**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:

Bájelo y adjúntelo en su IDE o entorno de trabajo para realizar el ejercicio de la Lección 3

El Script de la base de datos, tablas y datos están dentro del repositorio de github:

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.

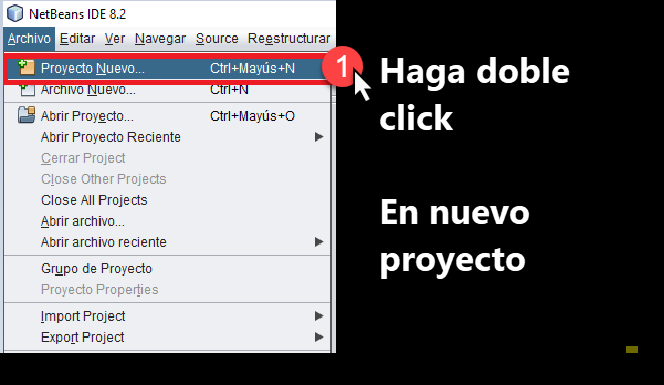
Sugerimos el uso de Payara Server ya que soporta microservicios y este proyecto es 100% compatible con Payara Server y WildFly 17.

GUIA RAPIDA

BACK-END

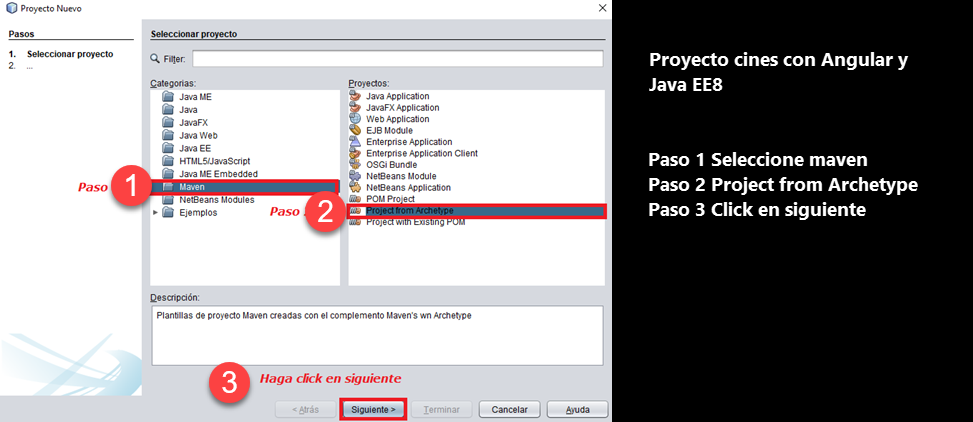
Paso 1

Crear nuevo proyecto

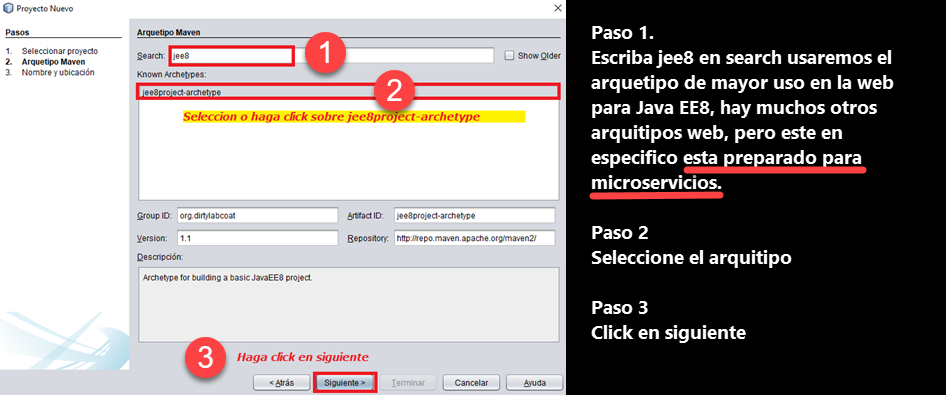


Paso 2

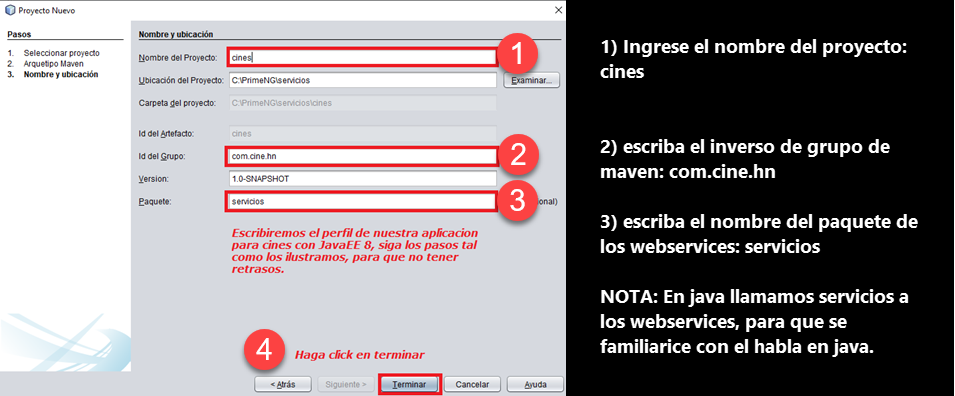
Seleccionaremos proyecto de arquetipo maven.



Paso 3, buscando el arquetipo maven

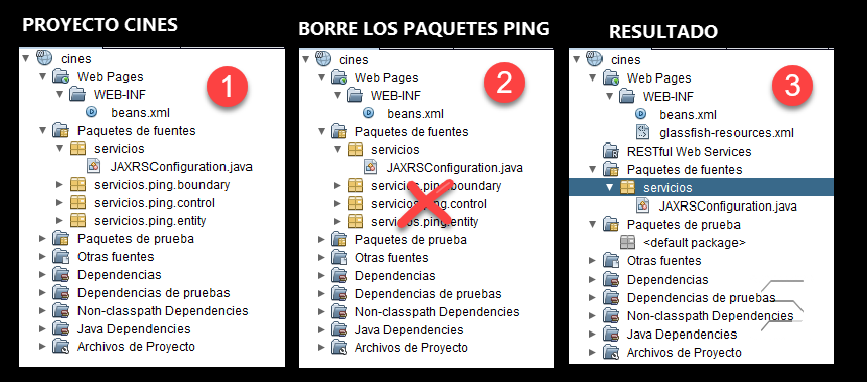


Paso 4, creando nuestro proyecto maven una vez seleccionado el arquetipo Maven.

