

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](#_Toc61465128)

[DISEÑO DEL ALGORITMO. 3](#_Toc61465129)

[CÓDIGO: 6](#_Toc61465130)

[RESULTADOS DEL ALGORITMO. 8](#_Toc61465131)

[BIBLIOGRAFÍA. 11](#_Toc61465132)

# 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.](https://drive.google.com/file/d/1I48KyTBLd3QLsyFGVizjReG2A60_C581/view?usp=sharing)

# 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 ≤ i ≤ n y 1 ≤ j ≤ n-1.

2 Jugadores -> 4 Jugadores

Indica jugador 1

Partidos del jugador 2

Agregado artificial

Esq.Sup.izda. Esq.Sup.dcha.

|  |  |
| --- | --- |
| 1 | 2 |
| 2 | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 2 | 1 | 4 | 3 |
| 3 | 4 | 1 | 2 |
| 4 | 3 | 2 | 1 |

K = 1

K = 2

Suma = 2

Proceso de crear k=2 a partir de k=1.

Esq.Inf.izda. Esq.Inf.dcha.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 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 |

Esq.Sup.izda. Esq.Sup.dcha.

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 2 | 3 | 4 |
| 2 | 1 | 4 | 3 |
| 3 | 4 | 1 | 2 |
| 4 | 3 | 2 | 1 |

K = 3

K = 2

Suma = 4

Proceso de crear k=3 a partir de k=2.

Esq.Inf.izda. Esq.Inf.dcha.

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.

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.

val(1,5)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| N |  |  |  |  |  |  |  |
| 1  (1,2) | 2  (1,1) | 3 | 4  (1,5) | 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  val(1,2)  val(1,1) | 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.

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 | 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 |

