

Código de Hamming: Detección y Corrección de errores

Quién iba a decir que fuera un juego -como lo lees un juego- el algoritmo de Hamming, que lo que trata es de corregir errores, sea como un juego de deducción -como el juego del Cluedo o el Sudoku- donde nos dan una serie de pistas iniciales, y necesitamos descubrir quién es el malo o que número va en cada posición por descarte.

Parece un sueño que si enviamos unos bits (una imagen, un audio, un texto, lo que sea) por Internet. Cuando llegue a su destino, y llegue mal (bits cambiados aleatoriamente; donde antes había un "1" ahora un "0", y viceversa) por alguna causa -como una tormenta electromagnética por el camino entre el ordenador que lo envía y el que lo recibe- podamos corregir los errores que han ocurrido por el camino en el destino.

Este algoritmo es, por así decirlo, un juego de detectives. Donde tenemos varias personas (los bits), y hay uno, varios o ningún malo (bits erróneos) entre ellas. A su vez tenemos varias pistas que van a hacer que descartemos a los buenos, hasta quedarnos con los malos; aunque ojo, también puede haber pistas que mientan. Realmente es bastante sencillo.

Es decir, hay **unos bits que corrigen a otros**. Estos bits que corrigen a otros se llaman **"bits de redundancia"**, pues son **bits extra a la información para intentar corregir los errores o al menos comprobar que todo esté correcto**. Un ejemplo en el mundo físico, sería como tener en un papel un texto cualquiera como "Mi mamá me mima", y más abajo en el mismo papel la suma de todas las letras del texto "M=6, I=2, A=3, E=1" (es información repetida/redundante, pues ya está contenido en el dato). De este modo si se nos cae líquido encima de la tinta del texto haciendo ilegible algunas palabras del texto "Xi XaXá me Xima" (las "X" representan la ilegibilidad del texto); todavía podemos recuperar el texto original con la suma de letras pues sabemos que hay 6 "Ms" y solo vemos legibles 2, por lo que se deduce que las letras ilegibles tienen que ser "Ms" también (lo mismo ocurriría si las equis "X" anteriores fueran letras y no borrones, la letra equis "X" no existe en las letras sumadas por lo que es una letra errónea). Al **dato original más los "bits de redundancia"** se les conoce como **"Bloque del código"** (en inglés "Block Code").

La "distancia mínima de Hamming" es muy interesante para corregir errores en bits (más información en el artículo de Hamming (<https://jarroba.com/hamming/>); te recomiendo que lo leas a continuación de éste artículo). Realmente sirve para corregir cualquier tipo de fichero (audio, imagen, etc) en informática, pues los bits son la base de cualquier fichero, y si los bits están correctos pues el fichero debería de estar correcto (debería, pues si, por ejemplo, con un ataque "man in the middle", un tercero modifica el fichero por el camino del fichero durante el envío; este método corregirá los errores del fichero si llegan mal al destino entre el tercero y el destino, pero no detecta si el contenido ha sido modificado deliberadamente).

Código Hamming

Para empezar con cualquier juego necesitamos unas normas. Y las normas de Hamming son las tres siguientes (si estas instrucciones no te aclaran mira la imagen que viene después, y luego vuelve, lo vas a entender enseguida):

1. Utilizaremos como **"bits de redundancia"** los conocidos como **"bits de paridad"** para comprobar si hay errores o no (serán las **pistas**): "1" significa "impar", y "0" significa "par" (el «Bit de Paridad» es la suma de "1s"; por ejemplo, la cadena de bits "1011" la

Uso de cookies

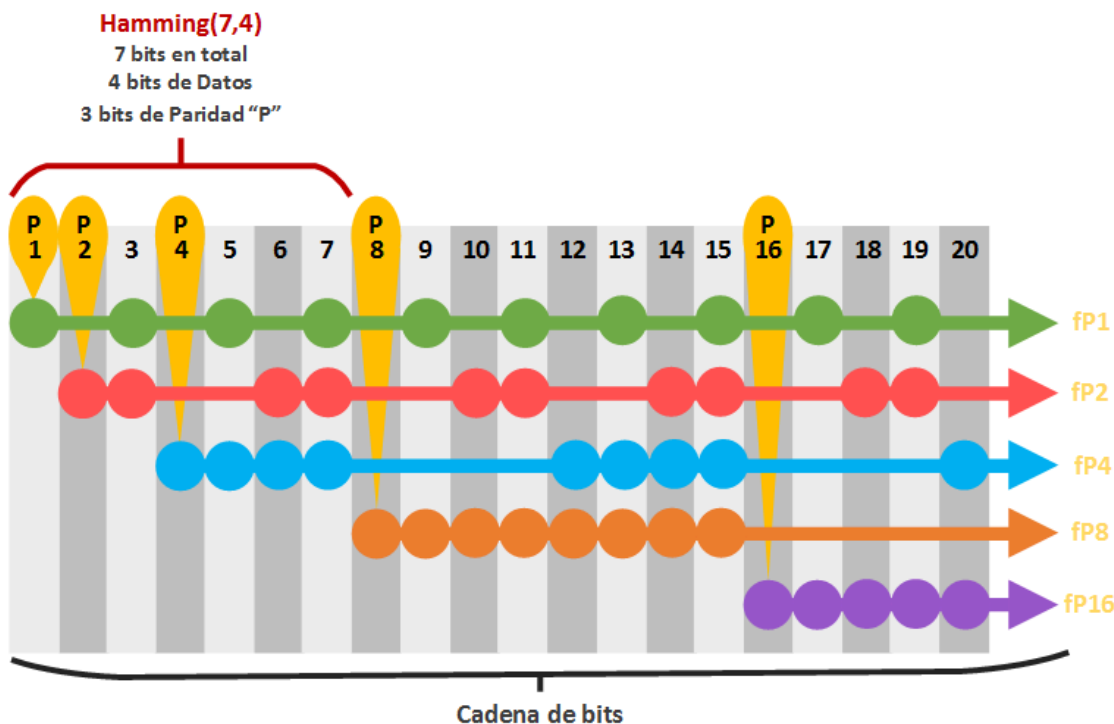
Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (<http://jarroba.com/faq/>)

