

Integrador Java + MySQL + JDBC + JPA

Puntos totales 88/100 ?

Correo *

raul300378@gmail.com

Datos personales

0 de 0 puntos

Correo Electrónico *

El mismo correo con el que Ingresas a tu Aula Virtual (con el que estás registrado en Egg)

raul300378@gmail.com

Nombre y Apellido *

Raul Herrera

DNI *

Escribe tu cédula de identidad sin puntos, ni espacios

26441076



Colecciones

17 de 17 puntos

✓ **Los mapas son colecciones de Llave/Valor. Las llaves son únicas pero los valores sí pueden repetirse.** *2/2

☒ Verdadero



☐ Falso

✓ **El Java Collection Framework es: *** 2/2

☐ Una base de datos

☐ Un conjunto de objetos

☒ Un conjunto de clases e interfaces



✓ **Cómo se define el tamaño de una colección? *** 3/3

☐ Con el método size()

☐ Se define al declarar la colección

☒ Las colecciones son dinámicas, su tamaño depende de la cantidad de objetos que le agreguemos o quitemos ✓



✓ **Un framework es un marco de trabajo el cual contiene un conjunto estandarizado de conceptos, prácticas y criterios para hacer frente a un tipo de problemática particular y resolver nuevos problemas de índole similar.** *2/2

☒ Verdadero



☐ Falso

✓ **Los conjuntos o Sets pueden contener elementos repetidos, simplemente se duplica el valor** *2/2

☒ Falso



☐ Verdadero

✓ **Qué puede almacenar una colección? *** 2/2

☐ Datos primitivos, como char e int

☐ Objetos mezclados sin importar de qué clase son

☒ Objetos de la misma clase



✓ **Las listas son colecciones de objetos ordenados por posición donde los elementos pueden repetirse.** *2/2

☐ Falso

☒ Verdadero



✓ **Cuál de las siguientes herramientas nos facilita recorrer una lista?** *2/2

- ☐ Un Switch con condicionales en cada caso
- ☐ Bucle For
- ☒ Bucle ForEach



Relaciones entre Clases

12 de 12 puntos

✓ **Las relaciones pueden ser *** 2/2

- ☐ Uno a Uno solamente
- ☐ Uno a Uno y Muchos a Muchos
- ☒ Uno a Uno, Uno a Muchos, Muchos a Muchos, Muchos a Uno



✓ **Las relaciones entre clases realmente significan que una clase contiene una referencia a un objeto u objetos, de la otra clase en la forma de un atributo.** *2/2

- ☐ Falso
- ☒ Verdadero



✓ **En una relación de agregación, un objeto depende del otro, no pueden existir individualmente.** *2/2

- ☐ Verdadero
- ☒ Falso



✓ **La composición es un tipo de relación donde un objeto complejo es conformado por objetos más pequeños.**

*2/2

- ☐ Falso
- ☒ Verdadero



✓ **Las clases no actúan aisladas entre sí, al contrario, las clases están relacionadas unas con otras.**

*2/2

- ☒ Verdadero
- ☐ Falso



✓ **Las relaciones son siempre unidireccionales ***

2/2

- ☐ Verdadero
- ☒ Falso



Herencia

14 de 15 puntos

✓ **Cuándo usamos la palabra reservada super? ***

2/2

- ☐ Cuando queremos acceder a los métodos, atributos y constructores de la clase inferior
- ☐ Cuando queremos referirnos a una clase que está muy bien hecha
- ☒ Cuando queremos acceder a los métodos, atributos y constructores de la clase superior



✓ En el siguiente código ¿cuál es la clase padre? *

3/3

```
1 public class Persona extends SerVivo{  
2  
3     String nombre;  
4  
5 }
```

- ☐ public class
- ☐ Java.util
- ☒ SerVivo
- ☐ Persona



✓ Qué es la Super Clase? *

2/2

- ☐ Una clase con muchos métodos
- ☒ Como se llama a la clase superior(Padre o Madre)
- ☐ Una clase con superpoderes



