

Universidad Autónoma Metropolitana Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

Reporte

Asistente de inscripción para alumnos de Ingeniería Física

Martin Hernández González

208204090

Ernesto Rodrigo Vázquez Cerón

Profesor Asociado

Departamento de Electrónica

Sergio Luis Pérez Pérez

Profesor Titular

UAM Cuajimalpa

Resumen

El presente documento es el reporte final relacionado al Proyecto de Integración de Ingeniería en Computación titulado “Asistente de Inscripción para alumnos de Ingeniería Física”, el cual se desarrolló en una aplicación móvil para dispositivos Android. Dicha aplicación generará una recomendación a los alumnos de Ingeniería Física de las UEA que podrá inscribir en su próximo trimestre.

Para la realización del proyecto se obtendrá la información necesaria del módulo de información escolar de alumnos de licenciatura proporcionado por la Universidad Autónoma Metropolitana, el cual provee una página en internet para que los alumnos puedan verificar su información académica entre otras cosas. Con dicho recurso la aplicación extraerá dos principales módulos de interés: la Información Académica y el Kardex. Estos módulos son arrojados por la página web en archivos con formato html, donde una de las principales tareas será el descomponer dichos archivos para el procesamiento de la información.

En el módulo de Información Académica se encuentran los datos personales de los alumnos con relación a la universidad, es decir, nombre completo, la carrera que cursa, trimestre de inicio, trimestre en el que se encuentra actualmente, número de créditos totales, etc.

Para el caso del módulo del Kardex se obtendrán los registros de las UEA que ha cursado el alumno durante su trayectoria en la universidad, obteniendo de éste la clave, nombre, calificación y tipo de evaluación de cada una de ellas. Una vez finalizada la obtención de la información, se hará el procesamiento correspondiente para generar la recomendación a los alumnos de Ingeniería Física con base en la recomendación emitida por el Coordinador de la carrera para la conclusión de la carrera en tiempo y forma.

Contenido

1. Objetivos.....	4
1.1 Objetivo General.....	4
1.2 Objetivos Específicos.....	4
2. Introducción	5
3. Desarrollo del Proyecto	6
3.1 Módulo de ingreso	7
3.2 Módulo de solicitud	10
3.3 Módulo de análisis	12
3.4 Módulo de recomendación.....	15
3.5 Módulo de presentación.....	21
4. Validación de la aplicación	23
5. Discusión de resultados.....	25
6. Conclusión	26
7. Bibliografía	27
8. Apéndice.....	28
A. Implementación de la vista LoginScreen.	28
B. Implementación de la validación de conexión a internet en Android.	30
C. Implementación de la clase Conexión.	31
D. Implementación de la clase DatosAcademicos.	34
E. Implementación de la vista UEAS.	39
F. Implementación de la clase SeriacionObligatoria	41
G. Implementación de la clase SeriacionOptativa.	75

1. Objetivos

1.1 Objetivo General

Implementar una aplicación móvil para dispositivos Android que le pueda recomendar a un alumno de Ingeniería Física las opciones de UEA para su próximo trimestre.

1.2 Objetivos Específicos

- Elaborar una interfaz que pida como entrada la matrícula y contraseña del alumno de Ingeniería Física y que establezca la conexión con el módulo de Información escolar de alumnos de licenciatura.
- Obtener del módulo de información escolar de alumnos de licenciatura el Kardex y la información académica del alumno.
- Descomponer los archivos con extensión html arrojados por el servidor del módulo de información escolar de alumnos de licenciatura.
- Diseñar e implementar un algoritmo que realice la recomendación de las UEA al alumno.
- Diseñar e implementar una interfaz gráfica que muestre la recomendación al usuario.

2. Introducción

Desde hace tiempo las carreras de ingeniería de la Universidad Autónoma Metropolitana unidad Azcapotzalco en la división de Ciencias Básicas e Ingeniería así como las demás divisiones, se preocupan por la conclusión de la carrera de los alumnos, para lo cual se realiza una revisión y modificación al plan de estudios atendiendo las necesidades académicas de cada programa, por ejemplo, programas de nivelación académica, proyectos de integración comunes para las 10 ingenierías de la unidad Azcapotzalco, programas de tutorías para becarios PRONABE, cursos no presenciales, entre otros. Con esto la universidad pretende incrementar el número de egresados de ingeniería en tiempo y forma como lo establece cada uno de sus planes de estudios.

Además de brindarle una formación profesional al estudiante, también es importante orientarlo a que tome decisiones que no impliquen un retraso con su plan de estudios. Contando con un coordinador de carrera que le sugiere un cierto recorrido a través de éste para que lo concluya. En ocasiones la falta de comunicación ó de consulta de información por parte del alumno con su coordinador provoca que se retrase en su trayectoria escolar. Este proyecto se desarrolló con la utilización de los avances tecnológicos en cuanto a dispositivos móviles (teléfonos celulares, tabletas, etc.) que utilizan un sistema operativo [1] Android [2], el cual proporciona las interfaces necesarias para desarrollar aplicaciones de una forma muy sencilla en el lenguaje de programación Java. Con ello se realizó una aplicación que accede al módulo de información escolar de alumnos de licenciatura [3] dando como resultado archivos con extensión html [4]. Esto se utilizó para proporcionar una recomendación de las distintas Unidades de Enseñanza y Aprendizaje (UEA) que el alumno pudiera elegir en la inscripción de su próximo trimestre. Dicho módulo, muestra la información académica del estudiante, Kardex, su horario de clases en el trimestre lectivo, impresión de comprobante de re-inscripción, entre otros detalles.

Con esto se pretende que el alumno no opte por dar de baja la inscripción de la UEA, así como también, no tenga problemas para acreditarlas debido a no contar con los conocimientos previos requeridos que no se describen en su seriación, evitando así retrasos de la conclusión de su carrera.

Con fundamento en lo anterior se espera que el alumno tenga un avance idóneo acorde a sus habilidades, pues además de tomarse en cuenta la recomendación proporcionada por el coordinador de la carrera, se realizará un estimado del número de créditos que el estudiante podría acreditar basados en el número de créditos aprobados en trimestres anteriores. Por lo tanto, los alumnos inscribirán las UEA que tienen una mayor prioridad en cuanto a su seriación. Y por último, realizarlo en una aplicación móvil hace que el alumno tenga la facilidad de consultarlo en cualquier momento sin tener que recurrir a un equipo de cómputo.

3. Desarrollo del Proyecto

Para la realización del presente proyecto se dividió en 5 módulos (ver Ilustración 1).

- Módulo de ingreso.
- Módulo de solicitud.
- Módulo de análisis.
- Módulo de recomendación.
- Módulo de presentación.

Esta estructura se diseño para hacer flexible la implementación de la programación, así como el manejo de datos y resultados.

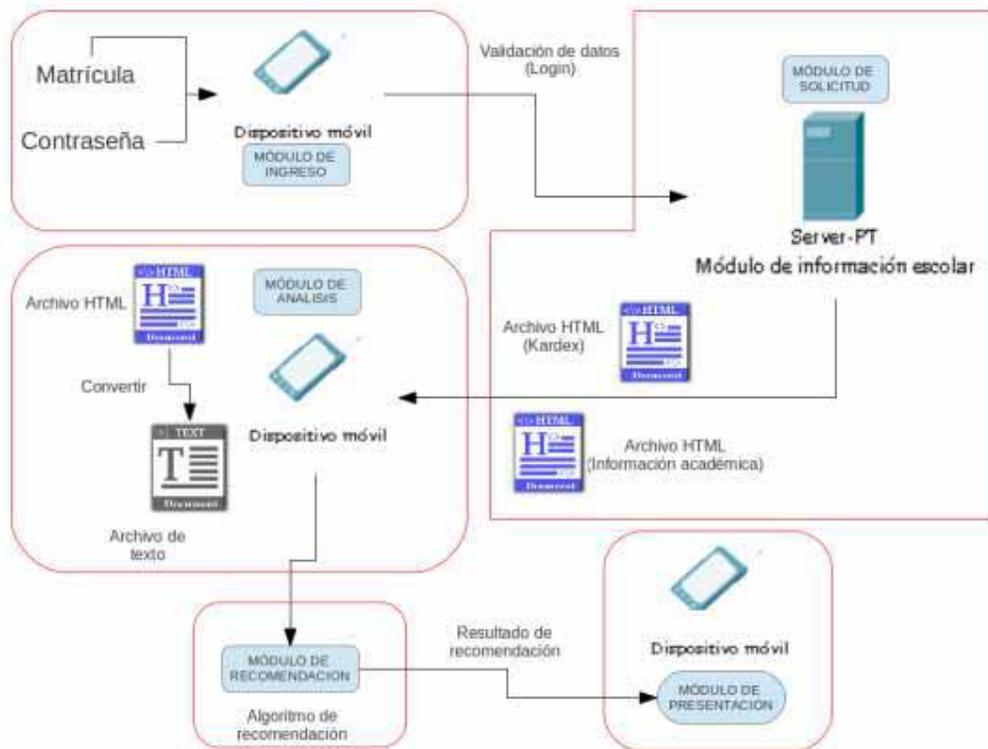


Ilustración 1 Módulos del proyecto.

Para el desarrollo y descripción de cada uno de los módulos mostrados en la ilustración 1 se muestra en las siguientes secciones la interfaz gráfica del proyecto en ejecución y posteriormente la explicación de los procedimientos de su funcionamiento, así como también una explicación técnica del funcionamiento de algunos componentes para alcanzar los objetivos de este proyecto. Con ello se pretende dejar un registro detallado del funcionamiento del código fuente para que pueda ser tomado como modelo en futuras aplicaciones y correcciones a esta aplicación en particular.

El recurso que se utilizó para simular funcionalidad de la aplicación fue el emulador proporcionado por el IDE (por sus siglas en inglés de Integrate Development Environment) de NetBeans 7.3.1 y un dispositivo móvil (celular) de la marca LG L7 modelo LG-P708g ambos con sistema operativo Android en su versión 4.0.3 IceCream Sandwich.

3.1 Módulo de ingreso

Para el desarrollo de este módulo se implementó una interfaz gráfica en xml donde se despliegan dos campos editables de texto y un botón. Esta vista corresponde a la pantalla de ingreso de la aplicación, tal y como se muestra en la Ilustración 2.

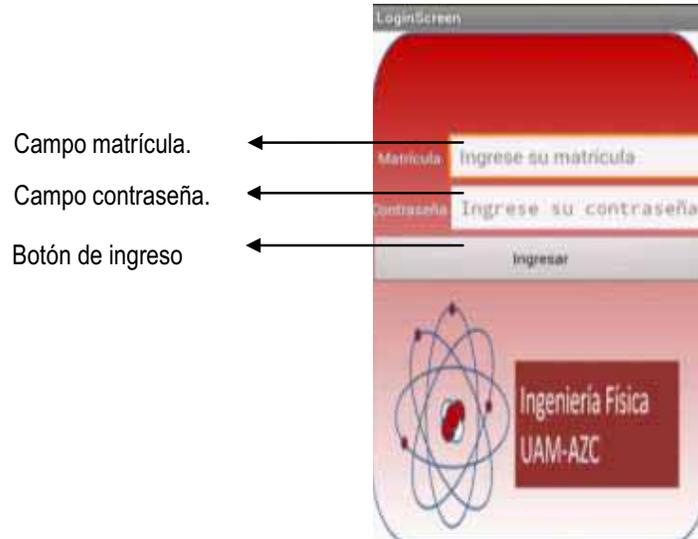


Ilustración 2 Interfaz de ingreso.

A continuación se describirá la funcionalidad de cada uno de los componentes que se utilizaron para la generación de dicha vista.

- Campo matrícula

Este componente es un campo editable de texto donde se ingresa la matrícula del alumno el cual fue programado solo para recibir tipos de datos numéricos, su código xml es el siguiente:

```
<EditText
    android:id="@+id/CampoTextoMatricula"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:hint="Ingrese su matrícula"
    android:numeric="integer">
</EditText>
```

- Campo contraseña

Este componente es un campo editable de texto donde se ingresa la contraseña del alumno el cual fue programado solo para recibir texto de entrada, su código xml es el siguiente:

```
<EditText
    android:id="@+id/CampoTextoContraseña"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:hint="Ingrese su contraseña"
    android:inputType="text"
    android:password="true">
</EditText>
```

- Botón de ingreso

Este componente es un botón que permite enviar los datos al servidor del módulo de información escolar de alumnos de licenciatura, su código xml es el siguiente:

```
<Button
    android:id="@+id/boton"
    android:text="Ingresar"
    android:layout_width="200dp"
    android:layout_height="wrap_content"
    android:gravity="center">
</Button>
```

Nota: Cabe enfatizar que la clave de usuario y contraseña corresponden al mismo que se utiliza en el módulo de información escolar de alumnos de licenciatura; la implementación en xml de la vista LoginScreen se encuentra en el Apéndice A anexo a este reporte.

La clase principal de la Activity se llama LoginScreen. Esta se encarga de instanciar los objetos xml para obtener los datos de entrada, validar campos vacíos y la conexión a internet, tal y como se muestra en las ilustraciones 3 y 4. Después de todas la validaciones antes mencionadas, esta clase crea un objeto de tipo Conexión al dar "click" sobre la opción "Ingresar". Posteriormente se retoma el método referido como Login, el cual recibe como parámetros las cadenas de texto instanciadas por los objetos xml, tanto de matrícula y contraseña (más adelante en el Módulo de solicitud se dará más a detalle el funcionamiento de este método). De esta manera no sólo se valida si el alumno pertenece a la carrera de Ingeniería Física, sino que también se realiza la obtención de la información escolar que se necesita, ver ilustración 5. En caso contrario la aplicación despliega un mensaje de que no pertenece a dicha licenciatura, ver ilustración 6.



Ilustración 3 Validación de campos vacíos.



Ilustración 4 Validación de conexión a Internet.



Ilustración 5 Validación de datos de ingreso.



Ilustración 6 Validación de alumnos de licenciatura.

La funcionalidad en Java servirá para desplegar la vista y obtener los datos de entrada, como se muestra en el siguiente pseudocódigo:

1. Se crean dos objetos de la clase EditText para la matrícula y la contraseña.
2. Se crea un objeto de la clase Button para realizar la conexión al servidor.
3. Se enlaza la funcionalidad del EditText de la matrícula y la contraseña con los id del xml.
4. Se enlaza la funcionalidad del Button de ingreso con el id del xml.
5. Finalmente se crea un manejador de eventos para el botón creado.

Para comenzar se describirá la validación de conexión a internet y campos de texto vacíos. Con lo cual se debe verificar que el dispositivo debe estar conectado a una red, ya sea Wi-Fi, 3G o 4G, esto para evitar que nuestra aplicación arroje excepciones por motivo de conectividad. Para ello se creó un método llamado "existeConexionInternet" en la Activity "LoginScreen", que proporciona un valor tipo booleano para realizar las operaciones necesarias de validación de conexión a una red. En el Apéndice B se encuentra la implementación del código para la validación de la conexión a Internet.

Para validar que los campos de ingreso no sean nulos se utiliza una estructura de control "if", el cual sirve para indicar al alumno que la aplicación necesariamente debe contener datos de entrada para proseguir con la funcionalidad del recomendador.