ACEPTAR

suma de "1s" es 3, que es impar, por lo que el bit de paridad es "1". Si tienes curiosidad tienes más información sobre «bits de Paridad» en este otro artículo sobre «Dígitos de control» haciendo click aquí (<https://jarroba.com/digito-de-control-deteccion-de-errores/>)).

2. Cada "bit de paridad" comprobará unos bits determinados, dependiendo de la posición que ocupe y siguiendo las normas del siguiente ejemplo. Por ejemplo, la posición es 4, desde esta misma posición se seleccionarán 4, luego dejará 4 sin seleccionar, seleccionará los siguientes 4, para luego dejar los siguientes 4 sin seleccionar, y así hasta el infinito. Lo mismo para todas las posiciones que son potencias de dos (para la posición 1 es uno sí, uno no, uno sí, uno no, etc; ¿Y para la posición 2? Dos sí, dos no, dos sí, dos no; la cuatro te la he dicho antes, ¿Y para los siguientes 8, 16, 32, etc? es muy fácil).
3. De las anteriores selecciones el "bit de paridad" es de lo seleccionado la posición de número más pequeña (evidentemente, porque si el bit de paridad dice si es par o impar la suma de los "1s", si se comprobara a sí mismo, entraríamos en un bucle infinito sin sentido), el resto son los bits que comprueba (que serán los que se sumen para saber si es par o impar).

Siguiendo estas instrucciones podríamos dibujar hasta el infinito una matriz infinita. Pero vamos a reducirla hasta la posición 20 que puedes ver en la siguiente imagen (el resto hasta el infinito es igual 😊).



(https://jarroba.com/wp-content/uploads/2016/10/Hamming-Matriz-sindrome-www.jarroba.com_.png)

Verás que la **cadena de bits se compone de bits de datos y bits de paridad** mezclados (pero no revueltos 😊). Sabemos que **los bits de paridad son las potencias de dos**: 1, 2, 4, 8, 16 (el siguiente es 32, y el ejemplo solo tiene 20 bits, así que ya paro la cuenta). Al "bit de paridad" en sí lo he llamado en la imagen "P" seguido de la posición que ocupa (por ejemplo, el que ocupa la posición 4, lo he llamado "P4"). No hay que confundir con las **filas de bits que comprueba cada "bit de paridad" e incluyéndose** así mismo; a cada fila de éstas la he denominado en la imagen como "fP" seguido de la posición en la que empieza (como "fP4", que es la fila que comprueba el "bit de paridad" llamado "P4" e incluyéndose así mismo). Esto es una matriz vulgar y corriente, lo único que un poco más emocionante. Esta matriz se llama "**Matriz de comprobación de paridad**" (Parity-check Matrix) o "**Matriz Síndrome**" (Syndrome Matrix); a esta matriz a mí me gusta llamarla "Matriz de deducción" que suena mucho más misterioso y porque esta matriz nos servirá para deducir con las pistas los bits que están mal para poder corregirlos, o simplemente detectar si hay errores.

Seguro que te has fijado en que encima de la "matriz síndrome" anterior aparece la palabra "Hamming(7,4)". Lo he querido señalar porque es el código Hamming más típico. Aunque existen infinitos, pero siempre siguiendo las normas, otros podrían ser el:

- Hamming(3,1)
- Hamming(15,11): en la matriz de la anterior imagen llegaría justo hasta antes de la posición 16
- Hamming(31,26): llegaría justo hasta la posición anterior a la 32

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra política de cookies (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

ACEPTAR

- etc...

Centrémonos en “Hamming(7,4)”, que quiere decir que **por cada 7 bits que representan datos hay 4 que son bits de Datos y los $(7 - 4 =) 3$ restantes son “bits paridad”**. Dicho de otro modo, si dividimos en cachos de 4 bits un vídeo/foto/audio/texto; obtendríamos, por ejemplo, un trozo que podría ser “1011”.

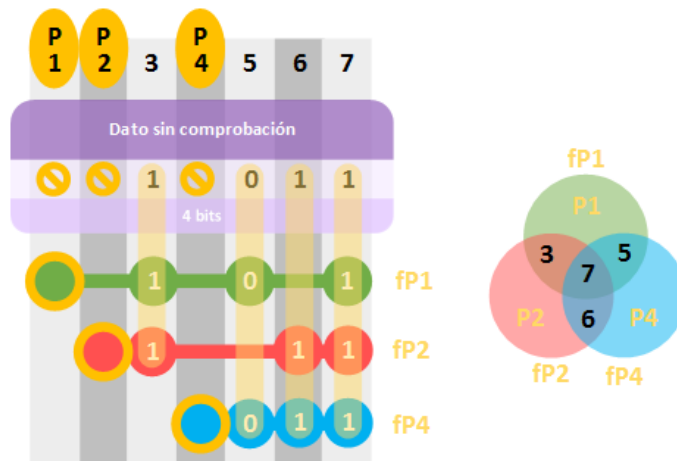


(https://jarroba.com/wp-content/uploads/2016/10/Hamming-Datos-en-crudo-www.jarroba.com_.png) A este cacho de 4 bits se le añaden 3 bits más que servirían para comprobar los datos (los “bits de paridad”). Para ello los ubicamos reservando los huecos para los “bits de paridad” de las posiciones 1, 2 y 4 (como en la “matriz síndrome” que vimos más arriba).

