



UPN, PASIÓN POR TRANSFORMAR VIDAS

UNIDAD 1: LISTAS, PILAS Y COLAS

SESIÓN 7: **Colas: Operaciones fundamentales, crear, destruir, agregar, remover**

Dr. Eric Gustavo Coronel Castillo
eric.coronel@upn.pe



DESAFIO DEL DIA

"La mente que se abre a una nueva idea,
nunca volverá a su tamaño original."

Albert Einstein



IDEA FUERZA

Cada nuevo conocimiento, paradigma o tecnología que aprendes transforma tu forma de pensar y programar; nunca vuelves a ser el mismo desarrollador.

LOGRO DE LA UNIDAD 1



Al finalizar la unidad, el estudiante implementa algoritmos utilizando listas simples, listas dobles, listas circulares, pilas y colas, a través de control de versiones y el lenguaje C# con entorno gráfico, demostrando lógica y habilidad en la implementación de los algoritmos.



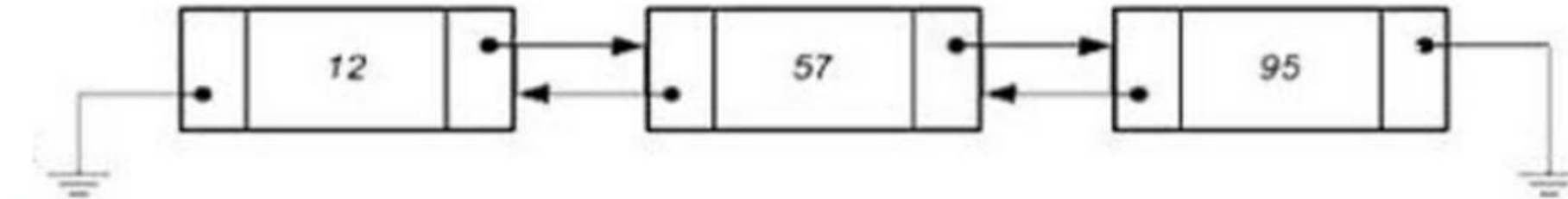
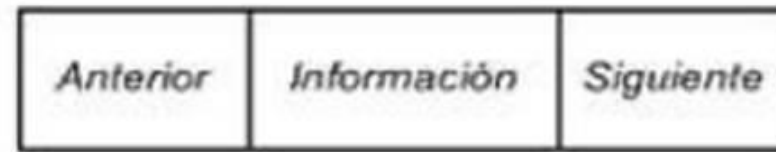
Al término de la sesión, el estudiante aprende y analiza algoritmos con pilas, usándolos con coherencia.

- Colas: Operaciones fundamentales: crear, destruir, agregar, remover.

