



Universidad Veracruzana

**Universidad
Veracruzana**



Facultad de
ESTADÍSTICA E INFORMÁTICA

Estructura de Datos

Luis Daniel Carmona Villa

4to Semestre

Estructuras Asociativas

Sánchez García Ángel Juan

09/Mayo/2018

Estructura de Datos

Estructuras Asociativas.

Una estructura asociativa como su nombre lo indica, una manera de asociar un elemento a través de otros objetos llamados clave.

Un ejemplo que encontré conforme a lo dicho anteriormente son las tablas hash.

Tabla hash: se puede ver como un conjunto de entradas. Cada una de estas entradas tiene asociada una clave única, por lo tanto, diferentes entradas de una misma tabla tendrán diferentes claves. Esto implica, que una clave identifica unívocamente a una entrada en una table hash.

Por otro lado, las entradas de las tablas hash están compuestas por dos componentes, la propia clave y la información que se almacena en dicha entrada.

En el cierto punto de vista se puede decir que estas tablas tienen relación a una base de datos, es decir cuando queremos almacenar por ejemplo algún producto, ese producto estará identificado por una clave única en este caso un ID, y a partir de el se puede acceder a la búsqueda del producto y después consultar ya sea descripción o existencias, etc.

Estas tablas son estructuras de datos que se usan en aplicaciones que manejan una secuencia de elementos. Con las tablas de dispersión se implementa eficientemente el tipo abstracto de datos Diccionario.

Estructura Set

Al referirnos al conjunto set nos referimos a una colección de valores únicos, La principal característica de los contenedores asociativos es su capacidad para almacenar objetos basados en un valor clave y la recuperación eficientes de objetos vía las claves

Ejemplo.

- *set<clave, comparar>: colección ordenada de claves. Esto no permite claves duplicadas y es un candidato ideal para un grupo de registros **ordenados único**.*

Estructura Pair

La estructura pair es muy similar al set, solo que el propósito de la estructura pair es contener una pareja de valores.

En este caso estos valores se llamarán first y second.

Cada uno de los miembros de pair pueden ser de tipos diferentes.

La función de pair es que sirve para construir contenedores de tipo MAPS.

Estructura Maps

Un mapa es una colección de objetos que se pueden referenciar por sus claves asociadas. En resumen, un mapa almacena los objetos asociados a un valor clave; un conjunto almacena solo la clave, esto sirve para recuperar la información de manera rápida a través del acceso del valor clave.

Ejemplo.

- *map<clave, comparar>: colección de objetos que pueden ser referenciados por un valor clave. Esto es un candidato ideal para búsquedas de elementos no **secuenciales**.*

Iterator

El iterador es aquel que nos permite el acceso a los elementos de una colección, en este caso podría ser de un map

En el ejemplo siguiente podemos ver como esta en función el iterador en una estructura de tipo map, el map va a reservar dos variables string, con el nombre y el número telefónico, es importante analizar que con la función pair implementamos el atributo nombre como first y el teléfono como second, a través del iterador asignado buscamos el elemento a través del valor clave, aquí entra en juego el pair por que al asignar el iterador decimos que p->second es el atributo via clave por el cual accederemos a la recolección de los datos.

```
[*] Sin Nombre | 2.-Pilastatica.cpp | ProyectoPilast3.cpp | ProyectoCola.cpp | Orde
1  #include <iostream>
2  #include <string>
3  #include <utility>
4  #include <map>
5  #include <iomanip>
6  using namespace std;
7  typedef pair<string, string >componente;
8
9  int main(int argc, char** argv) {
10     cout<<endl<<"Otra simple prueba de map"<<endl;
11     map<string,string>directorio;
12     directorio.insert(pair<string, string>("Blanca","555-6666"));
13     directorio.insert(make_pair("Oscar","555-5555"));
14
15     directorio.insert(componente("Teresa","555-4444"));
16     directorio.insert(componente("Carlos","555-3333"));
17     directorio.insert(componente("Ruben","555-2222"));
18     directorio.insert(componente("Juan","555-1111"));
19     directorio.insert(componente("Andrea","555-0000"));
20     string s= "Ruben";
21     map<string, string >::iterator p= directorio.find(s);
22     if(p!=directorio.end())
23         cout<<"Numero telefonico de: "<<s<<"="<<p->second<<endl;
24     else cout<<s<<"no esta en el directorio.\n";
25     return 0;
26 }
```

The image shows a C++ IDE with a project named 'ESTRUCTURA DE DATOS'. The left pane displays the source code for 'MapString.cpp'. The code defines a 'Map' class using a 'std::map' and a 'componente' struct. It includes a 'main' function that creates a map, inserts several entries, and then searches for 'Ruben'. The right pane shows the execution output, which matches the program's logic. At the bottom, a 'Compilation results...' window shows 0 errors and 0 warnings.

```
1 #include <iostream>
2 #include <string>
3 #include <utility>
4 #include <map>
5 #include <iomanip>
6 using namespace std;
7 typedef pair<string, string> componente;
8
9 int main(int argc, char** argv) {
10     cout<<endl<<"Otra simple prueba de map"<<endl;
11     map<string, string> directorio;
12     directorio.insert(pair<string, string>("Elaiza", "555-6666"));
13     directorio.insert(make_pair("Oscar", "555-5555"));
14
15     directorio.insert(componente("Teresa", "555-4444"));
16     directorio.insert(componente("Carlos", "555-3333"));
17     directorio.insert(componente("Ruben", "555-2222"));
18     directorio.insert(componente("Juan", "555-1111"));
19     directorio.insert(componente("Andrea", "555-0000"));
20     string s = "Ruben";
21     map<string, string>::iterator p = directorio.find(s);
22     if(p != directorio.end())
23         cout<<"Numero telefonico de: "<<s<<"="<<p->second<<endl;
24     else cout<<"no esta en el directorio."<<endl;
25     return 0;
26 }
```

Otra simple prueba de map
Numero telefonico de: Ruben=555-2222

Process exited after 2.971 seconds with return value 0
Presione una tecla para continuar . . .

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Danny\Desktop\ESTRUCTURA DE DATOS\Mapa\Mapa1.exe

Referencias:

Fuentes de libros Universidad Veracruzana, Fac. Estadística e Informática.

Carmelo Sánchez González. (2007). Estructura de Datos C++. Madrid: Impresos en Edigrafos, S.A.

Concepción Fernández Madrid. (2000). Programación en C++. Algoritmos, estructuras de datos y objetos. España: Impresos de Edigrafos, S. L.