



# Programación y estructuras de datos avanzadas. – PED1.

Curso 2020-2021.

Paulino Esteban Bermúdez Rodríguez. – 09146352B  
pbermudez30@alumno.uned.es

UNED - LAS TABLAS.



## Contenido

ENUNCIADO DE LA PRÁCTICA.....	3
DISEÑO DEL ALGORITMO. ....	3
CÓDIGO: .....	6
RESULTADOS DEL ALGORITMO. ....	8
BIBLIOGRAFÍA. ....	11

## ENUNCIADO DE LA PRÁCTICA

Necesitamos organizar un torneo de tenis con  $n$  jugadores en donde cada jugador ha de jugar exactamente una vez contra cada uno de sus posibles  $n-1$  competidores, y además ha de jugar un partido cada día, teniendo a lo sumo un día de descanso en todo el torneo. Se supone que hay campos de tenis suficientes para jugar cada día todos los partidos necesarios.

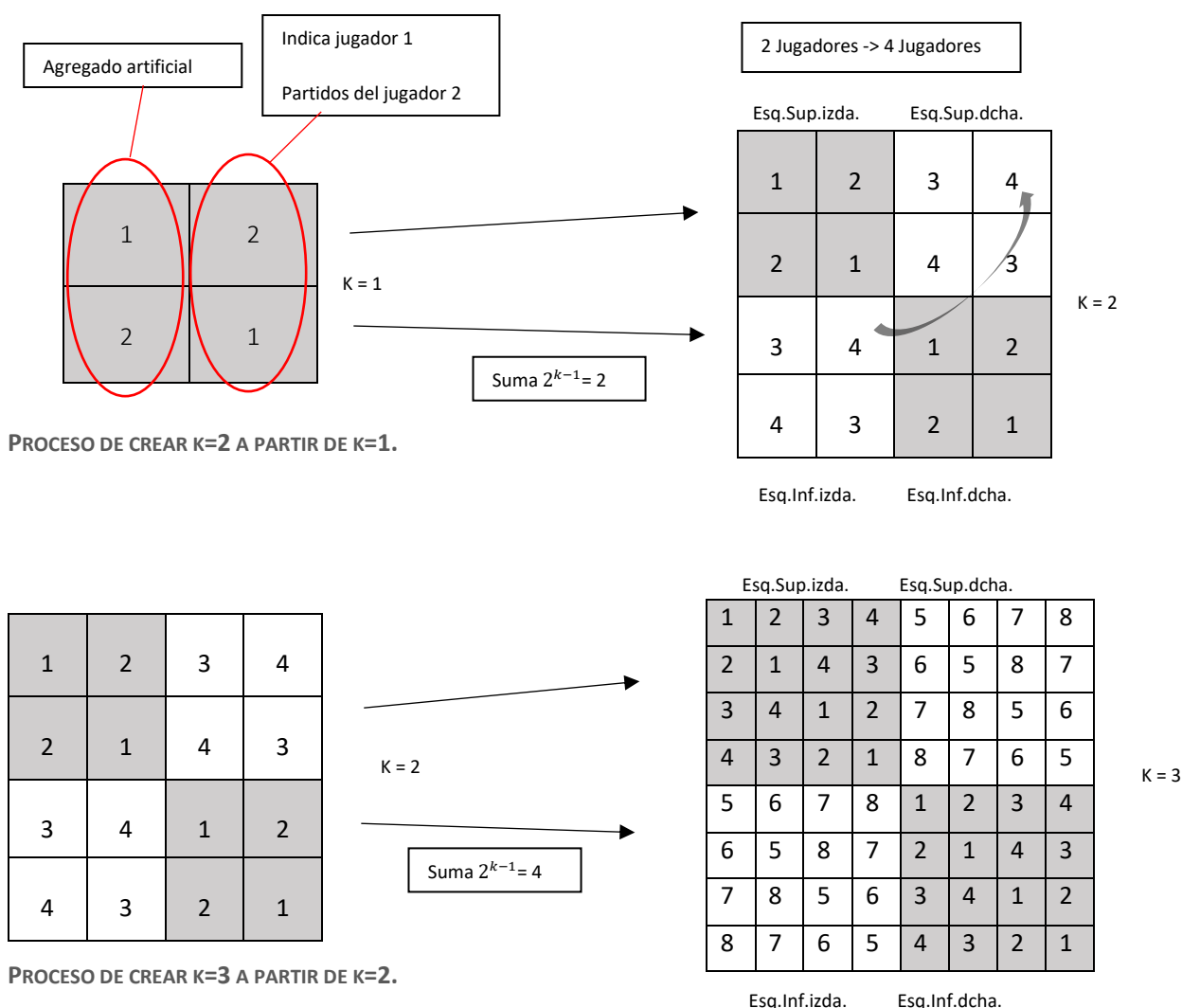
Se pide diseñar para resolverlo un algoritmo basado en el esquema de Divide y Vencerás.

