

## Guía 3-A

### Como crear y consumir servicios con Angular 8, utilizando arquitecturas middleware en Java EE 8 con Jax RS.



Los objetivos de esta guía son:

- Definir que es arquitectura middleware (tipos de herramientas).
- Aprender a crear un middleware en Java EE 8 con Jax RS.
- Aprender a integrar PrimeNG a Angular 8.
- Aprender a crear una clase de entidad o representación de una tabla en Angular.
- Aprender a crear un servicio en Angular.
- Aprender a crear la lógica para un CRUD o ABM para una entidad.
- Aprender a crear un front-end con un ejemplo CRUD usando PrimeNG.

Para logra dichos objetivos dividí la lección en dos partes:

La lección 3-A en como crear un servicio middleware en Java EE 8.

La lección 3-B se desarrollarán los siguientes temas:

- Como integrar PrimeNG 8 en Angular 8.
- Como crear clases de tipo entidad
- Como crear los servicios con angular
- Como definir la lógica de programación en Angular 8 (MVC).
- Como crear la Vista o Front-End Final al usuario.

## Lección 3-A

Creación de un servicio middleware en Java EE 8.

Esta lección contiene la documentación técnica para la creación del servicio middleware en Java EE 8.

-

El código fuente está en la siguiente dirección:

<https://github.com/arfloreshn/Leccion3-A-Curso-Angular-PrimefacesNG>

El script para crear las tabla de la base de datos, se encuentra dentro del repositorio de github

arfloreshn Script para crear Tablas ...		Latest commit 5734878 now
lib/unknown/binary	25102019 backend cines	10 days ago
src/main	04112019, RESPALDO	2 hours ago
target	04112019, RESPALDO	2 hours ago
Dockerfile	25102019 backend cines	10 days ago
Leccion3 Parte 1 Creando Back End con ...	04112019, LECCION3A BACKEND MIDDLEWARE JAVA EE 8	1 hour ago
README.md	25102019 backend cines	10 days ago
ScriptCrearTablas.sql	Script para crear Tablas	now
build-and-run.sh	25102019 backend cines	10 days ago
nb-configuration.xml	04112019, RESPALDO	2 hours ago
pom.xml	25102019 backend cines	10 days ago

Corresponde a cada interesado lo siguiente:

- ✓ Bajar las fuentes del middleware.
- ✓ Bajar y configurar el motor de bases de datos.
- ✓ Ejecutar los scripts de creación e importación de la data para realizar el ejercicio de la lección 3.

Esta lección es un explicativo de la construcción de los servicios middleware en Java EE 8, usaremos solo el estándar Java EE 8.

Se autoriza el uso de esta documentación a la comunidad de programadores, maestros de educación superior y centros de enseñanza técnica no formal, con el fin de que el hombre de Dios sea preparado para toda buena obra, se prohíbe el uso de esta documentación para fines de lucro personal o venta sin autorización del autor.

Requisitos técnicos mínimos:

El IDE de desarrollo que usaremos será en Netbeans 8.2 en español o superior.

JDK 1.8 (221u)

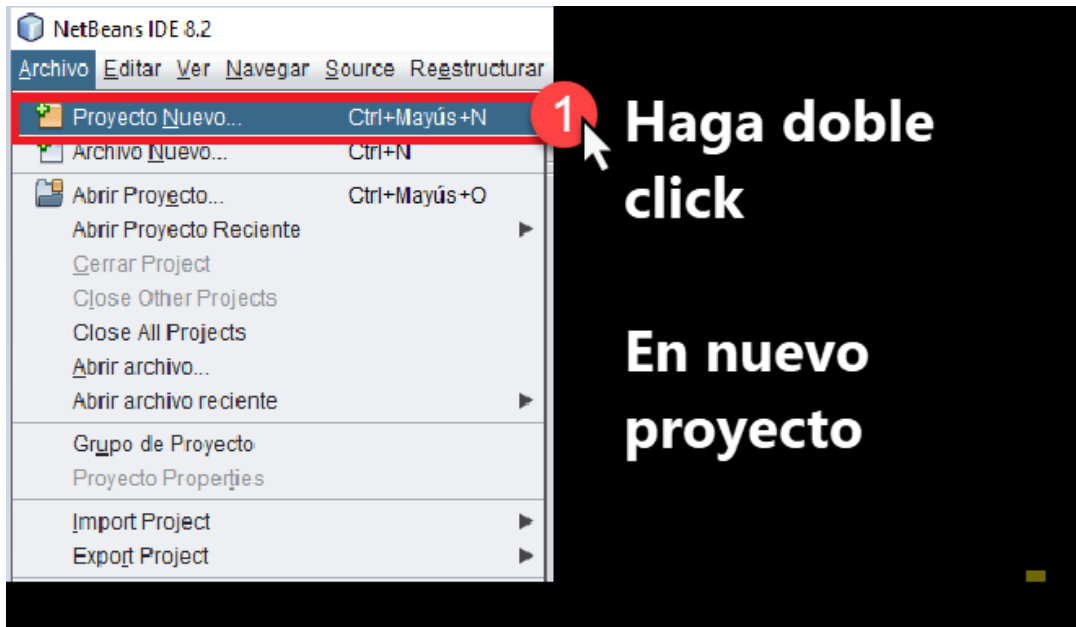
Eclipse Glassfish 5.1 versión completa.

Sugerimos el uso de Payara Server ya que soporta microservicios y este proyecto es 100% compatible con Payara Server y WildFly 17.

## GUIA RAPIDA BACK-END

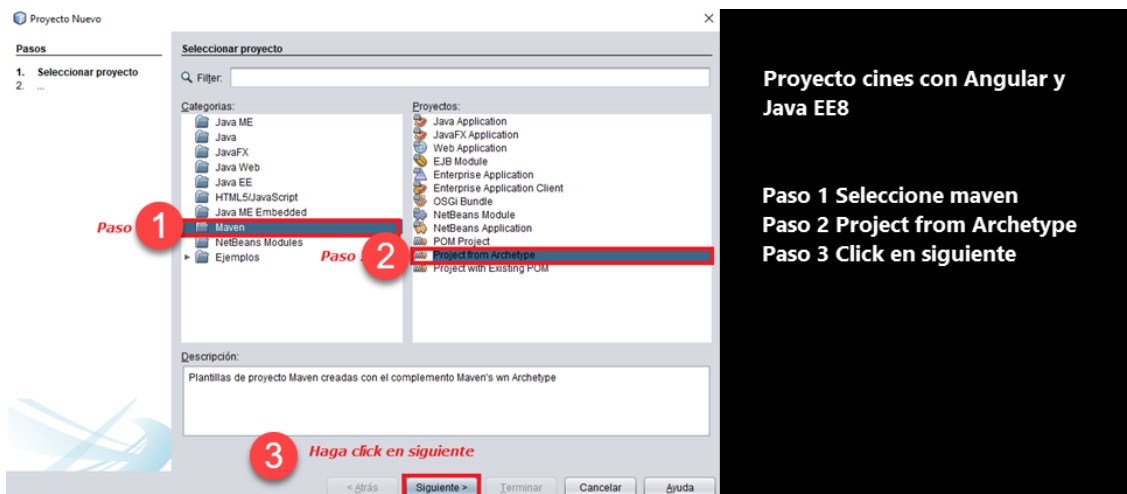
### Paso 1

#### Crear nuevo proyecto



### Paso 2

#### Seleccionaremos proyecto de arquetipo maven.



## Paso 3, buscando el arquetipo maven

Proyecto Nuevo

Pasos

1. Seleccionar proyecto
2. Arquetipo Maven
3. Nombre y ubicación

Arquetipo Maven

Search:  ☐ Show Older

Known Archetypes:

**jee8project-archetype**

*Selección o haga click sobre jee8project-archetype*

Group ID:  Artifact ID:

Version:  Repository:

Description:

Archetype for building a basic JavaEE8 project.

**3** Haga click en siguiente

< Atrás **Siguiente >** Terminar Cancelar Ayuda

### Paso 1.

Escriba jee8 en search usaremos el arquetipo de mayor uso en la web para Java EE8, hay muchos otros arquetipos web, pero este en específico esta preparado para microservicios.

