

Curso de Programación en Java



Juan Francisco Maldonado León
Arquitecto de Software



Java

Database Connectivity



Juan Francisco Maldonado León
Arquitecto de Software



Bases de Datos Relacionales

Una base de datos relacional es una representación lógica de datos que permite acceder a éstos sin necesidad de considerar su estructura física y almacena los datos en tablas



Bases de Datos Relacionales

	Número	Nombre	Departamento	Salario	Ubicación
Fila {	23603	Jones	413	1100	New Jersey
	24568	Kerwin	413	2000	New Jersey
	34589	Larson	642	1800	Los Angeles
	35761	Myers	611	1400	Orlando
	47132	Neumann	413	9000	New Jersey
	78321	Stephens	611	8500	Orlando
	Clave primaria		Columna		

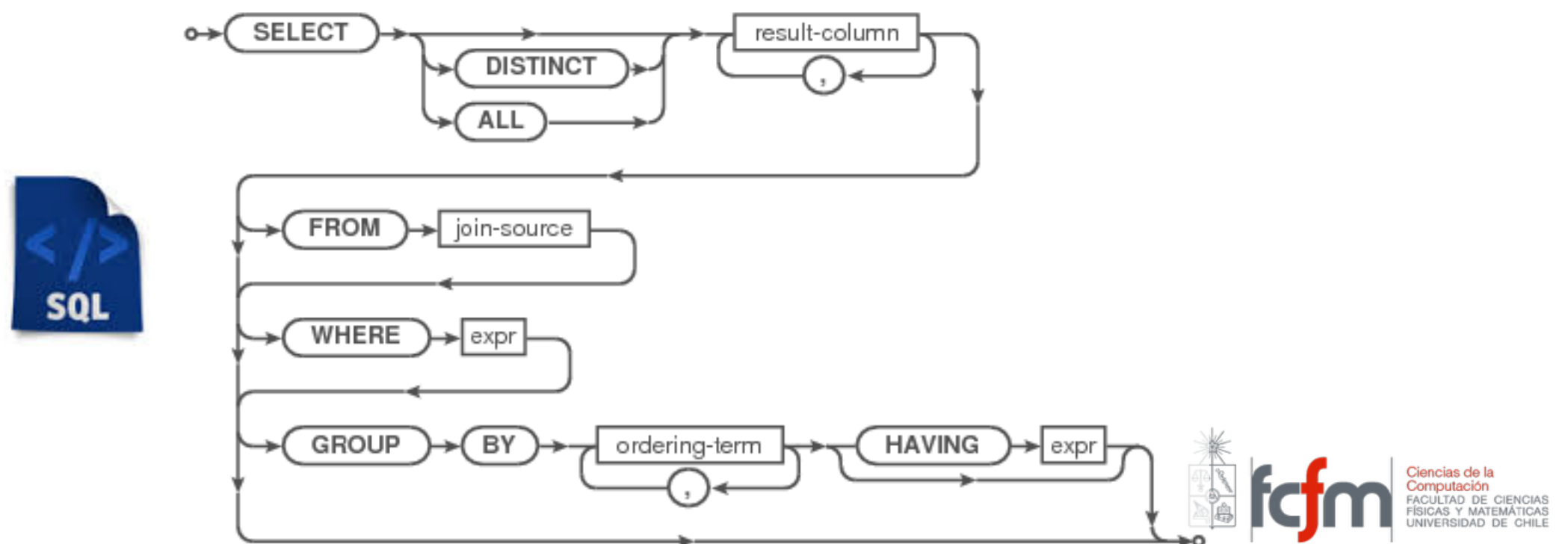
Bases de Datos Relacionales

- Una base de datos se compone de varias tablas o relaciones.
- No pueden existir dos tablas con el mismo nombre ni registro.
- Cada tabla es a su vez un conjunto de campos (columnas) y registros (filas).
- **Primary Key**: son la clave principal de un registro dentro de una tabla y estas deben cumplir con la integridad de datos.
- **Foreign Key**: se colocan en la tabla relacionada, contienen el mismo valor que la clave primaria del registro

SQL

Structured Query Language

es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.



SQL

Palabra clave de SQL	Descripción
SELECT	Obtiene datos de una o más tablas.
FROM	Las tablas involucradas en la consulta. Se requiere para cada SELECT.
WHERE	Los criterios de selección que determinan cuáles filas se van a recuperar, eliminar o actualizar. Es opcional en una consulta o instrucción de SQL.
GROUP BY	Criterios para agrupar filas. Es opcional en una consulta SELECT.
ORDER BY	Criterios para ordenar filas. Es opcional en una consulta SELECT.
INNER JOIN	Fusionar filas de varias tablas.
INSERT	Insertar filas en una tabla especificada.
UPDATE	Actualizar filas en una tabla especificada.
DELETE	Eliminar filas de una tabla especificada.



MySQL es un sistema de gestión de bases de datos relacional desarrollado bajo licencia dual GPL/Licencia comercial por Oracle Corporation y está considerada como la base datos open source más popular del mundo

