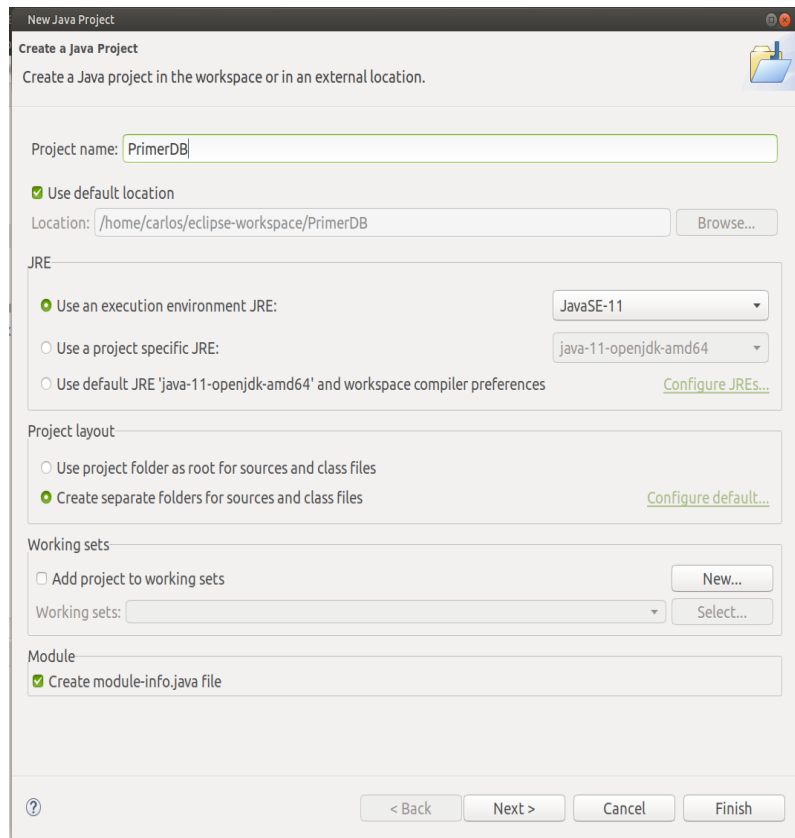


Cómo crear un proyecto eclipse y luego agregar Maven

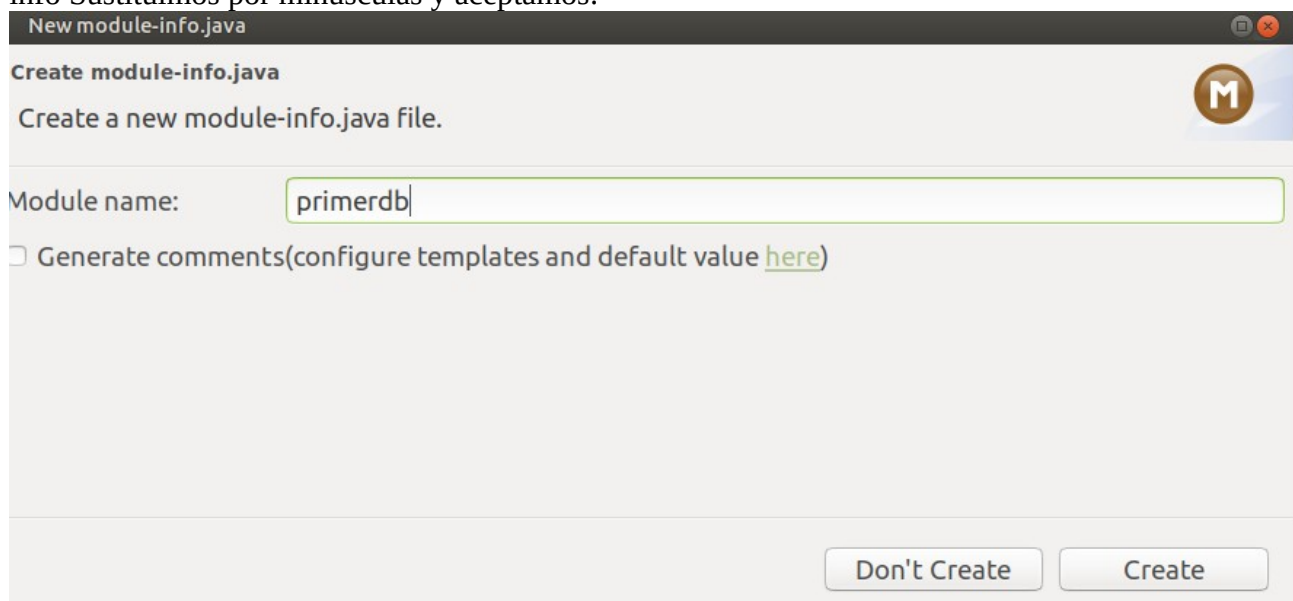
Primero generamos un proyecto java normal:

observar que nos está poniendo Java 11 por defecto. Si lo dejamos así cuando queramos trabajar con bases de datos, con javafx, etc vamos a tener que importar módulos en: module-info. Lo vamos a hacer para entender también esa parte

Fijarse que tenemos seleccionado por defecto que nos cree precisamente ese fichero



Pulsamos en Finish. Al hacerlo nos informa que recomienda usar cambiar a minúsculas el module-info. Sustituimos por minúsculas y aceptamos:



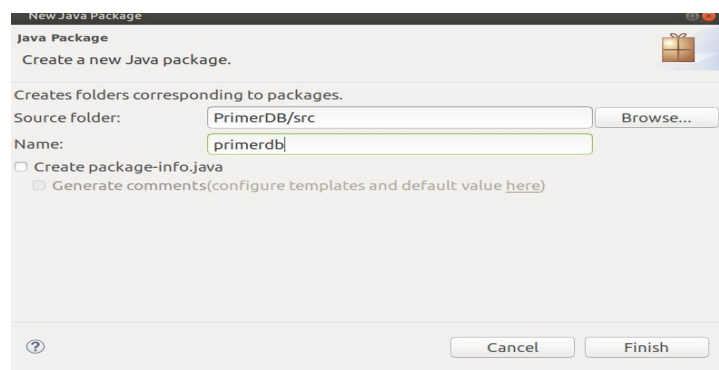
Si ahora miramos el contenido de ese fichero vemos que lo ha puesto en minúsculas:



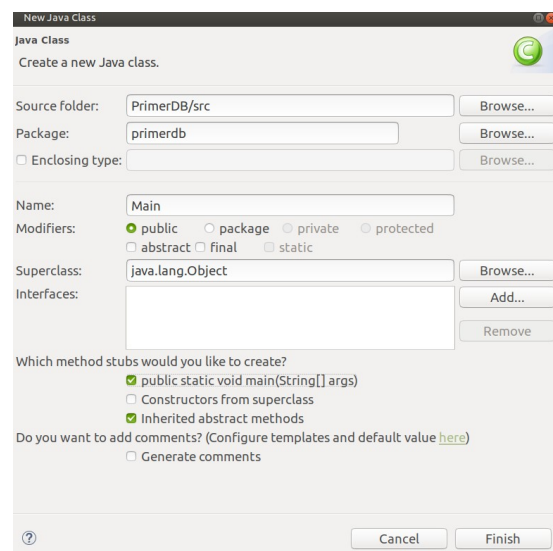
Ese fichero es el que define como particionar nuestra aplicación. A partir de Java 8 se decidió que era conveniente que únicamente se cargaran las partes de java que fuéramos a usar (javafx por ejemplo, es un módulo aparte) y es la razón de este fichero

Ahora vamos a crear un paquete y dentro vamos a poner una clase Main

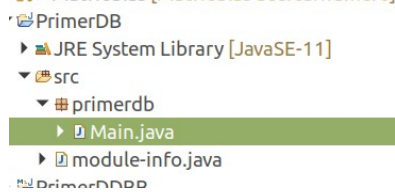
Botón derecho encima de:
“src” y agregamos el paquete:
primerdb



