

Ejercicio de EJB – Stateless SessionBeans

Introducción

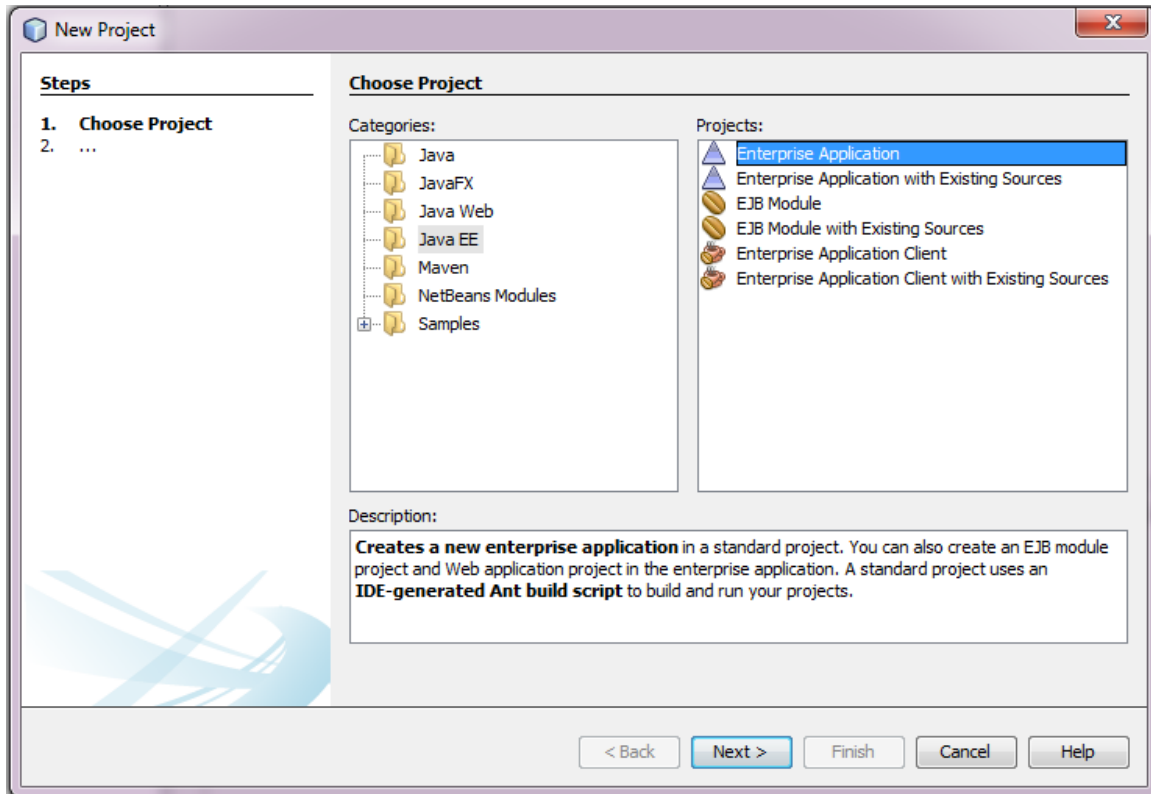
Tomaremos como ejemplo el escenario donde una aplicación java “cliente” requiere interactuar con otra aplicación java que ejecuta en un Application Server, para consumir un servicio que calcula el sueldo de un empleado. Esta aplicación “servidor” está construida con un componente EJB SessionBean que implementa la lógica concreta del cálculo del sueldo.

Se definirán en NetBeans tres proyectos:

1. Proyecto de tipo *Enterprise Application* que representará a nuestra aplicación “servidor”
2. Proyecto de tipo *EJB Module* donde construiremos el componente EJB
3. Proyecto de tipo *Java Application* para la aplicación “cliente”

Creación del proyecto Enterprise Application

1. Seleccionar del menú de NetBeans la opción para crear un nuevo proyecto. En la ventana que se muestra, seleccionar de la lista de categorías (a la izquierda) la opción Java EE y de la lista de proyectos la opción Enterprise Application.



2. Ingresar **CalcularSueldosEA** como nombre de Proyecto y especificar la ubicación del directorio de trabajo.

The screenshot shows the 'New Enterprise Application' wizard at Step 2, 'Name and Location'. The 'Steps' pane on the left lists: 1. Choose Project, 2. **Name and Location**, and 3. Server and Settings. The main area contains the following fields and options:

- Project Name:** CalcularSueldosEA
- Project Location:** C:\EjercicioArquitectura (with a 'Browse...' button)
- Project Folder:** C:\EjercicioArquitectura\CalcularSueldosEA
- ☐ **Use Dedicated Folder for Storing Libraries**
- Libraries Folder:** (empty, with a 'Browse...' button)
- Text: Different users and projects can share the same compilation libraries (see Help for details).

At the bottom, there are navigation buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

3. Seleccionar como servidor la opción *Glassfish Server 3.1.2*.

Seleccionar la versión JEE: *Java EE 6*

Desmarcar todas las opciones (crearemos el módulo EJB a mano luego).

