

Universidad Autónoma Metropolitana Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

Reporte Proyecto de Integración en Ingeniería en Computación I

Aplicación para smartphones sobre la recomendación de inscripción de UEA
para alumnos de Ingeniería Electrónica

Fernando Fabián Robles

208333679

Trimestre 2013 Otoño

Asesores:

Edgar Alejandro Andrade González
Departamento de Electrónica

Tabla de Contenidos

Resumen.....	3
Objetivos.....	3
Introducción.....	3
Desarrollo del proyecto.....	5
Análisis.....	5
Diseño.....	7
-Diagrama de Casos de Uso.....	7
Implementación.....	11
-Diagrama de clases.....	11
-Código fuente.....	12
• Clase IngresoSAE.....	12
• Clase SolicitudSae.....	16
• Clase AnalisisKardex.....	25
• Clase PlanEstudios.....	50
• Clase RecomendacionUEA.....	53
• Clase MuestraUEA.....	57
Pruebas del sistema.....	64
Pantalla de inicio:.....	64
Pantalla de recomendación:.....	69
Conclusiones.....	73
Referencias.....	73

Resumen

En este documento se presentan los detalles de desarrollo e implementación del proyecto “Recomendación de inscripción de UEA para alumnos de Ingeniería Electrónica”. Este proyecto consiste en desarrollar una aplicación para dispositivos móviles que ofrezca recomendaciones a los estudiantes de Ingeniería Electrónica relativas a las UEA que deberían inscribir en su próximo trimestre, tomando en cuenta su historial académico y la prioridad de las UEA.

Objetivos

Objetivo general:

Diseñar e implementar una aplicación para dispositivos móviles que ofrezca recomendaciones a los estudiantes de Ingeniería Electrónica relativas a las UEA que deberían inscribir en su próximo trimestre, teniendo en cuenta su historial académico.

Objetivos específicos:

- Abrir una sesión en el SAE (Subsistema de Administración Escolar) usando los datos de acceso del alumno a través de la aplicación.
- Extraer y organizar los datos relevantes del kardex del alumno usando la información devuelta por el SAE.
- Modelar el diagrama de seriación de la carrera de Ingeniería Electrónica en la aplicación.
- Diseñar e implementar un algoritmo que, considerando la información del kardex del alumno, el plan de estudios, el límite de carga académica y las recomendaciones del coordinador de la carrera, genere un listado de UEA que se recomiendan cursar el próximo trimestre.
- Diseñar una interfaz gráfica interactiva y entendible al usuario.

Introducción

Actualmente en la UAM Azcapotzalco, el proceso de reinscripción de los alumnos de licenciatura se lleva a cabo al inicio de cada trimestre, en el cual el alumno solicita la inscripción a UEA (Unidades Enseñanza-Aprendizaje) obligatorias y optativas por internet[1].

Este proceso se realiza a través del SAE, que es una herramienta web que permite a los alumnos, además de reinscribirse, consultar información académica como horarios de UEA inscritas, kardex, pagos de colegiatura, créditos acumulados y por trimestre[2].

Durante el proceso de reinscripción, el SAE simplemente lista todas las UEA posibles de inscribir para las cuales el alumno ya cubrió los requisitos que indica su plan de estudios, por lo que los alumnos inscriben sus UEA a criterio propio y generalmente sin tomar en cuenta las extensas seriaciones del plan de estudios, alargando inadvertidamente la duración de su licenciatura a más de 13 trimestres.

En el caso de la Licenciatura en Ingeniería Electrónica, se reporta que los alumnos están tardando en terminar su carrera en promedio más de 20 trimestres, cuando en realidad la duración prevista de la carrera es de 12 trimestres[3]. Para apoyar a los alumnos de Ingeniería Electrónica, se propone el desarrollo de una aplicación para dispositivos móviles que genere una recomendación personalizada de UEA a inscribir, dándole la opción al alumno de llevar una cantidad de UEA acorde a su ritmo real de aprobación de créditos.

Una de las razones principales de deserción y de que al alumno le tome más tiempo del necesario el poder concluir su Licenciatura es la falta de información sobre la importancia de ciertas ramas de seriación dentro del plan de estudios. Muchas veces los alumnos elijen sus UEA a inscribir sin tomar en cuenta esto, prolongando inadvertidamente su estadía en la universidad. Con esta aplicación se pretende proporcionar una recomendación que apoye al alumno en su elección de UEA a inscribir al considerar, entre otras cosas, la importancia académica de las diferentes UEA del plan de estudios, para que así pueda terminar su licenciatura en un tiempo razonable.

Adicionalmente, se pretende que la aplicación utilice un algoritmo que considere el promedio de créditos por trimestre en la trayectoria del alumno y posteriormente haga una estimación sobre el número conveniente de créditos a inscribir. Esto tendrá como ventaja que la recomendación dependerá del rendimiento académico del alumno, y si sigue la recomendación de la aplicación, inscribirá un número de créditos adecuado para él, reduciendo el riesgo de que deba dar de baja alguna UEA. La selección de UEA que cubrirán dichos créditos estará basada en la sugerencia para cubrir los créditos mínimos del plan de estudios diseñada por el coordinador de la Licenciatura en Ingeniería Electrónica[3]. Se estima que la aplicación brindará beneficios tanto a los alumnos como a la universidad al ayudar a mejorar la eficiencia terminal.

Cabe mencionar que en los últimos años, se ha incrementado el uso de dispositivos móviles con conexión a internet[4] ya que su facilidad de uso y portabilidad los hacen idóneos para consultas rápidas. Es por eso que esta aplicación se desarrollará en una plataforma móvil, facilitando a los alumnos la consulta de la recomendación de UEA a inscribir. Además, la aplicación considerará las modificaciones al plan de estudios de Ingeniería Electrónica que entrarán en vigor en el trimestre 13-O[8].

Desarrollo del proyecto

Análisis

Teniendo en cuenta la problemática a resolver y los requerimientos mencionados, se conceptualizó la aplicación como 5 módulos funcionales que interactúan e intercambian resultados parciales para obtener la recomendación de inscripción más adecuada para el alumno. A continuación se muestra un diagrama general del funcionamiento de la aplicación y su descripción:



Fig. 1: Diagrama general de la aplicación

1. Módulo de ingreso: Este módulo se encarga de establecer la conexión entre la aplicación y el SAE, usando los datos que el alumno usa para acceder al mismo (matrícula y contraseña), abriendo la sesión en el servidor. Dicha conexión e intercambio de información se realiza mediante los métodos de petición soportados por el protocolo http, los cuales definen las acciones deseadas en el recurso. En este caso se usan los métodos GET y POST. Primero se usa POST para enviar los datos del alumno e iniciar sesión con dichos datos en el SAE. El método GET se usa para recibir la respuesta. Si no se puede establecer una conexión, o los datos son incorrectos, la aplicación notificará al usuario.

Entrada: Matrícula y contraseña del SAE.

Salida: Respuesta del SAE al intento de inicio de sesión y datos de sesión en caso de éxito.

2. Módulo de solicitud: Este módulo hará la solicitud y recibirá la información académica del alumno del SAE, es decir, su kardex y corroborará que el alumno está inscrito y se encuentra cursando la Licenciatura en Ingeniería Electrónica, de lo contrario notificará al usuario que esta aplicación no corresponde con su Licenciatura y por lo tanto no podrá usarla.

Entrada: Datos de la sesión iniciada en el SAE.

Salida: Archivo del kardex del alumno.

3. Módulo de análisis: Este módulo se encargará de procesar y extraer la información académica relevante tal como claves de UEA, calificaciones, trimestre y el tipo de

evaluación en que se aprobó y UEA dadas de baja. También tomará la decisión sobre si el alumno puede inscribir exceso de créditos y hará un análisis del plan de estudios con el avance del alumno, para determinar qué UEA son posibles de inscribir.

Entrada: Archivo del kardex del alumno.

Salida: Lista de UEA posibles a inscribir y límite de créditos para el trimestre a cursar.

4. Módulo de recomendación: Este módulo se encargará de hacer una estimación sobre cuántos créditos le conviene inscribir al alumno en base a su rendimiento (promedio de créditos aprobados por trimestre) respetando los límites de créditos máximos y mínimos establecidos por el plan de estudios. Posteriormente seleccionará las UEA que cubran los créditos anteriormente determinados de acuerdo con su prioridad. La prioridad de las UEA está dada por una recomendación del coordinador de la Licenciatura en Ingeniería Electrónica, la cual se determinó en base a las siguientes características del plan de estudios:

- Cadenas de seriación
- Número de créditos
- Número de UEA que abren
- UEA que no tienen examen de recuperación

Entrada: Lista de UEA posibles de inscribir y créditos límite a inscribir.

Salida: Lista de UEA recomendadas a inscribir.

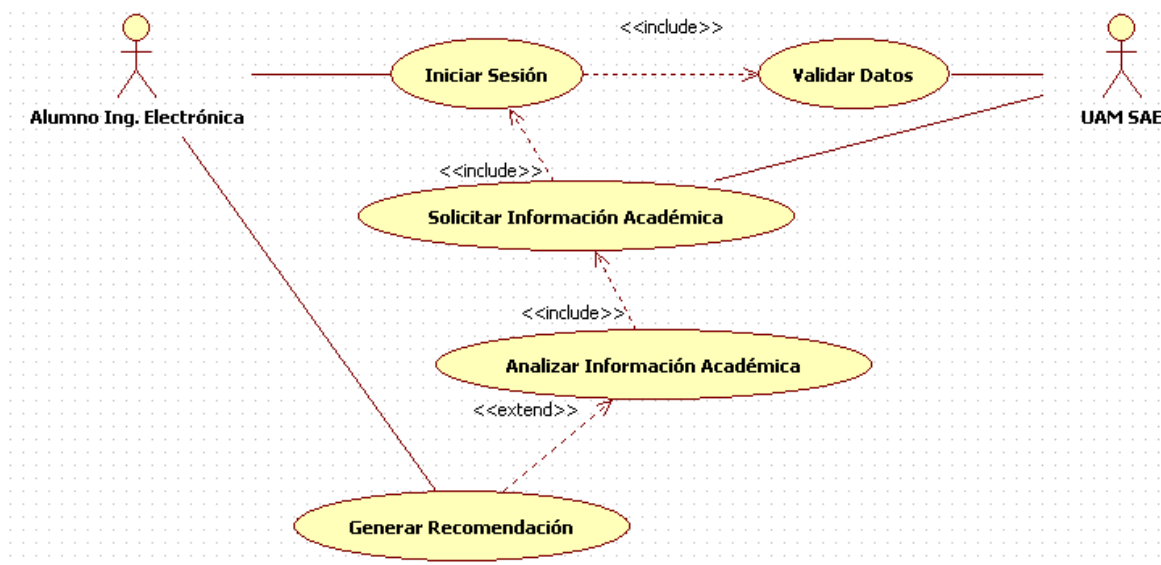
5. Módulo de presentación: Este módulo se encargará de visualizar la lista de UEA recomendadas. El alumno podrá volver a ejecutar la recomendación dándole a la aplicación un parámetro diferente de créditos a inscribir.

Entrada: Lista de UEA recomendadas a inscribir.

Salida: Lista de UEA recomendadas a inscribir.

Diseño

-Diagrama de Casos de Uso



➤ Caso de uso Iniciar Sesión

GUI
Iniciar Sesión
Caso de uso que permitirá abrir una sesión como alumno de Ing. Electrónica
Actores Primarios: Alumno de Ing. Electrónica
Disparador: El alumno de Ing. Electrónica requiere generar la recomendación de inscripción de UEA
Precondiciones: <ul style="list-style-type: none"> El alumno de Ing. Electrónica deberá autenticarse.
Postcondiciones: El alumno de Ing. Electrónica abrirá una sesión
Flujo principal <ol style="list-style-type: none"> La aplicación muestra la pantalla de login. El alumno de Ing. Electrónica ingresa su número de matrícula y su contraseña del SAE. La aplicación envía los datos al servidor del SAE para su verificación. La aplicación recibe la respuesta del servidor y se la muestra al usuario en pantalla. La aplicación abre la sesión en el servidor.
Flujo alternativo: <ol style="list-style-type: none"> La matrícula o la contraseña son incorrectas.

5.2. La aplicación muestra un mensaje notificando lo anterior
5.3. La aplicación vuelve a pedir los datos.
Requerimientos no funcionales: este proceso no debe tardar más de 5 segundos.

➤ Caso de uso Validar Datos

GU2
Validar Datos
Caso de uso que permitirá la validación de los datos introducidos por el alumno de Ing. Electrónica.
Actores Primarios: UAM SAE
Disparador: El alumno de Ing. Electrónica ha introducido su matrícula y su contraseña, los cuales deben ser validados.
Precondiciones: <ul style="list-style-type: none"> • El alumno de Ing. Electrónica debe haber introducido sus datos
Postcondiciones: el alumno de Ing. Electrónica podrá iniciar sesión o no en el SAE.
Flujo principal <ol style="list-style-type: none"> 1. La aplicación envía al SAE los datos del alumno a verificar. 2. El SAE responde la petición. 3. Se inicia la sesión en el SAE.
Flujo alternativo: <ol style="list-style-type: none"> 3.1. Datos incorrectos 3.2. La aplicación muestra el error al usuario
Requerimientos no funcionales: este proceso no debe tardar más de 5 segundos

➤ Caso de uso Solicitar Información Académica

GU3
Solicitar Información Académica
Caso de uso que permitirá solicitar Información Académica del alumno y su kardex
Actores Primarios: UAM SAE
Disparador: el alumno de Ing. Electrónica logró autenticarse y abrir una sesión en el SAE
Precondiciones: <ul style="list-style-type: none"> • El alumno de Ing. Electrónica se autenticó correctamente.
Postcondiciones: La aplicación recibirá la información académica del alumno y kárdex.
Flujo principal <ol style="list-style-type: none"> 1. La aplicación solicita al SAE la información académica y el kárdex del alumno. 2. El SAE envía el kárdex y la información académica del alumno a la aplicación 3. La aplicación guarda la información académica y el kárdex en memoria
Requerimientos no funcionales: este proceso no debe tardar más de 5 segundos.

➤ Caso de uso Analizar Información Académica

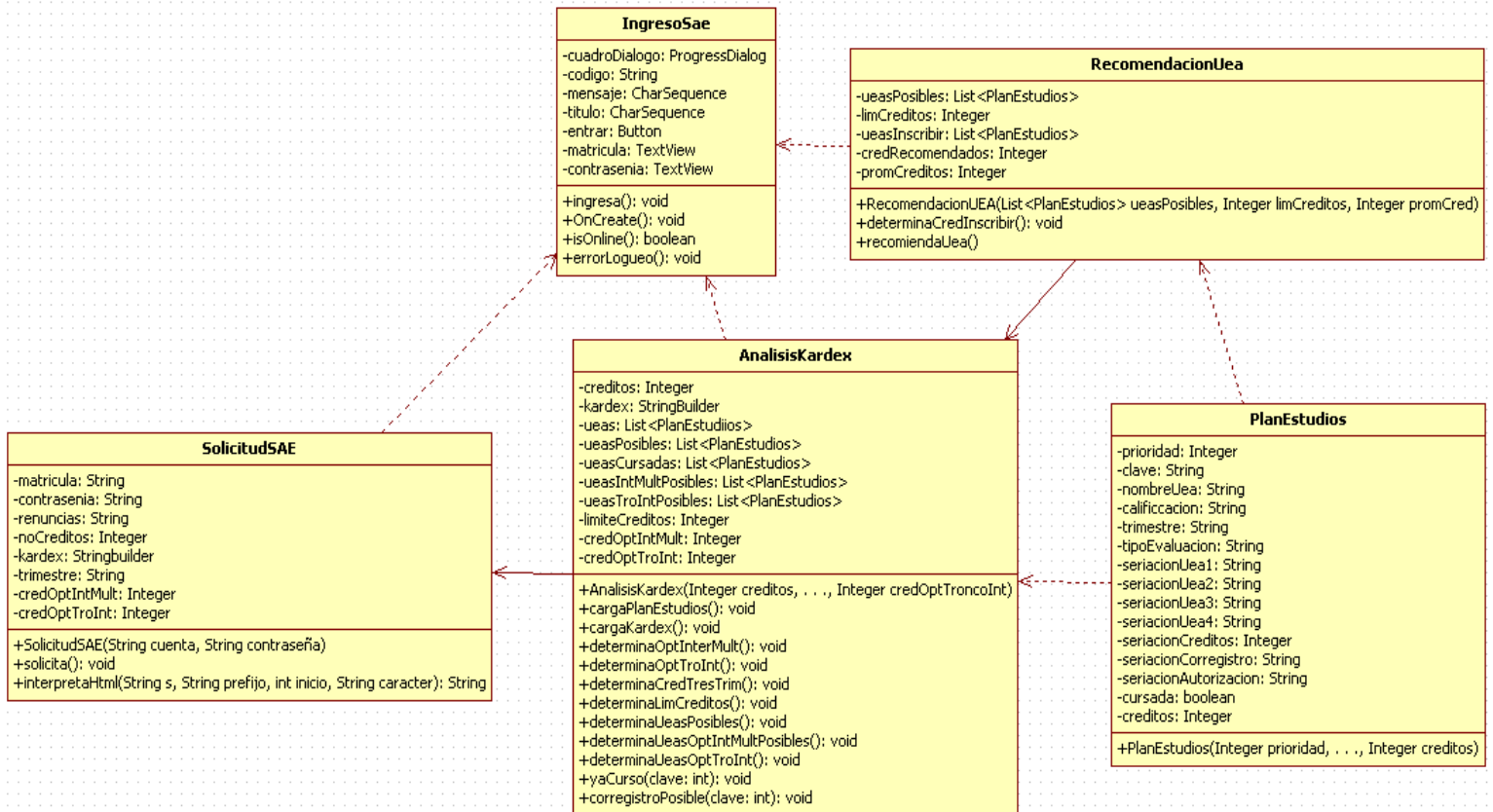
GU4
Analizar Información Académica
Caso de uso que permitirá el análisis del kárdex del alumno
Actores Primarios: La aplicación
Disparador: La aplicación requiere analizar la información almacenada en la memoria.
Precondiciones: La información académica debe estar almacenada en memoria.
Postcondiciones: Se crearán listas de UEAS que el alumno puede cursar de acuerdo a sus requerimientos y se calculará el límite de créditos de acuerdo a su información académica.
Flujo principal <ol style="list-style-type: none"> 1. La aplicación determina las UEAS posibles de inscribir de acuerdo a la información del kardex. 2. La aplicación determina el límite de créditos del alumno de acuerdo con la información académica.
Requerimientos no funcionales: este proceso no debe tardar más de 5 segundos.

➤ Caso de uso Generar recomendación

GU5
Generar recomendación
Caso de uso que permitirá generar la recomendación de inscripción de UEA
Actores Primarios: Alumno de Ing. Electrónica
Disparador: Se requiere determinar el número de créditos que se le recomienda llevar al alumno y las UEAS que cubrirán dichos créditos.
Precondiciones: <ul style="list-style-type: none">• Se han determinado el límite de créditos y las UEAS posibles de inscribir
Postcondiciones: La aplicación mostrará la información de las UEAS a inscribir y el número de créditos recomendado.
Flujo principal <ol style="list-style-type: none">1. Se determina el número de créditos recomendado.2. Se ordenan las UEAS en base a su prioridad.3. La aplicación muestra las UEAS a inscribir y sus respectivos créditos.
Flujo alternativo: <ol style="list-style-type: none">3.1. El alumno cambia el número de créditos.3.2. La aplicación recalcula las UEAS a inscribir
Requerimientos no funcionales: este proceso no debe tardar más de 5 segundos.

