

MATLAB® & SIMULINK®

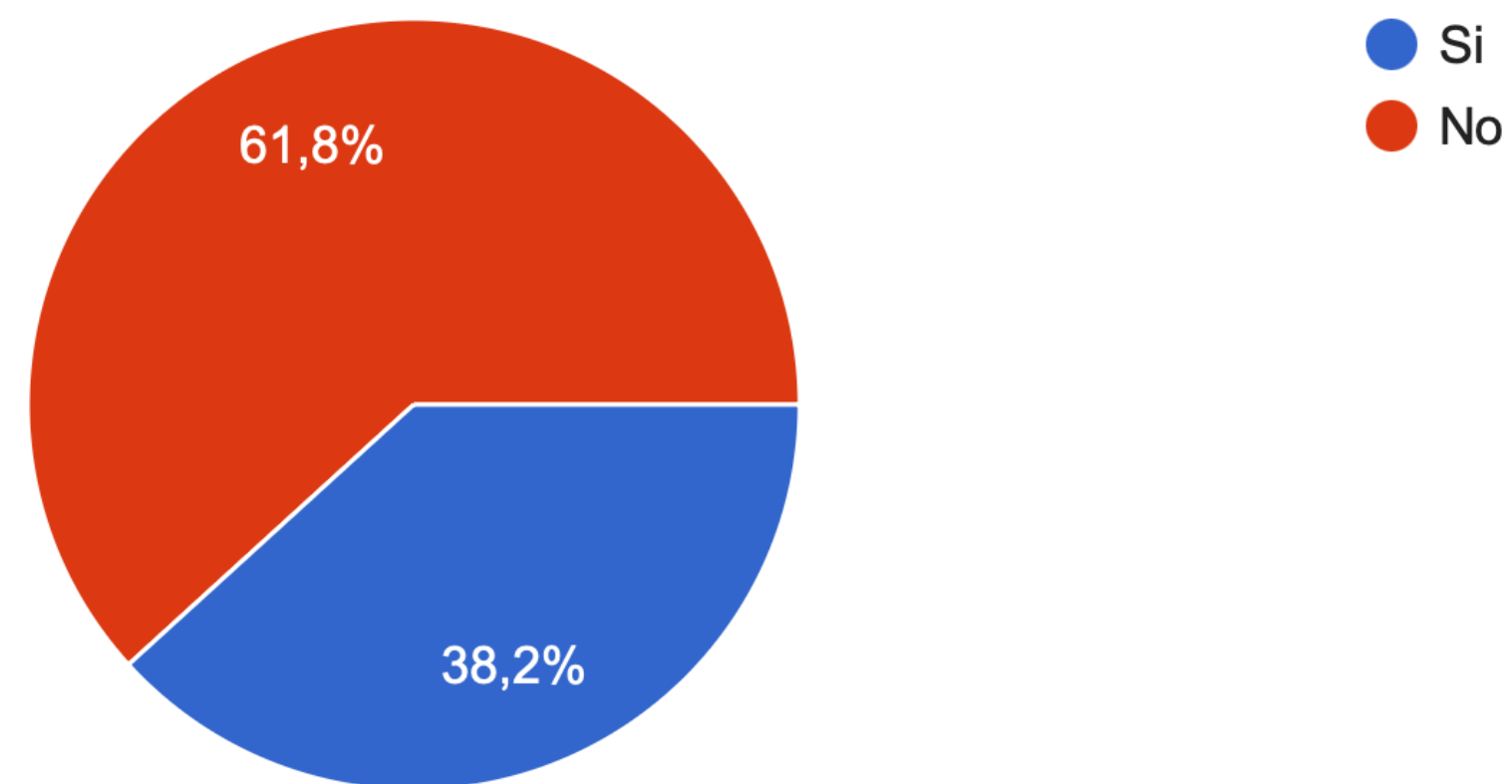
- Curso de introducción a la programación -

Presentación por: **David López García**

¿A quién va dirigido este curso?

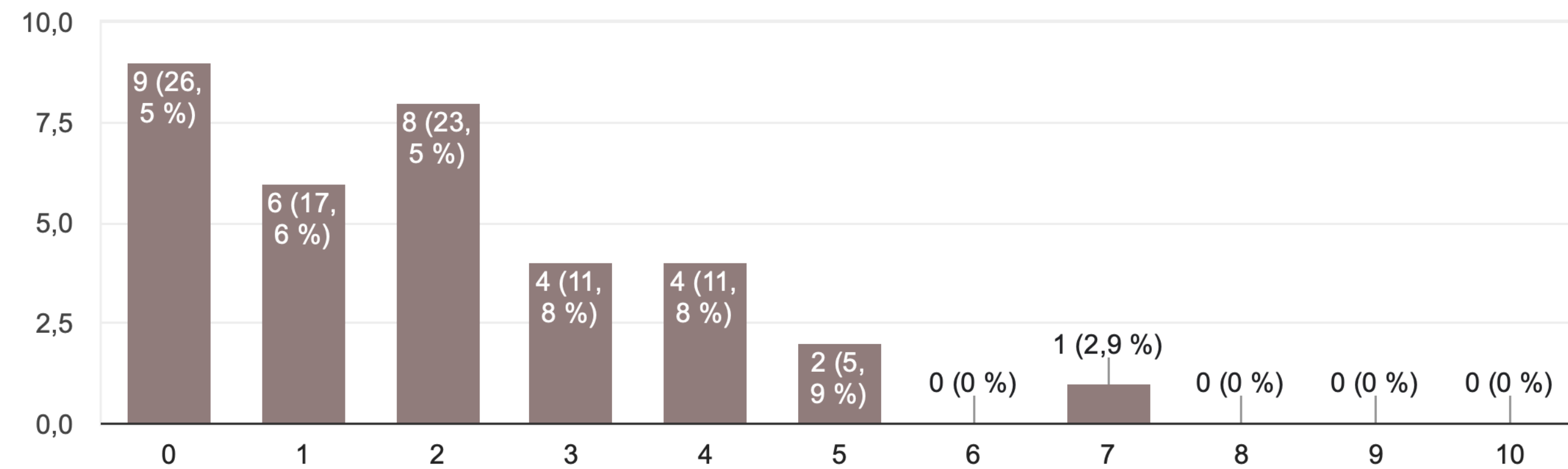
¿Tienes experiencia previa en programación?

34 respuestas

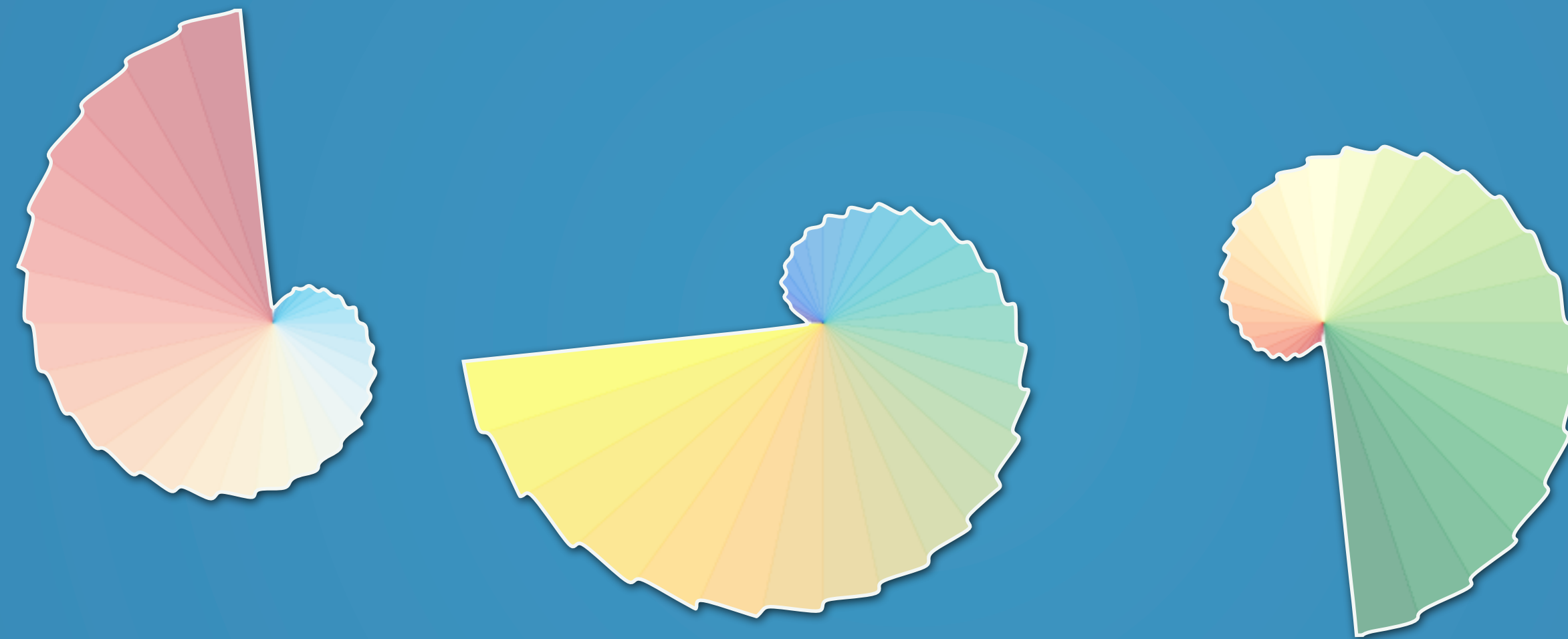


Del 0 al 10, ¿cuál dirías que es tu nivel de experiencia en programación ?

34 respuestas

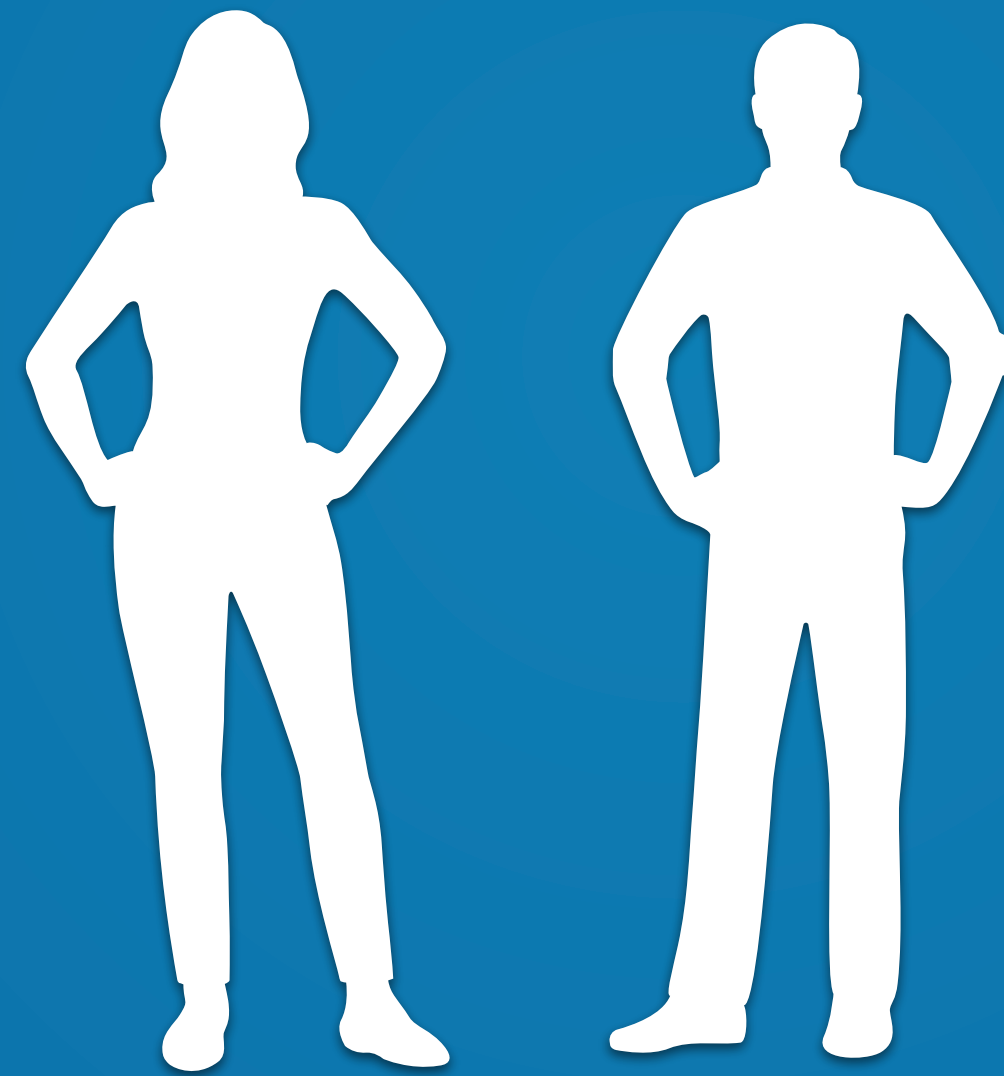


Organización de las sesiones y certificado



Introducción





¿Qué es programar?



Enseñarle al ordenador a hacer algo.

¿Qué es programar?



Enseñarle al ordenador a hacer algo.

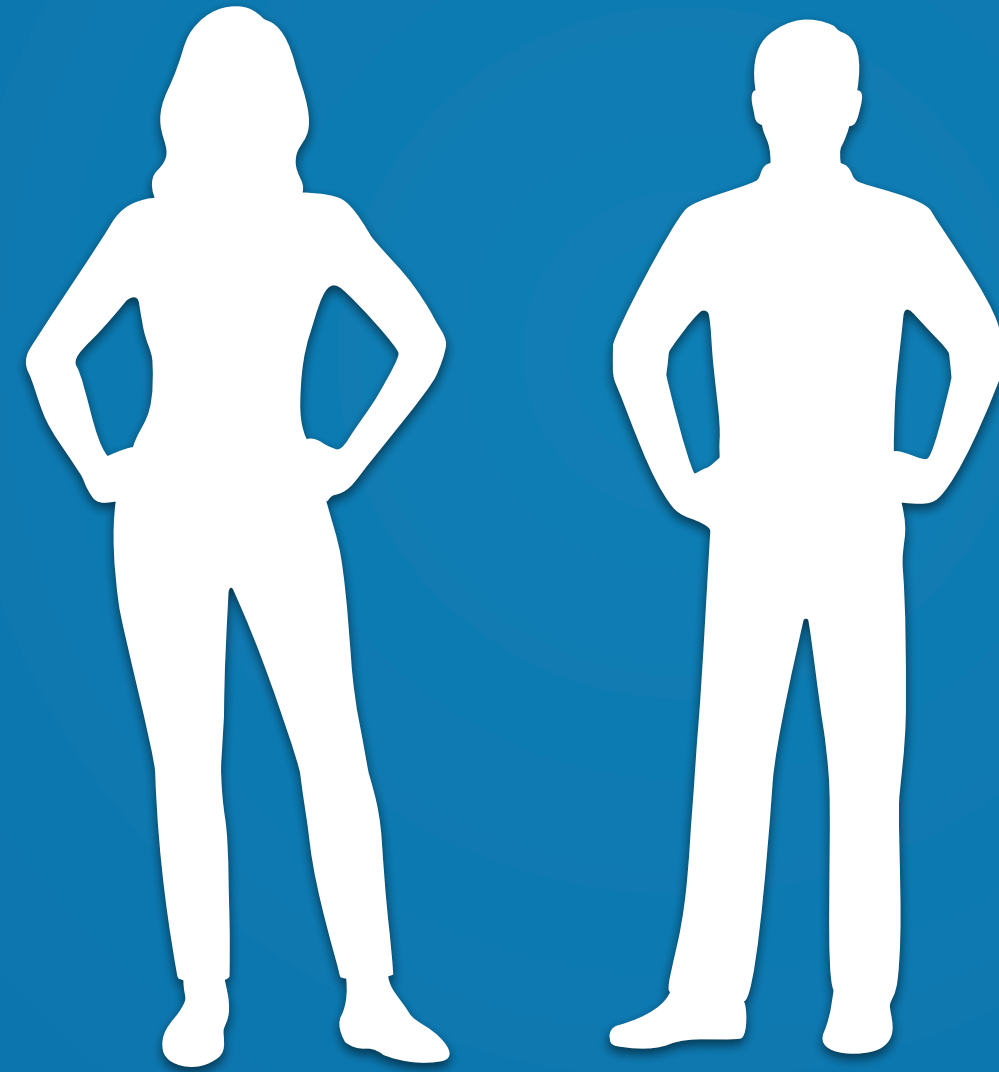
Darle instrucciones al ordenador para que haga algo por nosotros.

¿Qué es programar?

Crear un software usando un lenguaje de programación.

Enseñarle al ordenador a hacer algo.

Darle instrucciones al ordenador para que haga algo por nosotros.



¿Qué es programar?

Crear un software usando un lenguaje de programación.

Enseñarle al ordenador a hacer algo.

Entonces, ¿qué es un lenguaje de programación?

Darle instrucciones al ordenador para que haga algo por nosotros.



¿Qué es programar?





¿Qué es un lenguaje de programación?



LENGUAJE MUCHO MÁS SIMPLE QUE EL NUESTRO



¿Qué es un lenguaje de programación?



LENGUAJE MUCHO MÁS SIMPLE QUE EL NUESTRO



SINTAXIS, VOCABULARIO Y REGLAS MUCHO MÁS REDUCIDO



¿Qué es un lenguaje de programación?



LENGUAJE MUCHO MÁS SIMPLE QUE EL NUESTRO



SINTAXIS, VOCABULARIO Y REGLAS MUCHO MÁS REDUCIDO



REGLAS MUY ESTRUCTURADAS

¿Qué es un lenguaje de programación?



Software

Java



Software

Python



Software



Software

C++

R



Software

Javascript



Software





Software

Swift

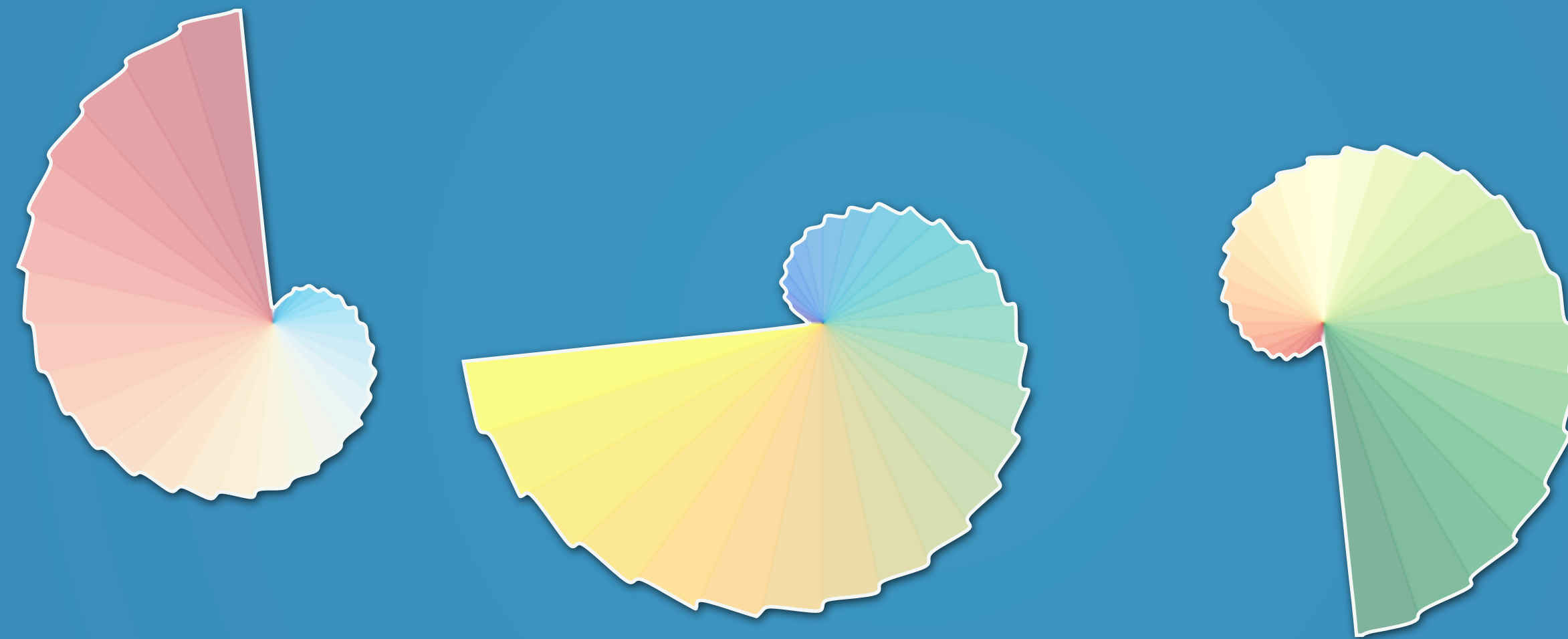
Matlab



Software

RESUMEN

Un programa es un conjunto de sentencias escritas en un lenguaje de programación que le dicen al ordenador qué tareas debe realizar y en qué orden, a través de una serie de instrucciones que detallan completamente ese proceso sin ambigüedad.



Variables





¿Qué es una variable?



¿Qué es una variable?



¿Qué es una variable?



¿Qué es una variable?



¿Qué es una variable?



¿Qué es una variable?



¿Qué es una variable?



¿Qué es una variable?



¿Qué es una variable?





















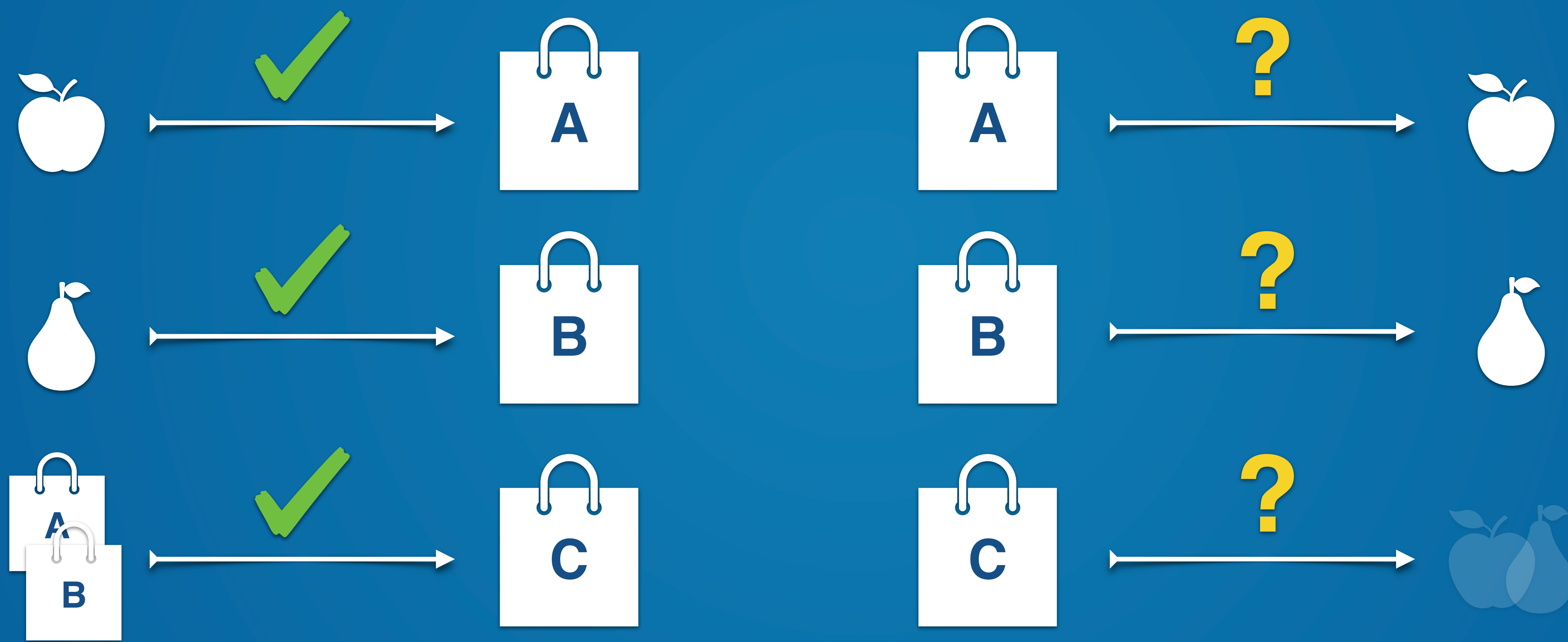


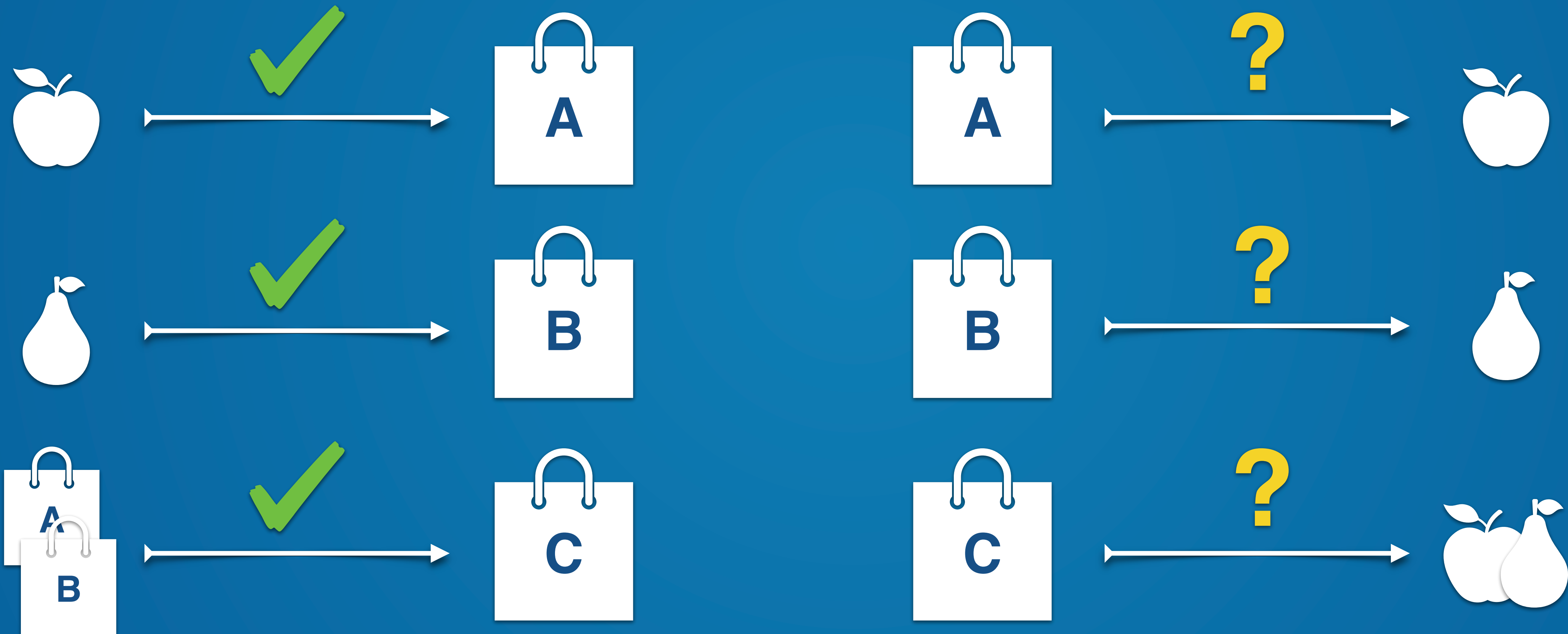








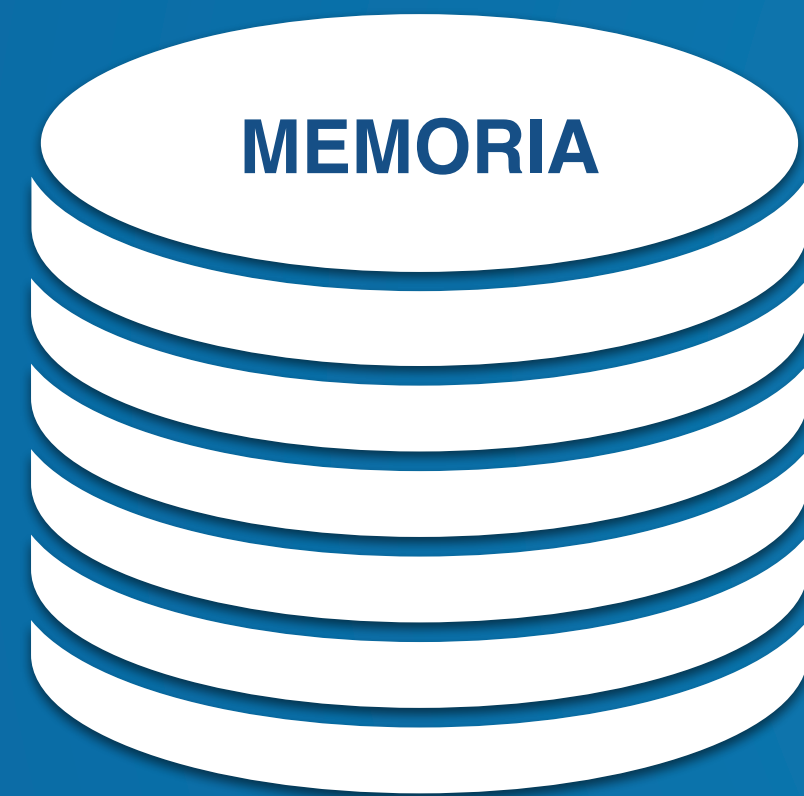




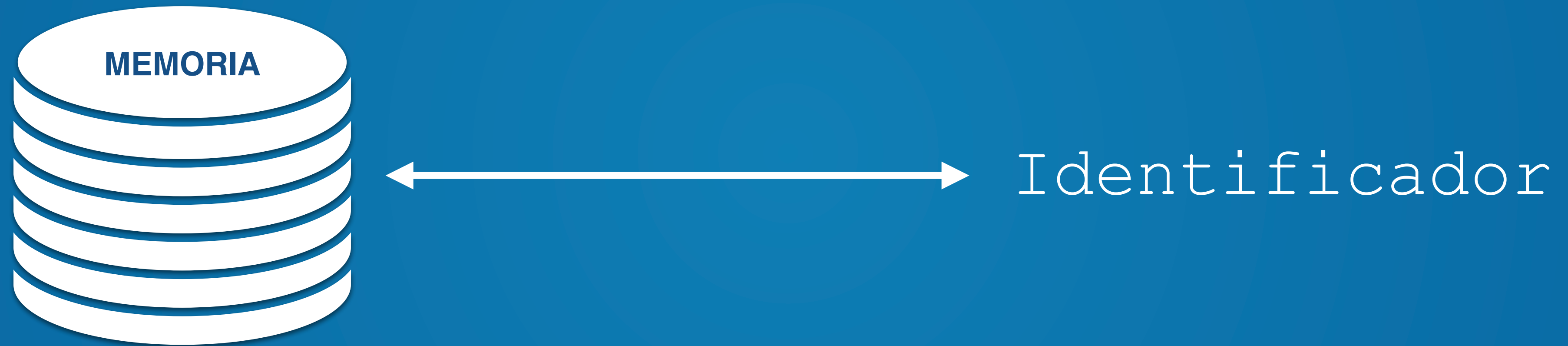




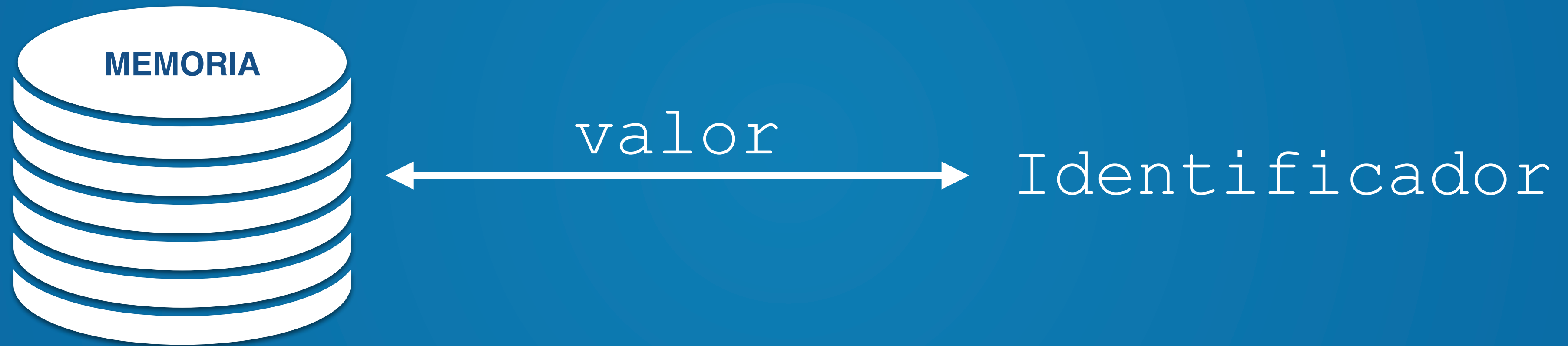
¿Entonces, qué es una variable?



¿Entonces, qué es una variable?



¿Entonces, qué es una variable?



¿Entonces, qué es una variable?


```
nBlocks = 4;           %Número total de bloques.
```

```
nBlocks = 4;           %Número total de bloques.  
nTrials = 9;           %Número total de ensayos.
```

```
nBlocks = 4;           %Número total de bloques.  
nTrials = 9;           %Número total de ensayos.  
stimDur = 200;        %Duración del estímulo.
```



```
acc = nHits / nTrials ;
```




```
acc = nHits / nTrials ;
```



```
acc = nHits / nTrials ;
```



```
acc = 175 / 200 ;
```

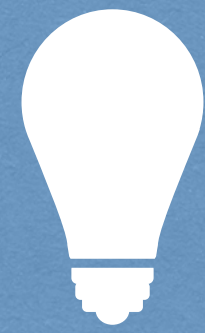


`acc = nHits / nTrials ;`



`acc = 175 / 200 ;`

`acc = 0.875 ;` 



Consejo

Usar nombres descriptivos
para las variables



```
nBlocks = 4;
```



```
nTrials = 9;
```



```
stimDur = 200;
```



```
a = 4;
```

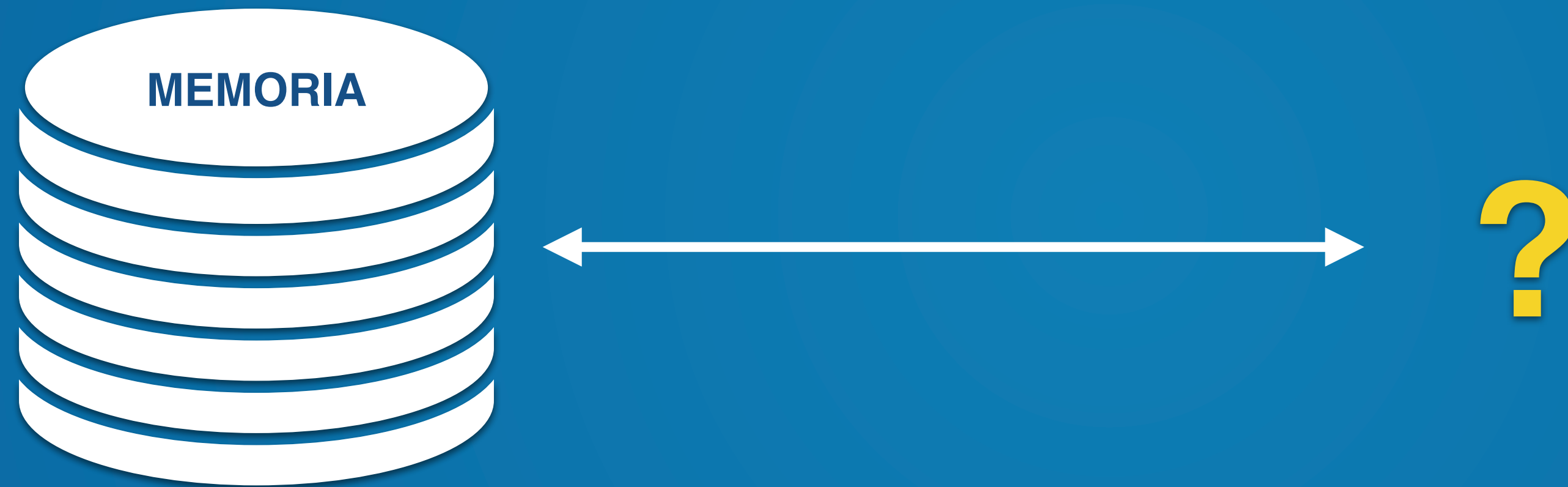


```
n = 9;
```

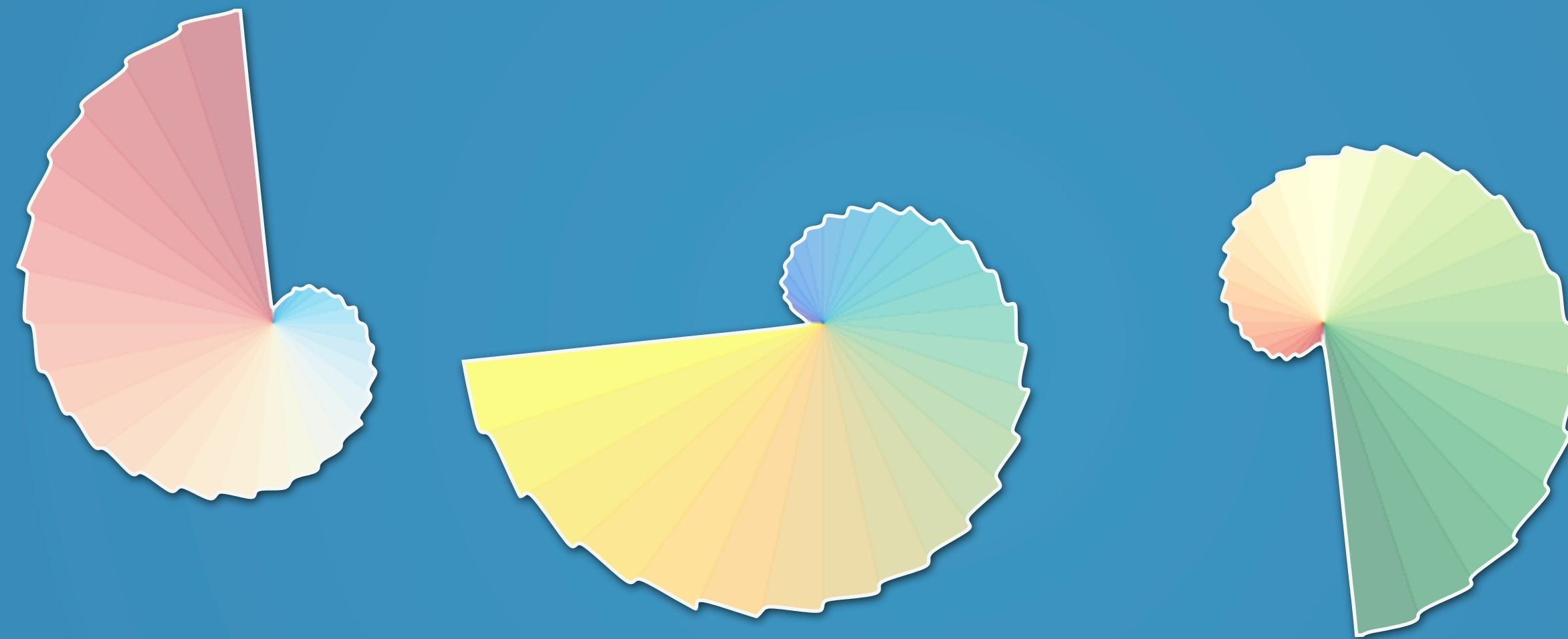





¿Qué podemos almacenar en una variable?



¿Qué podemos almacenar en una variable?



Tipos de datos



NÚMEROS ENTEROS



NÚMEROS ENTEROS



`intNum = 3;`



NÚMEROS ENTEROS



`intNum = 3;`



NÚMEROS REALES



NÚMEROS ENTEROS



`intNum = 3;`



NÚMEROS REALES



`realNum = 27.11;`



NÚMEROS ENTEROS



`intNum = 3;`



NÚMEROS REALES



`realNum = 27.11;`

NÚMEROS COMP.



NÚMEROS ENTEROS



`intNum = 3;`



NÚMEROS REALES



`realNum = 27.11;`

NÚMEROS COMP.



`imagNum = 1 + 2i;`

NÚMEROS ENTEROS



`intNum = 3;`



NÚMEROS REALES



`realNum = 27.11;`

NÚMEROS COMP.



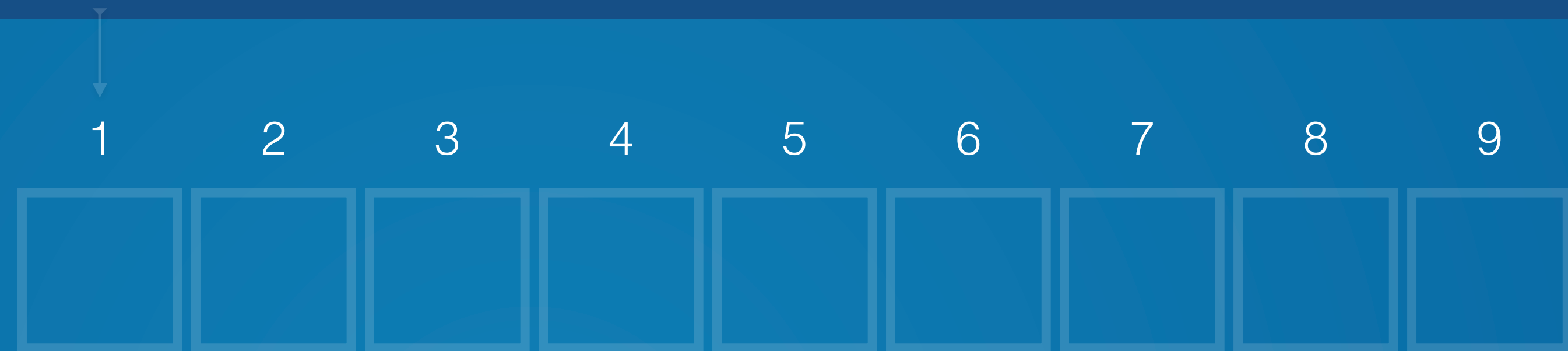
`imagNum = 1 + 2i;`



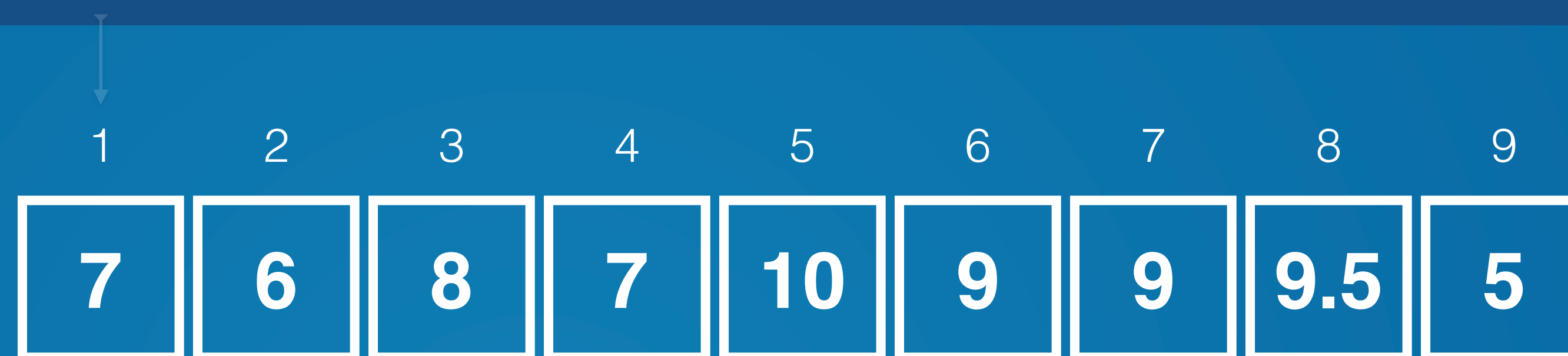
VECTOR
(Array)

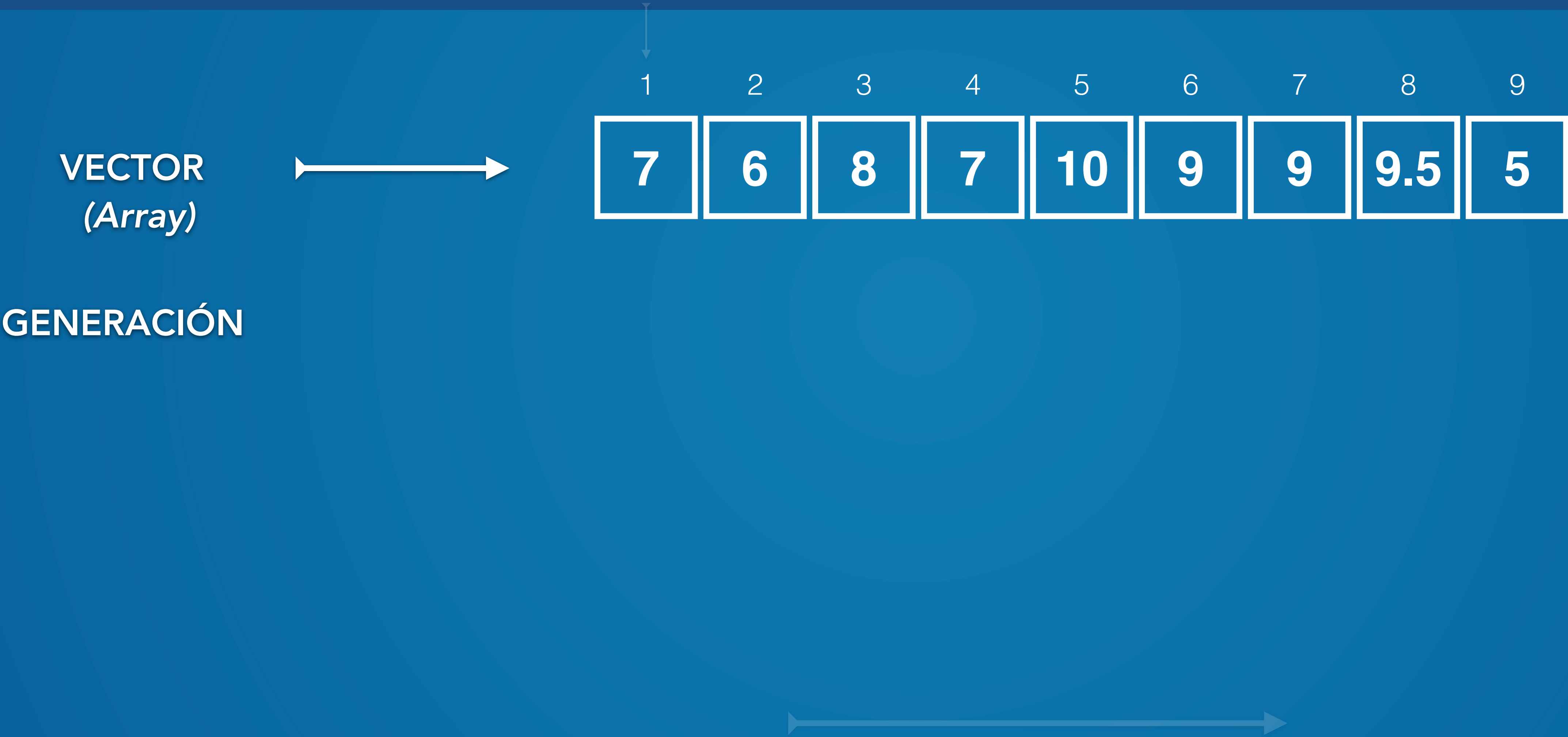


VECTOR
(Array)

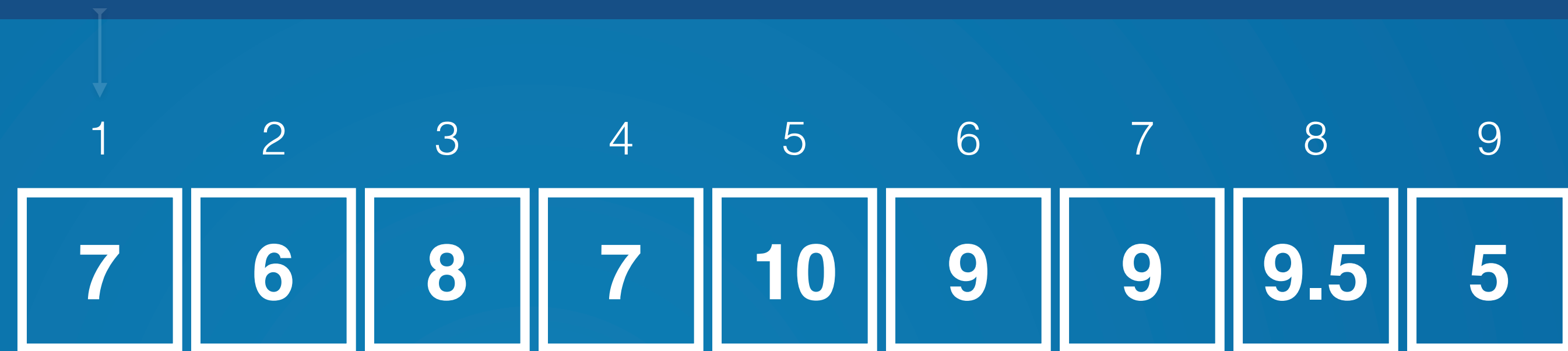


VECTOR
(Array)





VECTOR
(Array)

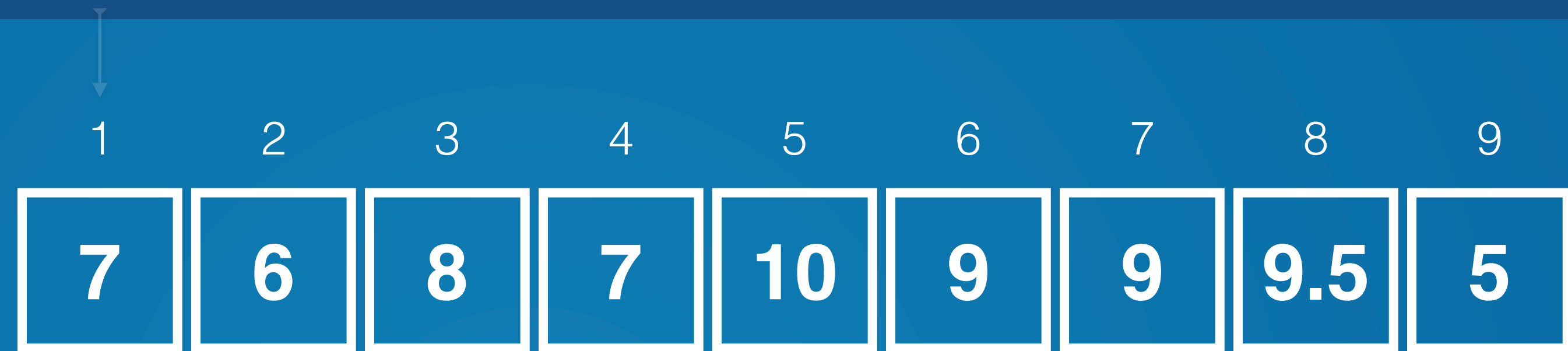


GENERACIÓN

```
marks = [7, 6, 8, 7, 10, 9, 9, 9.5, 5];
```



VECTOR
(Array)



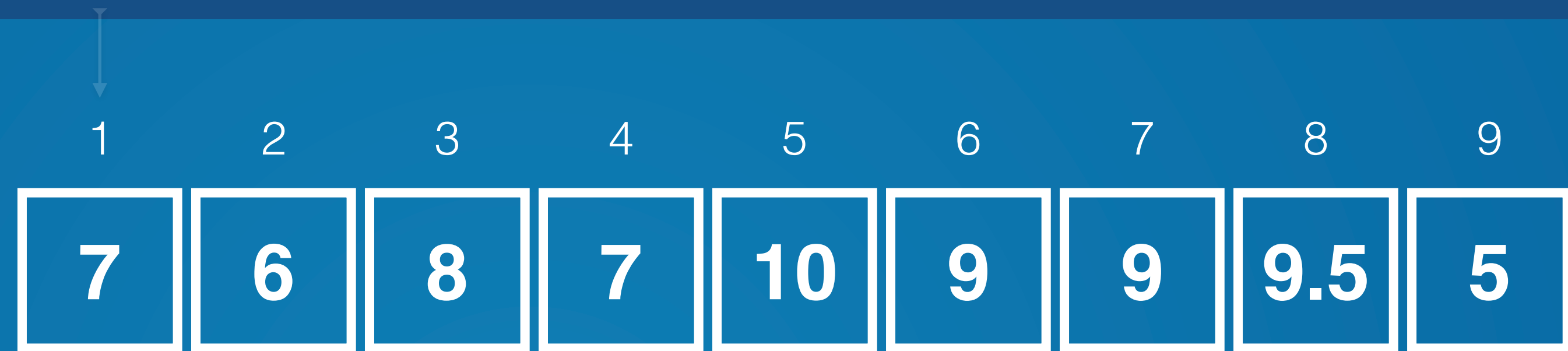
GENERACIÓN

```
marks = [7, 6, 8, 7, 10, 9, 9, 9.5, 5];
```

```
vec = [1:30]; vec = [1:2:30];
```



VECTOR
(Array)



GENERACIÓN

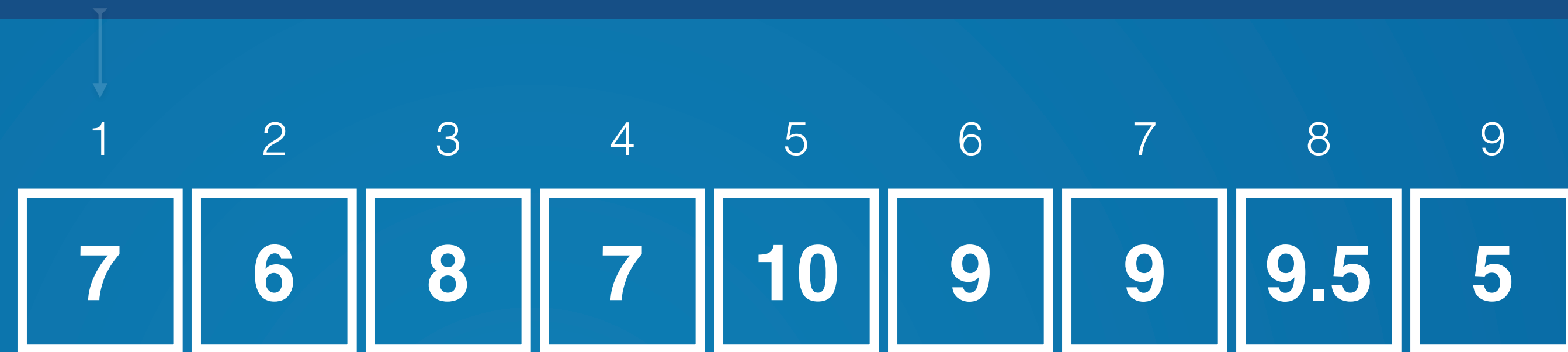
```
marks = [7, 6, 8, 7, 10, 9, 9, 9.5, 5];
```

```
vec = [1:30]; vec = [1:2:30];
```

MODIFICACIÓN



VECTOR
(Array)



GENERACIÓN

```
marks = [7, 6, 8, 7, 10, 9, 9, 9.5, 5];
```

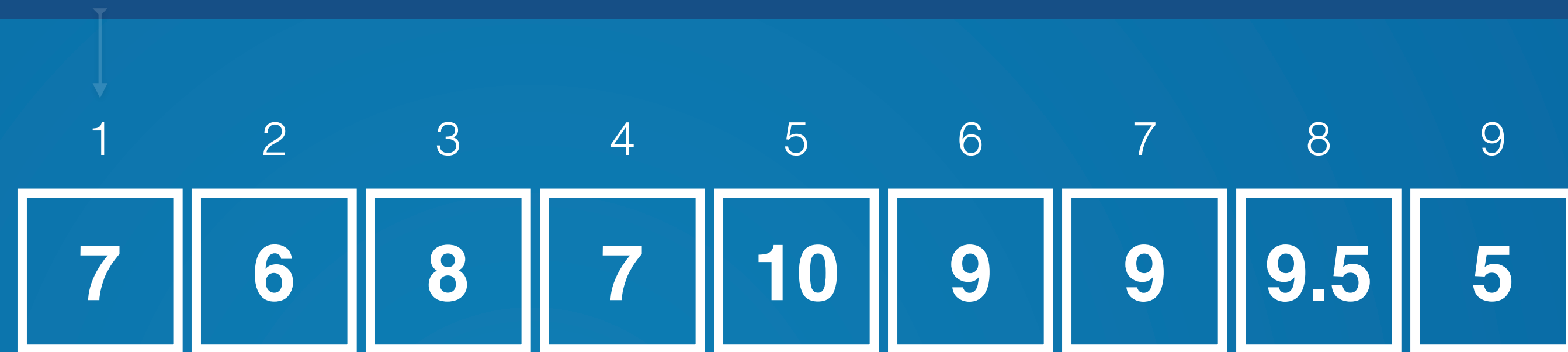
```
vec = [1:30]; vec = [1:2:30];
```

MODIFICACIÓN

```
marks(3) = 5;
```



VECTOR
(Array)



GENERACIÓN

```
marks = [7, 6, 8, 7, 10, 9, 9, 9.5, 5];
```

```
vec = [1:30]; vec = [1:2:30];
```

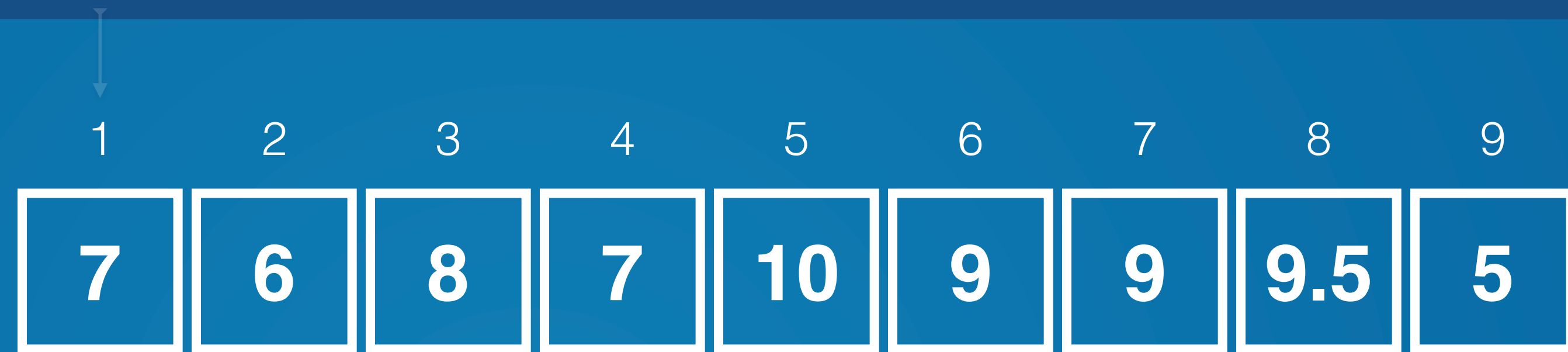
MODIFICACIÓN

```
marks(3) = 5;
```

RECUPERACIÓN



VECTOR
(Array)



GENERACIÓN

```
marks = [7, 6, 8, 7, 10, 9, 9, 9.5, 5];
```

```
vec = [1:30]; vec = [1:2:30];
```

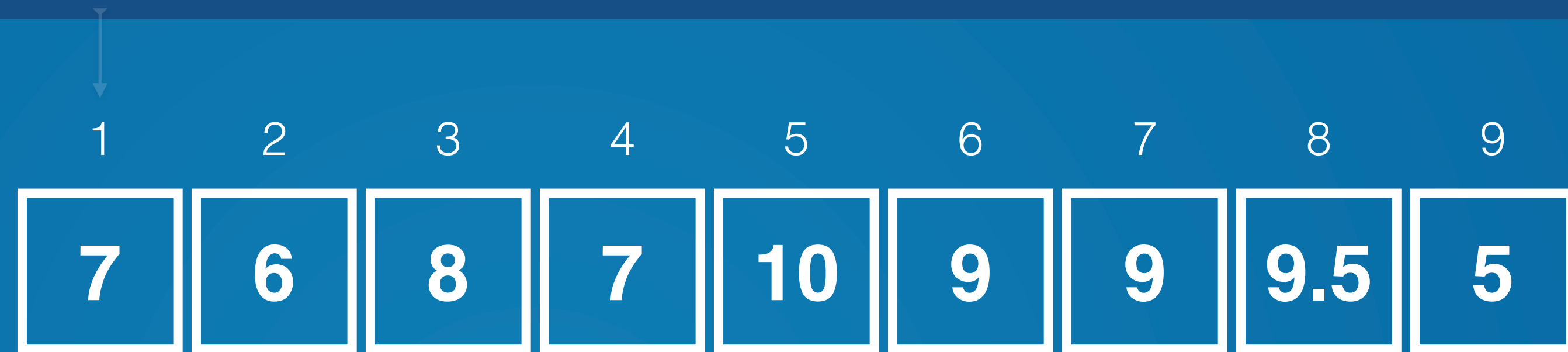
MODIFICACIÓN

```
marks(3) = 5;
```

RECUPERACIÓN

```
marks(4) →
```

VECTOR
(Array)



GENERACIÓN

```
marks = [7, 6, 8, 7, 10, 9, 9, 9.5, 5];
```

```
vec = [1:30]; vec = [1:2:30];
```

MODIFICACIÓN

```
marks(3) = 5;
```

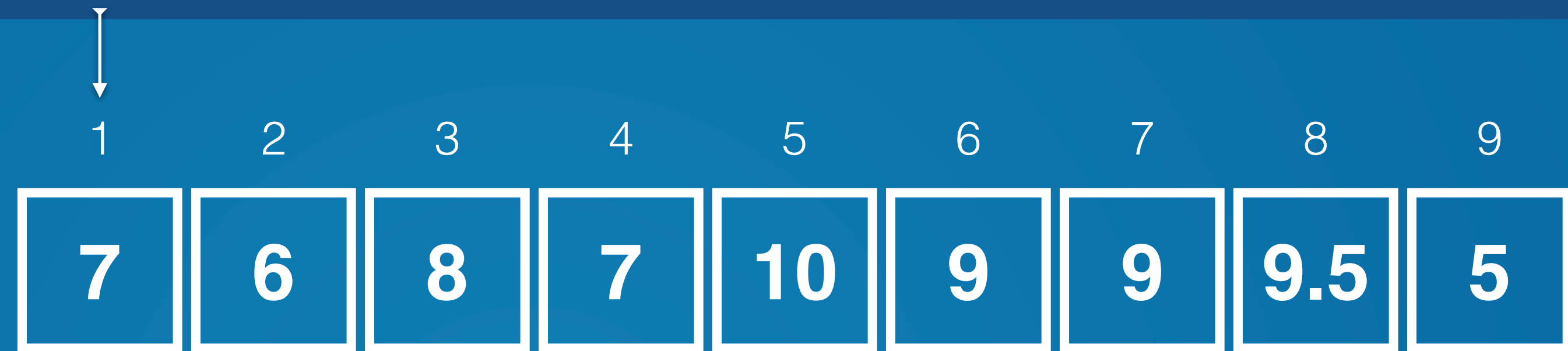
RECUPERACIÓN

```
marks(4) → ans = 7
```




El primer índice es el 1 y no el 0

VECTOR
(Array)



GENERACIÓN

```
marks = [7, 6, 8, 7, 10, 9, 9, 9.5, 5];
```

```
vec = [1:30]; vec = [1:2:30];
```

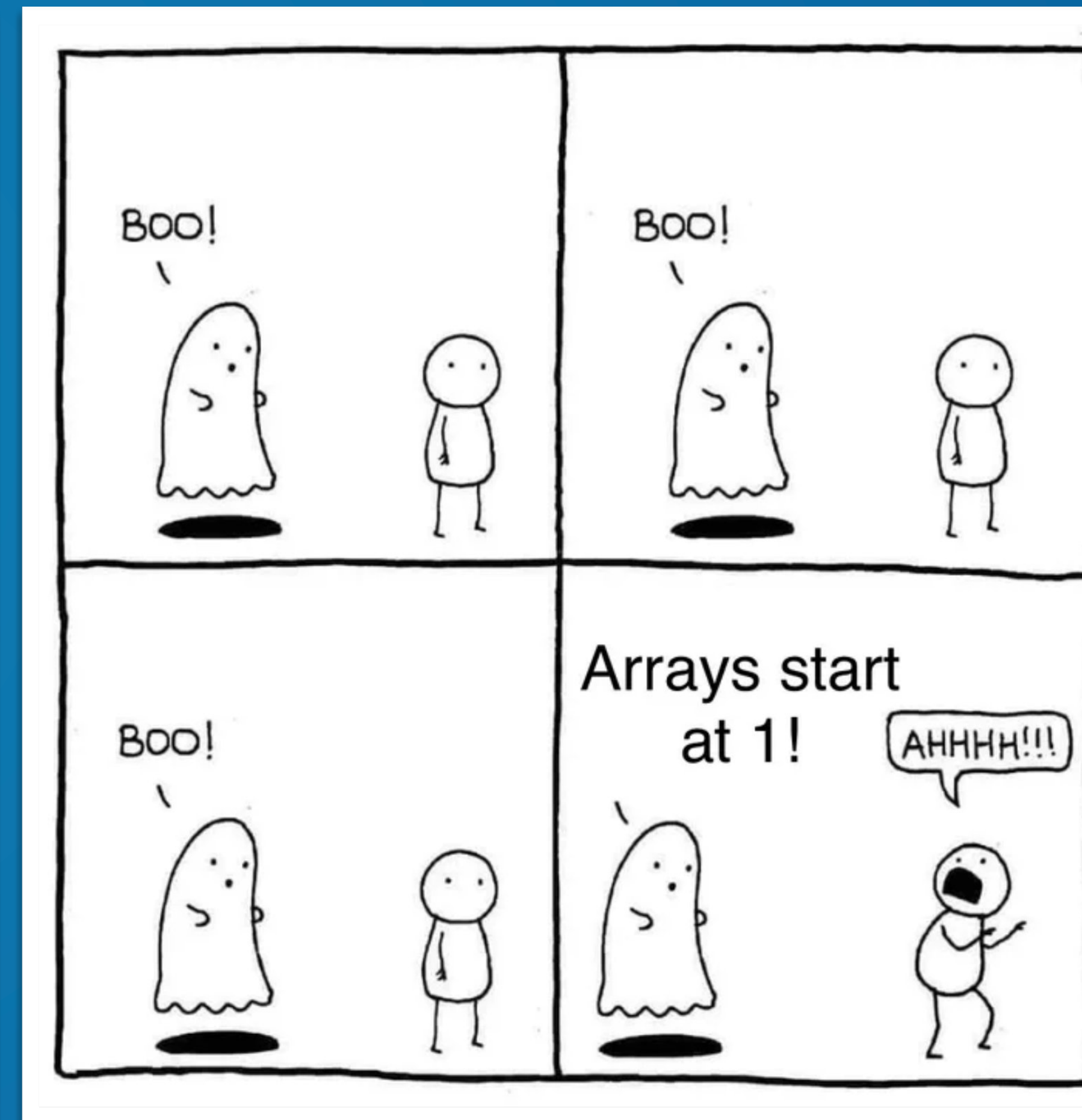
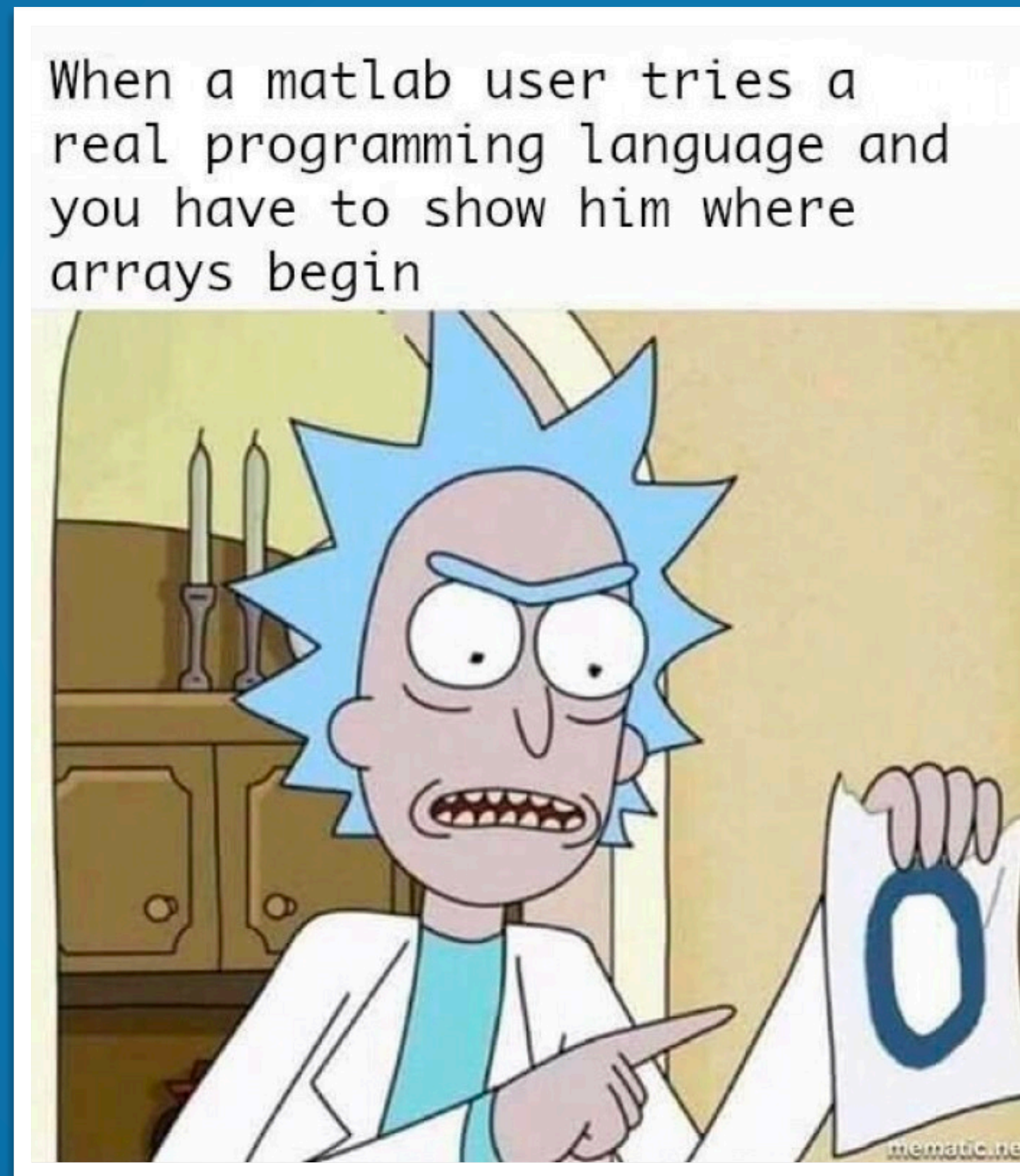
MODIFICACIÓN

```
marks(3) = 5;
```

RECUPERACIÓN

```
marks(4) → ans = 7
```


► Tipos de datos ► En Matlab los arrays empiezan en 1





EEGSignal = [10.5411, 10.6807, 13.7834, ...]

MATRIZ 4X4



	1	2	3	4
1	16	2	3	13
2	5	11	10	8
3	9	7	6	12
4	4	14	15	1



MATRIZ 4X4



	1	2	3	4
1	16	2	3	13
2	5	11	10	8
3	9	7	6	12
4	4	14	15	1

GENERACIÓN:



MATRIZ 4X4



	1	2	3	4
1	16	2	3	13
2	5	11	10	8
3	9	7	6	12
4	4	14	15	1

GENERACIÓN:

```
magic = [16, 2, 3, 13; 5, 11, 10, 8; ...];
```



Filas separadas por ; 

MATRIZ 4X4



	1	2	3	4
1	16	2	3	13
2	5	11	10	8
3	9	7	6	12
4	4	14	15	1

GENERACIÓN:

magic = [16,2,3,13;5,11,10,8;...];



Filas separadas por ; 

MATRIZ 4X4



	1	2	3	4
1	16	2	3	13
2	5	11	10	8
3	9	7	6	12
4	4	14	15	1

GENERACIÓN:



```
magic = [16,2,3,13;5,11,10,8;...];
```

MODIFICACIÓN:



Filas separadas por ;



MATRIZ 4X4



	1	2	3	4
1	16	2	3	13
2	5	11	10	8
3	9	7	6	12
4	4	14	15	1

GENERACIÓN:

`magic = [16, 2, 3, 13; 5, 11, 10, 8; ...];`

MODIFICACIÓN:

`magic(3, 2) = 5;`



Filas separadas por ;



MATRIZ 4X4



	1	2	3	4
1	16	2	3	13
2	5	11	10	8
3	9	7	6	12
4	4	14	15	1

GENERACIÓN:

`magic = [16, 2, 3, 13; 5, 11, 10, 8; ...];`

MODIFICACIÓN:

`magic(3, 2) = 5;`

RECUPERACIÓN:



Filas separadas por ;



MATRIZ 4X4



	1	2	3	4
1	16	2	3	13
2	5	11	10	8
3	9	7	6	12
4	4	14	15	1

GENERACIÓN:

`magic = [16, 2, 3, 13; 5, 11, 10, 8; ...];`

MODIFICACIÓN:

`magic(3, 2) = 5;`

RECUPERACIÓN:

`magic(1, 4)`

Filas separadas por ;



MATRIZ 4X4



	1	2	3	4
1	16	2	3	13
2	5	11	10	8
3	9	7	6	12
4	4	14	15	1

GENERACIÓN:

`magic = [16, 2, 3, 13; 5, 11, 10, 8; ...];`

MODIFICACIÓN:

`magic(3, 2) = 5;`

RECUPERACIÓN:

`magic(1, 4) → ans = 13`

MATRIZ 4X4



	1	2	3	4
1	16	2	3	13
2	5	11	10	8
3	9	7	6	12
4	4	14	15	1



MATRIZ 4X4



	1	2	3	4
1	16	2	3	13
2	5	11	10	8
3	9	7	6	12
4	4	14	15	1

EXTRACCIÓN DE UNA FILA:



MATRIZ 4X4



	1	2	3	4
1	16	2	3	13
2	5	11	10	8
3	9	7	6	12
4	4	14	15	1

EXTRACCIÓN DE UNA FILA:

`magic(2, :)`



MATRIZ 4X4



	1	2	3	4
1	16	2	3	13
2	5	11	10	8
3	9	7	6	12
4	4	14	15	1

EXTRACCIÓN DE UNA FILA:

`magic(2, :)`  `ans = 5 11 10 8`



MATRIZ 4X4



	1	2	3	4
1	16	2	3	13
2	5	11	10	8
3	9	7	6	12
4	4	14	15	1

EXTRACCIÓN DE UNA FILA:

`magic(2, :)`  `ans = 5 11 10 8`

EXTRACCIÓN DE UNA COLUMNA:



MATRIZ 4X4



	1	2	3	4
1	16	2	3	13
2	5	11	10	8
3	9	7	6	12
4	4	14	15	1

EXTRACCIÓN DE UNA FILA:

`magic(2, :)`  `ans = 5 11 10 8`

EXTRACCIÓN DE UNA COLUMNA:

`magic(:, 2)` 

MATRIZ 4X4



	1	2	3	4
1	16	2	3	13
2	5	11	10	8
3	9	7	6	12
4	4	14	15	1

EXTRACCIÓN DE UNA FILA:

`magic(2, :)`  `ans = 5 11 10 8`

EXTRACCIÓN DE UNA COLUMNA:

`magic(:, 2)`  `ans = 2 11 7 14`



MATRIZ 4X4



	1	2	3	4
1	16	2	3	13
2	5	11	10	8
3	9	7	6	12
4	4	14	15	1

EXTRACCIÓN DE UNA FILA:

`magic(2, :)`  `ans = 5 11 10 8`

EXTRACCIÓN DE UNA COLUMNA:

`magic(:, 2)`  `ans = 2 11 7 14`

EXTRACCIÓN SUBMATRIZ:



MATRIZ 4X4



	1	2	3	4
1	16	2	3	13
2	5	11	10	8
3	9	7	6	12
4	4	14	15	1

EXTRACCIÓN DE UNA FILA:

`magic(2, :)` → `ans = 5 11 10 8`

EXTRACCIÓN DE UNA COLUMNA:

`magic(:, 2)` → `ans = 2 11 7 14`

EXTRACCIÓN SUBMATRIZ:

`magic(1:2, 3:4)` →

MATRIZ 4X4



	1	2	3	4
1	16	2	3	13
2	5	11	10	8
3	9	7	6	12
4	4	14	15	1

EXTRACCIÓN DE UNA FILA:

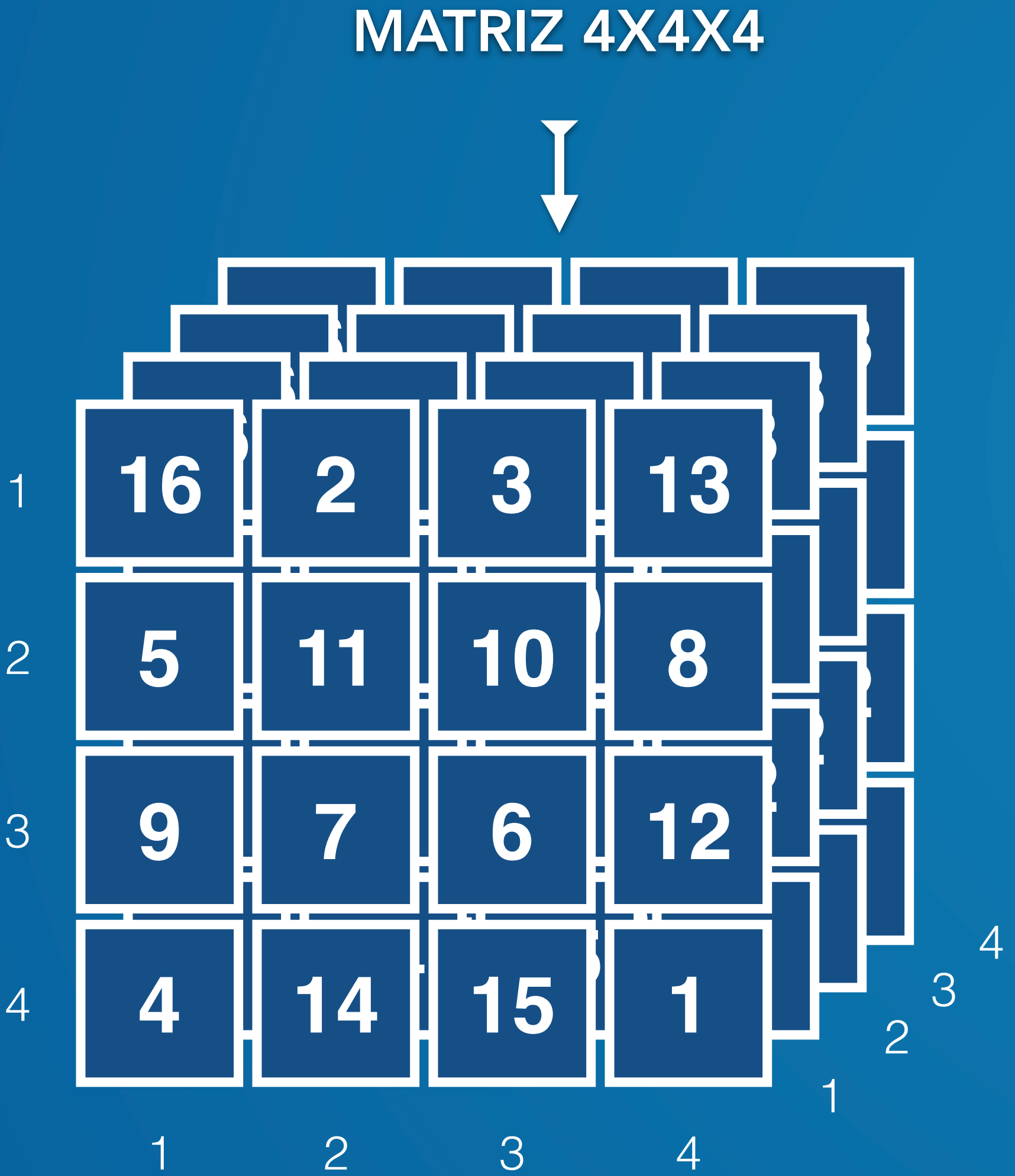
`magic(2, :)` \longrightarrow `ans = 5 11 10 8`

EXTRACCIÓN DE UNA COLUMNA:

`magic(:, 2)` \longrightarrow `ans = 2 11 7 14`

EXTRACCIÓN SUBMATRIZ:

`magic(1:2, 3:4)` \longrightarrow `ans = 3 13
10 8`



MATRIZ 4X4X4

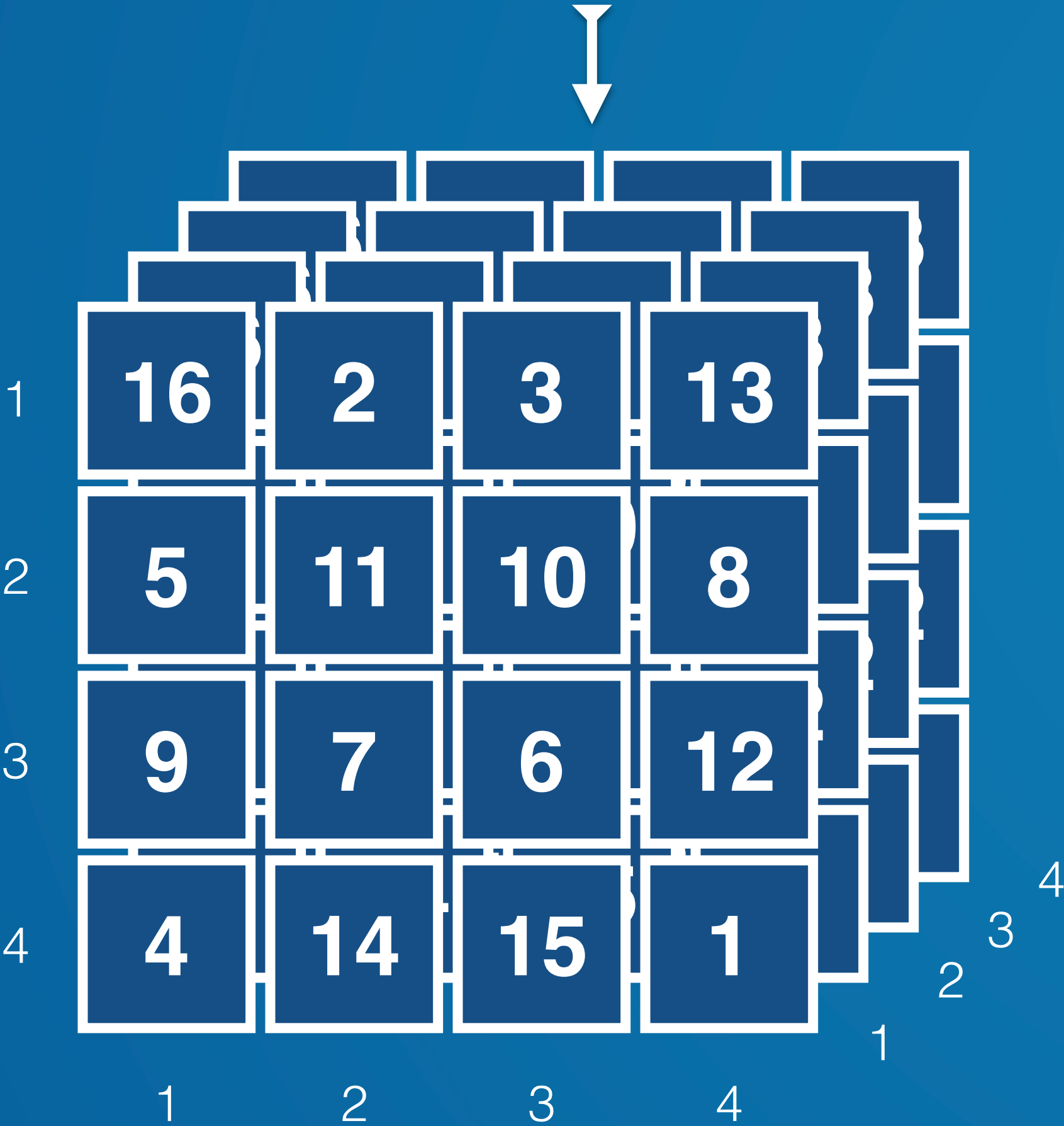


1	16	2	3	13			
2	5	11	10	8			
3	9	7	6	12			
4	4	14	15	1			
	1	2	3	4			

RECUPERACIÓN:



MATRIZ 4X4X4

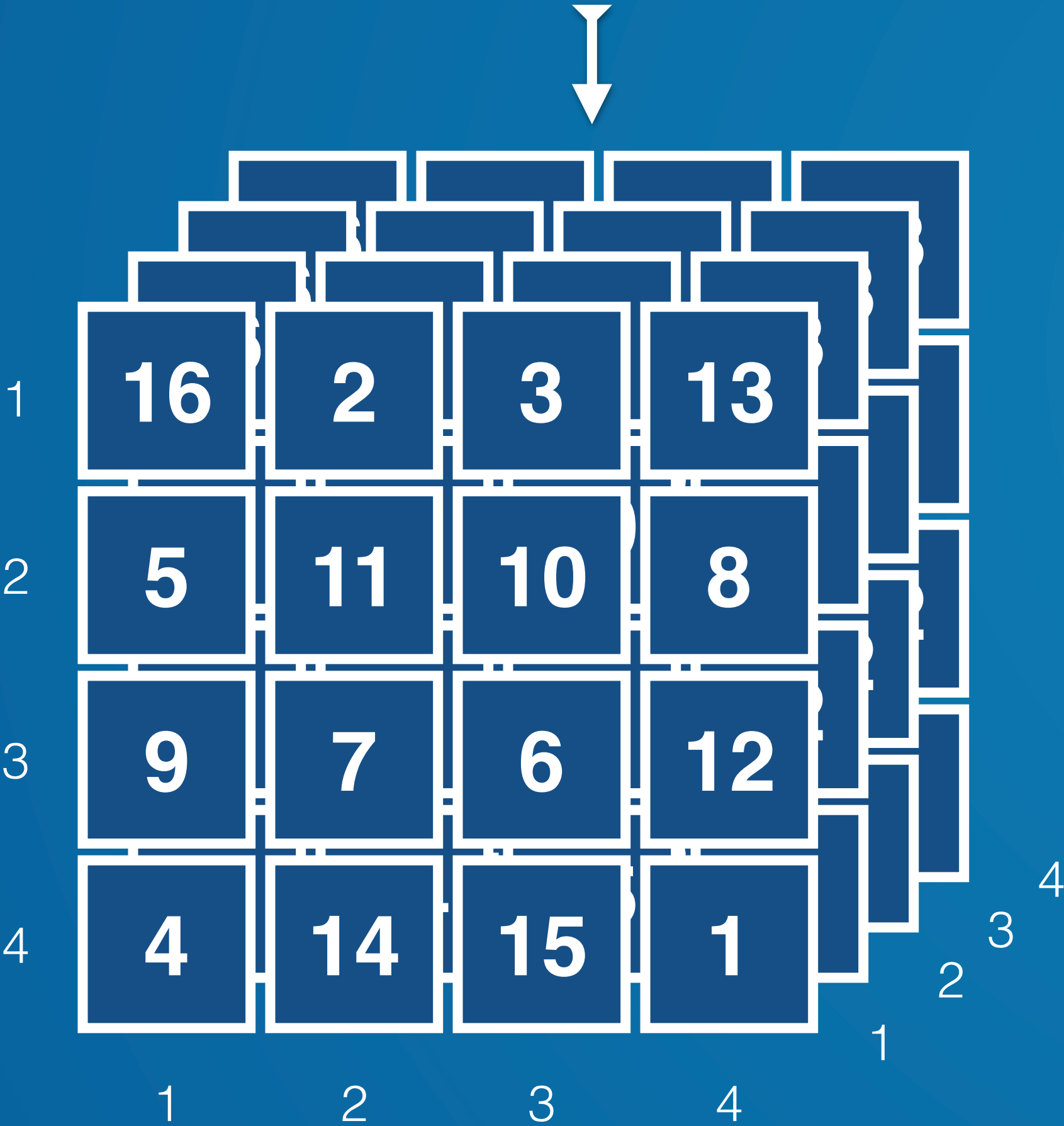


RECUPERACIÓN:

`magic(4, 4, 1)`



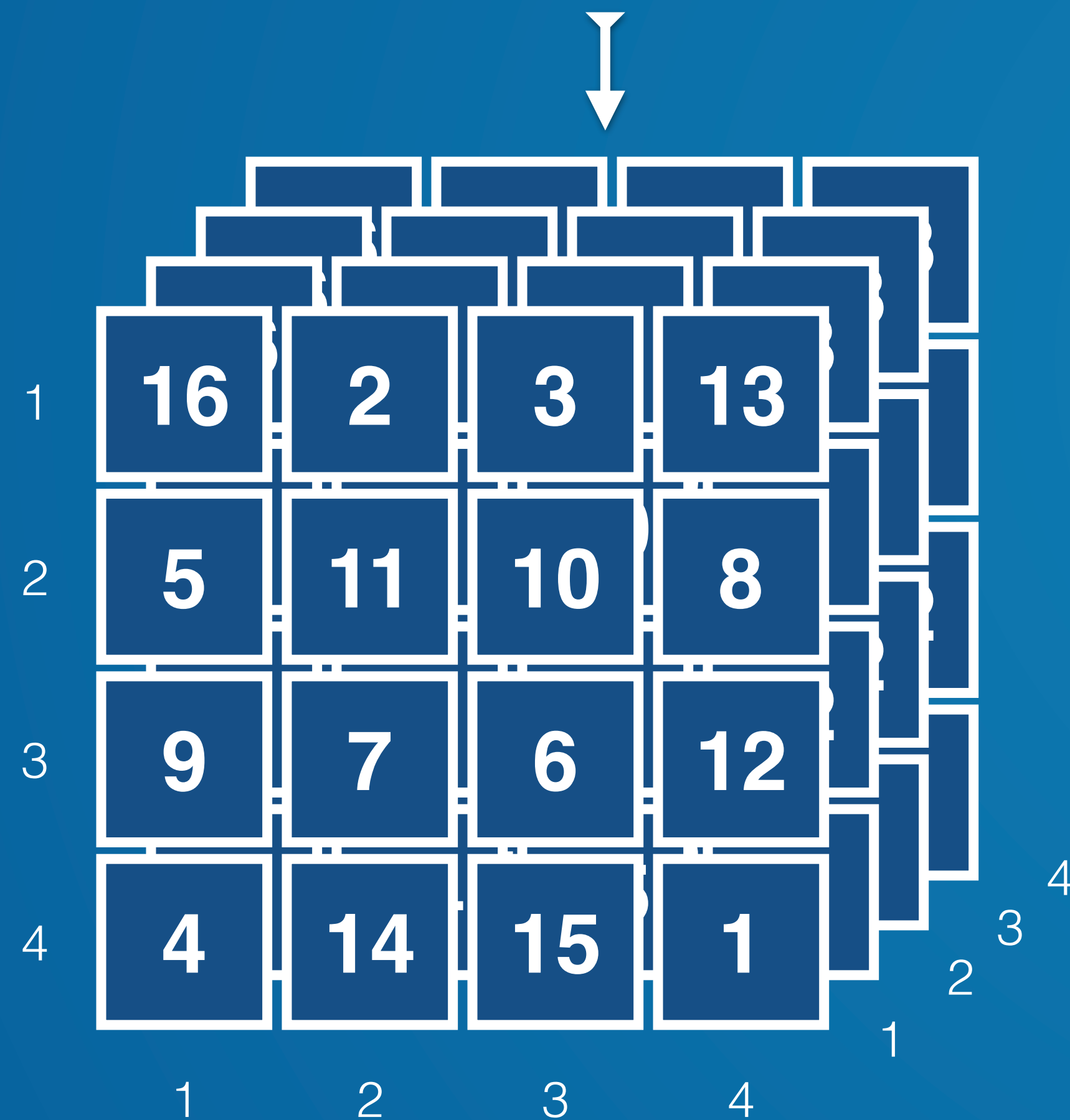
MATRIZ 4X4X4



RECUPERACIÓN:

`magic(4, 4, 1)` → `ans = 1`

MATRIZ 4X4X4

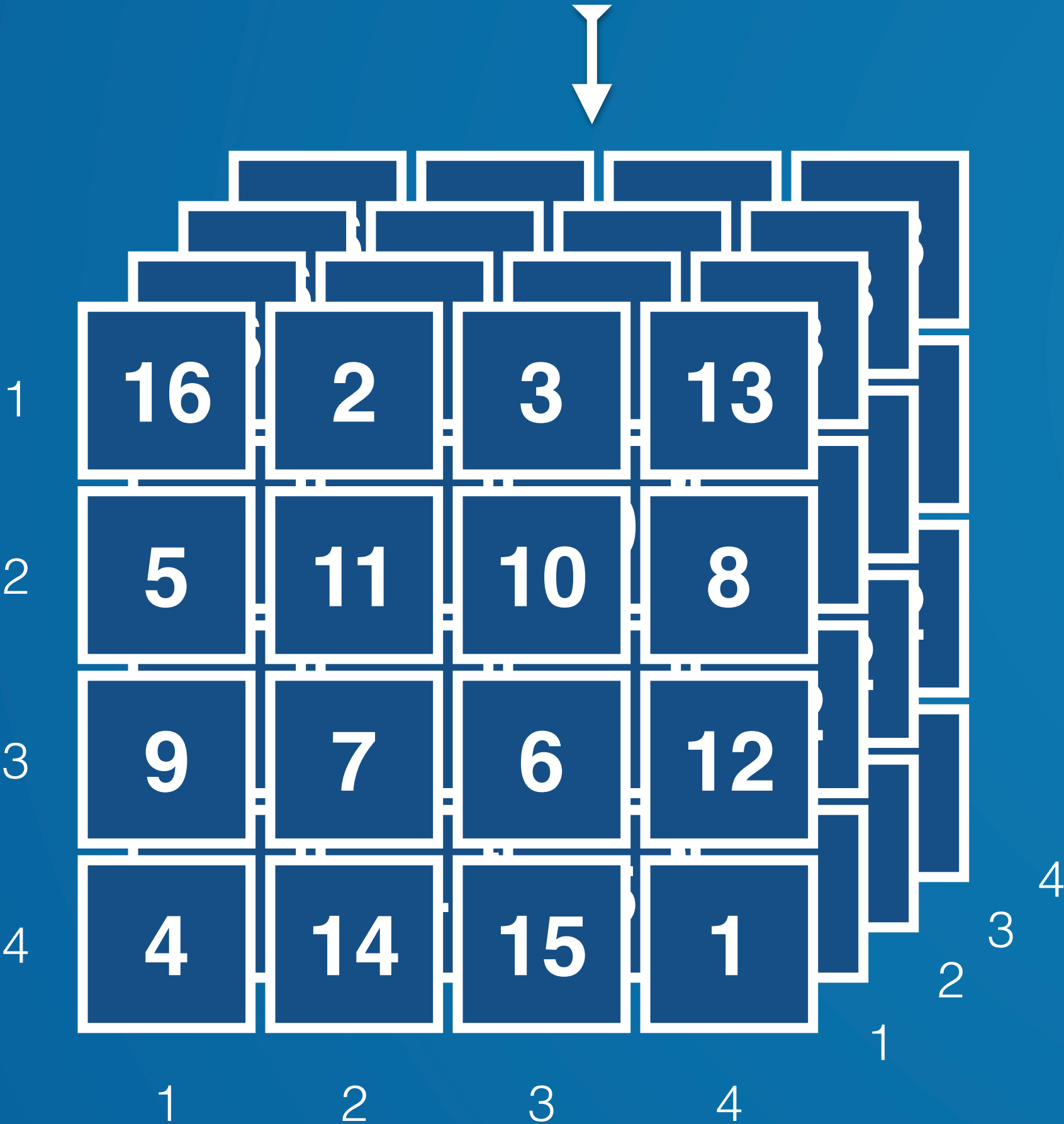


RECUPERACIÓN:

`magic(4, 4, 1)` → `ans = 1`

`magic(4, 4, 4)` →

MATRIZ 4X4X4

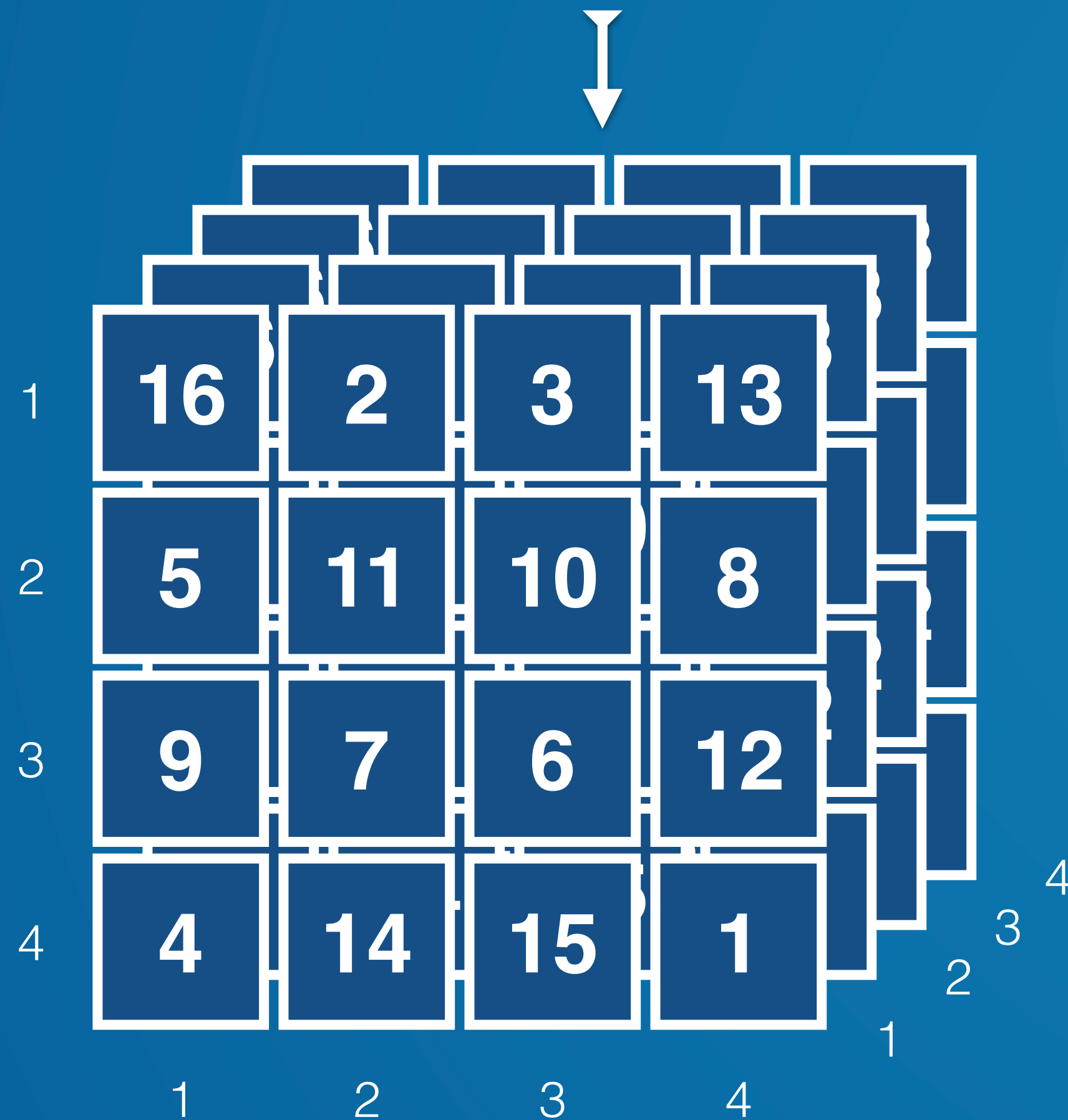


RECUPERACIÓN:

`magic(4, 4, 1)` \longrightarrow `ans = 1`

`magic(4, 4, 4)` \longrightarrow `ans = 6`

MATRIZ 4X4X4



RECUPERACIÓN:

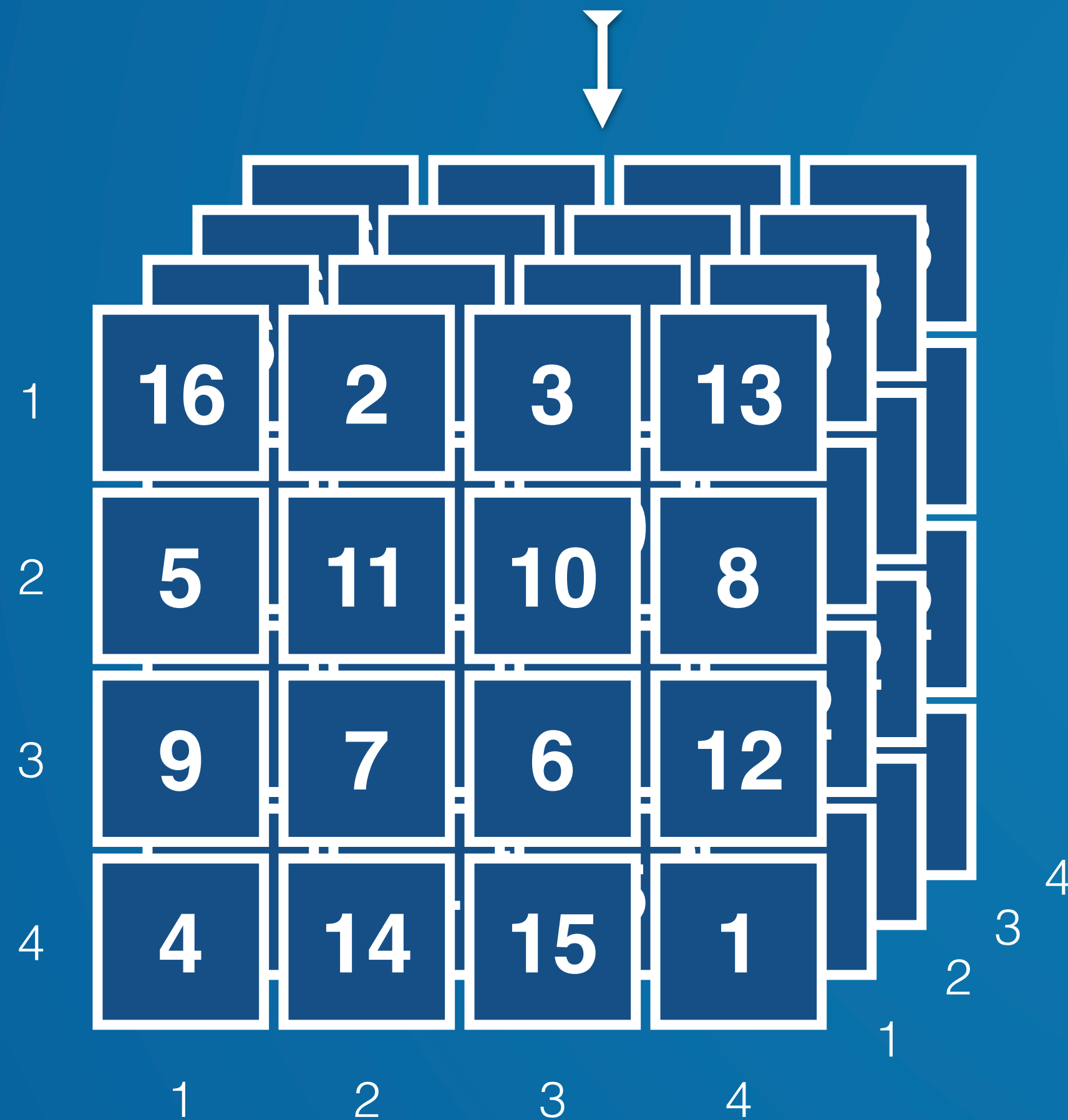
`magic(4, 4, 1)` → `ans = 1`

`magic(4, 4, 4)` → `ans = 6`



NO SOLO TENEMOS 3 DIMENSIONES

MATRIZ 4X4X4



RECUPERACIÓN:

`magic(4, 4, 1)` → `ans = 1`

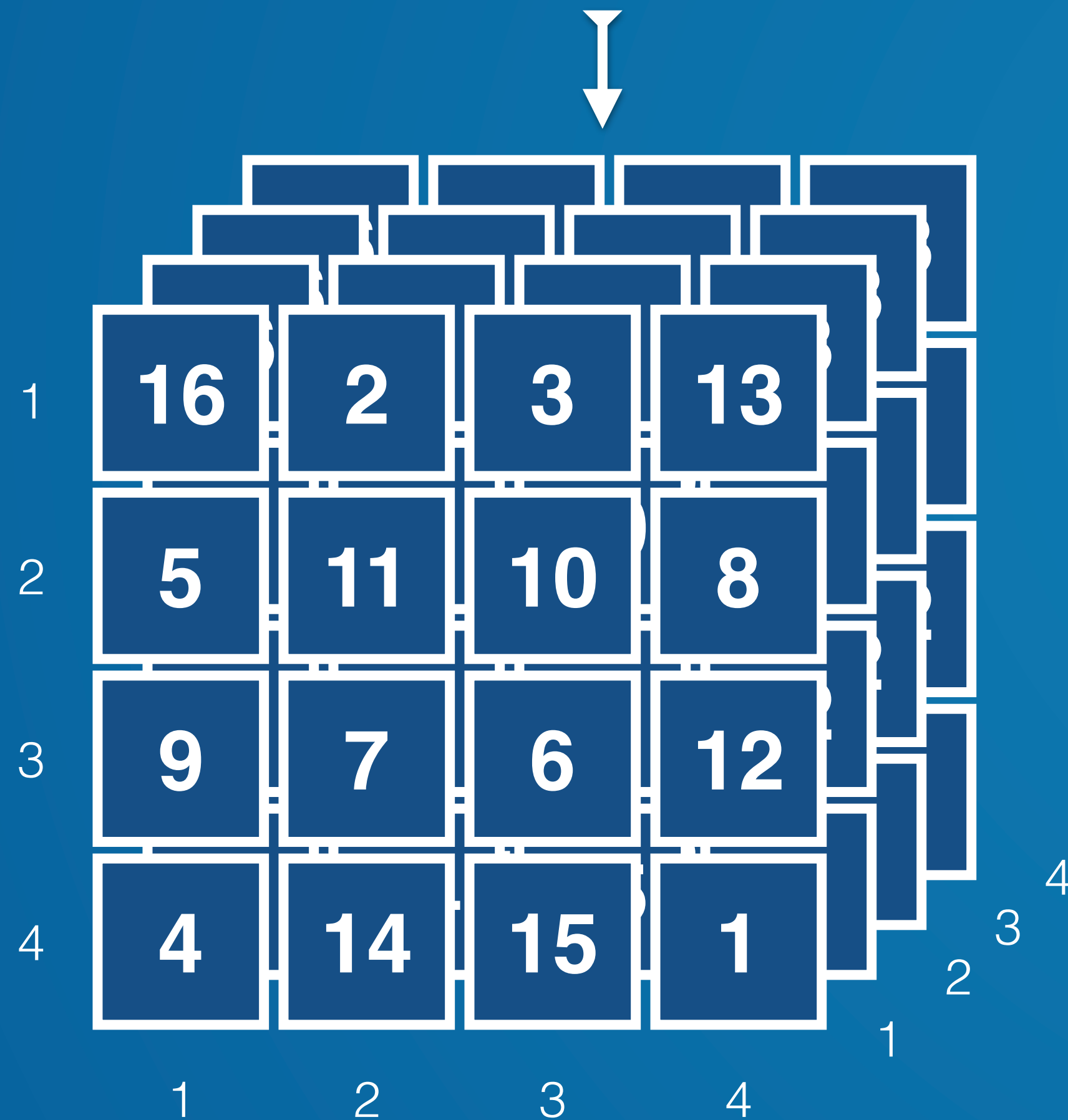
`magic(4, 4, 4)` → `ans = 6`



NO SOLO TENEMOS 3 DIMENSIONES

`magic(4, 4, 4, 4)`

MATRIZ 4X4X4



RECUPERACIÓN:

`magic(4, 4, 1)` \longrightarrow `ans = 1`

`magic(4, 4, 4)` \longrightarrow `ans = 6`



NO SOLO TENEMOS 3 DIMENSIONES

`magic(4, 4, 4, 4)`

`magic(4, 4, 4, 4, 4, 4)`



CARACTER
(Char)



CARACTER
(Char) → 'A'



CARACTER
(Char)



'A'

CADENA
(Char)



CARACTER
(Char) → 'A'

CADENA
(Char) → '¡Hola mundo!'



CARACTER
(Char)



'A'

CADENA
(Char)



'¡Hola mundo!'

CADENA
(String)



CARACTER
(Char) → 'A'

CADENA
(Char) → '¡Hola mundo!'

CADENA
(String) → "¡Hola mundo!"



(Char) **GENERACIÓN:**



(Char) GENERACIÓN:

```
saludo = '¡Hola!';
```



(Char) GENERACIÓN:

```
saludo = '¡Hola!';
```

(Char) COMBINACIÓN:



(Char) GENERACIÓN:

```
saludo = '¡Hola!';
```

(Char) COMBINACIÓN:

```
saludoMejorado = [saludo '¿Qué tal?'];
```



(Char) GENERACIÓN:

```
saludo = '¡Hola!';
```

(Char) COMBINACIÓN:

```
saludoMejorado = [saludo '¿Qué tal?'];
```

(Char) RECUPERACIÓN:



(Char) GENERACIÓN:

```
saludo = '¡Hola!';
```

(Char) COMBINACIÓN:

```
saludoMejorado = [saludo '¿Qué tal?'];
```

(Char) RECUPERACIÓN:

```
saludoMejorado ➡
```

(Char) GENERACIÓN:

```
saludo = '¡Hola!';
```

(Char) COMBINACIÓN:

```
saludoMejorado = [saludo '¿Qué tal?'];
```

(Char) RECUPERACIÓN:

```
saludoMejorado → '!Hola! ¿Qué tal?'
```



(String) GENERACIÓN:



(String) GENERACIÓN:

```
saludo = "¡Hola!";
```



(String) GENERACIÓN:

```
saludo = "¡Hola!";
```

(String) COMBINACIÓN:



(String) GENERACIÓN:

```
saludo = "¡Hola!";
```

(String) COMBINACIÓN:

```
saludos = ["¡Hola!", "¿Qué tal?", "Buenos días"];
```



(String) GENERACIÓN:

```
saludo = "¡Hola!";
```

(String) COMBINACIÓN:

```
saludos = ["¡Hola!", "¿Qué tal?", "Buenos días"];
```

(String) RECUPERACIÓN:



(String) GENERACIÓN:

```
saludo = "¡Hola!";
```

(String) COMBINACIÓN:

```
saludos = ["¡Hola!", "¿Qué tal?", "Buenos días"];
```

(String) RECUPERACIÓN:

```
saludos(2) ➡
```


(String) GENERACIÓN:

```
saludo = "¡Hola!";
```


(String) COMBINACIÓN:

```
saludos = ["¡Hola!", "¿Qué tal?", "Buenos días"];
```

(String) RECUPERACIÓN:

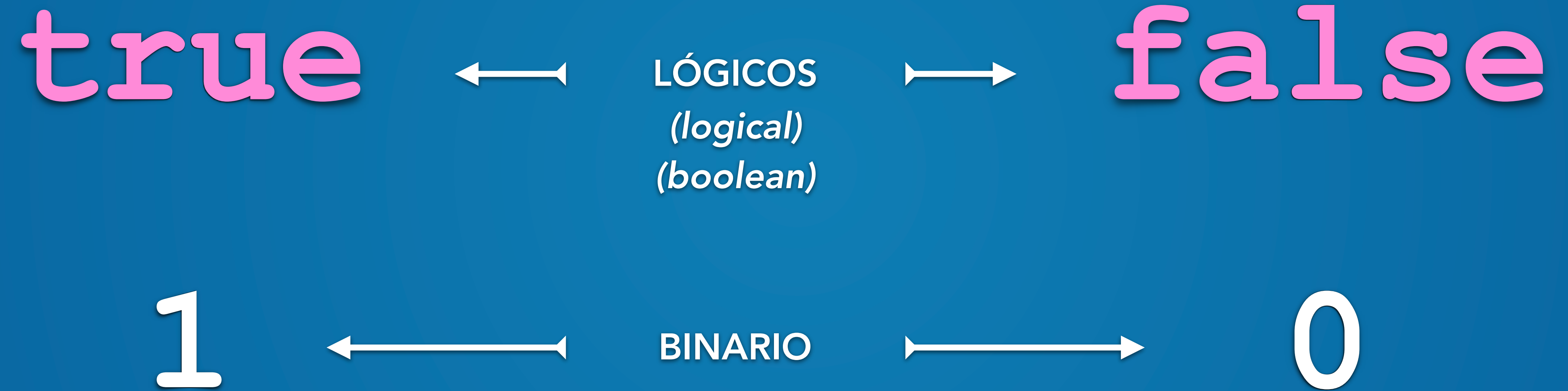
```
saludos(2) → ans = "¿Qué tal?"
```

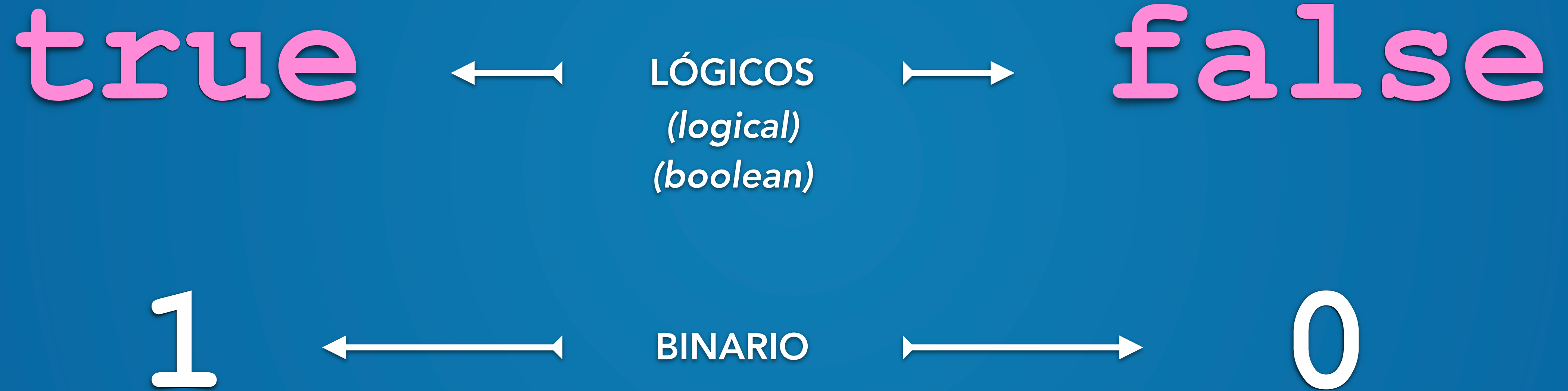




LÓGICOS
(logical)
(boolean)

true ← LÓGICOS
(logical)
(boolean) → **false**





Suelen usarse como variables de control



(logical) GENERACIÓN:



(logical) GENERACIÓN:

```
isMarch = true;
```



(logical) GENERACIÓN:

```
isMarch = true;
```

```
isMarch = 1;
```



(logical) GENERACIÓN:

```
isMarch = true;
```

```
isFebruary = false;
```

```
isMarch = 1;
```



(logical) GENERACIÓN:

```
isMarch = true;  
isFebruary = false;
```

```
isMarch = 1;  
isFebruary = 0;
```



(logical) GENERACIÓN:

```
isMarch = true;  
isFebruary = false;
```

```
isMarch = 1;  
isFebruary = 0;
```

(logical) RECUPERACIÓN:



(logical) GENERACIÓN:

```
isMarch = true;  
isFebruary = false;
```

```
isMarch = 1;  
isFebruary = 0;
```

(logical) RECUPERACIÓN:

```
isMarch →
```

(logical) GENERACIÓN:

```
isMarch = true;  
isFebruary = false;
```

```
isMarch = 1;  
isFebruary = 0;
```

(logical) RECUPERACIÓN:

```
isMarch        ans = true
```




{cell}{cellarray}

ARRAY DE CELDAS
(*cell array*)



{cell}{cellarray}

ARRAY DE CELDAS
(*cell array*)



{cell}{cellarray}

ARRAY DE CELDAS
(*cell array*)



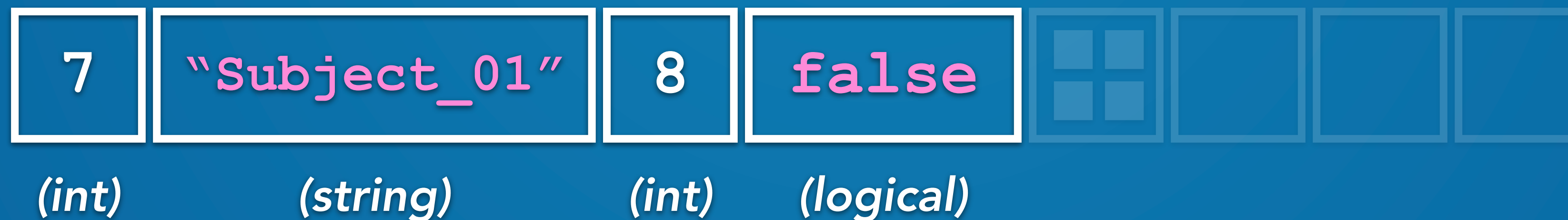
{cell}{cellarray}

ARRAY DE CELDAS
(*cell array*)



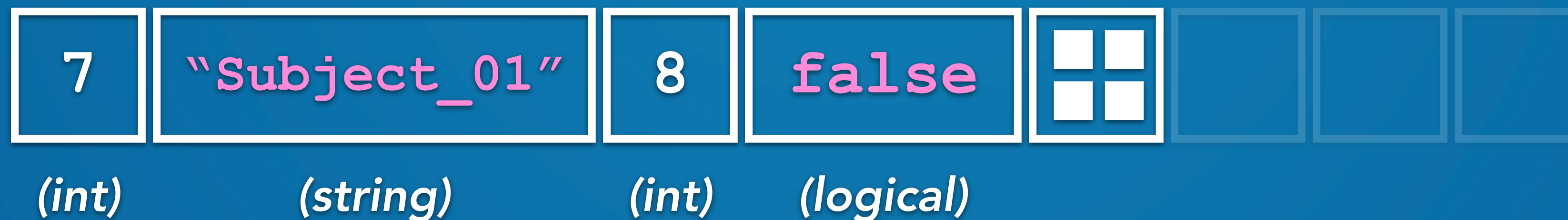
{cell}{cellarray}

ARRAY DE CELDAS
(*cell array*)



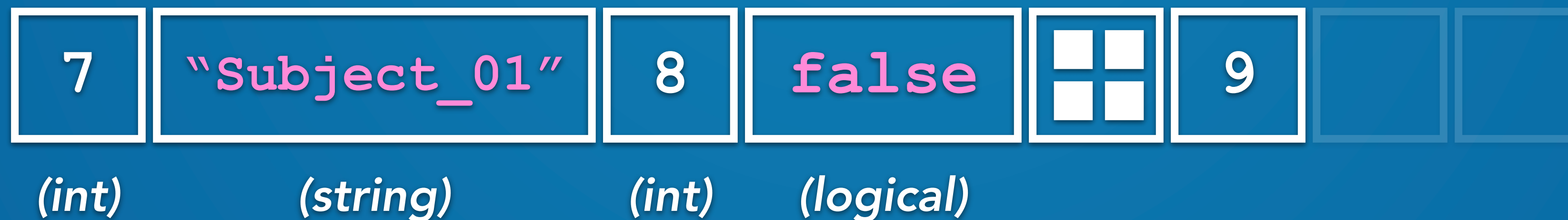
{cell}{cellarray}

ARRAY DE CELDAS
(*cell array*)



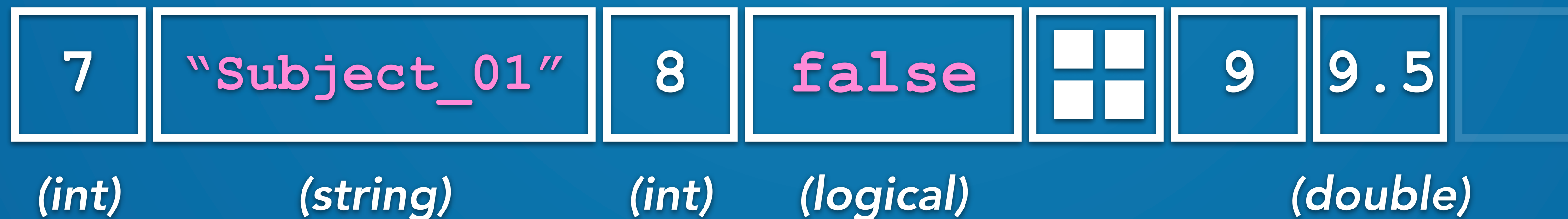
{cell}{cellarray}

ARRAY DE CELDAS
(*cell array*)



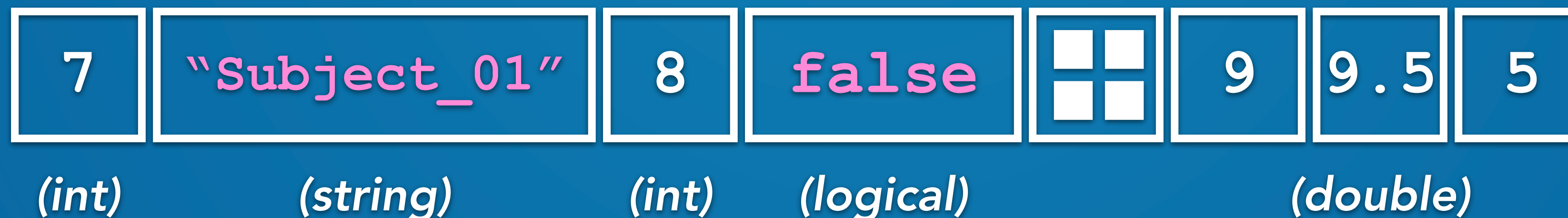
{cell}{cellarray}



ARRAY DE CELDAS
(*cell array*)

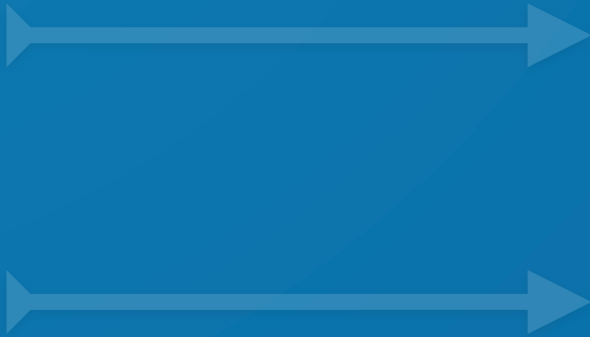



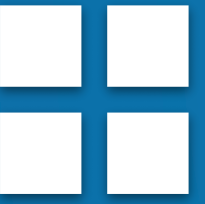
{cell}{cellarray}

ARRAY DE CELDAS
(*cell array*)

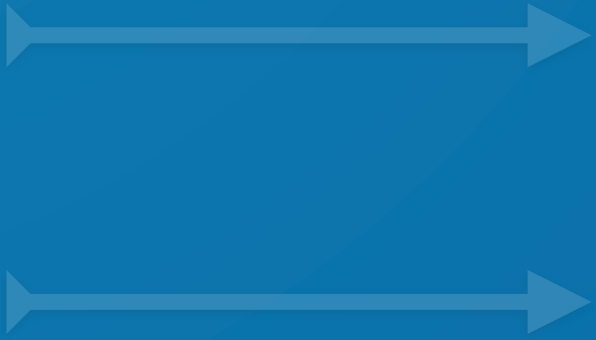




	1	2	3
1	(double)		(double)
2	(int)	(string)	(int)
3	(string)	(logical)	
4	(int)	(string)	(int)



	1	2	3
1	(double)		(double)
2	(int)	(string)	(int)
3	(string)	(logical)	
4	(int)	(string)	(int)

(cell) GENERACIÓN O EDICIÓN:





	1	2	3
1	(double)		(double)
2	(int)	(string)	(int)
3	(string)	(logical)	
4	(int)	(string)	(int)

(cell) GENERACIÓN O EDICIÓN:

`celdas{1,1} = 2.54;`




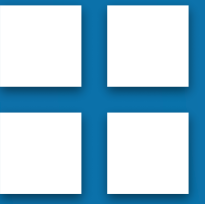
	1	2	3
1	(double)		(double)
2	(int)	(string)	(int)
3	(string)	(logical)	
4	(int)	(string)	(int)

(cell) GENERACIÓN O EDICIÓN:

`celdas{1,1} = 2.54;`

`celdas{1,2} = [1,2,3;4,5,6;7,8,9];`



	1	2	3
1	(double)		(double)
2	(int)	(string)	(int)
3	(string)	(logical)	
4	(int)	(string)	(int)

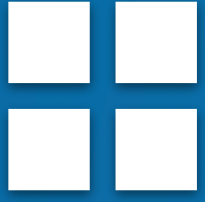
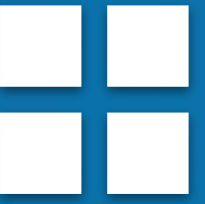
(cell) GENERACIÓN O EDICIÓN:

```
celdas{1,1} = 2.54;
```

```
celdas{1,2} = [1,2,3;4,5,6;7,8,9];
```

```
celdas{2,2} = "Subject_007";
```



	1	2	3
1	(double)		(double)
2	(int)	(string)	(int)
3	(string)	(logical)	
4	(int)	(string)	(int)

(cell) GENERACIÓN O EDICIÓN:

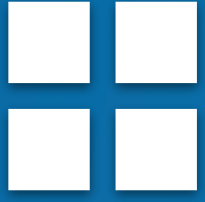

```
celdas{1,1} = 2.54;
```

```
celdas{1,2} = [1,2,3;4,5,6;7,8,9];
```

```
celdas{2,2} = "Subject_007";
```

(cell) RECUPERACIÓN:



	1	2	3
1	(double)		(double)
2	(int)	(string)	(int)
3	(string)	(logical)	
4	(int)	(string)	(int)

(cell) GENERACIÓN O EDICIÓN:

```
celdas{1,1} = 2.54;
```

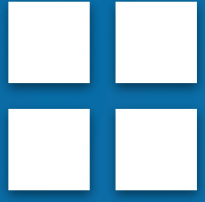

```
celdas{1,2} = [1,2,3;4,5,6;7,8,9];
```

```
celdas{2,2} = "Subject_007";
```

(cell) RECUPERACIÓN:

```
celdas(1,1) →
```

```
→
```

	1	2	3
1	(double)		(double)
2	(int)	(string)	(int)
3	(string)	(logical)	
4	(int)	(string)	(int)

(cell) GENERACIÓN O EDICIÓN:

```
celdas{1,1} = 2.54;
```

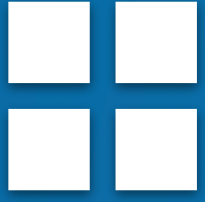

```
celdas{1,2} = [1,2,3;4,5,6;7,8,9];
```

```
celdas{2,2} = "Subject_007";
```

(cell) RECUPERACIÓN:

```
celdas(1,1) → ans = cell
```



	1	2	3
1	(double)		(double)
2	(int)	(string)	(int)
3	(string)	(logical)	
4	(int)	(string)	(int)

(cell) GENERACIÓN O EDICIÓN:

```
celdas{1,1} = 2.54;
```

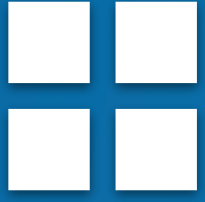

```
celdas{1,2} = [1,2,3;4,5,6;7,8,9];
```

```
celdas{2,2} = "Subject_007";
```

(cell) RECUPERACIÓN:

```
celdas(1,1) → ans = cell
```

```
celdas{1,1} →
```

	1	2	3
1	(double)		(double)
2	(int)	(string)	(int)
3	(string)	(logical)	
4	(int)	(string)	(int)

(cell) GENERACIÓN O EDICIÓN:

```
celdas{1,1} = 2.54;
```

```
celdas{1,2} = [1,2,3;4,5,6;7,8,9];
```

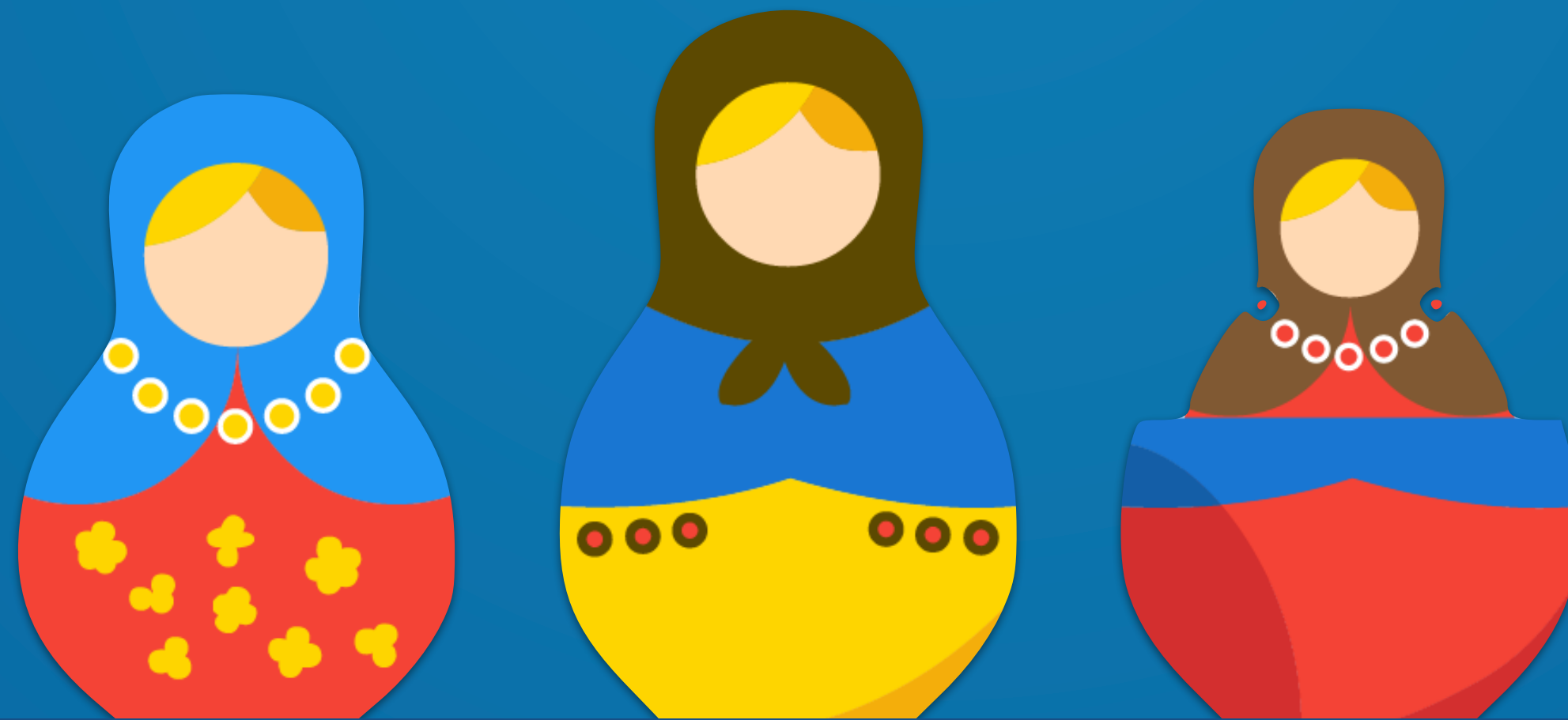
```
celdas{2,2} = "Subject_007";
```

(cell) RECUPERACIÓN:

```
celdas(1,1) → ans = cell
```

```
celdas{1,1} → ans = 2.54
```


ESTRUCTURAS DE DATOS (*struct*)





(struct) GENERACIÓN:



(struct) GENERACIÓN:

```
sujeto.edad = 23;
```



(struct) GENERACIÓN:

```
sujeto.edad = 23;
```

```
sujeto.nombre = "Laura";
```



(struct) GENERACIÓN:

```
sujeto.edad = 23;
```

```
sujeto.nombre = "Laura";
```

```
sujeto.puntuacion = [8, 7, 5, 9];
```



(struct) GENERACIÓN:

```
sujeto.edad = 23;
```

```
sujeto.nombre = "Laura";
```

```
sujeto.puntuacion = [8, 7, 5, 9];
```

(struct) RECUPERACIÓN:



(struct) GENERACIÓN:

```
sujeto.edad = 23;
```

```
sujeto.nombre = "Laura";
```

```
sujeto.puntuacion = [8, 7, 5, 9];
```

(struct) RECUPERACIÓN:

```
sujeto.puntuacion(2) ➡
```

(struct) GENERACIÓN:

```
sujeto.edad = 23;
```

```
sujeto.nombre = "Laura";
```

```
sujeto.puntuacion = [8, 7, 5, 9];
```

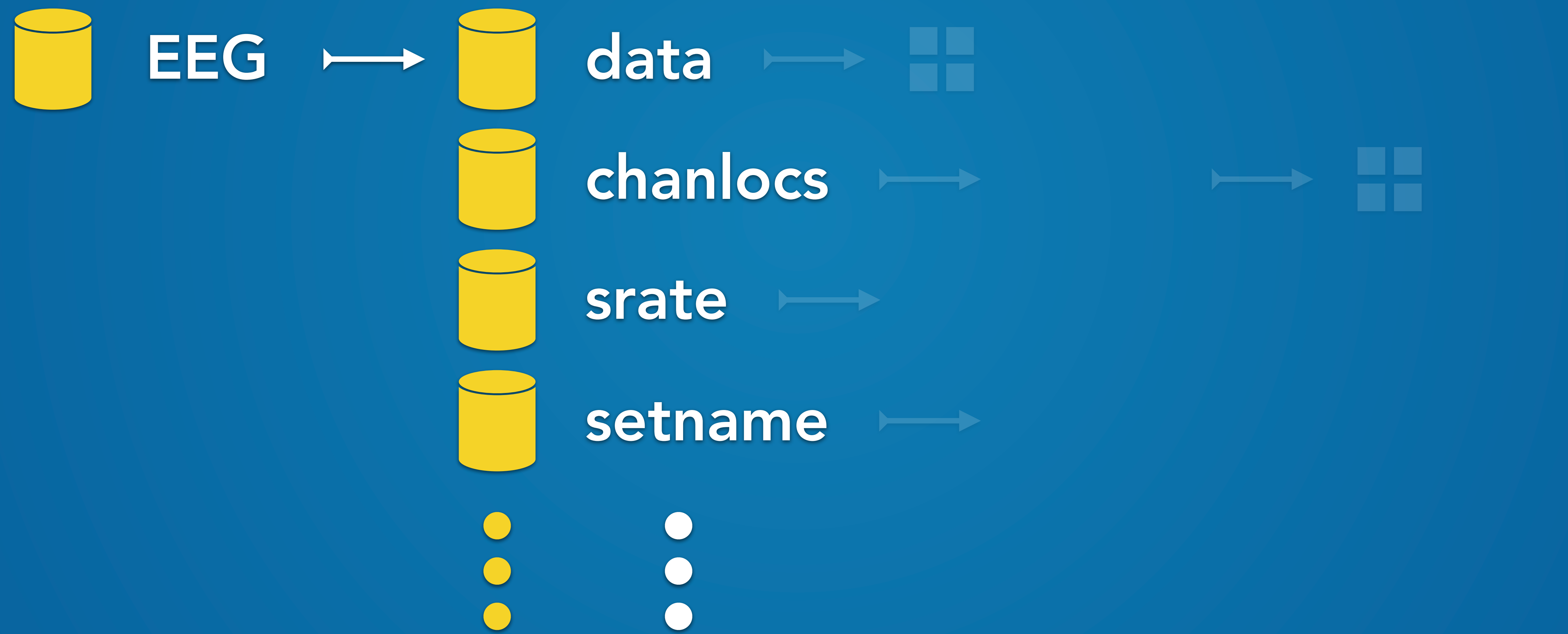
(struct) RECUPERACIÓN:

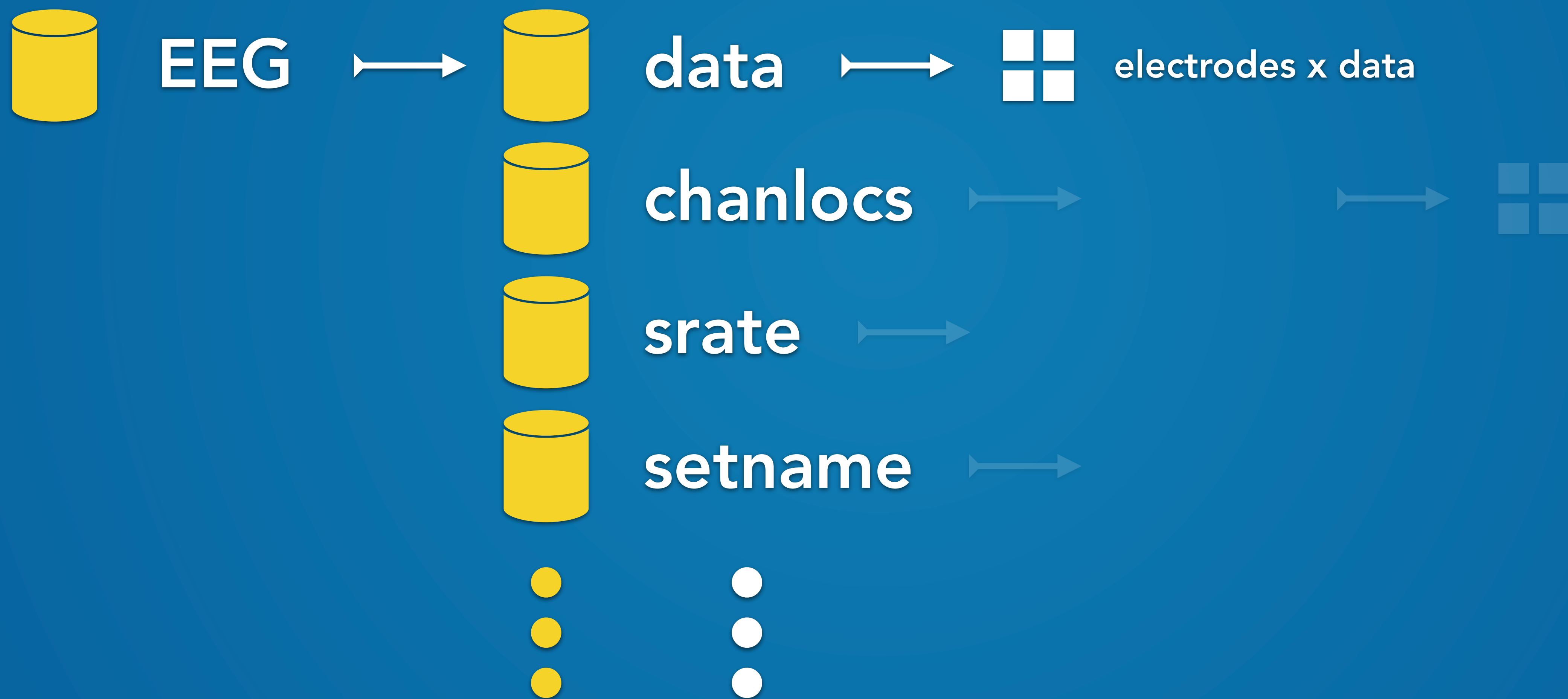
```
sujeto.puntuacion(2) → ans = 7
```

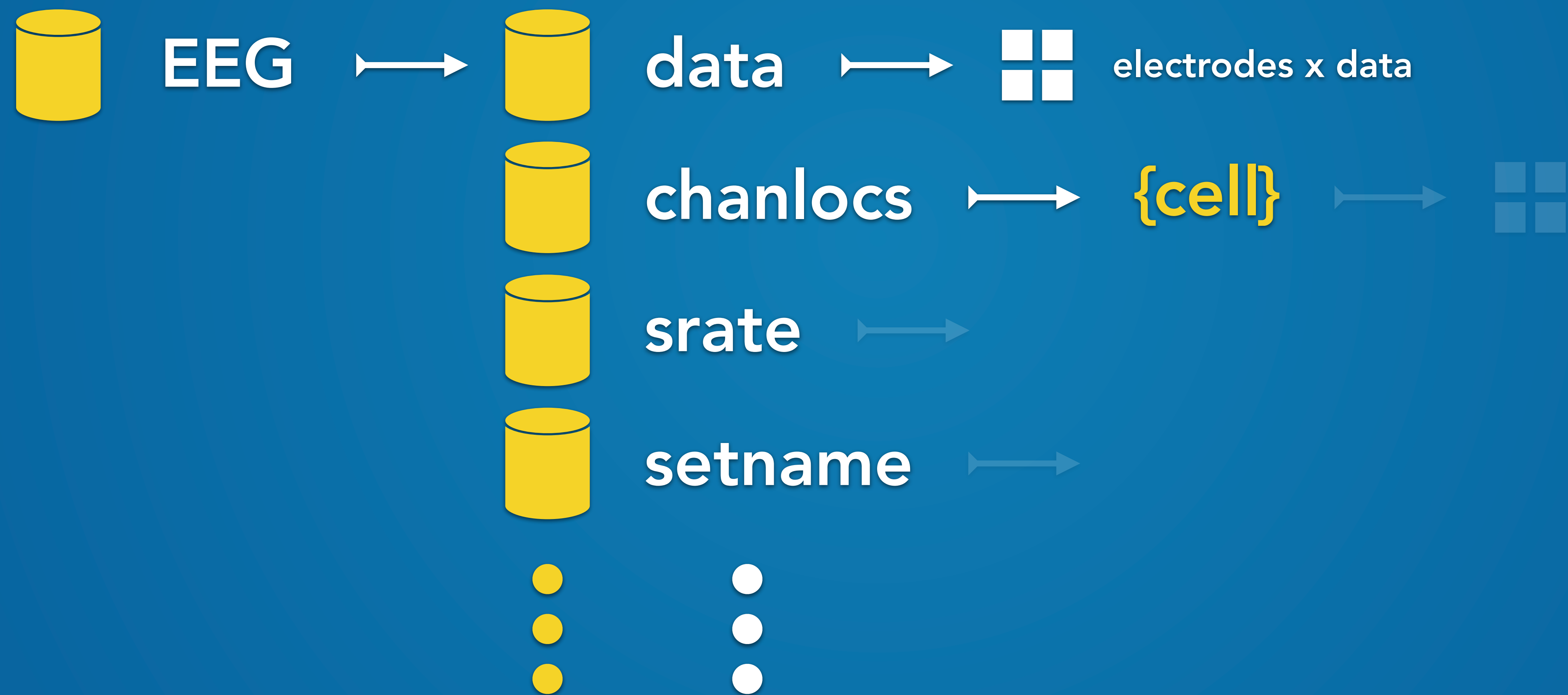


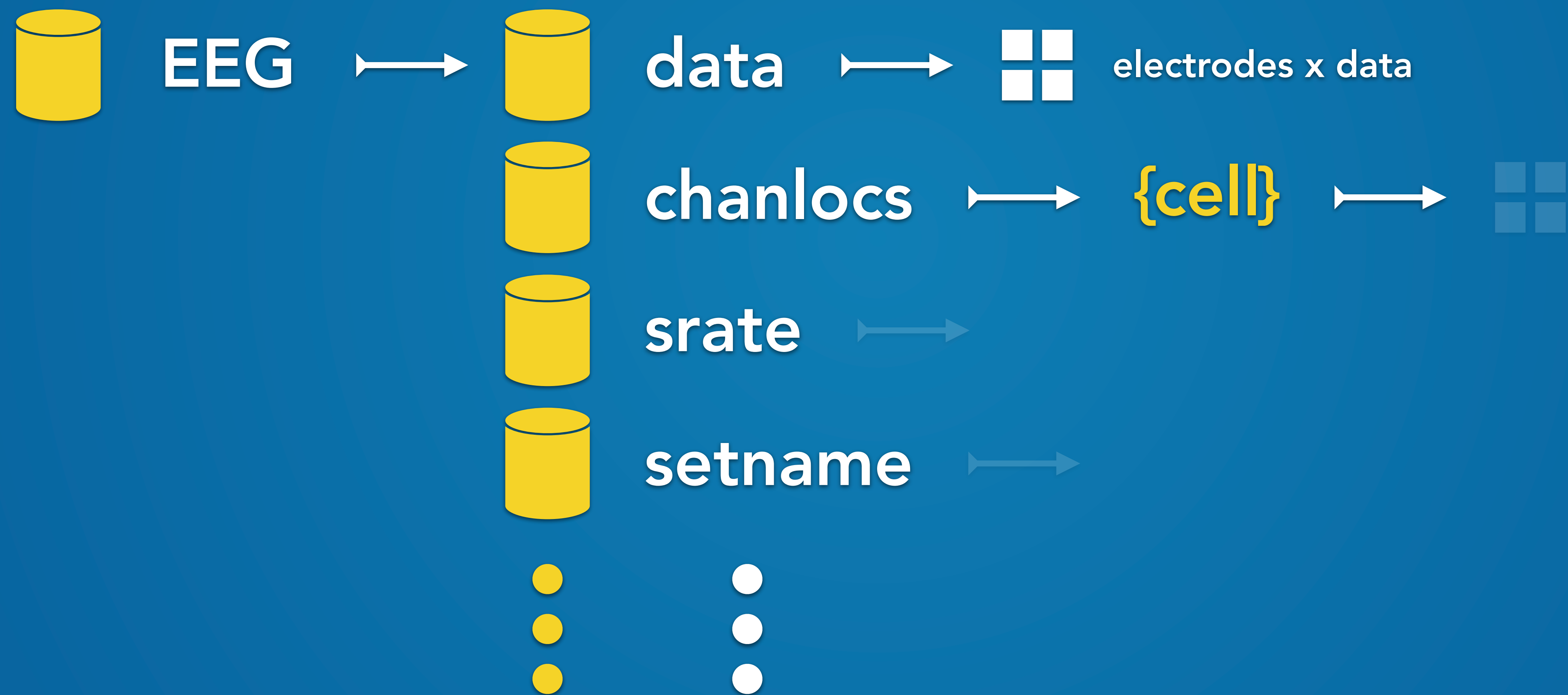


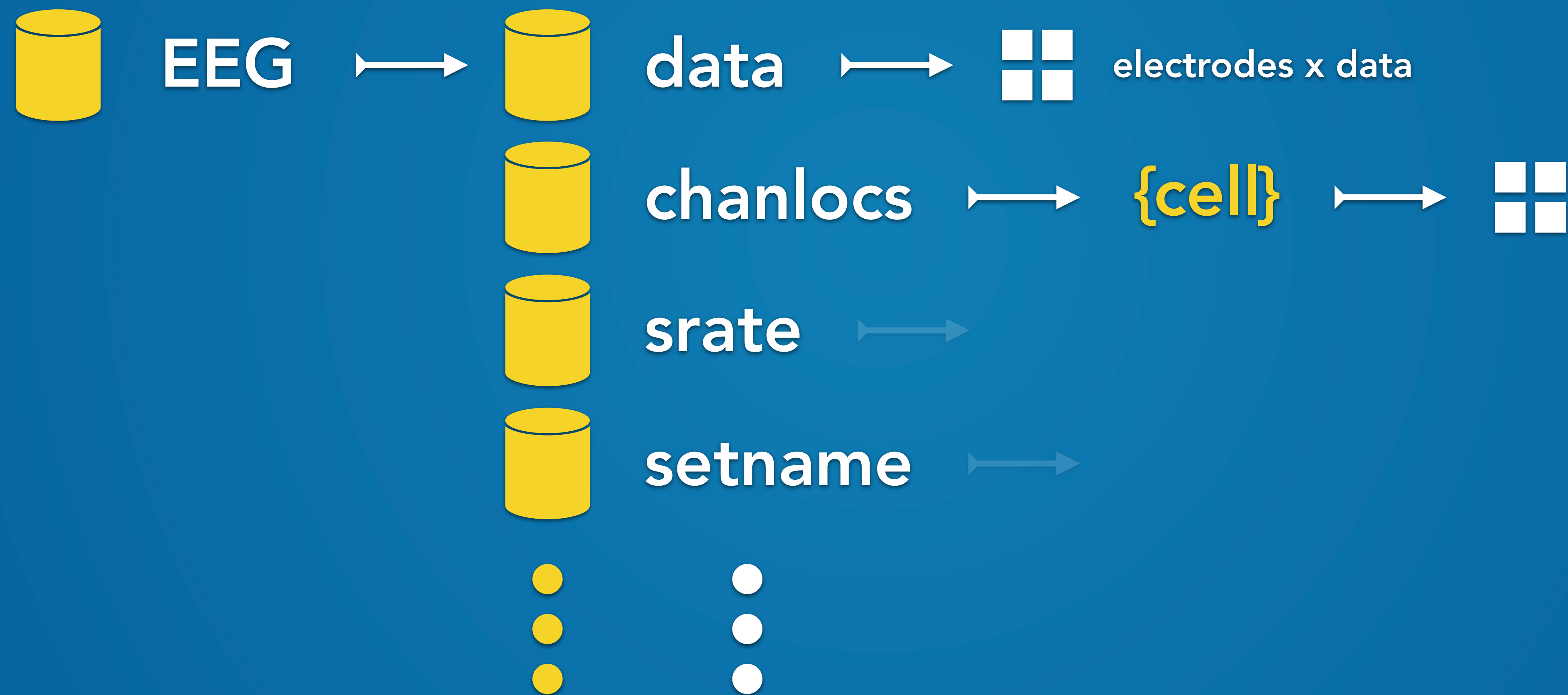


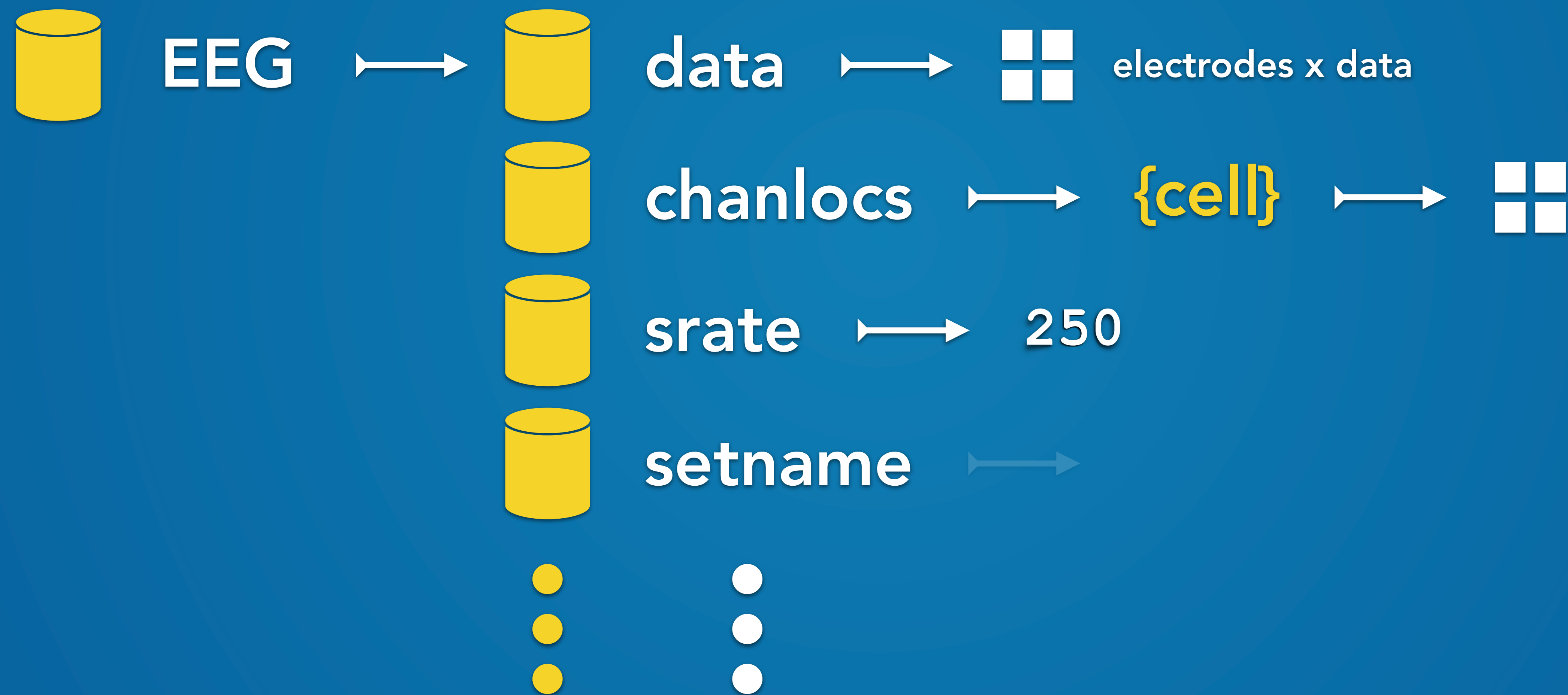


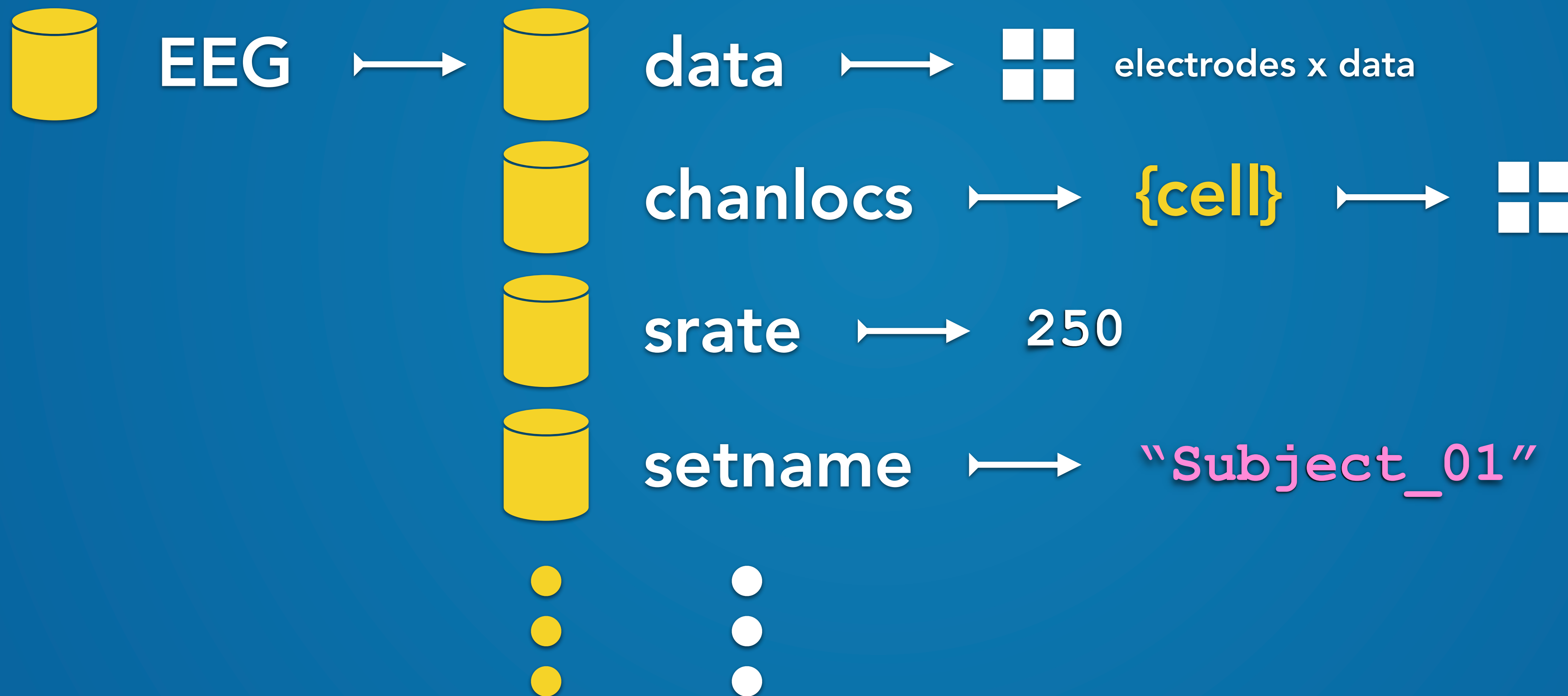














Matlab: tipado dinámico



Matlab: tipado dinámico

TIPADO ESTÁTICO



Matlab: tipado dinámico

TIPADO ESTÁTICO



```
int valor = 3;  
int valor = 'Tres';
```



```
valor = 3;  
valor = 'Tres';
```

Matlab: tipado dinámico

TIPADO ESTÁTICO

→ `int valor = 3;`
 `int valor = 'Tres';`



Matlab: tipado dinámico

TIPADO ESTÁTICO

→ `int valor = 3;`
 `int valor = 'Tres';`

TIPADO DINÁMICO



Matlab: tipado dinámico

TIPADO ESTÁTICO

→ `int valor = 3;`
 `int valor = 'Tres';`

TIPADO DINÁMICO

→ `valor = 3;`
`valor = 'Tres';`

Matlab: tipado dinámico

TIPADO ESTÁTICO

→ `int valor = 3;`



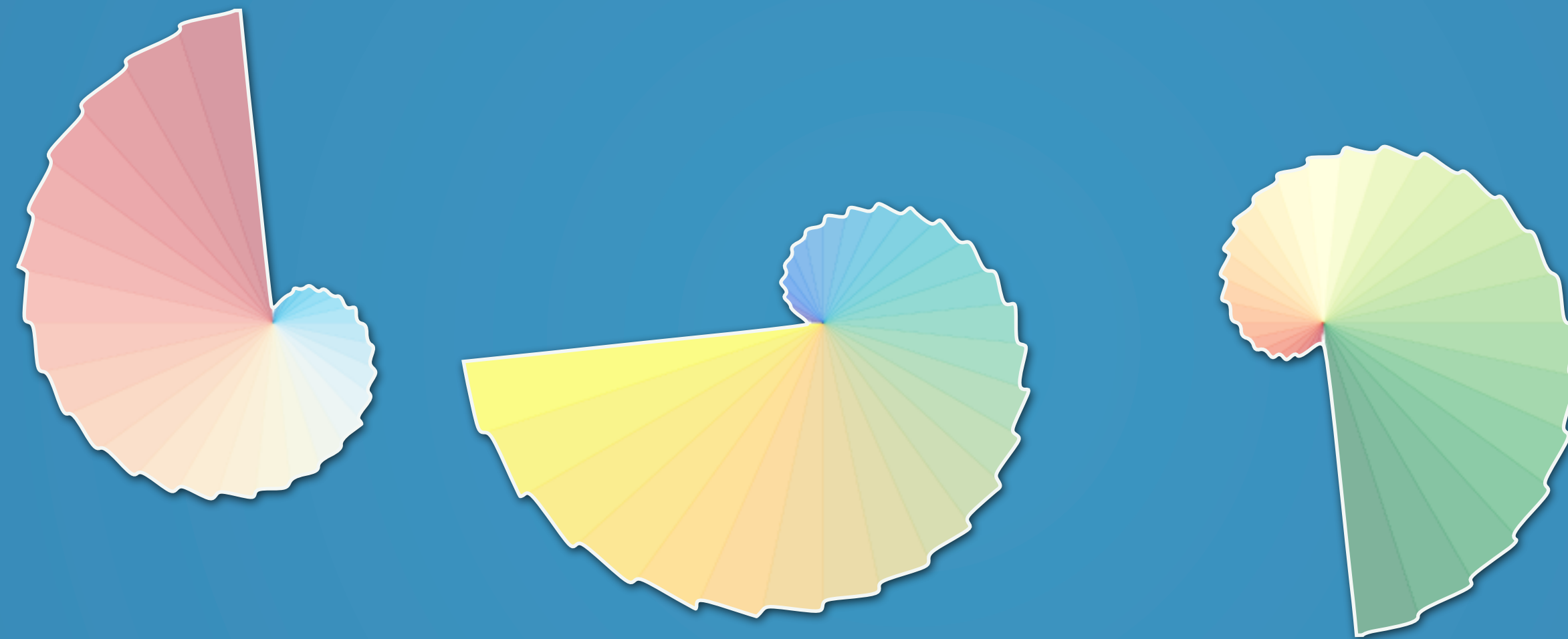
`int valor = 'Tres';`

TIPADO DINÁMICO

→ `valor = 3;`



`valor = 'Tres';`



Operadores



(suma) +



(suma)

+

result = 2 + pi;



(suma)

+

result = 2 + pi;

result



(suma)

+

result = 2 + pi;

result → ans = 5.1416

→

→



Existen valores predefinidos como π

(suma)

+

result = 2 + pi;

result → ans = 5.1416





Existen valores predefinidos como π

(suma)

+

result = 2 + pi;

result → ans = 5.1416

(resta)

-





Existen valores predefinidos como π

(suma)

+

result = 2 + pi;

result → ans = 5.1416

(resta)

-

result = result - 2;





Existen valores predefinidos como π

(suma)

+

result = 2 + pi;

result → ans = 5.1416

(resta)

-

result = result - 2;

result →

→



Existen valores predefinidos como π

(suma)

+

result = 2 + pi;

result → ans = 5.1416

(resta)

-

result = result - 2;

result → ans = 3.1416





Existen valores predefinidos como π

(suma) +

```
result = 2 + pi;
```

result → ans = 5.1416

(resta) -

```
result = result - 2;
```

result → ans = 3.1416

(producto) *





Existen valores predefinidos como π

(suma) +

```
result = 2 + pi;
```

result → ans = 5.1416

(resta) -

```
result = result - 2;
```

result → ans = 3.1416

(producto) *

```
length = 2 * result * r;
```





Existen valores predefinidos como π

(suma) **+**

```
result = 2 + pi;
```

result  ans = **5.1416**

(resta) **-**

```
result = result - 2;
```

result  ans = **3.1416**

(producto) *****

```
length = 2 * result * r;
```

length 



Existen valores predefinidos como π

(suma)	+	<code>result = 2 + pi;</code>	
		<code>result</code>	<code>ans = 5.1416</code>
(resta)	-	<code>result = result - 2;</code>	
		<code>result</code>	<code>ans = 3.1416</code>
(producto)	*	<code>length = 2 * result * r;</code>	
		<code>length</code>	<code>ans = 6.2832</code>



(cociente)



(cociente)



```
result = 2 / 0;
```



(cociente)



result = 2 / 0;



result



(cociente)



result = 2 / 0;

result  ans = **Inf**



En Matlab está definido el Inf.



(cociente)



result = 2 / 0;

result  ans = **Inf**



En Matlab está definido el Inf.



(cociente)



result = 2 / 0;

result  ans = **Inf**

(exponente)



En Matlab está definido el Inf.



(cociente)



result = 2 / 0;

result  ans = **Inf**

(exponente)



result = 2 ^ 8;



En Matlab está definido el Inf.



(cociente)



```
result = 2 / 0;
```

```
result → ans = Inf
```

(exponente)



```
result = 2 ^ 8;
```

```
result →
```

En Matlab está definido el Inf.



(cociente)



```
result = 2 / 0;
```

```
result → ans = Inf
```



(exponente)



```
result = 2 ^ 8;
```

```
result → ans = 256
```

En Matlab está definido el Inf.



(cociente)



```
result = 2 / 0;
```

```
result → ans = Inf
```



(exponente)



```
result = 2 ^ 8;
```

```
result → ans = 256
```

(orden)



En Matlab está definido el Inf.



(cociente)



```
result = 2 / 0;
```

```
result → ans = Inf
```



(exponente)



```
result = 2 ^ 8;
```

```
result → ans = 256
```

(orden)



```
result = 2 ^ (8 - 3) - pi * 2;
```



(mayor que)



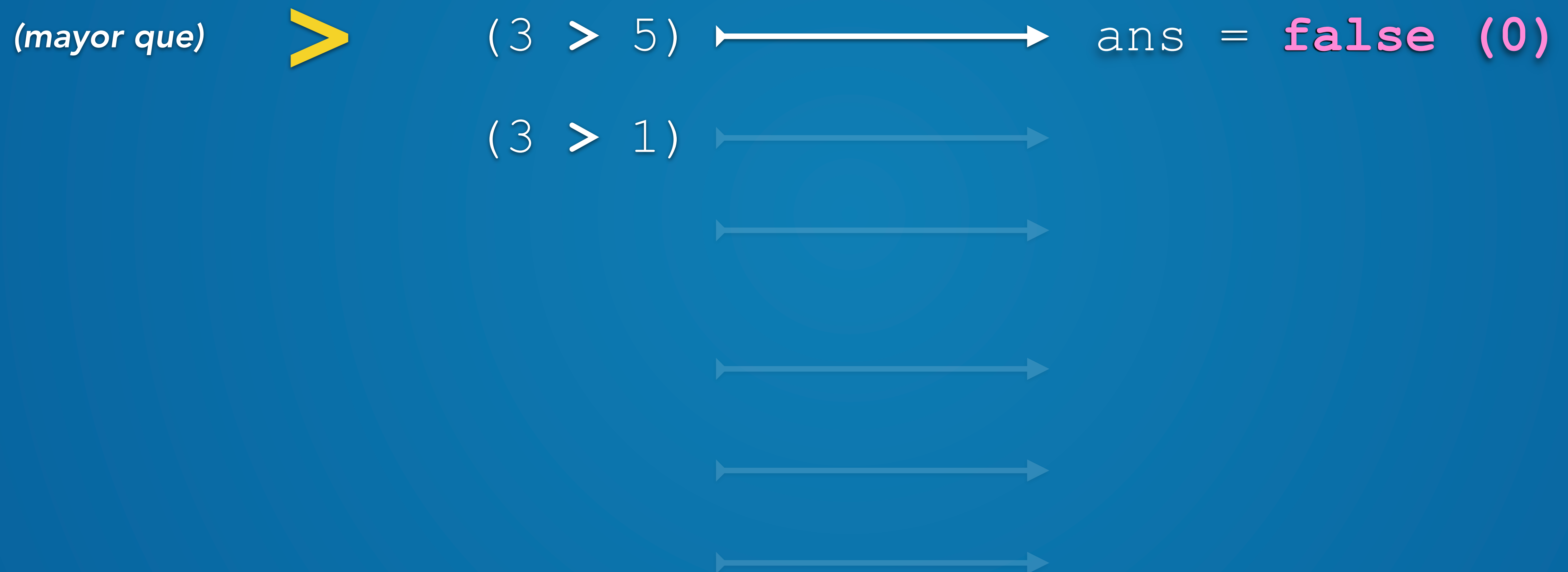
(mayor que)

>

(3 > 5)

(mayor que)  (3 > 5)  ans = **false (0)**





(*mayor que*)



(3 > 5)



ans = **false** (0)

(3 > 1)



ans = **true** (1)



(mayor que)



(3 > 5)



ans = **false (0)**

(3 > 1)



ans = **true (1)**

(3 > 3)



(mayor que)



(3 > 5) → ans = false (0)

(3 > 1) → ans = true (1)

(3 > 3) → ans = false (0)



(mayor que)



(3 > 5) → ans = false (0)

(3 > 1) → ans = true (1)

(3 > 3) → ans = false (0)

(menor que)



(mayor que)



(3 > 5)



ans = **false (0)**

(3 > 1)



ans = **true (1)**

(3 > 3)



ans = **false (0)**

(menor que)



(3 < 5)



(mayor que)



(3 > 5)



ans = **false (0)**

(3 > 1)



ans = **true (1)**

(3 > 3)



ans = **false (0)**

(menor que)



(3 < 5)



ans = **true (1)**



(mayor que)



(3 > 5) → ans = false (0)

(3 > 1) → ans = true (1)

(3 > 3) → ans = false (0)

(menor que)



(3 < 5) → ans = true (1)

(3 < 1) →



(mayor que)



(3 > 5) → ans = **false (0)**

(3 > 1) → ans = **true (1)**

(3 > 3) → ans = **false (0)**

(menor que)



(3 < 5) → ans = **true (1)**

(3 < 1) → ans = **false (0)**



(mayor que)



(3 > 5) → ans = false (0)

(3 > 1) → ans = true (1)

(3 > 3) → ans = false (0)

(menor que)



(3 < 5) → ans = true (1)

(3 < 1) → ans = false (0)

(3 < 3) →

(mayor que)



(3 > 5) → ans = false (0)

(3 > 1) → ans = true (1)

(3 > 3) → ans = false (0)

(menor que)



(3 < 5) → ans = true (1)

(3 < 1) → ans = false (0)

(3 < 3) → ans = false (0)



(mayor o igual) **>=**



(mayor o igual) **>=** (3 >= 5) 



(*mayor o igual*) **>=** (3 >= 5)  ans = **false (0)**



(*mayor o igual*) **>=** (3 >= 5)  ans = **false (0)**

(3 >= 1) 









(*mayor o igual*) **>=** (3 >= 5)  ans = **false (0)**

(3 >= 1)  ans = **true (1)**



(*mayor o igual*) **>=** (3 >= 5)  ans = **false (0)**

(3 >= 1)  ans = **true (1)**

(3 >= 3) 







(*mayor o igual*) **>=** (3 >= 5)  ans = **false (0)**

(3 >= 1)  ans = **true (1)**

(3 >= 3)  ans = **true (1)**



(mayor o igual) **>=** (3 >= 5) → ans = **false (0)**

(3 >= 1) → ans = **true (1)**

(3 >= 3) → ans = **true (1)**

(menor o igual) **<=**

→

→

→

(mayor o igual) **>=** (3 >= 5)  ans = **false (0)**

(3 >= 1)  ans = **true (1)**

(3 >= 3)  ans = **true (1)**

(menor o igual) **<=** (3 <= 5) 

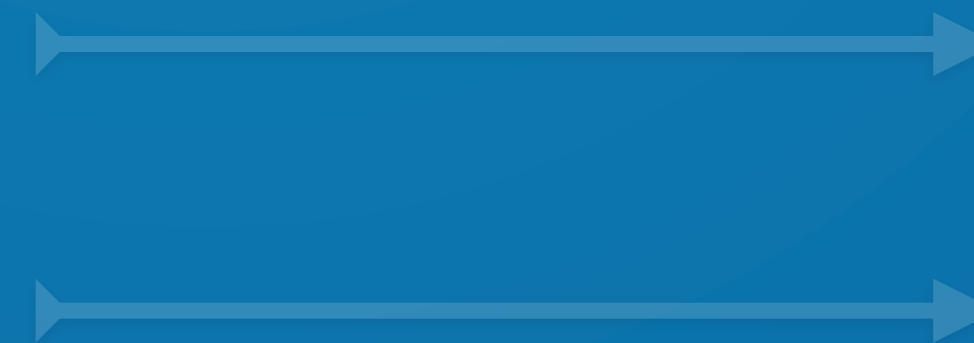


(mayor o igual) **>=** (3 >= 5) → ans = **false (0)**

(3 >= 1) → ans = **true (1)**

(3 >= 3) → ans = **true (1)**

(menor o igual) **<=** (3 <= 5) → ans = **true (1)**



(mayor o igual) **>=** (3 >= 5) → ans = **false (0)**

(3 >= 1) → ans = **true (1)**

(3 >= 3) → ans = **true (1)**

(menor o igual) **<=** (3 <= 5) → ans = **true (1)**

(3 <= 1) →

→

(mayor o igual) **>=** (3 >= 5)  ans = **false (0)**

(3 >= 1)  ans = **true (1)**

(3 >= 3)  ans = **true (1)**

(menor o igual) **<=** (3 <= 5)  ans = **true (1)**

(3 <= 1)  ans = **false (0)**



(mayor o igual) **>=** (3 >= 5) → ans = **false (0)**

(3 >= 1) → ans = **true (1)**

(3 >= 3) → ans = **true (1)**

(menor o igual) **<=** (3 <= 5) → ans = **true (1)**

(3 <= 1) → ans = **false (0)**

(3 <= 3) →

(mayor o igual) **>=** (3 >= 5)  ans = **false (0)**

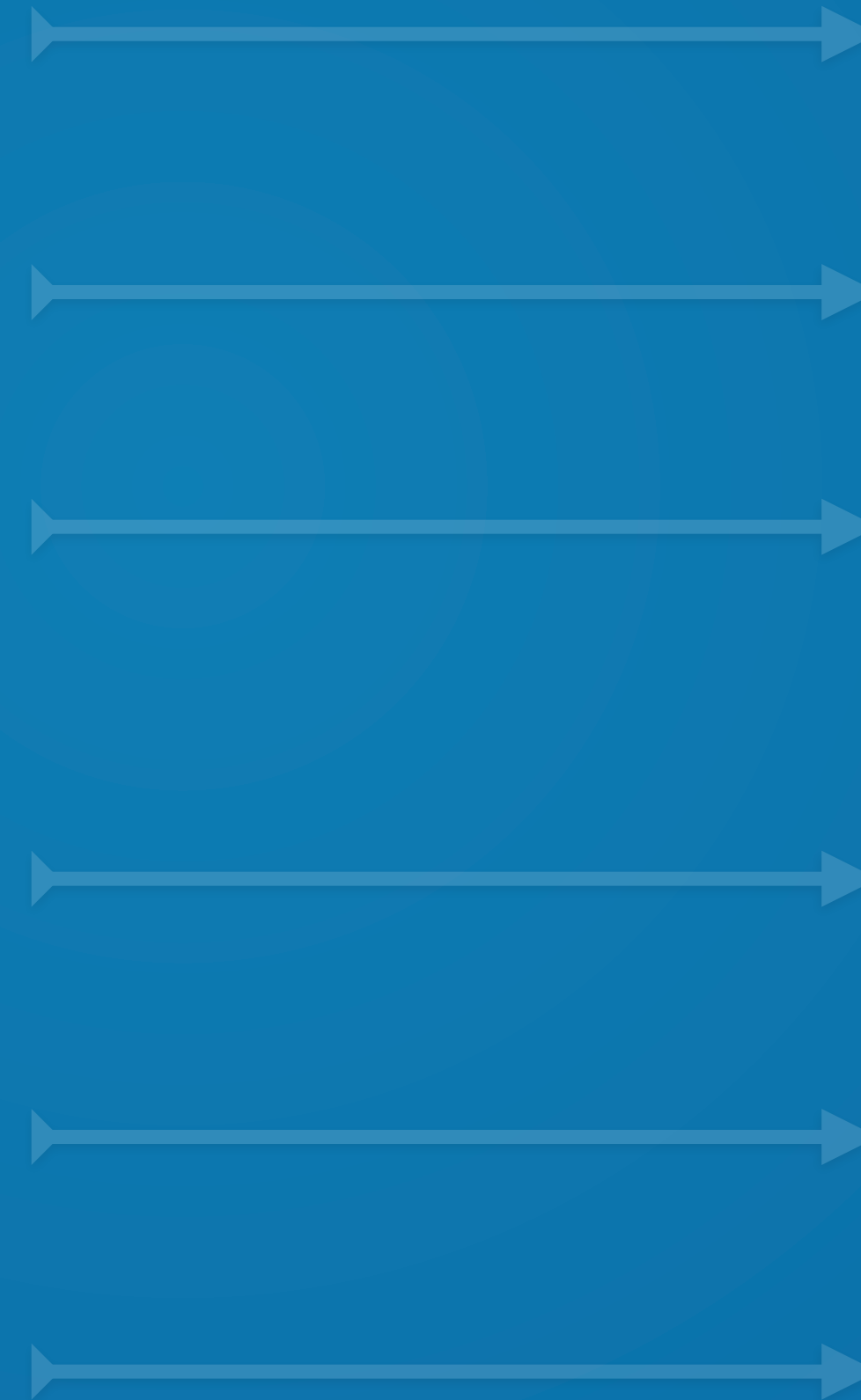
(3 >= 1)  ans = **true (1)**

(3 >= 3)  ans = **true (1)**

(menor o igual) **<=** (3 <= 5)  ans = **true (1)**

(3 <= 1)  ans = **false (0)**

(3 <= 3)  ans = **true (1)**



(igual que)



(igual que)



(3 == 5)



(igual que) **==** (3 == 5)  ans = **false** (0)



(igual que)

==

(3 == 5)  ans = **false (0)**

(3 == 1) 









(igual que)

==

(3 == 5)  ans = **false (0)**

(3 == 1)  ans = **false (0)**









(igual que)

==

(3 == 5)  ans = **false (0)**

(3 == 1)  ans = **false (0)**

(3 == 3) 







(igual que)

==

(3 == 5)  ans = **false (0)**

(3 == 1)  ans = **false (0)**

(3 == 3)  ans = **true (1)**







(igual que)

==

(3 == 5) → ans = **false** (0)

(3 == 1) → ans = **false** (0)

(3 == 3) → ans = **true** (1)

(distinto a)

~

→

→

→

(igual que)

==

(3 == 5) → ans = **false** (0)

(3 == 1) → ans = **false** (0)

(3 == 3) → ans = **true** (1)

(distinto a)

~=

(3 ~= 5) →

→

→

(igual que)

==

(3 == 5) → ans = **false (0)**

(3 == 1) → ans = **false (0)**

(3 == 3) → ans = **true (1)**

(distinto a)

~=

(3 ~= 5) → ans = **true (1)**

→

→

(igual que)

==

(3 == 5) → ans = **false (0)**

(3 == 1) → ans = **false (0)**

(3 == 3) → ans = **true (1)**

(distinto a)

~=

(3 ~= 5) → ans = **true (1)**

(3 ~= 1) →

→

(igual que)

==

(3 == 5) → ans = **false** (0)

(3 == 1) → ans = **false** (0)

(3 == 3) → ans = **true** (1)

(distinto a)

~=

(3 ~= 5) → ans = **true** (1)

(3 ~= 1) → ans = **true** (1)



(igual que)

==

(3 == 5) → ans = **false** (0)

(3 == 1) → ans = **false** (0)

(3 == 3) → ans = **true** (1)

(distinto a)

~=

(3 ~= 5) → ans = **true** (1)

(3 ~= 1) → ans = **true** (1)

(3 ~= 3) →

(igual que)

==

(3 == 5) → ans = **false (0)**

(3 == 1) → ans = **false (0)**

(3 == 3) → ans = **true (1)**

(distinto a)

~=

(3 ~= 5) → ans = **true (1)**

(3 ~= 1) → ans = **true (1)**

(3 ~= 3) → ans = **false (0)**



3

3

3

-

1

2

3

=

2

1

0

3	3	3	-	1	2	3	=	2	1	0
---	---	---	---	---	---	---	---	---	---	---

1	2	3	+	3	2	1	=	4	4	4
2	1	2		2	3	2		4	4	4
3	2	1		1	2	3		4	4	4

¡Deben de ser del mismo tamaño!



3	3	3	-	1	2	3	=	2	1	0
---	---	---	---	---	---	---	---	---	---	---

1	2	3	+	3	2	1	=	4	4	4
2	1	2		2	3	2		4	4	4
3	2	1		1	2	3		4	4	4



$$\begin{bmatrix} 3 & 3 & 3 \end{bmatrix} * \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} =$$



¡Cuidado con la multiplicación de matrices!

3 3 3

*

1 2 3

=

error



¡Cuidado con la multiplicación de matrices!

(multiplicación algebraica)
 $[1 \times 3] \cdot [1 \times 3]$

3	3	3
---	---	---

*

1	2	3
---	---	---

=

error



¡Cuidado con la multiplicación de matrices!

(multiplicación algebraica)
 $[1 \times 3] \cdot [1 \times 3]$

$$\begin{bmatrix} 3 & 3 & 3 \end{bmatrix}$$

*

$$\begin{bmatrix} 1 & 2 & 3 \end{bmatrix}$$

=

error

$$\begin{bmatrix} 3 & 3 & 3 \end{bmatrix}$$

*

$$\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

=



¡Cuidado con la multiplicación de matrices!

(multiplicación algebraica)
 $[1 \times 3] \cdot [1 \times 3]$



*



=

error

(multiplicación algebraica)
 $[1 \times 3] \cdot [3 \times 1]$



*



=

18





<table><tr><td>3</td><td>3</td><td>3</td></tr></table>	3	3	3	\cdot *	<table><tr><td>1</td><td>2</td><td>3</td></tr></table>	1	2	3	=	<table><tr><td>3</td><td>6</td><td>9</td></tr></table>	3	6	9
3	3	3											
1	2	3											
3	6	9											

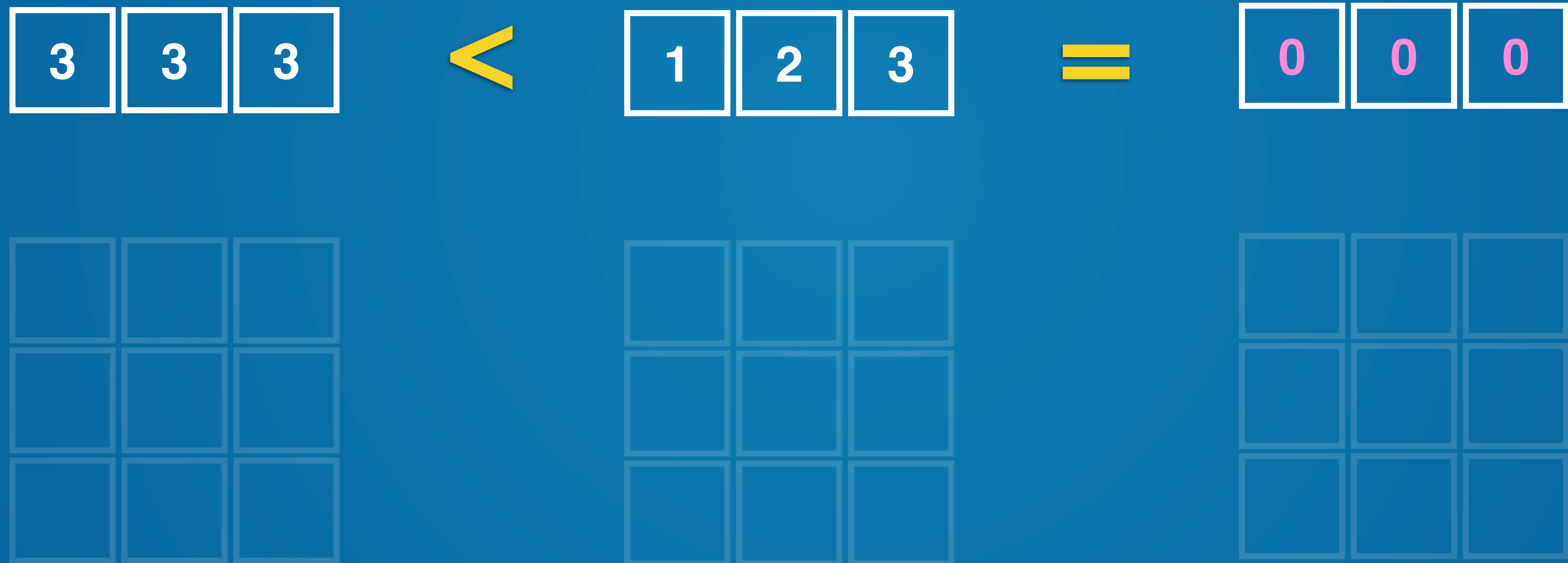
<table><tr><td>1</td><td>2</td><td>3</td></tr><tr><td>2</td><td>1</td><td>2</td></tr><tr><td>3</td><td>2</td><td>1</td></tr></table>	1	2	3	2	1	2	3	2	1	\cdot *	<table><tr><td>3</td><td>2</td><td>1</td></tr><tr><td>2</td><td>3</td><td>2</td></tr><tr><td>1</td><td>2</td><td>3</td></tr></table>	3	2	1	2	3	2	1	2	3	=	<table><tr><td>3</td><td>4</td><td>3</td></tr><tr><td>4</td><td>3</td><td>4</td></tr><tr><td>3</td><td>4</td><td>3</td></tr></table>	3	4	3	4	3	4	3	4	3
1	2	3																													
2	1	2																													
3	2	1																													
3	2	1																													
2	3	2																													
1	2	3																													
3	4	3																													
4	3	4																													
3	4	3																													

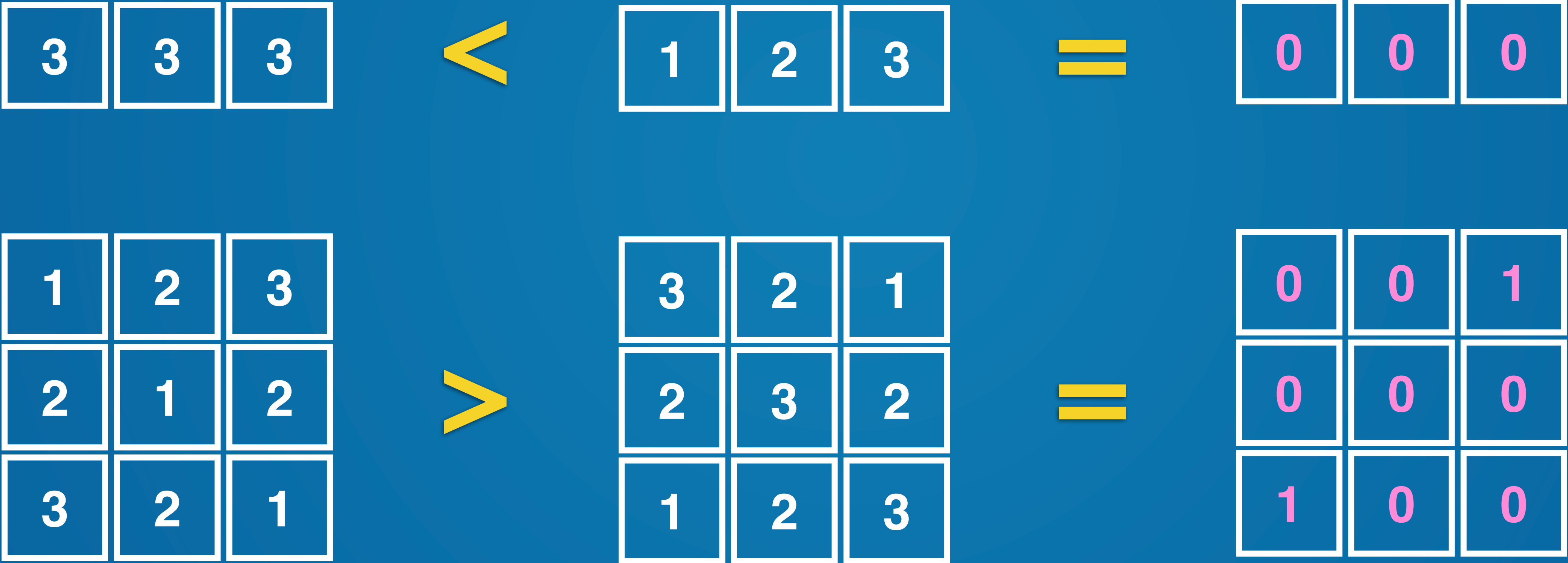
! .* multiplicación elemento a elemento

3	3	3	.*	1	2	3	=	3	6	9
---	---	---	----	---	---	---	---	---	---	---

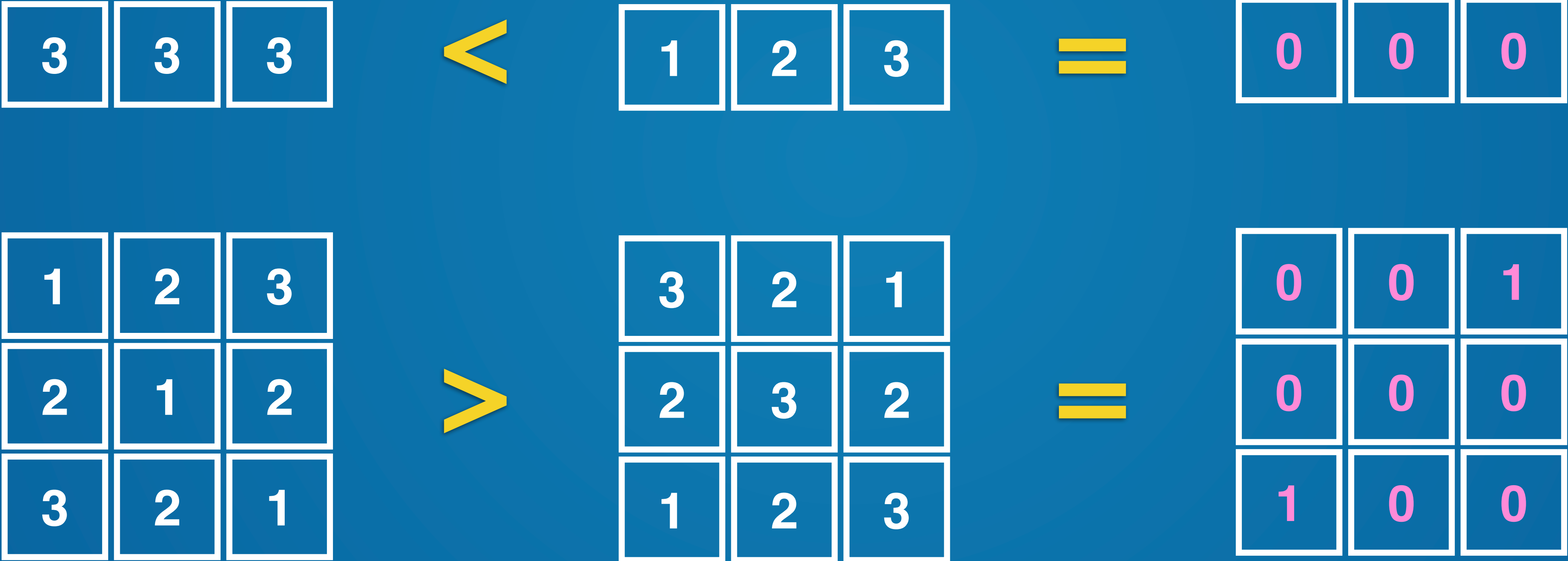
1	2	3	.*	3	2	1	=	3	4	3
2	1	2		2	3	2		4	3	4
3	2	1		1	2	3		3	4	3

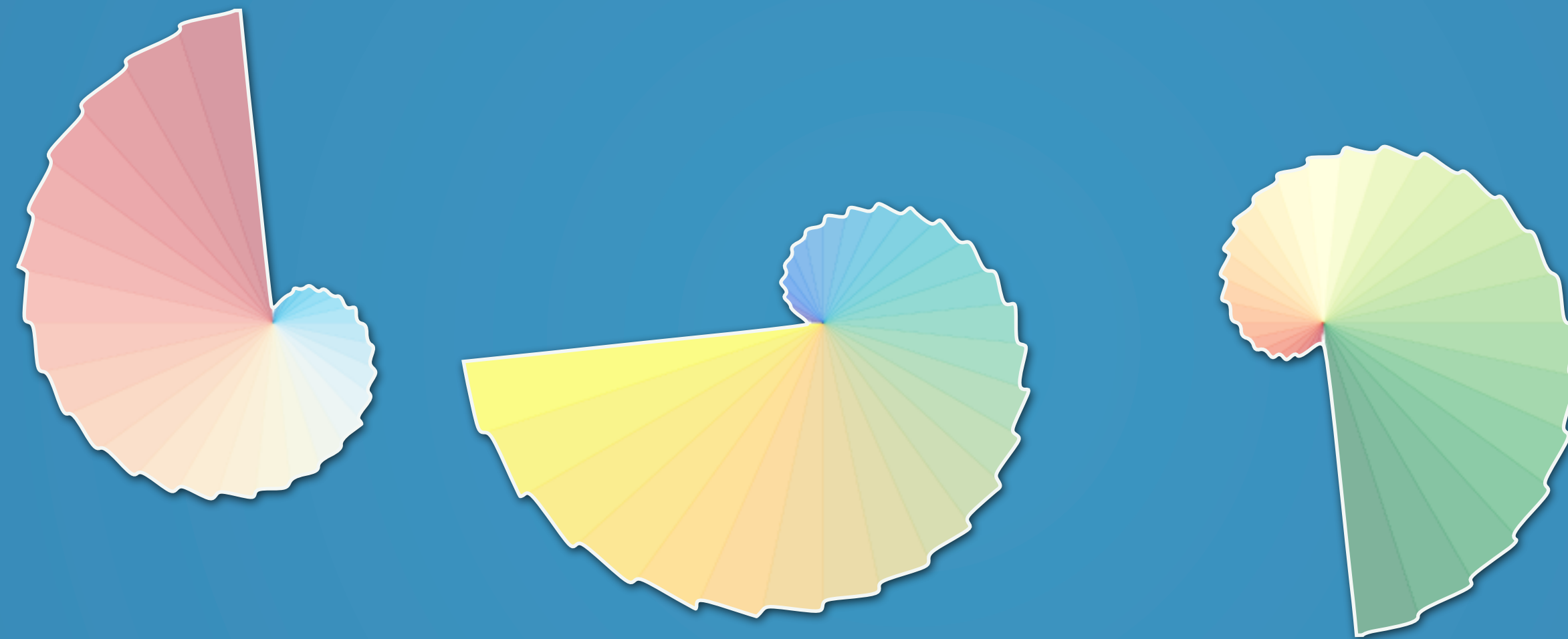






¡Deben de ser del mismo tamaño!





Funciones

¿Qué es una función?

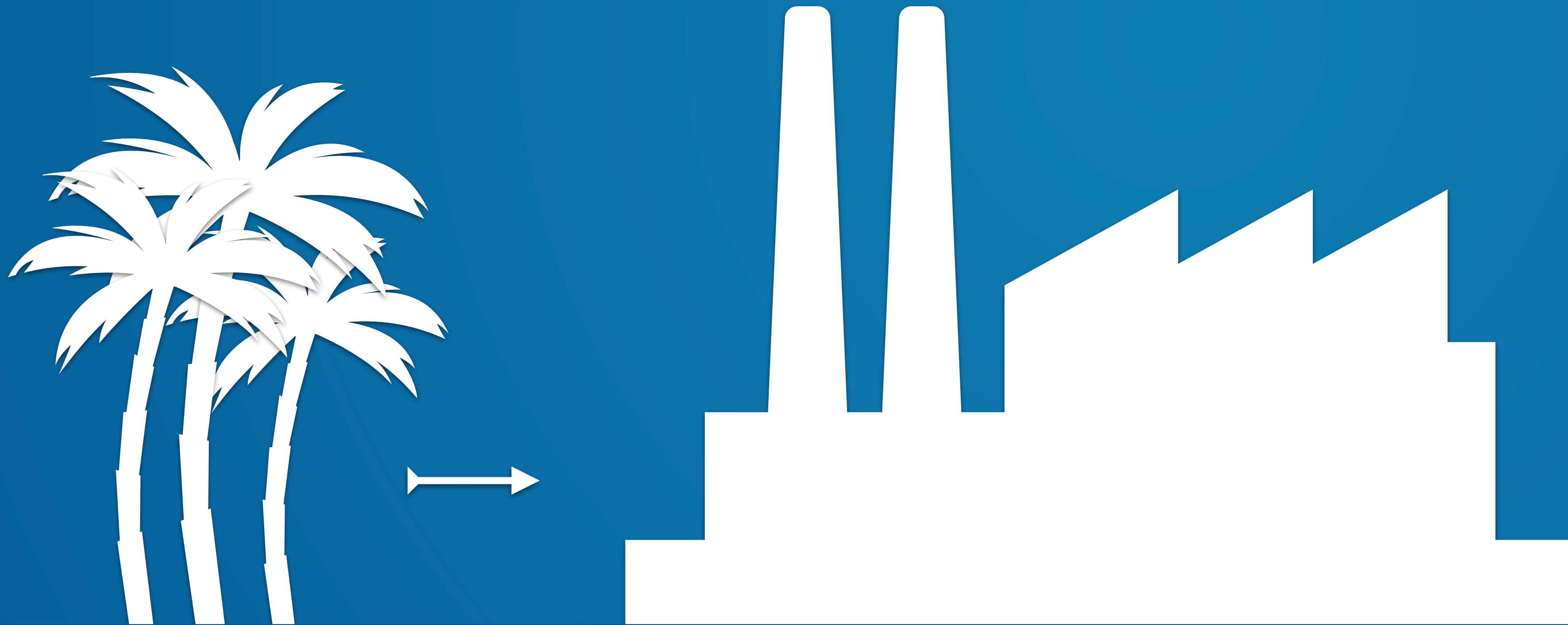
¿Qué es una función?



¿Qué es una función?



¿Qué es una función?



¿Qué es una función?



¿Qué es una función?

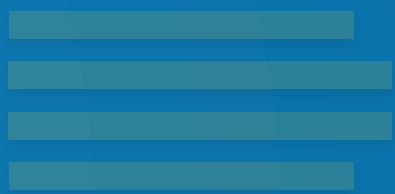


1. Definición de una función
2. Tipos de funciones
3. Funciones lineales
4. Funciones cuadráticas
5. Funciones racionales
6. Funciones irracionales
7. Funciones trascendentes
8. Funciones trigonométricas
9. Funciones exponenciales
10. Funciones logarítmicas
11. Funciones hiperbólicas
12. Funciones inversas
13. Funciones compuestas
14. Funciones implícitas
15. Funciones paramétricas
16. Funciones vectoriales
17. Funciones matriciales
18. Funciones de varias variables
19. Funciones de valor vectorial
20. Funciones de valor matricial
21. Funciones de valor escalar
22. Funciones de valor vectorial
23. Funciones de valor matricial
24. Funciones de valor escalar
25. Funciones de valor vectorial
26. Funciones de valor matricial
27. Funciones de valor escalar
28. Funciones de valor vectorial
29. Funciones de valor matricial
30. Funciones de valor escalar

1. Definición de una función
2. Tipos de funciones
3. Funciones lineales
4. Funciones cuadráticas
5. Funciones racionales
6. Funciones irracionales
7. Funciones trascendentes
8. Funciones trigonométricas
9. Funciones exponenciales
10. Funciones logarítmicas
11. Funciones hiperbólicas
12. Funciones inversas
13. Funciones compuestas
14. Funciones implícitas
15. Funciones paramétricas
16. Funciones vectoriales
17. Funciones matriciales
18. Funciones de varias variables
19. Funciones de valor vectorial
20. Funciones de valor matricial
21. Funciones de valor escalar
22. Funciones de valor vectorial
23. Funciones de valor matricial
24. Funciones de valor escalar
25. Funciones de valor vectorial
26. Funciones de valor matricial
27. Funciones de valor escalar
28. Funciones de valor vectorial
29. Funciones de valor matricial
30. Funciones de valor escalar

1. Definición de una función
2. Tipos de funciones
3. Funciones lineales
4. Funciones cuadráticas
5. Funciones racionales
6. Funciones irracionales
7. Funciones trascendentes
8. Funciones trigonométricas
9. Funciones exponenciales
10. Funciones logarítmicas
11. Funciones hiperbólicas
12. Funciones inversas
13. Funciones compuestas
14. Funciones implícitas
15. Funciones paramétricas
16. Funciones vectoriales
17. Funciones matriciales
18. Funciones de varias variables
19. Funciones de valor vectorial
20. Funciones de valor matricial
21. Funciones de valor escalar
22. Funciones de valor vectorial
23. Funciones de valor matricial
24. Funciones de valor escalar
25. Funciones de valor vectorial
26. Funciones de valor matricial
27. Funciones de valor escalar
28. Funciones de valor vectorial
29. Funciones de valor matricial
30. Funciones de valor escalar

PROGRAMA PRINCIPAL:



PROGRAMA PRINCIPAL:



FUNCIÓN EXTERNA:



PROGRAMA PRINCIPAL:



PROGRAMA PRINCIPAL:



FUNCIÓN EXTERNA:







CONJUNTO DE LÍNEAS DE CODIGO CON UNA FINALIDAD ESPECÍFICA





CONJUNTO DE LÍNEAS DE CODIGO CON UNA FINALIDAD ESPECÍFICA



RECIBE DATOS DE ENTRADA Y LOS DEVUELVE MODIFICADOS





CONJUNTO DE LÍNEAS DE CODIGO CON UNA FINALIDAD ESPECÍFICA



RECIBE DATOS DE ENTRADA Y LOS DEVUELVE MODIFICADOS



MODULAR EL CÓDIGO POR FUNCIONALIDAD





CONJUNTO DE LÍNEAS DE CODIGO CON UNA FINALIDAD ESPECÍFICA



RECIBE DATOS DE ENTRADA Y LOS DEVUELVE MODIFICADOS



MODULAR EL CÓDIGO POR FUNCIONALIDAD



REUTILIZAR EL CÓDIGO Y EVITAR ERRORES



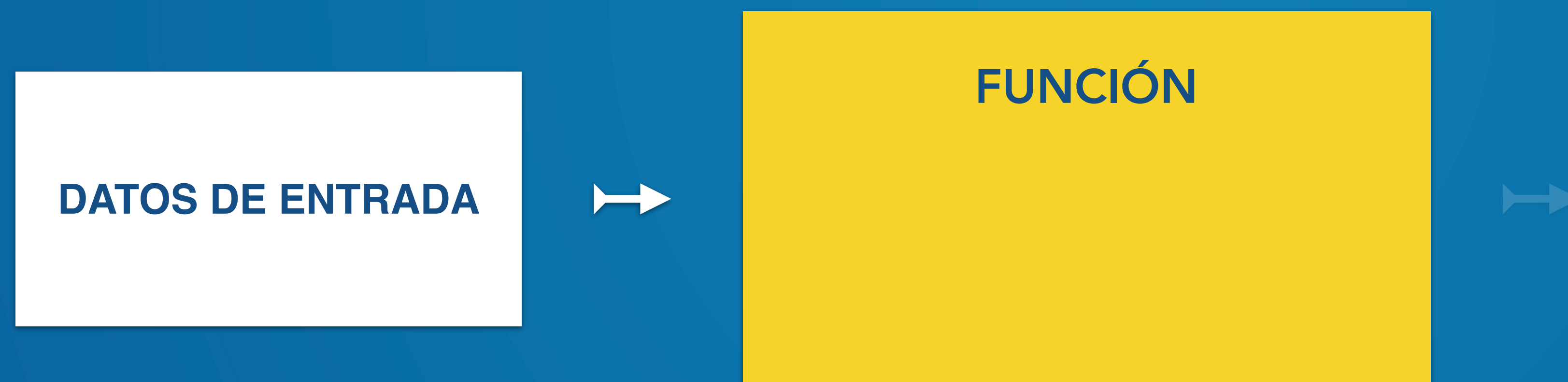
¿Qué es una función?



¿Qué es una función?



¿Qué es una función?



¿Qué es una función?



¿Qué es una función?





¿Qué es una función?

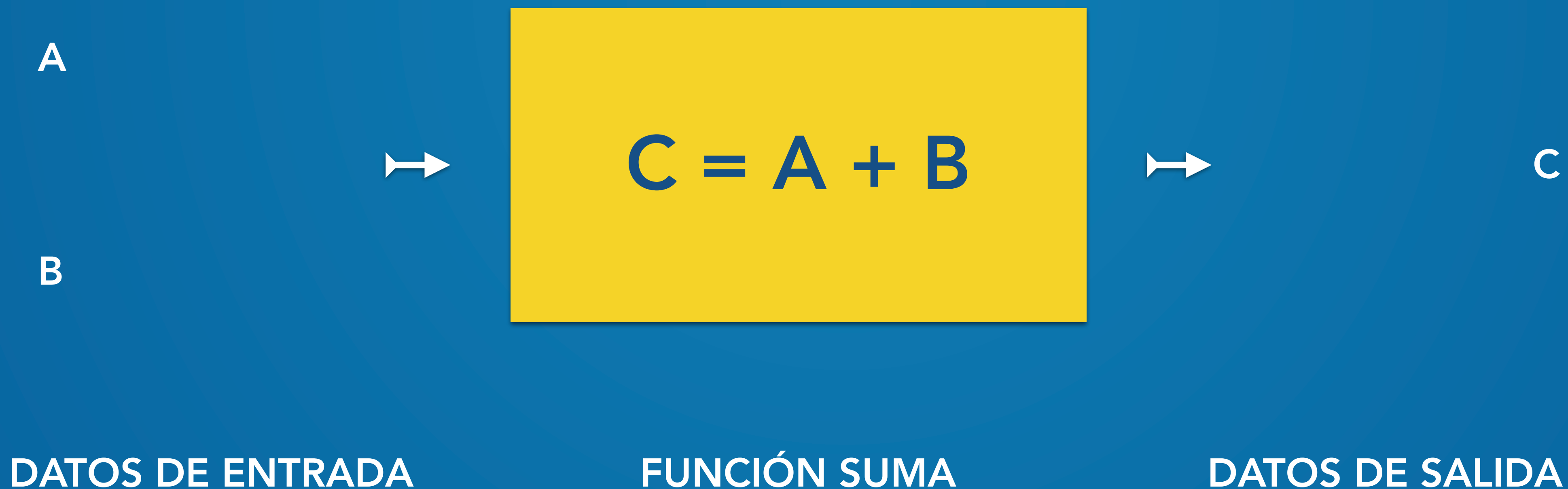


¿Qué es una función?

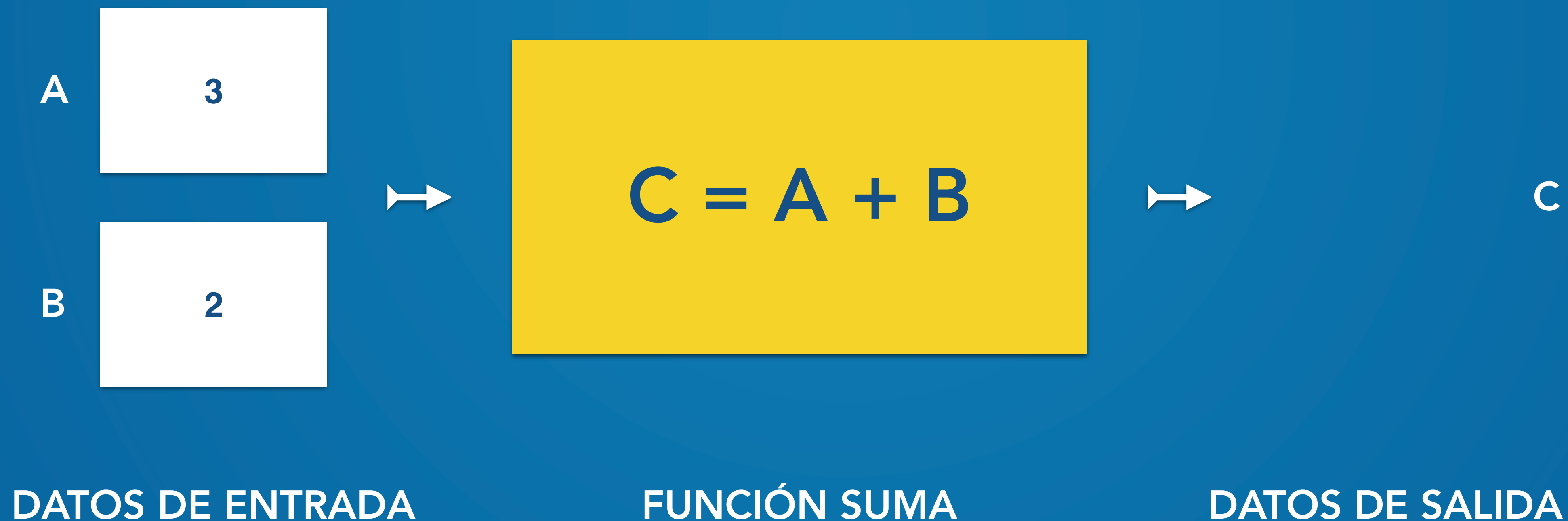


FUNCIÓN SUMA

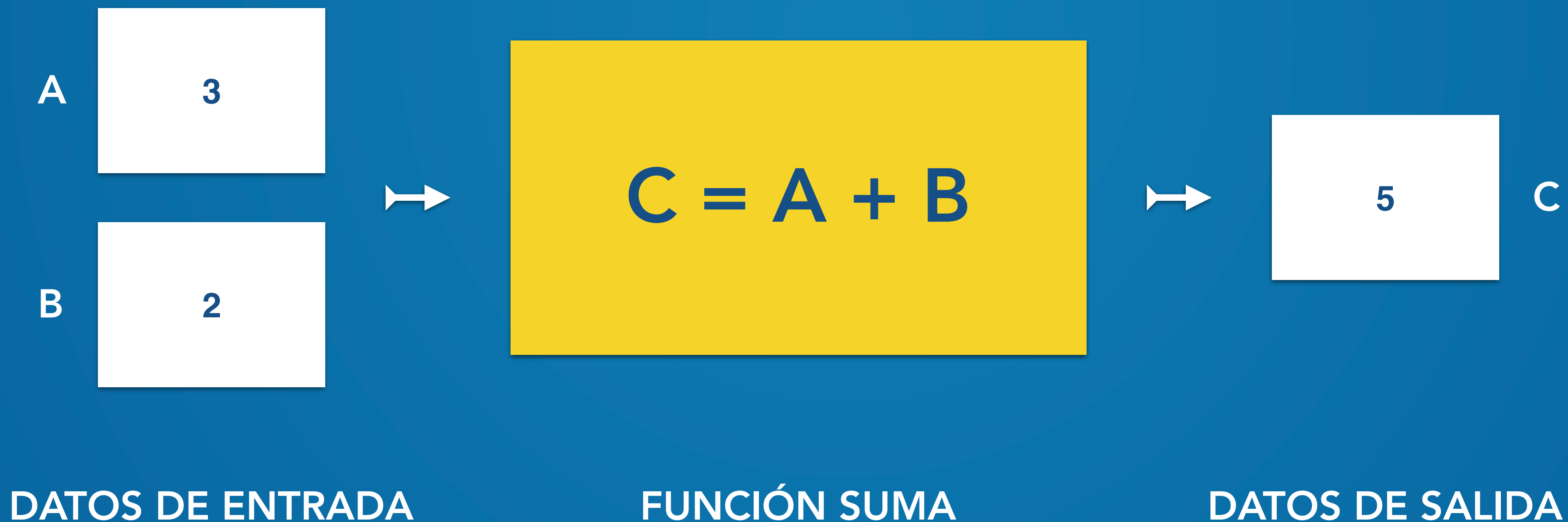
¿Qué es una función?



¿Qué es una función?



¿Qué es una función?



(function) **DEFINICIÓN:** archivo nuevo con el mismo nombre de la función — **suma.m**



(function) **DEFINICIÓN:** archivo nuevo con el mismo nombre de la función — **suma.m**

function

(function) DEFINICIÓN: archivo nuevo con el mismo nombre de la función — `suma.m`

Dato de salida



`function [resultado]`

(function) DEFINICIÓN: archivo nuevo con el mismo nombre de la función — `suma.m`

Dato de salida



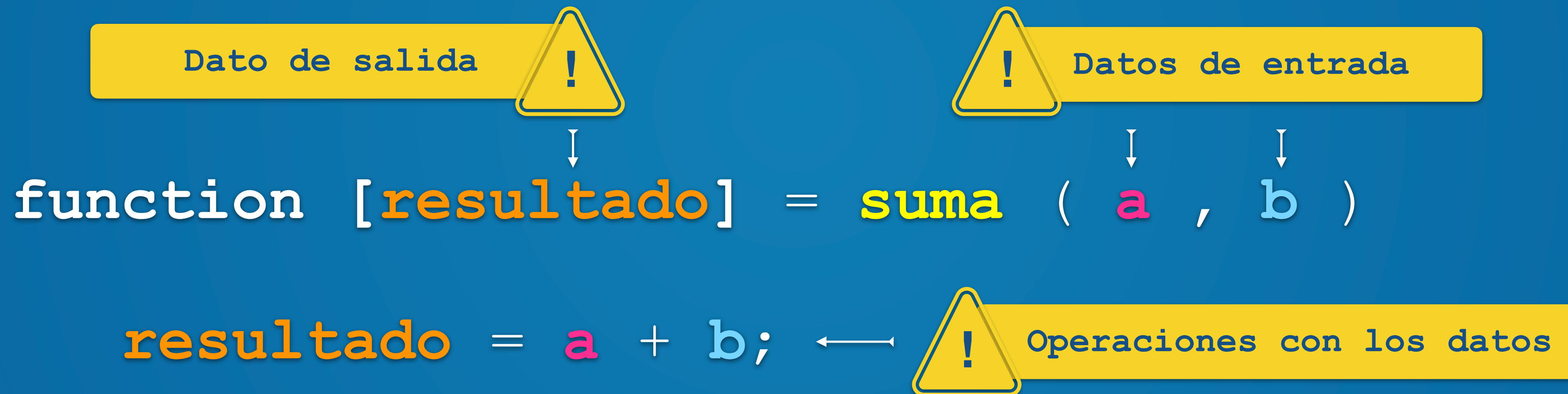
```
function [resultado] = suma
```


(function) DEFINICIÓN: archivo nuevo con el mismo nombre de la función — **suma.m**

The diagram illustrates the syntax of a function definition in MATLAB. It features a code snippet: `function [resultado] = suma (a , b)`. Above the opening square bracket of `[resultado]` is a yellow box containing the text "Dato de salida" and a warning triangle icon. A vertical arrow points from this box to the bracket. Above the opening parenthesis of `(a , b)` is a yellow box containing the text "Datos de entrada" and a warning triangle icon. Two vertical arrows point from this box to the variables `a` and `b` inside the parentheses. The variable `resultado` is highlighted in orange, `suma` is in yellow, `a` is in pink, and `b` is in blue.

```
function [resultado] = suma ( a , b )
```

(*function*) DEFINICIÓN: archivo nuevo con el mismo nombre de la función — **suma.m**



(*function*) DEFINICIÓN: archivo nuevo con el mismo nombre de la función — **suma.m**

The diagram illustrates the structure of a MATLAB function definition. It shows the code `function [resultado] = suma (a , b)` and `resultado = a + b;` with annotations. A yellow box labeled "Dato de salida" with a warning icon points to `[resultado]`. A yellow box labeled "Datos de entrada" with a warning icon points to `a` and `b`. A yellow box labeled "Operaciones con los datos" with a warning icon points to the assignment statement `resultado = a + b;`. The code is enclosed in a vertical line on the left, with `end` at the bottom.

```
function [resultado] = suma ( a , b )  
  
    resultado = a + b;  
  
end
```

(function) USOS:



(function) USOS:

```
respuesta = suma ( 1 , 2 );
```



(function) USOS:

```
respuesta = suma ( 1 , 2 );
```

```
respuesta
```



(function) USOS:

```
respuesta = suma ( 1 , 2 );
```

```
respuesta  ans = 3
```



(function) USOS:

```
respuesta = suma ( 1 , 2 );
```

```
respuesta  ans = 3
```

```
respuesta = 3 + suma ( 1 , 2 );
```



(function) USOS:

```
respuesta = suma ( 1 , 2 );
```

```
respuesta  ans = 3
```

```
respuesta = 3 + suma ( 1 , 2 );
```

```
respuesta 
```

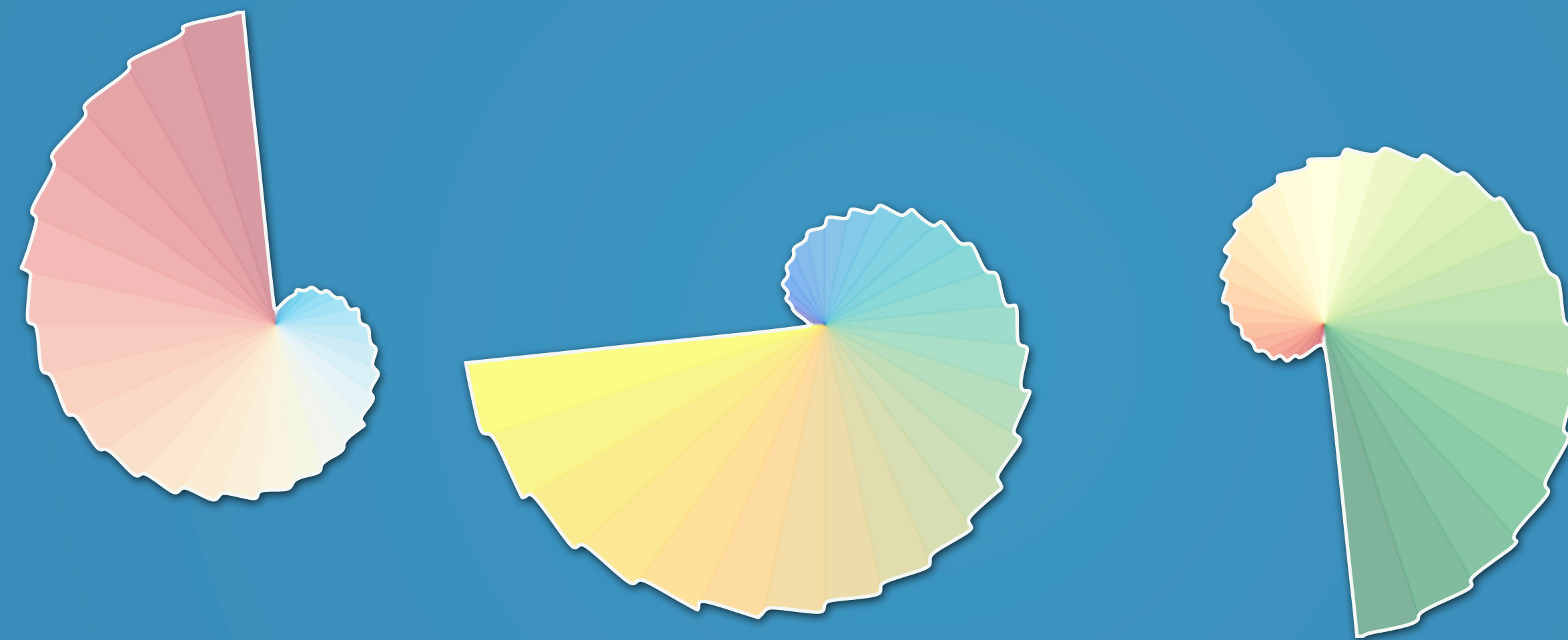
(function) USOS:

```
respuesta = suma ( 1 , 2 );
```

```
respuesta  ans = 3
```

```
respuesta = 3 + suma ( 1 , 2 );
```

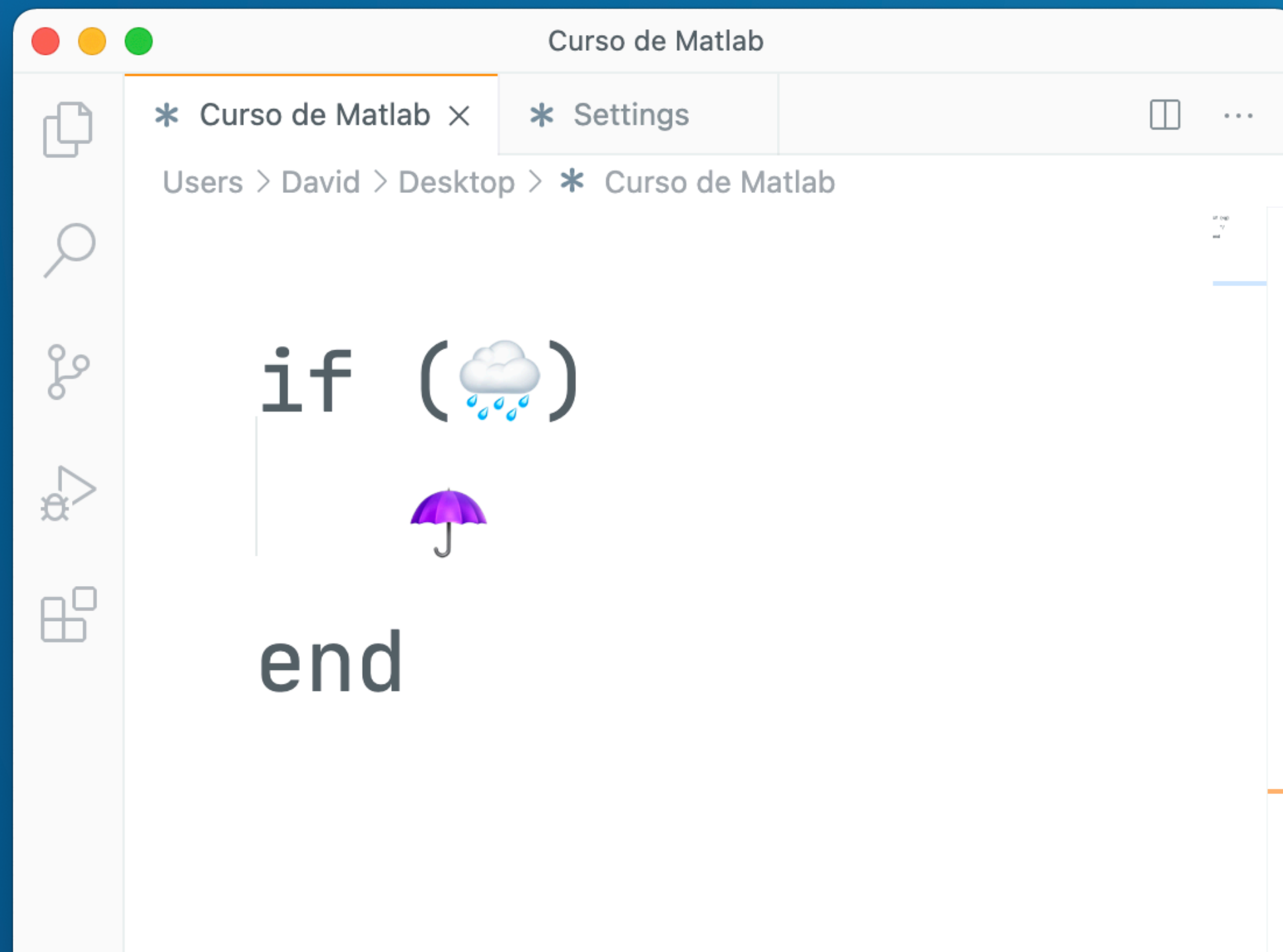
```
respuesta  ans = 6
```

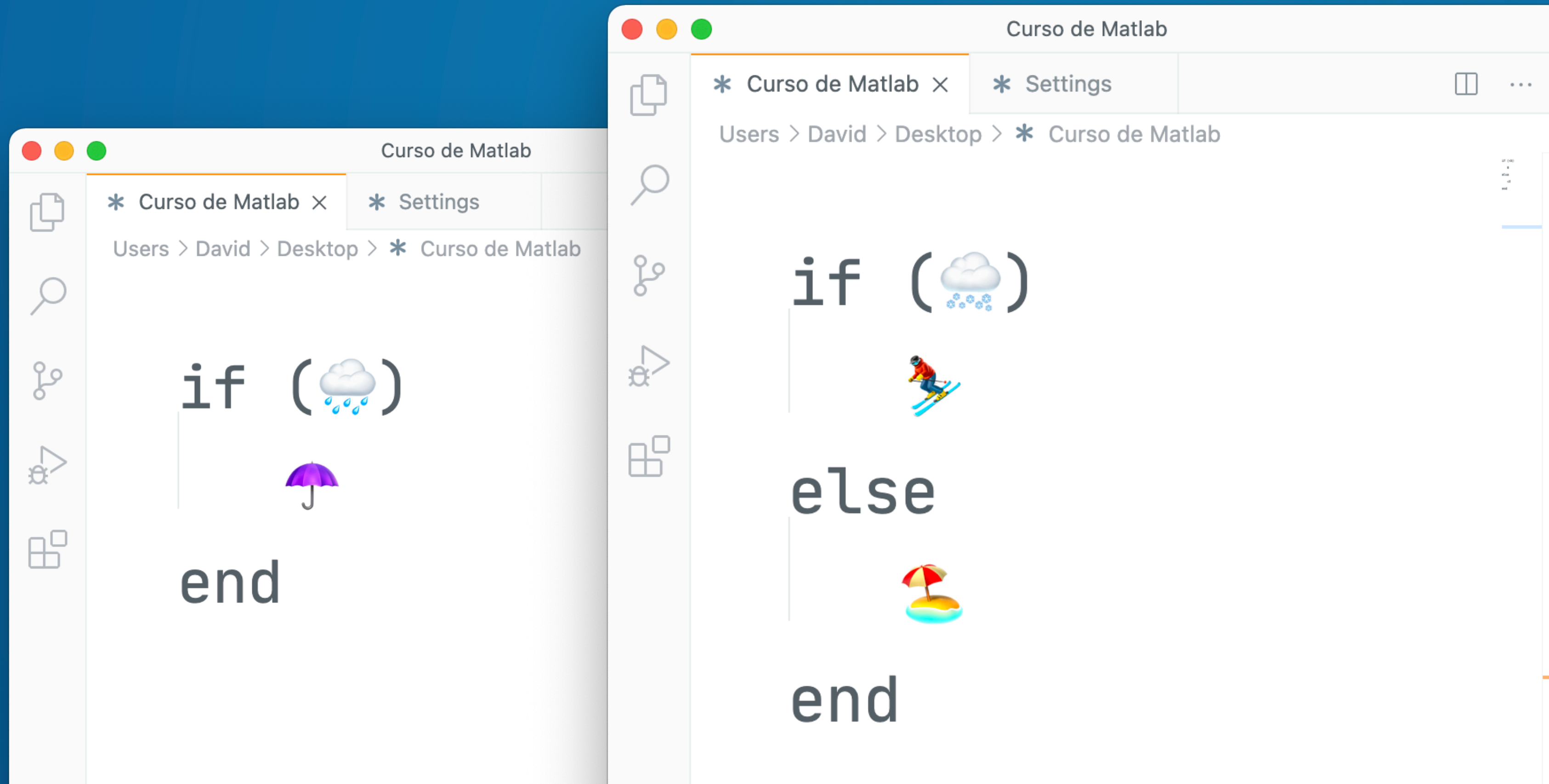
Estructuras condicionales

¿Qué es una estructura condicional?

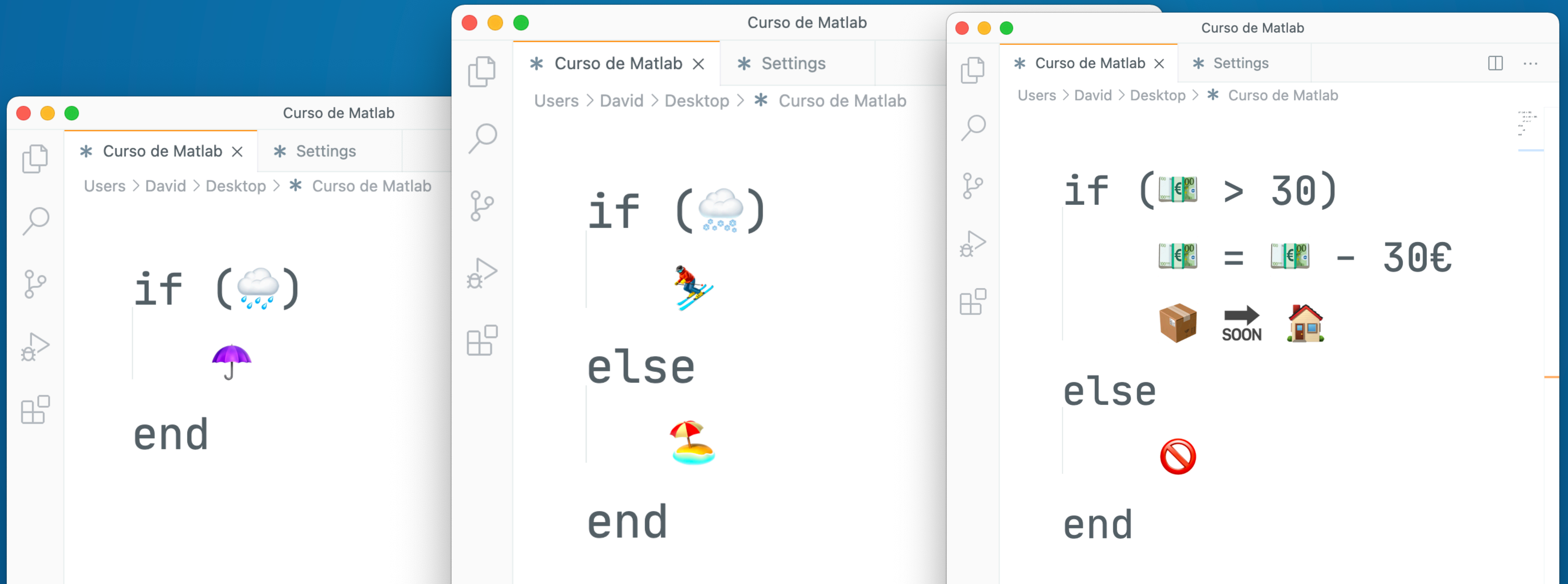
¿Qué es una estructura condicional?



¿Qué es una estructura condicional?



¿Qué es una estructura condicional?



Condicionan el flujo del programa

Condicionan el flujo del programa



Condicionan el flujo del programa



Condicionan el flujo del programa

< IF >

< IF • ELSE >

< IF • ELSEIF >

Condicionan el flujo del programa

< IF >

< IF • ELSE >

< IF • ELSEIF >

< SWITCH >

Diagrama de flujo:
representación gráfica de la
ejecución de un programa.

ESTRUCTURAS SIMPLES: **IF** — DEFINICIÓN:



Diagrama de flujo:
representación gráfica de la
ejecución de un programa.

ESTRUCTURAS SIMPLES: **IF** — DEFINICIÓN:

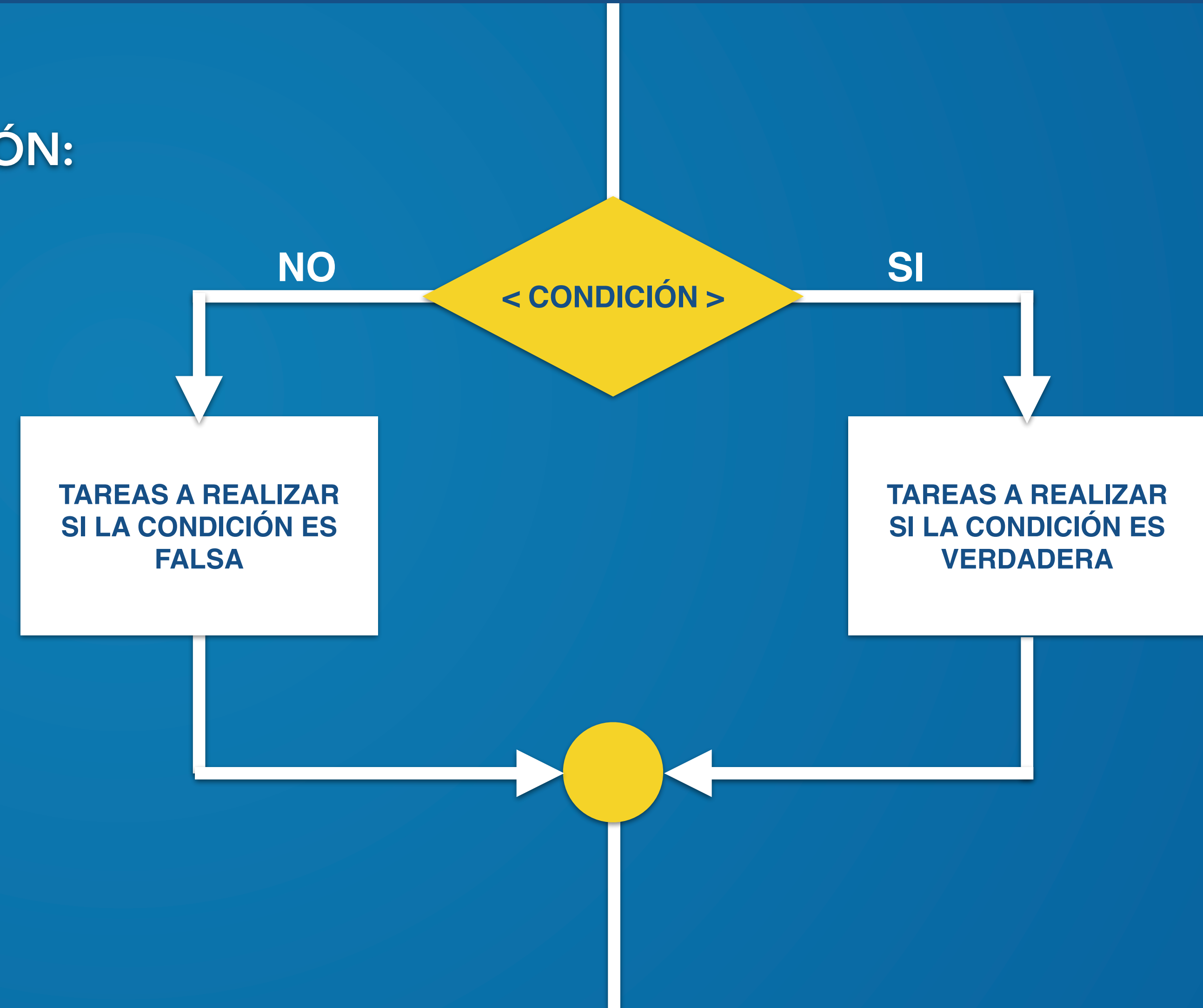
```
if (hayFuego == true)  
  % Acciones a realizar...
```

```
  encenderAlarma();  
  llamarBomberos();
```

```
end
```

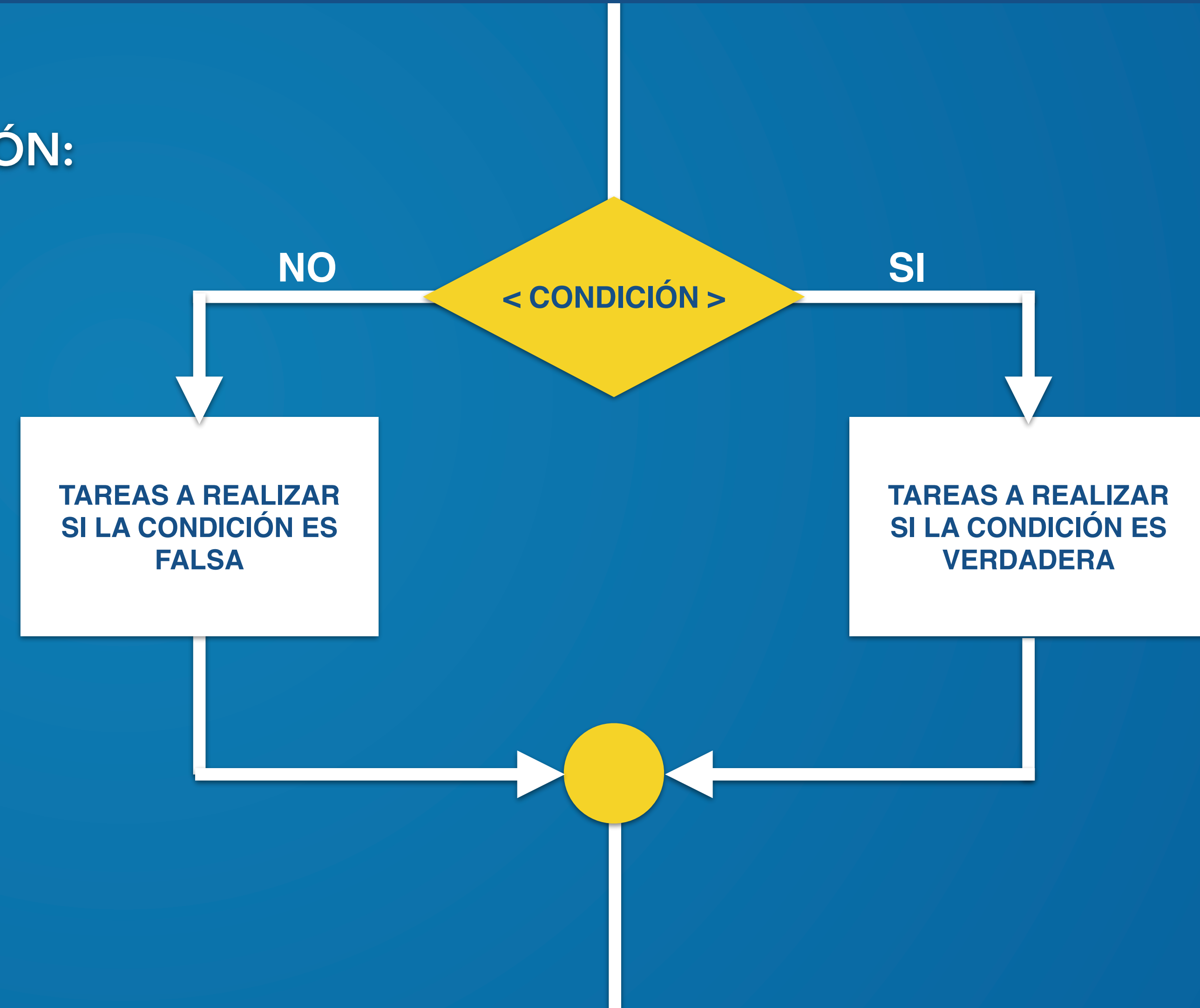


ESTRUCTURAS SIMPLES: **IF • ELSE** — DEFINICIÓN:



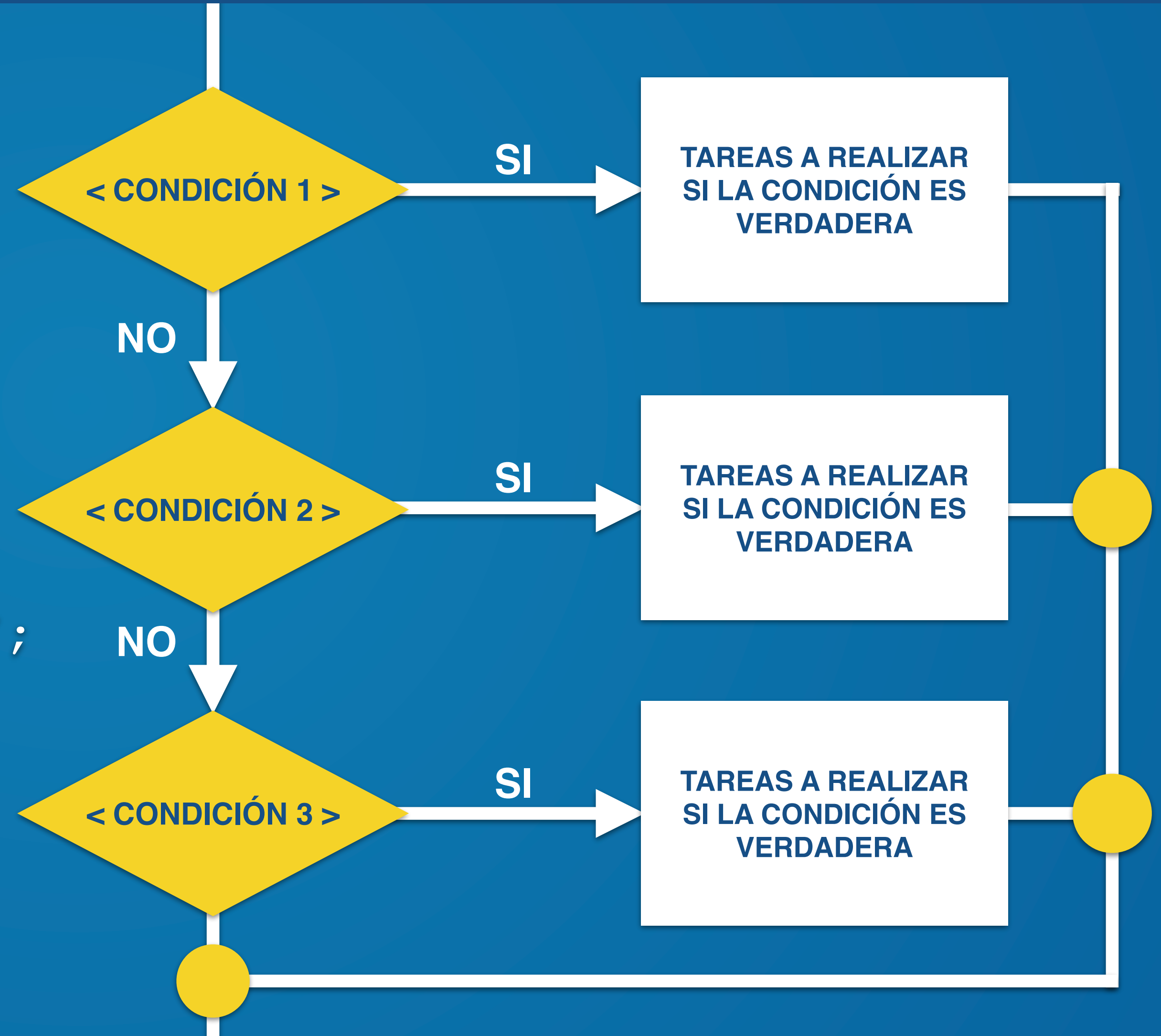
ESTRUCTURAS SIMPLES: **IF • ELSE** — DEFINICIÓN:

```
if (esCara == true)  
    elige = "Local";  
else  
    elige = "Visitante";  
end
```

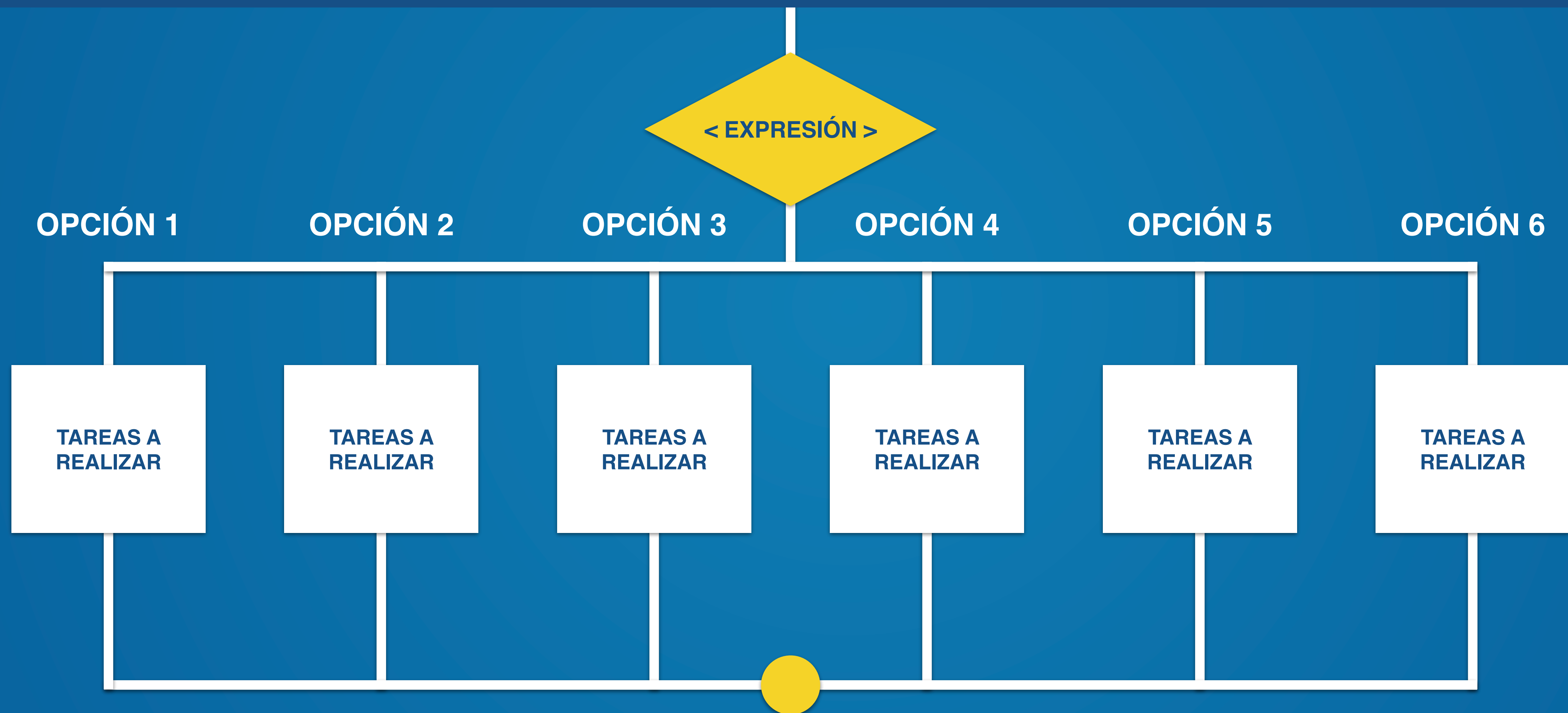


EJEMPLO MATLAB:

```
if (dato == 1)
    mensaje = "Avanza una casilla";
elseif (dato == 2)
    mensaje = "Avanza dos casillas";
elseif (dato == 3)
    mensaje = "Avanza tres casillas";
elseif (dato == 4)
    mensaje = "Avanza cuatro casillas";
...
end
```



► Estructuras condicionales ► Estructuras múltiples (**switch-case**)

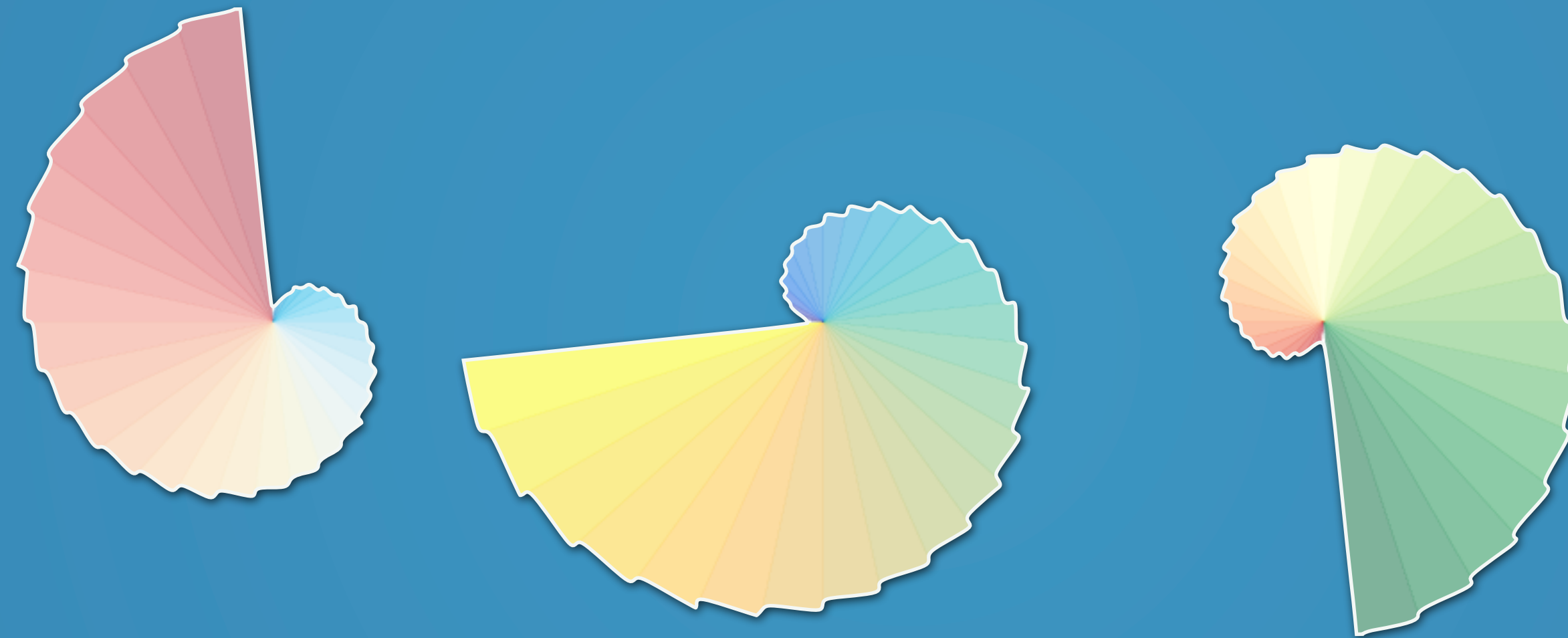


EJEMPLO MATLAB:

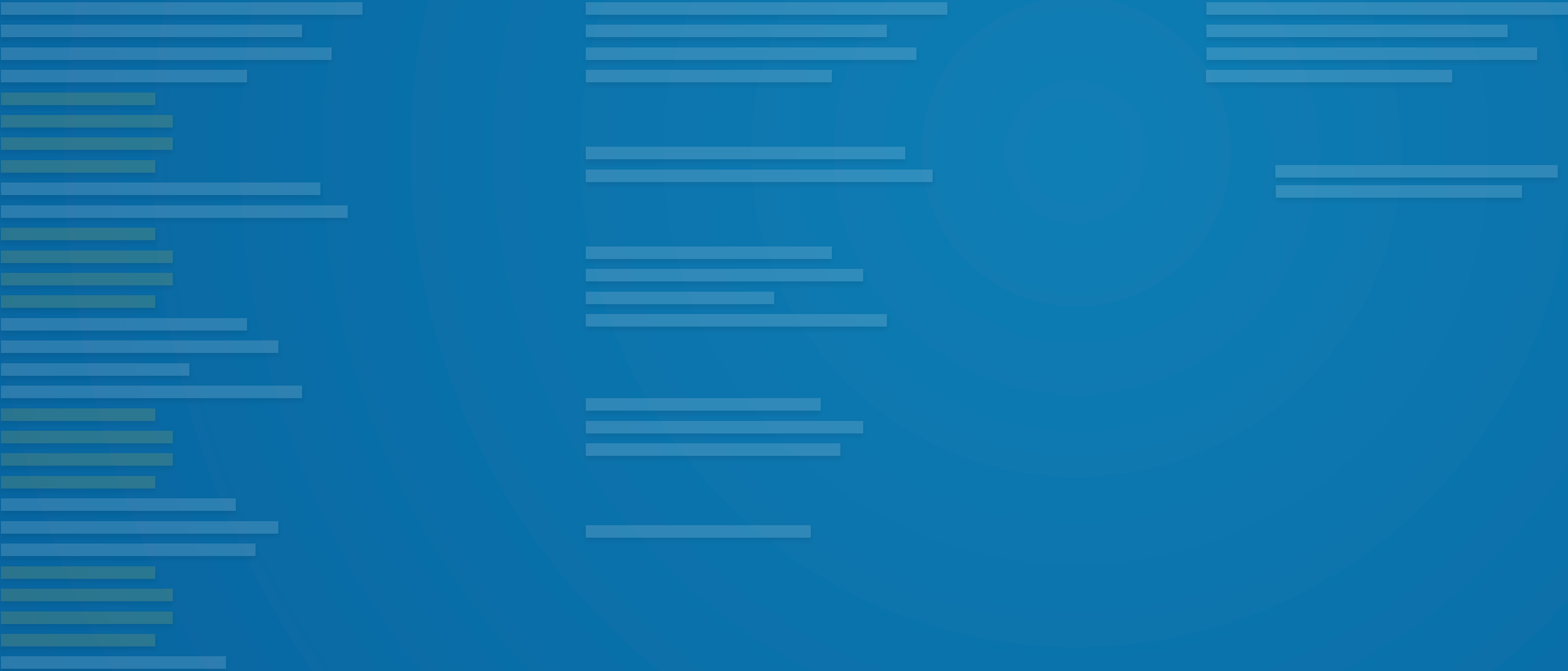


valorDado ➡ ans = 3

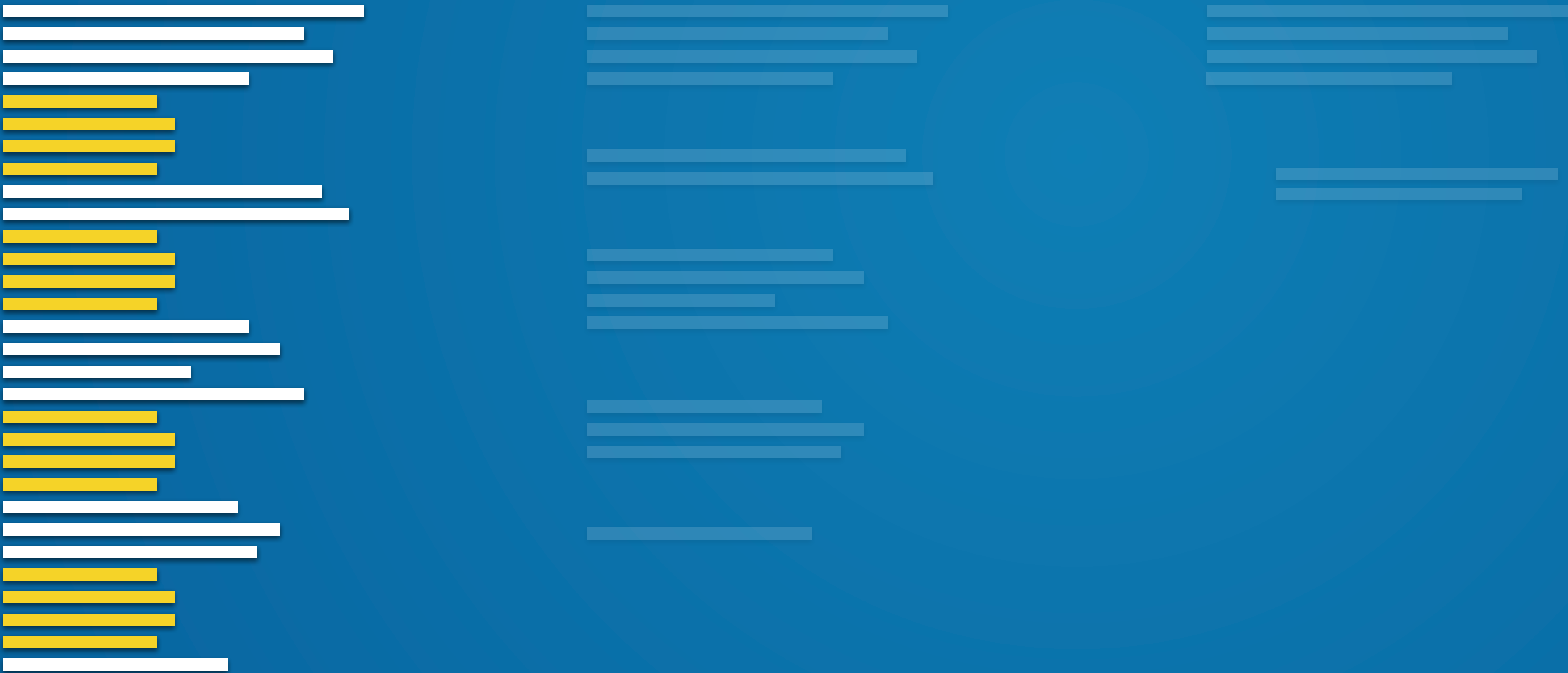
```
switch (valorDado)
    case 1
        mensaje = "Avanza una casilla";
    case 2
        mensaje = "Avanza dos casillas";
    case 3
        mensaje = "Avanza tres casillas";
    case 4
        mensaje = "Avanza cuatro casillas";
    case 5
        mensaje = "Avanza cinco casillas";
    otherwise
        mensaje = "Avanza seis casillas";
end
```



Bucles



PROGRAMA PRINCIPAL:



PROGRAMA PRINCIPAL:



USO DE FUNCIONES:



PROGRAMA PRINCIPAL:

A horizontal bar chart with a blue background. The y-axis lists 18 categories combining age groups and genders. The x-axis represents percentages from 0 to 100. Bars are colored white for 'Total' and 'Men' categories, and yellow for 'Women' categories. The bars are ordered from top to bottom: Total, Men, Women, Total, Men, Women, Total, Men, Women, Total, Men, Women, Total, Men, Women, Total, Men, Women. The lengths of the bars vary, with 'Total' and 'Men' generally being longer than 'Women' for the same age group.

Category	Percentage (%)
Total	100
Men	95
Women	85
18-29	90
30-49	85
50-69	80
70+	75
18-29	70
30-49	65
50-69	60
70+	55
18-29	50
30-49	45
50-69	40
70+	35
18-29	30
30-49	25
50-69	20
70+	15

USO DE FUNCIONES:



reactionTime (valores_1)

reactionTime (valores_2)

reactionTime (valores_3)

reactionTime (valores_4)

USO DE BUCLES:

```

for condition = 1 : 4
    reactionTime (valores_condicion)
end

```

¿Qué es un bucle?



¿Qué es un bucle?



CONJUNTO DE LÍNEAS DE CODIGO QUE SE EJECUTA REPETIDAMENTE



¿Qué es un bucle?



CONJUNTO DE LÍNEAS DE CODIGO QUE SE EJECUTA REPETIDAMENTE



MIENTRAS QUE LA CONDICIÓN ASIGNADA AL BUCLE SE CUMPLA



¿Qué es un bucle?



CONJUNTO DE LÍNEAS DE CODIGO QUE SE EJECUTA REPETIDAMENTE



MIENTRAS QUE LA CONDICIÓN ASIGNADA AL BUCLE SE CUMPLA



EJECUTAR LA MISMA ACCIÓN EN REPETIDAS OCASIONES



¿Qué es un bucle?



CONJUNTO DE LÍNEAS DE CODIGO QUE SE EJECUTA REPETIDAMENTE



MIENTRAS QUE LA CONDICIÓN ASIGNADA AL BUCLE SE CUMPLA



EJECUTAR LA MISMA ACCIÓN EN REPETIDAS OCASIONES



REUTILIZAR EL CÓDIGO Y EVITAR ERRORES

Tipos de bucles



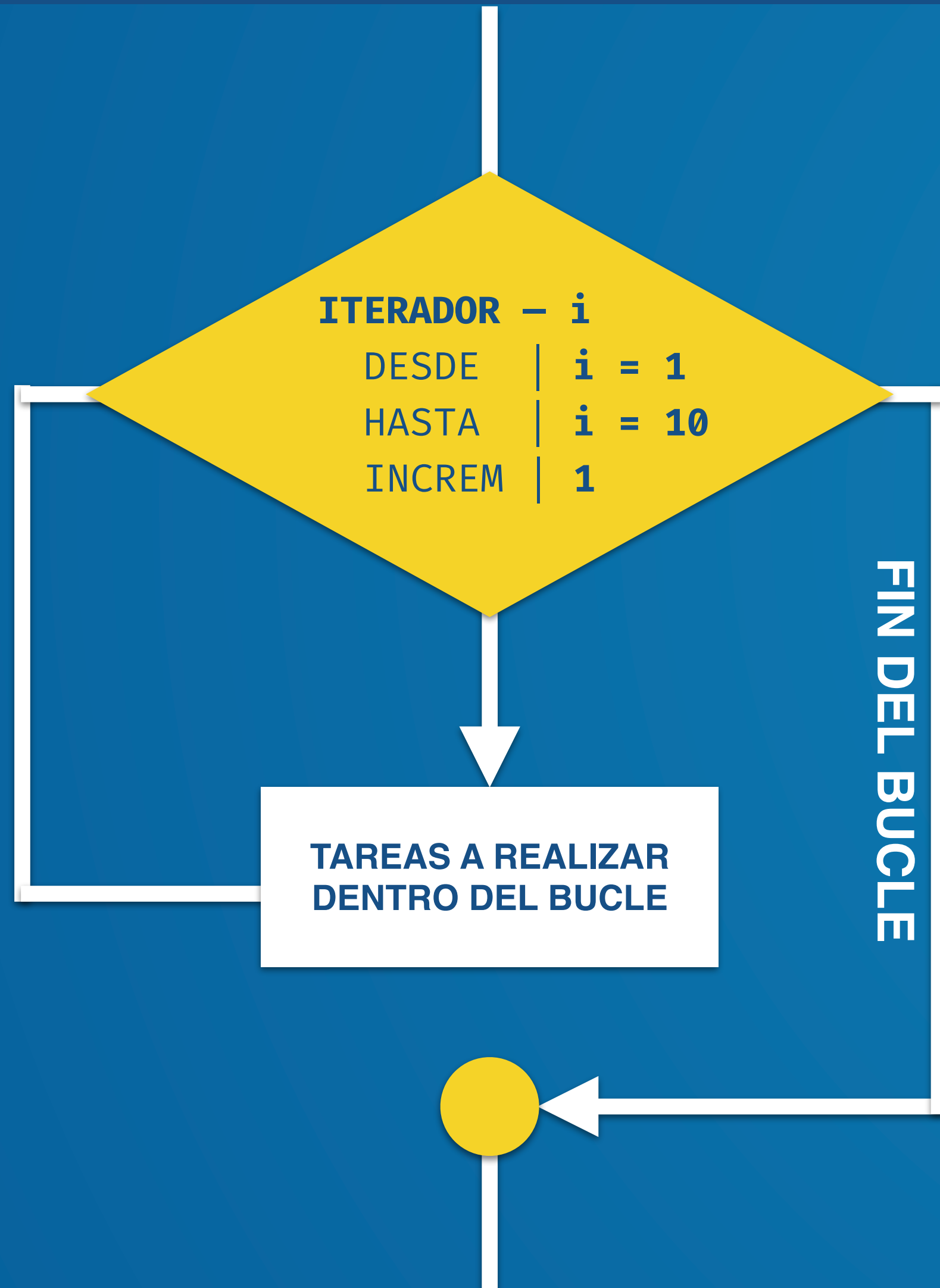
¿Qué es un bucle **for**?

```
for i = 1:10
```

```
    disp("El valor de i es: ");
```

```
    disp(i);
```

```
end
```



PROBLEMA: CALCULAR MI NOTA MEDIA DEL MÁSTER

```
nAsignaturas = 8;
```

```
vectorNotas = [7,10,9,9,9.5,5,7,6];
```

SOLUCIÓN:

1. SUMAR UNA A UNA TODAS LAS NOTAS DE LAS ASIGNATURAS
2. DIVIDIR ENTRE EL NÚMERO TOTAL DE ASIGNATURAS



SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;  
  
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end  
  
media = total/nAsignaturas;
```



SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;  
  
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end  
  
media = total/nAsignaturas;
```



SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;  
  
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end  
  
media = total/nAsignaturas;
```




SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```



SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```




SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas                                i = 1  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end  
media = total/nAsignaturas;
```



SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 1

```
    notaAsignatura = vectorNotas(i);
```

```
    total = total + notaAsignatura;
```

```
end
```

```
media = total/nAsignaturas;
```

SOLUCIÓN 1:

```
nAsignaturas = 8;
vectorNotas = [7,10,9,9,9.5,5,7,6];
total = 0;
```

```
for i = 1:nAsignaturas                                i = 1
    notaAsignatura = vectorNotas(i);
    total = total + notaAsignatura;
end
media = total/nAsignaturas;
```


SOLUCIÓN 1:

```
nAsignaturas = 8;
vectorNotas = [7,10,9,9,9.5,5,7,6];
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 1

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 7

```
    total = total + notaAsignatura;
```

```
end
```

```
media = total/nAsignaturas;
```

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```

i = 1

notaAsignatura = 7

SOLUCIÓN 1:

```
nAsignaturas = 8;
vectorNotas = [7,10,9,9,9.5,5,7,6];
total = 0;
```

```
for i = 1:nAsignaturas
    notaAsignatura = vectorNotas(i);
    total = total + notaAsignatura;
end
```

```
media = total/nAsignaturas;
```

i = 1

notaAsignatura = 7

total = 7

SOLUCIÓN 1:

```
nAsignaturas = 8;
vectorNotas = [7,10,9,9,9.5,5,7,6];
total = 0;
```

```
for i = 1:nAsignaturas
    notaAsignatura = vectorNotas(i);
    total = total + notaAsignatura;
end
```

```
media = total/nAsignaturas;
```

i = 1

notaAsignatura = 7

total = 7

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 1

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 7

```
    total = total + notaAsignatura;
```

total = 7

```
end
```

```
media = total/nAsignaturas;
```


SOLUCIÓN 1:

```
nAsignaturas = 8;
vectorNotas = [7,10,9,9,9.5,5,7,6];
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 2

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 7

```
    total = total + notaAsignatura;
```

total = 7

```
end
```

```
media = total/nAsignaturas;
```

SOLUCIÓN 1:

```
nAsignaturas = 8;
vectorNotas = [7,10,9,9,9.5,5,7,6];
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 2

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 7

```
    total = total + notaAsignatura;
```

total = 7

```
end
```

```
media = total/nAsignaturas;
```




SOLUCIÓN 1:

```
nAsignaturas = 8;
```

```
vectorNotas = [7,10,9,9,9.5,5,7,6];
```

```
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 2

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 7

```
    total = total + notaAsignatura;
```

total = 7

```
end
```

```
media = total/nAsignaturas;
```

SOLUCIÓN 1:

```
nAsignaturas = 8;
```

```
vectorNotas = [7,10,9,9,9.5,5,7,6];
```

```
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 2

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 10

```
    total = total + notaAsignatura;
```

total = 7

```
end
```

```
media = total/nAsignaturas;
```

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```

i = 2

notaAsignatura = 10

total = 7

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```

i = 2

notaAsignatura = 10

total = 17



SOLUCIÓN 1:

```
nAsignaturas = 8;
vectorNotas = [7,10,9,9,9.5,5,7,6];
total = 0;
```

```
for i = 1:nAsignaturas
    notaAsignatura = vectorNotas(i);
    total = total + notaAsignatura;
end
```

```
media = total/nAsignaturas;
```

```
i = 2
notaAsignatura = 10
total = 17
```




SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 2

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 10

```
    total = total + notaAsignatura;
```

total = 17

```
end
```

```
media = total/nAsignaturas;
```

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 3

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 10

```
    total = total + notaAsignatura;
```

total = 17

```
end
```

```
media = total/nAsignaturas;
```

SOLUCIÓN 1:

```
nAsignaturas = 8;
```

```
vectorNotas = [7,10,9,9,9.5,5,7,6];
```

```
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 3

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 10

```
    total = total + notaAsignatura;
```

total = 17

```
end
```

```
media = total/nAsignaturas;
```




SOLUCIÓN 1:

```
nAsignaturas = 8;
vectorNotas = [7,10,9,9,9.5,5,7,6];
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 3

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 10

```
    total = total + notaAsignatura;
```

total = 17

```
end
```

```
media = total/nAsignaturas;
```



SOLUCIÓN 1:

```
nAsignaturas = 8;
```

```
vectorNotas = [7,10,9,9,9.5,5,7,6];
```

```
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 3

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 9

```
    total = total + notaAsignatura;
```

total = 17

```
end
```

```
media = total/nAsignaturas;
```




SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```

i = 3

notaAsignatura = 9

total = 17



SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```

i = 3

notaAsignatura = 9

total = 26



SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```

i = 3

notaAsignatura = 9

total = 26



SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 3

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 9

```
    total = total + notaAsignatura;
```

total = 26

```
end
```

```
media = total/nAsignaturas;
```



SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 4

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 9

```
    total = total + notaAsignatura;
```

total = 26

```
end
```

```
media = total/nAsignaturas;
```




SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 4

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 9

```
    total = total + notaAsignatura;
```

total = 26

```
end
```

```
media = total/nAsignaturas;
```



SOLUCIÓN 1:

```
nAsignaturas = 8;
```

```
vectorNotas = [7,10,9,9,9.5,5,7,6];
```

```
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 4

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 9

```
    total = total + notaAsignatura;
```

total = 26

```
end
```

```
media = total/nAsignaturas;
```



SOLUCIÓN 1:

```
nAsignaturas = 8;
```

```
vectorNotas = [7,10,9,9,9.5,5,7,6];
```

```
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 4

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 9

```
    total = total + notaAsignatura;
```

total = 26

```
end
```

```
media = total/nAsignaturas;
```




SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```

i = 4

notaAsignatura = 9

total = 26



SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```

i = 4

notaAsignatura = 9

total = 35



SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```

i = 4
notaAsignatura = 9
total = 35



SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 4

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 9

```
    total = total + notaAsignatura;
```

total = 35

```
end
```

```
media = total/nAsignaturas;
```



SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 5

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 9

```
    total = total + notaAsignatura;
```

total = 35

```
end
```

```
media = total/nAsignaturas;
```




SOLUCIÓN 1:

```
nAsignaturas = 8;
```

```
vectorNotas = [7,10,9,9,9.5,5,7,6];
```

```
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 5

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 9

```
    total = total + notaAsignatura;
```

total = 35

```
end
```

```
media = total/nAsignaturas;
```

7	10	9	9	9.5			
---	----	---	---	-----	--	--	--

SOLUCIÓN 1:

```
nAsignaturas = 8;
```

```
vectorNotas = [7,10,9,9,9.5,5,7,6];
```

```
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 5

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 9

```
    total = total + notaAsignatura;
```

total = 35

```
end
```

```
media = total/nAsignaturas;
```


7	10	9	9	9.5			
---	----	---	---	-----	--	--	--

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 5

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 9.5

```
    total = total + notaAsignatura;
```

total = 35

```
end
```

```
media = total/nAsignaturas;
```

7	10	9	9	9.5			
---	----	---	---	-----	--	--	--

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```

i = 5

notaAsignatura = 9.5

total = 35



SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```

i = 5

notaAsignatura = 9.5

total = 44.5

7	10	9	9	9.5			
---	----	---	---	-----	--	--	--

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```

```
i = 5  
notaAsignatura = 9.5  
total = 44.5
```

7	10	9	9	9.5			
---	----	---	---	-----	--	--	--

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 5

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 9.5

```
    total = total + notaAsignatura;
```

total = 44.5

```
end
```

```
media = total/nAsignaturas;
```


7	10	9	9	9.5			
---	----	---	---	-----	--	--	--

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 6

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 9.5

```
    total = total + notaAsignatura;
```

total = 44.5

```
end
```

```
media = total/nAsignaturas;
```

7	10	9	9	9.5			
---	----	---	---	-----	--	--	--

SOLUCIÓN 1:

```
nAsignaturas = 8;
vectorNotas = [7,10,9,9,9.5,5,7,6];
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 6

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 9.5

```
    total = total + notaAsignatura;
```

total = 44.5

```
end
```

```
media = total/nAsignaturas;
```

7	10	9	9	9.5	5		
---	----	---	---	-----	---	--	--

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 6

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 9.5

```
    total = total + notaAsignatura;
```

total = 44.5

```
end
```

```
media = total/nAsignaturas;
```


7	10	9	9	9.5	5		
---	----	---	---	-----	---	--	--

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 6

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 5

```
    total = total + notaAsignatura;
```

total = 44.5

```
end
```

```
media = total/nAsignaturas;
```

7	10	9	9	9.5	5		
---	----	---	---	-----	---	--	--

SOLUCIÓN 1:

```
nAsignaturas = 8;
vectorNotas = [7,10,9,9,9.5,5,7,6];
total = 0;
```

```
for i = 1:nAsignaturas
    notaAsignatura = vectorNotas(i);
    total = total + notaAsignatura;
end
```

```
media = total/nAsignaturas;
```

i = 6

notaAsignatura = 5

total = 44.5

7	10	9	9	9.5	5		
---	----	---	---	-----	---	--	--

SOLUCIÓN 1:

```
nAsignaturas = 8;
vectorNotas = [7,10,9,9,9.5,5,7,6];
total = 0;
```

```
for i = 1:nAsignaturas
    notaAsignatura = vectorNotas(i);
    total = total + notaAsignatura;
end
```

```
media = total/nAsignaturas;
```

i = 6

notaAsignatura = 5

total = 49.5

7	10	9	9	9.5	5		
---	----	---	---	-----	---	--	--

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```

i = 6

notaAsignatura = 5

total = 49.5

7	10	9	9	9.5	5		
---	----	---	---	-----	---	--	--

SOLUCIÓN 1:

```
nAsignaturas = 8;
vectorNotas = [7,10,9,9,9.5,5,7,6];
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 6

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 5

```
    total = total + notaAsignatura;
```

total = 49.5

```
end
```

```
media = total/nAsignaturas;
```


7	10	9	9	9.5	5		
---	----	---	---	-----	---	--	--

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 7

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 5

```
    total = total + notaAsignatura;
```

total = 49.5

```
end
```

```
media = total/nAsignaturas;
```

7	10	9	9	9.5	5		
---	----	---	---	-----	---	--	--

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 7

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 5

```
    total = total + notaAsignatura;
```

total = 49.5

```
end
```

```
media = total/nAsignaturas;
```


7	10	9	9	9.5	5	7	
---	----	---	---	-----	---	---	--

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 7

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 5

```
    total = total + notaAsignatura;
```

total = 49.5

```
end
```

```
media = total/nAsignaturas;
```

7	10	9	9	9.5	5	7	
---	----	---	---	-----	---	---	--

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 7

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 7

```
    total = total + notaAsignatura;
```

total = 49.5

```
end
```

```
media = total/nAsignaturas;
```

7	10	9	9	9.5	5	7	
---	----	---	---	-----	---	---	--

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```

i = 7

notaAsignatura = 7

total = 49.5

7	10	9	9	9.5	5	7	
---	----	---	---	-----	---	---	--

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```

i = 7

notaAsignatura = 7

total = 56.5

7	10	9	9	9.5	5	7	
---	----	---	---	-----	---	---	--

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```

i = 7
notaAsignatura = 7
total = 56.5

7	10	9	9	9.5	5	7	
---	----	---	---	-----	---	---	--

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 7

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 7

```
    total = total + notaAsignatura;
```

total = 56.5

```
end
```

```
media = total/nAsignaturas;
```

7	10	9	9	9.5	5	7	
---	----	---	---	-----	---	---	--

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 8

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 7

```
    total = total + notaAsignatura;
```

total = 56.5

```
end
```

```
media = total/nAsignaturas;
```

7	10	9	9	9.5	5	7	
---	----	---	---	-----	---	---	--

SOLUCIÓN 1:

```
nAsignaturas = 8;
```

```
vectorNotas = [7,10,9,9,9.5,5,7,6];
```

```
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 8

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 7

```
    total = total + notaAsignatura;
```

total = 56.5

```
end
```

```
media = total/nAsignaturas;
```


7	10	9	9	9.5	5	7	6
---	----	---	---	-----	---	---	---

SOLUCIÓN 1:

```
nAsignaturas = 8;
```

```
vectorNotas = [7,10,9,9,9.5,5,7,6];
```

```
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 8

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 7

```
    total = total + notaAsignatura;
```

total = 56.5

```
end
```

```
media = total/nAsignaturas;
```

7	10	9	9	9.5	5	7	6
---	----	---	---	-----	---	---	---

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas
```

i = 8

```
    notaAsignatura = vectorNotas(i);
```

notaAsignatura = 6

```
    total = total + notaAsignatura;
```

total = 56.5

```
end
```

```
media = total/nAsignaturas;
```


7	10	9	9	9.5	5	7	6
---	----	---	---	-----	---	---	---

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```

i = 8

notaAsignatura = 6

total = 56.5

7	10	9	9	9.5	5	7	6
---	----	---	---	-----	---	---	---

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```

i = 8

notaAsignatura = 6

total = 62.5

7	10	9	9	9.5	5	7	6
---	----	---	---	-----	---	---	---

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```

i = 8
notaAsignatura = 6
total = 62.5

7	10	9	9	9.5	5	7	6
---	----	---	---	-----	---	---	---

SOLUCIÓN 1:

```
nAsignaturas = 8;
vectorNotas = [7,10,9,9,9.5,5,7,6];
total = 0;
```

```
for i = 1:nAsignaturas
    notaAsignatura = vectorNotas(i);
    total = total + notaAsignatura;
end
```

```
media = total/nAsignaturas;
```

i = 8

notaAsignatura = 6

total = 62.5

media = 7.8125

7	10	9	9	9.5	5	7	6
---	----	---	---	-----	---	---	---

SOLUCIÓN 1:

```
nAsignaturas = 8;  
vectorNotas = [7,10,9,9,9.5,5,7,6];  
total = 0;
```

```
for i = 1:nAsignaturas  
    notaAsignatura = vectorNotas(i);  
    total = total + notaAsignatura;  
end
```

```
media = total/nAsignaturas;
```

i = 8

notaAsignatura = 6

total = 62.5



media = 7.8125

SOLUCIÓN 2:

SOLUCIÓN 2:

```
total = sum(notas);
```

SOLUCIÓN 2:

```
total = sum(notas);
```

```
media = total/nAsignaturas;
```

SOLUCIÓN 3:

SOLUCIÓN 3:

```
media = mean(notas);
```


SOLUCIÓN 3:

```
media = mean(notas);
```

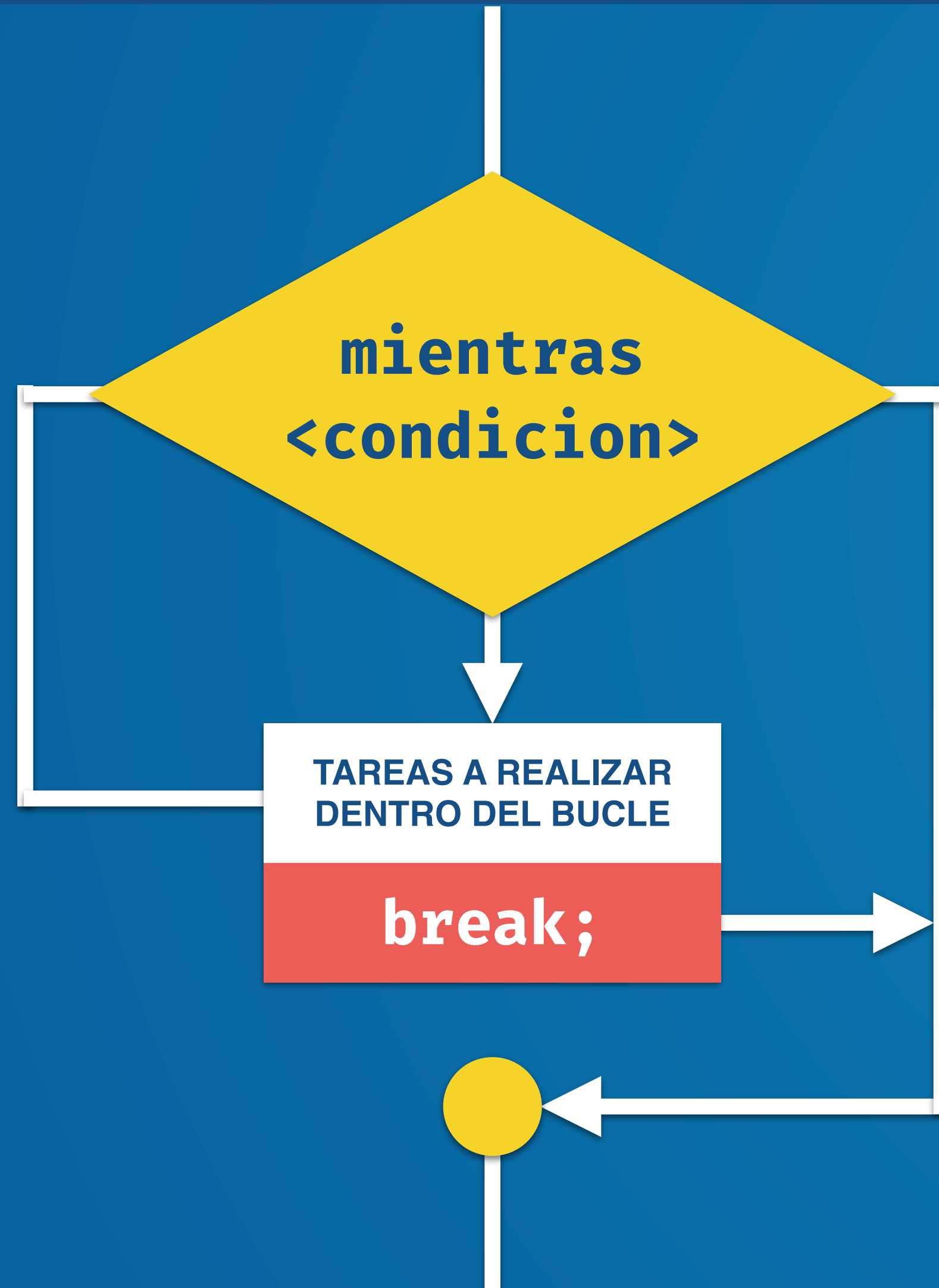
Matlab dispone de muchas (pero muchas) funcionalidades que nos facilitaran la programación.

SOLUCIÓN 3:

```
media = mean(notas);
```



Matlab dispone de muchas (pero muchas) funcionalidades que nos facilitaran la programación.



```
vuelta = 0;
```

```
while vuelta < 10
```

```
disp("Vuelta número: ");
```

```
disp(vuelta);
```

```
vuelta = vuelta + 1;
```

```
end
```

PROBLEMA: PROGRAMAR EL SISTEMA DE CONTROL DE UN DESPERTADOR

PROPUESTA:

1. SONAR DESPERTADOR MIENTRAS NO SE PULSE EL BOTON DE STOP.
2. SI NO SE PULSA EL BOTÓN, DESPUES DE SONAR 3 VECES, PARAR DE SONAR.

`comprobarBotonStop()` ; \longrightarrow `true` / `false`

`pararAlarma()` ;

`sonarAlarma()` ;

SOLUCIÓN:

```
nVeces = 0;
while (comprobarBotonStop() == false)
    if (nVeces < 3)
        sonarAlarma();
        nVeces = nVeces + 1;
    else
        pararAlarma();
        break;
    end
end
```


SOLUCIÓN:

```
nVeces = 0;
while (comprobarBotonStop() == false)
    if (nVeces < 3)
        sonarAlarma();
        nVeces = nVeces + 1;
    else
        pararAlarma();
        break;
    end
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

```
    if (nVeces < 3)
```

```
        sonarAlarma();
```

```
        nVeces = nVeces + 1;
```

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)           () ↦ false
```

```
    if (nVeces < 3)
```

```
        sonarAlarma();
```

```
        nVeces = nVeces + 1;
```

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```


SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false) () ↦ false
```

```
    if (nVeces < 3)
```

```
        sonarAlarma();
```

```
        nVeces = nVeces + 1;
```

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false) () ↦ false
```

```
    if (nVeces < 3) nVeces = 0
```

```
        sonarAlarma();
```

```
        nVeces = nVeces + 1;
```

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```


SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false) () ↦ false
```

```
    if (nVeces < 3) nVeces = 0
```

```
        sonarAlarma();
```

```
        nVeces = nVeces + 1;
```

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false) () ↦ false
```

```
    if (nVeces < 3) nVeces = 0
```

```
        sonarAlarma();
```

```
        nVeces = nVeces + 1;
```

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```



SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

() ↦ false

```
    if (nVeces < 3)
```

nVeces = 0

```
        sonarAlarma();
```



```
        nVeces = nVeces + 1;
```

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```


SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

() ↦ false

```
    if (nVeces < 3)
```

nVeces = 0

```
        sonarAlarma();
```



```
        nVeces = nVeces + 1;
```

nVeces = 1

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

$() \mapsto false$

```
    if (nVeces < 3)
```

$nVeces = 0$

```
        sonarAlarma();
```



```
        nVeces = nVeces + 1;
```

$nVeces = 1$

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```


SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

() ↦ false

```
    if (nVeces < 3)
```

nVeces = 0

```
        sonarAlarma();
```



```
        nVeces = nVeces + 1;
```

nVeces = 1

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
end
```

```
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

() ↦ false

```
    if (nVeces < 3)
```

nVeces = 0

```
        sonarAlarma();
```



```
        nVeces = nVeces + 1;
```

nVeces = 1

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)           () ↦ false
```

```
    if (nVeces < 3)                               nVeces = 0
```

```
        sonarAlarma();                             🔔
```

```
        nVeces = nVeces + 1;                       nVeces = 1
```

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```


SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false) () ↦ false
```

```
    if (nVeces < 3) nVeces = 0
```

```
        sonarAlarma(); 
```

```
        nVeces = nVeces + 1; nVeces = 1
```

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)           () ↦ false
```

```
    if (nVeces < 3)                               nVeces = 1
```

```
        sonarAlarma();                             🔔
```

```
        nVeces = nVeces + 1;                       nVeces = 1
```

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```


SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false) () ↦ false
```

```
    if (nVeces < 3) nVeces = 1
```

```
        sonarAlarma(); 
```

```
        nVeces = nVeces + 1; nVeces = 1
```

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false) () ↦ false
```

```
    if (nVeces < 3) nVeces = 1
```

```
        sonarAlarma();  
```

```
        nVeces = nVeces + 1; nVeces = 1
```

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

() ↦ false

```
    if (nVeces < 3)
```

nVeces = 1

```
        sonarAlarma();
```



```
        nVeces = nVeces + 1;
```

nVeces = 1

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```


SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

() ↦ false

```
    if (nVeces < 3)
```

nVeces = 1

```
        sonarAlarma();
```



```
        nVeces = nVeces + 1;
```

nVeces = 2

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

() ↦ false

```
    if (nVeces < 3)
```

nVeces = 1

```
        sonarAlarma();
```



```
        nVeces = nVeces + 1;
```

nVeces = 2

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```


SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

() ↦ false

```
    if (nVeces < 3)
```

nVeces = 1

```
        sonarAlarma();
```



```
        nVeces = nVeces + 1;
```

nVeces = 2

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

() ↦ false

```
    if (nVeces < 3)
```

nVeces = 1

```
        sonarAlarma();
```



```
        nVeces = nVeces + 1;
```

nVeces = 2

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)           () ↦ false
```

```
    if (nVeces < 3)
```

```
        sonarAlarma();
```

```
        nVeces = nVeces + 1;
```

nVeces = 1



nVeces = 2

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```


SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false) () ↦ false
```

```
    if (nVeces < 3) nVeces = 1
```

```
        sonarAlarma();  
```

```
        nVeces = nVeces + 1; nVeces = 2
```

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false) () ↦ false
```

```
    if (nVeces < 3) nVeces = 2
```

```
        sonarAlarma(); 🔔 🔔
```

```
        nVeces = nVeces + 1; nVeces = 2
```

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```


SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false) () ↦ false
```

```
    if (nVeces < 3) nVeces = 2
```

```
        sonarAlarma();  
```

```
        nVeces = nVeces + 1; nVeces = 2
```

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

() ↦ false

```
    if (nVeces < 3)
```

nVeces = 2

```
        sonarAlarma();
```



```
        nVeces = nVeces + 1;
```

nVeces = 2

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

() ↦ false

```
    if (nVeces < 3)
```

nVeces = 2

```
        sonarAlarma();
```



```
        nVeces = nVeces + 1;
```

nVeces = 2

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```


SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

() ↦ false

```
    if (nVeces < 3)
```

nVeces = 2

```
        sonarAlarma();
```



```
        nVeces = nVeces + 1;
```

nVeces = 3

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

() ↦ false

```
    if (nVeces < 3)
```

nVeces = 2

```
        sonarAlarma();
```



```
        nVeces = nVeces + 1;
```

nVeces = 3

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```


SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

() ↦ false

```
    if (nVeces < 3)
```

nVeces = 2

```
        sonarAlarma();
```



```
        nVeces = nVeces + 1;
```

nVeces = 3

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

() ↦ false

```
    if (nVeces < 3)
```

```
        sonarAlarma();
```

```
        nVeces = nVeces + 1;
```

nVeces = 2



nVeces = 3

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)           () ↦ false
```

```
    if (nVeces < 3)
```

```
        sonarAlarma();
```

```
        nVeces = nVeces + 1;
```

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```

nVeces = 2



nVeces = 3

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false) () ↦ false
```

```
    if (nVeces < 3) nVeces = 3
```

```
        sonarAlarma(); 
```

```
        nVeces = nVeces + 1; nVeces = 3
```

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

() ↦ false

```
    if (nVeces < 3)
```

nVeces = 3

```
        sonarAlarma();
```



```
        nVeces = nVeces + 1;
```

nVeces = 3

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```


SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

() ↦ false

```
    if (nVeces < 3)
```

nVeces = 3

```
        sonarAlarma();
```



```
        nVeces = nVeces + 1;
```

nVeces = 3

```
    else
```

```
        pararAlarma();
```

```
        break;
```

```
    end
```

```
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

() ↦ false

```
    if (nVeces < 3)
```

nVeces = 3

```
        sonarAlarma();
```



```
        nVeces = nVeces + 1;
```

nVeces = 3

```
    else
```

```
        pararAlarma();
```



```
        break;
```

```
    end
```

```
end
```

SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

() ↦ false

```
    if (nVeces < 3)
```

nVeces = 3

```
        sonarAlarma();
```



```
        nVeces = nVeces + 1;
```

nVeces = 3

```
    else
```

```
        pararAlarma();
```



```
        break;
```

```
    end
```

```
end
```


SOLUCIÓN:

```
nVeces = 0;
```

```
while (comprobarBotonStop() == false)
```

() ↦ false

```
    if (nVeces < 3)
```

nVeces = 3

```
        sonarAlarma();
```



```
        nVeces = nVeces + 1;
```

nVeces = 3

```
    else
```

```
        pararAlarma();
```



```
        break;
```

```
    end
```

```
end
```



```
while (true)
    disp( 'Trapped!' );
end
```




```
while (true)
    disp('Trapped!');
end
```

MATLAB® & SIMULINK®

- Curso de introducción 2025 -

Presentación por: **David López García**