****



INTRODUCCIÓN A LA PROGRAMACIÓN

**PRESENTACIÓN DEL CASO DE ESTUDIO FINAL**

**Presentado por:**

Andre Vargas Bermudez

Emilio Javier Gomez Lopez

Mauro Engelbert Delgado Saenz

Ulises Ramcess Chavez Rivera

Managua, julio de 2025

Índice

[**1. Introducción 3**](#_i63w54on4rf2)

[1.1. Planteamiento del problema 4](#_11sziend47wv)

[1.2. Objetivos 4](#_7f7awn9u6zb0)

[1.2.1. Objetivo general 4](#_70b92eb4o4d)

[1.2.2. Objetivos específicos 5](#_qh941qknxgw6)

[**2. Desarrollo 5**](#_yqxzu0byyr76)

[2.1. Fase de Análisis 5](#_84olqwwq6ua9)

[2.2. Fase de Algoritmo y Diagramas 6](#_b6ofscu19gie)

[2.3. Fase de Implementación 10](#_vwbk00j0ojcz)

[2.3.1. main.py 10](#_tpum3zffs3jt)

[2.3.2. sesion.py 11](#_240oabikhplk)

[2.3.3. sistema\_recargas.py 15](#_8oyot35lhfxp)

[**3. Ejecución del Programa 18**](#_6cy3vdq2zenv)

[**4. Conclusiones y Recomendaciones 20**](#_a2kt6cejyb09)

[4.1. Conclusiones 20](#_xrtjr29fm0ju)

[4.2. Recomendaciones 21](#_787lgtqcejkd)

[**5. Anexos 22**](#_bb5leqgbpmz2)

# Introducción

Nuestro proyecto consiste en la creación de un sistema automatizado diseñado específicamente para gestionar recargas telefónicas de forma sencilla, rápida y eficiente. Este sistema, al ser ejecutado, presenta un menú interactivo desde el cual el usuario puede seleccionar distintas opciones para realizar recargas a diferentes operadores móviles, introducir el número telefónico y el monto deseado, y obtener una confirmación del proceso.

El propósito de este sistema es simular, de manera funcional, una herramienta que responda a una necesidad común en el ámbito de las telecomunicaciones: la recarga de saldo. Este tipo de sistemas representa una solución accesible y práctica. Por ello, se ha trabajado en la lógica de programación y en la experiencia de usuario, procurando un uso intuitivo y directo que facilite el uso sin requerir conocimientos técnicos avanzados.

Durante el desarrollo del proyecto, se han aplicado conceptos clave como estructuras condicionales, funciones, validación de datos, manejo y registro de entradas y salidas.

Este proyecto refleja el compromiso del equipo con la creación de soluciones reales mediante el uso de la tecnología, demostrando cómo herramientas simples pueden automatizar procesos cotidianos y aportar valor en diferentes contextos.

## Planteamiento del problema

A pesar de los avances tecnológicos, aún existen sectores donde el acceso a plataformas robustas o sistemas especializados de recarga es limitado, ya sea por razones económicas, técnicas o de conectividad. En estos contextos, contar con una herramienta sencilla, automatizada y de fácil implementación que permita realizar recargas de forma segura y ordenada representa una solución viable y de alto valor.

Surge entonces la necesidad de desarrollar un sistema básico pero funcional que automatice el proceso de recargas telefónicas. Un sistema que, al ejecutarse, despliegue un menú con opciones claras, solicite los datos necesarios al usuario, valide la información ingresada y muestre confirmaciones del proceso de manera inmediata.

Nuestro proyecto responde precisamente a esta problemática, proponiendo una alternativa práctica que, aunque simple, refleja la aplicación real de los conocimientos adquiridos durante el primer semestre , y permite visualizar cómo la automatización puede impactar positivamente en tareas del día a día.

## Objetivos

### Objetivo general

Desarrollar un sistema interactivo y automatizado en el que se le facilite al usuario un menú intuitivo en el que pueda realizar una recarga de saldo de manera sencilla, eficiente y segura simplificando el proceso mediante una interfaz funcional.