Implementación

-Diagrama de clases



-Código fuente

La aplicación se dividió en 6 clases en el lenguaje de programación Java, de las cuales se presenta el código fuente a continuación:

- **Clase IngresoSAE**

```
package com.uamapps.appingelectronicauam;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.List;

import org.apache.http.Header;
import org.apache.http.HttpResponse;
import org.apache.http.HttpStatus;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.HttpClient;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.cookie.Cookie;
import org.apache.http.impl.client.AbstractHttpClient;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.util.EntityUtils;

import android.app.Activity;
import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;
import android.os.AsyncTask;
import android.os.Bundle;
import android.os.Parcelable;
import android.os.StrictMode;
import android.text.InputType;
import android.util.Log;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class IngresoSAE extends Activity implements OnClickListener {
    private ProgressDialog cuadroDialogo;
    private String codigo;
    private CharSequence msg;
    private CharSequence title;
    Button ok,back,exit;
```

```

        TextView result;

/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);
    StrictMode.ThreadPolicy policy = new
StrictMode.ThreadPolicy.Builder().permitAll().build();
    StrictMode.setThreadPolicy(policy);
    setContentView(R.layout.activity_ingreso_sae);

    // Login button clicked
    ok = (Button)findViewById(R.id.button1);
    ok.setOnClickListener(this);

}

public void Ingresa() throws IOException {
    EditText uname = (EditText)findViewById(R.id.editText1);
    uname.setInputType(InputType.TYPE_CLASS_NUMBER);
    String MY_ACC = uname.getText().toString();
    EditText pword = (EditText)findViewById(R.id.editText2);
    String MY_PASS = pword.getText().toString();

    SolicitudSAE modulo_solicitud = new SolicitudSAE(MY_ACC, MY_PASS);
    msg = "Enviando datos";
    runOnUiThread(changeMessage);
    codigo = modulo_solicitud.Solicita();
    if (!codigo.equals("Kardex_recibido")) {

    }
    else
    {
        title = "Sesión iniciada en el SAE";
        runOnUiThread(changeTitle);
        AnalisisKardex modulo_analisis = new
AnalisisKardex(modulo_solicitud.getNo_creditos(), modulo_solicitud.getKardex(),
modulo_solicitud.getRenuncias(), modulo_solicitud.getTrimestre(),
modulo_solicitud.getCred_opt_inter_mult(),
modulo_solicitud.getCred_opt_tronco_int());
        modulo_analisis.carga_plan_estudios();
        msg = "Analizando información";
        runOnUiThread(changeMessage);
        modulo_analisis.carga_kardex();
        modulo_analisis.determina_opt_inter_mult();
        modulo_analisis.determina_opt_tronco_int();
        msg = "Construyendo recomendación";
        runOnUiThread(changeMessage);
        modulo_analisis.determina_creditos();
        modulo_analisis.determina_UEAS_posibles();
        modulo_analisis.determina_UEAS_opt_inter_mult_cursadas();
        modulo_analisis.determina_UEAS_opt_tronco_int_cursadas();
        modulo_analisis.determina_UEAS_opt_inter_mult_posibles();
        modulo_analisis.determina_UEAS_opt_tronco_int_posibles();
        //modulo_analisis.imprime_UEAS_cursadas();
        modulo_analisis.cred_3_ult_trim();
        //modulo_analisis.imprime_UEAS_posibles();
    }
}

```

```

        RecomendacionUEA modulo_recomendacion = new RecomendacionUEA
(modulo_analisis.getUEAS_posibles(), modulo_analisis.getLimite_credits(),
modulo_analisis.getPrim_trim_cred(), modulo_analisis.getSeg_trim_cred(),
modulo_analisis.getTer_trim_cred());
        modulo_recomendacion.determina_cred_inscribir();
        modulo_recomendacion.recomienda_UEA();
        modulo_recomendacion.determina_min_UEA();
        //modulo_recomendacion.imprime_UEAS_inscribir();
        Intent intent = new Intent(this, Activity_muestra_uea.class);
        intent.putExtra("Creditos",
modulo_analisis.getCreditos().intValue());
        intent.putExtra("Renuncias", modulo_analisis.getRenuncias());
        intent.putExtra("LimCreditos",
modulo_recomendacion.getLimite_credits());
        intent.putExtra("RecCreditos",
modulo_recomendacion.getRec_cred());
        intent.putExtra("CredInterMult",
modulo_recomendacion.getCred_inter_mult());
        intent.putExtra("CredTroInt",
modulo_recomendacion.getCred_tronco_int());
        intent.putParcelableArrayListExtra("RecUEAS", (ArrayList<? extends
Parcelable>) modulo_recomendacion.getUEAS_inscribir());
        intent.putParcelableArrayListExtra("PosUEAS", (ArrayList<? extends
Parcelable>) modulo_recomendacion.getUEAS_posibles());
        intent.putParcelableArrayListExtra("RecUEASOptInterMult",
(ArrayList<? extends Parcelable>)
modulo_analisis.getUEAS_opt_inter_mult_posibles());
        intent.putParcelableArrayListExtra("PosUEASOptTroncoIntPosibles",
(ArrayList<? extends Parcelable>)
modulo_analisis.getUEAS_opt_tronco_int_posibles());
        intent.putExtra("Min", modulo_recomendacion.getMin());
        startActivity(intent);

    }
    //System.out.println(modulo_solicitud.renuncias);
}

private void error_logueo(String message) {
    AlertDialog ad = new AlertDialog.Builder(this).create();
    ad.setCancelable(false); // This blocks the 'BACK' button
    ad.setMessage(message);
    ad.setButton("OK", new DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.dismiss();
        }
    });
    ad.show();
}

private class YourAsyncTask extends AsyncTask<Void, Void, Void> {
    @Override
    protected void onPreExecute() {

        cuadroDialogo = ProgressDialog.show(IngresoSAE.this, "Conectándose
al SAE", "Conectando...");

    }
    protected Void doInBackground(Void... args) {
        // do background work here
    }
}

```

```

        try {
            if (!isOnline()) {
                codigo = "No hay conexión a Internet";
            } else {
                Ingresa();
            }
        } catch (IOException e) {
            // TODO Auto-generated catch block
            e.printStackTrace();
        }
        return null;
    }

    protected void onPostExecute(Void result) {
        // do UI work here
        cuadroDialogo.dismiss();
        if (!codigo.equals("Kardex_recibido") || codigo.equals("No hay
conexión a Internet"))
            error_logueo (codigo);
    }
}

@Override
public void onClick(View view) {
    if(view == ok){
        new YourAsyncTask().execute();

        //progressDialog =
ProgressDialog.show(IngresoSAE.this, "", "Loading...");

        //new Thread() {

        //public void run() {

        //    try {
        //        Ingresa();
        //    } catch (IOException e) {
        //        // TODO Auto-generated catch block
        //        e.printStackTrace();
        //    }

        //dismiss the progress dialog

        //    progressDialog.dismiss();

        //    }

        //    }.start();

        //    error_logueo (codigo);

        }
}

private Runnable changeMessage = new Runnable() {
    @Override
    public void run() {

        //Log.v(TAG, strCharacters);
        cuadroDialogo.setMessage(msg);
    }
}

```

```

    }
};
private Runnable changeTitle = new Runnable() {
    @Override
    public void run() {

        //Log.v(TAG, strCharacters);
        cuadroDialogo.setTitle(title);
    }
};

public boolean isOnline() {
    ConnectivityManager cm = (ConnectivityManager)
getSystemService(Context.CONNECTIVITY_SERVICE);
    NetworkInfo netInfo = cm.getActiveNetworkInfo();

    if (netInfo != null && netInfo.isConnectedOrConnecting()
        && cm.getActiveNetworkInfo().isAvailable()
        && cm.getActiveNetworkInfo().isConnected()) {
        return true;
    }
    return false;
}
}

```

- **Clase SolicitudSae**

```

package com.uamapps.appingelectronicauam;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.SocketTimeoutException;
import java.util.ArrayList;
import java.util.List;
import java.util.StringTokenizer;

import org.apache.http.Header;
import org.apache.http.HttpResponse;
import org.apache.http.HttpStatus;
import org.apache.http.NameValuePair;
import org.apache.http.client.ClientProtocolException;
import org.apache.http.client.entity.UrlEncodedFormEntity;
import org.apache.http.client.methods.HttpGet;
import org.apache.http.client.methods.HttpPost;
import org.apache.http.conn.ConnectTimeoutException;
import org.apache.http.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.params.BasicHttpParams;
import org.apache.http.params.HttpConnectionParams;
import org.apache.http.params.HttpParams;

import android.app.AlertDialog;
import android.content.DialogInterface;
import android.util.Log;
import android.widget.EditText;

public class SolicitudSAE {

```



```

private static final String TAG = "MyActivity";
private String MY_ACC;
private String MY_PASS;
private String renunciias;
private Integer no_creditos;
private StringBuilder kardex;
private String trimestre;
private Integer cred_opt_inter_mult;
private Integer cred_opt_tronco_int;

public SolicitudSAE (String cuenta, String contraseña) {
    this.MY_ACC = cuenta;
    this.MY_PASS = contraseña;
}

public String Solicita() {

    String UAM_KARDEX =
"https://ayamictlan.uam.mx:8443/sae/azc/IEWBC020.oConsulta";
    String UAM_LOGIN =
"https://ayamictlan.uam.mx:8443/sae/azc/AEWBU004.oIniSesWebLic?mod=1";
    String UAM_AC_INFO =
"https://ayamictlan.uam.mx:8443/sae/azc/IEWBC007.oConsulta";
    String UAM_SALIR =
"https://ayamictlan.uam.mx:8443/sae/azc/AEWBU005.oFinSesion";
    String UAM_SALIR1 =
"https://ayamictlan.uam.mx:8443/sae/azc/AEWBU002.oFin?mod=1";

    try {
        List<NameValuePair> nameValuePairs = new
ArrayList<NameValuePair>(11);
        nameValuePairs.add(new BasicNameValuePair("#.E_UNIDAD.AE02.1",
"AxJDMQ=="));
        nameValuePairs.add(new BasicNameValuePair("#.MODULO_UWAS.SAE01",
""));
        nameValuePairs.add(new BasicNameValuePair("#.USUARIO_ANEXO.SG02",
""));
        nameValuePairs.add(new BasicNameValuePair("#.WEB_INFO.SW01", ""));
        nameValuePairs.add(new BasicNameValuePair("#.WEB_MOD_ASO.SW01",
""));
        nameValuePairs.add(new BasicNameValuePair("#CRC.E_UNIDAD.AE02.1",
"00000022"));
        nameValuePairs.add(new
BasicNameValuePair("% .IDENTIFICACION.NONMODELED", ""));
        nameValuePairs.add(new
BasicNameValuePair("COMPLEMENTO.IDENTIFICACION.NONMODELED", MY_PASS));
        nameValuePairs.add(new
BasicNameValuePair("GO.IDENTIFICACION.NONMODELED", "Entrar"));
        nameValuePairs.add(new
BasicNameValuePair("NOMBRE.IDENTIFICACION.NONMODELED", MY_ACC));
        nameValuePairs.add(new
BasicNameValuePair("SIGLAS_UNI_XX.E_UNIDAD.AE02.1", "AZC"));

        HttpParams httpParameters = new BasicHttpParams();

        int timeoutConnection = 5000;
        HttpConnectionParams.setConnectionTimeout(httpParameters,
timeoutConnection);

        int timeoutSocket = 10000;
        HttpConnectionParams.setSoTimeout(httpParameters, timeoutSocket);

```

```

DefaultHttpClient httpClient = new DefaultHttpClient(httpParameters);
HttpPost httpPost = new HttpPost(UAM_LOGIN);

        httpPost.setEntity(new
UrlEncodedFormEntity(nameValuePairs));
        HttpResponse responsePost = httpClient.execute(httpPost);

        InputStream inp = responsePost.getEntity().getContent();
        BufferedReader buff = new BufferedReader(new
InputStreamReader(inp));
        String line = "anything";
        while (line != null) {
            line = buff.readLine();
            String found = interpretHtml(line, "<body class=\"\"
onload=\"Ualert('[info]', 37, '\"');
            if(null != found){

                httpClient.getConnectionManager().shutdown();
                return found;
            }
        }

        if (responsePost.getStatusLine().getStatusCode() ==
HttpStatus.SC_OK) {

            Header[] cookies = responsePost.getHeaders("Set-Cookie");

            HttpGet httpget = new HttpGet(UAM_AC_INFO);

            for (Header c : cookies) {
                httpget.addHeader("Cookie", c.getValue());
            }

            HttpResponse responseGet = httpClient.execute(httpget);

            InputStream ins = responseGet.getEntity().getContent();
            BufferedReader in = new BufferedReader(new
InputStreamReader(ins));

            Integer cred_op1 = 0;
            Integer cred_op2 = 0;
            Integer cred_op3 = 0;
            Integer cred_op4 = 0;
            Integer cred_op5 = 0;
            Integer cred_op6 = 0;
            Integer cred_op7 = 0;
            Integer cred_op8 = 0;
            Integer cred_op9 = 0;
            Integer cred_op10 = 0;
            Integer cred_op11 = 0;
            Integer cred_op12 = 0;
            Integer cred_op13 = 0;
            Integer cred_op14 = 0;

            String inputLine = "anything";
            while (inputLine != null) {

```

```

        inputLine = in.readLine();
        String found = interpretHtml(inputLine, "<td
class=\"tgruposub\">LICENCIATURA EN ", 38, "<");
        String trim_ac = interpretHtml(inputLine,
"TRIMESTRE_XX.TRIMESTRE_ACA", 42, "<");
        String creditos = interpretHtml(inputLine,
"CRED_ACU_AL_NU.V_ALUMNO_ACAD", 42, "<");
        String renuncia = interpretHtml(inputLine,
"RENUNCIA_RT_NU.V_RESUMEN_TRI", 4, "<");
        String op1 = interpretHtml(inputLine, "OPT. LINEA
ESTUDIOS CULTURALES\"", 4, "<");
        String op2 = interpretHtml(inputLine, "OPT. LINEA
FORMACION CIUDADANA\"", 4, "\"");
        String op3 = interpretHtml(inputLine, "OPT. LINEA
INDUCCION AL MERCADO LABORAL\"", 4, "<");
        String op4 = interpretHtml(inputLine, "OPT. LINEA ARTE Y
HUMANIDADES\"", 4, "<");
        String op5 = interpretHtml(inputLine, "OPT. LINEA
LENGUAJES FORMALES\"", 4, "<");
        String op6 = interpretHtml(inputLine, "OPTATIVAS INTER Y
MULTIDISCIPLINARES\"", 4, "<");

        String op7 = interpretHtml(inputLine, "OPT.
TUTORIALES\"", 4, "<");
        String op8 = interpretHtml(inputLine, "OPT. DE
MOVILIDAD\"", 4, "<");
        String op9 = interpretHtml(inputLine, "OPT. AREA DE
CONCENTRACION - REDES DE COMPUTADORAS\"", 4, "\"");
        String op10 = interpretHtml(inputLine, "OPT. AREA DE
CONCENTRACION - SISTEMAS EMBEBIDOS\"", 4, "\"");
        String op11 = interpretHtml(inputLine, "OPT. AREA DE
CONCENTRACION - INSTRUMENTACION Y CONTROL\"", 4, "<");
        String op12 = interpretHtml(inputLine, "OPT. AREA DE
CONCENTRACION - TELECOMUNICACIONES\"", 4, "\"");
        String op13 = interpretHtml(inputLine, "OPT. AREA DE
CONCENTRACION - MECATRONICA\"", 4, "<");
        String op14 = interpretHtml(inputLine, "OTRAS OPTATIVAS
DE INTEGRACION\"", 4, "\"");

        if(null != found && !found.equals("INGENIERIA
ELECTRONICA")){

            httpClient.getConnectionManager().shutdown();
            return "Esta aplicación sólo es para uso de alumnos
de Ingeniería Electrónica. Por favor, descargue la aplicación que corresponde a
su Licenciatura.";
        }
        if(null != creditos){
            this.no_creditos = Integer.parseInt(creditos);
        }
        if(null != renuncia){
            this.renuncias = renuncia;
        }
        if(null != trim_ac){

```

```

        this.trimestre = trim_ac;

    }
    if (null != op1) {
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        op1 = interpretHtml(inputLine,
"CR_CUB_VL_NU.ALUMNO", 79, "\"");

        cred_op1 = Integer.parseInt(op1.trim());

    }
    if (null != op2) {
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        op2 = interpretHtml(inputLine,
"CR_CUB_VL_NU.ALUMNO", 79, "\"");

        cred_op2 = Integer.parseInt(op2.trim());

    }
    if (null != op3) {
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        op3 = interpretHtml(inputLine,
"CR_CUB_VL_NU.ALUMNO", 79, "\"");

        cred_op3 = Integer.parseInt(op3.trim());

    }
    if (null != op4) {
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        op4 = interpretHtml(inputLine,
"CR_CUB_VL_NU.ALUMNO", 79, "\"");

        cred_op4 = Integer.parseInt(op4.trim());

    }
    if (null != op5) {
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
    }

```

```

op5 = interpretHtml(inputLine,
"CR_CUB_VL_NU.ALUMNO", 79, "\\");

cred_op5 = Integer.parseInt(op5.trim());

}
if (null != op6) {
inputLine = in.readLine();
inputLine = in.readLine();
inputLine = in.readLine();
inputLine = in.readLine();
inputLine = in.readLine();
op6 = interpretHtml(inputLine,
"CR_CUB_VL_NU.ALUMNO", 79, "\\");

cred_op6 = Integer.parseInt(op6.trim());

}
if (null != op7) {
inputLine = in.readLine();
inputLine = in.readLine();
inputLine = in.readLine();
inputLine = in.readLine();
inputLine = in.readLine();
op7 = interpretHtml(inputLine,
"CR_CUB_VL_NU.ALUMNO", 79, "\\");

cred_op7 = Integer.parseInt(op7.trim());

}
if (null != op8) {
inputLine = in.readLine();
inputLine = in.readLine();
inputLine = in.readLine();
inputLine = in.readLine();
inputLine = in.readLine();
op8 = interpretHtml(inputLine,
"CR_CUB_VL_NU.ALUMNO", 79, "\\");

cred_op8 = Integer.parseInt(op8.trim());

}
if (null != op9) {
inputLine = in.readLine();
inputLine = in.readLine();
inputLine = in.readLine();
inputLine = in.readLine();
inputLine = in.readLine();
op9 = interpretHtml(inputLine,
"CR_CUB_VL_NU.ALUMNO", 79, "\\");

cred_op9 = Integer.parseInt(op9.trim());

}
if (null != op10) {
inputLine = in.readLine();
inputLine = in.readLine();
inputLine = in.readLine();