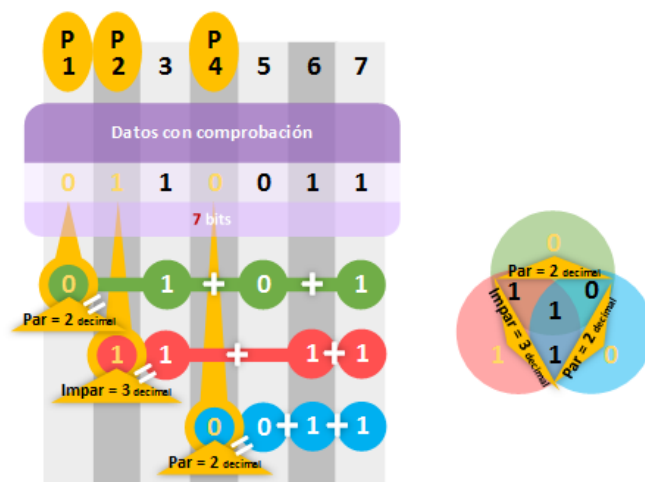


(https://jarroba.com/wp-content/uploads/2016/10/Hamming-Dejando-hueco-para-los-bits-de-paridad-www.jarroba.com_.png) Completamos las filas que comprobarán cada “bit de deducción” en la “matriz síndrome” (hasta la posición 7, no necesitamos más) con las normas antes descritas. Como puedes ver en la imagen siguiente (te recomiendo que ojees a la imagen siguiente a medida que lees este párrafo), tenemos 3 filas: “fP1”, “fP2” y “fP4”. Estas filas se pueden representar igualmente por conjuntos (pues cada fila contiene un conjunto/grupo de bits), y podemos ver que hay bits en algunas filas que comparten la misma posición (no confundas el “bit de paridad” como “P4”, con la fila que comprueba “fP4” en la que se incluye el propio “P4”, así como los datos de las posiciones 5, 6 y 7; así que nunca coincidirán las posiciones de los “bits de paridad”, pero sí de los bits que comprueba y esto veremos más adelante porqué es). De este modo, vemos que son conjuntos que represento con los mismos colores (o mismo “fP”) tanto en fila (en la imagen la “matriz síndrome” al a izquierda) como en círculo (en la imagen los círculos de colores):

- **fP1 o Verde:** con el “bit de paridad” P1, y las posiciones de datos 3, 5 y 7
- **fP2 o Rojo:** con el “bit de paridad” P2, y las posiciones de datos 3, 6 y 7
- **fP4 o Azul:** con el “bit de paridad” P4, y las posiciones de datos 5, 6 y 7



(https://jarroba.com/wp-content/uploads/2016/10/Hamming-Rellenar-matriz-de-deduci%C3%B3n-www.jarroba.com_.png) Solo nos queda calcular los “bits de paridad”. Simplemente sumamos los “1s” de cada fila, miramos si es par o impar (recordatorio: “par” = “0”, e “impar” = “1”), y completamos el “bit de paridad”. En los conjuntos circulares he cambiado las posiciones de la anterior imagen por los bits propiamente.



(https://jarroba.com/wp-content/uploads/2016/10/Hamming-Calcular-bits-de-paridad-www.jarroba.com_.png) Ya tenemos la cadena de bits que incluye los datos y los “bits de comprobación” siguiendo “Hammin(7,4)”.



(https://jarroba.com/wp-content/uploads/2016/10/Hamming-Datos-con-bits-de-paridad-www.jarroba.com_.png)

Nombres de los grupos de bits

No he querido poner nombres nuevos durante la explicación para no complicar la explicación (y continuaré sin hacerlo un buen rato para que la lectura del artículo sea armoniosa). Realmente al dato que se envía se llama “Mensaje” o “Secuencia de Datos”, que se convierte mediante el “Código de Hamming” a un “Bloque de código”.



Grado de Data Science

Aprende a analizar los datos de manera crea...

UOC

Grado de Turismo

El turismo sostenible ya está aquí. Conócelo...