### Objetivos específicos

* **Implementar un sistema de validación de datos** que asegure la correcta introducción del número telefónico y el monto de la recarga, evitando errores y garantizando la integridad de la transacción.
* **Proveer retroalimentación inmediata** al usuario, confirmando la realización exitosa de la recarga, con detalles como el número telefónico, el monto recargado y el operador seleccionado.
* **Facilitar el proceso** mediante el manejo adecuado de la información del usuario y el diseño de un sistema robusto que minimice errores y posibles fallos durante la transacción.

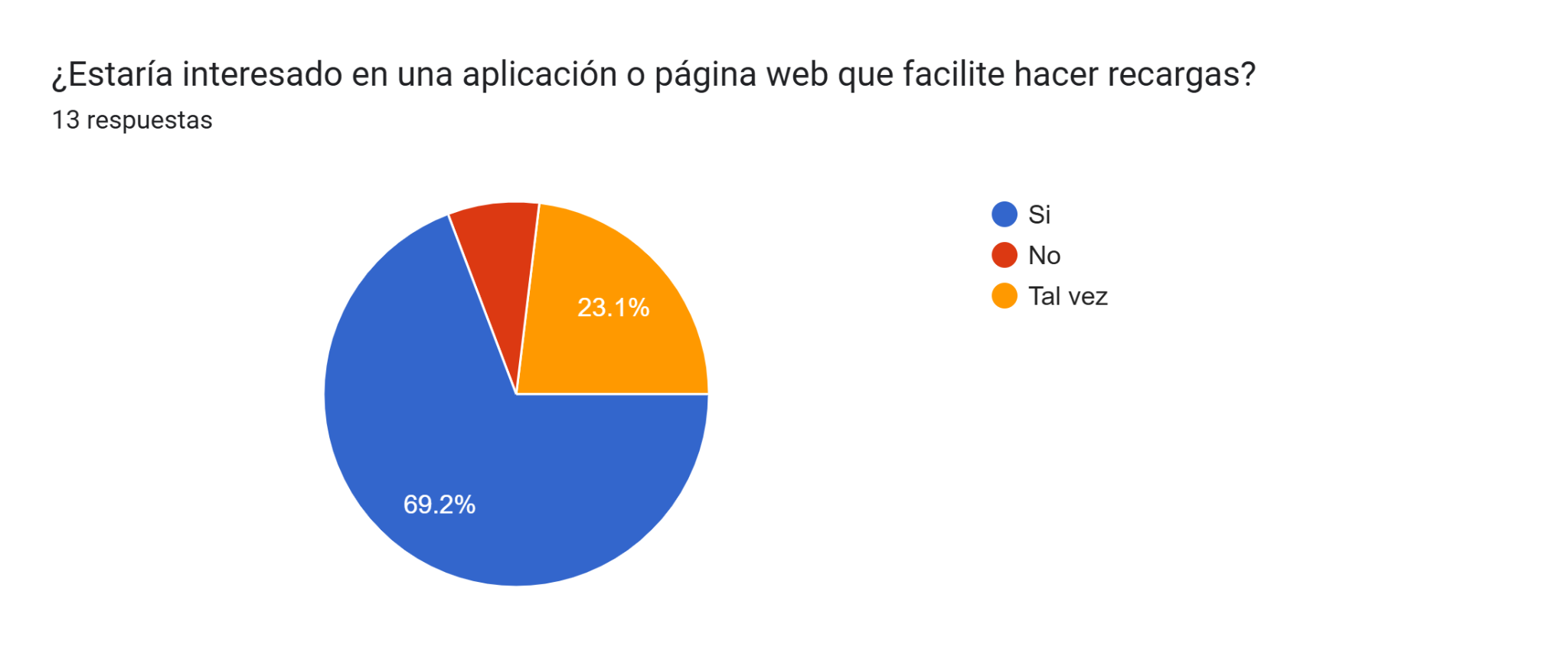
# Desarrollo

## Fase de Análisis

Se realizó una encuesta anónima a estudiantes de la UAM y se determinó que la mayoría de ellos no realizan recargas por medio de una aplicación, 35% lo hacen mediante tiendas y un 23% mediante mensajería o llamadas.

