

DOCUMENTO DE CAMBIOS

INGENIEROS A LO BESTIA

3ª INGENIERIA INFORMATICA

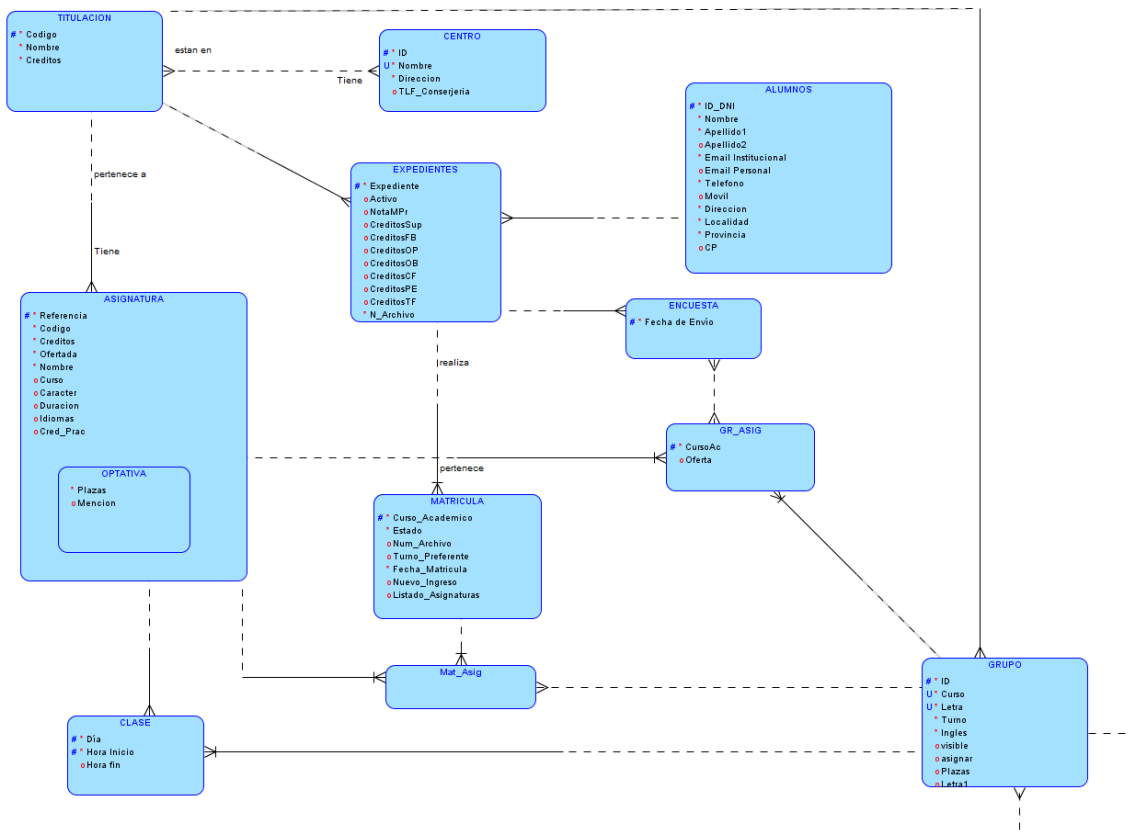
SISTEMAS DE INFORMACION PARA INTERNET

Descripción: Este documento trata de mostrar todos aquellos cambios realizado para la entrega T3 Aplicación Completa, con respecto a las entregas anteriores.

CAMBIOS EN LOS JPA

Actualización del JPA General

Como podremos observar en la imagen de abajo, la antigua base de datos planteada tenía problemas de implementación debido a un mal planteamiento que se ha ido corrigiendo según ha transcurrido el tiempo.



Debido a posteriores implementaciones y a la necesidad de un discriminante entre la entidad Alumno y la posteriormente añadida Secretaría, que representa a aquellos usuarios de la secretaría del centro encargados de gestionar la matriculación de los estudiantes.

Para ello, procedimos a desglosar Alumno y Secretaría en 2 entidades distintas unidas a través de una superentidad llamada Usuario.

Usuario agrupara los atributos principales de Secretaría y Alumno, como serían su ID, contraseña, nombre de usuario, correo institucional, etc...

Con ello logramos 2 cosas, la primera, cumplimos el requisito de un sistema de login donde ambas partes pueden iniciar sesión, y dos, obtenemos la presencia de la Secretaria en la aplicación.



Aparte de esto se realizaron otros cambios, pero antes de hablar de ellos haremos un inciso en esta parte, puesto que debemos hablar de las correcciones hechas a nivel de código, empezando por añadir las nuevas entidades y asignarle sus relaciones correspondientes.