UOC

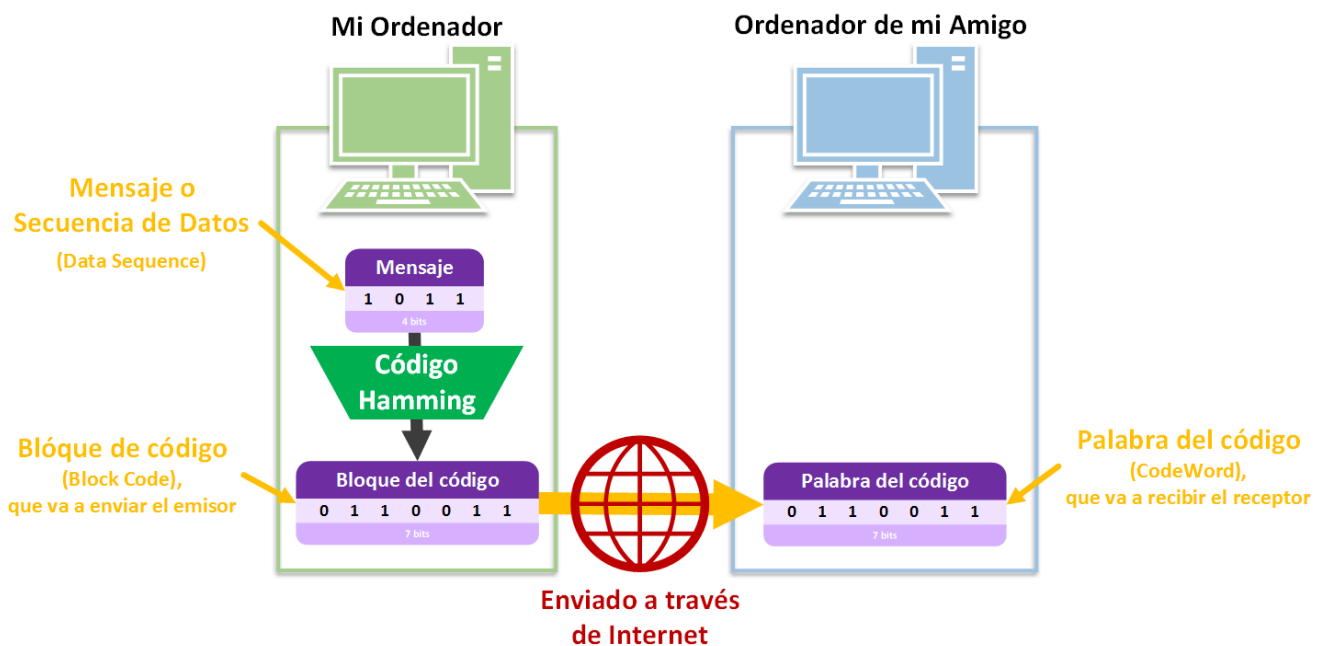
Grado de Artes

Entra en un programa pionero en Europa par...

UOC



(https://jarroba.com/wp-content/uploads/2016/10/Dato-a-Bloque-de-c%C3%B3digo-www.jarroba.com_.png) Si queremos enviar el "Bloque de código" por Internet al ordenador de, por ejemplo, un amigo, ya no se llamará "Bloque de código" sino que se llamará "Palabra del código", pues se preparará (añadiendo otros bits con significado, cuyo significado dependerá del protocolo de la comunicación) para entrar en un canal de comunicación que es Internet.



(https://jarroba.com/wp-content/uploads/2016/10/Envio-de-una-Palabra-de-Bloque-despu%C3%A9s-de-aplicar-Hamming-www.jarroba.com_.png) Resumen de las palabras:

- "Mensaje" o "Secuencia de Datos" (en inglés, "Data Sequence"): normalmente el trozo de lo queremos enviar (por ejemplo, un fichero grande, dividido en pequeños trozos que son los mensajes)
- "Bloque del código" ("Block Code"): Por teoría de la codificación, el resultado de aplicar un codificador (como el código Hamming) al mensaje para que le añada bits de redundancia (como los bits de paridad). (Nota sobre la traducción: No traduzco «Block Code» a «Código del Bloque» y sí a «Bloque del código», pues un «Código corrector de errores» codifica los datos en bloques, por lo que uno de estos bloques pertenece a un «Código» que lo ha creado, y podemos decir que es un «Bloque del código»)
- "Palabra del código" ("Code Word"): Por teoría de la comunicación, lo que queremos enviar por un canal de comunicación (como Internet), que normalmente va a ser un "Bloque de código" para corregir errores quien

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

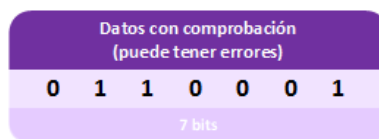
ACEPTAR

reciba la "Palabra del código" (Podría ser también un "Mensaje" la "Palabra del código", aunque si llegan bits erróneos al receptor no habrá forma de corregirlos al no tener bits de redundancia).

Buscar errores y corregirlos

Ahora lo mismo, pero al revés ¡Que dé comienzo el juego de deducción!

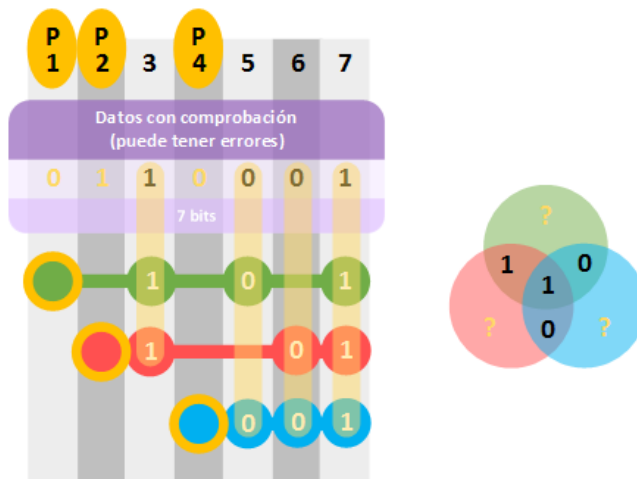
Supongamos que nos llega por Internet (o un dato de un disco duro) la siguiente cadena de bits, con bits de datos y "bits de paridad". Tenemos que comprobar que esté bien el dato antes de utilizarlo; no vaya a ser que por Internet debido a alguna interferencia se haya cambiado algún bit (o si estuviera en nuestro disco duro de plato, pudiera ser que se haya desmagnetizado algún bit; para los discos duros SSD podría ser que alguna celda de memoria estuviera defectuosa y no guardara bien algún bit, por ejemplo). Vamos, que queremos utilizar la siguiente cadena de bits y no sabemos si está correcto para poder utilizarlo ¡Comprobémoslo!



