Investiga los siguientes conceptos de C#:

¿Qué es una clase en C# y cómo se crea?

R/ Es una plantilla o modelo que define las propiedades y métodos que pueden tener los objetos. En otras palabras, es una representación de una entidad o concepto con características y comportamientos definido

//Capa Datos

public class Personal

{

public int ID\_Personal { get; set; }

public string Nombre { get; set; }

public string Apellido { get; set; }

public string Cargo { get; set; }

}

}

//Capa Logica

public class PersonalLogica

{

public List<Personal> ListarPersonal()

{

List<Personal> listaPersonal = new List<Personal>();

using (SqlConnection conn = new SqlConnection(DBcon.conexion))

{

try

{

conn.Open();

SqlCommand cmd = new SqlCommand("SELECT ID\_Personal, Nombre, Apellido, Cargo FROM Personal", conn);

cmd.CommandType = CommandType.Text;

using (SqlDataReader dr = cmd.ExecuteReader())

{

while (dr.Read())

{

listaPersonal.Add(new Personal

{

ID\_Personal = Convert.ToInt32(dr["ID\_Personal"]),

Nombre = dr["Nombre"].ToString(),

Apellido = dr["Apellido"].ToString(),

Cargo = dr["Cargo"].ToString(),

});

}

}

}

catch (Exception ex)

{

Console.WriteLine($"Error al listar el personal: {ex.Message}");

}

finally

{

conn.Close();

}

}

return listaPersonal;

}

}

}

2. ¿Cuál es la diferencia entre los delimitadores de acceso private y public?

R/ public: Un miembro marcado como public es accesible desde cualquier parte del código. Cualquier otra clase o método puede acceder a ese miembro sin restricciones.

private: Un miembro marcado como private solo es accesible dentro de la clase en la que fue declarado. No puede ser accedido directamente desde fuera de esa clase.

3. ¿Cómo funcionan los ciclos for y while en C#?

R/ El ciclo for se utiliza cuando se conoce de antemano cuántas veces se va a ejecutar un bloque de código. Está estructurado en tres partes: inicialización, condición y actualización

El ciclo while repite un bloque de código mientras una condición sea verdadera. Se utiliza cuando no se conoce de antemano cuántas veces se repetirá el ciclo.

4. Explica el uso de la estructura if-else en C#.

R/ La estructura if-else permite ejecutar bloques de código basados en una condición. Si la condición evaluada es true, se ejecuta el bloque dentro del if. Si la condición es false, se puede ejecutar un bloque alternativo con else.

int edad = 18;  
 if (edad >= 18) { Console.WriteLine("Es mayor de edad."); }   
else { Console.WriteLine("Es menor de edad."); }

5. ¿Cuál es la diferencia entre variables locales y variables de clase en C#?

**R/ Variables locales**: Son aquellas que se declaran dentro de un método o bloque de código y solo existen durante la ejecución de ese método o bloque. Una vez que el método termina, las variables locales se destruyen y no son accesibles fuera de ese método.

**Variables de clase (o campos)**: Son variables que se declaran dentro de una clase pero fuera de cualquier método. Estas variables existen mientras exista una instancia de la clase y pueden ser accedidas por cualquier método de la clase.