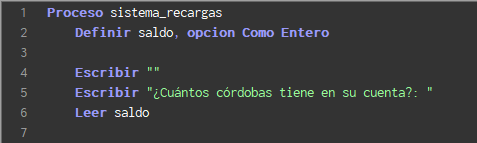


Esto demuestra que las aplicaciones todavía pueden ser más popularizadas como un método para realizar recargas, y el siguiente gráfico sólo lo confirma.

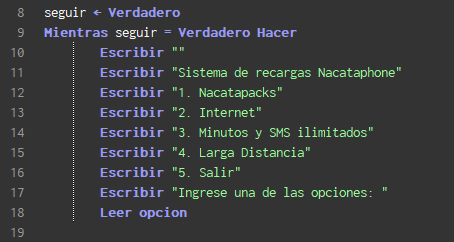
****

Con esta información podemos determinar que una aplicación es el siguiente paso necesario en esta área, ya que aunque una de las funciones de las aplicaciones de banca es la de hacer recargas, aún hay gente que prefiere otras opciones o no conoce que las aplicaciones de banca tiene dicha opción, posiblemente por falta de intuitividad.

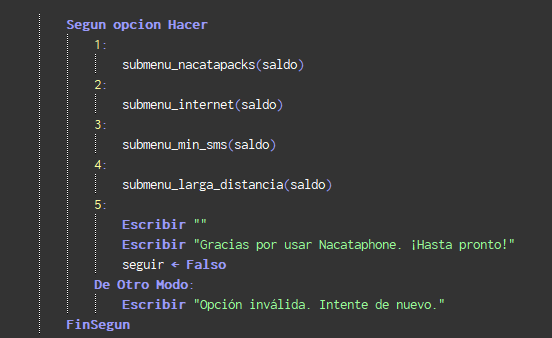
## Fase de Algoritmo y Diagramas



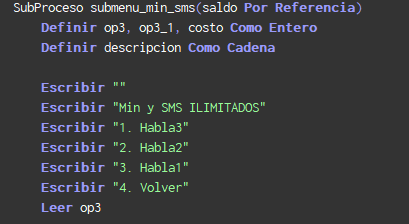
Primeramente en nuestro código le solicitamos al usuario que digite el saldo que tiene en su cuenta en córdobas y se utiliza “leer” para guardar la información en el sistema.



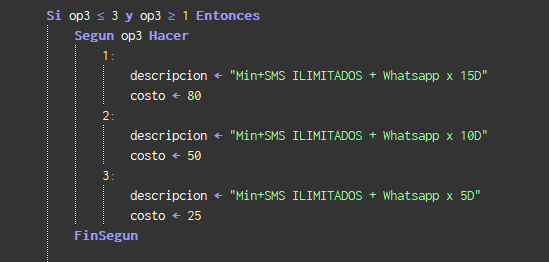
Una vez el usuario haya digitado su saldo y seleccionado seguir se abrirá un menú en el que puede seleccionar la opción de servicio que desea las cuales se enumeran del 1 al 5 y se lee el input del cliente. Según la función “seguir” sea verdadera.



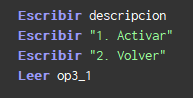
Después de leer el número del servicio que el cliente desea, utilizamos la función “según | hacer” para que el submenú indicado se abra según la opción seleccionada, en caso de que el cliente elija cinco se devolverá un mensaje de salida y no se seguirá con el subproceso ya que se determina que es falso.



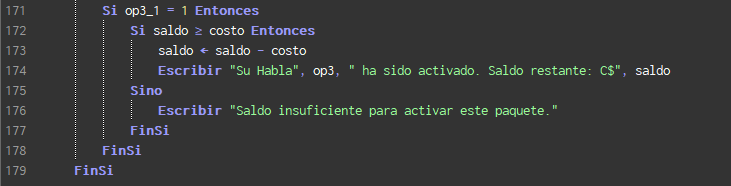
Al haber definido el subproceso que el cliente ha seleccionado se le redirigirá al mensaje que indica el subproceso del submenú en este caso se definirían las variables utilizadas en el subproceso necesarias para leer las opciones que necesite el cliente como entero para ingresar números y se procede a escribir las opciones.



