

Universidad Autónoma Metropolitana Unidad Azcapotzalco

División de Ciencias Básicas e Ingeniería

Licenciatura en Ingeniería en Computación

**Reporte de proyecto terminal**

**Sistema de recomendación de UEA para Ingeniería Industrial**

Francisco Jesús Cruz Guzmán

204201406

Trimestre 2013 Otoño

Miguel Ángel López Ontiveros

Profesor Titular

Departamento de Sistemas

Sergio Luis Pérez Pérez

Profesor Titular

UAM Cuajimalpa

## Tabla de contenido

Resumen.....	3
1. Introducción.....	4
1.1 Objetivo general.....	4
1.2 Objetivos específicos.....	4
2. Diseño de la aplicación.....	5
2.1 Módulo de ingreso.....	5
2.1.1 Interfaz gráfica del Módulo de ingreso.....	7
2.2 Módulo de solicitud.....	8
2.3 Módulo de análisis.....	9
2.3.1 Implementación del plan de estudios de Ingeniería Industrial en una estructura de datos.....	9
2.3.2 Análisis de la Información académica.....	13
2.3.3 Análisis de Kardex.....	14
2.4 Módulo de recomendación.....	14
2.4.1 Generación de prioridades.....	14
2.4.2 Cálculo del Mínimo Recomendado.....	15
2.4.3 Cálculo del Máximo Recomendado.....	16
2.4.4 Ordenar las UEA en el Grafo.....	17
2.5 Módulo de presentación: Interfaz gráfica.....	18
3. Conclusiones.....	20
ANEXO 1. Manual de instalación.....	22
ANEXO 2. Manual de usuario.....	24
ANEXO 3. Boligrama de plan de estudios de Ingeniería Industrial.....	29
ANEXO 4. Artículo 22, Fracción V del Reglamento de Estudios Superiores....	31
ANEXO 5. Código fuente.....	33
Bibliografía.....	61

## Resumen

El plan de estudios de la carrera de Ingeniería Industrial, ha sido elaborado de manera progresiva con el objetivo de que el alumno avance en la acumulación de conocimientos conforme va inscribiendo sus UEA. Sin embargo, la flexibilidad del mismo, permite que el alumno inscriba materias de distintos trimestres sin reflexionar en la importancia del conocimiento previo que debe tener para su siguiente trimestre. Esto puede ocasionar que el alumno deje inconclusa la materia, incrementando su índice de no aprobación, y en consecuencia, repercutir en la eficiencia terminal. Así mismo, existen materias que el alumno no siempre considera y que pueden ser esenciales para inscribir los cursos del siguiente bloque.

La finalidad del *Sistema de recomendación* es ofrecer una herramienta que oriente al alumno de Ingeniería Industrial, a escoger sus UEA, auxiliándose de la tecnología y de fácil acceso. Tomando en cuenta las actuales necesidades de los jóvenes universitarios es que se ha pensado en este sistema. El documento que se presenta a continuación divide su corpus en cinco módulos y un apéndice de cinco anexos: un manual para la instalación de la aplicación en dispositivos móviles con sistema operativo Android; un manual de usuario, el cual explica el funcionamiento y utilización del *Sistema de recomendación*; el boligrama del plan de estudios de Ingeniería Industrial; el Artículo 22, Fracción V del Reglamento de Estudios Superiores de la UAM, utilizada para la implementación de un algoritmo; y finalmente, el código fuente de la aplicación.

## 1. Introducción

La Universidad Autónoma Metropolitana (UAM), Unidad Azcapotzalco, hace uso del SAE<sup>1</sup> para que el alumno pueda seleccionar las UEA<sup>2</sup> que debe inscribir para su próximo trimestre. Sin embargo, actualmente el sistema muestra todas las UEA que puede inscribir sin hacer distinción entre las que tendrían mayor o menor prioridad, esto ocasiona, que el alumno no haga una buena elección provocando un retraso en su formación académica.

El reto, fue crear un *Sistema de recomendación* para dispositivos móviles con sistema operativo Android, que mostrará al alumno de la carrera de Ingeniería Industrial las UEA sugeridas a cursar en su próximo trimestre. En esta aplicación, el alumno ingresa su matrícula y contraseña; enseguida, se conecta al SAE e inicia sesión para obtener su **Kardex**<sup>3</sup> e **Información académica**<sup>4</sup>. El *Sistema de recomendación* haciendo uso de los algoritmos implementados mostrará las UEA sugeridas. De este modo, el alumno tendrá una guía que le permitirá reflexionar mejor al momento de acomodar su horario escolar.

### 1.1 Objetivo general

Diseñar e implementar un *Sistema de recomendación* inteligente que muestre al alumno de la carrera Ingeniería Industrial las UEA sugeridas para la inscripción en su próximo trimestre.

### 1.2 Objetivos específicos

Diseñar e implementar:

- Módulo de ingreso y conexión al SAE.
- Módulo de solicitud de información requerida del SAE, la cual es recibida en HTML.

---

<sup>1</sup> SAE: Subsistema de Administración Escolar del "Módulo de Información Escolar de Alumnos de Licenciatura" de la Universidad Autónoma Metropolitana Azcapotzalco.

<sup>2</sup> UEA: Unidad de Enseñanza-Aprendizaje.

<sup>3</sup> Kardex: es la sección del módulo de información que muestra las UEA aprobadas y no aprobadas, el trimestre en que fueron cursadas y el tipo de evaluación en la fue aprobada.

<sup>4</sup> Información académica: es la sección del módulo de información que muestra la licenciatura a la que pertenece el alumno, las UEA optativas cubiertas y el resumen trimestral.

- Módulo de análisis sintáctico del código HTML obtenido para extraer la información necesaria para la recomendación.
- Módulo de recomendación de las UEA a sugerir.
- Módulo que muestre las UEA recomendadas.
- La aplicación para un dispositivo móvil con sistema operativo Android.

## 2. Diseño de la aplicación

En esta sección se explicará la implementación del *Sistema de recomendación*, problemas encontrados y soluciones al mismo. Cabe mencionar, que este apartado está dividido en 5 módulos, los cuales se detallan de forma breve:

- Módulo de ingreso: la función de este módulo es explicar cómo el *Sistema de recomendación* se conecta al SAE.
- Módulo de solicitud: una vez realizada la conexión con el SAE, se expone la manera en que la aplicación hace la petición e importación del **Kardex** e **Información académica**, con el fin de conocer el estatus del alumno en la UAM Azcapotzalco, y corroborar que la matrícula del alumno corresponda a Ingeniería Industrial.
- Módulo de análisis: en este apartado, se explica la realización del análisis sintáctico de los archivos recibidos y la implementación de dos clases llamadas Grafo y Nodo, para manipular la información, y determinar el máximo de créditos viables a inscribir<sup>5</sup>.
- Módulo de recomendación: aquí se explica cómo se genera la sugerencia de UEA a inscribir.
- Módulo de presentación: en esta última sección, se explican y muestran las interfaces gráficas implementadas en el *Sistema de recomendación*.

### 2.1 Módulo de ingreso

Cuando el alumno de la UAM Azcapotzalco solicita ver su situación académica, ingresa al Módulo de Información Escolar (SAE), la página donde hace la consulta es:

---

<sup>5</sup> El mínimo viable siempre es cero, que se conoce como inscripción en blanco; mientras que el Máximo Viable será 40 ó 63.

<https://ayamictlan.uam.mx:8443/sae/azc/aewbf001.omuestraframes?mod=1>

En la página, el alumno ingresa su matrícula y una contraseña (de diez a quince caracteres, con ciertas especificaciones), el siguiente paso es dar clic en “Entrar” y el sistema se encargará de validar los datos ingresados.

El primer obstáculo al que nos enfrentamos fue establecer la conexión desde una aplicación en un dispositivo móvil, para resolverlo, se plantearon las siguientes preguntas como guía:

1. ¿Cuál es la URL<sup>6</sup> directa, donde sólo se pide y valida la matrícula y contraseña?
2. ¿Qué información, a parte de la matrícula y contraseña, se manda en la petición?

Para resolver ambas preguntas, se utilizó un programa llamado *Firebug*<sup>7</sup>, con el que se obtuvo el siguiente resultado:

1. URL: <https://ayamictlan.uam.mx:8443/sae/azc/AEWBU004>
2. Se determinó que aparte de la matrícula y contraseña, se envían de manera oculta, nueve datos.

Una vez resueltas las preguntas, se prosiguió con la implementación del módulo, haciendo uso de las bibliotecas que proporciona Android y de la información previamente obtenida.

Las funciones del módulo de ingreso son las siguientes:

- Validación de datos: al ingresar los datos, el *Sistema de recomendación* se encargará de leerlos y validar que los campos no estén vacíos. En caso de que alguno esté vacío, la aplicación mostrará un mensaje de error, si no, los datos se enviarán al SAE.

Adicionalmente, se encontró con un caso en el que un alumno de Ingeniería Industrial, había creado una contraseña de 16 caracteres,

---

<sup>6</sup> URL: es la dirección global de documentos y de otros recursos en la web.

<sup>7</sup> Extensión del navegador Firefox. Una de las funciones de este programa es el permitir ver todas peticiones que se realizan a una página web y la(s) respuesta(s) de la misma.

cuando el máximo 15, si el alumno iniciaba sesión en el SAE, la contraseña la validaba como correcta pero, si se realizaba desde la aplicación, la validaba como incorrecta, haciendo pruebas, se determinó que el SAE solo válida hasta 15 caracteres en la contraseña. Para resolver el problema, se implementó un algoritmo que recorta la contraseña al número de caracteres permitidos.

- Respuesta del SAE: una vez enviada la información, el SAE se encargará de validar los datos y devolverá como respuesta una cookie<sup>8</sup>.
- Verificación de la cookie: el *Sistema de recomendación* validará la longitud de la cookie. Si la longitud es diferente de tres<sup>9</sup>, la aplicación mostrará un mensaje de error indicando que la matrícula o contraseña son incorrectos, si no, continuará con el proceso para la recomendación.

### 2.1.1 Interfaz gráfica del Módulo de ingreso

Al iniciar sesión en el *Sistema de recomendación* (instalada previamente en el dispositivo móvil), el alumno de Ingeniería Industrial podrá ver dos campos donde se le solicitará ingresar su matrícula y contraseña (misma que ingresa cuando inicia sesión en el SAE) y oprimirá el botón “Enviar”, como se muestra a continuación:



Imagen 2.1.1 Módulo de ingreso

<sup>8</sup> La cookie contiene información de un sitio web específico que se relaciona con el usuario. Esta será usada para obtener el **Kardex** y la **Información académica** del alumno.

<sup>9</sup> Como el *Sistema de recomendación* no tiene acceso directo a la base datos del SAE, se realizaron pruebas y se determinó que cuando la longitud de la cookie es tres, la matrícula y contraseña son válidos.

## 2.2 Módulo de solicitud

Cuando el alumno inicia sesión en el SAE, el navegador (Chrome, Firefox, Explorer, etc.) es el encargado de mantener activa la sesión haciendo uso de la cookie que se obtiene, esto es, mientras no se termine o finalice la sesión, se podrá ingresar a todas las opciones que permita la página web. Por ende, una vez que el *Sistema de recomendación* establece la conexión con el SAE, utilizará la cookie obtenida, agregándola en cada petición (simulando la función que realiza el navegador), con el fin de mantener la sesión activa durante el proceso de importación del **Kardex** e **Información académica** en formato HTML<sup>10</sup>. Este proceso se realizó de esta manera:

- Solicitud del **Kardex**: igual que en el módulo de ingreso, es necesario determinar la URL de la sección Kardex del SAE a la que se le tiene que hacer la petición de la información requerida, para ello, se utilizó nuevamente *Firebug* y con la ayuda de este programa se obtuvo lo siguiente:

<https://ayamictlan.uam.mx:8443/sae/azc/IEWBC020.oConsulta>

A esta petición, utilizando las bibliotecas que proporciona Android, se le agregará la cookie obtenida en el módulo anterior.

- Solicitud de **Información académica**: se realizará el mismo procedimiento pero a la siguiente URL:

<https://ayamictlan.uam.mx:8443/sae/azc/IEWBC007.oConsulta>

Una vez obtenidos los archivos requeridos, se buscará en **Información académica**, que la matrícula del alumno corresponda con la carrera de Ingeniería Industrial, en caso de ser así, continuará con el proceso para la recomendación; en caso contrario, se mostrará un mensaje de error indicando que la aplicación es para alumnos de Ingeniería Industrial.

---

<sup>10</sup> HTML: por sus siglas en inglés, HyperText Markup Language, es un lenguaje para la elaboración de páginas web.

## 2.3 Módulo de análisis

En esta sección se analizarán los archivos obtenidos en el módulo 2.2. Este módulo, 2.3, se divide en los siguientes sub módulos:

- Implementación del plan de estudios de Ingeniería Industrial en una estructura de datos.
- Análisis de la **Información académica**.
- Análisis del **Kardex**.

### 2.3.1 Implementación del plan de estudios de Ingeniería Industrial en una estructura de datos.

Para poder realizar la estructura de datos que se requiere para el análisis sintáctico de la información, es importante mencionar algunas especificaciones del plan de estudios, mismas que se utilizarán para la implementación del mismo.

- Seriación: permite definir las UEA, o en su caso el número de créditos, que deben ser acreditados previamente para poder inscribir cada curso.
- Corregistro: se define a partir de la correlación entre dos, es decir, no se permite inscribir la UEA si no se cursa junto con la UEA indicada.
- NAs permitidas: es la cantidad de calificaciones No Aprobadas (NAs) permitidas para inscribir la UEA en evaluación global, y dependiendo de la UEA, las NAs permitidas serán uno o cuatro.

Una vez definidas las especificaciones, se prosigue con la creación de dos clases:

- Nodo: sirve como plantilla para crear las UEA del plan de estudios de Ingeniería Industrial. A la clase *Nodo*, se le agregó a sugerencia del coordinador de Ingeniería Industrial el trimestre el cual debe cursarse. Considerando de igual forma, la clave y nombre de la UEA, cantidad de créditos que la conforman, requisitos para inscribir la UEA (seriación y/o corregistro), número de NAs permitidas, etc. También se agregó un número de prioridad con valor inicial igual a cero.

- **Grafo:** sirve como plantilla para crear el plan de estudios de la carrera de Ingeniería Industrial (II), como se muestra en la imagen 2.3.1 (para ver el boligrama completo del plan de estudios de II, consultar Anexo 3). Al hacer la instancia de esta clase, se almacenarán y relacionarán los objetos de tipo Nodo.

