

Blog sobre Java EE

Prueba Google Ads con 75€
de publicidad gratuita.

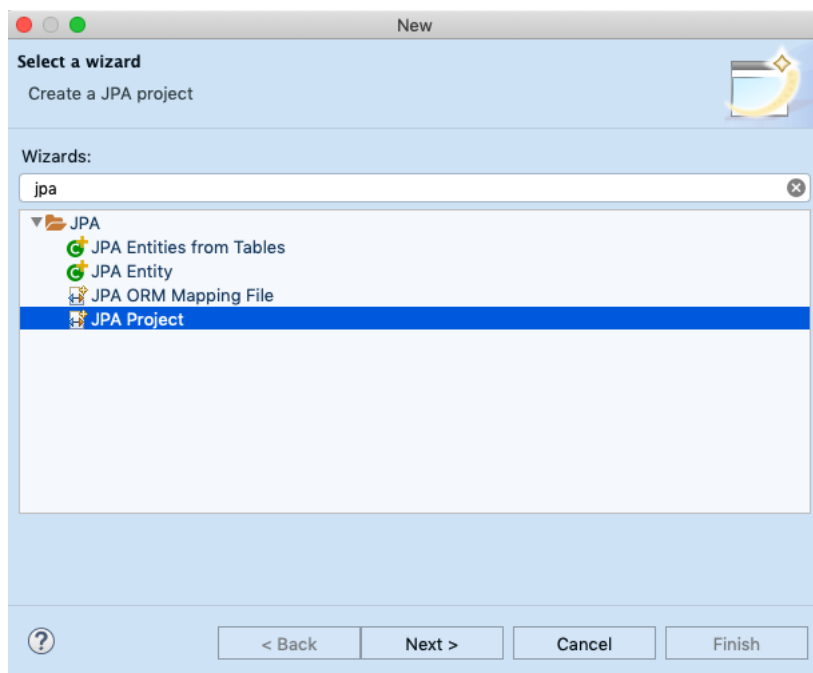
Continuar

Estás aquí: [Inicio](#) / [Java EE](#) / [JPA](#) / Eclipse JPA y clases de dominio

Eclipse JPA y clases de dominio

25 enero, 2019 por [Cecilio Álvarez Caules](#) — [Deja un comentario](#)

El uso de **Eclipse JPA** a la hora de automatizar la creación del modelo de dominio es bastante común . En muchas ocasiones necesitamos generar el modelo de dominio o de entidades de la forma mas rápida posible para Java Persistence API .Vamos a ver cómo podemos usar Eclipse para crear nuestras clases Java de JPA. El primer paso es construir un nuevo proyecto de JPA sobre Eclipse.



Una vez seleccionado el tipo de proyecto le asignamos un nombre y asignamos la última versión de JPA:

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

[ACEPTAR](#)[plugin cookies](#)

Location: [Browse...](#)

Target runtime

[New Runtime...](#)

JPA version

[New...](#)

Configuration

[Modify...](#)

A good starting point for working with Java SE 8 [1.8.0_151] runtime. Additional facets can later be installed to add new functionality to the project.

EAR membership

☐ Add project to an EAR

EAR project name: [New Project ...](#)

Working sets

☐ Add project to working sets [New...](#)

Working sets: [Select...](#)

[?< Back](#) [Next >](#) [Cancel](#) [Finish](#)

Pulsamos en continuar en Eclipse JPA y nos permitirá elegir su implementación , elegimos la última versión de Eclipse Link pulsando en + :

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

[ACEPTAR](#)[plugin cookies](#)

Type: **User Library**

☒ EclipseLink 2.5.2

☐ Include libraries with this application

Connection

<None>

[Add connection...](#)

[Connect](#)

☐ Add driver library to build path

Driver:

☐ Override default catalog from connection

Catalog:

☐ Override default schema from connection

Schema:

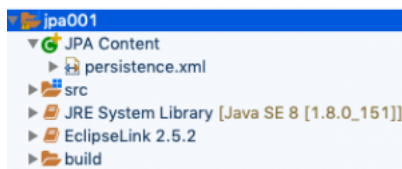
Persistent class management

☐ Discover annotated classes automatically

☒ Annotated classes must be listed in persistence.xml

[?](#) [< Back](#) [Next >](#) [Cancel](#) [Finish](#)

Ya disponemos de un proyecto de JPA vacío en donde podemos ver el persistence.xml:

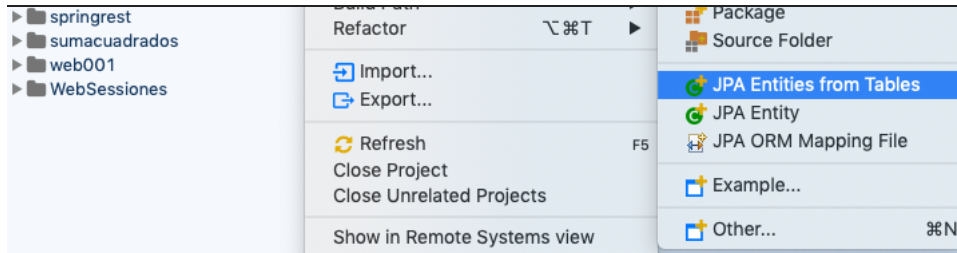


El siguiente paso es solicitar crear las entidades desde las tablas de la base de datos para ello nos apoyamos en el asistente del menu.

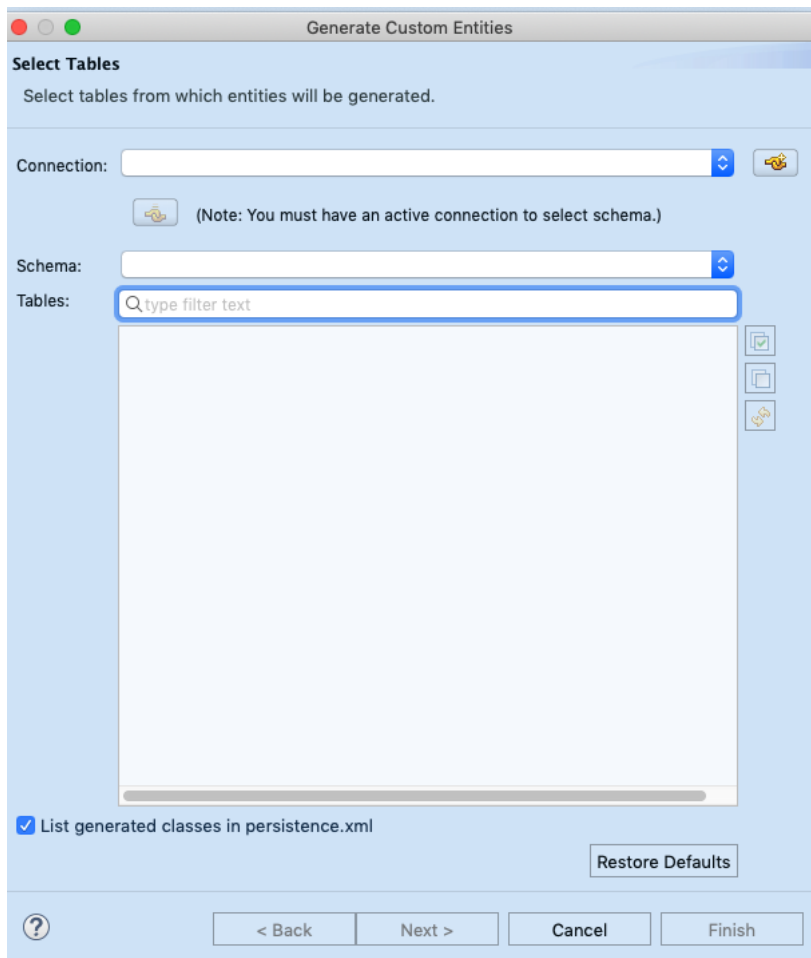
Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

ACEPTAR

plugin cookies



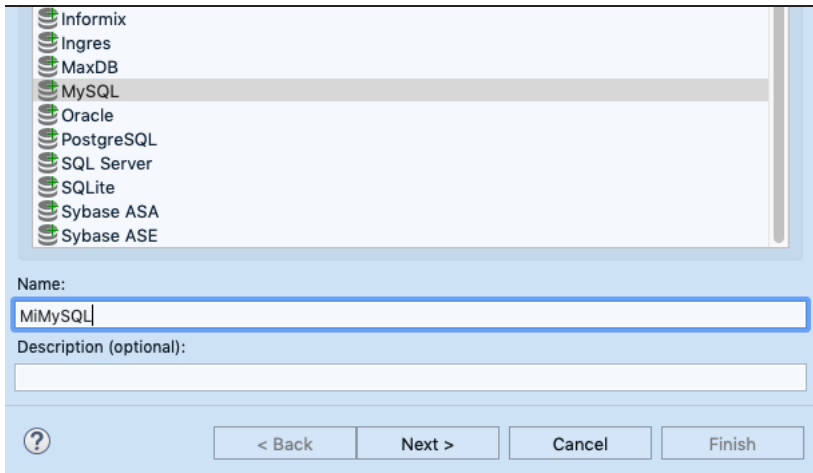
Esto nos presentara una nueva ventana en la que configurar una nueva conexión a la base de datos.



