Universidad Rafael Landívar.
Facultad de Ingeniería.
Análisis y Diseño II.
Ing. Dhaby Eugenio Xiloj Curruchiche.

Informe final:

Sistema para Centros Educativos. "ROXI"

Roca Rodríguez, Juan Carlos (15315-15)
Xicará Xicará, Wilson Giovanni (15146-15)

Quetzaltenango, 11 de julio de 2018.

Cambios realizados en la entrega anterior:

1. Nuevos paquetes desarrollados, cada uno representado como un módulo diferente y funcional dentro del sistema. Entre ellos se encuentran:

    a. **sce.asignacion.carrera:** En este paquete se crea la asignación de una carrera. Esta asignación permite manejar todos los cursos que pertenecen a una carrera. Dichos cursos se encuentran organizados por un pensum.

    b. **sce.asignacion.catedratico:** Dentro de este paquete se permite asignar a un catedrático con una carrera. Así como las opciones de asignar ciertos grados a un catedrático y también la asignación de cursos al mismo.

    c. **sce.asignacion.curso:** Desde este módulo se proveen las clases necesarias para la asignación de cursos. Se ha contemplado dos tipos de estas asignaciones: 1) asignación de cursos a grado: esta relación hace constatar qué cursos están disponibles en el transcurso del grado. 2) asignación de cursos a carrera: en esta categoría entran los cursos libres y otros que no necesitan de un grado para habilitarse. También provee la clase para la obtención de los cursos que han sido asignados a una carrera o a un grado en específico.

    d. **sce.asignacion.estudiante:** Desde este módulo se puede realizar la asignación de estudiantes. Se ha contemplado dos tipos de asignaciones: 1) asignación de estudiantes a grados: es la común y consiste en registrar las asignaciones de un estudiante a todos los cursos asignados en el grado seleccionado (esta funcionalidad es útil para los niveles de educación básica y anteriores). 2) asignación de estudiantes a cursos: similar a la anterior, ya que crea asignaciones de un estudiante a uno o más cursos, con la diferencia de que dichos cursos mostrados son aquellos que están directamente relacionados a una carrera. También provee la clase para la búsqueda y obtención de los estudiantes asignados ya sea a un grado o a cursos.

e. **sce.asignacion.grado:** Pemite la obtención de la asignación de una carrera, con el fin de poder asignarle a esta tantos grados sean requeridos por el centro educativo.
f. **sce.persona:** Permite obtener los diferentes tipos de personas que se manejan dentro del centro educativo. (Estudiantes, catedráticos, personal administrativo, etc.)
g. **sce.persona.educativo:** Incluye funciones enfocadas a las personas dentro del sistema. Entre ellas se encuentra la búsqueda y creación de personas.

2. Se añaden nuevos paquetes al motor del sistema. Entre ellos se encuentran:

    01. **sce.principal.command:** Parte del motor. Provee las interfaces con los métodos estándar para las asignaciones, personas y elementos de asignatura (ciclo escolar, grado y curso). Por lo que, cualquier módulo que necesite hacer uso de estos elementos mencionados anteriormente debe implementar alguna de estas interfaces.

    02. **sce.principal.entity:** Son todas las clases definidas que se utilizarán para el registro de información dentro de la base de datos del sistema.

    03. **sce.principal.ormjpa:** Son todos los controladores que manejan las operaciones relacionadas a la base de datos.

    04. **sce.principal.ormjpa.exceptions:** Manejo de excepciones propias de los controladores.

3. El modelo relacional se modificó. Esto para facilitar el manejo, inserción y la comunicación de datos entre los distintos módulos creados.

Algunos conceptos importantes utilizados en la descripción de los módulos mencionados y otros que se implementarán son:
Ciclo escolar: es la representación de un período de tiempo de estudios. En teoría, representa un año.
Carrera: este concepto hace referencia a los estudios especializados en un fin. Como ejemplo, podemos mencionar las carreras de Bachiller en Ciencias y Letras, Perito Contador, Secretaria, etc., siendo la analogía la misma en los niveles de estudios universitarios. En lo que respecta a los niveles Básico y anteriores, el concepto de carrera hace referencia al nivel en sí (carrera primaria y carrera básico).

