

Rúbrica de Evaluación

By Alberto Caro

Proyecto 1: Sistema de Monitoreo Ambiental (50% de la Calificación Total)

- **Servidor HTTP en Lazarus (Recepción y Almacenamiento de Datos JSON): (20%)**
 - Excelente (16-20%): Recibe correctamente los datos JSON, los almacena correctamente en la base de datos SQLite (clima.db, tabla estaciones) y gestiona los errores de forma adecuada.
 - Bueno (12-15%): Funciona en su mayoría, con pequeños problemas en la recepción de datos, el almacenamiento o la gestión de errores.
 - Regular (8-11%): Funciona parcialmente, con problemas significativos en la recepción de datos, el almacenamiento o la gestión de errores.
 - Deficiente (0-7%): No funcional o incompleto.
- **Implementación y Exportación de TChart: (15%)**
 - Excelente (12-15%): Muestra los datos de 1 a 10 estaciones en un TChart con series de líneas, implementa el desplazamiento izquierdo (left scrolling) y exporta el gráfico a una imagen PNG secuencial. La interfaz de usuario es intuitiva y funcional.
 - Bueno (9-11%): Funciona en su mayoría, con pequeños problemas en la visualización del TChart, el desplazamiento, la exportación o la interfaz de usuario.
 - Regular (6-8%): Funciona parcialmente, con problemas significativos en la visualización del TChart, el desplazamiento, la exportación o la interfaz de usuario.
 - Deficiente (0-5%): No funcional o incompleto.
- **Cliente Python (Envío de Datos JSON): (15%)**
 - Excelente (12-15%): Envía correctamente datos ambientales simulados como una estructura JSON al servidor Lazarus a una frecuencia de 1 segundo. La estructura de datos coincide con el formato especificado.
 - Bueno (9-11%): Funciona en su mayoría, con pequeños problemas en la frecuencia de envío de datos, el formato JSON o la gestión de errores.
 - Regular (6-8%): Funciona parcialmente, con problemas significativos en la frecuencia de envío de datos, el formato JSON o la gestión de errores.
 - Deficiente (0-5%): No funcional o incompleto.

Proyecto 2: Servidor HTTP de Imágenes (50% de la Calificación Total)

- **Emisor de Imágenes en Python: (25%)**
 - Excelente (20-25%): Envía correctamente las imágenes de la carpeta 'img' al servidor Lazarus utilizando HTTP POST cada 1 segundo. Maneja la lectura de archivos y las solicitudes HTTP de manera eficiente.
 - Bueno (15-19%): Funciona en su mayoría, con pequeños problemas en la frecuencia de envío de imágenes, el manejo de archivos o las solicitudes HTTP.
 - Regular (10-14%): Funciona parcialmente, con problemas significativos en la frecuencia de envío de imágenes, el manejo de archivos o las solicitudes HTTP.
 - Deficiente (0-9%): No funcional o incompleto.
- **Servidor Lazarus (Recepción y Visualización de Imágenes): (25%)**
 - Excelente (20-25%): Recibe imágenes, las muestra en las celdas disponibles y reemplaza las imágenes aleatoriamente si todas las celdas están ocupadas. La interfaz de usuario es similar al prototipo y funciona correctamente.
 - Bueno (15-19%): Funciona en su mayoría, con pequeños problemas en la recepción de imágenes, la visualización, la gestión de celdas o la interfaz de usuario.
 - Regular (10-14%): Funciona parcialmente, con problemas significativos en la recepción de imágenes, la visualización, la gestión de celdas o la interfaz de usuario.
 - Deficiente (0-9%): No funcional o incompleto.

Consideraciones Generales:

- **Calidad del Código:** (Bonificación/Penalización - hasta el 10%): Evalúa la legibilidad, el mantenimiento y la eficiencia del código. Deduce puntos por malas prácticas de codificación, falta de comentarios o algoritmos ineficientes. Agrega puntos por código excepcionalmente bien escrito y optimizado.
- **Manejo de Errores:** (Importante): Evalúa qué tan bien los programas manejan posibles errores (por ejemplo, problemas de red, datos no válidos). Un manejo robusto de errores es crucial.
- **Cumplimiento de los Requisitos:** (Importante): Los programas deben cumplir con todos los requisitos especificados.
- **Integridad:** Todos los componentes de ambos proyectos deben ser entregados.

Conversión de la Escala de Calificación:

Para alinearse con la escala de 1 a 7 donde el 60% = 4.0:

| | |
|------------|-------------|
| 0% - 59% | = 1.0 - 3.9 |
| 60% - 69% | = 4.0 - 4.6 |
| 70% - 79% | = 4.7 - 5.3 |
| 80% - 89% | = 5.4 - 6.2 |
| 90% - 100% | = 6.3 - 7.0 |

Cómo Utilizar la Rúbrica:

1. Evalúa cada componente de cada proyecto según las descripciones de la rúbrica.
2. Asigna un porcentaje a cada componente.
3. Calcula la puntuación ponderada para cada proyecto (Puntuación del Proyecto 1 * 0.5 + Puntuación del Proyecto 2 * 0.5).
4. Aplica cualquier punto de bonificación o penalización por la calidad del código.
5. Convierte la puntuación porcentual final a la escala de 1 a 7 utilizando la tabla de conversión.