MySQL Community Server 5.7.13

<https://dev.mysql.com/downloads/mysql/>

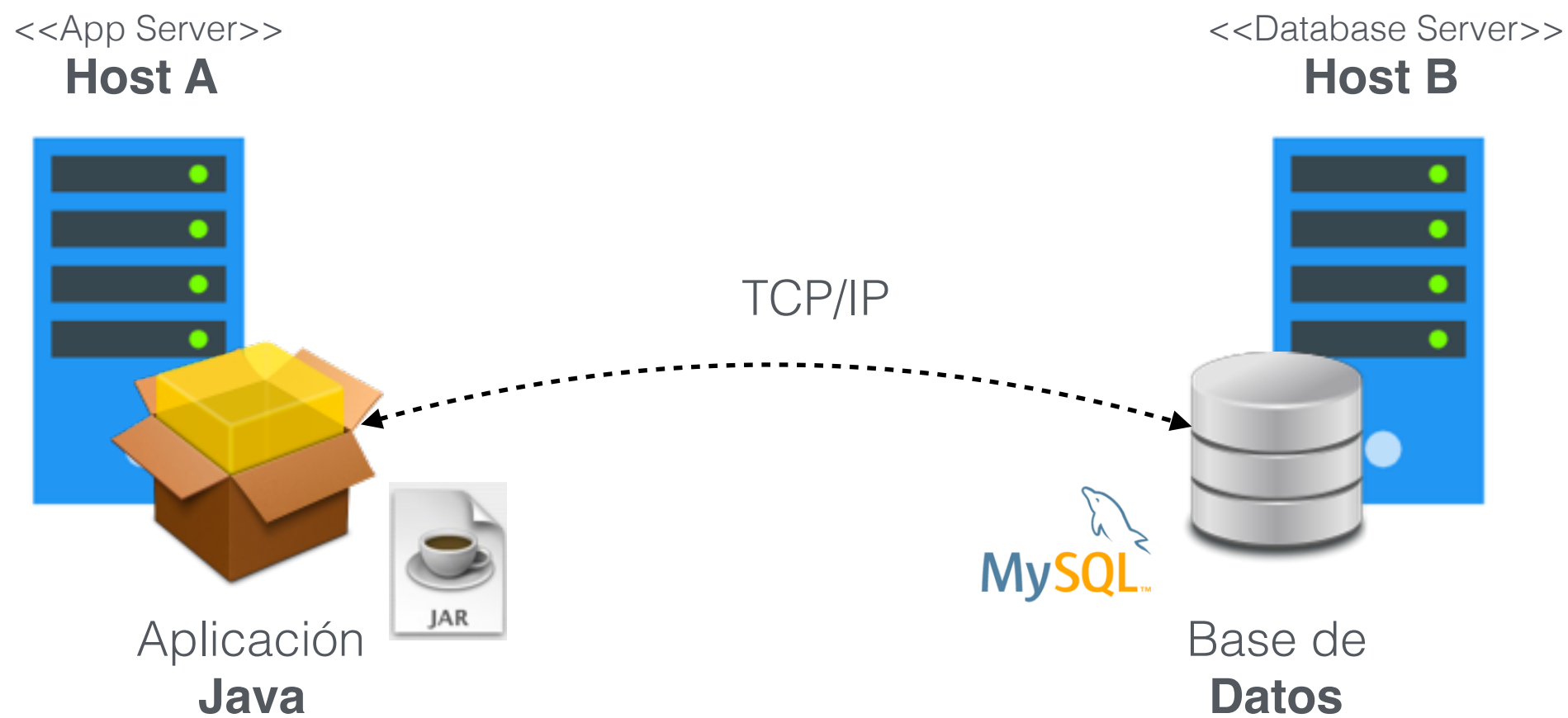
JDBC

Java Database Connectivity

Java Database Connectivity, más conocida por sus siglas JDBC es una API que permite la ejecución de operaciones sobre bases de datos relacionales, independiente de la base de datos a la cual se accede (Mysql, Oracle, SQLServer, etc).

JDBC es un api que viene incluida en la jdk, en el package
`java.sql`

JDBC



Conectar con una base de datos

Para conectarse a un base de datos, lo primero que debemos conocer, es el tipo de base de dato que utilizaremos, para cargar el driver de conexión.

