



Semana -09 -Blender

Cadenas de Caracteres



Temario

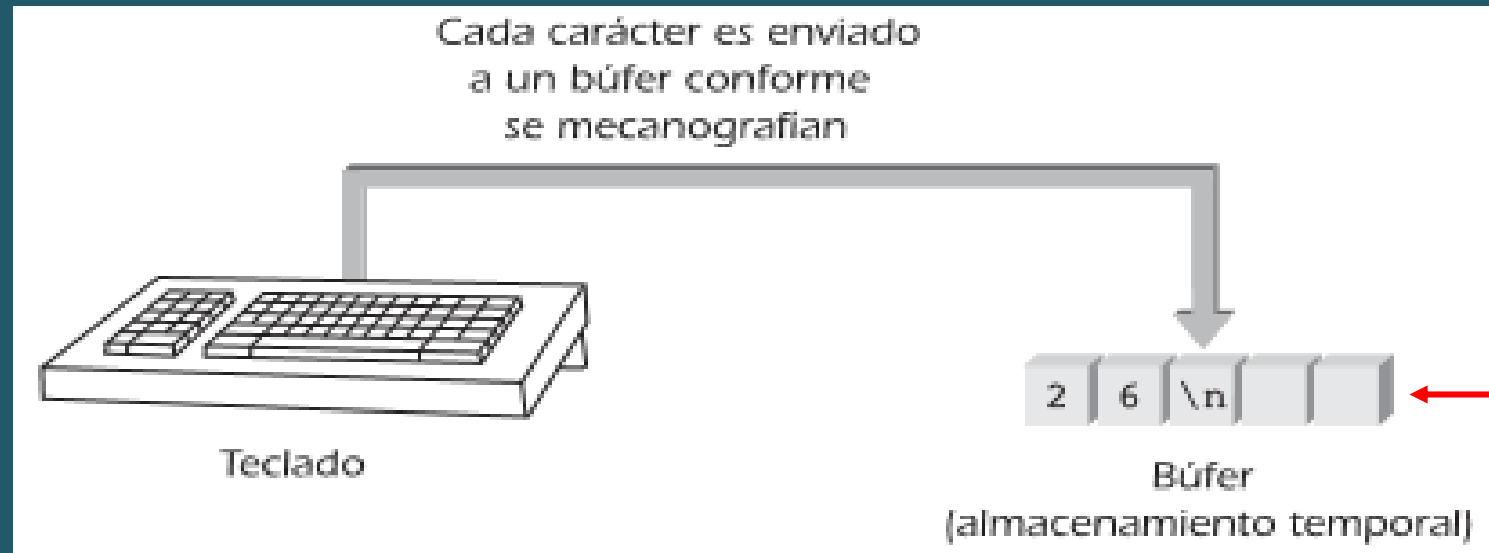


- Funciones adicionales para el manejo de cadenas

TIP IMPORTANTE



Recuerde lo que sucede cada vez que digitamos un carácter:



TIP IMPORTANTE



¡Usar **cin** y **getline** en 2 instrucciones diferentes puede traer problemas!

Ejemplo:

```
int numero; string mensaje;
```

```
cout<<"Ingrese un numero: ";  
cin>>numero; ←
```

Se asignará correctamente un valor en la variable entera *numero*

```
cout<<"Ingrese una cadena: ";  
getline(cin,mensaje); ←
```

Pero no se asignará ninguna cadena en el objeto *mensaje*.

¡Ejecute el programa y verá lo que sucede!



¿A qué se debe el problema mencionado en el ejemplo de la diapositiva anterior?

Se debe a que cuando se invoca a **cin** el carácter `\n` no es eliminado del buffer de entrada, por lo que afectará a un **getline** posterior.

¿Cómo se resuelve el problema anterior?

```
int numero; string mensaje;

cout<<"Ingrese un numero: ";
cin>>numero;

cin.ignore(); ←

cout<<"Ingrese una cadena: ";
getline(cin,mensaje);

cout<<"\n\nCadena leida: "<<mensaje;
```

Agregando esta instrucción:

Borra el contenido del buffer para poder hacer la siguiente lectura de la cadena

¡Ejecute el programa y notará que se arregló el problema!



