

# Algoritmos y Programación I

## FIUBA

Clase de Corte de Control  
Python

Pablo Guarna  
2do. cuatrimestre 2021

1

## Corte de Control

Es una estructura de programación usada para recorrer archivos secuenciales ordenados por uno de sus campos, la cual permite detectar los cortes en esa secuencia de ordenamiento.

Campo
1
1
1
7
7
9
9
9
9
20

2

## Corte de Control con Totales

Es una estructura de programación usada para recorrer archivos secuenciales ordenados por uno de sus campos, la cual permite detectar los cortes en esa secuencia de ordenamiento.

Clave	Importe	Total
1	10	
1	10	
1	10	30
7	5	
7	5	10
9	5	
9	5	
9	1	
9	1	12
20	20	20

3

## Si quiero recorrer un archivo

```
with open('movi.txt', 'rt') as movs:
    for linea in movs:
        print('linea', linea)
```

linea 001,10

linea 001,10

linea 001,10

linea 007,5

linea 007,5

linea 009,5

linea 009,5

linea 009,1

linea 009,1

linea 020,20

4

## Si quiero recorrer con WHILE

```
def leer_info(ejs, default):
    linea = ejs.readline()
    return linea if linea else default

with open('movimientos.txt', 'rt') as movs:
    linea = leer_info(movs, '999999,0')
    nro_cta, importe = linea.rstrip('\n').split(',')
    max = '999999'
    print(nro_cta, importe)
    total = 0
    while nro_cta != max:
        cta_ant = nro_cta
        tot_cta = 0
        print('cta', cta_ant)
        print('importe', importe)
        while nro_cta == cta_ant and nro_cta != max:
            print(importe)
            tot_cta += int(importe)
            print(nro_cta, importe)
            linea = leer_info(movs, '999999,0')
            nro_cta, importe = linea.rstrip().split(',')
        print('cierre')
        print('El total de la cuenta {} es {}'.format(cta_ant, tot_cta))
        total += tot_cta
    print('Total Gral:', total)
```

linea 001,10

linea 001,10

linea 007,5

linea 007,5

linea 009,5

linea 009,5

linea 009,1

linea 009,1

linea 020,20

linea

5

## Calculando total general

```
def leer_info(movs):
    linea = movs.readline()
    if linea:
        registro = linea.rstrip('\n').split(',')
    else:
        registro = ['', '']
    return registro

with open('movi.txt', 'rt') as movs:
    lista = leer_info(movs)
    total = 0
    while lista[0]:
        total += int(lista[1])
        lista = leer_info(movs)
        print('lista', lista)
        print('Total Gral:', total)
```

```
lista ['001', '10']
Total Gral: 10
lista ['001', '10']
Total Gral: 20
lista ['007', '5']
Total Gral: 30
lista ['007', '5']
Total Gral: 35
lista ['009', '5']
Total Gral: 40
lista ['009', '5']
Total Gral: 45
lista ['009', '1']
Total Gral: 50
lista ['009', '1']
Total Gral: 51
lista ['020', '20']
Total Gral: 52
lista ['', '']
Total Gral: 72
```

6

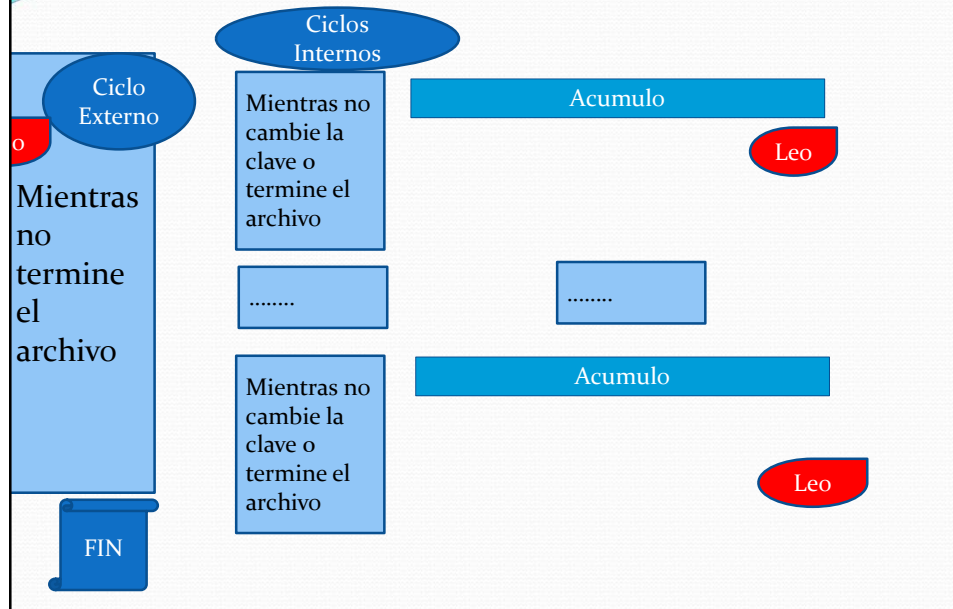
## Corte de Control con Totales

Es una estructura de programación usada para recorrer archivos secuenciales ordenados por uno de sus campos, la cual permite detectar los cortes en esa secuencia de ordenamiento.