Ejemplo

mysql-connector-java-5.1.39-bin.jar

<http://dev.mysql.com/downloads/connector/j/>



Introducción a Java

Java Database Connectivity

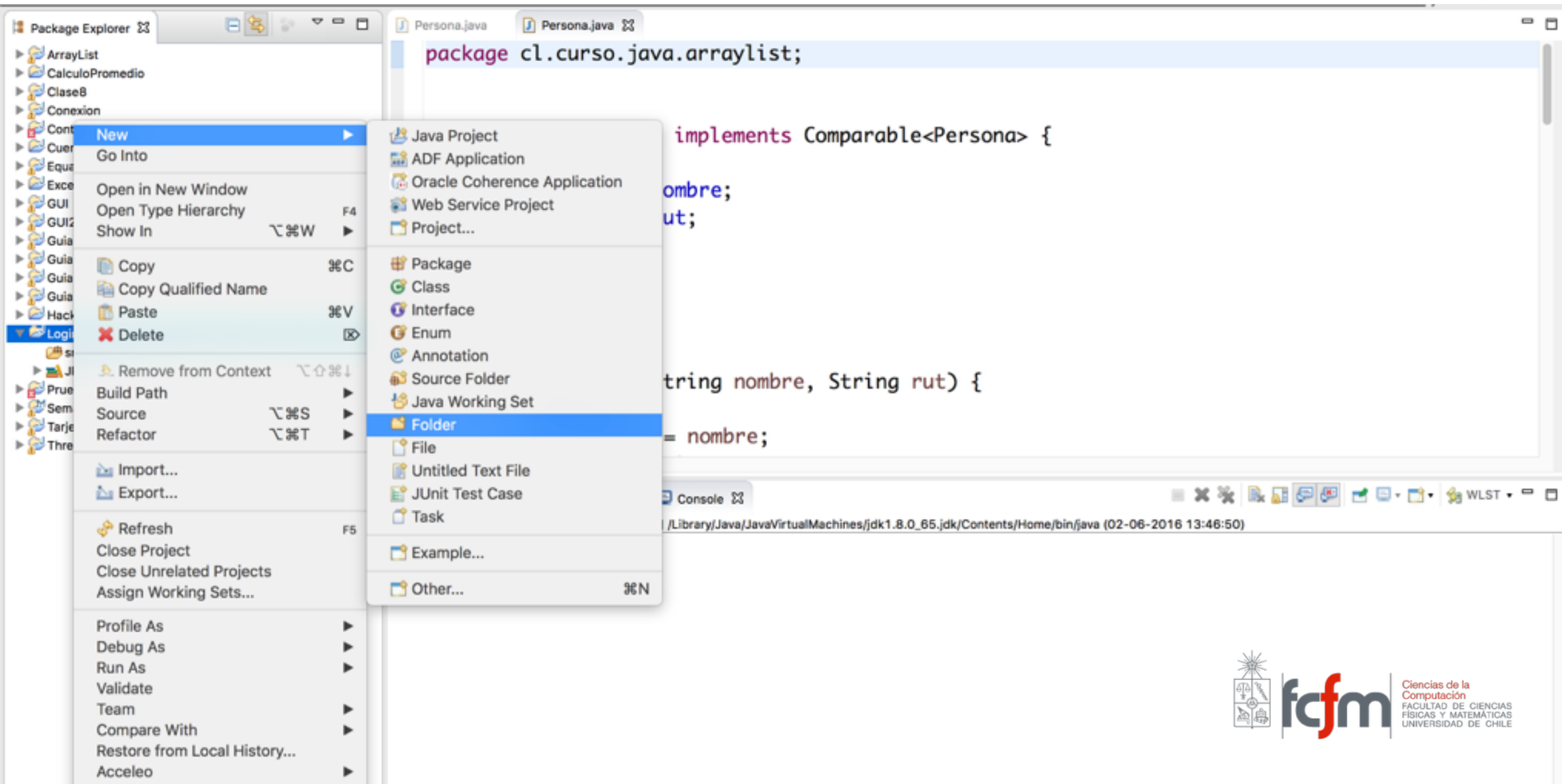
Conectar con una base de datos

Driver de la base de datos

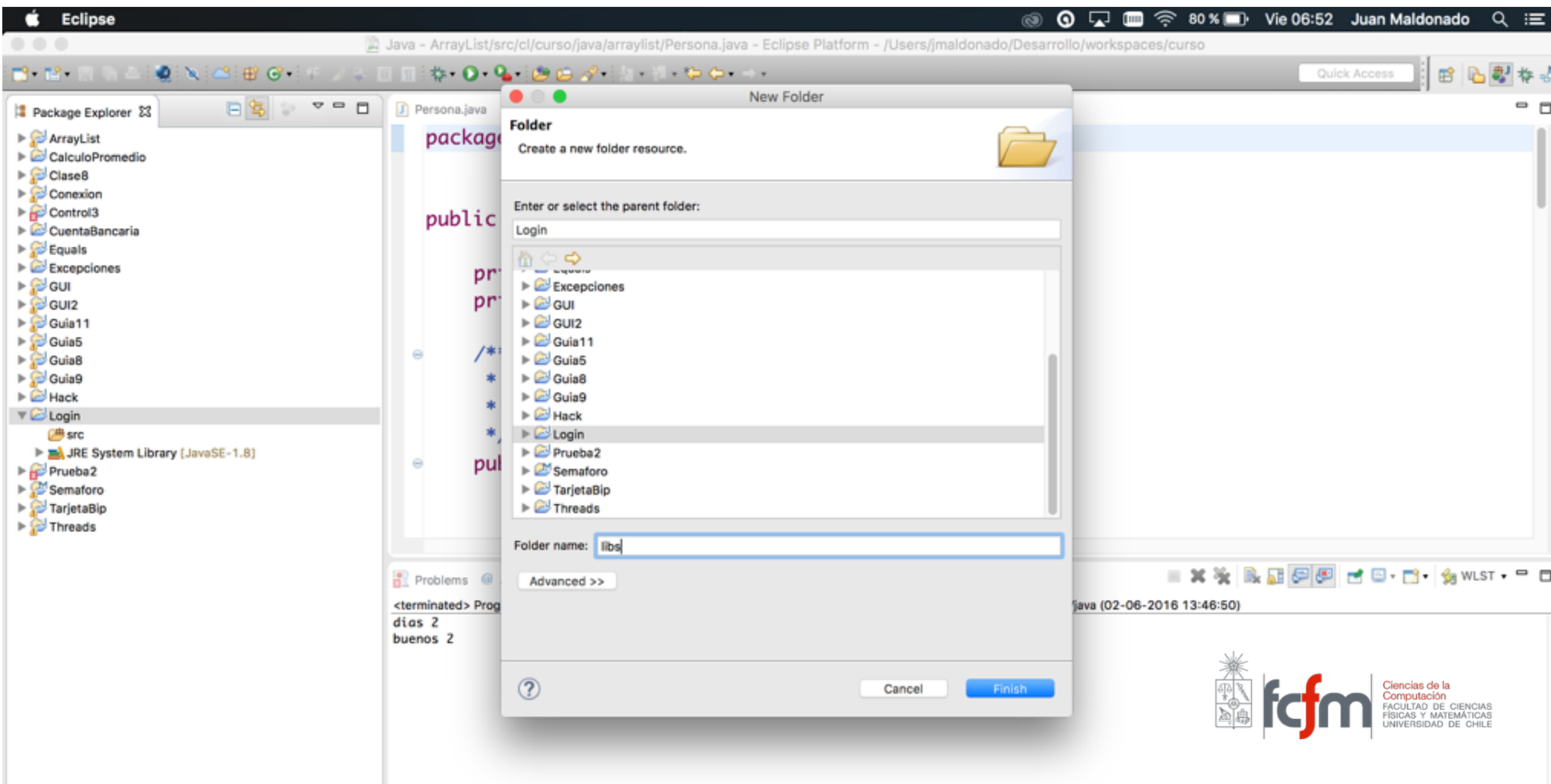
```
try {  
    Class.forName("com.mysql.jdbc.Driver");  
} catch (ClassNotFoundException e) {  
    System.out.println("El driver requerido no esta cargado.");  
}
```

Si el driver de la base de datos no esta cargado correctamente lanzara una excepción de `ClassNotFoundException`

