



Profundiza más

Recurso de Profundización

Flujo de ejecución secuencial y entrada/salida básica (Parte 2)

Autor: Damián Nicolalde Rodríguez

Objetivo del Documento: El objetivo de este recurso es ofrecer una referencia visual clara y concisa que sintetice los cinco pasos esenciales para desarrollar un programa secuencial eficiente en Python, desde la declaración y lectura de variables con `input()` hasta la conversión adecuada de tipos de datos.

Recurso de Profundización 1: Infografía pasos para un programa secuencial y eficiente

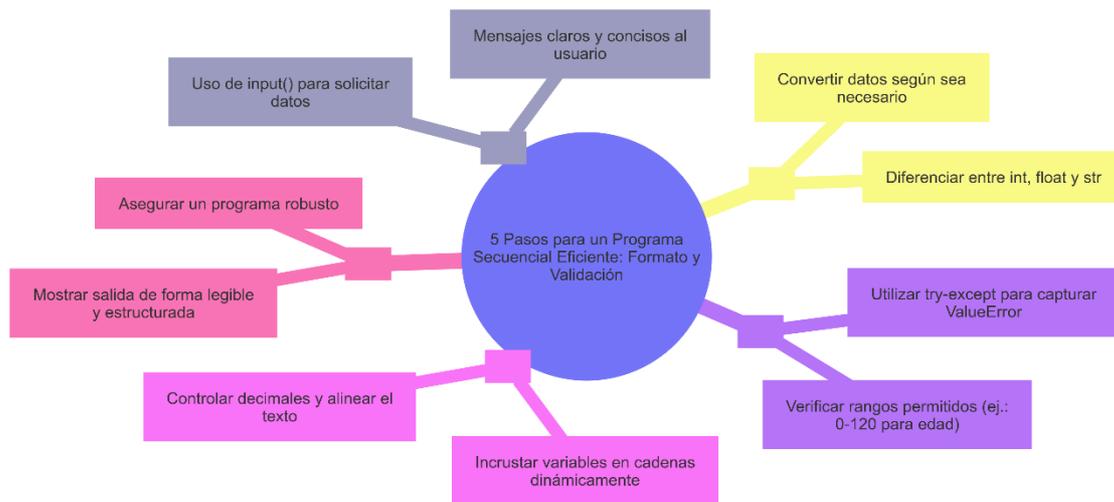


Imagen 1. Infografía 5 pasos para desarrollar un programa secuencial Damian Nicolalde (2025). Infografía 5 pasos para desarrollar un programa secuencial. La imagen es una infografía que presenta de manera clara y secuencial los cinco pasos esenciales para desarrollar un programa secuencial eficiente en Python, combinando un formato atractivo con validación robusta de datos.

Paso 1: Declaración y Lectura de Variables

Se resalta el uso de `input()` para solicitar datos al usuario, acompañado de mensajes claros y concisos que guían la interacción inicial.

Paso 2: Conversión de Tipos de Datos

Se enfatiza la importancia de diferenciar entre `int`, `float` y `str`, y se indica cuándo y cómo convertir los datos de acuerdo a las necesidades del programa.

Paso 3: Validación de Datos



Profundiza más

Se ilustra el uso del bloque try-except para capturar errores (por ejemplo, ValueError) y la verificación de rangos permitidos, como validar que la edad se encuentre entre 0 y 120.

Paso 4: Formateo con f-Strings

Se muestra cómo incorporar variables en cadenas de forma dinámica utilizando f-strings, controlando decimales y alineando el texto para lograr una presentación profesional de la salida.

Paso 5: Presentación Final

Se destaca la salida final del programa, resaltada como legible y estructurada, con énfasis en la robustez del programa y consejos prácticos para mejorar la implementación.

Esta representación visual permite al lector comprender rápidamente la secuencia lógica y las mejores prácticas para crear programas en Python que no solo sean funcionales, sino también robustos y bien presentados.