

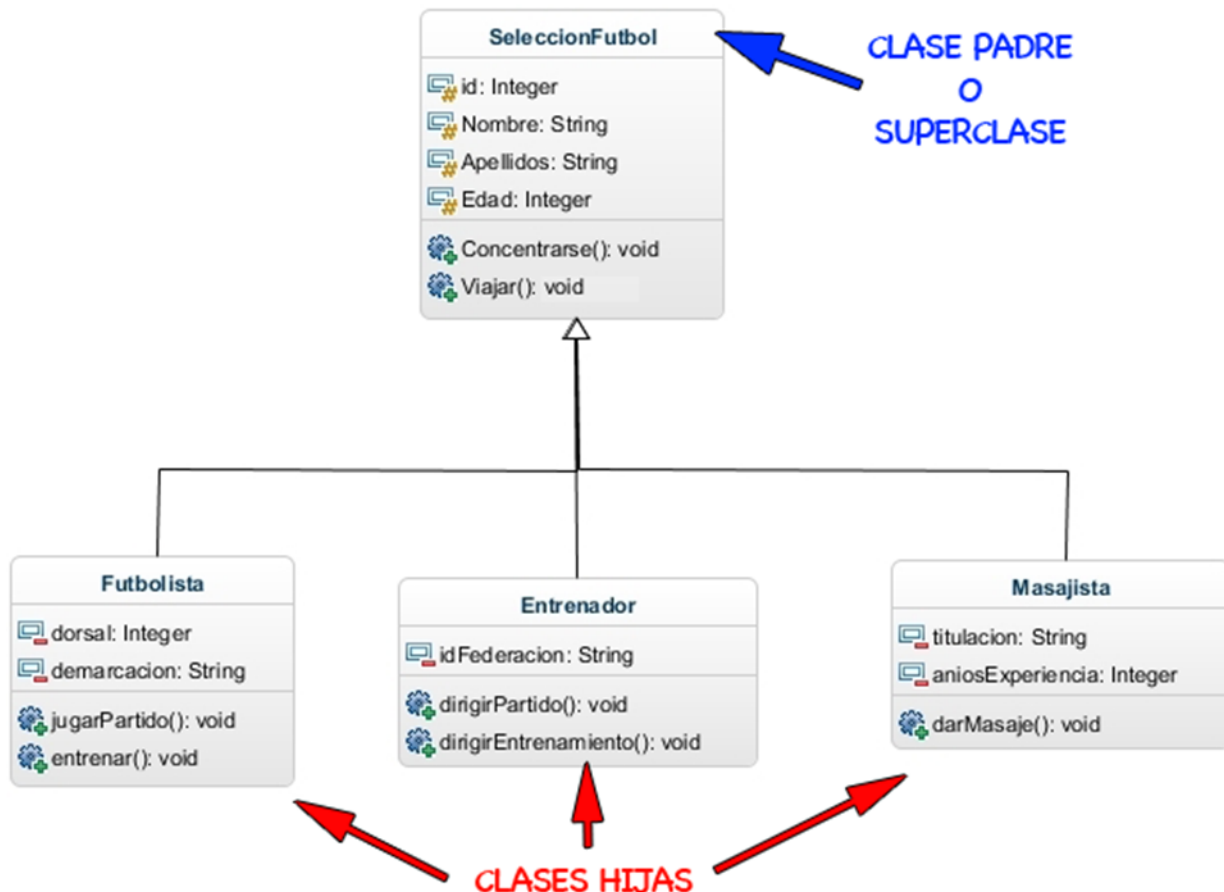
## Segundo Examen Parcial de SIS103

**Nombre:** Gonzales Suyo Franz Reinaldo

**C.U.** 111-500

**Carrera:** Ing. Ciencias de la Computación

1.- Desarrollar un programa en java, con una clase padreSeleccionFutbol y tres clases hijas Futbolista, Entrenador y masajista. las cuales heredan de la clase padre todos los atributos y métodos públicos o protegidos.



**Clase SeleccionFutbol:**

```
public class SeleccionFutbol {

    protected int id;
    protected String nombre;
    protected String apellido;
    protected int edad;

    public SeleccionFutbol(int id, String nombre, String apellido, int edad) {
        this.id = id;
        this.nombre = nombre;
        this.apellido = apellido;
        this.edad = edad;
    }
}
```

```

    }

    public void Concentrarse() {

        System.out.println("La seleccion de Futbol se concentra");
    }

    public void Viajar() {

        System.out.println("La seleccion de Futbol viaja");
    }

    public void presentarse(){
        System.out.println("ID: " + id);
        System.out.println("Nombre: " + nombre);
        System.out.println("Apellido: " + apellido);
        System.out.println("Edad: " + edad);
    }
}