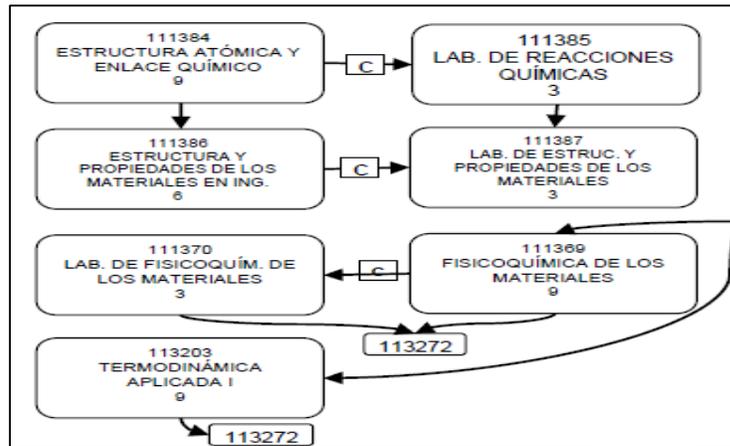


Imagen 2.3.1.1 Grafo del plan de estudios de Ingeniería Industrial

Dicho lo anterior, y utilizando la biblioteca gratuita *Jsoup*<sup>11</sup> (cargada previamente en el *Sistema de recomendación*), ambas clases se van a usar para manipular los datos que se extraerán del **Kardex** e **Información académica**. En lo referente al **Kardex**, los datos que se van a requerir son la “UEA”, que contiene la clave de la UEA; y “Calificación”, que contiene la información de la UEA aprobada o no aprobada, como se muestra en la siguiente imagen:

Registro	UEA	Nombre de la UEA	Horas	Tip	Eval	Calificación	Nº de Años	Créditos
1201008		COMPRESIÓN DE TEXTOS	990	GLC	B	28181348	4	
1111510		INTRODUCCIÓN A LA FÍSICA	990	GLC	UB	28181361	4	
1111205		ALLER DE MATEMÁTICAS	990	GLC	UB	28181362	7	
1111084		ESTRUCTURA ATÓMICA Y ENLACE QUÍMICO	990	GLC	EC	30000000	9	
1113049		LABORATORIO DE REACCIONES QUÍMICAS	990	GLC	EC	30000000	3	
1100037		INTRODUCCIÓN A LA INGENIERÍA	990	GLC	EC	30000000	9	
1113046		SEMIOLOGÍA	180	GLC	UB	210132068	2	
1111213		COMPLEMENTOS DE MATEMÁTICAS	180	GLC	UB	210131479	9	
1111219		MECÁNICA Y DINÁMICA DE PARTICULAS	180	GLC	EC	30000000	9	
1111084		ESTRUCTURA Y PROPIEDADES DE LOS MATERIALES EN INGENIERÍA	180	GLC	EC	30000000	9	
1111087		LABORATORIO DE ESTRUCTURA Y PROPIEDADES DE LOS MATERIALES	180	GLC	EC	30000000	3	
1100038		INTRODUCCIÓN AL DISEÑO DEL PRODUCTO SOSTENTABLE	180	GLC	EC	30000000	9	
1111051		MECÁNICA DEL CUERPO RÍGIDO	180	GLC	EC	30000000	9	
1111201		INTRODUCCIÓN AL CÁLCULO	180	GLC	EC	30000000	9	
1111202		CÁLCULO DIFERENCIAL	180	GLC	EC	30000000	9	
1100016		GÉNERO Y SEGURIDAD	180	GLC	EC	30000000	9	
1100019		LEDER Y GOBIERNO	180	GLC	EC	30000000	9	
1111102		LABORATORIO DE MOVIMIENTO DE UNA PARTICULA	180	GLC	EC	30000000	3	
1113051		ANÁLISIS MECÁNICO ASISTIDO POR COMPUTADORA	180	GLC	EC	30000000	9	

Imagen 2.3.1.2 Sección Kardex del SAE

<sup>11</sup> Es una biblioteca gratuita de Java para trabajar con HTML. Proporciona una API (Application Programming Interface) muy conveniente para la extracción y manipulación de datos.

En cuanto a la **Información académica**, los datos requeridos utilizando la siguiente nomenclatura son:

- **Renuncias**: información que indica si el alumno dio de baja alguna UEA el trimestre anterior.

Trimestre	Renuncias	Créditos	Créditos
09O	0	30	0
10I	0	39	0
10P	0	36	0
10O	0	42	0
11I	2	30	0
11P	1	39	0
11O	1	33	0
12I	0	42	0
12P	0	45	9
12O	0	45	0
13I	0	54	0
13P	0	36	0
13O	1	21	0

Imagen 2.3.1.3 Renuncias

- **trimestresCursados**: contiene el número de trimestres que el alumno ha cursado durante su estancia en la UAM Azcapotzalco.
- **Trimestres**: conjunto, donde cada posición, indica el número de trimestre cursado en la UAM Azcapotzalco.
- **créditosPorTrimestre**: información que indica los créditos aprobados por trimestre.

Trimestre	Renuncias	Créditos
09O	0	30
10I	0	39
10P	0	36
10O	0	42
11I	2	30
11P	1	39
11O	1	33
12I	0	42
12P	0	45
12O	0	45
13I	0	54
13P	0	36
13O	1	21

Imagen 2.3.1.4 Créditos por trimestre

- créditosAnterior: información que indica los créditos aprobados el trimestre anterior.

Trimestre	Renuncias	Créditos
09O	0	30
10I	0	39
10P	0	38
10O	0	42
11I	2	30
11P	1	39
11O	1	33
12I	0	42
12P	0	45
12O	0	45
13I	0	54
13P	0	58
13O	1	21

Imagen 2.3.1.5 Créditos

- cantidadMB, cantidadB, cantidadS y cantidadNA: contiene el número de las mismas obtenidas el trimestre anterior.

Trimestre	Renuncias	Créditos	Créditos	Créditos	Créditos	MB	B	S	NA
09O	0	30	0	16	0	2	1	0	0
10I	0	39	0	16	0	2	0	0	0
10P	0	38	0	0	0	0	0	0	0
10O	0	42	0	0	0	0	0	0	0
11I	2	30	0	9	0	1	0	0	0
11P	1	39	0	16	0	2	0	0	0
11O	1	33	0	0	0	1	0	0	0
12I	0	42	0	21	0	2	2	0	0
12P	0	45	0	36	0	3	2	0	0
12O	0	45	0	0	0	1	0	0	0
13I	0	54	0	24	0	2	1	0	0
13P	0	58	0	18	0	2	2	0	0
13O	1	21	0	21	0	1	2	0	0

Imagen 2.3.1.6 Cantidad de MB, B, S y NA

- créditosAcumulados: información que indica el total de créditos aprobados.

Trimestre	Renuncias	Créditos	Créditos	Créditos	Créditos	MB	B	S	NA	Trimestre	Créditos Acumulados
09O	0	30	0	16	0	2	1	0	0	1	30
10I	0	39	0	16	0	2	0	0	0	2	75
10P	0	38	0	0	0	0	0	0	0	3	123
10O	0	42	0	0	0	0	0	0	0	4	154
11I	2	30	0	9	0	1	0	0	0	5	164
11P	1	39	0	16	0	2	0	0	0	6	224
11O	1	33	0	0	0	1	0	0	0	7	241
12I	0	42	0	21	0	2	2	0	0	9	302
12P	0	45	0	36	0	3	2	0	0	9	348
12O	0	45	0	0	0	1	0	0	0	10	403
13I	0	54	0	24	0	2	1	0	0	11	457
13P	0	58	0	18	0	2	2	0	0	12	495
13O	1	21	0	21	0	2	2	0	0	13	814

Imagen 2.3.1.7 Créditos acumulados

- créditosUEAOptativas: información que contiene el total de créditos aprobados de las áreas como:
  - Optativas del área de Gestión Industrial.
  - Optativas del área de Producción y Manufactura.
  - Tutoriales.
  - De movilidad.
  - Otras optativas de integración.

Nivel	Área	Subárea	Tipo	Descripción	Creditos a cubrir	Evaluación a cubrir	Calificación
4	8	2	08	OPT. LINEA ESTUDIOS CULTURALES	0	03	12
4	8	2	11	OPT. LINEA FORMACIÓN TECNOLÓGICA	0	03	12
4	8	2	23	OPT. LINEA INDUCCIÓN AL MERCADO LABORAL	0	03	12
4	8	2	32	OPT. LINEA LENGUAJES FORMALES	0	03	12
4	8	2	198	OPT. AREA DE CONCENTRACION - PRODUCCION Y MANUFACTURA	0	100	00
4	8	2	200	OPT. AREA DE CONCENTRACION - GESTION INDUSTRIAL	0	100	00
4	8	2	130	OTRAS OPTATIVAS DE INTEGRACION	0	00	00

Imagen 2.3.1.8 UEA optativas acreditadas

### 2.3.2 Análisis de la Información académica

Una vez obtenidos los datos requeridos, para esta sección se necesitarán los siguientes: *cantidadMB*, *cantidadB*, *cantidadS*, *cantidadNA*, *Renuncias* y *créditosAnterior*, con el propósito para determinar el **Máximo Viable** (máximo de créditos viables a inscribir), tomando como referencia la información contenida en el Artículo 22, Fracción V del Reglamento de Estudios Superiores de la Universidad Autónoma Metropolitana (para mayor información ver Anexo 4) el siguiente pseudocódigo ejemplifica el algoritmo implementado:

- 1 Si  $(cantidadMB + cantidadB + cantidadS) = 0$  ó  $cantidadNA \geq 1$ , entonces
- 2 *Máximo Viable* ← 40
- 3 Otro Si  $(Renuncias = 0$  y  $créditosAnterior \geq 20$  y  $\frac{cantidadMB+10+cantidadB+8+cantidadS+6}{cantidadMB+cantidadB+cantidadS} \geq 8$ , entonces
- 4 *Máximo Viable* ← 63
- 5 Si no, entonces
- 6 *Máximo Viable* ← 40

El Máximo Viable será utilizado en el módulo de recomendación, con el propósito de que la suma de los créditos de las UEA a sugerir no exceda dicha cantidad.

### **2.3.3 Análisis de Kardex**

En esta sección, los datos requeridos son la *UEA* y la *Calificación*, todo ello con la finalidad de modificar el Grafo. Para ello se creó una función que recibe como parámetros los datos previamente mencionados.

La finalidad, es verificar cada UEA recibida y buscarla en el Grafo; una vez encontrada, la función analizará la calificación. En caso de que la UEA haya sido aprobada, se buscará si tiene UEA dependientes (Seriación), de ser afirmativo, se verifica si se pueden habilitar. En caso de no aprobada, se verificará que el número de NAs acumuladas no exceda el permitido (NAs permitidas), en caso de ser así, la UEA automáticamente se deshabilita para mostrarse en la recomendación.

## **2.4 Módulo de recomendación**

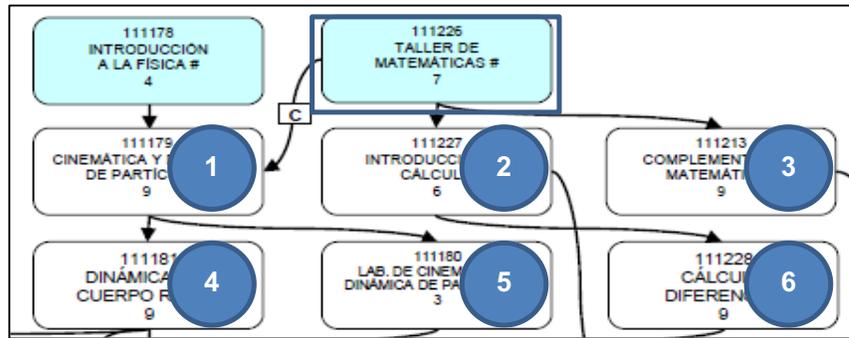
En este módulo es en donde el *Sistema de recomendación*, haciendo uso de algoritmos implementados, va a generar las UEA que se van a sugerir. La sección se divide en los siguientes sub módulos:

- Generación de prioridades.
- Cálculo del Mínimo Recomendado.
- Cálculo del Máximo Recomendado.
- Ordenar las UEA en el Grafo.

### **2.4.1 Generación de prioridades**

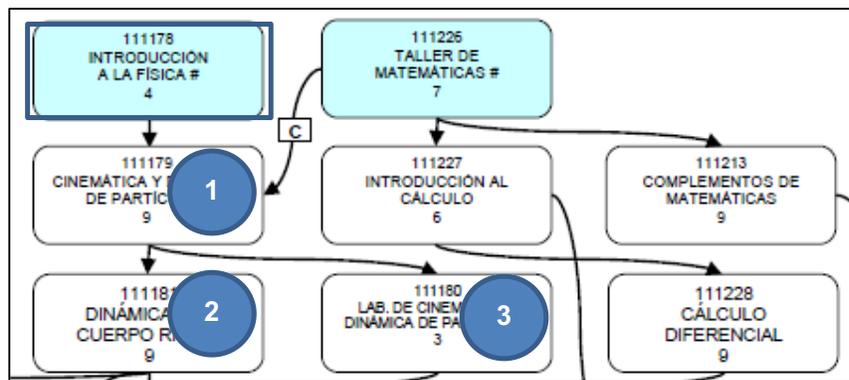
Es de suma importancia este sub módulo por la implementación de un algoritmo que permite generar un número de *Prioridad* para cada UEA (Nodo) contenida en el Grafo. Esto es, porque la sugerencia del coordinador de Ingeniería Industrial, solo indica en qué trimestre se debe cursar la UEA, dejando de lado la diferencia entre cuál tiene mayor o menor prioridad.

Para solucionar este problema, se implementó un algoritmo cuya función es contar para cada UEA todos sus dependientes, las siguientes imágenes ejemplifican el algoritmo:



2.4.1.1 Generación de Prioridad para la UEA Taller de Matemáticas

Se puede ver que la UEA, Taller de Matemáticas, tiene un número de *Prioridad* igual a seis.



2.4.1.2 Generación de Prioridad para la UEA Introducción a la Física

Para la UEA, Introducción a la Física, el número de *Prioridad* es igual a tres.

## 2.4.2 Cálculo del Mínimo Recomendado

Una vez obtenido el número de prioridad para cada UEA, el siguiente paso es determinar el **Mínimo Recomendado** (número mínimo de créditos a sugerir). Para ello, se requieren de los *trimestresCursados* y *créditosAcumulados* obtenidos en el Módulo de análisis.

Con base en el número de *trimestresCursados*, se generará un mínimo de créditos para que el alumno pueda terminar la carrera, ya sea en 12, 18 ó 30 trimestres. Una vez que se define en qué caso se encuentra, se utiliza la siguiente formula:

$$\text{Mínimo Recomendado} \leftarrow \frac{496 - \text{créditosAcumulados}}{n - \text{trimestresCursados}}, \text{ donde } n = 12, 18 \text{ ó } 30$$

### 2.4.3 Cálculo del Máximo Recomendado

Para calcular el **Máximo Recomendado** (máximo número de créditos a sugerir), se propuso utilizar la fórmula de promedios móviles<sup>12</sup>, tomando como referencia, los créditos aprobados de los tres últimos trimestres. Haciendo pruebas con la fórmula, se llegó a obtener un **Máximo Recomendado** igual a cero, esto ocurrió cuando el alumno no tenía actividad escolar.

Considerando esto, se optó por cambiar la fórmula por la del pronóstico<sup>13</sup>. Con esta fórmula, la ventaja que se obtiene es que utiliza como referencia todo el historial de créditos aprobados por trimestre, dando un resultado más eficiente, en las siguientes tablas se muestra la comparación de ambos algoritmos:

		Pronóstico											
		20	21										
<p style="text-align: center;"><b>Promedio móvil</b></p> <table border="1"> <thead> <tr> <th>Trimestres</th> <th>créditosPorTrimestre</th> </tr> </thead> <tbody> <tr> <td>26</td> <td>0</td> </tr> <tr> <td>27</td> <td>11</td> </tr> <tr> <td>28</td> <td>0</td> </tr> <tr> <td>29</td> <td>3,66666667</td> </tr> </tbody> </table>		Trimestres	créditosPorTrimestre	26	0	27	11	28	0	29	3,66666667	21	20
		Trimestres	créditosPorTrimestre										
		26	0										
		27	11										
		28	0										
		29	3,66666667										
		22	0										
		23	0										
		24	0										
		25	0										
26	0												
27	11												
28	0												
29	12,12451934												

Tabla 2.4.3.1 Comparación de los resultados de las fórmulas

Como se puede observar, el alumno en dos trimestres no tuvo actividad escolar, por ende, al aplicar promedio móvil el **Máximo Recomendado** da como resultado un número pequeño en comparación del pronóstico. Para obtener el máximo número de créditos a sugerir, se implementó la siguiente fórmula:

$$\text{Máximo Recomendado} \leftarrow \text{Pronóstico} \leftarrow a + bx, \text{ donde}$$

<sup>12</sup> Un promedio móvil es un método utilizado para analizar un conjunto de datos en modo de puntos para crear series de promedios.

