

HERENCIA Y POLIMORFISMO -01

- Construir las 3 clases en Dart tomando como base el diagrama de clases anterior.
- Para llamar el constructor de la clase padre desde la clase Futbolista y Programador, se deberá hacer utilizando el método correspondiente de la herencia

```
class Persona {
  String _nombre;
  int _edad;
  Persona(this._nombre, this._edad);
  setNombre(String nombre){
    _nombre = nombre;
  }
  getNombre(){
    return _nombre;
  }
  setEdad(int edad){
    _edad = edad;
  }
  getEdad (){
    return _edad;
  }
  void mostrarInfo1(){
    print("""
su nombre es $_nombre,
y la edad es $_edad
""");
  }
}
```



```
1  import 'Persona.dart';
2
3  class Futbolista extends Persona{
4      String _equipo;
5      String _posicion;
6      int _cantidadGoles;
7      Futbolista(this._equipo,this._posicion,this._cantidadGoles,
8      String _nombre,int _edad) : super(_nombre,_edad);
9      setEquipo(String equipo){
10         _equipo = equipo;
11     }
12     getEquipo(){
13         return _equipo;
14     }
15     setPosicion(String posicion){
16         _posicion = posicion;
17     }
18     getPosicion (){
19         return _posicion;
20     }
21     setCantGoles(int goles){
22         _cantidadGoles = goles;
23     }
24     getCantGoles(){
25         return _cantidadGoles;
26     }
27     void esTitular(esTitular){
28         bool titular = false;
29         if(esTitular.toUpperCase=='SI'){
30             titular = true;
31             print('El jugador es titular');
32         } else {
33             print('El jugador no es titular');
34         }
35     }
36     void mostrarInfo(){
37         this.mostrarInfo1();
38         print("""
39         el equipo es $_equipo,
40         la posicion es $_posicion,
41         la cantidad de goles es $_cantidadGoles
42         """);
43     }
44 }
```

```
1  import 'Persona.dart';
2
3  class Programador extends Persona {
4      String _empresa;
5      double _salario;
6      Programador(this._empresa, this._salario,
7      String _nombre, int _edad) : super(_nombr
8      e, _edad);
9      setEmpresa(String empresa){
10         _empresa = empresa;
11     }
12     getEmpresa(){
13         return _empresa;
14     }
15     setSalario(double salario){
16         _salario = salario;
17     }
18     getSalario(){
19         return _salario;
20     }
21     obtenerSalarioNeto(){
22     }
23     void mostrarInfo2(){
24         this.mostrarInfo();
25         print("""
26         La empresa en la que trabaja es $_empresa,
27         y su salario es de $_salario
28         """);
29     }
30 }
```

- c. Crear 1 objeto (instancia) de la clase Persona, 2 objetos (instancias) de la clase Futbolista y 2 objetos (instancias) de la clase Programador

```
void main(List<String> args) {  
    String nomPerson;  
    int edadPerson;  
  
    print("Indique el nombre de la persona");  
    nomPerson = stdin.readLineSync()!;  
    print('Indique la edad de la persona');  
    edadPerson = int.parse(stdin.readLineSync()!);  
  
    Persona myPerson = Persona(nomPerson, edadPerson);  
  
    myPerson.mostrarInfo();  
}
```

```

for (var i = 0; i < cantPerson; i++) {
    print(
        | ""En este momento ingresará los datos de la persona en el siguiente orden.
        | 1. Nombre de la Persona.
        | 2. Edad de la persona.
        | 3. Profesion de la persona.""");
    nomPerson = stdin.readLineSync();
    edadPerson = int.parse(stdin.readLineSync());
    profesion = stdin.readLineSync();
    //----- CREACION DE OBJETOS TIPO JUGADOR -----//
    if (profesion.toLowerCase() == 'jugador') {
        print("""Indique los datos de la profesion en el siguiente orden.
            | 1. Equipo en el que juega
            | 2. Posicion en la que juega
            | 3. Cantidad de goles en la temporada (Esto determinará si es Titular o no)""");
        equipoFut = stdin.readLineSync();
        posicionFut = stdin.readLineSync();
        cantGoles = int.parse(stdin.readLineSync());
        Futbolista myPerson = Futbolista(
            | equipoFut, posicionFut, cantGoles, nomPerson, edadPerson, profesion);
        myPerson.esTitular();
        listadoPersonas.add(myPerson);
        //----- CREACION DE OBJETOS TIPO PROGRAMADOR -----//
    } else if (profesion == 'programador') {
        print("""Indique los datos de la profesion en el siguiente orden.
            | 1. Empresa en la que trabaja
            | 2. Salario de la Persona""");
        empresaPerson = stdin.readLineSync();
        salario = double.parse(stdin.readLineSync());
        Programador myPerson =
            | Programador(empresaPerson, salario, nomPerson, edadPerson, profesion);
        myPerson.obtenerSalarioNeto();
        listadoPersonas.add(myPerson);
    } else {
        while (profesion != 'jugador' && profesion != 'programador') {
            print("Opción incorrecta intente nuevamente");
            profesion = stdin.readLineSync();
        }
    }
}

for (var i = 0; i < listadoPersonas.length; i++) {
    listadoPersonas[i].mostrarInfo();
}

