

## Parte 2. Hojas de Cálculo

Unai Pérez-Goya  
Área de Lenguajes y Sistemas Informáticos

Curso 2024/2025

## Parte 2.

# Hojas de Cálculo

Sesion IX: Condicionales y Funciones de Búsqueda

# Tabla de contenidos

Arrays (matrices)

Funciones condicionales

Búsqueda

Indexación

Repaso

# Funciones para la gestión de arrays



# Arrays Y las hojas de calculo

- ▶ Un **array**, es un rango formado por celdas llenas de datos en una hoja de cálculo.
- ▶ Existen funciones preparadas para su gestión como grupo de valores y no como celda individual.
- ▶ A veces el resultado será otro array.
  - ▶ Valores colocados en celdas en lugar de la fórmula.
- ▶ En el caso de array las cosas se hacen diferentes.

# Arrays

► Normas basicas:

1. Seleccione el rango resultante;
2. Una vez introducida la fórmula, pulse **Ctrl+Shift+Enter**, en lugar de **Enter**.

E1:G4								
  = {=A1:C4*10}								
	A	B	C	D	E	F	G	
1	1	2	3		10	20	30	
2	4	5	6		40	50	60	
3	7	8	9		70	80	90	
4	10	11	12		100	110	120	
5								
6								
7								
8								
9								

# Funciones de gestión de arrays

👉 **Sintaxis:** `TRANSPOSE (someArray)`


- ▶ **Acción:** Se obtiene la inversa de `someArray`.

The screenshot shows a spreadsheet with a formula bar at the top displaying `=TRANSPOSE(A1:C4)`. The spreadsheet data is as follows:

	A	B	C	D	E	F	G	H	I	J
1	1	2	3		1	4	7	10		
2	4	5	6		2	5				
3	7	8	9		3	6				
4	10	11	12							
5										
6										
7										
8										

An error dialog box is displayed over the spreadsheet. The dialog box has the title "LibreOffice 5.0.4.2" and a close button (X). It contains a lightbulb icon and the text "You cannot change only part of an array." Below the text is an "OK" button.

# Funciones de gestión de arrays

 **Sintaxis:** `SUMPRODUCT(arrayOne;arrayTwo;...)`

- ▶ **Acción:** Se obtiene la suma del producto de los elementos
- ▶ **Consideraciones a tener en cuenta:**
  - ▶ Todos los arrays deben tener la misma dimensión;
  - ▶ El resultado debe ser un valor escalar;
  - ▶ Puede generar el producto escalar de ambos vectores.




E1									
	A	B	C	D	E				
1	1	2	3		10				
2	3	2	1						
3									



# Funciones de gestión de arrays

👉 **Sintaxis:** `FREQUENCY (data; classes)`

- ▶ **Acción:** Calcula la frecuencia de los elementos del array (histograma).

F2					{=FREQUENCY(A1:C4;E2:E4)}		
	A	B	C	D	E	F	G
1	1	2	3		Criteria	Count	
2	4	5	6		5	5	
3	7	8	9		10	5	
4	10	11	12		>10	2	
5							
6							

# Funciones de gestión de arrays

 **Sintaxis:** `MMULT (arrayA; arrayB)`

- ▶ **Acción:** Se calcula el producto de dos arrays.
- ▶ El número de columnas de los dos arrays debe ser el mismo.

[=MMULT(B1:D4;F1:I3)]												
B	C	D	E	F	G	H	I	J	K	L	M	N
1	2	2		1	2	2	3		11	12	16	21
4	3	4		4	3	4	5		20	25	32	43
1	2	3		1	2	3	4		12	14	19	25
2	3	4							18	21	28	37

# Funciones de gestión de arrays

- ▶ Existen múltiples funciones para matrices;
- ▶ La clave para trabajar con arrays está en definir la forma de trabajar a la hoja de cálculo.
  - ▶ Recordar pulsar **Ctrl+Shift+Enter**!

# Funciones condicionales

# Funciones condicionales

- ▶ La mayoría de las celdas de LibreOffice Calc generan un valor formado por fórmulas.
  - ▶ Si se modifican los datos hay que actualizar toda la tabla.
- ▶ A veces algunas celdas se bloquean en manos de los valores.
- **Ejemplo:** Mostrar el valor absoluto del valor de una celda sin usar **ABS**.
- ▶ **Funciones condicionales** son funciones que adaptan la fórmula a los datos (no sólo su resultado).




# Funciones condicionales

👉 **Sintaxis:** `IF (Condition; AOption; BOption)`

- ▶ **Acción:** Se ejecuta `AOption` en la celda si `Condition` es cierta o `BOption` si no es cierta.
- ▶ **Condition:** Valores que pueden ser tomados como verdaderos o falsos.
- ▶ **AOption y BOption:** cualquier cosa que se pueda escribir en una celda (fórmulas, referencias, valores fijos, etc.).
- ▶ Suele ser normal que aparezcan más `IF` en los argumentos `XOption`.



# Funciones condicionales

- **Ejemplo:** ¿Un número es igual a 7?

D2					=	=IF(B2=7;"seven";"not seven")						
	A	B	C	D	E	F	G	H	I	J	K	
1												
2		7		seven								
3				positive								
4				positive 7								
5				positive and odd								
6												
7												
8												

# Funciones condicionales



- **Ejemplo:** ¿Un número es mayor que 0?

D3				=	=IF(B2>0;"positive";"negative")						
	A	B	C	D	E	F	G	H	I	J	K
1											
2		7		seven							
3				positive							
4				7							
5				positive and odd							
6											
7											
8											





# Funciones condicionales

- **Ejemplo:** Escribe un número, o cero si es negativo.

D4				  =	=IF(B2>0;B2;0)						
	A	B	C	D	E	F	G	H	I	J	K
1											
2		7		seven							
3				positive							
4				positive and odd							
5											
6											
7											
8											

# Funciones condicionales

- **Ejemplo:** ¿Un número positivo y/o par?

D5				  =	=IF(B2>0;IF(ISEVEN(B2);"positive and even";"positive and odd");"negative")						
	A	B	C	D	E	F	G	H	I	J	K
1											
2		7		seven							
3				positive							
4				7							
5				positive and odd							
6											
7											
8											

# Funciones condicionales

- 👉 **Sintaxis:** SUMIF (Range; Criteria)
- ▶ **Acción:** Sumará los valores comprendidos en Range si cumplen la condición definida en Criteria.

E1							
	A	B	C	D	E	F	
1	1	2	3		10		
2	4	5	6				
3	7	8	9				
4	10	11	12				
5							

# Funciones condicionales

 **Sintaxis:** SUMIF (Range; Criteria; SumRange)

- ▶ **Acción:** Sumará los valores comprendidos en SumRange si los valores en Range cumplen la condición definida en Criteria.
- ▶ Identico al anterior, pero el criterio se hace en otro tramo de celda.

=SUMIF(E1:G4;"=1";A1:C4)									
	A	B	C	D	E	F	G	H	I
1	1	2	3		1	0	1		45
2	4	5	6		0	0	1		
3	7	8	9		1	1	1		
4	10	11	12		0	1	0		
5									

# Funciones de búsqueda

# Búsqueda de datos

- ▶ A menudo utilizan hojas de cálculo con datos válidos derivados de procesos de negocio.
- ▶ Entonces, los usuarios deben leer la información desde otras tablas.
- ▶ Esto se hace utilizando funciones de búsqueda que ayudan a estudiar los arrays.

# Funciones de búsqueda

- ▶ Las funciones más importantes son `VLookUp` y `HLookUp`.
- ▶ Estas funciones exploran columnas y líneas viendo si se cumple el criterio definido.
- ▶ Pueden tener 3 o 4 parámetros. La última se puede poner por defecto.
  - ▶ Regla de parámetros: evitar parámetros por defecto.

# Funciones de búsqueda

👉 **Sintaxis:** `VLookUp(Criteria;Array;Index;Sorted)`

- ▶ **Acción:** Obtiene los datos de un `array` si su primera columna es igual a `Criteria`.
- ▶ **Prámetros:**
  1. `Criteria`: el valor que buscamos (siempre en la primera columna de `Array`);
  2. `Array`: rango de celdas que contiene los datos;
  3. Una vez se ha encontrado `Criteria` en el `Array`, devuelve el valor de la columna de `Array` definida por `Index`.
  4. Finalmente `Sorted` define el tipo de lista para valores concretos 0 y para intervalos 1 (se pone como true/false en Excel).



# Funciones de búsqueda

- **Ejemplo:** Buscar un identificador (Id) en una tabla.

F2							
=VLOOKUP(F1;A2:C7;2;FALSE())							
	A	B	C	D	E	F	G
1	ID	Name	Telephone		Which ID?	8	
2	1	John	664124		Name:	Wayne	
3	3	Ringo	234234		Telephone	854354	
4	35	Paul	2312334				
5	6	George	5677				
6	7	Alan	4473974				
7	8	Wayne	854354				

# Funciones de búsqueda

- **Ejemplo:** Busca el valor cercano.

E2					
=VLOOKUP(E1;A2:B8;2;TRUE())					
	A	B	C	D	E
1	Price	Discount		Which Price?	-1 €
2	0,00 €	0%		Discount	#N/A
3	10,00 €	1%			
4	25,00 €	2%			
5	50,00 €	5%			
6	100,00 €	10%			
7	500,00 €	15%			
8	1.000,00 €	25%			

# Funciones de búsqueda

- ▶ También existe `HLookUp`, muy parecido a `VLookUp`.
- ▶ Busca información de forma horizontal (desde la primera línea).
- ▶ El tercer parámetro nos especificará la **fila** de la que obtendremos la información (en lugar de la columna como en `VLookUp`).

# Funciones de indexación

# Funciones de indexación

- ▶ Indexación significa el acceso a un valor específico de un rango.
- ▶ Simple y a menudo muy útil.

# Funciones de indexación

👉 **Sintaxis:** `INDEX(Array;Line;Column)`

- ▶ **Acción:** Devuelve el valor en la línea `Line` y columna `Column` de los valores comprendidos por `Array`.
- ▶ El número de línea y columna se pueden calcular a partir de los datos.

# Funciones de indexación

- **Ejemplo:** Indexación con valores enteros.

G4							
=INDEX(B2:D5,G2,G3)							
	A	B	C	D	E	F	G
1							
2		1	2	3		Row:	5
3		4	5	6		Col:	3
4		7	8	9		Result:	Err:502
5		10	11	12			
6							

# Funciones de indexación

- **Ejemplo:** Construyendo texto.

F10						=F8&" "&F4&" of "&F7		
	A	B	C	D	E	F	G	
1								
2		Month	Week day		Date:	06/07/16		
3		January	Monday		Month:	6		
4		February	Tuesday		Month day:	7		
5		March	Wednesday		Week day:	3		
6		April	Thursday					
7		May	Friday		Month name:	June		
8		June	Saturday		Day name:	Wednesday		
9		July	Sunday					
10		August			Result:	Wednesday, 7 of June		
11		September						
12		October						
13		November						
14		December						
15								





# Funciones de indexación

- 👉 **Sintaxia:** `CHOOSE (Index; arg1; arg2; ...; argN)`
- ▶ **Acción:** Devuelve el elemento en la posición `Index`, de la subsecuencia creada por sus argumentos.
- ▶ Los argumentos pueden ser rangos o celdas, pero no todos los argumentos pueden entrar como rangos de celda. Es decir, `arg1; arg2; ...; argN` no puede ser un rango.

# Funciones de indexación

- **Ejemplo:** Construyendo un dato de tipo texto (usando CHOOSE).

F8				=	=CHOOSE(F5,C3,C4,C5,C6,C7,C8,C9)			
	A	B	C	D	E	F	G	
1								
2		Month	Week day		Date:	06/07/16		
3		January	Monday		Month:	6		
4		February	Tuesday		Month day:	7		
5		March	Wednesday		Week day:	3		
6		April	Thursday					
7		May	Friday		Month name:	June		
8		June	Saturday		Day name:	Wednesday		
9		July	Sunday					
10		August			Result:	Wednesday, 7 of June		
11		September						
12		October						
13		November						
14		December						
15								

# Funciones de indexación

- **Ejemplo:** Sumando columnas.

G4		fx		Σ		=		=SUM(CHOOSE(G3,B4:B8,C4:C5,D4:D9))	
	A	B	C	D	E	F	G		
1									
2		Columns							
3		1	2	3		Column?	2		
4		2	3	69		Result:	54		
5		7	51	5					
6		3		4					
7		34		5					
8		3		43					
9				8					
10									

# Repaso

# Funciones aprendidas

- ▶ Funciones condicionales.
  - ▶ IF, SUMIF, COUNTIF, ...
- ▶ Funciones de búsqueda.
  - ▶ VLookUp, HLookUp, ...
- ▶ Funciones de indexación.
  - ▶ INDEX
  - ▶ INDEX+MATCH