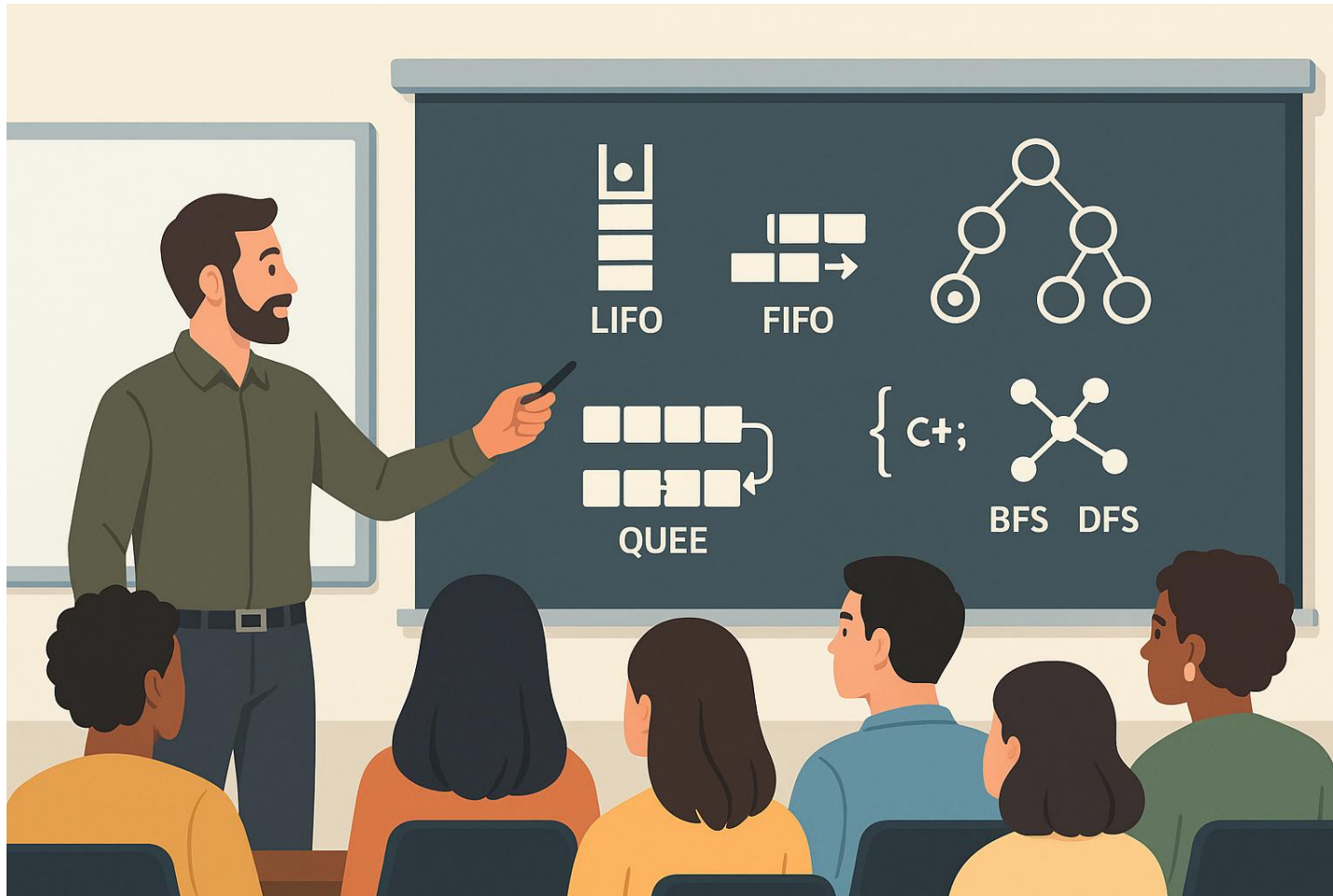


1. ¿Qué es una Cola?
2. ¿Qué operaciones se pueden realizar?

