



**ORGANIZACION **  
**COMPUTACIONAL**

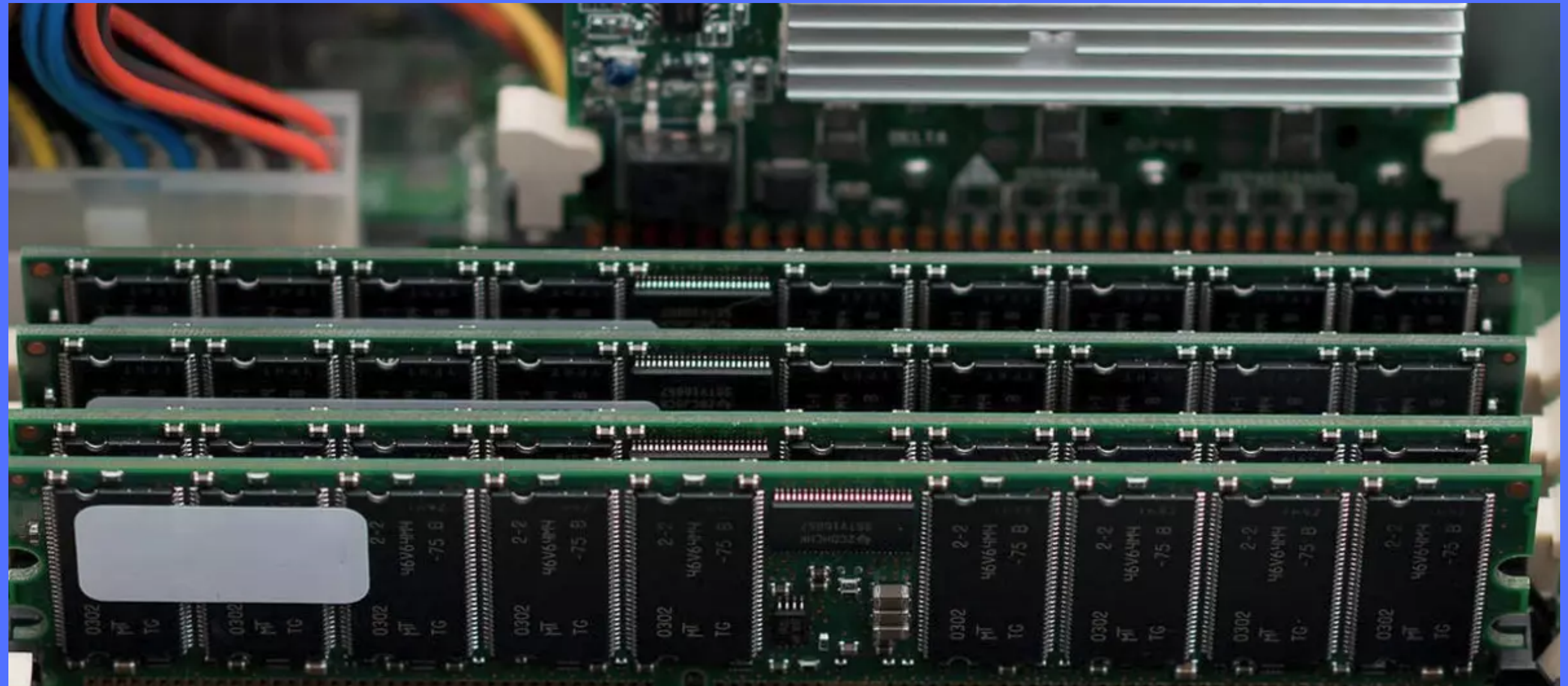
**CORRECCIONES DE ERRORES**

**DDR DRAM**

MANUEL EMILIO GONGORA CANTO



# ¿QUE ES UN ERROR DURO?



SE REFIERE A UN TIPO DE ERROR DE HARDWARE QUE AFECTA DE MANERA PERMANENTE A LA MEMORIA DDR (DOUBLE DATA RATE) DYNAMIC RANDOM ACCESS MEMORY. ESTE TIPO DE ERROR INDICA UN PROBLEMA FÍSICO O PERMANENTE EN LA MEMORIA QUE GENERALMENTE NO SE PUEDE CORREGIR MEDIANTE REINICIOS O AJUSTES SIMPLES DE SOFTWARE.

