

# Vectores y matrices

---

Oier Lopez de Lacalle y Patxi Ramírez

Lenguajes y Sistemas Informáticos

---

2021/2022



GASTEIZKO  
INGENIARITZAKO  
UNIBERTSITATE ESKOLA  
ESCUELA UNIVERSITARIA  
DE INGENIERÍA  
DE VITORIA-GASTEIZ

# Contenidos

- 1 **Matrices**
- 2 **Expresiones lógicas**
- 3 **Vector índice**
- 4 **Máscaras**
- 5 **Operaciones**

# Contents

- 1 Matrices**
- 2 Expresiones lógicas
- 3 Vector índice
- 4 Máscaras
- 5 Operaciones

# Matrices: creación (repass)

- Organizando valores en filas.

## Sintaxis

```
varname = [row values; ...]
```

## Ejemplo

```
mat=[5 8 -9; 78 13 -21]
```

- Mediante funciones:

Función	Descripción
<b>zeros</b>	Crea una matriz de ceros
<b>ones</b>	Crea una matriz de unos
<b>eye</b>	Crea una matriz de identidad
<b>rand</b>	Crea una matriz de números reales aleatorios
<b>randi</b>	Crea una matriz de números enteros aleatorios

# Acceder a elementos de la matriz

## Acceso a los elementos

Para acceder los elementos de una matriz, se necesitan dos índices, uno para determinar filas y otro para columnas.

### zenbZer(3)

numMat(1,2)

numMat	1	2	3	4	5
1	5	8	-9	78	13
2	13	18	9	-8	33
3	55	0	-9	-8	-1

numMat([1 2],[2 3])

numMat	1	2	3	4	5
1	5	8	-9	78	13
2	13	18	9	-8	33
3	55	0	-9	-8	-1

Prueba `numMat([1 2],[2 4])` y `numMat([1 2],[2:4])`

## Importante

Tanto los índices de fila como de columna deben ser valores válidos, con respecto al tamaño de la matriz.

# Acceso a filas y columnas de la matriz

## Acceso a filas y columnas

Se puede acceder a filas o columnas completas de una matriz utilizando el operador `:`.

### zenbZer(3)

`numMat(1,:)`

<code>numMat</code>	1	2	3	4	5
1	5	8	-9	78	13
2	13	18	9	-8	33
3	55	0	-9	-8	-1

`numMat(:,[2 3])`

<code>numMat</code>	1	2	3	4	5
1	5	8	-9	78	13
2	13	18	9	-8	33
3	55	0	-9	-8	-1

Prueba `numMat(:,[3:end])`

### Nota

El operador `end` puede usarse para referirse a la última fila o columna de la matriz.

# Asignación de nuevos valores una matriz

## Asignación de nuevos valores

Se pueden establecer nuevos valores para los elementos de la matriz usando el operador de **asignación**.

`zenbZer(3)`

`numMat(1,2)=0`

numMat	1	2	3	4	5
1	5	0	-9	78	13
2	13	18	9	-8	33
3	55	0	-9	-8	-1

`numMat([1 2],[2 3])=[1 2; 3 4]`

numMat	1	2	3	4	5
1	5	1	2	78	13
2	13	3	4	-8	33
3	55	0	-9	-8	-1

## Aviso

La submatriz a actualizar y la que se asignará deben tener las mismas dimensiones. Excepción, cuando se asigna un valor único,

# Contents

- 1 Matrices
- 2 Expresiones lógicas**
- 3 Vector índice
- 4 Máscaras
- 5 Operaciones



# Expresiones lógicas



## Expresiones lógicas

Expresiones que son conceptualmente **verdadero** o **falso**. Estas expresiones pueden usar tanto **operadores relacionales**, que relacionan dos expresiones de tipos compatibles, y operadores lógicos, que operan en **operandos lógicos**.

### Operaciones relacionales:

- $3 < 5$  (“3 es menor que 5”)

### Operadores lógicos:

- $2 < 4 \ || \ 'a' == 'c'$  (“2 es menor que 4 o el carácter 'a' es igual que 'c' ”)

# Operaciones relacionales

- En MATLAB, **true** está representado por el valor lógico 1, mientras que **false** está representado por el valor lógico 0.

