

# Integrador Java + MySQL

Total de puntos 87/100 ?

## Correo electrónico \*

juanignaciosalias@gmail.com

0 de 0 puntos

## DNI \*

Escribe tu cédula de identidad sin puntos, ni espacios

28608779

## Nombre y Apellido \*

Juan Ignacio Salias

## Correo Electrónico \*

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

juanignaciosalias@gmail.com

**Java Introducción**

11 de 11 puntos



✓ **Tipos de datos en Java: \***

- ☐ int, double, boolean, String, void, null
- ☒ byte, short, int, long, float, double, boolean, char, String
- ☐ Cadena, entero, real, lógico

✓ **Cuál es la salida del siguiente código? \***

```
1  int num1 = 6;  
2  int num2 = 8;  
3  
4  System.out.println(num1==num2);
```

- ☐ 6
- ☐ 8
- ☒ False
- ☐ True

✓ **Cual de los siguientes métodos NO es parte de la clase String?**

- ☐ .toUpperCase()
- ☐ .isEmpty()
- ☐ concat()
- ☒ .stringLength()



✓ El método main puede estar o no en un proyecto Java funcional.

- ☒ Falso
- ☐ Verdadero

✓ Qué hace el siguiente código? \*

```
1  int aux = 0;  
2  aux++;
```

- ☐ Agrega una nueva variable aux
- ☐ Multiplica por 2 a la variable aux
- ☒ Incrementa en 1 a la variable aux

✓ Las variables siempre deben especificar de qué tipo son al ser declaradas.

- ☐ Falso
- ☒ Verdadero

POO

24 de 24 puntos

✓ **Qué son los atributos? \***

- ☐ Funciones que el objeto puede realizar
- ☐ Métodos de consulta o modificación de un objeto
- ☒ Variables que guardan información sobre el objeto

✓ **Cuáles son los modificadores de acceso? \***

- ☐ Public, static, abstract
- ☒ Public, private, protected
- ☐ Public, notPublic

✓ **Cuál es la salida de este código? \***

```
1  Persona p1 = new Persona();
2  Persona p2 = new Persona();
3
4  p1.setNombre("Albus Dumbledore");
5  p2.setNombre(p1.getNombre());
6
7  System.out.println(p1.equals(p2));
8  System.out.println(p1.getNombre().equals(p2.getNombre()));
```

- ☒ False, True
- ☐ True, True
- ☐ True, False
- ☐ False, False

✓ Qué hace el siguiente código? \*



```
1  Persona p1 = new Persona();
```

- ☒ Instancia un objeto de tipo Persona llamado p1
- ☐ Duplica un objeto persona
- ☐ Crea una nueva clase llamada Persona

✓ Cómo accedo a los atributos de un objeto? \*

- ☐ Con el símbolo ! luego del nombre del objeto
- ☐ Con el símbolo \_ luego del nombre del objeto
- ☒ Con el símbolo . luego del nombre del objeto

✓ Qué significa POO? \*

- ☐ Programación Objetiva Orientada
- ☐ Programación, Observación y Orientación
- ☒ Programación Orientada a Objetos



✓ **Cuál es la diferencia entre clases y objetos? \***

- ☐ Los objetos son abstracciones del mundo real, las clases son de lunes a jueves
- ☒ Las clases son “moldes” y los objetos son modelados a partir de esos moldes, también llamados “instancias”
- ☐ Las clases y los objetos son los mismo

✓ **Qué es un constructor? \***

- ☐ Un método void
- ☐ Un método que nos define qué atributos tiene la clase
- ☒ Es un método que nos permite instanciar un objeto de la clase donde se encuentra

✓ **¿De que se encarga Clase servicio o control? \***

- ☒ Se encarga de crear y modificar objetos, teniendo cada clase de entidad una clase control.
- ☐ Se encarga de recibir los datos que ingresa el usuario por teclado
- ☐ Se encarga de controlar que el programa no consuma memoria al instanciar objeto

### ✓ Qué es el encapsulamiento? \*

- ☐ Hace referencia a los métodos constructores con y sin parámetros
- ☒ Hace referencia a crear métodos para obtener y modificar información de un objeto, manteniendo oculto lo que hace un objeto por dentro.
- ☐ Hace referencia a separar el código en "cápsulas" para mejor lectura del mismo.

