

Programación 2

2do Parcial - Viernes 6 de Noviembre

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

Pregunta 1 (10 pts)

```
class Cantante {
    public void Cantar() {
        Console.WriteLine ("Cantando con una voz privilegiada.");
    }
}

class ElGucci : Cantante {
    public void Cantar() {
        Console.WriteLine ("Voz poco privilegiada.");
    }
}
```

1.1) ¿Qué entiendes por encadenamiento estático y dinámico?, ¿dónde se origina?.

1.2) Modifica el código anterior para que el siguiente fragmento de código imprima "Voz poco privilegiada".

```
Cantante cantante = new ElGucci();
cantante.Cantar();
```

Pregunta 2 (20 pts)

```
class Parcial {
    public Boolean EsIdentico(Parcial otro) {
        /* retorna si son iguales los parciales */
    }
}

class Profesor {
    private IList<Parcial> parciales = new List<>();
    private int PonerNota (Parcial parcial) { /* pone nota a un parcial */ }
    public int Corregir (Parcial parcial) {
        int a = 0;
        foreach (Parcial elem in parciales) {
            if (elem.EsIdentico(parcial)) {
                Console.WriteLine("COPIA!!!");
                break;
            }
        }
        a = PonerNota (parcial);
        return a;
    }
}
```

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

Pregunta 3 (35 ptos)

```
class Abitab {
    public double PagarPorAbitab(Int32 valor) { /* */ }
}
class RedPagos {
    public double PagarPorRedPagos(Int32 valor) { /* */ }
}
class CompraPartido {
    public double ComprarEntrada (Int32 valor, Int32 cual) {
        double precioFinal = 0;
        if (cual == 0) {
            Abitab abitab = new Abitab();
            double a = abitab.PagarPorAbitab (valor);
            precioFinal = a * 0.25;
        }
        if (cual == 1) {
            RedPagos redpagos = new RedPagos();
            double a = redpagos.PagarPorRedPagos (valor);
            precioFinal = a * 0.35;
        }
        return precioFinal;
    }
}
```

3.1) Critica el código anterior en base a los principios SRP y OCP.

3.2) Modifica el código para que se cumplan los principios anteriores.

Pregunta 4 (35 ptos)

```
public class Cuadrado {
    public String Color { get; set; }
    public double Lado { get; set; }
    private double CalcularAreaCuadrado() { return Lado * Lado; }
    public void Pintar() {
        double h = CalcularAreaCuadrado(); /* Se usa la variable h para pintar */
    }
}
public class Rectangulo {
    public String Color { get; set; }
    public double Ancho { get; set; }
    public double Largo { get; set; }
    private double CalcularAreaRectangulo() { return Largo * Ancho; }
    public void Pintar() {
        double h = CalcularAreaRectangulo(); /* Se usa la variable h para pintar */
    }
}
```

Realiza los cambios necesarios en el código para poder soportar diferentes tipos de figuras geométricas (ejemplificando con un triángulo) enfocado a la reutilización de código.