Asignación carrera: la asignación de una carrera a un ciclo escolar. Esto hace referencia a la habilitación de una carrera por un período de tiempo. En el nivel universitario este concepto representa los semestres. Cada asignación carrera es única, identificada por un ciclo escolar y una carrera.
Pensum: como su nombre lo indica, es el registro de todos los cursos que componen una carrera. Un pensum organiza los cursos en forma de árbol ya que existe dependencia de unos cursos a otros. Además del curso, posee información de los prerrequisitos de los mismos (otros cursos o cantidad de créditos), así como el grado al que está asignado (esto último no es una referencia obligatoria sino más bien una sugerencia de en qué grado llevar el curso, pero cabe la posibilidad de llevarlo en un grado diferente siempre que dicho grado esté asignado a una carrera válida).
Grado: hace referencia a las etapas en sucesión en las que se divide una carrera. Este registro sólo es de referencia ya que no obliga en las asignaciones de estudiantes y catedráticos.
Asignación de Estudiante: se ha identificado dos tipos: 1) asignación de estudiantes a un grado, 2) asignación de estudiantes a cursos. La primera está pensado para los niveles de educación básica y anteriores, y la segunda para los posteriores. Pero en realidad la única válida es la asignación de estudiantes a cursos; esto debido a que el hecho de que un curso esté asignado a un grado no fuerza a que sólo en dicho grado pueda asignarse al curso (lo que pasa en el nivel universitario). Esto además proporciona la posibilidad de poder realizar asignaciones a otros cursos que no pertenezcan a la misma carrera (de hecho, el único requisito es que estén asignados a una carrera y un curso no es exclusivo a un pensum).
Asignación de Cursos: se ha identificado dos tipos: 1) asignación de cursos a grados, 2) asignación de cursos a carreras. El primer tipo hace referencia a una sugerencia de cuando se puede asignar a dicho curso (como ya se mencionó, el grado no es de carácter obligatorio). El segundo tipo es el que se mantiene en ambos casos. Aquí es donde se registra que un curso si puede ser asignado pues se asume que está asignado a una carrera habilitada (porque dicha carrera está asignada a un ciclo escolar y hacen un período de tiempo vigente). El segundo tipo, por lo tanto, contempla una variante de los cursos:

- **Cursos libres:** estos se caracterizan debido a que no pertenecen a un grado ni a una carrera en específico, y que puede tener o no una sucesión. Esto mismo hace que el grado no sea de carácter obligatorio. Por lo tanto, para los cursos libres se ha considerado lo siguiente:
    - Deben estar asignados a una carrera asignada (a un ciclo escolar) y vigente.
    - Por lo tanto, cada curso libre construye su propio pensum y la posibilidad de existir dependencias de uno hacia otro.

Al momento de realizar la asignación de un curso se especifica el valor de la puntuación sobre la cual se calificará dicho curso (sabemos que por defecto es 100, pero se pensó que dicho valor puede variar). A esto le llamamos una 'distribución de notas'.

Asignación de Catedráticos: se ha identificado dos tipos: 1) asignación de uno o varios grados a un catedrático, 2) asignación de uno o varios cursos a un catedrático. El primer caso se aplica para los niveles de educación básica y anteriores. Esto debido a que en asignación de cursos a grados se especificó los cursos relacionados a un grado y por lo todos esos cursos son asignados al catedrático. Lo que lleva al segundo tipo, en el que se registra todos lo cursos que el catedrático tiene a su cargo (ya sea cursos asociados a un grado o los cursos libres mencionados anteriormente). Una vez asignado el curso, el catedrático es libre de crear su distribución de notas a su criterio.

Distribución de notas: hace referencia a la cantidad de punteo sobre la cual se evalúa el curso en cuestión, y la posibilidad de distribuir dicha nota en varias actividades. Cuando un estudiante es asignado a un curso, automáticamente se le crea su referencia hacia la distribución de notas que tendrá.

Atributos adicionales: Pensado principalmente para los registros de información de Estudiantes y Catedráticos (y se extendería para otros registros de personas). Esto debido a que no se pretende dejar la opción de que el usuario especifique qué atributos adicionales desea manejar (y que no estén entre los atributos por defecto). En la tabla de registro de información hay un campo extra que guarda todos estos atributos y su valor. La propuesta es que dichos valores se almacenarán en formato JSON.

