



Facultad de Ingeniería y Ciencias Exactas – FICE

Carrera:

Licenciatura en Gestión de las Tecnologías de la Información

Materia:

Programación I - Martes turno noche

Docente:

Monasterio, Maria Julia

Alumnos:

Tobias Accorinti, 1184555

Nicolas Amarilla, 1164167

Enzo Acevedo, 1195724

Martina García Ríos, 1128267

Fecha de entrega: 25/11/2025

Empresa: HomeBanking

Documentación: Sistema de HomeBanking

Este documento describe brevemente la funcionalidad del código Python entregado en el Trabajo Práctico Obligatorio, el cual simula un Sistema de HomeBanking, con persistencia de datos de clientes y un registro de operaciones.

Funcionalidad General

El sistema permite a los usuarios crear cuentas, iniciar sesión, y realizar diversas operaciones bancarias con cuentas en pesos argentinos (ARS) y dólares estadounidenses (USD). Utiliza una estructura de diccionarios anidados para almacenar la información de los clientes, cuentas y saldos.

Estructura y Módulos

El código se organiza en funciones modulares que gestionan la interacción con el usuario, la lógica de las transacciones y la persistencia de datos.

- **Persistencia (`persistencia.py`):** Aunque no se incluye, se asume la existencia de un módulo que maneja el guardado (`guardar_clientes`) y la carga (`cargar_clientes`) de la lista de clientes (probablemente usando JSON o similar) para mantener los datos entre sesiones.
- **Registro de Operaciones:** Las funciones de transacción exitosas registran un movimiento en un archivo **CSV** (`operaciones.csv`) utilizando la función `registrar_operacion`.

Gestión de Usuarios y Acceso

Función	Descripción
<code>pedir_dni()</code>	Solicita y valida el DNI (numérico, 7 u 8 dígitos).
<code>registrar_usuario()</code>	Permite el alta de un nuevo cliente, validando que el DNI y el nombre de usuario sean únicos y que la contraseña cumpla requisitos de longitud, mayúsculas y números.
<code>iniciar_sesion()</code>	Autentica al usuario mediante DNI, usuario y contraseña.

Operaciones Bancarias

Las transacciones que modifican saldos (`depositar`, `extraer_dinero`, `transferir_entre_cuentas`) están diseñadas para devolver `True` o `False` para indicar si la operación fue exitosa, permitiendo que la función `main` registre el movimiento solo si la lógica de negocio se cumple (por ejemplo, tener saldo suficiente).

Función	Descripción
<code>crear_cuenta()</code>	Genera una nueva cuenta (Pesos o Dólares) con CBU/ALIAS aleatorio y saldo inicial \$0.00
<code>depositar()</code>	Acredita un monto a la cuenta seleccionada (Pesos o Dólares), con validación de monto positivo.
<code>extraer_dinero()</code>	Debita un monto, con validación de saldo suficiente y monto positivo.
<code>transferir_entre_cuentas()</code>	Permite la transferencia entre las cuentas ARS y USD del mismo cliente, aplicando una tasa de conversión fija (por defecto, 1000).
<code>cargar_celular()</code>	Debita de la cuenta en pesos para simular una carga de celular, solicitando el número y la compañía.
<code>consultar_saldo()</code>	Muestra el saldo actual de una cuenta específica.
<code>obtener_movimientos_usuario()</code>	Recupera el historial de transacciones del usuario desde <code>operaciones.csv</code> .

Interfaz y Utilidades

Función	Descripción
<code>limpiar_pantalla()</code>	Borra el contenido de la consola para una interfaz más limpia.
<code>pausar_y_volver()</code>	Pausa la ejecución hasta que se presione <code>ENTER</code> y luego limpia la pantalla. Incluye una condición para evitar el bloqueo durante la ejecución de tests automatizados (pytest).
<code>main()</code>	Contiene el ciclo principal del programa: gestiona la carga de datos, muestra los menús de inicio y operaciones, y llama a las funciones de transacción y utilidades.