**Programación básica**

**SC-115**

**MSc. Álvaro Camacho Mora**

**Sebastián Coto González**

**Gerson Elohim Betancourth Juarez**

**Eduardo Hernández Noguera**

Contenido

[Introducción 3](#_Toc133330877)

[objetivos 4](#_Toc133330878)

[Problema: 5](#_Toc133330879)

[Algoritmo: 6](#_Toc133330880)

[Registro de usuario: 6](#_Toc133330881)

[Usuario registrado 6](#_Toc133330882)

[Depósito 8](#_Toc133330883)

[Ver saldo 8](#_Toc133330884)

[pago servicios 9](#_Toc133330885)

[Eliminar usuario 10](#_Toc133330886)

[Venta y compra de divisas 11](#_Toc133330887)

[configuración avanzada 13](#_Toc133330888)

[Conclusión 15](#_Toc133330889)

[Bibliografía 16](#_Toc133330890)

# Introducción

Este texto trata acerca de como se realiza un software referente a una interfaz de un cajero automático con distintas funcionalidades que se logran ejecutar, realizando una serie de algoritmos, Este proyecto de programación tiene como objetivo simular el funcionamiento de un cajero automático a través de la implementación de un software que permita a los usuarios realizar operaciones bancarias de manera autónoma, además, para que pueda realizar las operaciones básicas de un cajero automático, tales como retiros de efectivo, depósitos de dinero, consulta de saldos y transferencias bancarias y en resumen, este proyecto de programación de un cajero automático representa un desafío interesante para los desarrolladores de software, ya que requiere la integración de diversas habilidades, como el diseño de interfaces de usuario, la programación de sistemas de seguridad y la implementación de algoritmos financieros.

# objetivos

* Uso getpass para la realización de seguridad al ingresar
* Desarrollar un algoritmo no muy complejo
* Realizar un interfaz funcional para el ingreso de pin
* Hacer un archivo de texto que se logre guardar la información adjunta
* Realizar un menú principal para el cajero automático y enseñar sus opciones
* Realizar un algoritmo para crear un usuario y se guarde esa información
* Hacer una solicitud de pin y cedula
* Hacer una opción para la parte de un deposito
* Realizar autenticaciones para el ingreso de algunas funciones
* Hacer un algoritmo que realice una transacción, haciendo una serie de acciones adicionales al tanto como también Retirar dinero, Depositar dinero, Ver saldo actual, Pagar servicios, Compra/Venta de Divisas o Eliminar usuario y pueda también salir al menú principal.
* Se representan una acción que le permita realizar pagos de diferentes servicios
* Hacer una opción de salir y que se finalice el programa

# Problema:

Desarrollar un programa que intente recrear la idea de un cajero automático ATM, para sustituir, agilizar y facilitar la interacción entre usuario y trabajador del banco, tendrá como objetivo principal: depósitos y retiros de dinero, estados de cuenta, pagos de servicios, registro de nuevos usuarios al banco entre otras cosas.

 Además, cada vez que ingrese a su cuenta deberá de ingresar una clave con 4 dígitos, se desplazara al menú principal una vez que ingrese el pin correcto, después se le presenta una serie de distintas opciones que pueda elegir la que vaya a necesitar en ese momento, al ingresar a una de las opciones el usuario podrá realizar la transacción necesario y poder volver al menú principal, también podrá pagar servicios ya sea Electricidad, Agua, Telefonía, Internet, Impuestos, Colegios Profesionales, Tarjeta de crédito, con opción de pago con las tres distintas monedas que se le otorga a disponibilidad, cambio de monedas y para finalizar con la opción de salir para finalizar la ejecución del programa.