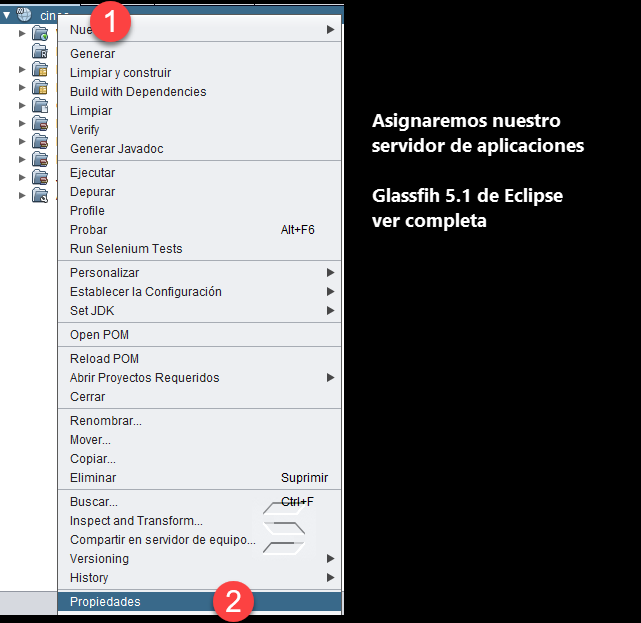


Paso 5, Depurando el proyecto Maven.

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

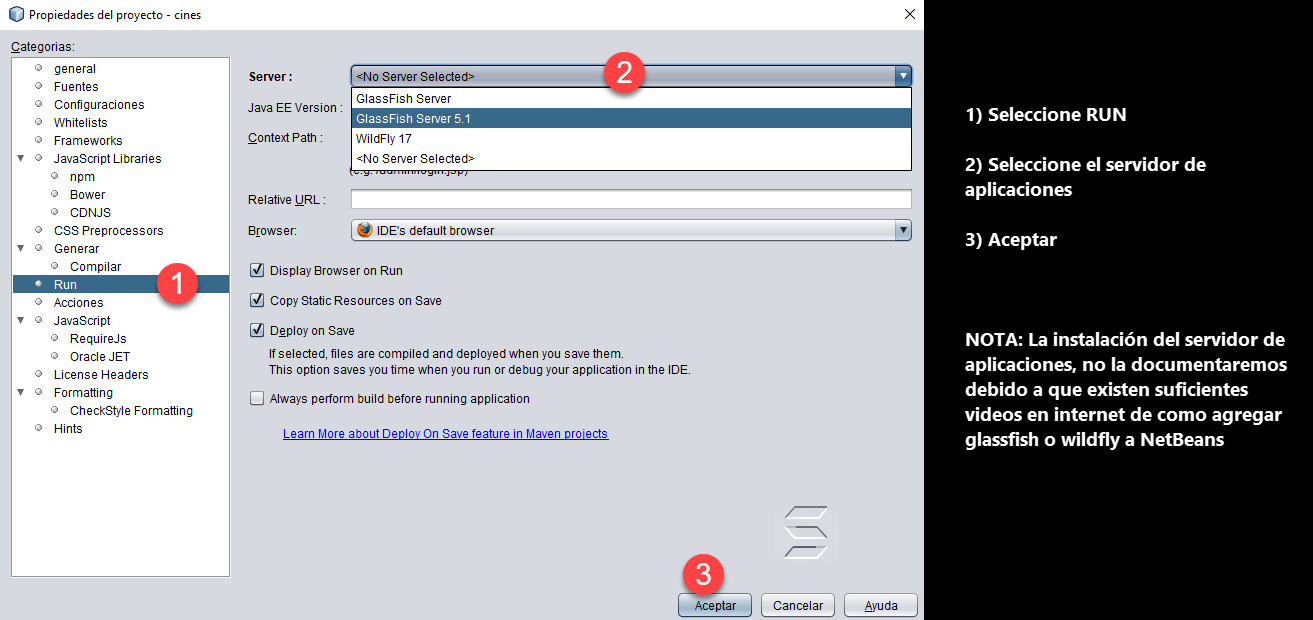


Paso 6, asignando nuestro servidor de aplicaciones al proyecto, esto se hace para evitar inconvenientes con la creación de nuestro archivo de unidad de persistencia.

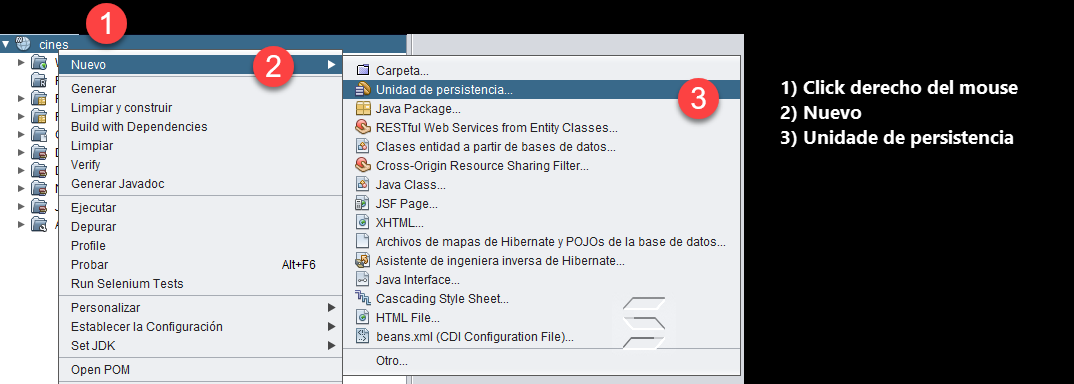


Paso 7, asignación del servidor de aplicaciones.

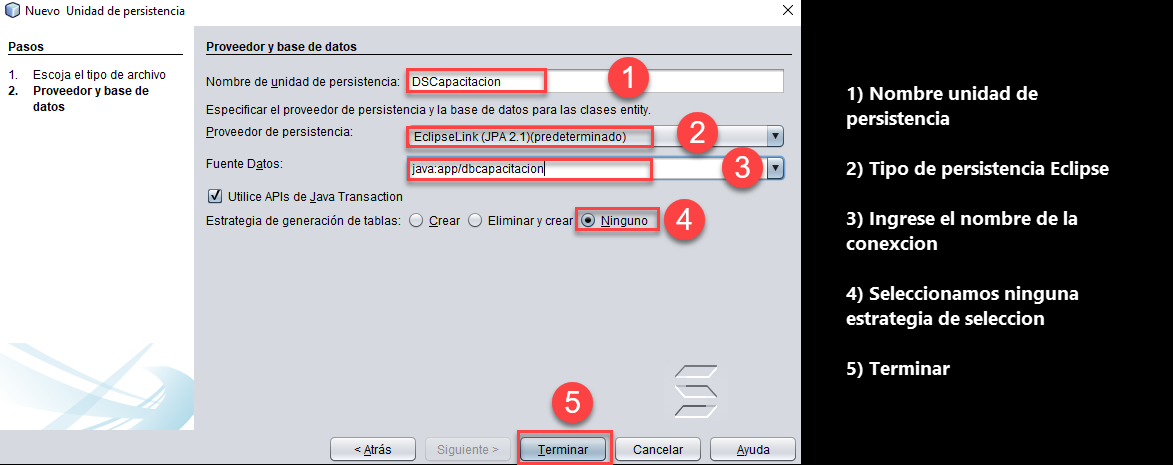
Asignaremos un servidor de aplicación a nuestro middleware de cines



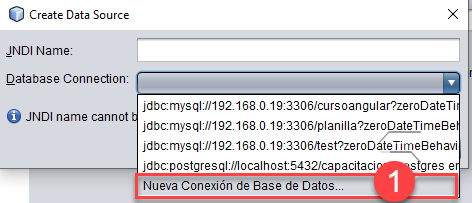
Paso 8, Crearemos la unidad de persistencia de nuestro middleware.



