

## SESIÓN 2

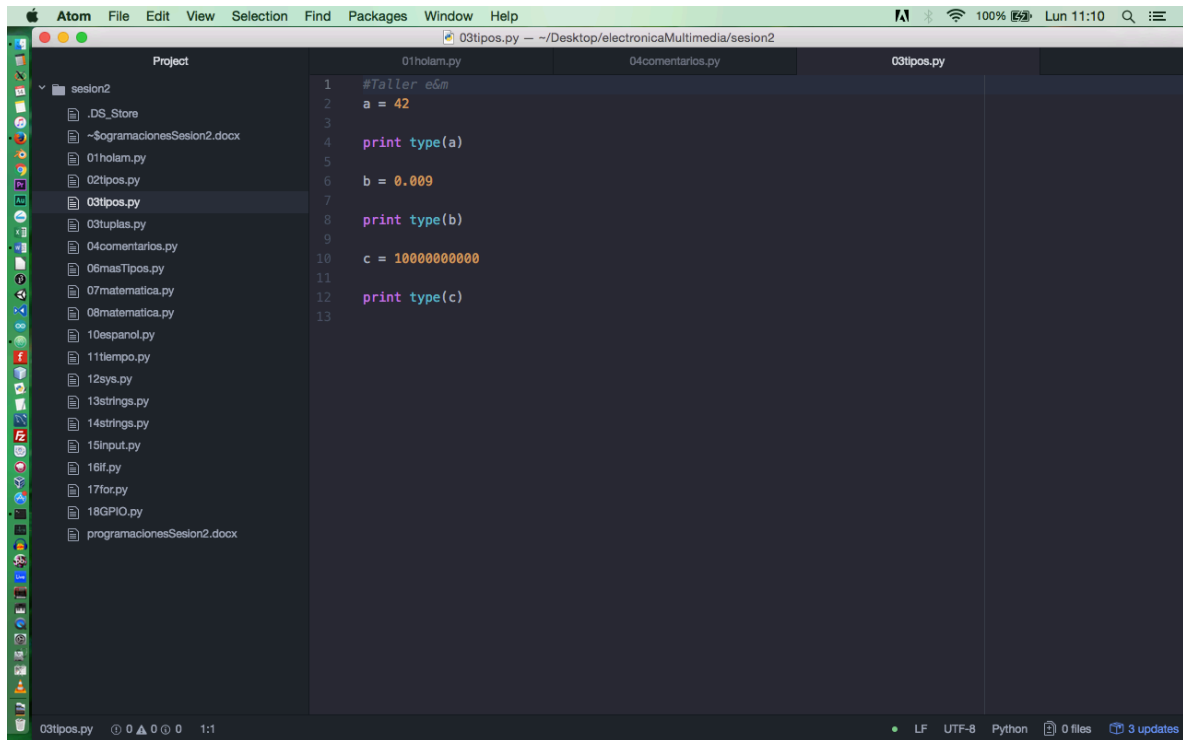
### 01holam.py

```
1 #Taller e6m
2
3 print('Electronica y multimedia')#imprime en terminal
4
```

### 02tipos.py

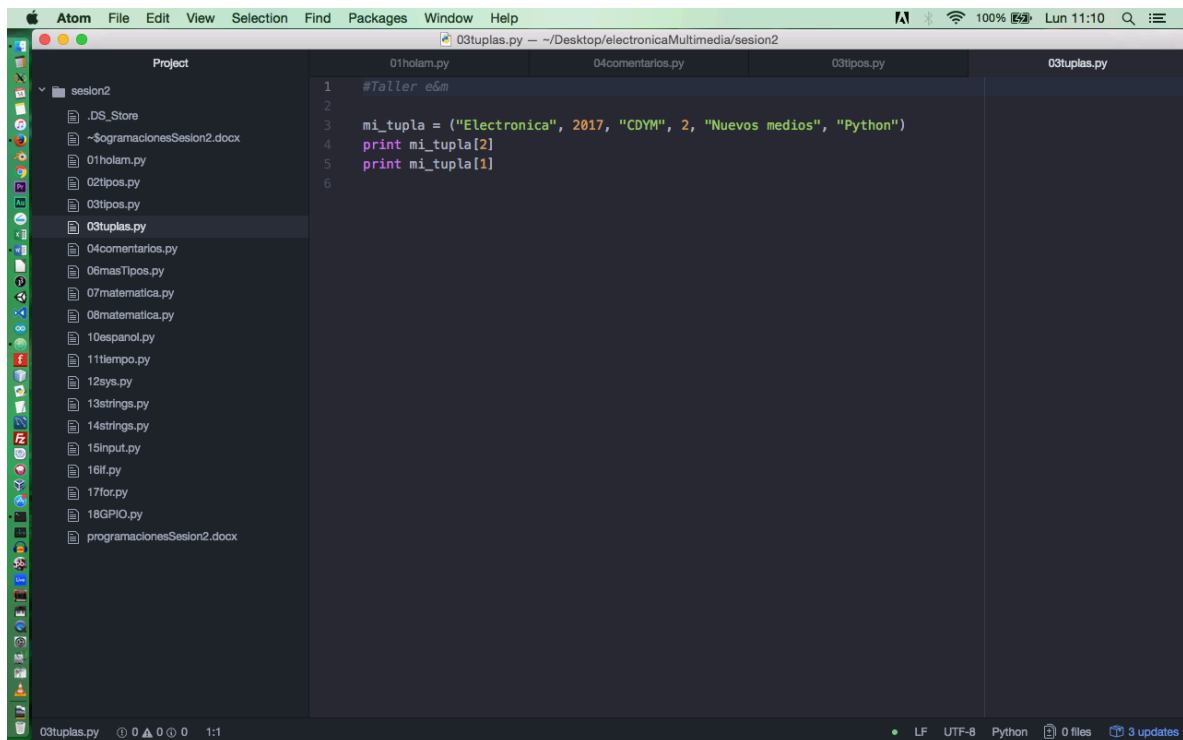
```
1 #Taller e6m
2 mi_var = "12345"
3
4 print mi_var
5
6 print type(mi_var) #imprime el tipo de dato o variable
7
8 mi_var_i = int(mi_var)
9
10 print mi_var_i
11
12 print type(mi_var_i)
13
```

