|  |
| --- |
| Pruebas de Caja Blanca |
| ***“Sistema de pagos de alícuotas”*** |
|  |
|  |
| **Integrantes:**  **Lucas Góngora**  **Gabriel Manosalvas**  **Jairo Molina**  **Yandry Vélez**  **Fecha 2025-02-10** |

**Prueba caja blanca de** RF 1.Login

1. **Código FUENTE**

import re

import getpass

from csv\_managment import \*

from consola import \*

validacion\_username = r'^[a-zA-Z]+$'

validacion\_clave = r'^(?=.\*[!@#$%^&\*(),.?":{}|<>])(?=.{8,})'

def iniciar\_sesion():

    limpiar\_pantalla()

    clave\_incorrecta = True

    intentos\_realizados = 0

    intentos\_permitidos = 3

    if obtener\_credenciales\_autorizadas() == []:

        registrar\_administrador()

    while clave\_incorrecta:

        print("\t----LOGIN---\t")

        usuario, clave = obtener\_credenciales()

        validacion\_exitosa = validar\_credenciales(usuario, clave)

        if validacion\_exitosa:

            clave\_incorrecta = False

            print("\nCredenciales válidas")

            input("Presiona ENTER para pasar al menú ...")

            limpiar\_pantalla()

        else:

            print("\nCredenciales inválidas\n")

            intentos\_realizados += 1

            print("Te quedan " + str(intentos\_permitidos - intentos\_realizados) + " intentos")

            input("Presiona ENTER para volver al Login ...")

            limpiar\_pantalla()

            if intentos\_realizados >= intentos\_permitidos:

                print("Número de intentos permitidos alcanzados")

                salir\_programa()

def registrar\_administrador():

    print("\tREGISTRAR\t")

    credenciales = obtener\_credenciales()

    insertar("./databases/administrador.csv", credenciales, "./databases")

    limpiar\_pantalla()

def validar\_credenciales(usuario, clave):

    usuario\_autorizado, clave\_autorizada = obtener\_credenciales\_autorizadas()[0]

    return usuario == usuario\_autorizado and clave == clave\_autorizada

def obtener\_credenciales\_autorizadas():

    credenciales\_autorizadas = leer("./databases/administrador.csv")

    return credenciales\_autorizadas

def obtener\_credenciales():

    usuario = validar\_entrada("username", validacion\_username, "Ingresa solo letras")

    clave = validar\_entrada("clave", validacion\_clave, "Ingresa al menos 8 caracteres y al menos un caracter especial")

    return [usuario, clave]

def validar\_entrada(tipo, patron, mensaje\_error):

    while True:

        if tipo == "clave":

            entrada = getpass.getpass("Ingresa la clave: ")

        elif tipo == "username":

            entrada = input("Ingresa el nombre de usuario: ")

        else:

            print("Ingresa un tipo de entrada válido")

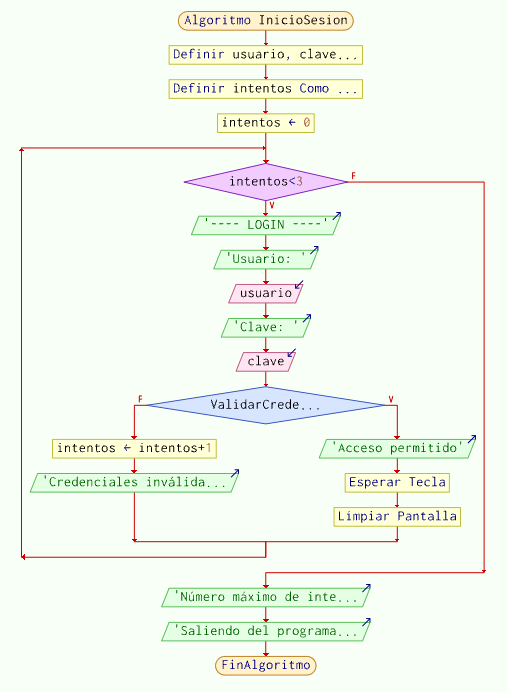
            break

        if re.match(patron, entrada):

            return entrada

print(mensaje\_error)

1. **Diagrama de flujo**

****

1. **Grafo de Flujo (GF)**
2. **Identificación de rutas**

R1:

R2:

1. **Complejidad Ciclomática**

Se puede calcular de las siguientes formas:

* V(G) = número de nodos predicados(decisiones)+1

V(G)= 2+1=3

* V(G) = A – N + 2

V(G)= 5−5+2=2

DONDE:

**P:** Número de nodos predicado

**A:** Número de aristas

**N:** Número de nodos

**Prueba caja blanca de** RF 1.Login

1. **Código FUENTE**
2. **Diagrama de flujo**
3. **Grafo de Flujo (GF)**
4. **Identificación de rutas**
   1. R1:
   2. R2:
5. **Complejidad Ciclomática**

Se puede calcular de las siguientes formas:

* V(G) = número de nodos predicados(decisiones)+1

V(G)= 2+1=3

* V(G) = A – N + 2

V(G)= 5−5+2=2

DONDE:

**P:** Número de nodos predicado

**A:** Número de aristas

**N:** Número de nodos