



CFP César Manrique

Breve repaso UML

**CFGS DAW – Desarrollo web en entorno
servidor**



Diagramas de clase

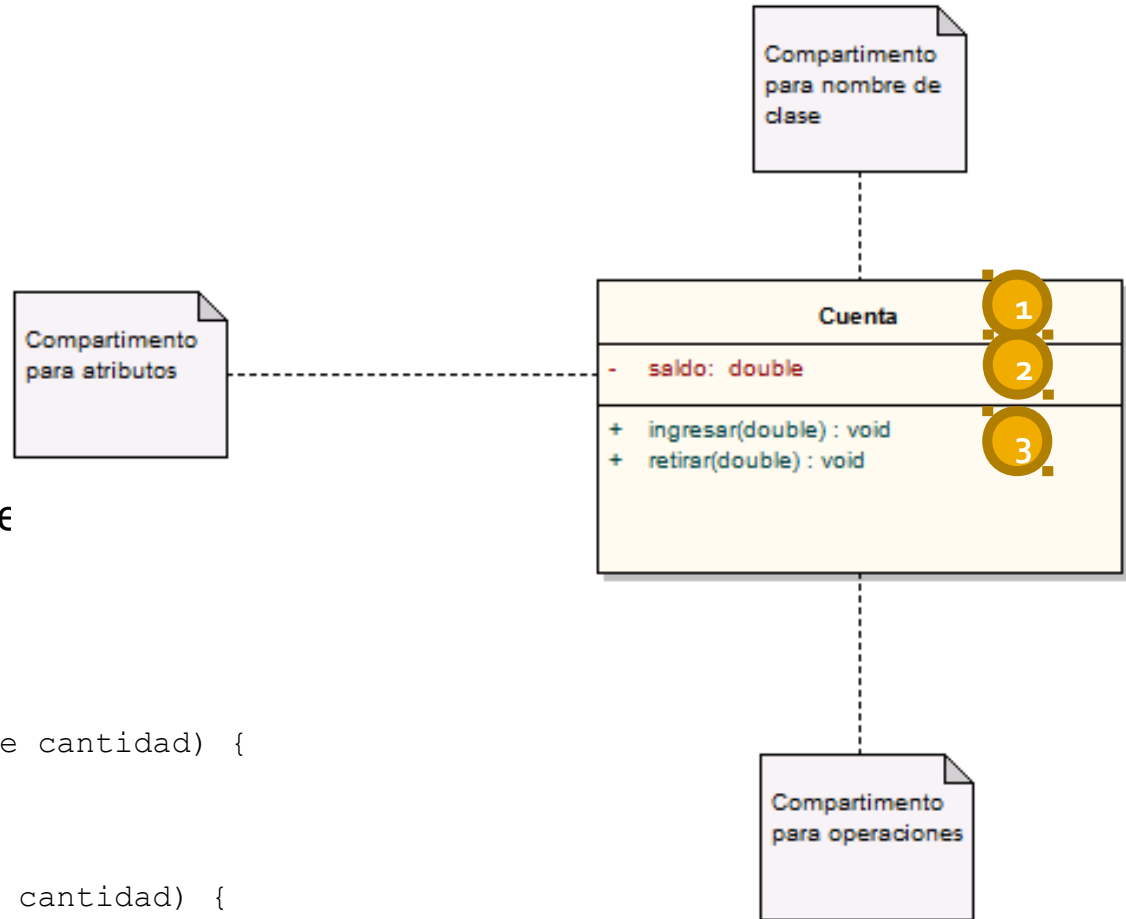
Employee
<pre>-name:String -payRate:double -EMPLOYEE_ID:int -<u>nextID:int</u> +<u>STARTING PAY RATE:double</u></pre>
<pre>+Employee(String) +Employee(String, double) +getName():String +getEmployeeID():int +getPayRate():double +changeName(String):void +changePayRate(double):void +<u>getNextID():int</u></pre>



Clasificadores

- Compartimentos
 - Nombre (1)
 - Atributos (2)
 - Operaciones (3)
- Estereotipos
 - Sobre el nombre Ej: «Single

```
public class Cuenta {  
  
    private double saldo;  
  
    public void ingresar (double cantidad) {  
        // ....  
    }  
  
    public void retirar (double cantidad) {  
        // ....  
    }  
}
```





Visibilidad de atributos

+ public

- private

protected

~ package

Variable miembro de instancia o clase

variableInstancia

variableClase (subrayada)



Atributos

Atributo de instancia, tipo double, acceso privado, valor por defecto 5.0

- saldo: double = 5.0

```
private double saldo = 5.0;
```

Atributo de instancia, tipo String, acceso público

+ numero_cuenta : String

```
public String numeroCuenta;
```

Atributo de instancia, tipo String, acceso protegido, multiplicidad 2

impositores : String[1..2]

```
protected String[] impositores;
```

Atributo de instancia calculado de tipo booleano, acceso privado

-/ numeros_rojos : boolean

```
private volatile boolean numerosRojos;
```

Atributo de clase de tipo integer

- numero_accesos : int

```
private static int numeroAccesos;
```

Variables estáticas (de clase) se indican mediante subrayado



Operaciones

Método de acceso público ingresar: parámetro de entrada importe de tipo entero

+ ingresar(importe: double): void

```
public void ingresar(double importe) { ... }
```

