

Rúbrica de Punto Único para Evaluar Estructuras de Decisión en Python - Ingeniería de Sistemas

Rúbrica de Punto Único | Ingeniería | Ingeniería de sistemas | 4 niveles

Descripción

Esta rúbrica está diseñada para evaluar el dominio y aplicación correcta de las estructuras de decisión en Python, enfocándose en la lógica, sintaxis, y claridad del código. Se ofrecen aspectos positivos y áreas de mejora para guiar el desarrollo y perfeccionamiento de los estudiantes.

Rúbrica

Rúbrica de Punto Único para Evaluar Estructuras de Decisión en Python - Ingeniería de Sistemas

Esta rúbrica está diseñada para evaluar el dominio y aplicación correcta de las estructuras de decisión en Python, enfocándose en la lógica, sintaxis, y claridad del código. Se ofrecen aspectos positivos y áreas de mejora para guiar el desarrollo y perfeccionamiento de los estudiantes.

Criterios	Aspectos Positivos	Aspectos a Mejorar
Comprensión y uso correcto de estructuras if, elif y else	Utiliza correctamente las estructuras condicionales, demostrando clara comprensión de la sintaxis y lógica.	Confusión en el uso de elif o else, falta de estructura condicional clara o errores sintácticos.
Aplicación adecuada de operadores lógicos y relacionales	Emplea operadores lógicos y relacionales correctamente para construir condiciones precisas y eficientes.	Uso incorrecto o redundante de operadores que genera resultados inesperados o código poco claro.
Claridad y legibilidad del código	El código está bien indentado, con nombres de variables descriptivos y comentarios que facilitan su comprensión.	Falta de indentación adecuada, variables con nombres confusos o ausencia de comentarios explicativos.
Correcta anidación de estructuras condicionales	Las estructuras anidadas están organizadas de manera lógica y eficiente, evitando complejidad innecesaria.	Anidación excesiva o desorganizada que dificulta la lectura y puede generar errores lógicos.
Implementación de soluciones que cumplen con los requisitos del problema	El código resuelve correctamente el problema planteado, cubriendo todos los casos necesarios.	Solución incompleta o que no considera varios escenarios, resultando en errores o salidas incorrectas.

Criterios	Aspectos Positivos	Aspectos a Mejorar
Uso adecuado de estructuras de decisión para optimizar el código	Optimiza el uso de condiciones para evitar redundancias y mejorar la eficiencia del programa.	Condiciones repetitivas o innecesarias que complican el código sin mejorar su funcionalidad.
Manejo de errores y validación de entradas mediante estructuras condicionales	Incluye validaciones y manejo de posibles errores o entradas incorrectas usando condiciones adecuadas.	Ignora la validación de datos o el manejo de errores, lo que puede provocar fallas durante la ejecución.
Capacidad para explicar y justificar la lógica empleada en las estructuras de decisión	Puede explicar claramente la lógica y el flujo del programa en base a las estructuras condicionales usadas.	Dificultad para justificar la elección y funcionamiento de las estructuras condicionales implementadas.