

Universidad Autónoma Metropolitana Unidad Azcapotzalco
División de Ciencias Básicas e Ingeniería
Licenciatura en Ingeniería en Computación

Reporte de proyecto de integración
Recomendación de inscripción para Ingeniería en Computación

Daniel Alberto Torres González
209329877

Trimestre 2013 Otoño
14 de noviembre de 2013

Asesores:

Francisco Javier Zaragoza Martínez
Profesor Titular
Departamento de Sistemas

Rodrigo Alexander Castro Campos
Profesor Asociado
Departamento de Sistemas

Tabla de Contenido

Resumen	
1. Objetivos	4
1.1 Objetivo general	4
1.2 Objetivos específicos	4
2. Introducción	5
3. Descripción técnica	6
3.1 Módulo de ingreso	6
3.2 Módulo de solicitud	6
3.3 Módulo de análisis	6
3.4 Módulo de recomendación	7
3.5 Módulo de presentación	7
4. Especificación técnica	9
4.1 Módulo de ingreso	9
4.2 Módulo de solicitud	9
4.3 Módulo de análisis	10
4.4 Módulo de recomendación	10
4.5 Módulo de presentación	11
5. Desarrollo	12
5.1 Módulo de ingreso	14
5.2 Módulo de solicitud	17
5.3 Módulo de análisis	18
5.4 Módulo de recomendación	20
5.5 Módulo de presentación	21
6. Conclusiones	26
7. Referencias	27
8. Apéndices	28

Resumen

En el presente reporte se presentan los detalles del proyecto terminal "Recomendación de inscripción para Ingeniería en Computación" de la Universidad Autónoma Metropolitana unidad Azcapotzalco. Este proyecto terminal fue desarrollado con la idea de ayudar a los alumnos de la carrera a seleccionar las UEA más convenientes a inscribir en su próximo trimestre, tomando el plan de estudios vigente. Esta selección de UEA puede resultar no muy benéfica para el alumno si no se toman en cuenta varios aspectos, como pueden ser el número de créditos que se otorgan en caso de acreditar la UEA, si algunas UEA más avanzadas solo pueden ser inscritas al acreditar una o más UEA, etc. Si se toman en cuenta estos aspectos, se puede tener como resultado un mejor desarrollo escolar y se puede reducir el tiempo de estancia en la Universidad. Se detalla el diseño y la implementación de los cinco módulos de que consta la aplicación.

1. Objetivos

1.1 Objetivo general

Implementar una aplicación para dispositivo móvil que le pueda recomendar a un alumno de Ingeniería en Computación las UEA que puede inscribir en su próximo trimestre.

1.2 Objetivos específicos

- I. Desarrollar una aplicación capaz de establecer una conexión entre el servidor que contiene el historial académico del alumno y el dispositivo.
- II. Diseñar e implementar un analizador sintáctico que recupere y almacene el historial académico del alumno.
- III. Diseñar y programar un algoritmo que seleccione las UEA más recomendadas a inscribir en el próximo trimestre.
- IV. Construir una interfaz gráfica intuitiva para facilitar la interacción entre la aplicación y el alumno.

2. Introducción

Actualmente los alumnos de Ingeniería en Computación deben elegir las Unidades de Enseñanza Aprendizaje (UEA) que pretenden cursar en su próximo trimestre, ya sean obligatorias u optativas y a veces resulta difícil decidir cuáles son las más adecuadas para inscribir y avanzar adecuadamente en la carrera.

Comúnmente los alumnos eligen las UEA solicitando información sobre los profesores que imparten las materias con compañeros de carrera más antiguos, consultando los horarios disponibles o simplemente las inscriben porque su seriación lo permite y omiten otros factores que pueden llegar a afectar su estadía en la universidad, como es la seriación de materias, el número de créditos recomendables a cursar, etc. Todos estos factores deben ser tomados en cuenta al momento de realizar la reinscripción, para que el alumno vaya planificando un desarrollo óptimo a lo largo de su carrera y procure terminar en un tiempo adecuado.

Hasta este momento, los alumnos no contaban con una herramienta que les ayudará a tomar esta decisión, por lo cual se decidió desarrollar un software que los oriente a lo largo de su estancia en la Universidad.

En este reporte de proyecto terminal se muestra como se desarrolló dicha herramienta para dispositivos móviles con sistema Android 4.0 o superior [1], la cual es capaz de sugerir al alumno las materias que puede cursar en su próximo trimestre, esto para procurar reducir el tiempo en que el alumno finaliza la carrera.

3. Descripción Técnica

3.1 Módulo de ingreso

Solicita al alumno su matrícula y contraseña proporcionadas al Sistema de Administración Escolar, además de solicitarle seleccionar el área de concentración que pretende cursar y un servidor al cual se conectará el dispositivo, para poder establecer una conexión válida entre ambos. En caso de que los datos sean válidos, se ejecuta el siguiente módulo. En caso de error, ya sea que el alumno no proporcionó los datos correctos o si se llega a presentar algún fallo al realizar la conexión con el servidor, se despliega un aviso con dicho error en la pantalla actual y este mismo módulo es ejecutado.

3.2 Módulo de solicitud

Ya teniendo una conexión válida, se solicita al servidor seleccionado previamente el historial académico completo del alumno para almacenar los campos que contienen la clave de la UEA, el tipo de evaluación en la que se cursó la UEA, el trimestre en que se cursó la UEA, la calificación que se obtuvo y el número de créditos que se otorgan al acreditar la UEA. Un analizador sintáctico es el encargado de transformar el historial académico del alumno en dicha estructura de datos. Además se solicita información académica sobre renunciaciones a UEA en el último trimestre cursado, la carrera a la que pertenece el alumno, las UEA que han sido autorizadas para cursar, el nombre completo del alumno, el número de créditos acumulados y el último trimestre con actividad académica. Al obtener toda la información se realiza el cierre de sesión. En caso de que el alumno no sea de Ingeniería en Computación se manda un error en la primer pantalla, se cierra la conexión y el módulo de ingreso se ejecuta nuevamente.

3.3 Módulo de análisis

Se carga una estructura de datos almacenada manualmente por el desarrollador, la cual contiene el plan de estudios vigente de acuerdo a la selección de área de concentración que realizó el alumno en el módulo de ingreso. Además se decide si se puede inscribir o no un exceso de créditos. Esto se logra con el dato almacenado sobre renunciaciones a UEA, con las calificaciones obtenidas en el último trimestre y con el promedio del trimestre anterior. En caso de haber renunciado o reprobado alguna UEA o no haber obtenido un promedio mínimo de B en el trimestre anterior, no se podrá inscribir un

exceso de créditos. Esto es, el número máximo de créditos que se podrá recomendar es de cuarenta. En caso contrario, el número máximo de créditos a recomendar será de sesenta y tres.

Finalmente se compara el historial académico obtenido del servidor seleccionado con el plan de estudios vigente para calcular una lista (ordenada de manera ascendente de acuerdo a una prioridad) de todas las UEA posibles que pueden ser inscritas en el siguiente trimestre.

3.4 Módulo de recomendación

Si es la primera vez que se ejecuta este módulo, se realiza el cálculo de un promedio móvil (últimos tres trimestres) para obtener un número mínimo y uno máximo de créditos que se recomendarán al alumno inscribir en su próximo trimestre, tomando en cuenta que dichos números no deben ser mayores al máximo calculado en el módulo de análisis.

Además se utiliza la lista de UEA posibles a inscribir calculada en el módulo anterior para calcular dos nuevas listas de UEA, una de UEA obligatorias y otra de optativas. Esta selección se realiza tomando en cuenta las prioridades de las UEA, asignadas previamente por el desarrollador. Si la prioridad de una UEA es mayor que otra, entonces la UEA con mayor prioridad será seleccionada siempre y cuando no se rebase el número máximo de créditos a recomendar.

Si no es la primera vez que se ejecuta este módulo, se omite el cálculo de créditos mínimos y máximos a recomendar.

3.5 Módulo de presentación

Aparece en la parte superior de la pantalla la matrícula, el nombre completo, el número de créditos acumulados, el número mínimo y máximo de créditos permitidos y el número mínimo y máximo de créditos recomendados a inscribir, seguido de una tabla que contendrá la lista final de UEA obligatorias calculada. Los datos que se incluyen en la tabla son la clave, el nombre y el número de créditos de las UEA. Posterior a esto, con un algoritmo se revisa si existe oportunidad de inscribir UEA optativas. En caso positivo, se despliegan las UEA optativas que pueden ser inscritas. Por otra parte, si la sugerencia no fue satisfactoria para el alumno, tendrá la opción de solicitar un nuevo cálculo cambiando los parámetros de créditos mínimos y créditos máximos a recomendar para después regresar al módulo de Recomendación. Además, se tendrá opción de realizar un nuevo cálculo para un alumno diferente, y finalmente se tendrá la opción de escribir un

comentario que ayudará a mejorar el desempeño de la aplicación.

La figura 1 muestra la interacción entre los cinco diferentes módulos:

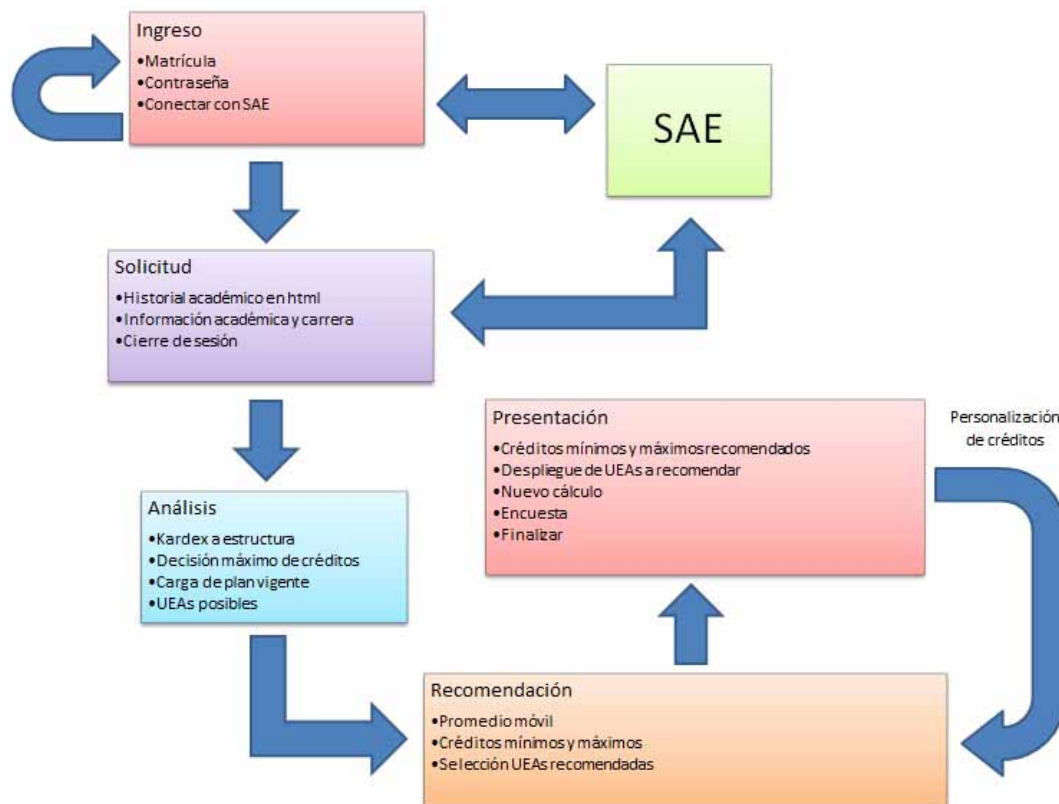


Figura 1: Interacción entre módulos

4. Especificación Técnica

La aplicación puede ser ejecutada en distintos dispositivos con sistema Android. Las versiones de Android que son soportadas van desde la versión *"Android 4.0-4.0.2 Ice Cream Sandwich (API nivel 14)"* hasta la última versión estable, *"Android 4.3 Jelly Bean (API nivel 18)"*. El tiempo en que se ejecuta la aplicación es de aproximadamente cuatro segundos. Se toma en cuenta que el servidor seleccionado de la Universidad Autónoma Metropolitana puede estar recibiendo muchas solicitudes, la conexión a internet del dispositivo puede no ser muy rápida, entre otras cosas.

4.1 Módulo de ingreso

Existen dos campos de texto en los cuales el alumno puede escribir su matrícula y password, además de dos menus desplegables. Uno sirve para seleccionar una de las cinco áreas de concentración de la carrera y otro para seleccionar el servidor al cual se solicita la información. Después se debe oprimir el botón que provoca que se inicialicen las variables que son utilizadas para almacenar la información que se solicita al servidor seleccionado concatenado con *".uam.mx:8443"*. Para esto se crea una nueva conexión con el servidor, se cifran los datos del alumno y mediante un método *"post"* se envían. En caso de error, ya sea que el alumno no ingreso datos en los campos de texto, ingreso datos incorrectos o no se puede establecer una conexión con el servidor, dicho error se desplegará en pantalla.

4.2 Módulo de solicitud

Mediante un método *"get"* se solicita toda la información académica requerida a través del directorio *"/sae/azc/IEWBC022.oConsulta"* alojado en el servidor seleccionado. Posteriormente, mediante otro método *"get"*, se solicita el historial académico del alumno a través del directorio *"/sae/azc/IEWBC020.oConsulta"*. Al terminar de leer todos los datos, se realiza el cierre de sesión. En caso de error, ya sea que la información académica no se pudo leer, no se estableció una sesión válida con el servidor o que el alumno no pertenece a la licenciatura de Ingeniería en Computación, dicho error se desplegará en pantalla.

4.3 Módulo de análisis

Se carga en memoria la estructura de datos que contiene todas las UEA del plan de estudios vigente, según el área de concentración seleccionada. Luego se recuperan los datos almacenados sobre renunciaciones a UEA, las calificaciones y el promedio obtenido en el trimestre anterior para decidir si se podrá inscribir o no un exceso de créditos. Después se comienza a calcular una lista de todas las UEA posibles a inscribir. Para esto, de la estructura de datos que contiene el plan de estudios vigente, se eliminan las UEA que se encuentran en el historial académico con calificación diferente de NA, las UEA que aún no pueden ser inscritas debido a la seriación o a cubrir un determinado número de créditos, las UEA que no han sido autorizadas a inscribir y las UEA que ya no se pueden cursar en evaluación global o recuperación. Al término de este cálculo se continúa con el siguiente módulo.

4.4 Módulo de recomendación

Se calcula el promedio móvil de los últimos 3 trimestres y se asigna como el número mínimo de créditos a recomendar. El número máximo de créditos a recomendar será la suma del número mínimo de créditos a recomendar más el número de créditos de una UEA, siempre y cuando esta suma no sea mayor al máximo permitido. Si la suma es mayor al máximo permitido, entonces el número máximo de créditos a recomendar será el máximo permitido.

Tomando en cuenta que el trimestre que se está cursando actualmente es el trimestre n , se tiene:

$$Min = \frac{creditos_{n-3} + creditos_{n-2} + creditos_{n-1}}{3}$$

$$Max = \frac{creditos_{n-3} + creditos_{n-2} + creditos_{n-1}}{3} + UEA$$

$$SA : 0 \leq Min < Max \leq max.permitido$$

Donde Min y Max son el número de créditos mínimo y máximo sugeridos, respectivamente, $creditos_{n-i}$ es el número de créditos obtenidos en el trimestre $n-i$ y UEA es el número de créditos de una UEA más a recomendar. Posteriormente se ejecuta un algoritmo que, tomando como entrada la lista ordenada de UEA posibles a inscribir calculada en el módulo de análisis, calcula dos listas, una de UEA obligatorias y otra de UEA optativas. El

algoritmo implementado revisa si una UEA es obligatoria u optativa. Al resultar ser obligatoria, la ingresa en la lista de UEA obligatorias a recomendar de acuerdo a su prioridad (mayor), siempre y cuando la suma total de créditos de las UEA ingresadas previamente no sea mayor que la cantidad de créditos máximos a recomendar. Si esto sucede, se deja de ingresar UEA a la lista de obligatorias. Si la UEA es optativa, la ingresa a la lista de UEA optativas sin realizar ninguna suma. Además, si la suma del número de créditos acumulados más el número de créditos a recomendar es mayor o igual a 491 (mínimo a cubrir para finalizar la carrera) se avisará al alumno en el módulo de presentación que puede terminar la carrera en el siguiente trimestre.

4.5 Módulo de presentación

Presenta en la parte superior de la pantalla la información del alumno especificada en la descripción técnica. Luego se crea dinámicamente la tabla de tres columnas por el número de UEA obligatorias que se recomendarán y se agregan las UEA. Posteriormente se verifica la lista de UEA optativas. En caso de haya una o más UEA optativas con un número de créditos tal que la suma total del número de créditos de la lista de UEA obligatorias más el número de créditos de la UEA optativa en cuestión sea menor o igual al número máximo de créditos a recomendar, se desplegará en pantalla, debajo de la lista final de UEA obligatorias, todas las UEA optativas que cumplan esta condición. Después se despliegan dos mensajes. El primer mensaje dice al alumno la cantidad de créditos que puede acumular si inscribe las UEA recomendadas y las acredita. El segundo mensaje dice al alumno si puede o no finalizar la carrera en el siguiente trimestre. Finalmente se presentan tres botones. El primer botón sirve al alumno para regresar al módulo de ingreso, esto en caso de que se desee realizar un nuevo cálculo con matrícula y password diferentes (alumno diferente). El segundo botón permite al alumno personalizar el número mínimo y máximo de créditos a recomendar y cambiar el área de concentración seleccionada previamente, esto con la intención de realizar un nuevo cálculo de UEA a recomendar en caso de que la sugerencia no sea atractiva para el alumno. El tercer botón permite al alumno realizar un comentario sobre el desempeño de la aplicación. El comentario escrito será enviado por correo electrónico al Coordinador de Ingeniería en Computación y al desarrollador de la aplicación con ayuda de algún cliente de correo electrónico que se encuentre instalado en el dispositivo.

5. Desarrollo

Las herramientas que fueron utilizadas para el desarrollo de la aplicación son las siguientes:

- I. Windows 8 para arquitectura de 64 bits [2]
- II. Java Development Kit (JDK) versión 7 actualización 21 [3]
- III. Eclipse Classic Software Development Kit (SDK) versión 4.2.2 [4]
- IV. Android SDK Manager revisión 21.1 [5]
- V. Complemento Android Development Tools en Eclipse con las versiones de API 14 a la 18 [6]
- VI. Emulador de dispositivos Android Virtual Device [7]

Dentro del IDE de desarrollo Eclipse se creó el nuevo proyecto Android, especificando en el archivo *AndroidManifest.xml* el paquete en el cual están contenidas las clases creadas (archivos *.java*), la versión, los permisos requeridos (acceso a internet), las versiones de Android mínima y máxima soportadas y las actividades *MainActivity*, *Recomendacion*, *Personalizacion* y *Comentar*, que son las actividades ejecutadas en la aplicación.

En el archivo *strings.xml*, el cual se crea por defecto en el proyecto, se incluyeron las cadenas de caracteres necesarias para ser desplegadas en las diferentes pantallas de la aplicación. En la sección 8 se muestra la forma en como fueron declaradas dichas cadenas.

