

# Oficinas de empleo

El ministerio de trabajo desea agilizar los trámites realizados por las oficinas de empleo. Nos piden diseñar un sistema, en el cual las personas se apuntan al paro indicando aquellos empleos para los que están cualificadas. Cuando se recibe en la oficina una oferta de empleo, se selecciona a la persona cualificada para ese empleo que lleva más tiempo apuntada en la oficina.

La implementación del sistema se deberá realizar como un TAD `oficinaEmpleo` con las siguientes operaciones:

- `altaOficina(nombre, empleo)`: añade una nueva persona (un `string`), a un empleo (`string`) para el que está cualificada. Si la persona no está apuntada en la oficina se dará de alta con el empleo para el que está cualificada. Si la persona ya está apuntada en la oficina se añadirá a su solicitud el nuevo empleo, si no estaba apuntada para él. Si ya estuviese apuntada para el empleo que solicita no se la volverá a apuntar.
- `ofertaEmpleo(empleo)`: La operación devuelve la persona cualificada para realizar este empleo que más tiempo lleva apuntada en la oficina. La persona dejará de estar apuntada en la oficina de empleo. Si un empleo sólo tenía a esta persona apuntada dejará de estar en la aplicación. Si el empleo no está en la aplicación lanza una excepción `domain_error` con mensaje `No existen personas apuntadas a este empleo`.
- `listadoEmpleos(persona)`: Obtiene un vector con los empleos para los que está cualificada una persona en orden alfabético. En caso de que la persona no exista, lanza una excepción `domain_error` con mensaje `Persona inexistente`.

## *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. La palabra `FIN` en una línea indica el final de cada caso.

Los nombres de las personas y las descripciones de los empleos son cadenas de caracteres sin blancos.

## Salida

Para cada caso de prueba se escribirán los datos que se piden. Las operaciones que generan salida son:

- `ofertaEmpleo`, que debe escribir una línea con el empleo que se ofrece seguido del nombre de la persona seleccionada.
- `listadoEmpleos`, que debe escribir una línea con el nombre de la persona seguido de todos los empleos para los que está cualificada la persona, en orden alfabético y separados por blancos.

Cada caso termina con una línea con tres guiones (`---`).

Si una operación produce un error, entonces se escribirá una línea con `ERROR:`, seguido del error que devuelve la operación, y no se escribirá nada más para esa operación.

## Entrada de ejemplo

```
altaOficina Carlota administracion
altaOficina Carlota secretariado
altaOficina Carlota recursosHumanos
altaOficina Juan administracion
altaOficina Juan riesgosLaborales
ofertaEmpleo administracion
altaOficina Beatriz administracion
altaOficina Beatriz biblioteca
altaOficina Beatriz jardineria
ofertaEmpleo administracion
listadoEmpleos Carlota
altaOficina Carlota informatica
altaOficina Beatriz informatica
listadoEmpleos Beatriz
ofertaEmpleo informatica
ofertaEmpleo informatica
ofertaEmpleo informatica
ofertaEmpleo biblioteca
FIN
```

## Salida de ejemplo

```
administracion: Carlota
administracion: Juan
ERROR: Persona inexistente
Beatriz: administracion biblioteca informatica jardineria
informatica: Carlota
informatica: Beatriz
ERROR: No existen personas apuntadas a este empleo
ERROR: No existen personas apuntadas a este empleo
---
```

**Autor:** Isabel Pita