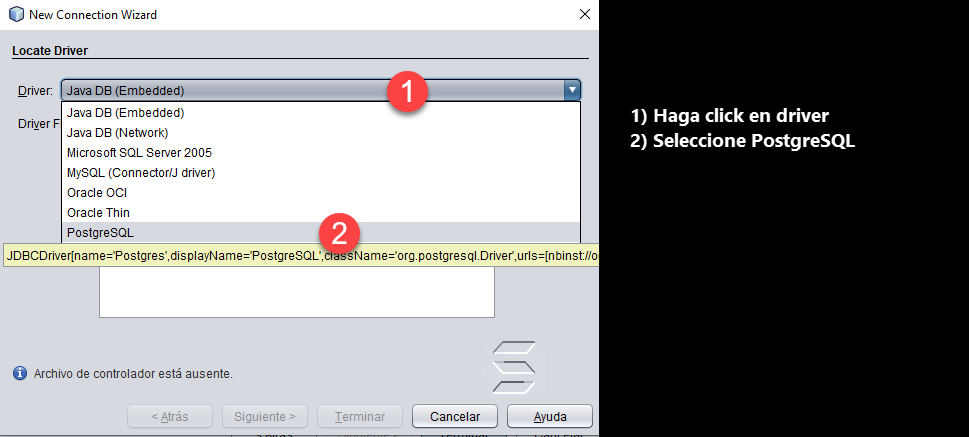
Paso 9, Definición de la unidad de persistencia de nuestro middleware.



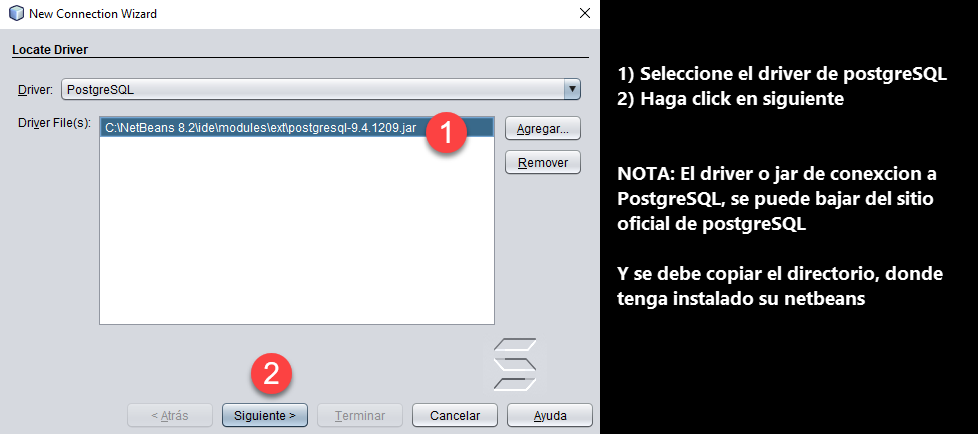
Paso 10, Creando conexiones de base datos para nuestro middleware.



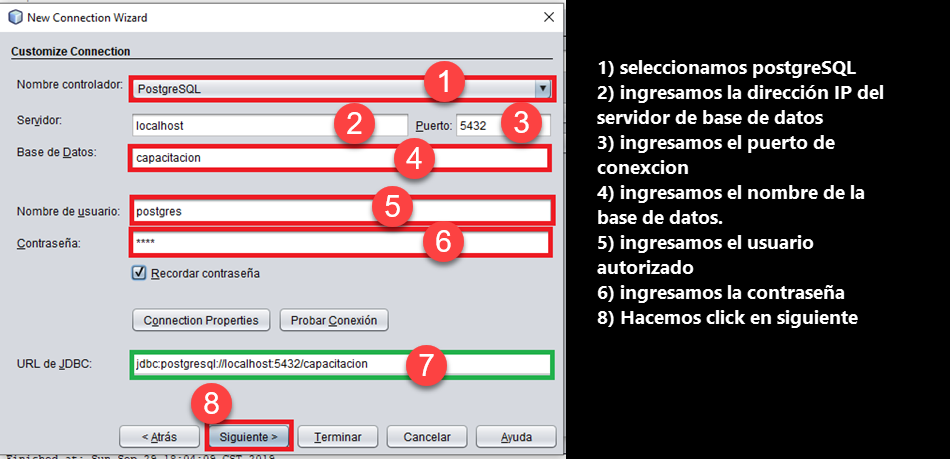
Paso 11, Seleccionado en driver de conexión de la base de datos.



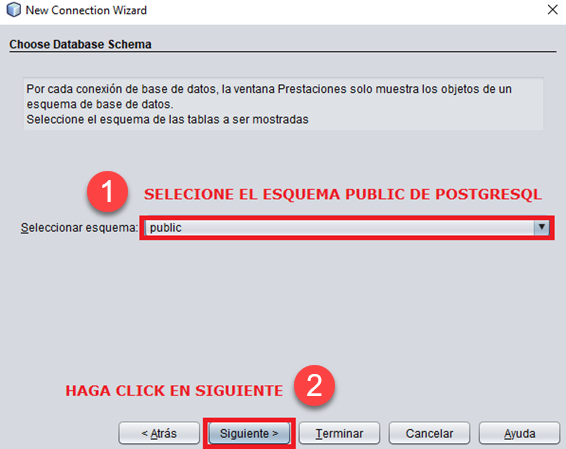
Paso 11, Seleccionado en driver de conexión de la base de datos.



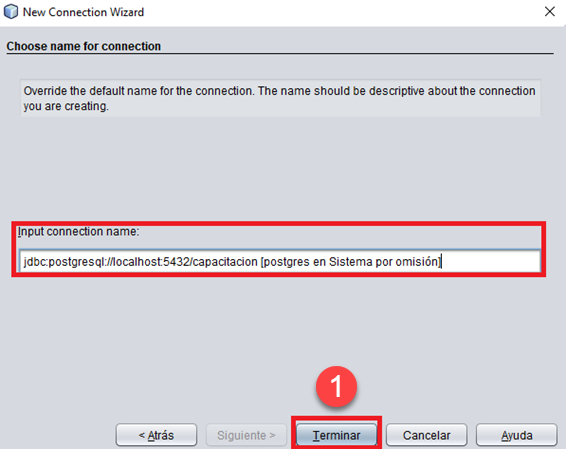
Paso 12, Parametrización de la conexión a base datos con PostgreSQL.