Nuevos cambios:

1. La estructura de paquetes creada anteriormente se mantiene, con la única diferencia que se designarán ciertos grupos como nuevos proyectos.
   Tomamos como referencia lo descrito en el blog de Le Funes(2008, septiembre) “Creando un sistema con plugins en Java”. Algunos puntos importantes a tomar en cuenta para el desarrollo de plugins son los siguientes:

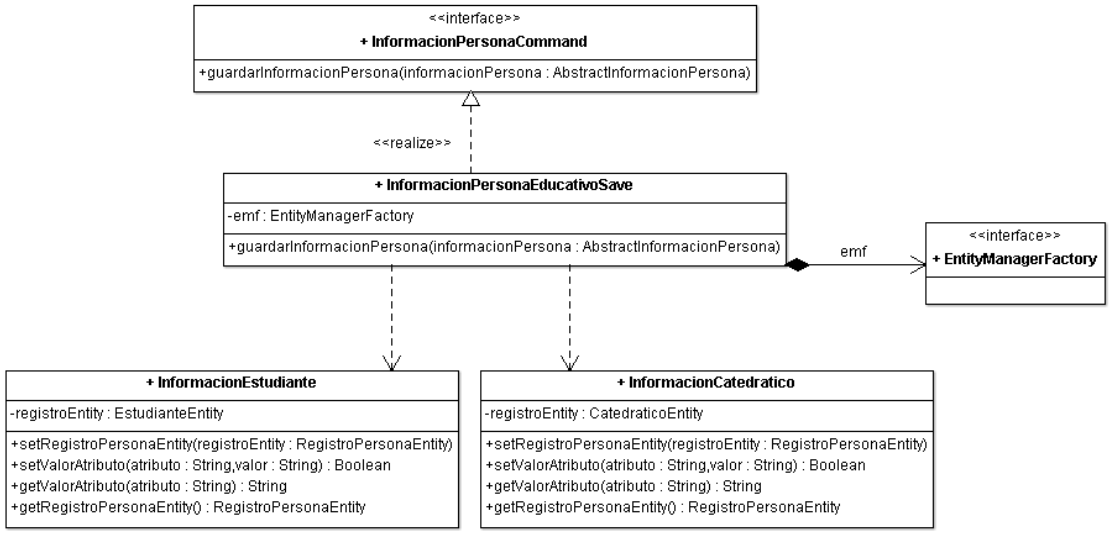
   A. Los plugins deben desarrollarse independientemente de los otros. Por lo tanto, dicho desarrollo debe realizarse en ‘proyectos’ diferentes.

   B. Durante el desarrollo, si un plugin necesita hacer uso de las funcionalidades de otro, sólo se debe importar el proyecto que contiene las clases que necesita el proyecto en desarrollo.

