IIC1103 - Sección 2

Clase 17: Archivos

Prof. Francisca Cattan

Universidad Católica de Chile

13 de octubre, 2022

Archivos

Archivos

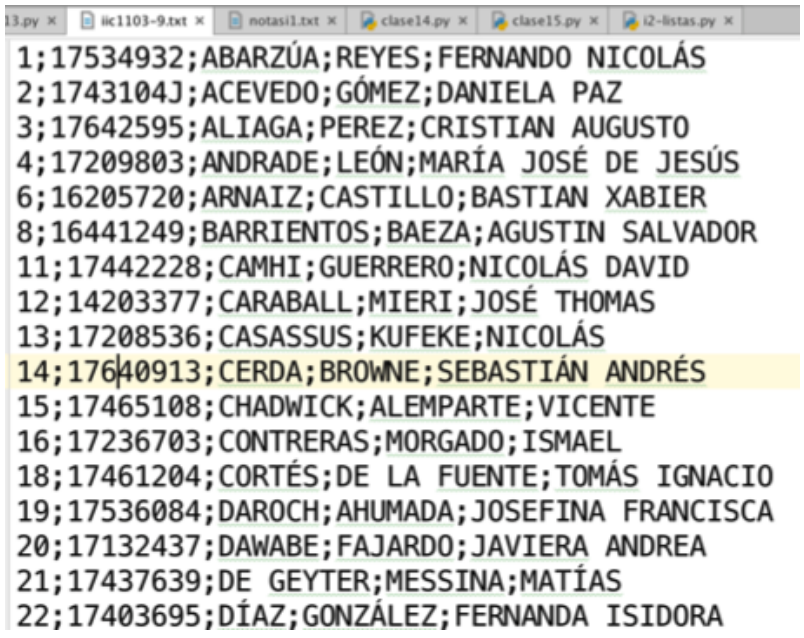
- Lectura de archivos.
- Escritura de archivos.

Archivos

- Hasta ahora, nuestros programas se han ejecutado pero no han podido dejar ningún registro permanente de su ejecución.
- Las variables con las que trabajamos se crean y destruyen con cada ejecución de nuestro programa, sin dejar rastro.
- También, las entradas que reciben nuestros programas hasta ahora han sido siempre entradas del usuario.
- Desde hoy esto va a cambiar. Aprenderemos a guardar informaci ón de nuestros programas en archivos y aprenderemos cómo nuestros programas podrán tomar estos archivos como un nuevo tipo de entrada.

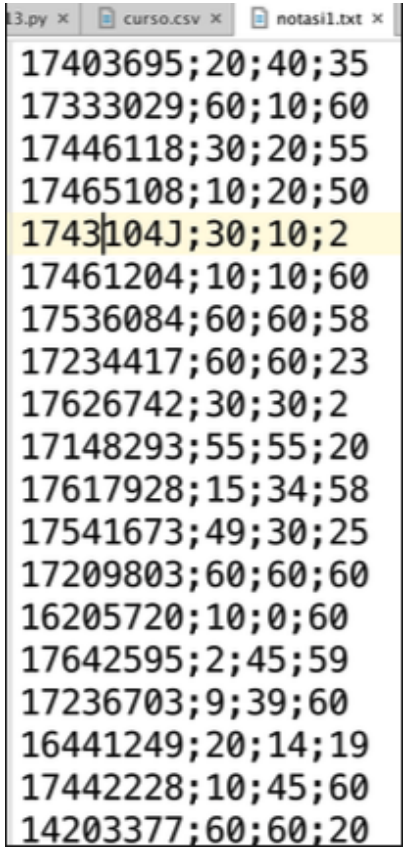
Archivos: lectura de acrchivos

Pensemos juntos: queremos leer los datos de los estudiantes desde un archivo, y luego cargar estos datos en una lista de estudiantes.



```
1;17534932;ABARZÚA;REYES;FERNANDO NICOLÁS
2;1743104J;ACEVEDO;GÓMEZ;DANIELA PAZ
3;17642595;ALIAGA;PEREZ;CRISTIAN AUGUSTO
4;17209803;ANDRADE;LEÓN;MARÍA JOSÉ DE JESÚS
6;16205720;ARNAIZ;CASTILLO;BASTIAN XABIER
8;16441249;BARRIENTOS;BAEZA;AGUSTIN SALVADOR
11;17442228;CAMHI;GUERRERO;NICOLÁS DAVID
12;14203377;CARABALL;MIERI;JOSÉ THOMAS
13;17208536;CASASSUS;KUFEKE;NICOLÁS
14;17640913;CERDA;BROWNE;SEBASTIÁN ANDRÉS
15;17465108;CHADWICK;ALEMPARTE;VICENTE
16;17236703;CONTRERAS;MORGADO;ISMAEL
18;17461204;CORTÉS;DE LA FUENTE;TOMÁS IGNACIO
19;17536084;DAROCH;AHUMADA;JOSEFINA FRANCISCA
20;17132437;DAWABE;FAJARDO;JAVIERA ANDREA
21;17437639;DE GEYTER;MESSINA;MATÍAS
22;17403695;DÍAZ;GONZÁLEZ;FERNANDA ISIDORA
```

