# Ejercicios Librerías

Descargar estos ejercicios

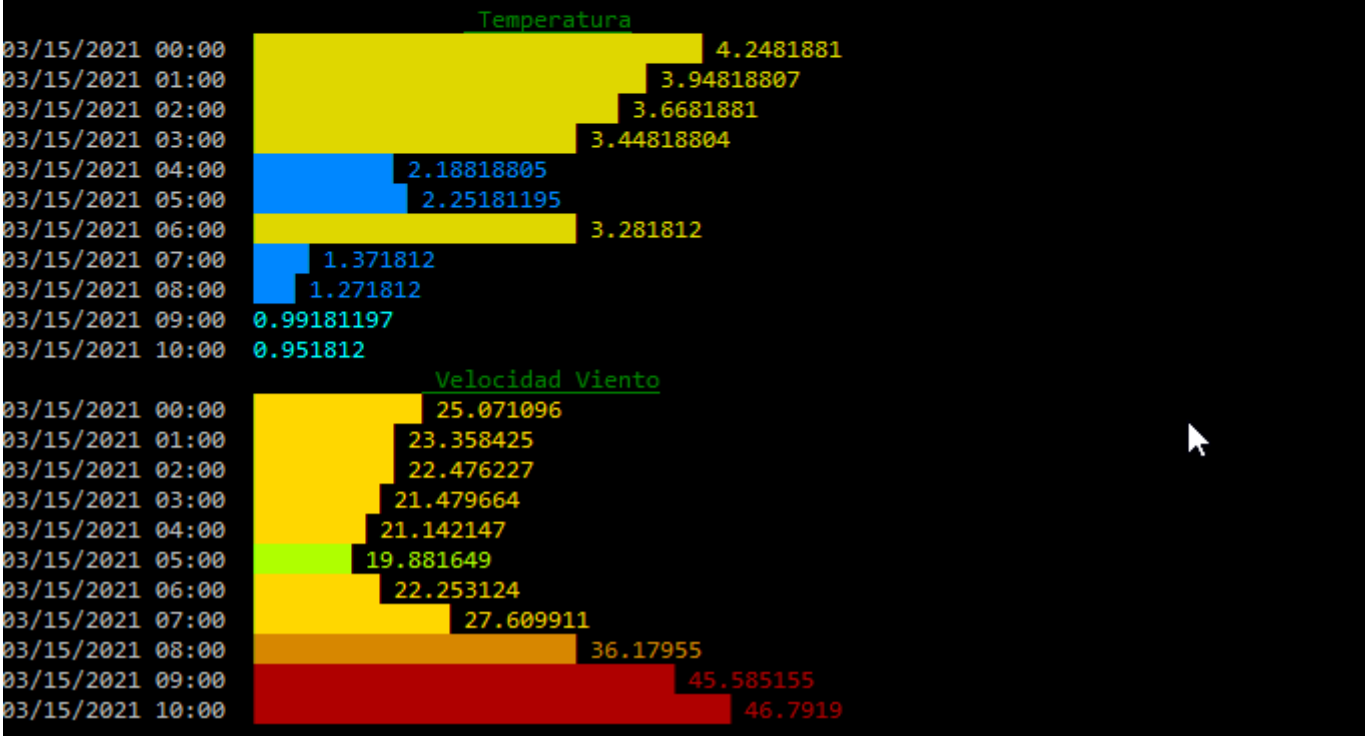
## Índice



## Ejercicio 1

> Este ejercicio se desarrollará a partir de una librería de terceros descargada de **NuGet** que facilita el diseño de la salida por consola, la librería se llama **spectre.console** y va a permitir, entre otras cosas, crear tablas y gráficos a partir de unos datos, Web Oficial Spectre Console. El ejercicio constará de leer la información del fichero **datos.csv**, pasado como recurso. A partir de la lectura del archivo y con la ayuda de la .dll, se creará una tabla y dos gráficos (uno con el dato temperatura y el otro con el dato velocidad del viento) con el aspecto parecido al de la imagen:

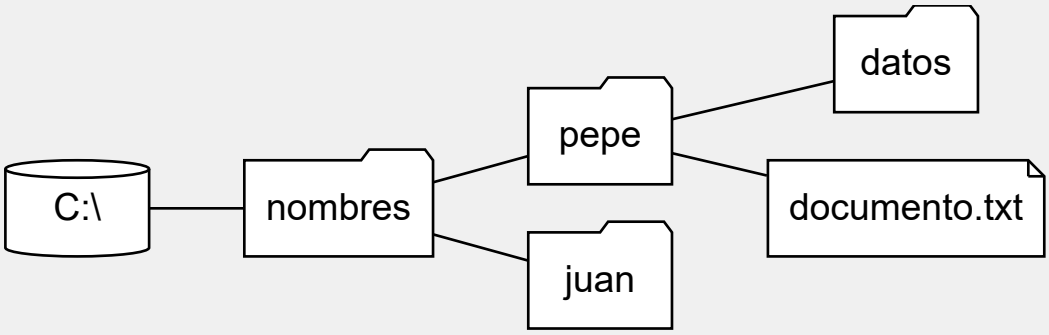
| timestamp | Temperature C ? | Relative Humidity % | Wind Speed km/h | Wind Direction ? |
|---|---|---|---|---|
| 03/15/2021 00:00 | 4.2481881 | 99.0 | 25.071096 | 195.83238 |
| 03/15/2021 01:00 | 3.94818807 | 97.0 | 23.358425 | 204.59012 |
| 03/15/2021 02:00 | 3.6681881 | 97.0 | 22.476227 | 211.90811 |
| 03/15/2021 03:00 | 3.44818804 | 97.0 | 21.479664 | 219.55966 |
| 03/15/2021 04:00 | 2.18818805 | 96.0 | 21.142147 | 227.07004 |
| 03/15/2021 05:00 | 2.25181195 | 96.0 | 19.881649 | 238.32454 |
| 03/15/2021 06:00 | 3.281812 | 96.0 | 22.253124 | 260.69006 |
| 03/15/2021 07:00 | 1.371812 | 95.0 | 27.609911 | 292.2176 |
| 03/15/2021 08:00 | 1.271812 | 94.0 | 36.17955 | 317.4195 |
| 03/15/2021 09:00 | 0.99181197 | 87.0 | 45.585155 | 335.25647 |
| 03/15/2021 10:00 | 0.951812 | 89.0 | 46.7919 | 343.10626 |



# Ejercicio 2

Crear un programa que tenga los siguientes métodos, que podrán llamarse mediante un menú con opción de salir con ESC. Usa las clases de utilidad **File y Directory** para resolver el ejercicio.

1. **`CreaArbolDeDirectorios`** : Que creará el siguiente árbol con directorios y ficheros incluidos, si ya existen, avisará mediante un mensaje.




2. **`EliminaDirectorio`** : Al que le pasarás la ruta de uno de los directorios del árbol y te lo eliminará, siempre y cuando exista, avisando en caso contrario.
3. **`EliminaFichero`** : Idem al anterior pero con ficheros.
4. **`MuestraInformación`** : Que mostrará todos los miembros de un directorio, siempre y cuando exista mostrando un mensaje en caso contrario.
5. **`MuestraAtributos`** : Que mostrará el estado de los atributos del fichero que le indiques.

# ☑ Ejercicio 3

Vamos a crear algunos comandos similares a los del S.O., que nos ayuden a entender mejor el funcionamiento de las clases **DirectoryInfo**, **FileInfo** y sus homólogas de utilidad. Para ello deberemos crear distintos programas, de forma que cada uno haga una de las siguientes funciones. El nombre de cada proyecto es el que aparece al principio de cada línea:

1. **listaCarpeta** - Se encargará de mostrar el contenido del directorio actual si no se le indica ruta, o del directorio que se indique en la ruta. Sino existe el directorio, se capturará la excepción mostrando un mensaje de error.
   Puedes usar la funcionalidad de las enumeraciones excluyentes para conocer el tipo de elemento:

```
//recuerda, podemos saber que elemento es un archivo si hacemos algo como lo siguiente
a.Attributes & FileAttributes.Archive==FileAttributes.Archive
```

Una posible salida si listamos la carpeta de usuario `listaCarpeta c:\users` , podría ser:

```
            All Users        Directory 07/12/2019 10:30:39
              Default        Directory 07/12/2019 10:03:44
         Default User        Directory 07/12/2019 10:30:39
          desktop.ini        Archive 07/12/2019 10:14:54
               Public        Directory 07/12/2019 10:14:52
                 xusa        Directory 07/10/2020 21:58:18
```

2. **creaDirectorio** - comando al que le pasas una ruta y te crea un directorio, tendrás que controlar si se indica más de una entrada o capturar las excepciones de ruta inválida.
3. **eliminaDirectorio** - comando al que le pasas una ruta y te elimina un directorio, tendrás que controlar si se indica más de una entrada o capturar las excepciones de ruta inválida.
4. **eliminaFichero** - comando al que le pasas una ruta y elimina un fichero, tendrás que controlar si se indica más de una entrada o capturar las excepciones de ruta inválida.
5. **copiaFicheros** - comando al que le pasas dos rutas: la primera de un directorio válido y la segunda de un directorio que no tiene porque existir. El comando se encargará de copiar todos los ficheros que hay en la primera ruta a la segunda, creando la carpeta si fuera necesario. Además deberas controlar las posibles excepciones.
   Este comando podrá, además, ejecutarse con la opción de añadir un filtro a la copia (solo se copiarán los archivos que cumplan el filtro, eso lo podemos hacer con la sobrecarga del método `EnumerateFile` ). Ejemplo del comando para copiar los ficheros que contengan en su nombre *windows*, desde la carpeta logs a la carpeta copia:

```
copiaFicheros c:\logs c:\copia *windows*
```

**Nota:** La ejecución y el paso de información al programa se realizará **a través de la línea de comandos**. La explicación está en el primer ejercicio del tema anterior de excepciones. Guarda todos los .cs en una carpeta Ejercicio3.