3.2 Módulo de solicitud

Para llevar a cabo la solicitud al servidor se creó una clase llamada Conexión, la cual consta de 2 atributos privados y 3 métodos, ver Ilustración 7.

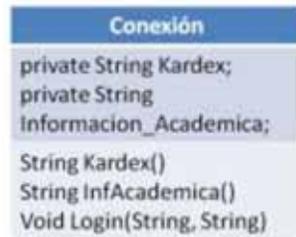


Ilustración 7 Clase Conexion().

El método Kardex sólo retorna el String Kardex en donde sea instanciado. El método InfAcademica solo devuelve el String Informacion_Academica donde sea instanciado.

El método Login sirve para llevar a cabo la conexión al servidor recibiendo como parámetros dos cadenas correspondientes a la matrícula y la contraseña del alumno enviados por el Módulo de Ingreso.

```
void Login(String Matricula, String Password)
{
    DefaultHttpClient httpClient = new DefaultHttpClient();
    List<NameValuePair> params=new ArrayList<NameValuePair>();
    params.add(new
        BasicNameValuePair("SIGLAS_UNI_XX.E_UNIDAD.AE02.1","AZC"));
    params.add(new BasicNameValuePair("%23.E_UNIDAD.AE02.1","AxJDMQ%3D%3D"));
    params.add(new BasicNameValuePair("%23CRC.E_UNIDAD.AE02.1","00000024"));
    params.add(new BasicNameValuePair("NOMBRE.IDENTIFICACION.NONMODELED",
        Matricula));
    params.add(new
        BasicNameValuePair("COMPLEMENTO.IDENTIFICACION.NONMODELED",Passw
        ord));
    params.add(new BasicNameValuePair("GO.IDENTIFICACION.NONMODELED","Entrar"));
    params.add(new BasicNameValuePair("%25.IDENTIFICACION.NONMODELED",""));
    params.add(new BasicNameValuePair("%23.WEB_INFO.SW01",""));
    params.add(new BasicNameValuePair("%23.WEB_MOD_ASO.SW01",""));
    params.add(new BasicNameValuePair("%23.USUARIO_ANEXO.SG02",""));
    params.add(new BasicNameValuePair("%23.MODULO_UWAS.SAE01",""));
    HttpPost httpPost = new
        HttpPost("https://ayamicatlan.uam.mx:8443/sae/azc/AEWBU004.olinici
        oSesion?mod=1");
}
```

Además de solo mandar los valores de matrícula y contraseña al servidor, también hay que mandar ciertos identificadores que le sirven al servidor para hacer válida la conexión que se pretende hacer, para ello se crea un List<NameValuePair> para almacenar los parámetros necesarios y enviarlos como una entidad en un objeto de la clase httpPost.

```
httpPost.setEntity(new UrlEncodedFormEntity(parametross));
```

Para almacenar la respuesta del servidor se crea un objeto de la clase HttpResponse [5] llamado responsePost para guardar la respuesta de nuestro objeto httpClient con ayuda del método execute.

```
HttpResponse responsePost = null;
responsePost = httpClient.execute(httpPost);
```

Después se crea un arreglo de cabeceras llamado “cookies” que nos va a servir para mantener la sesión iniciada.

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

A continuación se van a crear dos objetos de la clase HttpGet [5] que servirán para obtener la respuesta del servidor de la URL especificada para el Kardex y para la información académica del alumno. A continuación se muestran las direcciones donde se obtendrán los datos necesarios.

- a) <https://ayamictlan.uam.mx:8443/sae/azc/IEWBC020.oConsulta>
- b) <https://ayamictlan.uam.mx:8443/sae/azc/IEWBC007.oConsulta>

Después se agregan todas las cabeceras al objeto httpget y httpget2 hasta que se terminen las cookies en el objeto “cookies”.

```
for(Header c : cookies)
{
    httpget.addHeader("Cookie", c.getValue());
}
for (Header c : cookies)
{
    httpget2.addHeader("Cookie", c.getValue());
}
```

Ahora con los valores de las cookies almacenadas en la cabecera de nuestro objeto httpget y httpget2 se crea otro objeto llamado “responseGet” de la clase HttpResponse() [5], la cual obtendrá la respuesta del envío de datos con la sesión iniciada, mandando como parámetro dichos objetos.

```
HttpResponse responseGet = null;
responseGet = httpClient.execute(httpget);
```

Se crea un objeto “ent” de la clase HttpEntity() [5] para guardar tanto algunos encabezados como el cuerpo de la respuesta del servidor, después se llama al método EntityUtils(), el cual devuelve el valor de la solicitud al servidor y con el método toString() se convierte en cadena y así se obtiene el Kardex en una sola cadena con formato html.

```
HttpEntity ent=responseGet.getEntity();
Kardex=EntityUtils.toString(ent);
```

Para obtener la información académica se hace el mismo procedimiento que con la obtención del Kardex como se muestra a continuación.

```
responseGet = httpClient.execute(httpget2);
Informacion_Academica=EntityUtils.toString(ent);
```

Finalmente se declaran los String privados necesarios que contienen los valores de las páginas en formato html; que se requieren para realizar el análisis correspondiente. La implementación de la clase Conexión se encuentra en el Apéndice C anexo a este documento.

3.3 Módulo de análisis

Una vez que se obtienen los String en formato html se implementaron varias técnicas para procesar esa información, depurándola de etiquetas, signos alfanuméricos, signos especiales, etc. Con el objetivo de descomponer los String y hacer las operaciones correspondientes para obtener la información necesaria para este proyecto. Primeramente se debe verificar que los alumnos de la carrera de Ingeniería Física sean los únicos que pueden utilizar la aplicación, aunque en el Módulo de Solicitud manda la sesión y los String con los datos de usuario y contraseña de los alumnos de otra carrera, este módulo hará la tarea de hacer la clasificación pertinente de los alumnos de la carrera de Ingeniería Física.

Cuando el Módulo de Solicitud ha terminado su tarea satisfactoriamente necesitamos almacenar el trabajo hecho por la clase Conexión para ello se declaran dos String uno para el Kardex y otro para la información académica. Por consiguiente se llaman a los métodos "Kardex" e "InfAcademica" del objeto "con" de la clase Conexión almacenando el valor de retorno en los String privados.

Teniendo guardados los String se procede a la parte de validación de datos, es decir, verificar si la sesión se ha iniciado en el módulo de información escolar de alumnos de licenciatura. Para este procedimiento se necesita un reconocedor de patrones proporcionado por la librería "Regex" [6] en java, la cual busca patrones de cadenas en una cadena más grande. A continuación se describe un ejemplo de la declaración de un patrón:

```
Pattern nom patron = Pattern.compile("CADENA A COMPILAR");
```

Se declara un objeto con un nombre específico de la clase Pattern [6] haciendo que compile el String como parámetro, es decir, esa será la cadena que se buscará en el String de interés. En este caso se analizó el html en un navegador web y se encontró que la cadena "Error de acceso al" se encuentra en él. Por lo tanto, cuando un alumno se identifica de manera incorrecta en el módulo, aparecerá esta cadena como respuesta y es la que se utilizó como referencia para indicarle que se ha cometido un error en su identificación.

Sin menoscabo de lo anterior, se necesitó otra clase, la cual se va a encargar de tomar el objeto creado por Pattern y hace el cotejo de la cadena compilada en la cadena que reciba como parámetro. En este caso se utilizó el String que almacena la información académica para que busque el patrón de interés.

```
Matcher nom buscador = nom patron.matcher("CADENA DONDE SE QUIERE BUSCAR");
```

El método find() devuelve una bandera (o marcador) "true" si se encontró el patrón que se compiló. Si encontró el patrón que buscamos, la aplicación validará que se ha ingresado al módulo de forma correcta y accederá a los datos del usuario.

En caso de no encontrarlo, mandará una ventana emergente para hacer saber que los datos de ingreso han sido incorrectos y limpiará los campos de texto para volver a introducir los datos pertinentes. De lo contrario, se almacenarán los String de Kardex e información académica en archivos de texto para que sea más cómodo trabajar con las secuencias de escape como saltos de línea, tabuladores, etc., que con una cadena muy grande de caracteres. Para ello se necesitó crear archivos de texto como buffers, es decir, donde depositar los String recibidos por la clase Conexión. A continuación como se muestra el código que lleva a cabo dicha operación:

```
FileOutputStream fos = openFileOutput("textField.txt", MODE_PRIVATE);  
OutputStreamWriter osw = new OutputStreamWriter(fos);  
osw.write(htmlInfoAcademica);  
osw.flush();  
osw.close();
```

Una vez creado nuestro objeto de la clase `FileOutputStream` se necesita un objeto que escriba en él y para eso se crea un objeto de la clase `OutputStreamWriter` pasando como parámetro el objeto "fos". Después se llama al método `write` y se pasa como parámetro el String "htmlInfoAcademica", el cual escribirá todo en un archivo .txt. Una vez hecho esto se libera el flujo de salida y se cierra el archivo abierto con `flush` y `close` respectivamente.

Hasta el momento ya está guardado todo el código html en el archivo de texto, ahora se procederá a leer lo que hay en ese archivo, con la diferencia que estará escrito respetando las secuencias de escape. Cuando se crea un objeto de la clase `FileInputStream` que recibe un flujo de datos de archivo de entrada, se crea un objeto llamado "DatosAcade" de la clase que se implementó llamada `DatosAcademicos` y se pasara como parámetro el objeto de la clase `FileInputStream`.

```
DatosAcademicos DatosAcade = new DatosAcademicos();
Datos = DatosAcade.ObtenerDatos(objeto FileInputStream);
```

La clase `DatosAcademicos` descompondrá el archivo de texto para obtener la información que se necesita, la cual se muestra la ilustración 8.

```

DatosAcademicos
private int i, contTrim;
private String pTrim, uTrim, apeM, apeP,
name, mat;
private int CreditosTotales;
private boolean flag;
public boolean nn;
private String[] Datos;

String ObtenerDatos();
String[] DatosModulo();
String InfAcademica()
Void Login(String, String)

```

Ilustración 8 Clase DatosAcademicos().

El método `ObtenerDatos` recibe como parámetro el archivo de flujo de entrada enviado por la Activity, el cual utilizará para abrirlo y guardarlo en un Buffer de tipo `String[]`.

Finalmente se tiene la estructura de datos que se necesita para procesar la información del alumno. Después el mismo método manda a llamar al método `DatosModulo` de la misma clase para iniciar el proceso de análisis.

Dentro de esta clase se crearon objetos patrón, ya que en el html hay identificadores que señalan los datos que se necesitan.

```

Pattern nombre= Pattern.compile("NOMBRE_AL_NO.V_ALUMNO_ACAD.VW01");
Pattern apellidoP= Pattern.compile("APE_PAT_AL_XX.V_ALUMNO_ACAD.VW01");
Pattern apellidoM= Pattern.compile("APE_MAT_AL_XX.V_ALUMNO_ACAD.VW01");
Pattern matricula= Pattern.compile("MATRICULA_CL.V_ALUMNO_ACAD.VW01");
Pattern Licenciatura= Pattern.compile("LICENCIATURA EN INGENIERIA");
Pattern Carrera= Pattern.compile("FISICA");
Pattern primerTrim= Pattern.compile("TRIMESTRE_XX.TRIMESTRE_INI.AE02.1");
Pattern ultimoTrim= Pattern.compile("TRIMESTRE_XX.TRIMESTRE_ACA.AE02.1");
Pattern contadorTrim= Pattern.compile("TRIMESTRE_XX.TRIMESTRE_LEC.AE02.[0-9][0-9]-1|\"TRIMESTRE_XX.TRIMESTRE_LEC.AE02.[0-9]-1\"");
Pattern credAcumulados= Pattern.compile("CRED_ACU_AL_NU.V_ALUMNO_ACAD.VW01");

```

Así como también sus buscadores necesarios para hacer las validaciones de la información como fue descrito en la implementación del ejemplo anterior.

```
Matcher buscaNombre= nombre.matcher(html[i]);
Matcher buscaApellidoP= apellidoP.matcher(html[i]);
Matcher buscaApellidoM= apellidoM.matcher(html[i]);
Matcher buscaMatricula= matricula.matcher(html[i]);
Matcher buscaLic= Licenciatura.matcher(html[i]);
Matcher buscaPrimerTrim= primerTrim.matcher(html[i]);
Matcher buscaUltimoTrim= ultimoTrim.matcher(html[i]);
Matcher buscaCredAcumulados= credAcumulados.matcher(html[i]);
Matcher buscaContadorTrim= contadorTrim.matcher(html[i]);
```

La implementación de la clase DatosAcademicos se encuentra en el Apéndice D anexo a este reporte.

Al finalizar la tarea del objeto “DatosAcade” su valor de retorno da un arreglo de String conteniendo la información del alumno. Para validar si el alumno corresponde a la licenciatura, el objeto “DatosAcade” posee una bandera pública como atributo; si devuelve true, el alumno corresponde a la carrera y mandará a llamar al método “lanzar” y limpiará los campos. De lo contrario mandará una pantalla emergente indicando al alumno que no corresponde a la carrera de Ingeniería Física.

```
if(DatosAcade.perteneceLicenciatura)
{
    lanzar(view);
    matricula.setText("");
    contraseña.setText("");
}
else
{
    Toast.makeText(LoginScreen.this, "\t-Lo sentimos, pero la matricula \n del
alumno no corresponde a la carrera-", Toast.LENGTH_LONG).show();
    matricula.setText("");
    contraseña.setText("");
}
```

El método lanzar manda a llamar a otra Activity llamada “UEAS”, esta sirve de vista para mostrar la recomendación que se realizará al alumno, pasándole como referencia el arreglo con los datos y la cadena sin descomponer del Kardex.

```
public void lanzarSegundaActividad(View view)
{
    Intent i = new Intent(this, UEAS.class);
    i.putExtra("Datos", Datos);
    i.putExtra("htmlKardex", htmlKardex);
    startActivity(i);
}
```

3.4 Módulo de recomendación

Este módulo es el más complejo en cuanto al procesamiento de la información, ya que se hizo el algoritmo de seriación para el tronco profesional, las UEA optativas y el cálculo de créditos mínimos a inscribir.

Retomando el tema, la Activity “LoginScreen” manda a llamar a la Activity “UEAS” y le pasa los parámetros que se mencionaron anteriormente. Para la vista de la actividad se creó otro archivo xml con el Layout necesario para representar los datos de salida en este caso las UEA a recomendar, la implementación de la vista se encuentra en el Apéndice E anexo a este reporte. A continuación en la ilustración 9 se muestran las propiedades de la Actividad UEAS.

```
UEAS
Button botonRegresar;
EditText salida;
int k;
int p2;
public int f;
private String Datos[];
private String htmlKardex;
private String Tabla[][];
private String prioridad[][];
private String prioridad2[][];
private String fin[][];
int priori;
private String Kardex[][];
private String RecomOblig[];
String Buffer;
private String RecomOptOOI[];
private String RecomOptACE[];
private String RecomOptIE[];
private String RecomOptTM[];

public void onCreate();
public String[][] Limpia();
public void cerrar() ;
```

Ilustración 9 Clase UEAS().