```

```
su nombre es Daniel,  
y la edad es 20,  
la profesion es programador  
  
La empresa en la que trabaja es Getware,  
El salario neto del programador es de 27056.0  
  
su nombre es Juan,  
y la edad es 18,  
la profesion es jugador  
  
el equipo es Once,  
la posicion es delantero,  
la cantidad de goles es 5  
El jugador es Titular: No  
  
su nombre es Isabela,  
y la edad es 19,  
la profesion es programador  
  
La empresa en la que trabaja es setWare,  
El salario neto del programador es de 18245.0  
  
su nombre es Diego,  
y la edad es 25,  
la profesion es jugador  
  
el equipo es Nacional,  
la posicion es Medio campista Ofensivo,  
la cantidad de goles es 3  
El jugador es Titular: No
```

- d. El método **esTitular** de la clase Futbolista, será un método que devolverá un valor booleano que se determinará de la siguiente manera: Si el futbolista tiene más de 5 goles será titular del equipo, si tiene menos no lo podrá ser. El sistema deberá mostrar todos los datos del futbolista y si será titular o no. Tomar un pantallazo de los resultados.

```
bool esTitular() {  
    if (_cantidadGoles > 5) {  
        return true;  
    } else {  
        return false;  
    }  
}
```

```

    void mostrarInfo() {
        super.mostrarInfo();
        print("""
el equipo es $_equipo,
la posicion es $_posicion,
la cantidad de goles es $_cantidadGoles
El jugador es Titular: ${esTitular() ? 'SI' : 'No'}
"""); // ${esTitular() ? 'SI' : 'No'} es igual a ${esTitular()==true ? 'SI' : 'No'}
    }

```

- e. El método **obtenerSalarioNeto** de la clase Programador, será un método que devolverá el salario neto de un programador, este valor será calculado de la siguiente manera: al salario actual se le restará el 11% del salario. El sistema deberá mostrar todos los datos del programador y el salario neto. Tomar un pantallazo de los resultados.

```

    obtenerSalarioNeto() {
        double salarioNeto;
        salarioNeto = _salario * 0.11;
        return _salario -=salarioNeto;
    }
    void mostrarInfo() {
        super.mostrarInfo();
        print("""
La empresa en la que trabaja es $_empresa,
El salario neto del programador es de $_salario
""");
    }

```

- f. Crear un proyecto en Dart y realizar lo siguiente:

- a. Construir las clases: Figura (clase padre), Circulo, Cuadrado y Triángulo aplicando los conceptos de herencia y polimorfismo.

```

abstract class Figuras {
    double calcularArea();
}

```

```
import 'Figuras.dart';

class Circulo extends Figuras {
  double _radio;

  // Constructor para inicializar el radio
  Circulo(this._radio);

  // Implementación del método para calcular el área del círculo
  @override
  double calcularArea() {
    return pi * pow(_radio, 2); //  $\pi * r^2$ 
  }

  @override
  String toString() {
    return 'Círculo con radio $_radio tiene un área de ${calcularArea()}';
  }
}
```

```
import 'Figuras.dart';

class Cuadrado extends Figuras {
  double _lado;

  Cuadrado(this._lado);

  @override
  double calcularArea() {
    return pow(_lado, 2).toDouble(); //  $\text{lado}^2$ 
  }

  @override
  String toString() {
    return 'Cuadrado con lado $_lado tiene un área de ${calcularArea()}';
  }
}
```



```
import 'Figuras.dart';

class Triangulo extends Figuras {
  double _base, _altura;

  Triangulo(this._base, this._altura);

  @override
  double calcularArea() {
    return (_base * _altura) / 2;
  }

  @override
  String toString() {
    return 'Triángulo con base $_base y altura $_altura tiene un área de ${calcularArea()}';
  }
}
```

- b. Se deben crear objetos de cada una de las clases. Tomar un pantallazo de los resultados

```
Figuras circulo = Circulo(5);
Figuras cuadrado = Cuadrado(4);
Figuras triangulo = Triangulo(3, 6); // Base = 3, Altura = 6
```

- c. Se debe permitir calcular el área de la figura específica

```
print(circulo);
print(cuadrado);
print(triangulo);
```