curso.csv

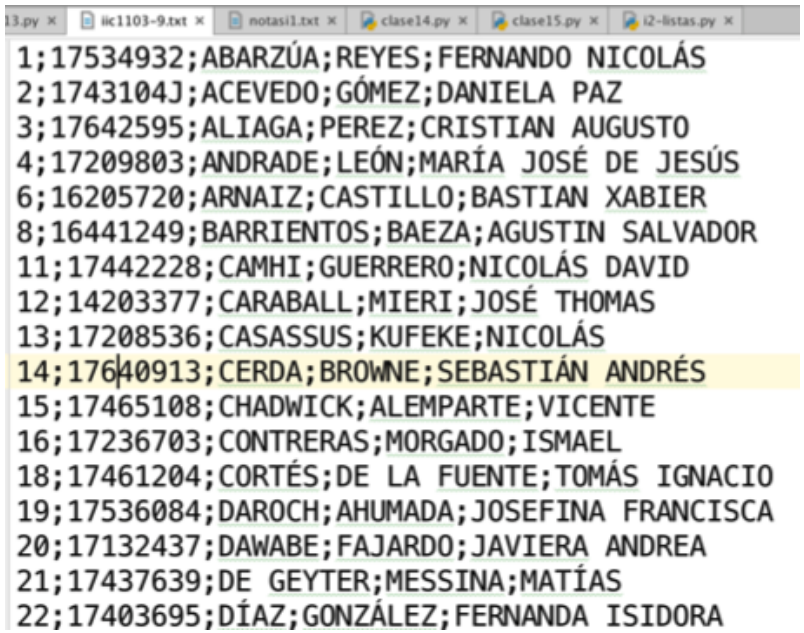


```
17403695;20;40;35
17333029;60;10;60
17446118;30;20;55
17465108;10;20;50
1743104J;30;10;2
17461204;10;10;60
17536084;60;60;58
17234417;60;60;23
17626742;30;30;2
17148293;55;55;20
17617928;15;34;58
17541673;49;30;25
17209803;60;60;60
16205720;10;0;60
17642595;2;45;59
17236703;9;39;60
16441249;20;14;19
17442228;10;45;60
14203377;60;60;20
```

