

**Fundamentos de Python**

Unidad 1. Introducción a Python y estructuras de control

PROJECTOS

1. Simulador de cajero automático
2. Sistema de inicio de sesión

ESTUDIANTE

Armando Gabriel Jacinto López

INSTITUCION

INFOTEC

FECHA

10/09/2025

**Introducción**

El presente informe analiza dos proyectos de desarrollo en Python, enfocados en las estructuras de control y tipos de datos. El primer proyecto es un **simulador básico de cajero automático**, diseñado para dispensar billetes de manera eficiente y gestionar un inventario limitado. El segundo proyecto consiste en un **sistema de inicio de sesión** que autentica usuarios con un número restringido de intentos.

El objetivo de estos proyectos es evaluar la funcionalidad, la estructura del código y las mejores prácticas aplicadas en ambos proyectos. Se examinarán tanto los aciertos en la implementación de la lógica, como el uso adecuado de las estructuras de control y datos, con el fin de poner en práctica los conceptos aprendidos durante esta primera unidad. Los proyectos, aunque resuelven sus problemas de manera funcional, sirven como una excelente base para discutir principios de diseño de software más avanzados.

**Desarrollo**

**Proyecto 1: Simulador de Cajero Automático**

Texto

El contenido generado por IA puede ser incorrecto.El proyecto de cajero automático cumple con sus requisitos principales de manera directa. El algoritmo utiliza un **ciclo while** para continuar la operación de dispensar billetes hasta que el monto solicitado se reduzca a cero o no se pueda completar.

Ilustración Variables globales

Texto

El contenido generado por IA puede ser incorrecto.La lógica principal reside en la función calculate, que usa un ciclo for para iterar a través de las denominaciones de billetes en orden descendente (de $1000 a $50). Este enfoque garantiza que se entregue la menor cantidad de billetes posible, como se especifica en los requisitos del proyecto.

Ilustración Función principal

Texto

El contenido generado por IA puede ser incorrecto.La gestión del inventario y el conteo de billetes dispensados se realiza a través de variables globales. El código utiliza la función globals() de Python para modificar estas variables desde dentro de la función calculate. Las variables FIFTY\_BILL, ONE\_HUNDRED\_BILL, etc., representan el inventario disponible, mientras que sus contrapartes con el sufijo \_COUNT almacenan los billetes entregados.

Ilustración Ciclo while e inicio del programa

Texto

El contenido generado por IA puede ser incorrecto.

Ilustración Ejecución del sistema de cajero

**Proyecto 2: Sistema de Inicio de Sesión**

Texto

El contenido generado por IA puede ser incorrecto.El sistema de inicio de sesión cumple con sus objetivos de forma sencilla y eficaz. Utiliza un **ciclo while** controlado por dos condiciones: una variable booleana VALID\_CREDENTIALS y un contador ATTEMPS\_LEFT. Este enfoque es un patrón de diseño común para bucles con un número limitado de iteraciones.

Ilustración Variables globales y la función para restar los intentos

Texto

El contenido generado por IA puede ser incorrecto.La lógica de autenticación se maneja a través de una serie de **estructuras if-elif-else** que evalúan las entradas del usuario. El código es capaz de proporcionar mensajes de error específicos para casos como "Usuario inválido", "Contraseña inválida" o "Credenciales inválidas" (ambas incorrectas), lo que mejora la experiencia del usuario.

Ilustración Inicio del programa y ciclo while

**Texto

El contenido generado por IA puede ser incorrecto.**

Ilustración Ejecución del sistema de sesión

**Conclusión**

Ambos proyectos demuestran una comprensión sólida de los conceptos fundamentales de las estructuras de control y tipos de datos en el lenguaje Python, como el uso de estructuras de control (if, elif, else) y ciclos (while, for). Cumplen de manera efectiva con los requisitos establecidos, resolviendo los problemas planteados con soluciones directas.

Sin embargo, se tienen áreas de mejora en ambos proyectos, pudiendo evolucionar a diseños más robustos y estructurados. Una posible mejora podría ser aplicar conceptos de la programación orientada a objetos para gestionar los sistemas de manera más efectiva. Al igual, la seguridad en ambos proyectos podría ser mejor, como en el caso del sistema de inicio de sesión, no escribiendo las credenciales directamente en el código para evitar que sean vistas por usuarios con acceso al código.

En conclusión, estos dos proyectos sirvieron como una base fundamental para en un futuro embarcarse en problemas con mayor complejidad, de momento, se ha adoptado una buena base de conocimientos para seguir progresando en el programa e indagar en conceptos con mayor complejidad.