

Laboratorio #2

Parte 2

Esquemas de detección y corrección de errores

Enlace al repositorio

Puede acceder al repositorio haciendo clic [aquí](#).

Descripción de la práctica

Consiste en la implementación y evaluación de algoritmos para detección y corrección de errores por medio de una aplicación de transmisión y detección de mensajes. Específicamente se desarrolló el algoritmo de Hamming para la corrección de errores y CRC-32 para la detección.

Dentro de la aplicación, se simula la comunicación entre emisor y receptor aplicando ruido a los mensajes transmitidos para evaluar las respuestas de los algoritmos. Esto implicó el uso de sockets para la transmisión de datos y la comprensión del modelo de capas como arquitectura.

Descripción de las pruebas

En total, se realizaron **80 pruebas**:

- Con longitud de 4 bits, fueron 40 pruebas.
 - Con una probabilidad de ruido del 25% fueron 20 pruebas (10 de Hamming y 10 de CRC-32).
 - Con una probabilidad de ruido del 50% fueron 20 pruebas (10 de Hamming y 10 de CRC-32).
- Con longitud de 11 bits, fueron 40 pruebas.

- Con una probabilidad de ruido del 25% fueron 20 pruebas (10 de Hamming y 10 de CRC-32).
- Con una probabilidad de ruido del 50% fueron 20 pruebas (10 de Hamming y 10 de CRC-32).

Estas se registraron con un formato de dos columnas, donde:

- ¿Ruido? = Indica si hubo ruido en la prueba.
 - “Sí” = Sí hubo ruido
 - “No” = No hubo ruido
- ¿Decoding? = Indica si la prueba fue decodificada con éxito.
 - “Sí” = Sí fue decodificada con éxito
 - “No” = No fue decodificada con éxito

Las pruebas realizadas de las distintas longitudes pueden ser consultadas en los **anexos** (*Cuadro 09 y Cuadro 10*).

Resultados

Resultados en cadenas de 4 bits

Longitud: 4 bits								
No:	Probabilidad de ruido							
	25%				50%			
	¿Ruido?		¿Decoding?		¿Ruido?		¿Decoding?	
	Hamming	CRC-32	Hamming	CRC-32	Hamming	CRC-32	Hamming	CRC-32
1	Sí	No	No	Sí	Sí	Sí	No	No
2	No	No	Sí	Sí	No	Sí	Sí	Sí
3	No	No	Sí	Sí	No	Sí	Sí	No
4	No	Sí	Sí	No	No	No	Sí	Sí
5	No	No	Sí	Sí	Sí	Sí	No	Sí
6	No	Sí	Sí	No	Sí	No	No	Sí
7	No	No	Sí	Sí	Sí	Sí	No	Sí
8	No	Sí	Sí	No	Sí	No	No	Sí
9	No	No	Sí	Sí	No	Sí	Sí	Sí
10	No	No	Sí	Sí	No	Sí	Sí	No

Cuadro 01. Resultados de las pruebas de 4 bits realizadas a los algoritmos Códigos de Hamming y CRC-32 con diferentes probabilidades de ruido.

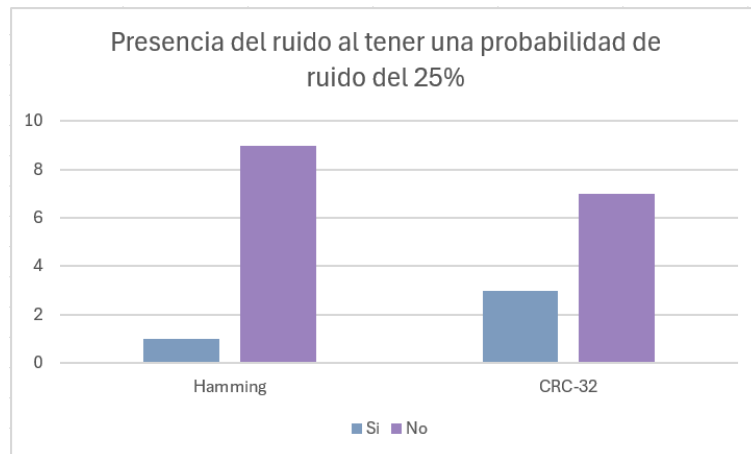


Gráfico 01. Presencia del ruido en las pruebas realizadas de 4 bits con un ajuste de probabilidad de ruido del 25%.

