

CFGS

## **DESARROLLO DE APLICACIONES MULTIPLATAFORMA Y DESARROLLO DE APLICACIONES WEB**

M03A: Programación.

Unidad Formativa 3: Fundamentos de Gestión de ficheros.

UF3\_PAC01\_ENUNCIADO



## UF3. [PAC 01]. Fundamentos de Gestión de ficheros.

### INFORMACIÓN

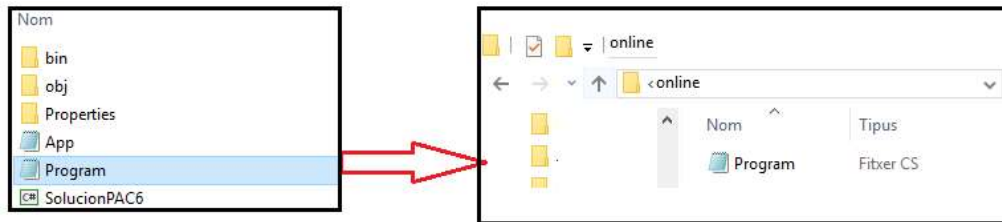
Para responder a las siguientes cuestiones deberás ayudarte del material didáctico y de internet.

#### Requisitos varios que deben cumplirse en vuestros trabajos:

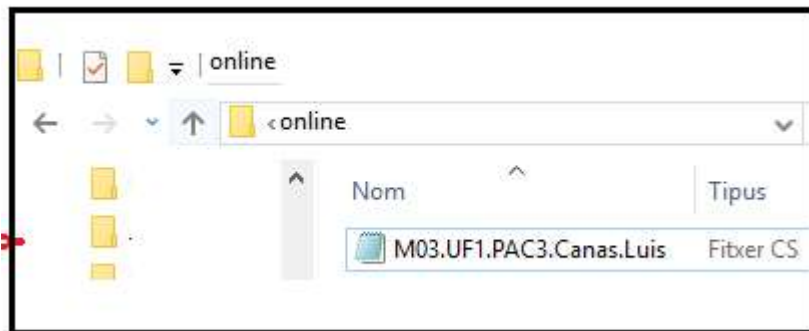
- En los ejercicios, si se requieren de cálculos, estos deben aparecer en la respuesta que planteéis.
- Siempre que utilicéis información de Internet para responder / resolver alguna pregunta, tenéis que citar la fuente (la página web) de dónde habéis sacado aquella información.
- Siempre que utilicéis información del material didáctico para responder / resolver alguna pregunta, tenéis que citar el tema y la página de dónde habéis sacado aquella información.
- No se aceptarán respuestas sacadas de Internet utilizando la metodología de copiar y pegar. Podéis utilizar Internet para localizar información, pero el redactado de las respuestas ha de ser vuestro.
- Las respuestas a las preguntas deben estar bien argumentadas, no se admiten respuestas escuetas o monosílabas.
- Se valorará la presentación de vuestro trabajo hasta con un punto y medio **(1)** de la nota final.

### Instrucciones PAC1 UF3

- La fecha de entrega es improrrogable y la tenéis especificada en el campus.
- **La PAC se debe realizar en C#**
- **En la primera línea del fichero, debe aparecer vuestro nombre, como comentario.**
- Para hacer la entrega, cuando tengáis la PAC terminada, debéis hacer lo siguiente:
  - Copiar el fichero Program.cs de vuestro proyecto, a donde queráis, por ejemplo a una carpeta que se llame “online”:



- Cambiar el nombre del fichero Program.cs, al siguiente: **M03A.UF3.PAC1.Apellido1.Nombre**. Los apellidos y nombres se escriben sin acentos, espacios, eñes....Por ejemplo, el alumno Luis Cañas Rojas, debe usar el siguiente nombre para el fichero de la PAC7: M03A.UF3.PAC7.Canas.Luis.



- **Éste es el fichero que debéis subir al campus.**

### REQUISITOS:

A parte de todo lo explicado anteriormente, se debe realizar la práctica usando funciones, ficheros y estructuras.

### PONDERACIÓN

Siempre que se cumplan los requisitos anteriores, se valorará de la siguiente manera

- 40% Tratamiento del fichero de entrada, con control de errores.
- 20% Almacenar datos en la estructura
- 20% Creación ficheros de salida y mostrar errores por pantalla.
- 10% Comentarios
- 10% Elegancia (uso de funciones, duplicidad de código, código “inútil”...)

### Enunciado PAC1 UF3

La empresa TENFE necesita un programa para gestionar sus trenes, y para gestionar las compras de billetes/asientos que se hacen. Para tal efecto, debemos realizar un programa que lea un fichero de texto con una serie de operaciones, (el programa debe pedir el nombre del fichero por teclado), y debe hacer lo siguiente:

- Crear el fichero ocupación.txt, con el nombre, número de plazas totales, y número de plazas disponibles de cada tren
- Indicar los errores encontrados

### FORMATO DEL FICHERO DE SALIDA

#### ocupacion.txt

Tren BC\_MA\_53 Plazas totales: 50 Plazas libres: 25

Tren LL\_SE\_48 Plazas totales: 25 Plazas libres: 0

Tren VA\_SE\_48 Plazas totales: 25 Plazas libres: 12

### FORMATO DEL FICHERO DE ENTRADA

El fichero está compuesto por líneas. Cada línea es una operación realizada en el centro de mando de TENFE. Y cada línea tiene la siguiente estructura:

EVENTO#NOMBRE#NUMERO (Salto de línea)