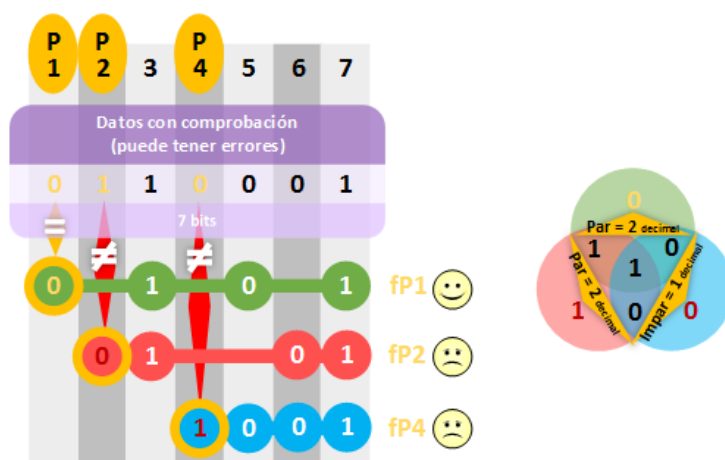
(https://jarroba.com/wp-content/uploads/2016/10/Hamming-Datos-con-bits-de-paridad-que-podr%C3%ADa-contener-errores-www.jarroba.com_.png)

No sabemos si tiene algún error o no, puede que esté correcto, puede que tenga algunos bits cambiados; tampoco sabemos si los que están mal son los bits que corresponden con los datos o los propios "bits de paridad" ¿Entonces, si está mal algún "bit de paridad" ya no podremos corregir o comprobar nada? También podemos saber si está mal algún "bit de paridad" e incluso corregirlo, veamos cómo.

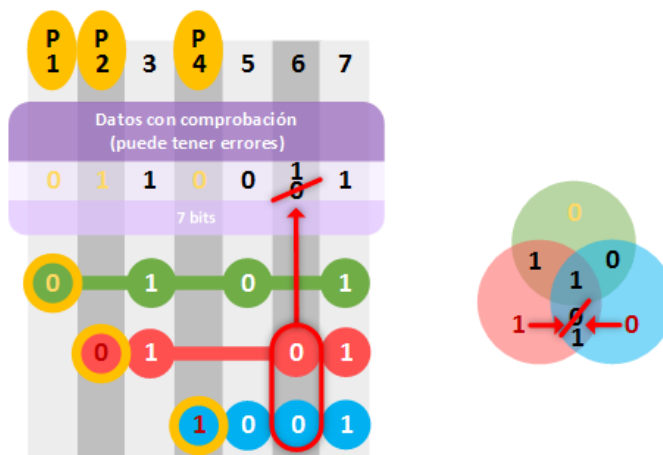
Lo primero que tenemos que hacer es rellenar la "matriz síndrome". No rellenamos los "bits de paridad", pues es lo que utilizamos para realizar las comprobaciones; es lo que tenemos que volver a calcular igual que antes.



(https://jarroba.com/wp-content/uploads/2016/10/Hamming-Rellenar-matriz-de-deduccion-a-la-inversa-www.jarroba.com_.png) Al igual que hicimos anteriormente calculamos los "bits de paridad". Una vez calculados comprobamos con los que ya tenía la cadena entera, miramos si coinciden.



(https://jarroba.com/wp-content/uploads/2016/10/Hamming-Calcular-bits-de-paridad-a-la-inversa-y-compararlos-www.jarroba.com_.png) En el ejemplo podemos comprobar que hay 2 "bits de paridad" que no coinciden en las filas "fP2" (rojo) y "fP4" (azul), con lo que hay algo mal seguro. Pero, ¿qué bit está mal? Con lo que deducimos que el bit erróneo debe pertenecer a ambas filas (en este juego de deducción, con las pruebas hemos logrado reconocer al malo). Por lo que, si no es "0" el bit erróneo, debe ser "1" y lo corregimos.



(https://jarroba.com/wp-content/uploads/2016/10/Hamming-Detectar-errores-y-corregir-los-que-se-puedan-www.jarroba.com_.png)

¿El bit de la posición 7 no está también coincidiendo en las dos filas marcadas como erróneas? Ten en cuenta que el bit "1" de la posición 7 ya nos está diciendo la fila "fP1" (verde) que está bien. Solo el "0" de la posición 6 coincide con las filas "fP2" (rojo) y "fP4" (azul) únicamente; y es el que está mal.

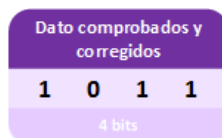
Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)

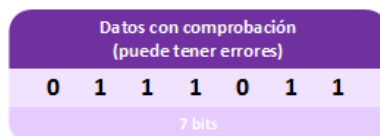
ACEPTAR

Una vez corregido podemos quitar los “bits de paridad” y obtener el dato en perfecto estado, así usarlo. Si era un trozo de una imagen, por ejemplo, después de comprobar el resto de cachos y componer todos los bits de la imagen, podemos ya por fin visualizarla sin ningún problema.