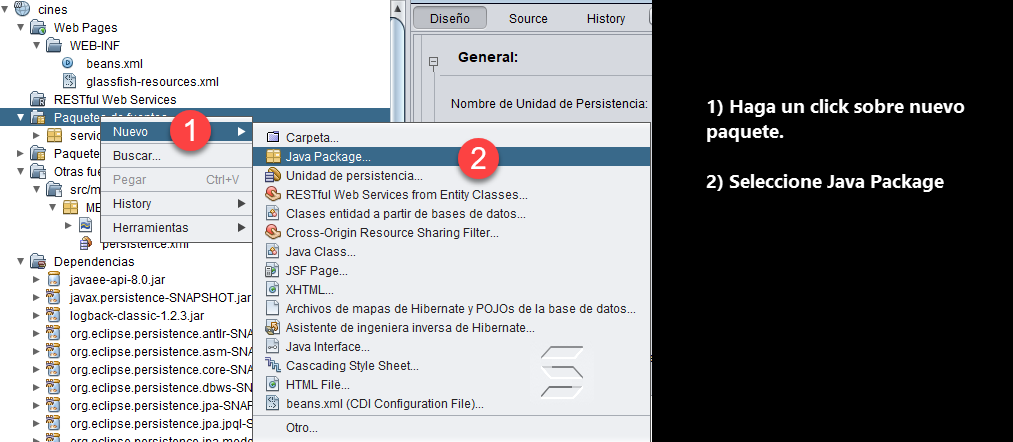
Paso 13, Seleccionando el esquema public

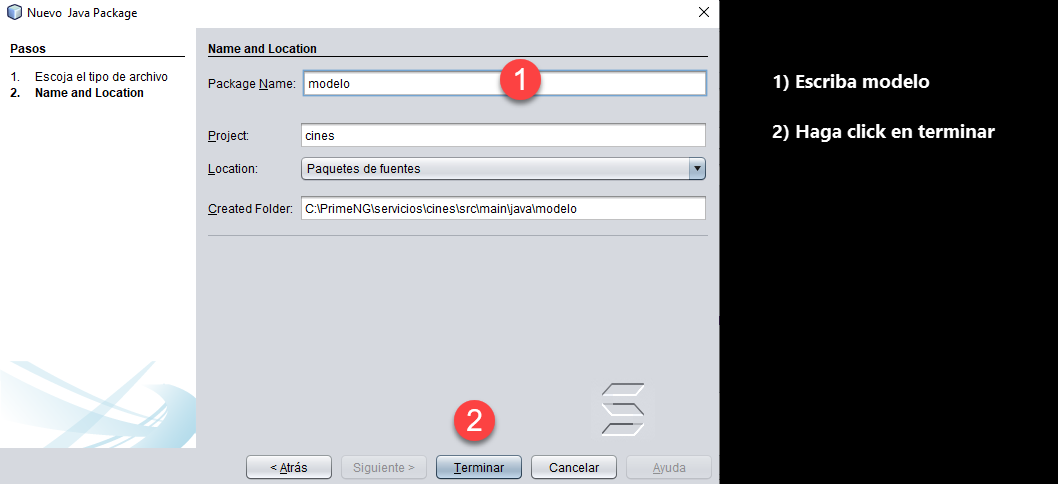


Paso 14, finalizando la parametrización de la conexión a la base de datos

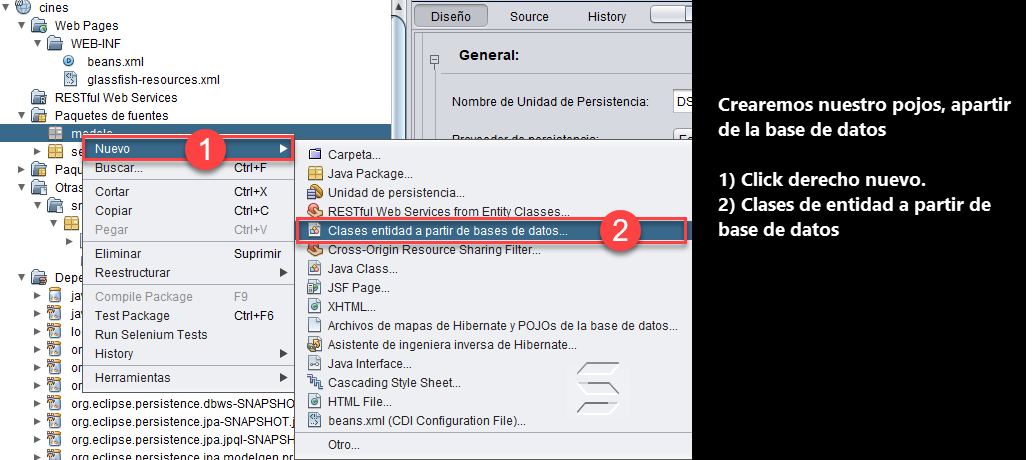


Paso 15, Creando nuestro paquete modelo, el paquete modelo es donde estará nuestros pojos que representan cada una de las tablas o entidades de la base de datos