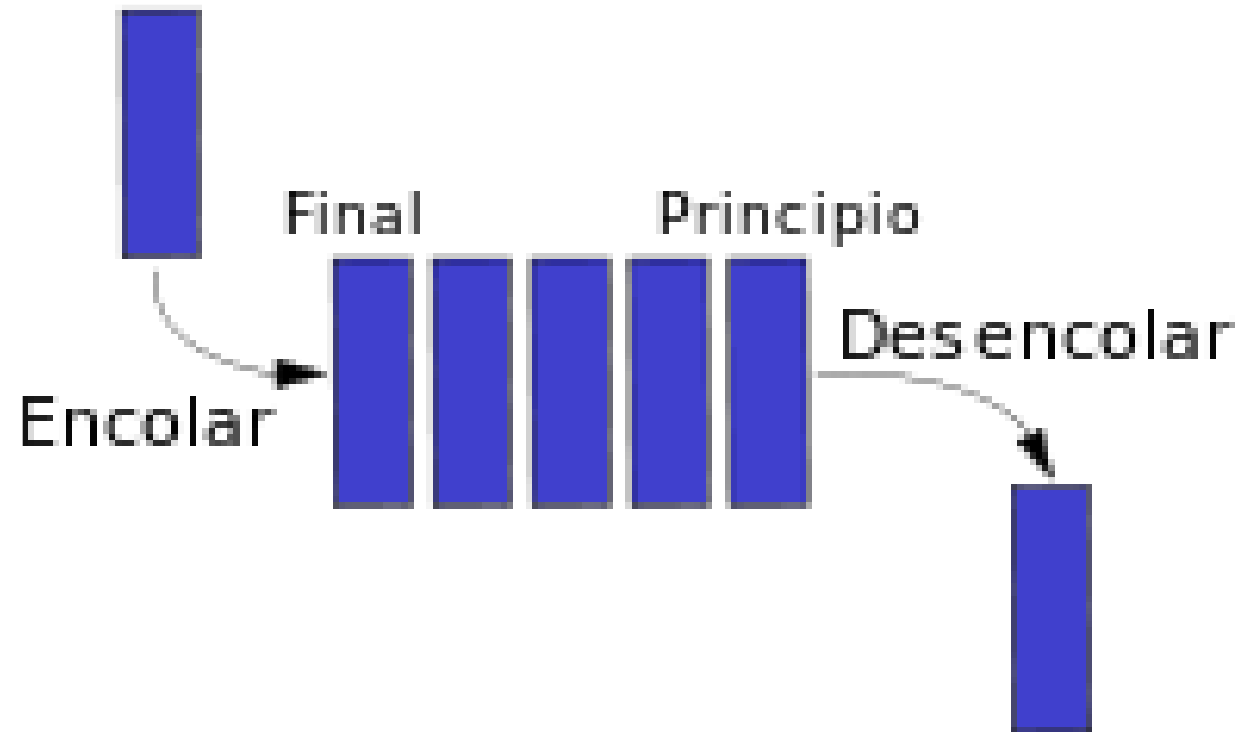
REFLEXIONA



DESARROLLO



¿QUÉ ES UNA COLA?



¿QUÉ ES UNA COLA?

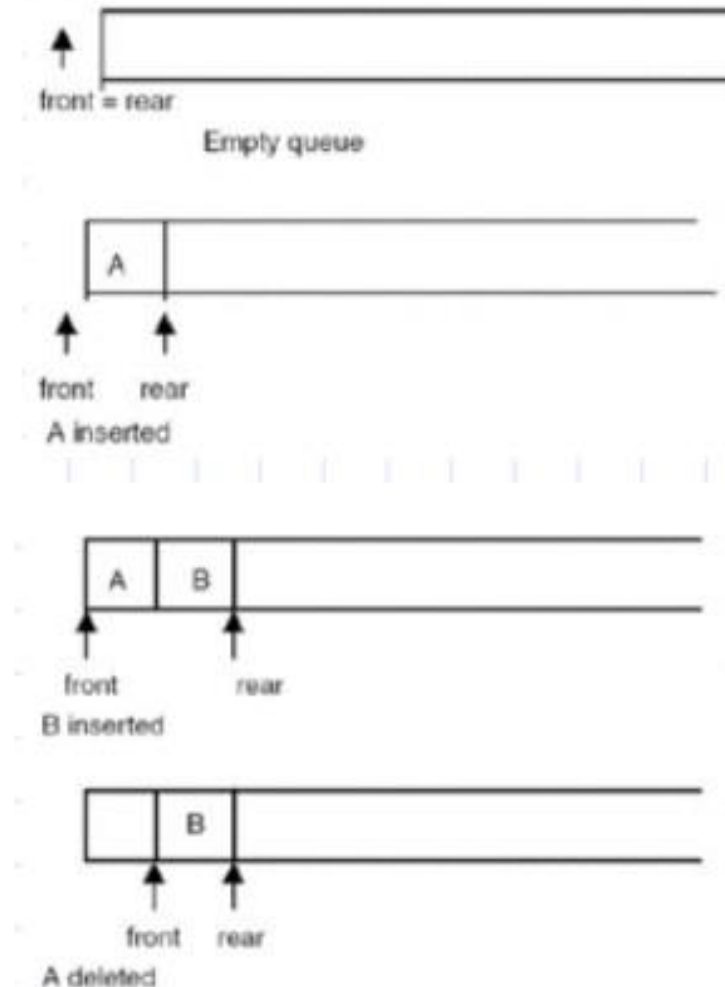


Encolar (enqueue)

