

## FP 2

### Práctica 2

**Fecha de entrega: 30 de mayo de 2021**

### ***Puzzle con matrices.***

En esta práctica mejoraremos la interfaz de entrada de la práctica 1. Para ello hemos creado un catálogo con la información de los puzzles que tenemos disponibles. La aplicación nos permitirá elegir el puzzle que queremos resolver a partir de su nombre, en lugar de tener que utilizar el nombre del fichero. La aplicación también permitirá añadir nuevos puzzles al catálogo.

### **1. Nueva funcionalidad de la práctica**

Al comenzar la aplicación se mostrará un menú con las siguientes opciones:

1. Resolver un puzzle 1D
2. Resolver un puzzle 2D
3. Añadir un puzzle al catálogo
0. Salir

Elige una opción:

Si el jugador elige resolver un puzzle, la aplicación mostrará una lista con los nombres de todos los puzzles disponibles de la categoría seleccionada y pedirá al usuario que seleccione el número del puzzle que quiere resolver. La lista estará ordenada inicialmente de menor a mayor por el número de movimientos máximos que se pueden realizar para resolverlo. A igualdad de movimientos máximos, la lista se ordenará por el nombre del puzzle. Por lo tanto, si se elige resolver un puzzle 1D se mostrará el listado contenido en la Tabla 1, mientras que si se elige resolver un puzzle en 2D se mostrará el listado contenido en la tabla 2.

Tabla 1

Elige una opción: 1
En esta versión están disponibles los siguientes retos
1 Torre con un máximo de 2 movimientos
2 Intercambiar columnas con un máximo de 3 movimientos
3 Intercambiar diagonales con un máximo de 3 movimientos
4 Intercambiar filas con un máximo de 3 movimientos
5 Voltear columna con un máximo de 4 movimientos
6 Voltear diagonal con un máximo de 4 movimientos
7 Voltear fila con un máximo de 4 movimientos
0 salir
-1 Mostrar la lista ordenada de mayor a menor
-2 Mostrar la lista ordenada de menor a mayor

Tabla 2

Elige una opción: 2

En esta versión están disponibles los siguientes retos

- 1 Rotar a la derecha con un máximo de 1 movimientos
- 2 Voltear respecto a la horizontal con un máximo de 1 movimientos
- 3 Voltear respecto a la vertical con un máximo de 1 movimientos
- 4 Intercambiar las posiciones vecinas de las posiciones con un máximo de 3 movimientos
- 5 Voltear respecto a la diagonal principal con un máximo de 3 movimientos
- 0 salir
- 1 Mostrar la lista ordenada de mayor a menor
- 2 Mostrar la lista ordenada de menor a mayor

Observa que las dos últimas opciones del listado de ambas Tablas son -1 y -2 que consisten en ordenar la lista de mayor a menor o de menor a mayor por el número de intentos y nombre en el caso de igualdad. Cuando se termine de seleccionar el puzzle con el que se quiere jugar, la lista se volverá a ordenar de menor a mayor, ya que este es su orden por defecto durante toda la ejecución del programa. Una vez elegido el puzzle a resolver, este se mostrará al usuario y se continuará resolviendo el puzzle como en la práctica 1.

Si el usuario selecciona añadir un puzzle al catálogo, el sistema mostrará un listado con todos los puzzles, primero mostrará el listado de los puzzles 1D y después el listado de los puzzles 2D. Ambos listados estarán ordenados, en orden creciente, por número de intentos y a igualdad de número de intentos en orden creciente del nombre del puzzle (es decir, en el orden en que se mantiene el catálogo durante toda la ejecución del programa). A continuación pedirá el nombre del nuevo puzzle y el nombre del fichero en que se encuentra, cargará las matrices del fichero y el máximo número de intentos, y si el proceso es correcto se añadirá la información a la lista ordenada del catálogo. Por último mostrará la lista de todos los puzzles que hay en el catálogo del tipo del nuevo puzzle. El puzzle que se ha añadido debe figurar en este listado.

Si el usuario desea salir de la aplicación (opción 0), el estado de las listas de los puzzles reunidos, con posibles nuevos retos, se guardará para las próximas partidas.

Recordad que antes de terminar la ejecución del programa se debe devolver al sistema toda la memoria dinámica que se haya reservado.

## 2. Modificaciones en la implementación

Implementaremos dos nuevos módulos `Puzzle` y `ListaPuzzles` con los tipos y funciones necesarios para manejar el catálogo de puzzles. El módulo `JuegoPM` que controlaba la resolución de un puzzle se encarga ahora de gestionar el catálogo, pasando las funciones que controlaban la resolución de un puzzle al nuevo módulo `Puzzle`. Los módulos `Matrices` y `Coordenadas` no se deben modificar.

