[[1]](#footnote-1)

1. **Sistema de Control de Acceso**

(Septiembre, 27, 2024)

Primer A. Juan Diego Duran Villamil, Segundo A. Juan Camila Guevara Duran

Tercer A. Cristian Ferney Buitrago López Cuarto A. Edison Samuel Rodríguez Rodríguez

Resumen -

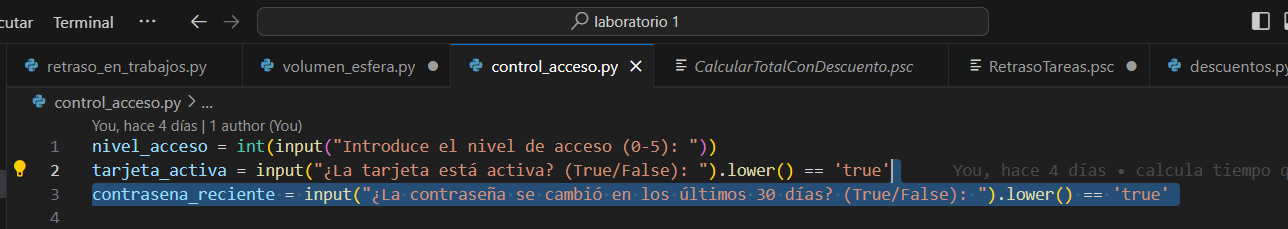
Desarrollar un algoritmo que verifique si un usuario puede acceder a áreas restringidas de un edificio de alta seguridad, basado en su nivel de acceso, la actividad de su tarjeta de identificación y si ha cambiado su contraseña recientemente.

### **INTRODUCCIÓN**

**E**neste laboratorio, desarrollaremos un algoritmo para un sistema de control de acceso a edificios de alta seguridad. Este sistema es crucial para garantizar que solo las personas autorizadas puedan acceder a áreas restringidas, protegiendo así la información y los recursos valiosos.

El sistema de control de acceso que vamos a crear verificará si los usuarios pueden ingresar a ciertas áreas restringidas dependiendo de su nivel de acceso y si cumplen con condiciones de seguridad adicionales, como haber cambiado su contraseña recientemente o si su tarjeta de identificación está activa. Utilizaremos operadores de comparación, operadores lógicos y la estructura if para determinar si un usuario puede o no acceder a un área específica.

Este laboratorio está diseñado para principiantes en programación, por lo que explicaremos cada paso con detalle para asegurar una comprensión completa del proceso.

***Explicación del código:***

**Solicitar Información del Usuario**

Primero, necesitamos obtener la información del usuario: su nivel de acceso, si su tarjeta está activa y si ha cambiado su contraseña recientemente.

**Python**

nivel\_acceso = int(input("Introduce el nivel de acceso (0-5): "))

tarjeta\_activa = input("¿La tarjeta está activa? (True/False): ").lower() == 'true'

contrasena\_reciente = input("¿La contraseña se cambió en los últimos 30 días? (True/False): ").lower() == 'true'

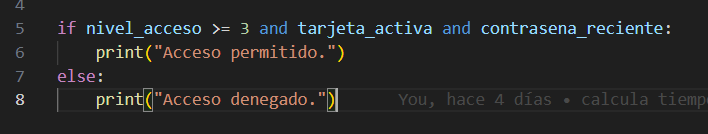
***Nivel de Acceso****: Solicita al usuario que introduzca su nivel de acceso y conviértelo a un número entero.*

***Tarjeta Activa****: Pregunta si la tarjeta está activa y convierte la respuesta a un valor booleano (True o False).*

***Contraseña Reciente****: Pregunta si la contraseña se cambió recientemente y convierte la respuesta a un valor booleano.*

***Verificar Condiciones de Acceso***

*Usaremos operadores de comparación y lógicos para verificar si el usuario cumple con todas las condiciones necesarias para acceder.*



**Python**

if nivel\_acceso >= 3 and tarjeta\_activa and contrasena\_reciente:

print("Acceso permitido.")

else:

print("Acceso denegado.")