notasi1.txt

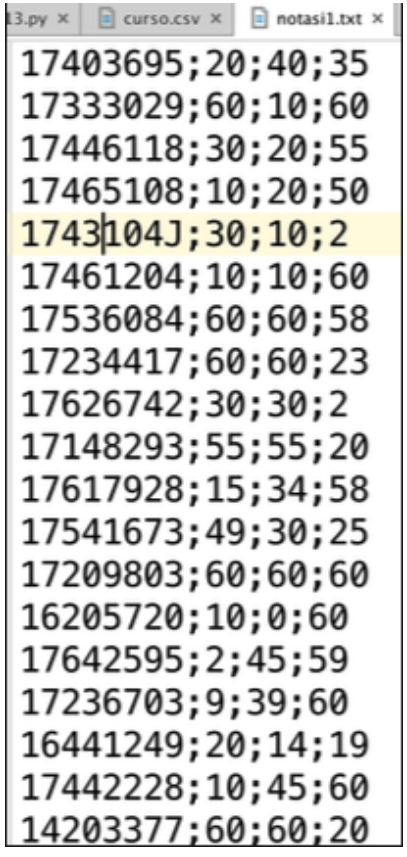
Archivos: lectura de acrchivos

Estos dos ejemplos son archivos de texto. Observemos cómo están cronstruidos.



```
1;17534932;ABARZÚA;REYES;FERNANDO NICOLÁS
2;1743104J;ACEVEDO;GÓMEZ;DANIELA PAZ
3;17642595;ALIAGA;PEREZ;CRISTIAN AUGUSTO
4;17209803;ANDRADE;LEÓN;MARÍA JOSÉ DE JESÚS
6;16205720;ARNAIZ;CASTILLO;BASTIAN XABIER
8;16441249;BARRIENTOS;BAEZA;AGUSTIN SALVADOR
11;17442228;CAMHI;GUERRERO;NICOLÁS DAVID
12;14203377;CARABALL;MIERI;JOSÉ THOMAS
13;17208536;CASASSUS;KUFEKE;NICOLÁS
14;17640913;CERDA;BROWNE;SEBASTIÁN ANDRÉS
15;17465108;CHADWICK;ALEMPARTE;VICENTE
16;17236703;CONTRERAS;MORGADO;ISMAEL
18;17461204;CORTÉS;DE LA FUENTE;TOMÁS IGNACIO
19;17536084;DAROCH;AHUMADA;JOSEFINA FRANCISCA
20;17132437;DAWABE;FAJARDO;JAVIERA ANDREA
21;17437639;DE GEYTER;MESSINA;MATÍAS
22;17403695;DÍAZ;GONZÁLEZ;FERNANDA ISIDORA
```

curso.csv



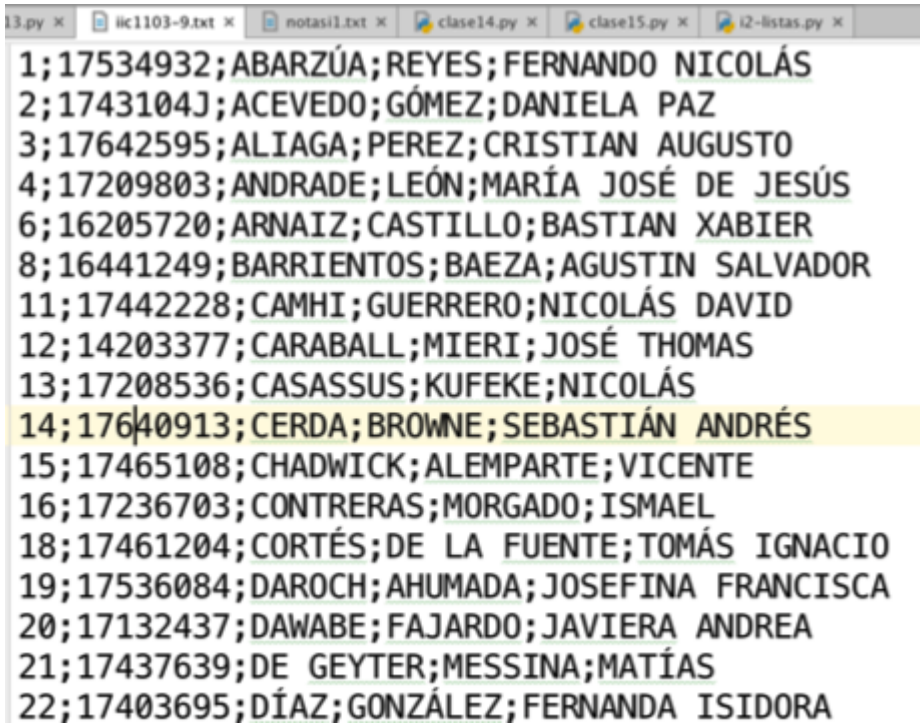
```
17403695;20;40;35
17333029;60;10;60
17446118;30;20;55
17465108;10;20;50
1743104J;30;10;2
17461204;10;10;60
17536084;60;60;58
17234417;60;60;23
17626742;30;30;2
17148293;55;55;20
17617928;15;34;58
17541673;49;30;25
17209803;60;60;60
16205720;10;0;60
17642595;2;45;59
17236703;9;39;60
16441249;20;14;19
17442228;10;45;60
14203377;60;60;20
```

notasi1.txt

Archivos: lectura de acrchivos

Archivo: curso.csv tiene la siguiente estructura en cada línea:

indice; *numeroDeAlumno*; *apellido*1; *apellido*2; *nombres*



Pensemos juntos: ¿Qué funciones que ya conocemos podemos usar en esta estructura de datos?

Archivos: lectura de acrchivos

Obs: Todos los archivos se encuentran en algún lugar de nuestro computador. A esta dirección se le llama *path*, que es el equivalente a la ruta que debe seguir el sistema para encontrar el archivo.