## 03tipos.py



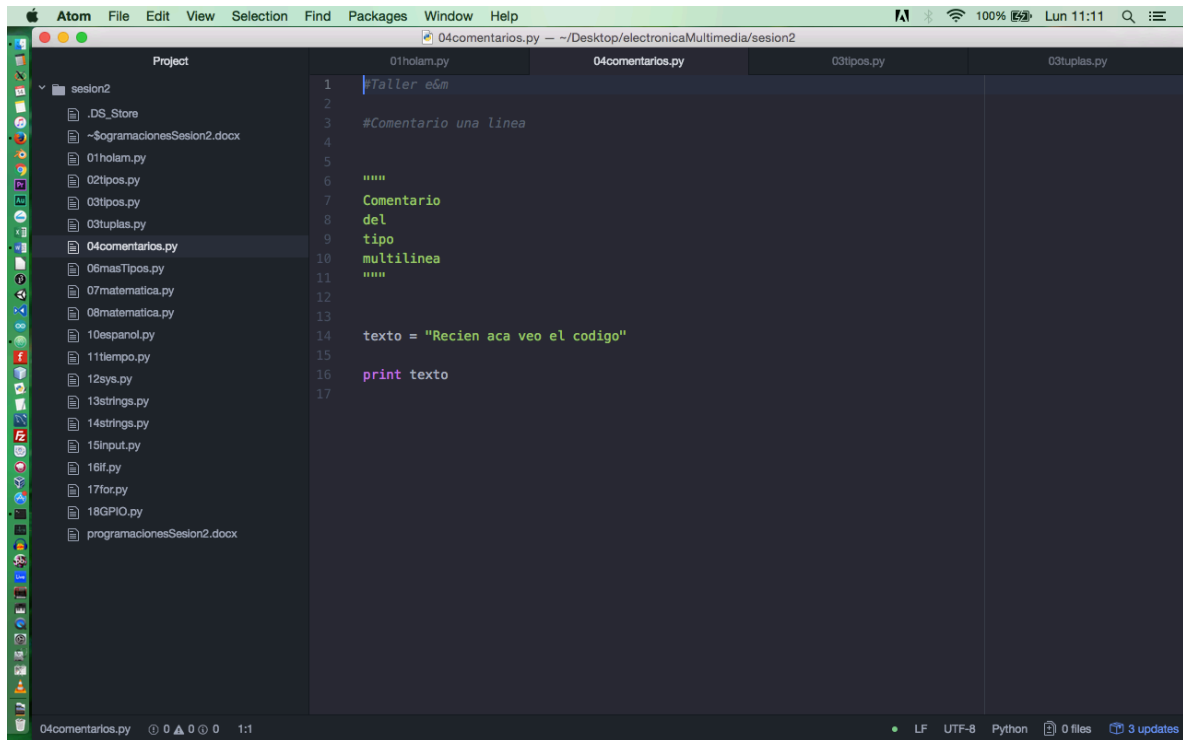
```
1 #Taller e6m
2 a = 42
3
4 print type(a)
5
6 b = 0.009
7
8 print type(b)
9
10 c = 10000000000
11
12 print type(c)
13
```

## 03tuplas.py



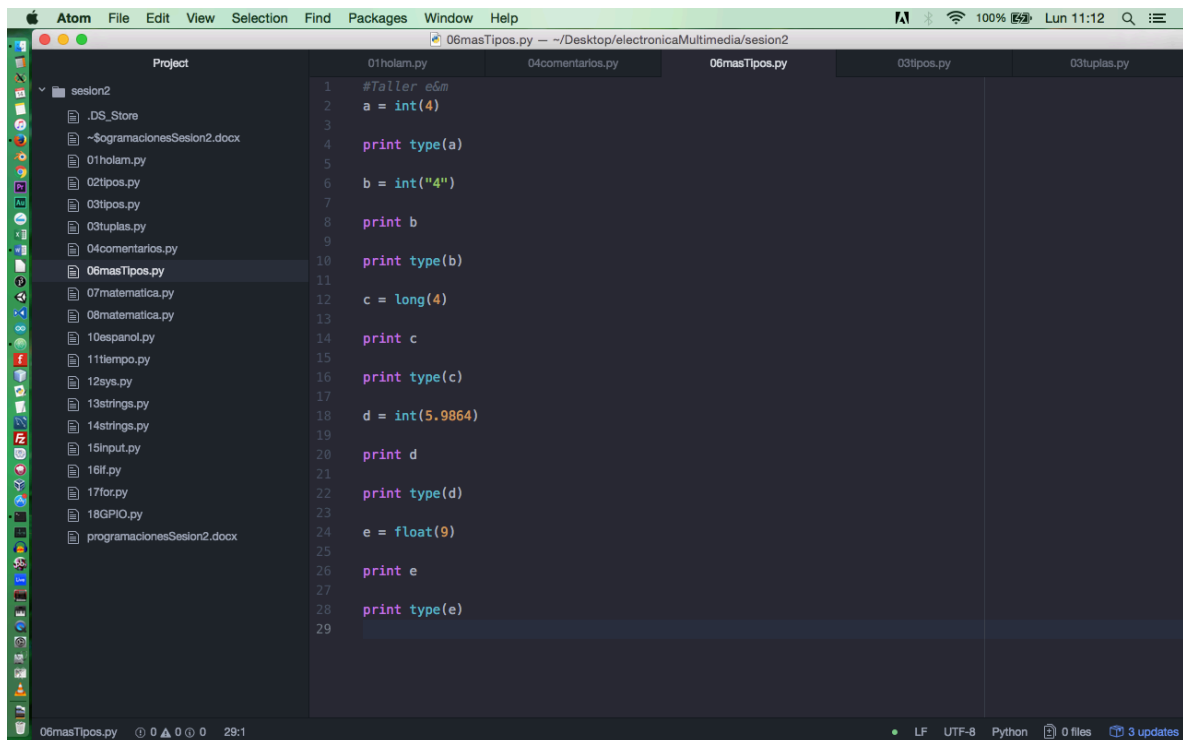
```
1 #Taller e6m
2
3 mi_tupla = ("Electronica", 2017, "CDYM", 2, "Nuevos medios", "Python")
4 print mi_tupla[2]
5 print mi_tupla[1]
6
```

