

# Ayudantía n° 7

Maps y maps con vectores

Diego Mena



# Mapas

Un mapa contiene dos tipos de datos, uno el cual funcionara como llave para poder acceder al otro tipo de dato, esto se puede observar de la siguiente manera:

LLave	LLave	LLave
Cont.	Cont.	Cont.

```
#include <map>
//Tipo de elemento=TD

map<TD1, TD2>Nombre;
map<string, int>Nombre;
map<string,
vector<NombreClase*>>Nombre;

Nombre[LLave]=Contenido;
Nombre[string]=Int;
```

¿Como se ve esto?

```
map<int, string>Compas;
```

2182	2171	2195
Lili	Ariel	Caña

```
Compas[2195]="Caña";
```

LLave                      Contenido  
2195                      →    Caña

Mapa con vector

Sec1	Sec2	Sec3
Ariel	Dani	Jorge
Lili	Aceitu	Chipi
Caña	Dante	Aldo
Basti	Branco	Xavi

```
map<int, vector<string>>Compas;
```

```
Compas[Grupo].push_back(nombre);
```

# Funciones

- Añadir algo al mapa: `map[llave] = valor;`
- `begin()`: Retorna el iterador de la primera posición.
- `end()`: Retorna el iterador antes de la última posición.
- `empty()`: Devuelve true si el mapa está vacío.
- `size()`: Devuelve el tamaño del mapa.
- `count(llave)`: Retorna true si la llave existe en el mapa, false en caso contrario.
- `find(llave)`: Retorna un iterador al elemento con la clave especificada.
- `erase(llave)`: Borra el contenido de la llave.
- `clear()`: Borra todos los elementos del mapa.
- `at(llave)`: Retorna el valor asociado a la llave; lanza una excepción si la llave no existe.
- `swap(map)`: Intercambia el contenido de dos mapas

¿Como podemos recorrer un map?

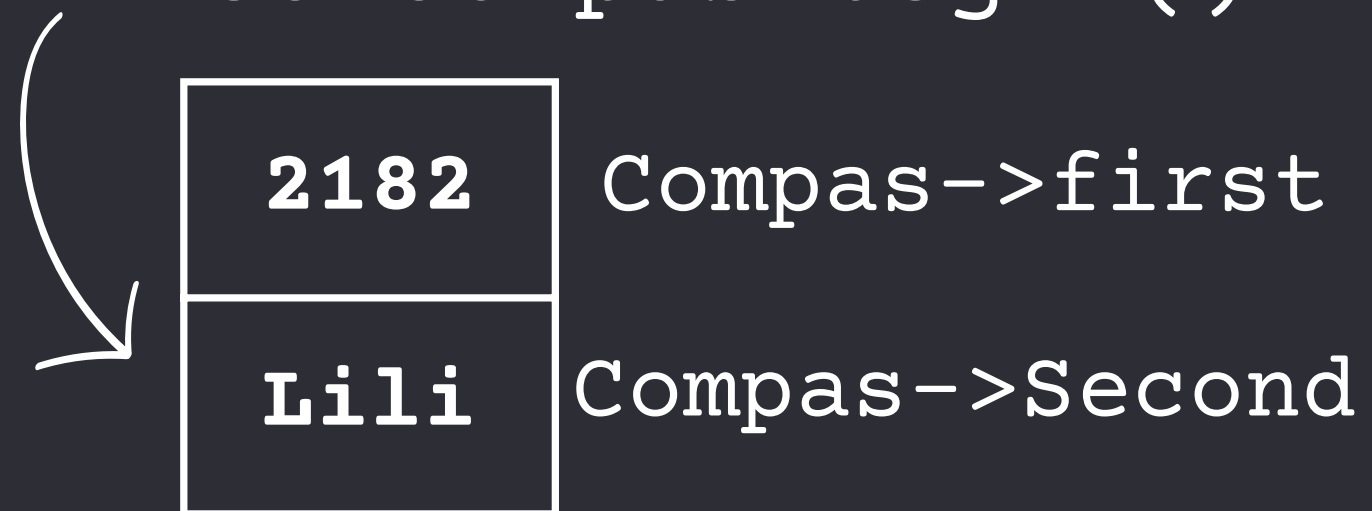
```
50 void MostrarTodosConIterador()
51 {
52     //map<int, string> Compas;
53     if (Compas.empty())
54     {
55         cout << "No hay compas registrados. :c" << endl;
56     }
57     else
58     {
59         map<string, int>::iterator rec;
60
61         for (rec = Compas.begin(); rec != Compas.end(); rec++)
62         {
63             cout << "Rut: " << rec->first << " Nombre: " << rec->second << endl;
64         }
65     }
66 }
```

## Compas

2182	2171	2195	
Lili	Ariel	Caña	

### Primera iteración del for

Rec=compas.begin()

