

## Ejercicio Carrera

Se tiene una clase Carrera que representa una carrera de atletismo. La carrera tiene un conjunto de participantes, cada uno con un identificador único y un nombre único.

Además, se registra el tiempo y la posición de llegada de cada participante que finaliza la carrera.

A continuación se detalla la estructura de representación elegida. Escribir en español y en lenguaje formal el invariante de representación de la misma.

```
class Carrera {
public:
    Carrera();

    const set<int> & participantes();

    // PRE: el participante 'id' está registrado
    string nombre(int id) const;

    // PRE: el participante 'id' está registrado y ya llegó a la meta
    float tiempo_de_llegada(int id) const;

    // PRE: el participante 'id' está registrado y ya llegó a la meta
    int posicion(int id) const;

    // PRE: ni el 'id' y ni el 'nombre' están registrados previamente
    void registrar_participante(int id, string nombre);

    // PRE: el participante 'id' está registrado y aún no llegó a la meta
    void registrar_llegada(int id, float tiempo);
private:
    vector<int> _tabla_posiciones;
    set<int> _ids_participantes;
    map<int, float> _tiempo_llegada;
    set<string> _nombres_participantes;
    map<string, int> _nombre_a_id;
};
```

## Propiedades individuales

### En español:

`_tabla_posiciones` tiene como mínimo un elemento. Para que sea una carrera, tiene que haber alguien corriendo, o, si se quiere, dos personas como mínimo. Esas personas estarán representadas con un ID.

`_ids_participantes` es un conjunto que no tiene elementos repetidos (porque es un conjunto) y puede ser cualquier cosa el número del ID, depende de qué método de identificación se use. Los IDs son los participantes de la carrera.

`_tiempo_llegada` tendrá como claves los IDs de los participantes, y como valor tendrá el tiempo que tardaron en correr (si no terminó, se pone como tiempo cero). Como es de esperar, el tiempo de llegada es un valor positivo y distinto de cero. No tiene elementos repetidos. No hay elementos repetidos.

`_nombres_participantes` no tiene elementos repetidos, y contiene los nombres de los participantes. Estos nombres están representados como strings, y no pueden ser vacíos.

`_nombre_a_id` tiene los nombres en strings asociados a IDs. Los strings no pueden ser vacíos.

### En lógica formal:

`_tabla_posiciones`:  $|e\_tabla\_posiciones| \geq 2$ .

Asumiendo que el método de identificación es arbitrario (no se pide nada específico en cuanto a esto) y deben correr como mínimo dos personas.

Skippeo esto porque, como es un conjunto, claramente no va a tener elementos repetidos.

`_tiempo_llegada`:  $(\forall id: \text{int})(id \in \text{claves}(e\_tiempo\_llegada) \Rightarrow e\_tiempo\_llegada[id] \geq 0)$

`_nombres_participantes`:  $(\forall s: \text{string} \in e\_nombres\_participantes) \Rightarrow (|s| \geq 1)$

`_nombre_a_id`:  $(\forall s: \text{string} \in \text{claves}(e\_nombre\_a\_id)) \Rightarrow |e\_nombre\_a\_id[s]| \geq 1$

## Relaciones

### En español:

En cuanto a tamaños, `_tabla_posiciones` tiene el mismo tamaño que el cardinal de `_ids_participantes`, que la cantidad de claves en `_tiempo_llegada` y `_nombre_a_id`, y que el cardinal de `_nombres_participantes`.

Los elementos de `_ids_participantes` son los mismos valores asociados en `_nombre_a_id`, las mismas claves que en `_tiempo_llegada` y deben estar presentes en `_tabla_posiciones`.

Los elementos de `_nombres_participantes` son también las claves en `_nombre_a_id`.

### En lógica formal:

*Alta paja de hacerlo. Por hoy, me fundí.*