### Formato del archivo de entrada (`datosPuzzles.txt`)

El catálogo se encuentra en el fichero "`datosPuzzles.txt`". El fichero comienza con el número de puzzles que se tienen del tipo 1D, seguido de la descripción de estos. Después se encuentra el número de puzzles de tipo 2D seguido de la descripción de estos. Cada puzzle se describe en dos líneas, en la primera línea se muestra el nombre del puzzle (cadena de caracteres que puede tener blancos), en la segunda línea se muestra el nombre del fichero donde se encuentra la información de ese puzzle. La lista de los dos tipos de puzzles se encuentra ordenada en el fichero, en orden creciente, por el máximo número de intentos y a igualdad de número de intentos por orden creciente de los nombres de los puzzles.

```
N1D      // Número de puzzles del tipo 1D
N1       // Nombre del puzzle 1 de tipo 1D
F1       // Fichero que contiene el puzzle 1
...      // 2 N1D - 2 filas con los nombres y ficheros de los puzzles 1D
N2D      // Número de puzzles del tipo 2D
N2       // Nombre del puzzle 1 de tipo 2D
F2       // Fichero que contiene el puzzle 1
...      // 2 N2D - 2 filas con los nombres y ficheros de los puzzles 2D
```

### Módulo `Puzzle`

En este módulo se definen los tipos y funciones necesarios para manejar la información correspondiente a un puzzle y resolverlo.

Declara un tipo estructurado `tPuzzle` con el nombre del puzzle y el nombre del fichero en que se encuentra, el tipo de puzzle, el número de movimientos máximo y las matrices inicial y final. Este tipo sustituye al tipo `tJuegoPM` del módulo `JuegoPM` de la práctica anterior.

Declara en este módulo las siguientes funciones que se encontraban en el módulo `JuegoPM` y elimínalas de aquel. No es necesario modificar la implementación realizada de estas funciones en la primera parte aunque modificaremos el nombre de la función

mainJuegoPM ya que ahora no se encuentra en ese módulo. Sobrecarga los operadores relacionales que necesites sobre el tipo Puzzle para realizar las operaciones de inserción ordenada y búsqueda del módulo ListaPuzzles

- `void mainPuzzle()`: función que resuelve un puzzle. Corresponde a la antigua función mainJuegoPM. Dependiendo de la implementación realizada por el alumno en la primera parte puede necesitar algún pequeño ajuste en el código.
- `bool cargar(tPuzzle & jpm, string tipo)`: abre el fichero que corresponda y llama a la función cargar del módulo Matriz.
- `void mostrar(const tPuzzle & jpm)`: muestra el estado del puzzle utilizando las facilidades del módulo utilidadesSYS. Debe mostrar primero la imagen que se está modificando y a continuación la imagen objetivo. Debe mostrar también el número de intentos que quedan.
- `bool jugar(tPuzzle & jpm)`: permite realizar las acciones necesarias para jugar y controla si se ha llegado al límite de acciones permitidas;
- `void accion(tPuzzle & jpm)`: según el comando de acción tecleado por el usuario, llama a la acción correspondiente definida en el módulo Matriz;

## Módulo ListaPuzzles

En este módulo se definen los tipos y funciones necesarios para manejar el catálogo de puzzles. Debe incluir el módulo Puzzle.

Define una constante MAX\_PUZZLES de valor 100, que representa el número máximo de puzzles, y un tipo lista, tListaPuzzles, que contiene **un array estático de punteros a variables dinámicas** de tipo tPuzzle. El tamaño máximo del array es el número máximo de puzzles. Esta lista se mantiene siempre ordenada en orden creciente del número de intentos máximo para realizar el puzzle. Si el usuario pide verla en orden inverso, opción -1 o -2, según corresponda, los punteros que contiene el array se modifican convenientemente, pero nunca se deben intercambiar o sobrescribir los datos de tipo tPuzzle, pues perderíamos eficiencia.

Declara también un array, tPuzzlesReunidos, de 2 posiciones, cada una de ellas de tipo tListaPuzzles. La primera lista representa el catálogo de puzzles de tipo 1D y la segunda lista el catálogo de puzzles de tipo 2D.

