

Contenedores utilizados

Despues de buscar informacion y leer sobre el tema, pude observar que (en mi caso) el contenedor que cumpliera con mis necesidades era **map**, ya que por su estructura de *Key-Value* me facilito las cosas al momento de enlazar o relacionar las **cuentas** con los **eventos** recibidos.

Decidi separar el problema en:

- Almacenamiento de **eventos**
- Almacenamiento de **cuentas**

Creando para cada uno un **map** y relacionandolos.

El **map** principal esta definido:

```
map<string, eventTypeMap> logMap;
```

Este contiene un `string` que son las **cuentas** y un `typedef eventTypeMap` que hace referencia a:

```
typedef map<string, int> eventTypeMap;
```

Este contiene un `string` que son los **eventos** recibidos y un `int` que es la cantidad de veces que se ha recibido ese evento.

Graficamente la estructura quedaria: Cuenta -> (tipoEvento -> contador)

Ese seria el almacenamiento principal.

Utilice tambien un almacenamiento auxiliar, para facilitar el manejo de datos al momento de mostrarlos.

`vector<pair<string, int>> accounts;` Almacena las cuentas con el numero total de eventos recibidos.

`vector<pair<string, int>> events;` Almacena los eventos con el numero total de llamadas.

Estos **vectores** fueron utilizados para facilitar el ordenamiento de los datos.