Días

# CÓDIGO:

Ilustración Main class



Ilustración Torneo class

# RESULTADOS DEL ALGORITMO.

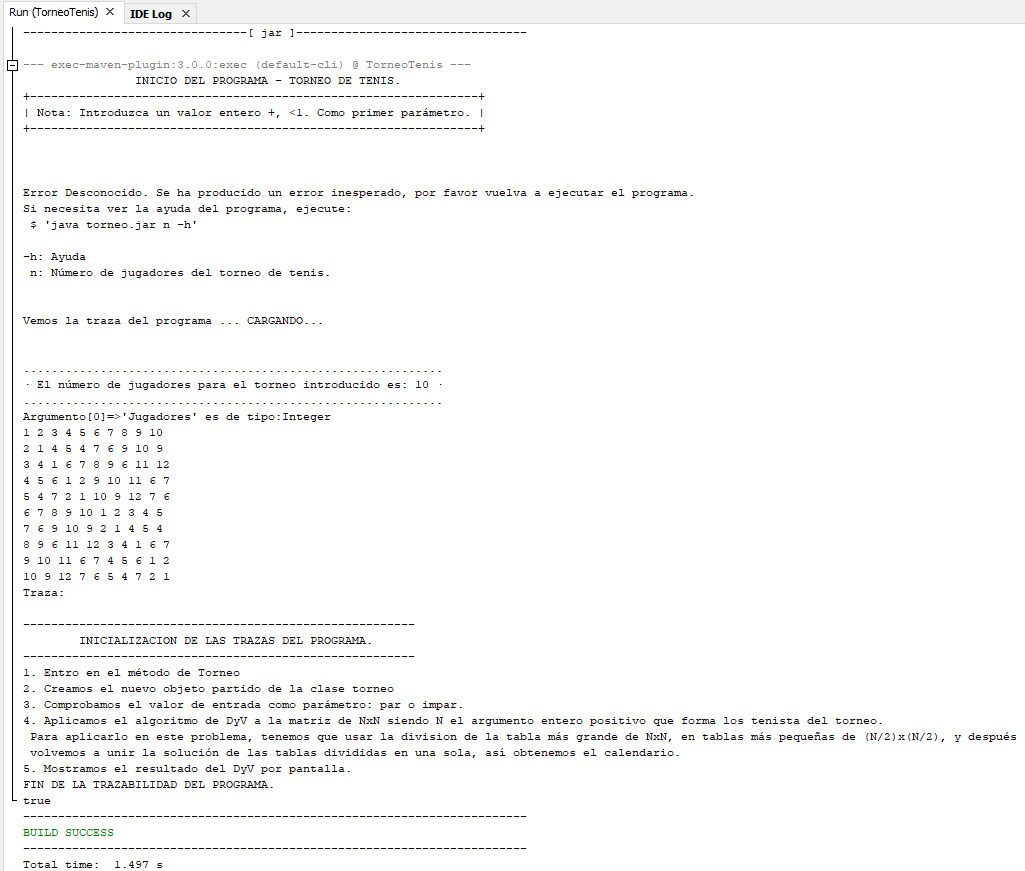


Ilustración Torneo n=10 + trazabilidad.

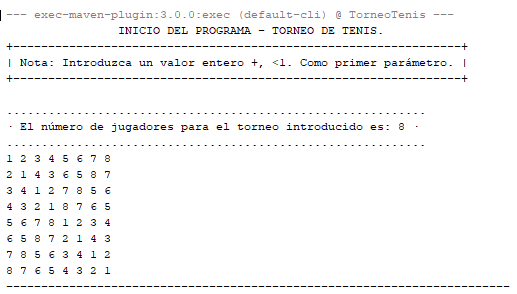


Ilustración Torneo n=8

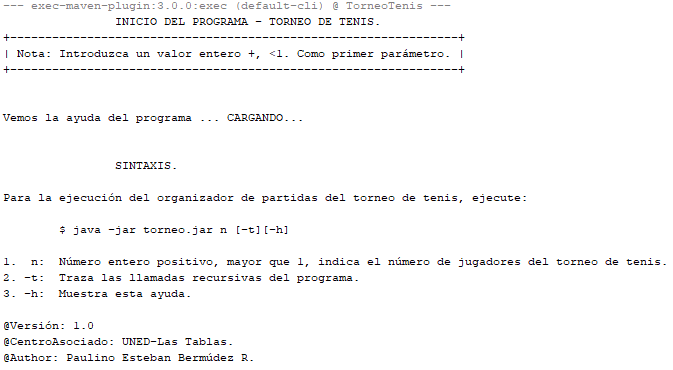


Ilustración argumento -H.