### ✓Cuál de estos servicios trabaja de la manera correcta? \*

```
1 public void conocerNombre(Persona x){
2     System.out.println("La persona se llama " + x.getNombre());
3 }
4
5 public void conocerNombre(String nombre){
6     System.out.println("La persona se llama " + nombre);
7 }
```

- ☐ El que obtiene el nombre desde un String
- ☒ El que obtiene el nombre a través de un objeto Persona

## Colecciones

11 de 17 preguntas

### ✗ Qué puede almacenar una colección? \*

- ☒ Objetos mezclados sin importar de qué clase son
- ☐ Objetos de la misma clase
- ☐ Datos primitivos, como char e int

#### Respuesta correcta

- ☒ Objetos de la misma clase

✗ **Los conjuntos o Sets pueden contener elementos repetidos, simplemente se duplica el valor**

- ☐ Falso
- ☒ Verdadero

Respuesta correcta

- ☒ Falso

✓ **Cómo se define el tamaño de una colección? \***

- ☒ Las colecciones son dinámicas, su tamaño depende de la cantidad de objetos que le agreguemos o quitemos
- ☐ Se define al declarar la colección
- ☐ Con el método size()

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

- ☐ Falso
- ☒ Verdadero

✓ **El Java Collection Framework es: \***

- ☐ Un conjunto de objetos
- ☒ Un conjunto de clases e interfaz
- ☐ Una base de datos





✓ **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.**

- ☒ Verdadero
- ☐ Falso

✗ **Cuál de las siguientes herramientas nos facilita recorrer una lista?**

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

Respuesta correcta

- ☒ Bucle ForEach

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

- ☐ Falso
- ☒ Verdadero

**Relaciones entre Clases**

10 de 12 preguntas

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

- ☒ Falso
- ☐ Verdadero

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

- ☒ Verdadero
- ☐ Falso

✓ **Las relaciones pueden ser \***

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

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

- ☒ Verdadero
- ☐ Falso



✗ 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.

- ☒ Falso
- ☐ Verdadero

Respuesta correcta

- ☒ Verdadero

✓ Las relaciones son siempre unidireccionales \*

- ☒ Falso
- ☐ Verdadero

## Herencia

12 de 15 puntos

✓ Cuándo usamos la palabra reservada super? \*

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

✓ **Qué hace la anotación @Override? \***

- ☐ Desbloquea nuevos métodos disponibles en Java
- ☐ Permite que la clase hija elimine métodos que considere innecesarios
- ☒ Permite que la clase hija haga funcionar un método de manera diferente a la clase padre

✓ **Qué es la Super Clase? \***

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

✗ **Una clase hija puede heredar de muchas clases padres. \***

- ☒ Verdadero
- ☐ Falso

Respuesta correcta

- ☒ Falso



✗ **Una clase padre puede heredar a muchas clases hijas. \***

☐ Verdadero

☒ Falso

Respuesta correcta

☒ Verdadero

✓ **La Sub Clase hereda el constructor \***

☐ Verdadero

☒ Falso

✓ **Para qué sirve el modificador Protected? \***

☐ Para que suene a que sabemos más que los demás

☐ Para proteger el código para que no sea borrado

☒ Para que los atributos sean accesibles sólo por las clases que heredan sin usar getters ni setters

✓ **La Sub Clase hereda atributos y métodos. \***

☒ Verdadero

☐ Falso



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

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

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

## Manejo de Excepciones

11 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.

- ☒ Verdadero
- ☐ Falso

✗ **Sólo se puede usar 1 bloque catch por bloque try. \***

☒ Verdadero

☐ Falso

Respuesta correcta

☒ Falso

✓ **Para aclarar que un método puede lanzar un error se coloca la palabra “extends”.**

☒ Falso

☐ Verdadero

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

☒ Verdadero

☐ Falso



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

```
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 }
```

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

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

- ☐ Verdadero
- ☒ Falso

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

- ☒ Falso
- ☐ Verdadero

MySQL

8 de 8 preguntas



## Resultados del ejercicio de los candados

✓ **Candado A \***

- ☐ Clave: 13539
- ☐ Clave: 15935
- ☒ Clave: 14043
- ☐ Clave: 16832

✓ **Candado C \***

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

✓ **Candado A \***

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



✓ **Candado C \***

☒ Posición 1

☐ Posición 2

☐ Posición 3

☐ Posición 4

✓ **Candado B \***

- ☐ Clave: 3830
- ☐ Clave: 4380
- ☒ Clave: 3480
- ☐ Clave: 4830

✓ **Candado B \***

☐ Posición 1

☐ Posición 2

☒ Posición 3

☐ Posición 4



✓ **Candado D \***

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



✓ **Candado D \***

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



El formulario se creó en Egg Cooperation.

Google Formularios







