

# Programación I

## Cadenas de Caracteres en C

Grado en Ingeniería Informática

## Cadenas de Caracteres en C

### Declaración

```
char nombre_cadena [tamaño_máximo_cadena + 1];
```

### Inicialización

Mediante el operador de asignación y sin necesidad de completar todo el vector.

1. Dándole valor a cada elemento del vector:

```
char codigo[11]={ 'H', 'o', 'l', 'a', '\0' };
```

El carácter nulo ('\0') indica el final de la cadena.

2. Asignándole directamente una constante de cadena de caracteres:

```
char codigo[11] = "Hola";
```

3. Mediante un array indeterminado:

```
char codigo[] = "hola";
```

### Indexar una cadena

```
char cad[]="esto es una prueba"; // para acceder a cada carácter
for(int i=0; cad[i]!='\0'; i++)
    printf("\nCaracter %d es: %d\n", i, cad[i]);
```

# Cadenas de Caracteres en C

## Operaciones

**strcpy (cadenaDestino, cadenaOrigen)**

Copia la cadena contenida en `cadenaOrigen` en `cadenaDestino`

**strcat (cadenaDestino, cadenaOrigen)**

Concatena la `cadenaOrigen` al final de la cadena que está en `cadenaDestino`.

**strlen (cadena\_origen)**

Devuelve la longitud de `cadenaOrigen`.

**strcmp (cadena1, cadena2)**

Compara las cadenas contenidas en `cadena1` y `cadena2` y devuelve uno de los siguientes valores:

- 0 si las dos cadenas son iguales;
- un entero menor que 0 si `cadena1` es menor que `cadena2`;
- un entero mayor que 0 si `cadena1` es mayor que `cadena2`.

# Cadenas de Caracteres en C

## Salida

Mediante printf:

```
char cadena[] = "Hola";  
printf("\nLa cadena es : %s\n", cadena);
```

## Entrada

Mediante fgets:

```
char cadena[100];  
fgets(cadena, 99, stdin);
```

Carácter a carácter:

```
int i; char c;  
for ( i = 0; i < 99 && c = getchar() != '\n'; i++)  
    cad[i] = c;  
cad[i] = '\0';  
if (c != '\n')  
    while ((c = getchar()) != '\n');
```