**Prueba técnica Software Developer**

**SECCIÓN 1: CONOCIMIENTO EN API´s**

**Marque con una x la(s) respuesta(s) correcta(s):**

1. En REST, Cuál de los siguientes Métodos de Peticiones HTTP es usado cuando se requiere actualizar información al servidor.

Por ejemplo, un archivo de actualización, información de formulario:

1. GET
2. POST
3. PUT
4. PATCH

2.

public class MyOtherJson {}

public class MyJson {

  String myString = "my string";

  int myInt = 5;

  double[] myArrayOfDoubles = new double[] { 3.14, 2.72 };

  MyOtherJson objectInObject = new MyOtherJson();

}

La anterior representación es en formato JSON?:

1. Si
2. No

Si su respuesta fue “No”, indique la forma correcta de json para el objeto.

{

        "descripcion": "My Json",

        "estado\_Producto": "Activo",

        "fecha\_Fabricacion": "2022-11-11",

        "fecha\_Validez": "2023-11-11",

        "codigo\_Proveedor": 1

}

**SECCIÓN 2: DISEÑO DE FORMULARIOS WEB:**

Se requiere crear una aplicación WEB utilizando tecnología MVC con las siguientes instrucciones.

* Crear una solución en blanco.
* Agregar un formulario WEB con el siguiente diseño.  (Competencia – Diseño interfaces)

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

* Los tipos de contacto de la lista son los siguientes
* Contacto Comercial
* Pago de factura
* Representante legal
* Retiro de mercadería

(Competencia – Manejo HTML)

* El sistema debe generar una contraseña de forma automática y aleatoria, la contraseña debe seguir el siguiente patrón
* Longitud 8 caracteres
* 4 caracteres alfabéticos en Mayúscula aleatorios
* 4 números aleatorios

se deben realizar validaciones del servidor y lado del cliente (Competencia – Manejo de JavaScript)

* Código de cliente:  xmxwebdemo2, solo lectura
* Usuario: debe iniciar con las letras XMX, longitud 6 caracteres.
* Nombre: mínimo 5 – máximo 15 caracteres.
* Cargo: mínimo 5 -máximo 10 caracteres
* Teléfono: debe mostrar como marca de agua el código +57, la longitud del número que digita el usuario debe ser de 7 caracteres. En la DB se deben almacenar los 10 caracteres.
* Correo: validar formato de correo.

**SECCIÓN 3: BASE DE DATOS**

Diagrama

Descripción generada automáticamente

1. Con el anterior MER elabore el query que permita:

1. Consultar el nombre y salario del empleado con mayor salario.

select employees.first\_name, salaries.salary from employees

inner join salaries on salaries.emp\_no = employees.emp\_no

where salaries.salary = (Select Max(salaries.salary) from salaries) order by employees.first\_name;

1. Consultar los departamentos y para cada uno incluir la suma de los salarios de los empleados de ese departamento.

select departments.dept\_name , SUM(salaries.salary) as Suma\_Salario

from salaries

inner join employees on employees.emp\_no = salaries.emp\_no

inner join dept\_manager on dept\_manager.emp\_no = employees.emp\_no

inner join departments on departments.dept\_no = dept\_manager.dept\_no

group by departments.dept\_name;

1. Consultar para cada empleado, todos sus datos y además el número de empleados que han nacido el mismo día.
2. Consultar para cada empleado, todos sus datos, el número de empleados que han nacido el mismo día, una columna con el número de empleados que nacieron el día antes y otra con los que nacieron el día después.

**SECCIÓN 4: PATRONES DE DISEÑO Y BUENAS PRACTICAS**

1. Cuál es el patrón de diseño que permite asegurar que una clase tenga una única instancia.

1. Singleton
2. Prototype
3. Abstract Factory
4. observer

1. En el patrón MVC ¿Qué capa se encarga de recibir y enrutar las peticiones del cliente?

1. Modelo
2. Vista
3. Controlador
4. Interfaces
5. Que son los principios SOLID, explique en sus propias palabras uno de ellos.

Rpt:

Los Principios Solid, son guías de código que se pueden aplicar dentro de cada código de una aplicación a desarrollar, ya sea a nivel de backend o Fronted.

Son 5 Principios:

S – Single Responsibility Principle (SRP)

En este principio una clase debería tener una, y solo una, razón para cambiar.

O – Open/Closed Principle (OCP)

En este principio se deberías ser capaz de extender el comportamiento de una clase, sin modificarla.

L – Liskov Substitution Principle (LSP)

En este principio las clases derivadas deben poder sustituirse por sus clases base.

I – Interface Segregation Principle (ISP)

En este principio nos recomienda que hagamos interfaces que sean específicas para un tipo de cliente.

D – Dependency Inversion Principle (DIP)

En este principio nos recomiendan dos importantes premisas:

Los módulos de alto nivel no deberían depender de módulos de bajo nivel.

Las abstracciones no deberían depender de los detalles.

**SECCIÓN 5: GIT.**

1. Para obtener una copia de un repositorio Git existente en GitHub, el comando que se necesita es:

1. git commit
2. git add
3. git log
4. git clone

1. ¿Cuál de los siguientes comandos crea una nueva rama local llamada develop?

1. git add develop
2. git checkout -b develop
3. git checkout – develop
4. git checkout develop