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

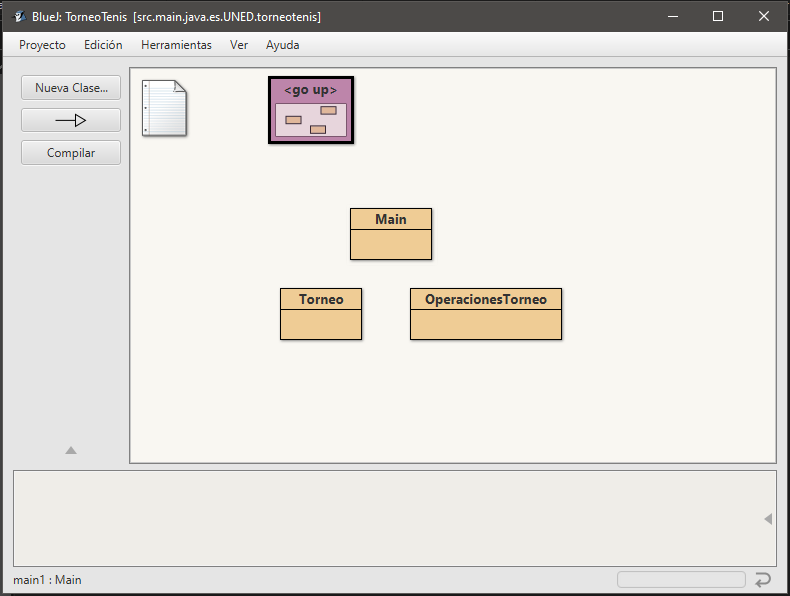


Ilustración BlueJ: torneo tenis

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

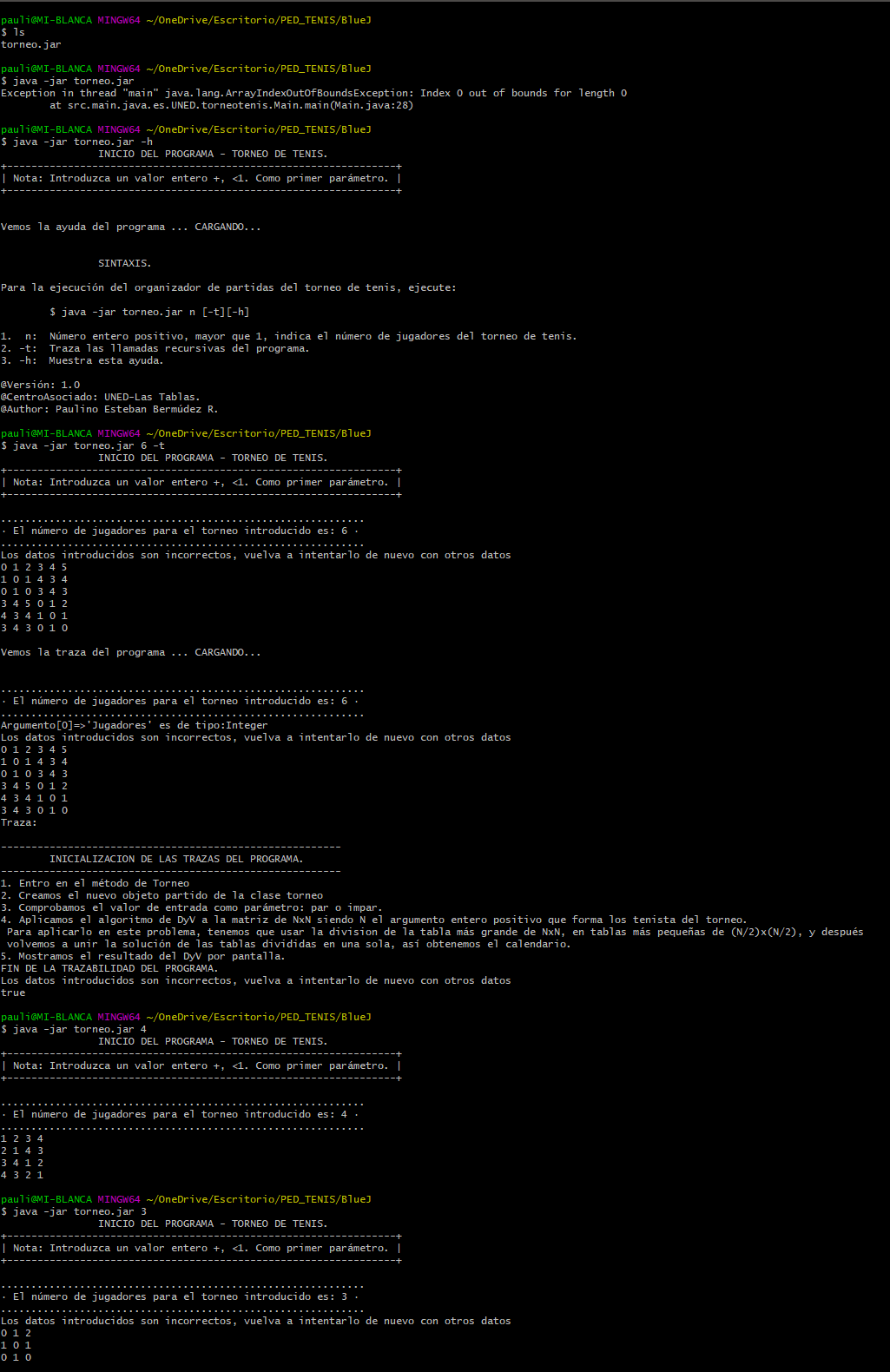


Ilustración Archivo .jar

