

Anexo: Código Ccuenta con Comentarios.

```
package cuentas;

/**
 * Clase para obtener objetos del tipo cuenta
 * @author Fatima.Atienza
 */

public class CCuenta {

    /*Esto es un comentario para hacer un commit*/

    /**
     * Atributo nombre. Declaración de la variable donde almacenar el nombre
     */
    private String nombre;

    /**
     * Atributo cuenta. Declaración de la variable donde almacenar la cuenta
     */
    private String cuenta;

    /**
     * Atributo saldo. Declaración de la variable para almacenar el saldo
     */
    private double saldo;

    /**
     * Atributo tipoInteres. Declaración de la variable del tipo de interés
    aplicado
     */
    private double tipoInterés;

    /** Método constructor sin parámetros*/
    public CCuenta()
    {
    }

    /** Método constructor con parámetros
     * @param nom nombre
     * @param cue cuenta
     * @param sal saldo
     * @param tipo tipoInterés
     */
    public CCuenta(String nom, String cue, double sal, double tipo)
    {
        setNombre(nom);
        setCuenta(cue);
        setSaldo(sal);
    }
}
```

```
}
```

```
/** Método que devuelve el saldo de la cuenta.
```

```
 * @return saldo en la cuenta
```

```
 */
```

```
public double estado()
```

```
{
```

```
    return getSaldo();
```

```
}
```

```
/** Método para ingresar una cantidad en una cuenta por parámetro
```

```
 * @param cantidad cantidad a ingresar
```

```
 * @throws Exception cantidad negativa
```

```
 */
```

```
public void ingresar(double cantidad) throws Exception
```

```
{
```

```
    if (cantidad<0)
```

```
        throw new Exception("No se puede ingresar una cantidad negativa");
```

```
    setSaldo(getSaldo() + cantidad);
```

```
}
```

```
/** Método para retirar una cantidad de la cuenta
```

```
 * @param cantidad cantidad a retirar
```

```
 * @throws Exception cantidad negativa
```

```
 */
```

```
public void retirar(double cantidad) throws Exception
```

```
{
```

```
    if (cantidad <= 0)
```

```
        throw new Exception ("No se puede retirar una cantidad negativa");
```

```
    if (estado()< cantidad)
```

```
        throw new Exception ("No se hay suficiente saldo");
```

```
    setSaldo(getSaldo() - cantidad);
```

```
}
```

```
/** Método para obtener el nombre
```

```
 * @return nombre de la cuenta
```

```
 */
```

```
    private String getNombre() {
```

```
        return nombre;
```

```
    }
```

```
    /** Método para modificar el nombre pasado
```

```
 * @param nombre nombre a modificar
```

```
 */
```

```
    private void setNombre(String nombre) {
```

```
        this.nombre = nombre;
```

```
    }
```

```
/** Método para obtener la cuenta
 * @return cuenta datos de la cuenta
 */
private String getCuenta() {
    return cuenta;
}
```

```
/** Método para modificar la cuenta pasada
 *
 * @param cuenta cuenta a modificar
 */
private void setCuenta(String cuenta) {
    this.cuenta = cuenta;
}
```

```
/** Método para obtener el saldo de la cuenta
 * @return saldo de la cuenta
 */
private double getSaldo() {
    return saldo;
}
```

```
/** Método para modificar el saldo de la cuenta pasado
 * @param saldo saldo a modificar
 */
private void setSaldo(double saldo) {
    this.saldo = saldo;
}
```

```
/** Método para obtener el tipo de interés
 * @return TipoInterés de la cuenta
 */
private double getTipoInterés() {
    return tipoInterés;
}
```

```
/** Método para modificar el tipo de interés pasado
 * @param tipoInterés tipo a modificar
 */
private void setTipoInterés(double tipoInterés) {
    this.tipoInterés = tipoInterés;
}
```

```
}
```