Primeramente en la actividad UEAS se debe obtener los parámetros de la otra actividad para ello se crea un objeto “bundle” de la clase Bundle y lo que hace es guardar los parámetros enviados por la otra actividad, posteriormente se obtiene el arreglo de datos y el String del Kardex almacenándolos en las variables locales correspondientes.

```
Bundle bunde= getIntent().getExtras();
Datos=bunde.getStringArray("Datos");
htmlKardex=bunde.getString("htmlKardex");
```

Para trabajar la cadena del Kardex se hará el mismo procedimiento que como se hizo para los datos académicos creando tanto un archivo de texto como el flujo de datos de salida y escribiendo la cadena completa del html del Kardex.

Una vez teniendo el archivo de flujo de entrada de datos se hace el mismo procedimiento que con la información académica, pasando el archivo a un objeto llamado “KardexModulo” de la clase Kardex, el cual su principal tarea será el de retornar en un arreglo con todas las UEA que ha cursado el alumno.

```
Kardex KardexModulo = new Kardex();
Kardex= KardexModulo.ObtenerDatos(objeto FileInputStream);
```

Es decir, la clave, el nombre y la calificación de la UEA. A continuación se muestra la ilustración 10 con las propiedades de la clase Kardex,



Ilustración 10 Clase Kardex().

En el método ObtenerDatos al igual que la clase DatosAcademicos se crea un archivo de entrada de flujo de datos para poder leer el archivo que se pasó como referencia, después se almacena línea por línea en un buffer de almacenamiento para poder procesar la cadena de texto.

```
DataInputStream entrada = new DataInputStream(fstream);
BufferedReader buffer = new BufferedReader(new InputStreamReader(entrada));
while((strLinea[]=buffer.readLine())!=null)
{
    i++;
}
buffer.close();
entrada.close();
fstream.close();
```

Después llamará a otro método llamado Tabla() en que se procesará lo necesario para obtener la tabla de las UEA cursadas. Para comenzar se crearon objetos patrones para buscar los identificadores referidos al nombre, la clave, trimestre en que la llevó, tipo de evaluación y la calificación. También la creación de los buscadores para cada uno de ellos.

```
Pattern claveUEA=Pattern.compile("CVE_UEA_CL.V_HIS_AC_VIG.VW01.[1-9]|CVE_UEA_CL.V_HIS_AC_VIG.VW01.[1-9][0-9]");
Pattern nombreUEA= Pattern.compile("NOM_UEA_NO.E_UEA.PE02.[1-9]|NOM_UEA_NO.E_UEA.PE02.[1-9][0-9]");
Pattern trimUEA= Pattern.compile("TRIMESTRE_XX.TRIMESTRE_LEC.AE02.[1-9]|TRIMESTRE_XX.TRIMESTRE_LEC.AE02.[1-9][0-9]");
Pattern evaluacionUEA= Pattern.compile("TIPO_EVAL_XX.TIPO_EVALUA.AE02.[1-9]|TIPO_EVAL_XX.TIPO_EVALUA.AE02.[1-9][0-9]");
Pattern califUEA= Pattern.compile("CALIFICA_CL.V_HIS_AC_VIG.VW01.[1-9]|CALIFICA_CL.V_HIS_AC_VIG.VW01.[1-9][0-9]");

Matcher buscaClaveUEA= claveUEA.matcher(html[i]);
Matcher buscaNombreUEA= nombreUEA.matcher(html[i]);
Matcher buscaTrimUEA= trimUEA.matcher(html[i]);
Matcher buscaEvaluacionUEA= evaluacionUEA.matcher(html[i]);
Matcher buscaCalifUEA= califUEA.matcher(html[i]);
```

Dando como valor de retorno, el arreglo con cada una de las UEA del alumno a la actividad UEAS. Una vez hecho esto en la actividad se declara un objeto llamado "T" de la clase TablaUEA, el cual llena una tabla con todas las UEA obligatorias y optativas del plan de estudios.

La forma de llenar la tabla tiene un orden especial para el procesamiento que se utilizará más adelante. Antes de comenzar el coordinador de carrera proporcionó un boligrama de las UEA obligatorias de la carrera a las cuales se les asignó un valor numérico, refiriéndose a la prioridad que tienen las UEA por cada trimestre, es decir, una UEA tiene mayor importancia que otra porque tiene más peso en cuanto a la seriación y podría retrasar al alumno de la carrera. En consecuencia, el coordinador estableció esta prioridad a cada una de las UEA analizando cada una para establecer así una recomendación óptima. La prioridad de las UEA optativas tienen un valor fijo y se recomendarán con forme vayan cubriendo sus créditos correspondientes. La ilustración 11 muestra el boligrama con las propiedades correspondientes.

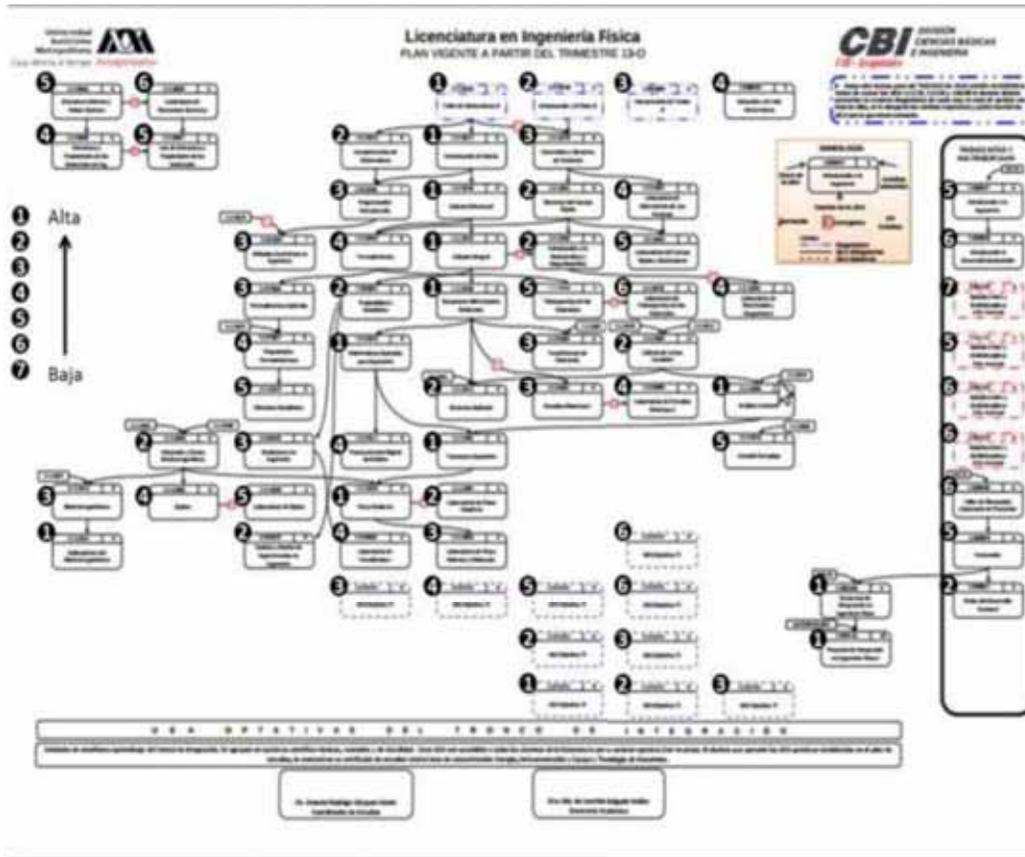


Ilustración 11 UEA obligatorias con prioridad correspondiente a cada trimestre.

El modo en que se programó la seriación fue ubicando ciertos caminos o recorridos principales que hacían cubrir todas la UEA, desde la UEA raíz del primer trimestre hasta donde llegaran en su seriación más distante.

La ilustración 12 muestra las propiedades de la clase TablaUEA:

```

TablaUeas
public String Tabla[][];

TablaUeas();
Tabla();
    
```

Ilustración 12 Clase TablaUEA().

Tiene un arreglo bidimensional de cadenas llamado Tabla y dos métodos. El constructor invoca al método Tabla que se va encargar de llenar el arreglo bidimensional con todo el plan de estudios con las siguientes características: la clave, el nombre, créditos, calificación, trimestre en el que debe cursarse y la

prioridad, esta clase tiene como finalidad mapear las UEA del plan de estudios para que posteriormente se sobre escriban las UEA que ya curso el alumno y sea más fácil la tarea a realizar por el módulo.

Ya teniendo las dos tablas correspondientes al plan de estudios y al Kardex del alumno, se refiere a la clase SeriaciónObligatoria aquella en la cual se crea un objeto llamado "S", en el que el constructor recibe la tabla del plan de estudios para inicializar su variable local con la misma. Con "S" declarado llamamos al método JuntaTablas, que recibe como parámetros la tabla del Kardex, los créditos totales y el número de UEA cursadas.

```
SeriacionObligatoria S = new SeriacionObligatoria(Tabla);
RecomOblig=S.JuntaTablas(Kardex, K.indiceKardex, Integer.parseInt(Datos[7]));
```

El método JuntaTablas sobre escribe la tabla del plan de estudios contra la tabla de las UEA del Kardex, dando como resultado solo una tabla actualizada con las calificaciones y UEA cursadas por el alumno, Este método manda a llamar al método seriacionOblig de la misma clase, el cual se va a encargar de retornar un arreglo con el resultado de la búsqueda de la clave de la UEA a recomendar, es decir, las UEA posibles para el próximo trimestre, en el Apéndice F anexo a este reporte se encuentra el código fuente de la clase SeriaciónObligatoria.

Ahora teniendo las claves de las UEA a recomendar se necesita la información adicional referida a esa UEA así que se crea un String llamado "prioridad" y su respectivo índice "priori" para guardarlas con la información completa buscando en el arreglo Tabla[][].

```
for(int i=0; i<S.indiceRecomendacion; i++)
{
    for(int j=0; j<UEATotalesDelPlan; j++)
    {
        if(RecomOblig[i].equals(Tabla[j][0]))
        {
            prioridad[priori][0]=Tabla[j][0]; //Clave de la UEA
            prioridad[priori][1]=Tabla[j][4]; //Trimestre que debe llevarla
            prioridad[priori][2]=Tabla[j][5]; //Prioridad sobre UEA
            prioridad[priori][3]=Tabla[j][2]; //Créditos
            prioridad[priori][4]=Tabla[j][1]; //Créditos
            priori++;
        }
    }
}
```

Este bucle generó una repetición de 9 caminos en el arreglo de recomendación debido a que existía una UEA que se repetía varias veces, por lo que se implementó un método llamado Limpia. Este método lo que hace es quitar todas las UEA repetidas generadas por la clase SeriacionObligatoria, el cual devuelve un arreglo sin UEA repetidas.

Con el valor de retorno del método Limpia, procedemos a calcular el número mínimo de créditos a inscribir según la trayectoria del alumno y contando con un plan a 13, 18 y 30 trimestres para la conclusión de su carrera dado por las siguientes fórmulas:

$$D.T. = \text{Plan a 13, 18 ó 30 trimestres} - \text{Trimestres cursados} \quad (1)$$

$$D.C. = 491 \text{ creditos totales} - \text{Créditos acumulados} \quad (2)$$

$$\text{CreditosMinimos} = \frac{D.C.}{D.T.} \quad (3)$$

- ❖ CreditosMinimos=Promedio recomendado de créditos para cada trimestre.
- ❖ D.T.= Diferencia de Trimestres.
- ❖ D.C.=Diferencia de Créditos.

Por lo tanto, la creación de esta distinción obligará al alumno a inscribir ese número de créditos obligatorios para concluir la carrera en un número de trimestres considerable, ya que el coordinador de la carrera tiene más interés que sus alumnos tengan mayor atención a las UEA obligatorias que las optativas, ahora bien, si el número de CréditosMínimos supera a los establecidos en la legislación escolar de 45 y 63 dentro de un plan de trimestres, el algoritmo tomará como referencia el siguiente plan de trimestres y hará la distribución de los créditos en esos trimestres restantes.

Los datos de los créditos y trimestres cursados se obtienen del arreglo Datos[] mandados por la otra Activity para hacer la operación de las fórmulas antes mencionadas. Ahora para la utilización de la prioridad se utilizan los valores de la tabla con el plan de estudios. La tabla se ve como una matriz bidimensional, el cual, las filas están integradas por los trimestres en que se deben inscribir las UEA y las columnas son la prioridad asignada a cada UEA.

A continuación, en la ilustración 13 se muestra el algoritmo de recomendación teniendo cada UEA a recomendar en el arreglo sin UEA repetidas, teniendo en cuenta que este algoritmo dará un orden a cada UEA según su posición en el trimestre que debe cursarse y la prioridad sobre otras UEA en el mismo trimestre.

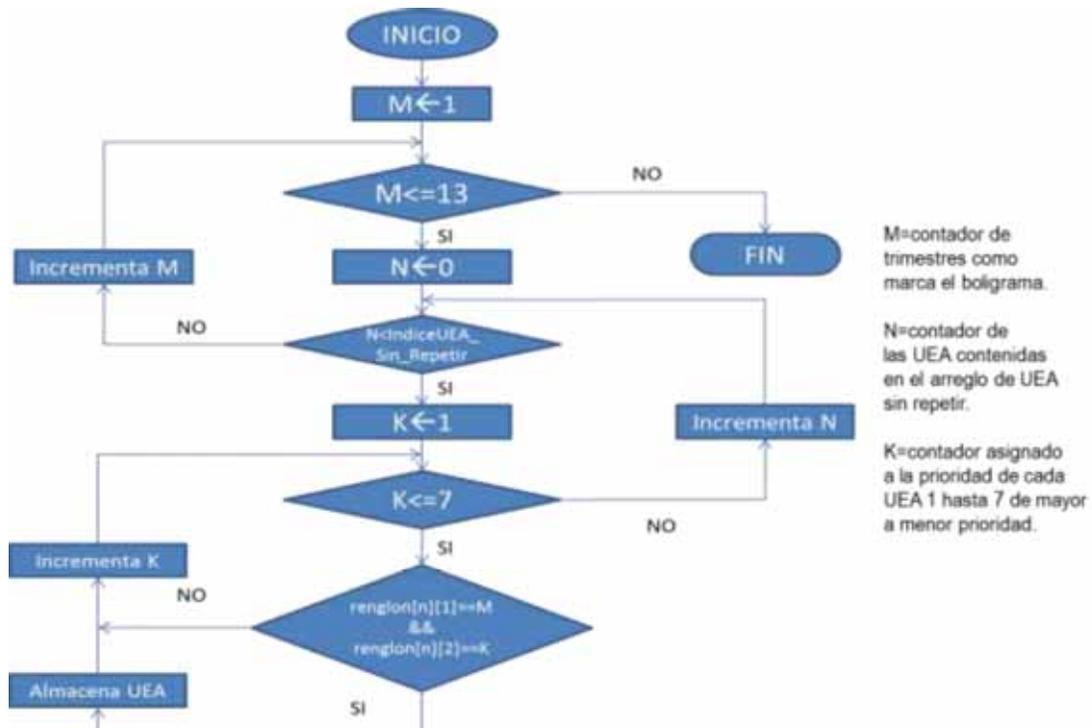


Ilustración 13 Diagrama de flujo del algoritmo de recomendación.