Archivos: lectura de acrchivos

lorem_ipsum.txt

lorem_ipsum.txt

```
1  Lorem ipsum dolor sit amet, consectetur adipiscing elit.
2  Donec placerat metus quis mollis gravida.
3  Donec vitae sapien sed eros consectetur luctus sit amet nec velit.
4  Vestibulum non ligula accumsan, vestibulum felis sit amet, feugiat leo.
5  In ac ante congue, consectetur orci ac, dapibus ligula.
6  Nam ornare tortor consequat accumsan fermentum.
7
8  Quisque et elit vitae velit tristique vehicula et sed elit.
9  Aliquam ut ipsum at neque interdum tristique.
10 Sed sodales urna mattis, vehicula lacus id, egestas massa.
11
12 Etiam accumsan ex elementum pretium finibus.
13 Pellentesque molestie neque imperdiet, tincidunt dui pharetra, semper
   nibh.
14 Ut vulputate sapien at odio accumsan, laoreet tincidunt risus vehicula.
15 Fusce pulvinar tortor nec porttitor tincidunt.
16 Ut semper augue sed condimentum tempus.
17 Nunc porta ligula et nisl pretium interdum.
```

Archivos: lectura de acrchivos

Antes de cualquier acción sobre un archivo debemos abrirlo.

- `f = open(p)`: Función que recibe una ruta a un archivo y retorna un objeto `f` del tipo `io.TextIOWrapper`.

Una vez que está abierto podemos ejecutar métodos sobre ese objeto:

- `f.readline()`: Método que lee la siguiente línea del archivo y retorna esa línea en un string.
- `f.close()`: Método que cierra un archivo.

Archivos: lectura de acrchivos

```
1  f = open('./lorem_ipsum.txt')
2
3  l = f.readline()
4  # Lorem ipsum dolor sit amet, consectetur adipiscing elit.\
   n
5  l = f.readline()
6  # Donec placerat metus quis mollis gravida.\n
7  l = f.readline()
8  # Donec vitae sapien sed eros consectetur luctus sit amet
   nec velit.\n
9
10 f.close()
```

Que hace el comando **readline()**:

- Lee hasta el siguiente '\n'.
- Avanza automáticamente a la siguiente línea.
- La línea retornada incluye un '\n' al final.
- Final del archivo es un string vacío ''.

Archivos: lectura de acrchivos

Ejemplo: Leer y mostrar en consola el archivo ./lorem_ipsum.txt

```
1 f = open('./lorem_ipsum.txt')
2
3 l = f.readline()
4 while(l != ''):            # linea vacia -> EOF
5     print(l.rstrip())
6     l = f.readline()       # leemos siguiente linea
7
8 f.close()                  # siempre cierren el archivo!
```

Observaciones:

- Con `rstrip()` eliminamos '\n' al final de cada línea.
- Una línea vacía no es '\n', ya que igual es un caracter.

Archivos: lectura de acrchivos

Si el archivo es una secuencia de datos, podemos pensar en formas alternativas de leer un archivo.
`for` nos permite iterar sobre secuencias finitas, en este caso sobre las líneas del archivo.

```
1 f = open('./lorem_ipsum.txt')
2 for linea in f:
3     print(linea.rstrip())
4 f.close()
```

`f.readlines()` retorna una lista con las líneas.

```
1 f = open('./lorem_ipsum.txt')
2 lineas  = f.readlines()
3 print(lineas)
4 f.close()
```

En ambos casos, hay que tener consideración de los '\n' al final del archivo.

RESUMEN: lectura de archivos

Antes de cualquier acción sobre un archivo debemos abrirlo.

- `f = open(p)`: Función que recibe una ruta a un archivo y retorna un objeto `f` del tipo `io.TextIOWrapper`.

Una vez que está abierto podemos ejecutar métodos sobre ese objeto:

- `f.readline()`: Lee la siguiente línea del archivo y retorna esa línea en un string.
- `f.readlines()`: Retorna una lista con todas las líneas.

Al finalizar nuestro código SIEMPRE debemos cerrar el/los archivo/s.

- `f.close()`: Cierra un archivo.

RESUMEN: lectura de archivos

Que hace el comando **readline()**:

- Lee hasta el siguiente '\n'.
- Avanza automáticamente a la siguiente línea. Tiene un contador interno.
- La línea retornada incluye un '\n' al final.
- Final del archivo es un string vacío ''.

Que hace el comando **readlines()**:

- Retorna en una variable del tipo lista todo el contenido del archivo.
- En ambos casos, hay que tener consideración de los '\n' al final del archivo.

Archivos