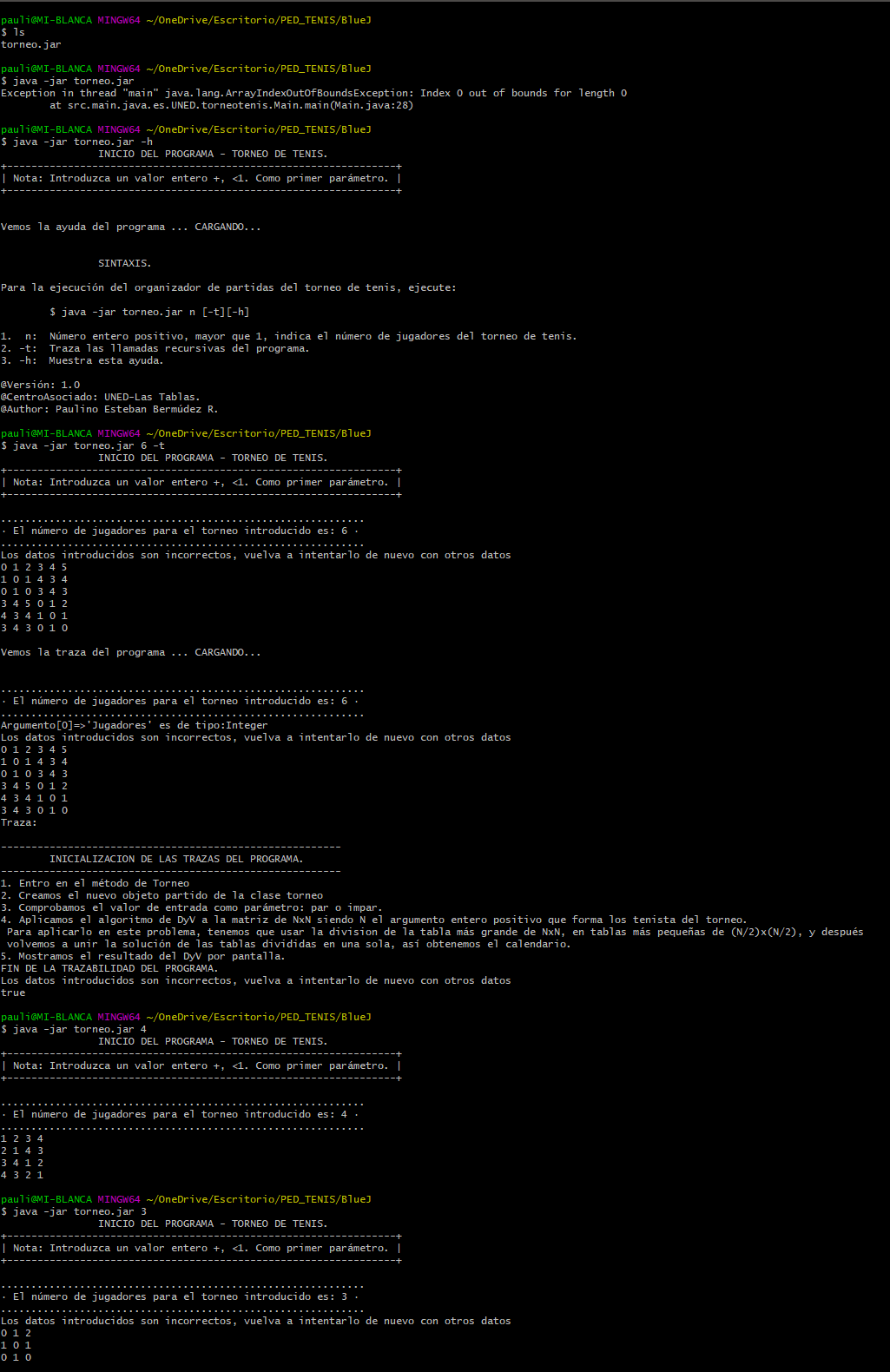


Ilustración Ayuda de torneo.jar

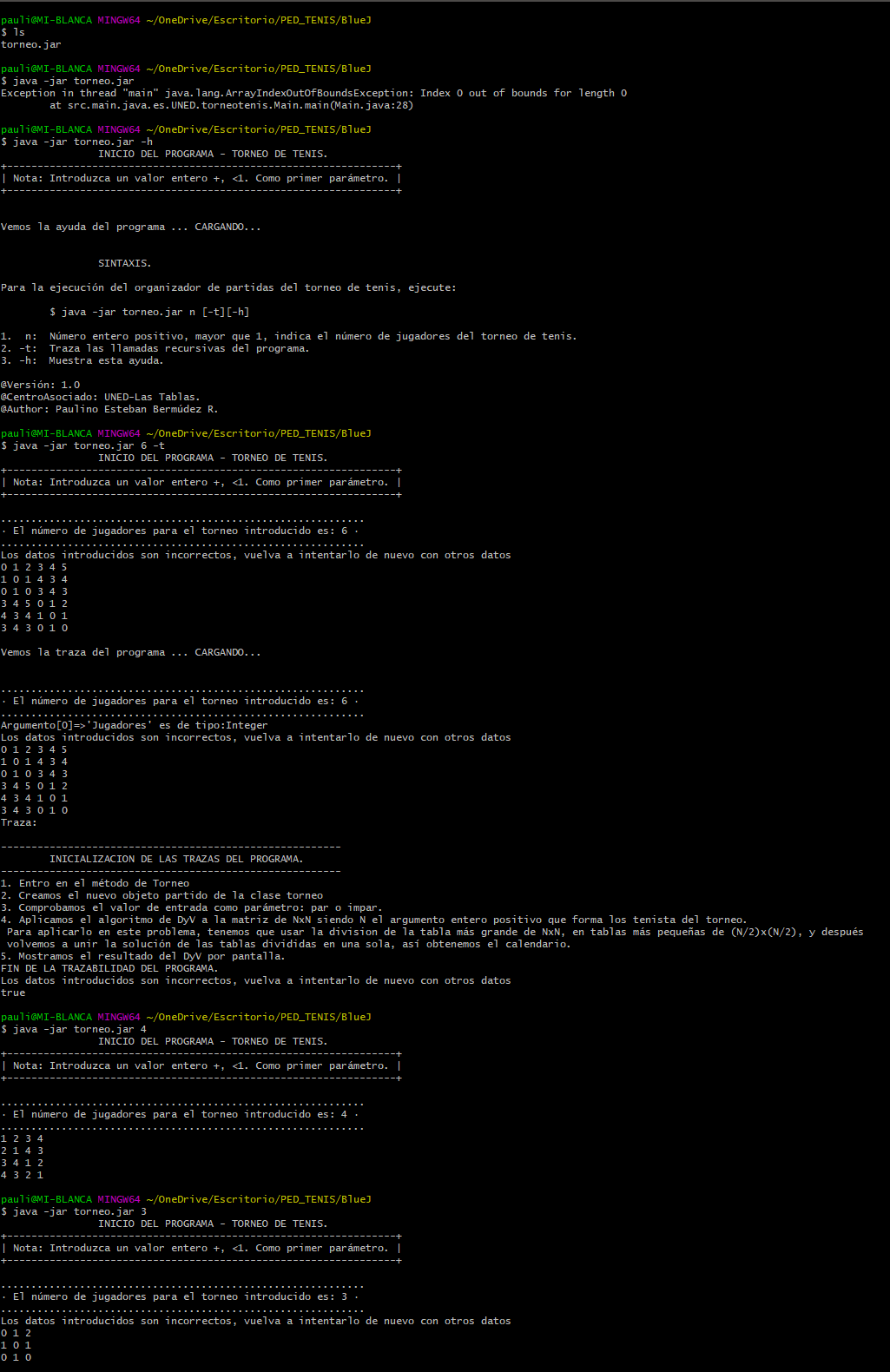


Ilustración Trazabilidad de torneo.jar

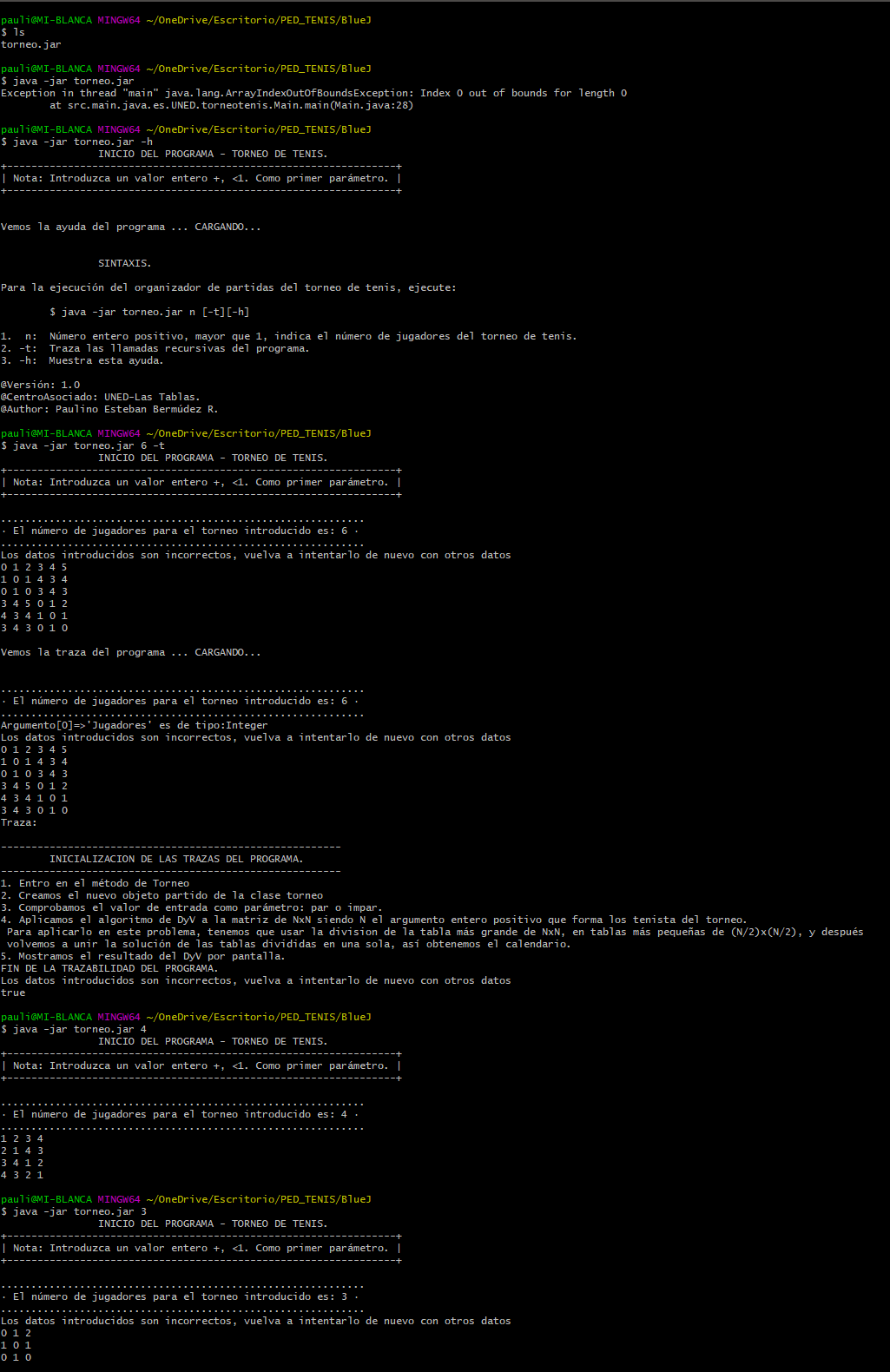


Ilustración Ejecucion de jar con nº par: 4 e impar: 3

Ilustración torneo.jar

con valor par:4

e impar:3

# BIBLIOGRAFÍA.

* Libro de texto. [PROGRAMACIÓN Y ESTRUCTURAS DE DATOS AVANZADAS](https://www.librosuned.com/LU13166/Programaci%C3%B3n-y-estructuras-de-datos-avanzadas.aspx).
* Wikipedia. [Algoritmo divide y vencerás](https://en.wikipedia.org/wiki/Divide-and-conquer_algorithm).
* Guerequeta R. & Vallevillo A. (1998), “[Tecnicas de Diseño de Algoritmos](http://www.lcc.uma.es/~av/Libro/)”. Servicio de Publicaciones de la Universidad de Málaga, 1998