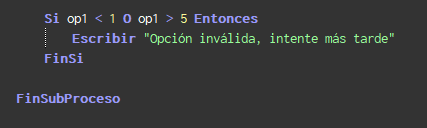
Se utiliza “Si y entonces” para determinar que el número ingresado sea menor o igual a 3 y mayor o igual a 1 porque esas son las únicas opciones y en caso de digitar un número menor o mayor este no será aceptado y devolverá un error. Según el usuario haya digitado un número entre 1 y 3 se devolverá la opción pertinente junto con la descripción y coste y se finalizará el segun.



Se le pide al usuario la confirmación de activación, en caso de que se elija 2 se le devolverá al menú anterior y si ha dictado 1 se redirigirá a lo siguiente:



Se utiliza “Si entonces” para determinar que el usuario haya escrito una de las opciones en cuyo caso si ha digitado 1 se utilizará otro “Si entonces” para determinar si el saldo es mayor al costo en cuyo caso se le restará el coste del paquete al saldo y devolverá que el paquete solicitado a sido activado, en caso de que el usuario no cuente con el saldo suficiente la función “sino” devolverá que el saldo es insuficiente, finSi.



Este “Si o entonces” se refiere a la opción 1 que el cliente introdujo primeramente por lo tanto que si es menor a 1 o mayor que 5 se devolverá que la opción es invalida y se finaliza el subproceso de igual manera devolviendo al cliente al menú principal tanto como si se realizó o no la recarga y así para cada subproceso.

## Fase de Implementación

### main.py

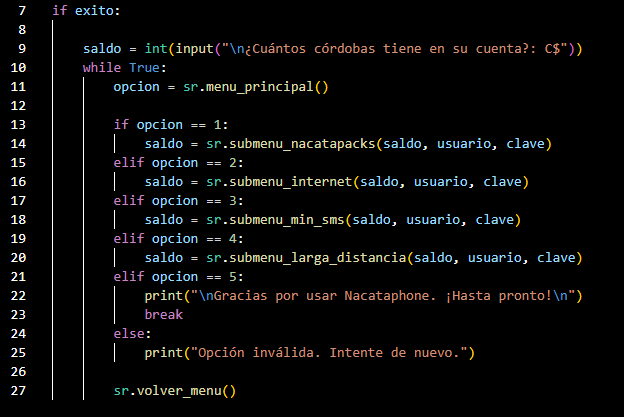
Este es el archivo principal, donde arranca todo. Lo primero que hace es importar dos módulos: uno que se llama sistema\_recargas (lo importa como sr) y otro que se llama sesión (se importa como ses). Eso es para no estar escribiendo los nombres largos cada vez que se usan funciones de esos archivos.



Después de eso, llama a una función llamada ses.inicio(), que básicamente te pregunta si querés iniciar sesión, registrarte o salir. Esa función te devuelve tres cosas: una variable llamada “exito” (que dice si lograste entrar o no), el número de teléfono del usuario, y su nombre. Si el login fue exitoso (o sea, si “exito” es True), entonces te deja seguir, si no, simplemente no hace nada más y termina el programa.

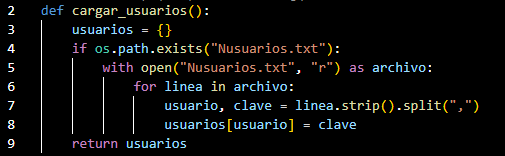


Luego, te pregunta cuánto saldo tenés. Te pide que escribas una cantidad, y eso se guarda en una variable que se llama saldo. Después de eso, empieza un ciclo infinito (while True) que te muestra un menú con varias opciones, como recargar paquetes, internet, mensajes, etc. Este menú viene del módulo sistema\_recargas. Según la opción que elijas, te manda al submenú correspondiente. Por ejemplo, si elegís la opción 1, te lleva al submenú de paquetes Nacata. Cada submenú te permite elegir un paquete, se te descuenta del saldo, y te regresa el saldo actualizado para seguir comprando si querés. Este ciclo sigue corriendo hasta que elegís la opción 5 que es "salir". Ahí te da un mensajito de despedida y se rompe el ciclo con un break.

