

# Samuel Espejo Gil

## Daniel Ayuso Del Campo

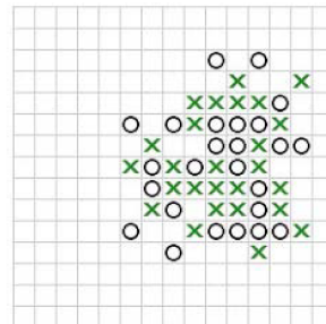
## Alejandro Cañas Borreguero

## Alberto Cortés Herranz

### 1ºB

#### 6. 5 en raya

El juego de las 5 en raya se juega sobre un tablero de  $N \times N$  y participan dos jugadores con  $(N^2/2)+1$  fichas de dos tipos distintos, uno para cada uno. El objetivo de cada jugador es conseguir antes que el otro una hilera de cinco fichas situadas sobre una misma recta vertical, horizontal o en diagonal. Se echa a suertes quien de los dos jugadores inicia el primer movimiento. Seguidamente el turno se va cambiando entre uno y otro de forma alternativa. Cada movimiento consiste en incorporar una nueva ficha al tablero. El juego finaliza cuando un jugador consigue una hilera de 5 fichas o cuando el tablero se ha completado y ya no se pueden poner más fichas.



Se pide realizar un programa que permita a dos jugadores jugar al 5 en raya de forma interactiva. Para ello, se pedirá por teclado el valor de  $N$ , así como los nombres de los dos jugadores. A continuación, el programa decidirá, de forma aleatoria, quién de los dos jugadores comenzará el juego, así como el tipo de ficha con el que jugará. El programa, entonces, irá indicando, alternativamente, a cuál de los dos jugadores le toca realizar su movimiento, para lo que, en cada momento, mostrará el estado del tablero. Es necesario controlar todos los errores posibles como que se quiera insertar datos sobre casillas ya ocupadas o casillas fuera de las dimensiones del tablero. Cuando el juego finalice se debe guardar en un fichero de texto el estado del tablero, para lo que en la primera línea se grabará el valor de  $N$  y, a continuación, cada línea representará una fila del tablero.

## *Análisis del problema*

Para realizar el análisis imaginemos la situación en la vida real. Yo quiero jugar contra ti al 5 en raya. Podemos diferenciar tres momentos: antes del juego, el juego y después del juego. Damos por sentado que conocemos las reglas. En el programa, esto último se traduce como la revisión de errores. Algunos ejemplos son la correcta colocación de las fichas en el tablero (ni fuera de él ni en casillas ya ocupadas) o un tablero “creíble” (sin dimensiones exageradamente grandes).

Lo primero que debemos saber es el tablero sobre el que vamos a jugar, sus dimensiones. Decidimos a suertes quién pone la primera ficha y qué ficha es. Es decir, quién usa las cruces y quién los círculos. El problema nos pide registrar los nombres de los jugadores. En la vida real, sabemos contra quién estamos jugando. Por eso, se omite pero hay que tenerlo en cuenta en el programa.

Una vez analizada la situación antes del juego, comienza la partida. Observamos el tablero y pensamos nuestra jugada. Se ha aclarado quién mueve primero por lo que el turno al principio ya viene indicado pero posteriormente el turno se alternará y será necesario indicarlo entre nosotros. Comunicación, al fin y al cabo. Se realiza el movimiento y, por tanto, se coloca la ficha en el tablero. Cada vez que se coloque una ficha, contemplaremos el mapa tras nuestra decisión, que únicamente tendrá sentido al pensar en la siguiente. Tendremos una estrategia con el fin de conseguir el 5 en raya. Por tanto, comprobaremos constantemente dónde puede ir el rival. Nosotros jugamos y evitamos que el rival nos gane mientras que llevamos a cabo nuestra estrategia. Por tanto, el programa debe comprobar con cada movimiento si hay un 5 en raya. Obviamente, no tiene que evitarlo sino confirmarlo para dar un veredicto.