Por último, la asignación de las UEA Optativas se obtienen a través de los métodos de la clase SeriacionOptativa que dan como resultado varios arreglos de cadenas donde se guarda la clave de la UEA posible que puede inscribir de cada una de las áreas de concentración. A continuación en la ilustración 13 se muestran las propiedades de la clase el cual consta de 16 atributos y 9 métodos.

```

SeriationOptativa
private boolean Autorizacion;
private String Tabla[][];
private String ueasRecomOOI[];
public int indiceRecomendacionOOI;
private String ueasRecomACE[];
public int indiceRecomendacionACE;
private String ueasRecomAIE[]=new String[100];
public int indiceRecomendacionAIE;
private String ueasRecomTM[];
public int indiceRecomendacionTM;
private String ueasRecomOT[];
public int indiceRecomendacionOT;
private String ueasRecomOM[];
public int indiceRecomendacionOM;
public int contOpt;
int creditos;

SeriationOptativa(String[][] T, int cred)
String[] OOI()
String[] ACE()
String[] AIE()
String[] TM()
String[] OT()
String[] OM()
void seriationOpt()
boolean buscaOpt(String UEA)

```

Ilustración 14 Clase SeriationOptativa().

Los métodos OOI, ACE, AIE, TM, OM y OT devuelven un arreglo con las UEA de cada área de concentración esto es para hacer la distinción al momento de mostrarlas al alumno, el método SeriationOpt realiza la búsqueda de UEA habilitadas por su seriación y sus créditos; y las almacena en sus arreglos respectivamente. El método buscaOpt busca la UEA si ya está inscrita, si la reprobó o si la aprobó, esto para fines de la recomendación de las UEA Optativas. La implementación de la clase SeriationOptativa se encuentra en el Apéndice G anexo a este reporte.

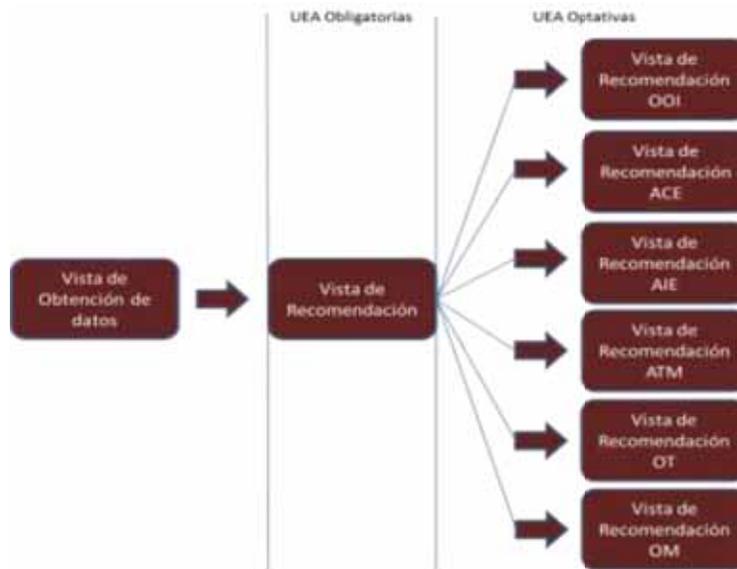
Así pues, cuando se instancian los atributos de la clase SeriationOptativa se guardan en variables locales para poder mostrarlas al alumno. Con ayuda de los bucles "for" almacena las UEA optativas a recomendar; en el caso ya haya cubierto las UEA optativas se desplegará una ventana emergente diciéndole que ya ha cubierto sus UEA optativas.

Finalmente, para pasar al módulo de presentación, previamente ya se tiene la recomendación de las UEA obligatorias y las UEA optativas para poder desplegarlas al usuario.

3.5 Módulo de presentación

En este módulo recaen todas las vistas, operaciones, algoritmos y estructuras de forma transparente al alumno, es decir, se dará una explicación a nivel de vista de cada una de las pantallas mostradas al alumno en el emulador o en su caso en el dispositivo móvil.

El módulo está constituido por las siguientes vistas:



Las abreviaturas de las vistas de recomendación de UEA Optativas corresponden al Tronco de Integración, las cuales se subdividen en OOI "Otras Optativas de Integración", ACE corresponde al "Área de Concentración Energía", AIE corresponde al área de concentración de "Instrumentación y Equipo", ATM corresponde al área de "Tecnología de Materiales", OT corresponde al área de "Optativas Tutoriales" y de OM corresponde a las de Movilidad.

En la Vista de Obtención de datos, como su nombre lo indica, pedirá el número de la matrícula y la contraseña del módulo de información escolar de alumnos de licenciatura, para acceder a la información de los alumnos de Ingeniería Física como se muestra en la ilustración 14. Presionando el botón de Ingresar pasará a la siguiente vista.



Ilustración 15 Vista obtención de datos.

En la Vista de Recomendación se muestran mensajes intuitivos al alumno de la información que se muestra en la pantalla, es decir, en la parte superior de la vista le indica al alumno una lista con cada UEA obligatoria que debe inscribir el siguiente trimestre en un orden de mayor a menor prioridad de arriba hacia abajo e indicando el número de créditos sugeridos.

En la parte baja de la pantalla le indica que presione una imagen que contiene las iniciales de las áreas de concentración que le muestran en cualquiera de las vistas OOI, AIE, ACE, etc. la lista de UEA que puede inscribir en su próximo trimestre de cada área en específico, ver ilustración 15.

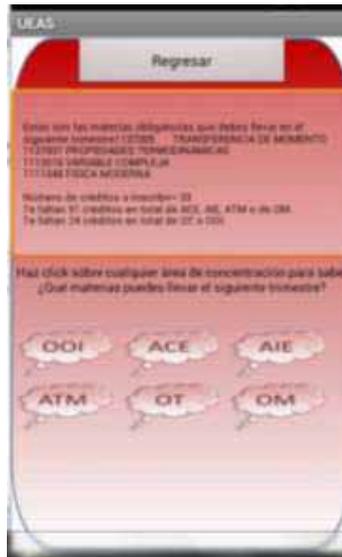


Ilustración 16 Vista de Recomendación

Tocando sobre cada área de concentración se seleccionan las siguientes vistas que es la que muestran la lista de UEA de cada área respectivamente, ver ilustración 16.



Ilustración 17 Vista de Recomendación Optativas.

Con el botón que se encuentra en la parte inferior de la Vista de Recomendación Optativas sirve para regresar a la Vista de Recomendación Obligatorias y seleccionar otra área de concentración.

4. Validación de la aplicación

El coordinador de la carrera proporcionó claves de acceso de alumnos de Ingeniería Física para hacer las pruebas correspondientes de la aplicación y para comprobar su buen funcionamiento. Se teclean los datos de entrada que solicita la aplicación, que son la matrícula y la contraseña, una vez hecho esto se toca el botón de “Ingresar” para generar la recomendación, ver ilustración 17.



Ilustración 18 Introducir datos.

En consecuencia, se despliega otra pantalla mostrando la recomendación de las UEA obligatorias y 6 botones de las áreas de concentración (ver ilustración 18).

Lista de UEA obligatorias que debe inscribir el alumno de Ingeniería Física en el siguiente trimestre.

Y la diferencia de los créditos restantes de las UEA optativas.

Lista de botones que acceden a las lista de UEA optativas de cada área de concentración

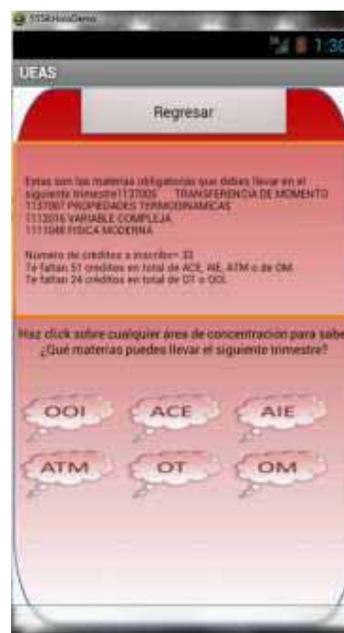


Ilustración 19 Recomendación.

Nota: La datos proporcionados de matrícula y contraseña es información confidencial.

Como se puede observar las UEA obligatorias se muestran de manera que le dé una idea al alumno de lo que puede inscribir en su próximo trimestre, notando que la aplicación tendría sentido utilizarla en los periodos inter-trimestrales, ya que las calificaciones están ya asentadas y los Kardex de los alumnos están actualizados. Además, se le indicará al alumno que ha cubierto el total de UEA optativas si así fuese, de lo contrario se le avisará cuantos créditos le faltan para completar su total mínimo necesario.

Esto significa que el alumno puede tomar 51 créditos de las tres áreas de concentración que corresponde a las optativas técnico-científicas y de movilidad. Ahora bien, los restantes 24 créditos pueden corresponder a las tutoriales y otras optativas. El requisito de los 51 créditos es fundamental para cubrir el 100% de créditos. Por otro lado, cabe resaltar que el alumno puede juntar los 75 créditos de las tres áreas de concentración que corresponden a las técnico-científicas, ya que el programa marca que al menos 51 deben ser de este rubro pero también puede ser todos de este tipo.

En la parte inferior de la pantalla se muestran los botones correspondientes a cada área de concentración con sus respectivas UEA a recomendar que pueda inscribir en el siguiente trimestre, tocando una imagen de la pantalla nos desplegará la lista de UEA a recomendar, ver ilustración 19.



Ilustración 20 Recomendación Optativas.

Los colores y las imágenes presentadas en los fondos de los Layouts fueron creados por el desarrollador con el consentimiento de los asesores. La información en cuanto a las claves de acceso; al finalizar dicho proyecto el coordinador de la carrera de Ingeniería Física deberá pedir a los alumnos que facilitaron sus claves de acceso que cambien su contraseña del modulo de información escolar de alumnos de licenciatura para deslindarse de responsabilidades tanto el coordinador de la carrera como el asesorado.

5. Discusión de resultados

Al comenzar la aplicación hubo problemas para la conexión al servidor ya que se pensaba utilizar una conexión SSL, pero no hubo la necesidad ya que sólo se utilizó el envío de parámetros y guardar la respuesta emitida por el servidor del módulo de información escolar de alumnos de licenciatura. Una vez obtenía la conexión sin problemas al servidor, se decidió ir programando la interfaz grafica de la petición de los datos ya que se utilizó el IDE de Netbeans, pero como no proporciona plantillas predeterminadas, los Layouts se tenían que elaborar desde el inicio. Esto para ir dando forma a la vista de entrada de nuestra aplicación.

Para el procesamiento de los archivos html no hubo la necesidad de hacer un análisis sintáctico como se estableció en la propuesta de este Proyecto de Integración, ya que se optó por la utilización de una clase que reconoce patrones dentro de una cadena de texto disponible dentro del mismo lenguaje, disminuyendo el costo computacional de validar carácter por carácter como lo haría un analizador sintáctico.

La determinación de la prioridad fue fundamental en este proyecto ya que va a proporcionar al alumno de la carrera de Ingeniería Física un panorama práctico de cómo debe recorrer su plan de estudios; pero un problema que fue fundamental para que la aplicación fuese aún más eficiente se encontró con las UEA optativas, y esto debido a que la disponibilidad de las UEA varía dependiendo de muchos factores entre ellos: la disponibilidad de profesores para la impartición de las UEA, la disponibilidad de grupos, la demanda de cierta UEA, el acoplamiento de los horarios con otras UEA, por mencionar algunos.

El coordinador de la carrera proporcionó una propuesta de cómo se distribuyen las UEA optativas pero lamentablemente no se ejecuta como lo plantea, por lo que la recomendación de las UEA optativa solo indica cuantos créditos le faltan al alumno para completar los mínimos necesarios para cubrir el total de UEA optativas indicándole las que puede inscribir de cada área de concentración.

La presentación de la recomendación se hizo con un diseño práctico y conciso, esto es, para que los alumnos solo consulten de manera rápida y sin problemas un tema que podría causar conflicto y confusión.

6. Conclusión

Durante la realización de la aplicación se puede concluir que esta aplicación será de gran utilidad para los alumnos, ya que al momento de programar la seriación obligatoria, hay varios caminos en que el alumno puede incurrir a retrasarse en su formación académica, tal es el caso de la UEA de Ecuaciones Diferenciales Ordinarias, con la prioridad de 1, siendo la más importante del 5° trimestre, es decir, poco menos de la mitad de la carrera.

Esta aplicación es una herramienta base de una más grande, es decir, a futuro se podría incluir los horarios de las UEA de los alumnos, también si se impartirán en el siguiente trimestre, también abrir un módulo de sugerencias que notifique al coordinador que UEA son las más demandadas en el siguiente trimestre para abrir grupos extras. Y así mantener una mayor comunicación no tan personal pero si tecnológica. Todas las aplicaciones móviles tienden a ser publicadas en las redes sociales, llamese como se llame, pero por cuestiones de tiempo se podría proponer la inclusión de compartir las recomendaciones emitidas por esta aplicación en la nube.

También se puede migrar a las diferentes versiones de Android e incluso a dispositivos móviles diferentes como Windows Mobile, Apple y BlackBerry. La elaboración de este proyecto fue satisfactoria y con aún más caminos de investigación para la administración escolar de los alumnos de la Universidad Autónoma Metropolitana en cuanto al aumento de la disponibilidad de los horarios, los profesores y disminuir los incidentes y requerimientos que se suscitan al momento de las inscripciones y las saturaciones de los grupos de muy alta demanda.