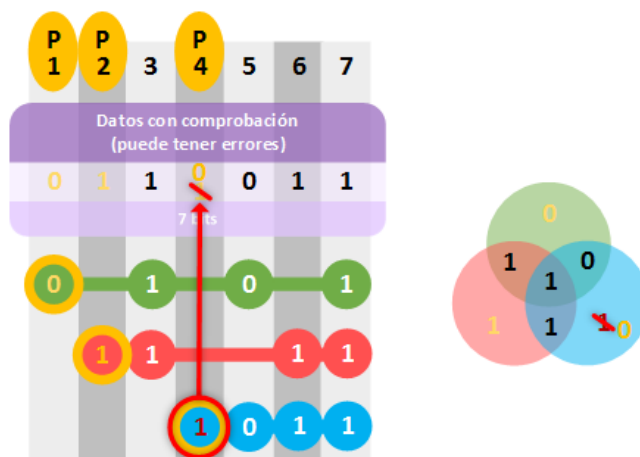


(https://jarroba.com/wp-content/uploads/2016/10/Hamming-Datos-corregidos-y-ya-sin-bits-de-paridad-www.jarroba.com_.png)

¿Y si el que está mal era un “bit de paridad”? Como el siguiente:



(https://jarroba.com/wp-content/uploads/2016/10/Hamming-Datos-con-bits-de-paridad-que-podr%C3%ADa-contener-errores-error-en-bit-de-paridad-www.jarroba.com_.png) Su “matriz síndrome” nos revelará que lo único que está mal es el propio “bit de paridad”. Pues como hemos dicho más arriba, los “bits de paridad” no son coincidentes con bits de otras filas; por lo que si el error lo tiene el “bit de paridad” siempre va a estar en soledad. Sabiendo que está mal podremos corregirlo o no, lo que nos importa es el dato realmente que está correcto.

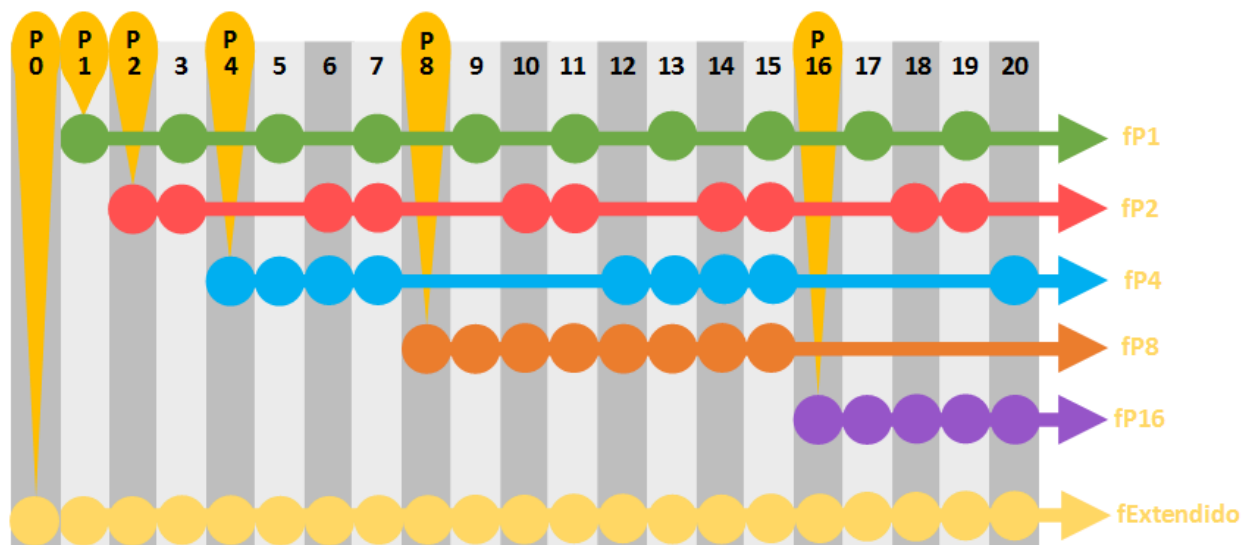


(https://jarroba.com/wp-content/uploads/2016/10/Hamming-Detectar-errores-y-corregir-los-que-se-puedan-error-en-bit-de-paridad-www.jarroba.com_.png)

¿El código de Hamming hasta cuantos bits puede corregir? Evidentemente si todos son erróneos, es imposible corregir nada, pues es otra cosa diferente que se validaría como correcta (Si tenemos una puerta a la que le falta una bisagra la podemos arreglar; lo que no podemos es suponer que una manzana es una puerta en mal estado e intentar “arreglarla” poniéndole una bisagra, pues una manzana es otra cosa, no es una puerta). Para saber cuántos bits se pueden corregir se necesita la “Distancia de Hamming” (la veremos en otro artículo)