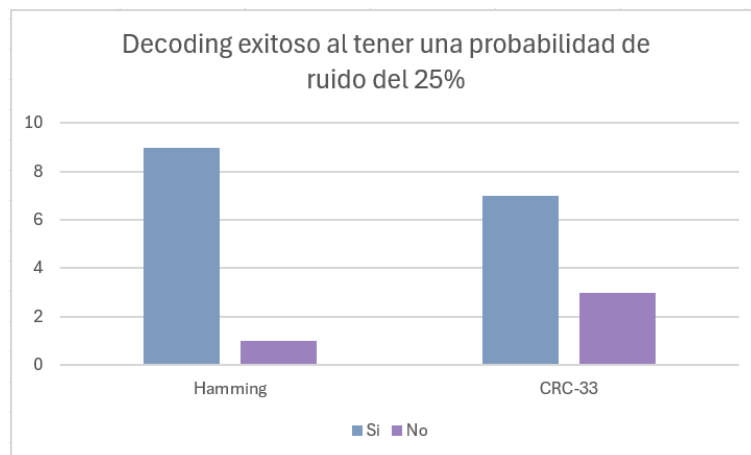


Gráfico 02. Éxito de la decodificación en las pruebas realizadas de 4 bits con un ajuste de probabilidad de ruido del 25%.

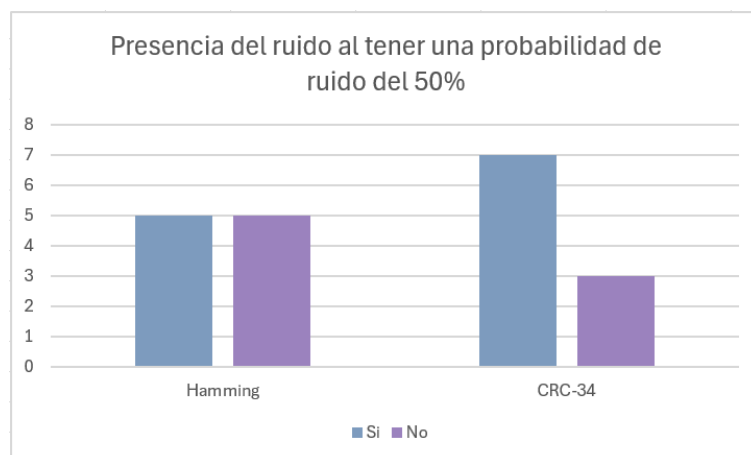


Gráfico 03. Presencia del ruido en las pruebas realizadas de 4 bits con un ajuste de probabilidad de ruido del 50%.

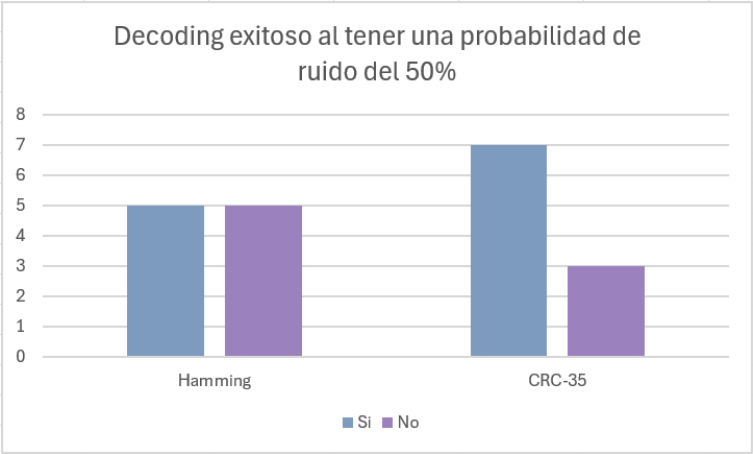


Gráfico 04. Éxito de la decodificación en las pruebas realizadas de 4 bits con un ajuste de probabilidad de ruido del 50%.