Si el tablero está completo (no hay más fichas) o hay un 5 en raya, se procede al resultado de la partida. El enunciado nos exige registrarlo en un documento de texto. Tiene cierta lógica para poder implementar, por ejemplo, un historial donde puedas ver las partidas jugadas con determinadas especificaciones como el resultado, la hora, el día... O simplemente para algo tan simple como recordar quién fue el perdedor de aquella vez.

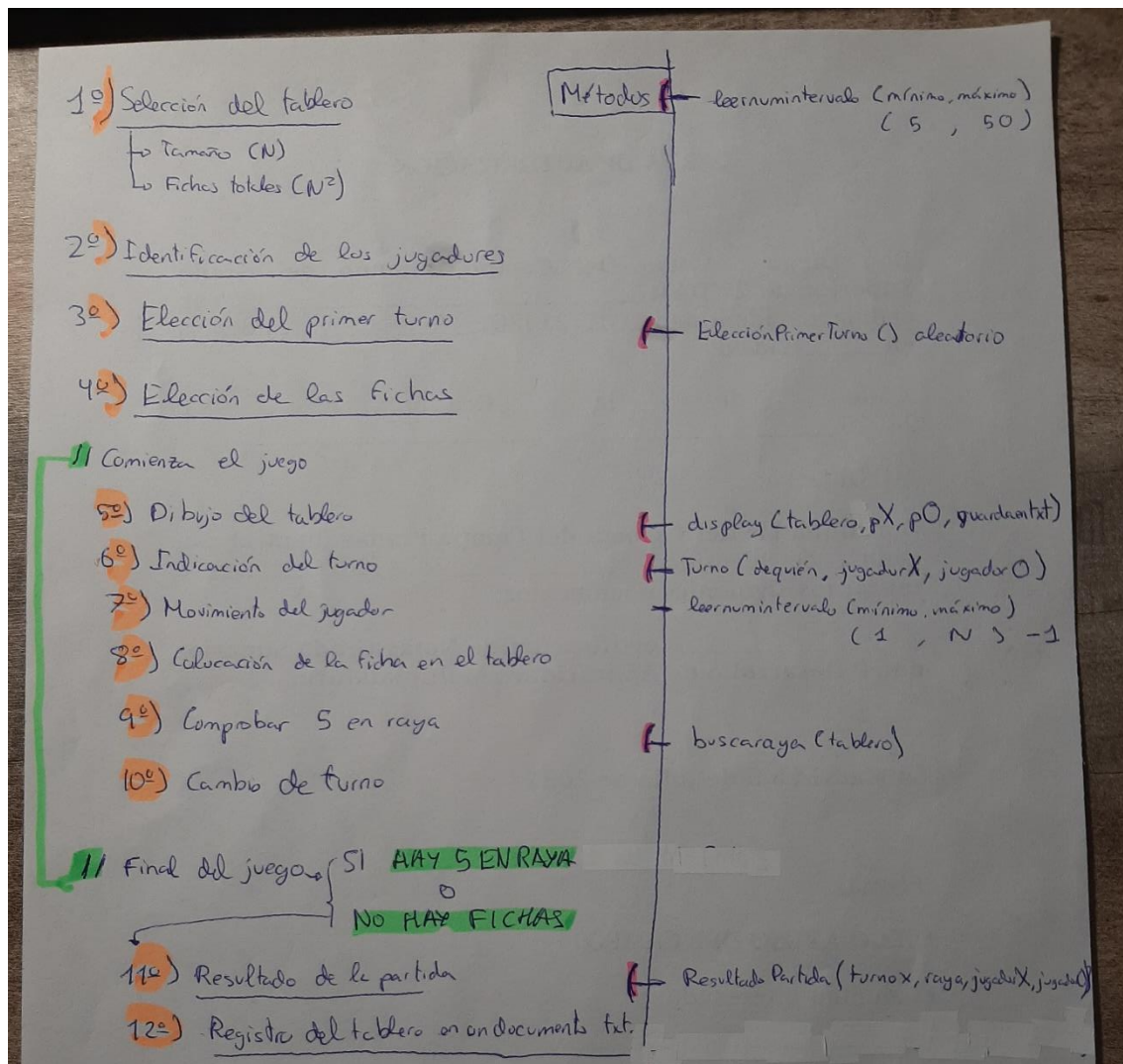
**6. 5 en raya**

El juego de las 5 en raya se juega sobre un tablero de  $N \times N$  y participan dos jugadores con  $(N^2/2)+1$  fichas de dos tipos distintos, uno para cada uno. El objetivo de cada jugador es conseguir antes que el otro una hilera de cinco fichas situadas sobre una misma recta vertical, horizontal o en diagonal. Se echa a suertes quien de los dos jugadores inicia el primer movimiento. Seguidamente el turno se va cambiando entre uno y otro de forma alternativa. Cada movimiento consiste en incorporar una nueva ficha al tablero. El juego finaliza cuando un jugador consigue una hilera de 5 fichas o cuando el tablero se ha completado y ya no se pueden poner más fichas.

Se pide realizar un programa que permita a dos jugadores jugar al 5 en raya de forma interactiva. Para ello, se pedirá por teclado el valor de  $N$ , así como los nombres de los dos jugadores. A continuación, el programa decidirá, de forma aleatoria, quien de los dos jugadores comenzará el juego, así como el tipo de ficha con el que jugará. El programa, entonces, irá indicando, alternativamente, a cuál de los dos jugadores le toca realizar su movimiento, para lo que, en cada momento, mostrará el estado del tablero. Es necesario controlar todos los errores posibles como que se quiera insertar datos sobre casillas ya ocupadas o casillas fuera de las dimensiones del tablero. Cuando el juego finalice se debe guardar en un fichero de texto el estado del tablero, para lo que en la primera línea se grabará el valor de  $N$  y, a continuación, cada línea representará una fila del tablero.

