



**UNIVERSIDAD DEL QUINDÍO**  
**FACULTAD DE INGENIERÍA**  
**PROGRAMA DE INGENIERÍA DE SISTEMAS Y COMPUTACIÓN**

Información general	
Actualizada por:	Carlos Andrés Flórez Villarraga
Duración estimada en minutos:	90
Docente:	Christian Andrés Candela y Einer Zapata
Guía no.	13
Nombre de la guía:	EJB de Sesión Singleton

<b>Información de la Guía</b>
-------------------------------

## OBJETIVO

Comprender que son los EJB de sesión singleton y aprender a crearlos y configurarlos para el desarrollo de la lógica de negocio (capa de negocio de la aplicación).

## CONCEPTOS BÁSICOS

Manejo de Eclipse, Java, Bases de Datos, JDBC, DataSource, Entidades y Glassfish.

## CONTEXTUALIZACIÓN TEÓRICA

Un EJB Singleton es un tipo de EJB disponible a partir de Java EE 6 que tiene la particularidad de que solo existe una instancia de él para toda nuestra aplicación Enterprise. Puede servir para acceder a datos que estén compartidos entre varias aplicaciones y se almacenen en el EJB como si se tratará de una pequeña cache. Por ejemplo podríamos disponer de un conjunto de rutas a carpetas o algo similar.

A diferencia de un EJB de Sesión sin estado, un EJB Singleton siempre tendrá solo una instancia por aplicación, sin importar cuántos clientes intenten acceder a ella.

## PRECAUCIONES Y RECOMENDACIONES

Tenga que en cuenta que si la aplicación a sido desplegada previamente y ha tenido errores, debe remover la aplicación mal desplegada antes de intentar ejecutarla nuevamente.

## ARTEFACTOS

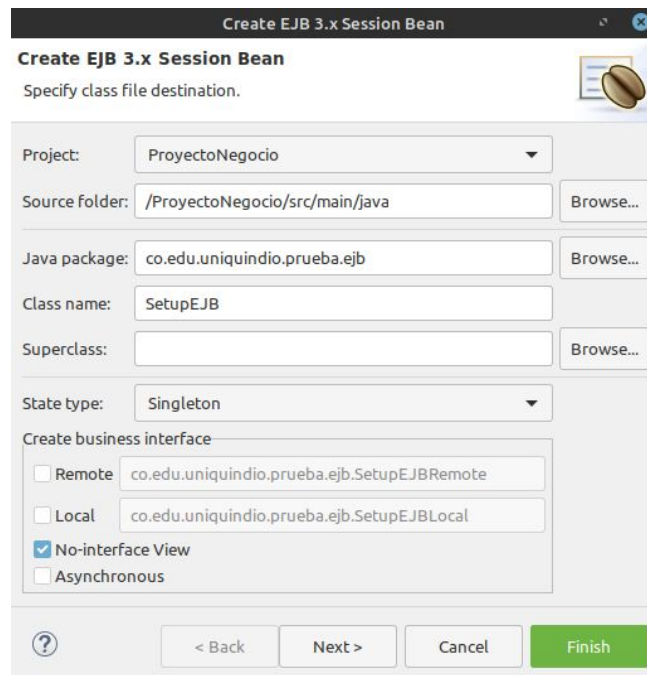
Se requiere tener instalado el JDK y un IDE para el desarrollo de aplicaciones (Eclipse JEE en su última versión), un servidor de aplicaciones que cumpla con las especificaciones de JEE, para esta práctica Glassfish y el motor de base de datos Mysql.

## EVALUACIÓN O RESULTADO

Se espera que el alumno pueda crear y usar exitosamente un EJB de sesión singleton para la creación de la lógica de negocio de las aplicaciones.

## Procedimiento

1. Para el desarrollo de esta guía necesitará una base de datos en MySQL, un proyecto de tipo Maven – POM, el cual debe contener un proyecto Maven con soporte para el uso de JPA, otro para la creación de EJB (negocio) y otro proyecto Maven configurado para la realización de pruebas. Y una conexión a dicha base de datos para ser usada en la generación de las tablas. De preferencia debería trabajar en el proyecto de clase.
2. Cree una consulta que le permita determinar el número de administradores registrados.
3. Cree una prueba que le permita verificar el correcto funcionamiento de su consulta.
4. En su Proyecto EJB (capa de negocio), cree un EJB de sesión singleton el cual se usará, en este caso, para establecer parámetros de configuración durante su primera ejecución. Para ello acceda al menú File – New - Other – Session Bean (EJB 3.x). Cuando se le pregunte por el paquete del EJB puede crear un paquete de su preferencia, lo mismo para el nombre del EJB. Seleccione como State type Singleton, la opción No-Interface View. Como resultado se debe haber creado una clase.



5. Adicione a su nuevo EJB un atributo de tipo EntityManager, con la anotación @PersistenceContext

```
@PersistenceContext  
private EntityManager entityManager;
```

6. Marque el EJB que acaba de crear con la anotación @Startup. Está anotación indica al sistema que el EJB debe ser creado al momento de iniciar la aplicación.

```
@Singleton
@LocalBean
@Startup
public class SetupEJB {

    @PersistenceContext
    private EntityManager entityManager;

}
```

7. En su EJB singleton cree un método público sin retorno y sin parámetros con el nombre de su preferencia.

```
@Singleton
@LocalBean
@Startup
public class SetupEJB {

    @PersistenceContext
    private EntityManager entityManager;

    public void config() {

    }

}
```

8. Marque el método creado con la anotación `@PostConstruct`. Esta anotación permite que el método sea invocado inmediatamente después de la construcción del objeto.

```
@Singleton
@LocalBean
@Startup
public class SetupEJB {

    @PersistenceContext
    private EntityManager entityManager;

    @PostConstruct
    public void config() {

    }

}
```

9. En el método creado en el punto anterior use una consulta para determinar el número de administradores registrados. En caso de que no haya ningún administrador registrado registre uno ( programe el método `registrarAdministrador`), el cual será usado para ingresar a la aplicación.
10. Para probar la correcta configuración despliegue el proyecto en el servidor de aplicaciones. Para ello de clic derecho sobre el proyecto EAR y acceda a la opción Run As – Run on Server.
11. Si todo ha salido bien, en la base de datos podrá verificar que se hayan creado las tablas correspondientes a las entidades de su proyecto. Y en la tabla de administradores deberá ver registrado el administrador que registro su EJB.



*Guía de laboratorio*  
*Área de Programación y Algoritmia*



12. Determine si existe información adicional que se necesite registrada al momento de ejecutar su aplicación, y de ser así adicione el código necesario para registrar dicha información.

**Para la próxima clase**

Leer sobre JSF (Java Server Faces), para qué sirve, qué críticas tiene, extensiones, qué alternativas existen a JSF. <https://javaee.github.io/javaxserverfaces-spec/>