## 04comentarios.py



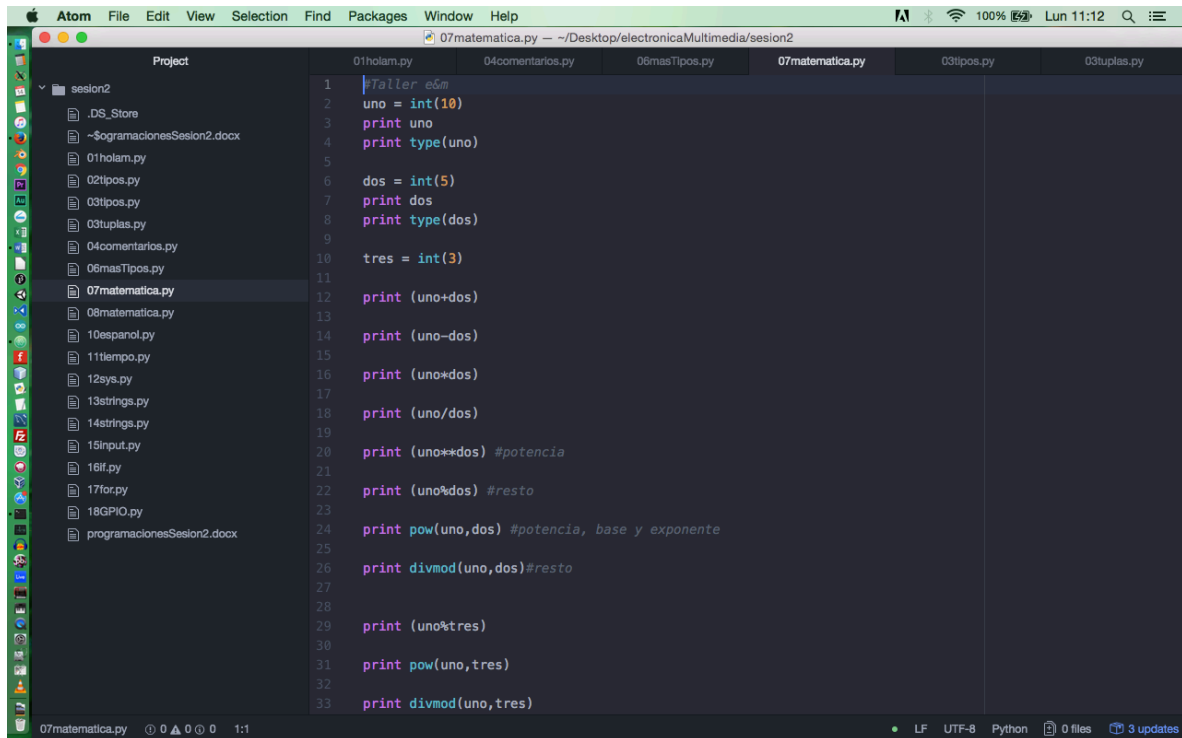
```
1 Taller e6m
2
3 #Comentario una linea
4
5
6 """
7 Comentario
8 del
9 tipo
10 multilinea
11 """
12
13
14 texto = "Recien aca veo el codigo"
15
16 print texto
17
```

## 06masTipos.py



```
1 #Taller e6m
2 a = int(4)
3
4 print type(a)
5
6 b = int("4")
7
8 print b
9
10 print type(b)
11
12 c = long(4)
13
14 print c
15
16 print type(c)
17
18 d = int(5.9864)
19
20 print d
21
22 print type(d)
23
24 e = float(9)
25
26 print e
27
28 print type(e)
29
```