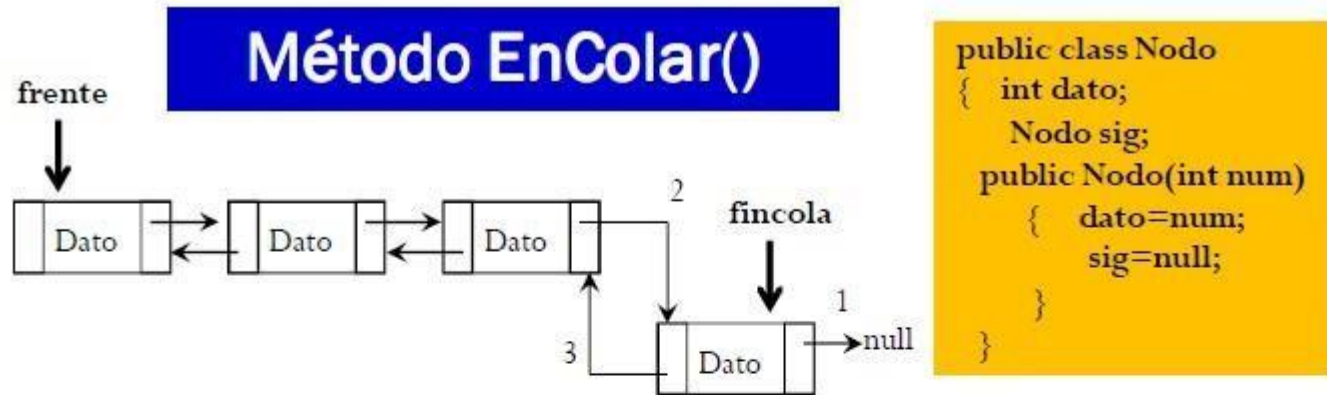
- Esta operación sirve para insertar un elemento e en la cola q.
- Enqueue(Q,e)

Desencolar (dequeue)

- Se usa para retirar un elemento de la cola Q y asignarlo a una variable del mismo tipo que el tipo de los elementos de la cola
- $V = \text{denqueue}(Q)$



ENCOLAR



```
private void EnColar(int num)
{
    Nodo nuevo=new Nodo(num);
    //Realizando los enlaces correspondientes
    if(frente==null)
        frente=nuevo;
    else
        fincola.sig=nuevo;
    fincola=nuevo;
    fincola.sig=null;
}
```

ACTIVIDAD EN CLASE



PROBLEMA 1



Se pide generar una cola de estructura FIFO (del inglés First In First Out) en consola que me permita encolar, desencolar, buscar, eliminar, mostrar y salir. Sabiendo que para eliminar puedo escoger la lista de la cola.

PROBLEMA 1



```
class Nodo
{
    private int dato;
    private Nodo siguiente;
    public int Dato
    {
        get { return dato; }
        set { dato = value; }
    }
    public Nodo Siguiente
    {
        get { return siguiente; }
        set { siguiente = value; }
    }
}
```

PROBLEMA 1



```
class Cola
{
    public Nodo primero = null;
    public Nodo ultimo = null;

    public Nodo Primero
    {
        get { return primero; }
        set { primero = value; }
    }
    public Nodo Ultimo
    {
        get { return ultimo; }
        set { ultimo = value; }
    }
}
```

```
public void queue(int dato)
{
    Nodo nuevoEncolado = new Nodo();
    nuevoEncolado.Dato = dato;
    if (Primero == null)
        Primero = nuevoEncolado;
    else
        Ultimo.Siguiente = nuevoEncolado;
    Ultimo = nuevoEncolado;
}
```

PROBLEMA 1



```
public int dequeue()
{
    int dato = Int32.MinValue;
    if (Primero != null)
    {
        dato = Primero.Dato;
        if (Primero == Ultimo)
            Ultimo = null;
        Primero =
Primero.Siguiente;
    }
    return dato;
}
```

```
public void MostrarCola(Cola cola) {
    Cola colaAuxiliar = new Cola();
    int datoDesencolado;
    if (cola.Primero != null) {
        while (cola.Primero != null) {
            datoDesencolado = cola.dequeue();
            Console.WriteLine(datoDesencolado);
            colaAuxiliar.queue(datoDesencolado);
        }
        while (colaAuxiliar.Primero != null)
            cola.queue(colaAuxiliar.dequeue());
    }
    else
        Console.WriteLine("NULL");
}
```