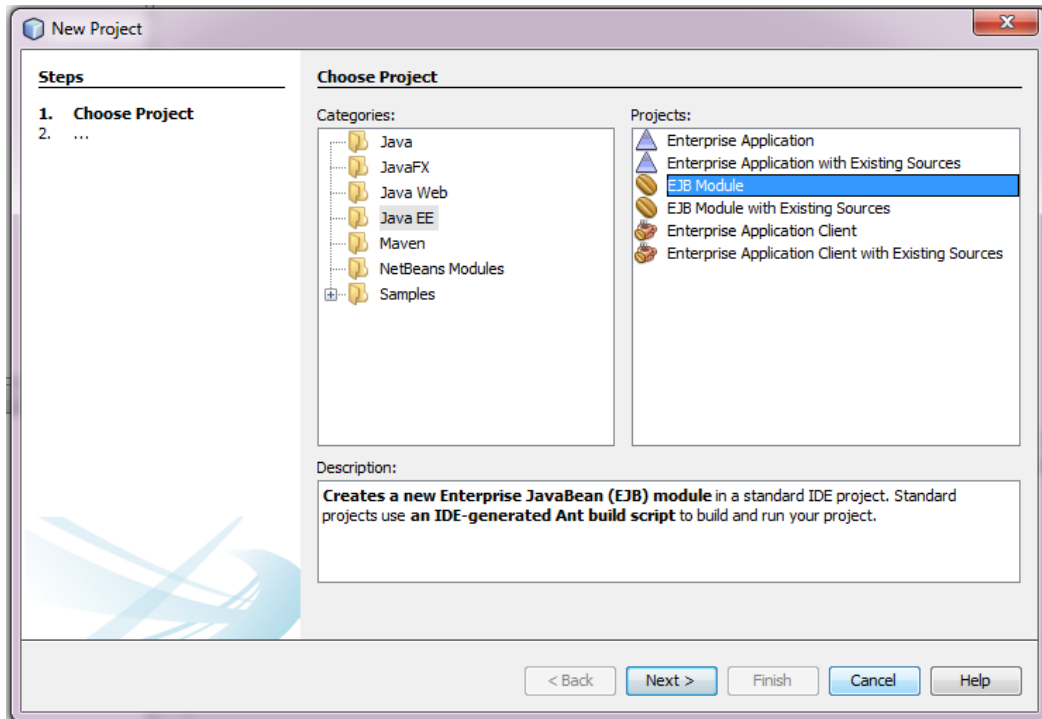
The screenshot shows the 'New Enterprise Application' wizard at Step 3, 'Server and Settings'. The 'Steps' pane on the left lists: 1. Choose Project, 2. Name and Location, and 3. **Server and Settings**. The main area contains the following fields and options:

- Server:** GlassFish Server 3.1.2 (with an 'Add...' button)
- Java EE Version:** Java EE 6 (dropdown menu)
- ☐ **Enable Contexts and Dependency Injection**
- ☒ **Set Source Level to 6**
- Text: Recommendation: Source Level 6 should be used in Java EE 6 projects.
- ☐ **Create EJB Module:** CalcularSueldosEA-ejb
- ☐ **Create Web Application Module:** CalcularSueldosEA-war

At the bottom, there are navigation buttons: '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

Creación del proyecto EJB Module

1. Seleccionar del menú de NetBeans la opción para crear un nuevo proyecto. En la ventana que se muestra, seleccionar de la lista de categorías (a la izquierda) la opción *Java EE* y de la lista de proyectos la opción *EJB Module*.



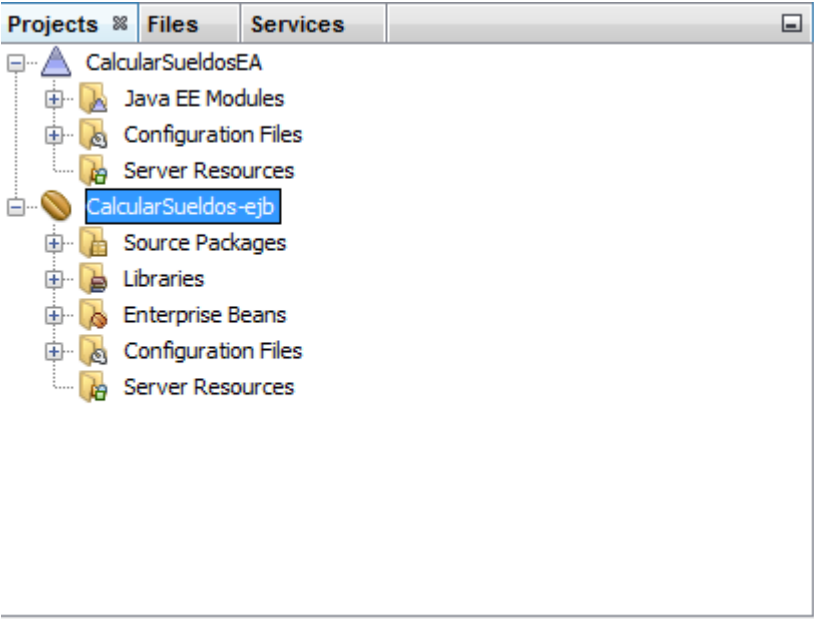
2. Ingresar **CalcularSueldo-ejb** como nombre de Proyecto y especificar la ubicación del directorio de trabajo.

The screenshot shows the 'New EJB Module' dialog box with the 'Name and Location' tab selected. The 'Steps' list on the left indicates the current step is '2. Name and Location'. The 'Project Name' field contains 'CalcularSueldos-ejb'. The 'Project Location' field contains 'C:\EjercicioArquitectura' with a 'Browse...' button next to it. The 'Project Folder' field contains 'C:\EjercicioArquitectura\CalcularSueldos-ejb'. There is an unchecked checkbox for 'Use Dedicated Folder for Storing Libraries' and a 'Libraries Folder' field with a 'Browse...' button. A note states: 'Different users and projects can share the same compilation libraries (see Help for details)'. At the bottom are buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

3. Asociar a la aplicación Enterprise que creamos en el paso anterior: *CalcularSueldosEA*
Seleccionar como servidor: *GlassFish Server 3.1.2*
Seleccionar la versión JEE: *Java EE 6*