Resultados en cadenas de 11 bits

Longitud: 11 bits								
No:	Probabilidad de ruido							
	25%				50%			
	¿Ruido?		¿Decoding?		¿Ruido?		¿Decoding?	
	Hamming	CRC-32	Hamming	CRC-32	Hamming	CRC-32	Hamming	CRC-32
1	No	No	Si	Si	Si	Si	No	Si
2	No	No	Si	Si	No	No	Si	Si
3	No	Si	Si	No	No	Si	Si	No
4	No	Si	Si	No	Si	No	Si	Si
5	No	No	Si	Si	Si	No	No	Si
6	No	No	Si	Si	No	No	Si	Si
7	No	No	No	Si	No	No	Si	Si
8	Si	Si	Si	No	Si	No	Si	Si
9	Si	No	No	Si	No	Si	Si	Si
10	No	No	Si	Si	Si	Si	Si	Si

Cuadro 02. Resultados de las pruebas de 11 bits realizadas a los algoritmos Códigos de Hamming y CRC-32 con diferentes probabilidades de ruido.

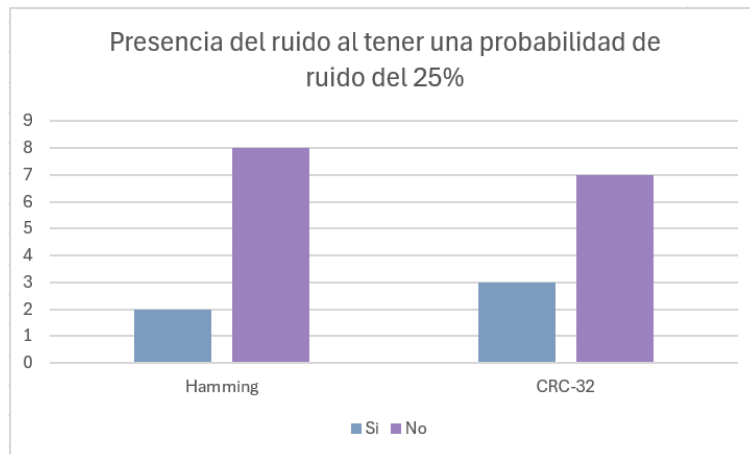


Gráfico 05. Presencia del ruido en las pruebas realizadas de 11 bits con un ajuste de probabilidad de ruido del 25%.

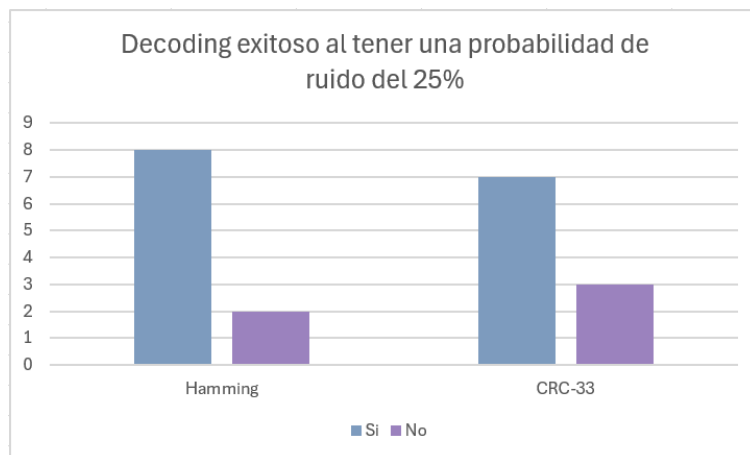


Gráfico 06. Éxito de la decodificación en las pruebas realizadas de 11 bits con un ajuste de probabilidad de ruido del 25%.

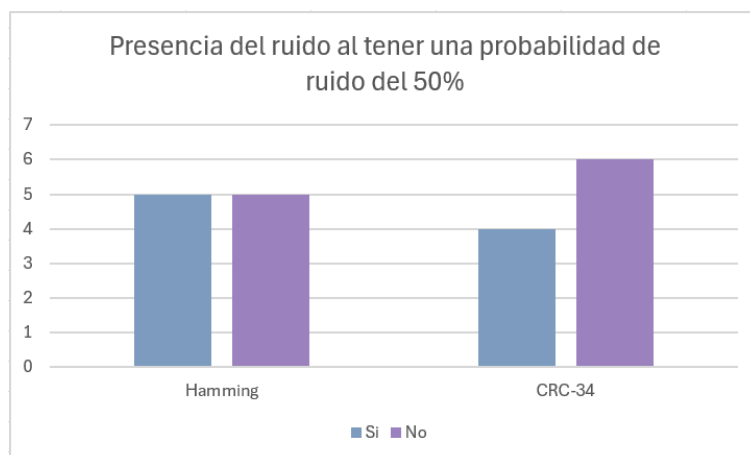


Gráfico 07. Presencia del ruido en las pruebas realizadas de 11 bits con un ajuste de probabilidad de ruido del 50%.