Cargar Driver

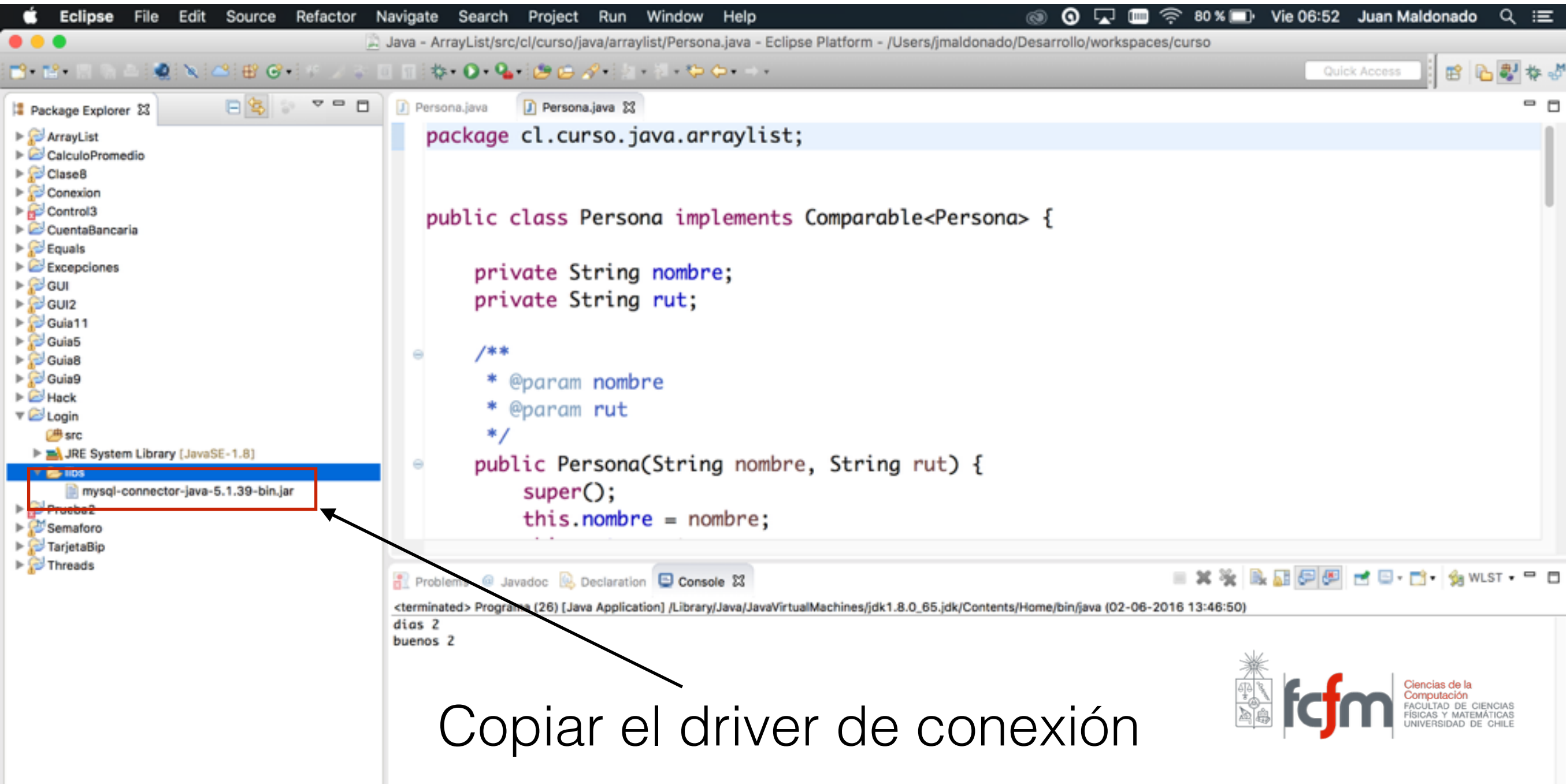


Cargar Driver



The screenshot shows the Eclipse IDE interface. The Package Explorer on the left lists several packages, including 'Login'. The 'New Folder' dialog box is open, prompting the user to 'Create a new folder resource.' The dialog shows a list of parent folders, including 'Login', 'Prueba2', 'Semaforo', 'TarjetaBip', and 'Threads'. The 'Folder name' field is set to 'libs'.

Cargar Driver



Eclipse File Edit Source Refactor Navigate Search Project Run Window Help

Java - ArrayList/src/cl/curso/java/arraylist/Persona.java - Eclipse Platform - /Users/jmaldonado/Desarrollo/workspaces/curso

Package Explorer

- ArrayList
- CalculoPromedio
- Clase8
- Conexion
- Control3
- CuentaBancaria
- Equals
- Excepciones
- GUI
- GUI2
- Guia11
- Guia5
- Guia8
- Guia9
- Hack
- Login
- src
 - JRE System Library [JavaSE-1.8]
 - mysql-connector-java-5.1.39-bin.jar**
- Prueba2
- Semaforo
- TarjetaBip
- Threads

Persona.java

```
package cl.curso.java.arraylist;

public class Persona implements Comparable<Persona> {

    private String nombre;
    private String rut;

    /**
     * @param nombre
     * @param rut
     */
    public Persona(String nombre, String rut) {
        super();
        this.nombre = nombre;
    }
}
```

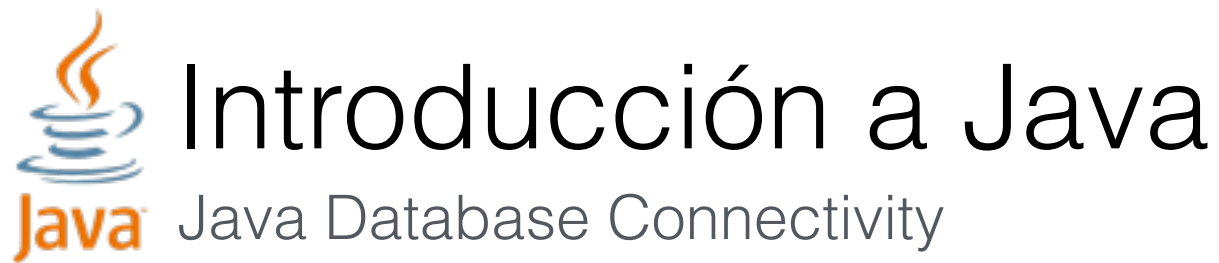
Problems Javadoc Declaration Console

<terminated> Programa (26) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_65.jdk/Contents/Home/bin/java (02-06-2016 13:46:50)