```

```

        inputLine = in.readLine();
        inputLine = in.readLine();
        op10 = interpretHtml(inputLine,
"CR_CUB_VL_NU.ALUMNO", 79, "\"");

        cred_op10 = Integer.parseInt(op10.trim());
    }
    if (null != op11) {
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        op11 = interpretHtml(inputLine,
"CR_CUB_VL_NU.ALUMNO", 79, "\"");

        cred_op11 = Integer.parseInt(op11.trim());
    }
    if (null != op12) {
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        op12 = interpretHtml(inputLine,
"CR_CUB_VL_NU.ALUMNO", 79, "\"");

        cred_op12 = Integer.parseInt(op12.trim());
    }
    if (null != op13) {
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        op13 = interpretHtml(inputLine,
"CR_CUB_VL_NU.ALUMNO", 79, "\"");

        cred_op13 = Integer.parseInt(op13.trim());
    }
    if (null != op14) {
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        inputLine = in.readLine();
        op14 = interpretHtml(inputLine,
"CR_CUB_VL_NU.ALUMNO", 79, "\"");

        cred_op14 = Integer.parseInt(op14.trim());
    }
}

```

```

        }
        cred_opt_inter_mult = cred_op1 + cred_op2 + cred_op3 +
cred_op4 + cred_op5 + cred_op6;
        cred_opt_tronco_int = cred_op7 + cred_op8 + cred_op9 +
cred_op10 + cred_op11 + cred_op12 + cred_op13 + cred_op14;

        HttpGet httpgetKardex = new HttpGet(UAM_KARDEX);
        for (Header c : cookies) {
            httpgetKardex.addHeader("Cookie", c.getValue());
        }
        HttpResponse responseGetKardex =
httpClient.execute(httpgetKardex);

        InputStream kardex =
responseGetKardex.getEntity().getContent();

        this.kardex = inputStreamToString(kardex);

        responseGetKardex.getEntity().consumeContent();

        HttpGet httpget1 = new HttpGet(UAM_SALIR);
        for (Header c : cookies) {
            httpget1.addHeader("Cookie", c.getValue());
        }
        HttpResponse responseGet1 = httpClient.execute(httpget1);
        responseGet1.getEntity().consumeContent();

        HttpPost httpPost1 = new HttpPost(UAM_SALIR1);
        HttpResponse responsePost1 = httpClient.execute(httpPost1);
        responsePost1.getEntity().consumeContent();

        in.close();
        httpClient.getConnectionManager().shutdown();
    } else {
        return "Error de respuesta: " +
responsePost.getStatusLine().getStatusCode();
    }

    } catch (SocketTimeoutException e) {
        return "No se pudo establecer una conexión, por favor intente más
tarde";
    }
    catch (ConnectTimeoutException e) {
        return "No se pudo establecer una conexión, por favor
intente más tarde";
    }

    catch (ClientProtocolException e) {
        e.printStackTrace();
    } catch (IOException e) {
        e.printStackTrace();
    }

        return "Kardex_recibido";

    }

    private String interpretHtml(String s, String prefijo, int inicio, String
carac){
        if (s == null)

```

```

        return null;
    if(s.contains(prefijo)){
        StringTokenizer tokens = new StringTokenizer(s.substring(inicio,
s.length()), carac);
        String contenido = tokens.nextToken();
        return contenido;
    }
    else
        return null;
}

private StringBuilder inputStreamToString(InputStream is) {
String line = "";
StringBuilder total = new StringBuilder();

BufferedReader rd = new BufferedReader(new InputStreamReader(is));

try {
        while ((line = rd.readLine()) != null) {
            total.append(line);
            total.append("\n");
        }
    } catch (IOException e) {
        e.printStackTrace();
    }

return total;
}

public String getMY_ACC() {
    return MY_ACC;
}

public void setMY_ACC(String mY_ACC) {
    MY_ACC = mY_ACC;
}

public String getMY_PASS() {
    return MY_PASS;
}

public void setMY_PASS(String mY_PASS) {
    MY_PASS = mY_PASS;
}

public String getRenuncias() {
    return renuncias;
}

public void setRenuncias(String renuncias) {
    this.renuncias = renuncias;
}

public Integer getNo_creditos() {
    return no_creditos;
}

public void setNo_creditos(Integer no_creditos) {
    this.no_creditos = no_creditos;
}
}

```



```

    public StringBuilder getKardex() {
        return kardex;
    }

    public void setKardex(StringBuilder kardex) {
        this.kardex = kardex;
    }

    public String getTrimestre() {
        return trimestre;
    }

    public void setTrimestre(String trimestre) {
        this.trimestre = trimestre;
    }

    public Integer getCred_opt_inter_mult() {
        return cred_opt_inter_mult;
    }

    public void setCred_opt_inter_mult(Integer cred_opt_inter_mult) {
        this.cred_opt_inter_mult = cred_opt_inter_mult;
    }

    public Integer getCred_opt_tronco_int() {
        return cred_opt_tronco_int;
    }

    public void setCred_opt_tronco_int(Integer cred_opt_tronco_int) {
        this.cred_opt_tronco_int = cred_opt_tronco_int;
    }

}

```

- **Clase AnalisisKardex**

```

package com.uamapps.appingelectronicauam;

import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
import java.util.StringTokenizer;

import org.apache.http.NameValuePair;

public class AnalisisKardex {
    private Integer creditos;
    private StringBuilder kardex;
    private List<PlanEstudios> UEAS;
    private List<PlanEstudios> UEAS_posibles;
    private List<PlanEstudios> UEAS_cursadas;
    private List<PlanEstudios> UEAS_opt_inter_mult;
    private List<PlanEstudios> UEAS_opt_tronco_int;
}

```

```

private List<PlanEstudios> UEAS_opt_inter_mult_posibles;
private List<PlanEstudios> UEAS_opt_tronco_int_posibles;
private String renunciias;
private Integer limite_credits;
private String Ult_trimestre;
private Integer prim_trim_cred;
private Integer seg_trim_cred;
private Integer ter_trim_cred;
private Integer cred_opt_inter_mult;
private Integer cred_opt_tronco_int;

// Constructor to initialize balance
public AnalisisKardex( Integer Credits, StringBuilder Kardex, String
renucia , String trimestre, Integer cred_opt_inter_mult, Integer
cred_opt_tronco_int)
{
    this.credits = Credits;
    this.kardex = Kardex;
    this.renunciias = renucia;
    this.Ult_trimestre = trimestre;
    this.prim_trim_cred = 0;
    this.seg_trim_cred = 0;
    this.ter_trim_cred = 0;
    this.cred_opt_inter_mult = cred_opt_inter_mult;
    this.cred_opt_tronco_int = cred_opt_tronco_int;
}

public void carga_plan_estudios (){
    this.UEAS_opt_inter_mult = new ArrayList<PlanEstudios>(68);
    this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100073", "El Paisaje
como Agente de los Asentamientos y de la Cultura", "SC", "ST", "SE", "0000000",
"0000000", "0000000", "0000000", 150, "0000000", "NA", 6));
    this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100074", "Familia y
Violencia en el México Contemporáneo ", "SC", "ST", "SE", "0000000", "0000000",
"0000000", "0000000", 150, "0000000", "NA", 6));
    this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100075", "Género y
Sexualidad", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000", 150,
"0000000", "NA", 6));
    this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100076", "Poder y
Género", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000", 150,
"0000000", "NA", 6));

    this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100077",
"Administración y Economía Política de la Ciencia y Tecnología", "SC", "ST",
"SE", "0000000", "0000000", "0000000", "0000000", 150, "0000000", "NA", 6));
    this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100078", "Derechos
Humanos", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000", 150,
"0000000", "NA", 6));
    this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100079", "Economía
Mundial", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000", 150,
"0000000", "NA", 6));
    this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100080", "Ética y
Valores", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000", 150,
"0000000", "NA", 6));
    this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100081", "Historia
Social de México en el Siglo XX", "SC", "ST", "SE", "0000000", "0000000",
"0000000", "0000000", 150, "0000000", "NA", 6));
    this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100082",
"Responsabilidad Social Organizacional", "SC", "ST", "SE", "0000000",
"0000000", "0000000", "0000000", 150, "0000000", "NA", 6));

    this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100083",
"Comunicación en Proyectos Multidisciplinarios", "SC", "ST", "SE", "0000000",
"0000000", "0000000", "0000000", 150, "0000000", "NA", 6));
}

```

```

        this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100084",
"Herramientas para el Emprendedor", "SC", "ST", "SE", "0000000", "0000000",
"0000000", "0000000", 150, "0000000", "NA", 6));
        this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100085", "Inserción
Laboral", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000", 150,
"0000000", "NA", 6));
        this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100086", "Planeación
Estratégica", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000",
150, "0000000", "NA", 6));
        this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100087", "Proyectos
de Inversión", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000",
150, "0000000", "NA", 6));

        this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100088", "Historia
del Arte", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000", 150,
"0000000", "NA", 6));
        this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100089", "Taller de
Dibujo", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000", 150,
"0000000", "NA", 6));
        this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100090", "Taller de
Fotografía", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000", 150,
"0000000", "NA", 6));
        this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100091", "Taller de
Teatro", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000", 150,
"0000000", "NA", 6));

        this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100092", "Divulgación
del Conocimiento", "SC", "ST", "SE", "0000000", "0000000", "0000000",
"0000000", 150, "0000000", "NA", 6));
        this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100093", "Habilidades
Creativas para el Ámbito Profesional", "SC", "ST", "SE", "0000000", "0000000",
"0000000", "0000000", 150, "0000000", "NA", 6));
        this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100094", "Laboratorio
de Usabilidad", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000",
150, "0000000", "NA", 6));
        this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100095", "Narrativa
para Medios Audiovisuales y Digitales", "SC", "ST", "SE", "0000000", "0000000",
"0000000", "0000000", 150, "0000000", "NA", 6));
        this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100096", "Taller de
Expresión Oral y Escrita ", "SC", "ST", "SE", "0000000", "0000000", "0000000",
"0000000", 150, "0000000", "NA", 6));

        this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100097", "Temas
Selectos Inter y Multidisciplinarios I", "SC", "ST", "SE", "0000000", "0000000",
"0000000", "0000000", 150, "0000000", "NA", 6));
        this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100098", "Temas
Selectos Inter y Multidisciplinarios II ", "SC", "ST", "SE", "0000000",
"0000000", "0000000", "0000000", 150, "0000000", "NA", 6));
        this.UEAS_opt_inter_mult.add(new PlanEstudios(1, "1100099", "Experiencia
Inter y Multidisciplinar", "SC", "ST", "SE", "0000000", "0000000", "0000000",
"0000000", 150, "0000000", "NA", 6));

        this.UEAS_opt_tronco_int = new ArrayList<PlanEstudios>(100);
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1100125", "Proyecto de
Integración en Ingeniería Electrónica II", "SC", "ST", "SE", "0000000",
"0000000", "0000000", "0000000", 240, "1100115", "NA", 18));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1100135", "Introducción
al Trabajo de Investigación en Ingeniería Electrónica", "SC", "ST", "SE",
"1100105", "0000000", "0000000", "0000000", 240, "0000000", "NA", 6));

        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1100021", "Optativa
Técnica de Movilidad I", "SC", "ST", "SE", "0000000", "0000000", "0000000",
"0000000", 240, "0000000", "NA", 3));

```

```

        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1100022", "Optativa
Técnica de Movilidad II", "SC", "ST", "SE", "0000000", "0000000", "0000000",
"0000000", 240, "0000000", "NA", 3));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1100023", "Optativa
Técnica de Movilidad III", "SC", "ST", "SE", "0000000", "0000000", "0000000",
"0000000", 240, "0000000", "NA", 6));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1100024", "Optativa
Técnica de Movilidad IV", "SC", "ST", "SE", "0000000", "0000000", "0000000",
"0000000", 240, "0000000", "NA", 6));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1100025", "Optativa
Técnica de Movilidad V", "SC", "ST", "SE", "0000000", "0000000", "0000000",
"0000000", 240, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1100026", "Optativa
Técnica de Movilidad VI", "SC", "ST", "SE", "0000000", "0000000", "0000000",
"0000000", 240, "0000000", "NA", 9));

        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1121025",
"Arquitectura de Computadoras", "SC", "ST", "SE", "1121060", "0000000",
"0000000", "0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1121043", "Diseño y
Administración de Redes de Computadoras", "SC", "ST", "SE", "1121038",
"0000000", "0000000", "0000000", 0, "0000000", "NA", 12));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1121044", "Seguridad
en Redes", "SC", "ST", "SE", "1121043", "0000000", "0000000", "0000000", 0,
"0000000", "NA", 6));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1121046", "Temas
Selectos de Redes de Computadoras I", "SC", "ST", "SE", "1121043", "0000000",
"0000000", "0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1121047", "Temas
Selectos de Redes de Computadoras II", "SC", "ST", "SE", "1121043", "0000000",
"0000000", "0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1121057", "Redes
Inalámbricas", "SC", "ST", "SE", "1121043", "0000000", "0000000", "0000000", 0,
"0000000", "NA", 6));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1121058", "Integración
de Redes de Voz y Datos", "SC", "ST", "SE", "1121043", "0000000", "0000000",
"0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1151018", "Sistemas
Operativos", "SC", "ST", "SE", "1121025", "1121038", "0000000", "0000000", 0,
"0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1151069", "Seguridad
en los Sistemas de Información", "SC", "ST", "SE", "1121044", "1121057",
"1121058", "0000000", 0, "0000000", "NA", 6));

        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1121025",
"Arquitectura de Computadoras", "SC", "ST", "SE", "1121060", "0000000",
"0000000", "0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1121039", "Laboratorio
de Arquitectura de Computadoras", "SC", "ST", "SE", "0000000", "0000000",
"0000000", "0000000", 0, "1121025", "NA", 3));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1121053",
"Arquitecturas Paralelas de Computadoras", "SC", "ST", "SE", "1121025",
"0000000", "0000000", "0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1121054", "Taller de
Sistemas Embebidos", "SC", "ST", "SE", "1151018", "1151073", "1121032",
"1121053", 0, "0000000", "NA", 6));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1121056", "Temas
Selectos de Sistemas Embebidos", "SC", "ST", "SE", "1121054", "0000000",
"0000000", "0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1151018", "Sistemas
Operativos", "SC", "ST", "SE", "1121025", "1121038", "0000000", "0000000", 0,
"0000000", "NA", 9));

```

```

        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1151041",
"Almacenamiento y Estructuras de Archivos", "SC", "ST", "SE", "1151042",
"1153001", "0000000", "0000000", 0, "0000000", "NA", 8));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1151046", "Sistemas
Distribuidos", "SC", "ST", "SE", "1151018", "1151041", "0000000", "0000000", 0,
"0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1151073",
"Programación de Sistemas", "SC", "ST", "SE", "1121060", "0000000", "0000000",
"0000000", 0, "0000000", "NA", 9));

        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1123024", "Impulsores
para Motores de CD y CA", "SC", "ST", "SE", "1123044", "1124050", "0000000",
"0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1123028", "Fuentes de
Alimentación Conmutadas", "SC", "ST", "SE", "1123044", "1124050", "0000000",
"0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1123042", "Diseño de
Instrumentos", "SC", "ST", "SE", "1123043", "1121034", "0000000", "0000000", 0,
"0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1123044", "Electrónica
de Potencia", "SC", "ST", "SE", "1123041", "1124003", "0000000", "0000000", 0,
"0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1123047", "Laboratorio
de Diseño de Instrumentos", "SC", "ST", "SE", "0000000", "0000000", "0000000",
"0000000", 0, "1123042", "NA", 3));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1123034", "Laboratorio
de Electrónica de Potencia", "SC", "ST", "SE", "0000000", "0000000", "0000000",
"0000000", 0, "1123044", "NA", 3));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1123050", "Laboratorio
de Instrumentación y Control mediante Programación Gráfica", "SC", "ST", "SE",
"0000000", "0000000", "0000000", 0, "1123051", "NA", 3));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1123051",
"Instrumentación y Control mediante Programación Gráfica ", "SC", "ST", "SE",
"1123043", "0000000", "0000000", "0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1124039",
"Controladores Lógicos Programables", "SC", "ST", "SE", "1124045", "0000000",
"0000000", "0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1124040", "Temas
Selectos de Control e Instrumentación I", "SC", "ST", "SE", "1124045",
"0000000", "0000000", "0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1124041", "Temas
Selectos de Control e Instrumentación II", "SC", "ST", "SE", "1124045",
"0000000", "0000000", "0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1124044", "Análisis de
Sistemas de Control", "SC", "ST", "SE", "1124045", "0000000", "0000000",
"0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1124046", "Control
Moderno", "SC", "ST", "SE", "1124045", "0000000", "0000000", "0000000", 0,
"0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1124047", "Interfaces
Hombre-Máquina", "SC", "ST", "SE", "1124039", "0000000", "0000000", "0000000",
0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1124055",
"Comunicaciones Industriales", "SC", "ST", "SE", "1124039", "0000000",
"0000000", "0000000", 0, "0000000", "NA", 9));

        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1122020",
"Comunicaciones Celulares", "SC", "ST", "SE", "1122015", "0000000", "0000000",
"0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1122021",
"Comunicaciones ópticas", "SC", "ST", "SE", "1122015", "0000000", "0000000",
"0000000", 0, "0000000", "NA", 9));

```

```

        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1122022",
"Comunicaciones Satelitales", "SC", "ST", "SE", "1122015", "0000000",
"0000000", "0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1122023", "Laboratorio
de Radiocomunicación", "SC", "ST", "SE", "1122014", "0000000", "0000000",
"0000000", 0, "0000000", "NA", 6));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1122024", "Laboratorio
de Simulación de Sistemas de Comunicaciones", "SC", "ST", "SE", "1122014",
"0000000", "0000000", "0000000", 0, "0000000", "NA", 6));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1122026", "Señales
Aleatorias", "SC", "ST", "SE", "1122009", "0000000", "0000000", "0000000", 0,
"0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1122027", "Temas
Avanzados de Comunicaciones", "SC", "ST", "SE", "1122015", "0000000",
"0000000", "0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1122028",
"Teletráfico", "SC", "ST", "SE", "1122026", "0000000", "0000000", "0000000", 0,
"0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1122029", "Tecnologías
de Radio Móvil", "SC", "ST", "SE", "1122014", "0000000", "0000000", "0000000",
0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1122030", "Temas
Selectos de Telecomunicaciones I", "SC", "ST", "SE", "1122027", "0000000",
"0000000", "0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1122031", "Temas
Selectos de Telecomunicaciones II", "SC", "ST", "SE", "1122027", "0000000",
"0000000", "0000000", 0, "0000000", "NA", 9));

        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1100034", "Taller de
Mecatrónica", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000", 0,
"1124043", "NA", 6));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1100035", "Temas
Selectos de Ingeniería Mecatrónica", "SC", "ST", "SE", "1100034", "0000000",
"0000000", "0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1124043",
"Automatización Industrial", "SC", "ST", "SE", "1121034", "1124050", "1123043",
"1133060", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1124051", "Temas
Selectos de Robótica", "SC", "ST", "SE", "1100034", "0000000", "0000000",
"0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1133009", "Laboratorio
de Mecanismos", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000",
0, "1133060", "NA", 3));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1133024", "Dinámica de
Máquinas", "SC", "ST", "SE", "1133060", "0000000", "0000000", "0000000", 0,
"0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1133032", "Diseño de
Mecanismos", "SC", "ST", "SE", "1133060", "0000000", "0000000", "0000000", 0,
"0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1133059", "Manufactura
Asistida por Computadora", "SC", "ST", "SE", "1133061", "0000000", "0000000",
"0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1133060",
"Mecanismos", "SC", "ST", "SE", "1133061", "1111081", "0000000", "0000000", 0,
"0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1133061", "Dibujo
Mecánico Asistido por Computadora", "SC", "ST", "SE", "1112013", "0000000",
"0000000", "0000000", 150, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1151045", "Temas
Selectos de Inteligencia Artificial", "SC", "ST", "SE", "1151062", "0000000",
"0000000", "0000000", 0, "0000000", "NA", 9));