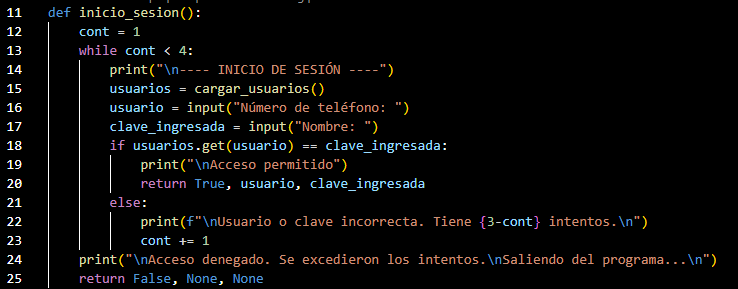


### sesion.py

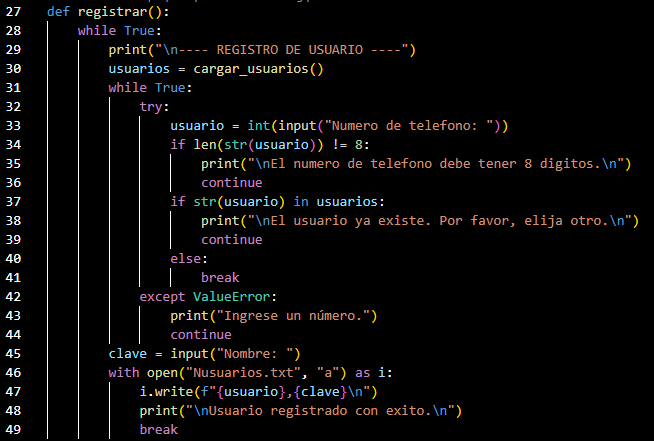
Este archivo se encarga de todo lo que tiene que ver con iniciar sesión o registrar usuarios. Lo que hace al principio es cargar un archivo de texto que se llama Nusuarios.txt. Este archivo guarda los usuarios registrados, y cada línea tiene el número de teléfono y el nombre del usuario, separados por una coma. Cuando el programa lee ese archivo, lo convierte en un diccionario de Python para poder buscar rápido si un usuario existe o no.



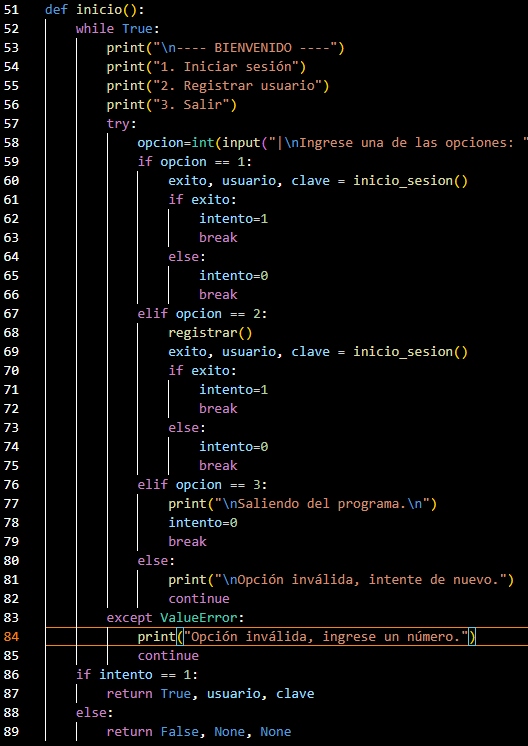
Después, hay una función que se llama inicio\_sesion() que te deja intentar entrar escribiendo tu número de teléfono y tu nombre. Tenés tres intentos para hacerlo bien. Si fallás las tres veces, te dice que no lograste iniciar sesión.



Otra función importante es registrar(), que sirve para crear un nuevo usuario. Te pide que ingresés un número de teléfono, valida que tenga 8 dígitos y que sea numérico, y también se fija si ese número ya existe en el archivo. Si todo está bien, guarda tu número y tu nombre en Nusuarios.txt.



La función principal de este archivo se llama inicio(), y ahí es donde te pregunta si querés iniciar sesión, registrarte o salir. Si elegís registrarte, una vez que te registrás te manda directo a iniciar sesión con los datos que acabás de meter.