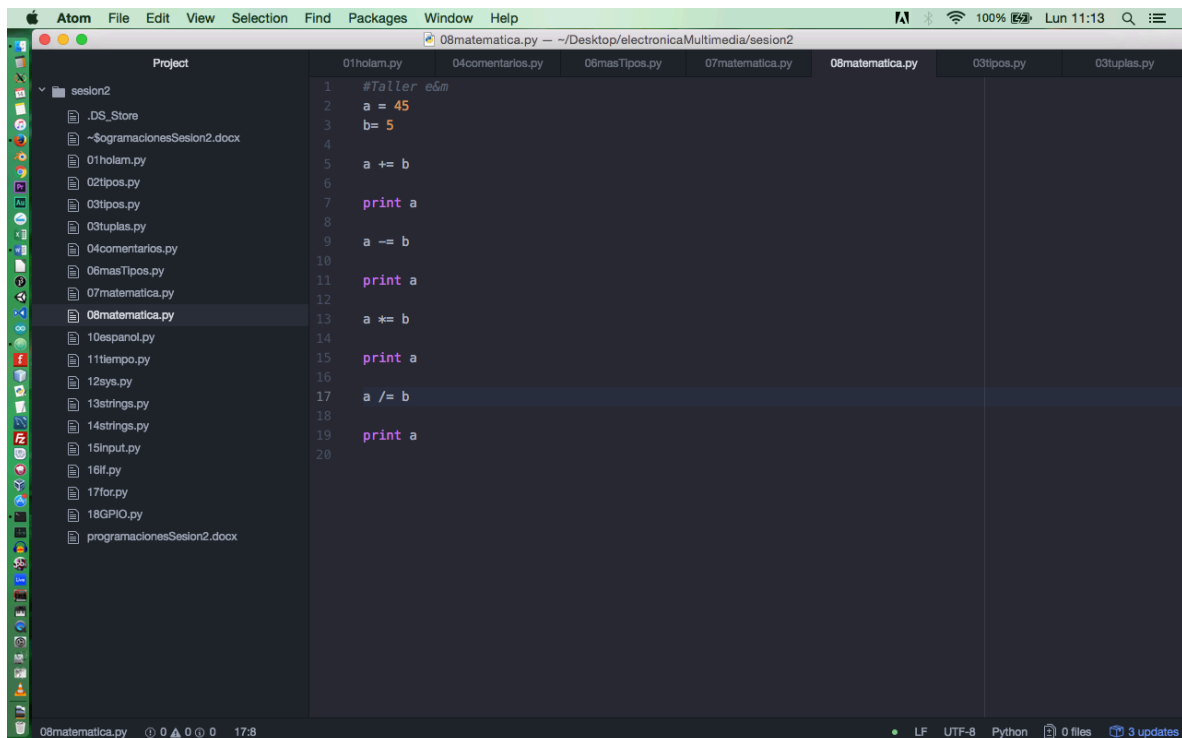
## 07matematica.py



The screenshot shows the Atom editor interface with the file 07matematica.py open. The left sidebar displays a project tree for 'sesion2' containing various Python files. The main editor area shows the following code:

```
1 #Taller e6m
2 uno = int(10)
3 print uno
4 print type(uno)
5
6 dos = int(5)
7 print dos
8 print type(dos)
9
10 tres = int(3)
11
12 print (uno+dos)
13
14 print (uno-dos)
15
16 print (uno*dos)
17
18 print (uno/dos)
19
20 print (uno**dos) #potencia
21
22 print (uno%dos) #resto
23
24 print pow(uno,dos) #potencia, base y exponente
25
26 print divmod(uno,dos)#resto
27
28
29 print (uno*tres)
30
31 print pow(uno,tres)
32
33 print divmod(uno,tres)
```

## 08matematica.py



The screenshot shows the Atom editor interface with the file 08matematica.py open. The left sidebar displays a project tree for 'sesion2'. The main editor area shows the following code:

```
1 #Taller e6m
2 a = 45
3 b = 5
4
5 a += b
6
7 print a
8
9 a -= b
10
11 print a
12
13 a *= b
14
15 print a
16
17 a /= b
18
19 print a
20
```

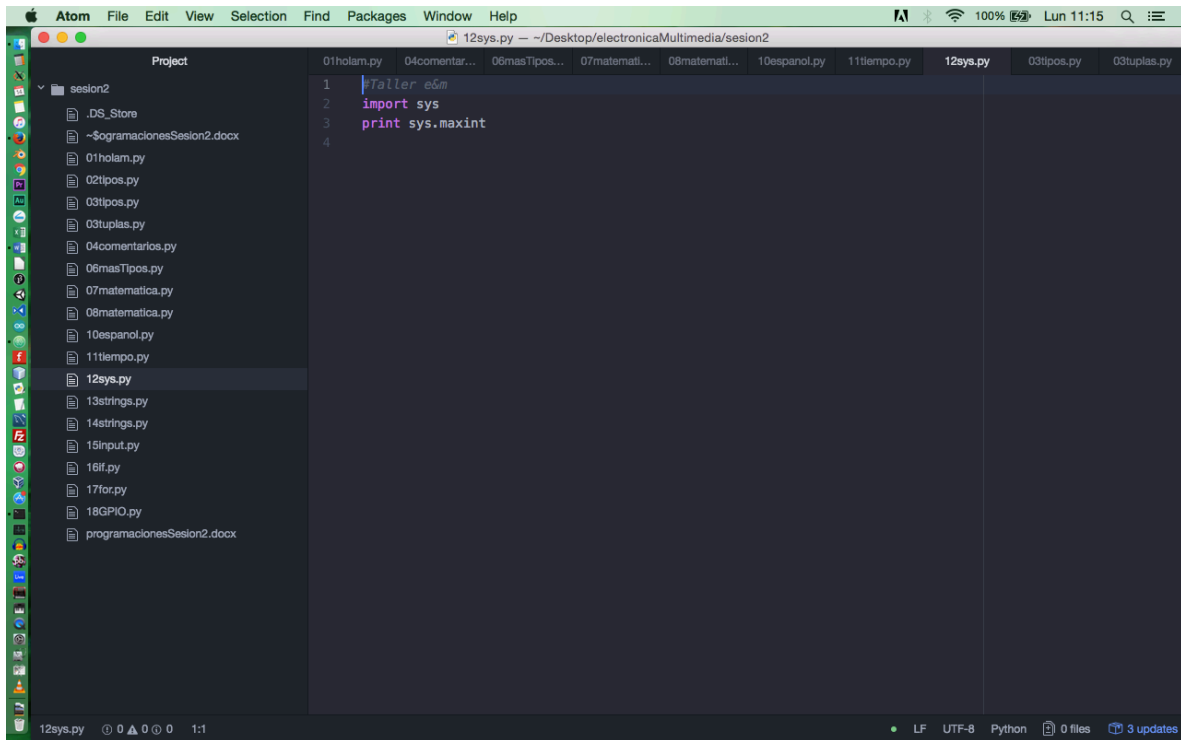
## 10espanol.py

```
1 #Taller e6m
2 # -*- coding: utf-8 -*-
3 #Codificar con caracteres especiales, español
4 print "%s tiene %d años" % ("José", 41)
5
```