```

```

        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1151062",
"Inteligencia Artificial", "SC", "ST", "SE", "1151042", "1153001", "0000000",
"0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1151063",
"Inteligencia Computacional", "SC", "ST", "SE", "1151042", "1153001",
"0000000", "0000000", 0, "0000000", "NA", 9));

        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1112005", "Cálculo de
Varias Variables", "SC", "ST", "SE", "1112029", "1112013", "0000000",
"0000000", 0, "0000000", "NA", 12));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1112016", "Variable
Compleja", "SC", "ST", "SE", "1112005", "0000000", "0000000", "0000000", 0,
"0000000", "NA", 6));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1112017",
"Introducción al Álgebra Lineal", "SC", "ST", "SE", "1151038", "0000000",
"0000000", "0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1112022", "Lógica",
"SC", "ST", "SE", "1151038", "0000000", "0000000", "0000000", 0, "0000000",
"NA", 6));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1112032",
"Introducción a las Ecuaciones Diferenciales Parciales ", "SC", "ST", "SE",
"0000000", "0000000", "0000000", "0000000", 0, "1112030", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1112033", "Matemáticas
Discretas", "SC", "ST", "SE", "1151038", "0000000", "0000000", "0000000", 0,
"0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1112036",
"Criptografía", "SC", "ST", "SE", "1151040", "0000000", "0000000", "0000000",
0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1121028", "Electrónica
Digital", "SC", "ST", "SE", "1123041", "1123046", "0000000", "0000000", 0,
"0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1123057", "Diseño
Electrónico Asistido por Computadora", "SC", "ST", "SE", "1123040", "0000000",
"0000000", "0000000", 0, "0000000", "NA", 6));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1131070", "Circuitos
Eléctricos de Corriente Alterna", "SC", "ST", "SE", "1124001", "1124005",
"0000000", "0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1131071", "Laboratorio
de Circuitos Eléctricos de Corriente Alterna", "SC", "ST", "SE", "0000000",
"0000000", "0000000", 0, "1131070", "NA", 3));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1133048", "Mediciones
en Ingeniería", "SC", "ST", "SE", "1153001", "0000000", "0000000", "0000000",
0, "0000000", "NA", 6));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1151040", "Análisis y
Diseño de Algoritmos", "SC", "ST", "SE", "1152001", "1151042", "1112033",
"0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1151044",
"Programación Orientada a Objetos", "SC", "ST", "SE", "1151038", "0000000",
"0000000", "0000000", 0, "0000000", "NA", 8));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1151051", "Gráficas
por Computadora", "SC", "ST", "SE", "1151042", "1112017", "0000000", "0000000",
0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1151072", "Laboratorio
de Programación Orientada a Objetos", "SC", "ST", "SE", "1151044", "0000000",
"0000000", "0000000", 0, "0000000", "NA", 3));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1152001",
"Investigación de Operaciones I", "SC", "ST", "SE", "1151039", "0000000",
"0000000", "0000000", 0, "0000000", "NA", 9));
        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1152002",
"Investigación de Operaciones II", "SC", "ST", "SE", "1152001", "1153001",
"0000000", "0000000", 0, "0000000", "NA", 9));

```

```

        this.UEAS_opt_tronco_int.add(new PlanEstudios(1, "1155006", "Análisis de
Problemas en Ingeniería", "SC", "ST", "SE", "0000000", "0000000", "0000000",
"0000000", 300, "0000000", "NA", 9));

        this.UEAS = new ArrayList<PlanEstudios>(100);
        this.UEAS.add(new PlanEstudios(1, "1113084", "Estructura Atómica y
Enlace Químico", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000",
0, "0000000", "NA", 9));
        this.UEAS.add(new PlanEstudios(2, "1113085", "Laboratorio de Reacciones
Químicas", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000", 0,
"1113084", "NA", 3));
        this.UEAS.add(new PlanEstudios(3, "1112026", "Taller de Matemáticas",
"SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000", 0, "0000000",
"NA", 7));
        this.UEAS.add(new PlanEstudios(4, "1201008", "Comprensión de Textos",
"SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000", 0, "0000000",
"NA", 4));
        this.UEAS.add(new PlanEstudios(5, "1100033", "Inducción a la Vida
Universitaria", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000",
0, "0000000", "NA", 3));
        this.UEAS.add(new PlanEstudios(6, "1111078", "Introducción a la Física",
"SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000", 0, "0000000",
"NA", 4));

        this.UEAS.add(new PlanEstudios(7, "1113086", "Estructura y Propiedades
de los Materiales en Ingeniería", "SC", "ST", "SE", "1113084", "0000000",
"0000000", "0000000", 0, "0000000", "NA", 6));
        this.UEAS.add(new PlanEstudios(8, "1113087", "Laboratorio de Estructura
y Propiedades de los Materiales", "SC", "ST", "SE", "1113085", "0000000",
"0000000", "0000000", 0, "1113086", "NA", 3));
        this.UEAS.add(new PlanEstudios(9, "1112013", "Complementos de
Matemáticas", "SC", "ST", "SE", "1112026", "0000000", "0000000", "0000000", 0,
"0000000", "NA", 9));
        this.UEAS.add(new PlanEstudios(10, "1112027", "Introducción al Cálculo",
"SC", "ST", "SE", "1112026", "0000000", "0000000", "0000000", 0, "0000000",
"NA", 6));
        this.UEAS.add(new PlanEstudios(11, "1111079", "Cinemática y Dinámica de
Partículas", "SC", "ST", "SE", "1111078", "0000000", "0000000", "0000000", 0,
"1112026", "NA", 9));

        this.UEAS.add(new PlanEstudios(12, "1151038", "Programación
Estructurada", "SC", "ST", "SE", "1112013", "1112027", "0000000", "0000000", 0,
"0000000", "NA", 7));
        this.UEAS.add(new PlanEstudios(13, "1112028", "Cálculo Diferencial",
"SC", "ST", "SE", "1112027", "0000000", "0000000", "0000000", 0, "0000000",
"NA", 9));
        this.UEAS.add(new PlanEstudios(14, "1111081", "Dinámica del Cuerpo
Rígido", "SC", "ST", "SE", "1111079", "0000000", "0000000", "0000000", 0,
"0000000", "NA", 9));
        this.UEAS.add(new PlanEstudios(15, "1111092", "Laboratorio de Movimiento
de una Partícula", "SC", "ST", "SE", "1111079", "0000000", "0000000",
"0000000", 0, "0000000", "NA", 3));
        this.UEAS.add(new PlanEstudios(16, "1100037", "Introducción a la
Ingeniería", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000", 50,
"0000000", "NA", 6));

        this.UEAS.add(new PlanEstudios(17, "1151042", "Algoritmos y Estructuras
de Datos", "SC", "ST", "SE", "1151038", "0000000", "0000000", "0000000", 0,
"0000000", "NA", 8));

```



```

        this.UEAS.add(new PlanEstudios(18, "1113046", "Termodinámica", "SC",
"ST", "SE", "1112028", "1111081", "0000000", "0000000", 0, "0000000", "NA",
6));
        this.UEAS.add(new PlanEstudios(19, "1112029", "Cálculo Integral", "SC",
"ST", "SE", "1112028", "0000000", "0000000", "0000000", 0, "0000000", "NA",
9));
        this.UEAS.add(new PlanEstudios(20, "1111083", "Introducción a la
Electrostática y Magnetostática", "SC", "ST", "SE", "1111081", "0000000",
"0000000", "0000000", 0, "1112029", "NA", 9));
        this.UEAS.add(new PlanEstudios(21, "1111093", "Laboratorio del Cuerpo
Rígido y Oscilaciones", "SC", "ST", "SE", "1111081", "1111092", "0000000",
"0000000", 0, "0000000", "NA", 3));

        this.UEAS.add(new PlanEstudios(22, "1121037", "Diseño lógico", "SC",
"ST", "SE", "1151038", "0000000", "0000000", "0000000", 0, "0000000", "NA",
12));
        this.UEAS.add(new PlanEstudios(23, "1121040", "Laboratorio de diseño
lógico", "SC", "ST", "SE", "1151038", "0000000", "0000000", "0000000", 0,
"1121037", "NA", 6));
        this.UEAS.add(new PlanEstudios(24, "1151039", "Métodos Numéricos en
Ingeniería", "SC", "ST", "SE", "1151038", "0000000", "0000000", "0000000", 0,
"1112029", "NA", 7));
        this.UEAS.add(new PlanEstudios(25, "1112030", "Ecuaciones Diferenciales
Ordinarias", "SC", "ST", "SE", "1112029", "0000000", "0000000", "0000000", 0,
"0000000", "NA", 9));
        this.UEAS.add(new PlanEstudios(26, "1124001", "Circuitos Eléctricos I",
"SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000", 0, "1112030",
"NA", 9));

        this.UEAS.add(new PlanEstudios(27, "1153001", "Probabilidad y
Estadística", "SC", "ST", "SE", "1112029", "0000000", "0000000", "0000000", 0,
"0000000", "NA", 9));
        this.UEAS.add(new PlanEstudios(28, "1111090", "Inducción y Ondas
Electromagnéticas", "SC", "ST", "SE", "1111083", "1112030", "0000000",
"0000000", 0, "0000000", "NA", 9));
        this.UEAS.add(new PlanEstudios(29, "1112040", "Transformada de Laplace y
Análisis de Fourier", "SC", "ST", "SE", "1112030", "0000000", "0000000",
"0000000", 0, "0000000", "NA", 9));
        this.UEAS.add(new PlanEstudios(30, "1124005", "Laboratorio de Circuitos
Eléctricos I", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000", 0,
"1124001", "NA", 3));
        this.UEAS.add(new PlanEstudios(31, "1122025", "Radiación y Propagación",
"SC", "ST", "SE", "1111083", "1112030", "0000000", "0000000", 0, "0000000",
"NA", 9));
        this.UEAS.add(new PlanEstudios(53, "110XXX1", "Optativa Inter y
Multidisciplinar I", "SC", "ST", "SE", "0000000", "0000000", "0000000",
"0000000", 150, "0000000", "NA", 6));

        this.UEAS.add(new PlanEstudios(32, "1121060", "Microprocesadores", "SC",
"ST", "SE", "1121037", "1151042", "0000000", "0000000", 0, "0000000", "NA",
9));
        this.UEAS.add(new PlanEstudios(33, "1123040", "Circuitos Electrónicos
I", "SC", "ST", "SE", "1124001", "1124005", "0000000", "0000000", 0, "0000000",
"NA", 9));
        this.UEAS.add(new PlanEstudios(34, "1123045", "Laboratorio de Circuitos
Electrónicos I", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000",
0, "1123040", "NA", 3));
        this.UEAS.add(new PlanEstudios(35, "1124003", "Circuitos Eléctricos II",
"SC", "ST", "SE", "1124001", "1112040", "0000000", "0000000", 0, "0000000",
"NA", 9));
        this.UEAS.add(new PlanEstudios(36, "1122016", "Introducción a las
Comunicaciones", "SC", "ST", "SE", "1122025", "1124001", "0000000", "0000000",
0, "0000000", "NA", 9));

```

```

this.UEAS.add(new PlanEstudios(53, "110XXX2", "Optativa Inter y
Multidisciplinar II", "SC", "ST", "SE", "0000000", "0000000", "0000000",
"0000000", 150, "0000000", "NA", 6));

this.UEAS.add(new PlanEstudios(37, "1121038", "Fundamentos de Redes de
Computadoras", "SC", "ST", "SE", "1112040", "1121060", "0000000", "0000000", 0,
"0000000", "NA", 9));
this.UEAS.add(new PlanEstudios(38, "1123041", "Circuitos Electrónicos
II", "SC", "ST", "SE", "1123040", "1123045", "0000000", "0000000", 0,
"0000000", "NA", 9));
this.UEAS.add(new PlanEstudios(39, "1122009", "Análisis de Señales",
"SC", "ST", "SE", "1153001", "1112040", "0000000", "0000000", 0, "0000000",
"NA", 9));
this.UEAS.add(new PlanEstudios(40, "1124048", "Laboratorio de Circuitos
Eléctricos II", "SC", "ST", "SE", "1124005", "0000000", "0000000", "0000000",
0, "1124003", "NA", 3));
this.UEAS.add(new PlanEstudios(41, "1122018", "Laboratorio de
Introducción a las Comunicaciones", "SC", "ST", "SE", "0000000", "0000000",
"0000000", "0000000", 0, "1122016", "NA", 3));
this.UEAS.add(new PlanEstudios(42, "1100038", "Introducción al
Desarrollo Sustentable", "SC", "ST", "SE", "1100037", "0000000", "0000000",
"0000000", 0, "0000000", "NA", 6));
this.UEAS.add(new PlanEstudios(53, "110XXX3", "Optativa Inter y
Multidisciplinar III", "SC", "ST", "SE", "0000000", "0000000", "0000000",
"0000000", 150, "0000000", "NA", 6));

this.UEAS.add(new PlanEstudios(43, "1121061", "Laboratorio de
Microprocesadores", "SC", "ST", "SE", "1121040", "0000000", "0000000",
"0000000", 0, "1121060", "NA", 3));
this.UEAS.add(new PlanEstudios(44, "1121034", "Sistemas Digitales con
Microcontroladores", "SC", "ST", "SE", "1121060", "1123040", "1151039",
"0000000", 0, "0000000", "NA", 9));
this.UEAS.add(new PlanEstudios(45, "1122014", "Comunicaciones
Analógicas", "SC", "ST", "SE", "1112040", "1122016", "1122018", "1123041", 0,
"0000000", "NA", 9));
this.UEAS.add(new PlanEstudios(46, "1123046", "Laboratorio de Circuitos
Electrónicos II", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000",
0, "1123041", "NA", 3));
this.UEAS.add(new PlanEstudios(47, "1124050", "Teoría de Control", "SC",
"ST", "SE", "1124003", "0000000", "0000000", "0000000", 0, "0000000", "NA",
9));
this.UEAS.add(new PlanEstudios(48, "1100040", "Taller de Planeación y
Ejecución de Proyectos", "SC", "ST", "SE", "0000000", "0000000", "0000000",
"0000000", 280, "0000000", "NA", 6));
this.UEAS.add(new PlanEstudios(63, "XXXXXX1", "Optativas del Tronco de
Integración I", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000",
300, "0000000", "NA", 9));
this.UEAS.add(new PlanEstudios(63, "XXXXXX2", "Optativas del Tronco de
Integración II", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000",
300, "0000000", "NA", 6));

this.UEAS.add(new PlanEstudios(49, "1123043", "Diseño de Sistemas
Electrónicos", "SC", "ST", "SE", "1123041", "1123046", "1121034", "0000000", 0,
"0000000", "NA", 9));
this.UEAS.add(new PlanEstudios(50, "1121032", "Laboratorio de Sistemas
Digitales con Microcontroladores", "SC", "ST", "SE", "0000000", "0000000",
"0000000", "0000000", 0, "1121034", "NA", 3));
this.UEAS.add(new PlanEstudios(51, "1122017", "Laboratorio de
Comunicaciones Analógicas", "SC", "ST", "SE", "0000000", "0000000", "0000000",
"0000000", 0, "1122014", "NA", 3));
this.UEAS.add(new PlanEstudios(52, "1122012", "Procesamiento Digital de
Señales", "SC", "ST", "SE", "1122009", "0000000", "0000000", "0000000", 0,
"0000000", "NA", 9));

```

```

        this.UEAS.add(new PlanEstudios(54, "1124049", "Laboratorio de Control",
"SC", "ST", "SE", "1121037", "0000000", "0000000", "0000000", 0, "1124050",
"NA", 3));
        this.UEAS.add(new PlanEstudios(55, "1100039", "Innovación", "SC", "ST",
"SE", "1100040", "0000000", "0000000", "0000000", 0, "0000000", "NA", 6));
        this.UEAS.add(new PlanEstudios(63, "XXXXXXX3", "Optativas del Tronco de
Integración III", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000",
300, "0000000", "NA", 9));
        this.UEAS.add(new PlanEstudios(63, "XXXXXXX4", "Optativas del Tronco de
Integración IV", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000",
300, "0000000", "NA", 6));

        this.UEAS.add(new PlanEstudios(56, "1123048", "Laboratorio de Diseño de
Sistemas Electrónicos", "SC", "ST", "SE", "0000000", "0000000", "0000000",
"0000000", 0, "1123043", "NA", 3));
        this.UEAS.add(new PlanEstudios(57, "1122015", "Comunicaciones
Digitales", "SC", "ST", "SE", "1122014", "1122017", "0000000", "0000000", 0,
"0000000", "NA", 9));
        this.UEAS.add(new PlanEstudios(58, "1122032", "Laboratorio de
Comunicaciones Digitales", "SC", "ST", "SE", "0000000", "0000000", "0000000",
"0000000", 0, "1122015", "NA", 3));
        this.UEAS.add(new PlanEstudios(59, "1124045", "Control Digital", "SC",
"ST", "SE", "1123043", "1124049", "1122012", "0000000", 0, "0000000", "NA",
9));
        this.UEAS.add(new PlanEstudios(63, "XXXXXXX5", "Optativas del Tronco de
Integración V", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000",
300, "0000000", "NA", 6));
        this.UEAS.add(new PlanEstudios(61, "1100105", "Seminario de Integración
en Ingeniería Electrónica", "SC", "ST", "SE", "1100039", "0000000", "0000000",
"0000000", 360, "0000000", "NA", 3));
        this.UEAS.add(new PlanEstudios(62, "1100041", "Retos del Desarrollo
Nacional", "SC", "ST", "SE", "1100039", "0000000", "0000000", "0000000", 0,
"0000000", "NA", 6));

        this.UEAS.add(new PlanEstudios(65, "XXXXXXX6", "Optativas del Tronco de
Integración VI", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000",
340, "0000000", "NA", 9));
        this.UEAS.add(new PlanEstudios(65, "XXXXXXX7", "Optativas del Tronco de
Integración VII", "SC", "ST", "SE", "0000000", "0000000", "0000000", "0000000",
340, "0000000", "NA", 9));
        this.UEAS.add(new PlanEstudios(60, "110XXX4", "Optativa Inter y
Multidisciplinar", "SC", "ST", "SE", "0000000", "0000000", "0000000",
"0000000", 150, "0000000", "NA", 6));
        this.UEAS.add(new PlanEstudios(67, "1100115", "Proyecto de Integración
en Ingeniería Electrónica I", "SC", "ST", "SE", "1100105", "0000000",
"0000000", "0000000", 0, "0000000", "NA", 18));
    }
    public void carga_kardex () throws IOException {
        List<String> trimestres = new ArrayList<String>(22);
        UEAS_cursadas = new ArrayList<PlanEstudios>(100);
        Scanner scan = new Scanner(this.kardex.toString());
        String line = "anything";
        String tiene_clave = null;
        String tiene_calif = null;
        String tiene_trim = null;
        String tiene_eva = null;
        while (scan.hasNextLine() ) {
            line = scan.nextLine();
            String clave = interpretHtml(line, "CVE_UEA_CL", 32, "<");
            String trim = interpretHtml(line, "TRIMESTRE_XX.TRIMESTRE_LEC",
0, "<");

