

Ejercicio JSF con Lógica

1. Convertir el proyecto a Web
 - a. Agregar la librería de Wildfly 13 en el build path
 - b. Agregar CDI, JSF, y Dynamic Web Module en los Facets del proyecto
 - c. Configurar la base de datos postgres en WildFly agregando la siguiente configuración en el datasource

```
<pool>
  <min-pool-size>5</min-pool-size>
  <max-pool-size>10</max-pool-size>
  <prefill>false</prefill>
  <use-strict-min>false</use-strict-min>
  <flush-strategy>IdleConnections</flush-strategy>
</pool>
```

- d. Cambiar el tipo de transacciones a JTA y agregar el datasource acorde al definido en WildFly

```
<jta-data-source>java:/PostgresDS</jta-data-source>
```
2. Anotar la clase EstudianteNB para que sea un Named Bean con scope de sesión
3. Conectar el Named Bean EstudianteNB con el EJB EstudiantesLogic. Consultar en la consola de WildFly la ruta de interfaz remota que empieza con java:global
4. Probar y corregir la interfaz para que la inserción funcione correctamente.
5. Obtener los programas del alumno en la lógica y persistirlos.
 - a. Modificar el método createAlumno para que reciba la colección de programas del alumno
 - b. Modificar la implementación de createAlumno para que persista la colección de programas del alumno
6. Al agregar estudiante en el Named Bean, realizar los siguiente:
 - a. obtener el código del programa (de la variable programa)
 - b. crear la colección de programas del alumno
 - c. agregar el programa escogido en la colección (crear la llave primaria también)
 - d. agregar el programa a la colección de programas del alumno
 - e. agregar la colección de programas al alumno.
 - f. Utilizar el método modificado de la lógica para pasar la colección de programas.
7. Implementar la funcionalidad en el NB y lógica para la consultar un estudiante.
 - a. Crear un método en la lógica para consultar el estudiante por código utilizando el tAlumnoDao.
 - b. Obtener el primer programa del estudiante y construir la cadena código-descripción para asignarla al atributo programa.
 - c. Crear un método en el NB que consuma la lógica y asigne los valores de los campos del objeto recibido.
8. Agregar un botón para consultar un estudiante por código y enlazarlo al método del NB.
9. Implementar la funcionalidad en el NB y lógica para la actualizar el estudiante.
10. Agregar un botón para actualizar un estudiante consultado y enlazarlo al método del NB.
11. Implementar la funcionalidad en el NB y lógica para el borrar el estudiante.
12. Agregar un botón para borrar un estudiante consultado y enlazarlo al método del NB.
13. Agregar un botón cancelar para salir del modo consulta y volver a insertar.
14. Activar y desactivar los botones acorde a la funcionalidad permitida (inserción/consulta).
 - a. Desactivar el campo código después de consultar

