



## Teoría de la Información y la Codificación

### Grado en Ingeniería Informática

#### Ejercicios del Tema 5

1. Enuncie el Teorema de Hamming para corrección de errores en canales con ruido.
2. Suponiendo que queremos diseñar un código capaz de corregir 1 error, ¿qué distancia de Hamming, como mínimo, debe tener dicho código? Con esa misma distancia, si en lugar de diseñar un código para corrección de errores diseñásemos uno para detección, ¿cuántos errores seríamos capaces de asegurar detectar?
3. Si un código para corrección de errores tiene distancia de Hamming  $d=6$ , ¿cuántos errores es capaz de corregir?
4. Diseñe un código que permita tener 4 palabras y que, según el Teorema de Hamming, pueda ser capaz de corregir errores en 3 bits. Explique cómo calcula el tamaño de cada palabra del código y cómo ha diseñado las palabras para que se puedan corregir 3 errores.
5. ¿Qué es un código lineal? ¿Qué significa que un código es  $F_q^n$ ?
6. Exponga un ejemplo de las palabras de un código que sea lineal, distinto a los ejemplos expuestos en clase.
7. Exponga las propiedades de los códigos lineales, explicando cada propiedad con un ejemplo distinto a los expuestos en clase.
8. ¿Qué es la base del espacio vectorial de un código? Exponga un ejemplo con el código diseñado en el ejercicio 6.
9. ¿Qué es la matriz generadora  $G$  de un código lineal? ¿Cuántas filas tiene? ¿cuántas columnas? ¿porqué debe tener ese número de filas y de columnas? Exponga un ejemplo para generar un código lineal que añada redundancia al código uniforme dado por las 4 palabras: {1100, 0101, 0110, 0011}, y que sea capaz de corregir 2 errores.
10. Explique cómo se codificaría una palabra del código uniforme del ejercicio anterior, utilizando la matriz generadora diseñada también en dicho ejercicio. Explique con un ejemplo cómo se codificaría la palabra 0011.
11. ¿Cómo se decodifica una palabra de un código lineal que ha sido recibida sin errores? Exponga un ejemplo de cómo se decodificaría la palabra generada con el código lineal del ejercicio anterior.
12. ¿Qué es una matriz de control de paridad de un código lineal? ¿Cuántas filas tiene? ¿cuántas columnas? ¿porqué? Indique la fórmula general para calcular una matriz de control de paridad de un código lineal.
13. Calcule la matriz de paridad del código lineal diseñado en el ejercicio 9.
14. ¿Qué es un síndrome? ¿para qué se utiliza? ¿cómo se calcula? Introduzca un error de un bit en el código lineal generado en el ejercicio 10, y calcule su síndrome.
15. Sabiendo que el código lineal diseñado en el ejercicio 9 es capaz de corregir 2 errores, calcule la tabla de síndromes del código.
16. Codifique la palabra 0110 del código uniforme utilizado en el ejercicio 9 en el código lineal diseñado en el mismo ejercicio. Seleccione 2 bits al azar del resultado y cambie su valor. Calcule su síndrome e indique en qué posiciones están los errores cometidos, según la tabla de síndromes. Indique cómo se debe corregir un error detectado por el código lineal y corrija el error.



