std::string

```
string()
string(size_type cant, char ch)
string(const string& otra)
string(const string& otra, size_type pos)
string(const string& otra, size_type pos, size_type cant)
string(const char* s)
string(const char* s, size_type cant)
string(iterator primero, iterator ultimo)

string& operator=(const string& otra)
string& operator=(const char* s)
string& operator=(char ch)

string& operator+=(const string& otra)
string& operator+=(char ch)
string& operator+=(char* s)

char& operator[](size_type pos)
char& at(size_type pos)

const char* c_str()

bool empty()
size_type size()
void clear()

string& insert(size_type idx, size_type cant, char ch)
string& insert(size_type idx, const char* s)
string& insert(size_type idx, const char* s, size_type cant)
string& insert(size_type idx, string& str)
string& insert(size_type idx, string& str,
               size_type str_idx, size_type str_count)

string& erase()
string& erase(size_type idx)
string& erase(size_type idx, size_type cant)

string substr()
string substr(size_type pos)
string substr(size_type pos, size_type cant)

size_type find(const string& str)
size_type find(const string& str, size_type pos)
size_type find(char ch)
size_type find(char ch, size_type pos)
size_type find(char* ch)
size_type find(char* ch, size_type pos)
size_type find(char* ch, size_type pos, size_type cant )

size_type rfind( /* mismos que find */)

size_type find_first_of(const string&)
size_type find_first_of(const string&, size_type pos)
size_type find_first_of(const char ch)
size_type find_first_of(const char ch, size_type pos)
size_type find_first_of(const char* s)
size_type find_first_of(const char* s, size_type pos)
size_type find_first_of(const char* s, size_type pos,
                        size_type cant)

size_type find_first_not_of( /* mismos que find_first_of */ )
size_type find_last_of( /* mismos que find_first_of */ )
size_type find_last_not_of( /* mismos que find_first_of */ )
```

Común a todos los contenedores

```
contenedor()
contenedor(const contenedor& otro)

bool empty()
size_type size()

void swap(contenedor& otro)
```

Común a vector, list, set y map

```
contenedor(iterator primero, iterator ultimo)

iterador begin()
iterador end()
iterador_inverso rbegin()
iterador_inverso rend()

void clear()

iterator erase(iterator pos)
iterator erase(iterator primero, iterator ultimo)
```

Común a vector<T> y list<T>

```
contenedor(size_type cantidad, const T& valor)

T& front()
T& back()
T* data()

iterator insert(iterator pos, const T& valor)
iterator insert(iterator pos, size_type cantidad, T& valor)
iterator insert(iterator pos, iterator primero,
                iterator ultimo)

void push_back(const T& valor)
void pop_back()
```

Común a set<K> y map<K,T>

```
size_type erase(const K& clave)

size_type count(const K& clave)
iterator find(const K& clave)
iterator lower_bound(const K& clave)
iterator upper_bound(const K& clave)
```

std::vector<T>

```
T& operator[](size_type pos)
T& at(size_type pos)
```

std::list<T>

```
void remove(const T& valor)

void push_front(const T& valor)
void pop_front()

void reverse()
void unique()
void sort()
void sort(compare cmp)

void merge(list& otra)
void merge(list& otra, compare cmp)
```

```
void splice(const_iterator pos, list& otra)
void splice(const_iterator pos, list& otra,
            const_iterator it)
void splice(const_iterator pos, list& otra,
            const_iterator primero, const_iterator ultimo)
```

std::set<K>

```
std::pair<iterator, bool> insert(const K& valor)
iterator insert(iterator pista, K& valor)
void insert(iterator primero, iterator ultimo)
```

std::map<K,T>

```
typedef std::pair<K,T> PAR

T& operator[](const K& clave)

std:pair<iterator, bool> insert(const PAR& par)
iterador insert(iterador pista, const PAR& par)
void insert(iterador primero, iterador ultimo)
```

std::stack<T>

```
T& top()
void push(const T& valor)
void pop()
```

std::queue<T>

```
T& front()
T& back()
void push(const T& valor)
void pop()
```

std::fstream

```
fstream(const char* ruta, openmode modo)

bool is_open()
void open(const char* ruta, openmode modo)
void close()

/* openmode puede ser: app, binary, in, out, trunc y/o ate */

fstream& operator>>(short& valor)
fstream& operator>>(int& valor)
fstream& operator>>(long& valor)
fstream& operator>>(long long& valor)
fstream& operator>>(float& valor)
fstream& operator>>(double& valor)
fstream& operator>>(long double& valor)
fstream& operator>>(bool& valor)
fstream& operator>>(void* valor)

char peek()
char get()
fstream& unget()
fstream& putback(char ch)

fstream& get(char& ch)
fstream& get(char* s, std::streamsize cant)
fstream& get(char* s, std::streamsize cant, char delim)
fstream& getline(char* s, std::streamsize cant)
fstream& getline(char* s, std::streamsize cant, char delim)

fstream& ignore(std::streamsize cant, char delim)

fstream& operator<<(short valor)
fstream& operator<<(int valor)
fstream& operator<<(long valor)
fstream& operator<<(long long valor)
fstream& operator<<(float valor)
fstream& operator<<(double valor)
fstream& operator<<(long double valor)
fstream& operator<<(bool valor)
fstream& operator<<(const void* valor)

fstream& put(char ch)
fstream& write(const char* s, std::streamsize cant)

bool good()
bool eof()
bool fail()
bool bad()
bool operator!()
operator bool()
```