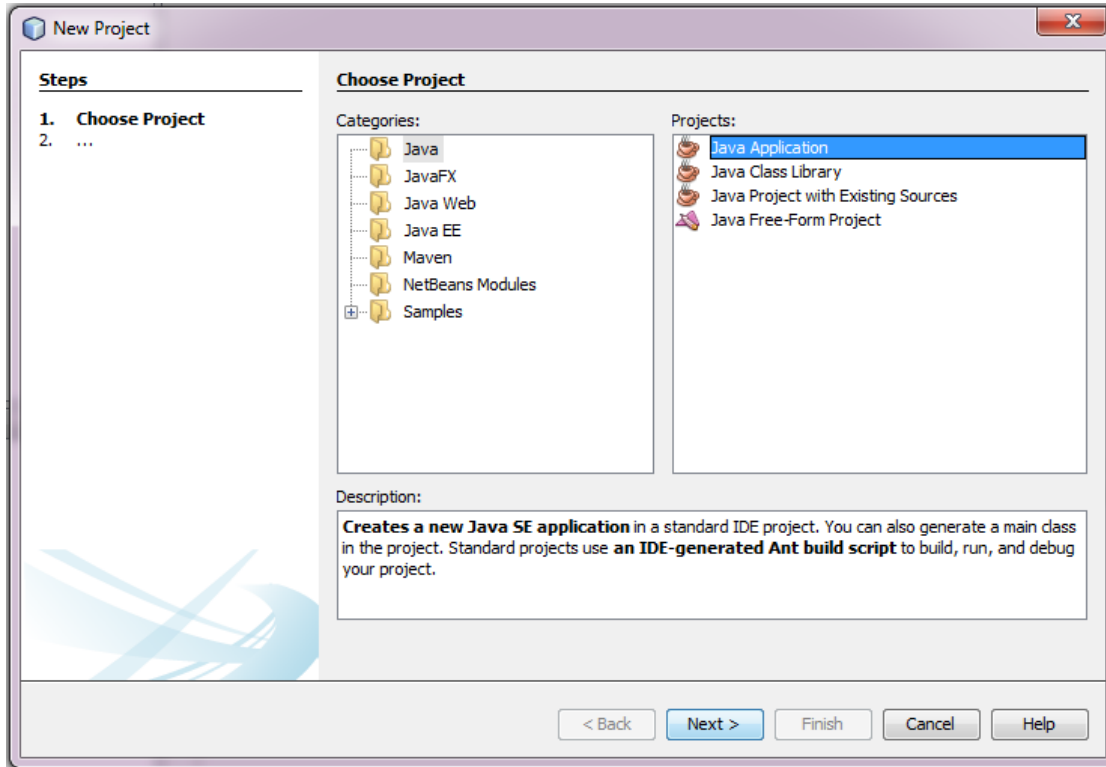
The screenshot shows the 'New EJB Module' dialog box with the 'Server and Settings' tab selected. The 'Steps' list on the left indicates the current step is '3. Server and Settings'. The 'Add to Enterprise Application' dropdown is set to 'CalcularSueldosEA'. The 'Server' dropdown is set to 'GlassFish Server 3.1.2' with an 'Add...' button next to it. The 'Java EE Version' dropdown is set to 'Java EE 6'. There are two checkboxes: 'Enable Contexts and Dependency Injection' (unchecked) and 'Set Source Level to 6' (checked). A recommendation note states: 'Recommendation: Source Level 6 should be used in Java EE 6 projects.' At the bottom are buttons for '< Back', 'Next >', 'Finish', 'Cancel', and 'Help'.

4. Una vez confirmada la finalización del asistente de creación de proyectos, se podrá ver en la vista **Projects** la estructura del proyecto recién creado:



Creación de una aplicación para contener las interfaces Remotas

1. Seleccionar del menú de NetBeans la opción para crear un nuevo proyecto. En la ventana que se muestra, seleccionar de la lista de categorías (a la izquierda) la opción *Java* y de la lista de proyectos la opción *Java Application*.



2. Ingresar **Interfaces** como nombre de Proyecto y especificar la ubicación del directorio de trabajo. Deseleccionar la opción *Create Main Class*.

New Java Application

Steps

1. Choose Project
2. **Name and Location**

Name and Location

Project Name:

Project Location:

Project Folder:

☐ Use Dedicated Folder for Storing Libraries

Libraries Folder:

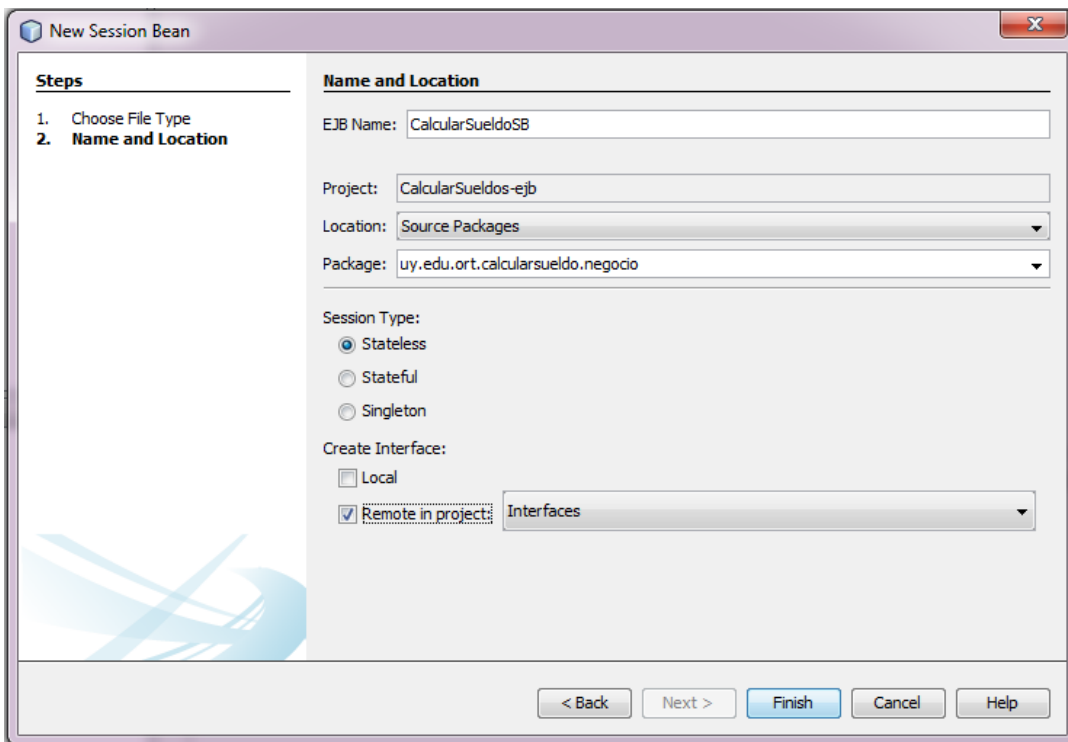
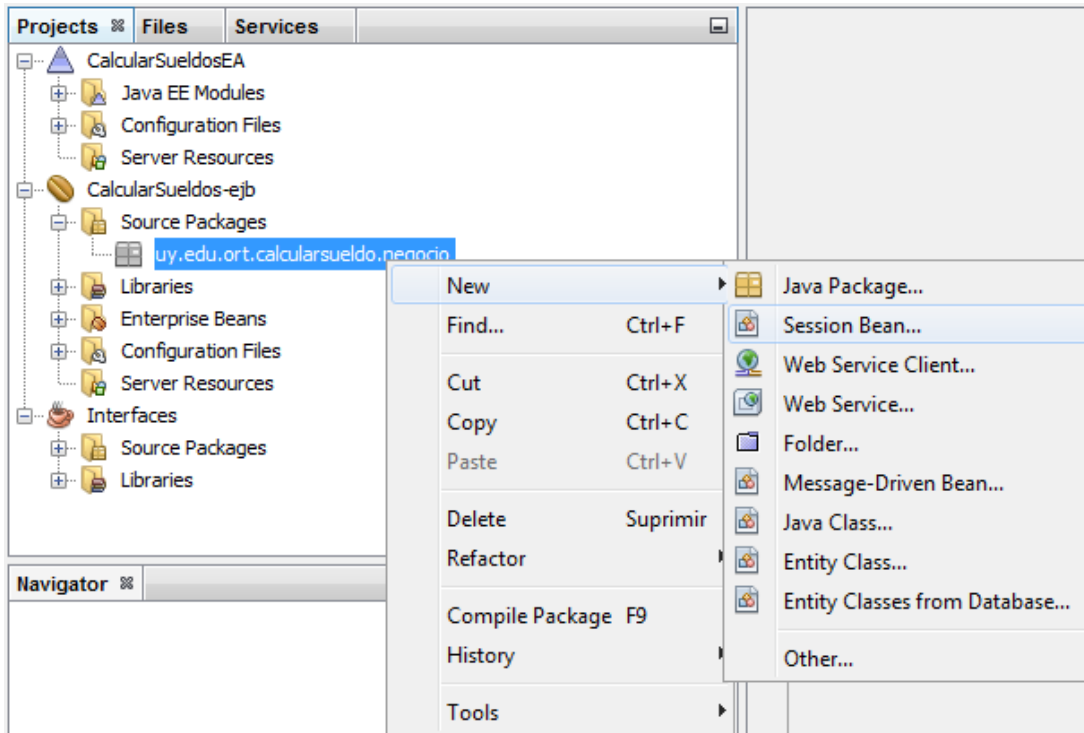
Different users and projects can share the same compilation libraries (see Help for details).

☒ **Create Main Class**

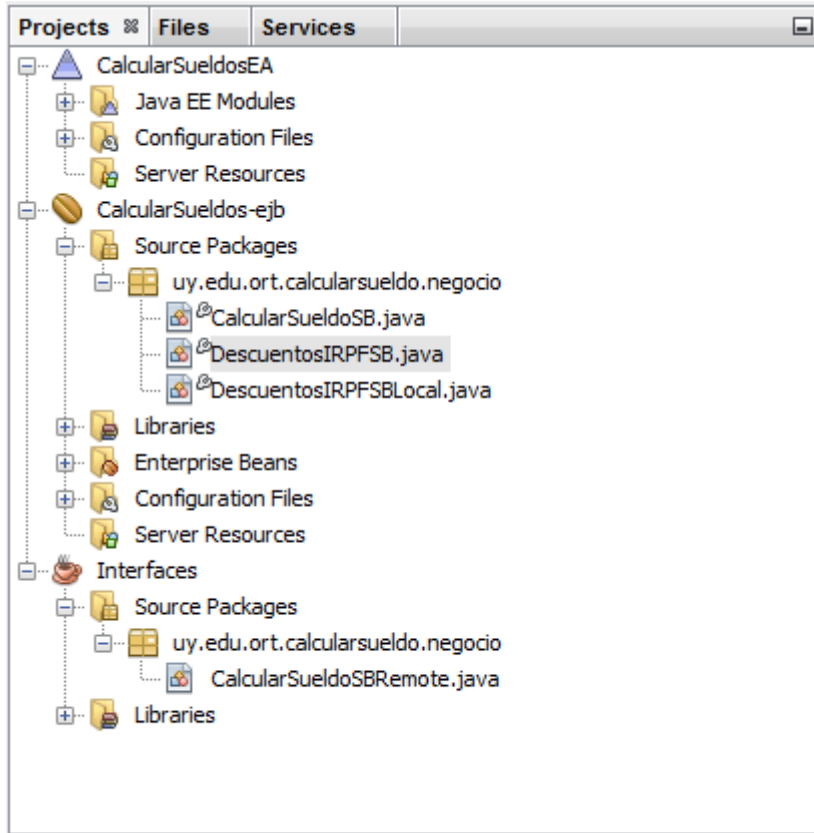
< Back Next > **Finish** Cancel Help

Creación del Stateless Session Bean

1. Posicionarse en la carpeta *Source Packages* y crear un componente *Session Bean* con el nombre **CalcularSueldoSB** en el package **uy.edu.ort.calcularsueldo.negocio** especificando que sea de tipo *Stateless* y marcando sólo la opción para crear su interfaz *Remote* (seleccionado la aplicación *Interface*):



