



UNIVERSIDAD
DE GRANADA

*Este documento está protegido por la Ley de Propiedad Intelectual ([Real Decreto Ley 1/1996 de 12 de abril](#)).
Queda expresamente prohibido su uso o distribución sin autorización del autor.*

Teoría de la Información y la Codificación

4º Grado en Ingeniería Informática

Tarea
Tema 5
Códigos QR

1. Objetivo.....	2
2. Materiales.....	2
3. Descripción de la tarea.....	2
4. Entrega de la práctica.....	3

© Prof. Manuel Pegalajar Cuéllar
Dpto. Ciencias de la Computación e I. A.
Universidad de Granada



DECSAI

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

Códigos QR

1. Objetivo

El objetivo de la tarea consiste en que el estudiante se familiarice con códigos cíclicos, y en particular con los códigos de Reed-Solomon y su aplicación para la construcción de códigos QR. **La tarea debe realizarse de forma individual.**

2. Materiales

Se proporciona al estudiante los siguientes materiales:

- **Video: 6.ImplementacionCodigosQR. Duración: 9' 39".** Contiene una explicación sobre cómo realizar las tareas en el cuaderno Jupyter proporcionado.
- **Cuaderno Jupyter: QRC.ipynb.** Contiene el esqueleto y formato de entrega de la tarea, para que sirva como base al estudiante para la realización del trabajo.

3. Descripción de la tarea

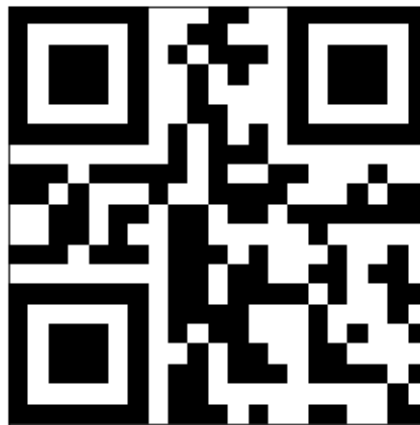
La tarea consiste en desarrollar técnicas para codificar una palabra de un código uniforme procedente de una secuencia de caracteres del charset ISO-8859-1, codificar esta palabra con un código cíclico de Reed-Solomon, e insertar el resultado en una matriz con formato de código QR. En particular, se deberá:

1. Elaborar una función en Python que, teniendo como entrada una cadena de caracteres (bytes) en formato ISO-8859-1, codifique dicha cadena en un array de números binarios.
2. Elaborar una función en Python que, teniendo como entrada un array de números binarios que codifican a una secuencia de caracteres ISO-8859-1, cree una palabra de un código uniforme de 19 valores enteros a codificar por un código QR versión 1, con capacidad de corrección de error L, conteniendo: a) Los bits del campo Mode Indicator del formato QR, con valor a codificación por byte; b) los bits del campo Character Count Indicator del formato QR, conteniendo el número de caracteres del mensaje original a enviar; c) los bits de terminación 0000 y los bytes 11101100 00010001, intercalados, en caso de que sea necesario para conseguir la palabra de 19 valores enteros del código uniforme requerida.
3. Elaborar una función que, teniendo como entrada una representación del mensaje creado con la función anterior, y una representación de un polinomio generador, devuelva una representación del resto de dividir el polinomio asociado al mensaje por el polinomio generador (secuencia FCS).
4. Elaborar una función que, teniendo como entrada la salida de las dos funciones anteriores, genere el código de bloque Reed-Solomon a codificar en un código QR.
5. Elaborar una función que, teniendo como entrada la salida de la función anterior, junto con una matriz de 21x21 componentes, con algunas componentes marcadas con un valor *colorFondo*, dibuje el código de bloque en la matriz, generando un código QR válido.
6. Hacer uso de las funciones implementadas para completar el ejemplo de uso expuesto en el cuaderno Jupyter de la tarea, para generar códigos QR capaces de ser leídos por cualquier lector.

La entrega de la tarea tiene dos partes: Una teórica y otra práctica. Con respecto a la parte teórica, se deberá indicar:

- Definición de códigos de Reed-Solomon.
- Relación entre códigos de Reed-Solomon y códigos QR.
- Pasos para generación de códigos QR.

La parte práctica consiste en implementar en Python diferentes funciones que calculen lo siguiente (cuaderno Jupyter **QRC.ipynb**), descritas anteriormente. Al finalizar la práctica, se deberá dibujar en la última celda del cuaderno un código QR conteniendo información. Un ejemplo de cómo debe visualizarse podría ser el siguiente dibujo:



4. Entrega de la práctica

Se deberá entregar un fichero cuaderno Jupyter (**QRC.ipynb**) con:

1. Varias celdas Markdown que contengan las soluciones a los conceptos teóricos, asociadas a cada función (código Python) a implementar.
2. Las implementaciones requeridas, en celdas de código separadas.
3. Una sección o varias (celda o celdas) donde se realicen las pruebas, y que sea fácilmente modificable por el profesor. **Cada prueba deberá ser analizada y sus resultados justificados por el estudiante. Este análisis crítico será una parte crucial en la evaluación, y su inexistencia puede hacer que la valoración de la parte teórica sea de 0.**

El fichero del cuaderno Jupyter (.ipynb) con la solución deberá estar suficientemente documentado como para que cualquier persona con conocimientos de los conceptos teóricos pueda entenderlo claramente.

La tarea se valorará de 0 a 10, siendo equitativo 50% para la parte de teoría, 50% para la parte práctica de la tarea. **Deberán entregarse ambas partes para que la tarea sea evaluada.** De otra forma, la calificación será 0.

El profesor podrá realizar, en horario de clase o tutorías, **en cualquier momento del curso**, una auditoría de las tareas realizadas por cualquier estudiante, debiendo responder adecuadamente a las cuestiones que se le planteen. **Es responsabilidad del estudiante llevar un estudio diario de todo el material entregado a lo largo del curso**, La auditoría podrá variar la calificación obtenida si el estudiante no es capaz de responder adecuadamente a las consultas planteadas por el profesor.