**Pontificia Universidad Javeriana**

**Introducción a Sistemas Distribuidos.**

**Documentación Segunda entrega Proyecto.**

**Presentado por: Daniel Esteban Valero, Juan Felipe Vales.**

**DOCUMENTACIÓN**

**Consola de operaciones (Cliente)**

**Cliente**

Clase en java que representa a un cliente en el sistema que se conecta con el coordinador (Banco), para realizar las operaciones.

* **Atributos:**
  + c: crea un nuevo cliente en el sistema.
  + reg: Crea un registro que conecta con las funciones remotas de ahorro.
  + reg2: Crea un registro que conecta con las funciones remotas de crédito.
  + reg3: Crea un registro que conecta con las funciones remotas de visa.
  + reg4: Crea un registro que conecta con las funciones remotas de MasterCard.
  + ad: conexion de registro con las funciones remotas de ahorro.
  + cd: conexion de registro con las funciones remotas de crédito.
  + vd: conexión de registro con las funciones remotas de visa.
  + md: conexión de registro con las funciones remotas de MasterCard.
* **Métodos:**
  + connectRemote(): clase que llama las funciones remotas con el fin de obtener las operaciones del banco.

**Coordinador (Banco)**

Clase en java que representa a la conexión que debe usar el cliente para poder conectarse con un servidor y recibe la información del servidor que luego es transmitida al cliente.

* **Atributos:**
  + listaParticipantes: crea una lista para guardar a múltiples clientes.
  + listaTransacciones: crea una lista de las transacciones enviadas por los múltiples clientes.
* **Métodos:**
  + openTransaccion(): función que recibe los parametros enviados por el cliente para enviarlos a los diferentes servidores y realizar las operaciones del banco, dependiendo de los parámetros enviados existen diferentes openTransaccion.

**Servidores**

**ServerVisaAhorro**

Clase en java que representa el servidor con los getters y setters de una tarjeta Visa de ahorro.

* **Atributos:**
  + saldoInicial: atributo que brinda un saldo inicial a los clientes del sistema.
  + reg: crea un registro para que pueda ser localizado por el banco y poder utilizar las funciones de este servidor.
* **Métodos:**
  + getBalance(): obtiene el saldo actual del cliente para su tarjeta de Visa de ahorro.
  + setBalance(): actualiza el saldo de una persona dependiendo del monto sacado de un cliente.

**ServerMCAhorro**

Clase en java que representa el servidor con los getters y setters de una tarjeta MasterCard de ahorros.

* **Atributos:**
  + saldoInicial: atributo que brinda un saldo inicial a los clientes del sistema.
  + reg: crea un registro para que pueda ser localizado por el banco y poder utilizar las funciones de este servidor.
* **Métodos:**
  + getBalance(): obtiene el saldo actual del cliente para su cuenta de MasterCard de ahorros.
  + setBalance(): actualiza el saldo de una persona dependiendo del monto sacado de un cliente.

**ServerVisa**

Clase en java que representa el servidor con los getters y setters de una tarjeta de credito Visa.

* **Atributos:**
  + Deuda: atributo que brinda la deuda de la tarjeta actual.
  + Cupo: atributo que el cupo total actual de la tarjeta actual.
  + reg: crea un registro para que pueda ser localizado por el banco y poder utilizar las funciones de este servidor.
* **Métodos:**
  + getDeudaVisa(): obtiene la deuda actual del cliente para su tarjeta Visa.
  + setDeudaVisa(): actualiza la deuda para una persona dependiendo del monto pagado de su cuenta de ahorros o crédito.
  + getCupoVisa(): obtiene el cupo actual del cliente para su tarjeta Visa.
  + setCupoVisa(): actualiza el cupo total de tarjeta para una persona dependiendo del monto pagado de su cuenta de ahorros o crédito.

**ServerMasterCard**

Clase en java que representa el servidor con los getters y setters de una tarjeta de crédito MasterCard.

* **Atributos:**
  + Deuda: atributo que brinda la deuda de la tarjeta actual.
  + Cupo: atributo que el cupo total actual de la tarjeta actual.
  + reg: crea un registro para que pueda ser localizado por el banco y poder utilizar las funciones de este servidor.
* **Métodos:**
  + getDeudaMasterMC(): obtiene la deuda actual del cliente para su tarjeta MasterCard.
  + setDeudaMC(): actualiza la deuda para una persona dependiendo del monto pagado de su cuenta de ahorros o crédito.
  + getCupoMC(): obtiene el cupo actual del cliente para su tarjeta Visa.
  + setCupoMC(): actualiza el cupo total de tarjeta para una persona dependiendo del monto pagado de su cuenta de ahorros o crédito.

**Interfaces**

**MCAhorro**

* **Métodos:**
  + getBalance(): permite llamar remotamente la función getBalance.
  + setBalance(): permite llamar remotamente la función setBalance.

**VisaAhorro**

* **Métodos:**
  + getBalance(): permite llamar remotamente la función getBalance.
  + setBalance(): permite llamar remotamente la función setBalance.

**MasterCard**

* **Métodos:**
  + getCupoVisa(): permite llamar remotamente la función getCupoVisa.
  + setCupoVisa(): permite llamar remotamente la funcion setCupoVisa.
  + getDeudaVisa(): permite llamar remotamente la función getDeudaVisa.
  + setDeudaVisa(): permite llamar remotamente la función setDeudaVisa.

**Visa**

* **Métodos:**
  + getCupoMC(): permite llamar remotamente la función getCupoMC.
  + setCupoMC(): permite llamar remotamente la funcion setCupoMC.
  + getDeudaMC(): permite llamar remotamente la funcion getDeudaMC.
  + setDeudaMC(): permite llamar remotamente la funcion setDeudaMC.