

Tipo D

Ante el cambio de equipo continuo por parte de los jugadores, la Liga de Fútbol Profesional necesita un nuevo gestor de futbolistas, más eficiente, que le permita conocer el equipo actual que tiene fichado a cada jugador, o ciertos datos sobre el historial de fichajes.

La implementación del sistema se realizará mediante un TAD `gestor_futbolistas` con las siguientes operaciones:

- `fichar(jugador, equipo)`: si el `jugador` (un `string`) no está dado de alta en el sistema, entonces se registra por primera vez como perteneciente al `equipo` (otro `string`). Si el `equipo` no estaba dado de alta, se le da en ese momento. Si el `jugador` ya estaba dado de alta, la operación supone un cambio de equipo, pasando a estar fichado por el nuevo `equipo`. Si el `jugador` ya estaba fichado por este `equipo`, la operación no tiene ningún efecto.
- `equipo_actual(jugador)`: devuelve el equipo actual por el que está fichado este `jugador`. En caso de que el `jugador` no esté dado de alta, lanzará una excepción `domain_error` con mensaje `Jugador inexistente`.
- `fichados(equipo)`: devuelve cuántos jugadores tiene fichados actualmente el `equipo`. En caso de que el equipo no esté dado de alta, lanzará una excepción `domain_error` con mensaje `Equipo inexistente`.
- `ultimos_fichajes(equipo, n)`: devuelve en un tipo de datos lineal los `n` últimos jugadores fichados por el `equipo` ($n > 0$) y que aún siguen estando fichados por ese `equipo`. La tipo lineal estará ordenada por el momento en el que fueron fichados, primero el último fichaje. En caso de que el `equipo` tenga menos de `n` jugadores, se devolverán todos, ordenados de la misma manera. En caso de que el equipo no esté dado de alta, lanzará una excepción `domain_error` con mensaje `Equipo inexistente`.
- `cuantos Equipos(jugador)`: devuelve el número de equipos distintos por los que ha estado fichado el `jugador`. Si el `jugador` no está dado de alta en el sistema, devolverá 0.

Requisitos de implementación.

La implementación de las operaciones debe ser lo más eficiente posible. Por tanto, debes elegir una representación adecuada para el TAD, implementar las operaciones y justificar la complejidad resultante.

Los métodos del TAD no deben mostrar nada por pantalla. El manejo de la entrada y salida de datos se realizará en funciones externas al TAD.

Entrada

La entrada consta de una serie de casos de prueba. Cada caso está formado por una serie de líneas, en las que se muestran las operaciones a llevar a cabo, una por cada línea: el nombre de la operación seguido de sus argumentos. Los nombres de jugadores y equipos serán cadenas de caracteres sin espacios. La palabra `FIN` en una línea indica el final de cada caso.

Salida

La operación `fichar` no produce salida, salvo en caso de error. Con respecto a las restantes:

- Tras llamar a `equipo_actual` debe imprimirse una línea con el texto `El equipo de XXX es YYY`, siendo `XXX` el jugador consultado e `YYY` su equipo.
- Tras llamar a `fichados` debe imprimirse una línea con el texto `Jugadores fichados por XXX: N`, siendo `XXX` el nombre del equipo consultado y `N` su número de jugadores.
- Tras llamar a `ultimos_fichajes` debe imprimirse una línea con el texto `Ultimos fichajes de XXX:` seguida por los últimos jugadores fichados, separados por espacios, y siendo `XXX` el nombre del equipo consultado.

- Tras llamar a `cuantos Equipos` debe imprimirse una línea con el texto **Equipos que han fichado a XXX: N**, donde **XXX** es el jugador consultado y **N** el número de equipos por los que ha estado fichado.

Cada caso termina con una línea con tres guiones (---). Si una operación produce un error, entonces se escribirá una línea con el mensaje **ERROR:**, seguido del error que devuelve la operación, y no se escribirá nada más para esa operación.

Entrada de ejemplo

```
fichar jugador1 equipo1
fichar jugador2 equipo1
ultimos_fichajes equipo1 3
fichar jugador2 equipo2
fichar jugador3 equipo1
ultimos_fichajes equipo1 2
fichar jugador1 equipo3
fichar jugador4 equipo1
cuantos Equipos jugador1
fichados equipo1
fichados equipo2
equipo_actual jugador4
fichar jugador5 equipo1
fichar jugador4 equipo1
ultimos_fichajes equipo1 8
FIN
fichados equipo2
cuantos Equipos jugador1
fichar jugador1 equipo1
fichar jugador1 equipo2
ultimos_fichajes equipo1 7
ultimos_fichajes equipo2 7
ultimos_fichajes equipo3 7
equipo_actual jugador10
FIN
```

Salida de ejemplo

```
Ultimos fichajes de equipo1: jugador2 jugador1
Ultimos fichajes de equipo1: jugador3 jugador1
Equipos que han fichado a jugador1: 2
Jugadores fichados por equipo1: 2
Jugadores fichados por equipo2: 1
El equipo de jugador4 es equipo1
Ultimos fichajes de equipo1: jugador5 jugador4 jugador3
---
ERROR: Equipo inexistente
Equipos que han fichado a jugador1: 0
Ultimos fichajes de equipo1:
Ultimos fichajes de equipo2: jugador1
ERROR: Equipo inexistente
ERROR: Jugador inexistente
---
```

Autor: Isabel Pita