

## RETO 4 (REFORMULADO)

# Canal con Corrección de Errores Hamming

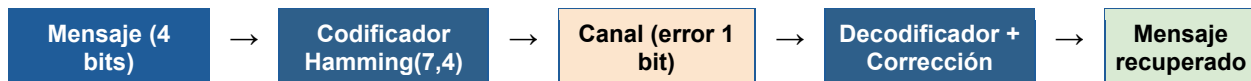
Contenido: Clase 7 — Códigos Correctores de Errores

## 1. Descripción del Reto

Los sistemas de comunicación digital enfrentan constantemente el problema de la corrupción de datos durante la transmisión. En este reto, los estudiantes diseñarán e implementarán un sistema completo de codificación y corrección de errores basado en el Código Hamming(7,4), cubriendo todo el flujo: codificación, simulación de canal con error, detección y corrección mediante síndrome.

## 2. Contexto del Problema

Un sistema de transmisión de datos envía paquetes de 4 bits a través de un canal ruidoso. Para proteger la información, el transmisor aplica el Código Hamming(7,4) antes de enviar. El canal puede introducir errores de un solo bit. El receptor debe ser capaz de detectar la posición del error, corregirlo y recuperar el mensaje original.



## 3. Actividades

### 3.1 Distancia de Hamming y Capacidad del Código

Calcule la distancia de Hamming entre cada par de cadenas binarias de la tabla siguiente. Para cada caso, indique cuántos errores puede detectar y cuántos puede corregir un código con esa distancia mínima.

| # | Cadena A      | Cadena B      | dH(A,B) = ? | Errores detectables | Errores corregibles |
|---|---------------|---------------|-------------|---------------------|---------------------|
| 1 | 1 0 1 1 0 1 0 | 1 0 1 1 1 0 0 |             |                     |                     |
| 2 | 1 1 0 0 0 1 1 | 1 0 0 0 1 1 1 |             |                     |                     |
| 3 | 0 1 1 0 1 0 1 | 0 1 1 1 0 0 1 |             |                     |                     |

#### Pregunta de análisis:

Justifique matemáticamente por qué el Código Hamming(7,4) necesita exactamente 3 bits de paridad. Utilice la fórmula  $2^r \geq m + r + 1$  y muestre la verificación para  $r = 2$  y  $r = 3$ .

| r<br>(paridad) | $2^r$ | $m + r + 1$ (con $m=4$ ) | ¿Se cumple $2^r \geq m+r+1$ ?                |
|----------------|-------|--------------------------|----------------------------------------------|
| $r = 2$        | 4     | $4 + 2 + 1 = 7$          | $4 \geq 7 \rightarrow \text{NO}$             |
| $r = 3$        | 8     | $4 + 3 + 1 = 8$          | $8 \geq 8 \rightarrow \text{SÍ } \checkmark$ |

## Codificación: Hamming(7,4)

### 3.2 Codificación de Mensajes

Codifique cada uno de los siguientes mensajes de 4 bits utilizando el Código Hamming(7,4). Para cada mensaje, muestre paso a paso el cálculo de los bits de paridad P1, P2 y P4, y la trama final de 7 bits.

**Recordatorio — Estructura Hamming(7,4):** Las posiciones 1, 2 y 4 son bits de paridad (P1, P2, P4). Las posiciones 3, 5, 6, 7 contienen los datos (D1, D2, D3, D4). P1 controla posiciones {1,3,5,7}. P2 controla posiciones {2,3,6,7}. P4 controla posiciones {4,5,6,7}.

| # | Mensaje (D1 D2 D3 D4) | Trabajo a mostrar (P1, P2, P4, trama final)                                                                   |
|---|-----------------------|---------------------------------------------------------------------------------------------------------------|
| A | 1 0 1 0               | Calcule P1, P2, P4. Complete la trama: P1 P2 D1 P4 D2 D3 D4 = _ _ 1 _<br>0 1 0<br>Código Hamming final: _____ |

|   |         |                                                                                                                              |
|---|---------|------------------------------------------------------------------------------------------------------------------------------|
| B | 1 1 0 1 | <p>Calcule P1, P2, P4. Complete la trama: P1 P2 D1 P4 D2 D3 D4 = _ _ 1 _</p> <p>1 0 1</p> <p>Código Hamming final: _____</p> |
| C | 0 0 1 1 | <p>Calcule P1, P2, P4. Complete la trama: P1 P2 D1 P4 D2 D3 D4 = _ _ 0 _</p> <p>0 1 1</p> <p>Código Hamming final: _____</p> |

## 3.3 Síndrome de Error y Corrección

El receptor recibe los siguientes códigos Hamming de 7 bits. En cada caso, un bit fue alterado durante la transmisión. Para cada código recibido:

- Recalcule P1, P2 y P4 a partir del código recibido.
- Construya el síndrome de error ( $S = P4 P2 P1$  en binario).
- Convierta el síndrome a decimal para identificar la posición del error.
- Corrija el bit erróneo y recupere el mensaje original (posiciones 3, 5, 6, 7).