### Paso 2

Seleccione el arquetipo

### Paso 3

Click en siguiente

## Paso 4, creando nuestro proyecto maven una vez seleccionado el arquetipo Maven.

Proyecto Nuevo

Pasos

1. Seleccionar proyecto
2. Arquetipo Maven
3. Nombre y ubicación

Nombre y ubicación

Nombre del Proyecto:  **1**

Ubicación del Proyecto:  Examinar...

Carpeta del proyecto:

Id del Artefacto:

Id del Grupo:  **2**

Version:

Paquete:  **3** (onall)

*Escribiremos el perfil de nuestra aplicación para cines con JavaEE 8, siga los pasos tal como los ilustramos, para que no tener retrasos.*

**4** Haga click en terminar

< Atrás **Terminar** Siguiente > Cancelar Ayuda

1) Ingrese el nombre del proyecto: cines

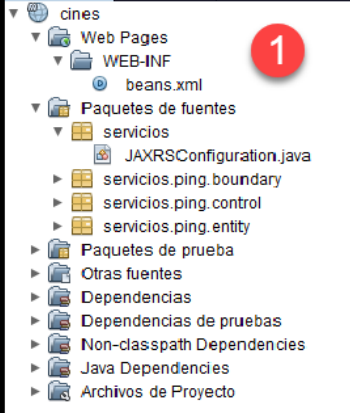
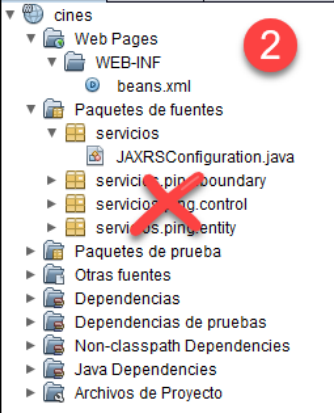
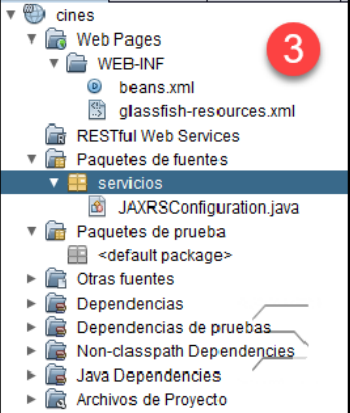
2) escriba el inverso de grupo de maven: com.cine.hn

3) escriba el nombre del paquete de los webservices: servicios

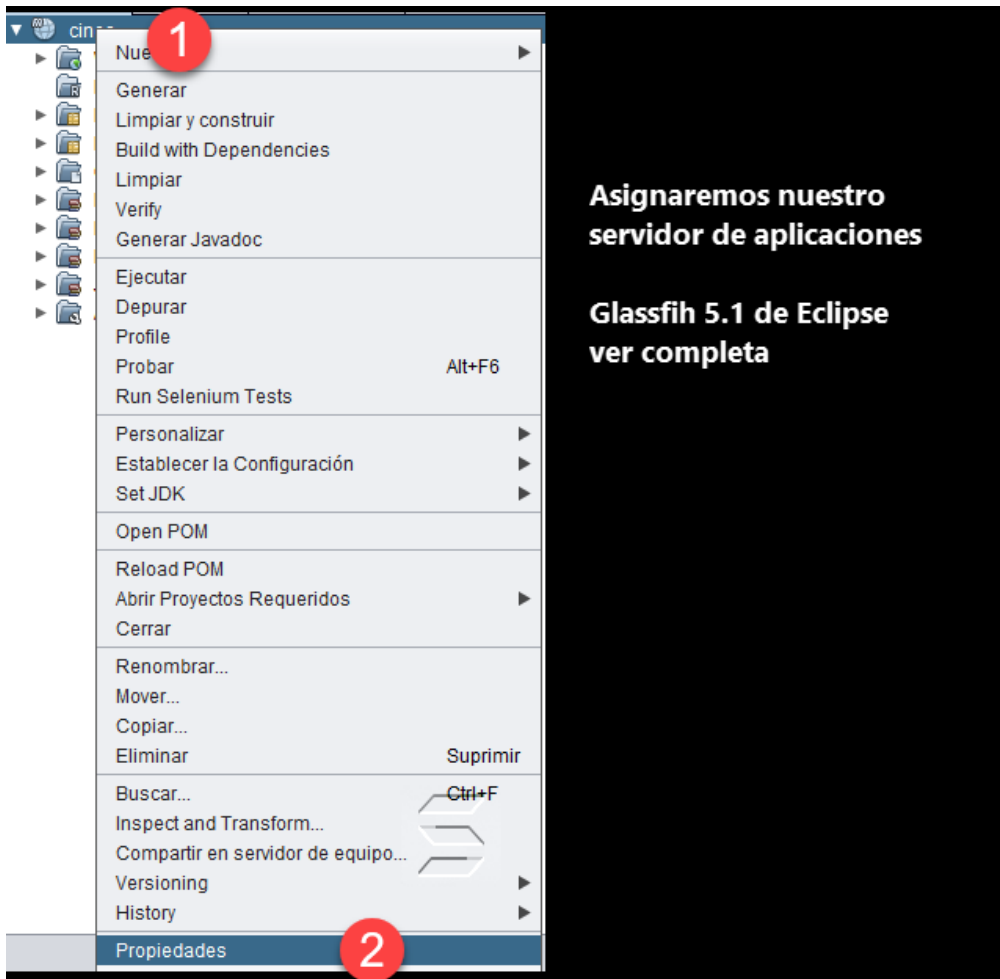
NOTA: En java llamamos servicios a los webservices, para que se familiarice con el habla en java.

## Paso 5, Depurando el proyecto Maven.

Eliminamos los paquetes de prueba creados por maven, con excepción de **JAXRSConfiguration.java**

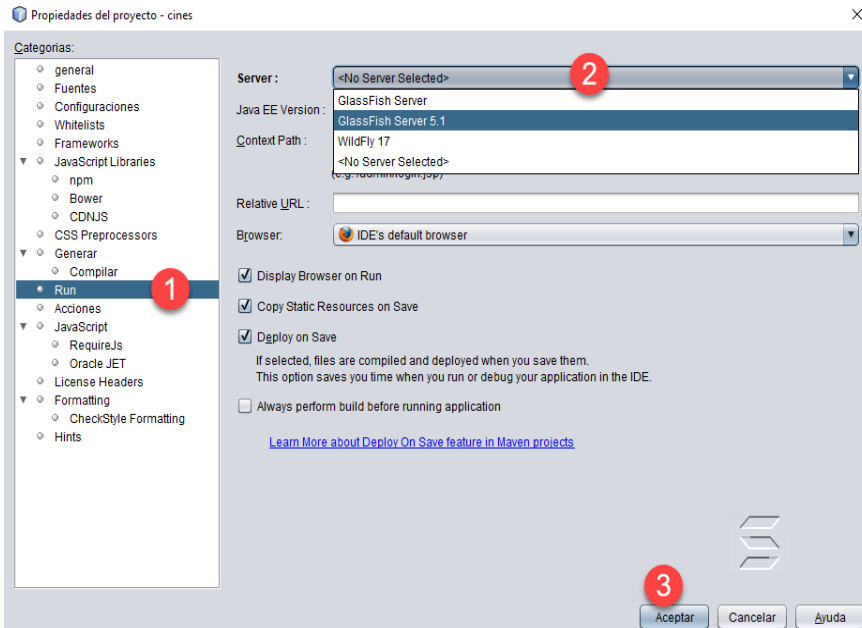
PROYECTO CINES	BORRE LOS PAQUETES PING	RESULTADO
		

Paso 6, asignando nuestro servidor de aplicaciones al proyecto, esto se hace para evitar inconvenientes con la creación de nuestro archivo de unidad de persistencia.



Paso 7, asignación del servidor de aplicaciones.