# Algoritmo:

# Registro de usuario:

* se importa el getpass (import getpass)
* se utiliza def(def)
* se define la constante: intento y intentos máximos(constante)
* se solicita la cédula, si no cumple los requisitos se le solicita nuevamente (tres intentos) (input,len,if,else,print,while)
* se le solicita el nombre(input)
* se le solicita un pin de 4 dígitos, si no cumple los requisitos lo hará intentarlo hasta que los cumpla (el PIN no se puede ver). (While,getpass,len,if,else,print)
* se le solicita el de nuevo el PIN para autenticar(while,getpass,if,else,print)
* se le solicita al usuario que deposite más de 100.000 colones para poder finalizar el registro(if,elif,else,while,input)
* escribe todo los datos en el archivo de texto(file,write,format)
* se cierra el archivo(file,close)

# Usuario registrado

* Se abre el archivo registrar usuarios para leer y meter lo del archivo en un arreglo(open,readlines.strip,for,append)
* se cierra el archivo(file,close)
* **Retirar dinero:**

* Se hace un bucle para poder pasar por todas las cuentas y ver la información (Enumerate,for)
* Luego se le solicita al usuario que ingrese en que cuenta desear retirar el dinero(int,imput)
* luego se muestra el saldo de la cuenta y se le solicita al usuario la cantidad de dinero que desea retirar(print,float,input)
* se usa un bucle para que solo tenga 3 intentos el usuario de poner un monto menor de lo que tiene y si no lo regresa al menú(for,if,else,print,float)
* sí cumple los 3 intentos envía un mensaje(print)

# 

# Depósito

* Leer el archivo de texto existente que contiene la información de la cuenta del usuario y el saldo actual.
* Pedir al usuario que ingrese el monto del depósito.
* Calcular el nuevo saldo de la cuenta sumando el monto del depósito al saldo actual.
* Actualizar el archivo de texto con el nuevo saldo de la cuenta.
* Imprimir un mensaje indicando que el depósito se ha realizado con éxito y mostrando el saldo actualizado.

# Ver saldo

* Se hace un bucle para poder pasar por todas las cuentas y ver la información del arreglo (Enumerate,for)
* se imprime la informacion(print)
* se le solicita cual cuenta quiere ver(int,input)
* se imprime el saldo de la cuenta seleccionada(print)

# pago servicios

* Definir las variables necesarias, como el saldo de las cuentas, los servicios activos y los precios de los servicios en cada moneda.
* Crear una función para mostrar el menú de servicios y permitir al usuario seleccionar un servicio.
* Una vez seleccionado el servicio, verificar si el usuario tiene este servicio activo. Si no lo tiene, mostrar un mensaje y regresar al menú de servicios.
* Si el usuario tiene el servicio activo, mostrar el saldo a pagar en la moneda local correspondiente al servicio.
* Consultar de qué cuenta desea hacer el débito el usuario (colones, dólares o bitcoin) y verificar que haya suficiente dinero en esa cuenta.
* Si el usuario escoge una divisa que no es colones, realizar la conversión de divisas correspondiente.
* Si hay suficiente dinero en la cuenta seleccionada, realizar el pago correspondiente y rebajar el saldo de la cuenta.
* Regresar al submenú de servicios para permitir al usuario seleccionar otro servicio o salir del programa.

# Eliminar usuario

* Se usa un bucle para verificar el pin del usuario(while)
* se importa el get pass (import,get pass)
* se le solicita el pin(getpass)
* se usa las condiciones para que el pin sea el correcto(if,else,print)
* se abre el archivo para pasarlo a un arreglo(open,readlines,strip)
* se cierra el archivo(file,close)
* se fija en todo el recorrido del arreglo(for,len)
* si la cedula está en los datos del usuario se borra(del,if,print)
* si no está la cédula manda un mensaje de que no se encontró(print)

