

Diseñar un sistema de “Billetera Electrónica o Virtual”, gestionando los procesos que se detallan, con el objetivo de facilitar en forma online, operaciones comerciales entre usuarios registrados y comercios adheridos, vinculados con bancos según tarjetas de débito y de crédito.

Dicho sistema pensado desde la gestión de la empresa Billetera Electrónica (dado que usuarios, comercios y bancos tendrán generadas Apps para ingresar a este sistema)

Se propone trabajar con 5 archivos:

C:\ayed\tp3\bancos.dat (*archivo sin ningún orden*)
C:\ayed\tp3\comercios.dat (*ordenado por código de comercio*)
C:\ayed\tp3\cuentas-virtuales.dat (*archivo sin ningún orden*)
C:\ayed\tp3\usuarios.dat (*archivo sin ningún orden*)
C:\ayed\tp3\movimientos.dat (*ordenado por fecha en forma ascendente*)

Se solicita que el sistema ofrezca el siguiente menú de opciones, pensado en 3 categorías, por un lado están las entidades bancarias desde donde se respaldarán las tarjetas, por otra parte están los comercios adheridos, y por último los usuarios que al iniciar sesión podrán disponer de sus cuentas con sus saldos utilizando diferentes tarjetas:

1. **Bancos**
2. **ABM de Comercios Adheridos**
3. **Usuarios**
 - a. **Inicio sesión**
 - b. **Cuentas**
 - c. **Envíos de Dinero**
 - d. **Compras en comercios**
 - e. **Movimientos**

La aplicación requiere que el usuario esté bancarizado. Podrá gestionar cuentas con tarjetas de Débito, o de Crédito de diferentes entidades.

Detalle de las opciones del menú:

1. **Bancos:** Esta opción listará primero todas las entidades bancarias que ya están cargadas en el sistema en bancos.dat, y permitirá ingresar nuevos bancos. Por cada nuevo ingreso se deberá registrar el código del banco y el nombre de la entidad.
2. **ABM** significa altas, bajas y modificaciones en archivos, en este caso de **comercios**
 - a) **Alta:** se genera un nuevo registro en el archivo comercios.dat. El nombre y el cuit ingresan por teclado, y el código es consecutivo secuencial a partir del último ingresado.
 - b) **Baja:** Se solicita el CODIGO DE COMERCIO a dar de baja, de encontrarse en el archivo, se actualiza el campo ESTADO a false.
 - c) **Modificación:** Se solicita el CODIGO DE COMERCIO y se podrán modificar todos los campos, menos su código. (recordar que un campo booleano no se puede leer – encontrar la forma para determinar si se desea modificar).

3. **Usuarios:** puede ingresar por primera vez o de forma rutinaria. Por tal motivo primero deberá iniciar sesión.
 - a) **Inicio:** se solicita el ingreso del DNI del usuario, se validará la existencia en el archivo usuarios.dat. Si existe, se le pide la contraseña y se valida que sea la correcta para dejarlo trabajar (tiene hasta 3 intentos para lograrlo, pasados todos los intentos fallidos no podrá acceder a los puntos b) en adelante). Si el usuario no existe en el archivo, se le notificará que deberá crear una clave y se le dará de alta en dicho archivo como nuevo usuario solicitándole su apellido y nombre y su mail.
Desde el punto b) en adelante solo podrá acceder si el punto a) se ejecutó correctamente.
 - b) **Cuentas:** muestra para el DNI que se acaba de registrar, todas las tarjetas que este usuario dispone, con sus saldos (máximo 5), y además muestra su saldo-billetera que es el dinero que dispone en efectivo. Si no tiene cuenta creada, se le deberá permitir generar este nuevo registro. Cada DNI, representa un único record en este archivo.
 - c) **Envíos:** permite el envío de dinero a otro usuario que posea la misma Billetera virtual. Mostrando el DATE() de la máquina, se solicita el DNI de usuario que recibirá el dinero, y que monto enviará. Se deberá validar que el usuario de destino exista, que el usuario que envía disponga de saldo en saldo-billetera. Si todo es correcto, el movimiento quedará registrado en el archivo movimientos.dat, el saldo del usuario que transfiere debe quedar reducido en el saldo-billetera, y el saldo del usuario que recibió el dinero debe quedar aumentado en su respectivo saldo-billetera.
 - d) **Compras:** Se solicita el código de comercio donde se realiza la compra, verificar la existencia del mismo, de existir mostrar la denominación, de lo contrario informar 'negocio no adherido'. A continuación, mostrar las cuentas vinculadas del usuario, ingresar el código del banco, el tipo de tarjeta y el monto a abonar (solo se podrá usar una tarjeta por cada compra). Pedir confirmación (implica validar todos los datos ingresados) y en caso afirmativo actualizar los archivos correspondientes de movimientos y usuarios. Emitir por pantalla el comprobante de compra, con un formato que contenga: Banco – Comercio– Fecha operación – monto a debitar.
 - e) **Movimientos:** Ofrece la consulta de los movimientos del usuario. Se deben ingresar dos fechas para establecer el rango a mostrar. Exhibir todos los movimientos de dicha cuenta entre las fechas establecidas.

Formato mínimo de registros de cada archivo (se podrán incorporar más campos)

USUARIOS

dni: string [8]
contraseña: integer
ape-nom: string [30]
mail: string [40]

CUENTAS-VIRTUALES

dni: string [8]
cuenta_virtual: array [1..5] of record (con los siguientes campos)
 cod-ban: integer
 tipo-tar: char (D: Débito / C:Crédito)
 saldo-x-tarjeta: real
 saldo_billetera: real

MOVIMIENTOS

Dni: string [8]
cod-ban: integer
tipo-tar: char (D: Débito / C:Crédito)
Importe: real
tipo-movi: char ('C' *Compras*, 'E' *Envío*)
día, mes, año: word
cod-com: integer (código de comercio)
dni-otro-usuario: string[8]

COMERCIOS

cod-com: integer (*secuencial consecutivo desde el ultimo ingresado*)
nombre: string [30]
cuit: string [12]
estado: boolean

BANCOS

cod-ban: integer; (código de banco)
nombre: string [30]