

# Programación Orientada a Objetos

2do Parcial

El parcial tiene una duración de 1 hora 40 mins. **No se permite el uso de material.**

## Pregunta 1 (30 ptos)

```
public class TanqueCilindrico {
    public String Color { get; set; }
    private double CalcularVolumenCilindro() { /* Implementacion */ }
    public double CalcLitrosCilindro() {
        ...
        double h = CalcularVolumenCilindro()
        /* Se usa la variable h para el cálculo de litros */
    }
}

public class TanqueRectangular {
    public String Color { get; set; }
    private double CalcularVolumenRectangular() { /* Implementacion */ }
    public double CalcLitrosRect() {
        ...
        double h = CalcularVolumenRectangular()
        /* Se usa la variable h para el cálculo de litros */
    }
}
```

Realiza los cambios necesarios en el código para poder soportar diferentes tipos de tanques enfocado a la reutilización de código. **Nota: NO es necesario desarrollar el cálculo del volumen**

## Pregunta 2 (35 ptos)

```
class AdminDiscos {
    public void Formatear (String unidad) { /* implementacion */ }
}

class Windows {
    public AdminDiscos Discos = new AdminDiscos();
}

class Linux {
    public AdminDiscos Discos = new AdminDiscos();
}

class PC {
    public Windows soWin = new Windows();
    public Linux soLnx = new Linux();
    public void Formatear (int tipo, String unidad) {
        if (tipo == 0) {
            soWin.Discos.Formatear(unidad);
        } else {
            soLnx.Discos.Formatear(unidad);
        }
    }
}
```

```

class Program {
    static void Main (String args[]) {
        PC miPC = new PC();
        miPC.Formatear(0, "c:/");
    }
}

```

Crítica el código anterior en base a los principios SRP y OCP y realiza los cambios necesarios para que los cumpla. Brinda como ejemplo el ingreso de un nuevo sistema operativo (MacOS).

### Pregunta 3 (20 ptos)

```

class Laboratorio {
    public Boolean EsCopiaDe(Laboratorio otro) {
        /* retorna si existe un plagio */
    }
}

class Profesor {
    private IList<Laboratorio> labs = new List<>();
    private int PonerNota (Laboratorio lab) { /* pone nota a un laboratorio */ }
    public int Corregir (Laboratorio lab) {
        int a = 0;
        foreach (Laboratorio elem in labs) {
            if (elem.EsCopiaDe(lab)) {
                Console.WriteLine("COPIA!!!");
                break;
            }
        }
        a = PonerNota (lab);
        return a;
    }
}

```

Realiza los cambios en el código para que, cuando se comete un plagio, se lance una excepción específica llamada "PlagioException". Debes proveer además un código de ejemplo (método Main en el Program) demostrando el manejo de la excepción (try, catch).

### Ejercicio 4 (15 ptos)

```

class Jugador {
    public void Jugar() {
        Console.WriteLine ("Jugando a la pelota.");
    }
}

class Messi : Jugador {
    public void Jugar() {
        Console.WriteLine ("Haciendo maravillas con la pelota.");
    }
}

```

- 4.1 Que entiendes por encadenamiento estático y dinámico?, donde se origina?.
- 4.2 Modifica el código anterior para que el siguiente fragmento de código imprima "Haciendo maravillas con la pelota".

```

Jugador jugador = new Messi();
jugador.Jugar();

```