Clave	Importe	Total
1	10	
1	10	
1	10	30
7	5	
7	5	10
9	5	
9	5	
9	1	
9	1	12
20	20	20

7

### ESQUEMA DE LA ESTRUCTURA de Corte de Control



8

```
def leer_info(movs):
    linea = movs.readline()
    if linea:
        registro= linea.rstrip('\n').split(',')
    else:
        registro=['','']
    return registro

with open('movi.txt','rt') as movs:
    lista=leer_info(movs)
    total_gral = 0
    while lista[0]:
        total_cta=0
        cta_ant=lista[0]
        while lista[0] == cta_ant:
            total_cta+=int(lista[1])
            total_gral += int(lista[1])
            lista=leer_info(movs)
        print(cta_ant, 'el toal de la cta es:',total_cta)
    print('Total Gral ', total_gral)
```

```
001 el toal de la cta es: 30
007 el toal de la cta es: 10
009 el toal de la cta es: 12
020 el toal de la cta es: 20
Total Gral 72
```

9

```
def leer_info(ejs, default):
    linea = ejs.readline()
    return linea if linea else default
```

10

```

with open('movimientos.txt', 'rt') as movs:
    linea = leer_info(movs, '999999,o')
    nro_cta, importe = linea.rstrip().split(',')
    max = '999999'
    print (nro_cta, importe)
    total = 0
    while nro_cta != max:
        cta_ant = nro_cta
        tot_cta = 0
        print ('cta',cta_ant)
        print ('importe',importe)
        while nro_cta == cta_ant and nro_cta != max:
            print (importe)
            tot_cta+=int(importe)
            print (nro_cta, importe)
            linea = leer_info(movs, '999999,o')
            nro_cta, importe = linea.rstrip().split(',')
            print ('cierre')
            print ('El total de la cuenta {} es {}'.format(cta_ant,
tot_cta))
            total += tot_cta
        print ("Total Gral:", total)

```

11

### Ejercicio Propuesto

empresa procesa la información de sus ventas a sus clientes  
Para cada venta tiene la siguiente información:

CUIL

Fecha de venta

Código de Producto

Cantidad de unidades

Precio unitario

Las ventas se guardan en un archivo de texto CSV ordenado  
por CUIL y código de producto de forma ascendente

**Desean tener en 1 archivo secuenciales ordenado por  
CUIL los totales en unidades y en pesos vendidos  
por CUIL vendidos**

**Caso B**

**Obtener los totales por CUIL y Producto**

12



## Resumiendo

- El archivo tiene que estar ordenada con el mismo criterio que los cortes de control
- Los cortes s realizan del mas externo al mas interno
- Cada corte chequea las condiciones del ciclo anterior y agrega una nueva condición
- La lectura esta solo en el interior del ciclo mas interno
- Al finalizar cada ciclo se muestran los totales
- Antes de entrar en cada ciclo se inicializa el total