Ver enunciado completo: [enunciadoPED](#).

## DISEÑO DEL ALGORITMO.

**Opción seleccionada:** Se supondrá que  $n$  es cualquier número natural mayor de 1. El torneo se realizará en  $n-1$  días si  $n$  es par o en  $n$  días si es impar.

De acuerdo con el requisito del enunciado, el calendario del juego está diseñado como una tabla con  $n$  filas y  $n-1$  columnas. En la fila  $i$ -ésima y la columna  $j$ -ésima de la tabla, complete el juego que el  $i$ -ésimo jugador enfrentó en el  $j$ -ésimo día. Entre ellos  $1 \leq i \leq n$  y  $1 \leq j \leq n-1$ .





Tomando como ejemplo un  $n=8$ , para la explicación de algoritmo, de acuerdo con la estrategia de dividir y vencerás, podemos separar a todos los jugadores en dos mitades, y el programa de juego de  $n$  jugadores puede determinarse por el programa de juego de  $n/2$  jugadores. Este proceso de solución es un proceso iterativo de abajo hacia arriba, donde la esquina superior izquierda y la esquina inferior izquierda son los horarios de competencia de los jugadores 1 a 4 y 5 a 8 respectivamente.

N
1
2
3
4
5
6
7
8

De acuerdo con esto, todos los números de la esquina superior izquierda se corresponden entre sí. Copie la posición en la esquina inferior derecha y copie todos los números de la esquina inferior izquierda en la esquina superior derecha de acuerdo con sus posiciones correspondientes.

N							
1 (1,1)	2 (1,2)	3	4	5 (1,5)	6	7	8
2	1	4	3	6	5	8	7
3	4	1	2	7	8	5	6
4	3	2	1	8	7	6	5
5 val(1,5)	6	7	8	1 val(1,1)	2 val(1,2)	3	4
6							
7							
8							

De esta forma se organizan los horarios de competición de los jugadores 1 a 4 y 5 a 8 en los próximos 4 días respectivamente. La situación con varios jugadores se puede deducir por analogía.

En cada iteración, el problema se divide en 4 partes:

- Esquina superior izquierda,
- Esquina inferior izquierda,
- Esquina superior derecha y
- Esquina inferior derecha.

El algoritmo es como sigue:

- Inicializa la matriz  $M[N][N]$ , ingresa el número de equipos  $n$ (par).
- Primero cuando  $n = 0$  y  $n=1$ , la matriz  $M[N][N] = 0$  y  $M[N][N] = 1$ , respectivamente, pero el enunciado nos pide que sea un entero positivo, mayor a 1, por ende, se anulan a la resolución del algoritmo.
- Luego, cuando  $N > 1$ , se divide el problema en dos y genera de forma recursiva la matriz en la esquina superior izquierda.
- De acuerdo con la matriz creada, generamos la matriz de la esquina superior derecha correspondiendo al elemento  $+ N/2$ .
- Los elementos inferiores son la copia exacta, de la matriz superior, pero en posición diagonal.

Día \ N	1	2	3	4	5	6	7
	1	2	3	4	5	6	7
1	2	3	4	5	6	7	8
2	1	4	3	6	5	8	7
3	4	1	2	7	8	5	6
4	3	2	1	8	7	6	5
5	6	7	8	1	2	3	4
6	5	8	7	2	1	4	3
7	8	5	6	3	4	1	2
8	7	6	5	4	3	2	1

De forma análoga, se aplica el mismo algoritmo para un número de jugadores impares,  $n=7$ , y tenemos que:

- Inicializar la matriz a  $[N][N]$ , se ingrese el número de equipos  $n$ (impar).
- Para los valores 0 y 1, se anulan a la resolución del algoritmo, de igual forma como en el caso anteriormente comentado.
- Luego, cuando  $N > 1$ , se divide el problema en dos y genera de forma recursiva la matriz en la esquina superior izquierda.
- De acuerdo con la matriz creada, generamos la matriz de la esquina superior derecha correspondiendo al elemento  $+ N/2$ .
- Los elementos inferiores son la copia exacta, de la matriz superior, pero en posición diagonal.
- Para no tener un error a la hora de rellenar la matriz, añadimos un jugador más como jugador auxiliar, en este caso será  $n=8$ , el cual, jugador  $N$  vs. jugador 8,  $N$ , descansa ese día del torneo.

N \ Día	1	2	3	4	5	6	7
	1	2	3	4	5	6	7
1	2	3	4	5	6	7	8=D
2	1	4	3	6	5	8	7
3	4	1	2	7	8	5	6
4	3	2	1	8	7	6	5
5	6	7	8	1	2	3	4
6	5	8	7	2	1	4	3
7	8	5	6	3	4	1	2







## RESULTADOS DEL ALGORITMO.

```

-----[ jar ]-----
--- exec-maven-plugin:3.0.0:exec (default-cli) @ TorneoTenis ---
      INICIO DEL PROGRAMA - TORNEO DE TENIS.
+-----+
| Nota: Introduzca un valor entero +, <1. Como primer parámetro. |
+-----+

Error Desconocido. Se ha producido un error inesperado, por favor vuelva a ejecutar el programa.
Si necesita ver la ayuda del programa, ejecute:
$ 'java torneo.jar n -h'

-h: Ayuda
n: Número de jugadores del torneo de tenis.

Vemos la traza del programa ... CARGANDO...

.....
· El número de jugadores para el torneo introducido es: 10 ·
.....
Argumento[0]=>'Jugadores' es de tipo:Integer
1 2 3 4 5 6 7 8 9 10
2 1 4 5 4 7 6 9 10 9
3 4 1 6 7 8 9 6 11 12
4 5 6 1 2 9 10 11 6 7
5 4 7 2 1 10 9 12 7 6
6 7 8 9 10 1 2 3 4 5
7 6 9 10 9 2 1 4 5 4
8 9 6 11 12 3 4 1 6 7
9 10 11 6 7 4 5 6 1 2
10 9 12 7 6 5 4 7 2 1
Traza:

-----
      INICIALIZACION DE LAS TRAZAS DEL PROGRAMA.
-----
1. Entro en el método de Torneo
2. Creamos el nuevo objeto partido de la clase torneo
3. Comprobamos el valor de entrada como parámetro: par o impar.
4. Aplicamos el algoritmo de DyV a la matriz de NxN siendo N el argumento entero positivo que forma los tenista del torneo.
   Para aplicarlo en este problema, tenemos que usar la division de la tabla más grande de NxN, en tablas más pequeñas de (N/2)x(N/2), y después
   volvemos a unir la solución de las tablas divididas en una sola, así obtenemos el calendario.
5. Mostramos el resultado del DyV por pantalla.
FIN DE LA TRAZABILIDAD DEL PROGRAMA.
true
-----

```

### ILUSTRACIÓN 3 TORNEO N=10 + TRAZABILIDAD.

```

| --- exec-maven-plugin:3.0.0:exec (default-cli) @ TorneoTenis ---
|      INICIO DEL PROGRAMA - TORNEO DE TENIS.
|
|-----+
| Nota: Introduzca un valor entero +, <1. Como primer parámetro. |
|-----+
|
|.....
|· El número de jugadores para el torneo introducido es: 8 ·
|.....
|
|1 2 3 4 5 6 7 8
|2 1 4 3 6 5 8 7
|3 4 1 2 7 8 5 6
|4 3 2 1 8 7 6 5
|5 6 7 8 1 2 3 4
|6 5 8 7 2 1 4 3
|7 8 5 6 3 4 1 2
|8 7 6 5 4 3 2 1
|-----

```

### ILUSTRACIÓN 4 TORNEO N=8

```

--- exec-maven-plugin:3.0.0:exec (default-cli) @ TorneoTenis ---
      INICIO DEL PROGRAMA - TORNEO DE TENIS.
+-----+
+| Nota: Introduzca un valor entero +, <1. Como primer parámetro. |
+-----+

Vemos la ayuda del programa ... CARGANDO...

      SINTAXIS.

Para la ejecución del organizador de partidas del torneo de tenis, ejecute:

      $ java -jar torneo.jar n [-t][-h]

1. n: Número entero positivo, mayor que 1, indica el número de jugadores del torneo de tenis.
2. -t: Traza las llamadas recursivas del programa.
3. -h: Muestra esta ayuda.

@Versión: 1.0
@CentroAsociado: UNED-Las Tablas.
@Author: Paulino Esteban Bermúdez R.