| # | Código recibido (con 1 error) | Análisis del síndrome                                                                                                                                                                                                                                                                                          |
|---|-------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | 1 0 1 1 1 1 0                 | P1 (pos. 1,3,5,7): resultado = ____ ¿Falla? ____<br>P2 (pos. 2,3,6,7): resultado = ____ ¿Falla? ____<br>P4 (pos. 4,5,6,7): resultado = ____ ¿Falla? ____<br>Síndrome ( $P4 P2 P1$ ) = ____ → Decimal: ____ → Error en posición: ____<br>Mensaje corregido: _ _ _ _ _ Datos recuperados (pos. 3,5,6,7): _ _ _ _ |
| 2 | 1 0 0 0 1 0 1                 | P1 (pos. 1,3,5,7): resultado = ____ ¿Falla? ____<br>P2 (pos. 2,3,6,7): resultado = ____ ¿Falla? ____<br>P4 (pos. 4,5,6,7): resultado = ____ ¿Falla? ____<br>Síndrome ( $P4 P2 P1$ ) = ____ → Decimal: ____ → Error en posición: ____<br>Mensaje corregido: _ _ _ _ _ Datos recuperados (pos. 3,5,6,7): _ _ _ _ |
| 3 | 1 0 0 0 0 0 1                 | P1 (pos. 1,3,5,7): resultado = ____ ¿Falla? ____<br>P2 (pos. 2,3,6,7): resultado = ____ ¿Falla? ____<br>P4 (pos. 4,5,6,7): resultado = ____ ¿Falla? ____<br>Síndrome ( $P4 P2 P1$ ) = ____ → Decimal: ____ → Error en posición: ____<br>Mensaje corregido: _ _ _ _ _ Datos recuperados (pos. 3,5,6,7): _ _ _ _ |

## Implementación y Documentación

### 3.4 Simulador por Software

Implemente un simulador del sistema Hamming(7,4) en el lenguaje de programación de su elección (De preferencia Python en una jupyter notebook). El simulador debe:

- Recibir un mensaje de 4 bits como entrada.
- Calcular y agregar los bits de paridad P1, P2 y P4 para producir el código Hamming de 7 bits.
- Simular un error de 1 bit en una posición aleatoria o indicada por el usuario.
- Calcular el síndrome de error a partir del código recibido.

- Identificar y corregir el bit erróneo, mostrando el mensaje recuperado.

#### Casos de prueba obligatorios en el simulador:

| # | Mensaje entrada | Código esperado | Error en pos. | Mensaje recuperado |
|---|-----------------|-----------------|---------------|--------------------|
| 1 | 1010            | 1011010         | 5             | 1010               |
| 2 | 1101            | 1010101         | 3             | 1101               |
| 3 | 0011            | 1000011         | 6             | 0011               |
| 4 | Libre           |                 | Libre         |                    |

#### Documentación requerida (incluir en el informe):

- Descripción del algoritmo implementado.
  - Capturas de pantalla o salida de consola de los 4 casos de prueba.
  - Conclusiones: ventajas y limitaciones del Código Hamming(7,4).
  - Referencia a los conceptos de la Clase 7 aplicados.
-

## 4. Entregables

| # | Entregable                  | Contenido mínimo                                                    | Formato                   |
|---|-----------------------------|---------------------------------------------------------------------|---------------------------|
| 1 | Informe técnico             | Ejercicios de días 1, 2 y 3 resueltos con procedimiento paso a paso | PDF o Word                |
| 2 | Código fuente del simulador | Archivo fuente con los 4 casos de prueba ejecutables                | Jupyter notebook o script |
| 3 | Evidencia de ejecución      | Capturas de pantalla de la salida del simulador para los 4 casos    | Incluido en el informe    |

## 5. Rúbrica de Evaluación

Puntaje total: 50 puntos

| Criterio                                                                           | Puntaje máximo | Aspectos evaluados                                                       | Puntaje obtenido |
|------------------------------------------------------------------------------------|----------------|--------------------------------------------------------------------------|------------------|
| <b>Análisis teórico</b><br>Distancia de Hamming + justificación de bits de paridad | <b>10 pts</b>  | Cálculo correcto de dH (6 pts)<br>Justificación con fórmula (4 pts)      |                  |
| <b>Codificación Hamming(7,4)</b><br>3 mensajes codificados con procedimiento       | <b>10 pts</b>  | Cálculo de P1, P2, P4 (6 pts)<br>Trama final correcta (4 pts)            |                  |
| <b>Decodificación y corrección</b><br>3 ejercicios con síndrome de error           | <b>10 pts</b>  | Síndrome correcto (6 pts)<br>Mensaje recuperado correcto (4 pts)         |                  |
| <b>Implementación + Documentación</b><br>Simulador funcional e informe             | <b>20 pts</b>  | Código funcional con 4 pruebas (6 pts)<br>Informe y conclusiones (4 pts) |                  |
| <b>TOTAL</b>                                                                       | <b>50 pts</b>  |                                                                          |                  |