17. Realice las mismas operaciones del ejercicio anterior, para codificar la palabra del código uniforme 0101. En este caso, introduzca primero un error en un bit, y calcule dónde está el error con la tabla de síndromes. Acto seguido, introduzca otro error en otro bit, y calcule dónde están los errores con la tabla de síndromes.
18. ¿Qué es un código de Hamming? ¿En qué se parece y en qué se diferencia de los códigos lineales?
19. Indique el procedimiento general para diseñar un código de Hamming.
20. Suponga el código uniforme dado por las 4 palabras: {1100, 0101, 0110, 0011}. ¿Sería capaz de diseñar un código de Hamming que corrija 2 errores? Justifique su respuesta.
21. Describa la estructura de una palabra de un código de Hamming capaz de añadir redundancia para corrección de errores de 1 bit a un código uniforme de longitud de palabra 7. Explique cómo se calcularían los bits de paridad, y calcule como ejemplo los bits de paridad que codificarían al mensaje (1011011). ¿Cuál sería la palabra del código de Hamming que codifica este mensaje?
22. ¿Cuál es la estructura de una matriz generadora de un código de Hamming? Explique cómo diseñar esta matriz para un código de Hamming (15, 11)
23. Explique cómo decodificar una palabra de un código de Hamming que ha sido recibida correctamente en el destino. Exponga un elemento con la palabra del código de Hamming codificada en el ejercicio 21.
24. ¿Cómo se construye la matriz de control de paridad de un código de Hamming? Exponga un ejemplo calculando la matriz de control de paridad del código de Hamming diseñado en el ejercicio 21.
25. ¿Cómo se calcula el síndrome de una palabra de un código de Hamming? Exponga un ejemplo para el código construido en el ejercicio 21.
26. ¿Cómo se calcula el bit que contiene el error (en caso de haberlo) en un código de Hamming? Explíquelo y exponga un ejemplo con una palabra del código de Hamming diseñado en el ejercicio 21, a la que se le introduce un error en un bit. Corrija dicho error atendiendo al síndrome detectado. ¿Porqué es más eficiente un código de Hamming que corrige errores de 1 bit que un código lineal que corrige el mismo número de errores? Justifique su respuesta con ejemplos.
27. ¿Qué es un código cíclico?
28. Dado un mensaje de 4 bits, que queremos codificar con un código cíclico con polinomio generador  $g(x)=x^3+x+1$ , calcule el mensaje resultante del código. ¿Cuántos bits tendrán las palabras de dicho código cíclico?
29. Decodifique la palabra del código cíclico resultante del ejercicio anterior, indicando el procedimiento para su decodificación.
30. Describa los códigos cíclicos que conoce.

*ugr*

**Universidad de Granada**  
Departamento de Ciencias de la Computación  
e Inteligencia Artificial



## Evaluación

Los ejercicios se realizarán individualmente, no siendo posible en grupos de alumnos. Deberán estar correctamente resueltos, incluyendo explicaciones, definiciones, fórmulas o ejemplos adicionales que se estimen oportunas para que la solución del ejemplo sea autoexplicativa y autocontenida. Deberán utilizar un lenguaje técnico apropiado, adaptado al nivel de la asignatura. Con estas pautas, los ejercicios se evaluarán atendiendo a los siguientes criterios:

- **Calificación NO APTO:** El ejercicio no está resuelto, no se responde a todo lo que se requiere o está mayormente incompleto.
- **Calificación APTO:** El ejercicio está resuelto, pero contiene fallos o no profundiza en la respuesta a la pregunta con el detalle requerido.
- **Calificación DESTACA:** El ejercicio está resuelto con la profundidad requerida y no tiene errores, o contiene mínimos fallos menores (tipográficos, etc.).

Una calificación **NO APTO** supone que el ejercicio se evalúa con 0 puntos. La calificación **APTO** indica que el ejercicio se evalúa con la mitad de su valor. Por último, la calificación **DESTACA** otorga la máxima calificación del ejercicio.

**La detección de copia en algún ejercicio supondrá la calificación de 0 en todos los ejercicios entregados.**

Considerando estos criterios de calificación, la puntuación de cada ejercicio se muestra en la siguiente tabla:

Ejercicio	Puntuación	Ejercicio	Puntuación
1	0,1	16	0,2
2	0,1	17	0,5
3	0,1	18	0,2
4	0,1	19	0,2
5	0,1	20	0,2
6	0,2	21	0,5
7	0,2	22	0,5
8	0,2	23	0,5
9	0,5	24	0,5
10	0,5	25	0,5
11	0,5	26	0,5
12	0,2	27	0,2



ugr

**Universidad de Granada**

Departamento de Ciencias de la Computación  
e Inteligencia Artificial



<b>13</b>	0,5	<b>28</b>	0,5
<b>14</b>	0,5	<b>29</b>	0,5
<b>15</b>	0,5	<b>30</b>	0,2