Asignaremos un servidor de aplicación a nuestro middleware de cines



1) Seleccione RUN

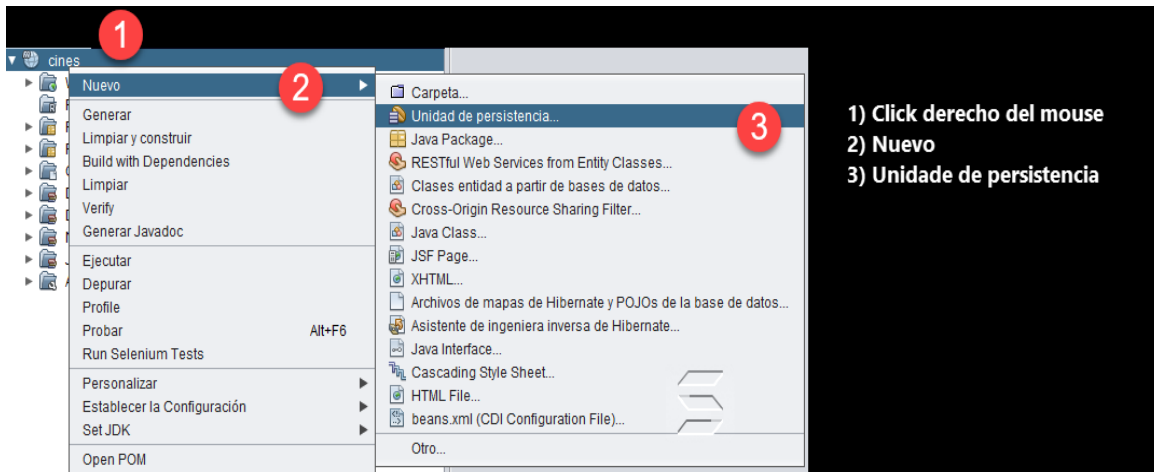
2) Seleccione el servidor de aplicaciones

3) Aceptar

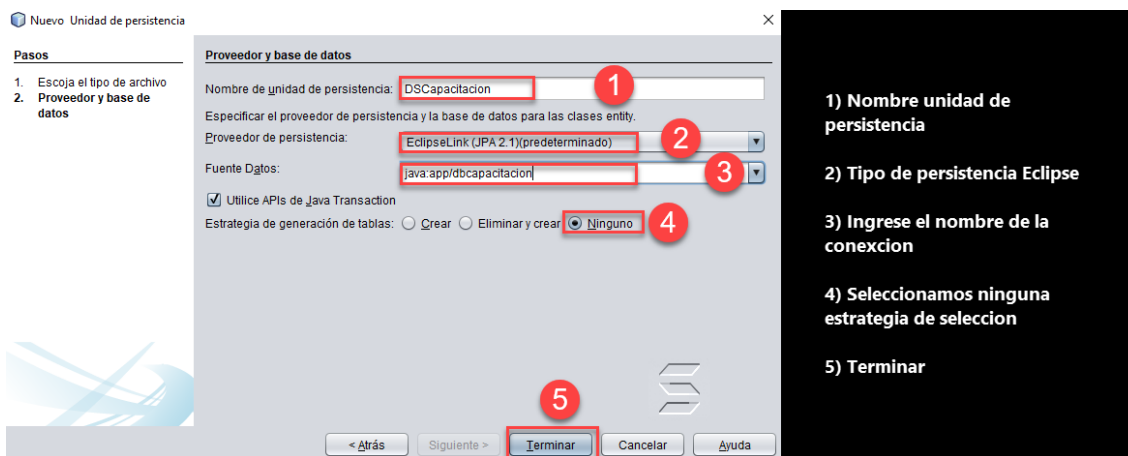
**NOTA:** La instalación del servidor de aplicaciones, no la documentaremos debido a que existen suficientes videos en internet de como agregar glassfish o wildfly a NetBeans



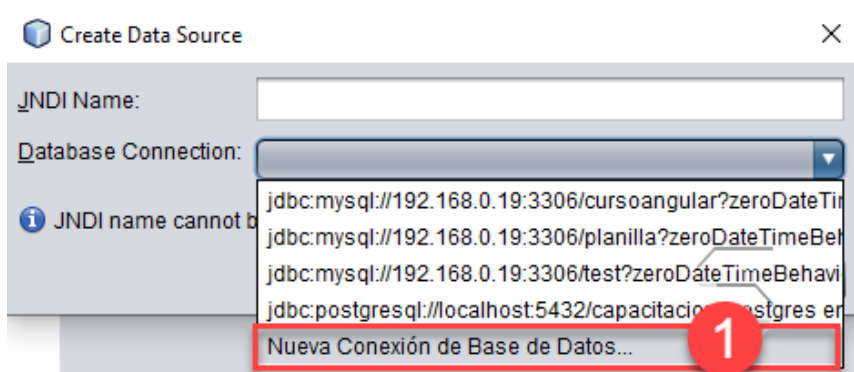
Paso 8, Crearemos la unidad de persistencia de nuestro middleware.



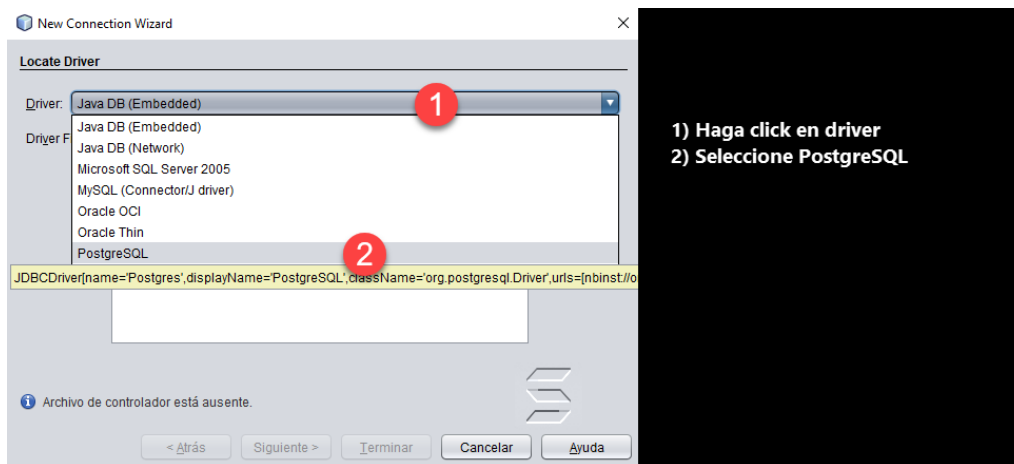
Paso 9, Definición de la unidad de persistencia de nuestro middleware.



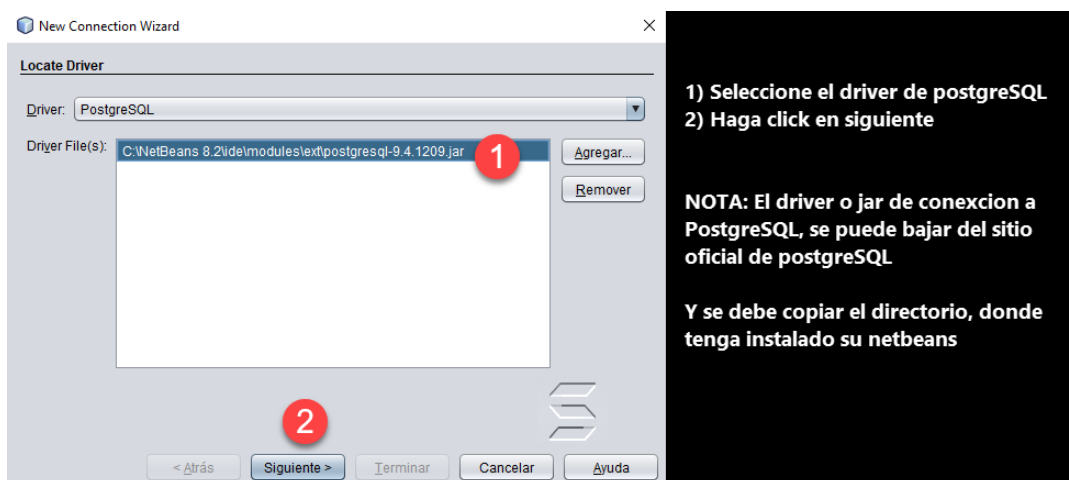
Paso 10, Creando conexiones de base datos para nuestro middleware.



