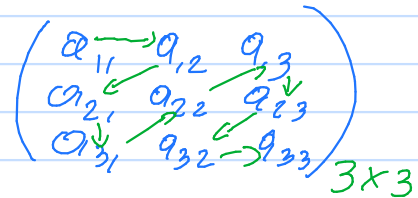
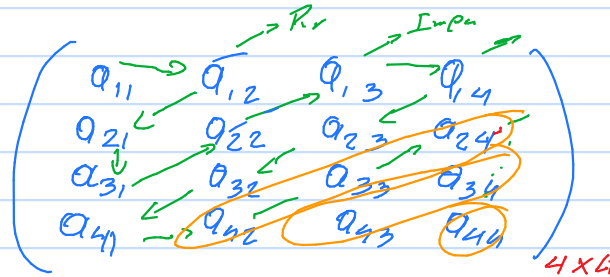


# Procesamiento de Imágenes

Fecha: 31-Julio-2020  
Inco: 1:00pm

## DCT-2D y Compresión de Imágenes

Note que el método Zig-Zag para codificación sigue la siguiente compactamiento:



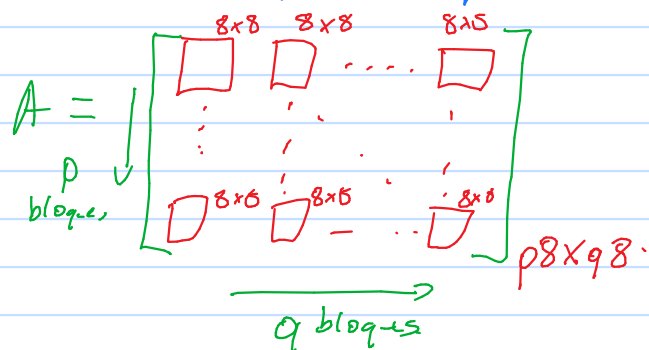
$$= x = [a_{11} \ a_{12} \ a_{21} \ a_{31} \ a_{22} \ a_{13} \ \dots \ ]$$

En Octave, implementamos el caso hasta cuando llega a la diagonal de tamaño  $n$

Queda de Tarea completar el método para recorrer las diagonales restantes de la matriz y guardar todos los valores en el vector  $x$ , hasta llegar a la última diagonal que no tiene todos los valores iguales a 0.

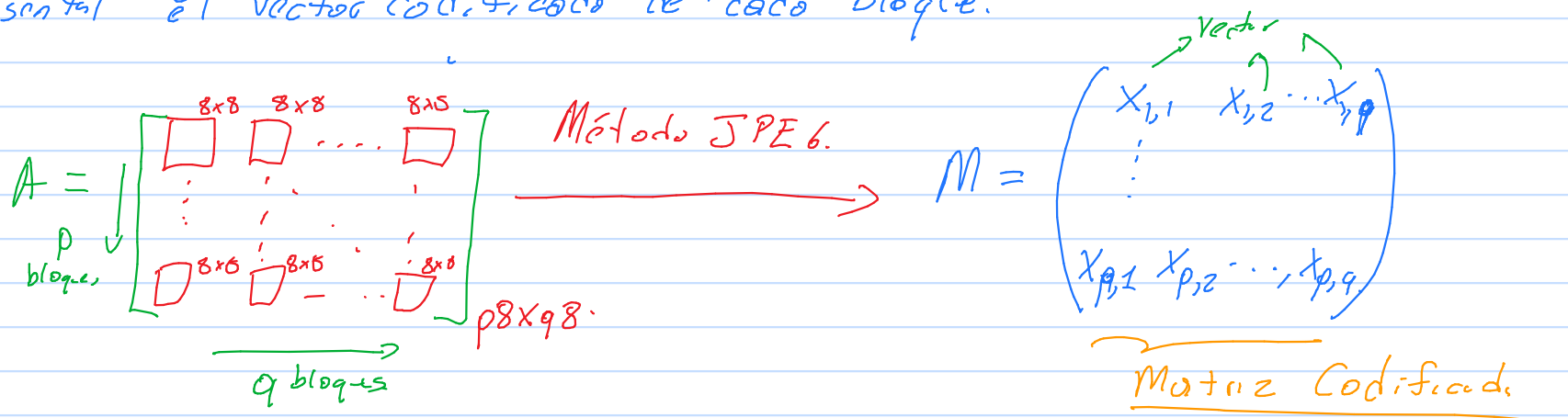
Hemos visto un ejemplo del algoritmo JPEG en una matriz de tamaño  $8 \times 8$ .

Sea  $A$  una matriz que se puede dividir en bloques de  $8 \times 8$ .

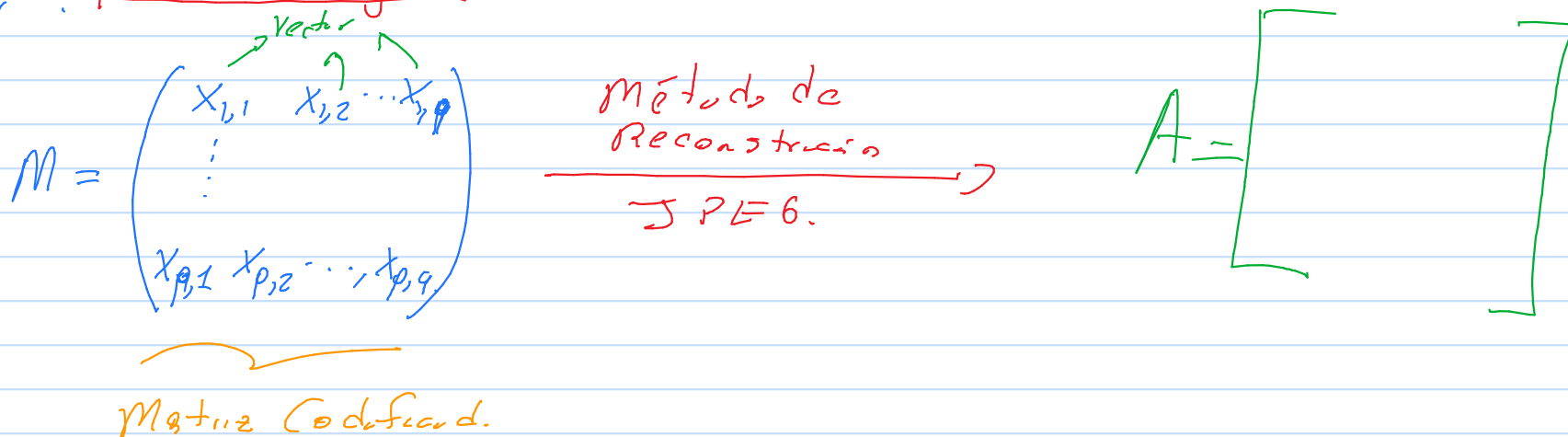


### Comprimir Imagen

Entonces para comprimir una imagen, se divide la imagen en bloques y se aplica el algoritmo visto anteriormente a cada bloque. Por comprimir, necesitamos los vectores de codificación, que se guardan en un arreglo  $M$  que tiene  $p \times q$  entradas, y cada entrada representa el vector codificado de cada bloque.



### Descomprimir la Imagen



Tarea: Implementar el método JPEG para comprimir y reconstruir una imagen de tamaño  $512 \times 512$ .