Por otra parte, el ícono diseñado para identificar la aplicación al momento de instalarse en un dispositivo Android es el siguiente:



Figura 2: Ícono de la Recomendación de UEA de Ingeniería en Computación

Al ejecutarse el archivo con extensión *.apk* en el dispositivo, se instalará la aplicación para poder ser ejecutada por el alumno desde su menú de aplicaciones. La figura siguiente muestra la forma en que se verá:

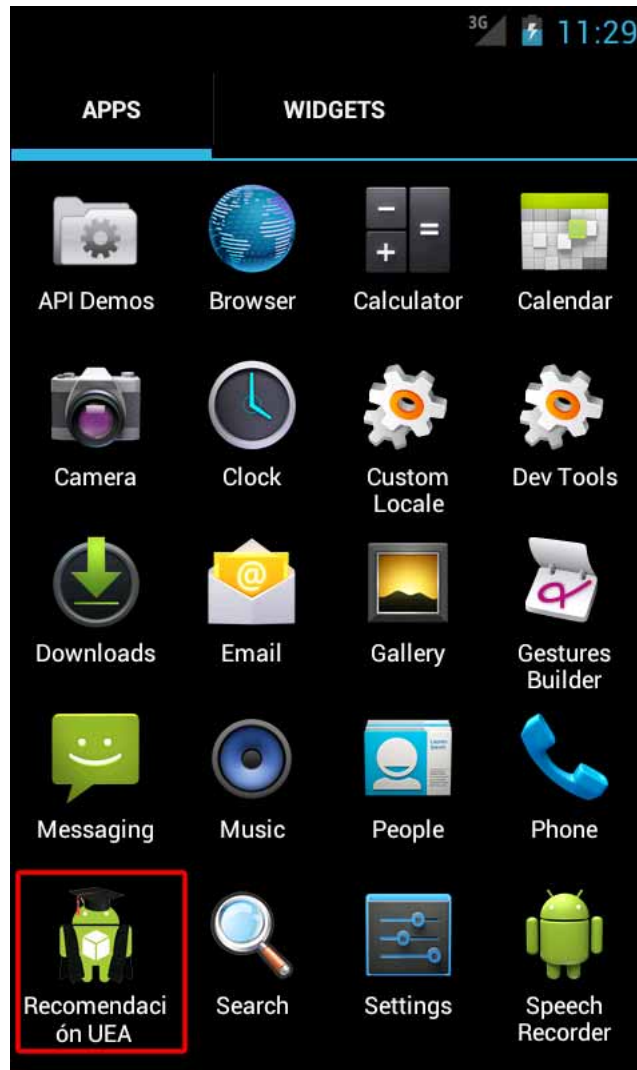


Figura 3: Dispositivo con la aplicación instalada

5.1 Módulo de ingreso

El diseño de la interfaz gráfica de la actividad principal se realizó en el archivo *activity_main.xml*, en el cual se agregaron seis componentes *TextView*, dos *EditText*, dos *Spinner* y un *Button*. La figura 4 muestra la interfaz gráfica del módulo de ingreso:

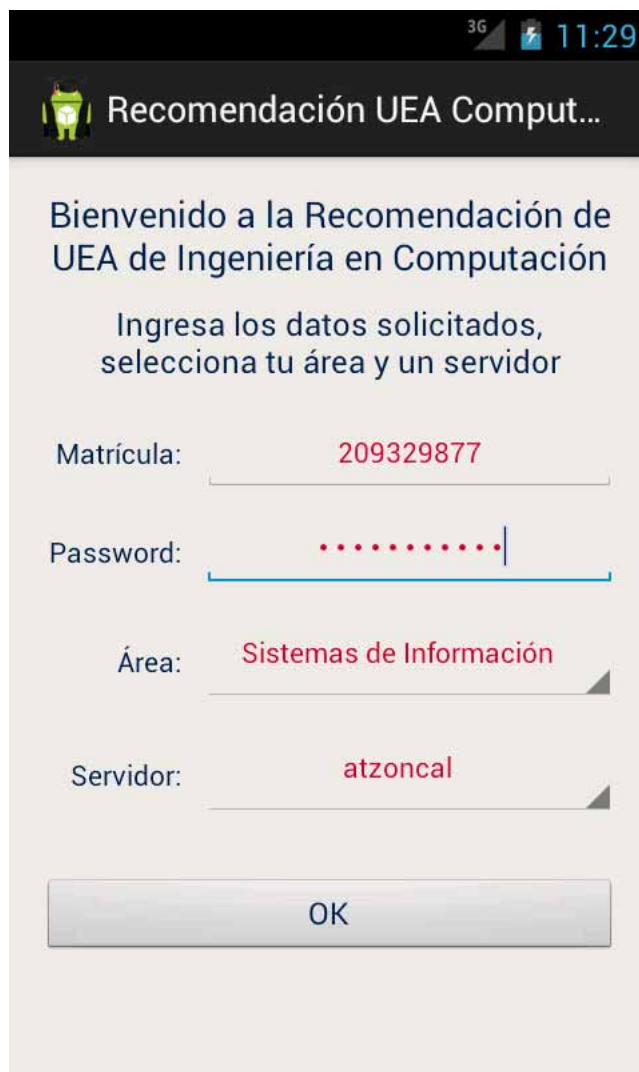


Figura 4: Módulo de Ingreso con información

Los componentes *TextView* sirven para visualizar cadenas de caracteres en la pantalla del dispositivo, con la finalidad de dar la bienvenida al alumno, proporcionarle instrucciones, indicarle que datos debe ingresar en los campos de texto existentes o que dato puede seleccionar de las listas desplegables. Los componentes *EditText* sirven para ingresar la matrícula y el password. En el campo de texto de la matrícula solo se pueden ingre-

sar números y en el de password números y caracteres. Los componentes *Spinner* son listas desplegables con un estilo único definido. Este estilo se definió en el archivo *spinner_main.xml*. La primer lista desplegable se utiliza para seleccionar una de las cinco áreas de concentración y la segunda para seleccionar el servidor del cual se extraerá la información del alumno. Al momento en que usuario selecciona la lista desplegable correspondiente al área de concentración, se le muestran las cinco diferentes áreas de la licenciatura de la siguiente forma:

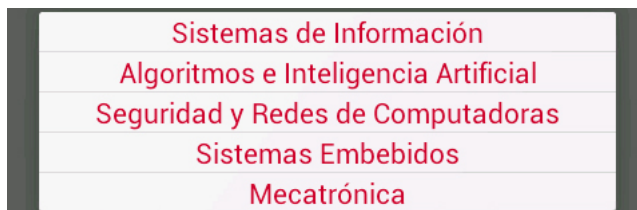


Figura 5: Lista desplegable de áreas de concentración

Teniendo las cinco áreas en pantalla, el alumno deberá seleccionar el área que pretende cursar a lo largo de su carrera. De igual manera, al seleccionar la lista correspondiente al servidor de la Universidad, se muestran los dos posibles servidores con los cuales se puede realizar una conexión y el alumno deberá seleccionar el de su preferencia:

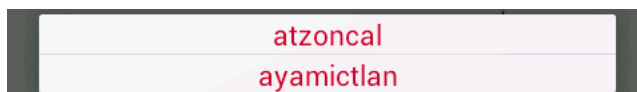


Figura 6: Lista desplegable de servidores

Por último el componente *Button* sirve para comenzar la ejecución del módulo de ingreso. Al oprimirse el botón se inicializan todos los datos estáticos de la clase *Datos*, definida dentro del paquete especificado en el archivo *AndroidManifest.xml*, se guardan los datos ingresados por el alumno dentro de las variables correspondientes y se comienza a verificar que cumplan las longitudes correctas. El Sistema de Administración Escolar establece que los passwords de los alumnos deben ser de mínimo diez caracteres y máximo quince caracteres. Si el password excede la longitud de quince caracteres, no se toman en cuenta los caracteres del dieciseis en adelante. Seguido de esto, se crea una nueva instancia de la clase *Conexion*, enviando como argumentos del método constructor la matrícula y password recién ingresados y se asigna la variable de referencia al dato correspon-

diente en la clase *Datos*. Luego se intenta ingresar los datos al servidor por medio del método *post* definido en la clase *Conexion*. Si se logra ingresar los datos, se almacenan las *cookies* recibidas en la variable correspondiente de la clase *Conexion* y se continúa con el módulo de solicitud. En caso de que el dispositivo no tenga acceso a internet, se crea una instancia de la clase *Error* enviando como argumento la actividad actual y por medio del método *despliega* definido en dicha clase se muestra el mensaje de error siguiente:

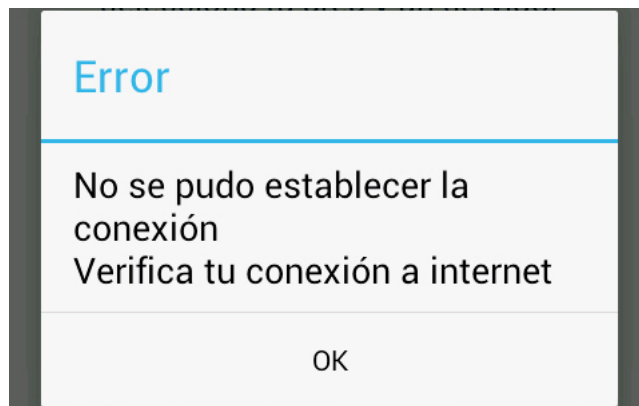


Figura 7: Error de conexión

Por otro lado, si el alumno llega a oprimir el botón antes de ingresar su matrícula o password completos, de igual manera se crea la instancia de la clase *Error* y se despliega en pantalla el siguiente error:

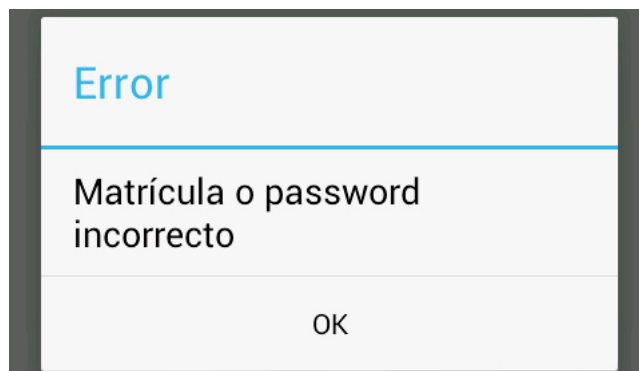


Figura 8: Error de matrícula o password

Finalmente si el alumno llega a ingresar datos con longitudes dentro del rango establecido, pero ingresa datos incorrectos, se despliega en pantalla el siguiente error:

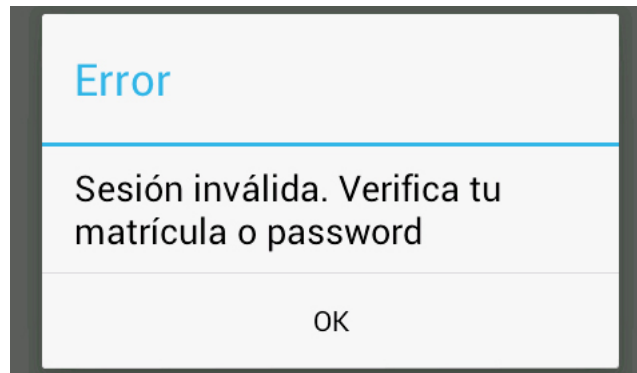


Figura 9: Error sesión inválida

5.2 Módulo de solicitud

Ya teniendo la conexión con el servidor se intenta obtener la información académica del alumno por medio del método *get* definido en la clase *Conexion*, el cual asigna un flujo de datos de entrada en la clase *Datos* a la respuesta del servidor. Al momento de recibir la respuesta se comienzan a leer los datos del alumno por medio del método *leeInfoAcademica* definido en la clase *Datos* y se asignan las variables correspondientes dentro de esta misma clase. Al terminar de leer la información académica se solicita el historial académico del alumno, nuevamente con el método *get* de la clase *Conexion*, y al obtener la respuesta se reasigna el flujo de datos de entrada de la clase *Datos* a esta nueva respuesta. El historial académico completo se lee a través del método *leeKardex* definido en la clase *Datos*. Al detectarse una nueva UEA, se almacenan sus componentes en una instancia temporal de la clase *UEA* y al terminar de leer dicha UEA se agrega a una lista de UEA definida en la clase *Datos*. La clase *UEA* se compone de los siguientes datos:

1. Una lista privada de datos tipo *String* que sirve para almacenar la seriación de la UEA.
2. Un dato tipo *String* para almacenar la clave de la UEA.
3. Un dato tipo *String* para almacenar el nombre de la UEA.

4. Un dato tipo *String* para almacenar el trimestre en que se cursó la UEA.
5. Un dato tipo *String* para almacenar el tipo de evaluación en que se cursó la UEA.
6. Un dato tipo *String* para almacenar la calificación obtenida en la UEA.
7. Un dato tipo *String* para almacenar si la UEA es obligatoria u optativa.
8. Un dato tipo *int* para almacenar el número de créditos que proporciona la UEA si se acredita.
9. Un dato tipo *int* para almacenar la prioridad asignada de la UEA.
10. Un dato tipo *int* para almacenar el número de oportunidades que le quedan al alumno de cursar la UEA en evaluación global.
11. Un dato tipo *int* para almacenar el número de oportunidades que le quedan al alumno de cursar la UEA en evaluación de recuperación.

Al terminar de leer el historial académico se cierra la conexión con el servidor a través del método *close* definido en la clase *Conexion*. En caso de error, ya sea éste de lectura de algún flujo de datos o que la matrícula ingresada no pertenezca a Ingeniería en Computación, se crea una instancia de la clase *Error*, se despliega el error correspondiente en pantalla y se cierra la conexión para después ejecutar nuevamente el módulo de ingreso.

5.3 Módulo de análisis

Se recupera la información de la clase *Datos* que contiene el área seleccionada y dependiendo del área, se instanciará una nueva variable de la clase *PlanEstudios* definida en el paquete del proyecto, enviando como argumento de su método constructor un número entero, que servirá para cargar en memoria las UEA correspondientes al área seleccionada más las UEA del tronco de nivelación, del general, del básico profesional, del inter y multidisciplinar y las no pertenecientes a ningún área del tronco de integración. Las variables definidas en la clase *PlanEstudios* son las siguientes:

1. Una lista privada de datos tipo *UEA* que almacena las UEA del tronco de nivelación.
2. Una lista privada de datos tipo *UEA* que almacena las UEA del tronco general.

3. Una lista privada de datos tipo *UEA* que almacena las UEA del tronco básico profesional.
4. Una lista privada de datos tipo *UEA* que almacena las UEA del tronco inter y multidisciplinar.
5. Una lista privada de datos tipo *UEA* que almacena las UEA del tronco de integración.
6. Una lista privada de datos tipo *UEA* que almacena las UEA del área seleccionada.
7. Un dato tipo *int* para almacenar el número del área seleccionada.
8. Un dato tipo *int* para almacenar la prioridad de algunas UEA del tronco inter y multidisciplinar.
9. Un dato tipo *int* para almacenar la prioridad de algunas UEA del tronco de integración.

Al término de esto, en caso de que aún no se hayan personalizado los créditos a recomendar, se calcula el promedio del alumno en el trimestre anterior. Para esto, solo se toman en cuenta las UEA con calificación igual a *S (6)*, *B (8)* o *MB (10)*. Si el resultado del cálculo es menor a 6.0, el promedio es *NA*; si se encuentra en el intervalo [6.0, 7.6), el promedio es *S*; si se encuentra en el intervalo [7.6, 8.6), el promedio es *B*; si es mayor o igual a 8.6 el promedio es *MB*. Luego se calcula el número máximo de créditos permitidos a inscribir recuperando la información sobre renunciadas a UEA en el trimestre anterior, verificando si el alumno reprobó alguna UEA en el mismo y con el promedio calculado con anterioridad. Si se reprobó o se renunció a alguna UEA o el promedio es igual a *NA* o *S*, se asigna el valor cuarenta a la variable de créditos máximos permitidos. En caso contrario, se asigna el valor de sesenta y tres. Posteriormente se calcula la lista de UEA posibles a inscribir con el método *calculaUEAPosibles* definido en la clase *PlanEstudios*. Con este método se comienzan a recorrer los troncos definidos en el plan de estudios con ayuda de los métodos *eliminaAcreditadas*, *eliminaPorUEA*, *eliminaPorCreditos*, *eliminaNoAutorizadas* y *eliminaNoOportunidad*, definidos en la clase *PlanEstudios*. El primer método elimina las UEA de un tronco o del área de concentración que ya fueron acreditadas. Si alguna UEA se cursó, ya sea en evaluación global o de recuperación y no fue aprobada, resta el número de oportunidades para

cursar dicha UEA en el tipo de evaluación correspondiente. Además si alguna UEA aprobada es del tronco inter y multidisciplinar o de integración, reasigna las prioridades de las UEA de esos troncos. El segundo método elimina las UEA que no pueden ser cursadas debido a la seriación de la UEA. El tercero elimina las UEA que no pueden ser cursadas debido a que no se cumple con un número mínimo de créditos acumulados. El cuarto elimina las UEA que no pueden ser cursadas debido a que no se ha autorizado cursar dicha UEA. El último elimina las UEA que no pueden ser cursadas debido a que ya han sido reprobadas dos veces en evaluación global y tres veces en evaluación de recuperación. Finalmente las UEA no eliminadas de los troncos y área de concentración se asignan a la lista posible de UEA a inscribir y se ordena de forma ascendente por medio del método *sort* de la clase *Collections*, de acuerdo a la prioridad y al número de créditos que proporciona la UEA. Si el alumno ya realizó una personalización de créditos, se omiten los cálculos de promedio y créditos máximos permitidos.

5.4 Módulo de recomendación

Si es la primera vez que se ejecuta este módulo, se calcula el promedio móvil de los últimos tres trimestres con el método *promedioMovil* definido en la clase *Datos* y se asigna el número mínimo de créditos a recomendar como el resultado de este cálculo. Si el resultado es menor a cero (no se ha finalizado ningún trimestre) se asigna un mínimo de treinta créditos a recomendar. Para obtener el número máximo de créditos a recomendar se realiza la suma del número mínimo de créditos a recomendar más nueve (promedio de seis, nueve y doce, créditos que otorgan la mayoría de las UEA). Si dicha suma es mayor al máximo permitido, entonces como máximo se recomendará al máximo permitido. En caso contrario se recomendará la suma antes descrita. Posteriormente se comienza a recorrer la lista de UEA posibles a inscribir. En cada UEA contenida en la lista se pregunta si la UEA es obligatoria u optativa. En caso de ser optativa se agrega a la lista de optativas definida en la clase *Datos*. En caso de ser obligatoria se verifica que la suma del número de créditos que otorga la UEA más el número de créditos de las UEA contenidas en la lista de UEA obligatorias (también definida en la clase *Datos*) sea menor o igual al número máximo de créditos a recomendar. Si es así, la UEA se agrega a la lista de UEA obligatorias a recomendar y se continúa con el recorrido. Al agregarse una UEA más a la lista de obligatorias se verifica si la suma del número de créditos acumulados más el número de créditos de las UEA contenidas en

dicha lista es mayor o igual a 491. En caso positivo, se asigna una variable tipo *boolean* con valor verdadero definida en la clase *Datos* que se utiliza en el módulo de presentación para dar aviso al alumno si está próximo a finalizar la licenciatura. Si ya se realizó la personalización de créditos se omiten los cálculos de créditos mínimos y máximos a recomendar.

5.5 Módulo de presentación

El diseño de la presentación se realizó en el archivo *recomendacion.xml*. En este archivo se agregaron diecisiete componentes *TextView* para indicar al alumno qué datos son los que se muestran. Catorce de estos componentes se insertaron en un componente *TableLayout* para mantenerlos organizados y se puedan visualizar de una mejor manera. Estos datos son los extraídos del servidor y los calculados en los módulos anteriores. Debajo de la tabla de datos se insertó una tabla más sin ningún componente dentro de ella, debido a que esta tabla se llena dinámicamente con las UEA obligatorias a recomendar desde el archivo *Recomendacion.java*, el cual se encuentra en la sección 8, y seguido de esta tabla, se encuentra un componente *ScrollView* que sirve para poder desplegar la lista de UEA optativas dentro de una tabla más que, en caso de que no se puedan recomendar UEA optativas, permanecerá vacía. Si hay oportunidad de recomendar UEA optativas, la tabla también se llenará dinámicamente desde el archivo *Recomendacion.java*. Después hay dos cadenas de texto que indican al alumno el número de créditos que podría acumular si llega a inscribir y acreditar las UEA obligatorias que se le recomiendan y si podría finalizar la carrera en el trimestre siguiente. Finalmente se insertaron tres componentes *Button*. El primero sirve para mandar llamar a la actividad principal y ejecutar nuevamente el módulo de ingreso. El segundo para iniciar la actividad de personalización de créditos, la cual se realizó en el archivo *personalizacion.xml* y el tercero para iniciar la actividad que brinda al alumno la oportunidad de enviar un mensaje de correo electrónico al Coordinador de la carrera y al desarrollador de la aplicación. La interfaz gráfica de la actividad del comentario se realizó en el archivo *comentar.xml*. En la interfaz gráfica de la personalización de créditos se agregaron los datos recolectados del servidor y se sustituyeron dos componentes *TextView* por dos componentes *EditText*, que sirven para que el alumno ingrese los valores de su preferencia, siempre y cuando el valor máximo sea menor o igual al máximo permitido y el valor mínimo sea menor que el valor máximo. Además se agregó un componente *Spinner* para que el alumno tenga oportunidad

de seleccionar otra área de concentración y un componente *Button* para ejecutar nuevamente el módulo de recomendación. En la interfaz gráfica de la actividad del comentario se agregó un componente *TextView* con un mensaje informativo para el usuario, además de los campos en los que se indican los correos electrónicos destino, el asunto del mensaje y un campo de texto multilínea que sirve para ingresar el comentario. Al último se encuentran dos botones; el primero sirve para enviar el mensaje con ayuda de un cliente de correo electrónico que se encuentre instalado en el dispositivo y el segundo para regresar a la actividad de recomendación.

En las siguientes dos figuras se muestran las interfaces realizadas para la actividad de la recomendación:



Figura 10: Recomendación de UEA sin optativas



Figura 11: Recomendación de UEA con optativas

Las siguientes imágenes corresponden a la interfaz gráfica de la actividad de personalización de créditos:

The screenshot shows a mobile application interface titled "Recomendación UEA Comput...". At the top, there is a status bar with "3G" and "10:30". Below the title bar, there is a small green robot icon. The main content area displays the following information:

- Matricula: 209329877
- Nombre: DANIEL ALBERTO TORRES GONZALEZ
- Créditos Acumulados: 439
- Créditos Mínimos Permitidos: 0
- Créditos Máximos Permitidos: 40

Below this information, there are two input fields for credit calculation:

- Créditos Mínimos a Calcular: 0
- Créditos Máximos a Calcular: 27

At the bottom, there is a dropdown menu for "Área:" with the selected option "Sistemas de Información". A large "OK" button is positioned at the bottom of the screen.

