

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 sesi
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 funciona correctamente.
5. Obtener el programa al agregar estudiante en el Named Bean e incluirlo para agregarlo al estudiante.
6. Obtener los programas del alumno en la lógica y persistirlos.
7. Agregar un botón para consultar un estudiante por código.
8. Implementar la funcionalidad en el NB y lógica para la consulta realizada.
9. Agregar un botón para actualizar un estudiante consultado.
10. Implementar la funcionalidad en el NB y lógica para la actualización realizada.
11. Agregar un botón para borrar un estudiante consultado.
12. Implementar la funcionalidad en el NB y lógica para el borrado realizado.
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).