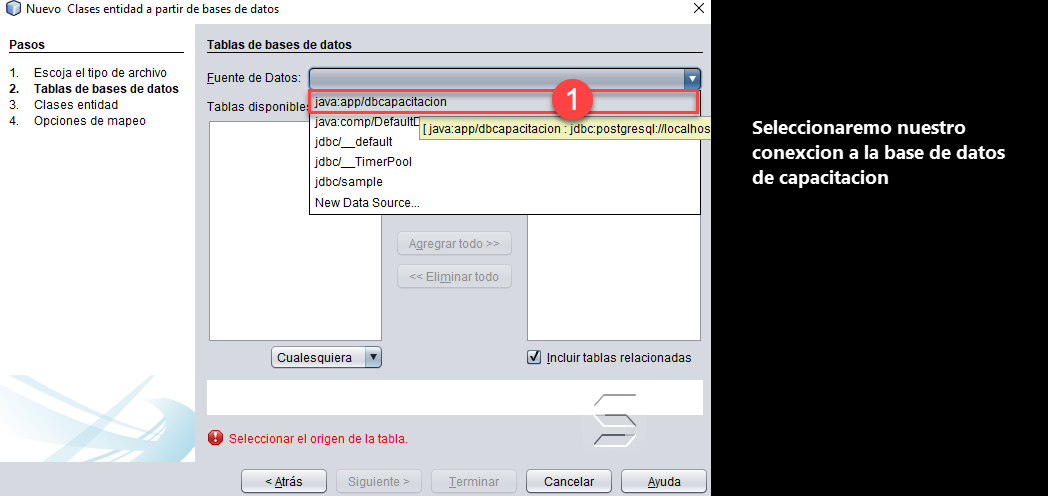




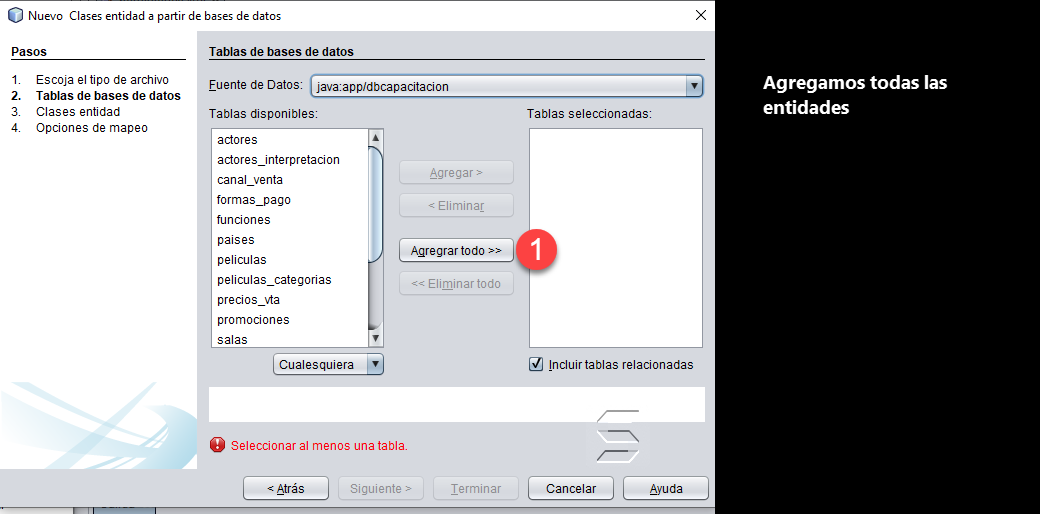
Paso 16, Creamos nuestro pojo a partir de la base de datos dentro del paquete modelo.



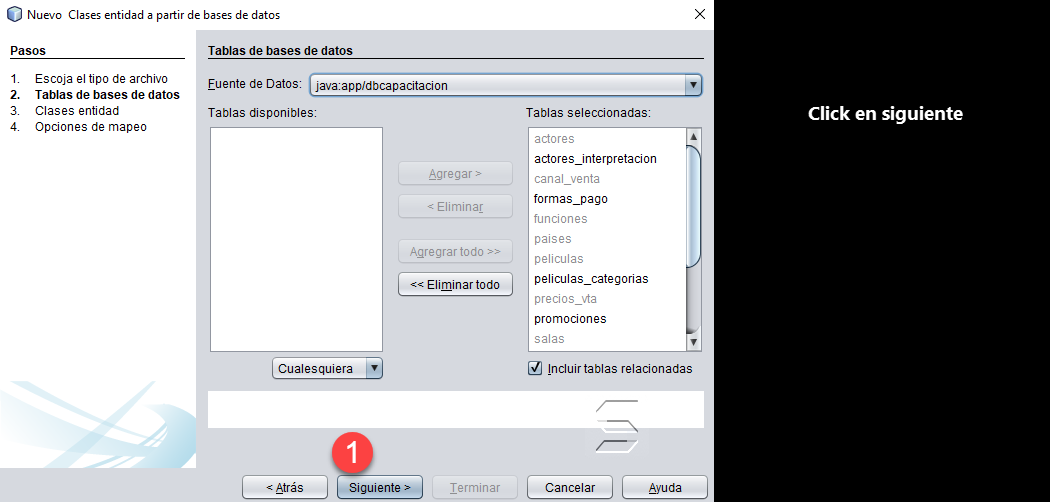
Paso 17, Seleccionamos la conexión de la base de datos.



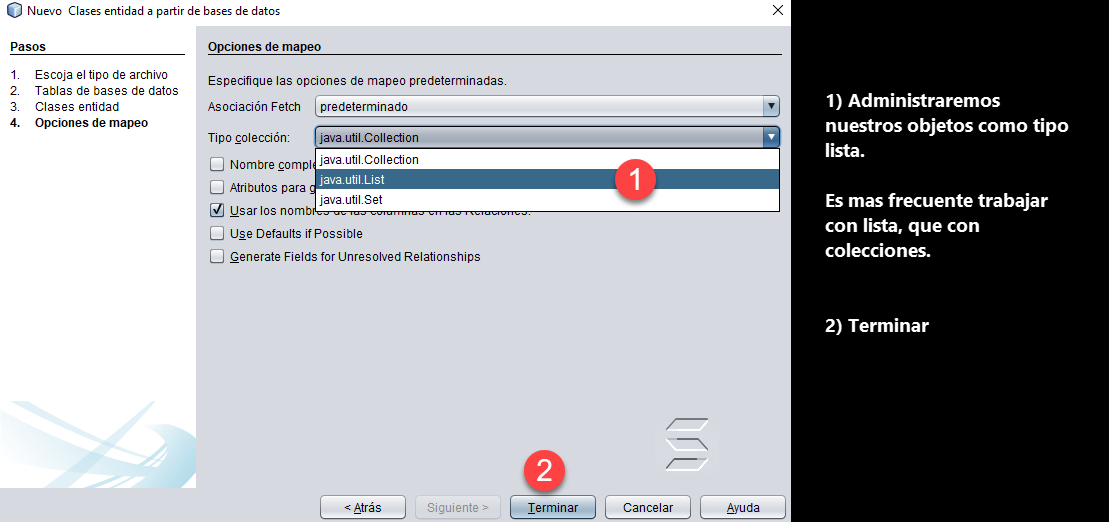
Paso 18, agregamos todas las tablas a la base de datos.



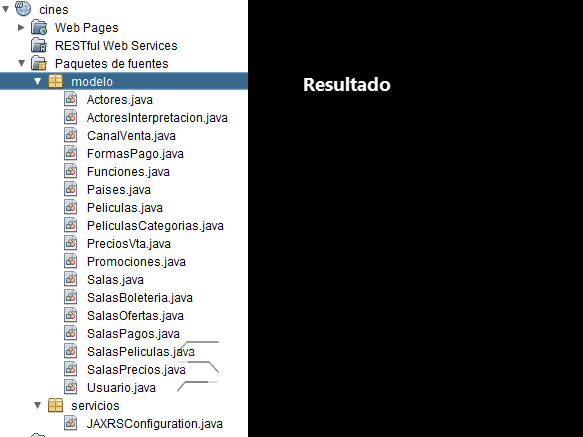
Paso 19, finalizamos el proceso de creación de los pojos que representan a las tablas en la base de datos.