## 11tiempo.py

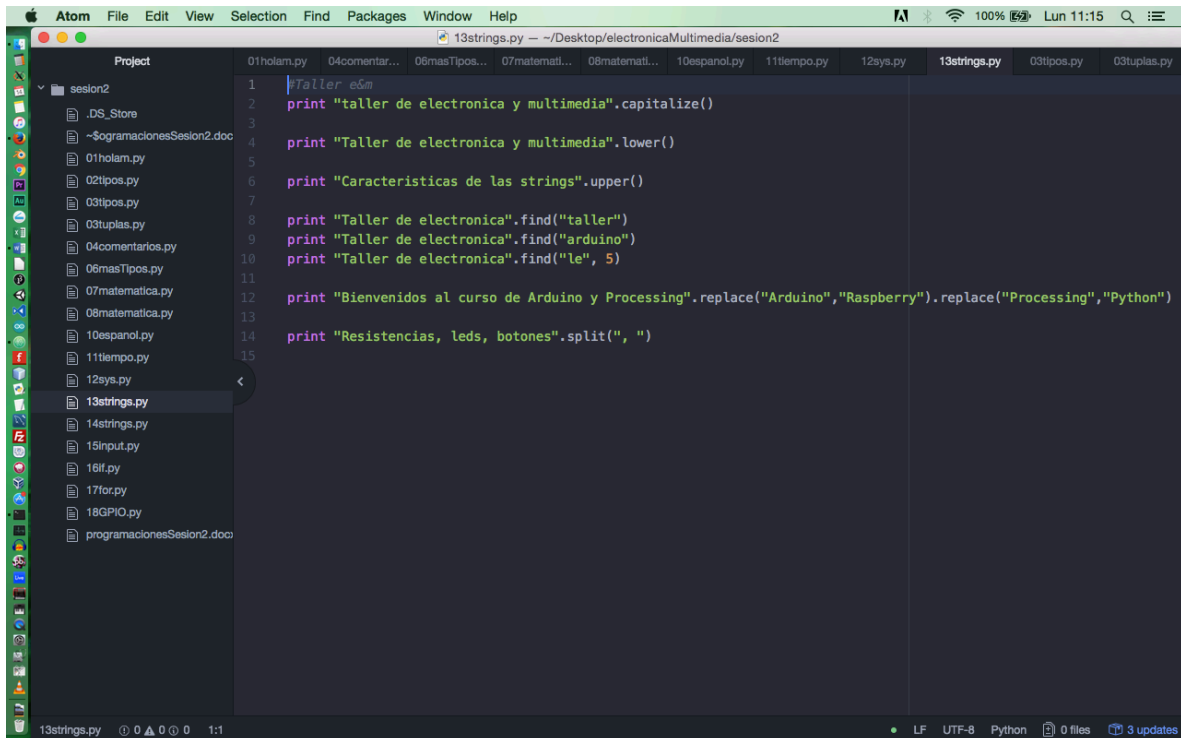
```
1 #Taller e6m
2 import time #importar modulos
3 print "Hoy es: " + time.ctime()
4
```

## 12sys.py



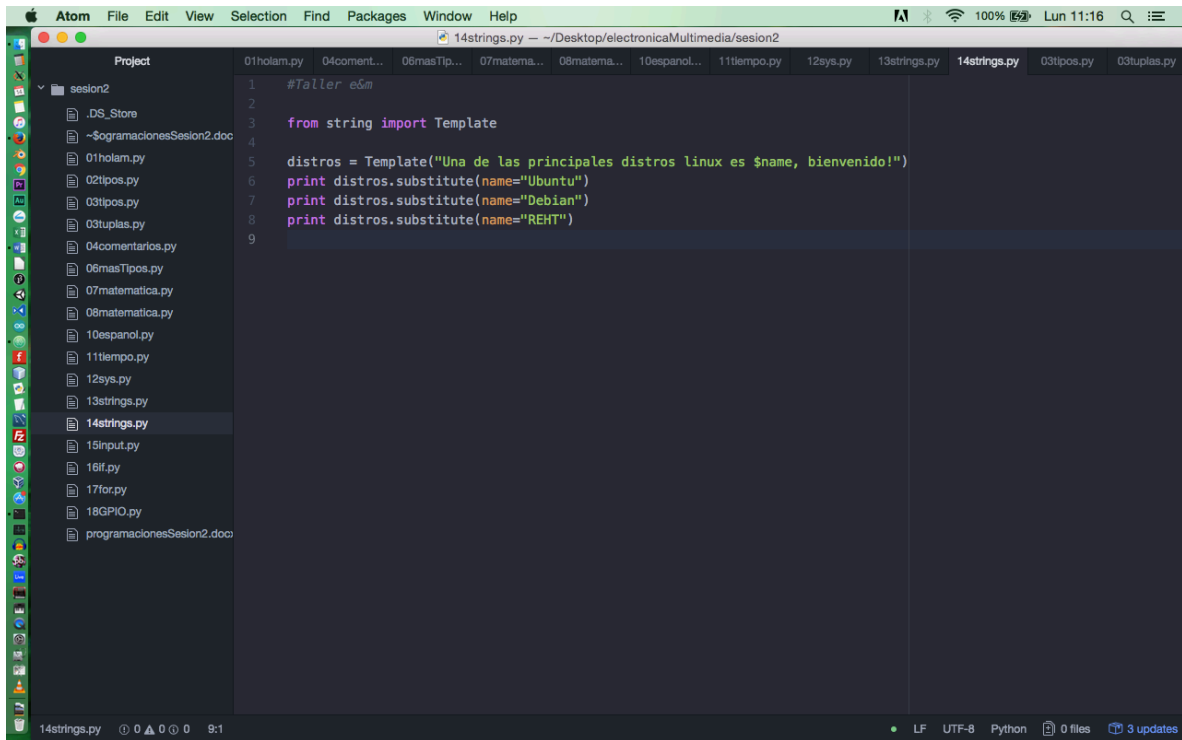
```
1 #Taller e5m
2 import sys
3 print sys.maxint
4
```