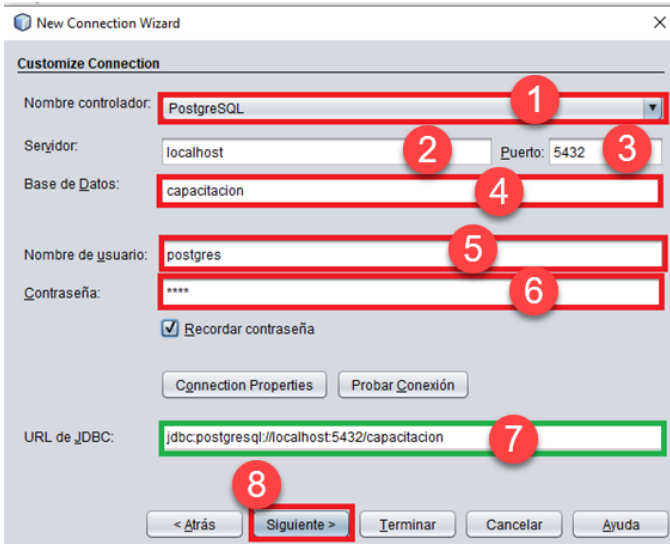
Paso 11, Seleccionado en driver de conexión de la base de datos.



Paso 11, Seleccionado en driver de conexión de la base de datos.



## Paso 12, Parametrización de la conexión a base datos con PostgreSQL.



1) seleccionamos postgresQL  
2) ingresamos la dirección IP del servidor de base de datos  
3) ingresamos el puerto de conexión  
4) ingresamos el nombre de la base de datos.  
5) ingresamos el usuario autorizado  
6) ingresamos la contraseña  
8) Hacemos click en siguiente

## Paso 13, Seleccionando el esquema public



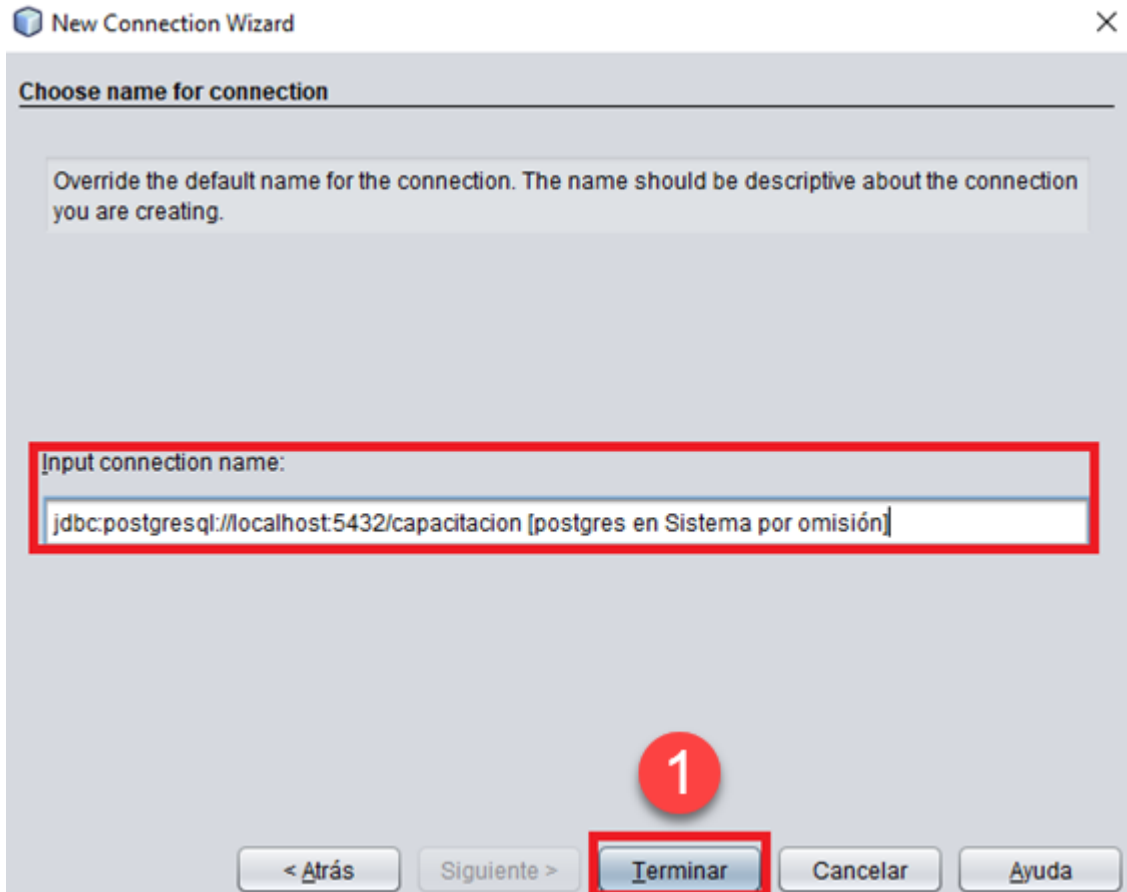
**1 SELECCIONE EL ESQUEMA PUBLIC DE POSTGRESQL**

Selección de esquema: public

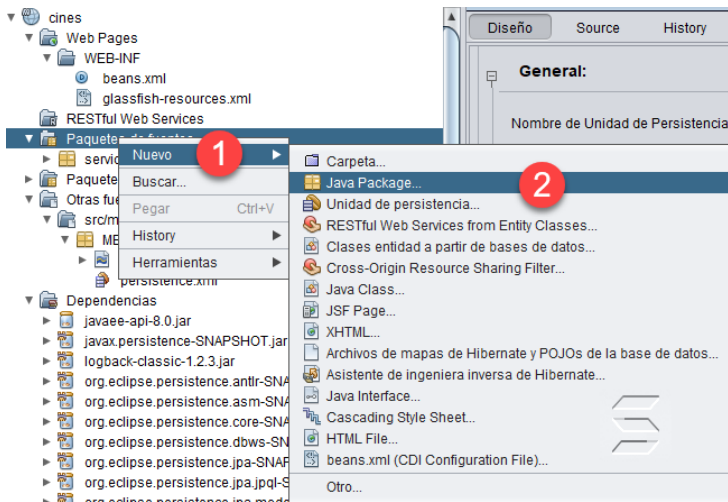
**HAGA CLICK EN SIGUIENTE**

**2**

Paso 14, finalizando la parametrización de la conexión a la base de datos

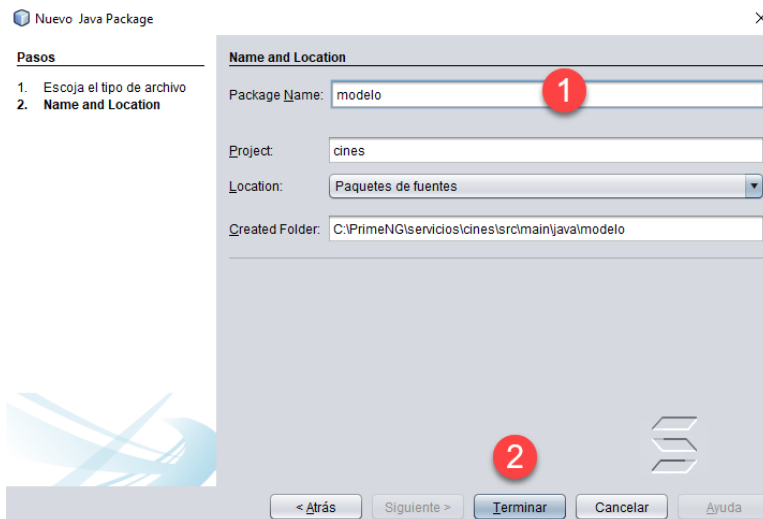


Paso 15, Creando nuestro paquete modelo, el paquete modelo es donde estará nuestros pojos que representan cada una de las tablas o entidades de la base de datos



1) Haga un click sobre nuevo paquete.

