

Circuito de torneos de tenis

Primavera 2022

Contenidos:

1. Enunciado de la práctica	2
1.1. Características del circuito y categorías de torneos	2
1.2. Cálculo del ranking	3
1.3. Fases de un torneo	3
2. Funcionalidades	8
2.1. Decisiones sobre los datos	8
2.2. Programa principal: estructura y comandos	8

1. Enunciado de la práctica

El tenis es un deporte que atrae a millones de aficionados y espectadores de todo el mundo. Los torneos de tenis a nivel internacional forman un circuito unificado desde 1970. En 1972 los principales jugadores profesionales masculinos crearon la ATP (Asociación de Tenistas Profesionales) y en 1973 la jugadora Billie Jean King fundó la WTA (*Women's Tennis Association*). Los rankings de jugadores y jugadoras se calculan de la misma manera automatizada, a partir de los resultados de los torneos, desde 1973 y 1974, respectivamente. En 1990 el circuito masculino pasó a denominarse *ATP tour* y en 1995 el circuito femenino pasó a denominarse *WTA tour*, que son sus nombres actuales.

En esta práctica simularemos la gestión deportiva y el cálculo del ranking de un circuito de torneos de tenis, muy similar al *ATP tour* y al *WTA tour*, aunque con algunas simplificaciones. Solamente consideraremos las competiciones y partidos de individuales (no los de dobles ni las competiciones por equipos), por lo que el único ranking a mantener será el de jugadores o jugadoras de individuales de un circuito.

1.1. Características del circuito y categorías de torneos

Un circuito está formado por un número T de torneos, que se celebran anualmente, y que están agrupados en C categorías. Los torneos se identifican por su nombre. Las categorías se identifican por un entero entre 1 y C , aunque también tienen un nombre asociado. Por ejemplo, un torneo denominado "Open_de_Australia" podría ser de categoría 1, cuyo nombre podría ser "Grand_Slam". Será posible dar de alta nuevos torneos en el circuito y dar de baja torneos existentes previamente. En cambio, el número C de categorías será constante.

Todos los torneos de la misma categoría ofrecen los mismos puntos de cara a la actualización del ranking y éstos dependen del nivel al que llega cada jugador o jugadora participante en el torneo. Supondremos que todas las categorías tienen un número máximo de niveles K (común a todas ellas), donde el nivel 1 corresponde al ganador o ganadora, el nivel 2 corresponde al o a la finalista, el nivel 3 corresponde a los jugadores o jugadoras derrotados en semifinales, el nivel 4 a los derrotados en cuartos de final, y así sucesivamente, hasta la ronda inicial del torneo.

Para cada par (i, j) , tal que i es una categoría entre 1 y C y j es un nivel entre 1 y K , existirá un valor entero $p_{ij} \geq 0$, que representa el número de puntos que gana un jugador o jugadora que llega al nivel j en un torneo de categoría i . Lógicamente, los puntos decrecen al aumentar el nivel y podrían llegar a ser cero a partir de un cierto nivel, que dependerá de la categoría.

Al igual que la mayoría del *ATP tour*, y que todo el *WTA tour*, asumiremos que todos los partidos de todos los torneos de todas las categorías se disputan al mejor de 3 *sets* (es decir, nunca al mejor de 5 *sets*). Asimismo, todos los *sets* en que se llegue a un empate a 6 juegos se resolverán mediante un *tie-break*.

Por otro lado, el circuito lleva un registro de los jugadores o jugadoras que pueden participar en sus torneos, identificados por su nombre. Inicialmente, habrá un conjunto de P jugadores o jugadoras, pero posteriormente podrán haber altas o bajas en el registro.

Asimismo existirá un ranking entre los jugadores o jugadoras, cuya inicialización y actualización se describen en la siguiente sección. El orden de ranking determinará el cuadro de emparejamientos entre los jugadores o jugadoras inscritos en un torneo, tal como se describirá más adelante.

1.2. Cálculo del ranking

Al mismo tiempo que se defina el conjunto inicial de P jugadores o jugadoras del circuito, se asignarán cero puntos a todos ellos/ellas y se establecerá un ranking inicial para los jugadores o jugadoras, con valores de ranking $1 \dots P$.

