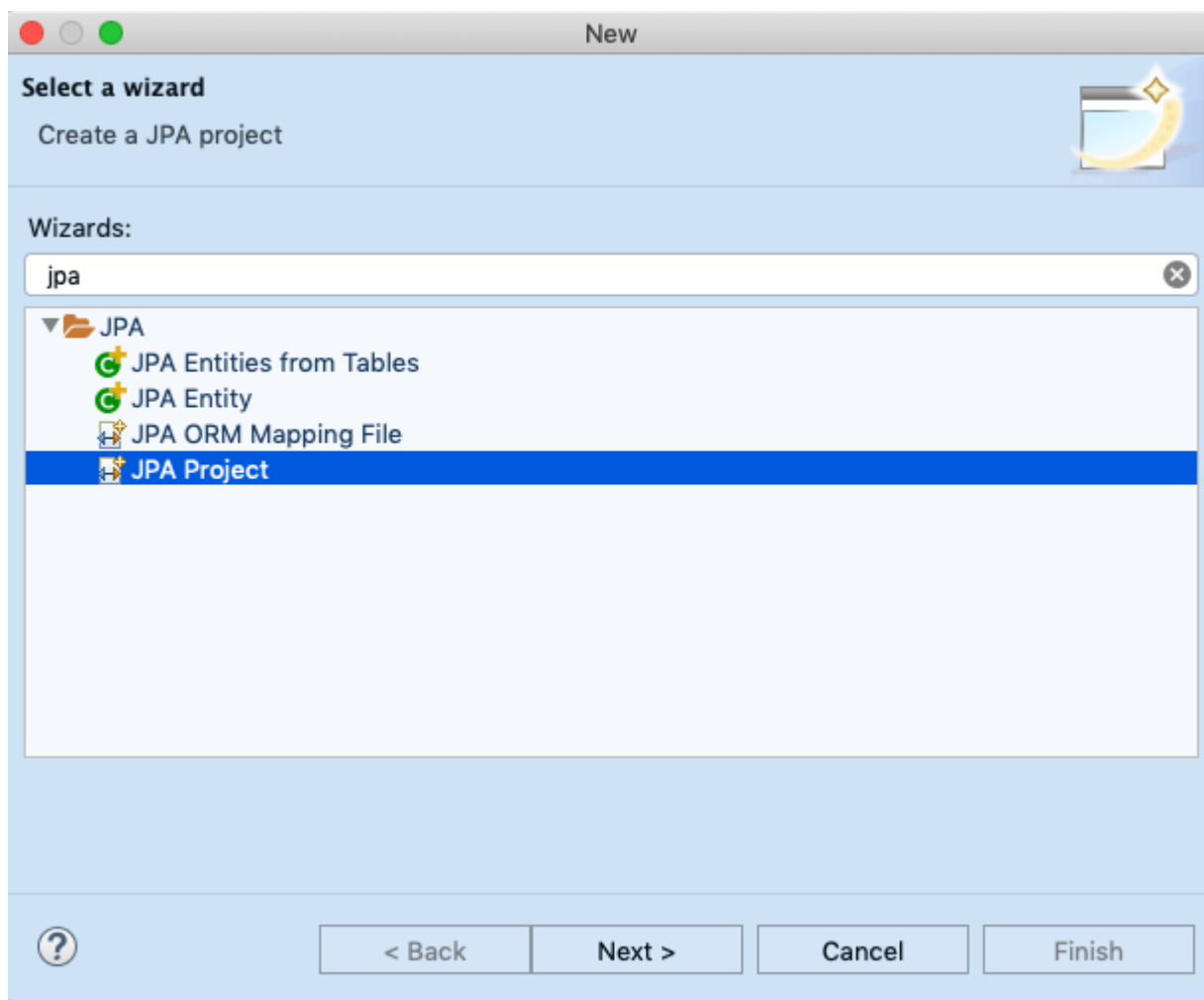


El uso de Eclipse JPA a la hora de automatizar la creación del modelo de dominio es bastante común . En muchas ocasiones necesitamos generar el modelo de dominio o de entidades de la forma mas rápida posible para Java Persistence API .Vamos a ver cómo podemos usar Eclipse para crear nuestras clases Java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

New JPA Project

JPA Project

Configure JPA project settings.