2. Repetir el paso para crear otro Session Beans con el nombre **DescuentosIRPFSB**, de tipo Stateless con una interface Local.
3. Una vez confirmada la finalización del asistente de creación de proyectos, se podrá ver en la vista **Projects** la estructura del proyecto recién creado:



4. Código de la interface Local (DescuentosIRPFSBLocal)

```
@Local
public interface DescuentosIRPFSBLocal {
    public double descuentoIRPF(double sueldo);
}
```

5. Código del Session Bean DescuentosIRPF SB

@Stateless

```
public class DescuentosIRPF SB implements DescuentosIRPF SBLocal {  
  
    @Override  
    public double descuentoIRPF(double sueldo) {  
        double desc=0;  
        if(sueldo>100 &&sueldo<200){  
            desc= 10;  
        }else if (sueldo>200){  
            desc= 20;  
        }  
        return desc;  
  
    }  
  
}
```

6. Código de la interface Remota (CalcularSueldoSBRemote)

@Remote

```
public interface CalcularSueldoSBRemote {  
  
    public double calcularSueldo(double sueldo);  
  
}
```

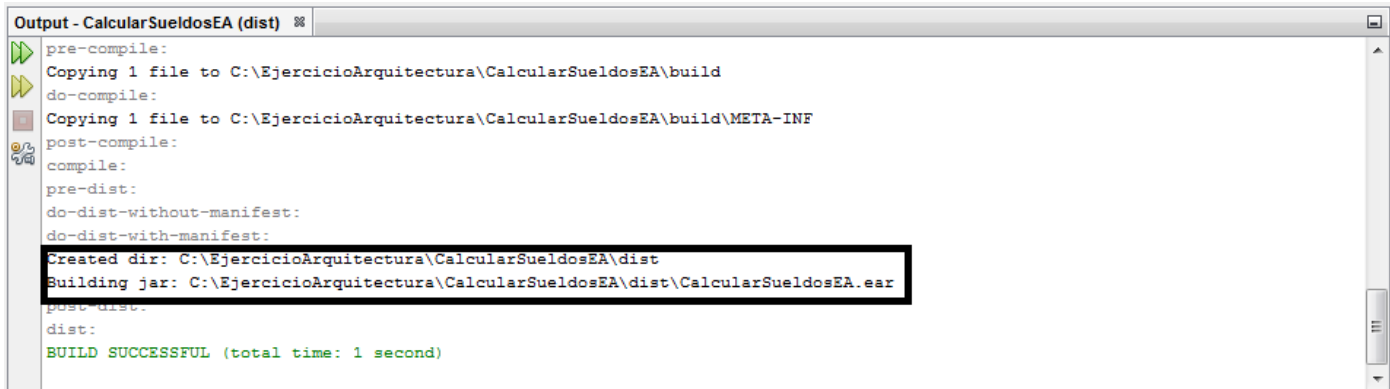
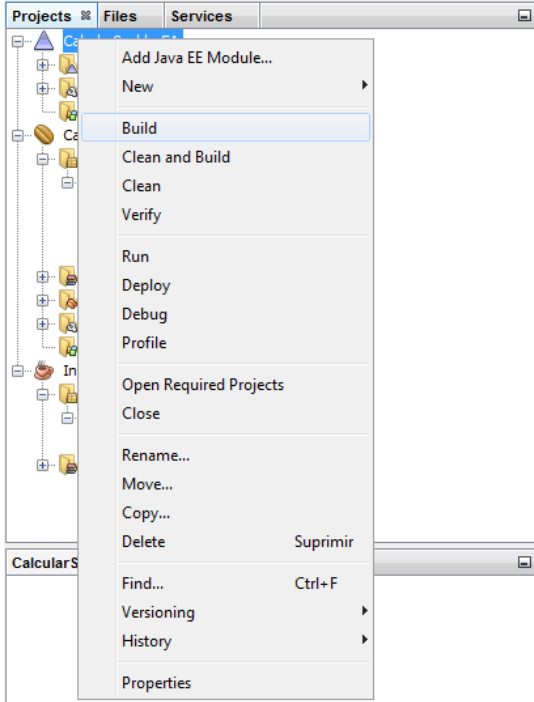
7. Código en el Session Beans CalcularSueldoSB