7. Bibliografía

- [1] Android. [En línea]. Available: <http://www.tutorialandroid.com/basico/como-chequear-si-existe-conexion-a-internet/>.
- [2] JavaYa. [En línea]. Available: <http://www.javaya.com.ar/androidya/detalleconcepto.php?codigo=144&inicio=>.
- [3] T. HTML. [En línea]. Available: <http://www.htmlquick.com/es/tutorials/tables.html>.
- [4] SDK. [En línea]. Available: <http://theandroidsdk.blogspot.mx/2013/02/how-to-get-source-code-from-apk-file-in.html>.
- [5] A. Developers. [En línea]. Available: <http://developer.android.com/index.html>.
- [6] P. D. a. H. Deitel, Como programar en Java, México, 2008, pp. 993-1035.
- [7] J. Lopez, HTML Y DHTML, AlfaOmega, 2008.
- [8] C. U. Azc. [En línea]. Available: <http://cbi.azc.uam.mx/planes/archivos/Ing Fisica/Ing Fisica.pdf>.


```

        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:text="">
    </TextView>
</TableRow>
<TableRow>
    <TextView
        android:id="@+id/TextoMat"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:text="Matrícula"
        android:textColor="#AFFFFF"
        android:maxLines="1"
        android:ellipsize="marquee"
        android:fadingEdge="horizontal"
        android:marqueeRepeatLimit="marquee_forever"
        android:scrollHorizontally="true">
    </TextView>
    <EditText
        android:id="@+id/CampoTextoMatricula"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="Ingrese su matrícula"
        android:numeric="integer">
    </EditText>
</TableRow>
<TableRow>
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="center_horizontal"
        android:text="Contraseña"
        android:textColor="#AFFDFF">
    </TextView>
    <EditText
        android:id="@+id/CampoTextoContraseña"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:hint="Ingrese su contraseña"
        android:inputType="text"
        android:password="true">
    </EditText>
</TableRow>
<Button
    android:id="@+id/boton"
    android:text="Ingresar"
    android:layout_width="200dp"

```

```
        android:layout_height="wrap_content"
        android:gravity="center">
    </Button>
</TableLayout>
```

B. Implementación de la validación de conexión a internet en Android.

```
import android.content.Context;
import android.net.ConnectivityManager;
import android.net.NetworkInfo;

public class LoginScreen extends Activity
{
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }

    public boolean existeConexionInternet()
    {
        ConnectivityManager cm = (ConnectivityManager)
        getSystemService(Context.CONNECTIVITY_SERVICE);
        NetworkInfo netInfo = cm.getActiveNetworkInfo();
        if (netInfo != null && netInfo.isConnectedOrConnecting())
        {
            return true;
        }
        return false;
    }
}
```

C. Implementación de la clase Conexión.

```
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.util.ArrayList;
import java.util.List;
import java.util.List;
import org.apache.http.Header;
import org.apache.http.HttpEntity;
import org.apache.http.HttpResponse;
import org.apache.http.NameValuePair;
import org.apache.http.ParseException;
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.impl.client.DefaultHttpClient;
import org.apache.http.message.BasicNameValuePair;
import org.apache.http.util.EntityUtils;

public class Conexion
{
    private String Kardex="";
    private String Informacion_Academica="";

    String Kardex()
    {
        return Kardex;
    }

    String InfAcademica()
    {
        return Informacion_Academica;
    }

    void Login(String Matricula, String Password)
    {
        DefaultHttpClient httpclient = new DefaultHttpClient();
        List<NameValuePair> params=new ArrayList<NameValuePair>();
        params.add(new BasicNameValuePair("SIGLAS_UNI_XX.E_UNIDAD.AE02.1","AZC"));
        params.add(new BasicNameValuePair("%23.E_UNIDAD.AE02.1","AxJDMQ%3D%3D"));
        params.add(new BasicNameValuePair("%23CRC.E_UNIDAD.AE02.1","00000024"));
        params.add(new BasicNameValuePair("NOMBRE.IDENTIFICACION.NONMODELED",
Matricula));
        params.add(new
BasicNameValuePair("COMPLEMENTO.IDENTIFICACION.NONMODELED",Password));
        params.add(new BasicNameValuePair("GO.IDENTIFICACION.NONMODELED","Entrar"));
        params.add(new BasicNameValuePair("%25.IDENTIFICACION.NONMODELED",""));
        params.add(new BasicNameValuePair("%23.WEB_INFO.SW01",""));
        params.add(new BasicNameValuePair("%23.WEB_MOD_ASO.SW01",""));
    }
}
```

```

        params.add(new BasicNameValuePair("%23.USUARIO_ANEXO.SG02", ""));
        params.add(new BasicNameValuePair("%23.MODULO_UWAS.SAE01", ""));
        HttpPost httpPost = new
HttpPost("https://ayamictlan.uam.mx:8443/sae/azc/AEWBU004.oInicioSesion?mod=1");
    try
    {
        httpPost.setEntity(new UrlEncodedFormEntity(params));
        HttpResponse responsePost = null;
        try{responsePost = httpclient.execute(httpPost);}
        catch(ClientProtocolException e)
        {e.printStackTrace();}
        catch(IOException e)
        {e.printStackTrace();}
        Header[] cookies = responsePost.getHeaders("Set-Cookie");
        HttpGet httpget = new
HttpGet("https://ayamictlan.uam.mx:8443/sae/azc/IEWBC020.oConsulta");
        HttpGet httpget2 = new
HttpGet("https://ayamictlan.uam.mx:8443/sae/azc/IEWBC007.oConsulta");
        for(Header c : cookies)
        {httpget.addHeader("Cookie", c.getValue());}
        for (Header c : cookies)
        {httpget2.addHeader("Cookie", c.getValue());}
        HttpResponse responseGet = null;
        try
        {
            responseGet = httpclient.execute(httpget);
        }
        catch(ClientProtocolException e)
        {
            e.printStackTrace();
        }
        catch(IOException e)
        {
            e.printStackTrace();
        }
        HttpEntity ent=responseGet.getEntity();

        try
        {
            Kardex=EntityUtils.toString(ent);
        }
        catch(ParseException e)
        {
            e.printStackTrace();
        }
        catch(IOException e)
        {
            e.printStackTrace();
        }
    }

```

```

    }
    try
    {
        responseGet = httpClient.execute(httpget2);
    }
    catch(ClientProtocolException e)
    {
        e.printStackTrace();
    }
    catch(IOException e)
    {
        e.printStackTrace();
    }
    ent=responseGet.getEntity();
    try
    {
        Informacion_Academica=EntityUtils.toString(ent);
        //return Informacion_Academica;
    }
    catch(ParseException e)
    {
        e.printStackTrace();
    }
    catch (IOException e)
    {
        e.printStackTrace();
    }
}
catch(UnsupportedEncodingException e)
{
    e.printStackTrace();
}
}
}

```

D. Implementación de la clase DatosAcademicos.

```
import java.io.*;
import java.util.StringTokenizer;
import java.util.regex.*;

public class DatosAcademicos
{
    private int i=0, contTrim=0;
    private String pTrim, uTrim, apeM, apeP, name, mat;
    private int CreditosTotales=0;
    private boolean flag;
    public boolean nn=true;
    private String[] Datos= new String[10];

    String[] ObtenerDatos(FileInputStream fstream)
    {
        String strLinea[]= new String[10000];
        int i=0;
        try
        {
            DataInputStream entrada = new DataInputStream(fstream);
            BufferedReader buffer = new BufferedReader(new InputStreamReader(entrada));
            while((strLinea[i]=buffer.readLine())!=null)
            {
                i++;
            }
            buffer.close();
            entrada.close();
        }
        catch (Exception e)
        {
            System.err.println("Ocurrio un error: " + e.getMessage());
        }
        return DatosModulo(strLinea);
    }

    String[] DatosModulo(String html[])
    {
        Pattern nombre= Pattern.compile("NOMBRE_AL_NO.V_ALUMNO_ACAD.VW01");
        Pattern apellidoP= Pattern.compile("APE_PAT_AL_XX.V_ALUMNO_ACAD.VW01");
        Pattern apellidoM= Pattern.compile("APE_MAT_AL_XX.V_ALUMNO_ACAD.VW01");
        Pattern matricula= Pattern.compile("MATRICULA_CL.V_ALUMNO_ACAD.VW01");
        Pattern Licenciatura= Pattern.compile("LICENCIATURA EN INGENIERIA");
        Pattern Carrera= Pattern.compile("FISICA");
        Pattern primerTrim= Pattern.compile("TRIMESTRE_XX.TRIMESTRE_INI.AE02.1");
        Pattern ultimoTrim= Pattern.compile("TRIMESTRE_XX.TRIMESTRE_ACA.AE02.1");
        Pattern contadorTrim= Pattern.compile("TRIMESTRE_XX.TRIMESTRE_LEC.AE02.[0-9][0-9]-1|\"TRIMESTRE_XX.TRIMESTRE_LEC.AE02.[0-9]-1\\\"");
```

```

Pattern credAcumulados= Pattern.compile("CRED_ACU_AL_NU.V_ALUMNO_ACAD.VW01");
while(html[i]!=null)
{
    Matcher buscaNombre= nombre.matcher(html[i]);
    Matcher buscaApellidoP= apellidoP.matcher(html[i]);
    Matcher buscaApellidoM= apellidoM.matcher(html[i]);
    Matcher buscaMatricula= matricula.matcher(html[i]);
    if(flag=buscaApellidoP.find()==true)
    {
        StringTokenizer token= new StringTokenizer(html[i]);
        while(token.hasMoreTokens()!=false)
        {
            String aux;
            aux=token.nextToken("\\");
            if(aux.equals(" value=")||aux.equals("value="))
            {
                apeP=token.nextToken("\\");
                Datos[2]=apeP;
            }
        }
        i++;
    }
    else if(flag=buscaApellidoM.find()==true)
    {
        StringTokenizer token= new StringTokenizer(html[i]);
        while(token.hasMoreTokens()!=false)
        {
            String aux;
            aux=token.nextToken("\\");
            if(aux.equals(" value=")||aux.equals("value="))
            {
                apeM=token.nextToken("\\");
                Datos[1]=apeM;
            }
        }
        i++;
    }
    else if(flag=buscaMatricula.find()==true)
    {
        StringTokenizer token= new StringTokenizer(html[i]);
        while(token.hasMoreTokens()!=false)
        {
            String aux;
            aux=token.nextToken("\\");
            if(aux.equals(" value=")||aux.equals("value="))
            {
                mat=token.nextToken("\\");
                Datos[3]=mat;
            }
        }
    }
}

```

```

    }
  }
  i++;
}
else if(flag=buscaNombre.find()==true)
{
  StringTokenizer token= new StringTokenizer(html[i]);
  while(token.hasMoreTokens()!=false)
  {
    String aux;
    aux=token.nextToken("\\");
    if(aux.equals(" value=")||aux.equals("value="))
    {
      name=token.nextToken("\\");
      Datos[0]=name;
    }
  }
  i++;
  break;
}
else
{
  i++;
}
}
while(html[i]!=null)
{
  Matcher buscaLic= Licenciatura.matcher(html[i]);
  if((flag=buscaLic.find())==true)
  {
    Matcher buscaCarrera= Carrera.matcher(html[i]);
    if((flag=buscaCarrera.find())==true)
    {
      while(html[i]!=null)
      {
        Matcher buscaPrimerTrim= primerTrim.matcher(html[i]);
        Matcher buscaUltimoTrim= ultimoTrim.matcher(html[i]);
        Matcher buscaCredAcumulados= credAcumulados.matcher(html[i]);
        if((flag=buscaPrimerTrim.find())==true)
        {
          primerTrim=Pattern.compile("value=\\[0-9][0-9][POI]\\");
          buscaPrimerTrim= primerTrim.matcher(html[i]);
          if((flag=buscaPrimerTrim.find())==true)
          {
            primerTrim=Pattern.compile("TRIMESTRE_XX.TRIMESTRE_INI.AE02.1");
            buscaPrimerTrim= primerTrim.matcher(html[i]);
            if((flag=buscaPrimerTrim.find())==true)
            {

```

```

StringTokenizer token= new StringTokenizer(html[i]);
while(token.hasMoreTokens()!=false)
{
    pTrim=token.nextToken();
    if(pTrim.length()==11)
    {
        Datos[4]=pTrim.substring(7, 10);
    }
}
}
}
i++;
}
else if((flag=buscaUltimoTrim.find())==true)
{
    ultimoTrim=Pattern.compile("value=\"[0-9][0-9][POI]\"");
    buscaUltimoTrim= ultimoTrim.matcher(html[i]);
    if((flag=buscaUltimoTrim.find())==true)
    {
        primerTrim=Pattern.compile("TRIMESTRE_XX.TRIMESTRE_ACA.AE02.1");
        buscaUltimoTrim= ultimoTrim.matcher(html[i]);
        if((flag=buscaUltimoTrim.find())==true)
        {
            StringTokenizer token= new StringTokenizer(html[i]);
            while(token.hasMoreTokens()!=false)
            {
                uTrim=token.nextToken();
                if(uTrim.length()==11)
                {
                    Datos[5]=uTrim.substring(7, 10);
                }
            }
        }
    }
    i++;
}
else if((flag=buscaCredAcumulados.find())==true)
{
    credAcumulados=Pattern.compile("value=\"[0-9][0-9]\"|value=\"[0-9][0-9][0-9]\"");
    buscaCredAcumulados= credAcumulados.matcher(html[i]);
    if((flag=buscaCredAcumulados.find())==true)
    {
        credAcumulados=Pattern.compile("CRED_ACU_AL_NU.V_ALUMNO_ACAD.VW01");
        buscaCredAcumulados= credAcumulados.matcher(html[i]);
        if((flag=buscaCredAcumulados.find())==true)
        {
            StringTokenizer token= new StringTokenizer(html[i]);
            while(token.hasMoreTokens()!=false)

```

```

        {
            String a;
            a=token.nextToken();
            if(a.length()==11)
            {
                CreditosTotales=Integer.parseInt(a.substring(7, 10));
            }
            if(a.length()==10)
            {
                CreditosTotales=Integer.parseInt(a.substring(7, 9));
            }
        }
    }
    }
    i++;
    break;
}
else
{
    i++;
}
}
}
else
{
    System.out.println("El alumno no corresponde al plan de estudios");
    nn=false;
}
break;
}
i++;
}
while(html[i]!=null)
{

    Matcher buscaContadorTrim= contadorTrim.matcher(html[i]);
    if((flag=buscaContadorTrim.find())==true)
    {
        while(html[i]!=null)
        {
            contadorTrim=Pattern.compile("\\\\TRIMESTRE_XX.TRIMESTRE_LEC.AE02.[0-9][0-9]-
1\\|\\\\TRIMESTRE_XX.TRIMESTRE_LEC.AE02.[0-9]-1\\");
            buscaContadorTrim= contadorTrim.matcher(html[i]);
            if((flag=buscaContadorTrim.find())==true)
            {
                StringTokenizer token= new StringTokenizer(html[i]);
                while(token.hasMoreTokens()!=false)
                {

```

```

        uTrim=token.nextToken();
        if(uTrim.length()==11)
        {
            contTrim++;
        }
    }
    i++;
}
else
{
    i++;
}
}
}
i++;
}
Datos[6]=String.valueOf(contTrim);
Datos[7]=String.valueOf(Creditos Totales);
return Datos;
}
}

```

E. Implementación de la vista UEAS.

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:gravity="center_horizontal"
    android:background="@drawable/fondo2"
    >
    <Button
        android:id="@+id/botonRegresar"
        android:text="Regresar"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:onClick="Regresar">
    </Button>
    <EditText
        android:id="@+id/datos"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:lines="13"
        android:enabled="false"
        android:textSize="10sp"
        android:textColor="#000000">

```

```

</EditText>
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:gravity="center_horizontal"
    android:textSize="12sp"
    android:textColor="#000000"
    android:text="Haz click sobre cualquier área de concentración para saber ¿Qué materias puedes
llevar el siguiente trimestre?">
</TextView>
<Button
    android:id="@+id/botonOOI"
    android:layout_width="100dp"
    android:layout_height="50dp"
    android:gravity="center"
    android:onClick="botonOOI"
    android:background="@drawable/botonooi">
</Button>
<Button
    android:id="@+id/botonACE"
    android:layout_width="100dp"
    android:layout_height="50dp"
    android:gravity="center"
    android:onClick="botonACE"
    android:background="@drawable/botonace">
</Button>
<Button
    android:id="@+id/botonIE"
    android:layout_width="100dp"
    android:layout_height="50dp"
    android:gravity="center"
    android:onClick="botonIE"
    android:background="@drawable/botonie">
</Button>
<Button
    android:id="@+id/botonTM"
    android:layout_width="100dp"
    android:layout_height="50dp"
    android:gravity="center"
    android:onClick="botonTM"
    android:background="@drawable/botontm">
</Button>
<Button
    android:id="@+id/botonOT"
    android:layout_width="100dp"
    android:layout_height="50dp"
    android:gravity="center"
    android:onClick="botonOT"

```