Supongamos que se acaba de celebrar un torneo t de una categoría c . Cada jugador/a p que haya participado en esta edición de t sumará a sus puntos de ranking los puntos conseguidos en dicha edición que, como hemos comentado anteriormente, dependen de la categoría c y del nivel alcanzado por p en el torneo. Si no se trata de la primera edición del torneo t entonces, además, a cada jugador/a q participante en la edición anterior de t se le restarán los puntos que q hubiera sumado en dicha edición anterior. Nótese que los/las participantes en dos ediciones consecutivas del torneo no tienen por qué ser los mismos.

El ranking se actualiza cada vez que finaliza un torneo, primero sumando y restando los puntos del torneo de la manera mencionada en el párrafo anterior, y luego ordenando decrecientemente por puntos totales a los jugadores o jugadoras del circuito. En caso de empate a puntos, se desempata según el valor de ranking anterior a la actualización (por orden creciente de ranking). Debido a esta forma de actualizar el ranking, se deduce que el ranking actual solo refleja los puntos obtenidos en la edición más reciente de cada torneo vigente del circuito.

Cuando se da de alta un jugador o jugadora, éste o ésta pasan a ocupar la última posición del ranking con cero puntos. Cuando se da de baja un jugador o jugadora, que ocupaba una cierta posición r del ranking, simplemente mejoran una posición en el ranking los jugadores o jugadoras que ocupaban las posiciones entre $r + 1$ y P .

Cuando se da de baja un torneo en el circuito, habrá que restar los puntos obtenidos en la última edición del mismo por los/las participantes y volver a actualizar el ranking.

1.3. Fases de un torneo

En cada edición de un torneo cualquiera del circuito podemos distinguir las siguientes fases:

1. Inscripciones

Existe un corto período de tiempo en el que los jugadores o jugadoras del circuito que lo deseen pueden inscribirse en el torneo. Sea n el número de inscritos una vez finalizado dicho período. Tanto en el ATP *tour* como en el WTA *tour*, se suele fijar el valor de n de cada torneo. Sin embargo, en esta práctica, consideraremos que el valor de n no está preestablecido y puede variar de una edición a otra. La única restricción que asumiremos en todo momento será que $8 \leq n \leq 2^{K-1}$, donde K es el número máximo de niveles fijado por el circuito (ya comentado

anteriormente). Notad que $K - 1$ es justamente el número máximo de rondas de un torneo, contando desde la ronda inicial hasta la final.

El resultado de esta fase es una secuencia ordenada crecientemente de n enteros que representan las posiciones en el ranking actual de los jugadores o jugadoras inscritos, a partir de la cual se pueden consultar sus nombres. Una vez se conocen los inscritos ordenados por ranking, las posiciones concretas en el ranking del circuito pasan a ser irrelevantes y (localmente al torneo) se asocia a cada jugador o jugadora un valor r entre 1 y n según su orden de ranking relativo. Por ejemplo, en la siguiente tabla representamos una posible lista ordenada de jugadores inscritos en un torneo de $n = 14$ jugadores:

Ranking	Nombre jugador	r
4	Tsitsipas	1
5	Nadal	2
6	Berrettini	3
13	Schwartzman	4
15	Bautista	5
17	Carreño	6
20	Alcaraz	7
24	Cilic	8
25	Monfils	9
26	Khachanov	10
27	Garin	11
31	Ramos	12
33	Dimitrov	13
37	Delbonis	14

2. Confección y publicación oficial del cuadro de emparejamientos

A continuación hay que confeccionar el cuadro de emparejamientos del torneo, que es un árbol binario que representa los enfrentamientos entre jugadores/as desde la ronda inicial hasta la final. La estructura de dicho árbol queda totalmente determinada por el número n de inscritos, que coincidirá con el número de hojas del árbol. También podrá determinarse de manera inambigua qué valor r corresponde a cada hoja, lo que dará lugar a los enfrentamientos de la primera ronda. Para esta construcción del cuadro, y solamente por conveniencia, se sigue la hipótesis que en todos los partidos del torneo ganará el jugador o jugadora de mejor ranking. Bajo esta hipótesis, el árbol resultado ha de cumplir las siguientes propiedades:

- Los dos participantes con mejor ranking sólo podrían jugar entre ellos en la final (si ganan todos los partidos anteriores), los cuatro mejores sólo podrían jugar entre ellos en las semifinales, los ocho mejores sólo podrían jugar entre ellos en cuartos de final, y así sucesivamente.
- En un emparejamiento directo entre a y b , si $a < b$ entonces a aparecerá en el árbol a la izquierda y b a la derecha. Esta regla se cumplirá inicialmente, antes

de disputarse los partidos, y mientras ganen los jugadores de mejor ranking, pero puede dejar de cumplirse a partir del momento en que un jugador de peor ranking vence a otro de mejor ranking, ya que entonces el jugador vencedor substituirá al jugador derrotado en el camino que marca el árbol hacia la final.

- c) La altura h del árbol viene dada por la ecuación $h = \lceil \log_2 n \rceil + 1$. Sea $m = 2^{h-1}$, el numero de nodos en el nivel h si éste está completo. En el caso particular $n = m$, todas las hojas estarían en el nivel h y éstas se agruparían en pares, con emparejamientos $(a \text{ vs. } b)$, según la fórmula $b = m + 1 - a, \forall a : 1 \leq a \leq m/2$. En el caso $n < m$, entonces tendríamos $m - n$ hojas en el nivel $h - 1$ (correspondientes a los mejores del ranking, es decir, a los valores r entre 1 y $m - n$, ya que según la fórmula anterior no tendrían rival en la ronda inicial y por tanto pasan automáticamente a la siguiente ronda) y $m - 2 * (m - n)$ hojas en el nivel h con el resto de jugadores emparejados así: $(a \text{ vs. } b)$, $b = m + 1 - a, \forall a : (m - n + 1) \leq a \leq m/2$.
- d) El resto de niveles del árbol (los niveles l tal que $1 \leq l < h$) estarán siempre completos de nodos. Inicialmente, y de forma oculta y provisional hasta que se jueguen los partidos, se podría asignar también un identificador r a todos los nodos internos (no hojas) del cuadro, siguiendo la misma fórmula que se ha descrito para el caso $n = m$ en el nivel h . Esto es, sea $z = 2^{l-1}$ el número de nodos del nivel l ; los nodos de ese nivel se emparejarían así: $(a \text{ vs. } b)$, donde $b = z + 1 - a, \forall a : 1 \leq a \leq z/2$. Para $l = 1$ sólo existe un nodo, la raíz, y ésta tendría provisionalmente el valor $r = 1$.

En la siguiente figura se muestra el árbol del cuadro de emparejamientos que resultaría para una inscripción de $n = 14$ jugadores o jugadoras (donde $h = 5$ y $m = 16$). En el momento de publicar el cuadro antes de disputarse el torneo, además de la estructura del árbol, sólo se visualiza la información de los jugadores o jugadoras asignados a las hojas del mismo.