```

```

String eva = interpretHtml(line, "TIPO_EVAL_XX.TIPO_EVALUA", 0,
"<");
String calif = interpretHtml(line, "CALIFICA_CL", 32, "<");

if(null != clave){
    tiene_clave = clave;
    //UEAS.contains(new PlanEstudios(1, Integer.parseInt(clave),
"Inducción a la Vida Universitaria", "SC", "ST", "SE", 0000000, 0, 0000000,
"NA", 3));
}
if(null != calif){
    tiene_calif = calif;
    //UEAS.contains(new PlanEstudios(1, Integer.parseInt(clave),
"Inducción a la Vida Universitaria", "SC", "ST", "SE", 0000000, 0, 0000000,
"NA", 3));
}
if(null != trim){
    tiene_trim = trim;
    if (!trimestres.contains(tiene_trim)) {
        trimestres.add(tiene_trim);
    }
    //UEAS.contains(new PlanEstudios(1, Integer.parseInt(clave),
"Inducción a la Vida Universitaria", "SC", "ST", "SE", 0000000, 0, 0000000,
"NA", 3));
}
if(null != eva){
    tiene_eva = eva;
    //UEAS.contains(new PlanEstudios(1, Integer.parseInt(clave),
"Inducción a la Vida Universitaria", "SC", "ST", "SE", 0000000, 0, 0000000,
"NA", 3));
}
if(tiene_clave != null && tiene_calif != null && tiene_trim !=
null && tiene_eva != null){
    for (int i = 0; i < this.UEAS.size(); i++) {
        if (this.UEAS.get(i).getClave().equals(tiene_clave))
        {
this.UEAS.get(i).setCalificacion(tiene_calif);
                this.UEAS.get(i).setTrimestre(tiene_trim);

this.UEAS.get(i).setTipo_evaluacion(tiene_eva);
                tiene_clave = null;
                tiene_calif = null;
                tiene_trim = null;
                tiene_eva = null;
                if
(!this.UEAS.get(i).getCalificacion().equals("NA")) {
                    this.UEAS.get(i).setCursada(true);
                    UEAS_cursadas.add(this.UEAS.get(i));
                    break;
                }
        }
    }
    for (int i = 0; i < this.UEAS_opt_inter_mult.size(); i++)
    {
        if
(this.UEAS_opt_inter_mult.get(i).getClave().equals(tiene_clave)) {
this.UEAS_opt_inter_mult.get(i).setCalificacion(tiene_calif);

```

```

this.UEAS_opt_inter_mult.get(i).setTrimestre(tiene_trim);

this.UEAS_opt_inter_mult.get(i).setTipo_evaluacion(tiene_eva);
    tiene_clave = null;
    tiene_calif = null;
    tiene_trim = null;
    tiene_eva = null;

    if
(!this.UEAS_opt_inter_mult.get(i).getCalificacion().equals("NA")) {

this.UEAS_opt_inter_mult.get(i).setCursada(true);

UEAS_cursadas.add(this.UEAS_opt_inter_mult.get(i));
    break;
    }

    }
    }
    for (int i = 0; i < this.UEAS_opt_tronco_int.size(); i++)
{
    if
(this.UEAS_opt_tronco_int.get(i).getClave().equals(tiene_clave)) {

this.UEAS_opt_tronco_int.get(i).setCalificacion(tiene_calif);

this.UEAS_opt_tronco_int.get(i).setTrimestre(tiene_trim);

this.UEAS_opt_tronco_int.get(i).setTipo_evaluacion(tiene_eva);
    tiene_clave = null;
    tiene_calif = null;
    tiene_trim = null;
    tiene_eva = null;
    if
(!this.UEAS_opt_tronco_int.get(i).getCalificacion().equals("NA")) {

this.UEAS_opt_tronco_int.get(i).setCursada(true);

UEAS_cursadas.add(this.UEAS_opt_tronco_int.get(i));
    break;
    }

    }

    }

//UEAS.contains(new PlanEstudios(1,
Integer.parseInt(tiene_clave), tiene_nombre, "SC", "ST", "SE", 0000000, 0,
0000000, "NA", 3));
    }

}

public void determina_opt_inter_mult () {
    if (cred_opt_inter_mult >= 6 && cred_opt_inter_mult < 12) {
        for (int i = 0; i < this.UEAS.size(); i++) {
            if (this.UEAS.get(i).getClave().equals("110XXX1")) {
                this.UEAS.get(i).setCursada(true);
            }
        }
    }
}

```

```

        }
    }
}

if (cred_opt_inter_mult >= 12 && cred_opt_inter_mult < 18) {
    for (int i = 0; i < this.UEAS.size(); i++) {
        if (this.UEAS.get(i).getClave().equals("110XXX1")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("110XXX2")) {
            this.UEAS.get(i).setCursada(true);
        }
    }
}

if (cred_opt_inter_mult >= 18 && cred_opt_inter_mult < 24) {
    for (int i = 0; i < this.UEAS.size(); i++) {
        if (this.UEAS.get(i).getClave().equals("110XXX1")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("110XXX2")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("110XXX3")) {
            this.UEAS.get(i).setCursada(true);
        }
    }
}

if (cred_opt_inter_mult >= 24) {
    for (int i = 0; i < this.UEAS.size(); i++) {
        if (this.UEAS.get(i).getClave().equals("110XXX1")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("110XXX2")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("110XXX3")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("110XXX4")) {
            this.UEAS.get(i).setCursada(true);
        }
    }
}

}

public void determina_opt_tronco_int () {

    if (cred_opt_tronco_int >= 6 && cred_opt_tronco_int < 9) {
        for (int i = 0; i < this.UEAS.size(); i++) {

```

```

        if (this.UEAS.get(i).getClave().equals("XXXXXX2")) {
            this.UEAS.get(i).setCursada(true);
        }
    }
}

if (cred_opt_tronco_int >= 9 && cred_opt_tronco_int < 12) {
    for (int i = 0; i < this.UEAS.size(); i++) {
        if (this.UEAS.get(i).getClave().equals("XXXXXX1")) {
            this.UEAS.get(i).setCursada(true);
        }
    }
}

if (cred_opt_tronco_int >= 12 && cred_opt_tronco_int < 18) {
    for (int i = 0; i < this.UEAS.size(); i++) {
        if (this.UEAS.get(i).getClave().equals("XXXXXX2")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXX4")) {
            this.UEAS.get(i).setCursada(true);
        }
    }
}

if (cred_opt_tronco_int >= 18 && cred_opt_tronco_int < 24) {
    for (int i = 0; i < this.UEAS.size(); i++) {
        if (this.UEAS.get(i).getClave().equals("XXXXXX1")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXX3")) {
            this.UEAS.get(i).setCursada(true);
        }
    }
}

if (cred_opt_tronco_int >= 24 && cred_opt_tronco_int < 27) {
    for (int i = 0; i < this.UEAS.size(); i++) {
        if (this.UEAS.get(i).getClave().equals("XXXXXX1")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXX3")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXX2")) {
            this.UEAS.get(i).setCursada(true);
        }
    }
}
}

```

```

}

if (cred_opt_tronco_int >= 27 && cred_opt_tronco_int < 30) {
    for (int i = 0; i < this.UEAS.size(); i++) {
        if (this.UEAS.get(i).getClave().equals("XXXXXXX1")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXXX3")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXXX6")) {
            this.UEAS.get(i).setCursada(true);
        }
    }
}

}

if (cred_opt_tronco_int >= 30 && cred_opt_tronco_int < 36) {
    for (int i = 0; i < this.UEAS.size(); i++) {
        if (this.UEAS.get(i).getClave().equals("XXXXXXX1")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXXX3")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXXX2")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXXX4")) {
            this.UEAS.get(i).setCursada(true);
        }
    }
}

}

if (cred_opt_tronco_int >= 36 && cred_opt_tronco_int < 45) {
    for (int i = 0; i < this.UEAS.size(); i++) {
        if (this.UEAS.get(i).getClave().equals("XXXXXXX1")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXXX3")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXXX2")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXXX4")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXXX5")) {
            this.UEAS.get(i).setCursada(true);
        }
    }
}

```



```

    }
}

if (cred_opt_tronco_int >= 45 && cred_opt_tronco_int < 54) {
    for (int i = 0; i < this.UEAS.size(); i++) {
        if (this.UEAS.get(i).getClave().equals("XXXXXXX1")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXXX3")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXXX2")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXXX4")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXXX5")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXXX6")) {
            this.UEAS.get(i).setCursada(true);
        }
    }
}

if (cred_opt_tronco_int >= 54) {
    for (int i = 0; i < this.UEAS.size(); i++) {
        if (this.UEAS.get(i).getClave().equals("XXXXXXX1")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXXX3")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXXX2")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXXX4")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXXX5")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXXX6")) {
            this.UEAS.get(i).setCursada(true);
        }
        if (this.UEAS.get(i).getClave().equals("XXXXXXX7")) {
            this.UEAS.get(i).setCursada(true);
        }
    }
}

```

```

        }
    }
}

public void cred_3_ult_trim() {
    char est = this.Ult_trimestre.charAt(2);

    String año = this.Ult_trimestre.substring(0, 2);

    int trim_act = 0;
    String se_trim = "-";
    String te_trim = "-";
    switch (est) {
        case 'I':
            trim_act = Integer.parseInt(año);
            se_trim = Integer.toString(trim_act - 1) + "O";
            te_trim = Integer.toString(trim_act - 1) + "P";
            for (int i = 0; i < this.UEAS_cursadas.size(); i++) {
                if (this.UEAS_cursadas.get(i).getCursada() == true)
                {
                    if
                    (this.UEAS_cursadas.get(i).getTrimestre().equals(this.Ult_trimestre)) {
                        this.prim_trim_cred +=
                    this.UEAS_cursadas.get(i).getCreditos();
                    }
                    if
                    (this.UEAS_cursadas.get(i).getTrimestre().equals(se_trim)) {
                        seg_trim_cred +=
                    this.UEAS_cursadas.get(i).getCreditos();
                    }
                    if
                    (this.UEAS_cursadas.get(i).getTrimestre().equals(te_trim)) {
                        ter_trim_cred +=
                    this.UEAS_cursadas.get(i).getCreditos();
                    }
                }
            }
            break;
        case 'P':
            trim_act = Integer.parseInt(año);
            se_trim = Integer.toString(trim_act) + "I";

            te_trim = Integer.toString(trim_act - 1) + "O";

            for (int i = 0; i < this.UEAS_cursadas.size(); i++) {
                if (this.UEAS_cursadas.get(i).getCursada() == true)
                {
                    if
                    (this.UEAS_cursadas.get(i).getTrimestre().equals(this.Ult_trimestre)) {
                        prim_trim_cred +=
                    this.UEAS_cursadas.get(i).getCreditos();
                    }
                }
            }
        }
    }
}

```

```

        if
        (this.UEAS_cursadas.get(i).getTrimestre().equals(se_trim)) {
            seg_trim_cred +=
this.UEAS_cursadas.get(i).getCreditos();

        }
        if
        (this.UEAS_cursadas.get(i).getTrimestre().equals(te_trim)) {
            ter_trim_cred +=
this.UEAS_cursadas.get(i).getCreditos();

        }
    }
}
break;
case '0':
    trim_act = Integer.parseInt(año);

    se_trim = año + "P";
    te_trim = año + "I";
    for (int i = 0; i < this.UEAS_cursadas.size(); i++) {
        if (this.UEAS_cursadas.get(i).getCursada() == true)
        {
            if
            (this.UEAS_cursadas.get(i).getTrimestre().equals(this.Ult_trimestre)) {
                prim_trim_cred +=
this.UEAS_cursadas.get(i).getCreditos();

            }
            if
            (this.UEAS_cursadas.get(i).getTrimestre().equals(se_trim)) {
                seg_trim_cred +=
this.UEAS_cursadas.get(i).getCreditos();

            }
            if
            (this.UEAS_cursadas.get(i).getTrimestre().equals(te_trim)) {
                ter_trim_cred +=
this.UEAS_cursadas.get(i).getCreditos();

            }
        }
    }
}
break;
default:
break;

}

}

public void determina_creditos () {
    if (!this.renuncias.equals("0")) {
        this.limite_creditos = 40;
    }
    else {
        this.limite_creditos = 63;
    }
}

}

public void determina_UEAS_posibles () {

```

```

        this.UEAS_posibles = new ArrayList<PlanEstudios>(10);
        for (int i = 0; i < this.UEAS.size(); i++) {
            if (!this.UEAS.get(i).getCursada()) {
                if (this.UEAS.get(i).getSeriacion_credits().intValue() <=
this.credits.intValue()) {
                    if (ya_curso(this.UEAS.get(i).getSeriacion_UEA_1()))
{
                        if
(ya_curso(this.UEAS.get(i).getSeriacion_UEA_2())) {
                            if
(ya_curso(this.UEAS.get(i).getSeriacion_UEA_3())) {
                                if
(ya_curso(this.UEAS.get(i).getSeriacion_UEA_4())) {
                                    if
(ya_curso(this.UEAS.get(i).getSeriacion_corregistro()) || corregistro_posible
(this.UEAS.get(i).getSeriacion_corregistro())) {
this.UEAS_posibles.add(this.UEAS.get(i));
                                }
                            }
                        }
                    }
                }
            }
        }

    }

    public void determina_UEAS_opt_inter_mult_cursadas () {
        for (int i = 0; i < this.UEAS_opt_inter_mult.size(); i++) {
            if (ya_curso(this.UEAS_opt_inter_mult.get(i).getClave())) {
                this.UEAS_opt_inter_mult.get(i).setCursada(true);
            }
        }
    }

    public void determina_UEAS_opt_tronco_int_cursadas () {
        for (int i = 0; i < this.UEAS_opt_tronco_int.size(); i++) {
            if (ya_curso(this.UEAS_opt_tronco_int.get(i).getClave())) {
                this.UEAS_opt_tronco_int.get(i).setCursada(true);
            }
        }
    }

    public void determina_UEAS_opt_inter_mult_posibles () {
        this.UEAS_opt_inter_mult_posibles = new ArrayList<PlanEstudios>(10);
        for (int i = 0; i < this.UEAS_opt_inter_mult.size(); i++) {
            if (!this.UEAS_opt_inter_mult.get(i).getCursada()) {
                if
(this.UEAS_opt_inter_mult.get(i).getSeriacion_credits().intValue() <=
this.credits.intValue()) {
                    if
(ya_curso(this.UEAS_opt_inter_mult.get(i).getSeriacion_UEA_1())) {

```



```

        }

    }

}

public boolean ya_curso (String clave) {
    for (int j = 0; j < this.UEAS_cursadas.size(); j++) {
        if (this.UEAS_cursadas.get(j).getCursada()) {
            if (this.UEAS_cursadas.get(j).getClave().equals(clave) ||
clave.equals("0000000")) {
                return true;
            }
        }
    }
    return false;
}

public boolean corregistro_posible (String clave) {
    for (int i = 0; i < this.UEAS_cursadas.size(); i++) {
        if (!this.UEAS_cursadas.get(i).getCursada()) {
            if (this.UEAS_cursadas.get(i).getClave().equals(clave)) {
                if
(this.UEAS_cursadas.get(i).getSeriacion_creditos().intValue() <=
this.creditos.intValue()) {
                    if
(ya_curso(this.UEAS_cursadas.get(i).getSeriacion_UEA_1())) {
                        if
(ya_curso(this.UEAS_cursadas.get(i).getSeriacion_UEA_2())) {
                            if
(ya_curso(this.UEAS_cursadas.get(i).getSeriacion_UEA_3())) {

                                if
(ya_curso(this.UEAS_cursadas.get(i).getSeriacion_UEA_4())) {
                                    if
(ya_curso(this.UEAS_cursadas.get(i).getSeriacion_corregistro())) {
                                        return true;
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
    return false;
}

private String interpretHtml(String s, String cadena, int inicio, String
carac){
    if (s == null)

```

```

        return null;
    if(s.contains(cadena)){
        StringTokenizer tokens = new StringTokenizer(s.substring(inicio,
s.length()), carac);
        String contenido = tokens.nextToken();
        return contenido;
    }
    else
        return null;
}

    public void imprime_UEAS_posibles() {
        for (int i = 0; i < this.UEAS_posibles.size(); i++) {

System.out.println(this.UEAS_posibles.get(i).getPrioridad());
                System.out.println(this.UEAS_posibles.get(i).getClave());

System.out.println(this.UEAS_posibles.get(i).getNombre_UEA());
System.out.println(this.UEAS_posibles.get(i).getCalificacion());
System.out.println(this.UEAS_posibles.get(i).getTrimestre());
System.out.println(this.UEAS_posibles.get(i).getTipo_evaluacion());
System.out.println(this.UEAS_posibles.get(i).getSeriacion_UEA_1());
System.out.println(this.UEAS_posibles.get(i).getSeriacion_UEA_2());
System.out.println(this.UEAS_posibles.get(i).getSeriacion_UEA_3());
System.out.println(this.UEAS_posibles.get(i).getSeriacion_UEA_4());
System.out.println(this.UEAS_posibles.get(i).getSeriacion_creditos());
System.out.println(this.UEAS_posibles.get(i).getSeriacion_corregistro());
System.out.println(this.UEAS_posibles.get(i).getSeriacion_autorizacion());
System.out.println(this.UEAS_posibles.get(i).getCursada());
System.out.println(this.UEAS_posibles.get(i).getCreditos());
                System.out.println("---");

        }

    }

    public void imprime_UEAS_cursadas() {
        for (int i = 0; i < this.UEAS_cursadas.size(); i++) {
            if (this.UEAS_cursadas.get(i).getCursada() == true) {

System.out.println(this.UEAS_cursadas.get(i).getPrioridad());
                System.out.println(this.UEAS_cursadas.get(i).getClave());

System.out.println(this.UEAS_cursadas.get(i).getNombre_UEA());
System.out.println(this.UEAS_cursadas.get(i).getCalificacion());
System.out.println(this.UEAS_cursadas.get(i).getTrimestre());
            }
        }
    }