Los campos van separados por una almohadilla. El significado de los campos es el siguiente:

- EVENTO: Solo hay dos tipos de Eventos. El evento CREAR y el evento COMPRAR (en mayúsculas). El CREAR, creará un tren, con el *nombre* dado, de *numero* asientos. El COMPRAR, significa que se han comprado *número* billetes(asientos) para el tren con el *nombre* dado
- NOMBRE: es el nombre del TREN. Siempre debe seguir la misma estructura: dos letras mayúsculas, seguidas de un guion bajo, seguidas de dos letras mayúsculas más, seguidas de otro guion bajo, seguido de dos dígitos.
- NUMERO: es un número. En caso de ser un evento CREAR, significa el número de asientos disponibles en el tren. En caso de ser un evento COMPRAR, significa el número de billetes que se han comprado para ese tren.

Dos ejemplos de línea del fichero de entrada seria:

CREAR LL\_SE\_48 25 (salto de línea)

COMPRAR LL\_SE\_48 3 (salto de línea)

La primera línea significa que se CREA el tren LL\_SE\_48 y tiene 25 plazas disponibles. La segunda significa que se compran 3 billetes para ese tren.

## LINEAS ERRONEAS

Todas las líneas, para ser correctas, deben seguir el formato especificado anteriormente.

Los siguientes casos pueden suceder y son errores, y por tanto, no deben ser tratadas las líneas. Deben mostrarse por pantalla indicando el número de línea junto al error.

- La línea no tiene 3 campos
- El nombre del tren no sigue el formato correcto
- El evento no es correcto
- El número no es un numero
- Evento COMPRA de un tren con menos asientos disponibles que billetes se desea comprar
- Evento COMPRA de un tren inexistente
- Evento CREAM de un tren existente

## FUNCIONAMIENTO DEL PROGRAMA

El programa debe incorporar una estructura de datos (struct) dónde almacenar el nombre de los trenes, junto a la cantidad de plazas disponibles, y a la cantidad de plazas libres.

Así, cuando leemos una línea correcta, si detectamos un evento CREAM de un tren que no existía en la estructura, lo crearemos y le asignaremos el número de asientos disponibles, y asientos libres (son los mismos). Si el tren ya existía, es un error.

Si detectamos un evento COMPRAR de un tren que no existía en la estructura, es un error. Si existe, y no tiene suficientes billetes, también es error. En caso contrario, se actualizará el número de plazas libres de ese tren.

Una vez procesado el fichero de entrada, el programa debe crear el fichero de salida, mostrar los errores por pantalla, e indicar el número de líneas tratadas e indicar cuántas de ellas son incorrectas.

## FUNCIONES RECOMENDADAS

La clase string posee algunos métodos interesantes para realizar esta práctica, por lo que recomiendo su uso.

- Substring, para conseguir una subcadena entre dos posiciones
- Split, para partir una cadena a un array a partir de un carácter
- Para comparar caracteres de un string, os será útil lo siguiente:
  - `if (Char.IsLetter(texto[i]))`
  - `if (Char.IsUpper(texto[i]))`

Si necesitáis ejemplos de cómo usar estas funciones, pedirlo por el foro.

## EJEMPLO DE EJECUCIÓN

En la siguiente página podéis ver un ejemplo de ejecución. También encontraréis en el campus el fichero de entrada usado y el de salida que se ha generado

```

Que fichero desea abrir?
UF3_ENTRADA.TXT

*** Leemos línea 1-->CREA#LL_SE_48#25
Error en línea 1: Evento incorrecto

*** Leemos línea 2-->COMP#LL_SE_48#25
Error en línea 2: Evento incorrecto

*** Leemos línea 3-->CREAR
Error en línea 3: Numero de # incorrecto

*** Leemos línea 4-->CREAR#LL_SE_48#25#JJJ
Error en línea 4: Numero de # incorrecto

*** Leemos línea 5-->CREAR#Lx_SE_48#25
Error en línea 5: Tren incorrecto

*** Leemos línea 6-->CREAR#BCN_SE_48#25
Error en línea 6: Tren incorrecto

*** Leemos línea 7-->CREAR#LL_SEV_48#25
Error en línea 7: Tren incorrecto

*** Leemos línea 8-->CREAR#LL_Se_48#25
Error en línea 8: Tren incorrecto

*** Leemos línea 9-->CREAR#LL_SE_48x#25
Error en línea 9: Tren incorrecto

*** Leemos línea 10-->CREAR#LL_SE_481#25
Error en línea 10: Tren incorrecto

*** Leemos línea 11-->CREAR#LL_SE_48_12#25
Error en línea 11: Tren incorrecto

*** Leemos línea 12-->CREAR#LL_SE_48#25

*** Leemos línea 13-->CREAR#BC_MA_55#255

*** Leemos línea 14-->CREAR#LL_SE_48#25
Error en línea 14: Evento CREAR con tren existente

*** Leemos línea 15-->COMPRAR#BC_MA_55#55

*** Leemos línea 16-->COMPRAR#BC_SE_55#55
Error en línea 16: Evento COMPRAR con tren inexistente

*** Leemos línea 17-->COMPRAR#LL_SE_48#55
Error en línea 17: Evento COMPRAR con numero de billetes insuficiente
Se trataron 17 líneas, siendo 14 incorrectas
Tren LL_SE_48 Plazas totales: 25 Plazas libres:25
Tren BC_MA_55 Plazas totales: 255 Plazas libres:200
  
```

# ¡Buen trabajo!