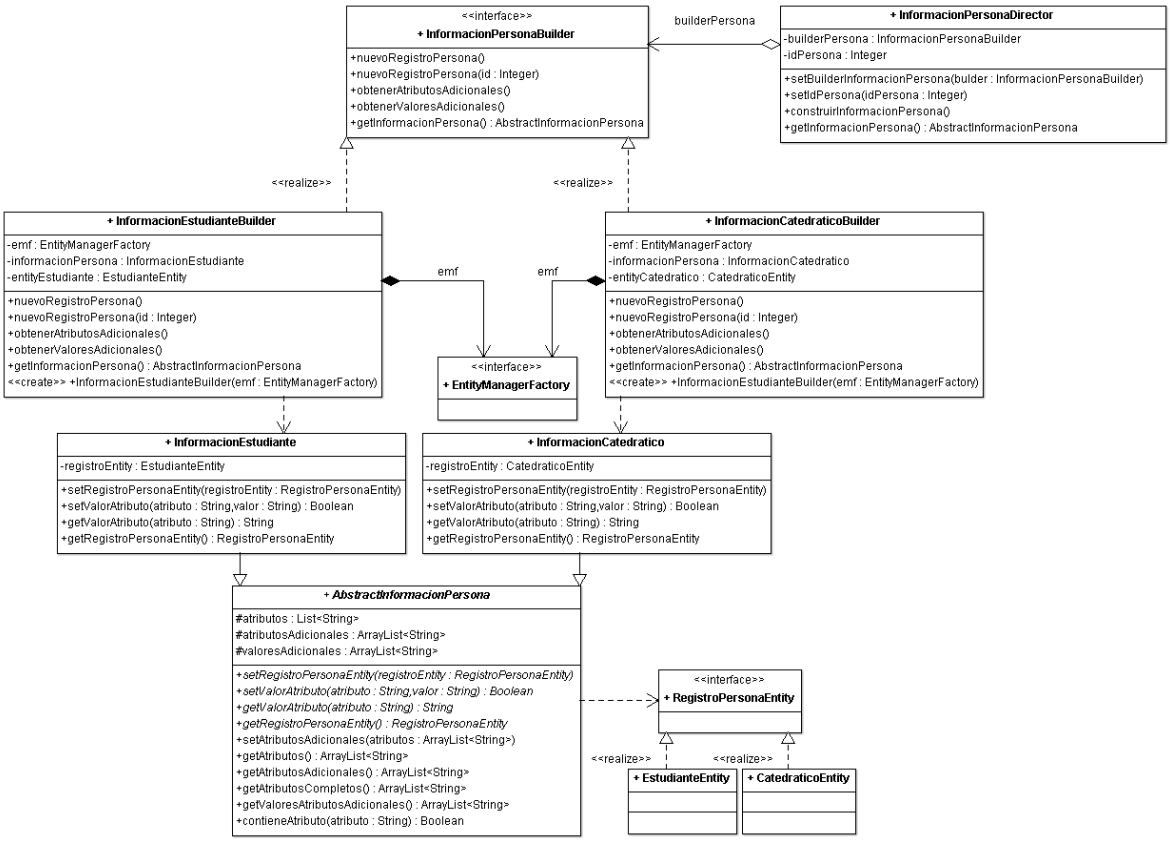
   C. Esto sugiere que nuestra aplicación principal, originalmente no debe tener precargado ningún plugin de forma estática. Éstos se irán añadiendo dinámicamente por medio de un Gestor de plugins que se desarrollará después.

2. Se realizaron algunos cambios en el modelo relacional de la Base de datos. Dichos cambios incluyen actualización de campos de las tablas, con base a las relaciones implicadas en la transferencia de datos entre módulos.