*A tener en cuenta*

La imagen a continuación es útil tanto para el análisis como para la descomposición modular.



# *Justificación de la representación de los datos elegida y descomposición modular*

## **Justificación de la representación de los datos elegida**

- Al pedir el número (n) el programa comprueba que se encuentre dentro del intervalo [a,b].
- Decidimos representar las fichas del cinco en raya como “x” y “o”, ya que es la manera más tradicional de hacerlo y las casillas sin contenido las representamos con corchetes con un espacio dentro [ ].
- La condición !r es para que cuando detecte el cinco en raya se detenga y no realice más iteraciones.
- El dato booleano turnox lo utilizamos para averiguar quién es el jugador que comienza la partida.
- TurnoX quiere decir que el turno será del jugador que utiliza la ficha “x”, ese jugador es el jugadorX y el caso contrario es el jugadorO, que utiliza las fichas “o”.
- A la hora de representar el tablero, representaremos la matriz del tablero y el número de las columnas y las filas, para esto usamos tres variables:
  - Dos variables enteras que irán desde -1 hasta el tamaño del tablero indicado.
  - Una variable de tipo string que empiece en nada y que guarde fila a fila que es lo que tiene que mostrar.
- Nos fijamos en los valores de las variables numéricas y si alguna de estas dos tiene el valor -1 en la variable display guardaremos el valor de dichas variables sumándoles 1. Si ninguna de las dos variables vale -1, guardaremos dependiendo de lo que haya en la matriz tablero, si en la matriz tablero no hay nada guardaremos un [ ], si hay una x guardaremos una “ x “ y si hay un o guardaremos un “ o “.
- Este proceso se realiza, primero, las columnas y luego las filas.
- Identificaremos al jugador que introduce primero su nombre como jugador1 y, posteriormente, jugador2. A la hora de asignar las fichas y el turno se convertirán en jugadorX o jugadorO según un número aleatorio.