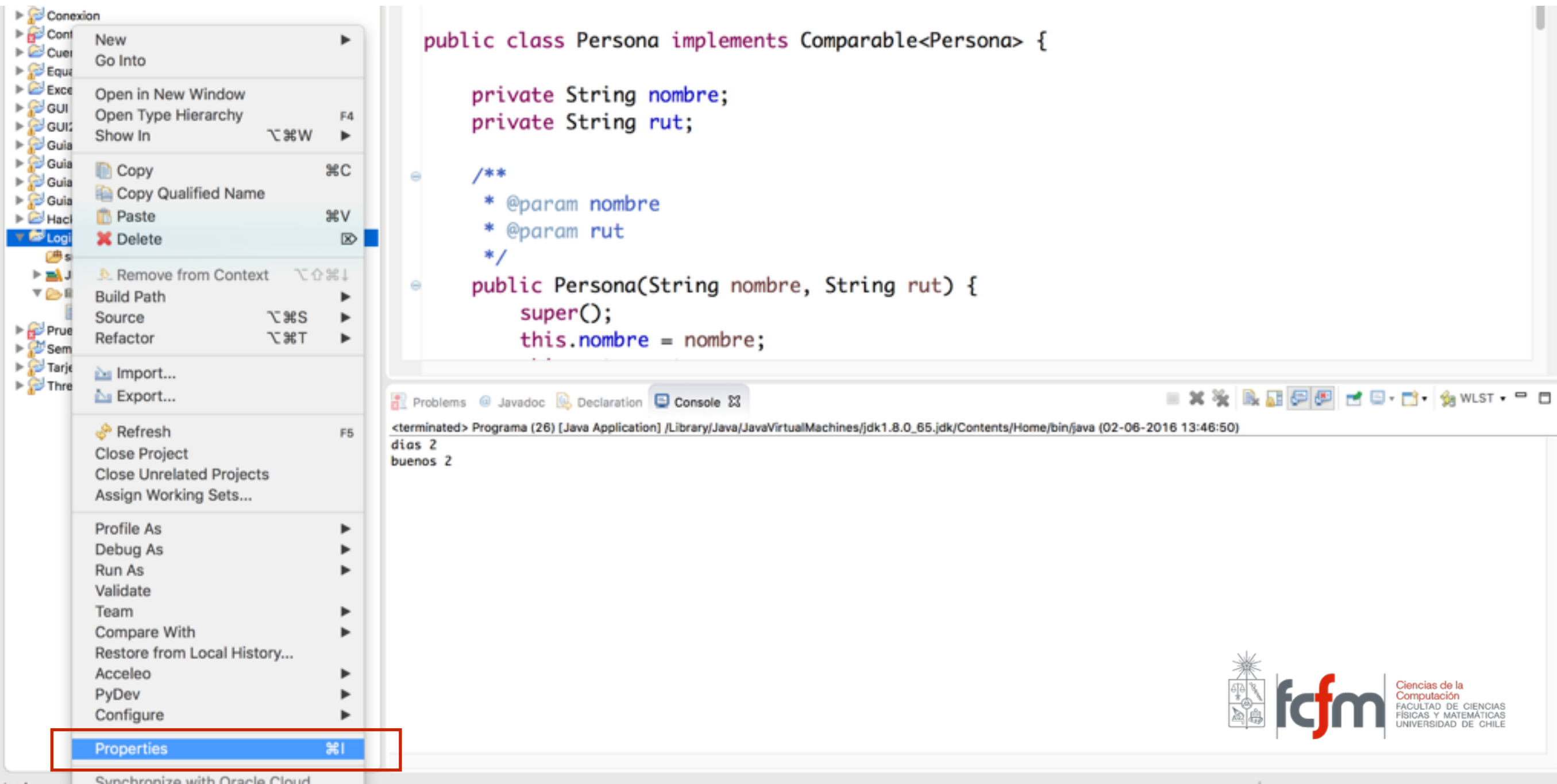
días 2
buenos 2

Copiar el driver de conexión

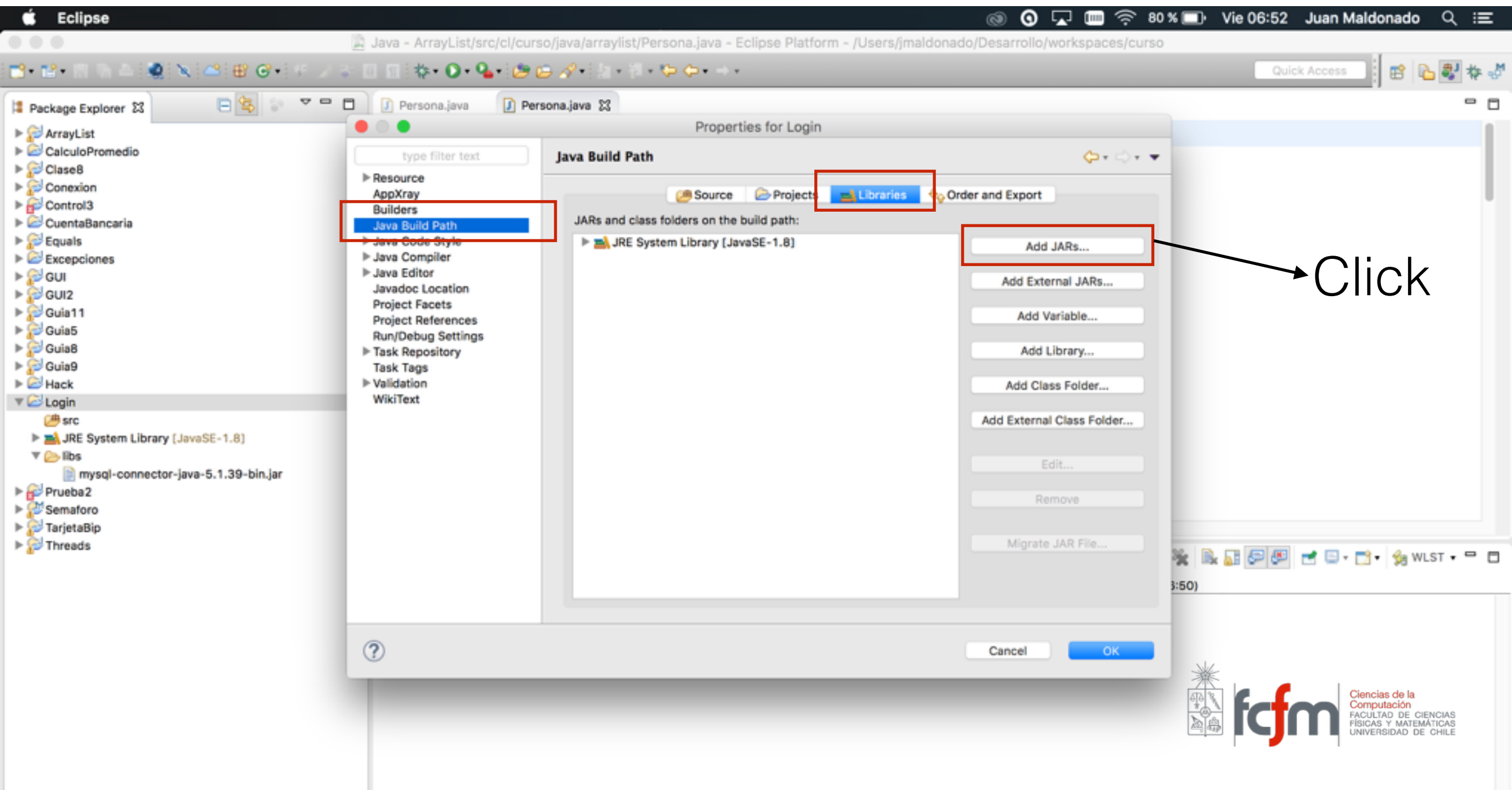
fcfm Ciencias de la Computación
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE



Cargar Driver



Cargar Driver

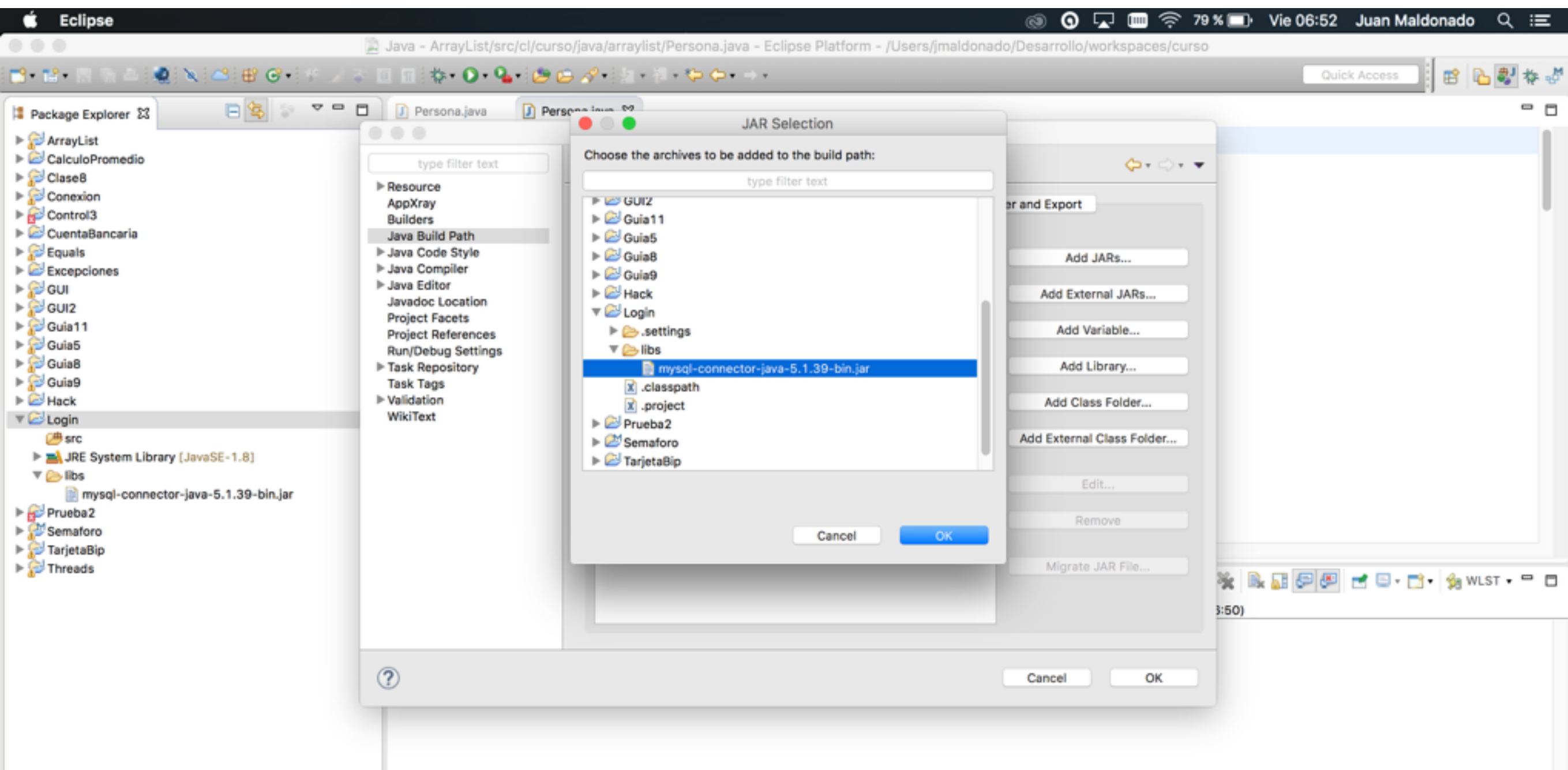


The screenshot shows the Eclipse IDE interface. The 'Package Explorer' on the left lists several packages, including 'Login'. The 'Properties for Login' dialog box is open, showing the 'Java Build Path' tab. The 'Libraries' sub-tab is selected, and the 'Add JARs...' button is highlighted with a red box. An arrow points to this button with the text 'Click'. The 'Libraries' list shows 'JRE System Library [JavaSE-1.8]'. The 'Add JARs...' button is located on the right side of the dialog box.