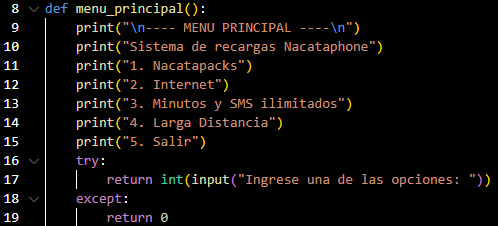
### sistema\_recargas.py

Este es el archivo que maneja todos los menús de paquetes que podés comprar, y también escribe un registro en un archivo para que quede guardado qué se activó y quién lo hizo.

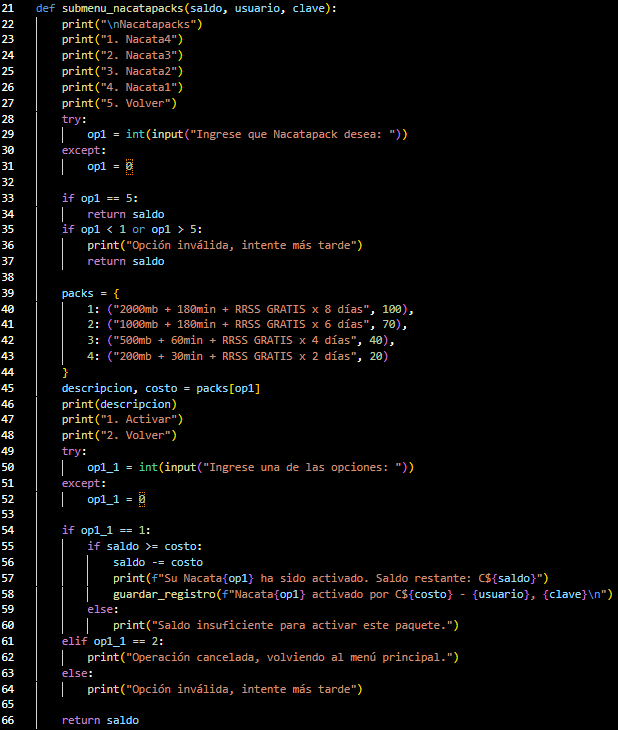
Primero hay una función que se llama guardar\_registro(), que simplemente escribe en un archivo llamado registro\_recargas.txt cada vez que alguien compra un paquete. Guarda el nombre del paquete, el costo, y el número y nombre del usuario que lo compró.



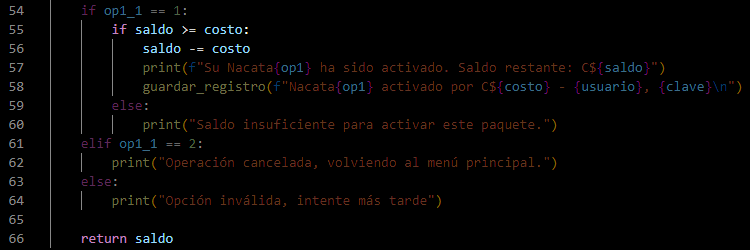
Luego está menu\_principal(), que muestra el menú principal donde elegís qué tipo de paquete querés: paquetes Nacata, internet, minutos/SMS, o llamadas internacionales.



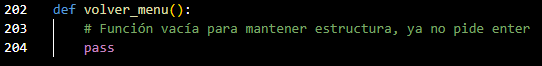
Después hay varios submenús, uno para cada tipo de paquete. Por ejemplo, submenu\_nacatapacks() tiene varios paquetes predefinidos, cada uno con un nombre (como Nacata1) y un precio. El usuario elige uno, se le descuenta del saldo, y se guarda el registro. Lo mismo pasa con los submenús de internet, minutos/SMS, y larga distancia. Todos funcionan igual: muestran opciones, validan lo que el usuario mete, descuentan saldo, y guardan un registro.



Cada submenú regresa el saldo actualizado para que se siga usando en el main.py.



También hay una función que se llama volver\_menu() pero está vacía, como si fuera un espacio reservado para más adelante. Podría usarse para poner un “presione enter para continuar” o algo así, pero por ahora no hace nada.