## Ejemplos

```
>> 3 < 5
ans = 1
>> 2 > 9
ans = 0
>> 2 < 4 || 'a' == 'c'
ans = 1
>> class(ans)
ans = logical
```

# Operaciones relacionales

Operador	Significado
>	mayor que
<	menor que
>=	mayor o igual que
<=	menor o igual que
==	igual a
~=	no igual a

Problemas con las igualdades:

<https://www.mathworks.com/matlabcentral/answers/57444-faq-why-is-0-3-0-2-0-1-not-equal-to-zero>

## Operadores lógicos

Operado	Significado
&&	and
	or
~	not

### Ejemplos

```
>> 2 < 4 || 'a' == 'c'  
ans = 1  
>> 2 < 4 && 'a' == 'c'  
ans = 0  
>> ~(2 < 4 || 'a' == 'c')  
ans = 0
```

# Contents

- 1 Matrices
- 2 Expresiones lógicas
- 3 Vector índice**
- 4 Máscaras
- 5 Operaciones

# Vector índice



## Vector índice

Un **vector de índices** es un vector cuyos valores son las posiciones de los elementos seleccionados de otro vector. Se utiliza para filtrar valores de otro vector.

```
v=[5 9 4 3 8 8] %vector
iv=[2 5 6] %vector
indice
v(iv) -> [9 8 8] %
resultado
```

	1	2	3	4	5	6
v	5	9	4	3	8	8

## Función `find` para crear un vector de índice

La forma más sencilla de crear un vector de índice es mediante una operación `find`:

```
% vector
v=[5 9 4 3 8 8]
% vector indice
iv = find(v>5)
% resultado
v(iv) -> [9 8 8]
% forma alternativa
v(find(v>5))-> [9 8 8]
```

	1	2	3	4	5	6
v	5	9	4	3	8	8

# find

Más ejemplos:

```
v=[5 9 4 3 8 8] %vector de valores
find(v>5 && v<9)%condicion doble
find(v>5,2)% las 2 primeras posiciones que contienen
valores mayor que 5
find(v>5,2,'last')%las 2 ultimas posiciones que
contienen valores mayor que 5
y = 0:0.1:1
k = find(y==0.3)%problemas de redondeo
k = find(abs(y-0.3)< 0.001)%aprox
```



# Contents

- 1 Matrices
- 2 Expresiones lógicas
- 3 Vector índice
- 4 Máscaras**
- 5 Operaciones

# Máscara



## Máscara

Un **máscara** es un vector lógico compuesto por unos y ceros. Se utiliza para filtrar valores de otro vector. Uno significa **true**, que establece que se debe considerar el valor correspondiente.

```
v=[5 9 4 3 8 8] %vector
m=logical([0 1 0 0 1 1])
% mascara
v(m) -> [9 8 8] %
resultado
```

	1	2	3	4	5	6
v	5	9	4	3	8	8

# Máscara como operación lógica

La forma más sencilla de crear una máscara es realizar una operación lógica:

```
v=[5 9 4 3 8 8] %vector  
m=v>5 %mascara  
v(m) -> [9 8 8] %resultado  
v(v>5) -> [9 8 8] %forma alternativa
```

## Máscaras de filtrado

Selecciona los días con una precipitación mayor que 5:

```
dias=[5 6 7 8 9 10] %dias
precips=[5 9 4 3 8 8] %precipitaciones
m=precips>5 %mascara
dias(m)-> [6 9 10] %resultado
dias(precips>5)-> [6 9 10] %alternativa
```

# Contents

- 1 Matrices
- 2 Expresiones lógicas
- 3 Vector índice
- 4 Máscaras
- 5 Operaciones**

# Operaciones

## Más operaciones:

```
v=[5 9 4 3 8 8] %valores
v(v==4)=[] %borrar los elementos con el valor 4
v(v<=5)=[] %borrar los elementos con valores menor o
igual que 5
any(v>5)% devuelve true (1)si existe algun element
mayor que 5 en v
any(v==8)% devuelve true (1)si existe algun elemento
igual a 8 en v
all(v>5)% devuelve true (1)si todos los elementos son
mayor que 5
all(v>2)% devuelve true (1)si todos los elementos son
mayor que 2
```

# Vectores y matrices

---

Oier Lopez de Lacalle y Patxi Ramírez

Lenguajes y Sistemas Informáticos

---

2021/2022



GASTEIZKO  
INGENIARITZAKO  
UNIBERTSITATE ESKOLA  
ESCUELA UNIVERSITARIA  
DE INGENIERÍA  
DE VITORIA-GASTEIZ