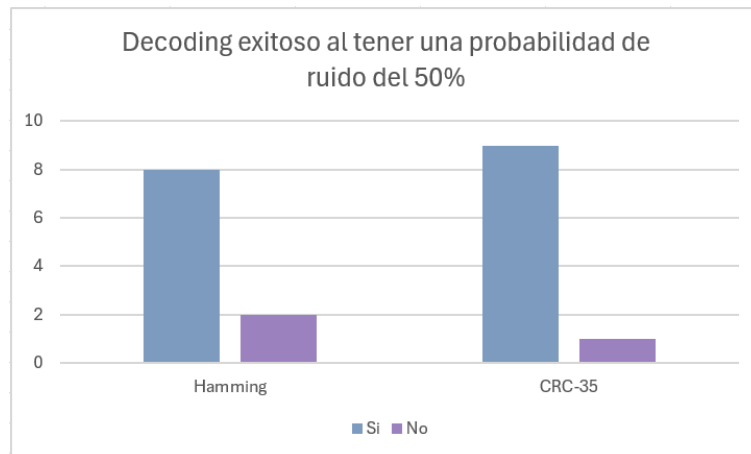


Gráfico 08. Éxito de la decodificación en las pruebas realizadas de 11 bits con un ajuste de probabilidad de ruido del 50%.

Discusión

Es bien sabido que para la transmisión de datos, la efectividad de los algoritmos de corrección y detección de errores es crucial para asegurar la integridad de la información. Por lo cual, se realizaron pruebas que abarcan una amplia variedad de condiciones, evaluando cómo los códigos de Hamming y CRC-32 manejan errores bajo diferentes longitudes de mensaje y tasas de ruido.

Los códigos de Hamming, diseñados para corregir errores de un solo bit, mostraron una capacidad relativamente buena de mantener la integridad de los datos en mensajes cortos, incluso con altas tasas de ruido. Sin embargo, su rendimiento disminuyó con el aumento de la longitud del mensaje y con tasas de ruido más elevadas, lo que evidencia sus limitaciones en contextos de mayor complejidad.

Por otro lado, el CRC-32, que es ampliamente utilizado para la detección de errores, demostró ser extremadamente eficaz en identificar errores en los mensajes, aunque carece de la capacidad de corregirlos. Esto limita su utilidad

en aplicaciones donde no es viable una retransmisión de datos para corregir errores detectados.

La introducción de ruido en la transmisión simulada mostró de manera efectiva cómo cada algoritmo maneja las alteraciones en los datos. Ambos algoritmos experimentaron una reducción en la efectividad conforme la probabilidad de ruido aumentaba, subrayando la necesidad de estrategias de manejo de errores más robustas en entornos altamente ruidosos.

Comentario grupal sobre el tema

La práctica nos resultó útil para reforzar los conocimientos sobre los algoritmos de corrección y detección de errores, pues ya se poseían conocimientos previos, pero al tener el reto de unir emisor y receptor de diferentes algoritmos nos dio otra perspectiva sobre el tema y por ende nos ayudó a terminar de comprender temas y conceptos que no estaban del todo claros.

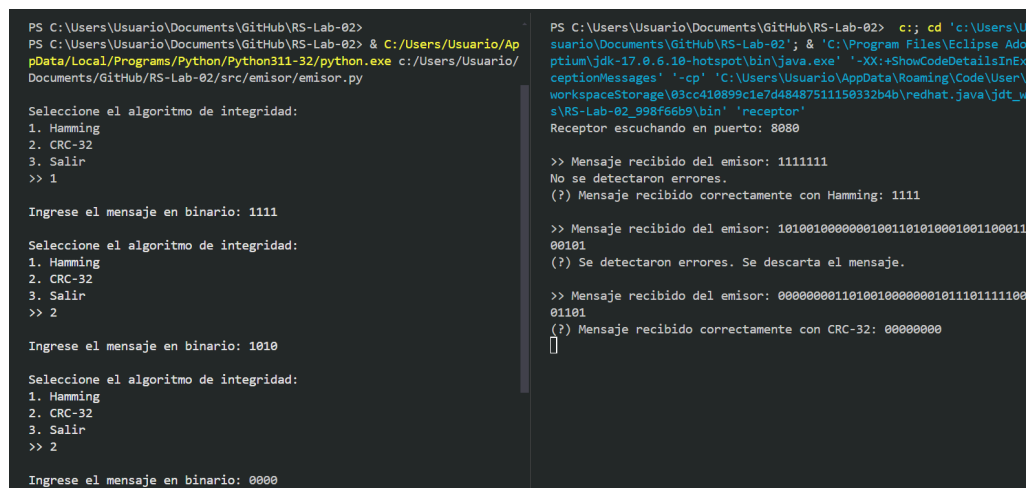
Sin embargo, fue desafiante la realización de las pruebas con ambos algoritmos, por lo que para prácticas futuras se recomienda buscar una forma de automatizar dichas pruebas con el fin de ahorrar tiempo y recursos.