```

        android:background="@drawable/botonot">
    </Button>
</LinearLayout>

```

F. Implementación de la clase SeriacionObligatoria

```

public class SeriacionObligatoria
{
    public String ueasRecomendadas[]=new String[100];
    public int indiceRecomendacion=0;
    public String Tabla[][]= new String[210][6];//Guarda todas la UEAS del plan de estudios

    SeriacionObligatoria(String[][] T)
    {
        Tabla=T;
    }

    String[] JuntaTablas(String ueasModulo[], int totalUeas, int creditos)//El metodo recibe todas la UEAS
    del modulo y el numero de Ueas totales.
    {
        int indiceUeasModulo;
        int indiceTabla;
        for(indiceUeasModulo=0; indiceUeasModulo<totalUeas; indiceUeasModulo++)
        {
            for(indiceTabla=0; indiceTabla<200; indiceTabla++)
            {
                if(ueasModulo[indiceUeasModulo][0].equals(Tabla[indiceTabla][0]))
                {
                    Tabla[indiceTabla][3]=ueasModulo[indiceUeasModulo][4];
                }
            }
        }
        return seriacionOblig(creditos);
    }

    String[] seriacionOblig(int creditos)
    {
        int cont;
        int band=0;
        if(!Tabla[28][3].equals("NA")&&!Tabla[28][3].equals(""))
        {
            System.out.println("ya no tiene seriacion0");
        }
        else
        {
            ueasRecomendadas[indiceRecomendacion]=Tabla[28][0];
            indiceRecomendacion++;
        }
        if(!Tabla[0][3].equals("NA")&&!Tabla[0][3].equals(""))

```

```

{
    System.out.println("ya no tiene seriacion000");
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[0][0];
    indiceRecomendacion++;
}
if(creditos>=50)
{
    if(!Tabla[24][3].equals("NA")&&!Tabla[24][3].equals(""))
    {
        if(!Tabla[25][3].equals("NA")&&!Tabla[25][3].equals(""))
        {
            System.out.println("ya no tiene seriacion0000000");
        }
        else
        {
            ueasRecomendadas[indiceRecomendacion]=Tabla[0][0];
            indiceRecomendacion++;
        }
    }
    else
    {
        ueasRecomendadas[indiceRecomendacion]=Tabla[0][0];
        indiceRecomendacion++;
    }
}
for(cont=0; cont<200; cont++)
{
    if(Tabla[cont][0].equals("1113084"))
    {
        if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
        {
            for( ;cont<200; cont++)
            {
                if(Tabla[cont][0].equals("1113085"))
                {
                    if(!Tabla[cont][3].equals("NA") && !Tabla[cont][3].equals(""))
                    {
                        band++;
                    }
                    else
                    {
                        ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
                        indiceRecomendacion++;
                    }
                }
            }
        }
    }
}

```

```

if(Tabla[cont][0].equals("1113086"))
{
    if(!Tabla[cont][3].equals("NA") && !Tabla[cont][3].equals(""))
    {
        band++;
    }
    else
    {
        ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
        indiceRecomendacion++;
    }
}
if(band==2)
{

    for( ;cont<200; cont++)
    {
        if(Tabla[cont][0].equals("1113087"))
        {
            if(!Tabla[cont][3].equals("NA") && !Tabla[cont][3].equals(""))
            {
                System.out.println("ya no tiene seriacion111111");
            }
            else
            {
                ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
                indiceRecomendacion++;
            }
        }
    }
}
break;
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
}
for(cont=0; cont<200; cont++)
{
    if(Tabla[cont][0].equals("1112026"))
    {
        if(!Tabla[cont][3].equals("NA") && !Tabla[cont][3].equals(""))
        {

```



```

    }
    else
    {
        ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
        indiceRecomendacion++;
        break;
    }
}
}
break;
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
for(cont=0; cont<200; cont++)
{
    if(Tabla[cont][0].equals("1112026"))
    {
        if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
        {
            for( ; cont<200; cont++)
            {
                if(Tabla[cont][0].equals("1112027"))
                {
                    if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                    {
                        for( ; cont<200; cont++)
                        {
                            if(Tabla[cont][0].equals("1112028"))
                            {
                                if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                                {
                                    for( ; cont<200; cont++)
                                    {
                                        if(Tabla[cont][0].equals("1112029"))
                                        {
                                            if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                                            {
                                                for( ; cont<200; cont++)
                                                {
                                                    if(Tabla[cont][0].equals("1112030"))
                                                    {

```



```

        {
            ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
            indiceRecomendacion++;
            break;
        }
    }
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
}
for(cont=0; cont<200; cont++)
{
    if(Tabla[cont][0].equals("1112026"))
    {
        if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
        {
            for( ; cont<200; cont++)
            {
                if(Tabla[cont][0].equals("1112027"))
                {
                    if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                    {
                        for( ; cont<200; cont++)
                        {
                            if(Tabla[cont][0].equals("1112028"))
                            {
                                if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                                {
                                    for( ; cont<200; cont++)
                                    {
                                        if(Tabla[cont][0].equals("1112029"))
                                        {

```

```

if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
{
for( ; cont<200; cont++)
{
if(Tabla[cont][0].equals("1153001"))
{
if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
{

for( ; cont<200; cont++)
{
if(Tabla[cont][0].equals("1154029"))
{
if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
{
System.out.println("ya no tiene seriacion4444444");
break;
}
else
{

ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
indiceRecomendacion++;
break;
}
}
}
}
}
else
{
ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
indiceRecomendacion++;
break;
}
}
}
}
else
{

```

```

        ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
        indiceRecomendacion++;
        break;
    }
}
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
}
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
}
for(cont=0; cont<200; cont++)
{
    if(Tabla[cont][0].equals("1112026"))
    {
        if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
        {
            for( ; cont<200; cont++)
            {
                if(Tabla[cont][0].equals("1112027"))
                {
                    if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                    {
                        for( ; cont<200; cont++)
                        {
                            if(Tabla[cont][0].equals("1112028"))
                            {
                                if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                                {
                                    for( ; cont<200; cont++)
                                    {
                                        if(Tabla[cont][0].equals("1112029"))
                                        {
                                            if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))

```



```

        break;
    }
}
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
}
for(cont=0; cont<200; cont++)
{
    if(Tabla[cont][0].equals("1112026"))
    {
        if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
        {
            for( ; cont<200; cont++)

```

```

{
    if(Tabla[cont][0].equals("1112027"))
    {
        if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
        {
            for( ; cont<200; cont++)
            {
                if(Tabla[cont][0].equals("1112028"))
                {
                    if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals("") &&
(buscaObligatorias("1111081")==true))
                    {
                        for( ; cont<200; cont++)
                        {
                            if(Tabla[cont][0].equals("1113046"))
                            {
                                if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                                {
                                    for( ; cont<200; cont++)
                                    {
                                        if(Tabla[cont][0].equals("1113069"))
                                        {
                                            if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                                            {
                                                for( ; cont<200; cont++)
                                                {
                                                    if(Tabla[cont][0].equals("1113070"))
                                                    {
                                                        if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                                                        {
                                                            System.out.println("ya no tiene seriacion6666666");
                                                            break;
                                                        }
                                                    }
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
else
{
    if(Tabla[cont][3].equals("NA"))
    {
        ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    }
}

```

```

        indiceRecomendacion++;
        break;
    }
    else if(Tabla[cont][3].equals(""))
    {
        ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
        indiceRecomendacion++;
        ueasRecomendadas[indiceRecomendacion]="1113070";
        indiceRecomendacion++;
        break;
    }
    }
    }
    }
    }
    else
    {
        ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
        indiceRecomendacion++;
        break;
    }
    }
    }
    }
    }
    else
    {
        if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
        {
            break;
        }
        else
        {
            ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
            indiceRecomendacion++;
            break;
        }
    }
    }
    }
    }
    }
    else
    {
        ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
        indiceRecomendacion++;
        break;
    }
}
}

```

```

    }
  }
  else
  {
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
  }
}
}
for(cont=0; cont<200; cont++)
{
  if(Tabla[cont][0].equals("1112026"))
  {
    if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
    {
      for( ; cont<200; cont++)
      {
        if(Tabla[cont][0].equals("1112027"))
        {
          if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
          {
            for( ; cont<200; cont++)
            {
              if(Tabla[cont][0].equals("1112028"))
              {
                if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                {
                  for( ; cont<200; cont++)
                  {
                    if(Tabla[cont][0].equals("1112029"))
                    {
                      if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                      {
                        for( ; cont<200; cont++)
                        {
                          if(Tabla[cont][0].equals("1112030"))
                          {
                            if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                            {
                              for( ; cont<200; cont++)
                              {
                                if(Tabla[cont][0].equals("1112015"))
                                {
                                  if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                                  {
                                    for( ; cont<200; cont++)

```



```

        else
        {
            ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
            indiceRecomendacion++;
            break;
        }
    }
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
}
for(cont=0; cont<200; cont++)
{
    if(Tabla[cont][0].equals("1112026"))
    {
        if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
        {
            for( ; cont<200; cont++)
            {
                if(Tabla[cont][0].equals("1112027"))
                {
                    if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                    {
                        for( ; cont<200; cont++)
                        {
                            if(Tabla[cont][0].equals("1112028"))
                            {
                                if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                                {
                                    for( ; cont<200; cont++)
                                    {
                                        if(Tabla[cont][0].equals("1112029"))

```

```

{
    if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
    {
        for( ; cont<200; cont++)
        {
            if(Tabla[cont][0].equals("1112030"))
            {
                if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                {

                    for( ; cont<200; cont++)
                    {
                        if(Tabla[cont][0].equals("1112015"))
                        {
                            if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                            {
                                for( ; cont<200; cont++)
                                {
                                    if(Tabla[cont][0].equals("1111091"))
                                    {

                                        if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals("")&& (buscaObligatorias("1111090")==true))
                                        {
                                            for( ; cont<200; cont++)
                                            {
                                                if(Tabla[cont][0].equals("1111048"))
                                                {

                                                    if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                                                    {
                                                        for( ; cont<200; cont++)
                                                        {
                                                            if(Tabla[cont][0].equals("1111088"))
                                                            {

                                                                if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                                                                {
                                                                    System.out.println("ya no tiene
seriacion8888888888888888");

                                                                    break;
                                                                }
                                                                else
                                                                {

                                                                    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];

                                                                    indiceRecomendacion++;
                                                                    break;
                                                                }
                                                            }
                                                        }
                                                    }
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

    }
  }
  for( cont=40; cont<200; cont++)
  {
    if(Tabla[cont][0].equals("1111087"))
    {
      if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
      {
        break;
      }
      else
      {
        ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
        indiceRecomendacion++;
        break;
      }
    }
  }
  else
  {
    if(Tabla[cont][3].equals("NA"))
    {
      ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
      indiceRecomendacion++;
      break;
    }
    else if(Tabla[cont][3].equals(""))
    {
      ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
      indiceRecomendacion++;
    }
  }
  ueasRecomendadas[indiceRecomendacion]="1111087";
  indiceRecomendacion++;
  break;
}
}
}
}
else
{

```



```

                break;
            }
        }
    }
    else
    {
        ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
        indiceRecomendacion++;
        break;
    }
}
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
}
for(cont=0; cont<200; cont++)
{
    if(Tabla[cont][0].equals("1112026"))
    {
        if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
        {
            for( ; cont<200; cont++)
            {
                if(Tabla[cont][0].equals("1112027"))
                {
                    if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                    {
                        for( ; cont<200; cont++)
                        {
                            if(Tabla[cont][0].equals("1112028"))
                            {
                                if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                                {
                                    for( ; cont<200; cont++)
                                    {
                                        if(Tabla[cont][0].equals("1112029"))
                                        {
                                            if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                                            {
                                                for( ; cont<200; cont++)

```

```

        {
            if(Tabla[cont][0].equals("1112030"))
            {
                if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals("")&&(buscaObligatorias("1111081")==true) &&
                (buscaObligatorias("1112005")==true))
                {
                    for( ; cont<200; cont++)
                    {
                        if(Tabla[cont][0].equals("1111013"))
                        {
                            if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                            {
                                System.out.println("ya no tiene
seriacion999999999999");
                                break;
                            }
                            else
                            {
                                ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
                                indiceRecomendacion++;
                                break;
                            }
                        }
                    }
                }
            }
            else
            {
                if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                {
                    break;
                }
                else
                {
                    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
                    indiceRecomendacion++;
                    break;
                }
            }
        }
    }
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}

```

```

        }
    }
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
}
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
}
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
}
}
for(cont=0; cont<200; cont++)
{
    if(Tabla[cont][0].equals("1112026"))
    {
        if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
        {
            for( ; cont<200; cont++)
            {
                if(Tabla[cont][0].equals("1112027"))
                {
                    if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                    {
                        for( ; cont<200; cont++)
                        {
                            if(Tabla[cont][0].equals("1112028"))
                            {
                                if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))

```

```

{
  for( ; cont<200; cont++)
  {
    if(Tabla[cont][0].equals("1112029"))
    {
      if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
      {
        for( ; cont<200; cont++)
        {
          if(Tabla[cont][0].equals("1112030"))
          {
            if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
            {
              for( ; cont<200; cont++)
              {
                if(Tabla[cont][0].equals("1124001"))
                {
                  if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                  {
                    for( ; cont<200; cont++)
                    {
                      if(Tabla[cont][0].equals("1124005"))
                      {

if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                        {
                          System.out.println("ya no tiene seriacion10");
                        }
                      else
                      {

ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
                          indiceRecomendacion++;
                          break;
                        }
                      }
                    }
                  }
                }
              else
              {
                if(Tabla[cont][3].equals("NA"))
                {

ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
                          indiceRecomendacion++;
                          break;
                        }
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}

```

```

else if(Tabla[cont][3].equals(""))
{
ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
indiceRecomendacion++;

ueasRecomendadas[indiceRecomendacion]="1124005";
indiceRecomendacion++;
break;
}
}
}
}
}
else
{
if(Tabla[cont][3].equals("NA"))
{
ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
indiceRecomendacion++;
break;
}
else if(Tabla[cont][3].equals(""))
{
ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
indiceRecomendacion++;
ueasRecomendadas[indiceRecomendacion]="1124001";
indiceRecomendacion++;
ueasRecomendadas[indiceRecomendacion]="1124005";
indiceRecomendacion++;
break;
}
}
}
}
}
else
{
ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
indiceRecomendacion++;
break;
}
}
}
}
else
{

```

```

        ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
        indiceRecomendacion++;
        break;
    }
}
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
}
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
}
for(cont=0; cont<200; cont++)
{
    if(Tabla[cont][0].equals("1112026"))
    {
        if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
        {
            for( ; cont<200; cont++)
            {
                if(Tabla[cont][0].equals("1112027"))
                {
                    if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                    {
                        for( ; cont<200; cont++)
                        {
                            if(Tabla[cont][0].equals("1112028"))
                            {
                                if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals("") &&
(buscaObligatorias("1111081")==true))
                                {
                                    for( ; cont<200; cont++)
                                    {
                                        if(Tabla[cont][0].equals("1113046"))
                                        {

```