<sup>13</sup> Calcula o pronostica un valor futuro a través de los valores existentes. La predicción del valor es un valor "y", teniendo en cuenta un valor "x". Los valores conocidos son "x" y "y", y el nuevo valor se pronostica utilizando regresión lineal. Esta función se puede utilizar para realizar previsiones de ventas, establecer requisitos de inventario o tendencias de los consumidores.

$$a = \bar{y} - b\bar{x}$$

y:

$$b = \frac{\sum (x - \bar{x})(y - \bar{y})}{\sum (x - \bar{x})^2}$$

Imagen 2.4.3.2 Formula del pronóstico

Y donde “x” corresponde a los valores de *Trimestres* e “y” corresponde a los valores de *créditosPorTrimestre*.

Una vez que se obtiene el **Máximo Recomendado**, el siguiente paso es compararlo contra el **Mínimo Recomendado** y **Máximo Viable**, de la siguiente manera:

- 1 Si *Mínimo Recomendado* > *Máximo Recomendado*, entonces
- 2 *Máximo Recomendado* ← *Mínimo Recomendado*
- 3 Si *Máximo Recomendado* > *Máximo Viable*, entonces
- 4 *Máximo Recomendado* ← *Máximo Viable*

#### 2.4.4 Ordenar las UEA en el Grafo

Para la selección de las UEA a sugerir, se implementó un algoritmo cuya función es ordenar las UEA en una lista, primero, en base a su *Prioridad* y segundo, con respecto al *Trimestre*, de la siguiente manera:

1. Ordenamiento de las UEA de mayor a menor *Prioridad*.

Grafo				
UEA	Nombre	Créditos	Prioridad	Trimestre
1112013	Complementos de Matemáticas	9	24	2
1111078	Introducción a la Física	4	17	1
1112026	Taller de Matemáticas	7	25	1
Sin ordenar				

Grafo				
UEA	Nombre	Créditos	Prioridad	Trimestre
1112026	Taller de Matemáticas	7	25	1
1112013	Complementos de Matemáticas	9	24	2
1111078	Introducción a la Física	4	17	1
Ordenado por Prioridad				

2. Ordenamiento de las UEA, de menor a mayor, con base en el *Trimestre*.

Grafo				
UEA	Nombre	Créditos	Prioridad	Trimestre
1112026	Taller de Matemáticas	7	25	1
1112013	Complementos de Matemáticas	9	24	2
1111078	Introducción a la Física	4	17	1
Ordenado por Prioridad				

Grafo				
UEA	Nombre	Créditos	Prioridad	Trimestre
1112026	Taller de Matemáticas	7	25	1
1111078	Introducción a la Física	4	17	1
1112013	Complementos de Matemáticas	9	24	2
Ordenado por Trimestre				

Una vez ordenada la lista, se seleccionan las UEA habilitadas, teniendo cuidado que el acumulado de los créditos de cada selección no exceda el **Máximo Recomendado**.

## 2.5 Módulo de presentación: Interfaz gráfica

El propósito de este último módulo es mostrar al alumno las UEA sugeridas por el *Sistema de recomendación*.

Una vez que el alumno inicia sesión en la aplicación, la primera pantalla le mostrará todas las UEA optativas de integración que ha acreditado, diferenciando con colores las UEA\*<sup>14</sup>. El color verde para el área de Gestión Industrial, y el color anaranjado para Producción y Manufactura, mostrada en la siguiente imagen:

Clave	UEA	Créditos
1133049	Metrología para Manufactura *	6
1121048	Sistemas Digitales y Redes	9
1154029	Análisis y Diseño de Procesos	9

\*UEA acreditada para obtener el certificado de Gestión  
 \*UEA acreditada para obtener el certificado de Producción

Imagen 2.5.1 Optativas de integración acreditadas

Dependiendo la opción que el alumno elija, se le mostrarán las UEA optativas del área de concentración que le faltaría acreditar para obtener el certificado del área seleccionada.

Clave	UEA	Créditos
1154036	Ingeniería de Mercados *	9
1154017	Sistemas de Gestión de la Calidad *	9
1154035	Planificación Estratégica *	9

Clave	UEA	Créditos
1133051	Logística de Mercados para Manufactura *	3
1154040	Módulo Avanzado de Procesos *	9
1154037	Diseño e Implementación de Procesos *	9

Figura 2.5.2 UEA\* optativas no acreditadas, necesarias para obtener el certificado del área seleccionada.

<sup>14</sup> El alumno que apruebe todas las UEA marcadas con asterisco le costará en su certificado de estudios total el área de concentración, ya sea el área de Gestión Industrial o el área de Producción y Manufactura.

Al dar clic en “Ver recomendación”, aparecerá la sugerencia de las UEA a inscribir en su próximo trimestre, adicionalmente la interfaz muestra dos botones, “Finalizar” y “Cambiar recomendación”.

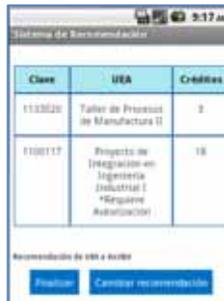


Imagen 2.5.3 Sugerencia de UEA a inscribir

En cuanto a estos botones, el de “Finalizar”, sirve para terminar la recomendación con la opción de poder enviar una opinión sobre la aplicación.



Imagen 2.5.4 Finalizar, con opción para enviar una opinión sobre la aplicación

Mientras que el botón “Cambiar recomendación” sirve para modificar la sugerencia, mostrando una interfaz donde se pide ingresar el número de créditos, tomando como mínimo, el número de créditos de la UEA que se encuentra al inicio de la lista en el Grafo y como máximo, el Máximo Viable.



Imagen 2.5.4 Cambiar recomendación, ingreso del número de créditos, mostrando la nueva recomendación

### 3. Conclusiones

La elaboración de este proyecto me permitió conocer desde otro enfoque, las problemáticas varias a las cuales se puede enfrentar un alumno al momento de inscribir sus materias. En cuanto al proyecto en general, el aporte fue ofrecer una herramienta que sirviera de guía al alumno de Ingeniería Industria, a tomar una mejor decisión de las UEA que debe inscribir en su próximo trimestre, haciendo uso de la tecnología.

Se logró dar solución a los objetivos planteados para el *Sistema de recomendación*. Cabe aclarar, que a lo largo del proceso para la implementación de la recomendación, se encontraron problemas de mayor a menor relevancia, que fueron resueltos satisfactoriamente. Por ejemplo, se pudo demostrar que es posible realizar una conexión directa al SAE, desde una aplicación externa, haciendo uso de las bibliotecas que proporciona Android.

Es importante aclarar, que esta herramienta se diseñó solamente para mostrar una sugerencia de UEA a inscribir para alumnos de la carrera de Ingeniería Industrial. Sin embargo, puede servir como base para futuros proyectos terminales, ya que:

- 1) El *Sistema de recomendación* no verifica la información de los grupos y horarios disponibles en el proceso de inscripción trimestral en la UAM Azcapotzalco, por ende, se pueden implementar nuevos sistemas que muestren diferentes recomendaciones, tomando la información antes mencionada.
- 2) En cada institución de educación, siempre hay materias con un mayor índice de dificultad o complejidad con respecto a otra, y por lo general, se manejan estadísticas con un índice de aprobación, no aprobación y abandono para cada materia. La UAM no es la excepción, por ello, si se utilizan esas estadísticas se puede obtener un índice de dificultad para cada UEA y agregarla a la recomendación.
- 3) Tampoco contempla el índice de demanda que tiene cada UEA, el cual se puede agregar para mejorar la recomendación.

- 4) Puede servir como base para ser implementada en otras carreras de la UAM, o tal vez, para la implementación de nuevas herramientas, que en vez de mostrar las UEA sugeridas, muestre las UEA que el alumno debe inscribir por trimestre, para terminar en cierto tiempo.
- 5) Normalmente un alumno tiene preferencia hacia un profesor por cada UEA, por diferentes razones. Si en un futuro se le pregunta a un grupo de alumnos, de qué profesor aprendieron más o cuál impartió mejor la materia, daría pie a agregar esta información en la sugerencia, como un filtro para mostrar las UEA con esa preferencia.

Cabe aclarar que éstas, son solo algunas áreas de oportunidad que se podrían implementar. Por último, y no menos importante, el *Sistema de recomendación* utiliza un algoritmo, que determina un número de *Prioridad* para cada UEA, éste puede servir como base para mejorar la recomendación que el coordinador de Ingeniería Industrial ofrece a sus alumnos.

## **ANEXO 1. Manual de instalación**

## Resumen

Este manual presenta la forma correcta de instalar el *Sistema de recomendación* de UEA para Ingeniería Industrial, en el dispositivo móvil del alumno de la carrera de Ingeniería Industrial.

### Requerimientos mínimos:

- Dispositivo móvil con sistema operativo Android 2.2 (Froyo).

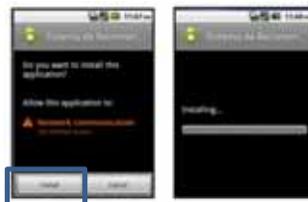
### Solicitud y descarga de la aplicación:

- El alumno debe solicitar la aplicación al coordinador de la carrera de Ingeniería Industrial. La cuál se le enviará al correo que indique.
- Descargar la aplicación en el dispositivo móvil.
- Buscar el archivo en la carpeta de Descargas del dispositivo móvil

### Instalación:

Para la instalación, seguir las indicaciones:

- 1) Dar clic al archivo “Sistema de Recomendación”.apk, a continuación dar clic en Install, como se muestra en la siguiente imagen.



Anexo 1.1 Apertura del archivo, “Sistema de Recomendación”.apk

- 2) Finalizada la instalación dar clic en Open, como se muestra a continuación.



Anexo 1.2 Apertura de la aplicación instalada

## **ANEXO 2. Manual de usuario**

## Tabla de contenido

1. Introducción
2. Limitaciones del software
3. Forma de uso
4. Errores

### 1. Introducción

En este manual de usuario se presenta la forma de utilizar el *Sistema de recomendación* de UEA, para alumnos de la carrera de Ingeniería Industrial de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco.

### 2. Limitaciones del software

Para poder hacer uso apropiado de la aplicación, es indispensable ser alumno de la carrera Ingeniería Industrial de la Universidad Autónoma Metropolitana, Unidad Azcapotzalco.

### 3. Formas de uso

Esta sección explica el uso y funcionamiento del Sistema de recomendación y se divide de la siguiente manera:

- Ingreso
- Optativas de integración acreditadas
- Optativas de integración no acreditadas
- Ver recomendación
- Cambiar recomendación
- Finalizar

### 3.1 Ingreso

Al abrir la aplicación (instalada previamente en el dispositivo móvil), aparecerán dos campos donde se le solicitará ingresar su matrícula y contraseña (misma que ingresa cuando inicia sesión en el SAE), oprimir el botón “Enviar”.



Anexo 2.3.1 Ingreso de matrícula y contraseña

### 3.2 Optativas de integración acreditadas

Una vez que se inicia sesión, se mostrarán todas las UEA optativas de integración que se han acreditado. Diferenciando con colores las UEA\*<sup>15</sup>, el color verde para el área de Gestión Industrial, y el color anaranjado para Producción y Manufactura. Se elige la recomendación en base a un área de concentración.



Clave	UEA	Créditos
1133049	Metrología para Manufactura *	6
1121048	Sistemas Digitales y Redes	9
1154029	Análisis y Diseño	9

Anexo 2.3.2 Optativas de integración acreditadas

### 3.3 Optativas de integración no acreditadas

<sup>15</sup> El alumno que apruebe todas las UEA marcadas con asterisco le costará en su certificado de estudios total el área de concentración, ya sea el área de Gestión Industrial o el área de Producción y Manufactura.

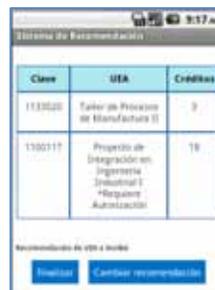
Al seleccionar un área de concentración, mostrará las UEA\* optativas que se deben acreditar para obtener el certificado con el perfil de dicha área. Para continuar con la recomendación, oprimir el botón “Ver recomendación”.



Anexo 2.3.3 UEA\* optativas necesarias para obtener el certificado del área seleccionada.

### 3.4 Ver recomendación

Muestra la sugerencia de las UEA a inscribir



Anexo 2.3.4 Sugerencia de UEA a inscribir

### 3.5 Cambiar recomendación

Esta opción permite modificar la recomendación, agregando un número dentro del rango establecido por la aplicación.



Anexo 2.3.5 Cambiar recomendación, ingreso del número de créditos, mostrando la nueva recomendación

### 3.6 Finalizar

Sirve para terminar la recomendación con la opción de poder enviar una opinión sobre la aplicación.



Anexo 2.3.6 Finalizar, con opción para enviar una opinión sobre la aplicación

## 4. Errores

En base a los errores detectados, este apartado se divide en las siguientes sub secciones:

- Error de conexión.
- Matrícula o Contraseña incorrectas.
- Problemas para enviar la opinión sobre la aplicación.

### 4.1 Error de conexión

El error se puede deber a dos situaciones, no hay conexión a una red de internet o existe un problema de conexión con el SAE.

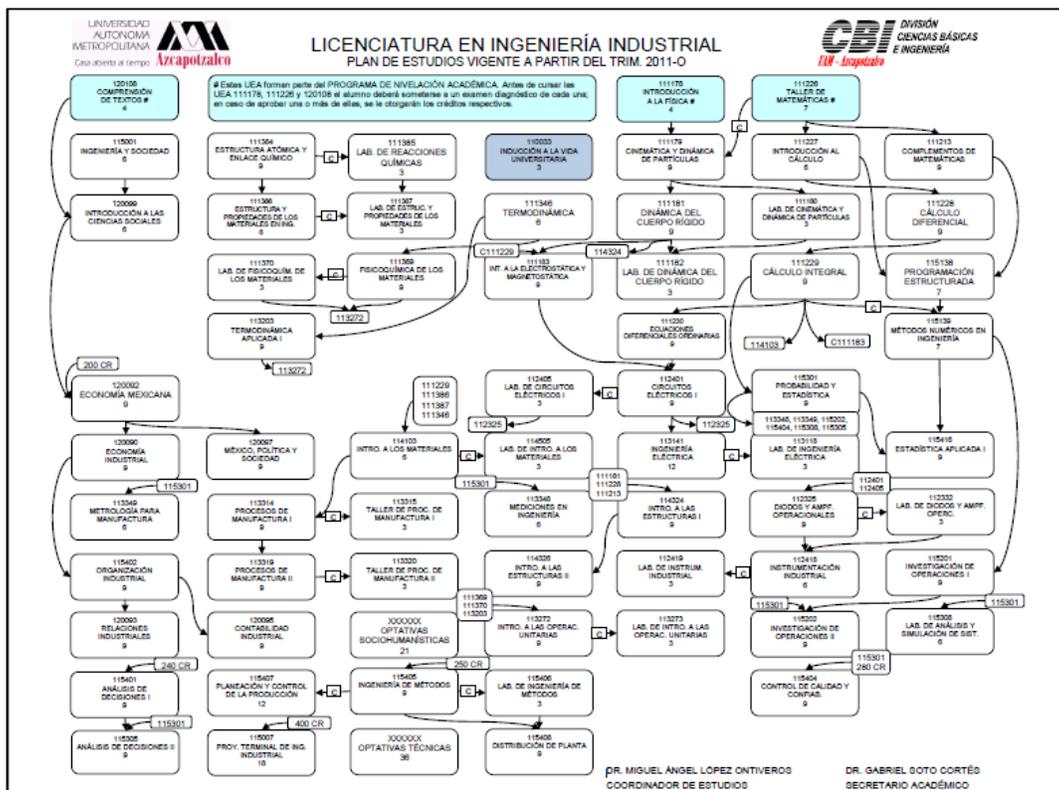
### 4.2 Matrícula o Contraseña incorrectos

Se tiene que verificar que los datos ingresados sean correctos. Recordando nuevamente que la contraseña es la misma que se ingresa en el SAE (Módulo de Información Escolar) para iniciar sesión.