Dentro del paquete creamos una nueva clase que llamamos Main y le decimos que nos incorpore ya el public static void main:

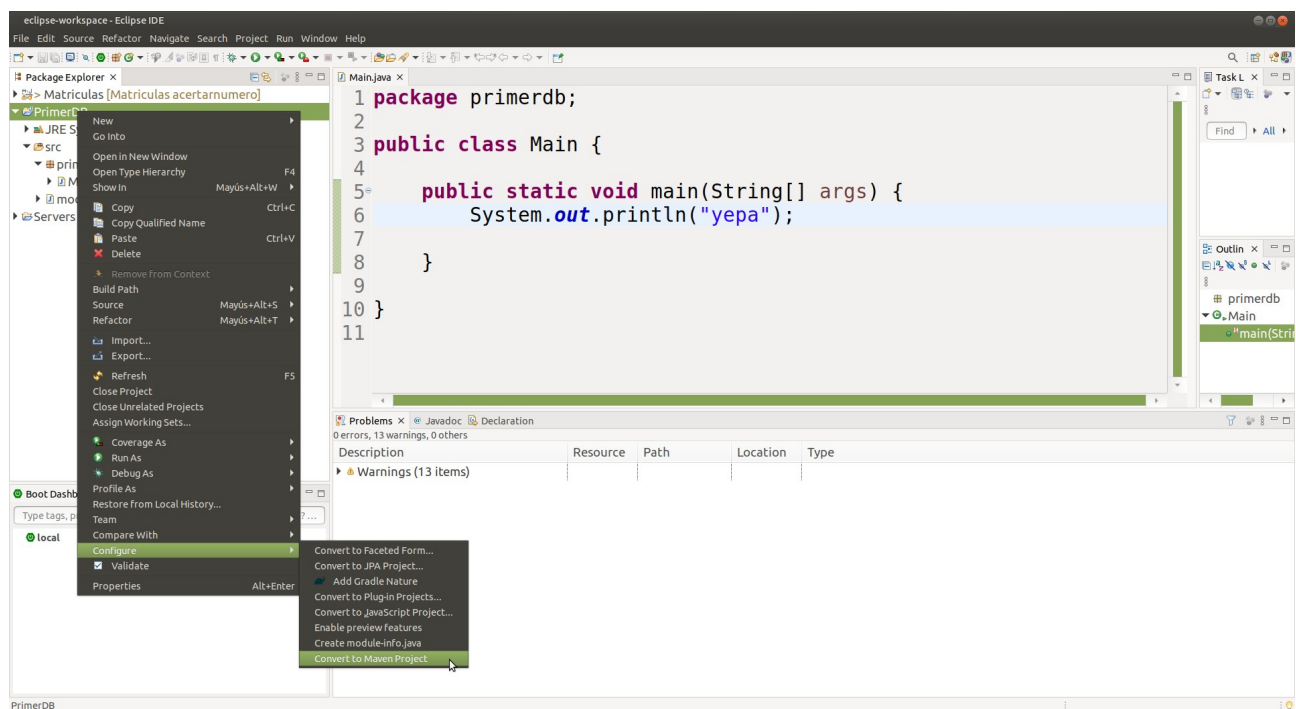


Nos quedará así:

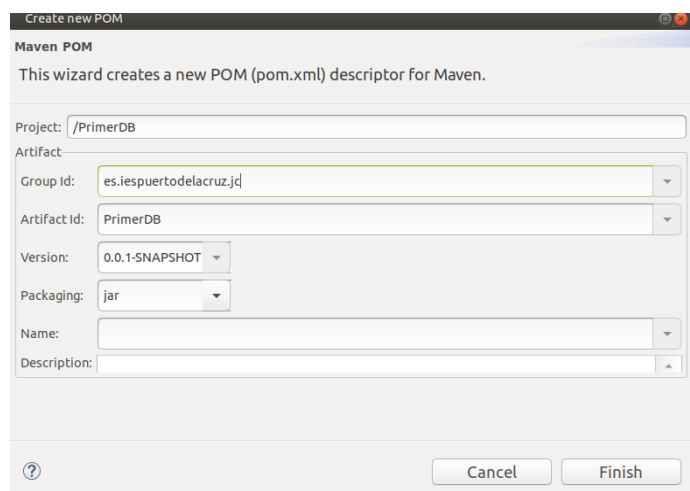


```
1 package primerdb;
2
3 public class Main {
4
5     public static void main(String[] args) {
6
7
```

