Informática I

Introducción al Lenguaje C

Claudio J. Paz

28 de marzo de 2018

Un poco de historia...

En los 60's

DEC Programmable Data Processor (PDP-7)

Un poco de historia...

Fines de los 60's y principios de los 70's

- 1966 - BCPL - Martin Richards.
- 1969 - B - Ken Thompson con Dennis Ritchie.
- 1969-1973 - C - Dennis Ritchie.

Un poco de historia...

Fines de los 70's hasta fines de los 90's... y más

- 1978 – Lenguaje de Programación C – Brian Kernighan & Dennis Ritchie
- 1989 – *American National Standards Institute* – ANSI C – C89
- 1990 – *International Organization for Standardization* – ISO C – C90
- 1999 – ANSI adopta el estándar ISO para C – C99
- 2011 – ISO C – C11

Primer programa...?

```c
#include <stdio.h>

/* Programa para mostrar
algunas caracteristicas de C */

/* la ejecución comienza en main */
int main(void)
{
  printf("Hola, mundo!\n");

  return 0; /* finalización con éxito */
}
```

```c
#include <stdio.h>

/* Programa para mostrar
algunas caracteristicas de C */

/* la ejecución comienza en main */
int main(void)
{
  printf("Hola, mundo!\n");

  return 0; /* finalización con éxito */
}
```

Las líneas que comienzan con # se llaman *Directivas del preprocesador*

Otros Ej.:
`#define`, `#ifdef`, etc.

```c
#include <stdio.h>

/* Programa para mostrar
algunas caracteristicas de C */

/* la ejecución comienza en main */
int main(void)
{
  printf("Hola, mundo!\n");

  return 0; /* finalización con éxito */
}
```

```c
#include <stdio.h>

/* Programa para mostrar
algunas caracteristicas de C */

/* la ejecución comienza en main */
int main(void)
{
  printf("Hola, mundo!\n");

  return 0; /* finalización con éxito */
}
```

Comentarios entre
/* y */
Pueden ser de una línea completa, varias líneas o una porción

```c
#include <stdio.h>

/* Programa para mostrar
algunas caracteristicas de C */

/* la ejecución comienza en main */
int main(void)
{
  printf("Hola, mundo!\n");

  return 0; /* finalización con éxito */
}
```

```c
#include <stdio.h>

/* Programa para mostrar
algunas caracteristicas de C */

/* la ejecución comienza en main */
int main(void)
{
  printf("Hola, mundo!\n");

  return 0; /* finalización con éxito */
}
```

Todos los programas deben tener una función `main()`

```c
#include <stdio.h>

/* Programa para mostrar
algunas caracteristicas de C */

/* la ejecución comienza en main */
int main(void)
{
  printf("Hola, mundo!\n");

  return 0; /* finalización con éxito */
}
```

```c
#include <stdio.h>

/* Programa para mostrar
algunas caracteristicas de C */

/* la ejecución comienza en main */
int main(void)
{
  printf("Hola, mundo!\n");

  return 0; /* finalización con éxito */
}
```

Las llaves definen *Bloques*
En este caso el bloque es el *cuerpo de la función* `main()`

```c
#include <stdio.h>

/* Programa para mostrar
algunas caracteristicas de C */

/* la ejecución comienza en main */
int main(void)
{
  printf("Hola, mundo!\n");

  return 0; /* finalización con éxito */
}
```

```c
#include <stdio.h>

/* Programa para mostrar
algunas caracteristicas de C */

/* la ejecución comienza en main */
int main(void)
{
  printf("Hola, mundo!\n");

  return 0; /* finalización con éxito */
}
```

`printf()` es una función de la biblioteca estándar `stdio.h`

```c
#include <stdio.h>

/* Programa para mostrar
algunas caracteristicas de C */

/* la ejecución comienza en main */
int main(void)
{
  printf("Hola, mundo!\n");

  return 0; /* finalización con éxito */
}
```

`printf()` es una función de la biblioteca estándar `stdio.h`
Espera como *argumento* una *cadena de caracteres* para imprimir en pantalla.

```c
#include <stdio.h>

/* Programa para mostrar
algunas caracteristicas de C */

/* la ejecución comienza en main */
int main(void)
{
  printf("Hola, mundo!\n");

  return 0; /* finalización con éxito */
}
```