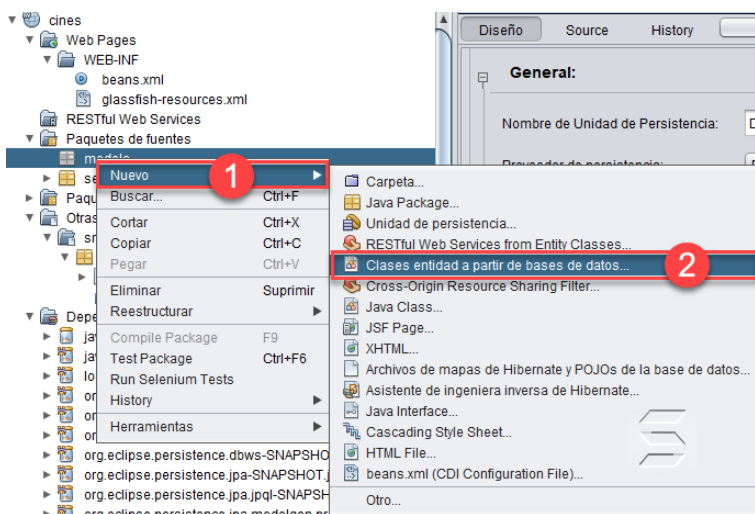
2) Seleccione Java Package



1) Escriba modelo

2) Haga click en terminar

Paso 16, Creamos nuestro pojo a partir de la base de datos dentro del paquete modelo.



Crearemos nuestro pojos, a partir de la base de datos

1) Click derecho nuevo.

2) Clases de entidad a partir de base de datos

## Paso 17, Seleccionamos la conexión de la base de datos.

Nuevo Clases entidad a partir de bases de datos

**Pasos**

1. Escoja el tipo de archivo
2. **Tablas de bases de datos**
3. Clases entidad
4. Opciones de mapeo

**Tablas de bases de datos**

Fuente de Datos: **java.app/dbcapacitacion** 1

Tablas disponibles:

- java.comp/Default
- jdbc/\_default
- jdbc/\_TimerPool
- jdbc/sample
- New Data Source...

Agregar todo >>

<< Eliminar todo

Cualesquiera

☒ Incluir tablas relacionadas

Seleccionar el origen de la tabla.

< Atrás    Siguiente >    Terminar    Cancelar    Ayuda

**Seleccionaremos nuestra conexión a la base de datos de capacitacion**

## Paso 18, agregamos todas las tablas a la base de datos.

Nuevo Clases entidad a partir de bases de datos

**Pasos**

1. Escoja el tipo de archivo
2. **Tablas de bases de datos**
3. Clases entidad
4. Opciones de mapeo

**Tablas de bases de datos**

Fuente de Datos: java.app/dbcapacitacion

Tablas disponibles:

- actores
- actores\_interpretacion
- canal\_venta
- formas\_pago
- funciones
- países
- películas
- películas\_categorias
- precios\_vta
- promociones
- salas

Agregar >

< Eliminar

Agregar todo >> 1

<< Eliminar todo

Cualesquiera

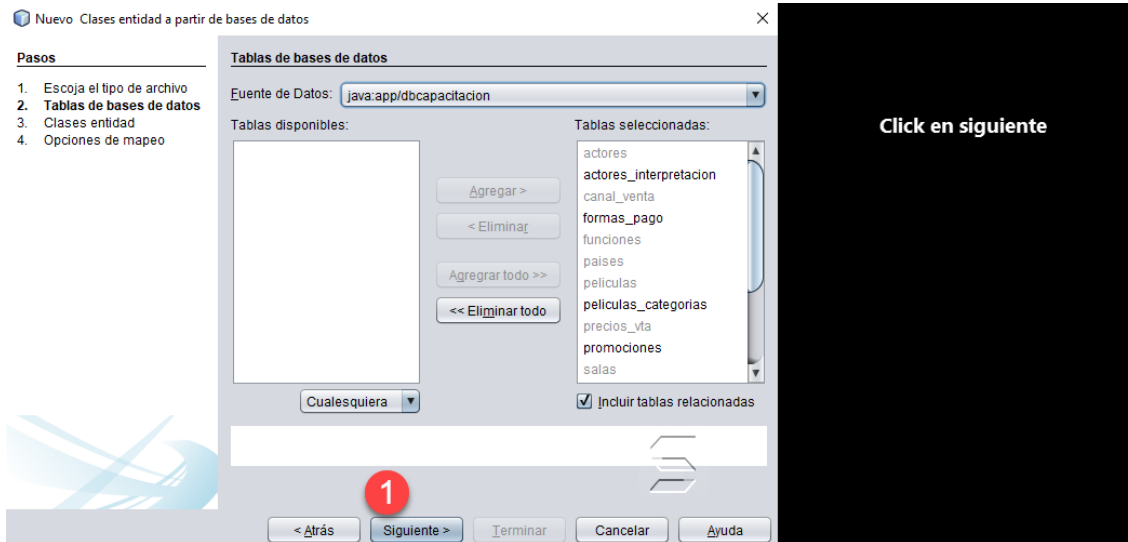
☒ Incluir tablas relacionadas

Seleccionar al menos una tabla.

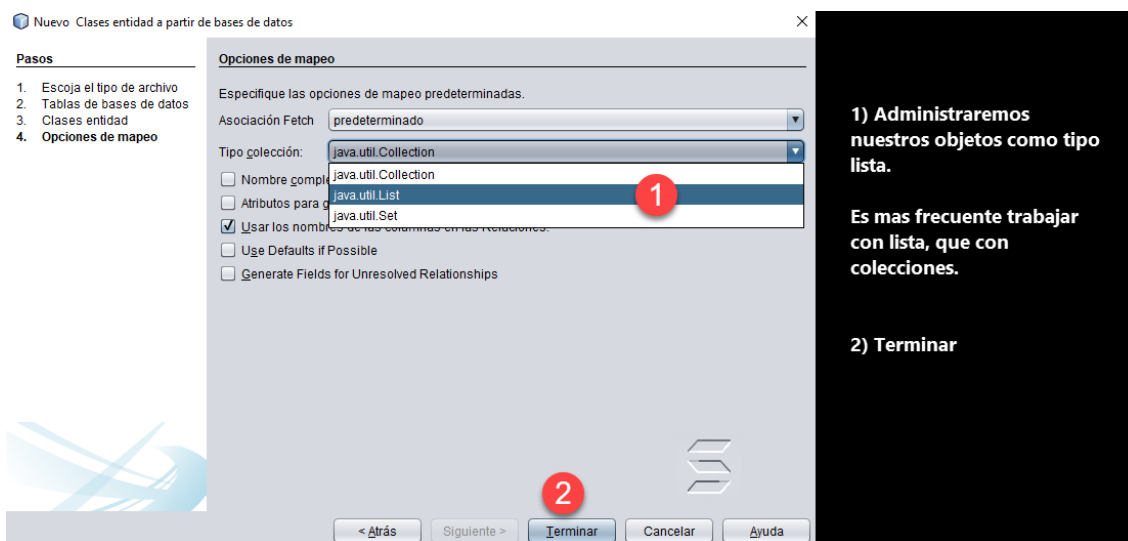
< Atrás    Siguiente >    Terminar    Cancelar    Ayuda

**Agregamos todas las entidades**

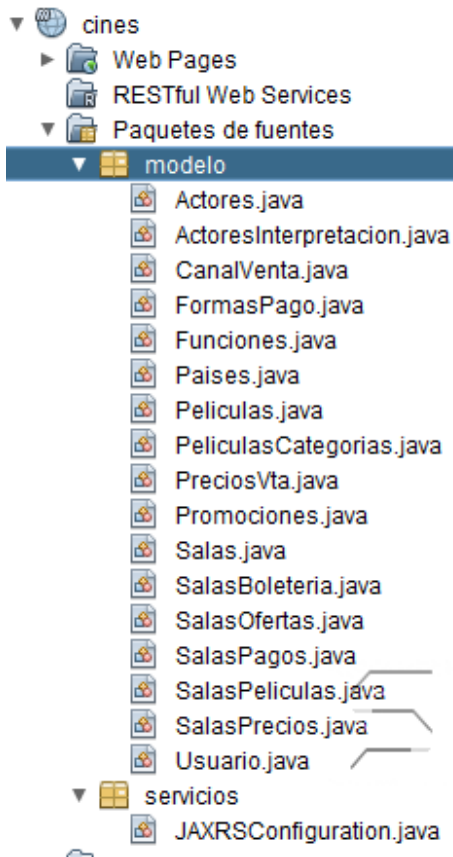
Paso 19, finalizamos el proceso de creación de los pojos que representan a las tablas en la base de datos.



Paso 20, Definiremos nuestros objetos como tipo lista para un mejor manejo en la programación.