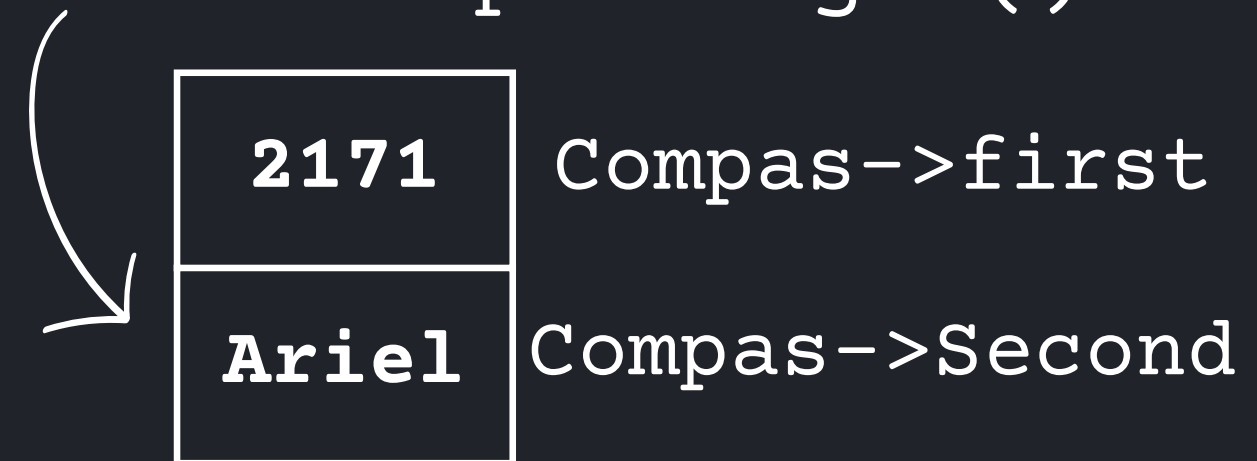


Salida:

Rut:2182 ; Nombre:Lili

### Segunda iteración del for

Rec=compas.begin()+1



Salida:

Rut:2171 ; Nombre:Ariel

.....

### Cuarta iteración del for

Rec=compas.begin()+3 == compas.end()

¿Como podemos recorrer un map con vector?

```
50 void MostrarTodosConIterador()
51 {
52     //map<int, string> Compas;
53     if (Compas.empty())
54     {
55         cout << "No hay compas registrados. :c" << endl;
56     }
57     else
58     {
59         map<string, int>::iterator rec;
60         for (rec = Compas.begin(); rec != Compas.end(); rec++)
61         {
62             cout << rec->first << endl;
63             for(int i=0;i<rec->second.size();i++)
64             {
65                 cout<<"Nombres: "<<rec->second[i]<<endl;
66             }
67         }
68     }
69 }
```

Primera iteración del for

`Rec=compas.begin()`



Sec1	Sec2	Sec3
Ariel	Dani	Jagsito
Lili	Benja.A	Chipi
Caña	Dante	Aldo
Basti	Branco	Xavi

Sec1
Ariel
Lili
Caña
Basti

`Compas->first`



`Compas->Second`

Salida:  
Grupo1  
Ariel  
Lili  
Caña  
Basti

```
for(int i=0;i<rec->second.size();i++)  
cout<<"Nombre: "<<rec->second[i]<<endl;
```





## Ejercicio mapas

Crea una clase GestionCalificaciones que permita gestionar las calificaciones de los estudiantes. Esta clase debe incluir las siguientes funcionalidades:

- **Agregar Estudiantes:** Un método agregarEstudiante que recibe el nombre del estudiante y lo añade a un registro con una calificación inicial de 1.0.
- **Agregar Notas:** Un método agregarNotas que permite ingresar tres notas para un estudiante específico. Este método calculará el promedio de las notas y lo actualizará en el registro del estudiante. Si el estudiante no existe, debe mostrar un mensaje indicando que no se encontró.
- **Imprimir Promedios:** Un método imprimir que muestra el nombre de cada estudiante junto con su promedio de notas.



# Ejercicio mapas con vector

Se quiere gestionar la información de las películas, las cuales están organizadas por género. El sistema debe permitir agregar géneros y títulos de películas, así como contar la cantidad de películas en cada género.

Requisitos:

Clase Netflix: Debe contener un mapa llamado peliculas que tenga como llave un string y como contenido un vector, el cual almacene los nombres de las películas.



Funciones pedidas:

- `agregarGenero(string genero)`: Agrega un nuevo género a la biblioteca. Si el género ya existe, debe informar al usuario.
- `agregarTitulo(string genero, string titulo)`: Agrega un título de película al género especificado. Si el género no está registrado, debe informar al usuario.
- `mostrarTitulosPorGenero(string genero)`: Muestra todos los títulos de películas bajo el género especificado. Si el género no está registrado, debe informar al usuario.
- `contarPeliculasPorGenero(string genero)`: Retorna la cantidad de películas en el género especificado. Si el género no está registrado, debe informar al usuario y retornar 0.

# Contacto {



**+56 9 6070 8865**



**diego.mena2@mail.udp.cl**



**<https://github.com/Dmena1/Ayudantia-prograAvanzada>**

}