```

        if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
        {
            for( ; cont<200; cont++)
            {
                if(Tabla[cont][0].equals("1137006"))
                {
                    if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals("")&&(buscaObligatorias("1112029")==true))
                    {
                        for( ; cont<200; cont++)
                        {
                            if(Tabla[cont][0].equals("1137007"))
                            {
                                if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                                {
                                    for( ; cont<200; cont++)
                                    {
                                        if(Tabla[cont][0].equals("1111019"))
                                        {
                                            if(!Tabla[cont][3].equals("NA") &&
!Tabla[cont][3].equals(""))
                                            {
                                                System.out.println("ya no tiene
seriacion3333333");
                                                break;
                                            }
                                            else
                                            {
                                                ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
                                                indiceRecomendacion++;
                                                break;
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
        else
        {
            ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
            indiceRecomendacion++;
            break;
        }
    }
}
}
else
{

```



```

else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
}
for(cont=0; cont<200; cont++)
{
    if(Tabla[cont][0].equals("1111078"))
    {
if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals("")&&(buscaObligatorias("1112026")==true))
    {
        for(cont=0;cont<200; cont++)
        {
            if(Tabla[cont][0].equals("1111079"))
            {
                if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                {
                    for(cont=0;cont<200; cont++)
                    {
                        if(Tabla[cont][0].equals("1111081"))
                        {
                            if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                            {
                                for(cont=0;cont<200; cont++)
                                {
                                    if(Tabla[cont][0].equals("1111083"))
                                    {
                                        if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                                        {
                                            for(cont=0;cont<200; cont++)
                                            {
                                                if(Tabla[cont][0].equals("1111094"))
                                                {
                                                    if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                                                    {
                                                        System.out.println("ya no tiene seriacion13");
                                                        break;
                                                    }
                                                }
                                            }
                                        }
                                    }
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
}

```



```

    }
  }
}
for(cont=0; cont<200; cont++)
{
  if(Tabla[cont][0].equals("1111078"))
  {

if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals("")&&(buscaObligatorias("1112026")==true))
  {
    for(;cont<200; cont++)
    {
      if(Tabla[cont][0].equals("1111079"))
      {
        if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
        {
          for(;cont<200; cont++)
          {
            if(Tabla[cont][0].equals("1111092"))
            {

if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals("")&&(buscaObligatorias("1111081")))
          {
            for(;cont<200; cont++)
            {
              if(Tabla[cont][0].equals("1111093"))
              {
                if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                {
                  break;
                }
                else
                {
                  ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
                  indiceRecomendacion++;
                  break;
                }
              }
            }
          }
        }
      }
    }
  }
  else
  {
    if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
    {
      break;
    }
    else
    {

```

```

        ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
        indiceRecomendacion++;
        break;
    }
}
}
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
}
}
else
{
    ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
    indiceRecomendacion++;
    break;
}
}
}
}
if((buscaObligatorias("1112029")==true)&&buscaObligatorias("1112013")==true)
{
    for(cont=0 ; cont<200; cont++)
    {
        if(Tabla[cont][0].equals("1112005"))
        {
            if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals("")&&(buscaObligatorias("1112029")==true))
            {
                for(; cont<200; cont++)
                {
                    if(Tabla[cont][0].equals("1111085"))
                    {
                        if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
                        {
                            break;
                        }
                    }
                    else
                    {
                        ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
                        indiceRecomendacion++;
                        break;
                    }
                }
            }
        }
    }
}

```

```

        }
    }
}
else
{
    if(!Tabla[cont][3].equals("NA")&&!Tabla[cont][3].equals(""))
    {
        break;
    }
    else
    {
        ueasRecomendadas[indiceRecomendacion]=Tabla[cont][0];
        indiceRecomendacion++;
        break;
    }
}
}
}
}
}
if(buscaObligatorias("1112005")==true)
{
    if(!Tabla[35][3].equals("NA") && !Tabla[35][3].equals(""))
    {
        System.out.println("ya no tiene seriacion14");
    }
    else
    {
        ueasRecomendadas[indiceRecomendacion]=Tabla[35][0];
        indiceRecomendacion++;
    }
}
if((buscaObligatorias("1111083")==true)&&buscaObligatorias("1112030")==true)
{
    if(!Tabla[46][3].equals("NA") && !Tabla[46][3].equals(""))
    {
        if(!Tabla[31][3].equals("NA")&&!Tabla[31][3].equals(""))
        {
            System.out.println("ya no tiene seriacion15");
        }
        else
        {
            ueasRecomendadas[indiceRecomendacion]=Tabla[31][0];
            indiceRecomendacion++;
        }
    }
}
else
{
    if(Tabla[cont][3].equals("NA"))

```

```

    {
        ueasRecomendadas[indiceRecomendacion]=Tabla[46][0];
        indiceRecomendacion++;
    }
    else if(Tabla[cont][3].equals(""))
    {
        ueasRecomendadas[indiceRecomendacion]=Tabla[46][0];
        indiceRecomendacion++;
        ueasRecomendadas[indiceRecomendacion]="1111069";
        indiceRecomendacion++;
    }
}
}
if((buscaObligatorias("1111090")==true)&&buscaObligatorias("1111091")==true)
{
    if(!Tabla[42][3].equals("NA")&&!Tabla[42][3].equals(""))
    {
        if(!Tabla[50][3].equals("NA")&&!Tabla[50][3].equals(""))
        {
            System.out.println("ya no tiene seriacion16");
        }
        else
        {
            ueasRecomendadas[indiceRecomendacion]=Tabla[31][0];
            indiceRecomendacion++;
        }
    }
    else
    {
        ueasRecomendadas[indiceRecomendacion]=Tabla[31][0];
        indiceRecomendacion++;
    }
}
if(creditos>=280)
{
    if(!Tabla[49][3].equals("NA")&&!Tabla[49][3].equals(""))
    {
        if(!Tabla[54][3].equals("NA")&&!Tabla[54][3].equals("")&&(creditos>=360))
        {
            if(!Tabla[55][3].equals("NA")&&!Tabla[55][3].equals(""))
            {
            }
            else
            {
                ueasRecomendadas[indiceRecomendacion]=Tabla[31][0];
                indiceRecomendacion++;
            }
        }
    }
}

```

```

    }
  }
  else
  {
    ueasRecomendadas[indiceRecomendacion]=Tabla[31][0];
    indiceRecomendacion++;
  }
}
else
{
  ueasRecomendadas[indiceRecomendacion]=Tabla[31][0];
  indiceRecomendacion++;
}
}

if(buscaObligatorias("1100039")==true)
{
  if(!Tabla[57][3].equals("NA")&&!Tabla[57][3].equals(""))
  {
  }
  else
  {
    ueasRecomendadas[indiceRecomendacion]=Tabla[57][0];
    indiceRecomendacion++;
  }
}
if(creditos>=150)
{
  for(cont=58; cont<=84; cont++)
  {
  }
}
return ueasRecomendadas;
}

```

```

boolean buscaObligatorias(String UEA)
{
  boolean band=false;
  for(int i=0; i<200; i++)
  {
    if(Tabla[i][0].equals(UEA))
    {
      if(!Tabla[i][3].equals(""))
      {
        if(!Tabla[i][3].equals("NA"))

```

```

        {
            return true;
        }
        else
        {
            return false;
        }
    }
    else
    {
        return false;
    }
}
}
return band;
}
}

```

G. Implementación de la clase SeriacionOptativa.

```

public class SeriacionOptativa
{
    private boolean Autorizacion=false;
    private String Tabla[][]= new String[210][6]; //Guarda todas la UEAS del plan de estudios
    private String ueasRecomOOI[]=new String[100];
    public int indiceRecomendacionOOI=0;
    private String ueasRecomACE[]=new String[100];
    public int indiceRecomendacionACE=0;
    private String ueasRecomIE[]=new String[100];
    public int indiceRecomendacionIE=0;
    private String ueasRecomTM[]=new String[100];
    public int indiceRecomendacionTM=0;
    private String ueasRecomOT[]=new String[100];
    public int indiceRecomendacionOT=0;
    public int contOpt=0;
    int creditos=0;

    SeriacionOptativa(String[][] T, int cred)
    {
        Tabla=T;
        creditos=cred;
        seriacionOpt();
    }

    String[] OOI()
    {
        return ueasRecomOOI;
    }
}

```

```

String[] ACE()
{
    return ueasRecomACE;
}

String[] IE()
{
    return ueasRecomIE;
}

String[] TM()
{
    return ueasRecomTM;
}
String[] OT()

{
    return ueasRecomOT;
}

void seriacionOpt()
{
    //AREA DE OTRAS OPTATIVAS DE INTEGRACION*****
    if((buscaOpt("1111015")&&buscaOpt("1137007"))==true)
    {
        if(buscaOpt("1111014")==true){}
        else
        {ueasRecomOOI[indiceRecomendacionOOI]="1111014";indiceRecomendacionOOI++;}
    }
    if(buscaOpt("1111014")==true)
    {
        if(buscaOpt("1111015")==true){}
        else
        {ueasRecomOOI[indiceRecomendacionOOI]="1111015";indiceRecomendacionOOI++;}
    }
    if((buscaOpt("1137007")&&buscaOpt("1141006")&&buscaOpt("1146038"))==true)
    {
        if(buscaOpt("1111035")==true){}
        else
        {ueasRecomOOI[indiceRecomendacionOOI]="1111035";indiceRecomendacionOOI++;}
    }
    if(creditos>=300)
    {
        if(buscaOpt("1111052")==true){}
        else
        {ueasRecomOOI[indiceRecomendacionOOI]="1111052";indiceRecomendacionOOI++;}
        if(buscaOpt("1111057")==true){}
        else

```

```

{ueasRecomOOI[indiceRecomendacionOOI]="1111057";indiceRecomendacionOOI++;}
if(buscaOpt("1111066")==true){}
else
{ ueasRecomOOI[indiceRecomendacionOOI]="1111066";indiceRecomendacionOOI++;}
if(buscaOpt("1111067")==true){}
else
{ueasRecomOOI[indiceRecomendacionOOI]="1111067";indiceRecomendacionOOI++;}
if(buscaOpt("1113071")==true){}
else
{ ueasRecomOOI[indiceRecomendacionOOI]="1113071";indiceRecomendacionOOI++;}
if(buscaOpt("1132095")==true){}
else
{ueasRecomOOI[indiceRecomendacionOOI]="1132095";indiceRecomendacionOOI++;}
if(buscaOpt("1131068")==true){}
else
{ ueasRecomOOI[indiceRecomendacionOOI]="1131068";indiceRecomendacionOOI++;}
}
if(buscaOpt("1111090")==true)
{
if(buscaOpt("1111053")==true){}
else
{ueasRecomOOI[indiceRecomendacionOOI]="1111053";indiceRecomendacionOOI++;}
}
if(buscaOpt("1111055")==true)
{
if(buscaOpt("1111059")==true){}
else
{ueasRecomOOI[indiceRecomendacionOOI]="1111059";indiceRecomendacionOOI++;}
}
if(buscaOpt("1111059")==true)
{
if(buscaOpt("1111061")==true){}
else
{ueasRecomOOI[indiceRecomendacionOOI]="1111061";indiceRecomendacionOOI++;}
}
if(buscaOpt("1151038")==true)
{
if(buscaOpt("1112017")==true){}
else
{ueasRecomOOI[indiceRecomendacionOOI]="1112017";indiceRecomendacionOOI++;}
}
if((buscaOpt("1113069")&&buscaOpt("1113070"))==true)
{
if(buscaOpt("1113057")==true){}
else
{ueasRecomOOI[indiceRecomendacionOOI]="1113057";indiceRecomendacionOOI++;}
}
if(buscaOpt("1111032")==true)

```

```

{
  if(buscaOpt("1121012")==true){}
  else
  {ueasRecomOOI[indiceRecomendacionOOI]="1121012";indiceRecomendacionOOI++;}
}
if(buscaOpt("1124050")==true)
{
  if(buscaOpt("1124049")==true){}
  else
  {ueasRecomOOI[indiceRecomendacionOOI]="1124049";indiceRecomendacionOOI++;}
}
if(buscaOpt("1124030")==true)
{
  if(buscaOpt("1124050")==true){}
  else
  {ueasRecomOOI[indiceRecomendacionOOI]="1124050";indiceRecomendacionOOI++;}
}
if(buscaOpt("1137005")==true)
{
  if(buscaOpt("1132009")==true){}
  else
  {ueasRecomOOI[indiceRecomendacionOOI]="1132009";indiceRecomendacionOOI++;}
  if(buscaOpt("1132040")==true){}
  else
  {ueasRecomOOI[indiceRecomendacionOOI]="1132040";indiceRecomendacionOOI++;}
}
if(buscaOpt("1132026")==true&&creditos>=300)
{
  if(buscaOpt("1132050")==true){}
  else
  {ueasRecomOOI[indiceRecomendacionOOI]="1132050";indiceRecomendacionOOI++;}
}
if(buscaOpt("1137006")==true&&creditos>=300)
{
  if(buscaOpt("1132067")==true){}
  else
  {ueasRecomOOI[indiceRecomendacionOOI]="1132067";indiceRecomendacionOOI++;}
}
if(buscaOpt("1111013")==true)
{
  if(buscaOpt("1133003")==true){}
  else
  {ueasRecomOOI[indiceRecomendacionOOI]="1133003";indiceRecomendacionOOI++;}
}
if(buscaOpt("1133060")==true)
{
  if(buscaOpt("1133009")==true){}
  else

```

```

        {ueasRecomOOI[indiceRecomendacionOOI]="1133009";indiceRecomendacionOOI++;}
    }
    if(buscaOpt("1133014")==true)
    {
        if(buscaOpt("1133015")==true){}
        else
        {ueasRecomOOI[indiceRecomendacionOOI]="1133015";indiceRecomendacionOOI++;}
    }
    if(buscaOpt("1111045")==true)
    {
        if(buscaOpt("1133016")==true){}
        else
        {ueasRecomOOI[indiceRecomendacionOOI]="1133016";indiceRecomendacionOOI++;}
    }
    if((buscaOpt("1133060")==true)&&(buscaOpt("1133016")==true))
    {
        if(buscaOpt("1133024")==true){}
        else
        {ueasRecomOOI[indiceRecomendacionOOI]="1133024";indiceRecomendacionOOI++;}
    }
    if(buscaOpt("1133060")==true)
    {
        if(buscaOpt("1132032")==true){}
        else
        {ueasRecomOOI[indiceRecomendacionOOI]="1132032";indiceRecomendacionOOI++;}
    }
    if(buscaOpt("1153001")==true)
    {
        if(buscaOpt("1133049")==true){}
        else
        {ueasRecomOOI[indiceRecomendacionOOI]="1133049";indiceRecomendacionOOI++;}
    }
    if(buscaOpt("1133049")==true)
    {
        if(buscaOpt("1133055")==true){}
        else
        {ueasRecomOOI[indiceRecomendacionOOI]="1133055";indiceRecomendacionOOI++;}
    }
    if(buscaOpt("1133048")==true)
    {
        if(buscaOpt("1133060")==true){}
        else
        {ueasRecomOOI[indiceRecomendacionOOI]="1133060";indiceRecomendacionOOI++;}
    }
    if((buscaOpt("1151039")==true)&&(buscaOpt("1112030")==true))
    {
        if(buscaOpt("1151009")==true){}
        else

```