## 13strings.py



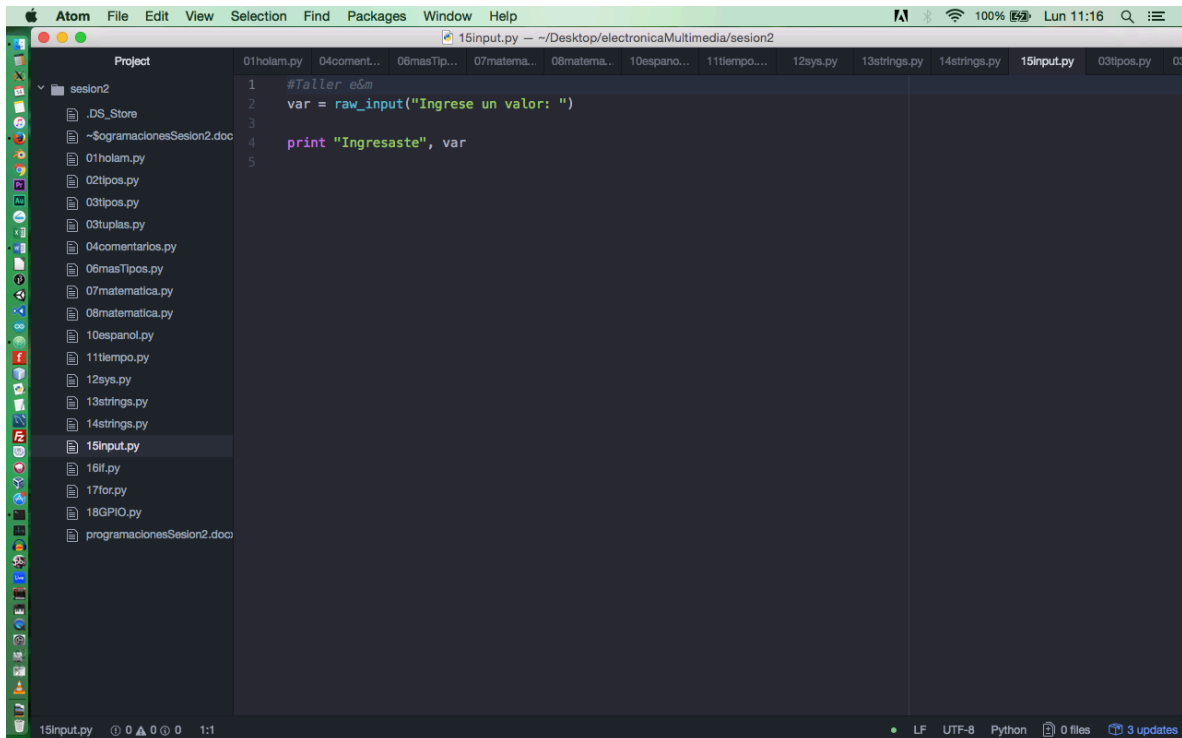
```
1 #Taller e5m
2 print "taller de electronica y multimedia".capitalize()
3
4 print "Taller de electronica y multimedia".lower()
5
6 print "Características de las strings".upper()
7
8 print "Taller de electronica".find("taller")
9 print "Taller de electronica".find("arduino")
10 print "Taller de electronica".find("le", 5)
11
12 print "Bienvenidos al curso de Arduino y Processing".replace("Arduino","Raspberry").replace("Processing","Python")
13
14 print "Resistencias, leds, botones".split(", ")
15
```

## 14strings.py



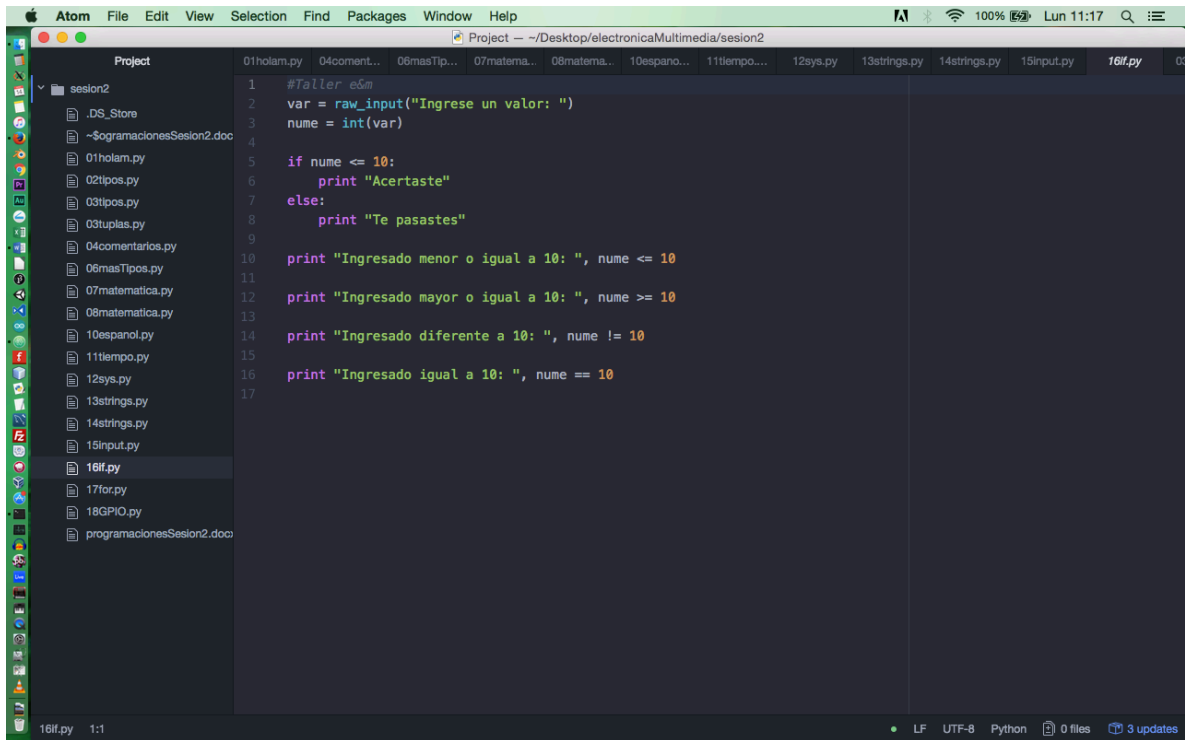
```
1 #Taller e5m
2
3 from string import Template
4
5 distros = Template("Una de las principales distros linux es $name, bienvenido!")
6 print distros.substitute(name="Ubuntu")
7 print distros.substitute(name="Debian")
8 print distros.substitute(name="REHT")
9
```