Figura 12: Personalización de créditos

Si el alumno no ingresa ambos datos se muestra el siguiente error:

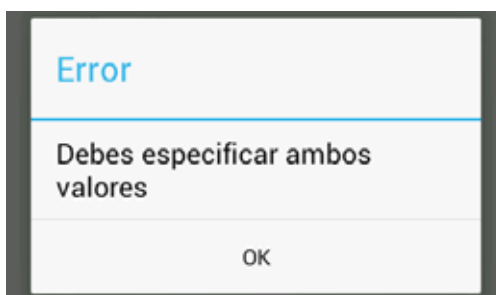


Figura 13: Error de ingreso de valores

Si en el campo especificado para los créditos mínimos se ingresa un valor mayor que en el campo especificado para los créditos máximos, se muestra el siguiente error:

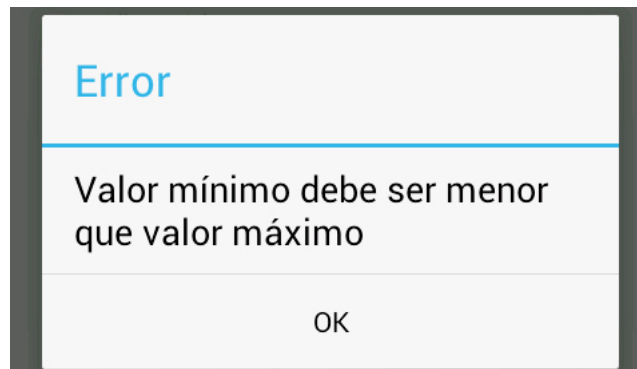


Figura 14: Error de valor mínimo

Finalmente, si en el campo especificado para los créditos máximos se ingresa un valor mayor que el máximo permitido, se muestra el siguiente error:

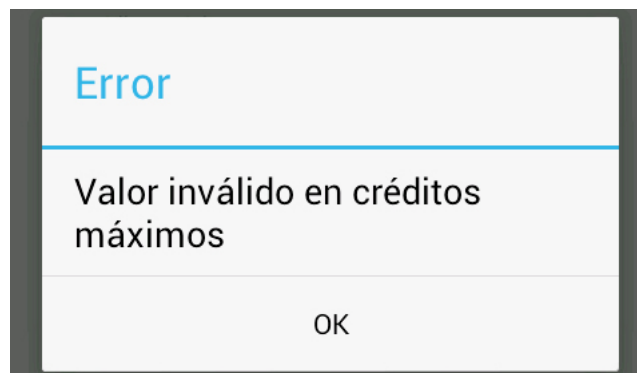


Figura 15: Error de valor máximo

La últimas 2 imágenes corresponden a la interfaz gráfica de la actividad del comentario:

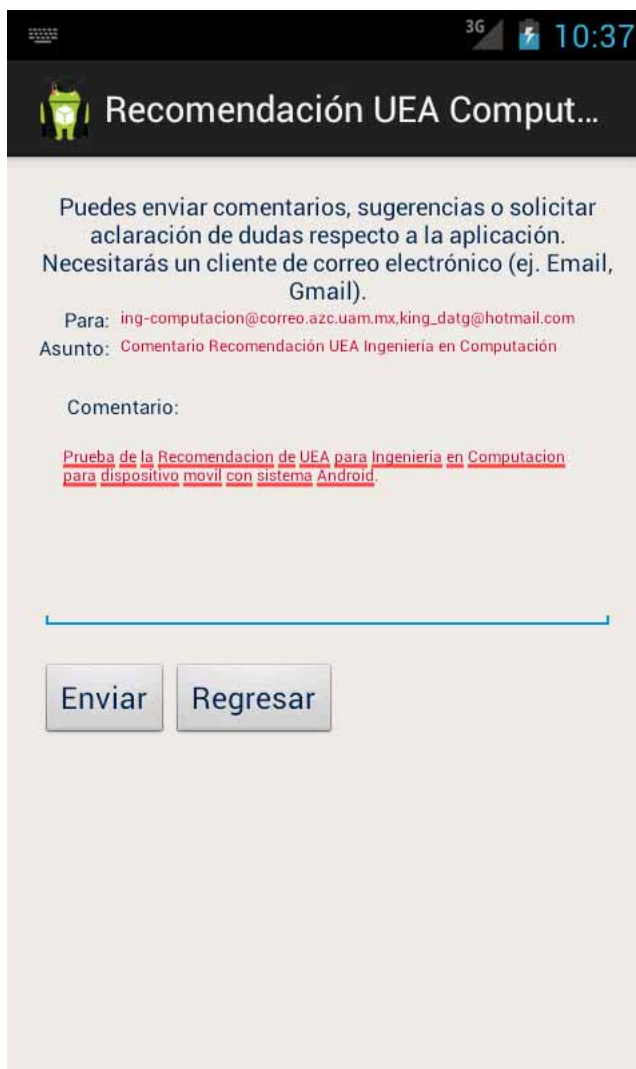


Figura 16: Enviar comentario

Al oprimir el botón "Enviar" se solicita al alumno seleccionar el cliente de correo electrónico de su preferencia:

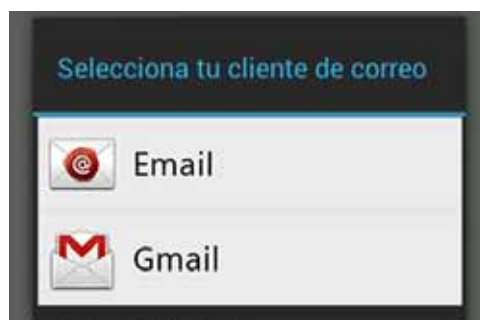


Figura 17: Selección de cliente de correo electrónico

6. Conclusiones

Conforme se fue desarrollando el proyecto se ha logrado comprender que en estos tiempos es de mucha utilidad el uso de las nuevas tecnologías, ya que desempeñan un papel muy importante entre la sociedad. Además se adquirió un mayor conocimiento sobre como se conforman y como operan estas tecnologías al momento de interactuar con el usuario final.

Cabe mencionar que al estar implementando la aplicación se reafirmaron muchos de los conocimientos adquiridos a lo largo de la carrera y se adquirió mayor experiencia, siendo de gran ayuda para el desarrollo y a su vez, para mejorar el desempeño de la aplicación.

Con respecto al objetivo general y a los objetivos específicos planteados inicialmente, podemos decir que todos fueron cumplidos, ya que se lograron implementar todos los módulos descritos en la propuesta, de forma que al trabajar en conjunto, se obtienen los resultados esperados. Esto se comprobó al momento de realizar pruebas en el emulador y en dispositivos móviles.

Pensando a futuro, se espera que la aplicación pueda anticipar las UEA que serán impartidas en el trimestre siguiente y de esa forma, sugerir al alumno solo las UEA que estarán disponibles, además de los horarios asegurando un cupo dentro del horario sugerido. Esperemos que próximamente se brinden estas nuevas herramientas de ayuda a la comunidad universitaria para mejorar su desempeño académico.

7. Referencias

- [1] Google Inc, Open Handset Alliance, Android Open Source Project. "Android OS." Internet: <http://www.android.com>, Sept. 23, 2008 [Jul. 9, 2013].
- [2] Microsoft Corporation, "Microsoft Windows 8." Internet: <http://windows.microsoft.com/es-mx/windows-8/meet>, Aug. 1, 2012 [Sept. 6, 2012].
- [3] Oracle Corporation, "Java Development Kit." Internet: <http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html>, Nov. 17, 2006 [May. 12, 2011].
- [4] The Eclipse Foundation, "Eclipse SDK." Internet: <http://www.eclipse.org/downloads/packages/eclipse-ide-java-developers/junosr2>, Jun. 27, 2012 [Jul. 9, 2013].
- [5] Creative Commons Attribution 2.5, "Android SDK." Internet: <http://developer.android.com/sdk/index.html>, Nov 12, 2007 [Jul. 9, 2013].
- [6] Google Inc, "ADT Plugin For Eclipse." Internet: <http://developer.android.com/tools/sdk/eclipse-adt.html/>, Jul. 2013 [Jul. 9, 2013].
- [7] Google Inc, "Virtual Device Manager." Internet: <http://developer.android.com/tools/devices/index.html>, Jul. 2008 [Jul. 9, 2013].
- [8] Oracle Corporation, "Overview (Java Platform SE 7)." Internet: <http://docs.oracle.com/javase/7/docs/api/>, Jan. 1993 [May. 5, 2011].
- [9] Google Inc, "App Components Android Developers." Internet: <http://developer.android.com/guide/components/index.html>, Sept. 2008 [Jul. 9, 2013].

8. Apéndices

Archivo: AndroidManifest.xml

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.proyecto.proyecto"
4     android:versionCode="1"
5     android:versionName="1.0"
6     android:installLocation="preferExternal">
7     <uses-permission
8         android:name="android.permission.INTERNET" />
9     <uses-sdk
10        android:minSdkVersion="14"
11        android:targetSdkVersion="18" />
12     <application
13        android:allowBackup="true"
14        android:icon="@drawable/ic_launcher"
15        android:label="@string/app_name"
16        android:theme="@style/AppTheme" >
17        <activity
18            android:name=".MainActivity"
19            android:label="@string/app_name"
20            android:screenOrientation="portrait"
21            android:configChanges="keyboard|keyboardHidden|
22                orientation" >
23            <intent-filter>
24                <action android:name="android.intent.action.MAIN" />
25                <category android:name="android.intent.category.
26                    LAUNCHER" />
27            </intent-filter>
28        </activity>
29        <activity
30            android:name=".Recomendacion"
31            android:label="@string/app_name"
32            android:screenOrientation="portrait"
33            android:configChanges="keyboard|keyboardHidden|
34                orientation">
35            <intent-filter>
36                <action android:name="com.proyecto.proyecto.
37                    RECOMENDACION" />
38                <category android:name="android.intent.category.
39                    DEFAULT" />
40            </intent-filter>
41        </activity>
42        <activity
43            android:name=".Personalizacion"
44            android:label="@string/app_name"
45            android:screenOrientation="portrait"
```

```

41         android:configChanges="keyboard|keyboardHidden|
           orientation">
42     <intent-filter>
43         <action android:name="com.proyecto.proyecto.
           PERSONALIZACION" />
44         <category android:name="android.intent.category.
           DEFAULT" />
45     </intent-filter>
46 </activity>
47 <activity
48     android:name=".Comentar"
49     android:label="@string/app_name"
50     android:screenOrientation="portrait"
51     android:configChanges="keyboard|keyboardHidden|
           orientation">
52     <intent-filter>
53         <action android:name="com.proyecto.proyecto.COMENTAR"
           />
54         <category android:name="android.intent.category.
           DEFAULT" />
55     </intent-filter>
56 </activity>
57 </application>
58 </manifest>

```

Archivo: strings.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <resources>
3     <string name="app_name">Recomendacion UEA Computacion</string>
4     <string name="action_settings">Settings</string>
5     <string name="Matricula">Matricula:</string>
6     <string name="Password">Password:</string>
7     <string name="Area">Area:</string>
8     <string name="Servidor">Servidor:</string>
9     <string name="Ok">OK</string>
10    <string name="Welcome">Bienvenido a la Recomendacion de UEA de
           Ingenieria en Computacion</string>
11    <string name="Instrucciones">Ingresa los datos solicitados ,
           selecciona tu area y un servidor</string>
12    <string name="Comenta">Puedes enviar comentarios, sugerencias o
           solicitar aclaracion de dudas respecto a la aplicacion.
           Necesitaras un cliente de correo electronico (ej. Email, Gmail
           ).</string>
13    <string name="Personalizar">Personalizar Creditos</string>
14    <string name="Nuevo">Nuevo Alumno</string>
15    <string name="Comentar">Comentar</string>
16    <string name="Salir">Salir</string>
17    <string name="Nombre">Nombre:</string>
18    <string name="CredAcu">Creditos Acumulados:</string>

```

```

19 <string name="CredMinPer">Creditos Minimos Permitidos:</string>
20 <string name="CredMaxPer">Creditos Maximos Permitidos:</string>
21 <string name="CredMinRec">Creditos Minimos Recomendados:</string>
22 <string name="CredMaxRec">Creditos Maximos Recomendados:</string>
23 <string name="CredMinCalc">Creditos Minimos a Calcular:</string>
24 <string name="CredMaxCalc">Creditos Maximos a Calcular:</string>
25 <string name="Nota">Se te recomienda inscribir las siguientes
    UEA:</string>
26 <string name="De">De:</string>
27 <string name="Para">Para:</string>
28 <string name="Asunto">Asunto:</string>
29 <string name="Comentario">Comentario:</string>
30 <string name="Enviar">Enviar</string>
31 <string name="Regresar">Regresar</string>
32 </resources>

```

Archivo: activity_main.xml

```

1 <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/
  android"
2   xmlns:tools="http://schemas.android.com/tools"
3   android:layout_width="fill_parent"
4   android:layout_height="fill_parent"
5   android:layout_gravity="center"
6   android:background="#ebebe6"
7   android:paddingBottom="@dimen/activity_vertical_margin"
8   android:paddingLeft="@dimen/activity_horizontal_margin"
9   android:paddingRight="@dimen/activity_horizontal_margin"
10  android:paddingTop="@dimen/activity_vertical_margin"
11  tools:context=".MainActivity">
12  <TextView
13      android:id="@+id/welcome"
14      android:layout_width="fill_parent"
15      android:layout_height="wrap_content"
16      android:layout_alignParentTop="true"
17      android:layout_centerHorizontal="true"
18      android:clickable="false"
19      android:gravity="center"
20      android:text="@string/Welcome"
21      android:textColor="#02224b"
22      android:textSize="18sp" />
23  <TextView
24      android:id="@+id/instrucciones"
25      android:layout_width="fill_parent"
26      android:layout_height="wrap_content"
27      android:layout_below="@+id/welcome"
28      android:layout_centerHorizontal="true"
29      android:layout_marginTop="10dp"
30      android:clickable="false"
31      android:gravity="center"

```

```

32         android:text="@string/Instrucciones"
33         android:textSize="16sp"
34         android:textColor="#02224b" />
35 <EditText
36     android:id="@+id/mat"
37     android:layout_width="fill_parent"
38     android:layout_height="40dp"
39     android:layout_below="@+id/instrucciones"
40     android:layout_marginLeft="10dp"
41     android:layout_marginTop="15dp"
42     android:layout_toRightOf="@+id/Matricula"
43     android:ems="10"
44     android:gravity="center_vertical|center_horizontal"
45     android:inputType="number"
46     android:textColor="#ce032e"
47     android:textSize="14sp" />
48 <TextView
49     android:id="@+id/Matricula"
50     android:layout_width="70dp"
51     android:layout_height="wrap_content"
52     android:layout_alignParentLeft="true"
53     android:layout_below="@id/instrucciones"
54     android:layout_marginTop="25dp"
55     android:clickable="false"
56     android:gravity="right"
57     android:text="@string/Matricula"
58     android:textColor="#02224b"
59     android:textSize="14sp" />
60 <TextView
61     android:id="@+id/Password"
62     android:layout_width="70dp"
63     android:layout_height="wrap_content"
64     android:layout_alignParentLeft="true"
65     android:layout_below="@+id/Matricula"
66     android:layout_marginTop="30dp"
67     android:clickable="false"
68     android:gravity="right"
69     android:text="@string/Password"
70     android:textColor="#02224b"
71     android:textSize="14sp" />
72 <EditText
73     android:id="@+id/pass"
74     android:layout_width="fill_parent"
75     android:layout_height="40dp"
76     android:layout_below="@id/mat"
77     android:layout_marginLeft="10dp"
78     android:layout_marginTop="7dp"
79     android:layout_toRightOf="@id/Password"
80     android:ems="10"
81     android:gravity="center_vertical|center_horizontal"
82     android:inputType="textPassword"

```

```

83         android:textColor="#ce032e"
84         android:textSize="14sp" />
85     <TextView
86         android:id="@+id/Area"
87         android:layout_width="70dp"
88         android:layout_height="wrap_content"
89         android:layout_alignParentLeft="true"
90         android:layout_below="@+id/Password"
91         android:layout_marginTop="35dp"
92         android:clickable="false"
93         android:gravity="right"
94         android:text="@string/Area"
95     android:textColor="#02224b"
96         android:textSize="14sp" />
97     <Spinner
98         android:id="@+id/ar"
99         android:layout_width="fill_parent"
100        android:layout_height="50dp"
101        android:layout_below="@+id/pass"
102        android:layout_marginLeft="10dp"
103        android:layout_marginTop="7dp"
104        android:layout_toRightOf="@id/Area"
105        android:spinnerMode="dialog" />
106     <TextView
107        android:id="@+id/Servidor"
108        android:layout_width="70dp"
109        android:layout_height="wrap_content"
110        android:layout_alignParentLeft="true"
111        android:layout_below="@+id/Area"
112        android:layout_marginTop="37dp"
113        android:clickable="false"
114        android:gravity="right"
115        android:text="@string/Servidor"
116    android:textColor="#02224b"
117        android:textSize="14sp" />
118     <Spinner
119        android:id="@+id/serv"
120        android:layout_width="fill_parent"
121        android:layout_height="50dp"
122        android:layout_below="@+id/ar"
123        android:layout_marginLeft="10dp"
124        android:layout_marginTop="7dp"
125        android:layout_toRightOf="@id/Servidor"
126        android:spinnerMode="dialog" />
127     <Button
128        android:id="@+id/btn_Ok"
129        style="@android:style/Widget.Button"
130        android:layout_width="match_parent"
131        android:layout_height="40dp"
132        android:layout_below="@+id/serv"
133        android:layout_centerHorizontal="true"

```



```

134         android:layout_marginTop="30dp"
135         android:text="@string/Ok"
136         android:textColor="#02224b"
137         android:textSize="15sp" />
138 </RelativeLayout>

```

Archivo: comentar.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:id="@+id/RelativeLayout1"
5     android:layout_width="fill_parent"
6     android:layout_height="fill_parent"
7     android:background="#ebeb6"
8     android:paddingBottom="@dimen/activity_vertical_margin"
9     android:paddingLeft="@dimen/activity_horizontal_margin"
10    android:paddingRight="@dimen/activity_horizontal_margin"
11    android:paddingTop="@dimen/activity_vertical_margin"
12    android:layout_gravity="center"
13    android:orientation="vertical" >
14    <TextView
15        android:id="@+id/comenta"
16        android:layout_width="fill_parent"
17        android:layout_height="wrap_content"
18        android:layout_alignParentTop="true"
19        android:layout_centerHorizontal="true"
20        android:clickable="false"
21        android:gravity="center"
22        android:text="@string/Comenta"
23        android:textSize="12sp"
24        android:textColor="#02224b" />
25    <TableLayout
26        android:id="@+id/datos"
27        android:layout_width="fill_parent"
28        android:layout_height="wrap_content"
29        android:layout_below="@+id/comenta" >
30        <TableRow
31            android:id="@+id/Para"
32            android:layout_width="fill_parent"
33            android:layout_height="40dp" >
34            <TextView
35                android:id="@+id/para"
36                android:layout_width="wrap_content"
37                android:layout_height="fill_parent"
38                android:gravity="right"
39                android:text="@string/Para"
40            android:textColor="#02224b"
41            android:textSize="10sp" />
42            <TextView

```

```

43         android:id="@+id/txt_para"
44         android:layout_width="fill_parent"
45         android:layout_height="fill_parent"
46         android:layout_marginLeft="5dp"
47         android:gravity="left"
48         android:textColor="#ce032e" />
49     </TableRow>
50     <TableRow
51         android:id="@+id/Asunto"
52         android:layout_width="fill_parent"
53         android:layout_height="40dp" >
54         <TextView
55             android:id="@+id/asunto"
56             android:layout_width="wrap_content"
57             android:layout_height="fill_parent"
58             android:gravity="right"
59             android:text="@string/Asunto"
60             android:textColor="#02224b"
61             android:textSize="10sp" />
62         <TextView
63             android:id="@+id/txt_asunto"
64             android:layout_width="fill_parent"
65             android:layout_height="fill_parent"
66             android:layout_marginLeft="5dp"
67             android:gravity="left"
68             android:textColor="#ce032e" />
69     </TableRow>
70 </TableLayout>
71 <TextView
72     android:id="@+id/Comentario"
73     android:layout_width="70dp"
74     android:layout_height="wrap_content"
75     android:layout_alignParentLeft="true"
76     android:layout_below="@+id/datos"
77     android:layout_marginTop="15dp"
78     android:clickable="false"
79     android:gravity="right"
80     android:text="@string/Comentario"
81     android:textSize="10sp"
82     android:textColor="#02224b" />
83 <EditText
84     android:id="@+id/txt_comentario"
85     android:layout_width="fill_parent"
86     android:layout_height="100dp"
87     android:layout_alignParentLeft="true"
88     android:layout_below="@+id/Comentario"
89     android:layout_marginTop="5dp"
90     android:gravity="left"
91     android:ems="10"
92     android:inputType="textMultiLine"
93     android:textColor="#ce032e" >

```

```

94 </EditText>
95 <Button
96     android:id="@+id/btn_enviar"
97     style="@android:style/Widget.Button"
98     android:layout_width="wrap_content"
99     android:layout_height="40dp"
100    android:layout_alignParentLeft="true"
101    android:layout_below="@+id/txt_comentario"
102    android:layout_marginTop="15dp"
103    android:text="@string/Enviar"
104    android:textColor="#02224b"
105    android:textSize="15sp" />
106 <Button
107     android:id="@+id/btn_regresar"
108     style="@android:style/Widget.Button"
109     android:layout_width="wrap_content"
110     android:layout_height="40dp"
111     android:layout_below="@+id/txt_comentario"
112     android:layout_toRightOf="@+id/btn_enviar"
113     android:layout_marginTop="15dp"
114     android:text="@string/Regresar"
115     android:textColor="#02224b"
116     android:textSize="15sp" />
117 </RelativeLayout>

```

Archivo: personalizacion.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:id="@+id/RelativeLayout2"
5     android:layout_width="fill_parent"
6     android:layout_height="fill_parent"
7     android:background="#ebeb66"
8     android:paddingBottom="@dimen/activity_vertical_margin"
9     android:paddingLeft="@dimen/activity_horizontal_margin"
10    android:paddingRight="@dimen/activity_horizontal_margin"
11    android:paddingTop="@dimen/activity_vertical_margin"
12    android:layout_gravity="center"
13    android:orientation="vertical" >
14    <TableLayout
15        android:id="@+id/Info2"
16        android:layout_width="fill_parent"
17        android:layout_height="wrap_content"
18        android:layout_alignParentTop="true" >
19        <TableRow
20            android:id="@+id/Matr2"
21            android:layout_width="fill_parent"
22            android:layout_height="40dp" >
23            <TextView

```

```

24         android:id="@+id/matr2"
25         android:layout_width="wrap_content"
26         android:layout_height="fill_parent"
27         android:layout_gravity="left"
28         android:text="@string/Matricula"
29         android:textSize="8sp"
30         android:textColor="#02224b" />
31     <TextView
32         android:id="@+id/txt_matr2"
33         android:layout_width="fill_parent"
34         android:layout_height="fill_parent"
35         android:layout_gravity="left"
36         android:layout_marginLeft="5dp"
37         android:textColor="#ce032e" />
38 </TableRow>
39 <TableRow
40     android:id="@+id/Nombre2"
41     android:layout_width="fill_parent"
42     android:layout_height="40dp" >
43     <TextView
44         android:id="@+id/nombre2"
45         android:layout_width="wrap_content"
46         android:layout_height="fill_parent"
47         android:layout_gravity="left"
48         android:text="@string/Nombre"
49         android:textSize="8sp"
50         android:textColor="#02224b" />
51     <TextView
52         android:id="@+id/txt_nombre2"
53         android:layout_width="fill_parent"
54         android:layout_height="fill_parent"
55         android:layout_gravity="left"
56         android:layout_marginLeft="5dp"
57         android:textColor="#ce032e" />
58 </TableRow>
59 <TableRow
60     android:id="@+id/CredAcumulados2"
61     android:layout_width="fill_parent"
62     android:layout_height="40dp" >
63     <TextView
64         android:id="@+id/credAcu2"
65         android:layout_width="wrap_content"
66         android:layout_height="fill_parent"
67         android:layout_gravity="left"
68         android:text="@string/CredAcu"
69         android:textSize="8sp"
70         android:textColor="#02224b" />
71     <TextView
72         android:id="@+id/txt_credAcu2"
73         android:layout_width="fill_parent"
74         android:layout_height="fill_parent"

