

# Lenguaje C++ & estructuras

29 de agosto de 2017

# Lenguaje

¿Cómo empezamos?

Para empezar debemos conocer algunos aspectos mínimos del lenguaje:

- Documentación oficial:
  - `http://en.cppreference.com/w/`
  - `http://www.cplusplus.com/doc/tutorial/`
- Tipos de datos básicos y su uso.`http://www.cplusplus.com/doc/tutorial/variables/`
- Tipos de datos armados.
- Las librerías que se necesitan importar en cada caso:
  - Entrada y salida: `iostream`.

# Estructuras

## Vector

### Definición

```
1 #include <vector>
2 vector<int> dias;
```

### Uso

```
1 cin >> cantdias;
2 dias.resize(cantdias);
3 dias.assign(cantdias,0);
4 dias[2]= dias[1]+1;
```

# Estructuras

## Map

### Definición

```
1 #include <map>
2 map<string, int> wIndex;
3 map<char, int> mimap;
4 map<int, string>::iterator it; //lo uso para recorrer
```

### Uso

```
1 wIndex[word[i]] = i+1;
2 Empleados[5234] = "Juan P.";
3 cout << "Employees[3374]="<< Empleados[3374] << endl << endl;
4 cout << "Tamano map: " << Empleados.size() << endl;
5 for (it = Empleados.begin(); it!=Empleados.end(); it++) {
6     cout << it->first << " => " << it->second << '\n';
7     cout << (*it).first << " => " << (*it).second << '\n';
8 }
```

# Estructuras

## Set

### Definición

```
1 set<long long> visited;
```

### Uso

```
1 if ( visited.find(result) != visited.end() )  
2 for (set<int>::iterator casa = casas.begin();  
3      casa != casas.end(); casa++){  
4     casa.insert(val);}
```

# Estructuras

## Pair

### Definición

```
1 pair<int, int> verticeorigen;  
2 pair<string, int> nodo;
```

### Uso

```
1 typedef pair<int, int> caso;  
2 pair<int, int> caso;  
3 vector<caso> result;  
4 sort(result.begin(), result.end(), cmp);  
5 for (int i =0; i<cant; i++){  
6     cout << result[i].first << " " << result[i].second << endl;  
7 }
```

# Estructuras

## list

### Definición

Info: <http://www.cplusplus.com/reference/list/list/?kw=list>

```
1 #include <list>
2 list <int> resultados;
```

### Uso

```
1 resultados.push_back(totaldias);
2 while (!resultados.empty()) {
3     cout<< resultados.front()<< endl;
4     resultados.pop_front();
5 }
```

# STL

## Funciones útiles

### Definición

Info:

<http://www.cplusplus.com/reference/algorithm/>

```
1 #include <algorithm>
2 STL_FUNC(begin iterator, end iterator)
```

### Uso

```
1 max_element(arr, arr+n)
2 sort(arr, arr+n) //ordenar un arreglo
3 count(arr, arr+n, val) // devuelve el número de elementos en
   el rango [pri, ult)
4 lower_bound(arr, arr+n, val) // devuelve un iterador
   comenzando en el valor pasado - upper
```