```

```

System.out.println(this.UEAS_cursadas.get(i).getTipo_evaluacion());
System.out.println(this.UEAS_cursadas.get(i).getSeriacion_UEA_1());
System.out.println(this.UEAS_cursadas.get(i).getSeriacion_UEA_2());
System.out.println(this.UEAS_cursadas.get(i).getSeriacion_UEA_3());
System.out.println(this.UEAS_cursadas.get(i).getSeriacion_UEA_4());
System.out.println(this.UEAS_cursadas.get(i).getSeriacion_creditos());
System.out.println(this.UEAS_cursadas.get(i).getSeriacion_corregistro());
System.out.println(this.UEAS_cursadas.get(i).getSeriacion_autorizacion());
System.out.println(this.UEAS_cursadas.get(i).getCursada());
System.out.println(this.UEAS_cursadas.get(i).getCreditos());
                System.out.println("---");
            }
        }
    }
    public Integer getCreditos() {
        return creditos;
    }
    public void setCreditos(Integer creditos) {
        this.creditos = creditos;
    }
    public StringBuilder getKardex() {
        return kardex;
    }
    public void setKardex(StringBuilder kardex) {
        this.kardex = kardex;
    }
    public List<PlanEstudios> getUEAS() {
        return UEAS;
    }
    public void setUEAS(List<PlanEstudios> uEAS) {
        UEAS = uEAS;
    }
    public List<PlanEstudios> getUEAS_posibles() {
        return UEAS_posibles;
    }
    public void setUEAS_posibles(List<PlanEstudios> uEAS_posibles) {
        UEAS_posibles = uEAS_posibles;
    }
    public List<PlanEstudios> getUEAS_cursadas() {
        return UEAS_cursadas;
    }
    public void setUEAS_cursadas(List<PlanEstudios> uEAS_cursadas) {
        UEAS_cursadas = uEAS_cursadas;
    }
    public List<PlanEstudios> getUEAS_opt_inter_mult() {
        return UEAS_opt_inter_mult;
    }
    public void setUEAS_opt_inter_mult(List<PlanEstudios>
uEAS_opt_inter_mult) {
        UEAS_opt_inter_mult = uEAS_opt_inter_mult;
    }
}

```



```

    public List<PlanEstudios> getUEAS_opt_tronco_int() {
        return UEAS_opt_tronco_int;
    }
    public void setUEAS_opt_tronco_int(List<PlanEstudios>
uEAS_opt_tronco_int) {
        UEAS_opt_tronco_int = uEAS_opt_tronco_int;
    }
    public List<PlanEstudios> getUEAS_opt_inter_mult_posibles() {
        return UEAS_opt_inter_mult_posibles;
    }
    public void setUEAS_opt_inter_mult_posibles(
        List<PlanEstudios> uEAS_opt_inter_mult_posibles) {
        UEAS_opt_inter_mult_posibles = uEAS_opt_inter_mult_posibles;
    }
    public List<PlanEstudios> getUEAS_opt_tronco_int_posibles() {
        return UEAS_opt_tronco_int_posibles;
    }
    public void setUEAS_opt_tronco_int_posibles(
        List<PlanEstudios> uEAS_opt_tronco_int_posibles) {
        UEAS_opt_tronco_int_posibles = uEAS_opt_tronco_int_posibles;
    }
    public String getRenuncias() {
        return renunciias;
    }
    public void setRenuncias(String renunciias) {
        this.renunciias = renunciias;
    }
    public Integer getLimite_creditos() {
        return limite_creditos;
    }
    public void setLimite_creditos(Integer limite_creditos) {
        this.limite_creditos = limite_creditos;
    }
    public String getUlt_trimestre() {
        return Ult_trimestre;
    }
    public void setUlt_trimestre(String ult_trimestre) {
        Ult_trimestre = ult_trimestre;
    }
    public Integer getPrim_trim_cred() {
        return prim_trim_cred;
    }
    public void setPrim_trim_cred(Integer prim_trim_cred) {
        this.prim_trim_cred = prim_trim_cred;
    }
    public Integer getSeg_trim_cred() {
        return seg_trim_cred;
    }
    public void setSeg_trim_cred(Integer seg_trim_cred) {
        this.seg_trim_cred = seg_trim_cred;
    }
    public Integer getTer_trim_cred() {
        return ter_trim_cred;
    }
    public void setTer_trim_cred(Integer ter_trim_cred) {
        this.ter_trim_cred = ter_trim_cred;
    }
    public Integer getCred_opt_inter_mult() {
        return cred_opt_inter_mult;
    }
    public void setCred_opt_inter_mult(Integer cred_opt_inter_mult) {
        this.cred_opt_inter_mult = cred_opt_inter_mult;
    }
}

```

```

    public Integer getCred_opt_tronco_int() {
        return cred_opt_tronco_int;
    }
    public void setCred_opt_tronco_int(Integer cred_opt_tronco_int) {
        this.cred_opt_tronco_int = cred_opt_tronco_int;
    }
}
}

```

• Clase PlanEstudios

```

package com.uamapps.appingelectronicauam;

import android.os.Parcel;
import android.os.Parcelable;

public class PlanEstudios implements Parcelable {
    private Integer prioridad;
    private String clave;
    private String nombre_UEA;
    private String calificacion;
    private String trimestre;
    private String tipo_evaluacion;
    private String seriacion_UEA_1;
    private String seriacion_UEA_2;
    private String seriacion_UEA_3;
    private String seriacion_UEA_4;
    private Integer seriacion_creditos;
    private String seriacion_corregistro;
    private String seriacion_autorizacion;
    private Boolean cursada;
    private Integer creditos;

    public PlanEstudios(Integer Prioridad, String Clave, String Nombre_UEA,
String Calificacion, String Trimestre, String Tipo_evaluacion, String
Seriacion_UEA_1, String Seriacion_UEA_2, String Seriacion_UEA_3, String
Seriacion_UEA_4, Integer Seriacion_creditos, String Seriacion_corregistro,
String Seriacion_autorizacion, Integer Creditos) {
        this.prioridad = Prioridad;
        this.clave = Clave;
        this.nombre_UEA = Nombre_UEA;
        this.calificacion = Calificacion;
        this.trimestre = Trimestre;
        this.tipo_evaluacion = Tipo_evaluacion;
        this.seriacion_UEA_1 = Seriacion_UEA_1;
        this.seriacion_UEA_2 = Seriacion_UEA_2;
        this.seriacion_UEA_3 = Seriacion_UEA_3;
        this.seriacion_UEA_4 = Seriacion_UEA_4;
        this.seriacion_creditos = Seriacion_creditos;
        this.seriacion_corregistro = Seriacion_corregistro;
        this.seriacion_autorizacion = Seriacion_autorizacion;
        this.cursada = false;
        this.creditos = Creditos;
    }

    private PlanEstudios(Parcel in) {

```

```

        this.prioridad = in.readInt();
        this.clave = in.readString();
        this.nombre_UEA = in.readString();
        this.calificacion = in.readString();
        this.trimestre = in.readString();
        this.tipo_evaluacion = in.readString();
        this.seriacion_UEA_1 = in.readString();
        this.seriacion_UEA_2 = in.readString();
        this.seriacion_UEA_3 = in.readString();
        this.seriacion_UEA_4 = in.readString();
        this.seriacion_creditos = in.readInt();
        this.seriacion_corregistro = in.readString();
        this.seriacion_autorizacion = in.readString();
        this.cursada = in.readByte() == 1;
        this.creditos = in.readInt();
    }

    public void writeToParcel(Parcel out, int flags) {

        out.writeInt(this.prioridad);
        out.writeString(this.clave);
        out.writeString(this.nombre_UEA);
        out.writeString(this.calificacion);
        out.writeString(this.trimestre);
        out.writeString(this.tipo_evaluacion);
        out.writeString(this.seriacion_UEA_1);
        out.writeString(this.seriacion_UEA_2);
        out.writeString(this.seriacion_UEA_3);
        out.writeString(this.seriacion_UEA_4);
        out.writeInt(this.seriacion_creditos);
        out.writeString(this.seriacion_corregistro);
        out.writeString(this.seriacion_autorizacion);
        out.writeByte((byte) (this.cursada ? 1 : 0));
        out.writeInt(this.creditos);
    }

    public int describeContents() {
        return 0;
    }

    public static final Parcelable.Creator<PlanEstudios> CREATOR = new
Parcelable.Creator<PlanEstudios>() {
        public PlanEstudios createFromParcel(Parcel in) {
            return new PlanEstudios(in);
        }

        public PlanEstudios[] newArray(int size) {
            return new PlanEstudios[size];
        }
    };

    public Integer getPrioridad() {
        return prioridad;
    }

    public void setPrioridad(Integer prioridad) {
        this.prioridad = prioridad;
    }

```

```

public String getClave() {
    return clave;
}

public void setClave(String clave) {
    this.clave = clave;
}

public String getNombre_UEA() {
    return nombre_UEA;
}

public void setNombre_UEA(String nombre_UEA) {
    this.nombre_UEA = nombre_UEA;
}

public String getCalificacion() {
    return calificacion;
}

public void setCalificacion(String calificacion) {
    this.calificacion = calificacion;
}

public String getTrimestre() {
    return trimestre;
}

public void setTrimestre(String trimestre) {
    this.trimestre = trimestre;
}

public String getTipo_evaluacion() {
    return tipo_evaluacion;
}

public void setTipo_evaluacion(String tipo_evaluacion) {
    this.tipo_evaluacion = tipo_evaluacion;
}

public String getSeriacion_UEA_1() {
    return seriacion_UEA_1;
}

public void setSeriacion_UEA_1(String seriacion_UEA_1) {
    this.seriacion_UEA_1 = seriacion_UEA_1;
}

public String getSeriacion_UEA_2() {
    return seriacion_UEA_2;
}

public void setSeriacion_UEA_2(String seriacion_UEA_2) {
    this.seriacion_UEA_2 = seriacion_UEA_2;
}

public String getSeriacion_UEA_3() {
    return seriacion_UEA_3;
}

public void setSeriacion_UEA_3(String seriacion_UEA_3) {
    this.seriacion_UEA_3 = seriacion_UEA_3;
}

```

```

    }

    public String getSeriacion_UEA_4() {
        return seriacion_UEA_4;
    }

    public void setSeriacion_UEA_4(String seriacion_UEA_4) {
        this.seriacion_UEA_4 = seriacion_UEA_4;
    }

    public Integer getSeriacion_creditos() {
        return seriacion_creditos;
    }

    public void setSeriacion_creditos(Integer seriacion_creditos) {
        this.seriacion_creditos = seriacion_creditos;
    }

    public String getSeriacion_corregistro() {
        return seriacion_corregistro;
    }

    public void setSeriacion_corregistro(String seriacion_corregistro) {
        this.seriacion_corregistro = seriacion_corregistro;
    }

    public String getSeriacion_autorizacion() {
        return seriacion_autorizacion;
    }

    public void setSeriacion_autorizacion(String seriacion_autorizacion) {
        this.seriacion_autorizacion = seriacion_autorizacion;
    }

    public Boolean getCursada() {
        return cursada;
    }

    public void setCursada(Boolean cursada) {
        this.cursada = cursada;
    }

    public Integer getCreditos() {
        return creditos;
    }

    public void setCreditos(Integer creditos) {
        this.creditos = creditos;
    }

}

```

- **Clase RecomendacionUEA**

```

package com.uamapps.appingelectronicauam;

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

```

```

public class RecomendacionUEA {
    private List<PlanEstudios> UEAS_posibles;
    private Integer Limite_credits;
    private Integer prim_trim_cred;
    private Integer seg_trim_cred;
    private Integer ter_trim_cred;
    private Integer cred_rec;
    private List<PlanEstudios> UEAS_inscribir;
    private Integer cred_inter_mult = 0;
    private Integer cred_tronco_int = 0;
    Integer rec_cred = 0;
    private Integer min = 0;

    public RecomendacionUEA( List<PlanEstudios> UEAS_Posibles, Integer
limite_credits, Integer Prim_trim_cred, Integer Seg_trim_cred, Integer
Ter_trim_cred )
    {
        this.UEAS_posibles = UEAS_Posibles;
        this.Limite_credits = limite_credits;
        this.prim_trim_cred = Prim_trim_cred;
        this.seg_trim_cred = Seg_trim_cred;
        this.ter_trim_cred = Ter_trim_cred;
    }

    public void imprime_UEAS_inscribir() {
        for (int i = 0; i < this.UEAS_inscribir.size(); i++) {
            System.out.println(this.UEAS_inscribir.get(i).getPrioridad());
            System.out.println(this.UEAS_inscribir.get(i).getClave());
            System.out.println(this.UEAS_inscribir.get(i).getNombre_UEA());
            System.out.println(this.UEAS_inscribir.get(i).getCalificacion());
            System.out.println(this.UEAS_inscribir.get(i).getTrimestre());

System.out.println(this.UEAS_inscribir.get(i).getTipo_evaluacion());

System.out.println(this.UEAS_inscribir.get(i).getSeriacion_UEA_1());

System.out.println(this.UEAS_inscribir.get(i).getSeriacion_UEA_2());

System.out.println(this.UEAS_inscribir.get(i).getSeriacion_UEA_3());

System.out.println(this.UEAS_inscribir.get(i).getSeriacion_UEA_4());

System.out.println(this.UEAS_inscribir.get(i).getSeriacion_credits());

System.out.println(this.UEAS_inscribir.get(i).getSeriacion_corregistro());

System.out.println(this.UEAS_inscribir.get(i).getSeriacion_autorizacion());
            System.out.println(this.UEAS_inscribir.get(i).getCursada());
            System.out.println(this.UEAS_inscribir.get(i).getCredits());
            System.out.println("----");
        }
    }

    public void determina_cred_inscribir () {
        Integer prom_cred_insc = (prim_trim_cred + seg_trim_cred +
ter_trim_cred)/ 3;

        if (prom_cred_insc <= 20) {

```

```

        cred_rec = 40;
    }
    else {
        if (prom_cred_insc > Limite_creditos) {
            cred_rec = Limite_creditos;
        }
        else {
            prom_cred_insc += 9;
            if (prom_cred_insc < Limite_creditos) {
                cred_rec = prom_cred_insc;
            }
            else {
                cred_rec = prom_cred_insc - 9;
            }
        }
    }
}

public void recomienda_UEA () {
    Integer rec = cred_rec;
    this.UEAS_inscribir = new ArrayList<PlanEstudios>(10);
    for (int i = 0; i < this.UEAS_posibles.size(); i++) {

        if (rec >= this.UEAS_posibles.get(i).getCreditos()) {
            this.UEAS_inscribir.add(this.UEAS_posibles.get(i));
            rec -= this.UEAS_posibles.get(i).getCreditos();
            if
(this.UEAS_posibles.get(i).getClave().contains("110XXX"))
                cred_inter_mult +=
this.UEAS_posibles.get(i).getCreditos();
            else
                if
(this.UEAS_posibles.get(i).getClave().contains("XXXXXX"))
                    cred_tronco_int +=
this.UEAS_posibles.get(i).getCreditos();
            else
                rec_cred +=
this.UEAS_posibles.get(i).getCreditos();
        }
        else {
            break;
        }
    }
}

public void determina_min_UEA () {
    min = this.UEAS_posibles.get(0).getCreditos();

    for (int i = 1; i < this.UEAS_posibles.size(); i++) {
        if (min > this.UEAS_posibles.get(i).getCreditos()) {
            min = this.UEAS_posibles.get(i).getCreditos();
        }
    }
}

public Integer getMin() {
    return min;
}
public void setMin(Integer min) {

```

```

        this.min = min;
    }
    public Integer getRec_cred() {
        return rec_cred;
    }
    public void setRec_cred(Integer rec_cred) {
        this.rec_cred = rec_cred;
    }
    public List<PlanEstudios> getUEAS_posibles() {
        return UEAS_posibles;
    }
    public void setUEAS_posibles(List<PlanEstudios> uEAS_posibles) {
        UEAS_posibles = uEAS_posibles;
    }
    public Integer getLimite_creditos() {
        return Limite_creditos;
    }
    public void setLimite_creditos(Integer limite_creditos) {
        Limite_creditos = limite_creditos;
    }
    public Integer getPrim_trim_cred() {
        return prim_trim_cred;
    }
    public void setPrim_trim_cred(Integer prim_trim_cred) {
        this.prim_trim_cred = prim_trim_cred;
    }
    public Integer getSeg_trim_cred() {
        return seg_trim_cred;
    }
    public void setSeg_trim_cred(Integer seg_trim_cred) {
        this.seg_trim_cred = seg_trim_cred;
    }
    public Integer getTer_trim_cred() {
        return ter_trim_cred;
    }
    public void setTer_trim_cred(Integer ter_trim_cred) {
        this.ter_trim_cred = ter_trim_cred;
    }
    public Integer getCred_rec() {
        return cred_rec;
    }
    public void setCred_rec(Integer cred_rec) {
        this.cred_rec = cred_rec;
    }
    public List<PlanEstudios> getUEAS_inscribir() {
        return UEAS_inscribir;
    }
    public void setUEAS_inscribir(List<PlanEstudios> uEAS_inscribir) {
        UEAS_inscribir = uEAS_inscribir;
    }
    public Integer getCred_inter_mult() {
        return cred_inter_mult;
    }
    public void setCred_inter_mult(Integer cred_inter_mult) {
        this.cred_inter_mult = cred_inter_mult;
    }
    public Integer getCred_tronco_int() {
        return cred_tronco_int;
    }
    public void setCred_tronco_int(Integer cred_tronco_int) {
        this.cred_tronco_int = cred_tronco_int;
    }
}

```



```
}
```

- **Clase MuestraUEA**

```
package com.uamapps.appingelectronicauam;

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

import android.os.Bundle;
import android.os.Parcelable;
import android.app.Activity;
import android.app.AlertDialog;
import android.app.ExpandableListActivity;
import android.app.ListActivity;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.SimpleExpandableListAdapter;
import android.widget.TextView;
import android.support.v4.app.NavUtils;
import android.text.InputFilter;
import android.text.InputType;
import android.annotation.TargetApi;
import android.content.DialogInterface;
import android.content.Intent;
import android.os.Build;

public class Activity_muestra_uea extends ExpandableListActivity {

    List<Map<String, String>> mGroupData = new ArrayList<Map<String,
String>>();
    List<List<Map<String, String>>> mChildData = new ArrayList<List<Map<String,
String>>>();
    Integer nuev_cred = 0;
    Integer rec_cred;
    Integer cred_inter_mult = 0;
    Integer cred_tronco_int = 0;
    Integer limcreditos = 0;
    Integer Min = 0;

    ArrayAdapter<String> adapter;

    int clickCounter=0;

    List<PlanEstudios> UEAS_inscribir;
    List<PlanEstudios> UEAS_posibles;
    List<PlanEstudios> UEAS_opt_inter_mult_posibles;
    List<PlanEstudios> UEAS_opt_tronco_int_posibles;
```

```

@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_activity_muestra_uea);
    SimpleExpandableListAdapter expListAdapter =
        new SimpleExpandableListAdapter(
            this,
            mGroupData, // Creating group List.
            R.layout.group_row, // Group item layout
            new String[] { " " }, // the key of group item.
            new int[] { R.id.row_name }, // ID of each group
            item.-Data under the key goes into this TextView.
            mChildData, // childData describes second-
            level entries.
            R.layout.child_row, // Layout for sub-level
            entries(second level).
            new String[] { " " }, // Keys in childData maps to
            display.
            new int[] { R.id.grp_child } // Data under the keys
            above go into these TextViews.
        );
    setListAdapter(expListAdapter);
    // Get the message from the intent
    Intent intent = getIntent();
    Integer creditos = intent.getIntExtra("Creditos", 0);
    String renuncias = intent.getStringExtra("Renuncias");
    limcreditos = intent.getIntExtra("LimCreditos", 0);
    rec_cred = intent.getIntExtra("RecCreditos", 0);
    cred_inter_mult = intent.getIntExtra("CredInterMult", 0);
    cred_tronco_int = intent.getIntExtra("CredTroInt", 0);
    Min = intent.getIntExtra("Min", 0);
    UEAS_inscribir = getIntent().getParcelableArrayListExtra("RecUEAS");
    UEAS_posibles = getIntent().getParcelableArrayListExtra("PosUEAS");
    UEAS_opt_inter_mult_posibles =
    getIntent().getParcelableArrayListExtra("RecUEASOptInterMult");
    UEAS_opt_tronco_int_posibles =
    getIntent().getParcelableArrayListExtra("PosUEASOptTroncoIntPosibles");

    setContentView(R.layout.activity_activity_muestra_uea);

    if (renuncias.equals("0")) {
        ((TextView)findViewById (R.id.textView2)).setText(" "+"No hay
renuncias a UEA");
    } else {
        ((TextView)findViewById (R.id.textView2)).setText(" "+"Si hay
renuncias a UEA");
    }

    ((TextView)findViewById (R.id.textView4)).setText(" "+"
creditos.toString());
    ((TextView)findViewById (R.id.textView6)).setText("
"+limcreditos.toString());
    ((TextView)findViewById (R.id.textView8)).setText("
"+rec_cred.toString());

    AlertDialog ad = new AlertDialog.Builder(this).create();
    ad.setCancelable(false);
    ad.setMessage("Datos recibidos exitosamente, análisis completo");
    ad.setButton("OK", new DialogInterface.OnClickListener() {