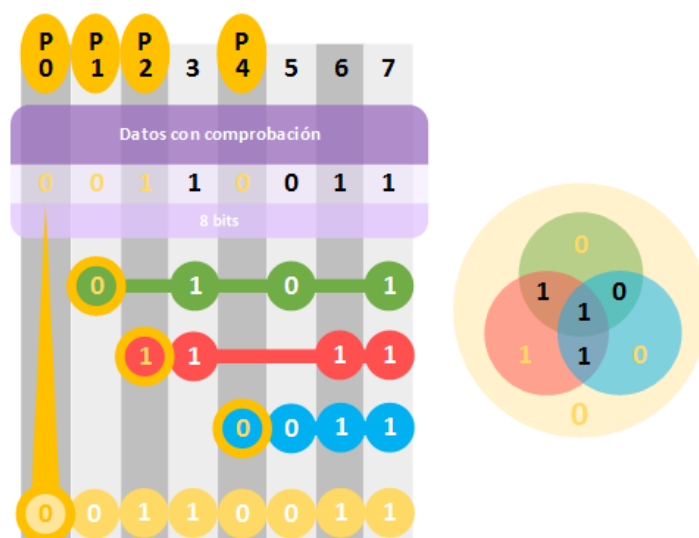
Hamming extendido

También se utiliza el “Hamming Extendido” para asegurar una mejor detección. Simplemente consiste en añadir una fila adicional completa, y un nuevo bit de paridad en la primera posición.



(https://jarroba.com/wp-content/uploads/2016/10/Hamming-Extendido-Matriz-sindrome-www.jarroba.com_.png) **Nota sobre la Matriz síndrome anterior:** La puedes utilizar para practicar código Hamming al imprimirla (en blanco y negro al ser posible) y escribir encima.

Un ejemplo del código Hamming Extendido aplicado con lo antes visto sería:



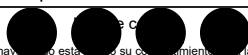
(https://jarroba.com/wp-content/uploads/2016/10/Hamming-Extendido-Ejemplo-www.jarroba.com_.png) Para terminar, fíjate bien que ya no sería "Hamming(7,4)", sino "**Hamming(8,4)**" (4 bits de datos más 4 bits de paridad).

Si sientes curiosidad por saber más, te recomiendo que eches un vistazo al artículo de Hamming (<https://jarroba.com/hamming/>), donde explicamos las distintas capacidades de un código corrector de errores ¡Incluido éste! el «Código de Hamming».

Bibliografía

- Experiencia personal
- https://en.wikipedia.org/wiki/Hamming_code (https://en.wikipedia.org/wiki/Hamming_code)
- https://en.wikipedia.org/wiki/Parity-check_matrix (https://en.wikipedia.org/wiki/Parity-check_matrix)
- https://en.wikipedia.org/wiki/List_decoding (https://en.wikipedia.org/wiki/List_decoding)
- https://en.wikipedia.org/wiki/Block_code (https://en.wikipedia.org/wiki/Block_code)
- https://en.wikipedia.org/wiki/Linear_code (https://en.wikipedia.org/wiki/Linear_code)

Comparte esta entrada en:



/plu /ww /ww /twi
 s.go w.li w.fa tter.
 ogle nke ceb co
 .co din. ook. m/h
 m/s co co om
 har m/s m/s e?
 e? har har stat
 url= eArt e.p us=
 http icle? hp? C%
 s%3 min u=h C3
 A% i=tr ttps %B
 2F% ue& %3 3dig



(<https://creativecommons.org/licenses/by-nc-sa/4.0/>) (<https://www.safecreative.org/userfeed/1401310112503>)

Código de Hamming: Detección y Corrección de errores (<https://jarroba.com/codigo-de-hamming-deteccion-y-correccion-de-errores/>)
 por "www.jarroba.com" esta bajo una licencia Creative Commons
 Reconocimiento-NoCompartir 3.0 Unported License.

Creado a partir de la obra en www.jarroba.com (<https://jarroba.com/codigo-de-hamming-deteccion-y-correccion-de-errores/>)
 om A% rro min

14 thoughts on “Código de Hamming: Detección y Corrección de errores”



Luni (<http://no%20tengo%20xd>) dice:

04/06/2020 a las 02:03 (<https://jarroba.com/codigo-de-hamming-deteccion-y-correccion-de-errores/#comment-282526>)

Si quiero comprobar un código de 11 bits que tiene paridad impar funciona ese tipo de agrupación de acuerdo a sus potencias(?)

Responder



Ramón [Admin Jarroba] dice:

04/06/2020 a las 12:40 (<https://jarroba.com/codigo-de-hamming-deteccion-y-correccion-de-errores/#comment-282544>)

Con paridad impar lo único que tienes que tener en cuenta es que el bit de paridad cuando marca 1 es par (e impar cuando marca 0), al revés. En el ejemplo, es dar la vuelta a los bits de paridad (pues son de paridad par, donde el 1 es el valor impar y 0 el par).

Para 11 bits tendrás un Hamming(11, 7) con 4 bits de paridad.

Responder



Francisca dice:

29/11/2019 a las 03:46 (<https://jarroba.com/codigo-de-hamming-deteccion-y-correccion-de-errores/#comment-272103>)

Buen artículo! Gracias 😊

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)

Responder

ACEPTAR

**Pablo dice:**

erro &t= 2F%

15/11/2019 a las 12:26 (<https://jarroba.com/codigo-de-hamming-deteccion-y-correccion-de-errores/#comment-271189>)

El mejor artículo sobre código Hamming de Internet. Muchas gracias.