Project name:

jpa001

Project location

☒ Use default location

Location: /Users/cecilioalvarezcaules/blog/jpa001

Browse...

Target runtime

Java SE 8 [1.8.0_151]

New Runtime...

JPA version

2.1

Configuration

Default Configuration for Java SE 8 [1.8.0_151]

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 + :

New JPA Project

JPA Facet

Configure JPA settings.

Platform

Generic 2.1

JPA implementation

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:

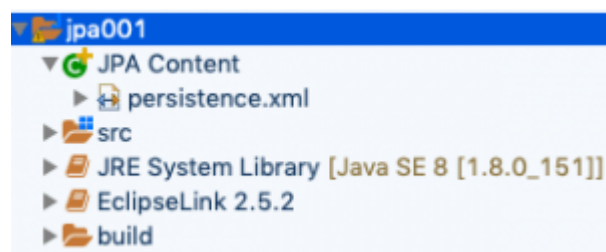
Persistent class management

☐ Discover annotated classes automatically

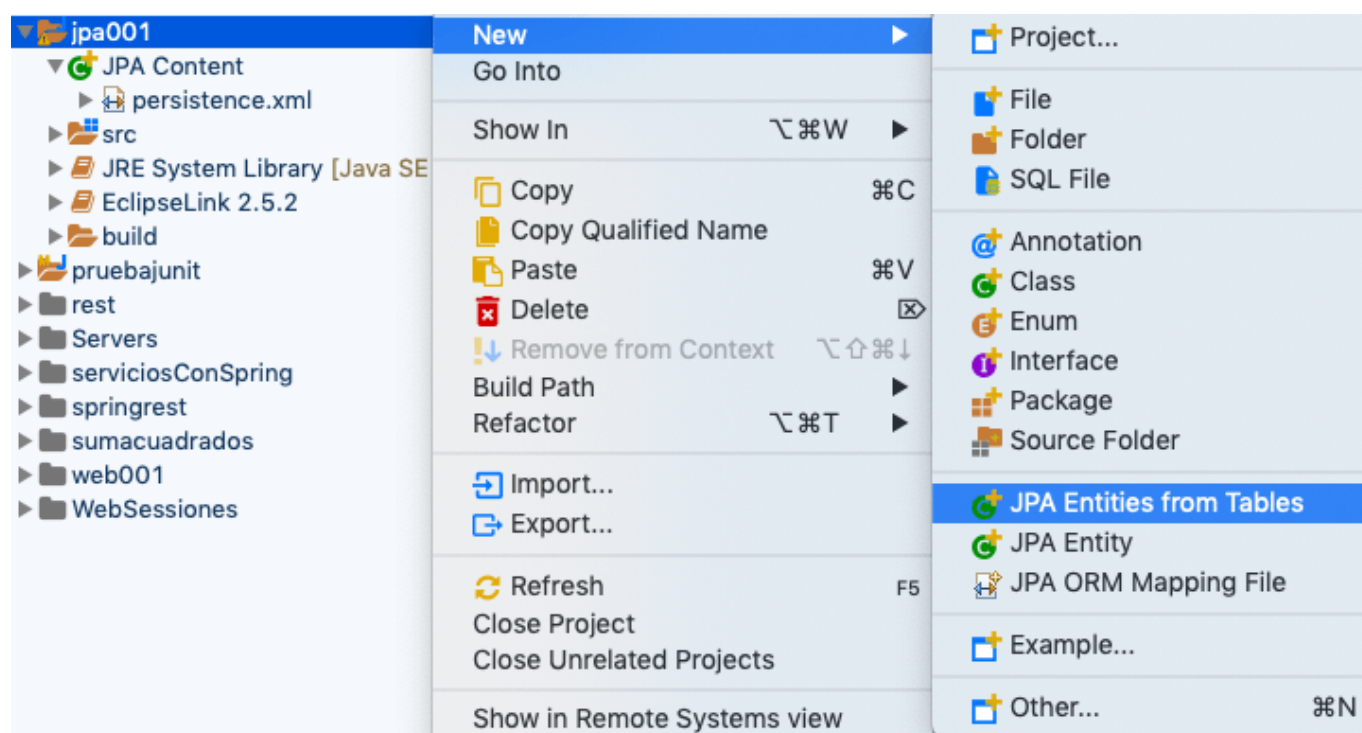
☒ Annotated classes must be listed in persistence.xml