Conclusiones

1. El código de Hamming es más efectivo con mensajes cortos y niveles de ruido moderados.
2. CRC-32 es eficaz en la detección de errores pero no ofrece corrección.
3. Ambos algoritmos sufren una reducción en su eficacia con el incremento del ruido.
4. El overhead de Hamming es considerable, especialmente en mensajes largos.
5. La longitud del mensaje influye significativamente en la eficacia de ambos algoritmos.
6. Es necesario equilibrar la eficacia de la detección y corrección de errores con el overhead de comunicación.

7. La adaptabilidad de los algoritmos a diferentes condiciones de ruido es clave para su efectividad práctica.
8. La selección del algoritmo debe basarse en el equilibrio entre la necesidad de corrección frente a la detección de errores y las características del canal de transmisión.

Anexos



```

PS C:\Users\Usuario\Documents\GitHub\RS-Lab-02>
PS C:\Users\Usuario\Documents\GitHub\RS-Lab-02> & C:/Users/Usuario/AppData/Local/Programs/Python/Python311-32/python.exe c:/Users/Usuario/Documents/GitHub/RS-Lab-02/src/emisor/emisor.py

Seleccione el algoritmo de integridad:
1. Hamming
2. CRC-32
3. Salir
>> 1

Ingrese el mensaje en binario: 1111

Seleccione el algoritmo de integridad:
1. Hamming
2. CRC-32
3. Salir
>> 2

Ingrese el mensaje en binario: 1010

Seleccione el algoritmo de integridad:
1. Hamming
2. CRC-32
3. Salir
>> 2

Ingrese el mensaje en binario: 0000
  
```

```

PS C:\Users\Usuario\Documents\GitHub\RS-Lab-02> c:: cd 'c:\Users\U
uario\Documents\GitHub\RS-Lab-02'; & 'C:\Program Files\Eclipse Ado
ptium\jdk-17.0.6.10-hotspot\bin\java.exe' '-XX:+ShowCodeDetailsInEx
ceptionMessages' '-cp' 'C:\Users\Usuario\AppData\Roaming\Code\User\
workspaceStorage\03cc410899c1e7d48487511150332b4b\redhat.java\jdt_w
s\RS-Lab-02_098f66b9\bin' 'receptor'
Receptor escuchando en puerto: 8080

>> Mensaje recibido del emisor: 11111111
No se detectaron errores.
(?) Mensaje recibido correctamente con Hamming: 1111

>> Mensaje recibido del emisor: 1010010000001001101010001001100011
00101
(?) Se detectaron errores. Se descarta el mensaje.

>> Mensaje recibido del emisor: 0000000011010010000000101110111100
01101
(?) Mensaje recibido correctamente con CRC-32: 00000000
  
```

Figura 01. Captura de la ejecución del programa con dos códigos sin ruido y uno con ruido.

No	Mensaje
1	1010
2	0111
3	1101
4	0010
5	1001
6	0100
7	1110
8	0001
9	1011
10	0110

Cuadro 09. Pruebas realizadas con longitud de 4 bits.

No	Mensaje
1	10101100101
2	01111010011
3	11010101110
4	00100101101
5	10011101010
6	01001010111
7	11101100010
8	00011011011
9	10111001100
10	01100111101

Cuadro 10. Pruebas realizadas con longitud de 11 bits.

Referencias

1. GeeksforGeeks. (2024, 26 julio). Hamming Code in Computer Network. GeeksforGeeks.
<https://www.geeksforgeeks.org/hamming-code-in-computer-network/>
2. Understanding CRC32. (s. f.).
<https://commandlinefanatic.com/cgi-bin/showarticle.cgi?article=art008>