- Una vez comienza el juego se dibuja el tablero mediante el display, se indica el turno con el Turno, se colocan las fichas y se leen su posición mediante el Leernuminintervalo, se comprueba el cinco en raya con el buscaraya y finalmente mediante el ResultadoPartida se da el ganador de la partida.

## Descomposición modular

### Primer método: leernuminintervalo

```
static int leernuminintervalo(int minimo, int máximo) {
    // Comprueba que el número introducido se encuentre en el intervalo [a,b]
    int n;
    do {
        n = teclado.nextInt();
        if (n < minimo || n > máximo)
            System.out.println(
                "Error el número debe ser mayor o igual a " + minimo + " y menor o igual que " + máximo + "."
            );
    } while (n < minimo || n > máximo);
    return n;
}
```

Mediante el método leernuminintervalo, el programa comprueba que el número introducido (n) esté dentro del intervalo [a,b]. Si el número introducido (n) no se encuentra dentro de dicho intervalo, el programa dará error y volverá a pedir el número hasta que esté dentro del intervalo.

### Segundo método: buscaraya

```
static boolean buscaraya(char[][] tablero) {
    // Método para buscar el 5 en raya
    int n = tablero.length;
    boolean r = false;

    for (int x = 0; x < n && !r; x++) // la condición de !raya es para que en cuanto se detecte que hay 5 en raya se
        // salga y no tenga que hacer más iteraciones
        for (int y = 0; y < n && !r; y++)
            switch (tablero[x][y]) { // Recorre todo el tablero y si hay algo, empieza a revisar.
                case 'x':
                case 'o':
                    for (int i = 0; i <= 1 && !r; i++)
                        for (int j = -1; j <= 1 && !r; j++) // En una matriz 3*3, la casilla del centro (x, y), la de
                            // arriba a la izquierda (x-1, y-1), la de arriba centro (x,
                            // y-1)...
                            if ((x + 4 * i < 0 || x + 4 * i >= n || y + 4 * j < 0 || y + 4 * j >= n) // Se encarga de
                                // que esté
                                // dentro de la
                                // matriz y no
                                // dé error
                                // indexOutOfBounds
                                && !(i == 0 && j == 0)) // Para no comprobar en la misma celda
                                if (tablero[x][y] == tablero[x + i][y + j]
                                    && tablero[x][y] == tablero[x + 2 * i][y + 2 * j]
                                    && tablero[x][y] == tablero[x + 3 * i][y + 3 * j]
                                    && tablero[x][y] == tablero[x + 4 * i][y + 4 * j])
                                    r = true;
            }
    return r;
}
```

Este método se encarga de buscar el cinco en raya. El programa recorre todo el tablero y si hay una casilla ocupada comienza a revisarlo. Para revisar el tablero lo que hace es



comprobar el contenido de las casillas de alrededor de la que tiene una “x” o un “o” y así sucesivamente.

### *Tercer método: ElecciónPrimerTurno*

```
static boolean ElecciónPrimerTurno() { // De forma aleatoria, se elige el primer turno.
    boolean turno;
    int primerturno = (int) (Math.random() * 2);
    if (primerturno == 0) // Se toma la parte entera del número aleatorio
        turno = true;
    else
        turno = false;
    return turno;
}
```

El método ElecciónPrimerTurno realiza la elección del primer turno de forma aleatoria mediante la función Math.random(), que genera un número aleatorio de 0 a 2 pero nunca será el 2. De la operación realizada con dicha función guardamos su parte entera y dependiendo de su valor (0 o 1) regresará turno con un valor u otro.

### *Cuarto método: Turno*

```
static void Turno(boolean turnox, String jugadorX, String jugadorO) { // Indicación del turno
    System.out.println();
    if (turnox)
        System.out.println("\nLe toca a " + jugadorX + ". Marca tu x.");
    else
        System.out.println("\nLe toca a " + jugadorO + ". Marca tu o.");
}
```

Este método únicamente indica cuál es el turno de cada jugador.