```

    {ueasRecomOOI[indiceRecomendacionOOI]="1151009";indiceRecomendacionOOI++;}
}
if(buscaOpt("1151038")==true)
{
    if(buscaOpt("1151042")==true){}
    else
    {ueasRecomOOI[indiceRecomendacionOOI]="1151042";indiceRecomendacionOOI++;}
}
if(buscaOpt("1151039")==true)
{
    if(buscaOpt("1152001")==true){}
    else
    {ueasRecomOOI[indiceRecomendacionOOI]="1152001";indiceRecomendacionOOI++;}
}
if((buscaOpt("1153001")==true)&&(buscaOpt("1151039")==true))
{
    if(buscaOpt("1154016")==true){}
    else
    {ueasRecomOOI[indiceRecomendacionOOI]="1154016";indiceRecomendacionOOI++;}
}
if(buscaOpt("1153001")==true&&creditos>=350)
{
    if(buscaOpt("1154042")==true){}
    else
    {ueasRecomOOI[indiceRecomendacionOOI]="1154042";indiceRecomendacionOOI++;}
}
if(creditos>=350)
{
    if(buscaOpt("1154045")==true){}
    else
    {ueasRecomOOI[indiceRecomendacionOOI]="1154045";indiceRecomendacionOOI++;}
}
//*****
//AREA CONCENTRACION ENERGIA
if(buscaOpt("1151039")&&creditos>=250)
{
    if(buscaOpt("1131065")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1131065";indiceRecomendacionACE++;}
}
if(buscaOpt("1112030")&&buscaOpt("1137006"))
{
    if(buscaOpt("1132026")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1132026";indiceRecomendacionACE++;}
}
if(buscaOpt("1137005"))
{

```

```

    if(buscaOpt("1132029")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1132029";indiceRecomendacionACE++;}
}
if(buscaOpt("1132029"))
{
    if(buscaOpt("1132030")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1132030";indiceRecomendacionACE++;}
}
if(buscaOpt("1137015"))
{
    if(buscaOpt("1132041")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1132041";indiceRecomendacionACE++;}
}
if(buscaOpt("1132026"))
{
    if(buscaOpt("1132042")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1132042";indiceRecomendacionACE++;}
}
if(buscaOpt("1132050"))
{
    if(buscaOpt("1132046")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1132046";indiceRecomendacionACE++;}
}
if(buscaOpt("1137005")&&buscaOpt("1137006"))
{
    if(buscaOpt("1132048")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1132048";indiceRecomendacionACE++;}
}
if(buscaOpt("1137005")&&buscaOpt("1137006")&&creditos>=300)
{
    if(buscaOpt("1132049")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1132049";indiceRecomendacionACE++;}
}
if(buscaOpt("1137006")&&creditos>=300)
{
    if(buscaOpt("1132052")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1132052";indiceRecomendacionACE++;}
}
if(buscaOpt("1132064"))
{

```

```

    if(buscaOpt("1132065")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1132065";indiceRecomendacionACE++;}
}
if(buscaOpt("1137005")==true&&buscaOpt("1132026")==true)
{
    if(buscaOpt("1132068")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1132068";indiceRecomendacionACE++;}
    if(buscaOpt("1132091")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1132091";indiceRecomendacionACE++;}
}
if(creditos>=300)
{
    if(buscaOpt("1132092")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1132092";indiceRecomendacionACE++;}
    if(buscaOpt("1132094")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1132094";indiceRecomendacionACE++;}
}
if(buscaOpt("1132092"))
{
    if(buscaOpt("1132093")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1132093";indiceRecomendacionACE++;}
}
if(buscaOpt("1112013")==true&&creditos>=150)
{
    if(buscaOpt("1133061")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1133061";indiceRecomendacionACE++;}
}
if(buscaOpt("1137005"))
{
    if(buscaOpt("1143002")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1134002";indiceRecomendacionACE++;}
}
if(buscaOpt("1134002"))
{
    if(buscaOpt("1134002")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1134003";indiceRecomendacionACE++;}
}
if(buscaOpt("1134002"))
{

```

```

    if(buscaOpt("1134004")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1134004";indiceRecomendacionACE++;}
}
if(buscaOpt("1132049"))
{
    if(buscaOpt("1135051")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1135051";indiceRecomendacionACE++;}
}
if(buscaOpt("1137005")==true &&creditos>=300)
{
    if(buscaOpt("1137015")==true){}
    else
    {ueasRecomACE[indiceRecomendacionACE]="1137015";indiceRecomendacionACE++;}
}
//*****
//AREA CONCENTRACION INSTRUMENTACION Y EQUIPO
if(buscaOpt("1123040"))
{
    if(buscaOpt("1111054")==true){}
    else
    {ueasRecomIE[indiceRecomendacionIE]="1111054";indiceRecomendacionIE++;}
}
if(buscaOpt("1123016"))
{
    if(buscaOpt("1111058")==true){}
    else
    {ueasRecomIE[indiceRecomendacionIE]="1111058";indiceRecomendacionIE++;}
    if(buscaOpt("1111060")==true){}
    else
    {ueasRecomIE[indiceRecomendacionIE]="1111060";indiceRecomendacionIE++;}
}
if(buscaOpt("1151038"))
{
    if(buscaOpt("1121037")==true){}
    else
    {ueasRecomIE[indiceRecomendacionIE]="1121037";indiceRecomendacionIE++;}
}
if(buscaOpt("1121037"))
{
    if(buscaOpt("1121040")==true){}
    else
    {ueasRecomIE[indiceRecomendacionIE]="1121040";indiceRecomendacionIE++;}
}
if(buscaOpt("1123040"))
{
    if(buscaOpt("1123016")==true){}

```

```

else
{ueasRecomIE[indiceRecomendacionIE]="1123016";indiceRecomendacionIE++;}
}
if(buscaOpt("1121037"))
{
if(buscaOpt("1123021")==true){}
else
{ueasRecomIE[indiceRecomendacionIE]="1123021";indiceRecomendacionIE++;}
}
if(buscaOpt("1111083")==true&&creditos>=375)
{
if(buscaOpt("1123026")==true){}
else
{ueasRecomIE[indiceRecomendacionIE]="1123026";indiceRecomendacionIE++;}
}
if(buscaOpt("1124001")==true&&buscaOpt("1124005")==true)
{
if(buscaOpt("1123040")==true){}
else
{ueasRecomIE[indiceRecomendacionIE]="1123040";indiceRecomendacionIE++;}
}
if(buscaOpt("1123040")==true&&buscaOpt("1123045")==true)
{
if(buscaOpt("1123041")==true){}
else
{ueasRecomIE[indiceRecomendacionIE]="1123041";indiceRecomendacionIE++;}
}
if(buscaOpt("1123043")==true&&buscaOpt("1123021")==true)
{
if(buscaOpt("1123042")==true){}
else
{ueasRecomIE[indiceRecomendacionIE]="1123042";indiceRecomendacionIE++;}
}
if(buscaOpt("11230421")==true&&buscaOpt("1123041")==true&&buscaOpt("1123046")==true)
{
if(buscaOpt("1123043")==true){}
else
{ueasRecomIE[indiceRecomendacionIE]="1123043";indiceRecomendacionIE++;}
}
if(buscaOpt("1124003")==true&&buscaOpt("1123041"))
{
if(buscaOpt("1123044")==true){}
else
{ueasRecomIE[indiceRecomendacionIE]="1123044";indiceRecomendacionIE++;}
}
if(buscaOpt("1123040"))
{
if(buscaOpt("1123045")==true){}

```

```

else
{ueasRecomIE[indiceRecomendacionIE]="1123045";indiceRecomendacionIE++;}
}
if(buscaOpt("1123041"))
{
if(buscaOpt("1123046")==true){}
else
{ueasRecomIE[indiceRecomendacionIE]="1123046";indiceRecomendacionIE++;}
}
if(buscaOpt("1123042"))
{
if(buscaOpt("1123047")==true){}
else
{ueasRecomIE[indiceRecomendacionIE]="1123047";indiceRecomendacionIE++;}
}
if(buscaOpt("1123044"))
{
if(buscaOpt("1123034")==true){}
else
{ueasRecomIE[indiceRecomendacionIE]="1123034";indiceRecomendacionIE++;}
}
if(buscaOpt("1124001")==true&&buscaOpt("1112015")==true)
{
if(buscaOpt("1124003")==true){}
else
{ueasRecomIE[indiceRecomendacionIE]="1124003";indiceRecomendacionIE++;}
}
if(buscaOpt("1111090"))
{
if(buscaOpt("1131010")==true){}
else
{ueasRecomIE[indiceRecomendacionIE]="1131010";indiceRecomendacionIE++;}
}
if(buscaOpt("1124001")==true&&buscaOpt("1124005")==true)
{
if(buscaOpt("1131070")==true){}
else
{ueasRecomIE[indiceRecomendacionIE]="1131070";indiceRecomendacionIE++;}
}
if(buscaOpt("1131070"))
{
if(buscaOpt("1131071")==true){}
else
{ueasRecomIE[indiceRecomendacionIE]="1131071";indiceRecomendacionIE++;}
}
//*****
//AREA CONCENTRACION TECNOLOGIA DE MATERIALES
if(buscaOpt("1111048"))

```

```

{
  if(buscaOpt("1111032")==true){}
  else
  {ueasRecomTM[indiceRecomendacionTM]="1111032";indiceRecomendacionTM++;}
}
if(buscaOpt("1111043"))
{
  if(buscaOpt("1111034")==true){}
  else
  {ueasRecomTM[indiceRecomendacionTM]="1111034";indiceRecomendacionTM++;}
}
if(buscaOpt("1111081")==true&&buscaOpt("1112005"))
{
  if(buscaOpt("1111045")==true){}
  else
  {ueasRecomTM[indiceRecomendacionTM]="1111045";indiceRecomendacionTM++;}
}
if(buscaOpt("1145054"))
{
  if(buscaOpt("1133014")==true){}
  else
  {ueasRecomTM[indiceRecomendacionTM]="1133014";indiceRecomendacionTM++;}
  if(buscaOpt("1145001")==true){}
  else
  {ueasRecomTM[indiceRecomendacionTM]="1145001";indiceRecomendacionTM++;}
  if(buscaOpt("1145052")==true){}
  else
  {ueasRecomTM[indiceRecomendacionTM]="1145052";indiceRecomendacionTM++;}
  if(buscaOpt("1145071")==true){}
  else
  {ueasRecomTM[indiceRecomendacionTM]="1145071";indiceRecomendacionTM++;}
  if(buscaOpt("1146038")==true){}
  else
  {ueasRecomTM[indiceRecomendacionTM]="1146038";indiceRecomendacionTM++;}
}
if(buscaOpt("1146038"))
{
  if(buscaOpt("1141006")==true){}
  else
  {ueasRecomTM[indiceRecomendacionTM]="1141006";indiceRecomendacionTM++;}
}
if(buscaOpt("1111045"))
{
  if(buscaOpt("1142025")==true){}
  else
  {ueasRecomTM[indiceRecomendacionTM]="1142025";indiceRecomendacionTM++;}
}
if(buscaOpt("111032"))

```

```

    {
        if(buscaOpt("1145050")==true){}
        else
        {ueasRecomTM[indiceRecomendacionTM]="1145050";indiceRecomendacionTM++;}
    }

if(buscaOpt("1112028")==true&&buscaOpt("1113086")==true&&buscaOpt("1113087")==true&&buscaOpt("1113046")==true)
    {
        if(buscaOpt("1145054")==true){}
        else
        {ueasRecomTM[indiceRecomendacionTM]="1145054";indiceRecomendacionTM++;}
    }
if(buscaOpt("1145054"))
    {
        if(buscaOpt("1145055")==true){}
        else
        {ueasRecomTM[indiceRecomendacionTM]="1145055";indiceRecomendacionTM++;}
    }
if(buscaOpt("1112030"))
    {
        if(buscaOpt("1145046")==true){}
        else
        {ueasRecomTM[indiceRecomendacionTM]="1145056";indiceRecomendacionTM++;}
    }
if(buscaOpt("1145056"))
    {
        if(buscaOpt("1145057")==true){}
        else
        {ueasRecomTM[indiceRecomendacionTM]="1145057";indiceRecomendacionTM++;}
    }
if(buscaOpt("1137006"))
    {
        if(buscaOpt("1145058")==true){}
        else
        {ueasRecomTM[indiceRecomendacionTM]="1145058";indiceRecomendacionTM++;}
    }
if(buscaOpt("1145052"))
    {
        if(buscaOpt("1145060")==true){}
        else
        {ueasRecomTM[indiceRecomendacionTM]="1145060";indiceRecomendacionTM++;}
    }
if(buscaOpt("1145060")==true&&buscaOpt("1145001")==true)
    {
        if(buscaOpt("1145061")==true){}
        else
        {ueasRecomTM[indiceRecomendacionTM]="1145061";indiceRecomendacionTM++;}
    }

```

```

}
if(buscaOpt("1145060"))
{
    if(buscaOpt("1145066")==true){}
    else
        {ueasRecomTM[indiceRecomendacionTM]="1145066";indiceRecomendacionTM++;}
}
if(buscaOpt("1145066"))
{
    if(buscaOpt("1145067")==true){}
    else
        {ueasRecomTM[indiceRecomendacionTM]="1145067";indiceRecomendacionTM++;}
}
if(buscaOpt("1145071"))
{
    if(buscaOpt("1145072")==true){}
    else
        {ueasRecomTM[indiceRecomendacionTM]="1145072";indiceRecomendacionTM++;}
}
if(buscaOpt("1145081"))
{
    if(buscaOpt("1145080")==true){}
    else
        {ueasRecomTM[indiceRecomendacionTM]="1145080";indiceRecomendacionTM++;}
}
if(buscaOpt("1145066")==true&&buscaOpt("1145067"))
{
    if(buscaOpt("1145081")==true){}
    else
        {ueasRecomTM[indiceRecomendacionTM]="1145081";indiceRecomendacionTM++;}
}
if(buscaOpt("1145052"))
{
    if(buscaOpt("1145091")==true){}
    else
        {ueasRecomTM[indiceRecomendacionTM]="1145091";indiceRecomendacionTM++;}
}
if(buscaOpt("1145066")==true&&buscaOpt("1145067")==true&&Autorizacion)
{
    if(buscaOpt("1145098")==true){}
    else
        {ueasRecomTM[indiceRecomendacionTM]="1145098";indiceRecomendacionTM++;}
}
//*****
//AREA CONCENTRACION OPTATIVAS TUTORIALES
if(buscaOpt("1132092")==true||buscaOpt("1132094")==true)
{
    if(buscaOpt("1132099")==true){}
}

```

```

        else
        {ueasRecomOT[indiceRecomendacionOT]="1132099";indiceRecomendacionOT++;}
    }
    if(creditos>=400)
    {
        if(buscaOpt("1145051")==true){}
        else
        {ueasRecomOT[indiceRecomendacionOT]="1145051";indiceRecomendacionOT++;}
    }
}

boolean buscaOpt(String UEA)
{
    boolean band=false;
    for(int i=0; i<200; i++)
    {
        if(Tabla[i][0].equals(UEA))
        {
            if(!Tabla[i][3].equals(""))
            {
                if(!Tabla[i][3].equals("NA"))
                {
                    contOpt+=Integer.parseInt(Tabla[i][2]);
                    return true;
                }
            }
            else
            {
                return false;
            }
        }
        else
        {
            return false;
        }
    }
}
return band;
}
}

```