



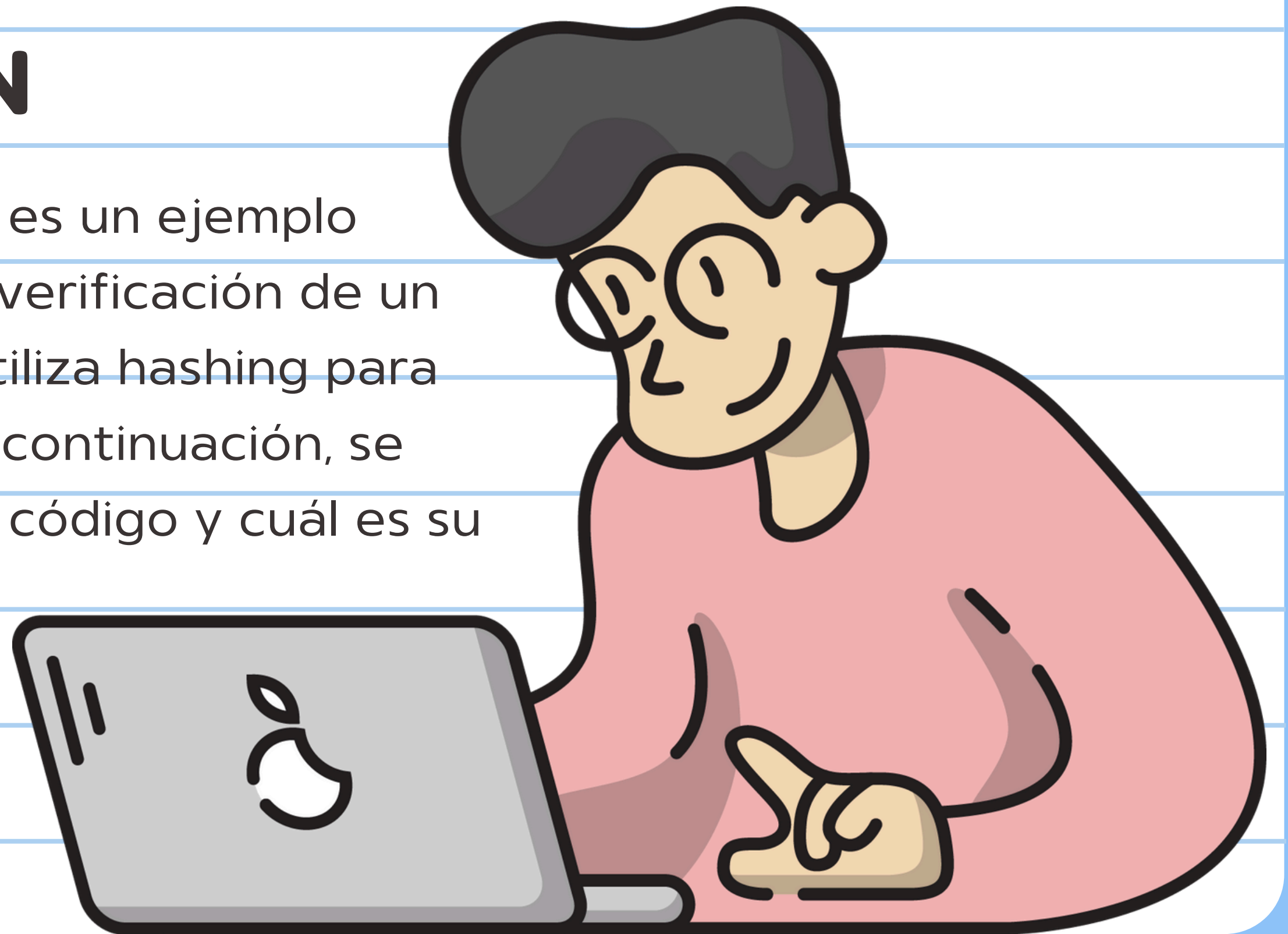
Powered by
Arizona State University®

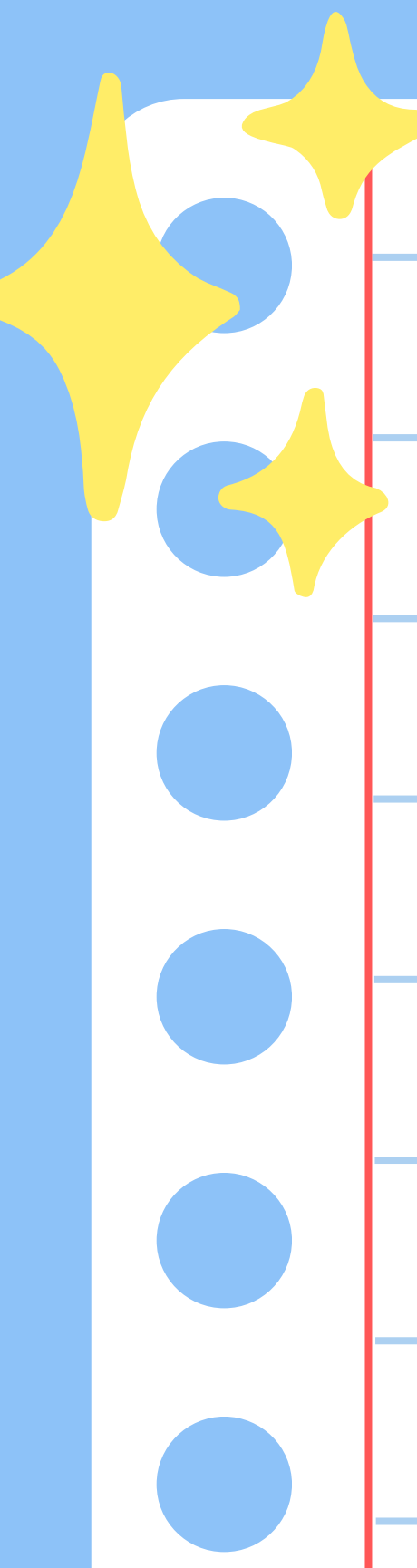
TRABAJO INTEGRADOR

Crhistopher Casanova

INTRODUCCIÓN

El código proporcionado es un ejemplo básico de un sistema de verificación de un usuario en Python que utiliza hashing para proteger contraseñas. A continuación, se detalla cómo funciona el código y cuál es su propósito.





El código está diseñado para acreditar a los usuarios mediante la comparación de contraseñas ingresadas con las contraseñas almacenadas de forma segura.

1.- IMPORTACIÓN DE MÓDULO

```
1 import hashlib
```

Se usará hashlib para crear hashes de contraseñas.



2.- FUNCIÓN HASH_PASSWORD

Su propósito es generar un hash seguro a la contraseña

```
# Función para hash de contraseñas
def hash_password(password):
    """Genera un hash seguro para la contraseña."""
    return hashlib.sha256(password.encode()).hexdigest()
```

3.- FUNCIÓN VERIFY_PASSWORD

Compara una contraseña ingresada con un hash almacenado para verificar si son equivalentes.

```
# Función para verificar la contraseña
def verify_password(stored_hash, password):
    """Verifica si la contraseña proporcionada coincide con el hash almacenado."""
    return stored_hash == hash_password(password)
```



4.- DICCIONARIO DE USUARIOS Y CONTRASEÑAS (HASH)

Aquí se almacena los usuarios y contraseñas en forma de hash

```
# Diccionario de usuarios y contraseñas (hash)
users = {
|   "Crhis06": hash_password("12345@")
}
```

5.- NÚMERO MÁXIMO DE INTENTOS

Define el número máximo de intentos de inicio de sesión permitidos.

```
# Número máximo de intentos
max_intentos = 3
```

6.- BUCLE PARA MANEJAR MÚLTIPLES INTENTOS DE INICIO DE SESIÓN

Su función es permitir al usuario intentar iniciar sesión varias veces.

```