```

ILUSTRACIÓN 5 ARGUMENTO -H.

El programa, se realizó en Apache NetBeans IDE 12.2 pero se ha desarrollado para que se pueda ejecutar con BlueJ.

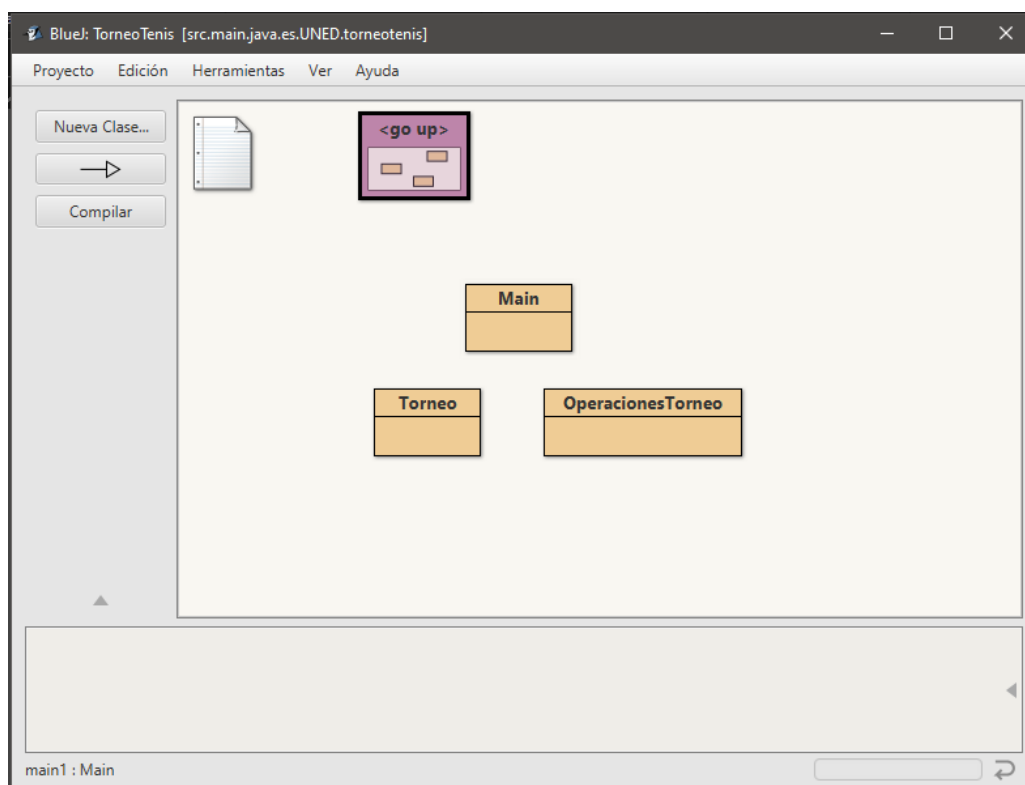


ILUSTRACIÓN 6 BLUEJ: TORNEO TENIS

Para la ejecución desde terminal, he creado un archivo llamado torneo.jar, siguiendo las indicaciones del enunciado.

```

pauli@MI-BLANCA MINGW64 ~/OneDrive/Escritorio/PED_TENIS/BlueJ
$ ls
torneo.jar