PROBLEMA 1



```
public void buscarDatoCola(Cola
cola, int datoBuscado)
{
    int contador = 0, datoDesencolado;
    Cola colaAuxiliar = new Cola();
    if (cola.Primerro != null) {
        while (cola.Primerro != null){
            datoDesencolado = cola.dequeue();
            if (datoDesencolado ==
                datoBuscado)
                contador++;
            colaAuxiliar.queue(datoDesencolado
        );
        }
    }
```

```
while (colaAuxiliar.Primerro != null)
    cola.queue(colaAuxiliar.dequeue());
if (contador == 0)
    Console.WriteLine("\nEl dato no fue encontrado
    en la cola.\n");
else if (contador == 1)
    Console.WriteLine("\nEl dato " + datoBuscado +
        " fue encontrado " + contador + " vez\n");
else
    Console.WriteLine("\nEl dato " + datoBuscado +
        " fue encontrado " + contador + " veces\n");
} else
    Console.WriteLine("\nLa cola no tiene
    datos.\n");
}
```

PROBLEMA 1



```
public void modificarDatoCola(Cola cola, int datoBuscado, int datoModificado)
{
    int contador = 0, datoDesencolado;
    Cola colaAuxiliar = new Cola();
    if (cola.Primeros != null)
    {
        while (cola.Primeros != null) {
            datoDesencolado = cola.dequeue();
            if (datoDesencolado == datoBuscado)
            {
                colaAuxiliar.queue(datoModificado);
                contador++;
            }else
                colaAuxiliar.queue(datoDesencolado);
        }
    }
}
```

PROBLEMA 1



```
while (colaAuxiliar.Primeros != null)
cola.queue(colaAuxiliar.dequeue());
if (contador == 0)
Console.WriteLine("\nEl dato que desea modificar no fue encontrado en la cola.\n");
else if (contador == 1)
Console.WriteLine("\nEl dato " + datoBuscado + " fue modificado " + contador + " vez
en la cola\n");
else
Console.WriteLine("\nEl dato " + datoBuscado + " fue modificado " + contador + " veces
en la cola\n");
} else
Console.WriteLine("\nLa cola no tiene datos.\n");
}
```

PROBLEMA 1



```
public void eliminarDatoCola(Cola cola, int datoAeliminar)
{
    int contador = 0, datoDesencolado;
    Cola colaAuxiliar = new Cola();
    if (cola.Primeros != null)
    {
        while (cola.Primeros != null)
        {
            datoDesencolado = cola.dequeue();
            if (datoDesencolado == datoAeliminar)
                contador++;
            else
                colaAuxiliar.queue(datoDesencolado);
        }
    }
}
```

PROBLEMA 1



```
while (colaAuxiliar.Primeros != null)
cola.queue(colaAuxiliar.dequeue());
if (contador == 0)
Console.WriteLine("\nEl dato que desea eliminar no fue encontrado en la pila.\n");
else if (contador == 1)
Console.WriteLine("\nEl dato " + datoAeliminar + " fue eliminado " +
contador + " vez en la pila\n");
else
Console.WriteLine("\nEl dato " + datoAeliminar + " fue eliminado " + contador + "
veces en la pila\n");
} else
Console.WriteLine("\nLa pila no tiene datos.\n");
}
```

CONCLUSIONES



- Las colas permiten gestionar la información en el orden en que llega, asegurando un procesamiento justo y organizado mediante la política FIFO (First In, First Out), lo que las hace esenciales en aplicaciones como sistemas de impresión, atención de procesos y gestión de solicitudes en servidores.
- El uso de colas optimiza la eficiencia y la organización en el manejo de datos y tareas, facilitando el control de flujos, la sincronización de procesos y la implementación de algoritmos en programación y sistemas operativos.

¿PREGUNTAS O COMENTARIOS?