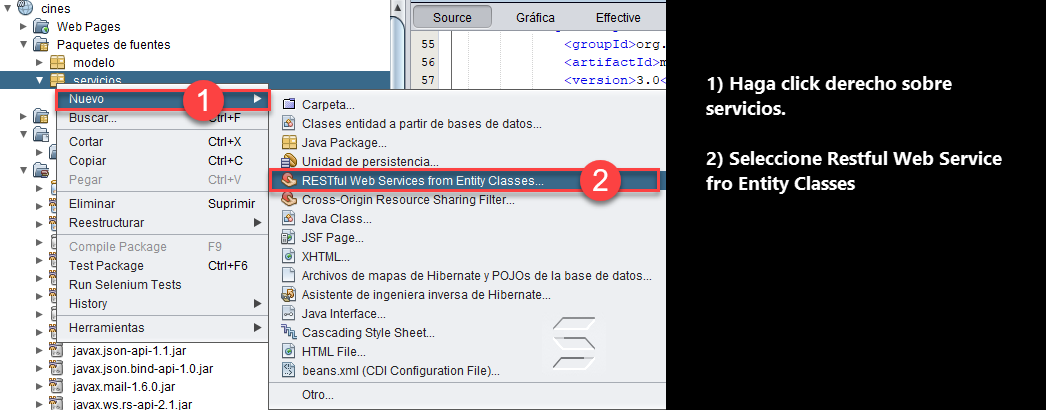
Paso 20, Definiremos nuestros objetos como tipo lista para un mejor manejo en la programación.



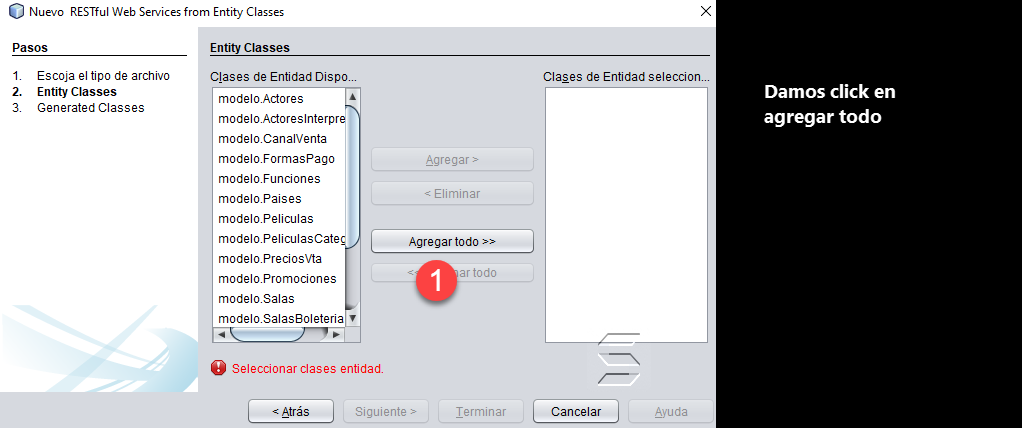
Paso 21, Resultado de la creación

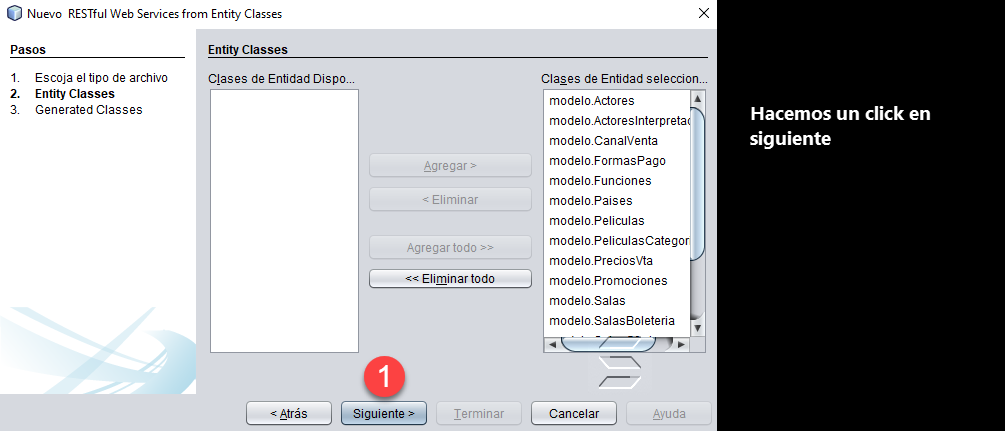


Paso 22, Creando los endpoint de nuestro middleware con Restful

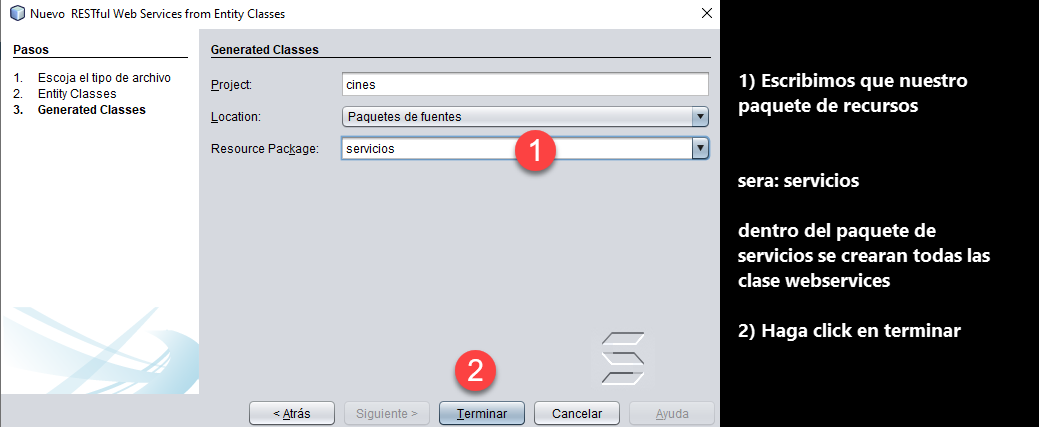


Paso 23, Seleccionaremos todos los pojos que hemos creado en el paquete modelo, para crear nuestros servicios middleware.

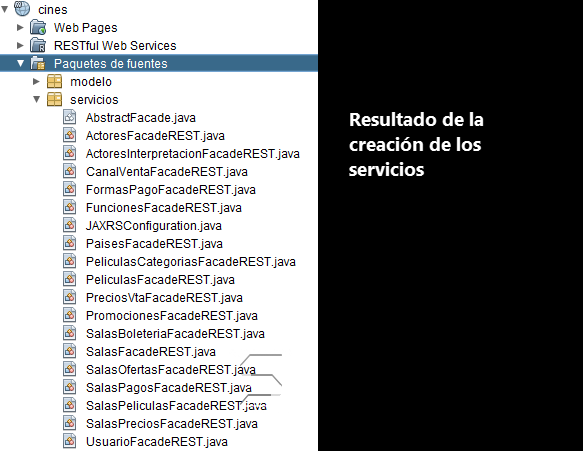




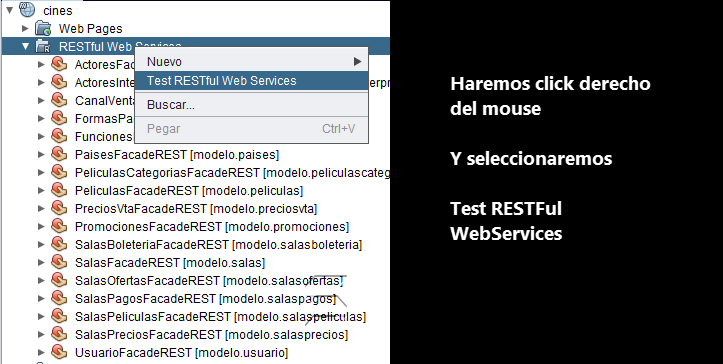
Paso 24, Definimos en que paquete vamos a guardar nuestro pojos de servicio o endpoint de RESTFUL.



