInfo();

    // Property
    dpPI.setName("nombre");
    dpPI.setValue(Nombre);
    dpPI.setType(String.class);

```

```

        // Add the property to request object
        request.addProperty(dpPI);

        // Create envelope
        SoapSerializationEnvelope envelope = new
SoapSerializationEnvelope(SoapEnvelope.VER11);
        // Set output SOAP object
        envelope.setOutputSoapObject(request);
        // Create HTTP call object
        HttpTransportSE androidHttpTransport = new
HttpTransportSE(URL+SERVICE_NAME+"?wsdl");

        try
        {
            // Invoke web service
            androidHttpTransport.call(NAMESPACE+"/"+METHOD_NAME,
envelope);

            // Get the response
            SoapObject response = (SoapObject) envelope.bodyIn;
            // Assign it to resTxt variable static variable
            int t = response.getPropertyCount();
            for(int i=0;i<t;i++)
            {
                if(response.getProperty(i) == null)
                    resTxt.add("null");
                resTxt.add(response.getProperty(i).toString());
            }
        }
        catch (Exception e)
        {
            //Print error
            e.printStackTrace();
            //Assign error message to resTxt
            resTxt.add("Error ocurred");
        }
        //Return resTxt to calling object
        return resTxt;
    }
}

```