```

```

75         android:layout_gravity=" left "
76         android:layout_marginLeft=" 5dp"
77         android:textColor="#ce032e" />
78     </TableRow>
79     <TableRow
80         android:id="@+id / CredMinPermitidos2"
81         android:layout_width=" fill_parent "
82         android:layout_height=" 40dp" >
83         <TextView
84             android:id="@+id / credMinPermitidos2"
85             android:layout_width=" wrap_content "
86             android:layout_height=" fill_parent "
87             android:layout_gravity=" left "
88             android:text=" @string / CredMinPer"
89             android:textSize=" 8sp"
90             android:textColor=" #02224b" />
91         <TextView
92             android:id="@+id / txt_credMinPer2"
93             android:layout_width=" fill_parent "
94             android:layout_height=" fill_parent "
95             android:layout_gravity=" left "
96             android:layout_marginLeft=" 5dp"
97             android:textColor="#ce032e" />
98     </TableRow>
99     <TableRow
100         android:id="@+id / CredMaxPermitidos2"
101         android:layout_width=" fill_parent "
102         android:layout_height=" 40dp" >
103         <TextView
104             android:id="@+id / credMaxPermitidos2"
105             android:layout_width=" wrap_content "
106             android:layout_height=" fill_parent "
107             android:layout_gravity=" left "
108             android:text=" @string / CredMaxPer"
109             android:textSize=" 8sp"
110             android:textColor=" #02224b" />
111         <TextView
112             android:id="@+id / txt_credMaxPer2"
113             android:layout_width=" fill_parent "
114             android:layout_height=" fill_parent "
115             android:layout_gravity=" left "
116             android:layout_marginLeft=" 5dp"
117             android:textColor="#ce032e" />
118     </TableRow>
119 </TableLayout>
120 <TextView
121     android:id="@+id / credMinRecomendados2"
122     android:layout_width=" 130dp"
123     android:layout_height=" wrap_content "
124     android:layout_alignParentLeft=" true"
125     android:layout_below=" @id / Info2"

```

```

126     android:layout_marginTop="5dp"
127     android:clickable="false"
128     android:gravity="center"
129     android:text="@string/CredMinCalc"
130     android:textColor="#02224b"
131     android:textSize="10sp" />
132 <EditText
133     android:id="@+id/edit_credMinRec"
134     android:layout_width="fill_parent"
135     android:layout_height="25dp"
136     android:layout_below="@id/Info2"
137     android:layout_marginLeft="10dp"
138     android:layout_marginTop="6dp"
139     android:layout_toRightOf="@+id/credMinRecomendados2"
140     android:gravity="center"
141     android:inputType="number"
142     android:textColor="#ce032e"
143     android:textSize="10sp" />
144 <TextView
145     android:id="@+id/credMaxRecomendados2"
146     android:layout_width="130dp"
147     android:layout_height="wrap_content"
148     android:layout_alignParentLeft="true"
149     android:layout_below="@id/credMinRecomendados2"
150     android:layout_marginTop="10dp"
151     android:clickable="false"
152     android:gravity="center"
153     android:text="@string/CredMaxCalc"
154     android:textColor="#02224b"
155     android:textSize="10sp" />
156 <EditText
157     android:id="@+id/edit_credMaxRec"
158     android:layout_width="fill_parent"
159     android:layout_height="25dp"
160     android:layout_below="@id/edit_credMinRec"
161     android:layout_marginLeft="10dp"
162     android:layout_marginTop="8dp"
163     android:layout_toRightOf="@+id/credMaxRecomendados2"
164     android:gravity="center"
165     android:inputType="number"
166     android:textColor="#ce032e"
167     android:textSize="10sp" />
168 <TextView
169     android:id="@+id/Area2"
170     android:layout_width="130dp"
171     android:layout_height="wrap_content"
172     android:layout_alignParentLeft="true"
173     android:layout_below="@+id/credMaxRecomendados2"
174     android:layout_marginTop="19dp"
175     android:clickable="false"
176     android:gravity="center"

```

```

177     android:text="@string/Area"
178     android:textColor="#02224b"
179     android:textSize="10sp" />
180 <Spinner
181     android:id="@+id/ar2"
182     android:layout_width="fill_parent"
183     android:layout_height="40dp"
184     android:layout_below="@+id/edit_credMaxRec"
185     android:layout_marginLeft="10dp"
186     android:layout_marginTop="7dp"
187     android:layout_toRightOf="@id/Area2"
188     android:spinnerMode="dialog" />
189 <Button
190     android:id="@+id/btn_Ok2"
191     style="@android:style/Widget.Button"
192     android:layout_width="match_parent"
193     android:layout_height="40dp"
194     android:layout_below="@+id/Area2"
195     android:layout_centerHorizontal="true"
196     android:layout_marginTop="20dp"
197     android:text="@string/Ok"
198     android:textColor="#02224b"
199     android:textSize="15sp" />
200 </RelativeLayout>

```

Archivo: recomendacion.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <RelativeLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:id="@+id/RelativeLayout1"
5     android:layout_width="fill_parent"
6     android:layout_height="fill_parent"
7     android:background="#ebebe6"
8     android:paddingBottom="@dimen/activity_vertical_margin"
9     android:paddingLeft="@dimen/activity_horizontal_margin"
10    android:paddingRight="@dimen/activity_horizontal_margin"
11    android:paddingTop="@dimen/activity_vertical_margin"
12    android:layout_gravity="center"
13    android:orientation="vertical" >
14    <TableLayout
15        android:id="@+id/Info"
16        android:layout_width="fill_parent"
17        android:layout_height="wrap_content"
18        android:layout_alignParentTop="true" >
19        <TableRow
20            android:id="@+id/Matr"
21            android:layout_width="fill_parent"
22            android:layout_height="40dp" >
23            <TextView

```

```

24         android:id="@+id/matr"
25         android:layout_width="wrap_content"
26         android:layout_height="fill_parent"
27         android:layout_gravity="left"
28         android:text="@string/Matricula"
29         android:textSize="8sp"
30         android:textColor="#02224b" />
31     <TextView
32         android:id="@+id/txt_matr"
33         android:layout_width="fill_parent"
34         android:layout_height="fill_parent"
35         android:layout_gravity="left"
36         android:layout_marginLeft="5dp"
37         android:textColor="#ce032e" />
38 </TableRow>
39 <TableRow
40     android:id="@+id/Nombre"
41     android:layout_width="fill_parent"
42     android:layout_height="40dp" >
43     <TextView
44         android:id="@+id/nombre"
45         android:layout_width="wrap_content"
46         android:layout_height="fill_parent"
47         android:layout_gravity="left"
48         android:text="@string/Nombre"
49         android:textSize="8sp"
50         android:textColor="#02224b" />
51     <TextView
52         android:id="@+id/txt_nombre"
53         android:layout_width="fill_parent"
54         android:layout_height="fill_parent"
55         android:layout_gravity="left"
56         android:layout_marginLeft="5dp"
57         android:textColor="#ce032e" />
58 </TableRow>
59 <TableRow
60     android:id="@+id/CredAcumulados"
61     android:layout_width="fill_parent"
62     android:layout_height="40dp" >
63     <TextView
64         android:id="@+id/credAcu"
65         android:layout_width="wrap_content"
66         android:layout_height="fill_parent"
67         android:layout_gravity="left"
68         android:text="@string/CredAcu"
69         android:textSize="8sp"
70         android:textColor="#02224b" />
71     <TextView
72         android:id="@+id/txt_credAcu"
73         android:layout_width="fill_parent"
74         android:layout_height="fill_parent"

```



```

75         android:layout_gravity=" left "
76         android:layout_marginLeft=" 5dp"
77         android:textColor="#ce032e" />
78     </TableRow>
79     <TableRow
80         android:id="@+id / CredMinPermitidos"
81         android:layout_width=" fill_parent "
82         android:layout_height=" 40dp" >
83         <TextView
84             android:id="@+id / credMinPermitidos"
85             android:layout_width=" wrap_content "
86             android:layout_height=" fill_parent "
87             android:layout_gravity=" left "
88             android:text=" @string / CredMinPer"
89             android:textSize=" 8sp"
90             android:textColor=" #02224b" />
91         <TextView
92             android:id="@+id / txt_credMinPer"
93             android:layout_width=" fill_parent "
94             android:layout_height=" fill_parent "
95             android:layout_gravity=" left "
96             android:layout_marginLeft=" 5dp"
97             android:textColor="#ce032e" />
98     </TableRow>
99     <TableRow
100         android:id="@+id / CredMaxPermitidos"
101         android:layout_width=" fill_parent "
102         android:layout_height=" 40dp" >
103         <TextView
104             android:id="@+id / credMaxPermitidos"
105             android:layout_width=" wrap_content "
106             android:layout_height=" fill_parent "
107             android:layout_gravity=" left "
108             android:text=" @string / CredMaxPer"
109             android:textSize=" 8sp"
110             android:textColor=" #02224b" />
111         <TextView
112             android:id="@+id / txt_credMaxPer"
113             android:layout_width=" fill_parent "
114             android:layout_height=" fill_parent "
115             android:layout_gravity=" left "
116             android:layout_marginLeft=" 5dp"
117             android:textColor="#ce032e" />
118     </TableRow>
119     <TableRow
120         android:id="@+id / CredMinRecomendados"
121         android:layout_width=" fill_parent "
122         android:layout_height=" 40dp" >
123         <TextView
124             android:id="@+id / credMinRecomendados"
125             android:layout_width=" wrap_content "

```

```

126         android:layout_height="fill_parent"
127         android:layout_gravity="left"
128         android:text="@string/CredMinRec"
129         android:textSize="8sp"
130         android:textColor="#02224b" />
131     <TextView
132         android:id="@+id/txt_credMinRec"
133         android:layout_width="fill_parent"
134         android:layout_height="fill_parent"
135         android:layout_gravity="left"
136         android:layout_marginLeft="5dp"
137         android:textColor="#ce032e" />
138 </TableRow>
139 <TableRow
140     android:id="@+id/CredMaxRecomendados"
141     android:layout_width="fill_parent"
142     android:layout_height="40dp" >
143     <TextView
144         android:id="@+id/credMaxRecomendados"
145         android:layout_width="wrap_content"
146         android:layout_height="fill_parent"
147         android:layout_gravity="left"
148         android:text="@string/CredMaxRec"
149         android:textSize="8sp"
150         android:textColor="#02224b" />
151     <TextView
152         android:id="@+id/txt_credMaxRec"
153         android:layout_width="fill_parent"
154         android:layout_height="fill_parent"
155         android:layout_gravity="left"
156         android:layout_marginLeft="5dp"
157         android:textColor="#ce032e" />
158 </TableRow>
159 </TableLayout>
160 <TextView
161     android:id="@+id/nota"
162     android:layout_width="fill_parent"
163     android:layout_height="wrap_content"
164     android:layout_below="@id/Info"
165     android:layout_marginTop="10dp"
166     android:text="@string/Nota"
167     android:textSize="10sp"
168     android:textColor="#02224b" />
169 <TableLayout
170     android:id="@+id/UEA"
171     android:layout_width="fill_parent"
172     android:layout_height="wrap_content"
173     android:layout_marginTop="5dp"
174     android:layout_below="@id/nota" >
175 </TableLayout>
176 <TextView

```

```

177         android:id="@+id/nota2"
178         android:layout_width="fill_parent"
179         android:layout_height="wrap_content"
180         android:layout_below="@id/UEA"
181         android:layout_marginTop="10dp"
182         android:textColor="#02224b" />
183     <ScrollView
184         android:id="@+id/scroll"
185         android:layout_width="fill_parent"
186         android:layout_height="50dp"
187         android:layout_below="@id/nota2"
188         android:scrollbars="none"
189         android:layout_marginTop="5dp" >
190         <TableLayout
191             android:id="@+id/UEA2"
192             android:layout_width="fill_parent"
193             android:layout_height="wrap_content" >
194         </TableLayout>
195     </ScrollView>
196     <TextView
197         android:id="@+id/acumularias"
198         android:layout_width="fill_parent"
199         android:layout_height="wrap_content"
200         android:layout_below="@id/scroll"
201         android:layout_marginTop="10dp"
202         android:textColor="#02224b" />
203 <TextView
204     android:id="@+id/fin"
205     android:layout_width="fill_parent"
206     android:layout_height="wrap_content"
207     android:layout_below="@id/acumularias"
208     android:layout_marginTop="5dp"
209     android:textColor="#02224b" />
210 <Button
211     android:id="@+id/btn_nuevo"
212     style="@android:style/Widget.Button"
213     android:layout_width="wrap_content"
214     android:layout_height="40dp"
215     android:layout_alignParentLeft="true"
216     android:layout_marginTop="30dp"
217     android:layout_below="@+id/fin"
218     android:gravity="center"
219     android:text="@string/Nuevo"
220     android:textSize="10sp"
221     android:textColor="#02224b" />
222 <Button
223     android:id="@+id/btn_personalizar"
224     style="@android:style/Widget.Button"
225     android:layout_width="wrap_content"
226     android:layout_height="40dp"
227     android:layout_below="@+id/fin"

```

```

228     android:layout_marginTop="30dp"
229     android:layout_marginRight="5dp"
230     android:layout_toRightOf="@id/btn_nuevo"
231     android:gravity="center"
232     android:text="@string/Personalizar"
233     android:textSize="10sp"
234     android:textColor="#02224b" />
235 <Button
236     android:id="@+id/btn_comentar"
237     style="@android:style/Widget.Button"
238     android:layout_width="wrap_content"
239     android:layout_height="40dp"
240     android:layout_below="@+id/fin"
241     android:layout_marginTop="30dp"
242     android:layout_marginRight="5dp"
243     android:layout_toRightOf="@id/btn_personalizar"
244     android:gravity="center"
245     android:text="@string/Comentar"
246     android:textSize="10sp"
247     android:textColor="#02224b" />
248 </RelativeLayout>

```

Archivo: spinner_main.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <TextView
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:id="@android:id/text1"
5     android:layout_width="fill_parent"
6     android:layout_height="fill_parent"
7     android:textSize="14sp"
8     android:gravity="center"
9     android:textColor="#ce032e" />

```

Archivo: spinner_personalizacion.xml

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <TextView
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     android:id="@android:id/text1"
5     android:layout_width="fill_parent"
6     android:layout_height="fill_parent"
7     android:textSize="8sp"
8     android:gravity="center"
9     android:textColor="#ce032e" />

```

Archivo: Comentar.java

```
1 package com.proyecto.proyecto;
2
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.os.Bundle;
6 import android.view.View;
7 import android.widget.Button;
8 import android.widget.TextView;
9
10 public class Comentar extends Activity{
11     private TextView para;
12     private TextView asunto;
13     private TextView comentario;
14     private Button enviar;
15     private Button regresar;
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState){
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.comentar);
21         this.para=(TextView)findViewById(R.id.txt_para);
22         this.para.setTextSize(8);
23         this.para.setText("ing-computacion@correo.azc.uam.mx,
24             king_datg@hotmail.com");
25
26         this.asunto=(TextView)findViewById(R.id.txt_asunto);
27         this.asunto.setTextSize(8);
28         this.asunto.setText("Comentario Recomendacion UEA Ingenieria en
29             Computacion");
30
31         this.comentario=(TextView)findViewById(R.id.txt_comentario);
32         this.comentario.setTextSize(8);
33
34         this.enviar=(Button)findViewById(R.id.btn_enviar);
35         this.enviar.setOnClickListener(new View.OnClickListener(){
36             @Override
37             public void onClick(View v){
38                 Intent i=new Intent(Intent.ACTION_SEND);
39                 i.putExtra(Intent.EXTRA_EMAIL,new String[]{"ing-
40                     computacion@correo.azc.uam.mx", "king_datg@hotmail.com"});
41                 i.putExtra(Intent.EXTRA_SUBJECT, asunto.getText().toString());
42                 i.putExtra(Intent.EXTRA_TEXT, comentario.getText().toString());
43                 ;
44                 i.setType("message/rfc822");
45                 startActivity(Intent.createChooser(i,"Selecciona tu cliente
46                     de correo"));
47             }
48         });
49     }
50 }
```

```

44
45     this.regresar=(Button)findViewById(R.id.btn_regresar);
46     this.regresar.setOnClickListener(new View.OnClickListener(){
47         @Override
48         public void onClick(View v){
49             Intent i=new Intent("com.proyecto.proyecto.RECOMENDACION");
50             startActivity(i);
51         }
52     });
53 }
54 }

```

Archivo: Conexion.java

```

1 package com.proyecto.proyecto;
2
3 import org.apache.http.Header;
4 import org.apache.http.HttpResponse;
5 import org.apache.http.NameValuePair;
6 import org.apache.http.client.ClientProtocolException;
7 import org.apache.http.client.entity.UrlEncodedFormEntity;
8 import org.apache.http.client.methods.HttpGet;
9 import org.apache.http.client.methods.HttpPost;
10 import org.apache.http.impl.client.DefaultHttpClient;
11 import org.apache.http.message.BasicNameValuePair;
12
13 import java.io.BufferedReader;
14 import java.io.InputStream;
15 import java.io.InputStreamReader;
16 import java.io.UnsupportedEncodingException;
17 import java.io.IOException;
18
19 import java.util.List;
20 import java.util.ArrayList;
21
22 public class Conexion{
23     private DefaultHttpClient cliente;
24     private HttpResponse respuestaPost;
25     private HttpResponse respuestaGet;
26     private Header [] cookies;
27     private String matricula,password;
28     /*
29      * Metodo constructor.
30      */
31     public Conexion(String mat,String pass){
32         this.cliente=new DefaultHttpClient();
33         this.respuestaPost=null;
34         this.respuestaGet=null;
35         this.matricula=mat;
36         this.password=pass;

```

```

37 }
38 /*
39  * Metodo para ingresar datos al servidor.
40  */
41 public boolean post(String link, boolean cierre){
42     List <NameValuePair> datos=new ArrayList <NameValuePair> (11);
43     if(!cierre){
44         datos.add(new BasicNameValuePair("#.E.UNIDAD.AE02.1", "AxJDMQ="));
45         datos.add(new BasicNameValuePair("#.MODULOUWAS.SAE01", ""));
46         datos.add(new BasicNameValuePair("#.USUARIO_ANEXO.SG02", ""));
47         datos.add(new BasicNameValuePair("#.WEB_INFO.SW01", ""));
48         datos.add(new BasicNameValuePair("#.WEB_MOD_ASO.SW01", ""));
49         datos.add(new BasicNameValuePair("#CRC.E.UNIDAD.AE02.1", "00000022"));
50         datos.add(new BasicNameValuePair("%IDENTIFICACION.NONMODELED", ""));
51         datos.add(new BasicNameValuePair("COMPLEMENTO.IDENTIFICACION.NONMODELED", password));
52         datos.add(new BasicNameValuePair("GO.IDENTIFICACION.NONMODELED", "Entrar"));
53         datos.add(new BasicNameValuePair("NOMBRE.IDENTIFICACION.NONMODELED", matricula));
54         datos.add(new BasicNameValuePair("SIGLAS.UNI.XX.E.UNIDAD.AE02.1", "AZC"));
55     }
56     HttpPost post=new HttpPost(link);
57     try{
58         post.setEntity(new UrlEncodedFormEntity(datos));
59     }
60     catch(UnsupportedEncodingException e){
61         return(false);
62     }
63     try{
64         this.respuestaPost=cliente.execute(post);
65     }
66     catch(ClientProtocolException e){
67         return(false);
68     }
69     catch(IOException e){
70         return(false);
71     }
72     this.cookies=this.respuestaPost.getHeaders("Set-Cookie");
73     return(true);
74 }
75 /*
76  * Metodo para solicitar datos al servidor.
77  */
78 public BufferedReader get(String link){
79     HttpGet get=new HttpGet(link);
80     for(Header c : this.cookies)

```

```

81     get.addHeader("Cookie",c.getValue());
82     try{
83         this.respuestaGet=this.cliente.execute(get);
84     }
85     catch(ClientProtocolException e){
86         return(null);
87     }
88     catch(IOException e){
89         return(null);
90     }
91     InputStream ins=null;
92     try{
93         ins=this.respuestaGet.getEntity().getContent();
94     }
95     catch(IllegalStateException e){
96         return(null);
97     }
98     catch(IOException e){
99         return(null);
100    }
101    BufferedReader in=new BufferedReader(new InputStreamReader(ins));
102    return(in);
103 }
104 /*
105  * Metodo para cerrar la conexion.
106  */
107 public void close(){
108     post("https://" + Datos.servidor + ".uam.mx:8443/sae/azc/AEWBU002
109         .oFin?mod=1",true);
110     try{
111         this.respuestaPost.getEntity().consumeContent();
112     }
113     catch(IOException e){}
114     catch(NullPointerException e){}
115 }

```

Archivo: Datos.java

```

1 package com.proyecto.proyecto;
2
3 import java.io.BufferedReader;
4 import java.io.IOException;
5
6 import java.util.ArrayList;
7 import java.util.List;
8 import java.util.StringTokenizer;
9
10 public class Datos{
11     private static String ultimoTrim;

```



