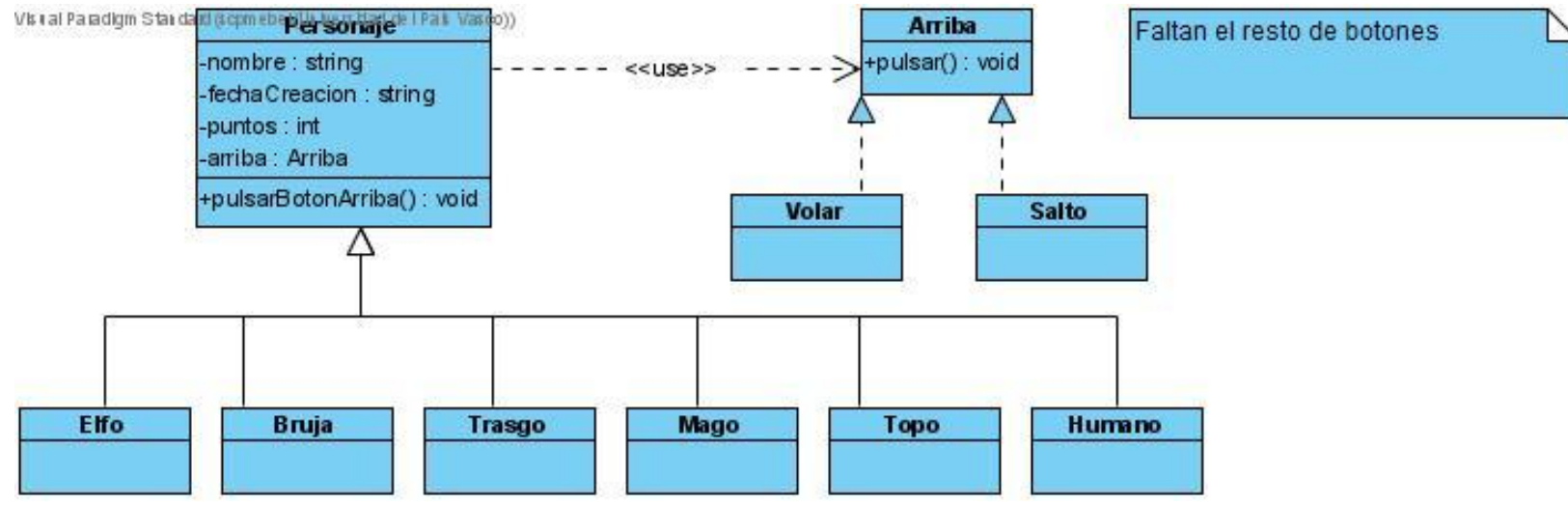


## DIAGRAMA DE CLASES (Videojuego (Solucion Strategy))



## CÓDIGO (Videojuego (Solucion Strategy))

```
public abstract class Personaje {
    private String nombre;
    private LocalDate fechaCreacion;
    private float puntos;
    private Arriba arriba;

    public Personaje(String nom, Arriba ar){
        nombre = nom;
        fechaCreacion = LocalDate.now();
        puntos = 0;
        arriba = ar;
    }
    public void pulsarBotonArriba(){
        arriba.pulsar();
    }
}
public class Bruja extends Personaje {
    public Bruja(String nom) {
        super(nom, new Volar());
    }
}
public class Humano extends Personaje {
    public Humano(String nom) {
        super(nom, new Salto());
    }
}
```