### 4.3 Problemas para enviar la opinión sobre la aplicación

Si al enviar una opinión aparece: “Debes tener activada una cuenta de correo electrónico en tu dispositivo móvil”, verificar que se tenga dada de alta una cuenta de correo electrónico “gmail” en el dispositivo móvil.

## **ANEXO 3. Boligrama de plan de estudios de Ingeniería Industrial**



Anexo 3.1, Boligrama de la carrera de Ingeniería Industrial

**ANEXO 4. Artículo 22, Fracción V del  
Reglamento de Estudios Superiores**

Conforme al Artículo 22, Fracción V del Reglamento de Estudios Superiores, tienen derecho a exceso en créditos los alumnos:

V. Que en aquellos casos en que el alumno desee inscribirse a unidades de enseñanza-aprendizaje, cuyos créditos trimestrales excedan al número fijado como normal en el plan de estudios, en el trimestre inmediato anterior:

- a) Haya estado inscrito cuando menos a un número de créditos igual o mayor a la mitad del fijado en el plan de estudios como normal.
- b) No haya renunciado a ninguna inscripción.
- c) Haber acreditado todas las unidades de enseñanza-aprendizaje a las que estuvo inscrito y obtenido un número de calificaciones MB igual o mayor al número de calificaciones S, o bien B en todas ellas.

Es importante aclarar que si se inscribe una Evaluación de Recuperación y se cumple con lo anterior, el exceso en créditos se pospone hasta el proceso de altas bajas y cambios, siempre y cuando no se modifiquen con la evaluación presentada, las condiciones del Reglamento.

## **ANEXO 5. Código fuente**

## Implementación de las clases:

### Clase Nodo

```
package com.example.sistema_de_recomendacion;

import java.util.ArrayList;

public class Nodo {

    private String Clave;
    private String UEA;
    private String Tipo; //OBLIGATORIA U OPTATIVA
    private int Creditos;
    private int Trimestre=0; //Ubicacion de la UEA en el trimestre
    private int NA_Permitidas;
    private boolean UEA_Habilitada=false;
    private boolean ok=false;

    private ArrayList<Nodo> hijos;

    private boolean requisito_adicional=false;
    private int Creditos_Necesarios=0;

    public int Prioridad=0;
    public String Calificacion="";
    public boolean Aceditada=false;
    public int NA=0;
    public int Numero_de_Padres=0;

    public boolean visitado=false;

    private boolean corregistro=false;
    String Corregistro="";

    public Nodo(String Clave, String UEA, String Tipo, int Creditos, int Trimestre, int NA_Permitidas, boolean requisito_adicional, int
    Creditos_Necesarios)
    {
        this.Clave=Clave;
        this.UEA=UEA;
        this.Tipo=Tipo;
        this.Creditos=Creditos;
        this.Trimestre=Trimestre;
        this.NA_Permitidas=NA_Permitidas;

        this.requisito_adicional=requisito_adicional;
        this.Creditos_Necesarios=Creditos_Necesarios;

        hijos=new ArrayList<Nodo>();
    }

    public void setrequisito_adicional(boolean requisito_adicional)
    {
        this.requisito_adicional=requisito_adicional;
    }

    public void setUEA_Habilitada(boolean UEA_Habilitada)
    {
        this.UEA_Habilitada=UEA_Habilitada;
    }

    public void setOK(boolean bandera)
    {
        ok=bandera;
    }

    public void setCorregistro(boolean bandera)
    {
        corregistro=bandera;
    }

    public void setCreditos(int Creditos)
    {
        this.Creditos=this.Creditos-Creditos;
    }
}
```

```

    }
    public String getClave()
    {
        return this.Clave;
    }

    public String getUEA()
    {
        return this.UEA;
    }

    public String getTipo()
    {
        return this.Tipo;
    }

    public int getCreditos()
    {
        return this.Creditos;
    }

    public boolean getrequisito_adicional()
    {
        return this.requisito_adicional;
    }

    public int getCreditos_Necesarios()
    {
        return this.Creditos_Necesarios;
    }

    public boolean getUEA_Habilitada()
    {
        return this.UEA_Habilitada;
    }

    public int getTrimestre()
    {
        return this.Trimestre;
    }

    public int getNA_Permitidas()
    {
        return this.NA_Permitidas;
    }

    public void setNodo(Nodo hijos)
    {
        this.hijos.add(hijos);
    }

    public ArrayList<Nodo> getHijos()
    {
        return hijos;
    }

    public boolean getOK()
    {
        return ok;
    }

    public boolean getCorregistro()
    {
        return corregistro;
    }
}

```

## Clase Grafo

```

package com.example.sistema_de_recomendacion;

import java.util.ArrayList;

public class Grafo {

    private ArrayList<Nodo> listaNodo;
    private int CreditosUEA_Obligatorias=0;
    private int CreditosUEA_OPT_INTER=0;

```

```

private int CreditosUEA_OPT_INTEGRACION=0;

public Grafo(){
    listaNodo=new ArrayList<Nodo>();
}

public void adjuntarNodo(Nodo nodo)
{
    listaNodo.add(nodo);
}

public Nodo buscarNodo(String Clave)
{
    Nodo temp=null;
    for(int i=0; i<listaNodo.size(); i++)
    {
        if(Clave.equals(listaNodo.get(i).getClave())){
            return listaNodo.get(i);
        }
    }
    return temp;
}

public void crearEnlaces(String UEA_Padre, String UEA_Hijo)
{
    Nodo padre=buscarNodo(UEA_Padre);
    Nodo hijo=buscarNodo(UEA_Hijo);
    if(padre!=null && hijo!=null)
    {
        hijo.Numero_de_Padres=hijo.Numero_de_Padres+1;
        padre.setNodo(hijo);
    }
}

public int TAM()
{
    return listaNodo.size();
}

public Nodo PosicionNodo(int i)
{
    return listaNodo.get(i);
}

public void PonerEnFalseVisitado()
{
    for(int i=0;i<listaNodo.size(); i++)
    {
        listaNodo.get(i).visitado=false;
    }
}

public int ContarCreditosAcumulados()
{
    int Acumulado=0;
    for(int i=0; i<listaNodo.size(); i++)
    {
        Acumulado=Acumulado+listaNodo.get(i).getCreditos();
    }

    return Acumulado;
}

public void Modificar_Grafo_Respecto_a_Creditos_Acumulados(int Creditos_Acumulados)
{
    for(int i=0; i<listaNodo.size(); i++)
    {
        if(listaNodo.get(i).getrequisito_adicional()==true && Creditos_Acumulados>=listaNodo.get(i).getCreditos_Necesarios()){
            listaNodo.get(i).setrequisito_adicional(false);
            if(listaNodo.get(i).Numero_de_Padres==0)
                listaNodo.get(i).setUEA_Habilitada(true);
        }
    }
}

public void Modificar_Grafo(String Clave, String Calificacion)
{
    Nodo nodo;

    nodo=buscarNodo(Clave);
}

```

```

if(nodo!=null)
{
    if(Calificacion=="NA")
    {
        nodo.NA=nodo.NA+1;
        nodo.Calificacion=Calificacion;
        if (nodo.NA>nodo.getNA_Permitidas())
        {
            nodo.setUEA_Habilitada(false);
            for(int i=0; i<nodo.getHijos().size(); i++)
            {
                if(nodo.getHijos().get(i).Corregistro==nodo.getClave())
                {
                    nodo.getHijos().get(i).setUEA_Habilitada(false);
                }
            }
        }
    }
}

else
{
    if(nodo.getTipo()=="OBL")
        CreditosUEA_Obligatorias+=nodo.getCreditos();
    if(nodo.getTipo()=="OPT-INTEGRACION"
        || nodo.getTipo()=="OPT-INTEGRACION-PRODUCCION"
        || nodo.getTipo()=="OPT-INTEGRACION-GESTION")
        CreditosUEA_OPT_INTEGRACION+=nodo.getCreditos();
    if(nodo.getTipo()=="OPT_INTER-MULTI")
        CreditosUEA_OPT_INTER+=nodo.getCreditos();

    nodo.Calificacion=Calificacion;
    nodo.Acreditada=true;
    /////Deshabilita las OPT_INTER en caso de haberse aprobado alguna
    if(nodo.getTipo()=="OPT_INTER-MULTI")
    {
        Nodo aux;

        aux=buscarNodo("xxxxx1");

        if(aux.Acreditada!=true)
            aux.Acreditada=true;

        else
        {
            aux=buscarNodo("xxxxx2");
            if(aux.Acreditada!=true)
                aux.Acreditada=true;

            else
            {
                aux=buscarNodo("xxxxx3");
                if(aux.Acreditada!=true)
                    aux.Acreditada=true;

                else
                {
                    aux=buscarNodo("xxxxx4");
                    if(aux.Acreditada!=true)
                        aux.Acreditada=true;
                }
            }
        }
    }
}

else
{
    for(int i=0; i<nodo.getHijos().size(); i++)
    {
        if(nodo.getClave()!=nodo.getHijos().get(i).Corregistro)
            nodo.getHijos().get(i).Numero_de_Padres--;

        ///Si UEA necesita corregistro decrementar en -1 Numero de Padres

        for(int j=0; j<nodo.getHijos().get(i).getHijos().size(); j++)
            if(nodo.getHijos().get(i).getHijos().get(j).Corregistro==nodo.getHijos().get(i).getClave())

```



```

if(nodo!=null)
    nodo.setUEA_Habilitada(false);

nodo=buscarNodo("1154037");
if(nodo!=null)
    nodo.setUEA_Habilitada(false);

nodo=buscarNodo("1154047");
if(nodo!=null)
    nodo.setUEA_Habilitada(false);

nodo=buscarNodo("1154049");
if(nodo!=null)
    nodo.setUEA_Habilitada(false);

//Deshabilitamos el resto de UEA OPT de Integracion Genericas

nodo=buscarNodo("xxxxx15");
if(nodo!=null)
    nodo.setUEA_Habilitada(false);

nodo=buscarNodo("xxxxx16");
if(nodo!=null)
    nodo.setUEA_Habilitada(false);

nodo=buscarNodo("xxxxx17");
if(nodo!=null)
    nodo.setUEA_Habilitada(false);
}
/*****/
else{

    if(Tipo==2){
        OPT=OPT+Produccion;

    }

    if(Tipo==1){
        OPT=OPT+Gestion;
    }

    if(OPT>=24){
        //Deshabilitamos el resto de UEA OPT de Integracion
        nodo=buscarNodo("xxxxx15");
        if(nodo!=null)
            nodo.setUEA_Habilitada(false);

        nodo=buscarNodo("xxxxx16");
        if(nodo!=null)
            nodo.setUEA_Habilitada(false);

        nodo=buscarNodo("xxxxx17");
        if(nodo!=null)
            nodo.setUEA_Habilitada(false);
    }
    else {

        if(OPT>=15){

            nodo=buscarNodo("xxxxx15");
            if(nodo!=null)
                nodo.setUEA_Habilitada(false);

            nodo=buscarNodo("xxxxx16");
            if(nodo!=null)
                nodo.setUEA_Habilitada(false);
            OPT=OPT-15;
            nodo=buscarNodo("xxxxx17");
            if(nodo!=null)
                nodo.setCreditos(OPT);

        }
        else{

            if(OPT>=9){

                nodo=buscarNodo("xxxxx15");

```



```

static String Matricula="";
static String Password="";

//Variables para el módulo de Solicitud
static String Informacion_Academica="";
static String Kardex="";
static final String Carrera="LICENCIATURA EN INGENIERIA INDUSTRIAL";
static Header[] cookies;
static boolean Verifica;
private ProgressDialog pDialog;

//Variable en donde se enviar datos a la Actividad resultado de la clase Resultado
static Intent Enviar;

public static Header[] Ingreso()
{
    DefaultHttpClient httpclient = new DefaultHttpClient();
    HttpPost httppost=new HttpPost("https://ayamiclan.uam.mx:8443/sae/azc/AEWBU004");
    HttpResponse responsePost = null;

    ArrayList<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_ÁNEXO.SG02",""));
    params.add(new BasicNameValuePair("%23.MODULO_UWAS.SAE01",""));

    try {
        httppost.setEntity(new UrlEncodedFormEntity(params));
    }
    catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }

    try {
        responsePost = httpclient.execute(httppost);
    }
    catch (ClientProtocolException e) {
        e.printStackTrace();
    }
    catch (IOException e) {
        e.printStackTrace();
    }

    return responsePost.getHeaders("Set-Cookie");
}

public static void Solicitud(Header[] cookies)
{
    DefaultHttpClient httpclient = new DefaultHttpClient();

    HttpGet httpget = new HttpGet("https://ayamiclan.uam.mx:8443/sae/azc/IEWBC020.oConsulta");
    HttpGet httpget2 = new HttpGet("https://ayamiclan.uam.mx:8443/sae/azc/IEWBC007.oConsulta");

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

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

httpClient.getConnectionManager().shutdown();

Verifica=Informacion_Academica.contains(Carrera);
}

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

    Validar=(TextView)findViewById(R.id.idValidar);
}

public void Enviar(View e)
{
    Enviar = new Intent(this, Procesar.class);
    TextView temp;

    pDialog = new ProgressDialog(this);
    pDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
    pDialog.setMessage("Solicitando...");

    temp=(TextView)findViewById(R.id.idCampoMatricula);
    Matricula=(temp.getText().toString());

    temp=(TextView)findViewById(R.id.idCampoPass);
    Password=(temp.getText().toString());

    if(Matricula.length()==0 && Password.length()==0){
        Toast.makeText(MainActivity.this, "Debes ingresar Matricula y Contraseña",
        Toast.LENGTH_SHORT).show();
    }
}

```

```

        else if(Matricula.length()==0){
            Toast.makeText(MainActivity.this, "Debes ingresar Matrícula", Toast.LENGTH_SHORT).show();
        }
        else if( Password.length()==0){
            Toast.makeText(MainActivity.this, "Debes ingresar Contraseña", Toast.LENGTH_SHORT).show();
        }
        else{
            //Se valida la conexion y se obtienen las cookies
            try{
                cookies=Ingreso();
                if(cookies.length!=3){
                    Toast.makeText(MainActivity.this, "Matricula o Contraseña incorrectos.",
Toast.LENGTH_SHORT).show();
                }
                else{
                    new Solicitar().execute();
                }
            }
            catch(Exception error)
            {
                Validar.setText("Error de conexión");
            }
        }
    }

public class Solicitar extends AsyncTask<Void, Integer, Boolean>{
    @Override
    protected Boolean doInBackground(Void... arg0){
        Solicitud(cookies);
        return true;
    }
    @Override
    protected void onPreExecute() {
        pDialog.show();
    }
    @Override
    protected void onPostExecute(Boolean result){
        if(result)
        {
            pDialog.dismiss();
            if(!Verifica){
                Toast.makeText(MainActivity.this, "Aplicación para alumnos de Ing. Industrial.",
Toast.LENGTH_SHORT).show();
            }
            else{
                Enviar.putExtra("Informacion_Academica", Informacion_Academica);
                Enviar.putExtra("Kardex", Kardex);
                startActivity(Enviar);
            }
        }
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}