### *Quinto método: ResultadoPartida*

```
static void ResultadoPartida(boolean turno, boolean raya, String jugadorX, String jugadorO) { // RESULTADO DE LA
                                                                    // PARTIDA
    if (!turno && raya)
        System.out.println("\n5 en raya para " + jugadorX + "!");
    else if (raya)
        System.out.println("\n5 en raya para " + jugadorO + "!");
    else
        System.out.println("\nSe han acabado las fichas!");
}
```

El método ResultadoPartida indica cuál ha sido el resultado de la partida. Pude haber tres casos:

- Que gane el jugador que usaba las x.
- Que gane el jugador que usaba los o.
- Empate, ya que se han acabado las fichas. Se ha completado el tablero sin que haya un cinco en raya.

## Sexto método: display

```
static void display(char[][] tablero, String pX, String pO, boolean guardaentxt) throws IOException {
    FileWriter fichero = new FileWriter("C:\\Users\\Public\\Tablero.txt");
    // Creamos el documento y añadimos la ruta donde queremos guardarlo
    int n = tablero.length;

    if (guardaentxt) {
        fichero.write("\r\n");
        fichero.write("Tablero: " + n + "x" + n + ".\n");
        fichero.write("Jugador X: " + pX + " | Jugador O: " + pO);
        fichero.write("\r\n");
    }

    for (int i = -1; i < n; i++) {
        String display = "\r\n";
        for (int j = -1; j < n; j++) {
            if (i == -1 && j == -1) {
                display = display + "\n0 ";
            } else if (i == -1) {
                if (j < 9) // if (i == -1 && j<9)
                    display = display + (j + 1) + " ";
                else // if (i == -1 && j>=9) {
                    display = display + (j + 1) + " ";
            } else if (j == -1) {
                if (i < 9) // if (j == -1 && i<9)
                    display = display + (i + 1) + " ";
                else // if (j == -1 && i>=9) {
                    display = display + (i + 1);
            } else {
                switch (tablero[j][i]) {
                    case 'x':
                        display = display + " x ";
                        break;
                    case 'o':
                        display = display + " o ";
                        break;
                    default:
                        display = display + "[ ]";
                }
            }
        }
        System.out.print(display);
        if (guardaentxt) {
            fichero.write(display);
        }
    }
    if (guardaentxt)
        fichero.close(); // Cerramos el fichero
}
```

Dentro del método display, realizaremos la tarea de guardar el tablero en un .txt mediante el parámetro guardaentxt y previamente hemos creado el documento y hemos añadido la ruta donde queremos guardarlo.

Dentro del display la tarea principal es la de crear el tablero. Primero pone el número de columnas y filas para que el jugador pueda identificarlas y seleccionarlás con mayor facilidad, luego crea el tablero con la medida que nosotros le hemos indicado al principio del programa y por último el programa comienza a poner las “x” y los “o”, pidiéndolo al jugador que introduzca la columna (j) y la fila (i). Es importante tener en cuenta los dígitos de los números del display. Por ejemplo, en un tablero de 15x15, el número 6 que mostraría sería “distinto” al número 14 por la cantidad de dígitos. Esto podría descolocar el tablero.

# *Pseudocódigo del algoritmo*

Añadir paquete Scanner

Añadir paquete FileWriter

Añadir paquete IOException

Función leerintervalo(entero mínimo, entero máximo): Entero

n: Entero

Hacer:

n <- Leer desde teclado

Si n es más pequeño que el extremo inferior del intervalo o mayor que el extremo superior

Mostrar (Error el número deber ser mayor que el extremo inferior o menor que el extremo superior)

Fin\_Si

Repetir hacer mientras n sea más pequeño que el extremo inferior del intervalo o mayor que el extremo superior

Fin\_Función leerintervalo

---

Función display(char[][] tablero, string pX, string pO, boolean guardaentxt)

Definimos la ruta

Guardar entero en n <- longitud del tablero

Si (guardaentxt)