Solicitamos crear una nueva conexión y para ello elegimso como base de datos MySQL:

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

ACEPTAR

[plugin cookies](#)

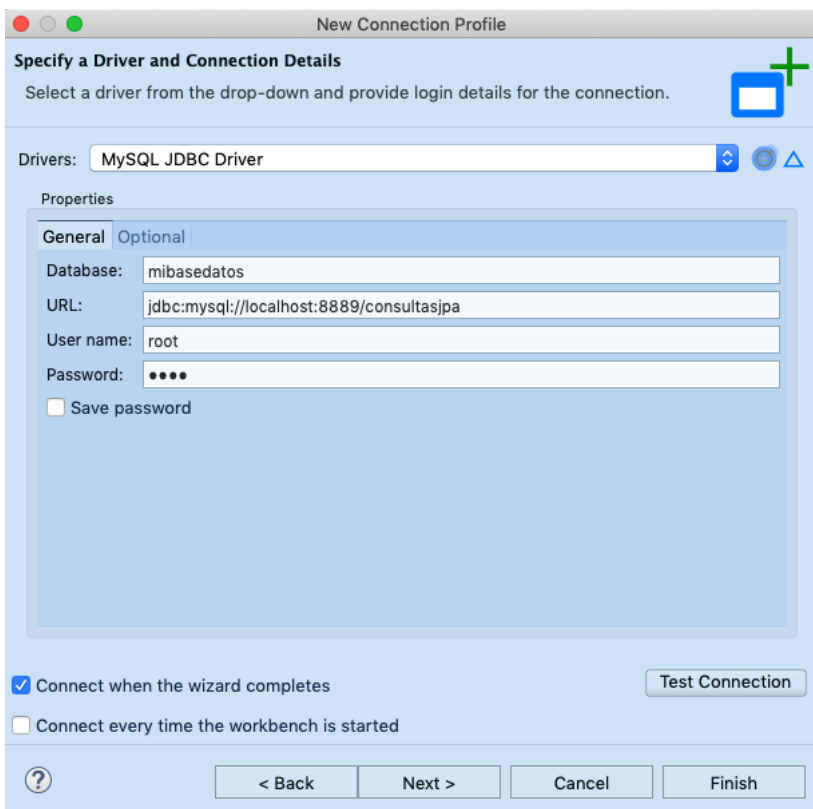
Informix
Ingres
MaxDB
MySQL
Oracle
PostgreSQL
SQL Server
SQLite
Sybase ASA
Sybase ASE

Name:
MiMySQL

Description (optional):

< Back Next > Cancel Finish

Configuramos para MySQL todos los parámetros usuario ,password y url:



New Connection Profile

Specify a Driver and Connection Details

Select a driver from the drop-down and provide login details for the connection.