@Stateless

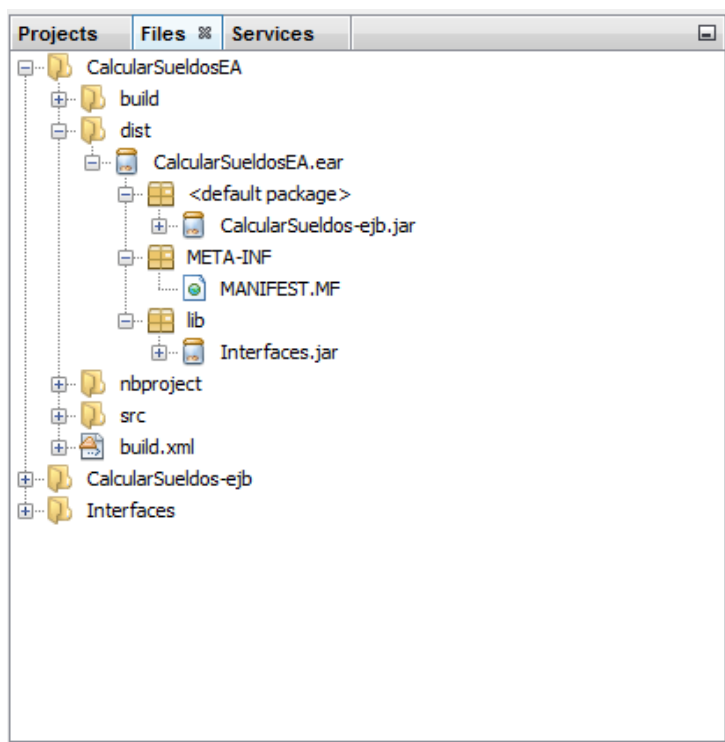
```
public class CalcularSueldoSB implements CalcularSueldoSBRemote {  
  
//Inyección de Dependencia  
  
    @EJB  
  
    private DescuentosIRPF SBLocal bLocal;  
  
    @Override  
  
    public double calcularSueldo(double sueldo) {  
        double descuento = bLocal.descuentoIRPF(sueldo);  
  
        return sueldo - descuento;  
  
    }  
  
}
```

Build y deploy del proyecto en el servidor de aplicaciones

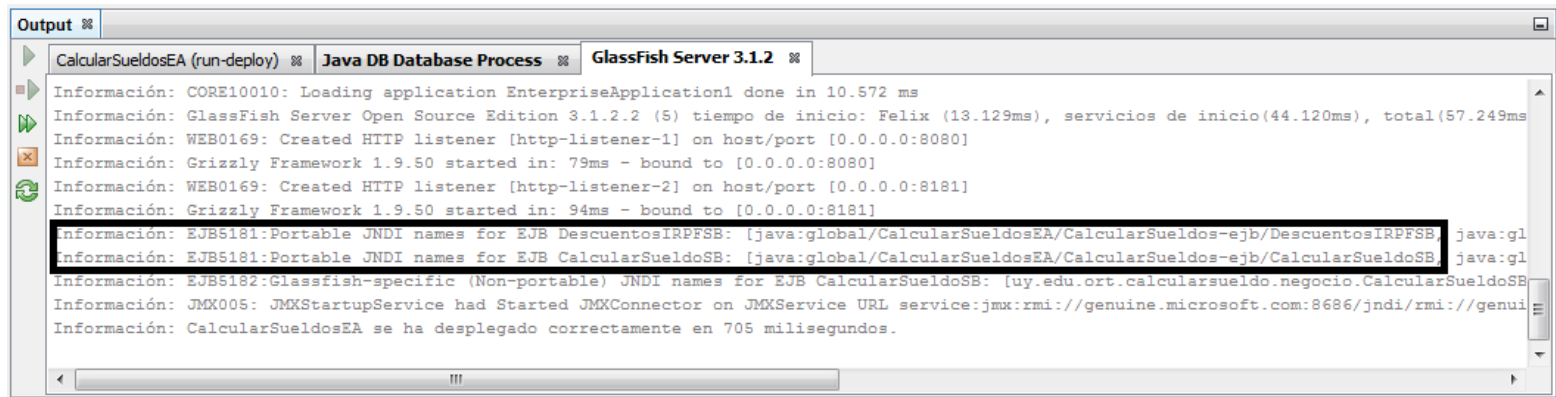
1. Posicionarse en la raíz del proyecto de nuestra enterprise application (**CalcularSueldosEA**) y seleccionar del menú contextual del mouse la opción **Build**. Se podrá verificar en la consola de NetBeans el empaquetamiento de los componentes y su ubicación para ser distribuidos al servidor:



2. En la vista Files de NetBeans, se podrá verificar que cada proyecto tiene creada una carpeta dist que contiene los componentes de nuestra aplicación:



3. Verificar que el server esté iniciado. Desde la raíz del proyecto de nuestra aplicación Enterprise seleccionar del menú contextual del mouse la opción **Deploy**. En la consola de NetBeans podrá verificarse el resultado del deploy:



Se registra en los Session beans en el JNDI con los nombre:

java:global/CalcularSueldosEA/CalcularSueldos-ejb/DescuentosIRPFEB

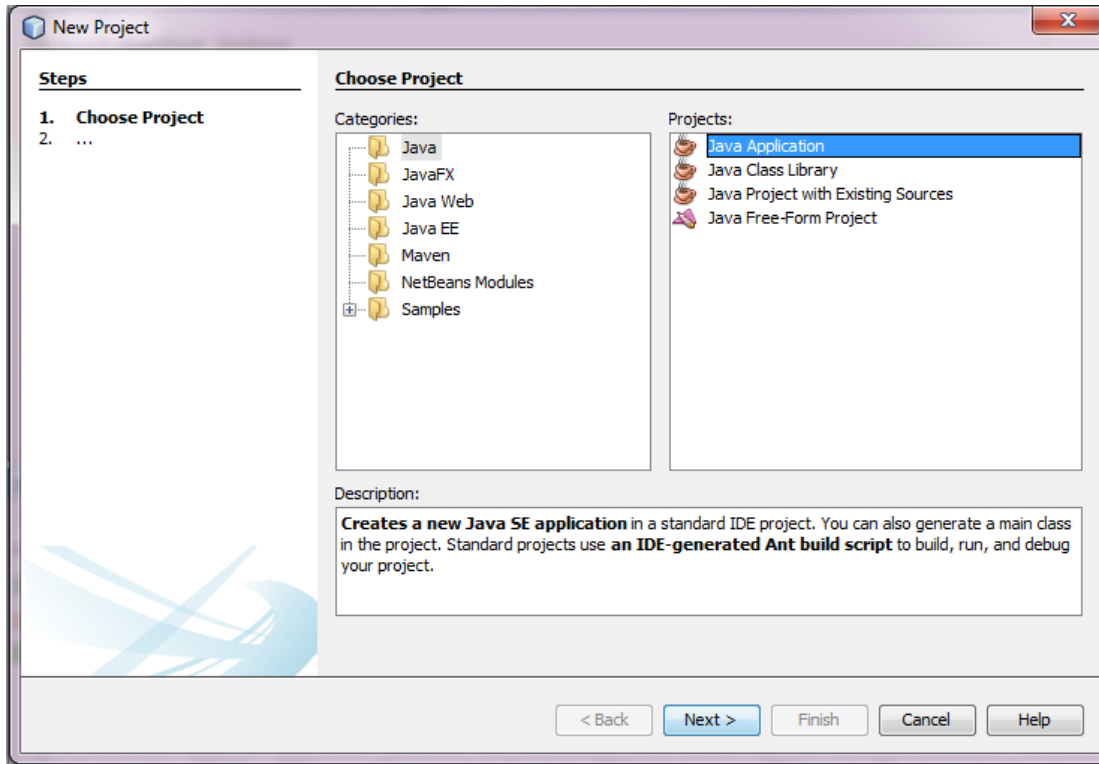
java:global/CalcularSueldosEA/CalcularSueldos-ejb/CalcularSueldoSB

Nombre por defecto:

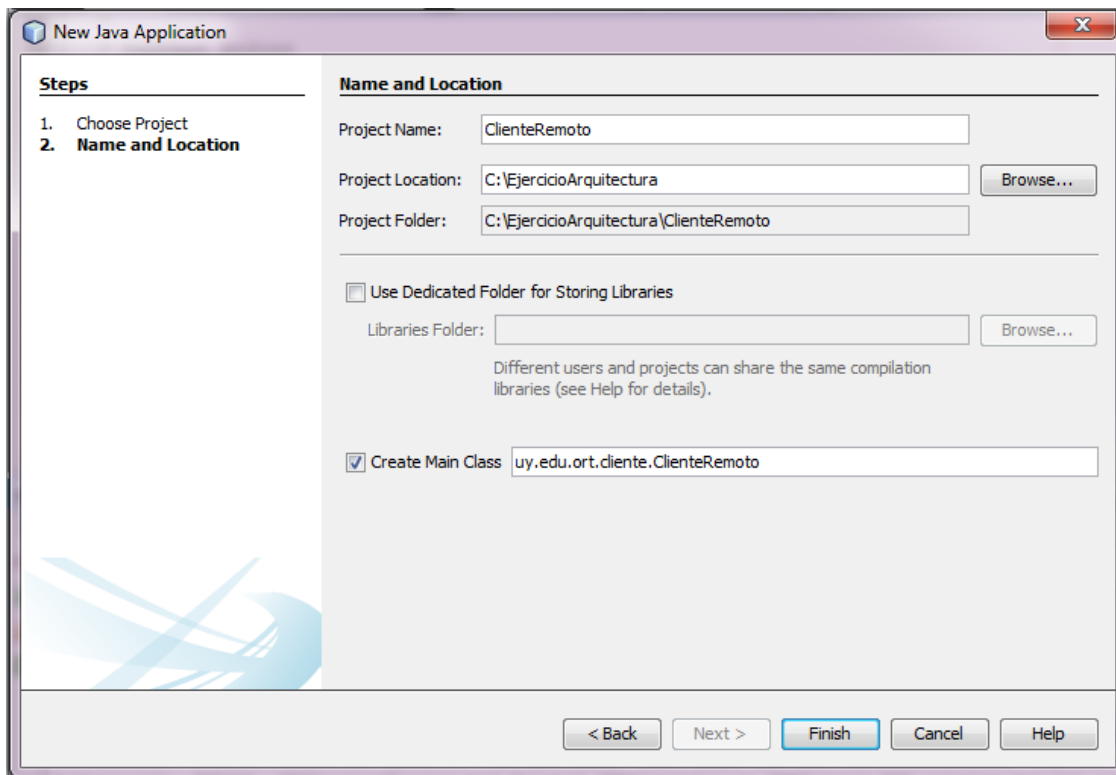
java:global/<Nombre del EA>/<Nombre del EJB>/<Nombre del SB>

Creación de la aplicación Cliente

1. Seleccionar del menú de NetBeans la opción para crear un nuevo proyecto. En la ventana que se muestra, seleccionar de la lista de categorías (a la izquierda) la opción *Java* y de la lista de proyectos la opción *Java Application*.

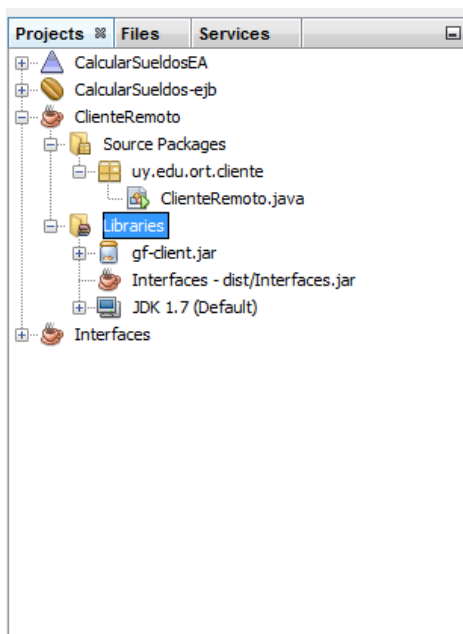


2. Ingresar **ClienteRemoto** como nombre de Proyecto, especificar la ubicación del directorio de trabajo, asignar el nombre **uy.edu.ort.cliente.ClienteRemoto** a la clase main del proyecto.