Aprovechando la ventaja de Java y su sistema de herencias, junto a la propiedades de JPA pudimos representar la relación de ambas tanto, a nivel de base de datos, como en el código, pudiendo así almacenar información de manera relacionada e implementar los métodos de la clase padre en sus hijos para facilitar la simplicidad en el código.

Especificaciones de Cambios JPA

Para realizar estas especificaciones, vamos a marcar todas las clases y en las clases que no tenga cambios, lo apuntaremos como (X). Además, hemos hecho cambios mínimos, como cambios en los paquetes/directorios cambiando el directorio “es.uma.platurno.jpa” por “es.uma.informatica.sii.ejb.practica.entidades”, cambios en anotaciones visibles en la base de datos como nombres, restricciones de unicidad o “nulabilidad”, todo esto con fin de llegar a un resultado final.

- 1.- Alumno: Para realizar pruebas con la aplicación, hemos añadido anotaciones @XML y @JSON. Además, hemos quitado la anotación @Id ya que hereda el ID directamente de la clase Usuario. Hemos cambiado la anotación de la relación Alumno-Expediente. Hemos agregado get y set de Expedientes. En el método toString hemos agregado dos métodos procedentes de Usuario para que muestre por pantalla la cadena de validación y el nombre de usuario asociado al Alumno (Opciones para debug).
- 2.- Asignatura: Realizamos un cambio en la estrategia de representación de la tabla de “Table per class” a “Single-Table”. Se han eliminado dos relaciones entre entidades.
- 3.- Centro: Se ha modificado la relación con titulaciones, cambiando la cardinalidad de uno a muchos a muchos a muchos.
- 4.- Clase: Se han agregado a la clase ClaseID, métodos de hashCode, equals y el propio constructor.
- 5.- ClaseMainPersistence: Se ha cambiado por completo la clase, ya que se ha utilizado dicha clase para realizar pruebas de Modelo e Inserción en la Base de Datos.
- 6.- Encuesta: Se ha reemplazado la relación uno a muchos de encuesta con GR_ASIG, por una relación muchos a muchos.

7.- Expediente: Hemos eliminado la relación Expediente con Matrícula (en la clase Matrícula si está implementado, por lo que no es un gran problema).

8.- GR_ASIG: Se han agregado a la clase GR_ASIGID, métodos de hashcode, equals y el propio constructor. Además, se ha reemplazado la relación uno a muchos de GR_ASIG con Encuesta, por una relación muchos a muchos.

9.- Grupo: Hemos eliminado algunas relaciones para poder realizar la implementación final.

10.- Mat_Asig: (X)

11.- Matricula: (X)

12.- Optativa: Se ha agregado una relación muchos a muchos con Titulación.

13.- Secretaría: Se han agregado varios atributos como el DNI, el nombre, el apellido, el email y el ID de serializacion. Se ha eliminado el atributo identificador. Se ha eliminado el constructor con argumentos. Se han agregado los setters, getters, equals y hashcode que faltaban.

14.- Titulación: Se ha reemplazado la relación muchos a uno de Titulacion con Centro, por una relación muchos a muchos. Se han agregado dos relaciones más, una muchos a muchos con Optativas, y otra uno a muchos con Grupos.

15.- Usuario: Se ha cambiado la estrategia de modelado de la base de datos de "Single_Table" a "Table_Per_Class". Hemos cambiado el id de la clase identificador por el username.

Por último, hay que añadir que hemos agregado las entidades que se nos proporcionaron en las ultimas prácticas para tenerlas como referencia, pero que no están agregadas como para tenerlas en la base de datos.

CAMBIOS EN LOS EJB

Actualización del JPA General

Realizando una breve exégesis sobre los EJB y sus implementaciones, a pesar de los grandes problemas que tuvimos (sobre todo de configuración y desarrollo), han sido solucionados en esta última actualización. Los principales cambios han sido:

A.-) Redefinición del método de importación de ficheros .csv y su posterior almacenaje en la base de datos. Tratamiento de comas, caracteres especiales y demás, tres saltos de línea al leer ya que no nos interesaba leer dichas líneas y, por último, el uso de un Buffer de lectura.

B.-) Añadir funcionalidades extra en las diversas clases. (get all devuelve todos los objetos de la bd). Todo esto con fin de poder trabajar con un objeto “Base de Datos” para que nos sea más sencillo trabajar y poder obtener datos.

C.-) Sistema de autenticación con el usuario. Sustitución de un método de autenticación, por uno que autentifica al usuario directamente, gracias a sus credenciales.