```

12  private static int renunciadas;
13
14  protected static List <UEA> kardex;
15  protected static List <String> autorizaciones;
16  protected static List <UEA> listaFinalObl;
17  protected static List <UEA> listaFinalOpt;
18  protected static List <UEA> listaPosible;
19  protected static String nombre;
20  protected static String apP;
21  protected static String apM;
22  protected static String matricula;
23  protected static String password;
24  protected static String area;
25  protected static String servidor;
26  protected static boolean carrera;
27  protected static boolean fin;
28  protected static int acumulados;
29  protected static int creditosMax;
30  protected static int creditosMinRec;
31  protected static int creditosMaxRec;
32  protected static int sumaCred;
33
34  protected static Conexion conn;
35  protected static BufferedReader in;
36
37  protected static void inicializaDatos () {
38      ultimoTrim="";
39      renunciadas=0;
40
41      kardex=new ArrayList <UEA> ();
42      autorizaciones=new ArrayList <String> ();
43      listaFinalObl=new ArrayList <UEA> ();
44      listaFinalOpt=new ArrayList <UEA> ();
45      listaPosible=new ArrayList <UEA> ();
46      nombre="";
47      apP="";
48      apM="";
49      matricula="";
50      password="";
51      area="";
52      servidor="";
53      carrera=false;
54      fin=false;
55      acumulados=0;
56      creditosMax=0;
57      creditosMinRec=0;
58      creditosMaxRec=0;
59      sumaCred=0;
60
61      conn=null;
62      in=null;

```

```

63     }
64     /*
65     * Metodo para leer el kardex del alumno.
66     */
67     protected static boolean leeKardex () {
68         UEA uea=new UEA();
69         String linea ,token;
70         try{
71             while((linea=in.readLine())!=null){
72                 if(linea.contains("CVEUEA.CL.V_HIS.AC.VIG.VW01.")){
73                     StringTokenizer st=new StringTokenizer(linea,"",false);
74                     token=st.nextToken("<td style=\"text-align: center;\">");
75                     uea=new UEA();
76                     uea.setClave(token);
77                 }
78                 else if(linea.contains("TRIMESTRE.XX.TRIMESTRE.LEC.AE02.")){
79                     StringTokenizer st=new StringTokenizer(linea,"",false);
80                     token=st.nextToken("<");
81                     uea.setTrimestre(token);
82                 }
83                 else if(linea.contains("TIPO_EVAL.XX.TIPO_EVALUA.AE02.")){
84                     StringTokenizer st=new StringTokenizer(linea,"",false);
85                     token=st.nextToken("<");
86                     uea.setEvaluacion(token);
87                 }
88                 else if(linea.contains("CALIFICA.CL.V_HIS.AC.VIG.VW01.")){
89                     StringTokenizer st=new StringTokenizer(linea,"",false);
90                     token=st.nextToken("<td style=\"text-align: center;\">");
91                     uea.setCalificacion(token);
92                 }
93                 else if(linea.contains("NO.CRED.UE.NU")){
94                     token="";
95                     int i=0;
96                     while(linea.charAt(i)!='/'){
97                         i++;
98                     }
99                     while(!Character.isDigit(linea.charAt(i))){
100                         i++;
101                     }
102                     while(linea.charAt(i)!='<'){
103                         token+=linea.charAt(i);
104                         i++;
105                     }
106                     uea.setNumCreditos(Integer.parseInt(token));
107                     kardex.add(uea);
108                 }
109             }
110         }
111         catch(IOException e){
112             return(false);
113         }

```

```

114     return(true);
115 }
116 /*
117  * Metodo para leer la carrera del alumno, el nombre completo, el
118  * ultimo trimestre inscrito,
119  * el numero de creditos acumulados, las autorizaciones y el numero
120  * de renunciaciones.
121  */
122 protected static boolean leeInfoAcademica() {
123     String linea;
124     boolean x=false;
125     try{
126         while((linea=in.readLine())!=null){
127             if(linea.contains("Sesion invalida"))
128                 return(false);
129             else if(linea.contains("LICENCIATURA EN INGENIERIA EN
130                 COMPUTACION"))
131                 carrera=true;
132             else if(linea.contains("APE.PAT.AL.XX.V.ALUMNO.ACAD")){
133                 StringTokenizer st=new StringTokenizer(linea,"",false);
134                 apP=st.nextToken("<td><strong>");
135             }
136             else if(linea.contains("APE.MAT.AL.XX.V.ALUMNO.ACAD")){
137                 StringTokenizer st=new StringTokenizer(linea,"",false);
138                 apM=st.nextToken("<td><strong>");
139             }
140             else if(linea.contains("NOMBRE.AL.NO.V.ALUMNO.ACAD")){
141                 StringTokenizer st=new StringTokenizer(linea,"",false);
142                 nombre=st.nextToken("<td><strong>");
143             }
144             else if(linea.contains("CRED.ACUAL.NU.V.ALUMNO.ACAD")){
145                 String aux="";
146                 int i=0;
147                 while(linea.charAt(i)!='>'){
148                     i++;
149                 }
150                 i++;
151                 while(linea.charAt(i)!='<'){
152                     aux+=linea.charAt(i);
153                     i++;
154                 }
155                 acumulados=Integer.parseInt(aux);
156             }
157             else if(linea.contains("CVE.UEA.CL.AL.AUTORIZACION.AE02.")){
158                 StringTokenizer st=new StringTokenizer(linea,"",false);
159                 autorizaciones.add(st.nextToken("<td>"));
160             }
161             else if(linea.contains("TRIMESTRE.XX.TRIMESTRE.ACA")){
162                 int i=0;
163                 while(linea.charAt(i)!='>'){
164                     i++;

```

```

162     }
163     i++;
164     while(linea.charAt(i)!='<'){
165         ultimoTrim+=linea.charAt(i);
166         i++;
167     }
168 }
169 else if(ultimoTrim.compareTo("")!=0){
170     if(linea.contains(ultimoTrim))
171         x=true;
172     if(x && linea.contains("RENUNCIA_RT_NU.V.RESUMEN_TRI.VW01."
173 )){
174         StringTokenizer st=new StringTokenizer(linea,"",false);
175         renunciaciones=Integer.parseInt(st.nextToken("<td>"));
176         x=false;
177     }
178 }
179 }
180 catch(IOException e){
181     return(false);
182 }
183 return(true);
184 }
185 /*
186  * Metodo que carga el plan de estudios vigente. Tambien calcula
187  * una lista posible de UEA
188  * a inscribir y la cantidad maxima de creditos a inscribir.
189  */
190 protected static void analisis(boolean personalizado){
191     PlanEstudios plan;
192     if(area.compareTo("Sistemas de Informacion")==0)
193         plan=new PlanEstudios(0);
194     else if(area.compareTo("Algoritmos e Inteligencia Artificial")
195 ==0)
196         plan=new PlanEstudios(1);
197     else if(area.compareTo("Seguridad y Redes de Computadoras")==0)
198         plan=new PlanEstudios(2);
199     else if(area.compareTo("Sistemas Embebidos")==0)
200         plan=new PlanEstudios(3);
201     else
202         plan=new PlanEstudios(4);
203     if(!personalizado){
204         String p=promedio();
205         creditosMax=(renunciaciones>0 || UEANoAprobada() || p.compareTo("S")
206 ==0 || p.compareTo("NA")==0) ? 40:63;
207     }
208     listaPosible=plan.calculaUEAPosibles();
209 }
210 /*
211  * Metodo que calcula el numero de creditos minimos y maximos a

```

```

209     * recomendar. Ademias calcula la
210     * lista final de UEA obligatorias y optativas a recomendar. Evalua
211     * si el alumno puede
212     * terminar en el trimestre a cursar.
213     */
214 protected static void recomendacion(boolean personalizado){
215     if (!personalizado){
216         int i=promedioMovil();
217         creditosMinRec=(i===-1) ? 30 : i;
218         creditosMaxRec=(creditosMinRec+9>creditosMax) ? creditosMax :
219             creditosMinRec+9;
220     }
221     for(UEA uea : listaPosible){
222         if(uea.getObl_opt().compareTo("OBL")==0){
223             if(sumaCred+uea.getNumCreditos()<=creditosMaxRec){
224                 listaFinalObl.add(uea);
225                 sumaCred+=uea.getNumCreditos();
226                 if(sumaCred+acumulados>=491)
227                     fin=true;
228             }
229         }
230     }
231     else
232         listaFinalOpt.add(uea);
233     }
234 }
235 /*
236 * Metodo para verificar que el alumno no reprobó alguna UEA en su
237 * ultimo trimestre.
238 */
239 private static boolean UEANoAprobada(){
240     int tam=kardex.size();
241     UEA aux;
242     for(int i=tam-1;i>=0;i--){
243         aux=kardex.get(i);
244         if(aux.getTrimestre().compareTo(ultimoTrim)==0 && aux.
245             getCalificacion().compareTo("NA")==0)
246             return(true);
247     }
248     return(false);
249 }
250 /*
251 * Metodo para verificar el promedio del alumno en su ultimo
252 * trimestre inscrito.
253 */
254 private static String promedio(){
255     int suma=0,cont=0,tam=kardex.size();
256     UEA aux;
257     for(int i=tam-1;i>=0;i--){
258         aux=kardex.get(i);
259         if(aux.getTrimestre().compareTo(ultimoTrim)==0){
260             if(aux.getCalificacion().compareTo("S")==0){

```

```

254         suma+=6;
255         cont++;
256     }
257     else if(aux.getCalificacion().compareTo("B")==0){
258         suma+=8;
259         cont++;
260     }
261     else if(aux.getCalificacion().compareTo("MB")==0){
262         suma+=10;
263         cont++;
264     }
265 }
266 }
267 float promedio=0.0f;
268 try{
269     promedio=suma/cont;
270 }
271 catch(ArithmeticException e){}
272 if(promedio<6.0f)
273     return("NA");
274 else if(promedio>=6.0f && promedio<7.6f)
275     return("S");
276 else if(promedio>=7.6f && promedio<8.6f)
277     return("B");
278 return("MB");
279 }
280 /*
281  * Metodo para calcular el promedio movil (ultimos 3 trimestres).
282  */
283 private static int promedioMovil(){
284     List<String> trims=new ArrayList<String> ();
285     for(UEA uea : kardex){
286         if(uea.getCalificacion().compareTo("NA")!=0 && !trims.contains(
287             uea.getTrimestre()))
288             trims.add(uea.getTrimestre());
289     }
290     int n_trim=trims.size();
291     if(n_trim==0)
292         return(-1);
293     int index, cred;
294     int [] sumas=new int[n_trim];
295     for(int i=0;i<n_trim;i++)
296         sumas[i]=0;
297     for(UEA uea : kardex){
298         if(uea.getCalificacion().compareTo("NA")!=0){
299             cred=uea.getNumCreditos();
300             index=trims.indexOf(uea.getTrimestre());
301             sumas[index]+=cred;
302         }
303     }
304     int sumT=0, iter=0;

```

```

304     if (n_trim==1)
305         return (sumas [0]);
306     else if (n_trim==2)
307         return ((int) ((sumas [0]+sumas [1]) /2));
308     else {
309         for (int i=n_trim-1; iter <3; i--, iter++)
310             sumT+=sumas [ i];
311     }
312     return ((int) (sumT/3));
313 }
314 }

```

Archivo: Error.java

```

1 package com.proyecto.proyecto;
2
3 import android.app.AlertDialog;
4 import android.content.Context;
5 import android.content.DialogInterface;
6
7 public class Error{
8     private Context cont=null;
9
10    public Error(Context c){
11        this.cont=c;
12    }
13
14    public void despliega(String msg){
15        AlertDialog.Builder builder=new AlertDialog.Builder(this.cont);
16        builder.setTitle("Error");
17        builder.setMessage(msg).setCancelable(false)
18            .setPositiveButton("OK",new DialogInterface.OnClickListener(){
19                public void onClick(DialogInterface dialog,int id){}
20            });
21        AlertDialog ad=builder.create();
22        ad.show();
23    }
24 }

```

Archivo: MainActivity.java

```

1 package com.proyecto.proyecto;
2
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.os.Bundle;
6 import android.os.StrictMode;
7 import android.view.View;
8 import android.view.View.OnClickListener;

```

```

9 import android.widget.AdapterView;
10 import android.widget.Button;
11 import android.widget.EditText;
12 import android.widget.Spinner;
13
14 public class MainActivity extends Activity{
15     private EditText matricula , password;
16     private Spinner area , servidor;
17     private Button ok;
18
19     private final String [] a={"Sistemas de Informacion",
20         "Algoritmos e Inteligencia Artificial",
21         "Seguridad y Redes de Computadoras",
22         "Sistemas Embebidos",
23         "Mecatronica"};
24     private final String [] s={"atzoncal",
25         "ayamictlan"};
26
27     @Override
28     protected void onCreate(Bundle savedInstanceState){
29         StrictMode.ThreadPolicy policy=new StrictMode.ThreadPolicy.
30             Builder().permitAll().build();
31         StrictMode.setThreadPolicy(policy);
32         super.onCreate(savedInstanceState);
33         setContentView(R.layout.activity_main);
34         this.matricula=(EditText)findViewById(R.id.mat);
35         this.password=(EditText)findViewById(R.id.pass);
36
37         ArrayAdapter <String> estilo1=new ArrayAdapter <String> (this,R.
38             layout.spinner_main , this.a);
39         estilo1.setDropDownViewResource(R.layout.spinner_main);
40         this.area=(Spinner)findViewById(R.id.ar);
41         this.area.setAdapter(estilo1);
42
43         ArrayAdapter <String> estilo2=new ArrayAdapter <String> (this,R.
44             layout.spinner_main , s);
45         estilo2.setDropDownViewResource(R.layout.spinner_main);
46         this.servidor=(Spinner)findViewById(R.id.serv);
47         this.servidor.setAdapter(estilo2);
48
49         this.ok=(Button)findViewById(R.id.btn_Ok);
50         this.ok.setOnClickListener(new OnClickListener(){
51             @Override
52             public void onClick(View v){
53                 Datos.inicializaDatos();
54                 Datos.matricula=matricula.getText().toString();
55                 Datos.password=password.getText().toString();
56                 Datos.area=area.getSelectedItem().toString();
57                 Datos.servidor=servidor.getSelectedItem().toString();
58                 if(Datos.matricula.length()!=9 || Datos.password.length()<10)
59                     {

```



```

56     Error e=new Error(MainActivity.this);
57     e.despliega("Matricula o password incorrecto");
58     return;
59 }
60 if(Datos.password.length()>15){
61     String aux="";
62     for(int i=0;i<15;i++)
63         aux+=Datos.password.charAt(i);
64     Datos.password=aux;
65 }
66
67 //-----MODULO DE INGRESO-----//
68 Datos.conn=new Conexion(Datos.matricula ,Datos.password);
69 if(!Datos.conn.post("https://" + Datos.servidor + ".uam.mx
70 :8443/sae/azc/AEWBU004.oInicioSesion?mod=1" ,false)){
71     Error e=new Error(MainActivity.this);
72     e.despliega("No se pudo establecer la conexion\nVerifica tu
73     conexion a internet");
74     return;
75 }
76 //-----FIN MODULO DE INGRESO
77 -----//
78
79 //-----MODULO DE SOLICITUD
80 -----//
81
82 if((Datos.in=Datos.conn.get("https://" + Datos.servidor + ".
83 uam.mx:8443/sae/azc/IEWBC007.oConsulta"))==null){
84     Error e=new Error(MainActivity.this);
85     e.despliega("No se pudo obtener la informacion academica");
86     Datos.conn.close();
87     return;
88 }
89 if(!Datos.leeInfoAcademica()){
90     Error e=new Error(MainActivity.this);
91     e.despliega("Sesion invalida. Verifica tu matricula o
92     password");
93     Datos.conn.close();
94     return;
95 }
96 if(!Datos.carrera){
97     Error e=new Error(MainActivity.this);
98     e.despliega("El alumno no pertenece a Ingenieria en
99     Computacion");
100    Datos.conn.close();
101    return;
102 }
103 if((Datos.in=Datos.conn.get("https://" + Datos.servidor + ".
104 uam.mx:8443/sae/azc/IEWBC020.oConsulta"))==null){
105     Error e=new Error(MainActivity.this);
106     e.despliega("No se pudo obtener el historial academico");
107     Datos.conn.close();

```

```

99         return;
100     }
101     if (!Datos.leeKardex()) {
102         Error e=new Error(MainActivity.this);
103         e.despliega("Error en lectura de kardex");
104         Datos.conn.close();
105         return;
106     }
107     Datos.conn.close();
108     //-----FIN MODULO DE SOLICITUD
109     //-----//
110     //-----MODULO DE ANALISIS
111     //-----//
112     Datos.analisis(false);
113     //-----FIN MODULO DE ANALISIS
114     //-----//
115     //-----MODULO DE RECOMENDACION
116     //-----//
117     Datos.recomendacion(false);
118     //-----FIN MODULO DE RECOMENDACION
119     //-----//
120     //-----MODULO DE PRESENTACION
121     //-----//
122     Intent i=new Intent("com.proyecto.proyecto.RECOMENDACION");
123     startActivity(i);
124     //-----FIN MODULO DE PRESENTACION
125     //-----//
126 }
127 });
128 }
129 }

```

Archivo: Personalizacion.java

```

1 package com.proyecto.proyecto;
2
3 import java.util.ArrayList;
4
5 import android.app.Activity;
6 import android.content.Intent;
7 import android.os.Bundle;
8 import android.os.StrictMode;
9 import android.view.View;
10 import android.view.View.OnClickListener;
11 import android.widget.AdapterView;
12 import android.widget.Button;
13 import android.widget.EditText;

```

```

14 import android.widget.Spinner;
15 import android.widget.TextView;
16
17 public class Personalizacion extends Activity{
18     private TextView matr;
19     private TextView nombre;
20     private TextView credAcu;
21     private TextView credMinPer;
22     private TextView credMaxPer;
23     private EditText minimos;
24     private EditText maximos;
25     private Button ok;
26     private Spinner area;
27
28     private final String [] a={"Sistemas de Informacion",
29         "Algoritmos e Inteligencia Artificial",
30         "Seguridad y Redes de Computadoras",
31         "Sistemas Embebidos",
32         "Mecatronica"};
33
34     @Override
35     protected void onCreate(Bundle savedInstanceState){
36         StrictMode.ThreadPolicy policy=new StrictMode.ThreadPolicy.
37             Builder().permitAll().build();
38         StrictMode.setThreadPolicy(policy);
39         super.onCreate(savedInstanceState);
40         setContentView(R.layout.personalizacion);
41         this.matr=(TextView)findViewById(R.id.txt_matr2);
42         this.matr.setTextSize(8);
43
44         this.nombre=(TextView)findViewById(R.id.txt_nombre2);
45         this.nombre.setTextSize(8);
46
47         this.credAcu=(TextView)findViewById(R.id.txt_credAcu2);
48         this.credAcu.setTextSize(8);
49
50         this.credMinPer=(TextView)findViewById(R.id.txt_credMinPer2);
51         this.credMinPer.setTextSize(8);
52         this.credMaxPer=(TextView)findViewById(R.id.txt_credMaxPer2);
53         this.credMaxPer.setTextSize(8);
54
55         this.minimos=(EditText)findViewById(R.id.edit_credMinRec);
56         this.maximos=(EditText)findViewById(R.id.edit_credMaxRec);
57         ArrayAdapter <String> estilo1=new ArrayAdapter <String> (this,R.
58             layout.spinner_personalizacion ,this.a);
59         estilo1.setDropDownViewResource(R.layout.spinner_personalizacion)
60         ;
61         this.area=(Spinner)findViewById(R.id.ar2);
62         this.area.setAdapter(estilo1);
63         this.ok=(Button)findViewById(R.id.btn_Ok2);
64         this.ok.setOnClickListener(new OnClickListener(){
65             @Override

```

```

62     public void onClick(View v){
63         String txt_min=minimos.getText().toString();
64         String txt_max=maximos.getText().toString();
65         if(txt_min.compareTo("")==0 || txt_max.compareTo("")==0){
66             Error e=new Error(Personalizacion.this);
67             e.despliega("Debes especificar ambos valores");
68             return;
69         }
70         int min=Integer.parseInt(txt_min);
71         int max=Integer.parseInt(txt_max);
72         if(min<0 || min>Datos.creditosMax){
73             Error e=new Error(Personalizacion.this);
74             e.despliega("Valor inválido en créditos máximos");
75             return;
76         }
77         if(max<0 || max>Datos.creditosMax){
78             Error e=new Error(Personalizacion.this);
79             e.despliega("Valor inválido en créditos máximos");
80             return;
81         }
82         if(min>=max){
83             Error e=new Error(Personalizacion.this);
84             e.despliega("Valor mínimo debe ser menor que valor máximo
85                ");
86             return;
87         }
88         Datos.creditosMinRec=min;
89         Datos.creditosMaxRec=max;
90         Datos.area=area.getSelectedItem().toString();
91         Datos.listaFinalObl=new ArrayList <UEA> ();
92         Datos.listaFinalOpt=new ArrayList <UEA> ();
93         Datos.listaPosible=new ArrayList <UEA> ();
94         Datos.fin=false;
95         Datos.sumaCred=0;
96         Datos.analisis(true);
97         Datos.recomendacion(true);
98         Intent i=new Intent("com.proyecto.proyecto.RECOMENDACION");
99         startActivity(i);
100     }
101     despliega();
102 }
103 /*
104  * Metodo para desplegar los datos de usuario.
105  */
106 private void despliega(){
107     this.matr.setText(Datos.matricula);
108     this.nombre.setText(Datos.nombre + " " + Datos.apP + " " + Datos.
109         apM);
110     this.credAcu.setText("" + Datos.acumulados);
111     this.credMinPer.setText("0");

```

```

111     this.credMaxPer.setText("" + Datos.creditosMax);
112 }
113 }

```

Archivo: PlanEstudios.java

