

# Programación 2

*Primer Parcial 2do Semestre 2015 - Viernes 4 de Setiembre del 2015*

El parcial tiene una duración de 1 hora 40 mins. Terminado el parcial, se realiza un recreo de 10 minutos y se procederá a dar la solución en clase. **No se permite el uso de material.**

## Pregunta 1 (20 pts)

Sean las siguientes interfaces

```
public interface IJugador {
    int ObtenerSalud();
    int ObtenerPoderAtaque();
    int ObtenerPoderDefensa();
    int Atacar();
    String Interactuar(String mensaje);
    String ObtenerNombre();
}

public interface INPC {
    String SolicitarQuest();
}

public interface IJugadorArtificial {
    void DefinirRuta(Ruta ruta);
    void DefinirAgresividad(int agresividad);
}

public interface IJugadorHumano {
    void Hablar(String mensaje);
    void Mover(Punto punto);
    void Cast(Spell spell);
}

public class GuardiaCastillo : IJugador, INPC { /*Codigo */ }
public class Monstruo : IJugador, IJugadorArtificial { /*Codigo */ }
public class JugadorHumano : IJugador, IJugadorHumano { /*Codigo */ }
public class Mascota : IJugador { /*Codigo */ }

public static void Main(String[] args) {
    /*1*/ IJugador j = new JugadorHumano();
    /*2*/ IJugadorArtificial m = new Monstruo();
    /*3*/ int p = m.ObtenerPoderAtaque();
    /*4*/ IJugador o = new Monstruo();
    /*5*/ int d = o.ObtenerPoderDefensa();
    /*6*/ m = o;
    /*7*/ j = m;
    /*8*/ Mascota dragon = new IJugador();
    /*9*/ dragon = new Mascota();
    /*10*/ dragon.SolicitarQuest();
}
```

1.1) Indique las líneas de código incorrectas y justifique brevemente.

1.2) ¿Qué tipos tienen los objetos creados en las líneas 1 y 4?

1.3) ¿Qué mensajes puede recibir un objeto como el creado en la línea 2?. ¿Cómo lo sabes?

1.4) ¿Qué mensajes puede recibir un objeto bajo los términos de la variable declarada en la línea 8? ¿Cómo lo sabes?

1.5) ¿Que contrato firmaron las variables declaradas en las líneas 2 y 4?

## Pregunta 2 (10 pts)

Teniendo en cuenta el ejercicio anterior, ahora suponga que tenemos un juego con su respectivo mapa. Para ello, nos fabricamos la siguiente clase:

```
public class Mapa {
    /*1*/ private ArrayList jugadores;

    public int EnviarMensajeGlobal() {
        /*2*/ String mensaje = "El servidor se encontrará cerrado a partir de las 14.00 EST";
        /*3*/ int jugadoresComunicados = 0;
        /*4*/ foreach (IJugador j in jugadores) {
            /*5*/ j.Interactuar(mensaje);
            /*6*/ jugadoresComunicados++;
            /*7*/ String str = new String();
            /*8*/ str = "Se comunicó al jugador: " + j.ObtenerNombre();
            /*9*/ Console.WriteLine (str);
        }
        /*10*/ return jugadoresComunicados;
    }

    public Mapa () {
        /*11*/ jugadores = new ArrayList();
    }

    public void AgregarJugador (IJugador jugador) {
        /*12*/ jugadores.Add(jugador);
    }

    public void EnviarMensajeGlobal(String mensaje){
        /*13*/ foreach (IJugador j in jugadores) {
            /*14*/ j.Interactuar(mensaje);
            /*15*/ String str = new String();
            /*16*/ str = "Se comunicó al jugador: " + j.ObtenerNombre();
            /*17*/ Console.WriteLine (str);
        }
    }
}
```

2.1) ¿Qué entiende por polimorfismo de tipos?

2.2) ¿Indique en cuál/es línea/s del código se ejecuta polimórficamente un método? ¿Cómo lo/s detectó?

### Pregunta 3 (15 ptos)

Dado el siguiente código

```
public class Clase {
    private int instanciasCreadas;
    private Metodo constructor;
    private ArrayList metodos;
    public String ImprimirCodigo() { return Imprimir(); }
    private String Imprimir() { /* código */ }
}

public class Program {
    public static void Main (String[] args) {
        // Código a implementar por el alumno.
    }
}
```

3.1) Modifica la clase Clase y escribe en el método Main del Program.cs un programa que genere variables que referencian a objetos del tipo Clase, que sean iguales e idénticos.

3.2) Realiza los cambios necesarios para que la clase Clase quede inmutable.

### Pregunta 4 (15 ptos)

Dada la siguiente clase

```
public class Objeto {
    private ArrayList implements;
    private ArrayList atributos;
    private ArrayList metodos;
    private Clase clase;

    public static Int32 MaxAtributos() { return 16495; }
    public void Llamar(String metodo, ArrayList params) {
        /* busca el método y lo ejecuta con los parámetros */
    }

    public Object ObtenerAtributo(String nombreAtributo) {
        /* Busca el atributo y lo setea en valor */
        return valor;
    }

    public Objeto (Clase clase) {
        this.clase = clase;
        /* construir objeto */
    }
}
```

4.1) Indica cuales son métodos de instancia y cuales son métodos de clase.

4.2) ¿Cuál es la diferencia entre ambos conceptos?

4.3) ¿Cómo se define la sobrecarga de métodos?. ¿Qué particularidad debe tener?.

4.4) En base a la clase anterior, ejemplifica la sobrecarga del método constructor y sobrecarga el método Llamar para cuando se quiere llamar un método que no toma parámetros.

**Pregunta 5 (40 ptos)**

Dentro de una empresa se trabaja con órdenes de pedidos. Para la orden de pedido es necesario que existan los siguientes datos: código del producto, cantidad, precio unitario y precio total de la orden.

Realiza un programa capaz de almacenar las órdenes de pedido e imprimir el valor total del conjunto de órdenes. Deberás proveer una solución que respete los principios de programación orientada a objetos que hemos visto.

Recuerda encapsular las clases, **brindar las interfaces correctas** y un programa principal que muestre lo realizado.