

## Recuperatorio Examen Integrador Final MISIONES

### Reperatorio Examen Integrador Teórico y Practico Final

En este formulario van a encontrar 20 preguntas teóricas para responder y al final van a encontrar pedazos de código que deberemos interpretar y responder de la manera correcta

## PROGRAMACION MISIONES

### 1) ¿Qué palabra se usa en Java para instanciar un objeto de una clase? 4 puntos

1. new
2. create
3. Cualquiera de las anteriores
4. Ninguna de las anteriores

### 2) El Java Collection Framework es: 4 puntos

- a) Un grupo de objetos.
- b) Un grupo de clases e interfaces
- c) Un grupo de procedimientos.
- d) Ninguna de las anteriores.

### 3) En base a la siguiente imagen, elegir la opción que describe la relación:4 puntos

#### IMAGEN NO ENCONTRADA

1. Auto usa un Motor
2. Motor tiene un Auto
3. Auto tiene un Motor
4. Motor usa un Auto

**4) Una clase que termina la cadena de una herencia se la declarara como: 4 punto**

1. final
2. abstract
3. public
4. Ninguna de las anteriores

**5) ¿Cuál de estas características de las interfaces es verdadera? 4 puntos**

1. Las clases la implementan mediante la palabra implements
2. Todos sus métodos son abstractos (métodos sin cuerpo)
3. Las variables declaradas dentro de una interfaz deben ser constantes
4. Todas las anteriores son verdaderas

**6) Cuando queremos informar que un método puede tirar una excepción usamos el bloque: 4 puntos**

- a) Finally
- b) Throws
- c) Try Catch
- d) Throw

**7) Una sentencia SELECT sin la cláusula WHERE devuelve : 4 puntos**

- a) Todos los registros existentes en la tabla que no estén relacionados con otra tabla
- b) Todos los registros existentes en la tabla
- c) No se puede ejecutar una sentencia SELECT sin la cláusula WHERE
- d) Las anteriores respuestas no son correctas

**8) El ORM nos permite mapear objetos como tablas de MySQL mediante: 8 puntos**

- a) Consultas SQL
- b) Interfaces
- c) Métodos
- d) Anotaciones

9) ¿Qué etiqueta se usa para denominar cada fila en una tabla html? 4 puntos

- a) <tr>
- b) <td>
- c) <th>
- d) <li>

10) Spring Framework es: 10 puntos

- a) Un framework para el desarrollo de aplicaciones PHP
- b) Un framework para el desarrollo de aplicaciones .net
- c) Un framework para el desarrollo de aplicaciones Java
- d) Todas las anteriores

**Interpretación Java**

**Interpretación de código en Java**

1-¿Que hacen los siguientes For con la matriz? 10 puntos

```
for (int i = 0; i < dim; i++) {  
    for (int j = 0; j < dim; j++) {  
        matriz2[i][j]=matriz[j][i];  
    }  
}
```

- a. Suma las dos matrices
- b. Asigna una matriz en otra matriz
- c. Valida si las matrices son iguales
- d. Elimina ambas matrices

## 2-¿Qué nos permite hacer este método de JPA?10 puntos

```
public void actualizar(Integer documento, String nombre) {  
    EntityManager em = Persistence.createEntityManagerFactory("CursoPU").createEntityManager();  
    em.getTransaction().begin();  
  
    Alumno alumno = em.find(Alumno.class, documento);  
  
    alumno.setNombre(nombre);  
  
    em.merge(alumno);  
    em.getTransaction().commit();  
}
```

- a. Guardar un alumno
- b. Eliminar un alumno
- c. Editar un alumno
- d. Ninguna de las anteriores

## 3-Que vinculación hay a continuación? 10 puntos

```
public class Humano extends Animal implements SerVivo{}
```

- a. Animal, es hija de Humano y de SerVivo
- b. SerVivo, es madre de Humano y Animal
- c. Animal es una interface
- d. Ninguna de las anteriores

## Interpretación MySQL

### Interpretación de consultas MySQL

#### ¿La siguiente query nativa de MySQL, que devuelve? 10 puntos

```
select * from producto where codigo_fabricante IN (1,3,5);
```

- b. Todos los elementos de la Tabla producto donde el codigo\_fabricante sea 1, 3 y 5
- a. Todos los elementos de la Tabla producto
- b. Todos los elementos de la Tabla producto donde el codigo\_fabricante no sea 1, 3 y 5
- d. Ninguna de las anteriores

## Interpretación Spring

### Interpretación de codigo Spring

1- ¿Para qué podemos usar este método? 10 puntos

```
@GetMapping("...")
public String ... (ModelMap modelo) {
    List<Autor> autores = autorRepositorio.findAll();
    modelo.put("autores", autores);
    return "autor/...";
}
```

1. Guardar un autor
2. Validar autores
3. Mostrar todos los autores
4. Buscar los autores a guardar