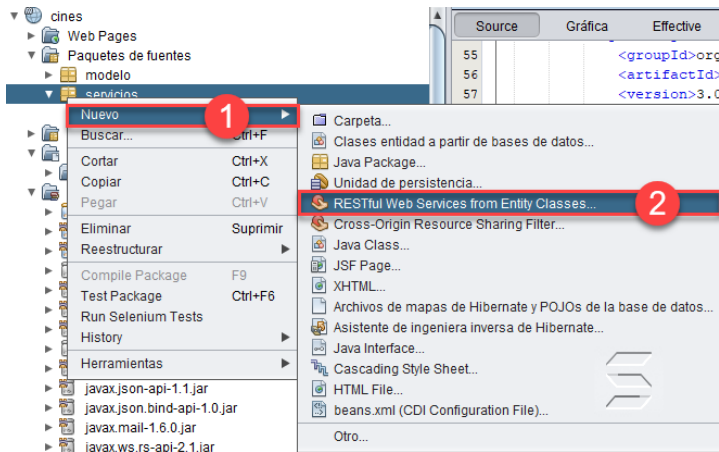


## Paso 21, Resultado de la creación



## Resultado

## Paso 22, Creando los endpoint de nuestro middleware con Restful

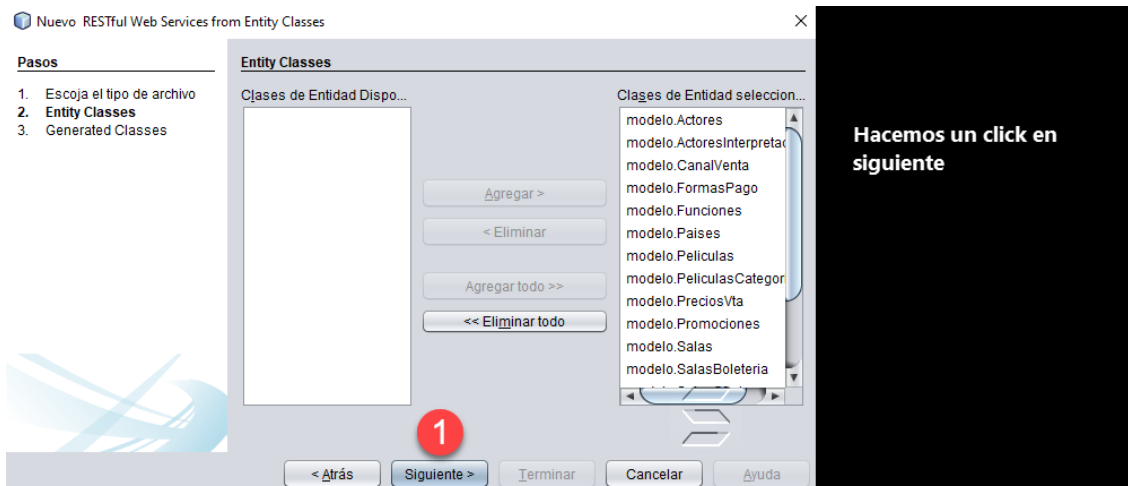
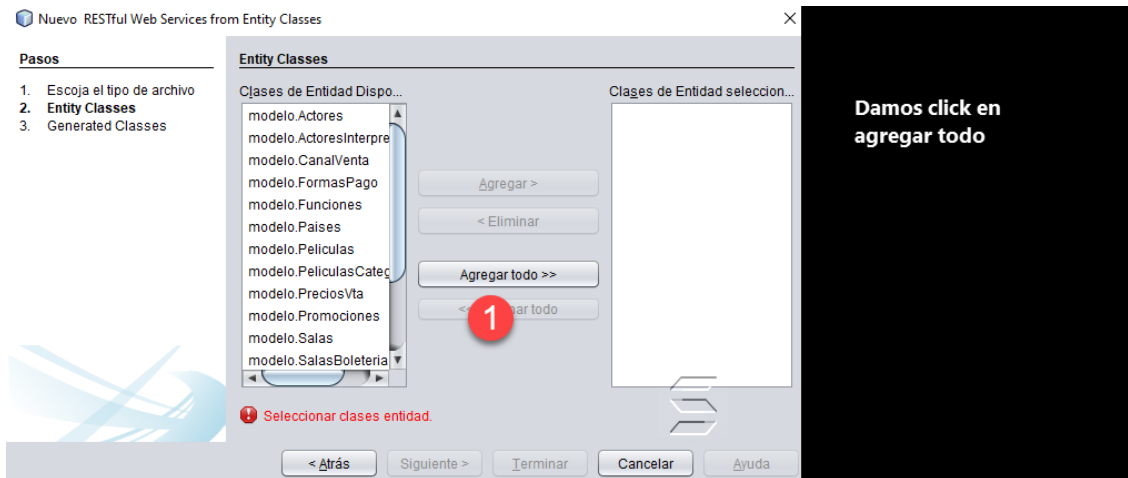


1) Haga click derecho sobre servicios.

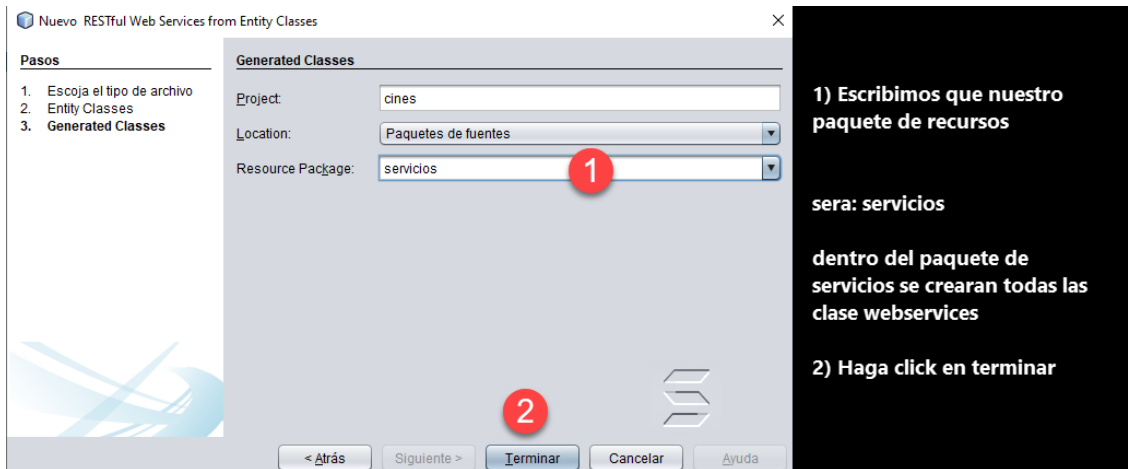
2) Seleccione Restful Web Service from Entity Classes