D.-)

Especificaciones de Cambios EJB

Para realizar estas especificaciones, vamos a marcar todas las clases y en las clases que no tenga cambios, lo apuntaremos como (X). Además, hemos hecho cambios mínimos, como cambios en los paquetes/directorios cambiando el directorio “es.uma.platurno.jpa” por “es.uma.informatica.sii.ejb.practica.entidades” que influirán directamente en los imports, se ha cambiado la unidad de persistencia de nombre de “ProyectoPlaturno.GrupoF” a “abc”, al cambiarse el nombre de los atributos de los JPA, se han cambiado las sentencias JPQL.

1.- AlertaColisionesHorarios: (X)

2.- AlumnoEjb: (X)

3.- AlumnoInterface: (X)

4.- ASIGNAR GRUPO AUTO: (X)

5.- ASIGNAR GRUPO AUTOInterfaz: (X)

6.- AsignaturasEjb: Se ha añadido la anotación @Inject para añadir las dependencias, un logger para comprobar fallos más tarde, en el método verAsignatura hemos eliminado la creación del atributo auth, ya que el atributo auth lo hemos sustituido por AuthInterface. Se ha añadido el método getAll(), y el método crearAsignatura, ya que no teníamos antes porque importábamos los datos.

7.- AsignaturasEjbInterfaz: Se ha añadido los dos métodos implementados en el anterior apartado.

8.- Autenticacion: Se ha cambiado el método registrarUsuario, validarCuenta, compruebaLogin por completo.

9.- AutenticacionInterfaz: Hemos cambiado el argumento de registrarUsuario “UriBuilder uribuilder” por “String cadena”.

10.- EncuestaEJB: Se ha agregado la anotación @Inject, se ha eliminado el constructor EncuestaEJB con argumentos y se ha modificado el crearEncuesta, se ha creado los métodos verEncuesta y borrarEncuesta.

11.- EncuestaInterfaceEJB: Se ha agregado los métodos verEncuesta y borrarEncuesta.

12.- ExpedienteEJB: Se ha añadido la anotación @Inject, un logger, un método getAll(), un método crearExpediente y solucionamos la autenticación con lo descrito en AsignaturasEjb (AuthInterface).

13.- ExpedienteInterfaz: Se ha modificado los argumentos de ReadExpediente (ahora no le pasamos el usuario) y hemos agregado el método getAll y crearExpediente.

14.- FiltroMatriculas: (X)

15.- FiltroMatriculasInterfaz: (X)

16.- GR ASIGEJB: Se ha agregado la anotación @Inject, un logger, un método getAll(), un método crearGR_ASIG y solucionamos la autenticación con lo descrito en AsignaturasEjb (AuthInterface).

17.- GruposEjbInterface: Se han eliminado todos los métodos, se les han cambiado el nombre y los argumentos, se han cambiado el nombre a la clase, y se han añadido los métodos getAll() y crearGrupo.

18.- IFCSV Interface: Se ha modificado el método leerCSV removiendo el argumento usuario y se ha creado el método leerCSVGrAsig.

19.- ImportarFicheroCSV: Se ha cambiado por completo la clase agregando los métodos leerCSV y leerCSVGrAsig.

20.- ListadoAlumnos: (X)

21.- ListadoAlumnosInterface: (X)

22.- Matricula ejb: Se ha añadido la anotación @Inject, un logger, un método getAll(), un método crearMatricula y solucionamos la autenticación con lo descrito en AsignaturasEjb (AuthInterface).

23.- MatriculaInterfaz: Se han añadido crearMatricula y getAll.

24.- SecretariaEjb: (X)

25.- SecretariaInterfaz: (X)

26.- Solicitud Cambio Grupo: Hemos removido del método generarSolicitud, la sentencia try-catch.

27.- Solicitud Cambio Grupo Interfaz: (X)

28.- TitulacionesEjb: Se ha agregado un logger, el método getAll() y crearTitulacion.

29.- TitulacionesEjbInterface: (X)

30.- UsuarioEjb: Se ha añadido la anotación @Inject, un logger, un método getAll, solucionamos la autenticación con lo descrito en AsignaturasEjb (AuthInterface), se ha removido el constructor con argumentos y se ha modificado el metodo crearUsuarioFromCsvExcel.

31.- UsuarioEjbInterfaz: Se ha modificado el argumento de crearUsuarioFromCsvExcel pasando un alumno, en vez de un DNI y hemos añadido el método getAll().

Se ha agregado la clase BaseDatos, y las clases dadas en las prácticas como referencia.