Responder

**Aldair dice:**

&titl %B ba.c

e-C 3dig om

%C o+d %2F

05/11/2017 a las 04:45 (<https://jarroba.com/codigo-de-hamming-deteccion-y-correccion-de-errores/#comment-187422>)

excelente! gracias

B3d am go-

igo+ min de-

de+ g%3 lia

Responder

**Juan Manuel Rodriguez dice:**

Ha A+D mm

12/09/2017 a las 02:32 (<https://jarroba.com/codigo-de-hamming-deteccion-y-correccion-de-errores/#comment-183746>)

Amigo mañana tengo que dar una exposición en la facultad y esta explicación me salvó la vida, en todos lados donde busqué, aparecían explicaciones poco convincentes y montones de fórmulas, usted se ha ganado todo mi respeto, desde ya muchísimas gracias.

mm eec ing

ng eir% det

%3 C3 ecci

A+D %B on-

Responder

**Ariel Aichino dice:**

ci% y+C corr

C3 orre ecci

14/05/2017 a las 03:14 (<https://jarroba.com/codigo-de-hamming-deteccion-y-correccion-de-errores/#comment-159814>)

Ramón, me encuentro dictando la asignatura Comunicaciones Digitales (un reemplazo) y no encontré en la bibliografía nada tan sencillo de entender como tus dos notas de Hamming. Muchas gracias por compartir.

%B cci on

3n+ %C de-

y+C 3% erro

Responder

**Damii dice:**

cci +de %2F

%C +err)

05/03/2017 a las 02:40 (<https://jarroba.com/codigo-de-hamming-deteccion-y-correccion-de-errores/#comment-128867>)

excelente explicacion!

3% ore

B3n s)

Responder

**Damii dice:**

+err

ore

05/03/2017 a las 02:40 (<https://jarroba.com/codigo-de-hamming-deteccion-y-correccion-de-errores/#comment-128865>)

Excelente!

s&s

our

Responder

**Christian dice:**

arro

13/02/2017 a las 00:25 (<https://jarroba.com/codigo-de-hamming-deteccion-y-correccion-de-errores/#comment-121139>)Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)[\(http://jarroba.com/faq/\)](http://jarroba.com/faq/)

uso de cookies

om)

ACEPTAR

Estupenda su explicación... He estado todo el día informando al respecto y no entendí nada hasta llegar aquí.... Mil gracias!

Responder



Carlos Córdova dice:

08/02/2017 a las 13:22 (<https://jarroba.com/codigo-de-hamming-deteccion-y-correccion-de-errores/#comment-119542>)

Excelente post!

Muchas gracias por la explicación, un favor, me puedes decir con qué herramientas haces tus animaciones/imágenes del post?

Responder



Ramón [Admin Jarroba] dice:

11/02/2017 a las 18:21 (<https://jarroba.com/codigo-de-hamming-deteccion-y-correccion-de-errores/#comment-120565>)

De nada Carlos 😊

Principalmente utilizo Gimp y Visio de Microsoft para hacer la mayoría de las imágenes.

Responder



Livio Panzera dice:

16/11/2016 a las 12:57 (<https://jarroba.com/codigo-de-hamming-deteccion-y-correccion-de-errores/#comment-78739>)

Gracias por tu explicación tan detallada y didáctica, Ramón, me servirá de mucho para una exposición sobre el tema. Sigue así.

Responder



Ramón [Admin Jarroba] dice:

27/11/2016 a las 20:43 (<https://jarroba.com/codigo-de-hamming-deteccion-y-correccion-de-errores/#comment-81512>)

Me alegro que sirva, muchos ánimos con la exposición 😊

Responder

Deja una respuesta

Tu dirección de correo electrónico no será publicada. Los campos obligatorios están marcados con *

Comentario *

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)

(<http://jarroba.com/faq/>)
Nombre *

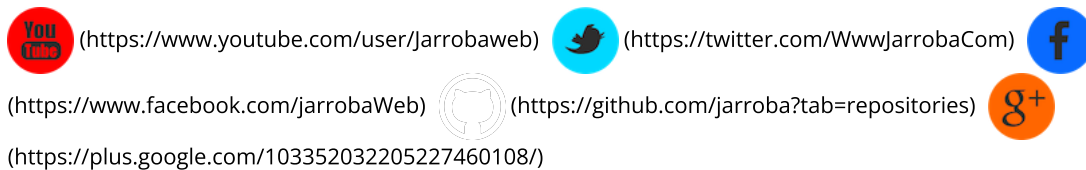
ACEPTAR

Correo electrónico * Web [Publicar el comentario](#)

Este sitio usa Akismet para reducir el spam. Aprende cómo se procesan los datos de tus comentarios (<https://akismet.com/privacy/>).



Siguenos en:

Contacto: jarrobaweb@gmail.com (<mailto:jarrobaweb@gmail.com>)[FAQ \(HTTPS://JARROBA.COM/FAQ/\)](https://jarroba.com/faq/)[VISITAS \(HTTPS://JARROBA.COM/VISITAS/\)](https://jarroba.com/visitas/)[MAPA DEL SITIO \(HTTPS://JARROBA.COM/SITEMAP_INDEX.XML\)](https://jarroba.com/sitemap_index.xml)Copyright © 2024 Jarroba.com (<https://www.jarroba.com>) - Todos los derechos reservados*By Reimon & Richard*[Configuración de la privacidad y las cookies](#)

Gestionado por Google Cumple el TCF de IAB. ID de CMP: 300

Uso de cookies

Este sitio web utiliza cookies para que usted tenga la mejor experiencia de usuario. Si continúa navegando está dando su consentimiento para la aceptación de las mencionadas cookies y la aceptación de nuestra [política de cookies](http://jarroba.com/faq/) (<http://jarroba.com/faq/>)

ACEPTAR