```

### Clase Futbolista:

```

public class Futbolista extends SeleccionFutbol {

    private int dorsal;
    private String demarcacion;

    public Futbolista(int id, String nombre, String apellido, int edad, int
dorsal, String demarcacion) {
        super(id, nombre, apellido, edad);
        this.dorsal = dorsal;
        this.demarcacion = demarcacion;
    }

    public void JugarPartido() {
        System.out.println("El futbolista " + nombre + " juega un partido");
    }

    public void Entrenar() {
        System.out.println("El futbolista " + nombre + " procede a entrenar");
    }

    public void presentarse(){
        super.presentarse();
        System.out.println("Dorsal: " + dorsal);
        System.out.println("Demarcación: " + demarcacion);
    }
}

```

### Clase Entrenador:

```
public class Entrenador extends SeleccionFutbol {

    private String idFederacion;

    public Entrenador(int id, String nombre, String apellido, int edad, String idFederacion) {

        super(id, nombre, apellido, edad);
        this.idFederacion = idFederacion;
    }

    public void DirigirPartido() {
        System.out.println("El entrenador " + nombre + " dirige un partido");
    }

    public void DirigirEntrenamiento() {
        System.out.println("El entrenador " + nombre + " dirige un entrenamiento");
    }

    public void presentarse(){
        super.presentarse();
        System.out.println("ID de Federación: " + idFederacion);
    }
}
```

### Clase Masajista:

```
public class Masajista extends SeleccionFutbol{

    private String titulacion;
    private int aniosExperiencia;

    public Masajista(int id, String nombre, String apellido, int edad, String titulacion, int aniosExperiencia) {

        super(id, nombre, apellido, edad);
        this.titulacion = titulacion;
        this.aniosExperiencia = aniosExperiencia;
    }

    public void darMasaje() {

        System.out.println("El masajista " + nombre + " da un masaje a los jugadores");
    }

    public void Entrenar() {
        System.out.println("El masajista " + nombre + " entrena");
    }
}
```

```

    public void presentarse(){
        super.presentarse();
        System.out.println("Título: " + titulacion);
        System.out.println("Años de experiencia: " + aniosExperiencia +
"años.");
    }
}

```

### Clase PruebaSeleccionFutbol:

```

public class PruebaSeleccionFutbol {

    public static void main(String[] args) {

        SeleccionFutbol seleccionFutbol = new SeleccionFutbol(1, "Carlos",
"Torrez", 30);

        Futbolista futbolista = new Futbolista(1, "Juan", "Perez", 22, 10,
"DeLantero");
        Entrenador entrenador = new Entrenador(2, "Alejandro", "Chavez", 45,
"Licenciado en Educacion Fisica");
        Masajista masajista = new Masajista(3, "Mario", "Lopez", 35, "Licenciado
en Fisioterapia", 10);

        System.out.println("===== Selección de Fútbol =====");
        seleccionFutbol.Concentrarse();
        seleccionFutbol.Viajar();
        System.out.println();

        System.out.println("===== Futbolista =====");
        futbolista.presentarse();
        futbolista.JugarPartido();
        futbolista.Entrenar();
        System.out.println();

        System.out.println("===== Entrenador =====");
        entrenador.presentarse();
        entrenador.DirigirPartido();
        entrenador.DirigirEntrenamiento();
        System.out.println();

        System.out.println("===== Masajista =====");
        masajista.presentarse();
        masajista.darMasaje();
    }
}

```

## Prueba de Ejecución

```
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS SQL CONSOLE COMMENTS powershell + v [ ] [ ] ... ^ x
PS C:\GONZALES\SIS103\Parciales\SegundoParcial> javac PruebaSeleccionFutbol.java SeleccionFutbol.java Futbolista.java Entrenador.java Masajista.java
PS C:\GONZALES\SIS103\Parciales\SegundoParcial> java PruebaSeleccionFutbol

===== Seleccción de Fútbol =====
La seleccion de Futbol se concentra
La seleccion de Futbol viaja

===== Futbolista =====
ID: 1
Nombre: Juan
Apellido: Perez
Edad: 22
Dorsal: 10
Demarcación: Delantero
El futbolista Juan juega un partido
El futbolista Juan procede a entrenar

===== Entrenador =====
ID: 2
Nombre: Alejandro
Apellido: Chavez
Edad: 45
ID de Federación: Licenciado en Educacion Fisica
El entrenador Alejandro dirige un partido
El entrenador Alejandro dirige un entrenamiento

===== Masajista =====
ID: 3
Nombre: Mario
Apellido: Lopez
Edad: 35
Título: Licenciado en Fisioterapia
Años de experiencia: 10años.
El masajista Mario da un masaje a los jugadores
PS C:\GONZALES\SIS103\Parciales\SegundoParcial> |
```

## 2.- Ejecutar el siguiente programa

Tenemos nuestra Clase Padre llamada:

Animal

Y 3 Clases Hijas llamadas:

Perro

Caballo

Gato

Cada una de las clases hijas tiene diferente raza, pero comparten las mismas propiedades de un Animal para ello crearemos nuestra clase Padre Animal:

## Clase Animal:

```
public class Animal {

    private String nombre, tipo_alimentacion;
    private int edad;

    public Animal(String nombre, String tipo_alimentacion, int edad){
        this.nombre = nombre;
        this.tipo_alimentacion = tipo_alimentacion;
        this.edad = edad;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public String getTipo_alimentacion() {
        return tipo_alimentacion;
    }

    public void setTipo_alimentacion(String tipo_alimentacion) {
        this.tipo_alimentacion = tipo_alimentacion;
    }

    public int getEdad() {
        return edad;
    }

    public void setEdad(int edad) {
        this.edad = edad;
    }
}
```

## Clase Perro:

```
public class Perro extends Animal {

    private String raza;

    public Perro(String nombre, String tipo_alimentacion, int edad, String raza){
        super(nombre, tipo_alimentacion, edad);
        this.raza = raza;
    }

    public String getRaza() {
        return raza;
    }
}
```

```

    public void setRaza(String raza) {
        this.raza = raza;
    }

    public void mostrar(){
        System.out.println(getNombre() + "-" + getTipo_alimentacion() + "-" +
getEdad() + "-" + getRaza());
    }
}

```

### Clase Caballo:

```

public class Caballo extends Animal {

    String raza;

    public Caballo(String nombre, String tipo_alimentacion, int edad,String
raza) {
        super(nombre, tipo_alimentacion, edad);
        this.raza = raza;
    }

    public String getRaza() {
        return raza;
    }

    public void setRaza(String raza) {
        this.raza = raza;
    }

    public void mostrar(){
        System.out.println(getNombre() + "-" +getTipo_alimentacion()+"-
"+getEdad()+"-"+getRaza());
    }
}

```

### Clase Gato:

```

public class Gato extends Animal {

    String raza;

    public Gato(String nombre, String tipo_alimentacion, int edad, String raza)
{
        super(nombre, tipo_alimentacion, edad);
        this.raza = raza;
    }

    public String getRaza() {
        return raza;
    }
}

```

```

    }

    public void setRaza(String raza) {
        this.raza = raza;
    }

    public void mostrar(){
        System.out.println(getNombre() + "-" + getTipo_alimentacion() + "-"
        + getEdad() + "-" + getRaza());
    }
}

```

### Clase Main:

```

public class Main {

    public static void main(String[] args) {

        Perro perro = new Perro("Teddy", "Croquetas", 10, "Chihuahua");
        Gato gato = new Gato("Pelusa", "Especial", 8, "Siames");
        Caballo caballo = new Caballo("Jhonny", "Pasto", 21, "Fino");

        //-->Nos muestra los detalles del objeto
        perro.mostrar();
        System.out.println("-----");
    };

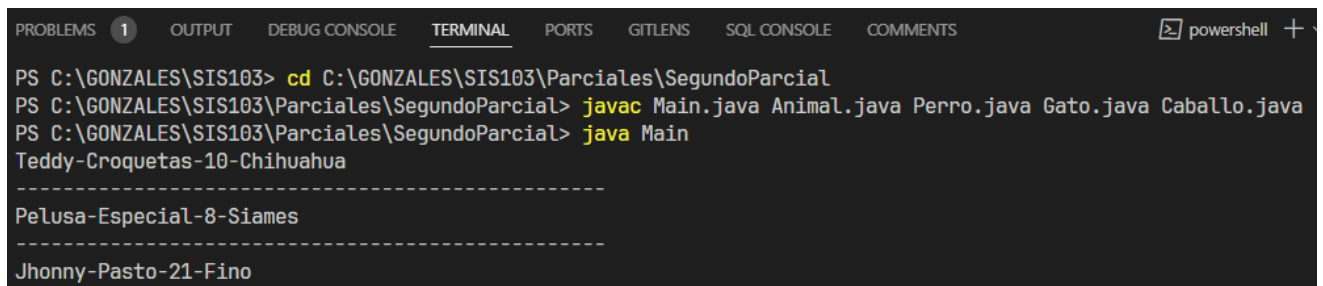
    gato.mostrar();
    System.out.println("-----");
    };

    caballo.mostrar();

    }
}

```

### Prueba de ejecución:



```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS SQL CONSOLE COMMENTS powershell +
PS C:\GONZALES\SIS103> cd C:\GONZALES\SIS103\Parciales\SegundoParcial
PS C:\GONZALES\SIS103\Parciales\SegundoParcial> javac Main.java Animal.java Perro.java Gato.java Caballo.java
PS C:\GONZALES\SIS103\Parciales\SegundoParcial> java Main
Teddy-Croquetas-10-Chihuahua
-----
Pelusa-Especial-8-Siames
-----
Jhonny-Pasto-21-Fino

```