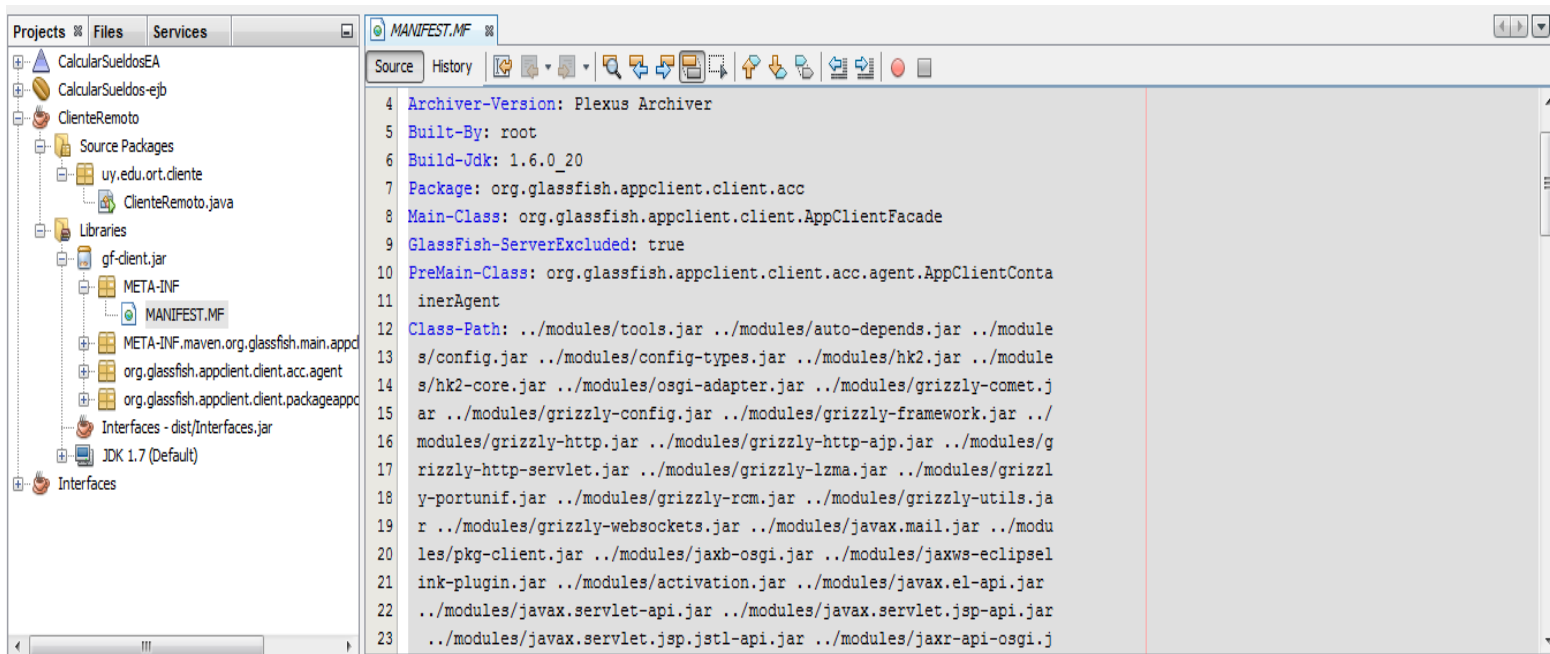


3. Posicionarse en la carpeta *Libraries* y agregar el JAR **gf-client.jar** que se encuentra en la ruta de instalación del servidor GlassFish 3.1.2 : *C:\Program Files (x86)\glassfish-3.1.2.2\glassfish\lib\gf-client.jar*

4. Posicionarse en la carpeta *Libraries* y seleccionar del menú contextual del mouse la opción **Add Project**. Seleccionar la ubicación del proyecto **Interface** (el proyecto que contiene las interfaces remotas de la aplicación) y confirmar.



Nota: Cuando se ejecute desde línea de consola el jar se debe agregar todas las librerías que hace desde el archivo MANIFEST.MF de la librería gf-client.jar



5. Código del ClienteRemoto

```
public class ClienteRemoto {  
  
    public static void main(String[] args) {  
  
        try {  
  
            //Seteo las ppropiedades del Contexto  
  
            Properties properties = new Properties();  
  
            properties.setProperty("org.omg.CORBA.ORBInitialHost", "localhost");  
  
            properties.setProperty("org.omg.CORBA.ORBInitialPort", "3700");  
  
            InitialContext context= new InitialContext(properties);  
  
            CalcularSueldoSBRemote servicio=(CalcularSueldoSBRemote)  
context.lookup("java:global/CalcularSueldosEA/CalcularSueldos-ejb/CalcularSueldoSB");  
  
            double sueldoTotal=servicio.calcularSueldo(300);  
  
            System.out.println(sueldoTotal);  
  
        } catch (NamingException ex) {  
  
            Logger.getLogger(ClienteRemoto.class.getName()).log(Level.SEVERE, null, ex);  
  
        }  
  
    }  
  
}
```

Build y ejecución de la aplicación cliente

1. Posicionarse en la raíz del proyecto Cliente y seleccionar del menú contextual del mouse la opción Build. Se podrá verificar en la consola de NetBeans el empaquetamiento de los componentes y su ubicación para ser distribuidos al servidor:
2. Posicionarse en la clase Cliente y seleccionar del menú contextual del mouse la opción Run File
3. Verificar el resultado del llamado en el log del proyecto en la consola de NetBeans