

# Clase 17 - Archivos

IIC1103-07 - Introducción a la Programación

---

Cristian Ruz – [cruz@ing.puc.cl](mailto:cruz@ing.puc.cl)

Martes 09-Octubre-2018

Departamento de Ciencia de la Computación  
Pontificia Universidad Católica de Chile

Sobre los laboratorios

Recapitulación

Archivos

Lectura de archivos

Escritura de archivos

Sobre los laboratorios

Recapitulación

Archivos

Lectura de archivos

Escritura de archivos

## Laboratorio 7. *Listas (II)*. Semana 8-14 Octubre.

- Recolectado el puntaje del Laboratorio 6
  - Si su puntaje está incorrecto avise a la brevedad
  - Todavía pueden resolver el lab 6 hasta el domingo 14-Oct
  - Mitad del puntaje que agreguen al lab 5 esta semana
- Salas de lunes a jueves. Entrega hasta el domingo.
  - W:5,6. Sala A5 (con laptop), Lab San Agustín
  - L:5,6. Lab San Agustín (si no tiene laptop)
  - M:5,6. CS101, B12, A5, Lab San Agustín
  - J:5,6. CS101, B12, K204, Lab San Agustín
- 6 problemas de programación
  - Máximo 1200 puntos por resolverlos durante la semana
  - Máximo 600 puntos por resolverlos en la semana siguiente

Sobre los laboratorios

Recapitulación

Archivos

Lectura de archivos

Escritura de archivos

# ¿En qué vamos?

1. **Variables y expresiones**
2. **Control de flujo**
3. **Funciones**
4. **Strings**
5. **Listas**
6. **Archivos**  $\Leftarrow$  **esto lo veremos HOY**
7. Programación orientada a objetos  $\Leftarrow$  *hasta aquí para la I2, V 26-Octubre, 18:30*
8. Ordenación y búsqueda
9. Recursión

Sobre los laboratorios

Recapitulación

Archivos

Lectura de archivos

Escritura de archivos

**Objetivo:** aprender a leer los datos de los estudiantes desde un archivo, y *cargarlos (load)* en una lista de estudiantes.

```
1.py x ac1101-9.txt x notasi1.txt x clase14.py x clase15.py x 2-hits.py x
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

```
3.py x curso.csv x notasi1.txt x
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



Ambos son **archivos de texto**.

Observen como están **construídos**.<sup>1</sup>

```
13.py x 11103-9.txt x notasi1.txt x clase14.py x clase15.py x i2-listas.py x
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`

```
13.py x curso.csv x notasi1.txt x
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
```

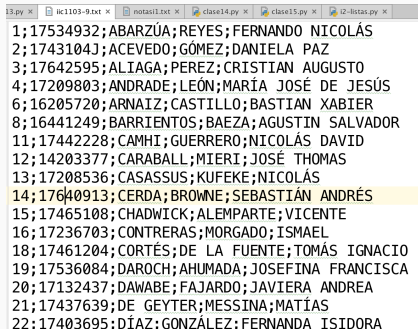
`notasi1.txt`

<sup>1</sup>Los puntajes son ficticios para proteger a los inocentes

Archivo: `curso.csv`

En cada línea:

*numeroDeAlumno; apellido1; apellido2; nombres*



A screenshot of a text editor window displaying a CSV file. The window has several tabs open at the top: '13.py', '11103-9.txt', 'notas1.txt', 'clase14.py', 'clase15.py', and '12-listas.py'. The main content area shows a list of 22 lines of text, each representing a student record. The records are separated by semicolons. The 14th line, '14;17640913; CERDA; BROWNE; SEBASTIÁN ANDRÉS', is highlighted in yellow. The records are as follows:

```
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;KUFKE;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
```

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.

# Lectura de archivos

```
1 El amor es un mejor profesor que el deber.
2 Información no es conocimiento.
3 Nunca pierdas la sagrada curiosidad.
4 Si no puedes explicarlo de forma simple, no lo entiendes lo suficientemente bien
5
6 Todos deben ser respetados como individuos, pero no idolatrados.
7
8 Nunca hagas algo contra tu conciencia incluso si la situación lo demanda.
9 Locura: hacer lo mismo una y otra vez, esperando diferentes resultados.
10 Un hombre debe buscar lo que es, no lo que piensa que es.
11 Una persona que nunca ha cometido un error nunca ha intentado algo nuevo.
12 Aprende del ayer, vive del hoy, espera del mañana.
13
14 Todo debe ser tan simple como es, pero no más sencillo.
15 La lógica te lleva del punto A al B. La imaginación te llevará donde sea.
```

./Einstein.txt

¿Cómo leemos esto desde python?

# Lectura de archivos

```
1 f = open('./Einstein.txt')
2
3 l = f.readline()
4 # "El amor es un mejor profesor que el deber.\n"
5 l = f.readline()
6 # "Información no es conocimiento.\n"
7 l = f.readline()
8 # "Nunca pierdas la sagrada curiosidad.\n"
9
10 f.close()
```

## Observaciones 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 ''.

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

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

## Observaciones:

- Con `rstrip()` eliminamos `'\n'` al final de cada línea.
- Una línea vacía será `'\n'`, no `''`.

Formas alternativas de leer un archivo:

for sobre las líneas del archivo.

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

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

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

En ambos casos, hay que manejar los '\n' final.

`f = open(p,m)`: Función que abre un archivos en modo `m`.

- `"r"` → Modo lectura (por defecto).
- `"w"` → Modo escritura.
- `"a"` → Modo append.

`f.write(s)`: Escribe `s` en el archivo (sin fin de línea)

`print(s, file=f)`: Escribe `s` en el archivo. (**¡preferido!**)

`f.close()`: Método que cierra el archivo.



¿Diferencia entre modo *escritura* y modo *append*?

`f = open(p, "w")`: Crea un nuevo archivo en ruta `p`. Si el archivo ya existe, borra su contenido.

`f = open(p, "a")`: Crea un nuevo archivo en ruta `p`. Si el archivo ya existe, lo lee y agrega el nuevo contenido al final.

**Ejemplo:** Escribir archivo con números del 1 al 5.

```
1 f = open('./test.txt', 'w')
2 for i in range(1,6):
3     f.write(str(i) + "\n")
4 f.close()
```

**Obs:** A diferencia de `print()`, el método `write()` no hace un salto de línea automático.

**Ejemplo:** Escribir archivo con números del 1 al 5.

```
1 f = open('./test.txt', 'w')
2 for i in range(1,6):
3     print(i, file=f)
4 f.close()
```

`print()`, es lo mismo que hemos hecho hasta ahora, pero en un archivo

**Resultado:** (Luego de ejecutarlo 2 veces)

```
1 1
2 2
3 3
4 4
5 5
```

`"/test.txt"`

¿Qué ocurre si hago lo mismo, pero en modo *append*?

```
1 f = open('./test2.txt', 'a')
2 for i in range(1,6):
3     print(i, file = f)
4 f.close()
```

**Resultado:** (Luego de ejecutarlo 2 veces)

```
1 1
2 2
3 3
4 4
5 5
6 1
7 2
8 3
9 4
10 5
```

`"./test2.txt"`