```

1 package com.proyecto.proyecto;
2
3 import java.util.List;
4 import java.util.ArrayList;
5 import java.util.StringTokenizer;
6 import java.util.Collections;
7 import java.util.Comparator;
8
9 public class PlanEstudios{
10     private List <UEA> troncoNivelacion;
11     private List <UEA> troncoGeneral;
12     private List <UEA> troncoBasicoProfesional;
13     private List <UEA> troncoInterMultidisciplinar;
14     private List <UEA> troncoIntegracion;
15     private List <UEA> area;
16     private int areaSel;
17     private int prioridadInter;
18     private int prioridadIntegracion;
19     /*
20      * Metodo constructor.
21      */
22     public PlanEstudios(int a){
23         this.areaSel=a;
24         this.troncoNivelacion=new ArrayList <UEA> ();
25         this.troncoGeneral=new ArrayList <UEA> ();
26         this.troncoBasicoProfesional=new ArrayList <UEA> ();
27         this.troncoInterMultidisciplinar=new ArrayList <UEA> ();
28         this.troncoIntegracion=new ArrayList <UEA> ();
29         this.area=new ArrayList <UEA> ();
30         this.prioridadInter=34;
31         this.prioridadIntegracion=3;
32         troncoNivelacion();
33         troncoGeneral();
34         troncoBasicoProfesional();
35         troncoInterMultidisciplinar();
36         troncoIntegracion();
37         if(this.areaSel==0)
38             areaSistemasInformacion();
39         else if(this.areaSel==1)
40             areaInteligencia();
41         else if(this.areaSel==2)
42             areaRedes();
43         else if(this.areaSel==3)
44             areaSistemasEmbebidos();

```

```

45     else
46         areaMecatronica();
47     }
48     /*
49     * Metodo que calcula las UEA posibles a inscribir y las ordena por
50     * prioridad de forma
51     * ascendente. Si las prioridades son las mismas, las ordena por
52     * numero de creditos.
53     */
54     public List <UEA> calculaUEAPosibles() {
55         for(int i=0;i<6;i++){
56             eliminaAcreditadas(i);
57             eliminaPorUEA(i);
58             eliminaPorCreditos(i);
59             eliminaNoAutorizadas(i);
60             eliminaNoOportunidad(i);
61         }
62         List <UEA> lista=copiaUEA();
63         Collections.sort(lista ,new Comparator <UEA> () {
64             public int compare(UEA uea1,UEA uea2){
65                 if(uea1.getPrioridad()==uea2.getPrioridad()){
66                     if(uea1.getNumCreditos()==uea2.getNumCreditos())
67                         return(0);
68                     else if(uea1.getNumCreditos(>uea2.getNumCreditos())
69                         return(1);
70                     return(-1);
71                 }
72                 else if(uea1.getPrioridad(>uea2.getPrioridad())
73                     return(1);
74                     return(-1);
75                 }
76             });
77         return(lista);
78     }
79     /*
80     * Metodo que elimina las UEA de un tronco o area de concentracion
81     * que ya fueron acreditadas.
82     * Si alguna UEA se curso, ya sea en evaluacion global o de
83     * recuperacion y no fue aprobada,
84     * resta el numero de oportunidades para cursar dicha UEA. Ademas,
85     * si alguna UEA aprobada
86     * es del tronco Inter y Multidisciplinar o de Integracion,
87     * reasigna las prioridades de las UEA.
88     */
89     private void eliminaAcreditadas(int op){
90         List <UEA> lista=cargaLista(op);
91         int tamL=lista.size();
92         UEA uea1;
93         for(int i=0;i<tamL;i++){
94             uea1=lista.get(i);
95             for(UEA uea2 : Datos.kardex){

```

```

90     if(uea2.getClave().compareTo(uea1.getClave())==0){
91         if(uea2.getCalificacion().compareTo("NA")!=0){
92             if(esOptInter(uea2))
93                 reasignaPrioridadesInter();
94             else if(esOptIntegracion(uea2))
95                 reasignaPrioridadesIntegracion();
96             lista.remove(i);
97             tamL--;
98             i--;
99             break;
100        }
101        else if(uea2.getEvaluacion().compareTo("GLO.")==0)
102            uea1.restaGlo();
103        else
104            uea1.restaRec();
105    }
106 }
107 }
108 }
109 /*
110  * Metodo que elimina las UEA de un tronco o area de concentracion
111  * que no pueden ser cursadas
112  * debido a la seriacion de UEA.
113  */
114 private void eliminaPorUEA(int op){
115     List <UEA> lista=cargaLista(op);
116     if(lista.isEmpty())
117         return;
118     int tamL=lista.size();
119     UEA uea1;
120     boolean enc;
121     for(int i=0;i<tamL;i++){
122 label:     for(String s1 : uea1.getSeriacion()){
123         if(s1.compareTo("AU")!=0 && !s1.startsWith("NC-")){
124             if(s1.startsWith("C")){
125                 StringTokenizer st=new StringTokenizer(s1,"",false);
126                 s1=st.nextToken("C");
127             }
128             enc=false;
129             for(UEA uea2 : Datos.kardex){
130                 if(uea2.getClave().compareTo(s1)==0){
131                     if(uea2.getCalificacion().compareTo("NA")==0){
132                         lista.remove(i--);
133                         tamL--;
134                         break label;
135                     }
136                 else{
137                     enc=true;
138                     break;
139                 }

```

```

140         }
141     }
142     if(!enc){
143         lista.remove(i--);
144         tamL--;
145         break label;
146     }
147 }
148 }
149 }
150 }
151 /*
152  * Metodo que elimina las UEA de un tronco o area de concentracion
153  * que no pueden ser cursadas
154  * debido a que no se cumple con un numero minimo de creditos
155  * acumulados.
156 */
157 private void eliminaPorCreditos(int op){
158     List <UEA> lista=cargaLista(op);
159     if(lista.isEmpty())
160         return;
161     int tamL=lista.size();
162     UEA uea;
163     for(int i=0;i<tamL;i++){
164         uea=lista.get(i);
165         for(String s1 : uea.getSeriacion()){
166             if(s1.startsWith("NC-")){
167                 StringTokenizer st=new StringTokenizer(s1,"",false);
168                 if(Datos.acumulados<Integer.parseInt(st.nextToken("NC-"))){
169                     lista.remove(i--);
170                     tamL--;
171                     break;
172                 }
173             }
174         }
175     }
176 }
177 /*
178  * Metodo que elimina las UEA de un tronco o area de concentracion
179  * que no pueden ser cursadas
180  * debido a que no se ha autorizado cursar dicha UEA.
181 */
182 private void eliminaNoAutorizadas(int op){
183     List <UEA> lista=cargaLista(op);
184     if(lista.isEmpty())
185         return;
186     int tamL=lista.size();
187     UEA uea;
188     for(int i=0;i<tamL;i++){
189         uea=lista.get(i);
190         if(uea.getSeriacion().contains("AU") && !Datos.autorizaciones.

```



```

        contains(uea.getClave())){
188     lista.remove(i--);
189     tamL--;
190 }
191 }
192 }
193 /*
194  * Metodo que elimina las UEA de un tronco o area de concentracion
        que no pueden ser cursadas
195  * debido que ya han sido reprobadas 2 veces en evaluacion global o
        3 veces en evaluacion de
196  * recuperacion.
197  */
198 private void eliminaNoOportunidad(int op){
199     List <UEA> lista=cargaLista(op);
200     if(lista.isEmpty())
201         return;
202     int tamL=lista.size();
203     UEA uea;
204     for(int i=0;i<tamL;i++){
205         uea=lista.get(i);
206         if(uea.getOpGlo()==0 && uea.getOpRec()==0){
207             lista.remove(i--);
208             tamL--;
209         }
210     }
211 }
212 /*
213  * Metodo que verifica si una UEA pertenece al tronco Inter y
        Multidisciplinar.
214  */
215 private boolean esOptInter(UEA uea){
216     for(UEA u : this.troncoInterMultidisciplinar){
217         if(u.getObl_opt().compareTo("OPT")==0 && uea.getClave().
                compareTo(u.getClave())==0)
218             return(true);
219     }
220     return(false);
221 }
222 /*
223  * Metodo que verifica si una UEA pertenece al tronco de
        Integracion.
224  */
225 private boolean esOptIntegracion(UEA uea){
226     for(UEA u : this.troncoIntegracion){
227         if(u.getObl_opt().compareTo("OPT")==0 && uea.getClave().
                compareTo(u.getClave())==0)
228             return(true);
229     }
230     for(UEA u : this.area){
231         if(u.getObl_opt().compareTo("OPT")==0 && uea.getClave().

```

```

232         compareTo(u.getClave())==0)
233         return(true);
234     }
235     return(false);
236 }
237 /*
238  * Metodo que reasigna las prioridades de las UEA del tronco Inter
239  * y Multidisciplinar.
240 */
241 private void reasignaPrioridadesInter() {
242     if(this.prioridadInter <=1)
243         this.prioridadInter--;
244     else if(this.prioridadInter==24)
245         this.prioridadInter=1;
246     else if(this.prioridadInter==29 || this.prioridadInter==34)
247         this.prioridadInter-=5;
248     for(UEA uea : this.troncoInterMultidisciplinar){
249         if(uea.getObl_opt().compareTo("OPT")==0 && !Datos.
250             autorizaciones.contains(uea.getClave()))
251             uea.setPrioridad(this.prioridadInter);
252     }
253 }
254 /*
255  * Metodo que reasigna las prioridades de las UEA del tronco de
256  * Integracion.
257 */
258 private void reasignaPrioridadesIntegracion() {
259     this.prioridadIntegracion--;
260     for(UEA uea : this.troncoIntegracion){
261         if(uea.getObl_opt().compareTo("OPT")==0 && !Datos.
262             autorizaciones.contains(uea.getClave()))
263             uea.setPrioridad(this.prioridadIntegracion);
264     }
265     for(UEA uea : this.area){
266         if(uea.getObl_opt().compareTo("OPT")==0 && !Datos.
267             autorizaciones.contains(uea.getClave()))
268             uea.setPrioridad(this.prioridadIntegracion);
269     }
270 }
271 /*
272  * Metodo que regresa un tronco o area de concentracion para
273  * verificar que UEA pueden ser cursadas.
274 */
275 private List <UEA> cargaLista(int op){
276     if(op==0)
277         return(this.troncoNivelacion);
278     else if(op==1)
279         return(this.troncoGeneral);
280     else if(op==2)
281         return(this.troncoBasicoProfesional);
282     else if(op==3)

```

```

276     return(this.troncoInterMultidisciplinar);
277 else if(op==4)
278     return(this.troncoIntegracion);
279 return(this.area);
280 }
281 /*
282  * Metodo que copia las UEA que no fueron eliminadas del plan de
283   estudios.
284 */
285 private List <UEA> copiaUEA(){
286     List <UEA> lista=new ArrayList <UEA> ();
287     while(!this.troncoNivelacion.isEmpty()){
288         lista.add(this.troncoNivelacion.remove(0));
289     }
290     while(!this.troncoGeneral.isEmpty()){
291         lista.add(this.troncoGeneral.remove(0));
292     }
293     while(!this.troncoBasicoProfesional.isEmpty()){
294         lista.add(this.troncoBasicoProfesional.remove(0));
295     }
296     while(!this.troncoInterMultidisciplinar.isEmpty()){
297         lista.add(this.troncoInterMultidisciplinar.remove(0));
298     }
299     while(!this.troncoIntegracion.isEmpty()){
300         lista.add(this.troncoIntegracion.remove(0));
301     }
302     while(!this.area.isEmpty()){
303         lista.add(this.area.remove(0));
304     }
305     return(lista);
306 }
307 /*
308  * Metodo para crear el tronco de nivelacion.
309 */
310 private void troncoNivelacion(){
311     List <String> seriacion=new ArrayList <String> ();
312     agregaUEA(seriacion,"1100033","Induccion a la Vida Universitaria"
313         ,"OBL",3,64,0);
314     agregaUEA(seriacion,"1111078","Introduccion a la Fisica","OBL"
315         ,4,65,0);
316     agregaUEA(seriacion,"1112026","Taller de Matematicas","OBL"
317         ,7,66,0);
318     agregaUEA(seriacion,"1201008","Comprension de Textos","OBL"
319         ,4,63,0);
320 }
321 /*
322  * Metodo para crear el tronco general.
323 */
324 private void troncoGeneral(){
325     List <String> seriacion=new ArrayList <String> ();
326     seriacion.add("1111078");

```

```

322     seriacion.add("C1112026");
323     agregaUEA(seriacion,"1111079","Cinematica y Dinamica de
        Particulas","OBL",9,59,1);
324     seriacion=new ArrayList <String> ();
325     seriacion.add("1111079");
326     agregaUEA(seriacion,"1111092","Laboratorio de Movimiento de una
        Particula","OBL",3,52,1);
327     agregaUEA(seriacion,"1111081","Dinamica del Cuerpo Rigido","OBL"
        ,9,54,1);
328     seriacion=new ArrayList <String> ();
329     seriacion.add("1111081");
330     seriacion.add("1111092");
331     agregaUEA(seriacion,"1111093","Laboratorio del Cuerpo Rigido y
        Oscilaciones","OBL",3,40,1);
332     seriacion=new ArrayList <String> ();
333     seriacion.add("1111081");
334     seriacion.add("C1112029");
335     agregaUEA(seriacion,"1111083","Introduccion a la Electrostatica y
        Magnetostatica","OBL",9,41,1);
336     seriacion=new ArrayList <String> ();
337     seriacion.add("1112026");
338     agregaUEA(seriacion,"1112013","Complementos de Matematicas","OBL"
        ,9,58,1);
339     agregaUEA(seriacion,"1112027","Introduccion al Calculo","OBL"
        ,6,60,1);
340     seriacion=new ArrayList <String> ();
341     seriacion.add("1112027");
342     agregaUEA(seriacion,"1112028","Calculo Diferencial","OBL",9,53,1)
        ;
343     seriacion=new ArrayList <String> ();
344     seriacion.add("1112028");
345     agregaUEA(seriacion,"1112029","Calculo Integral","OBL",9,49,1);
346     seriacion=new ArrayList <String> ();
347     seriacion.add("1112029");
348     agregaUEA(seriacion,"1112030","Ecuaciones Diferenciales
        Ordinarias","OBL",9,35,1);
349     seriacion=new ArrayList <String> ();
350     seriacion.add("1112028");
351     seriacion.add("1111081");
352     agregaUEA(seriacion,"1113046","Termodinamica","OBL",6,46,1);
353     seriacion=new ArrayList <String> ();
354     agregaUEA(seriacion,"1113084","Estructura Atomica y Enlace
        Quimico","OBL",9,62,1);
355     seriacion=new ArrayList <String> ();
356     seriacion.add("C1113084");
357     agregaUEA(seriacion,"1113085","Laboratorio de Reacciones Quimicas
        ","OBL",3,61,1);
358     seriacion=new ArrayList <String> ();
359     seriacion.add("1113084");
360     agregaUEA(seriacion,"1113086","Estructura y Propiedades de los
        Materiales en Ingenieria","OBL",6,57,1);

```

```

361     seriacion=new ArrayList <String> ();
362     seriacion.add("1113085");
363     seriacion.add("C1113086");
364     agregaUEA(seriacion,"1113087","Laboratorio de Estructura y
        Propiedades de los Materiales","OBL",3,56,1);
365     seriacion=new ArrayList <String> ();
366     seriacion.add("1112013");
367     seriacion.add("1112027");
368     agregaUEA(seriacion,"1151038","Programacion Estructurada","OBL"
        ,7,55,1);
369     seriacion=new ArrayList <String> ();
370     seriacion.add("1151038");
371     seriacion.add("C1112029");
372     agregaUEA(seriacion,"1151039","Metodos Numericos en Ingenieria","
        OBL",7,47,1);
373     seriacion=new ArrayList <String> ();
374     seriacion.add("1112029");
375     agregaUEA(seriacion,"1153001","Probabilidad y Estadistica","OBL"
        ,9,44,1);
376 }
377 /*
378  * Metodo para crear el tronco basico profesional.
379  */
380 private void troncoBasicoProfesional(){
381     List <String> seriacion=new ArrayList <String> ();
382     seriacion.add("1151038");
383     agregaUEA(seriacion,"1112017","Introduccion al Algebra Lineal","
        OBL",9,12,2);
384     agregaUEA(seriacion,"1112033","Matematicas Discretas","OBL"
        ,9,42,2);
385     seriacion=new ArrayList <String> ();
386     seriacion.add("1121037");
387     seriacion.add("1151042");
388     agregaUEA(seriacion,"1121060","Microprocesadores","OBL",9,39,2);
389     seriacion=new ArrayList <String> ();
390     seriacion.add("1151038");
391     agregaUEA(seriacion,"1121037","Diseno Logico","OBL",12,43,2);
392     seriacion=new ArrayList <String> ();
393     seriacion.add("C1121037");
394     agregaUEA(seriacion,"1121040","Laboratorio de Diseno Logico","OBL"
        ",6,36,2);
395     seriacion=new ArrayList <String> ();
396     seriacion.add("1151042");
397     seriacion.add("1153001");
398     agregaUEA(seriacion,"1151041","Almacenamiento y Estructuras de
        Archivos","OBL",8,37,2);
399     seriacion=new ArrayList <String> ();
400     seriacion.add("1151038");
401     agregaUEA(seriacion,"1151042","Algoritmos y Estructuras de Datos"
        ,"OBL",8,50,2);
402     agregaUEA(seriacion,"1151044","Programacion Orientada a Objetos",

```

```

    "OBL" ,8,48,2);
403   seriacion=new ArrayList <String> ();
404   seriacion.add("1151042");
405   seriacion.add("1112017");
406   agregaUEA(seriacion ,"1151051" ," Graficas por Computadora" ,"OBL"
    ,9,5,2);
407   seriacion=new ArrayList <String> ();
408   seriacion.add("1151044");
409   agregaUEA(seriacion ,"1151072" ," Laboratorio de Programacion
    Orientada a Objetos" ,"OBL" ,3,38,2);
410   if(this.areaSel==0)
411       agregaUEADistPrioridades
    (21,32,31,30,28,20,8,22,19,17,33,27,13,14,25,26,23,18);
412   else if(this.areaSel==1)
413       agregaUEADistPrioridades
    (22,27,33,30,23,13,9,14,28,10,32,26,6,18,25,31,19,17);
414   else if(this.areaSel==2)
415       agregaUEADistPrioridades
    (21,33,26,31,28,23,30,17,22,8,32,19,14,13,25,27,20,18);
416   else if(this.areaSel==3)
417       agregaUEADistPrioridades
    (17,32,33,30,28,9,27,23,16,15,31,10,6,14,8,21,22,18);
418   else
419       agregaUEADistPrioridades
    (19,33,32,31,27,10,28,20,18,14,30,9,6,13,25,26,21,17);
420 }
421 /*
422  * Metodo para crear el tronco inter y multidisciplinar.
423  */
424 private void troncoInterMultidisciplinar(){
425     List <String> seriacion=new ArrayList <String> ();
426     seriacion.add("NC.50");
427     agregaUEA(seriacion ,"1100037" ," Introduccion a la Ingenieria" ,"OBL"
    ,6,51,3);
428     seriacion=new ArrayList <String> ();
429     seriacion.add("1100037");
430     agregaUEA(seriacion ,"1100038" ," Introduccion al Desarrollo
    Sustentable" ,"OBL" ,6,45,3);
431     seriacion=new ArrayList <String> ();
432     seriacion.add("1100039");
433     agregaUEA(seriacion ,"1100041" ," Retos del Desarrollo Nacional" ,"
    OBL" ,6,7,3);
434     //ESTUDIOS CULTURALES
435     seriacion=new ArrayList <String> ();
436     seriacion.add("NC.150");
437     agregaUEA(seriacion ,"1100073" ," El Paisaje como Agente de los
    Asentamientos y de la Cultura" ,"OPT" ,6,this.prioridadInter,3);
438     agregaUEA(seriacion ,"1100074" ," Familia y Violencia en el Mexico
    Contemporaneo" ,"OPT" ,6,this.prioridadInter,3);
439     agregaUEA(seriacion ,"1100075" ," Genero y Sexualidad" ,"OPT" ,6,this.
    prioridadInter,3);

```

```

440 agregaUEA(seriacion ,"1100076" ,"Poder y Genero" ,"OPT" ,6 ,this .
    prioridadInter ,3);
441 //FORMACION CUIDADANA
442 agregaUEA(seriacion ,"1100077" ,"Administracion y Economia Politica
    de la Ciencia y Tecnologia" ,"OPT" ,6 ,this .prioridadInter ,3);
443 agregaUEA(seriacion ,"1100078" ,"Derechos Humanos" ,"OPT" ,6 ,this .
    prioridadInter ,3);
444 agregaUEA(seriacion ,"1100079" ,"Economia Mundial" ,"OPT" ,6 ,this .
    prioridadInter ,3);
445 agregaUEA(seriacion ,"1100080" ,"Etica y Valores" ,"OPT" ,6 ,this .
    prioridadInter ,3);
446 agregaUEA(seriacion ,"1100081" ,"Historia Social de Mexico en el
    Siglo XX" ,"OPT" ,6 ,this .prioridadInter ,3);
447 agregaUEA(seriacion ,"1100082" ,"Responsabilidad Social
    Organizacional" ,"OPT" ,6 ,this .prioridadInter ,3);
448 //INDUCCION AL MERCADO LABORAL
449 agregaUEA(seriacion ,"1100083" ,"Comunicacion en Proyectos
    Multidisciplinarios" ,"OPT" ,6 ,this .prioridadInter ,3);
450 agregaUEA(seriacion ,"1100084" ,"Herramientas para el Emprendedor" ,
    "OPT" ,6 ,this .prioridadInter ,3);
451 agregaUEA(seriacion ,"1100085" ,"Insercion Laboral" ,"OPT" ,6 ,this .
    prioridadInter ,3);
452 agregaUEA(seriacion ,"1100086" ,"Planeacion Estrategica" ,"OPT" ,6 ,
    this .prioridadInter ,3);
453 agregaUEA(seriacion ,"1100087" ,"Proyectos de Inversion" ,"OPT" ,6 ,
    this .prioridadInter ,3);
454 //ARTE Y HUMANIDADES
455 agregaUEA(seriacion ,"1100088" ,"Historia del Arte" ,"OPT" ,6 ,this .
    prioridadInter ,3);
456 agregaUEA(seriacion ,"1100089" ,"Taller de Dibujo" ,"OPT" ,6 ,this .
    prioridadInter ,3);
457 agregaUEA(seriacion ,"1100090" ,"Taller de Fotografia" ,"OPT" ,6 ,this
    .prioridadInter ,3);
458 agregaUEA(seriacion ,"1100091" ,"Taller de Teatro" ,"OPT" ,6 ,this .
    prioridadInter ,3);
459 //LENGUAJES FORMALES
460 agregaUEA(seriacion ,"1100092" ,"Divulgacion del Conocimiento" ,"OPT
    " ,6 ,this .prioridadInter ,3);
461 agregaUEA(seriacion ,"1100093" ,"Habilidades Creativas para el
    Ambito Profesional" ,"OPT" ,6 ,this .prioridadInter ,3);
462 agregaUEA(seriacion ,"1100094" ,"Laboratorio de Usabilidad" ,"OPT"
    ,6 ,this .prioridadInter ,3);
463 agregaUEA(seriacion ,"1100095" ,"Narrativa para Medios
    Audiovisuales Digitales" ,"OPT" ,6 ,this .prioridadInter ,3);
464 agregaUEA(seriacion ,"1100096" ,"Taller de Expresion Oral y Escrita
    " ,"OPT" ,6 ,this .prioridadInter ,3);
465 //OTRAS OPTATIVAS INTER Y MULTIDISCIPLINARES
466 agregaUEA(seriacion ,"1100097" ,"Temas Selectos Inter y
    Multidisciplinarios I" ,"OPT" ,6 ,this .prioridadInter ,3);
467 agregaUEA(seriacion ,"1100098" ,"Temas Selectos Inter y
    Multidisciplinarios II" ,"OPT" ,6 ,this .prioridadInter ,3);

```

