

Navegación por el cuestionario

1	1	2	3	4	5	6	7	8	9
10									

Finalizar revisión

Comenzado el	domingo, 21 de abril de 2019, 19:08
Estado	Finalizado
Finalizado en	martes, 23 de abril de 2019, 17:38
Tiempo empleado	1 día 22 horas
Calificación	5,67 de 10,00 (57%)

Información
 Marcar pregunta

No os guiéis por las correcciones porque lo único que hacen es comparar vuestro código con una posible solución.

Desarrollar y ejecutad el programa en vuestro IDE y comparad la solución con la planteada.
Se usará el usuario desvan con la contraseña desvan. Recordad configurar la IP de BBDD con la que estáis trabajando.

Todas las tablas que se usan son de la tarea de aprendizaje 2 (Desvan) de BD05.

Para cualquier duda ya sabéis donde estoy.

Pregunta 1
Correcta
Puntúa 1,00 sobre 1,00
 Marcar pregunta

El modelo orientado a objetos trata con conjuntos y relaciones, debido a su naturaleza matemática. ¿Verdadero o falso?

Seleccione una:

Verdadero

Falso ✓

La respuesta correcta es 'Falso'

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 2
Correcta
Puntúa 1,00 sobre 1,00
 Marcar pregunta

Para una conexión a una base de datos, se puede utilizar el método `getConnection()` de la clase `DriverManager`. ¿Verdadero o falso?

Seleccione una:

Verdadero ✓

Falso

La respuesta correcta es 'Verdadero'

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 3
Correcta
Puntúa 1,00 sobre 1,00
 Marcar pregunta

Un `ResultSet`:

Seleccione una o más de una:

a. Es un comando de SQL estándar.

b. Permite procesar el resultado de una consulta `SELECT` ✓

c. Ninguna es cierta

d. Sirve para contener el resultado del comando `SELECT` . ✓

La respuesta correcta es: Sirve para contener el resultado del comando `SELECT` ., Permite procesar el resultado de una consulta `SELECT`.

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 4
Correcta
Puntúa 0,67 sobre 1,00
 Marcar pregunta

Un ResultSet:

Seleccione una o más de una:

a. Va obteniendo de la base de datos, los datos según se van pidiendo. ✓

b. Tiene el inconveniente de que no posee métodos que permitan movernos hacia delante y hacia atrás en las filas, para obtener la información de cada fila.

c. No contiene todos los datos de la consulta en el mismo momento de hacer la consulta `SELECT` . ✓

d. Todas las respuestas son correctas.

La respuesta correcta es: Va obteniendo de la base de datos, los datos según se van pidiendo., No contiene todos los datos de la consulta en el mismo momento de hacer la consulta `SELECT` .

Correcta

Puntos para este envío: 1,00/1,00. Contando con los intentos anteriores, daría 0,67/1,00.

Pregunta 5
Correcta
Puntúa 1,00 sobre 1,00
 Marcar pregunta

El método `getMessage()` :

Seleccione una:

a. Devuelve un número entero que representa el código de error asociado

b. Todas son correctas.

c. imprime el mensaje de error asociado a la excepción que se ha producido ✓

La respuesta correcta es: imprime el mensaje de error asociado a la excepción que se ha producido

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 6
Correcta
Puntúa 1,00 sobre 1,00
 Marcar pregunta

Respecto a las consultas de actualización, `executeUpdate` , retorna el número de registros insertados, cuando estamos insertando datos. ¿Verdadero o falso?

Seleccione una:

Verdadero ✓

Falso

La respuesta correcta es 'Verdadero'

Correcta

Puntos para este envío: 1,00/1,00.

Pregunta 7
Incorrecta
 Marcar pregunta

De los ejemplos que hemos estudiado elige el que mejor se adapta a la siguiente consulta, modificalo y obtén los datos solicitados con el formato exigido. Para ello, se usará el usuario desvan con la contraseña desvan. Recuerda configurar correctamente la dirección IP de la BBDD.

Puntuación 0,00 sobre
1,00
▼ Marcar pregunta

El programa pedirá una letra, la pasará a mayúsculas y mostrará el código, el nombre y el número de cuenta de los empleados cuyo nombre empieza por esa letra.

La consulta que debemos realizar es:

```
SELECT codigo, nombre, cuenta FROM empleados WHERE nombre LIKE '%'
```

donde código será un número entero, nombre una cadena de caracteres y cuenta otra.

Los datos se mostrarán con el siguiente formato:

```
nombre (codigo): cuenta
```

Un posible resultado será:

```
Letra inicial: a
Alfonso Gutiérrez Lopez (67890): 12563478001234567890
Andrés Morales Martín (64738): 223411541162311563690
```

Solución: [Ejercicio4.java](#)

Respuesta: (penalty regime: 10, 20, ... %)

```
18 |     String consulta = "SELECT codigo, nombre, cuenta FROM empleados WHERE nombre LIKE ?";
19 |     PreparedStatement stat = conn.prepareStatement(consulta);
20 |
21 |     System.out.print(" Letra inicial: ");
22 |     String letra = teclado.next();
23 |     letra = letra.toUpperCase() + "%";
24 |
25 |     stat.setString(1, letra);
26 |     ResultSet rs = stat.executeQuery();
27 |
28 |     /*
29 |      * donde código será un número entero, nombre una cadena de caracteres y cuenta otra.
30 |      * Los datos se mostrarán con el siguiente formato:
31 |      * nombre (codigo): cuenta
32 |     */
33 |
34 |
35 |     while (rs.next()) {
36 |         System.out.println( rs.getString("NOMBRE") + ":" + rs.getInt("CODIGO") + " " + rs.getString("CUENTA"));
37 |     }

```

Expected	Got
<pre>✗ import java.sql.*; import java.util.*; public class Ejercicio4 { public static void main (String args[]) throws ClassNotFoundException, SQLException { // Datos necesarios para establecer la conexión con la BBDD String driver = "oracle.jdbc.driver.OracleDriver"; String url = "jdbc:oracle:thin:@192.168.1.41:1521:xe "; String usuario = "desvan"; String password = "desvan"; // Cargar el driver de la BBDD elegida Class.forName(driver); // Crear la conexión Connection conn = DriverManager.getConnection(url, usuario, password); // Crear y ejecutar una consulta preparada String consulta = "SELECT codigo, nombre, cuenta FROM empleados WHERE nombre LIKE ?"; PreparedStatement stat = conn.prepareStatement(consulta); // Obtenemos el número de hijos, lo introducimos en la consulta y la ejecutamos System.out.print("Letra inicial: "); Scanner leer = new Scanner(System.in); String letra = leer.next(); letra = letra.toUpperCase() + "%"; stat.setString(1, letra); ResultSet rs = stat.executeQuery(); // Procesar los resultados while (rs.next()) { System.out.println(rs.getString("nombre") + " (" + rs.getInt("codigo") + "): " + rs.getString("cuenta")); } // Cerrar conexión conn.close(); } }</pre>	<pre>***Error*** Traceback (most recent call File "prog.python3", line comment = "Expected {1} IndexError: tuple index out</pre>

Your code must pass all tests to earn any marks. Try again.

Question author's solution:

```
import java.sql.*;
import java.util.*;

public class Ejercicio4 {
    public static void main (String args[]) throws ClassNotFoundException, SQLException {

        // Datos necesarios para establecer la conexión con la BBDD
        String driver = "oracle.jdbc.driver.OracleDriver";
        String url = "jdbc:oracle:thin:@192.168.1.41:1521:xe ";
        String usuario = "desvan";
        String password = "desvan";

        // Cargar el driver de la BBDD elegida
        Class.forName(driver);

        // Crear la conexión
        Connection conn = DriverManager.getConnection(url, usuario, password);

        // Crear y ejecutar una consulta preparada
        String consulta = "SELECT codigo, nombre, cuenta FROM empleados WHERE nombre LIKE ?";
        PreparedStatement stat = conn.prepareStatement(consulta);

        // Obtenemos el número de hijos, lo introducimos en la consulta y la ejecutamos
        System.out.print("Letra inicial: ");
        Scanner leer = new Scanner(System.in);
        String letra = leer.next();
        letra = letra.toUpperCase() + "%";
        stat.setString(1, letra);
        ResultSet rs = stat.executeQuery();

        // Procesar los resultados
        while (rs.next()) {
            System.out.println(rs.getString("nombre") + " (" + rs.getInt("codigo") + "): " + rs.getString("cuenta"));
        }

        // Cerrar conexión
        conn.close();
    }
}
```

Incorrecta

Puntos para este envío: 0,00/1,00.

Puntuación 0,00 sobre
1,00

▢ Marcar pregunta

El programa mostrará el código y el nombre de todos los trabajadores en la tabla empleados.

La consulta que debemos realizar es:

```
SELECT codigo, nombre FROM empleados
```

donde código será un número entero y nombre una cadena de caracteres

Los datos se mostrarán con el siguiente formato:

```
codigo(hueco de 10, alineado a la izquierda)nombre
```

El resultado será:

```
11      Juan Ignacio Martinez
1       José Luis Pérez
2341    Fernando Romero Díaz
11223   Manuel López Martín
67890   Alfonso Gutiérrez López
111     Encarna López López
2031    Inés Montero Zafra
9876    Rosa Lorite López
96352   Lola Martínez Contreras
76543   Francisca Colate González
73152   María Pascual Rojo
64738   Andrés Morales Martín
```

Solución: [Ejercicio1.java](#)

Respuesta: (penalty regime: 10, 20, ... %)

```
1 import java.sql.*;           // Trabajar con BBDD
2
3 public class Ejercicio_08 {
4     public static void main (String args[]) throws ClassNotFoundException, SQLException {
5
6     // Datos necesarios para establecer la conexión con la BBDD
7     String driver = "oracle.jdbc.driver.OracleDriver";
8     String url = "jdbc:oracle:thin:@192.168.0.29:1521:xe ";
9     String usuario = "desvan";
10    String password = "desvan";
11
12    Class.forName(driver);
13
14    Connection conn = DriverManager.getConnection(url, usuario, password);
15
16    Statement stmt = conn.createStatement();
17
18    String consulta = "SELECT codigo, nombre FROM empleados";
19    ResultSet rs = stmt.executeQuery(consulta);
20}
```

Expected	Got
<pre>✗ import java.sql.*; // Trabajar con BBDD public class Ejercicio1 { public static void main (String args[]) throws ClassNotFoundException, SQLException { // Datos necesarios para establecer la conexión con la BBDD String driver = "oracle.jdbc.driver.OracleDriver"; String url = "jdbc:oracle:thin:@192.168.1.41:1521:xe "; String usuario = "desvan"; String password = "desvan"; // Crear la conexión Connection conn = DriverManager.getConnection(url, usuario, password); // Crear el objeto Statement, ejecutar consulta y obtener resultados Statement stmt = conn.createStatement(); String consulta = "SELECT codigo, nombre FROM empleados"; ResultSet rs = stmt.executeQuery(consulta); // Procesar resultados while (rs.next()) { // System.out.println(rs.getInt(1) + " " + rs.getString(2)); System.out.printf("%-10s%-n", rs.getInt("codigo"), rs.getString("nombre")); } // Cerrar conexión conn.close(); } }</pre>	<pre>***Error*** Traceback (most recent call last): File "prog.pytton3", line 74, in <module> comment = "Expected {1} lines, got {2}".format(li IndexError: tuple index out of range</pre>

Your code must pass all tests to earn any marks. Try again.

Question author's solution:

```
import java.sql.*;           // Trabajar con BBDD

public class Ejercicio1 {
    public static void main (String args[]) throws ClassNotFoundException, SQLException {

        // Datos necesarios para establecer la conexión con la BBDD
        String driver = "oracle.jdbc.driver.OracleDriver";
        String url = "jdbc:oracle:thin:@192.168.1.41:1521:xe ";
        String usuario = "desvan";
        String password = "desvan";

        // Crear la conexión
        Connection conn = DriverManager.getConnection(url, usuario, password);

        // Crear el objeto Statement, ejecutar consulta y obtener resultados
        Statement stmt = conn.createStatement();

        String consulta = "SELECT codigo, nombre FROM empleados";
        ResultSet rs = stmt.executeQuery(consulta);

        // Procesar resultados
        while (rs.next()) {
            // System.out.println(rs.getInt(1) + " " + rs.getString(2));
            System.out.printf("%-10s%-n", rs.getInt("codigo"), rs.getString("nombre"));
        }

        // Cerrar conexión
        conn.close();
    }
}
```

Incorrecta

Puntos para este envío: 0,00/1,00.

Pregunta 9

Incorrecta

Puntuación 0,00 sobre
1,00

▢ Marcar pregunta

De los ejemplos que hemos estudiado elige el que mejor se adapta a la siguiente consulta, modificalo y obtén los datos solicitados con el formato exigido.

Para ello, se usará el usuario desvan con la contraseña desvan. Recuerda configurar correctamente la dirección IP de la BBDD.

El programa da de alta a un nuevo departamento y luego lo borra. Pide el código y el nombre del nuevo departamento y lo añade. Luego lo borra. Comprobará si la operación se ha realizado bien o no.

Las consultas que debemos realizar son:

```
INSERT INTO departamentos (codigo, nombre) VALUES (?, ?)
DELETE FROM departamentos WHERE codigo = ?
```

donde código es un número entero y nombre una cadena de caracteres.

El resultado será:

```
Código: 7  
Nombre: Sistemas  
INSERT OK  
DELETE OK
```

Solución: Ejercicio7.java [View](#)

Respuesta: (penalty regime: 10, 20, ... %)

```
1 import java.sql.*;  
2 import java.util.*;  
3  
4 public class Ejercicio_09 {  
5     public static void main (String args[]) throws ClassNotFoundException, SQLException {  
6         Scanner tecaldo = new Scanner(System.in);  
7         // Datos necesarios para establecer la conexión con la BBDD  
8         String driver = "oracle.jdbc.driver.OracleDriver";  
9         String url = "jdbc:oracle:thin:@192.168.0.29:1521:xe";  
10        String usuario = "desvan";  
11        String password = "desvan";  
12  
13        // Crear la conexión  
14        Connection conn = DriverManager.getConnection(url, usuario, password);  
15  
16        /*  
17         El programa da de alta a un nuevo departamento y luego lo borra.  
18         Pide el código y el nombre del nuevo departamento y lo añade.  
19         Luego lo borra. Comprobará si la operación se ha realizado bien o no.  
20        */
```

Expected	Got
<pre>✗ import java.sql.*; import java.util.*; public class Ejercicio7 { public static void main (String args[]) throws ClassNotFoundException, SQLException { // Datos necesarios para establecer la conexión con la BBDD String driver = "oracle.jdbc.driver.OracleDriver"; String url = "jdbc:oracle:thin:@192.168.1.41:1521:xe "; String usuario = "desvan"; String password = "desvan"; // Cargar el driver de la BBDD elegida Class.forName(driver); // Crear la conexión Connection conn = DriverManager.getConnection(url, usuario, password); // Insertar datos con una consulta preparada String consulta = "INSERT INTO departamentos (codigo, nombre) VALUES (?, ?)"; PreparedStatement stat = conn.prepareStatement(consulta); // Obtenemos los valores, los introducimos en la consulta y la ejecutamos Scanner leer = new Scanner(System.in); System.out.print("Código: "); int codigo = leer.nextInt(); System.out.print("Nombre: "); String nombre = leer.next(); stat.setInt(1, codigo); stat.setString(2, nombre); if (stat.executeUpdate() == 1) { System.out.println("INSERT OK"); } // Borramos el dato anterior consulta = "DELETE FROM departamentos WHERE codigo = ?"; stat = conn.prepareStatement(consulta); // Obtenemos los valores, los introducimos en la consulta y la ejecutamos stat.setInt(1, codigo); if (stat.executeUpdate() == 1) { System.out.println("DELETE OK"); } // Cerrar conexión conn.close(); } }</pre>	<pre>***Error*** Traceback (most recent call last): File "prog.py", line 102, in <module> comment = "Expected {1} lines, got {2}".format(lin IndexError: tuple index out of range</pre>

Your code must pass all tests to earn any marks. Try again.

Question author's solution:

```
import java.sql.*;  
import java.util.*;  
  
public class Ejercicio7 {  
    public static void main (String args[]) throws ClassNotFoundException, SQLException {  
  
        // Datos necesarios para establecer la conexión con la BBDD  
        String driver = "oracle.jdbc.driver.OracleDriver";  
        String url = "jdbc:oracle:thin:@192.168.1.41:1521:xe ";  
        String usuario = "desvan";  
        String password = "desvan";  
  
        // Cargar el driver de la BBDD elegida  
        Class.forName(driver);  
  
        // Crear la conexión  
        Connection conn = DriverManager.getConnection(url, usuario, password);  
  
        // Insertar datos con una consulta preparada  
        String consulta = "INSERT INTO departamentos (codigo, nombre) VALUES (?, ?)";  
        PreparedStatement stat = conn.prepareStatement(consulta);  
  
        // Obtenemos los valores, los introducimos en la consulta y la ejecutamos  
        Scanner leer = new Scanner(System.in);  
        System.out.print("Código: ");  
        int codigo = leer.nextInt();  
        System.out.print("Nombre: ");  
        String nombre = leer.next();  
        stat.setInt(1, codigo);  
        stat.setString(2, nombre);  
        if (stat.executeUpdate() == 1) {  
            System.out.println("INSERT OK");  
        }  
  
        // Borramos el dato anterior  
        consulta = "DELETE FROM departamentos WHERE codigo = ?";  
        stat = conn.prepareStatement(consulta);  
  
        // Obtenemos los valores, los introducimos en la consulta y la ejecutamos  
        stat.setInt(1, codigo);  
        if (stat.executeUpdate() == 1) {  
            System.out.println("DELETE OK");  
        }  
  
        // Cerrar conexión  
        conn.close();  
    }  
}
```

Pregunta 10
Incorrecta
Puntuación 0,00 sobre
1,00
Marcar pregunta

De los ejemplos que hemos estudiado elige el que mejor se adapta a la siguiente consulta, modificalo y obtén los datos solicitados con el formato exigido.

Para ello, se usará el usuario desvan con la contraseña desvan. Recuerda configurar correctamente la dirección IP de la BBDD.

El programa pedirá el número de hijos y mostrará el nombre y el número de hijos de los empleados con más hijos que los indicados.

La consulta que debemos realizar es:

```
SELECT nombre, hijos FROM empleados WHERE hijos > XX
```

donde nombre es una cadena de caracteres, hijos un número entero y XX el valor que se ha introducido por teclado.

Los datos se mostrarán con el siguiente formato:

nombre: hijos

Un posible resultado será:

```
Número de hijos: 2
Francisca Colate Gonzalez: 3
María Pascual Rojo: 3
Andrés Morales Martín: 3
```

Solución: Ejercicio3.java

Respuesta: (penalty regime: 10, 20, ... %)

```
1 import java.sql.*; // Trabajar con BBDD
2 import java.util.*;
3
4 public class Ejercicio_10 {
5     public static void main (String args[]) throws ClassNotFoundException, SQLException {
6         Scanner teclado = new Scanner(System.in);
7
8         // Datos necesarios para establecer la conexión con la BBDD
9         String driver = "oracle.jdbc.driver.OracleDriver";
10        String url = "jdbc:oracle:thin:@localhost:1521:xe ";
11        String usuario = "desvan";
12        String password = "desvan";
13
14        // Crear la conexión
15        Connection conn = DriverManager.getConnection(url, usuario, password);
16
17        // Crear el objeto Statement, ejecutar consulta y obtener resultados
18        Statement stmt = conn.createStatement();
19
20    }
```

Expected	Got
<pre>✗ import java.sql.*; import java.util.*; public class Ejercicio3 { public static void main (String args[]) throws ClassNotFoundException, SQLException { // Datos necesarios para establecer la conexión con la BBDD String driver = "oracle.jdbc.driver.OracleDriver"; String url = "jdbc:oracle:thin:@192.168.1.41:1521:xe "; String usuario = "desvan"; String password = "desvan"; // Cargar el driver de la BBDD elegida Class.forName(driver); // Crear la conexión Connection conn = DriverManager.getConnection(url, usuario, password); // Crear y ejecutar una consulta preparada String consulta = "SELECT nombre, hijos FROM empleados WHERE hijos > ?"; PreparedStatement stat = conn.prepareStatement(consulta); // Obtenemos el número de hijos, lo introducimos en la consulta y la ejecutamos System.out.print("Número de hijos: "); Scanner leer = new Scanner(System.in); int numHijos = leer.nextInt(); stat.setInt(1, numHijos); ResultSet rs = stat.executeQuery(); // Procesar los resultados while (rs.next()) { System.out.println(rs.getString("nombre") + ": " + rs.getInt("hijos")); } // Cerrar conexión conn.close(); } }</pre>	<pre>***Error *** Traceback (most recent call last): File "prog.py", line 94, in <module> comment = "Expected {1} lines, got {2}".format(lin IndexError: tuple index out of range</pre>

Your code must pass all tests to earn any marks. Try again.

Question author's solution:

```
import java.sql.*;
import java.util.*;

public class Ejercicio3 {
    public static void main (String args[]) throws ClassNotFoundException, SQLException {

        // Datos necesarios para establecer la conexión con la BBDD
        String driver = "oracle.jdbc.driver.OracleDriver";
        String url = "jdbc:oracle:thin:@192.168.1.41:1521:xe ";
        String usuario = "desvan";
        String password = "desvan";

        // Cargar el driver de la BBDD elegida
        Class.forName(driver);

        // Crear la conexión
        Connection conn = DriverManager.getConnection(url, usuario, password);

        // Crear y ejecutar una consulta preparada
        String consulta = "SELECT nombre, hijos FROM empleados WHERE hijos > ?";
        PreparedStatement stat = conn.prepareStatement(consulta);

        // Obtenemos el número de hijos, lo introducimos en la consulta y la ejecutamos
        System.out.print("Número de hijos: ");
        Scanner leer = new Scanner(System.in);
        int numHijos = leer.nextInt();
        stat.setInt(1, numHijos);
        ResultSet rs = stat.executeQuery();

        // Procesar los resultados
        while (rs.next()) {
            System.out.println(rs.getString("nombre") + ": " + rs.getInt("hijos"));
        }

        // Cerrar conexión
        conn.close();
    }
}
```



Contacta con nosotros

- 📍 Dirección: Calle Álava 41, interior - Vitoria-Gasteiz
- 📞 Teléfono : 945 567 953
- ✉️ E-mail: ulhi@ulhi.net
- 🐦 Twitter: [@UrrutikoLH](https://twitter.com/UrrutikoLH)

