

# ***Programación Orientada a Objetos***

## **Unidad 5: Clase String**

*Teoría 07 - 29/09/2011 - Pablo Novara*

## Operaciones con cadenas de caracteres en C (cstrings)

```
char nom_mejor[50];  
char alumnos[100][50];  
int notas[100], mejor=0, n=0;  
  
while (true) {  
    cin.getline(alumnos[n],50);  
    if (strlen(alumnos)==0) break;  
    cin>>notas[n];  
    if (notas[n]>mejor) {  
        mejor=notas[n];  
        strcpy(nom_mejor,alumnos[n]);  
    }  
    n++; cin.ignore();  
}
```

una cadena era en realidad  
un arreglo de caracteres

un vector de cadenas  
era una matriz

hay que fijar el tamaño,  
sobredimensionando

para operar con cstrings  
se utilizan funciones  
de la biblioteca <cstring>

## Operaciones con la clase string en C++

```
string nom_mejor;  
string alumnos[100];  
int notas[100], mejor=0, n=0;  
  
while (true) {  
    getline(cin, alumnos[n]);  
    if (alumnos.size()==0) break;  
    cin>>notas[n];  
    if (notas[n]>mejor) {  
        mejor=notas[n];  
        nom_mejor=alumnos[n];  
    }  
    n++; cin.ignore();  
}
```

una cadena es una  
instancia de la clase string

un vector de cadenas  
es un vector de objetos

no hay que fijar un  
tamaño, la clase administra  
la memoria dinámicamente

para operar con strings  
se utilizan funciones  
operadores sobrecargados  
y métodos

## Operaciones con la clase string en C++

```
char nom[50]="Thomas", ape[50]="Anderson";  
  
char *mail=new char[1+strlen(ape)+16+1];  
mail[0]=nom[0];  
strcpy(mail+1,ape);  
strcat(mail,"@fich.unl.edu.ar");  
  
cout<<mail; // muestra "TAnderson@fich.unl.edu.ar"  
  
string nom("Thomas"), ape("Anderson");  
  
string mail(1,nom[0]);  
mail+=ape+"@fich.unl.edu.ar";  
  
cout<<mail; // muestra "TAnderson@fich.unl.edu.ar"
```

## Lectura y escritura de objetos string

```
string palabra;  
cout<<"Ingrese una palabra: ";  
cin>>nombre;
```

```
string frase;  
cout<<"Ingrese una frase: ";  
getline(cin,frase);
```

```
cout<<palabra<<endl;  
cout<<frase<<endl;
```

## Inicialización de objetos string

```
string s1;  
cout<<s1; // muestra ""  
s1="Inventa un mañana...";  
cout<<s1; // muestra "Inventa un mañana..."
```

```
string s2("...que no sea ayer");  
cout<<s2; // muestra "...que no sea ayer"
```

```
string s3(s1,11,6);  
cout<<s3; // muestra "mañana"
```

```
string s4(5,'A');  
cout<<s4; // muestra "AAAAA"
```

## Conversión, caracteres y tamaño

```
string str;  
bool es_numero;  
do {  
    cin>>str;  
  
    es_numero=true;  
    for (int i=0;i<str.size(); i++)  
        if (str[i]<'0' || str[i]>'9')  
            es_numero=false;  
  
} while (!es_numero);  
int cant = atoi(str.c_str());
```

## Concatenación de strings

```
string s1="Estudiar", s2="suerte";  
string s3=s1+" trae "+s2;  
cout<<s3; // muestra "Estudiar trae suerte"
```

```
string s4="Estudiar";  
s4+=" trae ";  
s4+="suerte";  
cout<<s4; // muestra "Estudiar trae suerte"
```

```
string s5="Estudiar", s6="suerte";  
s5+=string(" trae ")+s2;  
cout<<s5; // muestra "Estudiar trae suerte"
```



## Comparación entre strings

```
string s1,s2;  
cout<<"Ingrese dos palabras: ";  
cin>>s1>>s2;  
if (s1==s2)  
    cout<<"Las palabras son iguales";  
else if (s1<s2)  
    cout<<s1<<" esta primero en el diccionario";  
else  
    cout<<s2<<" esta primero en el diccionario";
```

## Extracción de sub-strings

```
string s1="Al infinito y más allá!";
```

```
string s2=s1.substr(3,8);
```

```
cout<<s2; // muestra "infinito"
```

```
string s3=s1.substr(14);
```

```
cout<<s3; // muestra "más allá!"
```

## Modificación de un string

```
string s1="Es menester que sea rock!";  
s1.erase(3,17);  
cout<<s1; // muestra "Es rock!"
```

```
string s2="Vengo remando de larga distancia";  
s2.erase(13);  
cout<<s2; // muestra "Vengo remando"
```

```
string s3="Andarás bien por la 66"  
s3.clear();  
cout<<s3; // muestra ""
```

## Modificación de un string

```
string s1="Hola a todos, yo soy el león!";
```

```
s1.replace(5,7,"mundo");
```

```
cout<<s1; // muestra "Hola mundo, yo soy el león!"
```

```
string s2="Maderas de nogal...";
```

```
s2.insert(11,"viejo");
```

```
cout<<s2; // muestra "Maderas de viejo nogal..."
```

## Modificación de un string

```
string str="sIGue aL cONejO BLancO";  
str[0] = toupper(str[0]);  
for (int i=1;i<str.size();i++)  
    str[i] = tolower(str[i]);  
cout<<str; // muestra "Sigue al conejo blanco"
```

## Busquedas en un string

```
string s1="Ser el elegido es como estar enamorado";  
string s2;  
cout<<"Ingrese una palabra: ";  
cin>>s2;  
  
size_t p=s1.find(s2,0);  
  
if (p==string::npos)  
    cout<<"La palabra no esta en la frase";  
  
else  
    cout<<"La palabra comienza en la posición "<<p;
```

## Busquedas en un string

```
string s1="Ser el elegido es como estar enamorado";  
  
string s2;  
cout<<"Ingrese una silaba: ";  
cin>>s2;  
  
int cant=0;  
size_t p=s1.find(s2,0);  
  
while (p!=string::npos) {  
    cant++;  
    p=s1.find(s2,p+s2.size());  
}  
  
cout<<"La silaba está "<<cant<<" veces"<<endl;
```

## Busquedas en un string

```
string nya="Guillermo Hernán Puertas";  
int pos_ape = nya.rfind(' ',nya.size());  
string nom=nya.substr(0,pos_ape);  
cout<<nom<<endl; // muestra "Guillermo Hernán"  
  
string ape=nya.substr(pos_ape+1);  
cout<<ape<<endl; // muestra "Puertas"
```



## Busquedas en un string

Escriba un programa que lea una lista de N palabras de hasta 12 letras (validando la longitud) y muestre:

- a) La primera según el orden alfabetico
- b) Las que empiezan con la sílaba “mar”

## Busquedas en un string

Escriba una función llamada *jerigonza* que reciba una cadena de texto con una frase y retorne la frase en *jerigonza*.

Por ejemplo, si recibe “Saludos Terrícolas!” debe retornar “Sapalupudopos Teperripicopolapas!”.

Implemente un programa cliente para verificar su funcionamiento.