```

468     seriacion=new ArrayList <String> ();
469     seriacion.add("NC_150");
470     seriacion.add("AU");
471     agregaUEA(seriacion,"1100099","Experiencia Inter y
        Multidisciplinar","OPT",6,this.prioridadInter,3);
472 }
473 /*
474  * Metodo para crear el tronco de integracion.
475  */
476 private void troncoIntegracion(){
477     List <String> seriacion=new ArrayList <String> ();
478     seriacion.add("NC_360");
479     seriacion.add("1100039");
480     agregaUEA(seriacion,"1100103","Seminario de Integracion en
        Ingenieria en Computacion","OBL",3,11,4);
481     seriacion=new ArrayList <String> ();
482     seriacion.add("1100103");
483     seriacion.add("AU");
484     agregaUEA(seriacion,"1100113","Proyecto de Integracion en
        Ingenieria en Computacion I","OBL",18,4,4);
485     //TUTORIALES
486     seriacion=new ArrayList <String> ();
487     seriacion.add("AU");
488     agregaUEA(seriacion,"1100123","Proyecto de Integracion en
        Ingenieria en Computacion II","OPT",18,this.
            prioridadIntegracion,4);
489     seriacion=new ArrayList <String> ();
490     seriacion.add("1100103");
491     seriacion.add("AU");
492     agregaUEA(seriacion,"1100133","Introduccion al Trabajo de
        Investigacion en Ingenieria en Computacion","OPT",6,this.
            prioridadIntegracion,4);
493     seriacion=new ArrayList <String> ();
494     seriacion.add("1100103");
495     agregaUEA(seriacion,"1151028","Trabajo de Investigacion de
        Ingenieria en Computacion","OPT",9,this.prioridadIntegracion
            ,4);
496     //DE MOVILIDAD
497     seriacion=new ArrayList <String> ();
498     seriacion.add("NC_240");
499     seriacion.add("AU");
500     agregaUEA(seriacion,"1100021","Optativa Tecnica de Movilidad I","
        OPT",3,this.prioridadIntegracion,4);
501     agregaUEA(seriacion,"1100022","Optativa Tecnica de Movilidad II",
        "OPT",3,this.prioridadIntegracion,4);
502     agregaUEA(seriacion,"1100023","Optativa Tecnica de Movilidad III"
        ,"OPT",6,this.prioridadIntegracion,4);
503     agregaUEA(seriacion,"1100024","Optativa Tecnica de Movilidad IV",
        "OPT",6,this.prioridadIntegracion,4);
504     agregaUEA(seriacion,"1100025","Optativa Tecnica de Movilidad V",
        "OPT",9,this.prioridadIntegracion,4);

```



```

505 agregaUEA(seriacion ,"1100026","Optativa Tecnica de Movilidad VI" ,
      "OPT" ,9, this.prioridadIntegracion ,4);
506 //OTRAS OPTATIVAS DE INTEGRACION
507 seriacion=new ArrayList <String> ();
508 seriacion.add("1112029");
509 seriacion.add("1112013");
510 agregaUEA(seriacion ,"1112005","Calculo de Varias Variables" ,"OPT"
      ,12, this.prioridadIntegracion ,4);
511 seriacion=new ArrayList <String> ();
512 seriacion.add("1112005");
513 agregaUEA(seriacion ,"1112016","Variable Compleja" ,"OPT" ,6, this.
      prioridadIntegracion ,4);
514 seriacion=new ArrayList <String> ();
515 seriacion.add("1153001");
516 seriacion.add("1112040");
517 agregaUEA(seriacion ,"1122009","Analisis de Senales" ,"OPT" ,9, this.
      prioridadIntegracion ,4);
518 seriacion=new ArrayList <String> ();
519 seriacion.add("1122009");
520 agregaUEA(seriacion ,"1122012","Procesamiento Digital de Senales" ,
      "OPT" ,9, this.prioridadIntegracion ,4);
521 agregaUEA(seriacion ,"1122026","Senales Aleatorias" ,"OPT" ,9, this.
      prioridadIntegracion ,4);
522 seriacion=new ArrayList <String> ();
523 seriacion.add("1122026");
524 agregaUEA(seriacion ,"1122028","Teletrafico" ,"OPT" ,9, this.
      prioridadIntegracion ,4);
525 seriacion=new ArrayList <String> ();
526 seriacion.add("NC_300");
527 agregaUEA(seriacion ,"1135062","Evaluacion Ambiental de
      Tecnologias" ,"OPT" ,9, this.prioridadIntegracion ,4);
528 agregaUEA(seriacion ,"1135096","Prevencion y Minimizacion de la
      Contaminacion Ambiental" ,"OPT" ,9, this.prioridadIntegracion ,4);
529 agregaUEA(seriacion ,"1151032","Temas Selectos de Ingenieria en
      Computacion I" ,"OPT" ,9, this.prioridadIntegracion ,4);
530 agregaUEA(seriacion ,"1151033","Temas Selectos de Ingenieria en
      Computacion II" ,"OPT" ,9, this.prioridadIntegracion ,4);
531 seriacion=new ArrayList <String> ();
532 seriacion.add("NC_150");
533 agregaUEA(seriacion ,"1136005","Retos Ambientales" ,"OPT" ,6, this.
      prioridadIntegracion ,4);
534 seriacion=new ArrayList <String> ();
535 seriacion.add("1151039");
536 seriacion.add("1112030");
537 agregaUEA(seriacion ,"1151009","Elemento Finito" ,"OPT" ,9, this.
      prioridadIntegracion ,4);
538 seriacion=new ArrayList <String> ();
539 seriacion.add("1154002");
540 agregaUEA(seriacion ,"1151029","Operacion de Centros de Datos" ,"
      OPT" ,9, this.prioridadIntegracion ,4);
541 agregaUEA(seriacion ,"1154038","Estudio del Metodo de Trabajo" ,"

```

```

OPT" ,6, this.prioridadIntegracion ,4);
542 seriacion=new ArrayList <String> ();
543 seriacion.add("1154002");
544 seriacion.add("NC_250");
545 agregaUEA(seriacion , "1154022" ," Desarrollo Organizacional" ,"OPT"
,9, this.prioridadIntegracion ,4);
546 agregaUEA(seriacion , "1154035" ," Planeacion Estrategica" ,"OPT" ,9,
this.prioridadIntegracion ,4);
547 seriacion=new ArrayList <String> ();
548 seriacion.add("1151048");
549 agregaUEA(seriacion , "1151030" ," Taller de Productos de Software" ,"
OPT" ,9, this.prioridadIntegracion ,4);
550 agregaUEA(seriacion , "1151034" ," Calidad de Software" ,"OPT" ,9, this.
prioridadIntegracion ,4);
551 agregaUEA(seriacion , "1151035" ," Auditoria Informatica" ,"OPT" ,9,
this.prioridadIntegracion ,4);
552 seriacion=new ArrayList <String> ();
553 seriacion.add("1153001");
554 agregaUEA(seriacion , "1154029" ," Analisis y Diseno de Experimentos
en Ingenieria" ,"OPT" ,9, this.prioridadIntegracion ,4);
555 seriacion=new ArrayList <String> ();
556 seriacion.add("1153001");
557 seriacion.add("1154001");
558 agregaUEA(seriacion , "1153005" ," Analisis de Decisiones II" ,"OPT"
,9, this.prioridadIntegracion ,4);
559 seriacion=new ArrayList <String> ();
560 seriacion.add("NC_200");
561 agregaUEA(seriacion , "1154001" ," Analisis de Decisiones I" ,"OPT" ,9,
this.prioridadIntegracion ,4);
562 seriacion=new ArrayList <String> ();
563 seriacion.add("NC_100");
564 agregaUEA(seriacion , "1154002" ," Organizacion Industrial" ,"OPT" ,9,
this.prioridadIntegracion ,4);
565 seriacion=new ArrayList <String> ();
566 seriacion.add("NC_320");
567 agregaUEA(seriacion , "1154015" ," Administracion de Proyectos" ,"OPT"
,6, this.prioridadIntegracion ,4);
568 seriacion=new ArrayList <String> ();
569 seriacion.add("1154042");
570 agregaUEA(seriacion , "1154017" ," Sistemas de Gestion de la Calidad"
,"OPT" ,9, this.prioridadIntegracion ,4);
571 agregaUEA(seriacion , "1154018" ," Administracion de la Calidad" ,"OPT
" ,6, this.prioridadIntegracion ,4);
572 seriacion=new ArrayList <String> ();
573 seriacion.add("1153005");
574 agregaUEA(seriacion , "1154025" ," Ingenieria Financiera" ,"OPT" ,9,
this.prioridadIntegracion ,4);
575 seriacion=new ArrayList <String> ();
576 seriacion.add("1153001");
577 seriacion.add("1154038");
578 agregaUEA(seriacion , "1154042" ," Control de Calidad y Confiabilidad

```

```

    " , "OPT" , 8 , this . prioridadIntegracion , 4 ) ;
579   seriacion = new ArrayList <String> ( ) ;
580   seriacion . add ( " NC_400 " ) ;
581   agregaUEA ( seriacion , " 1154054 " , " Habilidades Gerenciales " , " OPT " , 9 ,
        this . prioridadIntegracion , 4 ) ;
582 }
583 /*
584  * Metodo para crear el area de concentracion Sistemas de
        Informacion .
585  */
586 private void areaSistemasInformacion ( ) {
587   List <String> seriacion = new ArrayList <String> ( ) ;
588   seriacion . add ( " 1151048 " ) ;
589   seriacion . add ( " 1151047 " ) ;
590   seriacion . add ( " 1121038 " ) ;
591   agregaUEA ( seriacion , " 1151054 " , " Arquitectura e Integracion de
        Aplicaciones Empresariales " , " OPT * " , 7 , 16 , 5 ) ;
592   agregaUEA ( seriacion , " 1151057 " , " Programacion Orientada a Servicios
        " , " OPT * " , 6 , 15 , 5 ) ;
593   seriacion = new ArrayList <String> ( ) ;
594   seriacion . add ( " 1151048 " ) ;
595   seriacion . add ( " 1151047 " ) ;
596   agregaUEA ( seriacion , " 1151055 " , " Administracion de Proyectos de
        Software " , " OPT " , 8 , this . prioridadIntegracion , 5 ) ;
597   seriacion = new ArrayList <String> ( ) ;
598   seriacion . add ( " 1151048 " ) ;
599   agregaUEA ( seriacion , " 1151056 " , " Patrones de Diseno de Software " , "
        OPT * " , 8 , 9 , 5 ) ;
600   agregaUEA ( seriacion , " 1151071 " , " Temas Selectos de Sistemas de
        Informacion " , " OPT " , 9 , this . prioridadIntegracion , 5 ) ;
601   seriacion = new ArrayList <String> ( ) ;
602   seriacion . add ( " 1151054 " ) ;
603   seriacion . add ( " 1151056 " ) ;
604   seriacion . add ( " 1151057 " ) ;
605   seriacion . add ( " 1151074 " ) ;
606   agregaUEA ( seriacion , " 1151058 " , " Taller de Desarrollo de
        Aplicaciones Web " , " OPT * " , 7 , 6 , 5 ) ;
607   seriacion = new ArrayList <String> ( ) ;
608   seriacion . add ( " 1151054 " ) ;
609   agregaUEA ( seriacion , " 1151059 " , " Integracion de Servicios en
        Aplicaciones Empresariales " , " OPT " , 7 , this . prioridadIntegracion
        , 5 ) ;
610   seriacion = new ArrayList <String> ( ) ;
611   seriacion . add ( " 1151044 " ) ;
612   agregaUEA ( seriacion , " 1151060 " , " Temas Selectos de Lenguajes de
        Programacion " , " OPT " , 9 , this . prioridadIntegracion , 5 ) ;
613   seriacion = new ArrayList <String> ( ) ;
614   seriacion . add ( " 1151046 " ) ;
615   seriacion . add ( " 1151047 " ) ;
616   agregaUEA ( seriacion , " 1151074 " , " Bases de Datos Distribuidas " , " OPT *
        " , 8 , 10 , 5 ) ;

```

```

617 }
618 /*
619  * Metodo para crear el area de concentracion Algoritmos e
        Inteligencia Artificial.
620 */
621 private void areaInteligencia() {
622     List <String> seriacion=new ArrayList <String> ();
623     seriacion.add("1151040");
624     agregaUEA(seriacion,"1112035","Combinatoria","OPT*",9,15,5);
625     agregaUEA(seriacion,"1151061","Complejidad Computacional","OPT*"
        ,6,20,5);
626     agregaUEA(seriacion,"1151067","Temas Selectos de Algoritmos","OPT
        ",9,this.prioridadIntegracion,5);
627     seriacion=new ArrayList <String> ();
628     seriacion.add("1112017");
629     seriacion.add("1151040");
630     agregaUEA(seriacion,"1151065","Teoria de Juegos","OPT",9,this.
        prioridadIntegracion,5);
631     agregaUEA(seriacion,"1151066","Geometria Computacional","OPT",9,
        this.prioridadIntegracion,5);
632     seriacion=new ArrayList <String> ();
633     seriacion.add("1112034");
634     agregaUEA(seriacion,"1112037","Teoria de la Computacion","OPT",9,
        this.prioridadIntegracion,5);
635     agregaUEA(seriacion,"1112038","Temas Selectos de Matematicas
        Discretas","OPT",9,this.prioridadIntegracion,5);
636     seriacion=new ArrayList <String> ();
637     seriacion.add("1151062");
638     agregaUEA(seriacion,"1151045","Temas Selectos de Inteligencia
        Artificial","OPT",9,this.prioridadIntegracion,5);
639     seriacion=new ArrayList <String> ();
640     seriacion.add("1151042");
641     seriacion.add("1153001");
642     agregaUEA(seriacion,"1151062","Inteligencia Artificial","OPT*"
        ,9,21,5);
643     agregaUEA(seriacion,"1151063","Inteligencia Computacional","OPT*"
        ,9,16,5);
644     seriacion=new ArrayList <String> ();
645     seriacion.add("1151061");
646     seriacion.add("1151062");
647     seriacion.add("1151063");
648     seriacion.add("1112035");
649     agregaUEA(seriacion,"1151064","Taller de Analisis y Diseno de
        Algoritmos","OPT*",3,8,5);
650     seriacion=new ArrayList <String> ();
651     seriacion.add("1151051");
652     agregaUEA(seriacion,"1151068","Temas Selectos de Graficas por
        Computadora","OPT",9,this.prioridadIntegracion,5);
653     seriacion=new ArrayList <String> ();
654     seriacion.add("1152001");
655     seriacion.add("1153001");

```

```

656     agregaUEA(seriacion ,"1152002","Investigacion de Operaciones II","
        OPT" ,9,this.prioridadIntegracion ,5);
657 }
658 /*
659  * Metodo para crear el area de concentracion Seguridad y Redes de
        Computadoras.
660  */
661 private void areaRedes(){
662     List <String> seriacion=new ArrayList <String> ();
663     seriacion.add("1151040");
664     agregaUEA(seriacion ,"1112036","Criptografia","OPT*" ,9,9,5);
665     seriacion=new ArrayList <String> ();
666     seriacion.add("1121043");
667     agregaUEA(seriacion ,"1121044","Seguridad en Redes","OPT*" ,6,16,5)
        ;
668     agregaUEA(seriacion ,"1121046","Temas Selectos de Redes de
        Computadoras I","OPT" ,9,this.prioridadIntegracion ,5);
669     agregaUEA(seriacion ,"1121047","Temas Selectos de Redes de
        Computadoras II","OPT" ,9,this.prioridadIntegracion ,5);
670     agregaUEA(seriacion ,"1121057","Redes Inalambricas","OPT*" ,6,15,5)
        ;
671     agregaUEA(seriacion ,"1121058","Integracion de Redes de Voz y
        Datos","OPT*" ,9,10,5);
672     seriacion=new ArrayList <String> ();
673     seriacion.add("1121044");
674     seriacion.add("1121057");
675     seriacion.add("1121058");
676     agregaUEA(seriacion ,"1151069","Seguridad en los Sistemas de
        Informacion","OPT*" ,6,6,5);
677     seriacion=new ArrayList <String> ();
678     seriacion.add("1151069");
679     agregaUEA(seriacion ,"1151070","Temas Selectos de Seguridad en los
        Sistemas de Informacion","OPT" ,9,this.prioridadIntegracion ,5)
        ;
680 }
681 /*
682  * Metodo para crear el area de concentracion Sistemas Embebidos.
683  */
684 private void areaSistemasEmbebidos(){
685     List <String> seriacion=new ArrayList <String> ();
686     seriacion.add("C1121034");
687     agregaUEA(seriacion ,"1121032","Laboratorio de Sistemas Digitales
        con Microcontroladores","OPT*" ,3,19,5);
688     seriacion=new ArrayList <String> ();
689     seriacion.add("1121060");
690     seriacion.add("1124052");
691     agregaUEA(seriacion ,"1121034","Sistemas Digitales con
        Microcontroladores","OPT*" ,9,20,5);
692     seriacion=new ArrayList <String> ();
693     seriacion.add("C1121025");
694     agregaUEA(seriacion ,"1121039","Laboratorio de Arquitectura de

```

```

        Computadoras" ,"OPT" ,3 ,this.prioridadIntegracion ,5);
695   seriacion=new ArrayList <String> ();
696   seriacion.add("1121025");
697   agregaUEA(seriacion ,"1121053" ,"Arquitecturas Paralelas de
        Computadoras" ,"OPT*" ,9,26,5);
698   seriacion=new ArrayList <String> ();
699   seriacion.add("1151018");
700   seriacion.add("1151073");
701   seriacion.add("1121032");
702   seriacion.add("1121053");
703   agregaUEA(seriacion ,"1121054" ,"Taller de Sistemas Embebidos" ,"OPT
        *" ,6,13,5);
704   seriacion=new ArrayList <String> ();
705   seriacion.add("1121054");
706   agregaUEA(seriacion ,"1121056" ,"Temas Selectos de Sistemas
        Embebidos" ,"OPT" ,9 ,this.prioridadIntegracion ,5);
707   seriacion=new ArrayList <String> ();
708   seriacion.add("1121060");
709   agregaUEA(seriacion ,"1151073" ,"Programacion de Sistemas" ,"OPT*"
        ,9,25,5);
710 }
711 /*
712  * Metodo para crear el area de concentracion Mecatronica.
713  */
714 private void areaMecatronica(){
715     List <String> seriacion=new ArrayList <String> ();
716     seriacion.add("C1124043");
717     seriacion.add("1151063");
718     seriacion.add("1121032");
719     agregaUEA(seriacion ,"1100034" ,"Taller de Mecatronica" ,"OPT*"
        ,6,8,5);
720     seriacion=new ArrayList <String> ();
721     seriacion.add("1100034");
722     agregaUEA(seriacion ,"1100035" ,"Temas Selectos de Ingenieria
        Mecatronica" ,"OPT" ,9 ,this.prioridadIntegracion ,5);
723     seriacion=new ArrayList <String> ();
724     seriacion.add("C1121034");
725     agregaUEA(seriacion ,"1121032" ,"Laboratorio de Sistemas Digitales
        con Microcontroladores" ,"OPT*" ,3,22,5);
726     seriacion=new ArrayList <String> ();
727     seriacion.add("1121060");
728     seriacion.add("1124052");
729     agregaUEA(seriacion ,"1121034" ,"Sistemas Digitales con
        Microcontroladores" ,"OPT*" ,9,23,5);
730     seriacion=new ArrayList <String> ();
731     seriacion.add("1124001");
732     seriacion.add("1124005");
733     agregaUEA(seriacion ,"1123040" ,"Circuitos Electronicos I" ,"OPT" ,9 ,
        this.prioridadIntegracion ,5);
734     seriacion=new ArrayList <String> ();
735     seriacion.add("1123040");

```

