

SESION 7

CTIC

Cadenas en C++

Creando Objetos String

Cadena en C

Array de caracteres que terminan en el caracter nulo (**'\0'**).

Cadena en C++

- Objeto cuyo tipo string está definido en el archivo **<string>**
- Tiene un largo repertorio de funciones (ejm: length, replace, etc.)

```
char cs[ ] = "Napoleon"; // cadena en C  
string s = "Napoleon"; // cadena en C++
```

```
cout << s << " tiene " << s.length() << " caracteres.\n";  
s.replace(5, 2, "ia"); //cambia s to "Napolian"
```

Cadenas en C ++

- **Entrada con Formato:** Operator de extracción de flujo
 - `cin >> ObjetoString;`
 - el operador de extracción `>>` formatea los datos que recibe de su flujo de entrada; detiene la lectura al encontrar el espacio en blanco
- **Entrada sin Formato:** la función `getline` para un `string`
 - `getline(cin, s)`
 - no salta sobre un espacio en blanco
 - delimitado por un avance de línea
 - almacena una línea completa de caracteres en `s`

```
string s = "ABCDEFCFG";
```

```
getline(cin, s); //almacena una línea completa de caracteres en s
```

```
char c = s[2]; //asigna 'C' to c
```

```
s[4] = '*'; //cambia s to "ABCD*FG"
```

Cadenas en C ++

- No necesariamente terminan en **null**
- **string** no es un apuntador, sino una clase
- Muchas de sus funciones miembro toman la posición inicial y longitud
 - Si la longitud del argumento es muy grande, se elige max

Creando Objetos String

```
#include <string>  
//inicialización de string
```

incluir la biblioteca
<string>.

```
string s; //s contiene 0 caracteres  
string s1( "Hello" ); //s1 contiene 5 caracteres  
string s2 = "Hello"; //s2 contiene 5 caracteres  
//llama implícitamente al constructor
```

```
string s3( 8, 'x' ); //s3 contiene 8 caracteres 'x'  
string s4 = s3; //s4 contiene 8 caracteres 'x'
```

```
string s5(s2, 3, 2); //s5 copia una subcadena de s2; almacena "lo"
```

Objetos String

Las cadenas C++ pueden ser convertidas a cadenas en C.

```
string s = "ABCDEFGH";  
const char* cs = s.c_str();
```

Convierte **s** en una cadena C **cs**.

La función **c_str()** tiene un tipo de retorno **const char***

Objetos String

La clase C++ `string` también define una función `length()` para extraer cuantos caracteres están almacenados en un `string`.

```
cout << s.length() << endl;
```

Imprime 4 para el `string s == "Leon"`

Podemos usar el operador de *subindices* `[]` para acceder a caracteres individuales:

```
ejm    s[0] = 'N' ;    //el índice va de 0 a length-1
```

Objetos String

Las cadenas en C++ se pueden comparar usando operadores relacionales como tipos fundamentales :

```
If(s2 < s5)
```

```
    cout << "s2 lexicograficamente precede a s5 \n";
```

```
while(s4==s3) //...
```

'B' es lexicograficamente mayor que 'A'

Ejemplo de orden: 'A', "Arándano", "Banana", "Zanahoria", 'a',
"albaricoque", "lima"

Objetos String

Puede también concatenar strings C++ usando los operadores **+** y **+=** :

```
string s = "ABCD*FG";
```

```
string s2 = "Robot";
```

```
string s5 = "Soccer";
```

```
string s6 = s + "HIJK"; //cambia s6 a "ABCD*FGHIJK"
```

```
s2 += s5; //cambia s2 a "RobotSoccer"
```

Objetos String

Función Substring: substr()

```
s6 = "ABCD*FGHIJK";
```

```
s4 = s6.substr(5, 3); //cambia s4 a "FGH"
```

s4 obtiene una subcadena a partir de **s6**, que inicia en el índice **5** y extrae **3** caracteres

Objetos String

funciones `erase()` y `replace()` :

```
s6 = "ABCD*FGHIJK";
```

```
s6.erase(4, 2); //cambia s6 a "ABCDGHIJK";
```

```
s6.replace(5, 2, "xyz"); //cambia s6 a "ABCDGxyzJK";
```

reemplaza 2 caracteres de s6, que inician en el indice 5, con "xyz"

Objetos String

función find()

retorna el índice de la **primera ocurrencia** de una subcadena dada

```
string s7 = "Cuenca del río Mississippi"; //23 caracteres
cout << s7.find("ca") << endl; //imprime 4
cout << s7.find("so") << endl; //imprime 26, la longitud de la
cadena
```

Si la función find() **falla**, devuelve la longitud de la cadena que estaba buscando.

i.e. find() retorna **4,294,967,295**