# Examen diagnóstico Analista – Programador (Java – Web)

**Contesta únicamente las preguntas de los temas que conozcas:**

Java

1. Define qué es una clase, un objeto y cuál es la diferencia entre ellos.

R: una clase es una plantilla que define características y comportamientos de un objeto, es decir, define atributos y métodos.

Un objeto es una instancia de una clase, el cual, almacena en memoria sus atributos y métodos específicos distintos a otros objetos y, por ende, pueden interactuar enviando y recibiendo información.

La diferencia entre estos dos es que una clase es una plantilla genérica mientras que el otro, es una instancia concreta de una clase, con sus propios valores de atributos y métodos.

1. Explica que es una interfaz en Java, una clase abstracta y cuál es la diferencia entre éstas.

R: Una interfaz es una colección de métodos abstractos que pueden ser implementados por cualquier clase mediante la herencia.

Una clase abstracta es una clase que no se puede instanciar y son utilizadas para crear subclases con métodos abstractos y concretos, además de que cuentan con atributos y constructores.

La diferencia entre estas dos se centra en que la interfaz solo puede contener métodos abstractos mientras que la clase abstracta contiene métodos abstractos, métodos concretos, atributos y constructores.

1. ¿Para qué sirve un constructor?

R: Se utiliza para inicializar los objetos creados a partir de una clase (con su mismo nombre) pues se encarga de establecer valores iniciales en sus atributos.

1. ¿Qué son los annotations y cuáles conoces?

R: Las annotations proporcionan información adicional del programa, como información de depuración, documentación, restricciones de seguridad, etc.

* @Override
* @Deprecated
* @Test
* @GetMapping
* @PostMapping
* @PutMapping
* @DeleteMapping
* @Repository
* RequestBody
* @Service
* @Table
* @Id

1. Dado un objeto ArrayList “lista” con elementos de tipo String, escribe el ciclo *for each* que itere la lista e imprima sus elementos.

R:

import java.util.ArrayList;

import java.util.Arrays;

public class Main {

public static void main(String[] args) {

ArrayList<String> lista = new ArrayList<String>();

lista.add("elemento1");

lista.add("elemento2");

lista.add("elemento3");

for (String elemento : lista) {

System.out.println(elemento);

}

}

}

1. Realizar el código Java que ordene de menor a mayor los elementos del siguiente arreglo de enteros:

int []arr = {5, 8, 1, 9, 4};

R:

public class Main {

public static void main(String[] args) {

int[] arr = {5, 8, 1, 9, 4};

for (int i = 0; i < arr.length - 1; i++) {

for (int j = 0; j < arr.length - 1 - i; j++) {

if (arr[j] > arr[j + 1]) {

int temp = arr[j];

arr[j] = arr[j + 1];

arr[j + 1] = temp;

}

}

}

for (int i = 0; i < arr.length; i++) {

System.out.print(arr[i] + " ");

}

}

}

1. Si A es una interfaz y B es una clase ¿Es correcta la siguiente expresión? Explica.

public class C extends A implements B{}

R: No es correcta ya que la clase “C” esta intentando heredar de la interfaz “A” y, a su vez, implementar la clase “B”, lo cuál no es possible ya que “B” es una clase y no una interfaz. Si recordamos, una clase solo puede heredar de una clase base y puede implementar multiples interfaces.

1. Escriba el código que genere una pirámide de 1 a 50, formada de la siguiente forma:

1

12

123

1234

12345

123456

1234567

12345678

123456789

12345678910

R:

public class Main {

public static void main(String[] args) {

int n = 1;

for (int i = 1; i <= 10; i++) {

for (int j = 1; j <= i; j++) {

System.out.print(n);

n++;

}

System.out.println();

}

}

}

Web

1. Escribe el HTML de una tabla como la que se muestra a continuación

|  |  |
| --- | --- |
| A | C |
| B |

R:

<table>

<tr>

<td>A</td>

<td>C</td>

</tr>

<tr>

<td>B</td>

<td></td>

</tr>

</table>

Table se ocupa para crear la tabla, tr para definir filas y td para designar celdas.

1. Realiza una función javascript que valide que el valor de un combo no esté vacío.

R:

function validarCombo(idSelect) {

var select = document.getElementById(idSelect);

if (select.value == "") {

alert("Debe seleccionar una opción en el combo.");

return false;

}

return true;

}

1. Escribe el estilo (CSS) para que todas las tablas de un documento tengan un fondo rojo.

R:

table {

background-color: red;

}

1. ¿Qué es JSON?

R: Es un formato de intercambio de datos que representa información estructurada en forma de objetos y arrays.

JEE

1. ¿Qué es un Servlet y un JSP?

R: El Servlet es un componente que se ejecuta en un servidor web para procesar solicitudes HTTP de clientes.

El JSP permite a los desarrolladores crear páginas web dinámicas combinando HTML y Java.

Podemos decir que el Servlet procesa las solicitudes HTTP y el JSP es una página web dinámica que se procesa en el servidor y se convierte en Servlet para generar algún contenido en formato HTTP.

1. ¿Qué es el modelo MVC?

R: El MVC (Modelo, Vista, Controlador) es un patrón de arquitectura de software que se utiliza para desarrollar aplicaciones con separación de responsabilidades y estructura modular.

* Modelo, representa datos y lógica del negocio
* Vista, representa la interfaz de usuario
* Controlador, representa la interacción entre modelo y vista

Web Services

1. ¿Qué es un web service?

R: Un servicio web es una tecnología de comunicación entre aplicaciones que permite su interacción a través de la red y, se basa, en estándares XML, SOAP, HTTP, etc., para proporcionar una interfaz de API´s.

1. ¿Qué es un WSDL y para qué sirve?

R: El WSDL es un lenguaje basado en XML que ayuda a describir servicios web y su funcionalidad pues dicta como acceder a un servicio web y define los métodos, parámetros y formatos de datos utilizados para su comunicación.

JDBC

1. ¿Cuáles son las desventajas de utilizar JDBC puro?

R: Existe complejidad en el código, le falta portabilidad, se dificulta la gestión de conexiones, se dificulta el manejo de excepciones, limita el manejo de objetos complejos, dificulta en la gestión de transacciones, etc.

1. ¿En qué casos es mejor utilizar un Statement y en qué casos un PreparedStatement en JDBC?

R: Depende del escenario de uso y de los requisitos de la aplicación. Por ejemplo, Statement es adecuado para consultas simples que no contienen parámetros y que se ejecutan pocas veces. PreparedStatement es adecuado para consultas dinámicas con parámetros que se ejecutan muchas veces

1. ¿Qué mecanismo se puede emplear para optimizar el uso de las conexiones a la base de datos en una arquitectura J2EE?

R: En esta arquitectura se puede emplear un pool de conexiones pues se crean una sola vez y se reutilizan en múltiples solicitudes de aplicación lo que mejora el rendimiento y la escalabilidad de la aplicación.

BD

1. Dadas las siguientes tablas:

|  |
| --- |
| DEPTO |
| NUM\_DEPTO |
| DEPARTAMENTO |

|  |
| --- |
| EMPLEADO |
| NUM\_EMP |
| NOMBRE |
| NUM\_DEPTO |

¿Qué campos deberían ser llaves primarias y cuáles llaves foráneas?

R: En la tabla DEPTO, NUM\_DEPTO seria llave primaria. En la tabla EMPLEADO, NUM\_EMP debería ser la llave primaria.

En la tabla EMPLEADO, NUM\_DEPTO sería una llave foránea que haga referencia al campo NUM\_DEPTO de la tabla DEPTO, lo que permitiría establecer una relación entre las tablas.

1. Escribe un query de SQL para obtener los nombres de los departamentos 3, 7,10.

R:

SELECT DEPARTAMENTO

FROM DEPTO

WHERE NUM\_DEPTO IN (3, 7, 10);

1. Escribe un query para mostrar el nombre de TODOS los empleados y la descripción de su departamento, aun cuando no tengan un departamento asociado.

R:

SELECT E.NOMBRE, D.DEPARTAMENTO

FROM EMPLEADO E

LEFT JOIN DEPTO D

ON E.NUM\_DEPTO = D.NUM\_DEPTO;

1. Si queremos que un empleado pueda pertenecer a más de un departamento. ¿Qué se requiere?

R: Se requiere una relación muchos a muchos entre las dos tablas utilizando las llaves primarias de ambas como claves foráneas en la tabla de relación creada.

1. ¿Qué diferencia hay entre un “inner join” y un “outer join”?

R: INNER JOIN es un comando que combina solo los registros que tienen una clave correspondiente entre dos tablas. OUTER JOIN es un comando que devuelve todos los registros de una o mas tablas que se están uniendo sin ser necesaria las coincidencias en las claves de unión.

FRAMEWORKS

1. Escribe qué entiendes por Inversión de Control e Inyección de dependencias, ¿cuáles son sus ventajas y cómo se aplica en el framework Spring?

R: La inversión de control es el proceso de desacoplar componentes de una aplicación y la inyección de dependencias es el proceso de proporcionar a un objeto las dependencias necesarias para que realice su trabajo.

1. Explica cómo encaja cada uno de los elementos del patrón MVC con los frameworks JSF-Spring-Hibernate.

R: Para el Modelo, Hibernate permite mapear objetos Java a tablas de bases de datos y proporciona herramientas para realizar operaciones de lectura, escritura y consulta.

Para la Vista, JSF es un framework que permite crear interfaces de usuario mediante componentes visuales personalizables y proporciona herramientas para la gestión de eventos y validación de datos.

Para el controlador, Spring permite la gestión de peticiones de usuario y actualiza el modelo y la vista, además proporciona herramientas para la inyección de dependencias, la gestión de transacciones y la configuración de aspectos.

1. Escribe el código básico, con todos los elementos del patrón MVC, para generar una aplicación web para la vista de la tabla Empleado (del punto 23) utilizando los frameworks JSF-Spring-Hibernate.