NOMBRE		CICLO FORMATIVO	CURSO
		DAW/DAM	1ST
APELLIDOS		MODULO	CONVOCATORIA
		PROGRAMMING	3RD QUARTERLY (SERVER)
DNI	FECHA	NOTA	
	MAY 31st 2023		

After executing the examen.sql script perform the following actions:

```
(how to execute it: mysql -u root -p < examen.sql)
```

1. **(5p – 1p each function)** Create on table Person basic CRUD operations following the next interface:

```
public interface IPersonRepository {
    Person getPersonById(int id) throws SQLException;      // use CallableStatement
    List<Person> getAllPerson() throws SQLException;        // use Statement
    Person deletePersonById(int id) throws SQLException;    // use PrepareStatement
    Person updatePerson(Person person) throws SQLException; // use PrepareStatement
    Person addPerson(Person person) throws SQLException;   // use CallableStatement
}
```

You can use following class to access to MySQL database:

```
import com.mysql.cj.jdbc.MysqlDataSource;
import javax.sql.DataSource;

public class MyDataSource {
    public static DataSource getMyDataSource() {
        MysqlDataSource mySQL = new MysqlDataSource();
        mySQL.setURL("jdbc:mysql://localhost:3306/bbddJava");
        mySQL.setUser("examen");
        mySQL.setPassword("1234");
        return mySQL;
    }
}
```

HINT: To download the JDBC, remember to add the following lines in the pom.xml file:

```
<dependency>
    <groupId>mysql</groupId>
    <artifactId>mysql-connector-java</artifactId>
    <version>8.0.33</version>
</dependency>
```

2. **(1p.)** Now, create a Main class to test previous interfaces. This Main class will contain calls for all the methods.

NOMBRE	APELLIDOS

3. **(2p.)** Create a couple of methods to:
   a. Save all the person objects into a file, that is, serializing the person objects.
   b. Load a csv file having text with persons to add and separated by commas (,). This file will always contain the first name, the last name and the house identification fields. Example:

```
Pepa,Navarro,1
Luna,Guerrero,2
Blanca,Rosillo,3
```

4. **(2p.)** Create an APIRest application made with Spring Boot allowing the CRUD operations calls from a web application:
   - You must use the @RequestMapping("/api/")
   - In the controller part, all methods will return ResponseEntity<?>
   - Methods having id as parameter will use @PathVariable("id") int id as parameter in the controller part, and the ones having person object as parameter will use @RequestBody Person person

HINT: But if you are going to create the APIRest application using Spring Boot you need to add following dependences: Spring Boot Dev Tools, Spring Web, MySQL driver.