```

736     seriacion.add("1123045");
737     agregaUEA(seriacion,"1123041","Circuitos Electronicos II","OPT"
        ,9,this.prioridadIntegracion,5);
738     seriacion=new ArrayList <String> ();
739     seriacion.add("1121034");
740     seriacion.add("1123041");
741     seriacion.add("1123046");
742     agregaUEA(seriacion,"1123043","Diseno de Sistemas Electronicos",
        "OPT",9,this.prioridadIntegracion,5);
743     seriacion=new ArrayList <String> ();
744     seriacion.add("C1123040");
745     agregaUEA(seriacion,"1123045","Laboratorio de Circuitos
        Electronicos I","OPT",3,this.prioridadIntegracion,5);
746     seriacion=new ArrayList <String> ();
747     seriacion.add("C1123041");
748     agregaUEA(seriacion,"1123046","Laboratorio de Circuitos
        Electronicos II","OPT",3,this.prioridadIntegracion,5);
749     seriacion=new ArrayList <String> ();
750     seriacion.add("C1123043");
751     agregaUEA(seriacion,"1123048","Laboratorio de Diseno de Sistemas
        Electronicos","OPT",3,this.prioridadIntegracion,5);
752     seriacion=new ArrayList <String> ();
753     seriacion.add("C1112030");
754     agregaUEA(seriacion,"1124001","Circuitos Electricos I","OPT",9,
        this.prioridadIntegracion,5);
755     seriacion=new ArrayList <String> ();
756     seriacion.add("1124001");
757     seriacion.add("1112040");
758     agregaUEA(seriacion,"1124003","Circuitos Electricos II","OPT",9,
        this.prioridadIntegracion,5);
759     seriacion=new ArrayList <String> ();
760     seriacion.add("C1124001");
761     agregaUEA(seriacion,"1124005","Laboratorio de Circuitos
        Electricos I","OPT",3,this.prioridadIntegracion,5);
762     seriacion=new ArrayList <String> ();
763     seriacion.add("1124052");
764     agregaUEA(seriacion,"1124043","Automatizacion Industrial","OPT*
        ",9,16,5);
765     seriacion=new ArrayList <String> ();
766     seriacion.add("1123043");
767     seriacion.add("1124049");
768     seriacion.add("1122012");
769     agregaUEA(seriacion,"1124045","Control Digital","OPT",9,this.
        prioridadIntegracion,5);
770     seriacion=new ArrayList <String> ();
771     seriacion.add("C1124003");
772     seriacion.add("1124005");
773     agregaUEA(seriacion,"1124048","Laboratorio de Circuitos
        Electricos II","OPT",3,this.prioridadIntegracion,5);
774     seriacion=new ArrayList <String> ();
775     seriacion.add("1121037");

```

```

776     seriacion.add("C1124050");
777     agregaUEA(seriacion,"1124049","Laboratorio de Control","OPT",3,
        this.prioridadIntegracion,5);
778     seriacion=new ArrayList <String> ();
779     seriacion.add("1124003");
780     agregaUEA(seriacion,"1124050","Teoria de Control","OPT",9,this.
        prioridadIntegracion,5);
781     seriacion=new ArrayList <String> ();
782     seriacion.add("1100034");
783     agregaUEA(seriacion,"1124051","Temas Selectos de Robotica","OPT"
        ,9,this.prioridadIntegracion,5);
784     seriacion=new ArrayList <String> ();
785     seriacion.add("C1133060");
786     agregaUEA(seriacion,"1133009","Laboratorio de Mecanismos","OPT"
        ,3,this.prioridadIntegracion,5);
787     seriacion=new ArrayList <String> ();
788     seriacion.add("1133060");
789     agregaUEA(seriacion,"1133024","Dinamica de Maquinas","OPT",9,this
        .prioridadIntegracion,5);
790     agregaUEA(seriacion,"1133032","Diseno de Mecanismos","OPT",9,this
        .prioridadIntegracion,5);
791     seriacion=new ArrayList <String> ();
792     seriacion.add("1133061");
793     agregaUEA(seriacion,"1133059","Manufactura Asistida por
        Computadora","OPT",9,this.prioridadIntegracion,5);
794     seriacion=new ArrayList <String> ();
795     seriacion.add("1133061");
796     seriacion.add("1111081");
797     agregaUEA(seriacion,"1133060","Mecanismos","OPT",9,this.
        prioridadIntegracion,5);
798     seriacion=new ArrayList <String> ();
799     seriacion.add("1112013");
800     seriacion.add("NC_150");
801     agregaUEA(seriacion,"1133061","Dibujo Mecanico Asistido por
        Computadora","OPT",9,this.prioridadIntegracion,5);
802     seriacion=new ArrayList <String> ();
803     seriacion.add("1151042");
804     seriacion.add("1153001");
805     agregaUEA(seriacion,"1151063","Inteligencia Computacional","OPT*"
        ,9,15,5);
806 }
807 /*
808  * Metodo para agregar UEA con distintas prioridades, segun el area
        de concentracion seleccionada,
809  * al tronco basico profesional y al tronco inter y
        multidisciplinar.
810  */
811 private void agregaUEADistPrioridades(int... pr){
812     int i=0;
813     List <String> seriacion=new ArrayList <String> ();
814     seriacion.add("1112033");

```



```

815 agregaUEA(seriacion,"1112034","Lenguajes y Automatas","OBL",9,pr[
      i++],2);
816 seriacion=new ArrayList <String> ();
817 seriacion.add("1112030");
818 agregaUEA(seriacion,"1112040","Transformada de Laplace y Analisis
      de Fourier","OBL",9,pr[i++],2);
819 seriacion=new ArrayList <String> ();
820 seriacion.add("1121060");
821 agregaUEA(seriacion,"1121025","Arquitectura de Computadoras","OBL
      ",9,pr[i++],2);
822 seriacion=new ArrayList <String> ();
823 seriacion.add("C1121060");
824 seriacion.add("1121040");
825 agregaUEA(seriacion,"1121033","Laboratorio de Sistemas Digitales
      con Microprocesadores","OBL",3,pr[i++],2);
826 seriacion=new ArrayList <String> ();
827 seriacion.add("1112040");
828 seriacion.add("1121060");
829 agregaUEA(seriacion,"1121038","Fundamentos de Redes de
      Computadoras","OBL",9,pr[i++],2);
830 seriacion=new ArrayList <String> ();
831 seriacion.add("1121038");
832 agregaUEA(seriacion,"1121043","Diseno y Administracion de Redes
      de Computadoras","OBL",12,pr[i++],2);
833 seriacion=new ArrayList <String> ();
834 seriacion.add("1111083");
835 seriacion.add("1113086");
836 agregaUEA(seriacion,"1124052","Microelectronica","OBL",9,pr[i
      ++],2);
837 seriacion=new ArrayList <String> ();
838 seriacion.add("1121025");
839 seriacion.add("1121038");
840 agregaUEA(seriacion,"1151018","Sistemas Operativos","OBL",9,pr[i
      ++],2);
841 seriacion=new ArrayList <String> ();
842 seriacion.add("1152001");
843 seriacion.add("1151042");
844 seriacion.add("1112033");
845 agregaUEA(seriacion,"1151040","Analisis y Diseno de Algoritmos","
      OBL",9,pr[i++],2);
846 seriacion=new ArrayList <String> ();
847 seriacion.add("1151018");
848 seriacion.add("1151047");
849 agregaUEA(seriacion,"1151046","Sistemas Distribuidos","OBL",9,pr[
      i++],2);
850 seriacion=new ArrayList <String> ();
851 seriacion.add("1151072");
852 seriacion.add("1151041");
853 agregaUEA(seriacion,"1151047","Bases de Datos","OBL",12,pr[i
      ++],2);
854 agregaUEA(seriacion,"1151048","Analisis y Diseno de Sistemas de

```

```

    Informacion" ,"OBL" ,8 ,pr [ i++ ],2);
855   seriacion=new ArrayList <String> ();
856   seriacion.add("1121060");
857   seriacion.add("1112034");
858   agregaUEA(seriacion ,"1151049" ,"Compiladores" ,"OBL" ,9 ,pr [ i++ ],2);
859   seriacion=new ArrayList <String> ();
860   seriacion.add("C1100040");
861   agregaUEA(seriacion ,"1151050" ,"Etica y Legislacion Informatica" ,"
      OBL" ,6 ,pr [ i++ ],2);
862   seriacion=new ArrayList <String> ();
863   seriacion.add("1151072");
864   agregaUEA(seriacion ,"1151052" ,"Interaccion Humano-Computadora" ,"
      OBL" ,7 ,pr [ i++ ],2);
865   seriacion=new ArrayList <String> ();
866   seriacion.add("1151039");
867   agregaUEA(seriacion ,"1152001" ,"Investigacion de Operaciones I" ,"
      OBL" ,9 ,pr [ i++ ],2);
868   seriacion=new ArrayList <String> ();
869   seriacion.add("NC_280");
870   agregaUEA(seriacion ,"1100040" ,"Taller de Planeacion y Ejecucion
      de Proyectos" ,"OBL" ,6 ,pr [ i++ ],3);
871   seriacion=new ArrayList <String> ();
872   seriacion.add("1100040");
873   agregaUEA(seriacion ,"1100039" ,"Innovacion" ,"OBL" ,6 ,pr [ i++ ],3);
874 }
875 /*
876  * Metodo para agregar una UEA a algun tronco o area de
      concentracion.
877  */
878 private void agregaUEA(List <String> seriacion ,String clave ,String
      nombre ,String obl_opt ,int cred ,int prio ,int op){
879   UEA uea=new UEA();
880   uea.setClave(clave);
881   uea.setNombre(nombre);
882   uea.setObl_opt(obl_opt);
883   uea.setNumCreditos(cred);
884   uea.setPrioridad(prio);
885   uea.setSeriacion(seriacion);
886   if(op==0)
887     this.troncoNivelacion.add(uea);
888   else if(op==1)
889     this.troncoGeneral.add(uea);
890   else if(op==2)
891     this.troncoBasicoProfesional.add(uea);
892   else if(op==3)
893     this.troncoInterMultidisciplinar.add(uea);
894   else if(op==4)
895     this.troncoIntegracion.add(uea);
896   else
897     this.area.add(uea);
898 }

```

Archivo: Recomendacion.java

```
1 package com.proyecto.proyecto;
2
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.graphics.Color;
6 import android.os.Bundle;
7 import android.view.Gravity;
8 import android.view.View;
9 import android.widget.Button;
10 import android.widget.TableLayout;
11 import android.widget.TableRow;
12 import android.widget.TextView;
13
14 public class Recomendacion extends Activity{
15     private TableLayout resultado;
16     private TableLayout optativas;
17     private TextView matr;
18     private TextView nombre;
19     private TextView credAcu;
20     private TextView credMinPer;
21     private TextView credMaxPer;
22     private TextView credMinRec;
23     private TextView credMaxRec;
24     private TextView acumularias;
25     private TextView fin;
26     private TextView textoOpt;
27     private Button nuevo;
28     private Button personalizar;
29     private Button comentar;
30     private boolean flag;
31
32     @Override
33     protected void onCreate(Bundle savedInstanceState){
34         super.onCreate(savedInstanceState);
35         setContentView(R.layout.recomendacion);
36         this.resultado=(TableLayout)findViewById(R.id.UEA);
37         this.resultado.setStretchAllColumns(true);
38         this.resultado.setShrinkAllColumns(true);
39         this.optativas=(TableLayout)findViewById(R.id.UEA2);
40         this.optativas.setStretchAllColumns(true);
41         this.optativas.setShrinkAllColumns(true);
42         this.matr=(TextView)findViewById(R.id.txt_matr);
43         this.matr.setTextSize(8);
44
45         this.nombre=(TextView)findViewById(R.id.txt_nombre);
46         this.nombre.setTextSize(8);
```

```

47
48     this.credAcu=(TextView) findViewById(R.id.txt_credAcu);
49     this.credAcu.setTextSize(8);
50
51     this.credMinPer=(TextView) findViewById(R.id.txt_credMinPer);
52     this.credMinPer.setTextSize(8);
53     this.credMaxPer=(TextView) findViewById(R.id.txt_credMaxPer);
54     this.credMaxPer.setTextSize(8);
55
56     this.credMinRec=(TextView) findViewById(R.id.txt_credMinRec);
57     this.credMinRec.setTextSize(8);
58     this.credMaxRec=(TextView) findViewById(R.id.txt_credMaxRec);
59     this.credMaxRec.setTextSize(8);
60
61     this.acumularias=(TextView) findViewById(R.id.acumularias);
62     this.acumularias.setTextSize(8);
63
64     this.fin=(TextView) findViewById(R.id.fin);
65     this.fin.setTextSize(8);
66
67     this.textoOpt=(TextView) findViewById(R.id.nota2);
68     this.textoOpt.setTextSize(9);
69
70     this.nuevo=(Button) findViewById(R.id.btn_nuevo);
71     this.nuevo.setOnClickListener(new View.OnClickListener(){
72         @Override
73         public void onClick(View v){
74             Intent i=new Intent(Recomendacion.this, MainActivity.class);
75             i.setFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
76             startActivity(i);
77         }
78     });
79
80     this.personalizar=(Button) findViewById(R.id.btn_personalizar);
81     this.personalizar.setOnClickListener(new View.OnClickListener(){
82         @Override
83         public void onClick(View v){
84             Intent i=new Intent("com.proyecto.proyecto.PERSONALIZACION");
85             startActivity(i);
86         }
87     });
88
89     this.comentar=(Button) findViewById(R.id.btn_comentar);
90     this.comentar.setOnClickListener(new View.OnClickListener(){
91         @Override
92         public void onClick(View v){
93             Intent i=new Intent("com.proyecto.proyecto.COMENTAR");
94             startActivity(i);
95         }
96     });
97     this.flag=true;

```

```

98     presentacion();
99 }
100 /*
101  * Metodo que despliega en pantalla el calculo de UEA final.
102  */
103 private void presentacion() {
104     boolean opt=false;
105     this.matr.setText(Datos.matricula);
106     this.nombre.setText(Datos.nombre + " " + Datos.apP + " " + Datos.
107         apM);
108     this.credAcu.setText("" + Datos.acumulados);
109     this.credMinPer.setText("0");
110     this.credMaxPer.setText("" + Datos.creditosMax);
111     this.credMinRec.setText("" + Datos.creditosMinRec);
112     this.credMaxRec.setText("" + Datos.creditosMaxRec);
113     cabecera(false);
114     for(UEA uea : Datos.listaFinalObl)
115         agregaFila(uea.getClave(), uea.getNombre(), "" + uea.
116             getNumCreditos(), false);
117     for(UEA uea : Datos.listaFinalOpt) {
118         if(Datos.sumaCred+uea.getNumCreditos()<=Datos.creditosMaxRec) {
119             if(!opt) {
120                 this.textoOpt.setText("Mas alguna(s) UEA optativa(s):");
121                 cabecera(true);
122                 opt=true;
123             }
124             agregaFila(uea.getClave(), uea.getNombre(), "" + uea.
125                 getNumCreditos(), true);
126         }
127     }
128     this.acumularias.setText("\nN̄mero de cr̄ditos que acumular̄as
129     : " +
130     ((opt) ? (Datos.acumulados+Datos.sumaCred) + " m̄s los
131     cr̄ditos de la(s) optativa(s)" :
132     (Datos.acumulados+Datos.sumaCred) + ""));
133     this.fin.setText("¿Podrias terminar este trimestre?: " + ((Datos
134     .fin) ? "Si":"No"));
135 }
136 /*
137  * Metodo para agregar una cabecera a la tabla final.
138  */
139 private void cabecera(boolean opt) {
140     TextView obl_opt=new TextView(this);
141     obl_opt.setText((opt) ? "Optativas" : "Obligatorias");
142     obl_opt.setTextSize(10);
143     obl_opt.setTextColor(Color.rgb(2,34,75));
144     obl_opt.setGravity(Gravity.CENTER.HORIZONTAL);
145     obl_opt.setBackgroundResource(android.R.color.holo_red_dark);
146     obl_opt.setLayoutParams(new TableRow.LayoutParams(TableRow.
147         LayoutParams.WRAP_CONTENT, TableRow.LayoutParams.WRAP_CONTENT))
148     ;

```

```

141     TableRow fila=new TableRow(this);
142     fila.addView(new TextView(this));
143     fila.addView(obl_opt);
144     fila.addView(new TextView(this));
145     if(opt)
146         this.optativas.addView(fila);
147     else
148         this.resultado.addView(fila);
149     TextView clave=new TextView(this);
150     clave.setText("Clave");
151     clave.setTextSize(8);
152     clave.setTextColor(Color.WHITE);
153     clave.setGravity(Gravity.CENTER_HORIZONTAL);
154     clave.setBackgroundResource(android.R.color.holo_red_dark);
155     clave.setLayoutParams(new TableRow.LayoutParams(TableRow.
        LayoutParams.WRAP_CONTENT, TableRow.LayoutParams.WRAP_CONTENT))
        ;
156     TextView nombre=new TextView(this);
157     nombre.setText("Nombre");
158     nombre.setTextSize(8);
159     nombre.setTextColor(Color.WHITE);
160     nombre.setGravity(Gravity.CENTER_HORIZONTAL);
161     nombre.setBackgroundResource(android.R.color.holo_red_dark);
162     nombre.setLayoutParams(new TableRow.LayoutParams(TableRow.
        LayoutParams.WRAP_CONTENT, TableRow.LayoutParams.WRAP_CONTENT))
        ;
163     TextView cred=new TextView(this);
164     cred.setText("Cr ditos");
165     cred.setTextSize(8);
166     cred.setTextColor(Color.WHITE);
167     cred.setGravity(Gravity.CENTER_HORIZONTAL);
168     cred.setBackgroundResource(android.R.color.holo_red_dark);
169     cred.setLayoutParams(new TableRow.LayoutParams(TableRow.
        LayoutParams.WRAP_CONTENT, TableRow.LayoutParams.WRAP_CONTENT))
        ;
170     fila=new TableRow(this);
171     fila.addView(clave);
172     fila.addView(nombre);
173     fila.addView(cred);
174     if(opt)
175         this.optativas.addView(fila);
176     else
177         this.resultado.addView(fila);
178 }
179 /*
180  * Metodo para agregar una UEA con prioridad unica a la tabla final
181  */
182 private void agregaFila(String cl, String nom, String n_cred, boolean
    opt){
183     TextView clave=new TextView(this);

```

```

184     clave.setText(c1);
185     clave.setTextSize(7);
186     clave.setTextColor(Color.BLACK);
187     clave.setGravity(Gravity.CENTER_HORIZONTAL);
188     clave.setLayoutParams(new TableRow.LayoutParams(TableRow.
        LayoutParams.WRAP_CONTENT, TableRow.LayoutParams.WRAP_CONTENT))
        ;
189     clave.setPadding(0,1,0,0);
190     TextView nombre=new TextView(this);
191     nombre.setText(nom);
192     nombre.setTextSize(7);
193     nombre.setTextColor(Color.BLACK);
194     nombre.setGravity(Gravity.CENTER_HORIZONTAL);
195     nombre.setLayoutParams(new TableRow.LayoutParams(TableRow.
        LayoutParams.WRAP_CONTENT, TableRow.LayoutParams.WRAP_CONTENT))
        ;
196     nombre.setPadding(5,1,0,0);
197     TextView cred=new TextView(this);
198     cred.setText(n_cred);
199     cred.setTextSize(7);
200     cred.setTextColor(Color.BLACK);
201     cred.setGravity(Gravity.CENTER_HORIZONTAL);
202     cred.setLayoutParams(new TableRow.LayoutParams(TableRow.
        LayoutParams.WRAP_CONTENT, TableRow.LayoutParams.WRAP_CONTENT))
        ;
203     cred.setPadding(3,1,0,0);
204     if(this.flag){
205         clave.setBackgroundResource(android.R.color.holo_red_light);
206         nombre.setBackgroundResource(android.R.color.holo_red_light);
207         cred.setBackgroundResource(android.R.color.holo_red_light);
208     }
209     else{
210         clave.setBackgroundResource(android.R.color.white);
211         nombre.setBackgroundResource(android.R.color.white);
212         cred.setBackgroundResource(android.R.color.white);
213     }
214     this.flag=!this.flag;
215     TableRow fila=new TableRow(this);
216     fila.addView(clave);
217     fila.addView(nombre);
218     fila.addView(cred);
219     if(opt)
220         this.optativas.addView(fila);
221     else
222         this.resultado.addView(fila);
223 }
224 }

```

Archivo: UEA.java

```
1 package com.proyecto.proyecto;
2
3 import java.util.List;
4 import java.util.ArrayList;
5
6 public class UEA{
7     private List <String> seriacion;
8     private String clave_UEA;
9     private String nombre_UEA;
10    private String trimestre;
11    private String tipo_evaluacion;
12    private String calificacion;
13    private String obl_opt;
14    private int numCreditos;
15    private int prioridad;
16    private int opGlo;
17    private int opRec;
18    /*
19     * Metodo constructor.
20     */
21    public UEA() {
22        this.clave_UEA="";
23        this.nombre_UEA="";
24        this.trimestre="";
25        this.tipo_evaluacion="";
26        this.calificacion="";
27        this.obl_opt="";
28        this.numCreditos=0;
29        this.prioridad=0;
30        this.opGlo=2;
31        this.opRec=3;
32        this.seriacion=new ArrayList <String> ();
33    }
34    /*
35     * Metodos get.
36     */
37    public String getClave() {
38        return(this.clave_UEA);
39    }
40    public String getNombre() {
41        return(this.nombre_UEA);
42    }
43    public String getTrimestre() {
44        return(this.trimestre);
45    }
46    public String getEvaluacion() {
47        return(this.tipo_evaluacion);
48    }
```



```

49 public String getCalificacion(){
50     return(this.calificacion);
51 }
52 public String getObl_opt(){
53     return(this.obl_opt);
54 }
55 public int getNumCreditos(){
56     return(this.numCreditos);
57 }
58 public int getPrioridad(){
59     return(this.prioridad);
60 }
61 public int getOpGlo(){
62     return(this.opGlo);
63 }
64 public int getOpRec(){
65     return(this.opRec);
66 }
67 public List <String> getSeriacion(){
68     return(this.seriacion);
69 }
70 /*
71  * Metodos set.
72  */
73 public void setClave(String dato){
74     this.clave_UEA=dato;
75 }
76 public void setNombre(String dato){
77     this.nombre_UEA=dato;
78 }
79 public void setTrimestre(String dato){
80     this.trimestre=dato;
81 }
82 public void setEvaluacion(String dato){
83     this.tipo_evaluacion=dato;
84 }
85 public void setCalificacion(String dato){
86     this.calificacion=dato;
87 }
88 public void setObl_opt(String dato){
89     this.obl_opt=dato;
90 }
91 public void setNumCreditos(int num){
92     this.numCreditos=num;
93 }
94 public void setPrioridad(int n){
95     this.prioridad=n;
96 }
97 public void setSeriacion(List <String> lista){
98     this.seriacion=lista;
99 }

```

```
100  /*
101   * Metodos de resta de oportunidades.
102   */
103  public void restaGlo(){
104      this.opGlo-=1;
105  }
106  public void restaRec(){
107      this.opRec-=1;
108  }
109 }
```