



Apache  
**NetBeans IDE**

El banco “de la Plaza” ha decidido realizar una nueva aplicación para sus clientes que mantienen su dinero depositado. Los retiros de dinero y los depósitos entregarán el saldo resultante de cada operación.

Para esto se requiere crear un sistema que almacene información de sus usuarios y sus cuentas.

- De los usuarios se requiere almacenar el Rut, nombre, clave de 4 dígitos.
- Por otra parte, la cuenta debe tener un número de identificador, saldo (cantidad de dinero que mantiene), un campo que almacena tipo de cuenta (cuenta corriente o cuenta vista), si su cuenta de usuario es empresa o cuenta persona.

Aparte de ello se requiere tener ciertos requisitos a cumplir:

- El nombre del usuario no debe venir en blanco
- El tipo de cuenta debe ser únicamente Corriente o Vista
- La clave numérica debe tener 4 dígitos y puede comenzar con ceros
- Validar la cuenta de usuario.
- Cuenta empresas puede girar hasta 1 millón si es que les alcanza el saldo.
- Cuenta personas pueden girar hasta 200 mil si es que les alcanza el saldo.

## PryBanco.java

```
package pry03.banco;

public class Pry03Banco {

    public static void main(String[] args) {
        // creamos entidades de la clase fuerte USUARIO
        Usuario usu1 = new Usuario();
        usu1.setRut("111-1");
        usu1.setNombre("Alonso Cuevas");
        usu1.setClave("0123");
        //System.out.println(usu1);

        Cuenta c1 = new Cuenta();
        // asignar inmediatamente el enlace
        c1.setUsuario(usu1); // enlace fisico
        c1.setId(1001);
        c1.setSaldo(75000);
        c1.setTipo('V');
        c1.setEmpresa(false);
        System.out.println(c1);
        // imprimir el rut del dueño de la cuenta
        System.out.println("rut de c1 " + c1.getUsuario().getRut()); /*
muestra una ruta de navegacion. primero entra a Usuario y luego
al dato requerido que en este
caso es rut */
        c1.getUsuario().setRut("222-2");// modificamos el rut del usuario

        Cuenta c2 = new Cuenta();
        c2.setUsuario(usu1);
        c2.setId(20001);
        c2.setSaldo(250000);
        c2.setTipo('C');
        c2.setEmpresa(true);
        System.out.println(c2);

        c1.listarDatos(); // Método listar datos:
        c2.listarDatos(); // Método listar datos:

        System.out.println("datos correctos?: " +
c1.getUsuario().validar("222-2", "0123"));

        System.out.println("Nuevo saldo: " + c1.depositar(15000));//
imprsión de saldo total + un deposito efectuado

        System.out.println("Max riro c1: " + c1.maxGiro());
        System.out.println("Max riro c2: " + c2.maxGiro());
    }
}
```

## Cuenta.java

```
package pry03.banco;

public class Cuenta {
    private int id;
    private int saldo;
    private char tipo; // C V
    private boolean empresa; // tipo empresa o tipo persona

    private Usuario usuario; // FK esto enlaza con Usuario.java
    /* en el constructor vacio
    enlazamos fidicamente los objetos */

    public Cuenta() {
        // atributo usuario se enlaza con la clase
        usuario = new Usuario();
        /* cada vez que se crea una nueva cuenta la va a
        enlazar con un usuario */
    }

    public Cuenta(int id, int saldo, char tipo, boolean empresa, Usuario
usuario) {
        this.id = id;
        this.saldo = saldo;
        this.tipo = tipo;
        this.empresa = empresa;
        this.usuario = usuario;
    }

    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    public int getSaldo() {
        return saldo;
    }

    public void setSaldo(int saldo) {
        this.saldo = saldo;
    }

    public char getTipo() {
        return tipo;
    }

    public void setTipo(char tipo) {
        if(tipo == 'C' || tipo == 'V') {
            this.tipo = tipo;
        }
    }
}
```





```
}
```

## Usuario.java

```
package pry03.banco;

public class Usuario {
    private String rut;
    private String nombre;
    private String clave; // permitir solo 4 digitos

    public Usuario() {
    }

    public Usuario(String rut, String nombre, String clave) {
        this.rut = rut;
        this.nombre = nombre;
        this.clave = clave;
    }

    public String getRut() {
        return rut;
    }

    public void setRut(String rut) {
        this.rut = rut;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        if(nombre.length() == 0){
            this.nombre = nombre;
        } else {
            this.nombre = null;
            System.out.println("ERROR, nombre no debe estar vacio");
        }
    }

    public String getClave() {
        return clave;
    }

    public void setClave(String clave) {
        // regla de negocio, acepta solo 4 digitos
        if(clave.length() == 4){
            // validar que todos los caracteres sean numeros
            // opcion 1: recorrer y comparar todos los char
            boolean soloDigitos = true;

```



## Output – pry03Banco (run)

```
run:
ERROR, nombre no debe estar vacio
Cuenta{id=1001, saldo=75000, tipo=V, empresa=false,
usuario=Usuario{rut=111-1, nombre=null, clave=0123}}
rut de c1 111-1
Cuenta{id=20001, saldo=250000, tipo=C, empresa=true,
usuario=Usuario{rut=222-2, nombre=null, clave=0123}}
```

DATOS CUENTA

```
-----
null 222-2 75000 CUENTA VISTA
```

```
#####
```

DATOS CUENTA

```
-----
null 222-2 250000 CUENTA CORRIENTE
```

```
#####
```

```
datos correctos?: true
Nuevo saldo: 90000
Max riro c1: 90000
Max riro c2: 250000
BUILD SUCCESSFUL (total time: 0 seconds)
```