13



14

## MERGE



15

## REPASO Corte de Control

Recorrido  
de UN  
archivo

While hasta  
el final

Recorrido  
Con 1 Corte  
de control

2While  
anidados

Externo hasta  
fin de archivo

Interno  
Hasta cambio  
de clave



16



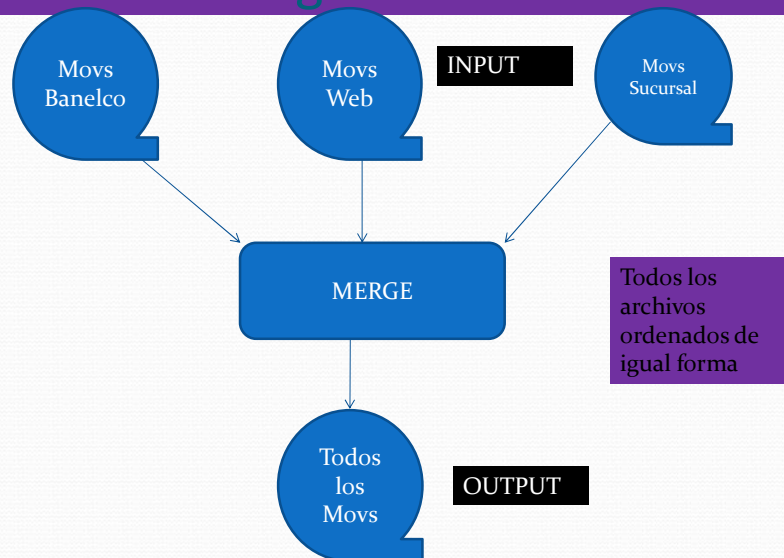
## REPASO CC

```
def leer_info(movs):
    linea = movs.readline()
    if linea:
        registro= linea.rstrip('\n').split(',')
    else:
        registro=['','']
    return registro

with open('movi.txt','rt') as movs:
    lista=leer_info(movs)
    total_gral = 0
    while lista[0]:
        total_cta=0
        cta_ant=lista[0]
        while lista[0] == cta_ant:
            total_cta+=int(lista[1])
            total_gral += int(lista[1])
            lista=leer_info(movs)
        print(cta_ant, 'el toaal de la cta es:',total_cta)
    print('Total Gral ', total_gral)
```

17

## Estructura Merge



18

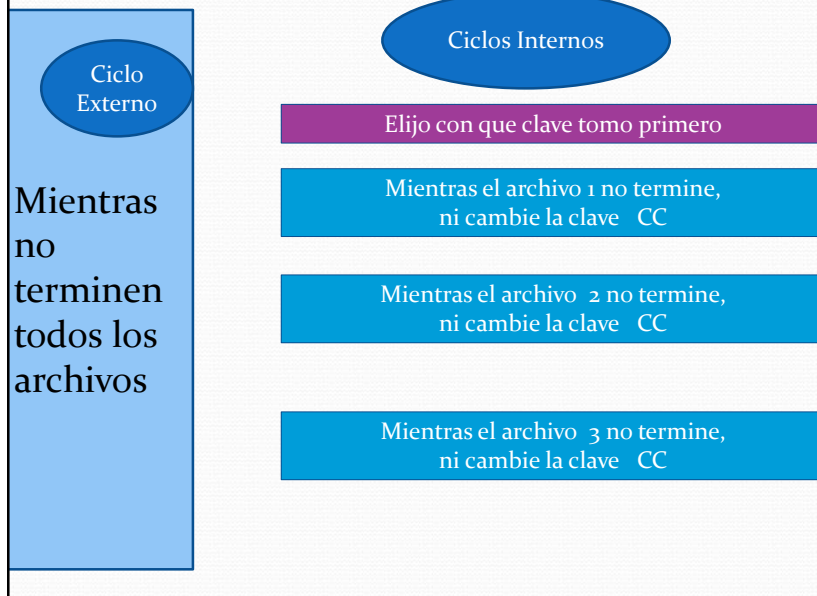
## Merge Ejemplo

SUCURSAL		ONLINE		CAJERO	
cta	imp	cta	imp	cta	imp
4	50	1	100	2	20
4	50	1	100	4	10
5	100	1	200	4	10
		4	100	10	1000
		5	200	10	1000
		5	200	12	500
		10	500		
		10	2000		

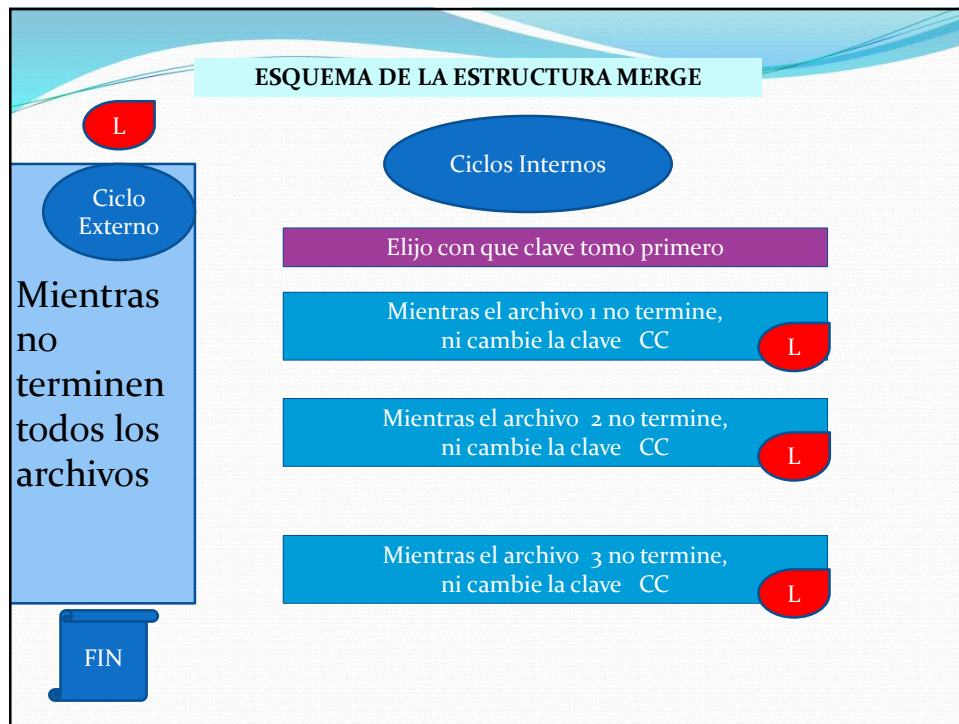
Resultado	
cta	total
1	400
2	20
4	220
5	500
10	4500
12	500