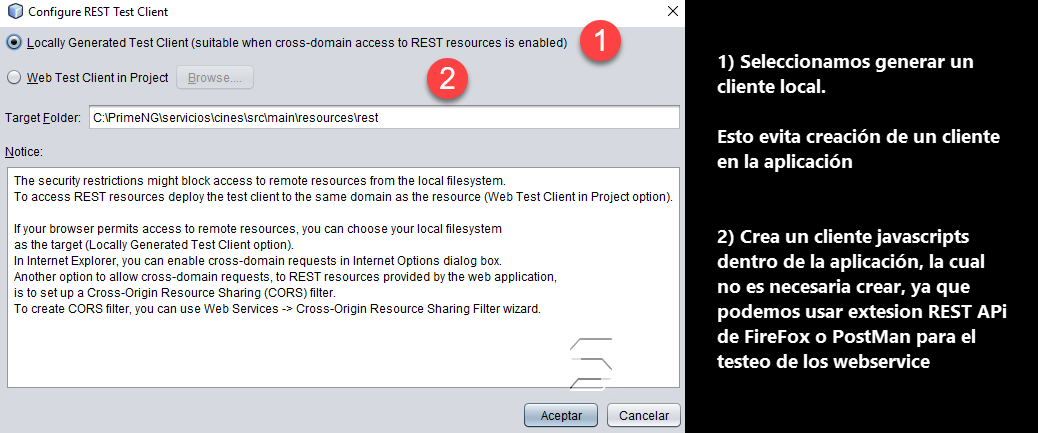
Paso 25, resultado de la creación de los endpoint.

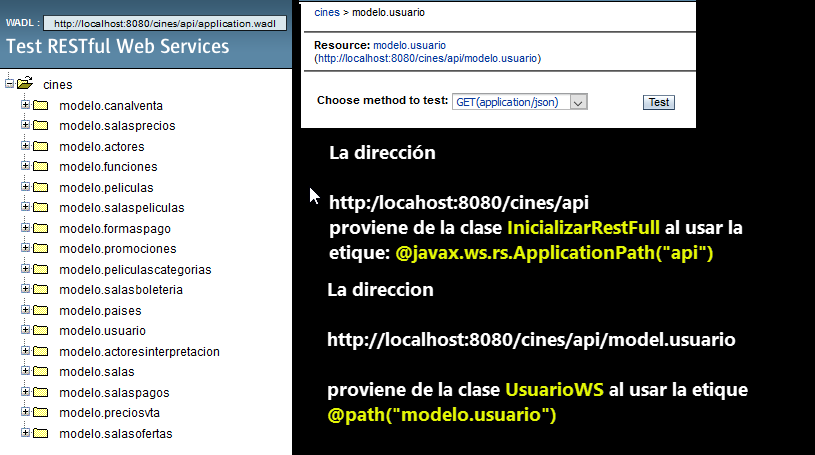


Paso 25, para testear los endpoints basta con hacer click derecho en la carpeta de RESTful Web Services, eso levantara el servidor de aplicaciones.



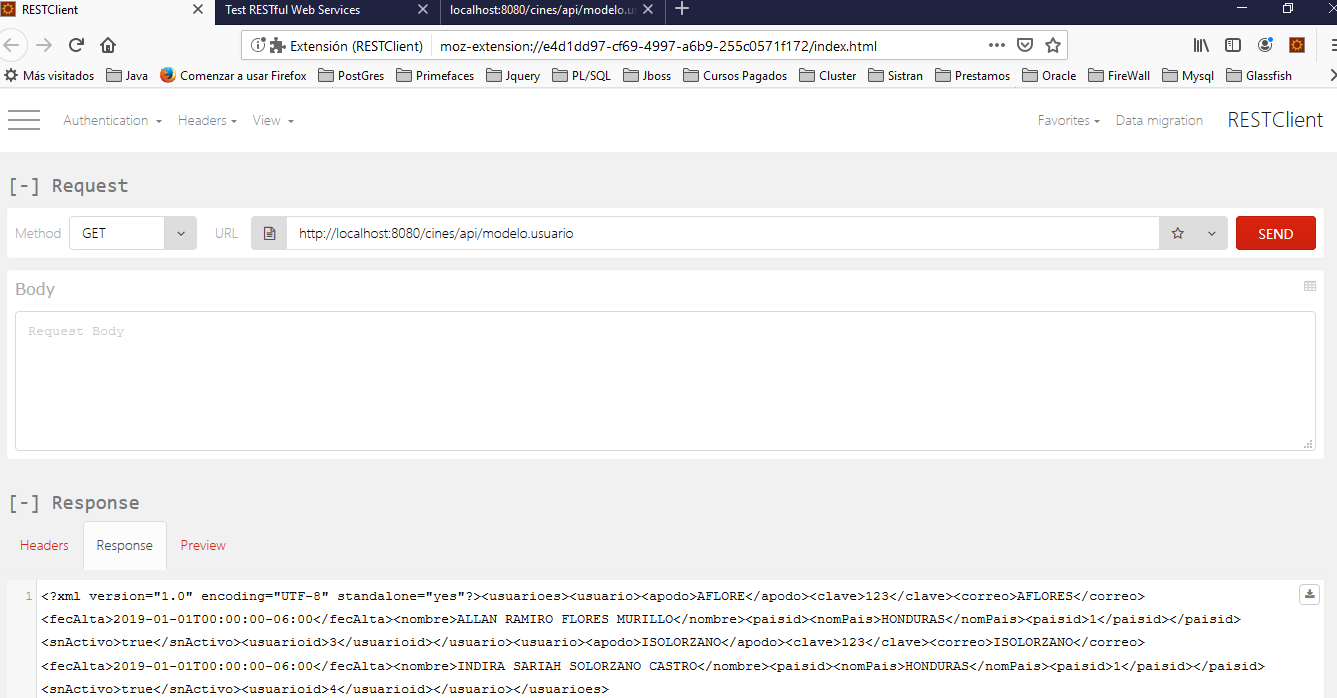
Seleccionamos el tipo de cliente que usaremos.