```

ILUSTRACIÓN 7 ARCHIVO .JAR

```
pauli@MI-BLANCA MINGW64 ~/OneDrive/Escritorio/PED_TENIS/BlueJ
$ java -jar torneo.jar -h
      INICIO DEL PROGRAMA - TORNEO DE TENIS.
+-----+
| Nota: Introduzca un valor entero +, <1. Como primer parámetro. |
+-----+

Vemos la ayuda del programa ... CARGANDO...

      SINTAXIS.

Para la ejecución del organizador de partidas del torneo de tenis, ejecute:

      $ java -jar torneo.jar n [-t][-h]

1. n: Número entero positivo, mayor que 1, indica el número de jugadores del torneo de tenis.
2. -t: Traza las llamadas recursivas del programa.
3. -h: Muestra esta ayuda.

@Versión: 1.0
@CentroAsociado: UNED-Las Tablas.
@Author: Paulino Esteban Bermúdez R.
```

ILUSTRACIÓN 8 AYUDA DE TORNEO.JAR

```
pauli@MI-BLANCA MINGW64 ~/OneDrive/Escritorio/PED_TENIS/BlueJ
$ java -jar torneo.jar 6 -t
      INICIO DEL PROGRAMA - TORNEO DE TENIS.
+-----+
| Nota: Introduzca un valor entero +, <1. Como primer parámetro. |
+-----+

.....
. El número de jugadores para el torneo introducido es: 6 .
.....
Los datos introducidos son incorrectos, vuelva a intentarlo de nuevo con otros datos
0 1 2 3 4 5
1 0 1 4 3 4
0 1 0 3 4 3
3 4 5 0 1 2
4 3 4 1 0 1
3 4 3 0 1 0

Vemos la traza del programa ... CARGANDO...

.....
. El número de jugadores para el torneo introducido es: 6 .
.....
Argumento[0]=>'Jugadores' es de tipo:Integer
Los datos introducidos son incorrectos, vuelva a intentarlo de nuevo con otros datos
0 1 2 3 4 5
1 0 1 4 3 4
0 1 0 3 4 3
3 4 5 0 1 2
4 3 4 1 0 1
3 4 3 0 1 0
Traza:

-----
      INICIALIZACION DE LAS TRAZAS DEL PROGRAMA.
-----
1. Entro en el método de Torneo
2. Creamos el nuevo objeto partido de la clase torneo
3. Comprobamos el valor de entrada como parámetro: par o impar.
4. Aplicamos el algoritmo de DyV a la matriz de NxN siendo N el argumento entero positivo que forma los tenista del torneo.
   Para aplicarlo en este problema, tenemos que usar la division de la tabla más grande de NxN, en tablas más pequeñas de (N/2)x(N/2), y después
   volvemos a unir la solución de las tablas divididas en una sola, así obtenemos el calendario.
5. Mostramos el resultado del DyV por pantalla.
FIN DE LA TRAZABILIDAD DEL PROGRAMA.
Los datos introducidos son incorrectos, vuelva a intentarlo de nuevo con otros datos
true
```

ILUSTRACIÓN 9 TRAZABILIDAD DE TORNEO.JAR

```
pauli@MI-BLANCA MINGW64 ~/OneDrive/Escritorio/PED_TENIS/BlueJ
$ java -jar torneo.jar 4
      INICIO DEL PROGRAMA - TORNEO DE TENIS.
+-----+
| Nota: Introduzca un valor entero +, <1. Como primer parámetro. |
+-----+

.....
. El número de jugadores para el torneo introducido es: 4 .
.....
1 2 3 4
2 1 4 3
3 4 1 2
4 3 2 1

pauli@MI-BLANCA MINGW64 ~/OneDrive/Escritorio/PED_TENIS/BlueJ
$ java -jar torneo.jar 3
      INICIO DEL PROGRAMA - TORNEO DE TENIS.
+-----+
| Nota: Introduzca un valor entero +, <1. Como primer parámetro. |
+-----+

.....
. El número de jugadores para el torneo introducido es: 3 .
.....
Los datos introducidos son incorrectos, vuelva a intentarlo de nuevo con otros datos
0 1 2
1 0 1
0 1 0
```

ILUSTRACIÓN 11 TORNEO.JAR

CON VALOR PAR:4

E IMPAR:3

## BIBLIOGRAFÍA.

- Libro de texto. [PROGRAMACIÓN Y ESTRUCTURAS DE DATOS AVANZADAS](#).
- Wikipedia. [Algoritmo divide y vencerás](#).
- Guerequeta R. & Vallevillo A. (1998), "[Técnicas de Diseño de Algoritmos](#)". Servicio de Publicaciones de la Universidad de Málaga, 1998