# Bucle para manejar múltiples intentos de inicio de sesión
for intento in range(max_intentos):
    print("Bienvenido, ingresa tu usuario")
    user = input("Usuario: ")

    print("Contraseña: ")
    pwd = input()

    if user in users and verify_password(users[user], pwd):
        print("Bienvenido de vuelta", user)
        break # Salir del bucle si el inicio de sesión es exitoso
    else:
        print("Usuario o contraseña incorrectos")
        if intento < max_intentos - 1:
            print("Tienes", max_intentos - (intento + 1), "intentos restantes")
        else:
            print("Has agotado el número máximo de intentos. Por favor, intenta más tarde.")
```




7.- MENSAJE FINAL

Imprime in mensaje final para indicar que ha finalizado el inicio se sesión.

```
# Este bloque se ejecuta después de salir del bucle  
print("Fin del proceso de inicio de sesión.")
```

CONCLUSIÓN

La protección de contraseñas es un aspecto fundamental de la seguridad en el desarrollo de software. La función hash_password demuestra cómo se puede utilizar el hashing para almacenar contraseñas de manera segura, protegiéndolas de accesos no autorizados.



CREDENCIALES CORRECTAS

```
Bienvenido, ingresa tu usuario
Usuario: Crhis06
Contraseña:
12345@
Bienvenido de vuelta Crhis06
Fin del proceso de inicio de sesión.
```

CREDENCIALES INCORRECTAS

```
Bienvenido, ingresa tu usuario
Usuario: Crhis06
Contraseña:
12345
Usuario o contraseña incorrectos
Tienes 2 intentos restantes
Bienvenido, ingresa tu usuario
Usuario: Cris
Contraseña:
122
Usuario o contraseña incorrectos
Tienes 1 intentos restantes
Bienvenido, ingresa tu usuario
Usuario: cris
Contraseña:
12333
Usuario o contraseña incorrectos
Has agotado el número máximo de intentos. Por favor, intenta más tarde.
Fin del proceso de inicio de sesión.
```




**¡GRACIAS POR
SU ATENCIÓN!**