Drivers: MySQL JDBC Driver

Properties

General Optional

Database: mibasedatos

URL: jdbc:mysql://localhost:8889/consultasjpa

User name: root

Password: ****

☐ Save password

☒ Connect when the wizard completes

☐ Connect every time the workbench is started

Test Connection

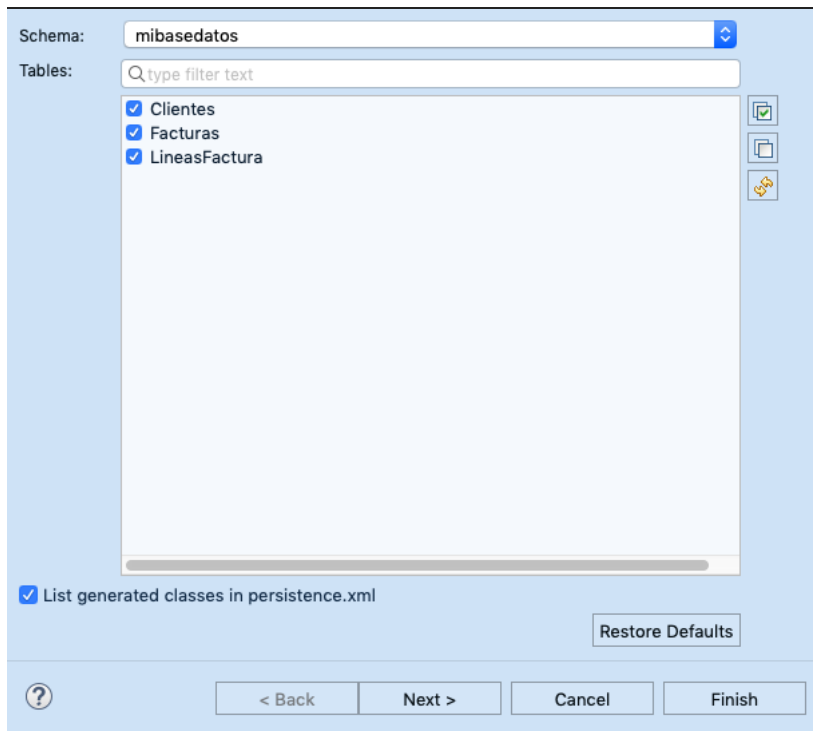
< Back Next > Cancel Finish

Pulsamos en continuar y nos permitirá elegir las tablas de la base de datos para generar las clases de dominio de forma automática en este caso tenemos Clientes, Facturas y LineasFactura:

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

[plugin cookies](#)

ACEPTAR



Realizada esta operación pulsamos en finalizar y en el Eclipse nos habrá generado las clases que nosotros necesitamos así de sencillo:



Cómo se puede observar Eclipse JPA ha sido capaz incluso de generar las claves compuestas que tenía el modelo Entidad Relación en el que nos hemos basado. Eso sí recordemos que siempre un asistente genera un modelo muy general y tendremos la posibilidad y sobre todo la obligación de afinarlo nosotros modificando nuestras clases, las relaciones y temas como **eager** y **lazy feching**.

Otros artículos relacionados

1. [Utilizando un JPA Stream con JPA 2.2](#)
2. [Java Generic Repository y JPA](#)
3. [JPA DTO \(Data Transfer Object\) y JPQL](#)
4. [Ejemplo de JPA , Introducción \(I\)](#)
5. [Java Persistence API](#)



PDF

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

ACEPTAR

[plugin cookies](#)

con *

Comentario

Nombre *

Correo electrónico *

Web

PUBLICAR COMENTARIO

Este sitio usa Akismet para reducir el spam. [Aprende cómo se procesan los datos de tus comentarios.](#)

BUSCAR

Buscar en esta web

Mis Cursos de Java Gratuitos

Java Herencia

Java JDBC



Servlets

Intro JPA



Mis Cursos de Java

Programación Orientada a Objeto en Java

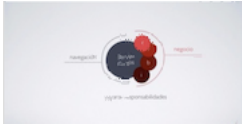
Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra **política de cookies**, pinche el enlace para mayor información.

[plugin cookies](#)

ACEPTAR



Java Web



Pack Java Core



Arquitectura Java Solida con Spring



POPULAR

- [Spring Boot JPA y su configuración](#)
- [Los Frameworks y su lado oscuro](#)
- [Spring REST CORS y su configuración](#)
- [Static Method vs instance method y su uso correcto](#)
- [Java new String y la creación de objetos](#)
- [Arquitecturas RESTFul y agregados](#)
- [Spring Boot JSP y su configuración](#)
- [Single Page Application y REST](#)
- [@RepositoryRestResource y Spring Framework](#)
- [Spring JdbcTemplate y el principio DRY](#)

CONTACTO

contacto@arquitecturajava.com

LO MAS LEIDO

- [¿Qué es Spring Boot?](#)
- [Java Optional Repository y JPA](#)
- [Java Constructores this\(\) y super\(\)](#)
- [Usando Java Session en aplicaciones web](#)
- [Ejemplo de Java Singleton \(Patrones y ClassLoaders\)](#)

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](#), pinche el enlace para mayor información.

ACEPTAR plugin cookies

- Java Override y encapsulación
- REST JSON y Java
- Ejemplo de JPA , Introducción (I)
- Uso de Java Generics (I)
- ¿Cuales son las certificaciones Java?
- ¿Qué es Gradle?
- ¿Qué es un Microservicio?
- El modelo Entidad Relacion con DBDesigner
- El patrón de inyección de dependencia y su utilidad
- Mis Libros
- Eclipse refactor move y como utilizarlo
- Angular ngFor la directiva y sus opciones
- ¿ Que es REST ?
- Spring MVC Configuración (I)
- Java Stream forEach y colecciones