## 15input.py



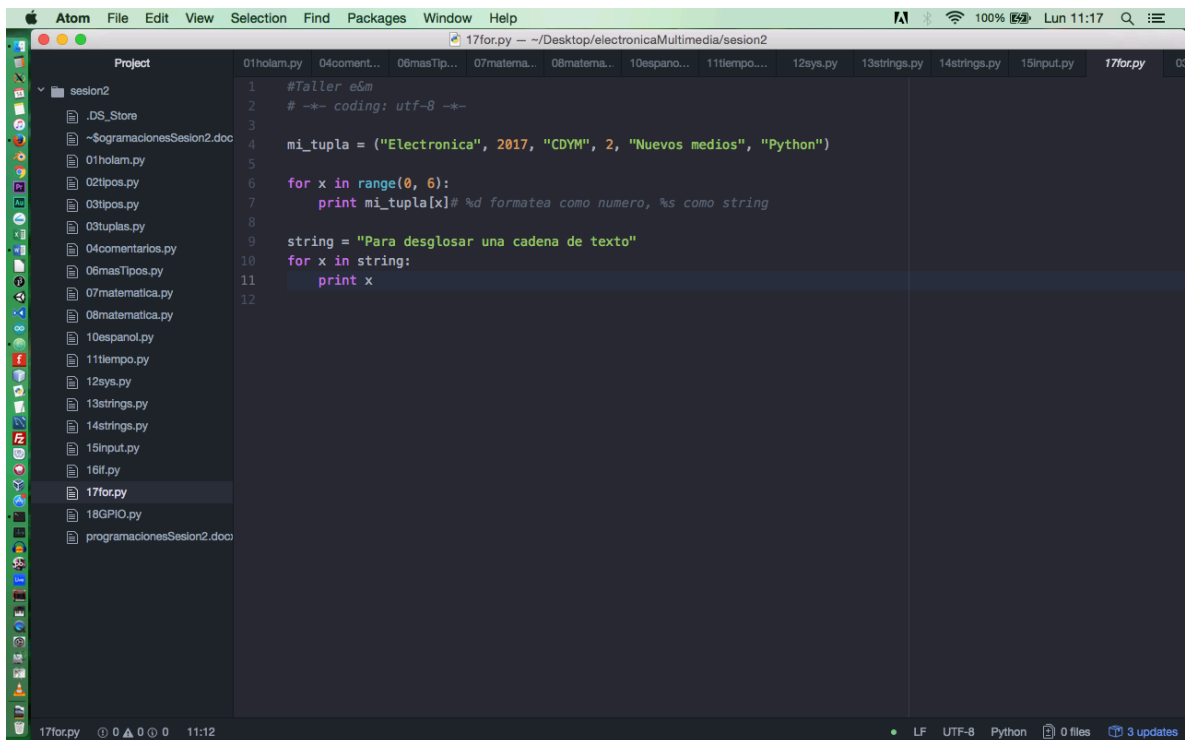
```
1 #Taller e5m
2 var = raw_input("Ingrese un valor: ")
3
4 print "Ingresaste", var
5
```

## 16if.py



```
1 #Taller e5m
2 var = raw_input("Ingrese un valor: ")
3 nume = int(var)
4
5 if nume <= 10:
6     print "Acertaste"
7 else:
8     print "Te pasastes"
9
10 print "Ingresado menor o igual a 10: ", nume <= 10
11
12 print "Ingresado mayor o igual a 10: ", nume >= 10
13
14 print "Ingresado diferente a 10: ", nume != 10
15
16 print "Ingresado igual a 10: ", nume == 10
17
```

## 17for.py























```
1 #Taller e5m
2 # -*- coding: utf-8 -*-
3
4 mi_tupla = ("Electronica", 2017, "CDYM", 2, "Nuevos medios", "Python")
5
6 for x in range(0, 6):
7     print mi_tupla[x]# %d formatea como numero, %s como string
8
9 string = "Para desglosar una cadena de texto"
10 for x in string:
11     print x
12
```



## GPIO

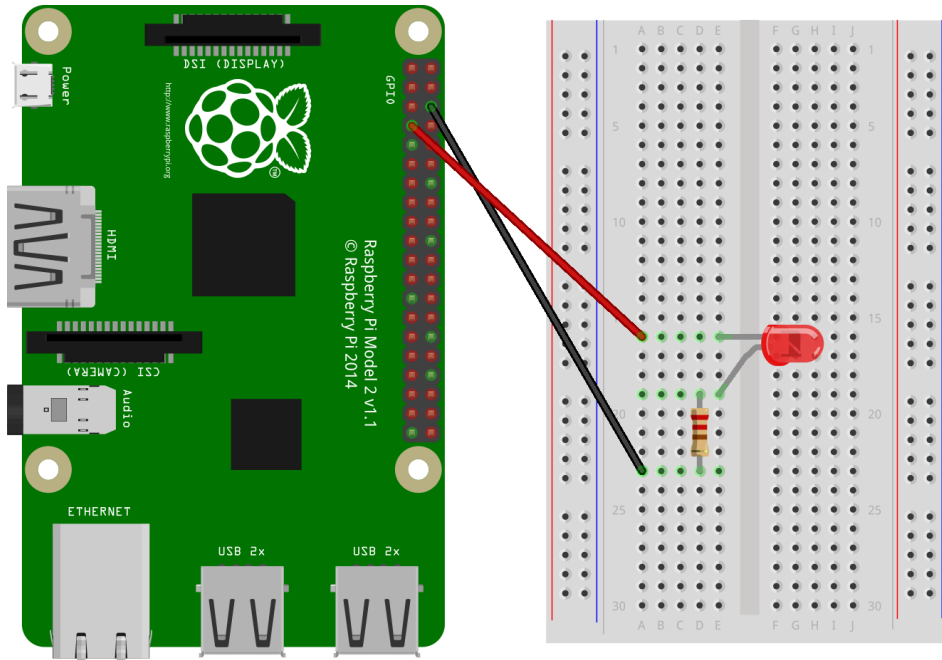
### Raspberry Pi 3 GPIO Header

<i>Pin#</i>	<i>NAME</i>		<i>NAME</i>	<i>Pin#</i>
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I <sup>2</sup> C)		DC Power 5v	04
05	GPIO03 (SCL1 , I <sup>2</sup> C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I <sup>2</sup> C ID EEPROM)		(I <sup>2</sup> C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

Rev. 2  
29/02/2016

[www.element14.com/RaspberryPi](http://www.element14.com/RaspberryPi)

## CIRCUITO LED



fritzing

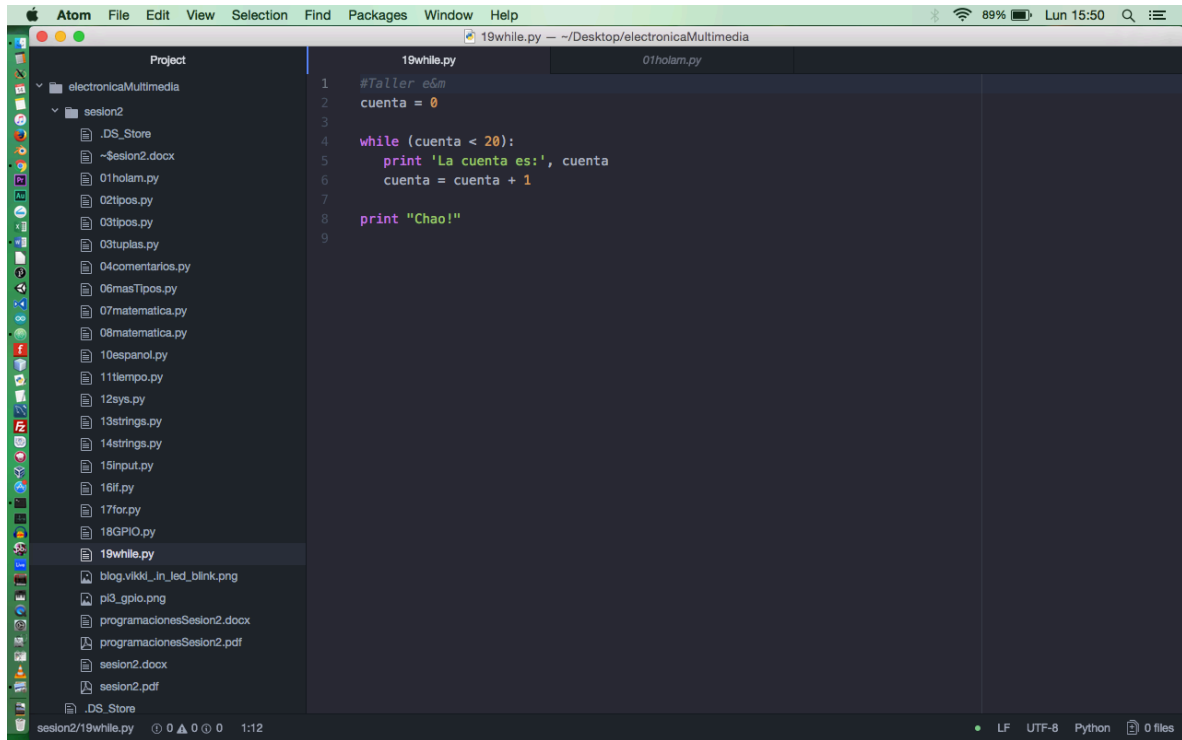
## 18GPIO.py

```
Atom File Edit View Selection Find Packages Window Help
18GPIO.py - ~/Desktop/electronicaMultimedia/sesion2

Project
  session2
    .DS_Store
    ~$ogramacionesSesion2.doc
    01holam.py
    02tipos.py
    03tipos.py
    03tuplas.py
    04comentarios.py
    06masTipos.py
    07matematica.py
    08matematica.py
    10espanol.py
    11tiempo.py
    12sys.py
    13strings.py
    14strings.py
    15input.py
    16if.py
    17for.py
    18GPIO.py
    programacionesSesion2.doc

18GPIO.py
1  #Taller eSm
2  # -*- coding: utf-8 -*-
3
4  import RPi.GPIO as GPIO
5  from time import sleep
6
7  GPIO.setmode(GPIO.BCM)
8
9  led_gpio = 4
10 GPIO.setup(led_gpio, GPIO.OUT)
11
12 for _ in range(10):
13
14     GPIO.output(led_gpio, not GPIO.input(led_gpio))
15     sleep(0.1)
16 GPIO.cleanup()
17
```

## 19while.py



```
1 #Taller: e5m
2 cuenta = 0
3
4 while (cuenta < 20):
5     print 'La cuenta es:', cuenta
6     cuenta = cuenta + 1
7
8 print "Chao!"
9
```

The screenshot shows the Atom code editor interface. The left sidebar displays a project tree with the following structure:

- Project
  - electronicaMultimedia
    - sesion2
      - .DS\_Store
      - ~\$esion2.docx
      - 01holam.py
      - 02tipos.py
      - 03tipos.py
      - 03tuplas.py
      - 04comentarios.py
      - 06masTipos.py
      - 07matematica.py
      - 08matematica.py
      - 10espanol.py
      - 11tiempo.py
      - 12sys.py
      - 13strings.py
      - 14strings.py
      - 15input.py
      - 16if.py
      - 17for.py
      - 18GPIO.py
      - 19while.py**
      - blog.vikki\_in\_led\_blink.png
      - pi3\_gpio.png
      - programacionesSesion2.docx
      - programacionesSesion2.pdf
      - sesion2.docx
      - sesion2.pdf
      - .DS\_Store

The main editor area shows the code for 19while.py. The status bar at the bottom indicates the file is named 'sesion2/19while.py', has 0 errors, 0 warnings, and 0 info messages, and is 1:12 lines long. The encoding is UTF-8 and the language is Python.