COMPROBACIÓN DEL LOGRO





Cuestionario en la plataforma

Nivel	Rango
Nivel 4	17 – 20
Nivel 3	13 – 16
Nivel 2	9 – 12
Nivel 1	0 - 8

Comprobación del Logro

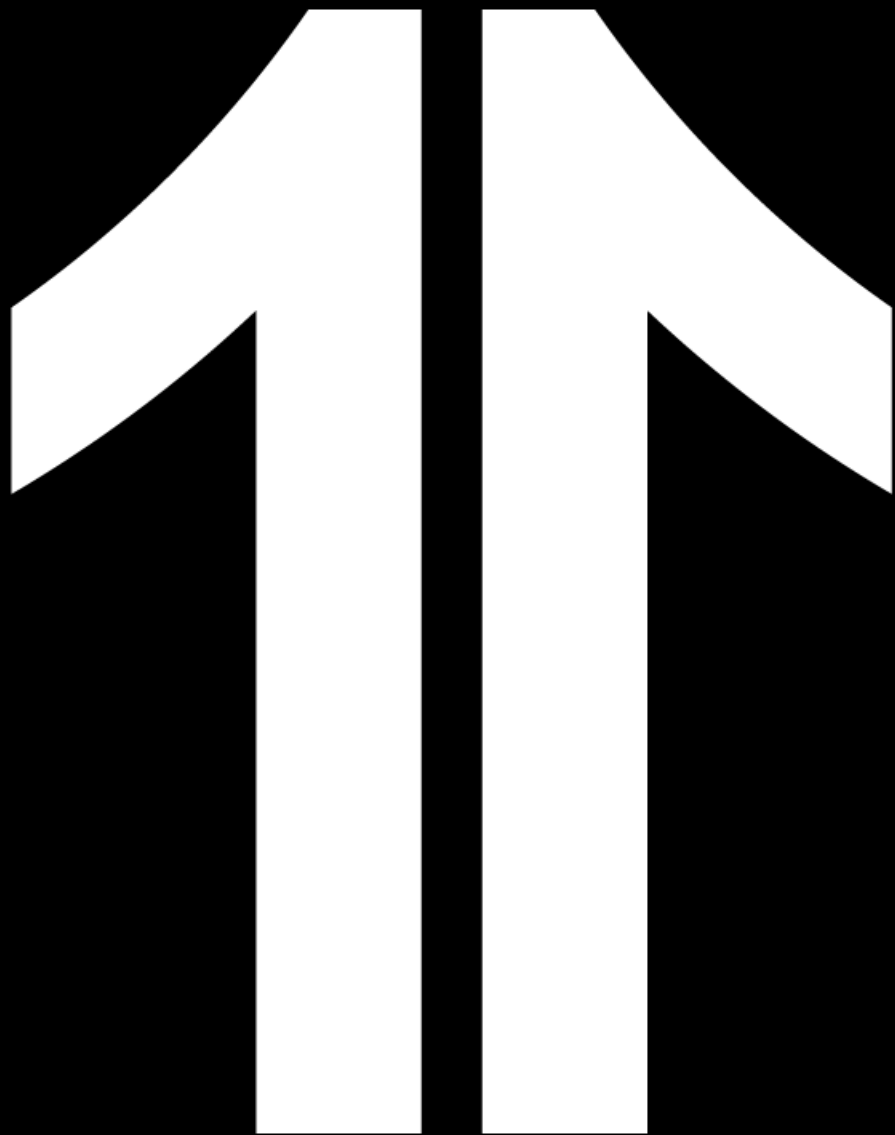


Criterio	Nivel 4 (Excelente)	Nivel 3 (Bueno)	Nivel 2 (Regular)	Nivel 1 (Deficiente)
Comprensión conceptual	Demuestra comprensión completa de los conceptos clave.	Comprende la mayoría de los conceptos clave.	Muestra comprensión parcial, con algunas confusiones.	No demuestra comprensión clara de los conceptos.
Aplicación de conocimientos	Aplica correctamente los conceptos en ejemplos o escenarios.	Aplica conceptos con algunos errores menores.	Aplica de forma superficial o con errores evidentes.	No logra aplicar los conceptos o lo hace incorrectamente.
Claridad y precisión en respuestas	Las respuestas son claras, bien estructuradas y precisas.	Las respuestas son mayormente claras, con leves imprecisiones.	Respuestas poco claras o con ideas mal organizadas.	Respuestas confusas, incompletas o incoherentes.
Cobertura del contenido esperado	Responde completamente las 5 preguntas, desarrollando bien cada una.	Responde correctamente 4 de las 5 preguntas.	Responde correctamente 2 o 3 preguntas.	Responde solo 1 pregunta correctamente o no responde.

BIBLIOGRAFIA REFERENCIAL



- Ceballos Sierra, F. Microsoft C#: Curso de Programación (2a.ed.) 2014
<https://elibronet.eu1.proxy.openathens.net/es/lc/upnorte/titulos/106417>
- Cesar Liza Avila; Estructura de datos con C/C++



GRACIAS