```

## Clase Procesar

```
package com.example.sistema_de_recomendacion;

import java.util.ArrayList;

import org.jsoup.Jsoup;
import org.jsoup.nodes.Document;
import org.jsoup.nodes.Element;
import org.jsoup.select.Elements;

import android.app.Activity;
import android.app.ProgressDialog;
import android.content.Intent;
import android.content.res.Resources;
import android.graphics.Color;
import android.os.AsyncTask;
import android.os.Bundle;
import android.view.Gravity;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.view.ViewGroup.LayoutParams;
import android.widget.Button;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;
import android.widget.Toast;

public class Procesar extends Activity {

    TableLayout tabla;
    TableLayout cabecera2;
    TableLayout cabecera;
    TableLayout BotonOPTI;

    TableRow.LayoutParams layoutFila;
    TableRow.LayoutParams layoutFila2;
    TableRow.LayoutParams layoutFila3;
    TableRow.LayoutParams layoutGestion;
    TableRow.LayoutParams layoutProduccion;
    TableRow.LayoutParams layoutBoton;
    TableRow.LayoutParams layoutTitulo;
    TableRow.LayoutParams layoutClave;
    TableRow.LayoutParams layoutUEA;
    TableRow.LayoutParams layoutCreditos;
    private ProgressDialog pDialog;

    Resources rs;

    static Grafo grafo;

        static String Informacion_Academica;
        static String Kardex;

    static int Acumulado=0;
    static int Creditos_Acumulados=0;
    static int Creditos[]=new int[100];
    static int Creditos_Inscritos_el_trimestre_Anterior=0;
    static int Numero_de_Trimestre[];
    static int Trimestres=0;
    static int MB=0;
    static int B=0;
    static int S=0;
    static int NA=0;
    static int Numero_de_Renuncias=0;

    static final String Perfil_Manufactura="OPT. AREA DE CONCENTRACION - PRODUCCION Y MANUFACTURA";
    static final String Perfil_Gestion="OPT. AREA DE CONCENTRACION - GESTION INDUSTRIAL";
    static int Creditos_Gestion=0, Creditos_Produccion=0, Creditos_OPT_Integracion=0;

        static String UEA_Habilidades="";

    static int Minimo_Viable=0;
```

```

static int Maximo_Viable=0;

static int Minimo_Recomendado=0;
static int Maximo_Recomendado=0;

static ArrayList<String> Clave=new ArrayList<String>();
static ArrayList<String> Nombre_UEA=new ArrayList<String>();
static ArrayList<Integer> creditos=new ArrayList<Integer>();

static ArrayList<String> Clave_OPTI=new ArrayList<String>();
static ArrayList<String> Nombre_UEAOPTI=new ArrayList<String>();
static ArrayList<Integer> Creditos_OPTI=new ArrayList<Integer>();
static ArrayList<String> Tipo_OPTI=new ArrayList<String>();

static ArrayList<String> ClaveGestion=new ArrayList<String>();
static ArrayList<String> UEA_Gestion=new ArrayList<String>();
static ArrayList<Integer> Creditos_OGestion=new ArrayList<Integer>();

static ArrayList<String> ClaveProduccion=new ArrayList<String>();
static ArrayList<String> UEA_Produccion=new ArrayList<String>();
static ArrayList<Integer> Creditos_OProduccion=new ArrayList<Integer>();

static Intent Gestion;
static Intent Produccion;

public static void llenandoGrafo()
{
    grafo=new Grafo();

    /******Limpiamos el Grafo*****/
    grafo.setLimpiarGrafo();
    /*(ClAVE, UEA, Tipo, Numero de Creditos, Trimestre sugerido, NAs Aceptadas, requisito_adicional, Creditos_Necesarios)*/

    /******Tronco de Nivelacion Academica*****/
    grafo.adjuntarNodo(new Nodo("1100033", "Introducción a la Vida Universitaria", "OBL",3,1,1, false, 0));
    grafo.adjuntarNodo(new Nodo("1111078", "Introducción a la Fisica", "OBL",4, 1,1, false, 0));
    grafo.adjuntarNodo(new Nodo("1112026", "Taller de Matemáticas", "OBL",7, 1,1, false, 0));
    grafo.adjuntarNodo(new Nodo("1201008", "Comprensión de Textos", "OBL",4, 1,1, false, 0));

    grafo.buscarNodo("1100033").setUEA_Habilitada(true);
    grafo.buscarNodo("1111078").setUEA_Habilitada(true);
    grafo.buscarNodo("1112026").setUEA_Habilitada(true);
    grafo.buscarNodo("1201008").setUEA_Habilitada(true);

    /******

    /******Tronco General*****/
    grafo.adjuntarNodo(new Nodo("1111079", "Cinématica y Dinámica de Particulas", "OBL", 9, 2,1, false, 0));
    grafo.adjuntarNodo(new Nodo("1111092", "Laboratorio de Movimiento de una Partícula", "OBL", 3, 3,4, false, 0));
    grafo.adjuntarNodo(new Nodo("1111081", "Dinámica del Cuerpo Rígido", "OBL", 9, 3,1, false, 0));
    grafo.adjuntarNodo(new Nodo("1111093", "Laboratorio del Cuerpo Rígido y Oscilaciones", "OBL", 3, 4,4, false, 0));
    grafo.adjuntarNodo(new Nodo("1111083", "IntroducciAn a la Electrostatica y Magnetostática", "OBL", 9, 4,1, false, 0));
    grafo.adjuntarNodo(new Nodo("1112013", "Complementos de Matemáticas", "OBL", 9, 2,1, false, 0));
    grafo.adjuntarNodo(new Nodo("1112027", "Introducción al Cálculo", "OBL", 6, 2,1, false, 0));
    grafo.adjuntarNodo(new Nodo("1112028", "Cálculo Diferencial", "OBL", 9, 3,1, false, 0));
    grafo.adjuntarNodo(new Nodo("1112029", "Cálculo Integral", "OBL", 9, 4,1, false, 0));
    grafo.adjuntarNodo(new Nodo("1112030", "Ecuaciones Diferenciales Ordinarias", "OBL", 9, 5,1, false, 0));
    grafo.adjuntarNodo(new Nodo("1113046", "Termodinámica", "OBL", 6, 5,1, false, 0));

    grafo.adjuntarNodo(new Nodo("1113084", "Estructura Atómica y Enlace Químico", "OBL",9, 1,1, false, 0));
    grafo.buscarNodo("1113084").setUEA_Habilitada(true);
    grafo.adjuntarNodo(new Nodo("1113085", "Laboratorio de Reacciones Químicas", "OBL",3, 1,4, false, 0));
    grafo.buscarNodo("1113085").setUEA_Habilitada(true);
    grafo.buscarNodo("1113085").Corregistro="1113084";

    grafo.adjuntarNodo(new Nodo("1113086", "Estructura y Propiedades de los Materiales en Ingeniería", "OBL",6, 2,1, false, 0));

    grafo.adjuntarNodo(new Nodo("1113087", "Laboratorio de Estructura y Propiedades de los Materiales", "OBL",3,2,4, false, 0));
    grafo.buscarNodo("1113087").Corregistro="1113086";
    grafo.adjuntarNodo(new Nodo("1151038", "Programación Estructurada", "OBL", 7, 3, 1, false, 0));
    grafo.adjuntarNodo(new Nodo("1151039", "Métodos Numéricos en Ingeniería", "OBL",7, 4, 1, false,0));
    grafo.adjuntarNodo(new Nodo("1153001", "Probabilidad y Estadística", "OBL", 9, 5, 1, false, 0));
    /******

    /******Tronco Basico Profesional*****/
    grafo.adjuntarNodo(new Nodo("1123052", "Fundamentos de Electricidad y Electrónica de Manufactura", "OBL", 9, 5,1, false, 0));
    grafo.adjuntarNodo(new Nodo("1123053", "Laboratorio de Sistemas Electrónicos de Ingeniería Industrial", "OBL", 3, 5, 4, false,
0));
    grafo.adjuntarNodo(new Nodo("1133014", "Procesos de Manufactura I", "OBL", 9, 6, 1, false, 0));
    grafo.adjuntarNodo(new Nodo("1133015", "Taller de Procesos de Manufactura I", "OBL", 3, 6, 1, false, 0));
    grafo.adjuntarNodo(new Nodo("1133019", "Procesos de Manufactura II", "OBL", 9, 7, 1, false, 0));
    grafo.adjuntarNodo(new Nodo("1133020", "Taller de Procesos de Manufactura II", "OBL", 3, 7, 1, false, 0));

```

```

grafo.adjuntarNodo(new Nodo("1133048", "Mediciones en Ingeniería", "OBL", 6, 7, 1, false, 0));
grafo.adjuntarNodo(new Nodo("1137019", "Procesos Físicos Industriales", "OBL", 9, 11, 1, true, 350));
grafo.adjuntarNodo(new Nodo("1145054", "Ingeniería de los Materiales", "OBL", 9, 4, 1, false, 0));
grafo.adjuntarNodo(new Nodo("1145055", "Laboratorio de Ingeniería de los Materiales", "OBL", 3, 4, 4, false, 0));
grafo.adjuntarNodo(new Nodo("1152001", "Investigación de Operaciones I", "OBL", 9, 6, 1, false, 0));
grafo.adjuntarNodo(new Nodo("1152002", "Investigación de Operaciones II", "OBL", 9, 7, 1, false, 0));
grafo.adjuntarNodo(new Nodo("1153005", "Análisis de Decisiones II", "OBL", 9, 10, 1, false, 0));
grafo.adjuntarNodo(new Nodo("1153006", "Ingeniería de Costos", "OBL", 9, 9, 1, false, 0));
grafo.adjuntarNodo(new Nodo("1154001", "Análisis de Decisiones I", "OBL", 9, 8, 1, true, 200));
grafo.adjuntarNodo(new Nodo("1154002", "Organización Industrial", "OBL", 9, 6, 1, true, 100));
grafo.adjuntarNodo(new Nodo("1154016", "Estadística Aplicada I", "OBL", 9, 7, 1, false, 0));
grafo.adjuntarNodo(new Nodo("1154031", "Planeación de la Producción", "OBL", 9, 9, 1, false, 0));
grafo.adjuntarNodo(new Nodo("1154032", "Administración de la Producción", "OBL", 6, 10, 1, false, 0));
grafo.adjuntarNodo(new Nodo("1154038", "Estudio del Método del Trabajo", "OBL", 6, 8, 1, false, 0));
grafo.adjuntarNodo(new Nodo("1154039", "Estudio de la Medición del Trabajo", "OBL", 6, 9, 1, false, 0));
grafo.adjuntarNodo(new Nodo("1154040", "Laboratorio de Estudio del Método del Trabajo", "OBL", 3, 8, 4, false, 0));
grafo.adjuntarNodo(new Nodo("1154041", "Laboratorio de Estudio de la Medición del Trabajo", "OBL", 3, 9, 4, false, 0));
grafo.adjuntarNodo(new Nodo("1154042", "Control de Calidad y Confiabilidad", "OBL", 8, 9, 1, false, 0));
grafo.adjuntarNodo(new Nodo("1154043", "Diseño de Instalaciones y Manejo de Materiales", "OBL", 8, 10, 1, false, 0));
grafo.adjuntarNodo(new Nodo("1154044", "Logística", "OBL", 8, 10, 1, false, 0));
grafo.adjuntarNodo(new Nodo("1154045", "Seguridad e Higiene Industrial", "OBL", 8, 12, 1, false, 0));
grafo.adjuntarNodo(new Nodo("1133061", "Dibujo Mecánico Asistido por Computadora", "OBL", 9, 6, 1, true, 150));
grafo.adjuntarNodo(new Nodo("1200089", "Introducción a la Economía", "OBL", 9, 8, 1, true, 120));
grafo.adjuntarNodo(new Nodo("1200095", "Contabilidad Industrial", "OBL", 9, 8, 1, false, 0));
/*****

//System.out.println(grafo.ContarCreditosAcumulados());

/*(CIAVE, UEA, Tipo, Numero de Creditos, Trimestre sugerido, NAs Aceptadas, requisito_adicional, Creditos_Necesarios)*/
/*****Tronco Inter y Multidisciplinar*****/
grafo.adjuntarNodo(new Nodo("1100037", "Introducción a la Ingeniería", "OBL", 6, 3, 1, true, 50));
grafo.adjuntarNodo(new Nodo("1100038", "Introducción al desarrollo Sustentable", "OBL", 6, 5, 1, false, 0));
grafo.adjuntarNodo(new Nodo("1100040", "Taller de Planeación y Ejecución de Proyectos", "OBL", 6, 9, 1, true, 280));
grafo.adjuntarNodo(new Nodo("1100039", "Innovación", "OBL", 6, 10, 1, false, 0));
grafo.adjuntarNodo(new Nodo("1100041", "Retos del Desarrollo Nacional", "OBL", 6, 12, 1, false, 0));
/*****

/*****Tronco de Integración*****/
grafo.adjuntarNodo(new Nodo("1100107", "Seminario de Integración en Ingeniería Industrial", "OBL", 3, 12, 4, true, 350));
grafo.adjuntarNodo(new Nodo("1100117", "Proyecto de Integración en Ingeniería Industrial I *Requiere Autorización", "OBL", 18,
13, 1, false, 0));
/*****

/*****Tronco Inter y Multidisciplinar OPT*****/

//Estudios Culturales
grafo.adjuntarNodo(new Nodo("1100073", "El Paisaje como Agente de los Asentamientos y de la Cultura", "OPT_INTER-MULTI",
6, 14, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1100074", "Familia y Violencia en el México Contemporáneo", "OPT_INTER-MULTI", 6, 14, 1, true,
1500));
grafo.adjuntarNodo(new Nodo("1100075", "Género y Sexualidad", "OPT_INTER-MULTI", 6, 14, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1100076", "Poder y Género", "OPT_INTER-MULTI", 6, 14, 1, true, 1500));

//Formación Ciudadana
grafo.adjuntarNodo(new Nodo("1100077", "Administración y Economía Política de la Ciencia y Tecnología", "OPT_INTER-MULTI",
6, 7, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1100078", "Derechos Humanos", "OPT_INTER-MULTI", 6, 7, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1100079", "Economía Mundial", "OPT_INTER-MULTI", 6, 7, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1100080", "Ética y Valores", "OPT_INTER-MULTI", 6, 7, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1100081", "Historia Social de México en el Siglo XX", "OPT_INTER-MULTI", 6, 7, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1100082", "Responsabilidad Social Organizacional", "OPT_INTER-MULTI", 6, 7, 1, true, 1500));

//Inducción al Mercado Laboral
grafo.adjuntarNodo(new Nodo("1100083", "Comunicación en Proyectos Multidisciplinarios", "OPT_INTER-MULTI", 6, 7, 1, true,
1500));
grafo.adjuntarNodo(new Nodo("1100084", "Herramientas para el Emprendedor", "OPT_INTER-MULTI", 6, 7, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1100085", "Inserción Laboral", "OPT_INTER-MULTI", 6, 7, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1100086", "Planeación Estratégica", "OPT_INTER-MULTI", 6, 7, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1100087", "Proyectos de Inversión", "OPT_INTER-MULTI", 6, 7, 1, true, 1500));

//Arte y Humanidades
grafo.adjuntarNodo(new Nodo("1100088", "Historia del Arte", "OPT_INTER-MULTI", 6, 7, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1100089", "Taller de Dibujo", "OPT_INTER-MULTI", 6, 7, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1100090", "Taller de Fotografía", "OPT_INTER-MULTI", 6, 7, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1100091", "Taller de Teatro", "OPT_INTER-MULTI", 6, 7, 1, true, 1500));

//Lenguajes Formales
grafo.adjuntarNodo(new Nodo("1100092", "Divulgación del Conocimiento", "OPT_INTER-MULTI", 6, 7, 1, true, 1500));