Escribir en el fichero (Tablero de: n por n)

Escribir en el fichero(Jugador X es pX y jugador O es pO)

Fin\_Si

Para i desde -1 hasta i menor que n incrementando 1

String display es igual a siguiente línea

Para j desde -1 hasta j menor que n incrementando 1

Si i es igual a 1 y j es igual a -1

display es igual a display más salto de línea y después pone un 0

O Si i es igual a -1

Si j es menor que 9



display es igual a display, añadiendo la posición j más 1 y más 2 espacios

Sino

display es igual a display, añadiendo la posición j más 1 y más 1 espacios

Fin\_Si

O Si j es igual a -1

Si i es menor que 9

display es igual a display, añadiendo la posición i más 1 y más 1 espacios

Sino

display es igual a display, añadiendo la posición i más 1

Fin\_Si

Sino

Posición j i del tablero:

Hay X:

display es igual a display, añadiendo una x

Hay O:

display es igual a display, añadiendo una o

Por defecto:

display es igual a display, añadiendo [ ]

Fin\_Si

Fin\_Para

Mostrar (display)

Si (guardaentxt)

Escribe en el fichero display

Fin\_Si

Fin\_Para

Si (guardaentxt)

Cerrar fichero

Fin\_Si

Fin\_Funcióndisplay

---

Función buscaraya(char[][] tablero): boolean

Guardamos un entero en n <- Longitud del tablero

Guardamos un boolean en r <- false

Para entero x igual a 0 hasta x menor que n y que no haya raya incrementando 1

Para entero e igual a 0 hasta y menor que n y que no haya raya incrementando 1

Si en la posición x y del tablero hay un X u O

Para entero i igual a 0 hasta i menor o igual que 1 y que no haya raya incrementando 1

Para entero j igual a -1 hasta j menor o igual a 1 y que no haya raya incrementando 1

Si la posición x+i y+j se encuentra dentro del tablero y no ocurre que (i es igual a 0 y j es igual a 0)

Si las posiciones de X o O coinciden 5 veces

hay 5 en raya (r <- true)

Fin\_Si

Fin\_Si

Fin\_Para

Fin\_Para

Fin\_proceso

Fin\_Para

Fin\_Para

Fin\_Funciónbuscaraya

---

FunciónElecciónPrimerTurno(): Boolean

turnox: Boolean

Guardar entero en primerturno <- la parte entera de (Número elegido aleatoriamente hasta 2 sin incluir)

Si primerturno es igual a 0

turnox <- true

Sino

turnox <-false

Fin\_Si

Devuelve turnox

Fin\_ FunciónElecciónPrimerTurno

---

FunciónTurno(boolean turnox\_, string jugadorX, string jugador O)

Si turnox

Mostrar (Le toca a jugadorX)

Sino

Mostrar (Le toca a jugadorO)

Fin\_Si

Fin\_ FunciónTurno

---

FunciónResultadoPartida (boolean turnox, boolean raya, string jugador, string jugadorO)

Si es el contrario de turnox y hay 5 en raya

Mostrar (5 en raya del jugadorX)

O Si hay 5 en raya y es turnox

Mostrar (5 en raya del jugadorO)

Sino

Mostrar (Se han acabado las fichas)

Fin\_Si

Fin\_ FunciónResultadoPartida

---

Algoritmo 5 en raya

Datos NecesariosRaya: Boolean

Mostrar (Introducir tamaño del tablero)

Guardar en entero (N) &lt;- leernumintervalo(5, 50)

Guardar en entero (nfichas) &lt;- N por N

Crear matriz tablero de N filas por N columnas

Mostrar (Introduce el nombre del primer jugador: )

Guardar en string (p1) &lt;- Leer desde teclado

Mostrar (Introduce el nombre del segundo jugador: )

Guardar en string (p2) &lt;- Leer desde teclado

Guardar en boolean (turnox) <- ElecciónPrimerTurno()