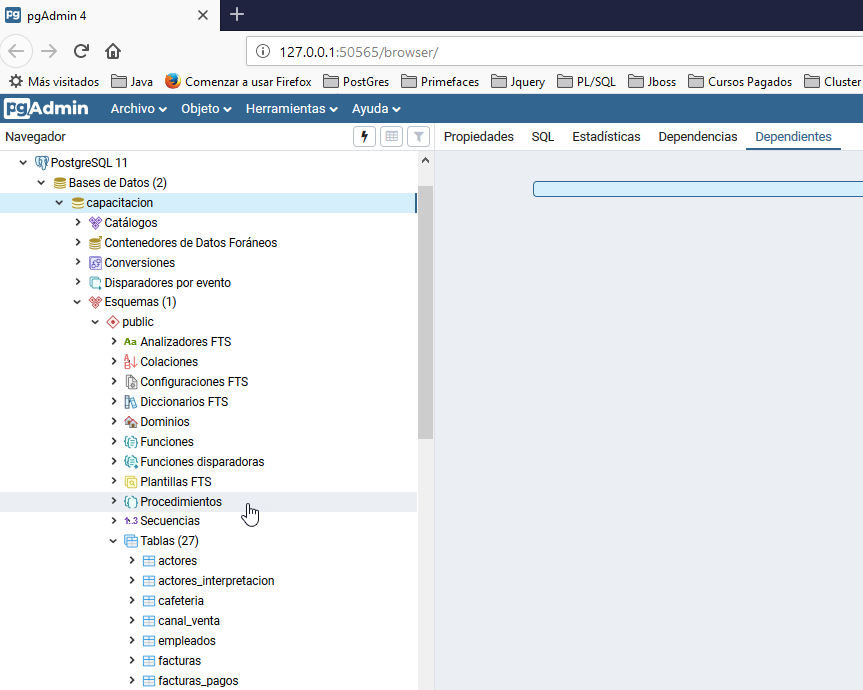


Los clientes Postman y ResClient, solo pueden ser usados mientras en servidores de aplicaciones esta levantado y el middleware este corriendo dentro del servidor de aplicaciones.

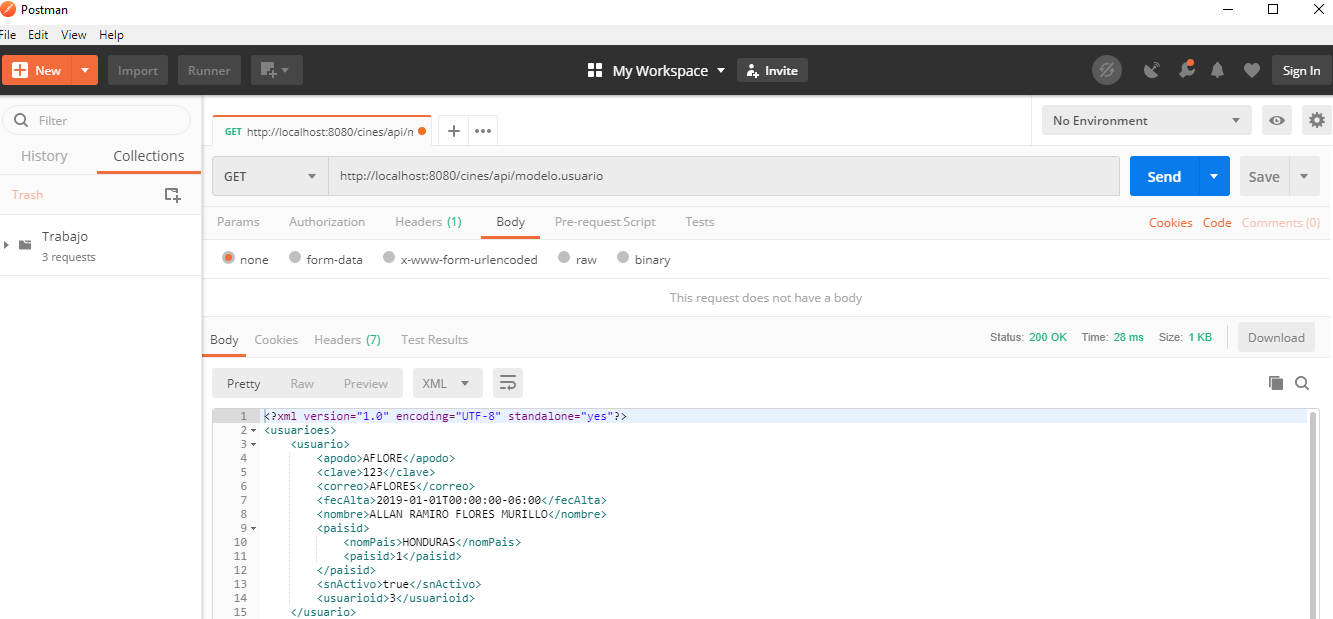
RestClient Mozilla



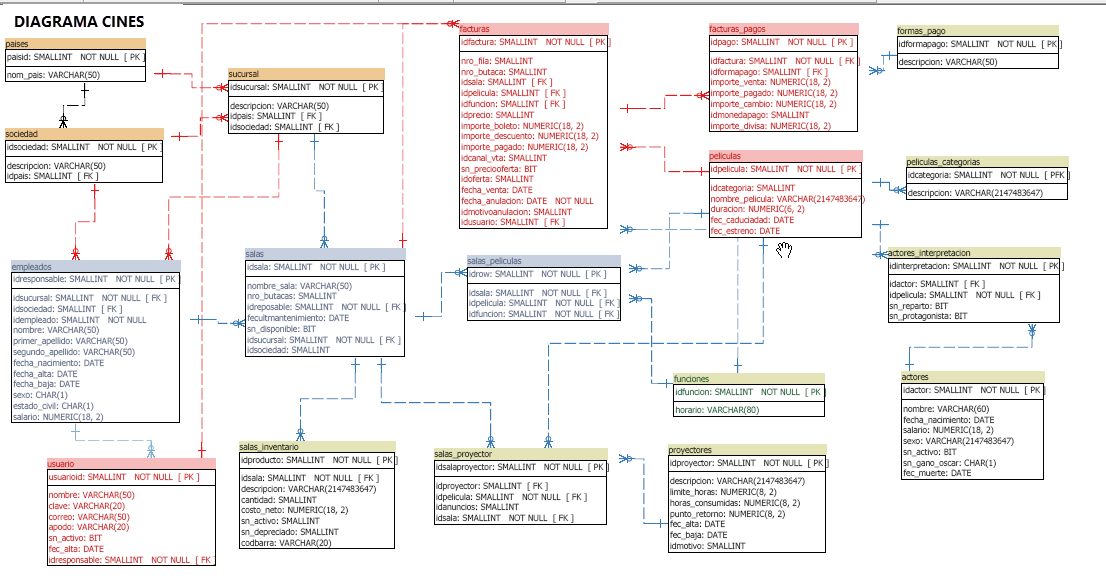
**PGADMIN4 v4 y PostgreSQL 11**



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 Arquitect versión comunitaria.**

**Sql Power Arquitect**