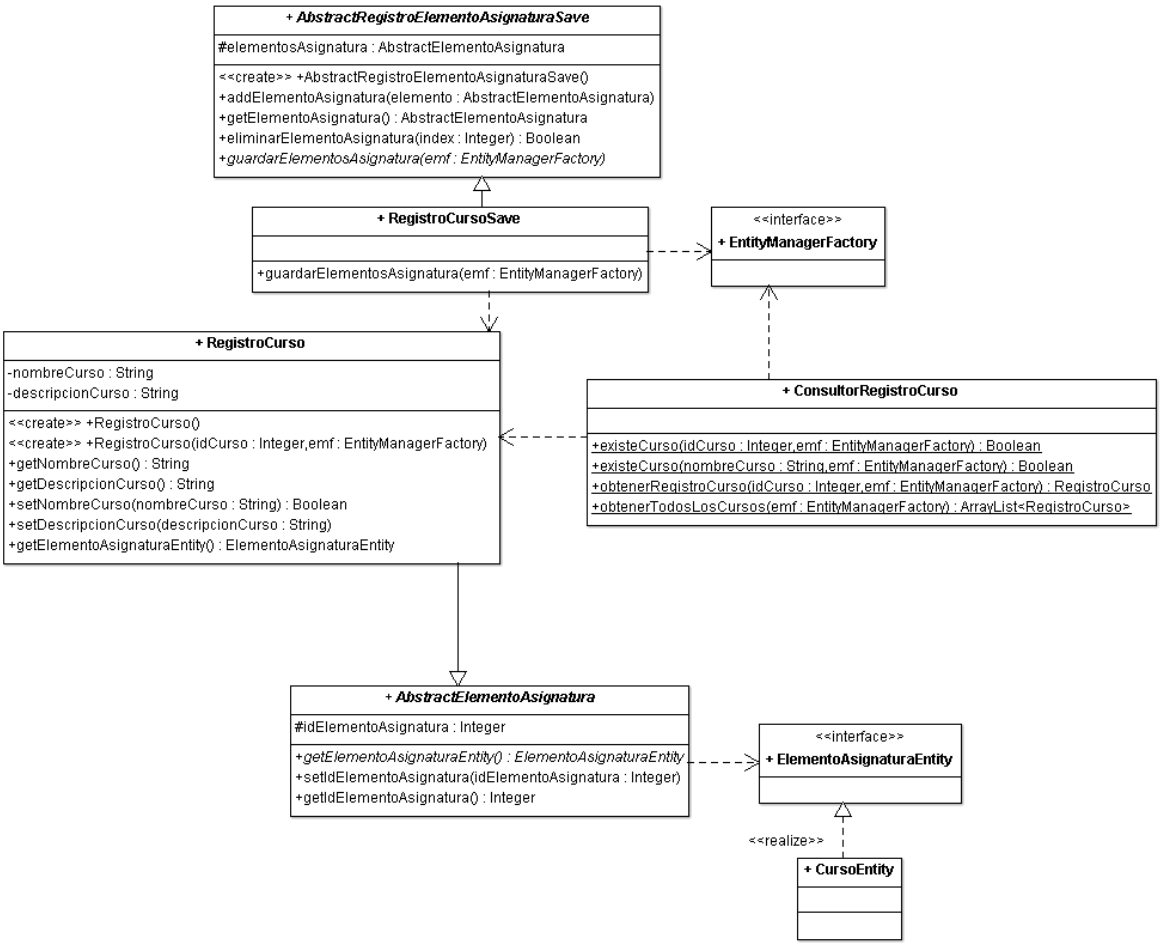
3. Mejoras en la estructuración de diagramas de clases, aplicando patrones de diseño.



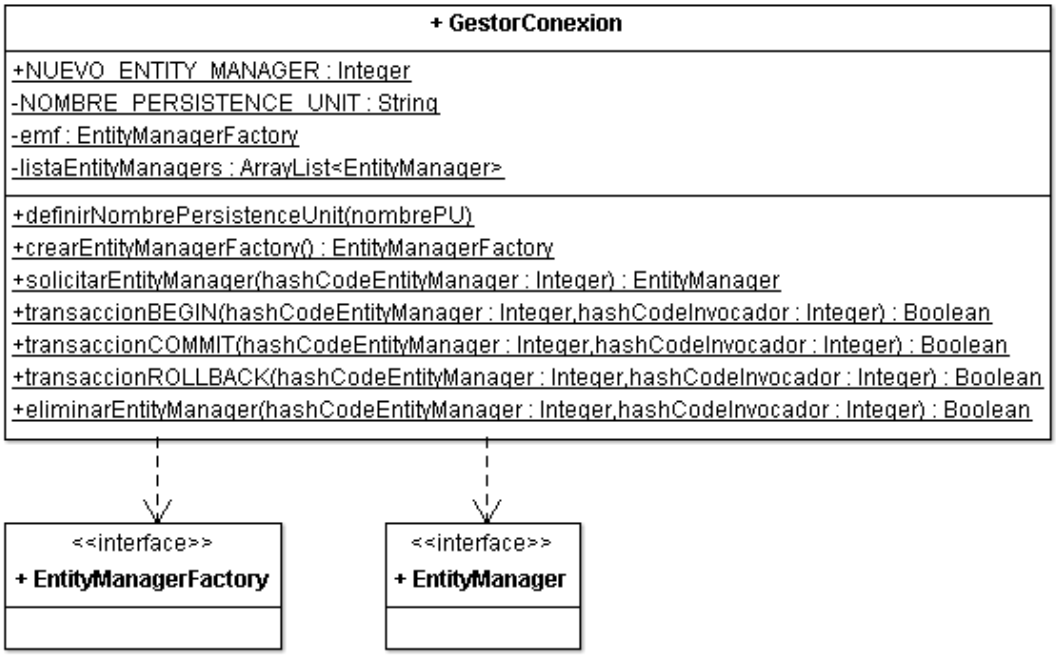
Siguiendo el patrón de diseño Command, implementa la funcionalidad para guardar el registro de información de una persona. Como se ha mencionado con anterioridad, un registro de persona posee sus atributos básicos (que representan un campo en la tabla correspondiente dentro de la base de datos) y un listado de atributos adicionales (que se guardan en un solo campo, en un formato JSON). Por lo tanto, para agregar un nuevo tipo de persona se debe implementar dichas clases para guardar el registro correspondiente.