Guardar en entero (XoO) <- parte entera de (Número elegido aleatoriamente hasta 2 sin incluir)

Crear string pX y pO

Si XoO es igual a 0

px es igual a p1

pO es igual a p2

Sino

Px es igual a p2

pO es igual a p1

### Programa

Hacer:

Mostrar tablero <- Display(tablero)

Turno()

x, y: Enteros

Hacer:

Mostrar (Indica columna)

x <- leernumintervalo (1,N) menos 1

Mostrar (Indica la fila)

y <- leernumintervalo (1,N) menos 1

Si posición x y del tablero existe X o O

Mostrar (Error, la celda esta ocupada)

Fin\_Si

Repetir hacer mientras posición x y del tablero sea igual a X o O.

Si turno es igual a X

Guardar X en la posición x y del tablero

Sino

Guardar O en la posición x y del tablero

Fin\_Si

raya <- buscaraya()

turnox es igual al contrario de turnox

nfichas es igual a nfichas menos 1

Repetir hacer mientras no haya 5 en raya y el número de fichas sea mayor que 0

display(tablero,pX,pO, true)

ResultadoPartida()

Fin\_algoritmo\_5\_en\_raya

## *Lecciones aprendidas*

No tiene sentido usar un tablero gigante ya que se termina viendo mal y la partida podría durar mucho. Tampoco tiene sentido usar uno de menos de 5 filas y columnas ya que sería imposible realizar 5 en raya, incluso este último tiene poco valor por las pocas posibilidades de ganar.

No se necesita que el array bidimensional de char tablero esté inicializado.

A la hora de mostrar el tablero, poner números que indiquen la fila y la columna ayuda al usuario a colocar sus fichas, se debe tener en cuenta si el número tiene dos dígitos o uno para que el tablero no se mueva.

No es necesario enviar el parámetro N (el tamaño del tablero) a los diferentes métodos ya que se puede conseguir mediante la sentencia: `N=tablero.length;`

Para buscar raya hay dos formas principalmente, una en la que sólo comprobamos la celda en la que se ha producido el cambio y sus alrededores y la otra que mira todo el tablero. Al hacerlo de la primera forma se puede dar el caso de que la celda a comprobar caiga entre fichas y se complica el buscar 5 en raya. Es más óptimo recorrer todo el tablero, se necesitan más pasos, pero es más fácil la búsqueda de la raya ya que no tenemos en cuenta si la ficha está entre medias de dos.

Al buscar raya, si compruebas que la posible ficha más externa está dentro del tablero, no es necesario comprobar que todas las fichas que estén más cerca de la celda a comprobar



estén dentro ya que si la más externa está dentro, las internas también lo estarán. Tampoco tiene sentido comprobar las internas si la externa está fuera ya que sabemos que en esa dirección no se obtendrá un 5 en raya. Para optimizar más el algoritmo de buscar raya no es necesario mirar las fichas por las que ya se ha pasado anteriormente.

No se debe tener en cuenta la ficha  $x+0$  e  $y+0$  ya que es la misma celda en donde se realiza la comprobación y cualquier número multiplicado por 0 es 0 por lo que al intentar comprobar fichas de diferentes “capas” realmente sólo se comprueba la misma ficha y por ende da raya siempre.

Una vez se ha encontrado una raya no es necesario seguir buscando otra posibilidad.

No es necesario comprobar si en una raya son X u O, con que sean iguales basta, el main ya tiene en cuenta de quién es el turno, y por ende, de quién la raya ya que si en el anterior turno no se ha hecho una raya, la que se haya podido producir tiene que ser necesariamente del jugador que ha colocado ficha.

