**Tarea: Fundamentos de C#**

**Parte 1: Investigación Teórica** Investiga los siguientes conceptos de C#:

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

En C#, una **clase** es una plantilla o modelo que define los atributos (campos) y comportamientos (métodos) de los objetos que se crearán a partir de ella. Es uno de los pilares de la programación orientada a objetos (POO), lo que permite encapsular datos y operaciones relacionadas.

Para crear una clase en C#, se utiliza la palabra clave **class**, seguida del nombre de la clase.

Ejemplo:

Texto

Descripción generada automáticamente

En este ejemplo, Persona es una clase con dos atributos (Nombre y Edad), un constructor y un método (MostrarInfo).

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

Los **modificadores de acceso** en C# controlan la visibilidad y accesibilidad de los miembros de una clase (campos, propiedades, métodos).

* **public**: Permite que el miembro sea accesible desde cualquier parte del código, incluso desde otras clases y ensamblados.



**private**: Restringe el acceso solo a la clase en la que se define. No es accesible desde otras clases o partes externas.



La diferencia clave es el nivel de acceso que otorgan. Con public, cualquier parte del programa puede acceder y modificar ese miembro. Con private, el acceso está limitado solo a la propia clase.

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

**Ciclo for**: Se utiliza cuando conoces de antemano cuántas veces necesitas repetir un bloque de código. Se compone de tres partes: inicialización, condición y actualización.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

En este ejemplo, i comienza en 0, se ejecuta el bloque de código mientras i sea menor que 5, y i se incrementa en 1 en cada iteración.

**Ciclo while**: Ejecuta un bloque de código (una serie de instrucciones) mientras la condición especificada sea verdadera. Se utiliza cuando no sabes cuántas iteraciones habrá, pero necesitas seguir mientras una condición sea verdadera.

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

El ciclo se ejecuta mientras i sea menor que 5, incrementando i en cada iteración.

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

Interfaz de usuario gráfica, Texto

Descripción generada automáticamenteLa estructura if-else permite tomar decisiones basadas en una condición booleana (verdadero o falso). Si la condición es verdadera, se ejecuta el bloque de código del if; si es falsa, se ejecuta el bloque del else (si está presente).

Ejemplo:

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

**Variables locales**: Son aquellas que se declaran dentro de un método, constructor o bloque y solo existen dentro de ese ámbito. Una vez que el método o bloque termina, la variable local deja de existir.

Texto

Descripción generada automáticamente

**Variables de clase** (también llamadas campos o atributos): Se declaran dentro de la clase pero fuera de cualquier método. Son accesibles desde cualquier método de la clase y existen mientras exista una instancia de la clase (si no son estáticas).

Texto

Descripción generada automáticamente

La diferencia principal es que las variables locales tienen un alcance limitado al método donde se declaran, mientras que las variables de clase tienen un alcance que depende de la instancia de la clase o de la propia clase si son estáticas.

Fuentes consultadas:

**Microsoft Docs**. (2024). *Classes (C# Programming Guide)*. Disponible en: [https://learn.microsoft.com](https://learn.microsoft.com/en-us/dotnet/csharp/programming-guide/classes-and-structs/classes)

OpenAI. (2024). Respuesta a la consulta sobre clases en C#. ChatGPT. <https://chat.openai.com/>

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**Parte 2: Ejercicio Práctico**

using System;

using System.Collections.Generic;

class Estudiante

{

**// Propiedades, declaración de las variables del programa**

public string Nombre;

private int Edad;

public double Promedio;

**// Método Constructor de la Clase**

public Estudiante(string nombre, int edad, double promedio)

**//recibe las variables como párametros**

{

Nombre = nombre;**//asigna los valores recibidos a las variables del método**

Edad = edad;

Promedio = promedio;

}

**// Método para verificar si el estudiante es mayor de edad**

public bool EsMayorDeEdad()

**//Método booleano devuelve verdadero si se cumple la condición**

{

return Edad >= 18;

}

**// Método para mostrar la información del estudiante**

public void MostrarInfo()

{

Console.WriteLine("Nombre:"+Nombre,"Promedio:"+Promedio);

**//Imprime en consola los datos**

}

}

class Program **//Se declara la clase del programa**

{

static void Main(string[] args) **// Inicio**

{

// Solicitar la cantidad de estudiantes

Console.Write("Ingrese la cantidad de estudiantes que desea registrar: ");

int cantidadEstudiantes = int.Parse(Console.ReadLine());

**//guarda el dato ingresado en la variable convertido en entero**

**// Crear una lista para almacenar los estudiantes**

List<Estudiante> estudiantes = new List<Estudiante>();

**// Ciclo for para ingresar los datos de cada estudiante desde 0 hasta la cantidad almacenada en la variable**

for (int i = 0; i < cantidadEstudiantes; i++)

{

Console.WriteLine("\n--- Registro del estudiante"+i+1+"---");

**//imprime el título con el # del estudiante i+1**

**// Solicitar nombre**

Console.Write("Ingrese el nombre del estudiante: ");

string nombre = Console.ReadLine();

**// Solicitar edad**

Console.Write("Ingrese la edad del estudiante: ");

int edad = int.Parse(Console.ReadLine());**//guarda el dato en entero**

**// Solicitar promedio**

Console.Write("Ingrese el promedio del estudiante: ");

double promedio = double.Parse(Console.ReadLine());

**//guarda el dato en double--> permite decimales**

**// Crear un nuevo objeto Estudiante y agregarlo a la lista**

Estudiante estudiante = new Estudiante(nombre, edad, promedio);

**//crea el objeto estudiante con las propiedades**

estudiantes.Add(estudiante);//Agrega el nuevo objeto estudiante a la lista

}

**// Ciclo while para mostrar los estudiantes con promedio mayor o igual a 70**

int index = 0;//declara index como una variable entera que inicia en 0

Console.WriteLine("\n--- Estudiantes con promedio mayor o igual a 70 ---");

while (index < estudiantes.Count)

**// Count propiedad que me da el total de la lista de estudiantes**

{

Estudiante estudiante = estudiantes[index];

**// accesa a cada elemento de la lista según subindice index**

if (estudiante.Promedio >= 70)

**//accede a la información del promedio y verifica si es mayor o igual a 70**

{

estudiante.MostrarInfo(); **// llama a la función MostrarInfo**

**// Verificar si el estudiante es mayor de edad**

if (estudiante.EsMayorDeEdad())

{

Console.WriteLine("Es mayor de edad.");

}

else

{

Console.WriteLine("No es mayor de edad.");

}

}

index++; **//aumenta en 1 index**

}

}

}

Texto

Descripción generada automáticamente