Siguiendo el patrón de diseño Builder, esta implementación permite crear un nuevo registro con información de una persona. Es principalmente para cuando se obtiene un registro almacenado en la base de datos ya que dicho registro puede tener asociado una serie de atributos adicionales. Dichos atributos son aquellos almacenados en un solo campo en formato JSON, por lo que uno de los procedimientos internos se encargará de decodificar y extraer todos los atributos y su respectivo valor de para poder ser manipulados.



Siguiendo el patrón de diseño Command y Facade (fachada), se presenta el diseño de las clases para guardar los registros correspondientes a los elementos de asignatura. Los elementos de asignatura proporcionados por defecto son el ciclo escolar, grado y curso. Además se proporciona la clase encargada de la comunicación con el módulo correspondiente; dichas clases tendrán las funcionalidades de Consultar el estado de los registros y validar su existencia, Buscar registros en base a parámetros especificados, Anular registros y Crear los mismos.



Siguiendo el patrón de diseño Singleton, esta es la clase encargada de manejar la conexión a la base de datos y gestionar las transacciones. La arquitectura JPA permite manejar un propio espacio de gestión a través de los EntityManager por lo que esta clase provee la funcionalidad de gestionar los EntityManager's y las transacciones de cada una.

Nuevos cambios realizados:

1. **Unión vista - controlador**
   Se inició el proceso de unión entre todos los módulos con su vista gráfica y su controlador. Por el momento se está haciendo de manera estática, pero el objetivo es que cada módulo se pueda ir acoplando dinámicamente.

2. **Creación de jars (módulos)**
   Cada módulo se clasificó en proyectos diferentes e independientes, con el fin de crear su respectivo jar e ir incluyéndose en el proyecto principal.
   Cada módulo trae consigo funciones que permiten el ingreso y manejo de información.

3. **Modificación de clases**
   Por el momento se siguen actualizando algunas clases con el fin de implementar una mejor organización de información.

Cambios finales:

1. Todo el proyecto se estará manejando desde un JFrame llamado VentanaPrincipal. Este se encargará de la interacción entre el usuario o cliente, con el sistema creado.

2. Dentro de `VentanaPrincipal` se encuentra un JMenuBar, el cual contiene los las opciones de acceso hacia los módulos. Este componente realiza una búsqueda con el objetivo de detectar qué módulos son los que se encuentran habilitados. Una vez detectados, permite la creación de un nuevo ítem en el menú, el cual admite al usuario acceder a las funcionalidades proveídas por dicho módulo, todo esto de manera dinámica.

3. Cada módulo consiste en un proyecto independiente. Se agrupan las clases y controladores que se estarán manejando y son compiladas en un proyecto. El archivo JAR generado, será el módulo resultante. Estos JARS se irán adhiriendo al proyecto principal, dependiendo la necesidad que tenga el usuario.

4. Con lo mencionado anteriormente, la idea de la modularidad y extensibilidad consiste principalmente en el desarrollo independiente de un proyecto (módulo que se convierte en un archivo con extensión jar), permitiendo así actualizar dichos módulos sin afectar el funcionamiento principal. Ya que al iniciar la aplicación se registran todos los plugins, se tiene la opción de aumentar o disminuir las funcionalidades agregando o quitando archivos jar, respectivamente.

Bibliografia.

Le Funes. (2008, septiembre). Creando un sistema con plugins en Java. Disponible en https://lefunes.wordpress.com/2008/09/03/creando-un-sistema-con-plugins-en-java/

El proyecto se encuentra disponible en:
https://github.com/WilsonXicara/Sistema-Centro-Educativo