? < Back Next > Cancel Finish

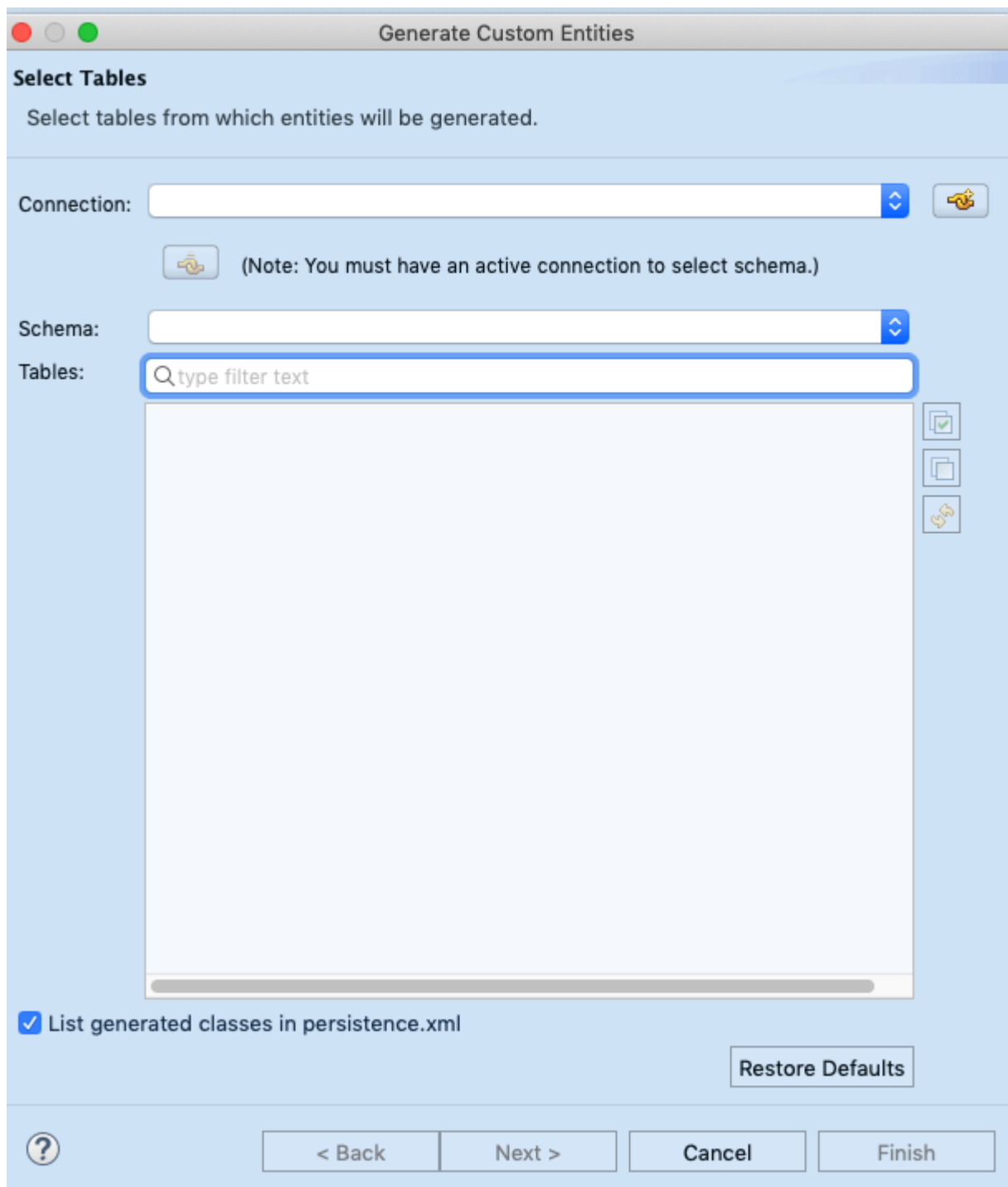
Ya disponemos de un proyecto de JPA vacío en donde podemos ver el persistence.xml:



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.