Se implementarán al menos las siguientes funciones de la interfaz:

- `void` `inicializar(tPuzzlesReunidos & jr)`: inicializa las dos listas de puzzles.
- `bool` `cargar (tPuzzlesReunidos & jr)`: carga los datos contenidos en el fichero `datosPuzzles.txt`. Los datos se encuentran ordenados en el fichero en el mismo orden en que deben estar en la lista, en orden creciente del número de intentos máximo para realizar el puzzle y a igualdad de intentos por orden alfabético del nombre del puzzle.
- `void` `guardar(const tPuzzlesReunidos & listas)`: guarda en el fichero `datosPuzzles.txt` la lista de puzzles 1D y 2D con el mismo formato del fichero de entrada, y ordenados según están en las listas.
- `int` `elegirPuzzle (tListaPuzzles & lp)`: muestra la lista de puzzles y permite al usuario elegir uno de ellos. Debe dar la opción de mostrar la lista de mayor a menor número de movimientos o a la inversa.
- `bool` `insertarOrdenado(tListaPuzzles & l, tPuzzle * p)`: Inserta ordenado en la lista de puzzles un nuevo puzzle.

Se implementarán al menos las siguientes funciones no pertenecientes a la interfaz:

- `bool` `buscar(const tListaPuzzles & lista, tPuzzle* puzzle, int &pos)`: busca en una lista ordenada de puzzles un puzzle.

## Módulo JuegoPM

En este módulo dejamos las funciones que se encargan de la gestión del catálogo de puzzles.

- Se mantiene la constante `numOpc`.
- Se modifica la función `menu` para que muestre el nuevo menú.
- Se implementa una nueva función `mainPuzzlesReunidos()` que carga los datos de entrada, muestra el menú y gestiona las opciones seleccionadas por el usuario. Cuando el usuario pide salir del programa guarda en el fichero el catálogo de puzzles y libera la memoria dinámica que se haya utilizado.
- El resto de tipos y funciones que había en este módulo en la práctica 1 se eliminan de él.

## 3. Entrega de la práctica

La práctica se entregará en el Campus Virtual por medio de una tarea, que permitirá subir los archivos con extensiones `h` y `cpp` con el código fuente. Uno de los dos miembros del grupo será el encargado de subirlo, no lo suben los dos.

Recordad poner el nombre de los miembros del grupo en un comentario al principio de cada archivo del código fuente.

**No se permite** el uso de variables globales, ni el uso de instrucciones de salto: goto, exit, continue, break (salvo en instrucciones switch), ni return dentro de bucles.

### **Se valorará**

- 1) La estructuración del código: instrucciones condicionales que cubran todos los casos, instrucciones iterativas que sigan los esquemas vistos en clase, definición y uso de constantes y funciones.
- 2) Documentación clara y concisa

## **4. Ejemplo de ejecución;**

Página de códigos activa: 1252

- 1 Resolver un puzzle - 1D
- 2 Resolver un puzzle - 2D
- 3 Añadir un puzzle al catálogo
- 0 Salir

Elige una opción: 1

En esta versión están disponibles los siguientes retos

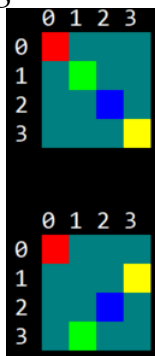
- 1 Torre con un máximo de 2 movimientos
- 2 Intercambiar columnas con un máximo de 3 movimientos
- 3 Intercambiar diagonales con un máximo de 3 movimientos
- 4 Intercambiar filas con un máximo de 3 movimientos
- 5 Voltar columna con un máximo de 4 movimientos
- 6 Voltar diagonal con un máximo de 4 movimientos
- 7 Voltar fila con un máximo de 4 movimientos
- 0 salir
- 1 Ordenar la lista de mayor a menor
- 2 Ordenar la lista de menor a mayor

Elige un reto: -1

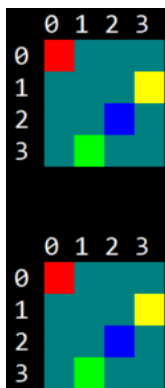
En esta versión están disponibles los siguientes retos

- 1 Voltear fila con un máximo de 4 movimientos
- 2 Voltear diagonal con un máximo de 4 movimientos
- 3 Voltear columna con un máximo de 4 movimientos
- 4 Intercambiar filas con un máximo de 3 movimientos
- 5 Intercambiar diagonales con un máximo de 3 movimientos
- 6 Intercambiar columnas con un máximo de 3 movimientos
- 7 Torre con un máximo de 2 movimientos
- 0 salir
- 1 Ordenar la lista de mayor a menor
- 2 Ordenar la lista de menor a mayor

Elige un reto: 4



SF 1 3



Te quedan 2 intentos.