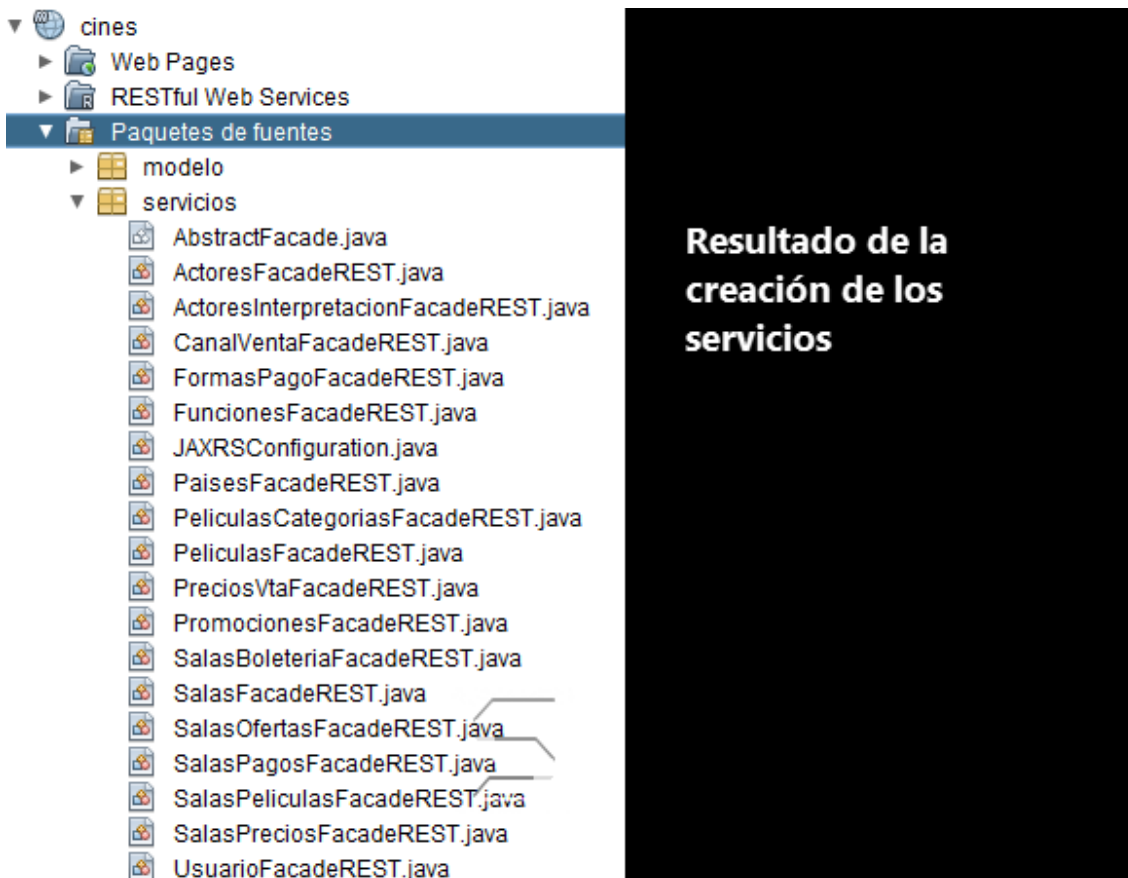
Paso 23, Seleccionaremos todos los pojos que hemos creado en el paquete modelo, para crear nuestros servicios middleware.



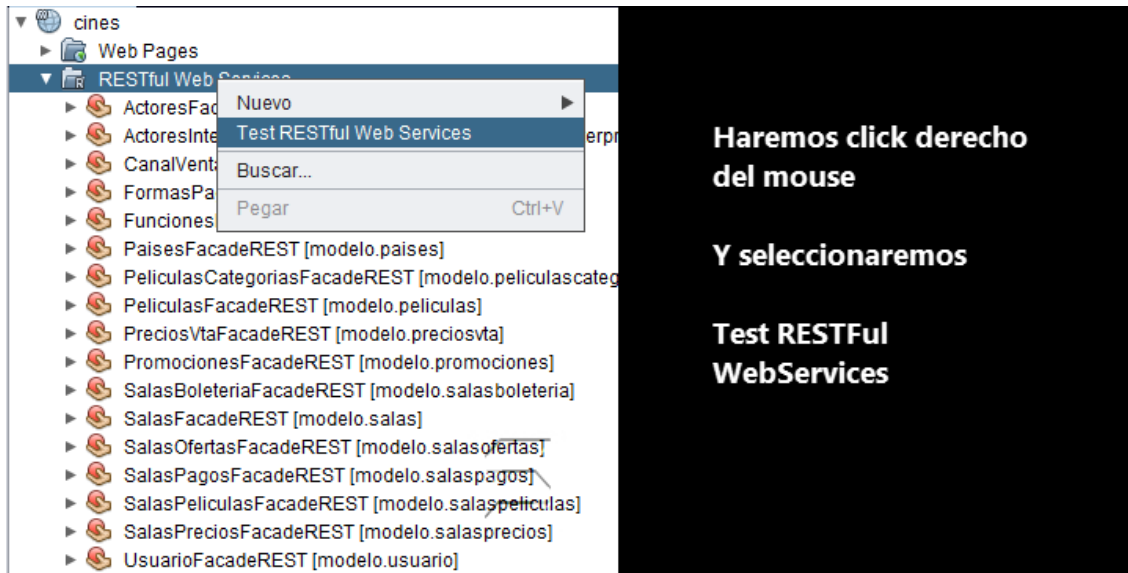
Paso 24, Definimos en que paquete vamos a guardar nuestro pojos de servicio o endpoint de RESTFUL.



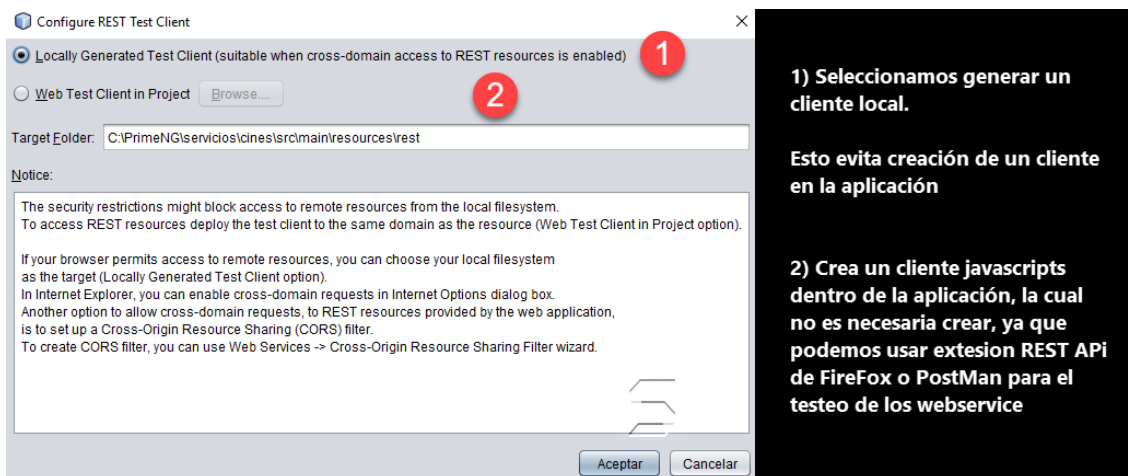
Paso 25, resultado de la creación de los endpoint.



Paso 25, para testear los endpoints basta con hacer click derecho en la carpeta de RESTful Web Services, eso levantara el servidor de aplicaciones.



Seleccionamos el tipo de cliente que usaremos.



WADL : <http://localhost:8080/cines/api/application.wadl>

### Test RESTful Web Services

- cines
  - modelo.canalventa
  - modelo.salasprecios
  - modelo.actores
  - modelo.funciones
  - modelo.peliculas
  - modelo.salaspeliculas
  - modelo.formaspago
  - modelo.promociones
  - modelo.peliculascategorias
  - modelo.salasboleteria
  - modelo.paises
  - modelo.usuario
  - modelo.actoresinterpretacion
  - modelo.salas
  - modelo.salaspagos
  - modelo.preciosvta
  - modelo.salasofertas

cines > modelo.usuario

Resource: modelo.usuario  
(<http://localhost:8080/cines/api/modelo.usuario>)

Choose method to test:

**La dirección**

http://localhost:8080/cines/api  
proviene de la clase **InicializarRestFull** al usar la etique: **@javax.ws.rs.ApplicationPath("api")**

**La direccion**

http://localhost:8080/cines/api/model.usuario  
proviene de la clase **UsuarioWS** al usar la etique **@path("modelo.usuario")**

Los clientes Postman y ResClient, solo pueden ser usados mientras en servidores de aplicaciones esta levantado y el middleware este corriendo dentro del servidor de aplicaciones.