***Condición de Nivel de Acceso****: Verifica si el nivel de acceso del usuario es mayor o igual a 3.*

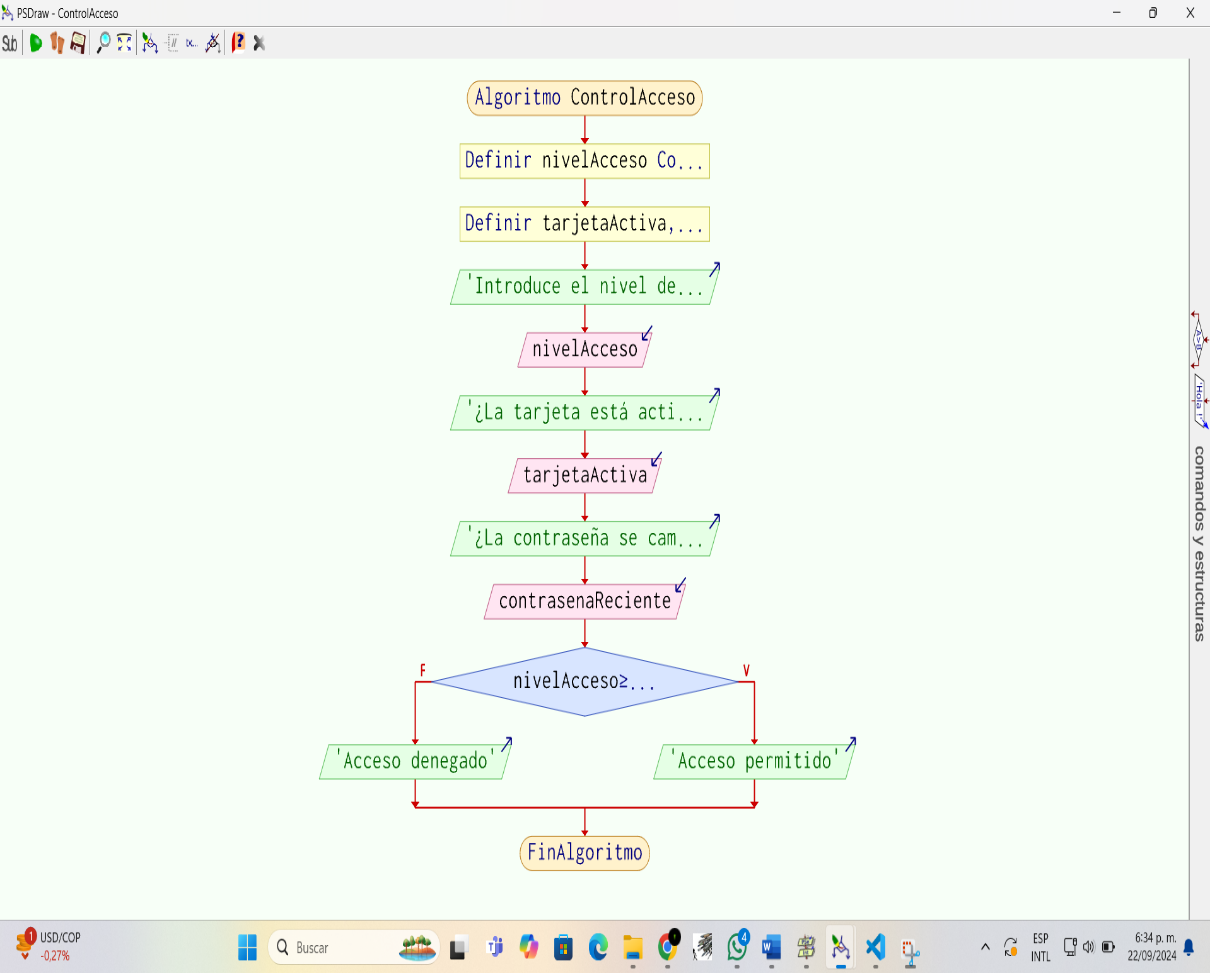
***Condición de Tarjeta Activa****: Verifica si la tarjeta del usuario está activa.*

***Condición de Contraseña Reciente****: Verifica si la contraseña del usuario se cambió en los últimos 30 días.*

***Decisión de Acceso****: Si todas las condiciones se cumplen, imprime “Acceso permitido”. De lo contrario, imprime “Acceso denegado”.*

**“Prueba del Programa**

Ejecuta el programa con diferentes combinaciones de entradas para asegurarte de que funciona correctamente en todos los casos posibles.



**Prueba con Acceso Permitido**: Introduce un nivel de acceso de 4, tarjeta activa como True y contraseña reciente como True. El programa debería imprimir “Acceso permitido.”

**Prueba con Acceso Denegado**: Introduce un nivel de acceso de 2, tarjeta activa como False y contraseña reciente como False. El programa debería imprimir “Acceso denegado.**”**

(Rossum, 2024)

## **Pasos:**

**Solicitar Información del Usuario**

Primero, necesitamos obtener la información del usuario: su nivel de acceso, si su tarjeta está activa y si ha cambiado su contraseña recientemente.

**Verificar Condiciones de Acceso**

Usaremos operadores de comparación y lógicos para verificar si el usuario cumple con todas las condiciones necesarias para acceder.

### **Conclusión**

Este laboratorio te ha guiado a través del desarrollo de un algoritmo básico para un sistema de control de acceso. Has aprendido a usar operadores de comparación, operadores lógicos y la estructura if-else para tomar decisiones basadas en múltiples condiciones.

##### Objetivo General:

##### Implementar un sistema de control de acceso eficiente y seguro que proteja las instalaciones y la información sensible de la organización.

##### Objetivos Específicos:

##### 1. Evaluar las necesidades de seguridad de la organización para determinar los puntos críticos de acceso.

##### 2. Instalar tecnología de control de acceso, como tarjetas magnéticas o biometría, en todas las entradas principales.

##### 3. Capacitar al personal sobre el uso adecuado del sistema y la importancia de la seguridad en el acceso a las instalaciones.

##### 4. Realizar auditorías trimestrales del sistema para identificar y resolver posibles vulnerabilidades.

##### 5. Establecer protocolos claros para la gestión de accesos temporales y visitantes.

1. [↑](#footnote-ref-1)