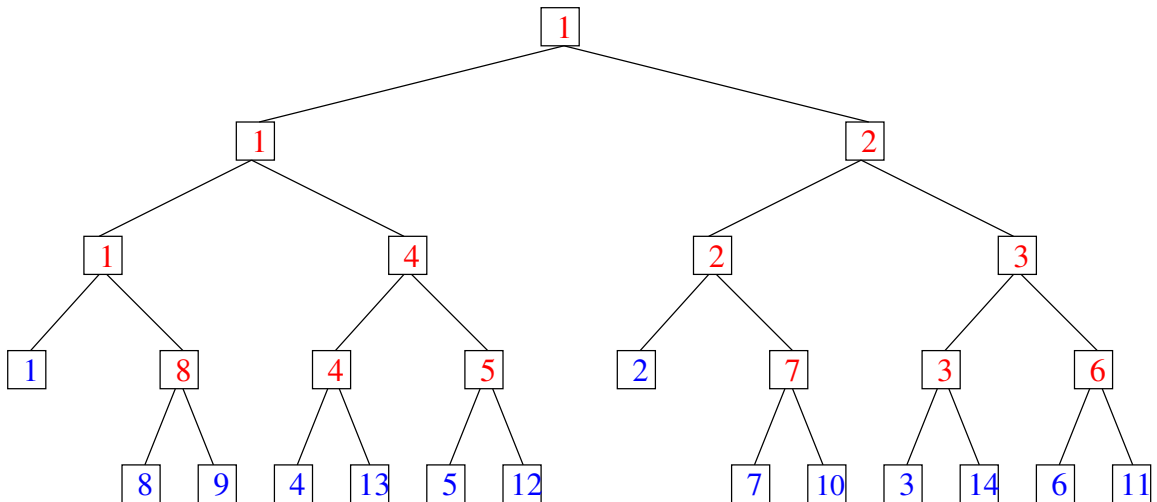


Figura 1: Cuadro de emparejamientos para $n = 14$. En rojo: valores auxiliares durante la construcción; en azul: valores definitivos

3. Desarrollo del torneo

El torneo se va disputando ronda a ronda siguiendo el cuadro de emparejamientos. El orden en que se disputan los partidos de una misma ronda es irrelevante. Al acabar cada partido (o *match*), la organización del torneo toma nota del resultado del mismo y anota el identificador r del vencedor (quien pasa a la siguiente ronda) en el cuadro de emparejamientos, concretamente en el nodo padre de los nodos correspondientes a los jugadores/as que han disputado el *match*. De esta manera, al acabar el torneo, el identificador r del ganador del torneo ocupará el nodo raíz del cuadro de emparejamientos.

4. Publicación oficial del cuadro de resultados

Cuando acaba el torneo, la organización publica oficialmente el cuadro de resultados, que también es un árbol binario, pero en el que en cada nodo tendremos los nombres de los dos jugadores que han disputado el *match* y el resultado del mismo. Por tanto, la raíz tendrá el nombre de los dos finalistas y el resultado de la final.

Se puede deducir que la estructura del árbol del cuadro de resultados será como la del árbol del cuadro de emparejamientos, excepto que los nodos hoja del cuadro de emparejamientos no serán nodos del cuadro de resultados (pero, en cambio, los nombres de los jugadores asociados a las hojas del cuadro de emparejamientos sí han de aparecer en el cuadro de resultados, concretamente en el nodo correspondiente a su primer partido).

El resultado de un *match* entre a y b , con a a la izquierda y b a la derecha en el cuadro, podrá ser de una de las tres formas siguientes:

a) $a_1-b_1, a_2-b_2, a_3-b_3$

Resultado de un partido en el que se han disputado 3 *sets*, para lo que, necesariamente, a y b habrán ganado un *set* cada uno en los dos primeros *sets*. a_1 , a_2 y a_3 representan los juegos ganados por a en el primer, segundo y tercer *set*, respectivamente, mientras que b_1 , b_2 y b_3 representan los juegos ganados por b en dichos *sets*. Los valores de a_i y b_i estarán entre 0 y 7, pero no todas las combinaciones son posibles. Sólo se pueden dar en un *set* los siguientes resultados: 7-6, 6-7, 7-5, 5-7, 6- x y x -6, donde x puede ser 0, 1, 2, 3 o 4.

b) a_1-b_1, a_2-b_2

Resultado de un partido en el que sólo se han disputado 2 *sets*, porque o bien a o bien b han ganado los dos primeros *sets*. a_1 y a_2 representan los juegos ganados por a en el primer y segundo *set*, respectivamente, mientras que b_1 y b_2 representan los juegos ganados por b en dichos *sets*. Los valores posibles de a_i y b_i son los ya mencionados.

c) 1-0 o 0-1

Resultado de un partido en el que un jugador ha vencido a otro por incomparecencia o abandono durante el partido (cualquiera que sea la causa: lesión, enfermedad, etc.). Nótese que, en caso de haberse disputado parcialmente el

partido, el resultado de los juegos o *sets* disputados no se tiene en cuenta ni se publica. El resultado será 1-0 si el vencedor es *a* y 0-1 si el vencedor es *b*.

El análisis del resultado, bajo las restricciones anteriores, permite determinar en todos los casos si el vencedor del *match* ha sido *a* o ha sido *b*.

En la siguiente figura se muestra el árbol de un cuadro de resultados posible para un torneo de $n = 14$ jugadores:

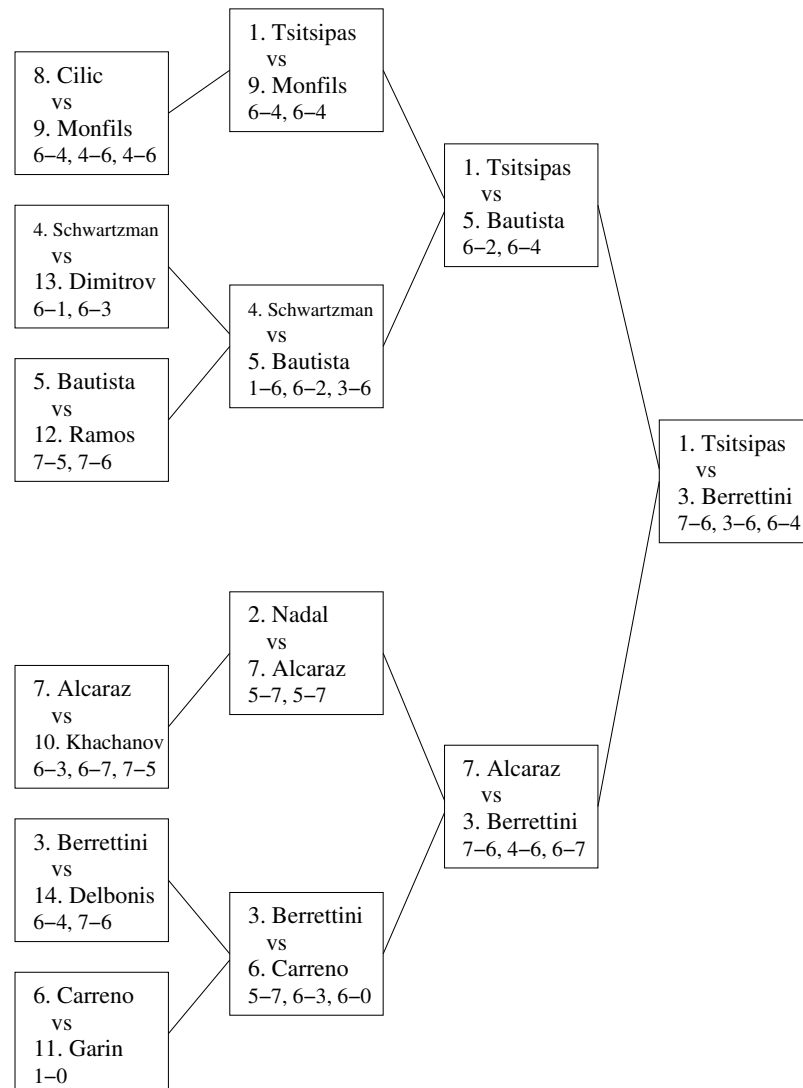


Figura 2: Posible cuadro de resultados para $n = 14$

5. Procesamiento de los resultados para el cálculo de puntos y estadísticas

Finalmente, la organización del torneo se encarga de calcular y publicar los puntos que ha ganado cada jugador/a participante, que dependen de la categoría del torneo y del nivel alcanzado en el cuadro de emparejamientos: el nivel 1, es decir, la raíz del árbol, corresponde al ganador del torneo, el nivel 2 corresponde a los finalistas, el nivel 3 a los semifinalistas, etc.

Los puntos obtenidos por los jugadores/as en la edición actual, junto con los obtenidos por los jugadores/as en la edición anterior del torneo (si la hubiera), se usarán para actualizar el ranking del circuito, tal como se ha descrito en un apartado previo.

Por último, el circuito mantiene un registro con algunas estadísticas globales de los jugadores/as (torneos disputados, número de partidos ganados y perdidos, número de *sets* ganados y perdidos, número de juegos ganados y perdidos), por lo que la organización actualizará las estadísticas globales de los participantes en la presente edición del torneo, según los resultados obtenidos en la misma. Los resultados del tipo 1-0 o 0-1 sólo afectarán a las estadísticas de partidos ganados y perdidos, pero no a las de *sets* y juegos.

2. Funcionalidades

2.1. Decisiones sobre los datos

Los datos del programa se dividen en dos partes. Las categorías y un primer grupo de torneos y jugadores se usarán para la inicialización del sistema. Una vez realizada dicha inicialización, el programa ofrecerá un repertorio de funcionalidades. Cada funcionalidad podrá requerir a su vez unos datos y producirá una salida que describimos con más precisión en el siguiente apartado.

En todo momento, se leerán solo datos sintácticamente correctos y no será necesario comprobar dicha corrección. Por ejemplo, si decimos que inicialmente el sistema contiene P jugadores registrados, el programa recibirá un entero P mayor o igual que 0 y a continuación exactamente P strings con los nombres de los jugadores sin repeticiones. Los datos de las categorías y los torneos serán asimismo sintácticamente correctos.

2.2. Programa principal: estructura y comandos

El programa principal empezará leyendo el número de categorías $C \geq 1$ y el número máximo de niveles $K \geq 4$, ambos constantes. A continuación se leerá una secuencia de C strings con los nombres asociados a las categorías identificadas por los valores entre 1 y C en orden creciente (de identificador, no de nombre). Luego se leerán $C \times K$ enteros mayores o iguales que 0, que serán los puntos por categoría y nivel, ordenados crecientemente por categorías y dentro de cada categoría ordenados crecientemente por nivel.

Seguidamente se leerá un número inicial de torneos $T \geq 0$ y una secuencia de T pares de string t y entero c , donde t será el nombre que identifica al torneo (sin repeticiones) y c la categoría entre 1 y C a la que pertenece el torneo.

Por último se leerá un número inicial de jugadores $P \geq 0$ y una secuencia de P strings con los nombres que identifican a los jugadores (sin repeticiones). El orden en que se lean los jugadores determinará el orden de ranking inicial (el i -ésimo jugador leído será el número i del ranking, para todo i entre 1 y P). Excepto dicho valor de ranking inicial, el resto de estadísticas de los jugadores se inicializarán a cero, incluidos sus puntos.

Hay que notar que las magnitudes T y P no serán constantes ya que podrán aumentar o disminuir al añadir posteriormente nuevos torneos o jugadores y al dar de baja torneos o jugadores registrados previamente.

Terminadas las inicializaciones, se procesará una serie de comandos acabados en un comando `fin`. La estructura general del programa principal será la siguiente:

```
lectura del conjunto de categorías;
lectura del conjunto inicial de torneos;
lectura del conjunto inicial de jugadores;
lee comando;
while (comando != "fin") {
    procesa comando;
    lee comando;
}
```

Los comandos aceptados se describen a continuación. Todo ellos se presentan en dos versiones, una con el nombre completo y otra con el nombre abreviado. La sintaxis exacta de la entrada y la salida de cada comando se podrá derivar del juego de pruebas público del ejercicio creado en el Jutge para cada entrega.

1. `nuevo_jugador p`: lee y añade un nuevo jugador con identificador p (si éste es válido). El comando admite la forma abreviada `nj`. Si ya existe un jugador en el circuito con el mismo identificador se imprime un mensaje de error. En caso contrario se añade el jugador con estadísticas a cero y última posición en el ranking y se imprime el número de jugadores P en el circuito después de añadirlo.
2. `nuevo_torneo t c`: lee un nuevo torneo con identificador t y categoría c y lo añade al circuito (si los datos son válidos). El comando admite la forma abreviada `nt`. Si la categoría c no está entre 1 y C se imprime un mensaje de error. Si lo está, pero ya existe un torneo en el circuito con el nombre t , se imprime otro mensaje de error. En caso contrario se imprime el número de torneos T en el circuito después de añadirlo.
3. `baja_jugador p`: da de baja en el circuito al jugador con identificador p . El comando admite la forma abreviada `bj`. Si no existe un jugador con identificador p se imprime un mensaje de error. En caso contrario se actualiza el ranking (desplazando una posición hacia arriba los jugadores siguientes en el ranking) y se imprime el número de jugadores P en el circuito después de darlo de baja. Si más tarde se añade otro jugador con el mismo nombre es como si el anterior no hubiera existido.
4. `baja_torneo t`: da de baja en el circuito al torneo con identificador t . El comando admite la forma abreviada `bt`. Si no existe un torneo con identificador t se imprime un mensaje de error. En caso contrario se actualiza el ranking después de restar los puntos conseguidos por los jugadores en la última edición del torneo (si se hubiera jugado alguna) y se imprime el número de torneos T en el circuito después de darlo de baja. Si más tarde se añade otro torneo con el mismo nombre es como si el anterior no hubiera existido.

5. `iniciar_torneo t`: se lee la inscripción en el torneo t y se confecciona e imprime el cuadro de emparejamientos de los jugadores inscritos. Se garantiza que el torneo t existe en el circuito. El comando admite la forma abreviada `it`. Primero se lee el número n de jugadores inscritos y luego se leen n enteros con sus posiciones en el ranking actual ordenadas crecientemente. El formato de escritura del cuadro de emparejamientos se deducirá del juego de pruebas público.
6. `finalizar_torneo t`: se leen los resultados del torneo t , se produce e imprime el cuadro oficial de resultados, y se listan los puntos para el ranking ganados por cada uno de los participantes; también se actualiza el ranking y las estadísticas de los jugadores. Se garantiza que el torneo t existe en el circuito y que previamente se ha ejecutado el comando `iniciar_torneo t` correspondiente[†]. El comando admite la forma abreviada `ft`. Los resultados del torneo se leen como un árbol binario de strings en preorden siguiendo la estructura del cuadro de emparejamientos. El formato de escritura del cuadro de resultados se deducirá del juego de pruebas público. La lista de jugadores y puntos ganados por cada uno se escribe por orden de ranking de los jugadores en el momento de iniciar el torneo. La actualización del ranking ha de tener en cuenta, además de los puntos ganados en la presente edición que se suman, los puntos ganados en la edición anterior del torneo, si la hubiera, que se restan.
7. `listar_ranking`: se listan, por orden creciente de ranking actual, la posición, el nombre y los puntos de cada jugador del circuito. El comando admite la forma abreviada `lr`.
8. `listar_jugadores`: se listan, por orden creciente de identificador (nombre), el nombre, la posición en el ranking, los puntos y el resto de las estadísticas de cada jugador del circuito. El comando admite la forma abreviada `lj`.
9. `consultar_jugador p`: lo mismo que la anterior, pero solamente para el jugador con identificador p . Si no existe un jugador con identificador p se imprime un mensaje de error. El comando admite la forma abreviada `cj`.
10. `listar_torneos`: se listan, por orden creciente de identificador (nombre), el nombre y la categoría de cada torneo del circuito. El comando admite la forma abreviada `lt`.
11. `listar_categorías`: se listan, por orden creciente de identificador, el nombre y la tabla de puntos por niveles (en orden creciente de nivel) de cada categoría del circuito. El comando admite la forma abreviada `lc`.
12. `fin`: termina la ejecución del programa

[†]Aunque no es estrictamente necesario, puede suponerse la consistencia temporal de la secuencia de torneos: entre dos ediciones consecutivas de un torneo se celebran los demás torneos vigentes (pero no siempre en el mismo orden), un torneo no se puede iniciar si ya se ha iniciado antes y aún no se ha finalizado, etc. Del mismo modo se pueden suponer otras condiciones “realistas”, como que si se inicia un torneo antes de finalizar algún otro, esos torneos no tendrán jugadores en común