



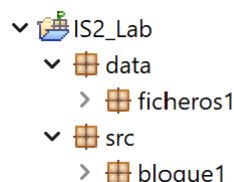
Este bloque abarcará las tres primeras sesiones de laboratorio, y se abordarán nociones de dos conceptos importantes: funciones y lectura/escritura de ficheros. Para ello, seguiremos los siguientes pasos.

1. Construye el proyecto

Ejecuta *Eclipse*¹ y construya un proyecto *PyDev*² de nombre **IS2-Lab**³. Posteriormente, cree:

- El paquete **src** y, dentro de este, el paquete **bloque1** (ubicación donde se guardarán los módulos implementados).
- El paquete **data** y, dentro de este, el paquete **ficheros1** (ubicación donde se guardarán los ficheros de texto o ficheros csv utilizados).

Quedando una estructura análoga a la siguiente:



2. Funciones

Dentro del paquete **src.bloque1** cree el módulo⁴ **funciones.py** e implemente y pruebe las siguientes funciones:

1. Dada una lista de números reales, implemente funciones que devuelvan la suma, la media, el máximo, el mínimo y la varianza.
2. Dada una lista con números enteros, implemente una función que devuelva otra lista con las diferencias entre cada valor y el anterior.
3. Dado un número real $n > 0$ y un número entero $x \geq 0$, implemente una función que devuelva n pero truncado a x decimales. Por ejemplo, si $n = 3,1416$ y $x = 2$, debe devolver 3,14.
4. Dados dos números enteros n, k con $n \geq k$, implemente una función que calcule el producto $\prod_{i=0}^{k-1} (n - i)$.
5. Dado un número entero $n > 1$, implemente una función para decidir si un número, pasado como parámetro, es primo o no.
6. Dado un número entero $n > 1$, implemente una función que devuelva una lista con sus divisores ordenados de menor a mayor.
7. Dada una cadena de caracteres con todos ellos en mayúsculas, implemente una función que devuelva otra con todos los caracteres en minúsculas salvo el comienzo de cada palabra que debe quedar en mayúscula.
8. Dada una cadena de caracteres, implemente una función que devuelva la misma cadena pero invertida.
9. Dada una cadena de caracteres, implemente una función que devuelva la misma cadena pero sustituyendo las vocales por guiones.
10. Dada una cadena de caracteres, implemente una función que devuelva el número de palabras que contiene.
11. Dada una lista de cadenas de caracteres, implemente una función que devuelva la cadena más larga.

¹ Entorno de desarrollo integrado (IDE, acrónimo del término inglés *Integrated Development Environment*).

² Complemento Python para Eclipse.

³ La intención es usar este mismo proyecto en todas las sesiones del bloque 1.

⁴ Un módulo en Python es un fichero .py que alberga un conjunto de funciones, variables, ... y que puede ser usado por otros módulos.



2. Escritura y lectura de ficheros

2.1. Escritura y lectura de un fichero de texto.

Los pasos a seguir son:

a. Dentro del paquete **scr.bloque1**, cree ahora el módulo **tratarFicherosTexto.py** y realice los siguientes pasos:

a.1. Implemente la función **crearFicheroTexto(nombre, texto)**.

- Esta función recibe un dos parámetro, uno con el **nombre** del fichero de texto a crear, p.e. **"refran.txt"**, y otro con el **texto** que se desea guardar en dicho fichero.
- La función debe devolver un valor lógico:
 - o **True** si ha ido bien la creación del fichero
 - o **False**, en caso contrario.

a.2. Implemente la función **leerFicheroTexto(nombre)**. Esta función recibe un parámetro con el **nombre** del fichero de texto a leer y muestra por consola su contenido (print).

a.3. Implemente también:

- Función **numPalabras(fichero)** que devuelva el número de palabras que contiene el fichero de texto pasado como parámetro de entrada.
- Función **numRepeticiones(fichero, palabra)** que devuelva las veces que se repite una palabra en un fichero, ambos dados como parámetros de entrada.

b. Pruebe todas las funciones implementadas en el mismo módulo, es decir, en **tratarFicherosTexto.py** y utilice el siguiente texto para la escritura del fichero:

"Una hora duerme el gallo, dos el caballo, tres el santo, cuatro el que no es tanto, cinco el capuchino, seis el peregrino, siete el caminante, ocho el estudiante, nueve el caballero, diez el pordiosero, once el muchacho y doce el borracho."

2.2. Escritura y lectura de un fichero .csv

Los pasos a seguir son:

a. Dentro del paquete **scr.bloque1**, cree el módulo **tratarFicherosCSV.py** y realice los siguientes pasos:

a.1. Implemente la función **crearFicheroCSV(nombre, texto)**.

- Esta función recibe un dos parámetro, uno con el **nombre** del fichero csv a crear, p.e. **"provincias.csv"**, y otro con el texto, es decir, los registros que se desean guardar en dicho fichero.
- Poner como delimitador el carácter **"|"** (Barra vertical)
- La función debe devolver un valor lógico:
 - o **True** si ha ido bien la creación del fichero
 - o **False**, en caso contrario.

a.2. Implemente la función **leerFicheroCSV(nombre, delimitador)**. Esta función recibe dos parámetros, uno con el **nombre** del fichero CSV a leer y, otro, con el carácter delimitador. Como resultado debe mostrar por consola su contenido (print).



- a.3. Implemente también una función **numFilasCSV(ficheroCSV)** que devuelva el número de filas (int) que tiene el fichero csv pasado como parámetro de entrada.
- b. Pruebe todas las funciones implementadas en el mismo módulo, es decir, en **tratarFicherosCSV.py** y utilice el siguiente texto para la escritura del fichero:



"Código, Provincia

01, Álava

02, Albacete

03, Alicante

04, Almería

33, Asturias

05, Ávila

06, Badajoz

07, Baleares

08, Barcelona

09, Burgos

10, Cáceres

11, Cádiz

39, Cantabria

12, Castellón

13, Ciudad Real

14, Córdoba

16, Cuenca

17, Gerona

18, Granada

19, Guadalajara

20, Guipúzcoa

21, Huelva

22, Huesca

23, Jaén

15, La Coruña

26, La Rioja

35, Las Palmas

24, León

25, Lérida

27, Lugo

28, Madrid

29, Málaga

30, Murcia

31, Navarra

32, Orense

34, Palencia

36, Pontevedra

37, Salamanca

38, Santa Cruz de Tenerife

40, Segovia

41, Sevilla

42, Soria

43, Tarragona

44, Teruel

45, Toledo

46, Valencia

47, Valladolid

48, Vizcaya

49, Zamora

50, Zaragoza

51, Ceuta

52, Melilla"