Lo has conseguido

- 1 Resolver un puzzle - 1D
- 2 Resolver un puzzle - 2D
- 3 Añadir un puzzle al catálogo
- 0 Salir

Elige una opción: 2

En esta versión están disponibles los siguientes retos

- 1 Rotar a la derecha con un máximo de 1 movimientos
- 2 Voltrear respecto a la horizontal con un máximo de 1 movimientos
- 3 Voltrear respecto a la vertical con un máximo de 1 movimientos
- 4 Intercambiar las posiciones vecinas de las posiciones con un máximo de 3 movimientos
- 5 Voltrear respecto a la diagonal principal con un máximo de 3 movimientos
- 0 salir
- 1 Ordenar la lista de mayor a menor
- 2 Ordenar la lista de menor a mayor

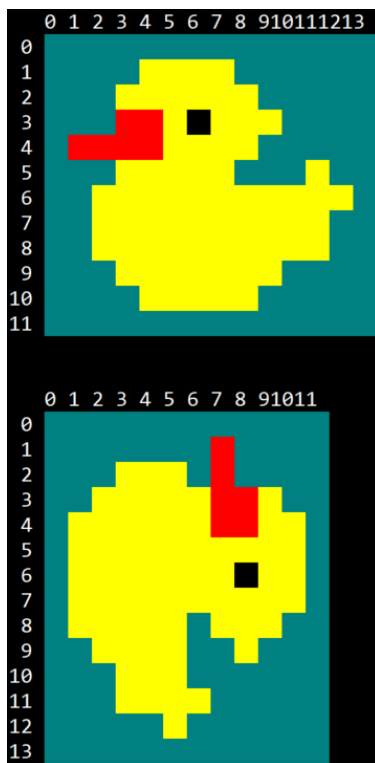
Elige un reto: -1

En esta versión están disponibles los siguientes retos

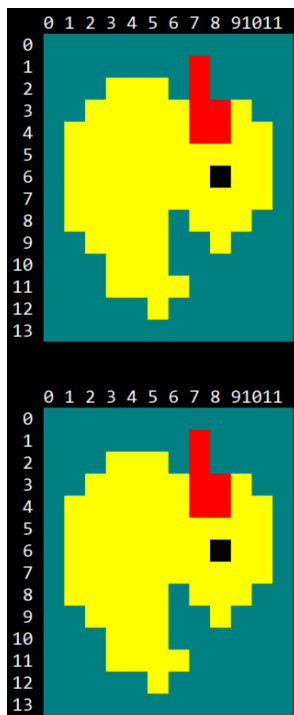
- 1 Voltrear respecto a la diagonal principal con un máximo de 3 movimientos
- 2 Intercambiar las posiciones vecinas de las posiciones con un máximo de 3 movimientos
- 3 Voltrear respecto a la vertical con un máximo de 1 movimientos
- 4 Voltrear respecto a la horizontal con un máximo de 1 movimientos
- 5 Rotar a la derecha con un máximo de 1 movimientos
- 0 salir
- 1 Ordenar la lista de mayor a menor
- 2 Ordenar la lista de menor a mayor

Elige un reto: 5





RD



Te quedan 0 intentos.

Lo has conseguido

- 1 Resolver un puzzle - 1D
- 2 Resolver un puzzle - 2D
- 3 Añadir un puzzle al catálogo
- 0 Salir

Elige una opción: 3

#### Juegos 1D

- 1 Torre con un máximo de 2 movimientos
- 2 Intercambiar columnas con un máximo de 3 movimientos
- 3 Intercambiar diagonales con un máximo de 3 movimientos
- 4 Intercambiar filas con un máximo de 3 movimientos
- 5 Voltar columna con un máximo de 4 movimientos
- 6 Voltar diagonal con un máximo de 4 movimientos
- 7 Voltar fila con un máximo de 4 movimientos

#### Juegos 2D

- 1 Rotar a la derecha con un máximo de 1 movimientos
- 2 Voltar respecto a la horizontal con un máximo de 1 movimientos
- 3 Voltar respecto a la vertical con un máximo de 1 movimientos
- 4 Intercambiar las posiciones vecinas de las posiciones con un máximo de 3 movimientos
- 5 Voltar respecto a la diagonal principal con un máximo de 3 movimientos

Escriba el nombre del puzzle: Cara misteriosa

Escriba nombre del fichero en que se encuentra: cara\_1D.txt

#### Juegos 1D

- 1 Torre con un máximo de 2 movimientos
- 2 Cara misteriosa con un máximo de 3 movimientos
- 3 Intercambiar columnas con un máximo de 3 movimientos
- 4 Intercambiar diagonales con un máximo de 3 movimientos
- 5 Intercambiar filas con un máximo de 3 movimientos
- 6 Voltar columna con un máximo de 4 movimientos
- 7 Voltar diagonal con un máximo de 4 movimientos
- 8 Voltar fila con un máximo de 4 movimientos

Fichero añadido correctamente. Seleccione su siguiente operación.

- 1 Resolver un puzzle 1D
- 2 Resolver un puzzle 2D
- 3 Añadir un puzzle al catálogo
- 0 Salir

Elige una opción.