

Rúbrica Analítica para Evaluación de Algoritmos en Ciencia de Datos

Rúbrica Analítica | Ciencias Exactas y Naturales | Ciencia de datos | 3 niveles

Descripción

Esta rúbrica está diseñada para evaluar de forma detallada las habilidades de los estudiantes universitarios en la elaboración, implementación y análisis de algoritmos aplicados a la ciencia de datos utilizando Python. Cada criterio refleja un aspecto fundamental del aprendizaje esperado y se evalúa en tres niveles de desempeño: Excelente, Bueno y Bajo.

Rúbrica

Rúbrica Analítica para Evaluación de Algoritmos en Ciencia de Datos

Esta rúbrica está diseñada para evaluar de forma detallada las habilidades de los estudiantes universitarios en la elaboración, implementación y análisis de algoritmos aplicados a la ciencia de datos utilizando Python. Cada criterio refleja un aspecto fundamental del aprendizaje esperado y se evalúa en tres niveles de desempeño: Excelente, Bueno y Bajo.

Criterios de Evaluación	Excelente	Bueno	Bajo
1. Elaboración de pruebas de escritorio mediante diferentes métodos de algoritmos	Realiza pruebas de escritorio completas y precisas, utilizando múltiples métodos para validar cada algoritmo, identificando claramente errores y resultados esperados.	Ejecuta pruebas de escritorio adecuadas con algunos métodos, detectando la mayoría de errores y resultados, aunque con algunas omisiones menores.	Pruebas de escritorio incompletas o superficiales, con falta de variedad en métodos y poca identificación de errores o resultados esperados.
2. Simplificación del lenguaje de código mediante estructuras de control de flujo	Usa estructuras de control de flujo (condicionales, bucles) de manera eficiente y clara para simplificar y optimizar el código, mejorando su legibilidad y mantenimiento.	Emplea estructuras de control de flujo correctamente, aunque con cierta redundancia o falta de optimización en la simplificación del código.	Uso inadecuado o limitado de estructuras de control, resultando en código complejo o difícil de entender y mantener.

Criterios de Evaluación	Excelente	Bueno	Bajo
3. Implementación y evaluación de algoritmos de búsqueda y ordenamiento considerando eficiencia temporal	Implementa algoritmos de búsqueda y ordenamiento correctamente y analiza con precisión su complejidad temporal, justificando decisiones de implementación.	Implementa algoritmos funcionales y realiza una evaluación básica de su eficiencia temporal, aunque con análisis poco detallados o justificaciones superficiales.	Implementaciones incorrectas o incompletas de algoritmos y ausencia de evaluación o comprensión de la eficiencia temporal.
4. Aplicación de estructuras de datos (listas, diccionarios, matrices) para manipulación y preparación de datos en Python	Utiliza de forma adecuada y avanzada listas, diccionarios y matrices para organizar y preparar datos, demostrando comprensión profunda y manipulación eficiente.	Aplica estructuras de datos básicas correctamente para manipulación de datos, aunque con limitaciones en la eficiencia o manejo avanzado.	Uso limitado o incorrecto de estructuras de datos, dificultando la manipulación y preparación efectiva de los datos.
5. Construcción de funciones reutilizables con buenas prácticas de programación orientada al análisis	Diseña funciones modulares, reutilizables y bien documentadas, integrando buenas prácticas que facilitan el análisis y mantenimiento del código.	Construye funciones funcionales pero con documentación o modularidad limitada, mostrando conocimiento básico de buenas prácticas.	Funciones poco reutilizables, mal estructuradas o sin respeto a buenas prácticas, dificultando la comprensión y mantenimiento del código.
6. Implementación de arreglos, vectores y matrices en Python con relación a fundamentos matemáticos del Álgebra Lineal	Implementa estructuras matriciales relacionando claramente su representación computacional con conceptos matemáticos, evidenciando comprensión integral.	Implementa arreglos y matrices funcionales con cierta relación a fundamentos matemáticos, aunque sin profundidad en la explicación o integración.	Implementaciones incorrectas o sin relación clara con los fundamentos matemáticos de Álgebra Lineal.
7. Construcción de clases básicas con atributos y métodos en programación orientada a objetos	Desarrolla clases completas con atributos y métodos bien definidos, aplicando correctamente principios básicos de programación orientada a objetos.	Crea clases funcionales con atributos y métodos, pero con limitaciones en la estructura o aplicación de conceptos OOP.	Clases incompletas, mal estructuradas o sin aplicación de los principios básicos de programación orientada a objetos.