Ahora que tenemos el proyecto creado vamos a convertirlo en proyecto maven:
botón derecho sobre proyecto → configure → convert to maven project



Ponemos un artifactid y un groupid
y pulsamos en finish



Para trabajar con bases de datos agregamos la dependencia maven de mysql:

Dependencia(en texto copiable) e imagen de todo el pom.xml:

```
<dependencies>
  <dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.26</version>
  </dependency>
</dependencies>
```

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>es.iespuertodelacruz.jc</groupId>
  <artifactId>PrimerDB</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <dependencies>
    <!-- https://mvnrepository.com/artifact/mysql/mysql-connector-java -->
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
      <version>8.0.26</version>
    </dependency>
  </dependencies>
  <build>
    <sourceDirectory>src</sourceDirectory>
    <plugins>
      <plugin>
        <artifactId>maven-compiler-plugin</artifactId>
        <version>3.8.1</version>
        <configuration>
          <release>11</release>
        </configuration>
      </plugin>
    </plugins>
  </build>
</project>
```

Ahora ya hacemos lo de actualizar maven (botón derecho sobre el proyecto → maven → update project)

Haciendo las modificaciones en los módulos para agregar bases de datos para Java > 8

Vamos a copiar el siguiente código en el Main (van a aparecer múltiples errores de import)

```
public class Main {
    private static void cargarDriverMysql(){
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
        } catch (ClassNotFoundException ex) {
            System.err.println("no carga el driver");
            System.exit(1);
        }
    }

    public static void main(String[] args) {

        cargarDriverMysql();
        Connection conexion = null;
        try {
            conexion = DriverManager.getConnection(
                "jdbc:mysql://localhost/oficina?serverTimezone=UTC", "root", "1q2w3e4r");

            Statement s = conexion.createStatement();
            String sql = "select * from lapices";
            ResultSet rs = s.executeQuery(sql);

            while(rs.next()){

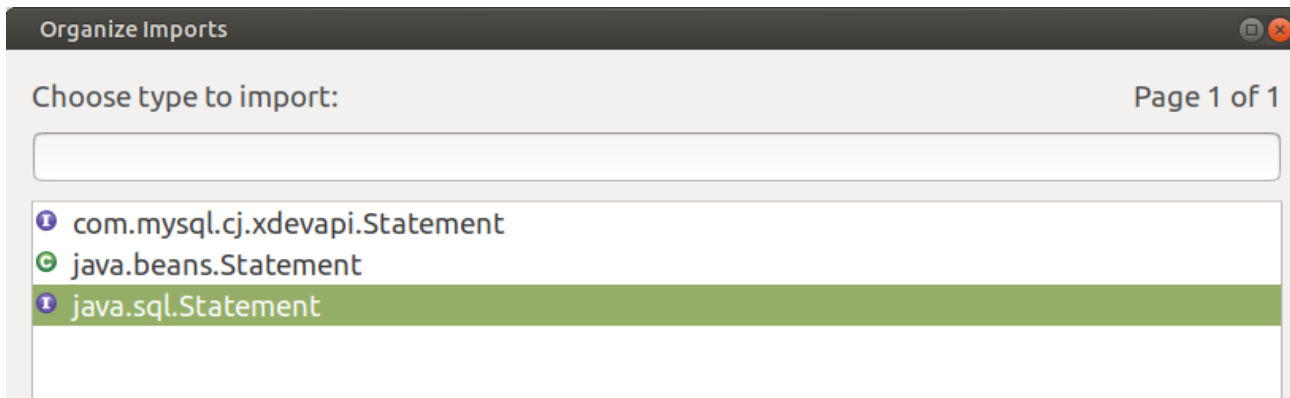
                int id = rs.getInt("idlapiz");
                String marca = rs.getString("marca");
                int numero = rs.getInt("numero");