# ¿QUE ES UN ERROR SUAVE?

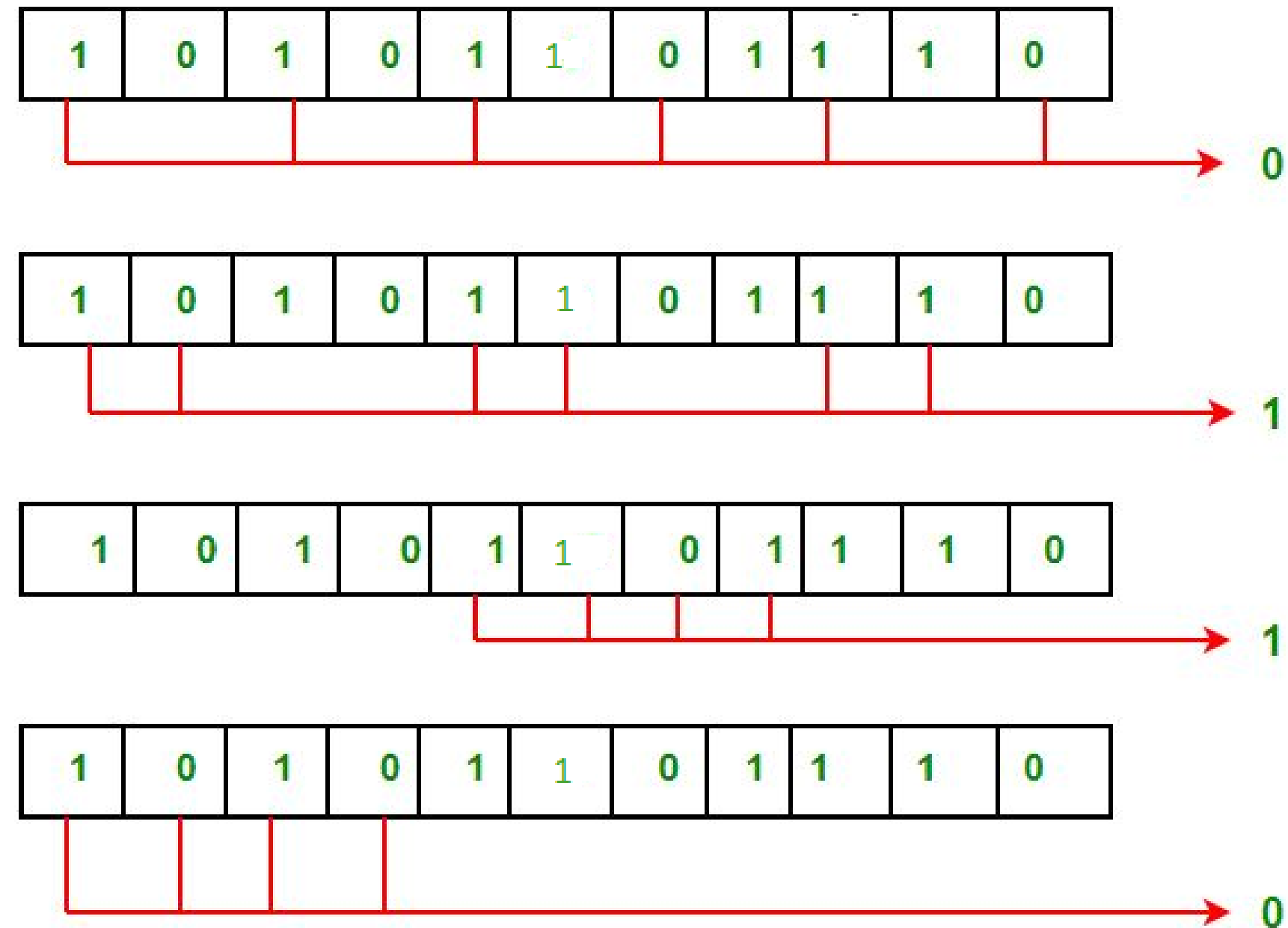
SE REFIERE A UN TIPO DE ERROR TEMPORAL O INTERMITENTE QUE PUEDE OCURRIR DEBIDO A DIVERSAS RAZONES, PERO QUE NO REPRESENTA UN PROBLEMA FÍSICO PERMANENTE EN LA MEMORIA. LOS ERRORES SUAVES SON GENERALMENTE MÁS TRANSITORIOS Y PUEDEN SER CORREGIDOS CON ACCIONES COMO REINICIOS DEL SISTEMA, RECONFIGURACIÓN DE LA MEMORIA O AJUSTES EN LA CONFIGURACIÓN DEL HARDWARE. AQUÍ HAY ALGUNAS CARACTERÍSTICAS Y CAUSAS COMUNES DE LOS ERRORES SUAVES EN LA DDR DRAM:





# ¿QUE SON LOS CODIGOS ERROR - CORRECTING?

TAMBIÉN CONOCIDOS COMO CÓDIGOS DE CORRECCIÓN DE ERRORES, SON TÉCNICAS UTILIZADAS EN LA TRANSMISIÓN Y ALMACENAMIENTO DE DATOS PARA DETECTAR Y CORREGIR ERRORES QUE PUEDAN OCURRIR DURANTE LA TRANSMISIÓN O LECTURA DE INFORMACIÓN.



# EXPLICA DE FORMA SENCILLA DEL CODIGO HAMMING

**Es un tipo de código de error-corrección utilizado para detectar y corregir errores en la transmisión o almacenamiento de datos. Fue desarrollado por Richard W. Hamming en la década de 1950 y es uno de los códigos más simples y efectivos para corregir errores de un solo bit.**



### 1.- BITS DE DATOS Y BITS DE PARIDAD:

- EN EL CÓDIGO DE HAMMING, COMENZAMOS CON UN CONJUNTO DE BITS DE DATOS QUE QUEREMOS TRANSMITIR O ALMACENAR.
- LUEGO, AÑADIMOS BITS DE PARIDAD ADICIONALES PARA PERMITIR LA DETECCIÓN Y CORRECCIÓN DE ERRORES.

### 2.- CÁLCULO DE LOS BITS DE PARIDAD:

- LOS BITS DE PARIDAD SE CALCULAN COLOCANDO LOS BITS DE DATOS EN POSICIONES ESPECÍFICAS SEGÚN UNA FÓRMULA MATEMÁTICA.

### 3.- DETECCIÓN Y CORRECCIÓN DE ERRORES:

- CUANDO SE TRANSMITE O ALMACENA UN CONJUNTO DE BITS (DATOS MÁS BITS DE PARIDAD), EL RECEPTOR CALCULA NUEVAMENTE LOS BITS DE PARIDAD

## EJEMPLO SIMPLIFICADO

1

- Supongamos que queremos enviar 4 bits de datos (D1, D2, D3, D4).
- Añadimos 3 bits de paridad (P1, P2, P3) de acuerdo con ciertas reglas de posicionamiento.