✓ Para qué sirve el modificador Protected? *

2/2

- ☐ Para proteger el código para que no sea borrado
- ☐ Para que suene a que sabemos más que los demás
- ☒ Para que los atributos sean accesibles sólo por las clases que heredan sin usar getters ni setters



✗ La Sub Clase hereda el constructor *

0/1

☒ Verdadero☐ Falso

Respuesta correcta

☒ Falso**✓ Qué hace la anotación @Override? ***

1/1

☒ Permite que la clase hija haga funcionar un método de manera diferente a la clase padre ✓☐ Desbloquea nuevos métodos disponibles en Java☐ Permite que la clase hija elimine métodos que considere innecesarios**✓ La Sub Clase hereda atributos y métodos. ***

1/1

☐ Falso☒ Verdadero**✓ Una clase padre puede heredar a muchas clases hijas. ***

2/2

☒ Verdadero☐ Falso

✓ **Una clase hija puede heredar de muchas clases padres. *** 1/1

☐ Verdadero

☒ Falso



Manejo de Excepciones

12 de 13 puntos

✓ **Una excepción es un evento que ocurre durante la ejecución de un programa que interrumpe el flujo normal de las instrucciones del programa. *** 1/1

☐ Falso

☒ Verdadero



✗ **Para aclarar que un método puede lanzar un error se coloca la palabra "extends". *** 0/1

☐ Falso

☒ Verdadero



Respuesta correcta

☒ Falso

✓ **Sólo se puede usar 1 bloque catch por bloque try. *** 2/2

☐ Verdadero

☒ Falso



✓ **Un objeto Exception contiene información sobre un error ocurrido.** *2/2

- ☐ Falso
- ☒ Verdadero



✓ **Si una excepción se produce en el bloque catch, puede atraparse en el bloque try y manejarla.** *2/2

- ☐ Verdadero
- ☒ Falso



✓ **Cuál es el resultado del siguiente código? siguiendo el ejemplo que venimos viendo** *4/4

```
1 Persona p1 = new Persona();
2 String nombre = null;
3
4 try {
5     if (!nombre.equals(null)) {
6         p1.setNombre(nombre);
7     }
8 } catch (Exception e) {
9     System.out.println("El nombre no puede ser nulo");
10 }
```

- ☐ El nombre de p1 queda vacío
- ☒ Se imprime por consola "El nombre no puede ser nulo"
- ☐ El nombre de p1 es null



✓ **Cuando se lanza una excepción no hay nada más que hacer, termina el programa.** *1/1

☒ Falso



☐ Verdadero

MySQL

8 de 8 puntos

Resultados del ejercicio de los candados

✓ **Candado A *** 1/1

☐ Clave: 13539

☐ Clave: 15935

☐ Clave: 16832

☒ Clave: 14043



✓ **Candado B *** 1/1

☐ Clave: 4830

☐ Clave: 3830

☒ Clave: 3480

☐ Clave: 4380



✓ **Candado A ***

1/1

- ☐ Posición 1
- ☒ Posición 2
- ☐ Posición 3
- ☐ Posición 4



✓ **Candado B ***

1/1

- ☐ Posición 1
- ☐ Posición 2
- ☒ Posición 3
- ☐ Posición 4



✓ **Candado C ***

1/1

- ☒ Posición 1
- ☐ Posición 2
- ☐ Posición 3
- ☐ Posición 4



✓ **Candado D ***

1/1

- ☐ Posición 1
- ☐ Posición 2
- ☐ Posición 3
- ☒ Posición 4



✓ **Candado C ***

1/1

- ☒ Clave: 631
- ☐ Clave: 963
- ☐ Clave: 145
- ☐ Clave: 601



✓ **Candado D ***

1/1

- ☒ Clave: 191
- ☐ Clave: 1063
- ☐ Clave: 153
- ☐ Clave: 101

**JDBC**

14 de 20 puntos



✓ ¿Cuál es la forma correcta de manejar excepciones en JDBC? * 2/2

- ☐ B) Lanzar una nueva excepción personalizada en cada método de JDBC.
- ☐ A) Ignorar las excepciones y continuar con la ejecución del programa.
- ☒ C) Capturar excepciones específicas de JDBC y manejarlas adecuadamente. ✓
- ☐ D) Utilizar el bloque finally para liberar recursos, sin importar si se produce una excepción o no.