```

```

        @Override
        public void onClick(DialogInterface dialog, int which) {
            dialog.dismiss();
            //llena_tabla ();
        }
    });
    ad.show();
    findViewById(R.id.button2).setEnabled(false);
}

public void addItem(final View v) {
    mGroupData.clear();
    mChildData.clear();

    for( int n = 0 ; n < 3 ; n++ ) {
        if (n == 0) {
            HashMap<String, String> child = new HashMap<String,
String>();
            child.put( " ", " " + "Créditos: " + rec_cred + " de UEAS
Obligatorias" );
            mGroupData.add(child);
        }
        if (n == 1) {
            if (cred_inter_mult > 0) {
                HashMap<String, String> child = new HashMap<String,
String>();
                child.put( " ", " " + "Créditos: " + cred_inter_mult
+ " de UEAS Optativas Inter y Multidisciplinar" );
                mGroupData.add(child);
            }
        }
        if (n == 2) {
            if (cred_tronco_int > 0) {
                HashMap<String, String> child = new HashMap<String,
String>();
                child.put( " ", " " + "Créditos: " + cred_tronco_int
+ " de UEAS Optativas de Integración" );
                mGroupData.add(child);
            }
        }
    }

    if(n == 0) {
        ArrayList secList = new ArrayList();
        for (int i = 0; i <
this.UEAS_inscribir.size(); i++) {
            if
(!this.UEAS_inscribir.get(i).getClave().contains("XXX")) {
                HashMap<String,
String> par = new HashMap<String, String>();
                par.put( " ", " " +
this.UEAS_inscribir.get(i).getClave() + " -
"+this.UEAS_inscribir.get(i).getNombre_UEA()+" - Créditos: " +
this.UEAS_inscribir.get(i).getCredito() );
            }
        }
    }
}

```

```

                secList.add( par );
            }
        }
        mChildData.add( secList );
    }
    if(n == 1) {
        ArrayList secList = new ArrayList();

        if (cred_inter_mult > 0) {
            for (int i = 0; i <
this.UEAS_opt_inter_mult_posibles.size(); i++) {

                if
(this.UEAS_opt_inter_mult_posibles.get(i).getCreditos() <= cred_inter_mult) {

                    HashMap<String, String> par = new HashMap<String, String>();
                    par.put( "
", " " + this.UEAS_opt_inter_mult_posibles.get(i).getClave() + " -
"+this.UEAS_opt_inter_mult_posibles.get(i).getNombre_UEA()+" - Créditos: " +
this.UEAS_opt_inter_mult_posibles.get(i).getCreditos() );
                    secList.add(
par );

                }
            }
            mChildData.add( secList );
        }
    }
    if(n == 2) {
        ArrayList secList = new ArrayList();

        if (cred_tronco_int > 0) {
            for (int i = 0; i <
this.UEAS_opt_tronco_int_posibles.size(); i++) {

                if
(this.UEAS_opt_tronco_int_posibles.get(i).getCreditos() <= cred_tronco_int) {

                    HashMap<String, String> par = new HashMap<String, String>();
                    par.put( "
", " " + this.UEAS_opt_tronco_int_posibles.get(i).getClave() + " -
"+this.UEAS_opt_tronco_int_posibles.get(i).getNombre_UEA()+" - Créditos: " +
this.UEAS_opt_tronco_int_posibles.get(i).getCreditos() );
                    secList.add(
par );

                }
            }
            mChildData.add( secList );
        }
    }
}

```

```

        super.onContentChanged();
    }

```

```

        super.onContentChanged();
        Integer cred_totales = rec_cred + cred_inter_mult + cred_tronco_int;
        ((TextView)findViewById (R.id.textView8)).setText("
"+cred_totales.toString());

        if (cred_totales == 0) {

```

```

        AlertDialog ad = new AlertDialog.Builder(this).create();
        ad.setCancelable(false);
        ad.setMessage("El número de créditos que introduciste no es
suficiente para cubrir las UEAS que puedes cursar. Introduce un número mayor de
créditos.");
        ad.setButton("OK", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                dialog.dismiss();
                showPopUp2(v);
            }
        });
        ad.show();
    } else if (UEAS_inscribir.isEmpty() && cred_inter_mult == 0 &&
cred_tronco_int == 0) {
        AlertDialog ad = new AlertDialog.Builder(this).create();
        ad.setCancelable(false);
        ad.setMessage("Haz terminado la carrera.");
        ad.setButton("OK", new DialogInterface.OnClickListener() {
            @Override
            public void onClick(DialogInterface dialog, int which) {
                dialog.dismiss();
            }
        });
        ad.show();
    }
    findViewById(R.id.button1).setEnabled(false);
    findViewById(R.id.button2).setEnabled(true);
}

public void recomienda_UEA (Integer cred) {
    Integer rec = cred;
    rec_cred = 0;
    cred_inter_mult = 0;
    cred_tronco_int = 0;
    UEAS_inscribir.clear();
    this.UEAS_inscribir = new ArrayList<PlanEstudios>(10);
    for (int i = 0; i < this.UEAS_posibles.size(); i++) {

        if (rec >= this.UEAS_posibles.get(i).getCreditos()) {
            this.UEAS_inscribir.add(this.UEAS_posibles.get(i));
            rec -= this.UEAS_posibles.get(i).getCreditos();
            if
(this.UEAS_posibles.get(i).getClave().contains("110XXX"))
                cred_inter_mult +=
this.UEAS_posibles.get(i).getCreditos();
            else
                if
(this.UEAS_posibles.get(i).getClave().contains("XXXXXX"))
                    cred_tronco_int +=
this.UEAS_posibles.get(i).getCreditos();
            else
                rec_cred +=
this.UEAS_posibles.get(i).getCreditos();
        }
        else {

            break;
        }
    }
    findViewById(R.id.button1).setEnabled(true);
    findViewById(R.id.button2).setEnabled(false);
}

```

```

public void showPopUp2(View v) {

    AlertDialog.Builder helpBuilder = new AlertDialog.Builder(this);
    helpBuilder.setTitle("Cambio de No. créditos");
    helpBuilder.setMessage("Ingrese el No. de créditos deseado (mayor a " +
Min + " y menor a 63)");
    final EditText input = new EditText(this);
    input.setSingleLine();
    input.setInputType(InputType.TYPE_CLASS_NUMBER);
    input.setFilters(new InputFilter[]{new InputFilter.LengthFilter(2)});
    input.setText("");
    helpBuilder.setView(input);
    helpBuilder.setNeutralButton("Recalcular",
        new DialogInterface.OnClickListener() {

            @Override
            public void onClick(DialogInterface dialog, int which) {

                }
            });

    final AlertDialog helpDialog = helpBuilder.create();
    helpDialog.show();

    Button positiveButton =
helpDialog.getButton(DialogInterface.BUTTON_NEUTRAL);
    positiveButton.setOnClickListener( new View.OnClickListener() {
        @Override
        public void onClick(View onClick) {

            if (input.getText().toString().equals("") ||
Integer.parseInt(input.getText().toString())>63 ||
Integer.parseInt(input.getText().toString())<Min ) {

                } else {
                    nuev_cred = Integer.parseInt(input.getText().toString());
                    recomienda_UEA
(Integer.parseInt(input.getText().toString()));
                    helpDialog.dismiss();
                }

            }
        });
    mGroupData.clear();
    mChildData.clear();
    super.onContentChanged();

}

/**
 * Set up the {@link android.app.ActionBar}, if the API is available.
 */
@TargetApi(Build.VERSION_CODES.HONEYCOMB)
private void setupActionBar() {
    if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.HONEYCOMB) {
        getActionBar().setDisplayHomeAsUpEnabled(true);
    }
}

```

```
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {

        getMenuInflater().inflate(R.menu.activity_muestra_uea, menu);
        return true;
    }

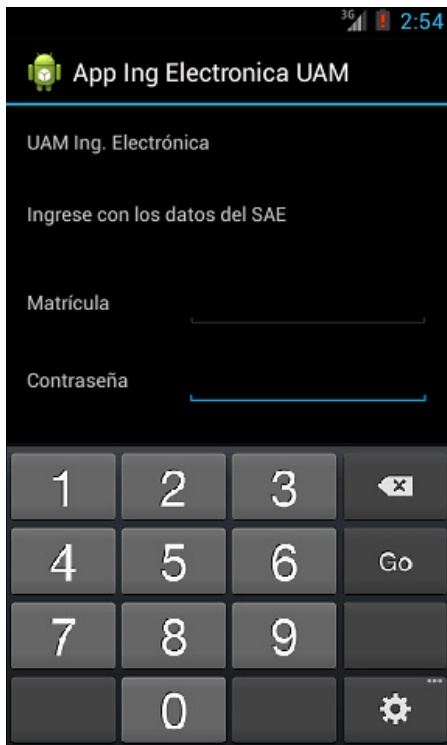
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        switch (item.getItemId()) {
            case android.R.id.home:

                NavUtils.navigateUpFromSameTask(this);
                return true;
            }
        return super.onOptionsItemSelected(item);
    }
}
```

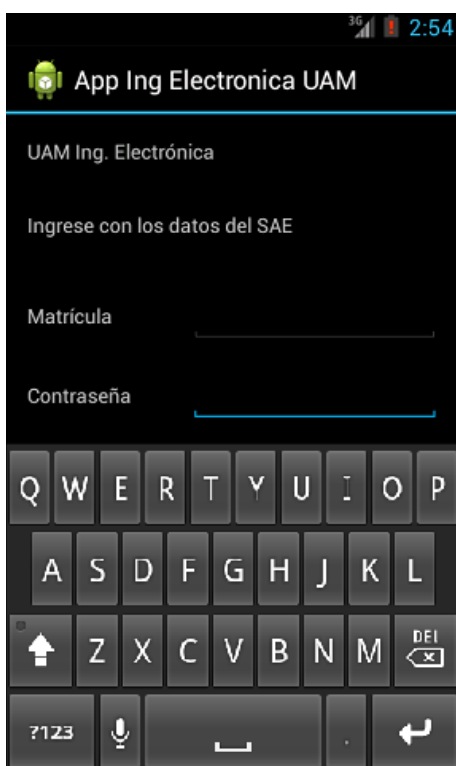
Pruebas del sistema

A continuación se presentan los estados de la interfaz gráfica de la aplicación para diferentes casos en el uso del programa

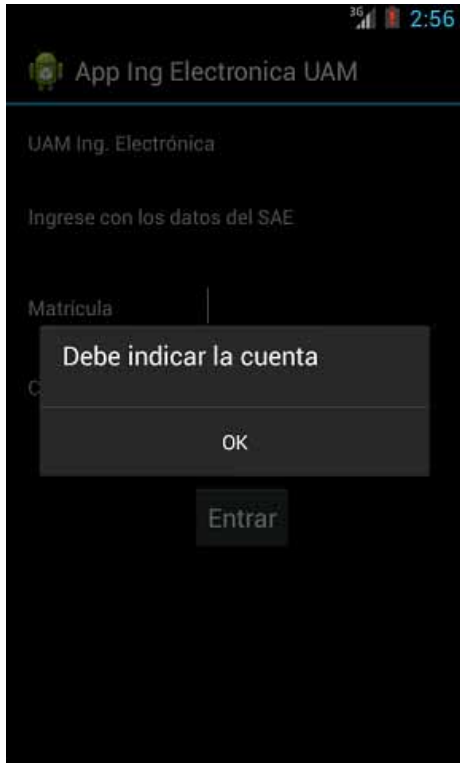
Pantalla de inicio:



En la pantalla principal de la aplicación, se le pedirá al alumno ingresar con los datos del SAE. El campo de la matrícula está configurado con el teclado numérico, por lo que se asegura que sólo se podrán introducir dígitos.



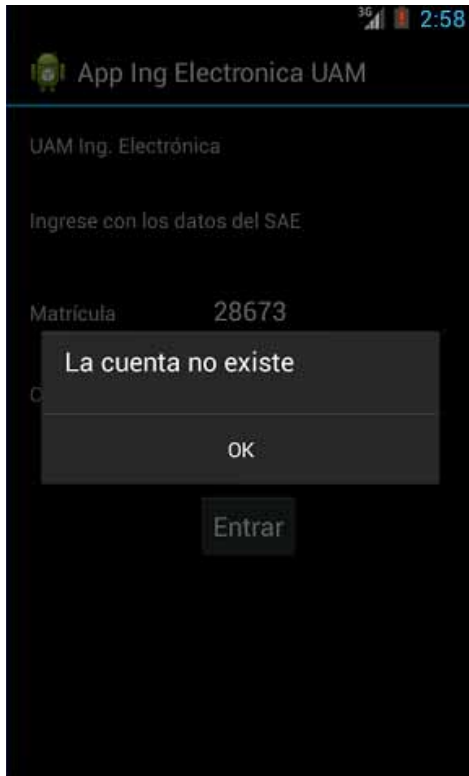
El campo de la contraseña está configurado con el teclado QWERTY completo, asegurando que se pueden introducir todas las letras, números y símbolos permitidos por el SAE para la contraseña.



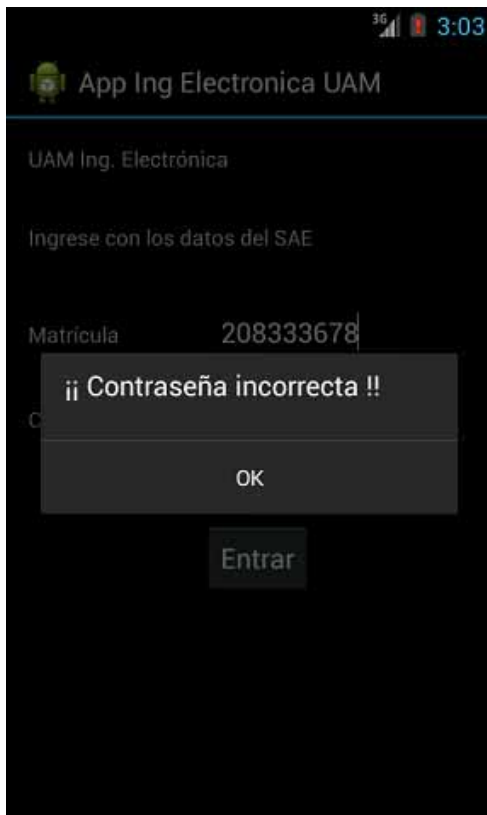
Si el alumno ingresa sin datos, la aplicación le informará mediante un mensaje que indique la cuenta.



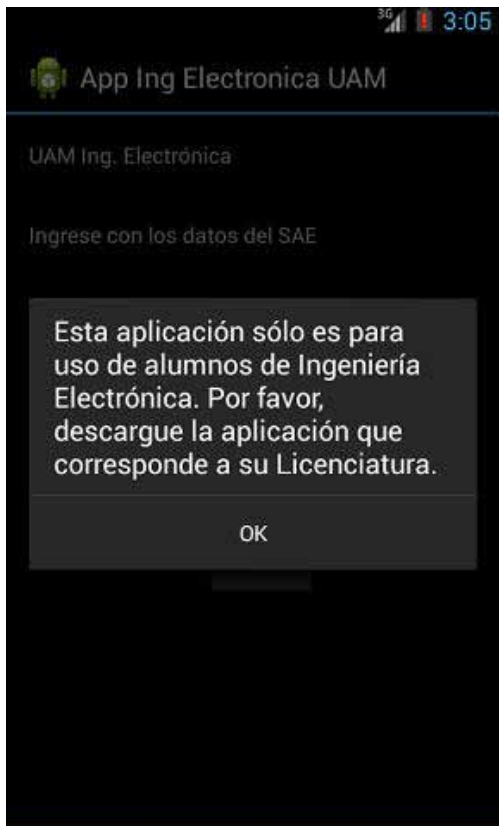
Si el alumno únicamente ingresa su matrícula pero no su contraseña, le mostrará un mensaje que debe indicar la contraseña



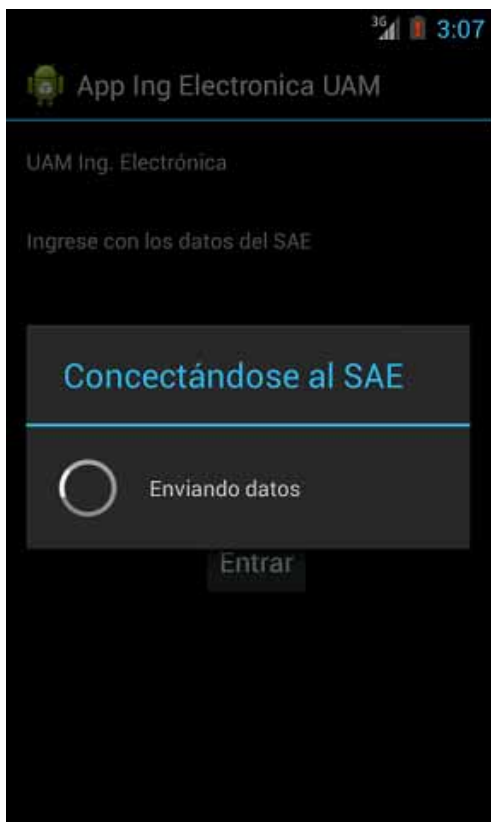
Si el alumno ingresa una matrícula y una contraseña no existentes, le informará en un mensaje dicha advertencia.



