

1. **Cambios en JPA:**

- Hemos cambiado el nombre y la duración de la clase Asignatura :
 - El nombre se ha puesto como un varchar(75) y la duración como un varchar(20) para así tenerlo como el modelo de Administración de Base de Datos porque hemos tenido que modificar los tamaños de ambos atributos ya que habían datos que excedían el tamaño que habíamos puesto.
- Hemos modificado la clase Asignaturas_Matricula :
 - Se tenía que eliminar el @Id cuando ya había una notación @EmbeddedId.
 - Teníamos que eliminar la notación @JoinColumn(nullable=false) en el atributo grupo ya que en esta clase debería poder ser nulo en caso de no tener asignado ninguno.
- Hemos creado la clase Usuario:
 - Teníamos que crear esta clase para cubrir alguno de los requisitos que hemos propuesto para cubrir la aplicación.
- Hemos modificado la clase Grupo:
 - El atributo Turno_Mañana_Tarde hemos tenido que cambiarlo de longitud 6 a 7 ya que habían datos que excedían el tamaño que habíamos puesto.

2. Cambios en EJB:

- Hemos modificado la clase Asignaturas_MatriculImpl :
 - Se ha cambiado la manera de aplicar el filtro en los métodos denominados “AplicarFiltros” para que coincida el argumento que se la pasa por parámetro con la referencia de las asignaturas por matrícula en lugar de ordenar según el tipo de parámetro.
- Hemos modificado la clase MatriculaPrueba:
 - Se ha realizado el test de visualizarAlumnoPorGrupo que estaba incompleto.
- Hemos modificado la clase ExpedienteImpl:
 - Había que arreglar el método Importar() porque faltaba modificar los CSVFormat de cada importar, cambiando los CSVFormat.DEFAULT por CSVFormat.newFormat('\n') y teniendo en cuenta que antes de importar expedientes es necesario importar los alumnos y las titulaciones antes de comenzar.
 - Métodos mostrarDatosAdmin() para mostrar todos los datos de la base de datos de la clase que llame al método.
 - El método Importar() hemos tenido que cambiarlo para que si lee la clave primaria compruebe si ese dato ya estaba en la base de datos o no, para que así no lo importe si ya estaba.
- Hemos modificado la clase AlumnoImpl :
 - Métodos mostrarDatosAdmin() para mostrar todos los datos de la base de datos de la clase que llame al método.
 - El método validarAcceso() lo hemos eliminado para así ponerlo en la clase UsuarioImpl que es donde debería ir.
 - El método Importar() hemos tenido que cambiarlo para que si lee la clave primaria compruebe si ese dato ya estaba en la base de datos o no, para que así no lo importe si ya estaba.
- Hemos modificado la clase AsignaturalImpl:
 - Métodos mostrarDatosAdmin() para mostrar todos los datos de la base de datos de la clase que llame al método.

- El método Importar() hemos tenido que cambiarlo para que si lee la clave primaria compruebe si ese dato ya estaba en la base de datos o no, para que así no lo importe si ya estaba.
- Hemos modificado la clase ClaseImpl:
 - Métodos mostrarDatosAdmin() para mostrar todos los datos de la base de datos de la clase que llame al método.
 - El método Importar() hemos tenido que cambiarlo para que si lee la clave primaria compruebe si ese dato ya estaba en la base de datos o no, para que así no lo importe si ya estaba.
- Hemos modificado la clase MatriculaImpl:
 - Métodos mostrarDatosAdmin() para mostrar todos los datos de la base de datos de la clase que llame al método.
 - El método Importar() hemos tenido que cambiarlo para que si lee la clave primaria compruebe si ese dato ya estaba en la base de datos o no, para que así no lo importe si ya estaba.
- Hemos modificado la clase OptativaImpl:
 - Métodos mostrarDatosAdmin() para mostrar todos los datos de la base de datos de la clase que llame al método.
 - El método Importar() hemos tenido que cambiarlo para que si lee la clave primaria compruebe si ese dato ya estaba en la base de datos o no, para que así no lo importe si ya estaba.
- Hemos modificado la clase TitulacionImpl:
 - Métodos mostrarDatosAdmin() para mostrar todos los datos de la base de datos de la clase que llame al método.
 - El método Importar() hemos tenido que cambiarlo para que si lee la clave primaria compruebe si ese dato ya estaba en la base de datos o no, para que así no lo importe si ya estaba.
- Hemos creado la clase UsuarioImpl:
 - Esta clase nos hará falta para todos los métodos de la clase Usuario que nos hagan falta para cubrir los requisitos .
 - Hemos añadido el método validarAcceso() ya que en esta clase es donde debería haber ido desde un principio.

➤ Hemos modificado la clase GrupoImpl:

- Se han modificado los métodos crear ,buscarPorCursoLetra porque nos conviene más la implementación que hay actual del método y hemos creado el método mostrarDatosAdmin() para mostrar básicamente los datos .

3. Requisitos que no hemos realizado:

RF-06 (Comprobación de plazas disponibles):

- Este requisito se cubre con EJB; no se ha realizado en los tests de Selenium debido a que no lo utilizamos en asignar grupos.

RF-08 (Preferencia de asignación de grupo a alumnos que elijan alguna asignatura en inglés):

- Este requisito hemos decidido que al final no lo vamos a implementar. Se ha conseguido que en la base de datos los de inglés tengan preferencia, pero no hemos conseguido transportarlo a la aplicación

RF-09 (Preferencia a los alumnos de tiempo parcial):

- Este requisito hemos decidido que no lo vamos a implementar.
- Para acreditar las circunstancias que te permite matricularte a tiempo parcial, hay que entregar una instancia general en Secretaría, proceso el cual la Universidad tiene su propio flujo y del que no poseemos acceso.

RF-10 (Solicitud de cambio de grupo):

- Este requisito no lo hemos implementado al final ya que no hay un proceso fijo, ya que depende de lo que justifique cada alumno.

RF-12 (Subida de documentación):

- Este requisito hemos decidido que no lo vamos a implementar.
- Para poder realizar este requisito, hay que entregar una instancia general en Secretaría, proceso el cual la Universidad tiene su propio flujo y del que no poseemos acceso.

RF-14 (Mostrar horario de asignaturas):

- Este requisito no se ha implementado del todo ya que está pensado para el punto de vista del alumno, nosotros de momento lo tenemos implementado desde el administrador.

RF-15 (Aviso de solapamiento):

- Este requisito hemos decidido que no lo vamos a implementar ya que resulta bastante complejo ya que habría que seguir una serie de pasos bastante larga.

RF-16 (No sobrescribir modificaciones manuales hechas por secretaría):

- Este requisito hemos decidido no implementarlo ya que las modificaciones se trata de un proceso muy específico y resulta un procedimiento muy complejo.