✗ ¿Cuál es la forma correcta de cerrar una conexión JDBC correctamente? *0/2

- ☐ B) Llamar al método close() en el objeto ResultSet.
- ☒ D) Llamar al método close() en todos los objetos anteriores. ✗
- ☐ C) Llamar al método close() en el objeto Connection.
- ☐ A) Llamar al método close() en el objeto Statement.

Respuesta correcta

- ☒ C) Llamar al método close() en el objeto Connection.

✓ ¿Cuál de las siguientes interfaces de JDBC se utiliza para ejecutar *2/2 consultas parametrizadas?

- ☒ D) PreparedStatement. ✓
- ☐ C) Connection.
- ☐ A) Statement.
- ☐ B) ResultSet.



✓ ¿Cuál es el propósito del objeto Connection en JDBC? *

2/2

- ☒ C) Representar una conexión física a una base de datos. ✓
- ☐ D) Proporcionar métodos para obtener metadatos de la base de datos.
- ☐ B) Almacenar los resultados de una consulta.
- ☐ A) Ejecutar consultas SQL.

✗ ¿Cuál es la forma más eficiente de realizar múltiples inserciones en una base de datos utilizando JDBC? *0/2

- ☐ A) Ejecutar una consulta INSERT para cada fila a insertar.
- ☒ D) Utilizar el objeto BatchStatement para ejecutar consultas en lote. ✗
- ☐ C) Utilizar un objeto PreparedStatement y una transacción.
- ☐ B) Utilizar un bucle y ejecutar consultas INSERT individuales.

Respuesta correcta

- ☒ C) Utilizar un objeto PreparedStatement y una transacción.

✓ ¿Cuál de las siguientes declaraciones es cierta acerca de los controladores (drivers) JDBC?

*2/2

- ☒ D) Los controladores JDBC son interfaces utilizadas por la aplicación para interactuar con una base de datos. ✓
- ☐ C) Los controladores JDBC son necesarios solo para la conexión inicial a la base de datos.
- ☐ B) Los controladores JDBC son proporcionados por el servidor de la base de datos.
- ☐ A) Los controladores JDBC son responsables de la administración de transacciones.



✓ ¿Cuál es el propósito del objeto ResultSet en JDBC?

2/2

- ☐ D) Define los parámetros de una consulta parametrizada.
- ☐ C) Ejecuta una consulta SQL.
- ☐ A) Representa una conexión a una base de datos.
- ☒ B) Almacena el resultado de una consulta SQL.



✓ ¿Cuál de las siguientes afirmaciones es cierta acerca de las transacciones en JDBC?

*2/2

- ☐ D) Las transacciones solo son necesarias cuando se utilizan controladores de base de datos específicos.
- ☐ A) Las transacciones solo se utilizan para consultas SELECT.
- ☒ C) Las transacciones permiten agrupar múltiples consultas en una única operación atómica.
- ☐ B) Las transacciones solo se utilizan para consultas de modificación de datos (INSERT, UPDATE, DELETE).



✗ ¿Cuál es la forma más segura de ejecutar consultas SQL en JDBC para prevenir ataques de inyección de SQL? *0/2

- ☐ C) Utilizar el objeto PreparedStatement con parámetros vinculados.
- ☐ B) Utilizar el objeto Statement para ejecutar consultas sin parámetros.
- ☒ D) Utilizar el objeto ResultSet para obtener resultados de consultas.
- ☐ A) Concatenar los parámetros de consulta directamente en la cadena SQL.