Método de acceso público obtenerSaldo: sin parámetros, retorno de tipo double

+ obtenerSaldo() : double

```
public double obtenerSaldo() { ... }
```

Constructor privado

- GestorCuentas()

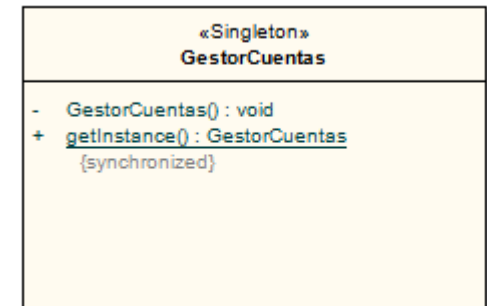
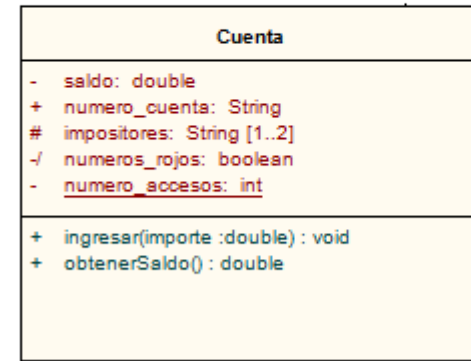
```
private GestorCuentas() { ... }
```

Método estático getInstance: acceso público, sincronizado, sin parámetros, retorno de tipo GestorCuentas

+ getInstance() : GestorCuentas {synchronized}

```
public static synchronized GestorCuentas getInstance() { ... }
```

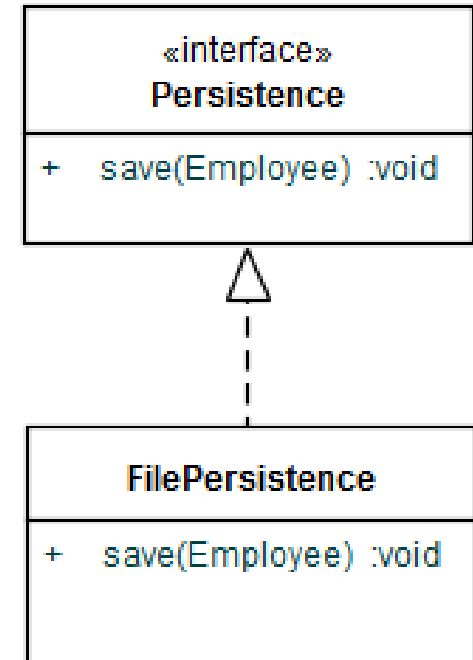
numeros_rojos=true si saldo <0





Realización **implements** (implementación de interfaz)

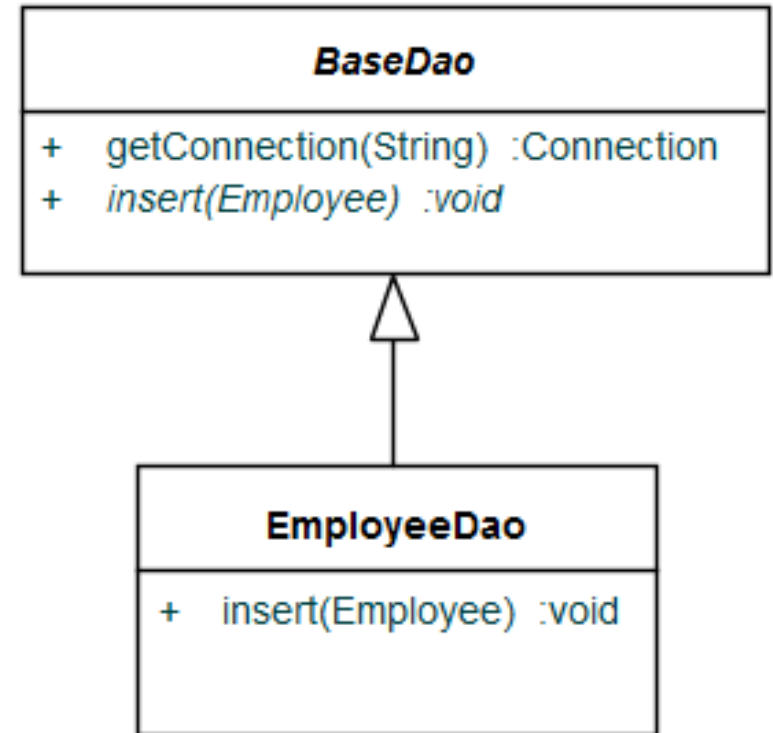
```
public interface Persistence {  
  
    void saveEmployee(Employee emp);  
}  
  
public class FilePersistence implements Persistence {  
  
    @Override  
    public void saveEmployee(Employee emp) {  
        //...  
    }  
}
```





Herencia **extends**

```
public abstract class BaseDao {  
  
    public Connection getConnection(String url) {  
        // ....  
        return null;  
    }  
  
    public abstract void insert(Employee emp);  
}  
  
public class EmployeeDao extends BaseDao {  
  
    @Override  
    public void insert(Employee emp) {  
        //...  
    }  
}
```





PREGUNTAS

