# Instructivo Detallado: Conexión de IntelliJ IDEA con MySQL Workbench a través de Hibernate

## Pre-requisitos:

- **XAMPP instalado y en funcionamiento:** Asegúrate de que el servidor MySQL de XAMPP esté iniciado.
- **Base de datos "movies_db" creada:** Esta base de datos debe contener las tablas que deseas mapear con tus entidades Java.
- **IntelliJ IDEA instalado:** Con el plugin de Maven configurado.
- **Dependencias descargadas:** hibernate-core, jakarta.persistence-api, mysql-connector-java (disponibles en tu Google Drive).

## Paso a Paso:

### 1. Crear un Nuevo Proyecto Maven en IntelliJ IDEA:

- **File -> New -> Project -> Maven**
- Selecciona el arquetipo deseado (por ejemplo, Maven project).
- Define el groupId, artifactId y la versión de tu proyecto.

### 2. Configurar el POM.xml:

- **Agregar las dependencias:**

  XML

```
<dependencies>
    <dependency>

                hibernate-core
    <dependency>

                jakarta.persistence-api
    </dependency>
    <dependency>

                mysql-connector-java
    </dependency>
</dependencies>
```

- **Crear la carpeta META-INF en src/main/resources:**
- **Copiar el archivo persistence.xml:** Pega el contenido del archivo de configuración de Hibernate que tienes en tu Google Drive dentro de META-INF/persistence.xml. Asegúrate de ajustar las propiedades de conexión (URL, usuario, contraseña) para que coincidan con tu base de datos.

**3. Crear los Paquetes:**

- Dentro del paquete principal de tu proyecto (por ejemplo, com.example), crea los siguientes paquetes:
    - controllers
    - entities
    - persistence

**4. Crear las Clases:**

- **En el paquete persistence:**
    - **ConfigJPA:** Esta clase se encargará de configurar Hibernate.
    - **MovieJPA:** Esta clase será un DAO (Data Access Object) para realizar operaciones CRUD sobre la entidad Movie.
- **En el paquete entities:**
    - **Movie:** Esta entidad mapeará la tabla "movies" de tu base de datos. Anótalo con las anotaciones de Hibernate para definir la correspondencia entre las propiedades de la clase y las columnas de la tabla.

**5. Crear el Controller:**

- **En el paquete controllers:**
    - **MovieController:** Esta clase contendrá los métodos para realizar las operaciones CRUD sobre la entidad Movie.

**6. Configurar el Main.java:**

- En el método main, crea una instancia MovieController para realizar operaciones CRUD.

## Ejemplo de código (ConfigJPA):

Java

```
import jakarta.persistence.EntityManagerFactory;
import jakarta.persistence.Persistence;
```

```java
public class ConfigJPA {
    private static final EntityManagerFactory emf =
Persistence.createEntityManagerFactory("examplePU");

    public static EntityManager getEntityManager() {
        return emf.createEntityManager();
    }

    public static void close() {
        emf.close();
    }
}
```

**Importante:**

- **Ajusta las propiedades de conexión en persistence.xml:** Asegúrate de que la URL, el usuario y la contraseña sean correctos para tu base de datos.
- **Mapeo de entidades:** Utiliza las anotaciones de Hibernate (por ejemplo, `@Entity`, `@Table`, `@Id`, `@Column`) para mapear correctamente las entidades con las tablas de la base de datos.
- **Manejo de transacciones:** Utiliza las transacciones de Hibernate para garantizar la integridad de los datos.
- **Error handling:** Implementa un manejo de errores adecuado para capturar excepciones y proporcionar mensajes informativos.

**Este es un esquema general.** El código completo dependerá de la complejidad de tu aplicación y de las características específicas de tu base de datos.