## RestClient Mozilla

RESTClient

Test RESTful Web Services

localhost:8080/cines/api/modelo.usuario

Extensión (RESTClient) moz-extension://e4d1dd97-cf69-4997-a6b9-255c0571f172/index.html

Más visitados Java Comenzar a usar Firefox PostGRES Primefaces JQuery PL/SQL Jboss Cursos Pagados Cluster Sistran Prestamos Oracle FireWall Mysql Glassfish

Authentication Headers View Favorites Data migration RESTClient

**[ - ] Request**

Method: GET URL: <http://localhost:8080/cines/api/modelo.usuario>

Body

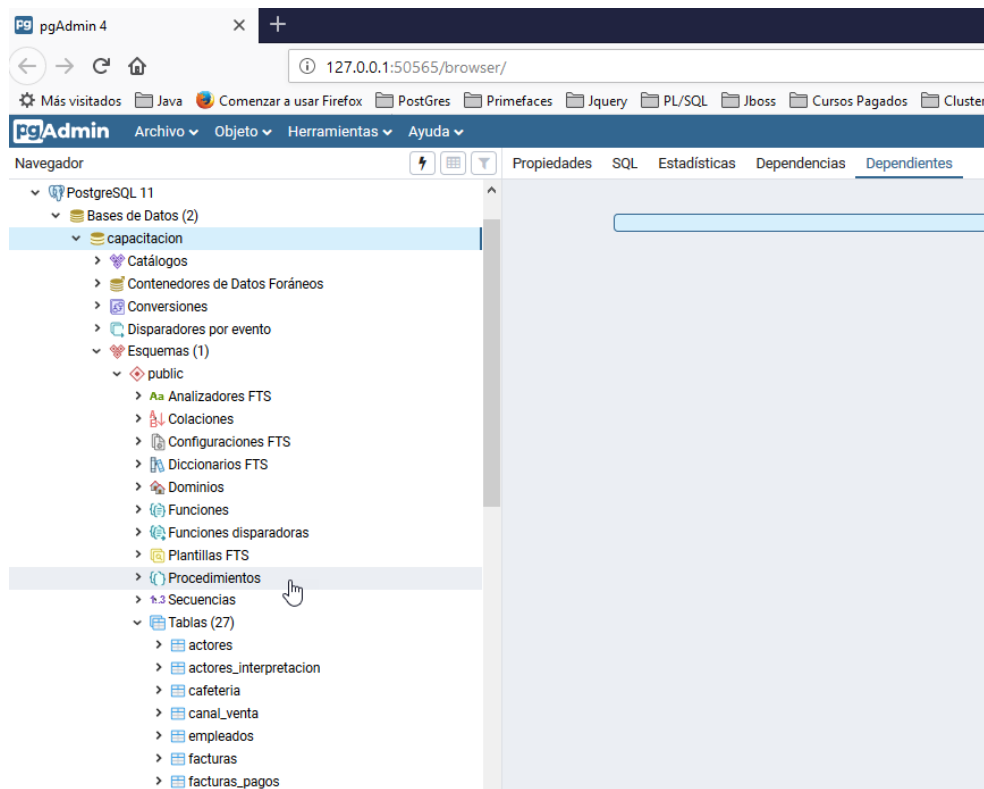
Request Body

**[ - ] Response**

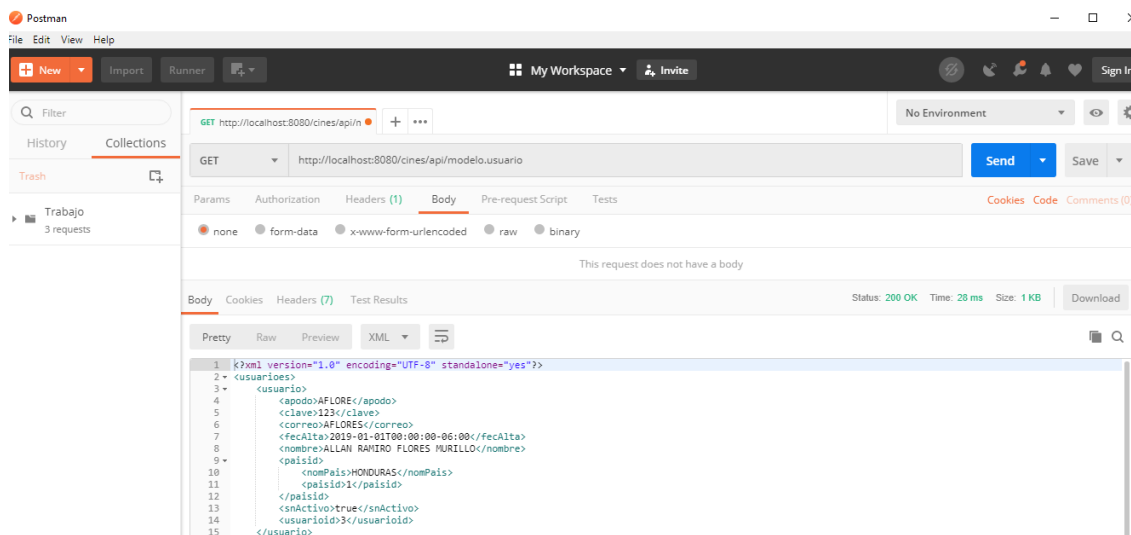
Headers Response Preview

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?><usuarios><usuario><apodo>AFLORE</apodo><clave>123</clave><correo>AFLORES</correo><fechaAlta>2019-01-01T00:00:00-06:00</fechaAlta><nombre>ALLAN RAMIRO FLORES MURILLO</nombre><paisid><nomPaís>HONDURAS</nomPaís><paisid></paisid></usuario><usuario><apodo>ISOLORZANO</apodo><clave>123</clave><correo>ISOLORZANO</correo><fechaAlta>2019-01-01T00:00:00-06:00</fechaAlta><nombre>INDIRA SARIAN SOLORZANO CASTRO</nombre><paisid><nomPaís>HONDURAS</nomPaís><paisid></paisid></usuario></usuarios>
```

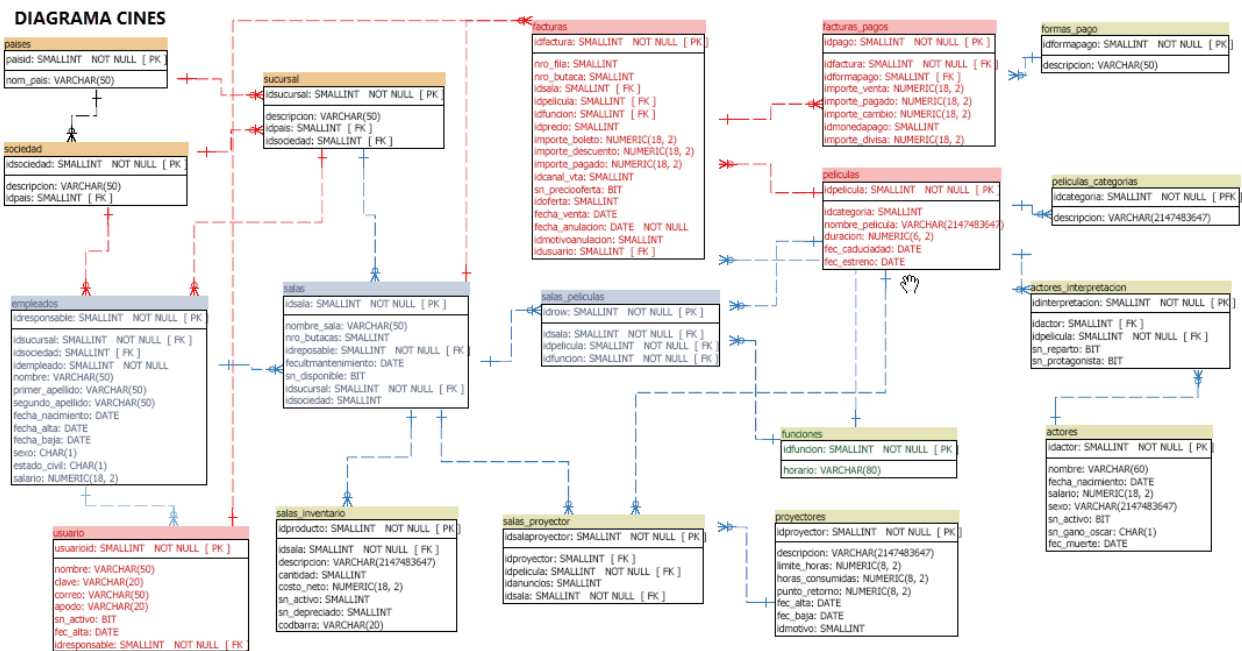
## PGADMIN4 v4 y PostgreSQL 11



## Postman



## Creación del modelo entidad relación



El color rojo identifica las áreas críticas del módulo de cines la app no debe tener ningún error, sobre todo en lo operativo una vez concluidas las pruebas de QA.

Para crear modelos entidad relación puede utilizar Sql Power Architect versión comunitaria.

## Sql Power Architect

diagramaCines - SQL Power Architect

File Edit Connections ETL OLAP Enterprise Tools Profile Window Help