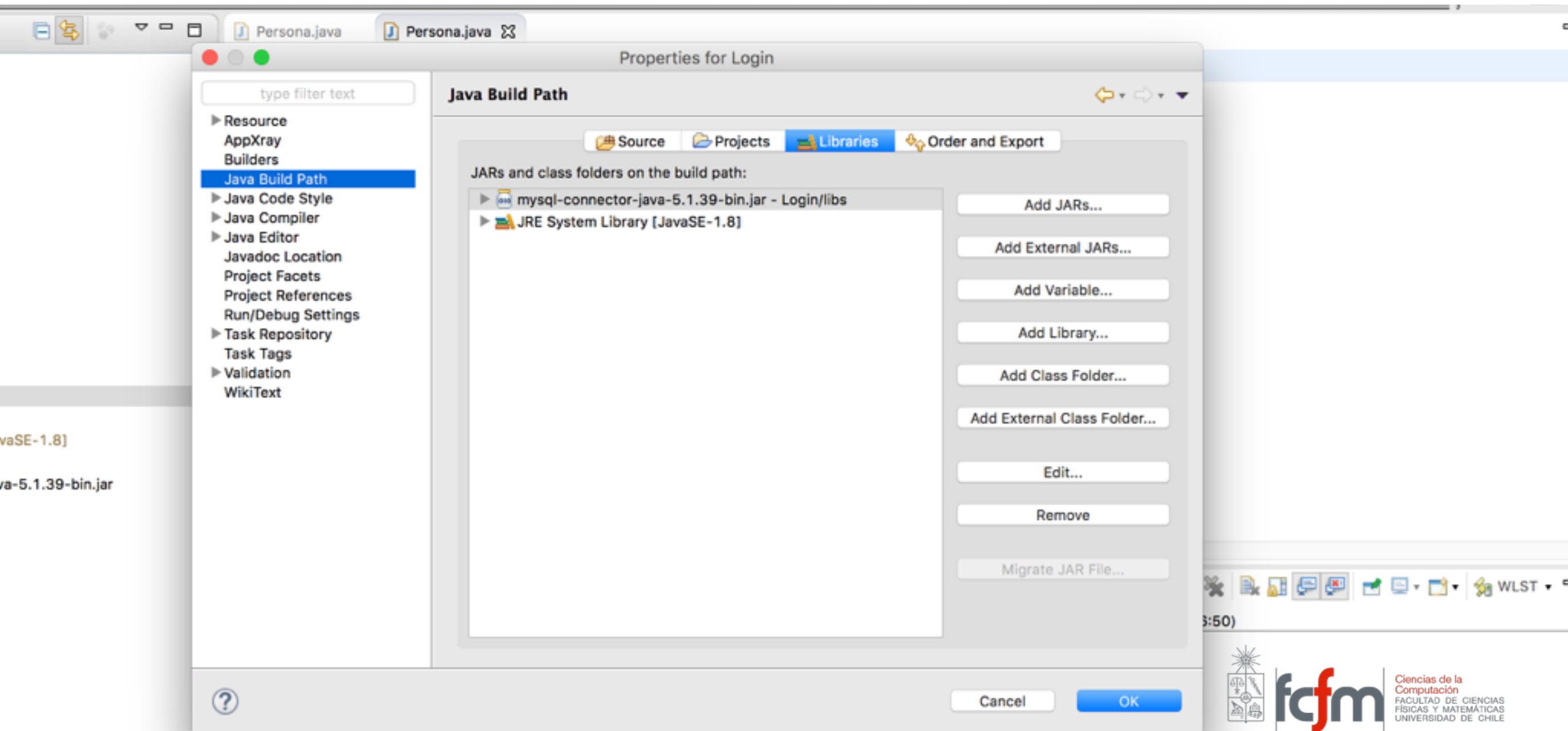
Introducción a Java

Java Database Connectivity

Cargar Driver



Cargar Driver

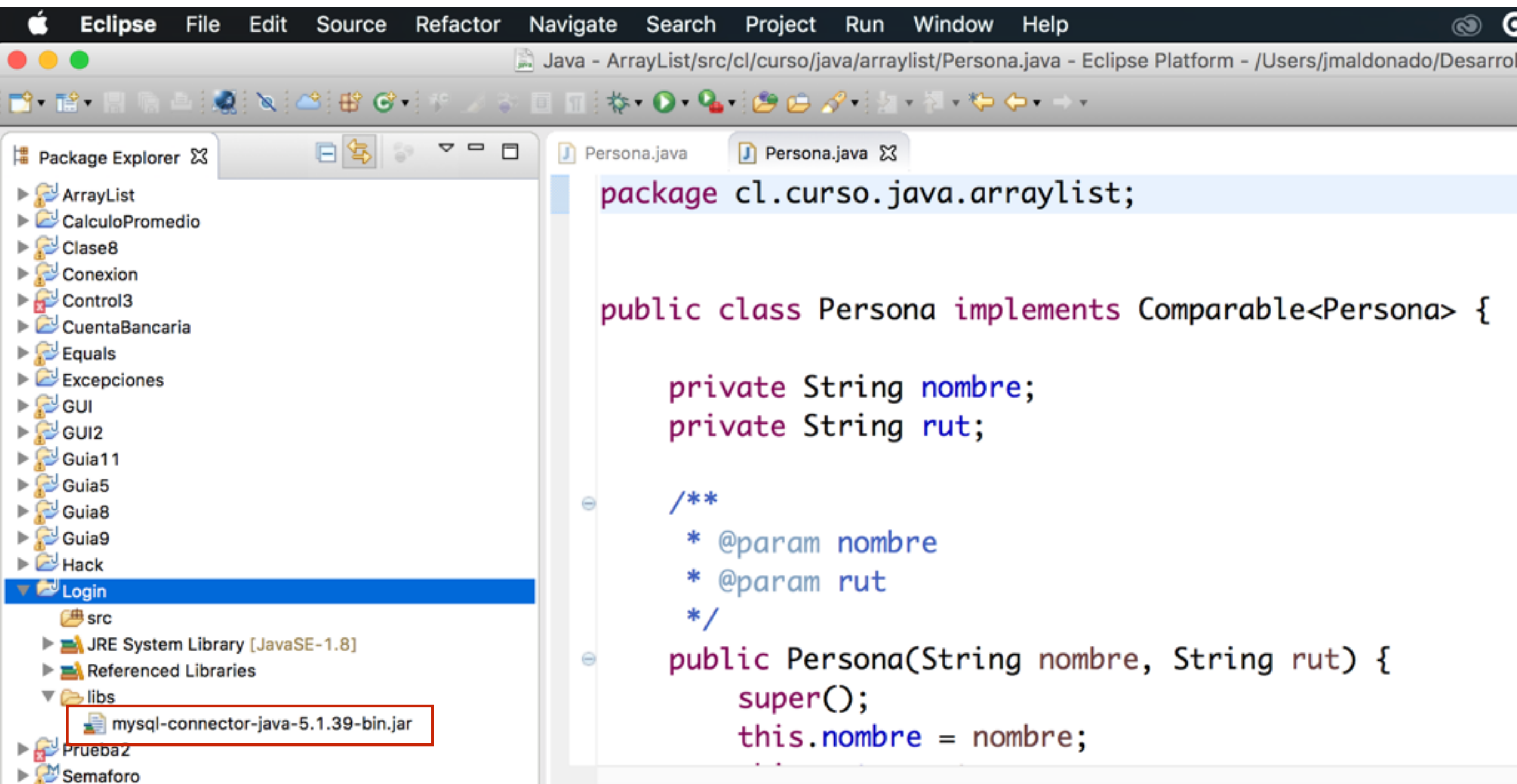




Introducción a Java

Java Database Connectivity

Cargar Driver



Conectar con una base de datos

También debemos conocer los datos de conexión los cuales son

- Dirección IP
- Puerto
- Nombre de la Base de datos
- Usuario de la base de datos
- Password

Conectar con una base de datos

```
Connection conexion = DriverManager.  
    getConnection ("jdbc:mysql://192.168.0.108:3306/curso", "alumnos", "java123");
```

- Dirección IP : **192.168.0.108**
- Puerto : **3306**
- Nombre de la Base de datos : **curso**
- Usuario de la base de datos : **alumnos**
- Password : **java123**

Conectar con una base de datos

Columnas de la tabla alumnos

idalumno	nombre	apellido	apellidoMater...	edad
1	juan	maldonado	leon	27
2	francisco	leon	maldonado	26

registros de la tabla alumnos

Clave Primaria

Consultas sobre BD

SQL para realizar consulta a la tabla alumno

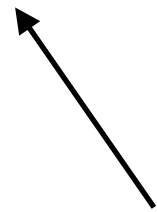
"select * from alumnos"

Elementos a retornar de la tabla alumnos, * es para traer todas las columnas de la tabla

Nombre de la tabla

Consultas sobre BD

```
Statement s = conexion.createStatement();  
ResultSet rs = s.executeQuery ("select * from alumnos");
```



El resultado de la consulta se almacenara en el objeto
ResultSet



Introducción a Java

Java Database Connectivity

Iterar sobre ResultSet

```
Statement s = conexion.createStatement();
ResultSet rs = s.executeQuery ("select * from alumnos");

while (rs.next()) {
    System.out.println(rs.getInt( 1));
    System.out.println(rs.getInt("idalumno"));

    System.out.println(rs.getString("nombre"));
    System.out.println(rs.getString(2));
}
```



fcjm

Ciencias de la
Computación
FACULTAD DE CIENCIAS
FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Insertar un registro

Sentencia SQL

Nombre de la Tabla

Campos de la tabla

```
INSERT INTO curso.alumnos (idalumno, nombre, apellido, apellidoMaterno)  
VALUES ('3', 'James', 'Gosling', 'Lion');
```

Valores que se desean insertar

Insertar un registro

Desde Java

```
Statement sentenciaInsert = conexion.createStatement();  
int resultadoInsert = sentenciaInsert  
    .executeUpdate("INSERT INTO alumnos (idalumno, nombre, apellido, apellidoMaterno)"  
        + " VALUES ('3', ' James', 'Gosling', 'lion');");  
  
System.out.println("Resultado : " + resultadoInsert);
