

[Área personal](#) / [Mis cursos](#) / [\[2-2022\] INF220-SA](#) / [Examen Parcial 2](#) / [Examen Parcial 2](#)

<b>Comenzado el</b>	Tuesday, 17 de January de 2023, 07:57
<b>Estado</b>	Finalizado
<b>Finalizado en</b>	Tuesday, 17 de January de 2023, 08:17
<b>Tiempo empleado</b>	20 minutos 17 segundos
<b>Calificación</b>	<b>36,00</b> de 100,00

## Pregunta 1

Parcialmente correcta

Se puntúa 36,00 sobre 100,00

## // Clase del TAD Lista

```

public class clsLista
{
    // Se define la clase Nodo
    class Nodo
    {
        public int info;
        public Nodo sig;
    }

    // Se define los nodos tope (primero) y rear (ultimo)
    private Nodo tope, rear;

    // Constructor de la Clase Lista
    public clsLista()
    {
        tope = null;
        rear = null;
    }

    public clsLista Lista
    {
        get { return Lista; }
        set { Lista = value; }
    }
}

```

**Rellene el espacio en blanco con la lógica correcta para que funcione correctamente la función que adicione un nuevo nodo a una lista simple con valor "info", este nodo debe estar ubicado antes de un nodo que contenga valor "infX". Todas las instrucciones debe ser rellenada sin espacios en blanco.**

```
public clsLista AdicionarAntes( int infoX , int info)
```

```
{
```

```
    clsLista
```

```
    NL=new
```

✓ clsLista(); // se define una nueva lista con el nombre NL

```
    Nodo
```

```
    anterior;
```

✗ // se define un nodo con nombre anterior y se asigna valor nulo

```
    Nodo
```

```
    actual=tope;
```

✓ // se define un nodo con nombre actual y se asigna el puntero y el valor del nodo tope

```
    while
```

```
    (actual!=null)
```

✓ // recorrer toda la lista desde el nodo Tope hasta el nodo Rear

```
    {
```

```
        if
```

```
        actual.info==infoX
```

✗ // Valida si el nodo leído contiene el valor "infoX"

```

{
    Nodo nuevo;
    nuevo = new Nodo();    // Se declara un nodo con el nombre nuevo

    nuevo.info
    ✗    // Al nodo nuevo se le asigna el valor "info"

    nuevo.sig=actual;
    ✓    // El nuevo Nodo apunta al actual Nodo

    anterior.sig=nuevo
    ✗    // El anterior Nodo apunta al Nuevo nodo
}

anterior=actual;
✓    // El anterior nodo toma el puntero y el valor del nodo actual

actual=actual.sig;
✓    // el actual nodo toma el puntero y el valor del nodo sucesor del nodo actual
}

NL.tope=tope;
✓    // El tope del nueva lista toma los valores del nodo tope

NL.rear=rear;
✓    // El rear del nuevo lista toma los valores del tope rear

return
NL;
✓    // Retorna el valor y el puntero del nuevo nodo
}
}
}

```

**NOTA: Si la función no funciona después de rellenar las instrucciones, se descuenta el 40% de la nota final**

Comentario:

Nota	60
-40%	-24
Nota Final	36

◀ Lab3-T7

Ir a...