Laboratorio 2 - Parte 2: Algoritmos de detección
Redes

Hugo Rivas - 22500
Alexis Mesias - 22562

1. **Descripción de la práctica**

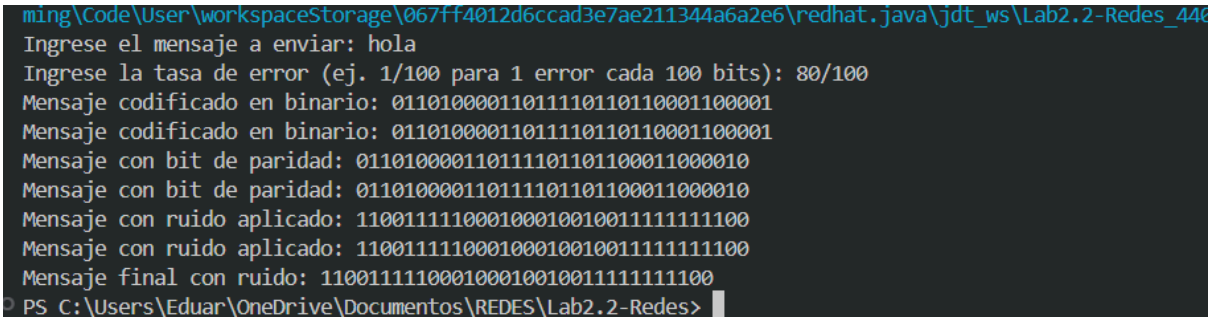
En esta segunda parte del laboratorio sobre esquemas de detección y corrección de errores, se implementó la aplicación del lado del emisor utilizando una arquitectura por capas. El objetivo principal fue simular el proceso de transmisión de datos en un canal no confiable, aplicando un esquema de detección de errores y ruido aleatorio.

La implementación se dividió en cuatro capas:

- **Aplicación**: solicitó al usuario un mensaje de texto a enviar.
- **Presentación**: codificó el mensaje a formato binario utilizando el código ASCII.
- **Enlace**: añadió un bit de paridad simple al mensaje binario para permitir la detección de errores.
- **Ruido**: simuló un canal no confiable aplicando errores aleatorios (flip de bits) con una probabilidad predefinida.

El resultado fue la generación de un mensaje alterado que representa la transmisión a través de un canal con ruido, dejando listo el sistema para ser recibido y evaluado por el receptor en la siguiente fase del laboratorio.

2. **Resultados**



3. **Discusión**

En esta práctica se observó el proceso completo desde la entrada del mensaje hasta su alteración por ruido simulado. Al enviar la palabra *"Hola"*, esta fue correctamente codificada en binario y se le añadió un bit de paridad como verificación de integridad. Posteriormente, se aplicó una tasa de error del 1%, lo que provocó la alteración de uno o más bits en la cadena transmitida.

Este resultado evidencia cómo incluso un canal con baja probabilidad de error puede afectar la integridad del mensaje. Aunque el sistema aún no corrige los errores, el uso de paridad permite detectarlos y justifica la necesidad de implementar una capa receptora con mecanismos de verificación más robustos para evitar pérdidas de información.

4. **Conclusiones**

1. La arquitectura en capas facilita la organización del proceso de transmisión, permitiendo modular el manejo de codificación, verificación y ruido.

2. La introducción de ruido incluso con baja probabilidad (1%) puede afectar significativamente la integridad del mensaje.

3. El uso de un bit de paridad permite detectar errores simples, pero no es suficiente para corregirlos, lo que resalta la importancia de implementar esquemas de corrección en etapas posteriores.

5. **Citas y Referencias**

Citas

1. “One extra bit is sent along with the original bits to make number of 1s either even in case of even parity, or odd in case of odd parity.”
 — Tutorials Point (n.d.) (TutorialsPoint)

2. “Parity bit method is a simple method which uses only one bit extra parity with the message bits to make codeword at sender side. It is used to detect errors of odd number of bits at receiver side.”
 — GeeksforGeeks (2025, July 15) (GeeksforGeeks)

Referencias

- Tutorials Point. (n.d.). *Error Detection & Correction*. Retrieved July 25, 2025, from https://www.tutorialspoint.com/data_communication_computer_network/error_detection_and_correction.htm (TutorialsPoint)

- GeeksforGeeks. (2025, July 15). *Error Detection Codes: Parity Bit Method*. Retrieved July 25, 2025, from https://www.geeksforgeeks.org/digital-logic/error-detection-codes-parity-bit-method/ (GeeksforGeeks)