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:

Connection Profile
Create a MySQL connection profile.

Connection Profile Types:

type filter text

- DB2 for z/OS
- Derby
- Generic JDBC
- HSQLDB
- 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

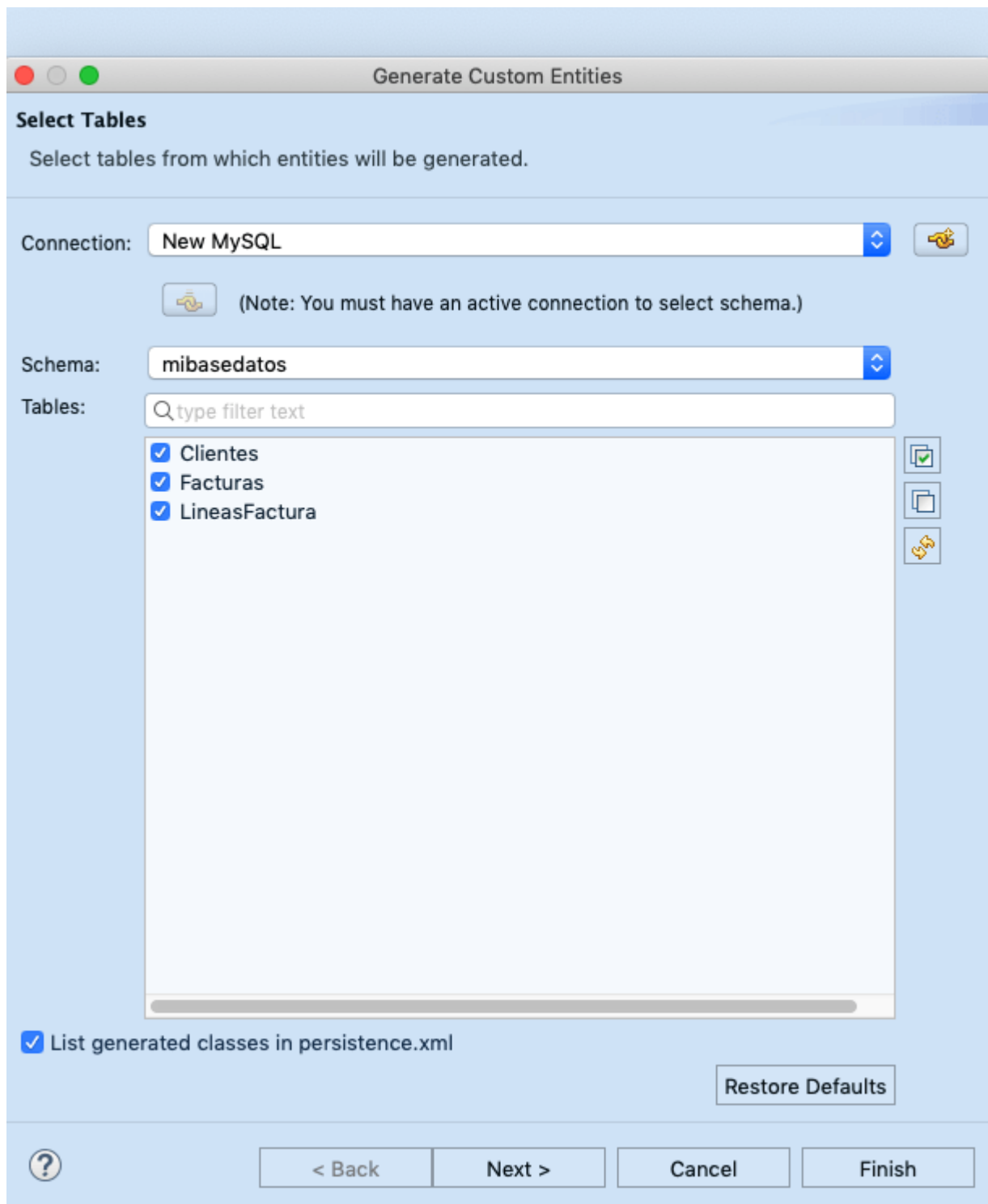
☒ Connect when the wizard completes

☐ 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:



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](#)