19

### ESQUEMA DE LA ESTRUCTURA MERGE



20



21

```

def leer_info(movs):
    linea = movs.readline()
    if linea:
        registro= linea.rstrip('\n').split(',')
    else:
        registro=['999999','o']
    return registro
  
```

22

```

movsbane= open("MoviBANE.txt", 'rt')
movshb = open("MoviHB.txt", 'rt')
movssuc = open("MoviSUC.txt", 'rt')
max = '999999'
bane_cta, bane_importe = leer_info(movsbane)
hb_cta, hb_importe = leer_info(movshb)
suc_cta, suc_importe = leer_info(movssuc)
total = 0
while bane_cta != max or hb_cta != max or suc_cta != max:
    tot_cta = 0
    men = min(hb_cta, bane_cta, suc_cta) # min es una función de python
    print('cta:', men)
    while bane_cta == men and bane_cta != max:
        tot_cta += int(bane_importe)
        bane_cta, bane_importe = leer_info(movsbane)
    while hb_cta == men and hb_cta != max:
        tot_cta += int(hb_importe)
        hb_cta, hb_importe = leer_info(movshb)
    while suc_cta == men and suc_cta != max:
        tot_cta += int(suc_importe)
        suc_cta, suc_importe = leer_info(movssuc)
    print("Total por cta:", tot_cta)
    total += tot_cta
print("Total Gral:", total)
movsbane.close()
movshb.close()
movssuc.close()

```

23

```

movsbane= open("MoviBANE.txt", 'rt')
movshb = open("MoviHB.txt", 'rt')
movssuc = open("MoviSUC.txt", 'rt')
max = '999999'
bane_cta, bane_importe = leer_info(movsbane)
hb_cta, hb_importe = leer_info(movshb)
suc_cta, suc_importe = leer_info(movssuc)
total = 0
while bane_cta != max or hb_cta != max or suc_cta != max:
    tot_cta = 0
    men = min(hb_cta, bane_cta, suc_cta) # min es una función de python
    print('cta:', men)
    while bane_cta == men and bane_cta != max:
        tot_cta += int(bane_importe)
        bane_cta, bane_importe = leer_info(movsbane)
    while hb_cta == men and hb_cta != max:
        tot_cta += int(hb_importe)
        hb_cta, hb_importe = leer_info(movshb)
    while suc_cta == men and suc_cta != max:
        tot_cta += int(suc_importe)
        suc_cta, suc_importe = leer_info(movssuc)
    print("Total por cta:", tot_cta)
    total += tot_cta
print("Total Gral:", total)
movsbane.close()
movshb.close()
movssuc.close()

```

24

Ej 1. Se cuenta con 3 archivos de texto con información de los movimientos de stock en sus 3 sucursales, que están ordenados **código de producto** en los que se registra el movimiento de los productos con el siguiente formato:

fecha

codigo\_producto

cantidad\_ingresos

cantidad\_ventas

Se pide, 1- generar un archivo con igual formato y orden que los anteriores que contenga un resumen de los registros de los 3 archivos de entrada por Código de producto, y ordenado por dicho código. (un registro por código)

2-

formato:

fecha

codigo\_producto

cantidad\_ingresos

cantidad\_ventas

Ordenado por producto fecha

Se pide sacar el stock para cada producto, fecha, (suponer que se comienza con cero productos).

25

### PARA PENSAR.

Como resolver el mismo ejercicio, si los archivos en lugar de estar ordenados solo por artículo, están ordenados por **fecha y código** y la salida tiene que ser por fecha y código.

Se cuenta con 3 archivos de texto con información de los movimientos de stock en sus 3 sucursales, que están ordenados **por fecha y código** de producto en los que se registra el movimiento de los productos con el siguiente formato:

fecha

codigo\_producto

cantidad\_ingresos

cantidad\_ventas

Se pide, 1- generar un archivo con igual formato y orden que los anteriores que contenga un registro por cada uno de los registros de los 3 archivos de entrada.

26





27