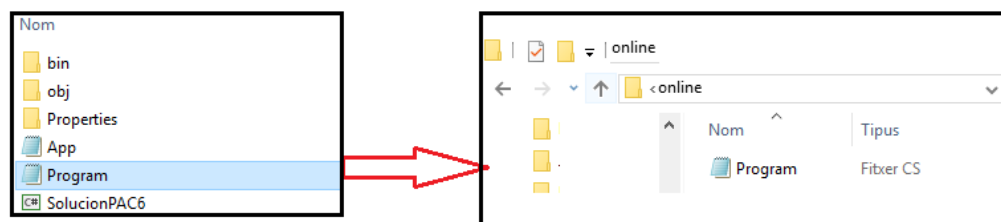
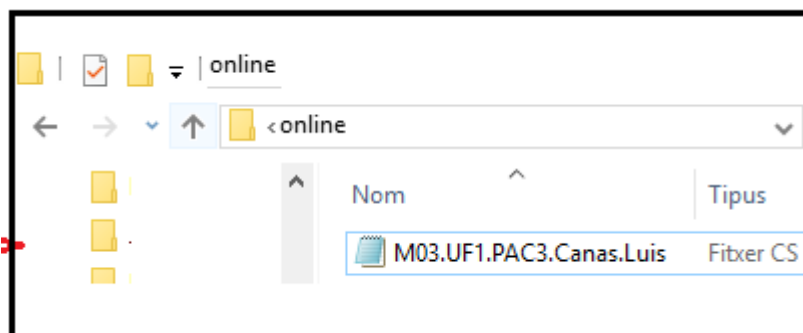


## Instrucciones UF1 PAC5

- La fecha de entrega es improporrogable 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.UF1.PAC5.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 PAC5: M03A.UF1.PAC5.Canas.Luis.



- Éste es el fichero que debéis subir al campus.

## Evaluación genérica de los ejercicios

Los ejercicios entregados deben **compilar** para ser evaluados. Para cada ejercicio se verificará todo lo que se expone a continuación.

- **Funcionalidad(75%):** El programa debe mostrar resultados correctos para entradas razonables. Haced pruebas en casos "críticos" o "fronterizos". Por ejemplo, si el programa debe pedir la edad del usuario, obviamente introduciré un número cuando yo lo pruebe(no letras), pero... qué pasa si entro el -7? tiene sentido? **Es obligatorio el uso de bucles para trabajar con arrays.**
- **Comentarios (10%):** El código debe ser comentado, explicando la manera que habéis usado para programar vuestra solución al problema.
- **Elegancia y eficiencia de la solución(15%):** el programa no debe hacer operaciones que malgasten tiempo de cálculo. El código debe estar bien estructurado, con nombres de variables claros.

## Prohibiciones

**No se pueden crear funciones ni usar ficheros**

## Enunciado UF1 PAC5

1. – Realizad un programa con un bucle principal, que permita escoger entre la opcion1, opcion2, y la opcion3 (salir del programa). Cada opción (la 1 y la 2), debe solucionar, respectivamente, los ejercicios planteados.

**NO HACE FALTA CONTROLAR ERRORES. TODAS LAS ENTRADAS SERAN CORRECTAS, NUMEROS POSITIVOS EN CASO DEL PRIMER EJERCICIO, Y NUMEROS DEL 1 AL 9 EN EL SEGUNDO.**

### Opcion1

En los sorteos de la lotería más mediáticos, como el del Gordo de Navidad o el del Niño, es habitual que los fetichistas busquen números con algún tipo de significado extraño o que los matemáticos y estadísticos se entretengan informando sobre los números más habituales.

Esto último es interesante si queremos maximizar la probabilidad de ganar algo. Dado que los números que terminan con el mismo dígito que el del premio principal tienen un reintegro, una buena forma de intentar, al menos, no perder dinero es jugar un número con la terminación más habitual en el histórico de sorteos.

Esta opción empieza pidiendo el número N de sorteos que se nos informará. Seguidamente se introducirán los N sorteos.

El programa debe decir, dados los números de lotería, la terminación (de un dígito) que ha sido más premiada a lo largo de los sorteos. En caso que varias terminaciones aparezcan el máximo de veces, debemos decir VARIAS.

Ejemplo de ejecución en la siguiente página.

### Opcion2

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

El sudoku es un pasatiempo lógico que consiste en rellenar una cuadrícula de 9×9 casillas dividida en nueve regiones 3×3 (las separadas con líneas más gruesas en la imagen) con los números del 1 al 9 de tal forma que no se repitan números en ninguna fila, columna o región. El sudoku inicialmente se presenta con algunas casillas ya rellenas, a modo de *pistas*, y el jugador debe

deducir los valores de las casillas vacías. Si el sudoku está bien planteado, la solución es única.

Dado un sudoku completamente relleno por teclado, ¿sabrías construir un programa que comprobara si es correcto (es decir, cada fila, columna o región contiene los números del 1 al 9 exactamente una vez)?

El sudoku se debe leer por FILAS. Y El programa debe decir SI, si es correcto, o NO, si no es correcto.

Ejemplo de ejecución en la siguiente página.

(Recordad que en negrita aparece lo que introduce el usuario por teclado).

## **EJEMPLO DE EJECUCIÓN DEL EJERCICIO 1**

*Por ejemplo:*

Numero de Sorteos? **3**

Numero 1: **00004**

Numero 2: **03847**

Numero 3: **39804**

La mejor terminación es la 4.

*Otro ejemplo:*

Numero de Sorteos? **4**

Numero 1: **15154**

Numero 2: **03847**

Numero 3: **39804**

Numero 4: **35997**

La mejor terminación es VARIAS.

## **EJEMPLO DE EJECUCIÓN DE LA OPCION2**

El sudoku se debe leer por FILAS. Es decir, el programa debe hacer algo semejante a:

Dado por ejemplo el sudoku de la imagen de la página anterior...

Entra fila 1 columna 1: **4**

Entra fila 1 columna 2: **1**

Entra fila 1 columna 3: **3**

...

Entra fila 1 columna 9: **9**

Entra fila 2 columna 1: **5**

...

...

...

El sudoku SI es correcto.