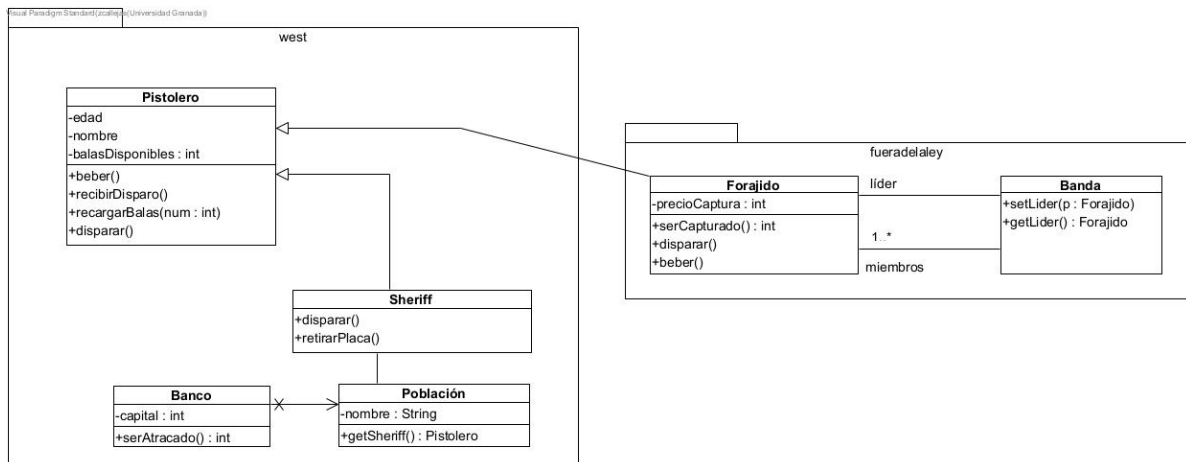


# CLASE HERENCIA Y POLIMORFISMO



## RUBY

```
module West
  class Pistolero
    def initialize(edad,nombre,balas)
      @edad = edad
      @nombre = nombre
      @balasDisponibles = balas
    end

    def beber
      puts "Estoy bebiendo"
    end

    def recibirDisparo
      puts "Me han diparado"
    end

    def recargarBalas(num)
      @balasDisponibles = @balasDisponibles + num
    end

    def disparar
      if(@balasDisponibles > 0)
        @balasDisponibles = @balasDisponibles - 1
        puts "pium"
      end
    end
  end
end
```

```
module West
  class Sheriff < Pistolero

    def initialize(edad,nombre,balas, poblacion)
      #Llama al initialize de superclase pasandole los parametros de ahora
    end
  end
end
```

```

        #Si pongo solo super, paso los mismos argumentos que arriba, si
pongo super          y los argumentos, se le pasa los argumentos que yo
indique.

        super(edad,nombre,balas)
        @poblacion = poblacion
    end
    def disparar
        puts "Alto"
        super # ponemos super para acceder al metodo disparar del pistolero
    end

    def retirarPlaca
        puts "Me dejan sin placa"s
    end
end
end
end

```

```

require_relative "../west/pistolero.rb"
module fueradelrey
    class Forajido < West::Pistolero
        def initialize(edad,nombre,balas,precioCaptura,banda)
            super(edad,nombre,balas)
            @precioCaptura = precioCaptura
            @miembros = banda
        end

        def serCapturado
            puts "Me han capturado"
        end

        def disparar
            puts "Disparo a matar"
            super
        end

        def beber
            puts "Bebo lo que pillo"
        end
    end
end
end

```

```

require("../west/Pistolero")
require("../west/Sheriff")

elbueno = West::Pistolero.new(25,"David",4)
sheriff = West::Sheriff.new(26,"Pepe",4,45)

elbueno.beber #ejecuta el metodo beber de pistolero
sheriff.beber # ejecuta el metodo beber del pistolero porque hereda el metodo
elbueno.retirarPlaca #el pistolero no conoce este metodo

# El siguiente codigo funciona perfectamente.
# Hace: alto piumm (sheriff) y despues piumm (pistolero)

```

```
# Anonimo es un polimorfismo, esto significa que la misma variable en diferentes
posiciones del código pueden ser objetos diferentes. Al principio es un sheriff y
luego es un pistolero
anonimo = sheriff
anonimo.disparar
anonimo = elbueno
anonimo.disparar

# Creo un array con todos
pistoleros = Array.new
pistoleros << elbueno
pistoleros << sheriff

# Se ponen todos a disparar
pistoleros.each do |p|
  p.disparar
end
```

## JAVA

---

```
public class Sheriff extends Pistolero{
    // para indicar que estoy redefiniendo un metodo de la clase de la que
hereda
    @Override
    public void disparar(){
        System.out.println("Alto");
        super.disparar();
    }
}
```