# Ejecución del Programa

La ejecución comienza en la gestión de usuarios. El archivo tiene funciones clave como **cargar\_usuarios()**, que carga los usuarios existentes desde un archivo .txt, y **guardar\_registro()** que escribe las operaciones realizadas.

La función **inicio\_sesion()** permite a un usuario ingresar su número de teléfono y nombre. Se tienen 3 intentos antes de bloquear el acceso.

En cambio, **registrar()** permite crear nuevos usuarios validando que el número tenga 8 dígitos y no exista previamente.

La función **inicio()** es la que permite escoger entre las 2 opciones anteriormente descritas y una tercera que permite terminar la ejecución del programa.

* Iniciar sesión
* Registrar nuevo usuario
* Salir del programa

Si el usuario inicia sesión con éxito, se continúa con el menú principal el cual se ejecuta con la función de **menu\_principal()**, el cual permite al usuario elegir entre 5 opciones de recarga:

* Nacatapacks
* Internet
* Minutos y SMS ilimitados
* Larga distancia
* Salir

Todos los submenús tienen esencialmente la misma estructura; Por ejemplo, si el usuario elige "Nacatapacks", se entra a la función **submenu\_nacatapacks()**. Ahí hay 4 tipos de paquetes como Nacata4, Nacata3, etc., cada uno con un costo y duración distintos.

En el caso de los otros submenús:

* **submenu\_internet()** para paquetes de navegación.
* **submenu\_min\_sms()** para paquetes de minutos y sms.
* **submenu\_larga\_distancia()** para llamadas internacionales.

Cada uno sigue el mismo flujo: se muestra una lista, el usuario elige, se valida si tiene saldo suficiente, y si sí, se activa el paquete y se descuenta el saldo. Además, se registra la operación con **guardar\_registro()**.

Para concluir, hay algunos detalles técnicos y del flujo general del sistema los cuales no fueron detallados previamente.

El sistema mantiene una estructura modular, cada parte se encarga de una tarea específica: registro, autenticación, selección y activación de paquetes. Usa archivos .txt para guardar datos de forma persistente.

El saldo se maneja dentro del código (aunque actualmente no hay una función directa para recargar), y se reduce según el paquete activado. Se podría mejorar esto integrando una base de datos o una interfaz gráfica.

Finalmente, también hay una función llamada **volver\_menu()** que está vacía, pero se mantiene para estructura futura o compatibilidad.

Este sistema es un buen ejemplo de lógica de menús, manejo de archivos, validaciones y modularidad en Python.

# Conclusiones y Recomendaciones

## Conclusiones

El desarrollo de este programa de recargas de minutos en Python responde a la necesidad de contar con una herramienta práctica que permita realizar recargas de manera simple, rápida y eficiente. En contextos donde la comunicación móvil es fundamental, disponer de un sistema automatizado ayuda a evitar errores humanos y mejora significativamente la gestión de las recargas.

Este tipo de solución facilita tareas que de otro modo serían repetitivas o propensas a fallos, asegurando que el proceso se lleve a cabo de forma consistente. Además, al estar desarrollado en Python, el programa es fácil de mantener y expandir, permitiendo futuras mejoras o adaptaciones según las necesidades del usuario.

En resumen, este programa representa una forma efectiva de optimizar un servicio básico, demostrando cómo con herramientas accesibles es posible resolver problemas reales del día a día.

## Recomendaciones

Para continuar con la mejora del programa, el próximo paso debería de ser el desarrollo de una interfaz gráfica, con el objetivo de hacer el sistema más accesible y fácil de utilizar para los usuarios. Esto permitiría una experiencia más intuitiva y organizada.

Asimismo, sería beneficioso agregar funcionalidades que permitan modificar los datos de la cuenta, como cambiar la contraseña o eliminarla, brindando al usuario mayor control sobre su información y aumentando la seguridad del sistema. Además, se recomienda agregar una base de datos que mantenga el saldo de cada usuario, lo cual facilita la gestión de la información financiera y permitiría implementar nuevas funciones como por un historial de movimientos más detallado y consistente.

# Anexos

**Formulario**