2

3

1 0 0 1 1 0 0

0 1 0 0 1 0 1

1 1 0 0 1 1 0

0 0 0 1 1 1 1

1 1 1 0 0 0 0

0 0 1 1 0 0 1

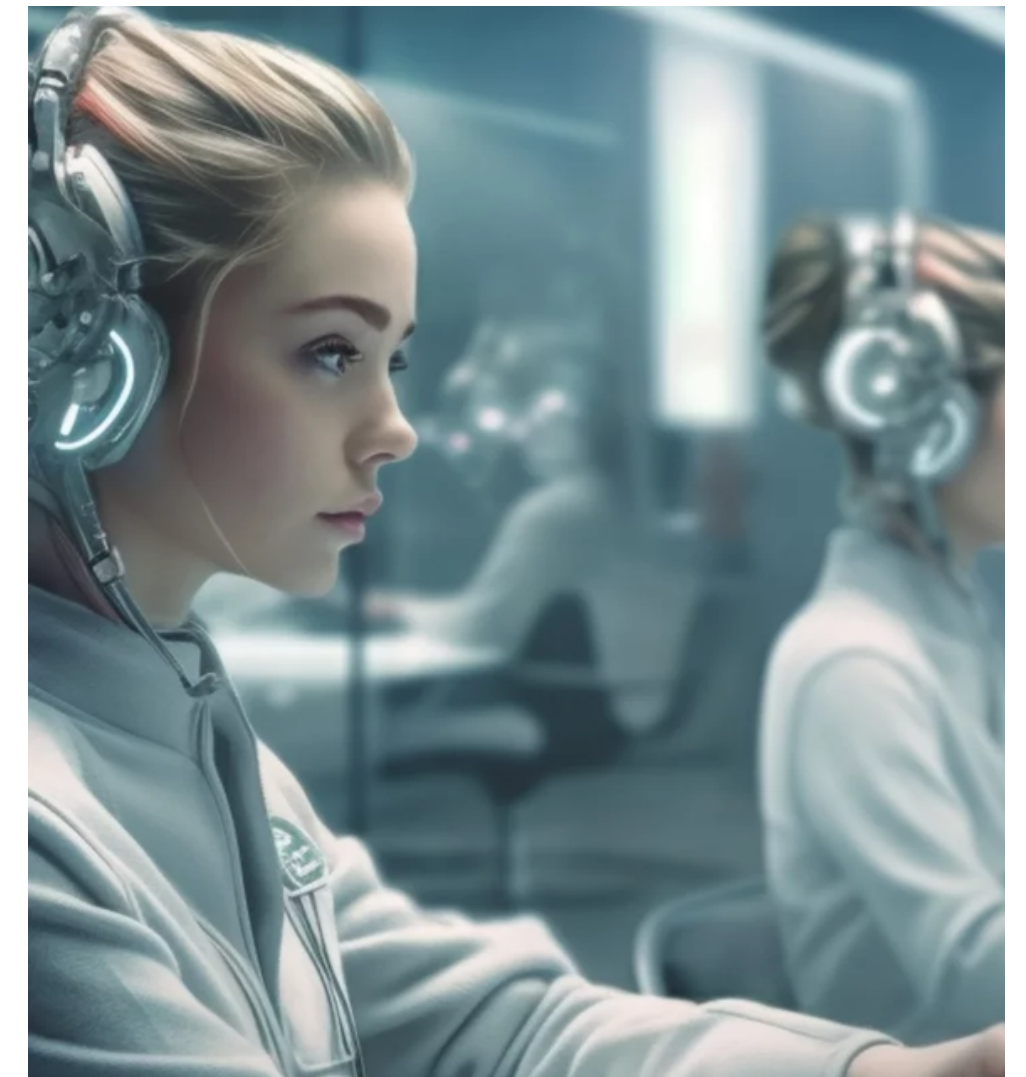
1 0 1 1 0 1 0

0 1 1 0 0 1 1

0 1 1 1 1 0 0

# **EXPLICA DE FORMA SENCILLA EL CODIGO SEC**

**ES UN TIPO DE CÓDIGO DE DETECCIÓN Y CORRECCIÓN DE ERRORES DISEÑADO PARA DETECTAR Y CORREGIR UN ÚNICO ERROR EN UN CONJUNTO DE DATOS. ESTE CÓDIGO SE UTILIZA EN SISTEMAS DONDE ES IMPORTANTE GARANTIZAR LA INTEGRIDAD DE LOS DATOS Y CORREGIR ERRORES QUE PUEDAN OCURRIR DURANTE LA TRANSMISIÓN O ALMACENAMIENTO.**





# EXPLICA DE FORMA SENCILLA EL CODIGO SEC - DED

EL CÓDIGO SEC-DED (SINGLE ERROR CORRECTION, DOUBLE ERROR DETECTION) ES UNA EXTENSIÓN DEL CÓDIGO SEC (SINGLE ERROR CORRECTION) QUE PUEDE DETECTAR LA PRESENCIA DE HASTA DOS ERRORES EN UN CONJUNTO DE DATOS Y CORREGIR UN SOLO ERROR. ESTE TIPO DE CÓDIGO DE ERROR-CORRECCIÓN ES COMÚNMENTE UTILIZADO EN SISTEMAS DONDE LA INTEGRIDAD DE LOS DATOS ES CRÍTICA Y SE NECESITA UNA ALTA FIABILIDAD EN LA DETECCIÓN Y CORRECCIÓN DE ERRORES.

2 B		1 B		1 B		29 B					
e bits .1010	Sync word	Header		Data field (payload data)							
		length field	address field								
$p,0$	$H(0,op)$	$V(3,0)$	$V(2,0)$	$V(1,0)$	$V(0,0)$	$H(0,3)$	$H(0,2)$	$H(0,1)$	$H(0,0)$	$d(0,10)$	...
$p,1$	$H(1,op)$	$V(3,1)$	$V(2,1)$	$V(1,1)$	$V(0,1)$	$H(1,3)$	$H(1,2)$	$H(1,1)$	$H(1,0)$	$d(1,10)$	...
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	...
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	...
$p,10$	$H(10,op)$	$V(3,10)$	$V(2,10)$	$V(1,10)$	$V(0,10)$	$H(10,3)$	$H(10,2)$	$H(10,1)$	$H(10,0)$	$d(10,10)$	...
$OP_H$		Hamming code per vertical direction				Hamming code per horizontal direction				resolution of sens $SE_i \ (0 \leq i \leq k, -1 k, =1$	