`printf()` es una función de la biblioteca estándar `stdio.h`
Espera como *argumento* una *cadena de caracteres* para imprimir en pantalla.
También es una *instrucción*. Todas las intrucciones deben terminar en un ;

```c
#include <stdio.h>

/* Programa para mostrar
algunas caracteristicas de C */

/* la ejecución comienza en main */
int main(void)
{
  printf("Hola, mundo!\n");

  return 0; /* finalización con éxito */
}
```

```c
#include <stdio.h>

/* Programa para mostrar
algunas caracteristicas de C */

/* la ejecución comienza en main */
int main(void)
{
  printf("Hola, mundo!\n");

  return 0; /* finalización con éxito */
}
```

La función `main()` debe terminar con un `return` para informar el estado de finalización

Biblioteca `stdio.h`

- `printf()`
- `scanf()`
- `...`

Secuencias de escape de la función `printf()`

Escape	Descripción
\n	Nueva línea. El cursor queda al principio de la siguiente línea.
\t	Tabulador horizontal. El cursor queda en la posición de Tab.
\v	Tabulador vertical. El cursor queda en la misma posición en la linea de abaj
\b	Retroceso. El cursor vuelve una posición en la misma línea.
\r	Retorno de carro. El cursor vuelve al principio de la misma línea.
\\	Diagonal invertida. Inserta una diagonal en una cadena.
\”	Comillas. Idem anterior.

```c
#include <stdio.h>

/* Sequencias de escape de printf() */

int main(void)
{
  printf("Hola, mundo!\n");
  printf("Hola,\tmundo!\n");
  printf("Hola,\vmundo!\n");
  printf("Hola,\bmundo!\n");
  printf("Hola,\rmundo!\n");
  printf("Hola,\\mundo!\n");
  printf("Hola,\"mundo!\"\n");

  return 0;
}
```

```c
#include <stdio.h>

/* Sequencias de escape de printf() */

int main(void)
{
  printf("Hola, mundo!\n");
  printf("Hola,\tmundo!\n");
  printf("Hola,\vmundo!\n");
  printf("Hola,\bmundo!\n");
  printf("Hola,\rmundo!\n");
  printf("Hola,\\mundo!\n");
  printf("Hola,\"mundo!\"\n");

  return 0;
}
```

```
Hola, mundo!
Hola,    mundo!
Hola,
     mundo!
Holamundo!
mundo!
Hola,\mundo!
Hola,"mundo!"
```