Respuesta correcta

- ☒ C) Utilizar el objeto PreparedStatement con parámetros vinculados.



✓ ¿Cuál es el propósito principal de JDBC? *

2/2

- ☐ D) Realizar operaciones CRUD en una base de datos.
- ☐ A) Crear consultas SQL.
- ☒ B) Establecer la conexión con una base de datos. ✓
- ☐ C) Mapear objetos a tablas de base de datos.

JPA

11 de 15 puntos

✗ ¿Cuál de las siguientes opciones describe mejor el concepto de "carga diferida" (lazy loading) en JPA? *0/2

- ☒ C) Cargar solo las propiedades necesarias de una entidad cuando se accede a ella. ✗
- ☐ D) Cargar todas las propiedades de una entidad de forma anticipada.
- ☐ A) Cargar solo las entidades relacionadas necesarias cuando se accede a ellas.
- ☐ B) Cargar todas las entidades relacionadas de forma anticipada.

Respuesta correcta

- ☒ A) Cargar solo las entidades relacionadas necesarias cuando se accede a ellas.

✓ ¿Cuál de las siguientes anotaciones se utiliza para marcar una propiedad como una clave primaria en JPA? *2/2

- ☐ D) @Column
- ☒ A) @Id ✓
- ☐ B) @PrimaryKey
- ☐ C) @GeneratedValue



✓ ¿Cuál de las siguientes anotaciones se utiliza para establecer una relación de uno a muchos en JPA? *2/2

- ☐ D) @ManyToMany
- ☐ C) @OneToOne
- ☒ A) @OneToMany
- ☐ B) @ManyToOne



✓ ¿Cuál de las siguientes afirmaciones describe mejor JPA (Java Persistence API)? *3/3

- ☐ D) Es un framework para el desarrollo de interfaces de usuario en Java.
- ☐ B) Es una biblioteca para el acceso a bases de datos NoSQL.
- ☒ C) Es una especificación de Java para el mapeo objeto-relacional.
- ☐ A) Es un lenguaje de consulta para bases de datos relacionales.



✓ ¿Cuál de las siguientes anotaciones se utiliza para marcar una clase como una entidad en JPA? *2/2

- ☒ A) @Entity
- ☐ D) @OneToMany
- ☐ B) @Table
- ☐ C) @PersistenceContext



✓ ¿Cuál de las siguientes opciones describe mejor el concepto de "cascada" (cascade) en JPA? *2/2

- ☐ A) La forma de almacenar entidades en cascada utilizando JPA.
- ☐ D) La forma de cargar entidades relacionadas en cascada utilizando JPA.
- ☒ B) La forma en que se propagan las operaciones en cascada a entidades relacionadas. ✓
- ☐ C) La forma de generar automáticamente identificadores únicos para las entidades.

✗ ¿Cuál de las siguientes opciones describe mejor la diferencia entre CascadeType.PERSIST y CascadeType.MERGE en JPA? *0/2

- ☒ B) CascadeType.PERSIST guarda todas las entidades relacionadas, mientras que CascadeType.MERGE guarda solo las entidades modificadas. ✗
- ☐ A) CascadeType.PERSIST realiza una operación de inserción, mientras que CascadeType.MERGE realiza una operación de actualización.
- ☐ C) CascadeType.PERSIST realiza una operación de actualización, mientras que CascadeType.MERGE realiza una operación de inserción.
- ☐ D) CascadeType.PERSIST guarda solo las entidades modificadas, mientras que CascadeType.MERGE guarda todas las entidades relacionadas.

Respuesta correcta

- ☒ A) CascadeType.PERSIST realiza una operación de inserción, mientras que CascadeType.MERGE realiza una operación de actualización.

Este formulario se creó en Egg Cooperation.

Google Formularios