```

```

    grafo.adjuntarNodo(new Nodo("1100093", "Habilidades Creativas para el Ámbito Profesional", "OPT_INTER-MULTI", 6, 7, 1, true,
1500));
    grafo.adjuntarNodo(new Nodo("1100094", "Laboratorio de Usabilidad", "OPT_INTER-MULTI", 6, 7, 1, true, 1500));
    grafo.adjuntarNodo(new Nodo("1100095", "Narrativa para Medios Audiovisuales y Digitales", "OPT_INTER-MULTI", 6, 7, 1, true,
1500));
    grafo.adjuntarNodo(new Nodo("1100096", "Taller de Expresión Oral y Escrita", "OPT_INTER-MULTI", 6, 7, 1, true, 1500));

//Otras Optativas Inter y Multidisciplinares
    grafo.adjuntarNodo(new Nodo("1100097", "Temas Selectos Inter y Multidisciplinar I", "OPT_INTER-MULTI", 6, 7, 1, true, 1500));
    grafo.adjuntarNodo(new Nodo("1100098", "Temas Selectos Inter y Multidisciplinar II", "OPT_INTER-MULTI", 6, 7, 1, true, 1500));
    grafo.adjuntarNodo(new Nodo("1100098", "Experiencia Inter y Multidisciplinar", "OPT_INTER-MULTI", 6, 7, 1, true, 1500));

//OPT_INTER-MULTI
    grafo.adjuntarNodo(new Nodo("xxxxx1", "Optativa Inter y Multidisciplinar", "OPT_INTER-MULTI", 6, 7, 1, true, 150));
    grafo.adjuntarNodo(new Nodo("xxxxx2", "Optativa Inter y Multidisciplinar", "OPT_INTER-MULTI", 6, 8, 1, true, 150));
    grafo.adjuntarNodo(new Nodo("xxxxx3", "Optativa Inter y Multidisciplinar", "OPT_INTER-MULTI", 6, 11, 1, true, 150));
    grafo.adjuntarNodo(new Nodo("xxxxx4", "Optativa Inter y Multidisciplinar", "OPT_INTER-MULTI", 6, 12, 1, true, 150));

/*****/

/*****Tronco de Integracion OPT-INTEGRACION*****/
//Tutoriales
    grafo.adjuntarNodo(new Nodo("1100127", "Proyecto de integración de Ingeniería Industrial II", "OPT-INTEGRACION", 18, 12, 1,
true, 1500));
    grafo.adjuntarNodo(new Nodo("1100137", "Introducción al trabajo de investigación en Ingeniería Industrial", "OPT-
INTEGRACION", 6, 12, 1, true, 1500));
    grafo.adjuntarNodo(new Nodo("1150013", "Prácticas Profesionales de Ingeniería Industrial", "OPT-INTEGRACION", 18, 12, 1, true,
1500));

//De Movilidad
    grafo.adjuntarNodo(new Nodo("1100021", "Optativa Técnica de Movilidad I", "OPT-INTEGRACION", 3, 12, 1, true, 1500));
    grafo.adjuntarNodo(new Nodo("1100022", "Optativa Técnica de Movilidad II", "OPT-INTEGRACION", 3, 12, 1, true, 1500));
    grafo.adjuntarNodo(new Nodo("1100023", "Optativa Técnica de Movilidad III", "OPT-INTEGRACION", 6, 12, 1, true, 1500));
    grafo.adjuntarNodo(new Nodo("1100024", "Optativa Técnica de Movilidad IV", "OPT-INTEGRACION", 6, 12, 1, true, 1500));
    grafo.adjuntarNodo(new Nodo("1100025", "Optativa Técnica de Movilidad V", "OPT-INTEGRACION", 9, 12, 1, true, 1500));
    grafo.adjuntarNodo(new Nodo("1100026", "Optativa Técnica de Movilidad VI", "OPT-INTEGRACION", 9, 12, 1, true, 1500));

//Científico-Tecnicas

//Area de Concentración: Producción y Manufactura
    grafo.adjuntarNodo(new Nodo("1121048", "Sistemas Digitales y Redes", "OPT-INTEGRACION", 9, 12, 1, true, 1500));
    grafo.adjuntarNodo(new Nodo("1123054", "Tecnologías de la información industrial", "OPT-INTEGRACION", 9, 12, 1, true, 1500));
    grafo.adjuntarNodo(new Nodo("1124042", "Introducción a la Automatización Industrial", "OPT-INTEGRACION", 9, 12, 1, true,
1500));
    grafo.adjuntarNodo(new Nodo("1131018", "Laboratorio de Ingeniería Eléctrica", "OPT-INTEGRACION", 3, 12, 4, true, 1500));
    grafo.adjuntarNodo(new Nodo("1131041", "Ingeniería Eléctrica", "OPT-INTEGRACION", 12, 12, 1, true, 1500));
    grafo.adjuntarNodo(new Nodo("1133044", "Taller de Manufactura Asistido por Computadora", "OPT-INTEGRACION", 9, 12, 1,
true, 1500));
    grafo.adjuntarNodo(new Nodo("1133046", "Control Numérico Computarizado", "OPT-INTEGRACION", 9, 12, 1, true, 1500));
    grafo.adjuntarNodo(new Nodo("1133049", "Metrología para Manufactura **", "OPT-INTEGRACION-PRODUCCION", 6, 11, 1, false,
0));
    grafo.adjuntarNodo(new Nodo("1133055", "Laboratorio de Metrología para Manufactura **", "OPT-INTEGRACION-PRODUCCION",
3, 11, 4, false, 0));
    grafo.adjuntarNodo(new Nodo("1154029", "Análisis y Diseño de Experimentos en Ingeniería", "OPT-INTEGRACION", 9, 12, 1,
true, 1500));
    grafo.adjuntarNodo(new Nodo("1154037", "Diseño e Innovación de Productos **", "OPT-INTEGRACION-PRODUCCION", 9, 13, 1,
false, 0));
    grafo.adjuntarNodo(new Nodo("1154046", "Ergonomía Industrial", "OPT-INTEGRACION", 6, 12, 1, true, 1500));
    grafo.adjuntarNodo(new Nodo("1154047", "Mantenimiento Industrial **", "OPT-INTEGRACION-PRODUCCION", 9, 13, 1, false, 0));
    grafo.adjuntarNodo(new Nodo("1154048", "Ecodiseño", "OPT-INTEGRACION", 9, 12, 1, true, 1500));
    grafo.adjuntarNodo(new Nodo("1154049", "Mejora Continua de Procesos **", "OPT-INTEGRACION-PRODUCCION", 9, 12, 1,
false, 0));
    grafo.adjuntarNodo(new Nodo("1154050", "Temas Selectos de Producción y Manufactura I", "OPT-INTEGRACION", 9, 12, 1, true,
1500));
    grafo.adjuntarNodo(new Nodo("1154051", "Temas Selectos de Producción y Manufactura II", "OPT-INTEGRACION", 9, 12, 1, true,
1500));
    grafo.adjuntarNodo(new Nodo("1154059", "Sistemas de Manufactura Esbelta", "OPT-INTEGRACION", 6, 12, 1, true, 1500));
    grafo.adjuntarNodo(new Nodo("1431048", "Normalización y Estandarización", "OPT-INTEGRACION", 6, 12, 1, true, 1500));

//Area de Concentración: Gestión Industrial
    grafo.adjuntarNodo(new Nodo("1154015", "Administración de Proyectos", "OPT-INTEGRACION", 6, 12, 1, true, 1500));
    grafo.adjuntarNodo(new Nodo("1154017", "Sistemas de Gestión de la Calidad **", "OPT-INTEGRACION-GESTION", 9, 12, 1, false,
0));
    grafo.adjuntarNodo(new Nodo("1154018", "Administración de la Calidad", "OPT-INTEGRACION", 6, 12, 1, true, 1500));
    grafo.adjuntarNodo(new Nodo("1154022", "Desarrollo Organizacional", "OPT-INTEGRACION", 9, 12, 1, true, 1500));
    grafo.adjuntarNodo(new Nodo("1154025", "Ingeniería Financiera", "OPT-INTEGRACION", 9, 12, 1, true, 1500));
    grafo.adjuntarNodo(new Nodo("1154027", "Métodos Cuantitativos para la Toma de Decisiones", "OPT-INTEGRACION", 9, 12, 1,
true, 1500));

```

```

grafo.adjuntarNodo(new Nodo("1154030", "Psicología Industrial", "OPT-INTEGRACION", 9, 12, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1154034", "Taller integral de Ingeniería Industrial", "OPT-INTEGRACION", 3, 12, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1154035", "Planeación Estratégica **", "OPT-INTEGRACION-GESTION", 9, 13, 1, true, 250));
grafo.adjuntarNodo(new Nodo("1154036", "Procesos de Mercadeo **", "OPT-INTEGRACION-GESTION", 9, 11, 1, false, 0));
grafo.adjuntarNodo(new Nodo("1154052", "Temas Selectos de Gestión Industrial I", "OPT-INTEGRACION", 9, 12, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1154053", "Temas Selectos de Gestión Industrial II", "OPT-INTEGRACION", 9, 12, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1154054", "Habilidades Gerenciales", "OPT-INTEGRACION", 9, 12, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1154055", "Innovación y Administración de la Tecnología en la Empresa", "OPT-INTEGRACION",
9, 12, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1154056", "Ingeniería de Producto", "OPT-INTEGRACION", 9, 12, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1154057", "Gestión de Servicios **", "OPT-INTEGRACION-GESTION", 9, 13, 1, false, 0));

//Otras Optativas de Integración
grafo.adjuntarNodo(new Nodo("1135062", "Evaluación Ambiental de Tecnologías", "OPT-INTEGRACION", 9, 12, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1135096", "Prevención y Minimización de la Contaminación Ambiental", "OPT-INTEGRACION", 9,
12, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1136005", "Retos Ambientales", "OPT-INTEGRACION", 6, 12, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1155005", "Análisis de Problemas", "OPT-INTEGRACION", 6, 12, 1, true, 1500));
grafo.adjuntarNodo(new Nodo("1155008", "Creatividad y Solución de Problemas", "OPT-INTEGRACION", 6, 12, 1, true, 1500));

//OPT-INTEGRACION GENERICAS
grafo.adjuntarNodo(new Nodo("xxxxx15", "Optativa del Tronco del Integración", "OPT-INTEGRACION", 9, 11, 1, false, 0));
grafo.buscarNodo("xxxxx15").setUEA_Habilitada(true);
grafo.adjuntarNodo(new Nodo("xxxxx16", "Optativa del Tronco del Integración", "OPT-INTEGRACION", 6, 11, 1, false, 0));
grafo.buscarNodo("xxxxx16").setUEA_Habilitada(true);
grafo.adjuntarNodo(new Nodo("xxxxx17", "Optativa del Tronco del Integración", "OPT-INTEGRACION", 9, 12, 1, false, 0));
grafo.buscarNodo("xxxxx17").setUEA_Habilitada(true);

/*****/

/***** Creacion de los enlaces *****/

//Tronco General *****/

grafo.crearEnlaces("1111078", "1111079");
grafo.crearEnlaces("1112026", "1111079");
grafo.buscarNodo("1111079").setCorregistro(true);
grafo.buscarNodo("1111079").Corregistro="1112026";

grafo.crearEnlaces("1111079", "1111092");

grafo.crearEnlaces("1111079", "1111081");

grafo.crearEnlaces("1111081", "1111093");
grafo.crearEnlaces("1111092", "1111093");

grafo.crearEnlaces("1111081", "1111083");
grafo.crearEnlaces("1112029", "1111083");
grafo.buscarNodo("1111083").setCorregistro(true);
grafo.buscarNodo("1111083").Corregistro="1112029";

grafo.crearEnlaces("1112026", "1112013");

grafo.crearEnlaces("1112026", "1112027");

grafo.crearEnlaces("1112027", "1112028");

grafo.crearEnlaces("1112028", "1112029");

grafo.crearEnlaces("1112029", "1112030");

grafo.crearEnlaces("1112028", "1113046");
grafo.crearEnlaces("1111081", "1113046");

grafo.crearEnlaces("1113084", "1113085");
grafo.buscarNodo("1113085").setCorregistro(true);
grafo.buscarNodo("1113085").Corregistro="1113084";

grafo.crearEnlaces("1113084", "1113086");

grafo.crearEnlaces("1113085", "1113087");
grafo.crearEnlaces("1113086", "1113087");
grafo.buscarNodo("1113087").setCorregistro(true);
grafo.buscarNodo("1113087").Corregistro="1113086";

grafo.crearEnlaces("1112013", "1151038");
grafo.crearEnlaces("1112027", "1151038");

grafo.crearEnlaces("1151038", "1151039");