# Venta y compra de divisas

* se le solicita al usuario qué tipo de cambio desea hacer(input)
* **compra**:
* se usa un bucle para pasar por todo el arreglo y leer la información(for,enumerate)
* se imprime la información(print)
* se le solicita al usuario que cuenta va usar(int,input)
* se usa las condiciones dependiendo qué opción escoja(if,elif)
* se imprime el saldo de la cuenta escogida(print)
* se le solicita la cantidad de dinero que quiera cambiar(int,input)
* se usa condiciones dependiendo lo que ingrese(if,else)
* si la cantidad es mayor al monto de la cuenta se imprime un mensaje de que no se puede hacer la acción(print)
* sí cumple con el saldo que hay en la cuenta se imprime un mensaje de la cantidad retirada, de cómo queda la cuenta después de la acción y cuánto recibe en la otra moneda(print)
* multiplica el total por el monto de compra de la moneda para obtener el monto exacto
* **venta:**
* se usa un bucle para pasar por todo el arreglo y leer la información(for,enumerate)
* se imprime la información(print)
* se le solicita al usuario a qué cuenta van los fondos(int,input)
* se usa las condiciones dependiendo qué opción escoja(if,elif)
* se usan las condiciones dependiendo que escoja el usuario(if,elif)
* dependiendo de la opción imprime la cantidad de dinero en la cuenta escogida(print)
* si el saldo es que quiere pasar el mayor al saldo que se tiene imprime un mensaje(print)
* sí cuenta con la cantidad insertada, imprime cuánto dinero se retiró de una cuenta para la otra imprime el saldo con el que se quedó dicha cuenta e imprime cuanto se pasó(print)
* divide el total entre el monto de compra de la moneda para obtener el monto exacto
* **y estos procesos se repiten en todas las opciones**

# configuración avanzada

* para acceder a esta opción se le pide la clave
* en caso de que digite mal el pin se le indica un error de ingreso
* se hace uso de la biblioteca getpass
* se despliega un menú con distintas opciones (while, if, else, elif, variables, input)
* se realiza la opción de eliminar usuario (while true, if, elif, variable, input)
* se realiza la opción de modificar el tipo de cambio
* Abrir el archivo de texto que contiene los factores de conversión. Este archivo puede tener un formato de tabla con filas y columnas que representan los diferentes tipos de conversión y sus respectivos factores.
* Desplegar un menú en la consola que muestre cada uno de los tipos de conversión existentes. Para ello, se puede usar la función "print" de Python y una estructura de control como "for" o "while" para iterar sobre los elementos del menú.
* Solicitar al usuario que seleccione el tipo de cambio que desea modificar mediante la entrada de un número. Para ello, se puede usar la función "input" de Python y un mensaje de texto para solicitar la entrada del usuario.
* Leer el factor de conversión actual del archivo de texto correspondiente al tipo de cambio seleccionado. Esto se puede hacer usando la función "readline" para leer la línea correspondiente al tipo de cambio seleccionado.
* Solicitar al usuario que ingrese el nuevo factor de conversión para el tipo de cambio seleccionado mediante la entrada de un número. Para ello, se puede usar la función "input" de Python y un mensaje de texto para solicitar la entrada del usuario.
* Escribir el nuevo factor de conversión en el archivo de texto correspondiente al tipo de cambio seleccionado. Esto se puede hacer usando la función "write" para escribir el nuevo factor de conversión en la línea correspondiente.
* Cerrar el archivo de texto para guardar los cambios realizados.
* Si el usuario selecciona la opción "Salir" en el menú principal, terminará la ejecución del programa.

# Conclusión

En conclusión, al realizar el proyecto se fueron creando varios algoritmos con distintas funcionalidades, haciendo referente al funcionamiento de un cajero automático, con menú de inicio, pedir pin para el acceso requerido, almacenamiento de información, diseño de interfaz similar al cajero automático, se hicieron mejoras para tener una buena fluidez del software, se hizo opciones para salir y se finalice la ejecución del programa, se desarrolló con gran lógica básica y lo mas interactivo posible.

# Bibliografía

Documento realizado con argumentos propios

(Betancourth, 2023)

(Coto, 2023)

(2001-2023 Python Software Foundation. Todos los derechos reservados, [getpass — Entrada de contraseña portátil — documentación de Python - 3.11.3](https://docs.python.org/3/library/getpass.html))