                System.out.println("Lápiz: " + id + " " + marca + " " + numero);

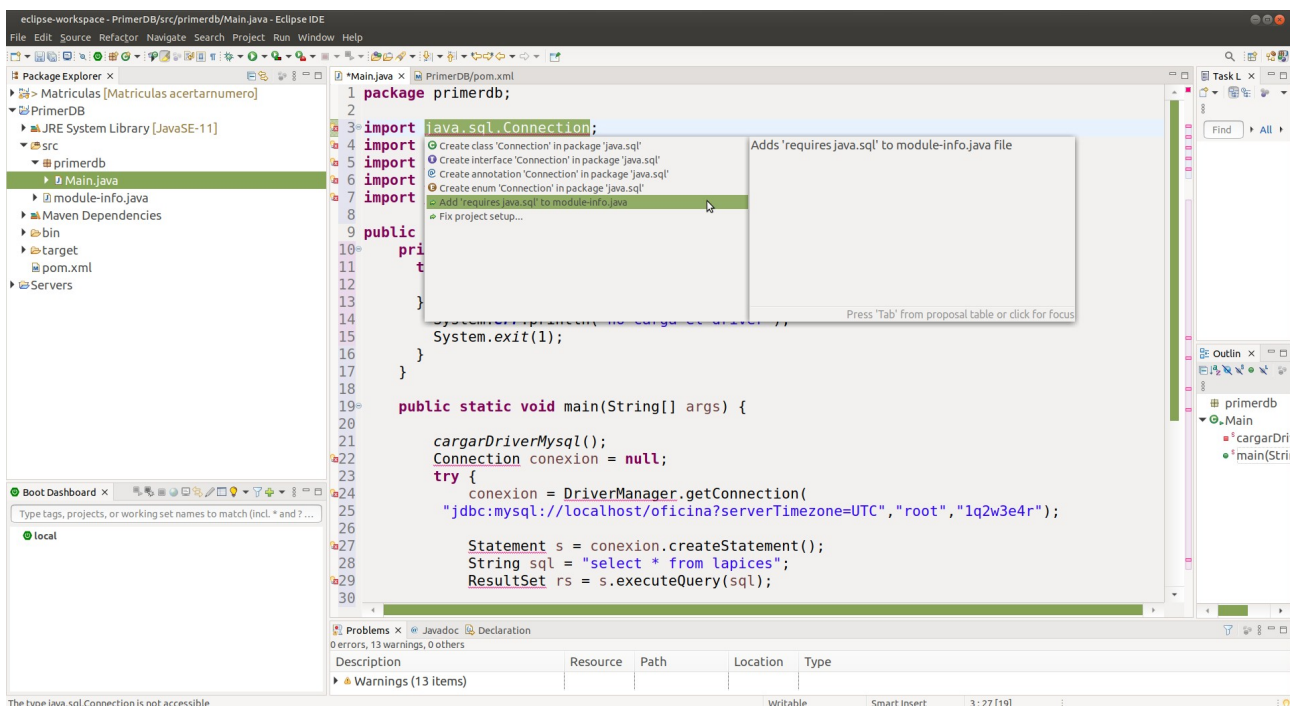
            }
            s.close();
            conexion.close();

        } catch (SQLException ex) { ex.printStackTrace(); }
    }
}
```

Pulsamos: CTRL + SHIFT + O para que nos arregle los import.
Nos preguntará que import elegir. Elegimos java.sql.Statement



Una vez hecho veremos que sigue dando error (ahora en los import que nos ha generado) Si nos ponemos con botón derecho encima del import podemos solucionarlo haciendo click con el ratón y diciéndole: **Add 'requires.java.sql' to module-info.java**



Veamos el module.info que nos ha generado:



(Si aún se viera algún error decirle de nuevo que actualice maven: update maven project)