```

```
grafo.crearEnlaces("1112029", "1151039");
grafo.buscarNodo("1151039").setCorregistro(true);
grafo.buscarNodo("1151039").Corregistro="1112029";

grafo.crearEnlaces("1112029", "1153001");
```

```
//Tronco B sico Profesional*****
```

```
grafo.crearEnlaces("1111083", "1123052");
grafo.crearEnlaces("1111093", "1123052");

grafo.crearEnlaces("1123052", "1123053");
grafo.buscarNodo("1123053").setCorregistro(true);
grafo.buscarNodo("1123053").Corregistro="1123052";

grafo.crearEnlaces("1145054", "1133014");

grafo.crearEnlaces("1133014", "1133015");
grafo.buscarNodo("1133015").setCorregistro(true);
grafo.buscarNodo("1133015").Corregistro="1133014";

grafo.crearEnlaces("1133014", "1133019");

grafo.crearEnlaces("1133019", "1133020");
grafo.buscarNodo("1133020").setCorregistro(true);
grafo.buscarNodo("1133020").Corregistro="1133019";

grafo.crearEnlaces("1153001", "1133048");

grafo.crearEnlaces("1113086", "1145054");

grafo.crearEnlaces("1145054", "1145055");
grafo.buscarNodo("1145055").setCorregistro(true);
grafo.buscarNodo("1145055").Corregistro="1145054";

grafo.crearEnlaces("1151039", "1152001");

grafo.crearEnlaces("1152001", "1152002");
grafo.crearEnlaces("1153001", "1152002");

grafo.crearEnlaces("1153001", "1153005");
grafo.crearEnlaces("1154001", "1153005");
grafo.crearEnlaces("1153006", "1153005");

grafo.crearEnlaces("1200095", "1153006");

grafo.crearEnlaces("1153001", "1154016");
grafo.crearEnlaces("1151039", "1154016");

grafo.crearEnlaces("1154038", "1154031");

grafo.crearEnlaces("1154039", "1154032");
grafo.crearEnlaces("1154031", "1154032");

grafo.crearEnlaces("1154002", "1154038");

grafo.crearEnlaces("1154038", "1154039");

grafo.crearEnlaces("1154038", "1154040");
grafo.buscarNodo("1154040").setCorregistro(true);
grafo.buscarNodo("1154040").Corregistro="1154038";

grafo.crearEnlaces("1154039", "1154041");
grafo.buscarNodo("1154041").setCorregistro(true);
grafo.buscarNodo("1154041").Corregistro="1154039";
grafo.crearEnlaces("1154040", "1154041");

grafo.crearEnlaces("1153001", "1154042");
grafo.crearEnlaces("1154038", "1154042");

grafo.crearEnlaces("1154031", "1154043");
grafo.crearEnlaces("1154039", "1154043");

grafo.crearEnlaces("1154031", "1154044");

grafo.crearEnlaces("1154039", "1154045");

grafo.crearEnlaces("1112013", "1133061");

grafo.crearEnlaces("1154002", "1200095");
```

```

//Tronco Inter y Multidisciplinar*****/
grafo.crearEnlaces("1100037", "1100038");
grafo.crearEnlaces("1100040", "1100039");
grafo.crearEnlaces("1100039", "1100041");

//Tronco de IntegraciÃ³n*****/
grafo.crearEnlaces("1100039", "1100107");
grafo.crearEnlaces("1100107", "1100117");

//Tronco IntegraciÃ³n*****/
//Perfil Manufactura
grafo.crearEnlaces("1153001", "1133049");

grafo.crearEnlaces("1133049", "1133055");
grafo.buscarNodo("1133055").setCorregistro(true);
grafo.buscarNodo("1133055").Corregistro="1133049";

grafo.crearEnlaces("1154032", "1154037");

grafo.crearEnlaces("1154043", "1154047");

grafo.crearEnlaces("1154043", "1154049");

//Perfil GestÃ³n
grafo.crearEnlaces("1154042", "1154017");

grafo.crearEnlaces("1154002", "1154035");

grafo.crearEnlaces("1200095", "1154036");

grafo.crearEnlaces("1154043", "1154057");

/*****/
}

public static void Hijos(Nodo nodo)
{
    if(!nodo.getHijos().isEmpty())
    {
        for(int i=0; i<nodo.getHijos().size(); i++)
        {
            if(nodo.getHijos().get(i).visitado==false)
            {
                nodo.getHijos().get(i).visitado=true;
                Acumulado=Acumulado+1;
            }
            Hijos(nodo.getHijos().get(i));
        }
    }
}

public static void GenerarPrioridades()
{
    Nodo nodo;

    //Se crean las prioridades para cada UEA
    for(int i=0; i<grafo.TAM(); i++)
    {
        nodo=grafo.PosicionNodo(i);
        Hijos(nodo);
        nodo.Prioridad=Acumulado;
        Acumulado=0;
        grafo.PonerEnFalseVisitado();
    }
}

public static int Analisis_Informacion_Academica()
{
    /*****Extraccion del Resumen Trimestre*****/
    try{
        Document doc=Jsoup.parse(Informacion_Academica);

```

```

Element elemento=doc.getElementByld("tab4");
Elements td;
td=elemento.select("td");

ArrayList<String> Analisis_1=new ArrayList<String>();

for(Element TD : td)
{
    Analisis_1.add(TD.text());
}

ArrayList<String> Descripcion=new ArrayList<String>();
ArrayList<Integer> Creditos_Cubietos_OPT = new ArrayList<Integer>();
ArrayList<String> Basura = new ArrayList<String>();
int indice=0;

for(int i=0; i<Analisis_1.size(); i++)
{
    if(Analisis_1.get(i).contains("Nivel"))
        break;
    else{
        Basura.add(Analisis_1.get(i));
        i++;
        Basura.add(Analisis_1.get(i));
        i++;
        Basura.add(Analisis_1.get(i));
        i++;
        Basura.add(Analisis_1.get(i));
        i++;
        Descripcion.add(Analisis_1.get(i));
        i++;
        Basura.add(Analisis_1.get(i));
        i++;
        Basura.add(Analisis_1.get(i));
        i++;
        Creditos_Cubietos_OPT.add(Integer.parseInt(Analisis_1.get(i)));

        if(Descripcion.get(indice).contains("OPT. AREA DE CONCENTRACION - PRODUCCION Y MANUFACTURA"))
            Creditos_Produccion=Creditos_Cubietos_OPT.get(indice);
        if(Descripcion.get(indice).contains("OPT. AREA DE CONCENTRACION - GESTION INDUSTRIAL"))
            Creditos_Gestion=Creditos_Cubietos_OPT.get(indice);
        if(Descripcion.get(indice).contains("OPT. TUTORIALES")
            || Descripcion.get(indice).contains("OPT. DE MOVILIDAD")
            || Descripcion.get(indice).contains("OTRAS OPTATIVAS DE INTEGRACION") )
            Creditos_OPT_Integracion=Creditos_OPT_Integracion+Creditos_Cubietos_OPT.get(indice);

        indice++;
    }
}

elemento=doc.getElementByld("tab5");
td=elemento.select("td");
ArrayList<String> Analisis_2=new ArrayList<String>();

for(Element TD : td)
{
    // System.out.println(TD.text());
    Analisis_2.add(TD.text());
}

ArrayList<String> Trimestre=new ArrayList<String>();
ArrayList<Integer> Renuncias=new ArrayList<Integer>();
ArrayList<Integer> mb=new ArrayList<Integer>();
ArrayList<Integer> b=new ArrayList<Integer>();
ArrayList<Integer> s=new ArrayList<Integer>();
ArrayList<Integer> na=new ArrayList<Integer>();
ArrayList<Integer> Creditos_Acumulados=new ArrayList<Integer>();
int Acumulado=0, indice2=0;

for(int i=0; i<Analisis_2.size(); i++)
{
    Trimestre.add(Analisis_2.get(i)); i++;
    Renuncias.add(Integer.parseInt(Analisis_2.get(i))); i++;

    Basura.add(Analisis_2.get(i)); i++;
    Basura.add(Analisis_2.get(i)); i++;
    Basura.add(Analisis_2.get(i)); i++;
    Basura.add(Analisis_2.get(i)); i++;
}

```

```

mb.add(Integer.parseInt(Analisis_2.get(i))); i++;
b.add(Integer.parseInt(Analisis_2.get(i))); i++;
s.add(Integer.parseInt(Analisis_2.get(i))); i++;
na.add(Integer.parseInt(Analisis_2.get(i))); i++;

Basura.add(Analisis_2.get(i)); i++;

Creditos_Acumulados.add(Integer.parseInt(Analisis_2.get(i)));

Creditos[indice2]=Creditos_Acumulados.get(indice2)-Acumulado;

Acumulado=Creditos_Acumulados.get(indice2);

indice2++;
}

MB=mb.get(indice2-1);
B=b.get(indice2-1);
S=s.get(indice2-1);
NA=na.get(indice2-1);
Numero_de_Renuncias=Renuncias.get(indice2-1);
Trimestres=indice2-1;
Creditos_Inscritos_el_trimestre_Anterior=Creditos[indice2-1];

//Obtencion del Maximo Viable
if((MB+B+S)==0 || NA>=1)
    Maximo_Viable=40;
else{
    if(Numero_de_Renuncias==0 && (Creditos_Inscritos_el_trimestre_Anterior)>=20
    &&(((MB*10+B*8+S*6)/(MB+B+S))>=8)
        Maximo_Viable=63;
    else
        Maximo_Viable=40;
}

Informacion_Academica="";
return Creditos_Acumulados.get(indice2-1);

}catch(Exception error)
{
    System.out.println("Error en Analisis de InformaciÃ³n");
    return -1;
}

}

}

public static void Analisis_Kardex()
{
    try{
        Document doc;
        doc=Jsoup.parse(Kardex);

        Element elemento=doc.getElementById("t1");
        Elements td;
        td=elemento.select("td");

        ArrayList<String> CadenasTD=new ArrayList<String>();

        for(Element TD : td)
        {
            CadenasTD.add(TD.text());
        }

        ArrayList<String> UEA = new ArrayList<String>();
        ArrayList<String> Calificacion = new ArrayList<String>();
        ArrayList<String> Basura=new ArrayList<String>();
        ArrayList<String> Nombre=new ArrayList<String>();

        int indice=0;

        for(int i=0; i<CadenasTD.size(); i++)
        {
            Basura.add(CadenasTD.get(i));
            i++;

            UEA.add(CadenasTD.get(i));
            i++;

            Nombre.add(CadenasTD.get(i));

```

```

        i++;

        Basura.add(CadenasTD.get(i));
        i++;
        Basura.add(CadenasTD.get(i));
        i++;

        Calificacion.add(CadenasTD.get(i));

        grafo.Modificar_Grafo(UEA.get(indice), Calificacion.get(indice));

        i++;

        Basura.add(CadenasTD.get(i));

        i++;
        Basura.add(CadenasTD.get(i));

        indice++;
    }
    CadenasTD.clear();
    Kardex="";
} catch (Exception error)
{
    System.out.println("Error en Kardex");
}
}

public static void Recomendacion(int Creditos_Aprobados, int Tipo)
{
    int Pronostico;
    int a;
    int b;
    int x2_suma=0;
    int y_promedio;
    int x_promedio;
    int x_suma=0;
    int y_suma=0;
    int xy_suma=0;

    //Calculo del mínimo de créditos sugeridos
    if(Trimestres<12 && ((496-Creditos_Aprobados)/(12-Trimestres))<=Maximo_Viable)
        Minimo_Recomendado=((496-Creditos_Aprobados)/(12-Trimestres)); // Para terminar en 12 trimestres
    else if(Trimestres<18 && ((496-Creditos_Aprobados)/(18-Trimestres))<=Maximo_Viable)
        Minimo_Recomendado=((496-Creditos_Aprobados)/(18-Trimestres)); // Para terminar en 18 trimestres
    else if(Trimestres<30 && ((496-Creditos_Aprobados)/(30-Trimestres))<=Maximo_Viable)
        Minimo_Recomendado=((496-Creditos_Aprobados)/(30-Trimestres)); // Para terminar en 30 trimestres

    //Calculo del máximo de créditos sugeridos utilizando la fórmula del Pronostico
    for(int i=1; i<=Trimestres;i++){

        xy_suma=xy_suma+(i*Creditos[i-1]);
        x_suma=x_suma+i;
        y_suma=y_suma+Creditos[i-1];
        x2_suma=x2_suma+(i*i);
    }

    y_promedio=y_suma/Trimestres;
    x_promedio=x_suma/Trimestres;

    b=(xy_suma-(y_promedio*x_suma))/(x2_suma-(x_promedio*x_suma));
    a=y_promedio-(b*x_promedio);
    Pronostico=a+(b*(Trimestres+1));

    Maximo_Recomendado=Pronostico;

    if(Minimo_Recomendado>Maximo_Recomendado)
        Maximo_Recomendado=Minimo_Recomendado;

    if(Maximo_Recomendado>Maximo_Viable)
        Maximo_Recomendado=Maximo_Viable;

    grafo.Ordenar();

    String Clave_AUX;

```

```

String UEA_AUX;
int Creditos_AUX;
String TIPO_AUX;

Clave.clear();
Nombre_UEA.clear();
creditos.clear();
Clave_OPTI.clear();
Nombre_UEAOPTI.clear();
Creditos_OPTI.clear();
Tipo_OPTI.clear();
ClaveGestion.clear();
UEA_Gestion.clear();
Creditos_OGestion.clear();
ClaveProduccion.clear();
UEA_Produccion.clear();
Creditos_OProduccion.clear();

for(int i=0;i<grafo.TAM(); i++){

    if((grafo.PosicionNodo(i).getTipo()=="OPT-INTEGRACION"
        || grafo.PosicionNodo(i).getTipo()=="OPT-INTEGRACION-PRODUCCION"
        || grafo.PosicionNodo(i).getTipo()=="OPT-INTEGRACION-GESTION")
        && grafo.PosicionNodo(i).Aceditada==true)
        {
            Clave_OPTI.add(grafo.PosicionNodo(i).getClave());
            Nombre_UEAOPTI.add(grafo.PosicionNodo(i).getUEA());
            Creditos_OPTI.add(grafo.PosicionNodo(i).getCreditos());
            Tipo_OPTI.add(grafo.PosicionNodo(i).getTipo());
        }

    if(grafo.PosicionNodo(i).getTipo()=="OPT-INTEGRACION-GESTION" && grafo.PosicionNodo(i).Aceditada==false)
    {
        ClaveGestion.add(grafo.PosicionNodo(i).getClave());
        UEA_Gestion.add(grafo.PosicionNodo(i).getUEA());
        Creditos_OGestion.add(grafo.PosicionNodo(i).getCreditos());
    }
    if(grafo.PosicionNodo(i).getTipo()=="OPT-INTEGRACION-PRODUCCION" && grafo.PosicionNodo(i).Aceditada==false)
    {
        ClaveProduccion.add(grafo.PosicionNodo(i).getClave());
        UEA_Produccion.add(grafo.PosicionNodo(i).getUEA());
        Creditos_OProduccion.add(grafo.PosicionNodo(i).getCreditos());
    }
}

if(grafo.PosicionNodo(i).getUEA_Habilitada()==true && grafo.PosicionNodo(i).Aceditada!=true)
{
    if(Tipo==1 && grafo.PosicionNodo(i).getTipo()=="OPT-INTEGRACION-PRODUCCION")
        continue;
    if(Tipo==2 && grafo.PosicionNodo(i).getTipo()=="OPT-INTEGRACION-GESTION")
        continue;

    Clave.add(grafo.PosicionNodo(i).getClave());
    Nombre_UEA.add(grafo.PosicionNodo(i).getUEA());
    creditos.add(grafo.PosicionNodo(i).getCreditos());
}
}

}

@Override
protected void onCreate(Bundle savedInstanceState) {
    // TODO Auto-generated method stub
    super.onCreate(savedInstanceState);
    setContentView(R.layout.procesar);

    rs = this.getResources();
    tabla = (TableLayout)findViewById(R.id.tabla);
    cabecera = (TableLayout)findViewById(R.id.cabecera);
    cabecera2 = (TableLayout)findViewById(R.id.cabecera2);
    BotonOPTI=(TableLayout)findViewById(R.id.idBotonOPTI);

    layoutFila2 = new TableRow.LayoutParams(TableRow.LayoutParams.WRAP_CONTENT);

    layoutFila = new TableRow.LayoutParams(TableRow.LayoutParams.WRAP_CONTENT,
    TableRow.LayoutParams.WRAP_CONTENT,
    TableRow.LayoutParams.WRAP_CONTENT);
}

```

```

layoutFila3 = new TableRow.LayoutParams(TableRow.LayoutParams.WRAP_CONTENT,
    TableRow.LayoutParams.WRAP_CONTENT);

layoutGestion=new TableRow.LayoutParams(115,TableRow.LayoutParams.WRAP_CONTENT,
    TableRow.LayoutParams.WRAP_CONTENT);

layoutProduccion=new TableRow.LayoutParams(115,TableRow.LayoutParams.WRAP_CONTENT,
    TableRow.LayoutParams.WRAP_CONTENT);

layoutTitulo= new TableRow.LayoutParams(230,TableRow.LayoutParams.FILL_PARENT);
layoutClave = new TableRow.LayoutParams(60,TableRow.LayoutParams.FILL_PARENT);
layoutUEA = new TableRow.LayoutParams(110,TableRow.LayoutParams.FILL_PARENT);
layoutCreditos = new TableRow.LayoutParams(60,TableRow.LayoutParams.FILL_PARENT);

pDialog = new ProgressDialog(Procesar.this);
    pDialog.setProgressStyle(ProgressDialog.STYLE_SPINNER);
    pDialog.setMessage("Procesando...");

Bundle bundle=getIntent().getExtras();

Gestion = new Intent(this, Gestion.class);
Produccion = new Intent(this, Produccion.class);

    Informacion_Academica=bundle.getString("Informacion_Academica");
    Kardex=bundle.getString("Kardex");

    //Se hace un llamado al método llenandoGrafo() de la clase Grafo() para crear el grafo y sus enlaces
    llenandoGrafo();

    //Función que genera las prioridades de las UEA por trimestre
    GenerarPrioridades();

    //Ejecución de AsyncTask para trabajar en Segundo Plano
    try{
        new ProcesarRecomendacion().execute();
    }
    catch(Exception e){
        Toast.makeText(Procesar.this, "Se interrumpió el proceso", Toast.LENGTH_SHORT).show();
    }
}

public class ProcesarRecomendacion extends AsyncTask<Void, Integer, Boolean>{

    @Override
    protected Boolean doInBackground(Void... arg0){

        //int Creditos_OBL=0, Creditos_OPT_Integracion=0, Creditos_OPT_Inter=0;

        Creditos_Acumulados=Análisis_Informacion_Academica();

        grafo.Modificar_Grafo_Respecto_a_Creditos_Acumulados(Creditos_Acumulados);

        //Se extrae la información del Kardex y se modifica el grafo
        Análisis_Kardex();

        //Modificamos las UEA Optativas de Integración

        //grafo.setOptIntegracion(Creditos_Produccion, Creditos_Gestion, Creditos_OPT_Integracion);

        Recomendacion(Creditos_Acumulados, 0);
        System.out.println(Maximo_Recomendado);

        return true;
    }

    @Override
    protected void onPreExecute() {

        pDialog.show();
    }

    @Override
    protected void onPostExecute(Boolean result){

        if(result)
        {

            TextView Bienvenido;
            TextView OPT_Gestion;
            TextView OPT_Produccion;