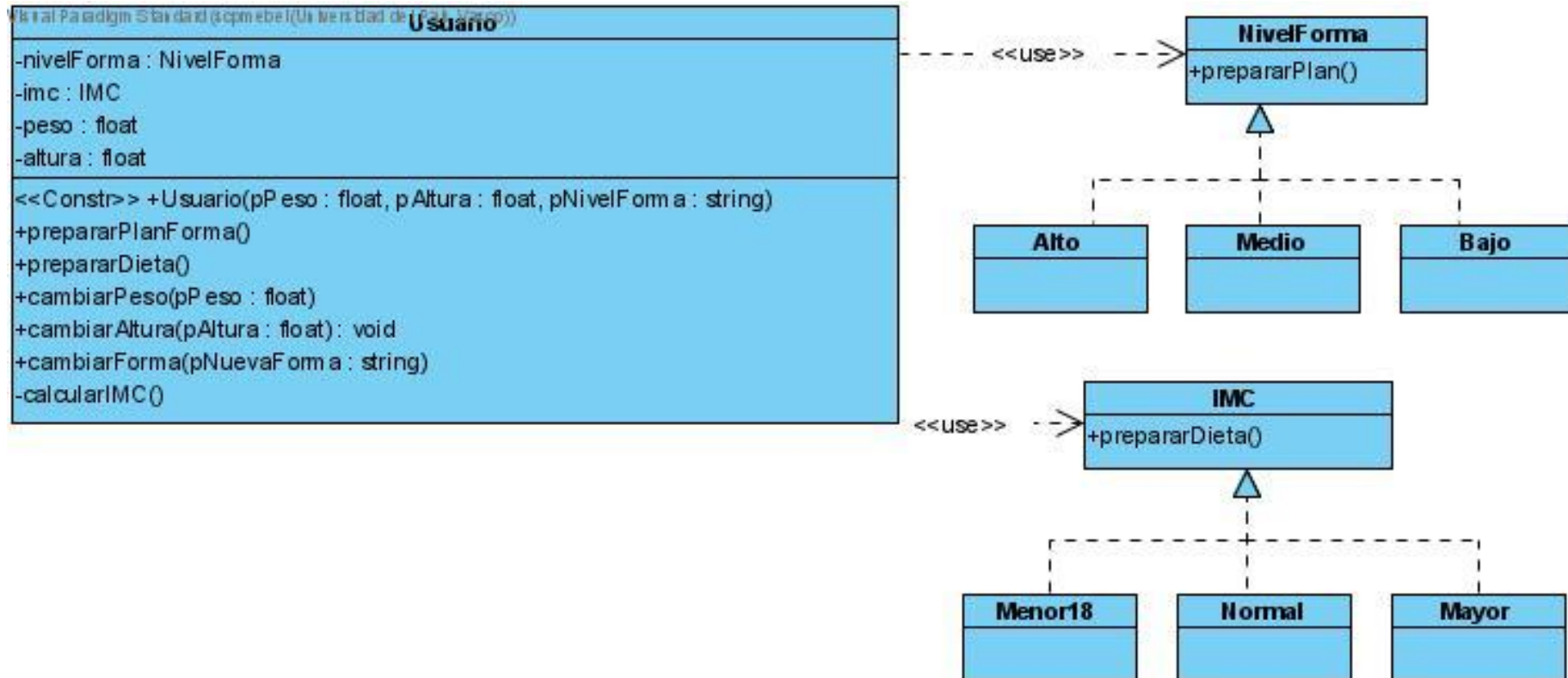
### STRATEGY

```
public interface Arriba {
    public void pulsar();
}

public class Volar implements Arriba {
    public void pulsar() {
        System.out.println("volar");
    }
}

public class Salto implements Arriba {
    public void pulsar() {
        System.out.println("saltar");
    }
}
```

## DIAGRAMA DE CLASES (Fitness Tracker (Solucion State))



## CÓDIGO (Fitness Tracker (Solucion State))

```
public class Usuario {
    private float altura;
    private float peso;
    private IMC imc;
    private NivelForma nivelForma;
    public Usuario(float pAltura, float pPeso, String pForma) {
        this.altura = pAltura;
        this.peso = pPeso;
        calcularIMC();
        cambiarForma(pForma);
    }
    public void prepararPlanForma() {
        nivelForma.prepararPlan();
    }
    public void prepararDieta() {
        imc.prepararDieta();
    }
    public void cambiarForma(String pNuevaForma){
        if (pNuevaForma.equals("alto")) {
            nivelForma = new Alto();
        } else if (pNuevaForma.equals("medio")) {
            nivelForma = new Medio();
        } else if (pNuevaForma.equals("bajo")) {
            nivelForma = new Bajo();
        }
    }
    public void calcularIMC() {
        float division = peso / altura;
        if (division > 25) {
            imc = new Mayor25();
        } else if (division > 18.5) {
            imc = new Normal();
        } else {
            imc = new Menor18();
        }
    }
}
```

```
public void cambiarPeso(float pPeso){
    this.peso = pPeso;
    calcularIMC();
}

public void cambiarAltura(float pAltura) {...}
}
```

```
public interface IMC {
    public void prepararDieta();
}
public class Menor18 implements IMC {
    public void prepararDieta() {
        System.out.println("dieta hipercalorica");
    }
}
public class Normal implements IMC {...}
public class Mayor25 implements IMC {...}
```

### STATE IMC

```
public interface NivelForma {
    public void prepararPlan();
}
public class Alto implements NivelForma {
    public void prepararPlan() {
        System.out.println("ejercicio para profesionales");
    }
}
public class Medio implements NivelForma {...}
public class Bajo implements NivelForma {...}
```

### STATE FORMA