

Repaso Integrador

Gimnasios (GobsGym)

GobsGym es una plataforma simple para registrar los accesos de los deportistas a los diferentes aparatos de un gimnasio y conseguir estadísticas de las rutinas efectuadas. Para modelar la plataforma, se introducen las estructuras de datos que se describen a continuación. Existen 3 tipos de Ejercicios que se ofrecen en la plataforma y se representan con el siguiente tipo de datos:

```
type TipoDeEjercicio is variant {  
    case Musculación{}  
    case Cardio{}  
    case Fitness {}  
}
```

De cada máquina deportiva conocemos su nombre, el tipo de ejercicio, el precio por uso del mismo, y una indicación sobre si el equipo se puede usar o está siendo reparada, por lo que no se permite su uso. Además tenemos un tipo para los Tickets de uso de las máquinas

```
type Máquina is record {  
    /*  
        PROPÓSITO: modela un máquina de GobsGym  
        INV.REP.:  
            * "nombre" no puede estar vacío  
            * "precio" es >= 0  
    */  
    field nombre          // String  
    field tipo            // TipoDeEjercicio  
    field enReparación    // Booleano  
    field precio          // Número  
}
```

```
type TicketDeUso is record {  
    /*  
        PROPÓSITO: modela el uso de una máquina por un deportista  
        INV.REP.:  
            * "númeroDeTarjeta" es un número >=0  
            * "nombreDeMáquina" no puede estar vacío  
    */  
    field númeroDeTarjeta // Número  
    field nombreDeMáquina // String  
}
```

El sistema se representa por una lista de las máquina que están disponibles y la lista de los usos realizados por los deportistas, mediante el siguiente tipo:

```
type GobsGym is record {  
    /*  
        PROPÓSITO: modela una plataforma de registro de las  
                    máquinas usadas en el gimnasio  
        INV.REP.: "máquinas" no contiene dos máquinas con el mismo nombre  
    */  
    field usosEfectuados // [TicketDeUso]  
    field máquinas       // [Máquina]
```

```
}
```

En los ejercicios que siguen, programaremos algunos aspectos del sistema utilizando el lenguaje Gobstones.

EJERCICIOS

Ejercicio 1)

Definir la función `cantidadDeAccesosAl_RegistradosEn_(nombreDeMaquina, gobsGym)` que dado un nombre de máquina y un sistema GobsGym, describe la cantidad de contrataciones que se realizaron para esa máquina.

Observación: notar que si el nombre de la máquina no existe, debe describir al número 0.

Ejercicio 2)

Definir la función `máquinasQueSeUsaronMásDe_RegistradosEn_(cantidad, gobsGym)` que dado un número y un sistema GobsGym retorna la lista de los nombres de las máquinas que registraron más tickets que la cantidad recibida.

Ejercicio 3)

Definir `gymConAccesosDe_A_En_(númeroDeTarjeta, listaDeNombresDeMáquinas, gobsGym)` que describe el GobsGym resultante de intentar registrar el uso de la tarjeta dada por el número de tarjeta, que se supone existente, en cada uno de las máquinas cuyos nombres se reciben en la lista `listaDeNombresDeMáquinas`. Notar que solo se podrán registrar los tickets para máquinas que estén actualmente disponibles, es decir que NO estén en reparación.

Ejercicio 4)

Definir la función `máquinasContratadasPor_En_(listaDeNrosDeTarjetas, gobsGym)` que describe una lista de listas de Máquinas. El primer elemento es la lista de todas las Máquinas que se usaron con el número de tarjeta que es el primer elemento de la `listaDeNrosDeTarjetas`, el segundo elemento son todos los Máquinas que se usaron con la tarjeta cuyo número es el segundo elemento de la lista `listaDeNrosDeTarjetas`.

Notar que si `listaDeNrosDeTarjetas` tiene `n` elementos, la lista resultante también tendrá `n` elementos, y cada uno será una lista de `Máquinas`.

Ejercicio 5)

Definir la función `recaudaciónDeMáquinasParaFitnessEn_(gobsGym)` que describe un número que es la cantidad de dinero recaudado por las máquinas de tipo Fitness.

Ejercicio 6)

Definir la función `hayMáquinaEnReparaciónConGananciaMayorA_En_(ganancia, gobsGym)` que describe si alguna máquina recaudó esa ganancia o un número mayor, y está actualmente siendo reparada.