```

```

        Bienvenido=(TextView)findViewById(R.id.idCargando);
        OPT_Gestion=(TextView)findViewById(R.id.idOPT_Gestion);
        OPT_Produccion=(TextView)findViewById(R.id.idOPT_Produccion);

        agregarCabeceraPrincipal();
        agregarCabecera();
        agregarFilasTabla();
        agregarBoton();

        Bienvenido.setText("Cuenta: " + MainActivity.Matricula);
        Bienvenido.setGravity(Gravity.RIGHT);
        Bienvenido.setTextColor(Color.BLACK);

        OPT_Gestion.setText("**UEA acreditada para obtener el certificado de Gestión");
        OPT_Gestion.setTextColor(Color.rgb(102, 205, 0));

        OPT_Produccion.setText("**UEA acreditada para obtener el certificado de Producción");
        OPT_Produccion.setTextColor(Color.rgb(255, 69, 00));

        pDialog.dismiss();
        //Toast.makeText(Procesar.this, "Tarea finalizada!", Toast.LENGTH_SHORT).show();
    }

}

}

public void agregarCabeceraPrincipal(){
    TableRow fila;

    fila = new TableRow(this);
    fila.setLayoutParams(layoutFila2);

    TextView Titulo;
    Titulo = new TextView(this);

    Titulo.setText("OPTATIVAS DE INTEGRACION ACREDITAS");
    Titulo.setGravity(Gravity.CENTER_HORIZONTAL);
    Titulo.setTextAppearance(this,R.style.etiqueta);
    Titulo.setBackgroundResource(R.drawable.tabla_celda_cabecera);
    Titulo.setLayoutParams(layoutTitulo);

    fila.addView(Titulo);

    cabecera2.addView(fila);
}
public void agregarCabecera(){
    TableRow fila;
    TextView clave;
    TextView UEA;
    TextView Creditos;

    fila = new TableRow(this);
    fila.setLayoutParams(layoutFila);

    clave = new TextView(this);
    UEA = new TextView(this);
    Creditos = new TextView(this);

    clave.setText(rs.getString(R.string.clave));
    clave.setGravity(Gravity.CENTER_HORIZONTAL);
    clave.setTextAppearance(this,R.style.etiqueta);
    clave.setBackgroundResource(R.drawable.tabla_celda_cabecera);
    clave.setLayoutParams(layoutClave);

    UEA.setText(rs.getString(R.string.UEA));
    UEA.setGravity(Gravity.CENTER_HORIZONTAL);
    UEA.setTextAppearance(this,R.style.etiqueta);
    UEA.setBackgroundResource(R.drawable.tabla_celda_cabecera);
    UEA.setLayoutParams(layoutUEA);

    Creditos.setText(rs.getString(R.string.Creditos));
    Creditos.setGravity(Gravity.CENTER_HORIZONTAL);
    Creditos.setTextAppearance(this,R.style.etiqueta);
    Creditos.setBackgroundResource(R.drawable.tabla_celda_cabecera);
    Creditos.setLayoutParams(layoutCreditos);

    fila.addView(clave);
    fila.addView(UEA);
    fila.addView(Creditos);
}

```

```

        cabecera.addView(fila);
    }

    public void agregarFilasTabla(){
        TableRow fila;
        TextView clave;
        TextView UEA;
        TextView Creditos;

        for(int i = 0;i<Clave_OPTI.size();i++){
            fila = new TableRow(this);
            fila.setLayoutParams(layoutFila);

            clave = new TextView(this);
            UEA = new TextView(this);
            Creditos = new TextView(this);

            //UEA INTEGRACION-GESTION VERDE
            if(Tipo_OPTI.get(i)=="OPT-INTEGRACION-GESTION"){
                clave.setTextAppearance(this,R.style.verde);
                UEA.setTextAppearance(this,R.style.verde);
                Creditos.setTextAppearance(this,R.style.verde);
            }
            else if(Tipo_OPTI.get(i)=="OPT-INTEGRACION-PRODUCCION"){
                clave.setTextAppearance(this,R.style.naranja);
                UEA.setTextAppearance(this,R.style.naranja);
                Creditos.setTextAppearance(this,R.style.naranja);
            }

            clave.setText(Clave_OPTI.get(i));
            clave.setGravity(Gravity.CENTER_HORIZONTAL);
            clave.setBackgroundResource(R.drawable.tabla_celda);
            clave.setLayoutParams(layoutClave);

            UEA.setText(Nombre_UEAOPTI.get(i));
            UEA.setGravity(Gravity.CENTER_HORIZONTAL);
            UEA.setBackgroundResource(R.drawable.tabla_celda);
            UEA.setLayoutParams(layoutUEA);

            Creditos.setText(Integer.toString(Creditos_OPTI.get(i)));
            Creditos.setGravity(Gravity.CENTER_HORIZONTAL);
            Creditos.setBackgroundResource(R.drawable.tabla_celda);
            Creditos.setLayoutParams(layoutCreditos);

            fila.addView(clave);
            fila.addView(UEA);

            fila.addView(Creditos);

            tabla.addView(fila);
        }

        Clave_OPTI.clear();
        Nombre_UEAOPTI.clear();
        Creditos_OPTI.clear();
    }

    public void agregarBoton(){
        TableRow fila;
        fila = new TableRow(this);
        fila.setLayoutParams(layoutFila3);

        Button gestion = new Button(this);
        gestion.setText("Gestión");
        gestion.setTextAppearance(this, R.style.boton);
        gestion.setLayoutParams(layoutGestion);
        gestion.setBackgroundResource(R.drawable.color_boton2);

        gestion.setOnClickListener(new Button.OnClickListener(){
            public void onClick(View arg0){

                startActivity(Gestion);
            }
        });

        Button produccion= new Button(this);
        produccion.setText("Producción");
    }

```

```

        produccion.setTextAppearance(this, R.style.boton);
        produccion.setLayoutParams(layoutProduccion);
        produccion.setBackgroundResource(R.drawable.color_boton3);

        produccion.setOnClickListener(new Button.OnClickListener(){
public void onClick(View arg0){

            startActivity(Produccion);

        }
});

        fila.addView(gestion);
        fila.addView(produccion);
        BotonOPT1.addView(fila);
    }

}

```

## Clase Recomendación

```

package com.example.sistema_de_recomendacion;

import android.app.Activity;
import android.content.Intent;
import android.content.res.Resources;
import android.graphics.Color;
import android.os.Bundle;
import android.view.Gravity;
import android.view.View;
import android.widget.TableLayout;
import android.widget.TableRow;
import android.widget.TextView;

public class Recomendacion extends Activity {

    TableLayout tabla;
    TableLayout cabecera2;
    TableLayout cabecera;
    TableRow.LayoutParams layoutFila;
    TableRow.LayoutParams layoutFila2;
    TableRow.LayoutParams layoutTitulo;
    TableRow.LayoutParams layoutClave;
    TableRow.LayoutParams layoutUEA;
    TableRow.LayoutParams layoutCreditos;

    Resources rs;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.recomendacion);

        TextView UEA;

        UEA=(TextView)findViewById(R.id.idUEA);
        TextView Bienvenido=(TextView)findViewById(R.id.idBienvenido);

        rs = this.getResources();
        tabla = (TableLayout)findViewById(R.id.tabla);
        cabecera = (TableLayout)findViewById(R.id.cabecera);
        layoutFila2 = new TableRow.LayoutParams(TableRow.LayoutParams.WRAP_CONTENT);
        layoutFila = new TableRow.LayoutParams(TableRow.LayoutParams.WRAP_CONTENT,
            TableRow.LayoutParams.WRAP_CONTENT,
            TableRow.LayoutParams.WRAP_CONTENT);
        layoutTitulo= new TableRow.LayoutParams(230,TableRow.LayoutParams.FILL_PARENT);
        layoutClave = new TableRow.LayoutParams(60,TableRow.LayoutParams.FILL_PARENT);
        layoutUEA = new TableRow.LayoutParams(110,TableRow.LayoutParams.FILL_PARENT);
        layoutCreditos = new TableRow.LayoutParams(60,TableRow.LayoutParams.FILL_PARENT);

        agregarCabecera();
        agregarFilasTabla();

        Bienvenido.setText("Cuenta: " + MainActivity.Matricula);
        Bienvenido.setGravity(Gravity.RIGHT);
    }
}

```

```

        Bienvenido.setTextColor(Color.BLACK);

        UEA.setText("Recomendación de UEA a incibir");
    }

    public void agregarCabecera(){
        TableRow fila;
        TextView clave;
        TextView UEA;
        TextView Creditos;

        fila = new TableRow(this);
        fila.setLayoutParams(layoutFila);

        clave = new TextView(this);
        UEA = new TextView(this);
        Creditos = new TextView(this);

        clave.setText(rs.getString(R.string.clave));
        clave.setGravity(Gravity.CENTER_HORIZONTAL);
        clave.setTextAppearance(this, R.style.etiqueta);
        clave.setBackgroundResource(R.drawable.tabla_celda_cabecera);
        clave.setLayoutParams(layoutClave);

        UEA.setText(rs.getString(R.string.UEA));
        UEA.setGravity(Gravity.CENTER_HORIZONTAL);
        UEA.setTextAppearance(this, R.style.etiqueta);
        UEA.setBackgroundResource(R.drawable.tabla_celda_cabecera);
        UEA.setLayoutParams(layoutUEA);

        Creditos.setText(rs.getString(R.string.Creditos));
        Creditos.setGravity(Gravity.CENTER_HORIZONTAL);
        Creditos.setTextAppearance(this, R.style.etiqueta);
        Creditos.setBackgroundResource(R.drawable.tabla_celda_cabecera);
        Creditos.setLayoutParams(layoutCreditos);

        fila.addView(clave);
        fila.addView(UEA);
        fila.addView(Creditos);

        cabecera.addView(fila);
    }

    public void agregarFilasTabla(){
        TableRow fila;
        TextView clave;
        TextView UEA;
        TextView Creditos;
        int Creditos_Viables=0;
        String Corregistro;
        for(int i = 0; i<Procesar.Clave.size(); i++){

            if(Procesar.grafo.buscarNodo(Procesar.Clave.get(i)).getCorregistro()==true)
            {
                Corregistro=Procesar.grafo.buscarNodo(Procesar.Clave.get(i)).Corregistro;

                if(Procesar.grafo.buscarNodo(Corregistro).Aceditada==false)
                {
                    if(Procesar.grafo.buscarNodo(Corregistro).getOK()==false)
                        continue;
                }
            }

            Creditos_Viables=Creditos_Viables+Procesar.creditos.get(i);

        }

        if(Creditos_Viables>Procesar.Maximo_Recomendado){
            Creditos_Viables=Creditos_Viables-Procesar.creditos.get(i);
            Procesar.grafo.buscarNodo(Procesar.Clave.get(i)).setOK(false);
        }
        else{
            Procesar.grafo.buscarNodo(Procesar.Clave.get(i)).setOK(true);
            fila = new TableRow(this);
            fila.setLayoutParams(layoutFila);

            clave = new TextView(this);

```

```

        UEA = new TextView(this);
        Creditos = new TextView(this);

//UEA INTEGRACION-GESTION VERDE
        if(Procesar.grafo.buscarNodo(Procesar.Clave.get(i)).getTipo()=="OPT-INTEGRACION-GESTION"){
            clave.setTextAppearance(this,R.style.verde);
            UEA.setTextAppearance(this,R.style.verde);
            Creditos.setTextAppearance(this,R.style.verde);
        }
        if(Procesar.grafo.buscarNodo(Procesar.Clave.get(i)).getTipo()=="OPT-INTEGRACION-PRODUCCION"){
            clave.setTextAppearance(this,R.style.naranja);
            UEA.setTextAppearance(this,R.style.naranja);
            Creditos.setTextAppearance(this,R.style.naranja);
        }
        clave.setText(Procesar.Clave.get(i));
        clave.setGravity(Gravity.CENTER_HORIZONTAL);
        clave.setBackgroundResource(R.drawable.tabla_celda);
        clave.setLayoutParams(layoutClave);

        UEA.setText(Procesar.Nombre_UEA.get(i));
        UEA.setGravity(Gravity.CENTER_HORIZONTAL);
        UEA.setBackgroundResource(R.drawable.tabla_celda);
        UEA.setLayoutParams(layoutUEA);

        Creditos.setText(Integer.toString(Procesar.creditos.get(i)));
        Creditos.setGravity(Gravity.CENTER_HORIZONTAL);
        Creditos.setBackgroundResource(R.drawable.tabla_celda);
        Creditos.setLayoutParams(layoutCreditos);

        fila.addView(clave);
        fila.addView(UEA);

        fila.addView(Creditos);

        tabla.addView(fila);
    }
}

public void Modificar(View e)
{
    Intent Continuar = new Intent(this, Modificar.class);

    startActivity(Continuar);
}

public void Finalizar(View e)
{
    Intent Finalizar = new Intent(this, Finalizar.class);

    startActivity(Finalizar);
}
}

```

## **Bibliografía**

Ceballos Sierra, F.J, 2006, Java 2 Curso de programación, Alfaomega.

Weiss, M.A, 2000, Estructura de datos en Java, Pearson.

<http://darkbyteblog.wordpress.com/2011/03/31/java-arreglos-ordenamiento-de-objetos-algoritmo-de-insercion-directa>, Java-Arreglos-Ordenamiento de objetos, consultado el 12 de agosto de 2013.

<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=listas>, Ordenar Listas en Java, consultado el 15 de agosto de 2013.

[http://www.sgoliver.net/blog/?page\\_id=3011](http://www.sgoliver.net/blog/?page_id=3011), Curso Programación Android, consultado del 26 de agosto al 30 de septiembre de 2013.

<http://android.alorse.net/2012/11/27/como-crear-actividades-activity-y-pasar-de-una-a-la-otra-en-android-ii/>, Creación actividades en Android, consultada el 27 de agosto de 2013.

<http://programacion-anexo4.blogspot.mx/2011/09/practicar-android-enviando-un-email-una.html>, Envío de e-mail en Android, consultada el 8 de octubre de 2013.

<http://nosoandroid.blogspot.mx/search/label/boton>, Creación de botón dinámico en Android, consultada el 17 de octubre.