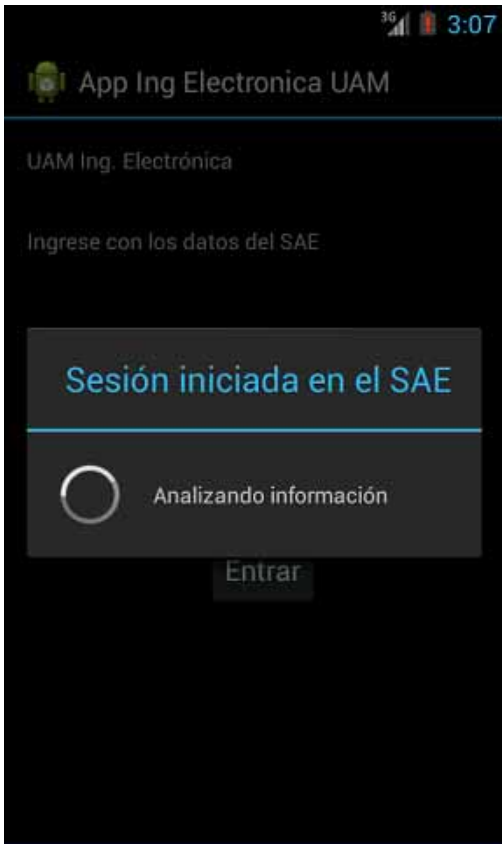
Si el alumno ingresa con una matrícula que sí existe pero una contraseña incorrecta, reinformará mediante un mensaje.



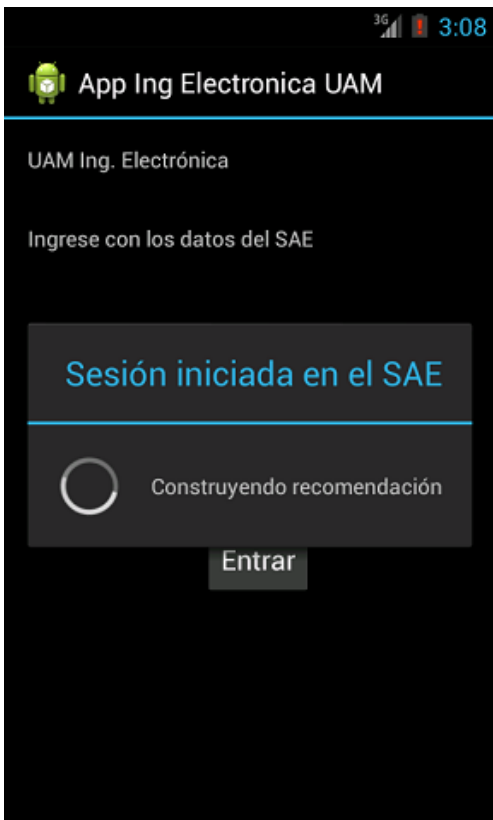
Si el alumno ingresa una matrícula y contraseña válidas y existentes, pero correspondientes a una ingeniería diferente a la Ingeniería Electrónica, se le informará que la aplicación es únicamente para uso de los alumnos de Ingeniería Electrónica.



Si el alumno introduce una matrícula y contraseña existentes, válidas y correspondientes a la Ingeniería Electrónica, la aplicación abrirá una sesión en el SAE. Mediante el cuadro de diálogo, la aplicación informará el estado de la sesión y del análisis de la información.

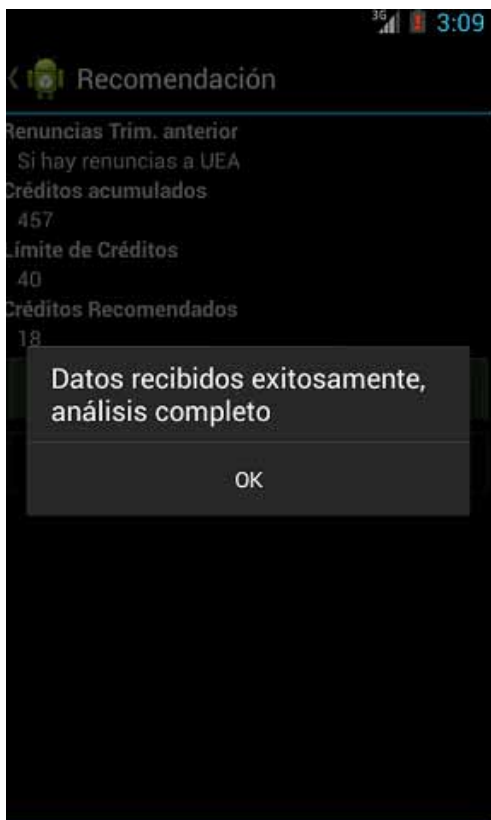


En este estado, la aplicación ya inició la sesión en el SAE y está recibiendo y analizando la información académica del alumno.

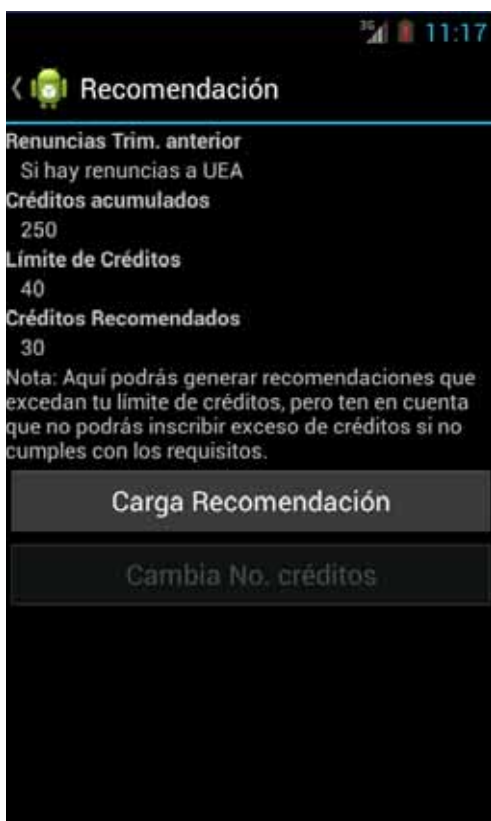


Este estado indica que la aplicación está construyendo la recomendación en base al análisis previamente realizado.

Pantalla de recomendación:



En cuanto el análisis termine, la aplicación informará al alumno.

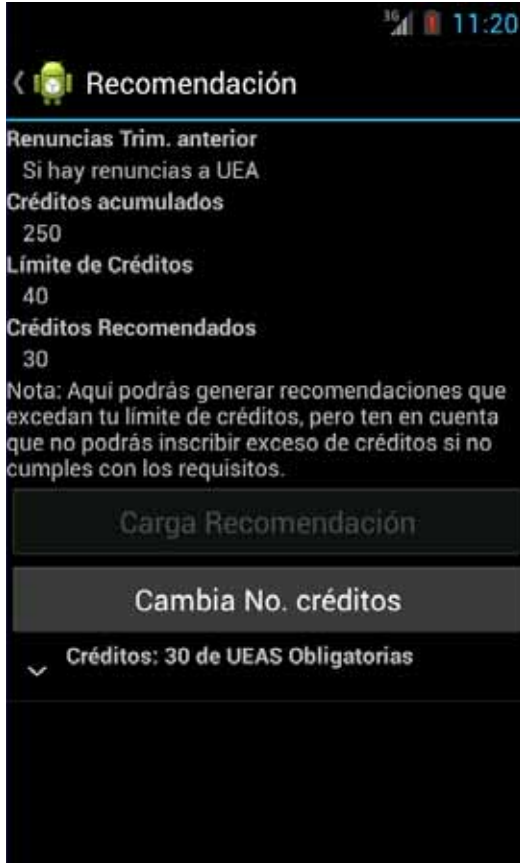


Se presentarán datos generales del alumno:

- Si dio UEA de baja
- Créditos acumulados
- Límite de créditos
- Créditos recomendados por la aplicación
- Botón de “Carga recomendación”
- Botón de “Cambia no. créditos”

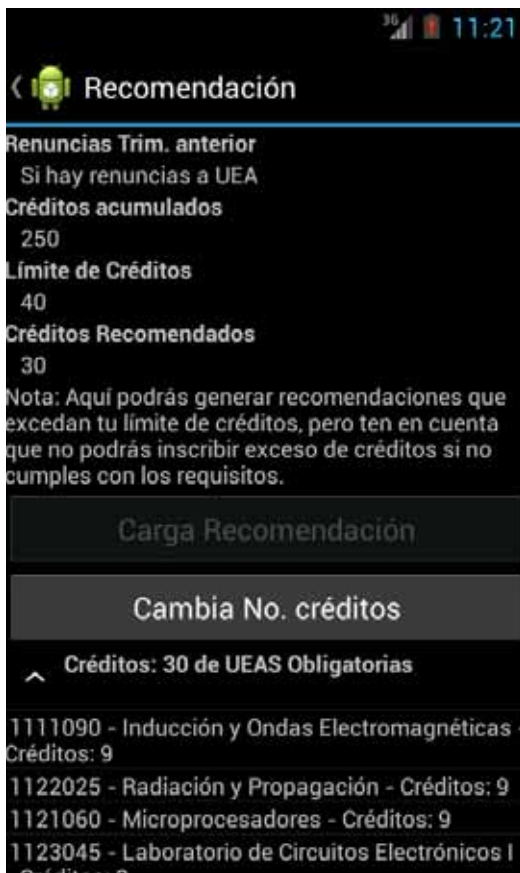
Los datos anteriores influyen en la recomendación al alumno:

- Si hay renuncias, el límite de créditos es 40 créditos
- Los créditos recomendados dependen del promedio de los últimos 3 trimestres cursados y de cuántos créditos se aprobaron.



Si el alumno presiona el botón de “Carga recomendación” la aplicación cargará en pantalla la lista de UEAs recomendadas que cubren los créditos recomendados. Estarán divididas en:

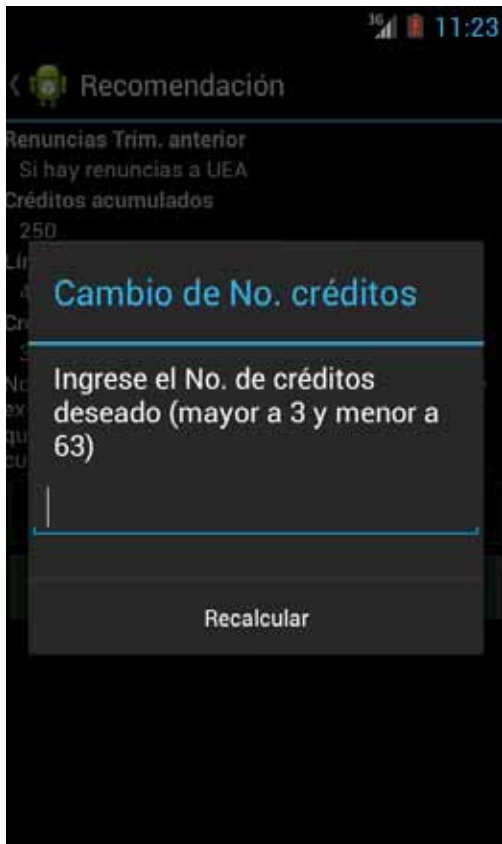
- UEAs Obligatorias
- UEAs Tronco Inter y Multidisciplinar
- UEAs Tronco de Integración



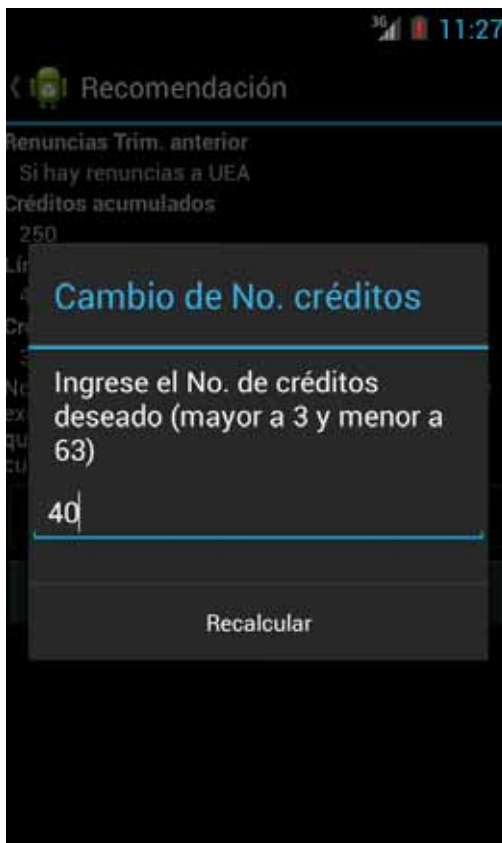
Al expandir las listas aparecerá la información en detalle de las UEAs recomendadas:

- Clave
- Nombre de la UEA
- No. créditos

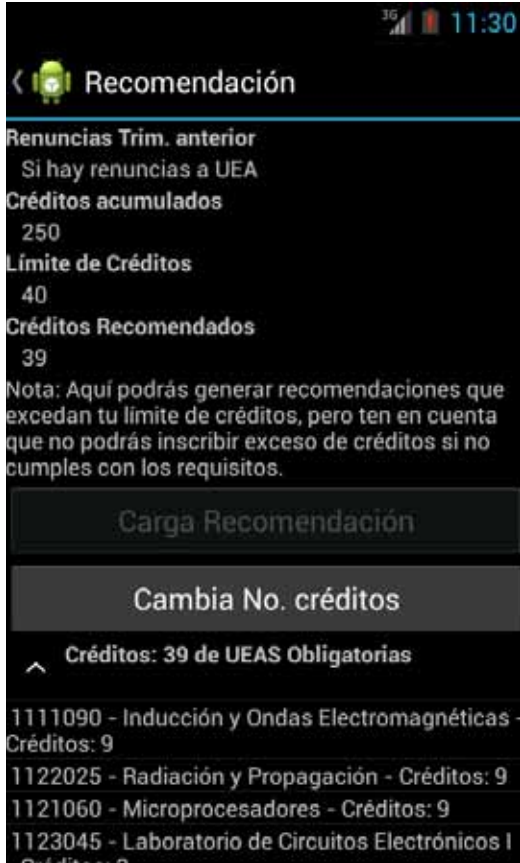
En este momento si el alumno no está conforme con el No. de créditos recomendados, puede modificarlo presionando el botón “Cambia no. créditos”.



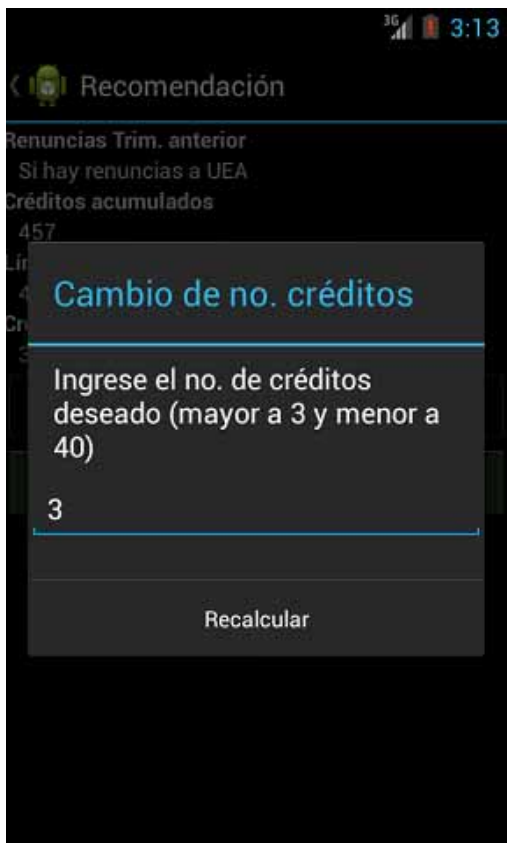
Si el alumno presiona el botón “Cambia no. créditos” aparecerá un cuadro de diálogo donde podrá introducir un nuevo número de créditos.



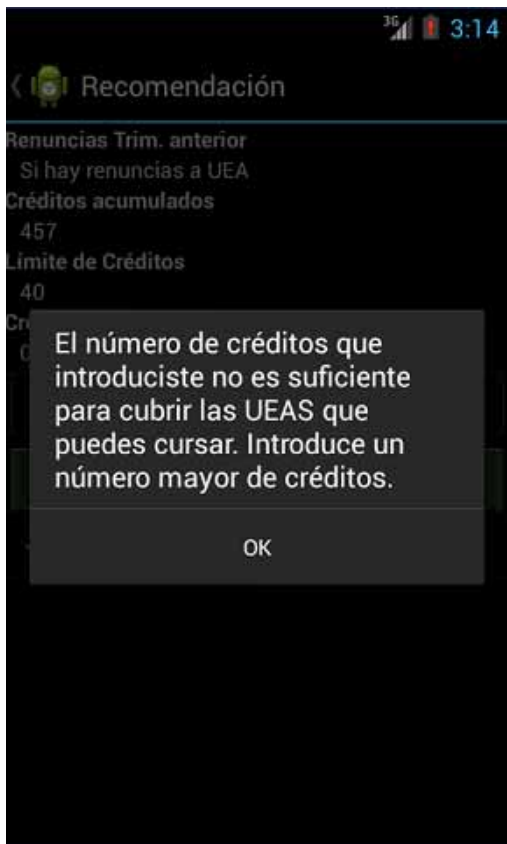
El nuevo No. de créditos deberá ser mayor a 3 pero menor al límite de créditos. En cuanto introduzca un número válido y presione el botón recalcular, la aplicación volverá a realizar la recomendación de UEAs.



El alumno deberá presionar el botón de “Carga recomendación” para cargar en pantalla la nueva recomendación de UEAs. En este momento, el alumno puede cambiar el número de créditos las veces que desee y generar las recomendaciones que desee.



Si en algún momento el alumno ingresa un número muy pequeño de créditos, el cual no es suficiente para cubrir las UEAs que puede cursar, la aplicación le mostrará un mensaje informándole lo anterior.



Si ocurre lo anterior, deberá introducir un número de créditos mayor.

Conclusiones

Se obtuvo una aplicación para dispositivos móviles capaz de recomendar las UEA a los alumnos de Ing. Electrónica para concluir sus estudios con una administración de su carga académica y así aumentar la eficiencia terminal.

Gracias a este proyecto, pude aprender el desarrollo de aplicaciones móviles y el impacto que pueden llegar a tener, debido a su practicidad y facilidad de uso, estas últimas dependientes de las habilidades y capacidades del programador.

Esta aplicación ofrece una herramienta de orientación al alumno respecto a la planeación de su avance académico dentro de la Universidad tomando en cuenta factores que a veces no son considerados por el alumno por falta de información o incluso por desinformación

Espero que la aplicación cumpla su objetivo primordial, orientando a los alumnos para un mejor desempeño sobre todo en su eficiencia terminal.

Referencias

[1] Guía Proceso de Reinscripción Web Trimestre 2013-Primavera. Coordinación de Sistemas Escolares, UAM Azcapotzalco, 2013. Acceso (9 de julio de 2013). <http://www.azc.uam.mx/sae/avisos/2013/A174.pdf>

[2] Universidad Autónoma Metropolitana Azcapotzalco. Módulo de Información Escolar. Acceso (9 de julio de 2013). <http://alumnosa.uam.mx:8080/sae/>

[3] Universidad Autónoma Metropolitana Azcapotzalco, División de Ciencias Básicas e Ingeniería, "Plan de estudios Licenciatura en Ingeniería Electrónica". Acceso (9 de julio de 2013).
http://cbi.azc.uam.mx/index.php?option=com_content&view=category&layout=blog&id=40&Itemid=597, 2013

[8] Universidad Autónoma Metropolitana Azcapotzalco, División de Ciencias Básicas e Ingeniería, "Modificaciones a los planes de estudio que entrarán en vigor en el trimestre 13-O". Acceso (9 de julio de 2013).<http://cbi.azc.uam.mx/planes/>, 2013