Sigamos conociendo mas funciones para procesar cadenas



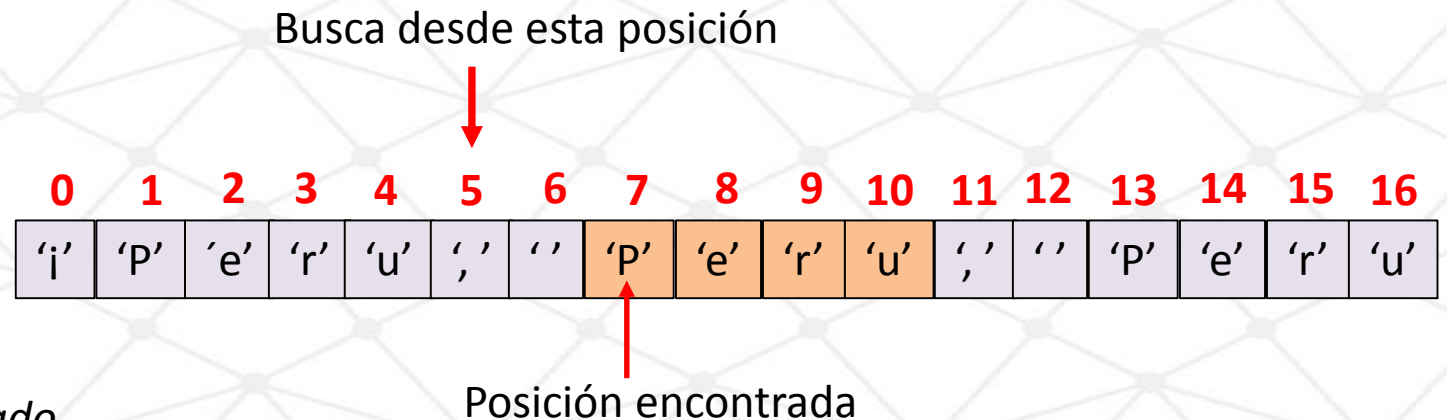
¿ Recuerda a la función **find()** ? También se puede usar de otra forma

Ejemplo:

```
string frase = "¡Peru, Peru, Peru!";  
int ind;  
ind = frase.find("Peru",5);
```

Busca desde la posición **5**, la primera ocurrencia de la cadena **"Peru"** en la cadena **frase**

Si la búsqueda prospera se retorna la **posición** desde donde fue encontrada la cadena. Por lo que a la variable **ind** se le asignará el valor **7**



Revise el programa en el block de notas adjuntado



Se puede eliminar parte de una cadena, usando la función **erase()**

Ejemplo 1:

```
string trama = "111100011";  
trama.erase(4,3);
```

De la cadena **trama** se eliminan **3** caracteres empezando desde la posición **4**

0	1	2	3	4	5	6	7	8
'1'	'1'	'1'	'1'	'0'	'0'	'0'	'1'	'1'

Cadena actualizada:
(su tamaño se reduce)

0	1	2	3	4	5
'1'	'1'	'1'	'1'	'1'	'1'

Revise el programa en el block de notas adjuntado

Ejemplo 2:



```
string trama = "110000";  
trama.erase(2);
```

De la cadena **trama** se eliminan todos los caracteres desde la posición **2**

0	1	2	3	4	5
'1'	'1'	'0'	'0'	'0'	'0'

Cadena actualizada:
(su tamaño se reduce)

0	1
'1'	'1'

Revise el programa en el block de notas adjuntado



Se puede reemplazar parte de una cadena con otra cadena, usando la función **replace()**

Ejemplo:

```
string cadena = "Clasifico Chile al repechaje";
cadena.replace(10,5,"Peru");
```

De la cadena original primero se eliminan 5 caracteres empezando desde la posición 10

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
'c'	'l'	'a'	's'	'i'	'f'	'i'	'c'	'o'	' '	'C'	'h'	'i'	'l'	'e'	' '	'a'	'l'	' '	'r'	'e'	'p'	'e'	'c'	'h'	'a'	'j'	'e'

Cadena después de la eliminación:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
'c'	'l'	'a'	's'	'i'	'f'	'i'	'c'	'o'	' '	' '	'a'	'l'	' '	'r'	'e'	'p'	'e'	'c'	'h'	'a'	'j'	'e'

A la cadena actualizada se inserta la cadena "Peru" desde la posición 10, obteniéndose el resultado final

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
'c'	'l'	'a'	's'	'i'	'f'	'i'	'c'	'o'	' '	'P'	'e'	'r'	'u'	' '	'a'	'l'	' '	'r'	'e'	'p'	'e'	'c'	'h'	'a'	'j'	'e'



Se pueden comparar dos cadenas utilizando los operadores relacionales :

`== != < > <= >=`

Ejemplo:

```
string clave;  
cout<<"Ingrese la clave: ";  
getline(cin,clave);  
  
if(clave == "03FT60A")  
    cout<<"\nClave correcta";  
else  
    cout<<"\nClave incorrecta";
```

Compara si el contenido de la cadena **clave** es igual a la cadena "03FT60A"

Revise el programa en el block de notas adjuntado



Métodos de conversión (de string a número): Uso de las funciones **stoi()** y **stof()**

Ejemplo:

```
string cadena1 = "345";  
string cadena2 = "3.1415";  
  
int    numero1;  
float  numero2;  
  
//se convierte cadena1 a número entero  
numero1 = stoi(cadena1);  
cout<<numero1 + 1;  //para verificar  
  
//se convierte cadena2 a numero float  
cout<<endl<<endl;  
numero2 = stof(cadena2);  
cout<<numero2 + 1;  //para verificar
```

Importante:

Tener configurado el **ISO C++ 11** para no tener problemas con el uso de estas funciones

Revise el programa en el block de notas adjuntado