## Dos ejemplos de entrada

```

Escribe el tamaño del tablero:6
Nombre del primer jugador:
Dani
Nombre del segundo jugador:
Samu

0 1 2 3 4 5 6
1 [ ][ ][ ][ ][ ][ ]
2 [ ][ ][ ][ ][ ][ ]
3 [ ][ ][ ][ ][ ][ ]
4 [ ][ ][ ][ ][ ][ ]
5 [ ][ ][ ][ ][ ][ ]
6 [ ][ ][ ][ ][ ][ ]

Le toca a Dani. Marca tu x.

Indica la columna:3
Indica la fila:2
Error, la celda está ocupada.

Indica la columna:0
Error el número debe ser mayor o igual a 1 y menor o igual que 6.
2
Indica la fila:0
Error el número debe ser mayor o igual a 1 y menor o igual que 6.
. . . . .
Le toca a Dani. Marca tu x.

Indica la columna:2
Indica la fila:3

0 1 2 3 4 5 6
1 x [ ] o [ ] o [ ]
2 [ ] x x o [ ][ ]
3 [ ] x x x [ ][ ]
4 [ ][ ][ ] x [ ][ ]
5 x o o o o [ ]
6 [ ][ ][ ][ ][ ][ ]

Le toca a Samu. Marca tu o.

Indica la columna:4
Indica la fila:2

0 1 2 3 4 5 6
1 [ ][ ][ ][ ][ ][ ]
2 [ ][ ] x o [ ][ ]
3 [ ][ ][ ][ ][ ][ ]
4 [ ][ ][ ][ ][ ][ ]
5 [ ][ ][ ][ ][ ][ ]
6 [ ][ ][ ][ ][ ][ ]

Le toca a Dani. Marca tu x.
. . . . .

0 1 2 3 4 5 6
1 x [ ] o [ ] o [ ]
2 [ ] x x o [ ][ ]
3 [ ] x x x [ ][ ]
4 [ ][ ][ ] x [ ][ ]
5 x o o o o o
6 [ ][ ][ ][ ][ ][ ]

¡5 en raya para Samu!

Fin del programa

```

### Ejemplo 1

## 5 en Raya

```
Escribe el tamaño del tablero:-2
Error el número debe ser mayor o igual a 5 y menor o igual que 50.
51
Error el número debe ser mayor o igual a 5 y menor o igual que 50.
5
Nombre del primer jugador:
Alberto
Nombre del segundo jugador:
Álex

0 1 2 3 4 5
1 [ ][ ][ ][ ][ ]
2 [ ][ ][ ][ ][ ]
3 [ ][ ][ ][ ][ ]
4 [ ][ ][ ][ ][ ]
5 [ ][ ][ ][ ][ ]

Le toca a Alberto. Marca tu o.

Indica la columna:4
Indica la fila:2

0 1 2 3 4 5
1 [ ][ ][ ][ ][ ]
2 [ ][ ][ ]o[ ]
3 [ ][ ][ ][ ][ ]
4 [ ][ ][ ][ ][ ]
5 [ ][ ][ ][ ][ ]

Le toca a Álex. Marca tu x.

Indica la columna:3
Indica la fila:2

0 1 2 3 4 5
1 [ ][ ][ ][ ][ ]
2 [ ][ ]x o[ ]
3 [ ][ ][ ][ ][ ]
4 [ ][ ][ ][ ][ ]
5 [ ][ ][ ][ ][ ]

Le toca a Alberto. Marca tu o.

Indica la columna:7
Error el número debe ser mayor o igual a 1 y menor o igual que 5.
```

Samuel Espejo, Daniel Ayuso  
Alejandro Cañas, Alberto Cortés

```

•      •      •      •

Le toca a Álex. Marca tu x.

Indica la columna:2
Indica la fila:4

0 1 2 3 4 5
1 o o o x o
2 x x x o x
3 x o o x o
4 o x [ ] o x
5 x x o x o

Le toca a Alberto. Marca tu o.

Indica la columna:3
Indica la fila:4

0 1 2 3 4 5
1 o o o x o
2 x x x o x
3 x o o x o
4 o x o o x
5 x x o x o

¡Se han acabado las fichas!

Fin del programa
```

• • • • • •

## Ejemplo 2