```


Actualizar un registro

Sentencia SQL

Nombre de la Tabla

Campos de la tabla

```
UPDATE curso.alumnos SET nombre='juan francisco'
WHERE idalumno='1';
```

clausula WHERE para determinar
que campos se actualizaran

Actualizar un registro

Desde Java

```
Statement sentenciaUpdate = conexion.createStatement();  
int resultadoUpdate = sentenciaUpdate  
    .executeUpdate("UPDATE alumnos SET nombre='juan francisco'"  
        + " WHERE idalumno='1';");  
  
System.out.println("Resultado modificacion : " + resultadoUpdate);
```

Eliminar un registro

Sentencia SQL

```
DELETE FROM curso.alumnos WHERE idalumno='3';
```

Eliminar un registro

Desde Java

```
Statement sentenciaDelete = conexion.createStatement();  
int resultadoDelete = sentenciaDelete  
    .executeUpdate("DELETE FROM curso.alumnos WHERE idalumno='3'");  
  
System.out.println("Resultado modificacion : " + resultadoDelete);
```

API

Clases para utilizar una base de datos

Clase	Descripción
DriverManager	Para cargar un driver
Connection	Para establecer conexiones con las bases de datos
Statement	Para ejecutar sentencias SQL y enviarlas a las BBDD
PreparedStatement	La ruta de ejecución está predeterminada en el servidor de base de datos que le permite ser ejecutado varias veces
ResultSet	Para almacenar el resultado de la consulta



Introducción a Java

Java Database Connectivity

```
Class.forName("com.mysql.jdbc.Driver");
Connection conexion = DriverManager.getConnection(
    "jdbc:mysql://localhost:3306/curso", "jmaldonado",
    "java123");

Statement sentenciaInsert = conexion.createStatement();
int resultadoInsert = sentenciaInsert
    .executeUpdate("INSERT INTO alumnos (idalumno, nombre, apellido, apellidoMaterno)"
        + " VALUES ('3', ' James', 'Gosling', 'lion');");
System.out.println("Resultado : " + resultadoInsert);

Statement sentenciaUpdate = conexion.createStatement();
int resultadoUpdate = sentenciaUpdate
    .executeUpdate("UPDATE alumnos SET nombre='juan francisco'"
        + " WHERE idalumno='1';");
System.out.println("Resultado modificacion : " + resultadoUpdate);

Statement sentenciaDelete = conexion.createStatement();
int resultadoDelete = sentenciaDelete
    .executeUpdate("DELETE FROM curso.alumnos WHERE idalumno='3';");
System.out.println("Resultado modificacion : " + resultadoDelete);
```



fcfm

Ciencias de la
Computación
FACULTAD DE CIENCIAS
FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE