

Programación

Clase 09

U

4
Arreglos
Matrices



Objetivo

Comprender qué es un arreglo bidimensional y cómo utilizarlo.

Utilizar arreglos unidimensionales y bidimensionales en conjunto para resolver problemas de mayor complejidad.

Aplicar los conceptos para la resolución de problemas.

¿Qué sabemos hasta ahora?

- Variables!
 - Es la base de todo
- Condicional (if/else/elif)
 - En todas sus formas
- Ciclos (for/while)
- Listas!

¿Es todo lo que necesitamos?

Un ejemplo

Leyendo desde un archivo, en que cada línea contiene un nombre y una edad, determinar la **cantidad de nombres distintos** presentes en el archivo (**nombres sin duplicar**)

Listas

Veamos este problema

Queremos evaluar el riesgo que tienen las personas al vivir cerca del mar.

Para eso, tengo un archivo en que cada línea contiene el rut de una persona, su nombre, la ciudad donde vive y la distancia de su casa al borde del mar.

Necesito procesar el archivo y determinar la cantidad total de personas que viven a menos de 30 metros del mar.

¿Se puede hacer?

Continuando...

El programa también debe indicar, para cada ciudad, la cantidad de personas que viven a menos de 30 metros del mar.

¿Se puede hacer?

¡Si! Resuélvelo en tu casa

Continuando la continuación...

Dependiendo de la distancia al mar, las personas se clasifican "**en riesgo**" (<20m) "**en precaución**" (entre 20 y 50m) y "**fuera de peligro**" (>50m)

Usando apropiadamente un montón de listas puedes responder estas preguntas.

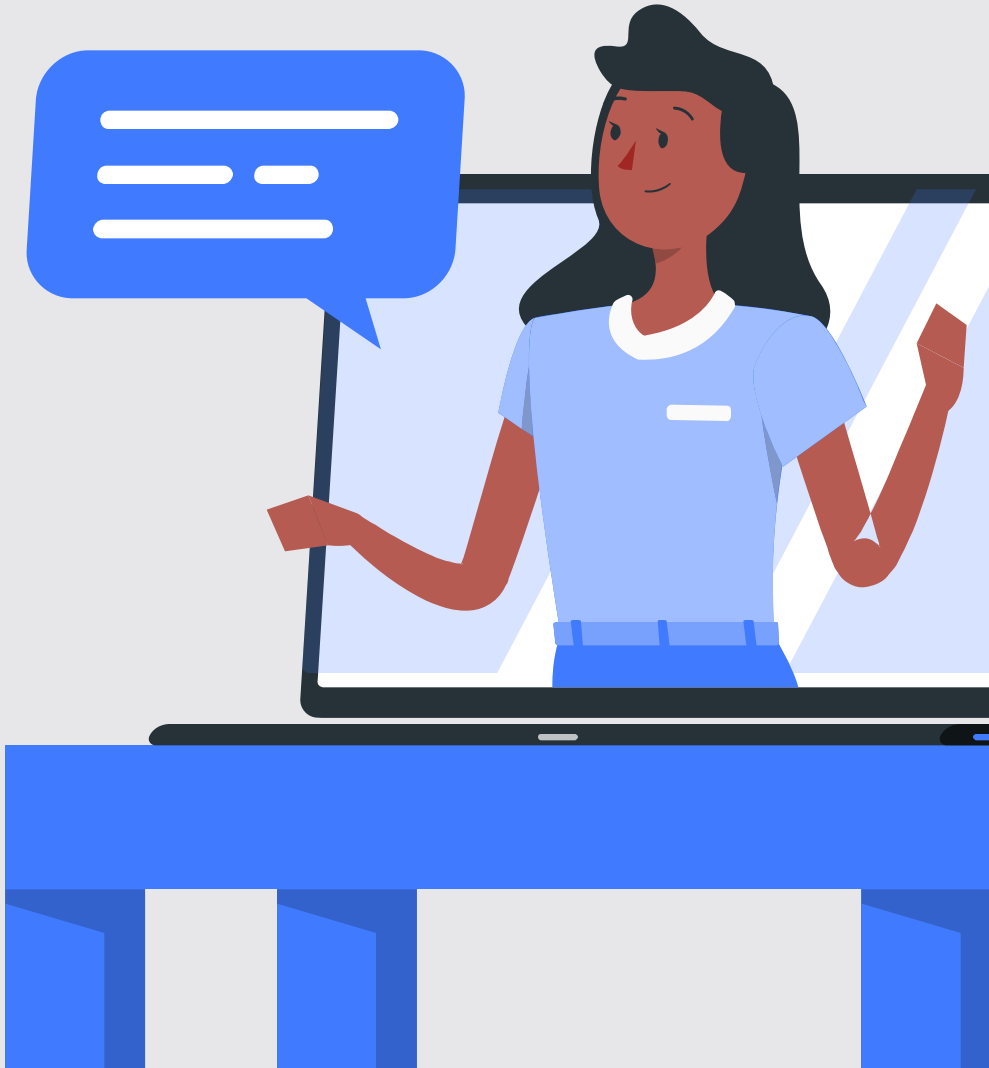
Además, **Pero hay una manera más fácil** de hacerlo "en riesgo" de perderlo.

¿Se puede hacer?

Clase 09: Matrices

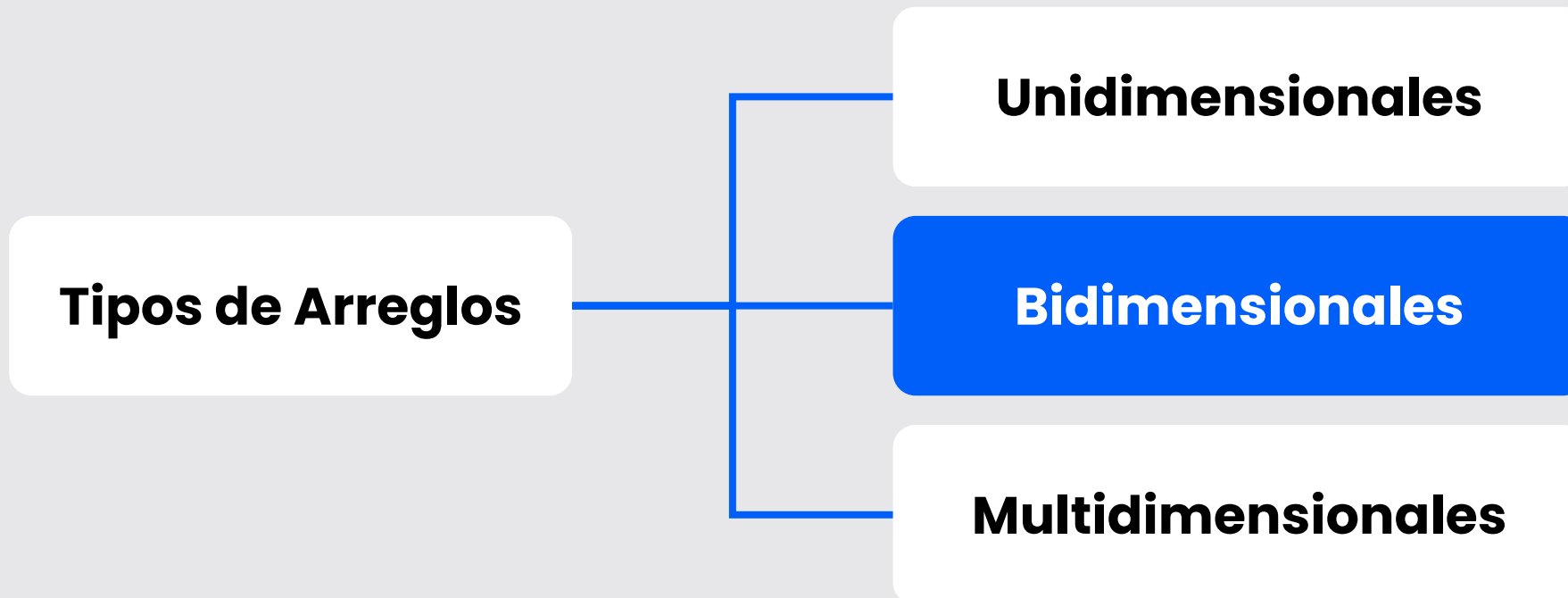
Matrices

La segunda estructura de datos

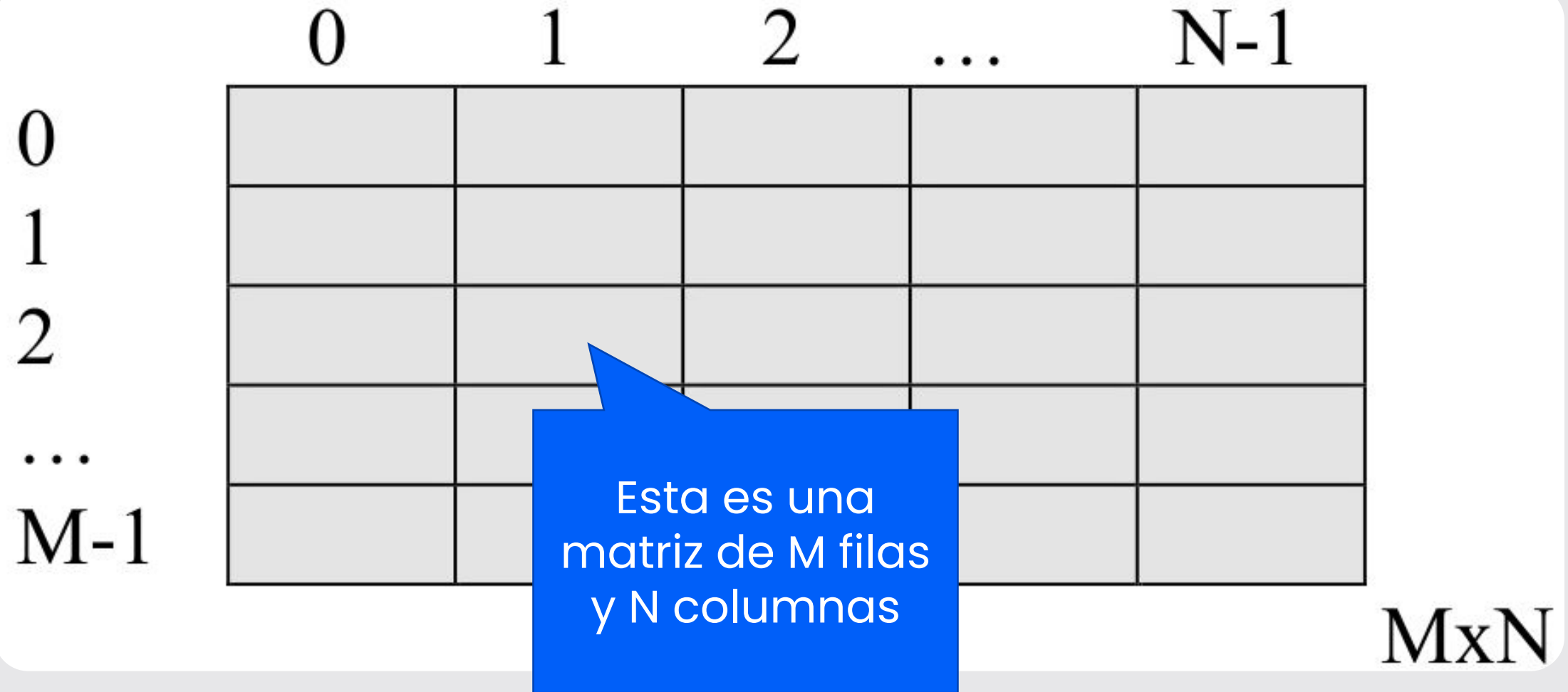


Arreglos

Un **arreglo** consiste en un número **finito** y **ordenado** de elementos, bajo un **nombre común** para todos ellos.



Arreglos bidimensionales



Trabajando con matrices

Para trabajar con matrices necesitamos:

- Crearla
 - De forma que exista la variable
- Ingresar elementos a la matriz
 - Estos son los datos que vamos a procesar para solucionar el problema
- Obtener elementos de la matriz
 - Dependiendo del problema, necesitaremos leer los datos para procesarlos, escribirlos, etc.

Un pequeño AVISO

```
# Vamos a "importar" la biblioteca numpy.  
# Desde ahora, usando el prefijo np.  
import numpy as np  
# Usamos la función "zeros" para crear una matriz de  
# 3 filas y 4 columnas  
matriz = np.zeros([3,4])  
# Ponemos el valor 10 en la primera posición  
matriz[0][0] = 10  
# Ponemos el valor 20 en la última posición  
matriz[2][3] = 20  
# Escribimos la matriz por pantalla  
print(matriz)  
# Escribimos un elemento en particular de la matriz  
print(matriz[2][2])
```

```
[[ 10.   0.   0.   0.]  
 [  0.   0.   0.   0.]  
 [  0.   0.   0.  20.]]
```

¿Cuál es la diferencia?

```
import numpy as np
# Usamos la función "zeros" para
# crear una matriz de
# 3 filas y 4 columnas
matriz = np.zeros(3, 4)

cont = 1
```

Acá estamos
recorriendo
primero por
columna

```
for fila in range(0, 3):
```

```
    for col
```

```
        matriz[fila][col] = cont
        cont += 1
```

```
print(matriz)
```

```
import numpy as np
# Usamos la función "zeros" para
# crear una matriz de
# 3 filas y 4 columnas
matriz = np.zeros(3, 4)

cont = 1
```

Acá estamos
recorriendo
primero por fila

```
for col in range(0, 4):
```

```
    for fila
```

```
        matriz[fila][col] = cont
        cont += 1
```

```
print(matriz)
```

Un ejemplo 1/3

MATRICULA

	1	2	3	4	5	6	7	8
1	35	30	32	32	34	35	34	28
2	40	33	40	37	36	39	40	29
3	25	23	26	21	24	24	25	15
4	33	33	33	32	34	30	32	20
5	45	44	45	44	43	40	44	33
6	24	20	22	22	24	25	24	12

Asignatura 7

Grupo 6

Hay 21 alumnos del grupo 3 en la asignatura 4

Un ejemplo 2/3

Construya un programa que lea el archivo, pueble la matriz e imprima por pantalla el total de estudiantes matriculados por cada asignatura

Las asignaturas son:

1:Matemáticas, **2:**Lengua Española, **3:**Formación Humanística,
4:Ciencias Naturales, **5:**Inglés, **6:**Informática Básica, **7:**Estructura de la Información, **8:**Metodología de la Programación

Un ejemplo 3/3

Para solucionar este ejemplo, lo que hay
abajo por cada asignatura

Una vez leídos los datos desde el archivo, la parte importante es:

MATRICULA								Asignatura
	1	2	3	4	5	6	7	8
1	35	30	32	32	34	35	34	28
2	40	33	40	37	36	39	40	29
3	25	23	26	21	24	24	25	15
4	33	33	33	32	34	30	32	20
5	45	44	45	44	43	40	44	33
6	24	20	22	22	24	25	24	12

Hay 21 alumnos del grupo 3 en la asignatura

```
def escribirEstudiantesPorAsignatura(matriz):  
    for c in range(8):  
        suma = 0  
        for f in range(6):  
            suma = suma + matriz[f][c]  
        print(obtenerNombreAsignatura(c), suma)
```


Clarificando las cosas

Muchas veces sucede que **NO** sabemos la cantidad de columnas y/o filas que necesitamos almacenar

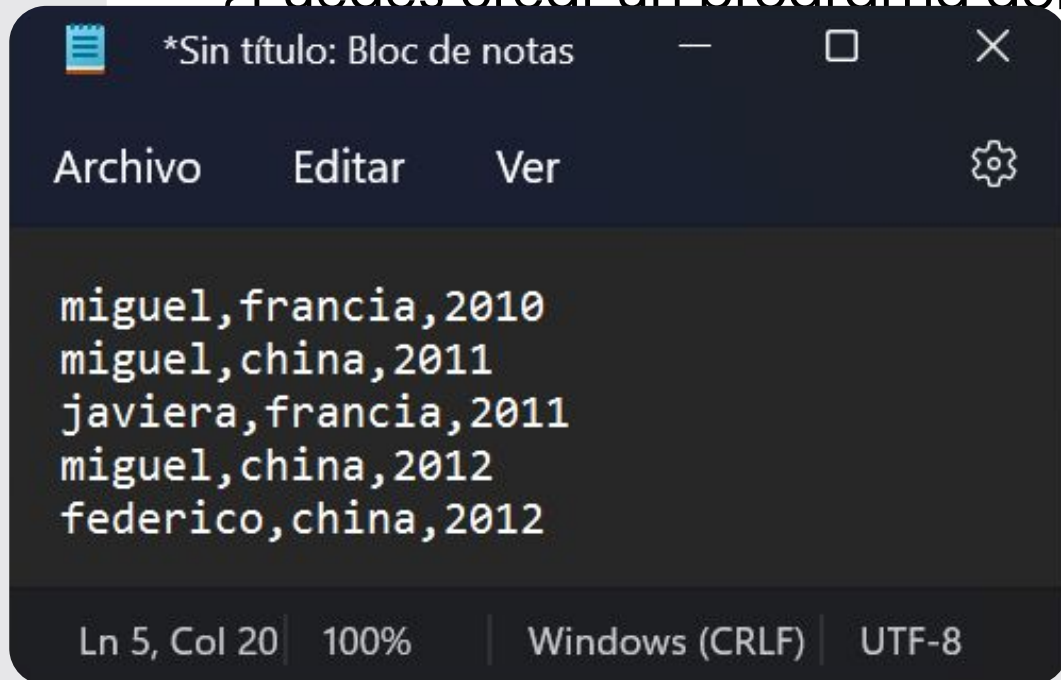
Ni tampoco sabemos a priori los nombres de lo que almacenaremos (como en el ejemplo anterior, ya sabíamos el nombre de las asignaturas).

Otro ejemplo

Tú y tus amigos son fanáticos de los viajes

Han visitado muchos países, en diferentes años

¿Puedes crear un programa donde tus amigos puedan ingresar todos los países que han visitado, y te diga cuál es el país favorito (o los países) de tus amigos lo visitaron?



```
*Sin título: Bloc de notas

Archivo  Editar  Ver

miguel,francia,2010
miguel,china,2011
javiera,francia,2011
miguel,china,2012
federico,china,2012

Ln 5, Col 20 | 100% | Windows (CRLF) | UTF-8
```

Países más visitados:

China:

- Miguel
- Federico

Propuesta

En la intersección
se almacenará el
dato que
corresponde al
nombre y país

The diagram shows a grid of blue squares representing data. A red outline highlights a specific region within the grid. The text 'países' is written vertically on the left, and 'nombres' is written horizontally at the top.

¿Cómo funcionaría en el pr

```
# una lista inicialmente vacía  
países = [ ]
```

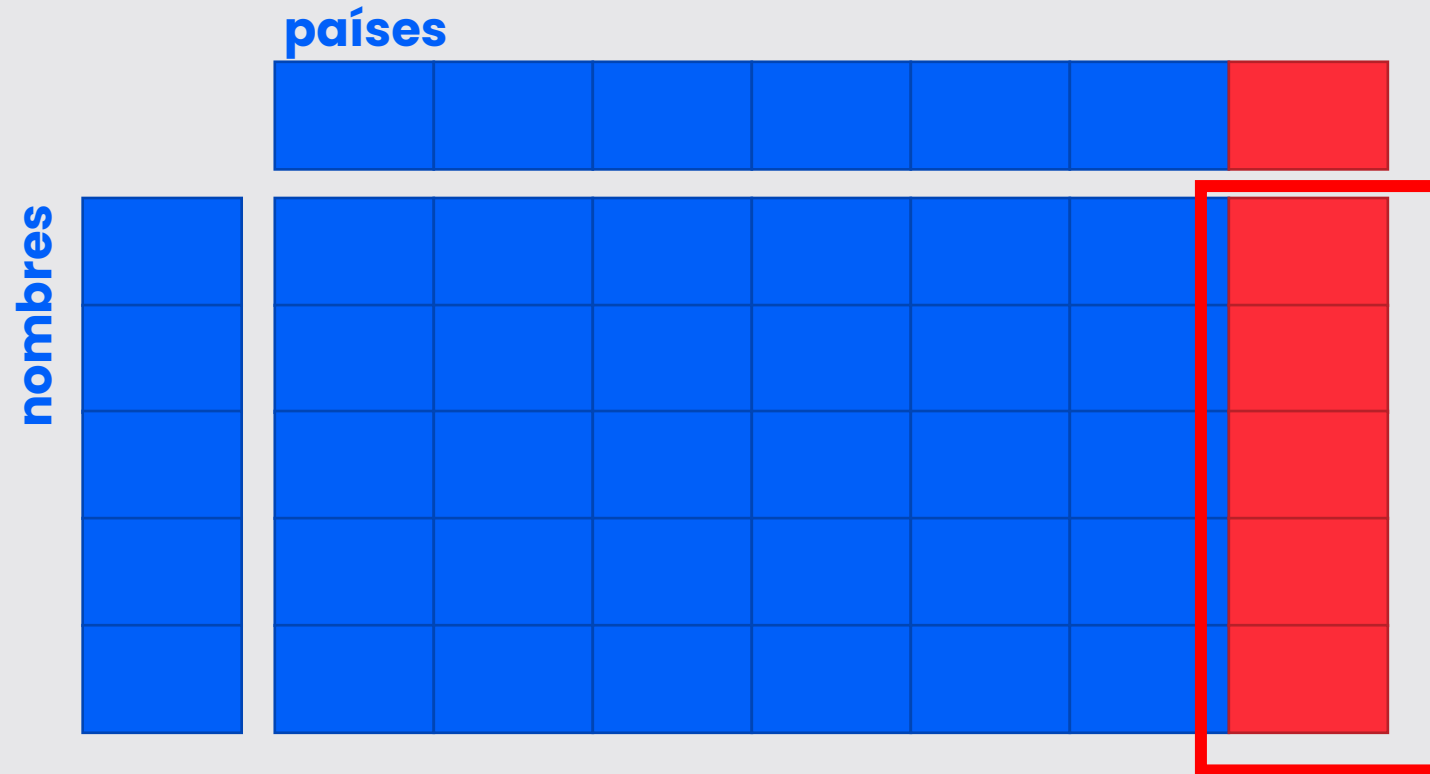
```
nombres = [ ]
```

nombres

países

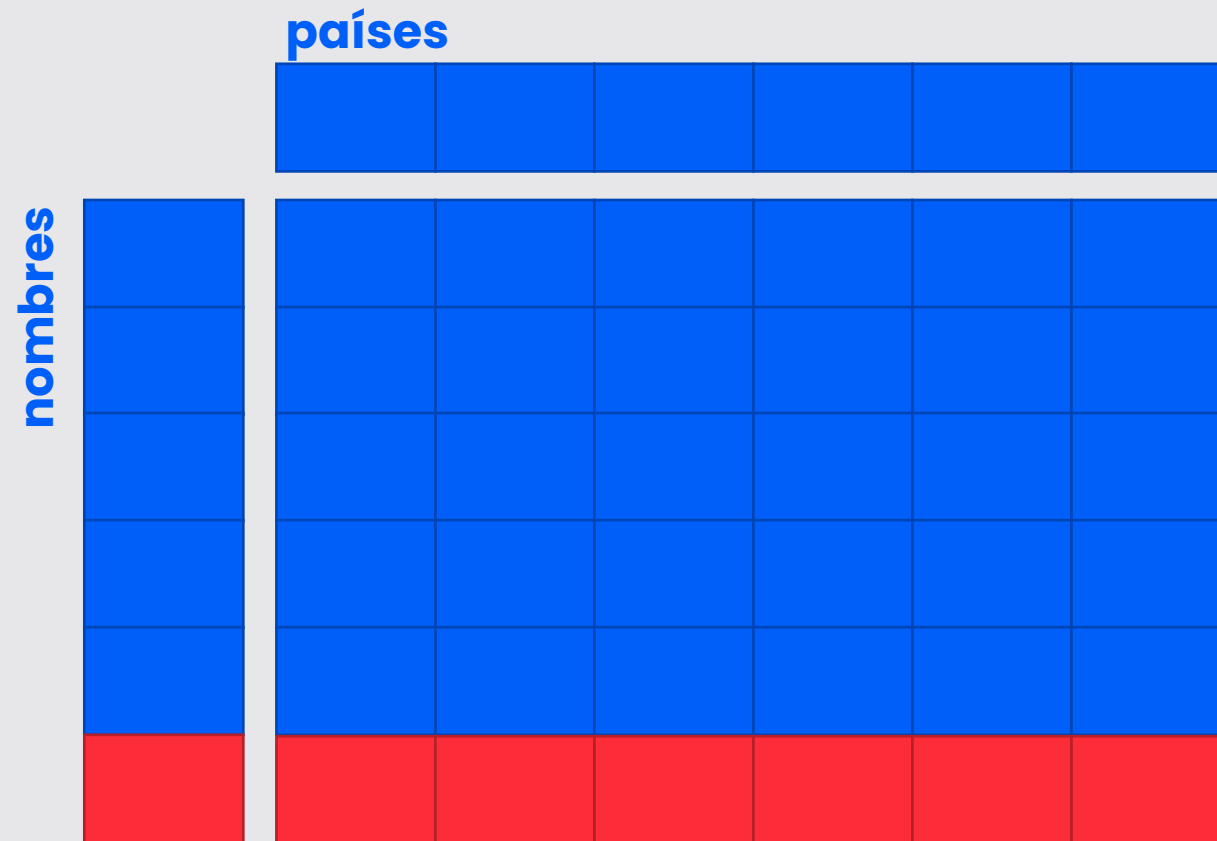
Una matriz
donde
almacenarás
los datos

Pero, ¿qué pasa con la matriz?



Antes de comenzar a leer, vamos a crear suficiente espacio para almacenar todos los datos que necesitamos

Algo equivalente pasa con los nombres



Paso a paso

Comienza con las listas de países y nombres vacías, y la matriz de contadores suficientemente grande para almacenar todos los datos. Esta matriz está llena de ceros.

		países					
nombres							
		0	0	0	0	0	0
		0	0	0	0	0	0
		0	0	0	0	0	0
		0	0	0	0	0	0
		0	0	0	0	0	0

Paso a paso

El programa lee Miguel, Francia, 2010

¿Existe Miguel?

No, así que lo agregas al final de la lista

(La lista está actualmente vacía, así que al agregarlo al final, queda en primer lugar)

¿Existe Francia?

No, así que lo agregas al final de la lista

		países					
nombres	M	F					
		1	0	0	0	0	0
		0	0	0	0	0	0
		0	0	0	0	0	0
		0	0	0	0	0	0
		0	0	0	0	0	0

Miguel visitó Francia, así que corresponde dejar registrado el hecho

Paso a paso

El programa lee Miguel, China, 2011

¿Existe China?

No, así que lo
agregas al final
de la lista

¿Existe Miguel?

Si, así no
agregas nada

	países					
	F	Ch				
nombres	M	1	1	0	0	
		0	0	0	0	0
		0	0	0	0	0
		0	0	0	0	0
		0	0	0	0	0

Miguel visitó
China, así que
corresponde
dejar registrado
el hecho

Paso a paso

El programa lee Javiera, Francia, 2011

¿Existe Francia?

Si. No tienes nada que hacer

países

F	Ch				
---	----	--	--	--	--

nombres

M	1	1	0	0	0	0
J	1	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0

¿Existe Javiera?

No, así que la agregas al final de la lista

Javiera visitó Francia, así que corresponde dejar registrado el hecho

Paso a paso

El programa lee Miguel, China, 2011

¿Existe
China?

¿Existe
Miguel?

nombres

países

	países					
	F	Ch				
M	1	2				
J	1	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0

Lo correcto es
INCREMENTAR en
uno el contador
de visitas de esa
persona en ese
país

Paso a paso

El programa lee Federico, China, 2012

¿Existe
China?

¿Existe
Federico?

nombres

países

	países					
	F	Ch				
M	1	2	0	0	0	0
J	1	0	0	0	0	0
F	0	1	0	0	0	0
	0	0	0	0	0	0
	0	0	0	0	0	0

Federico en
China

Paso a paso

Finalizó el ingreso de datos

¿Cuál es el país más visitado?

Tienes que sumar hacia abajo, por columna

Te convi...
para ir g...
por país

Una vez que
tengas las suma
por cada país,
puedes determinar
el mayor

		países			
		F	Ch		
nombres	M	1	2	0	0
	J	1	0	0	0
	F	0	1	0	0
		0	0	0	0
		0	0	0	0
		0	0	0	0
		2	3	0	0

¡El
mayor!

Paso a paso

Ya sabes cuál país fue el más visitado

Ahora, necesitas recorrer esa lista, e informar los nombres de las personas que visitaron dicho país

		países					
		F	Ch				
nombres	M	1	2	0	0	0	0
	J	1	0	0	0	0	0
	F	0	1	0	0	0	0
		0	0		0	0	0
		0	0	0	0	0	0
		2	3	0			0

Un CERO significa que la persona NO visitó ese país

¿Y cómo se haría en Python?

Tienes que abrir el archivo

Tienes que leer línea a línea
hasta que el archivo se termine

Tienes que separar cada parte
de la línea

**¡Todo esto ya sabes cómo
hacerlo!**



¿Y cómo se haría en Python?

Tienes que preguntar si el país existe o no. Si es que no existe, hay que agregarlo al final.

```
# Este código estará dentro del ciclo  
# de lectura:  
partes = linea.split()  
pais = partes[1]  
  
columna = buscarAgregar(paises, pais)
```

```
# Esta función retorna el índice en que el  
# elemento está en la lista.  
# Si es que el elemento no está en la lista, lo  
# agrega al final. En este caso, igual retorna  
# el índice donde se encuentra el elemento
```

```
def buscarAgregar(lista, elemento):  
    if elemento not in lista:  
        lista.append(elemento)  
    return lista.index(elemento)
```

¿Y cómo se haría en Python?

Tienes que preguntar si el nombre existe o no. Si es que no existe, hay que agregarlo al final.

```
partes = linea.split()  
nombre = partes[0]  
  
fila = buscarAgregar(nombres, nombre)
```

¿Y cómo se haría en Python?

Independiente de si el nombre y país existen o no, hay que incrementar en uno la intersección en la matriz.

```
# Después que hemos procesado el nombre y el país  
# (y por lo tanto sabemos la fila y la columna  
# en la matriz), incrementamos el contador en la  
# intersección indicada:
```

```
matriz[fila][columna] = matriz[fila][columna] + 1
```

En resumen

```
def buscarAgregar(lista, elemento):  
    if not elemento in lista:  
        lista.append(elemento)  
    return lista.index(elemento)
```

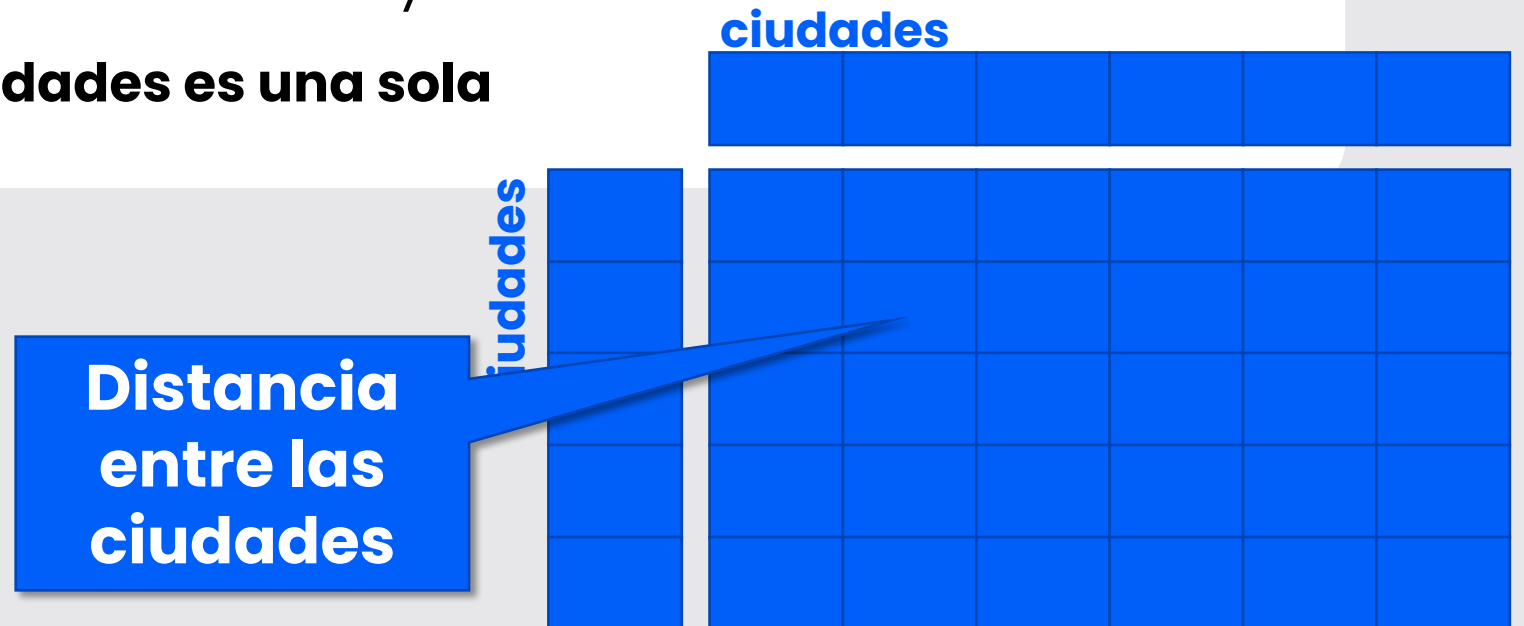
```
países = []  
nombres = []  
matriz = np.zeros([100,100])  
  
arch = open("archivo.txt", "r")  
línea = arch.readline().strip()  
while línea != "":  
    partes = línea.split(",")  
    nombre = partes[0]  
    país = partes[1]  
    año = int(partes[2])  
  
    columna = buscarAgregar(países, país)  
    fila = buscarAgregar(nombres, nombre)  
    matriz[fila][columna] =  
matriz[fila][columna] + 1  
  
    línea = arch.readline().strip()
```

Matriz de Adyacencia

Algo muy útil que se puede almacenar dentro de una matriz son las distancias entre lugares

En este caso, el valor almacenado en la intersección representa la distancia entre las ciudades de la fila y columna

OJO, que la lista de ciudades es una sola



Recapitulando...

¿Qué es una matriz?

Un arreglo de tipo **bidimensional** que nos permite almacenar datos de manera estructurada.

¿De qué nos sirve utilizarlas por sobre las listas?



Trabajo autónomo mínimo

Revisar capítulo 9 libro guía.

Resolver ejercicios 6, 7, 8, 9.