Especificadores de conversión de la función `printf()`

Especificadores	Descripción
%c	Caracter.
%d o %i	Entro decimal con signo.
%u	Entro decimal sin signo.
%f	Decimal de punto flotante.

```c
#include <stdio.h>

/* Programa para mostrar
especificadores de conversión de printf() */

int main(void)
{
  printf("Un caracter: %c\n", 'a');
  printf("Un caracter: %c\n", 97);
  printf("Un número decimal: %d\n", 'a');
  printf("Un número decimal: %d\n", 97);
  printf("Un número decimal con ceros: %05d\n", 97);

  return 0;
}
```

```c
#include <stdio.h>

/* Programa para mostrar
especificadores de conversión de printf() */

int main(void)
{
  printf("Un caracter: %c\n", 'a');
  printf("Un caracter: %c\n", 97);
  printf("Un número decimal: %d\n", 'a');
  printf("Un número decimal: %d\n", 97);
  printf("Un número decimal con ceros: %05d\n", 97);

  return 0;
}
```

```
Un caracter: a
Un caracter: a
Un número decimal: 97
Un número decimal: 97
Un número decimal con ceros: 00097
```

```c
#include <stdio.h>

/* Programa para mostrar
especificadores de conversión de printf() */

int main(void)
{

  printf("Un número flotante: %.2f\n", 3.14159);
  printf("Un número flotante: %9f\n", 3.14159);
  printf("Un número flotante: %09.4f\n", 3.14159);

  return 0;
}
```

```c
#include <stdio.h>

/* Programa para mostrar
especificadores de conversión de printf() */

int main(void)
{

  printf("Un número flotante: %.2f\n", 3.14159);
  printf("Un número flotante: %9f\n", 3.14159);
  printf("Un número flotante: %09.4f\n", 3.14159);

  return 0;
}
```

```
Un número flotante: 3.14
Un número flotante:  3.141590
Un número flotante: 0003.1416
```

```c
#include <stdio.h>

/* Programa para mostrar
el uso de scanf() */

int main(void)
{
  char c1;
  char c2;

  printf("Ingrese un caracter: ");
  scanf("%c", &c1);

  printf("Ingrese un segundo caracter: ");
  scanf("%c", &c2);
  printf("Usted ingresó \"%c\" y \"%c\"\n", c1, c2);

  return 0;
}
```

```c
#include <stdio.h>

/* Programa para mostrar
el uso de scanf() */

int main(void)
{
  char c1;
  char c2;

  printf("Ingrese un caracter: ");
  scanf("%c", &c1);

  printf("Ingrese un segundo caracter: ");
  scanf("%c", &c2);
  printf("Usted ingresó \"%c\" y \"%c\"\n", c1, c2);

  return 0;
}
```

```
Ingrese un caracter: a
Ingrese un segundo caracter: Usted ingresó "a" y "
"
```

```c
#include <stdio.h>

/* Programa para mostrar
el uso de scanf() */

int main(void)
{
  char c1;
  char c2;

  printf("Ingrese un caracter: ");
  scanf(" %c", &c1);

  printf("Ingrese un segundo caracter: ");
  scanf(" %c", &c2);
  printf("Usted ingresó \"%c\" y \"%c\"\n", c1, c2);

  return 0;
}
```

```c
#include <stdio.h>

/* Programa para mostrar
el uso de scanf() */

int main(void)
{
  char c1;
  char c2;

  printf("Ingrese un caracter: ");
  scanf(" %c", &c1);

  printf("Ingrese un segundo caracter: ");
  scanf(" %c", &c2);
  printf("Usted ingresó \"%c\" y \"%c\"\n", c1, c2);

  return 0;
}
```

```
Ingrese un caracter: a
Ingrese un segundo caracter: b
Usted ingresó "a" y "b"
```

```c
#include <stdio.h>

/* Programa para mostrar
el uso de scanf() */

int main(void)
{
  int n1;
  int n2;

  printf("Ingrese un número: ");
  scanf("%d", &n1);

  printf("Ingrese un segundo número: ");
  scanf("%d", &n2);

  printf("%d + %d = %d\n", n1, n2, n1+n2);

  return 0;
}
```

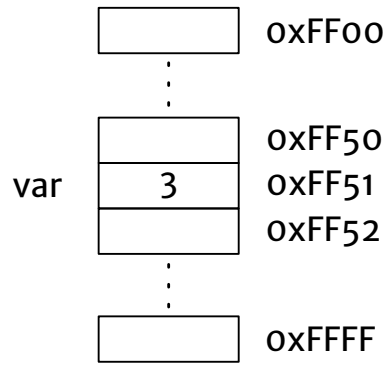
```c
#include <stdio.h>

/* Programa para mostrar
el uso de scanf() */

int main(void)
{
  int n1;
  int n2;

  printf("Ingrese un número: ");
  scanf("%d", &n1);

  printf("Ingrese un segundo número: ");
  scanf("%d", &n2);

  printf("%d + %d = %d\n", n1, n2, n1+n2);

  return 0;
}
```

```
Ingrese un número: 1
Ingrese un segundo número: 2
1 + 2 = 3
```

Variables

Todas las variables tienen

- Nombre o identificador
- Valor almacenado
- Dirección de memoria donde está amacenado ese valor
- Tipo



```c
#include <stdio.h>

int main(void)
{

  char caracter = 'a';
  int entero;
  float flotante = 3.14159;

  entero = 3;

  printf("Un caracter: %c\n", caracter);
  printf("Un número entero: %d\n", entero);
  printf("Un número flotante: %.2f\n", flotante);

  return 0;
}
```

Tipos de datos y sus tamaños

Tipo	Min.	Máx.	Bytes
char	-128	127	1
unsigned char	0	255	1
short int	-32768	32767	2
int	-32768	32767	2
unsigned int	0	65535	2
long	-2147483648	2147483647	4
float	3.4E-38	3.4E+38	4
double	1.7E-308	1.7E+308	8
long double	1.1E+4932	3.4E-4932	10

```c
#include <stdio.h>

int main(void)
{
  printf("Tamaños de tipos de datos\n");
  printf("char\t\t%d\n",sizeof(char));
  printf("short\t\t%d\n",sizeof(short));
  printf("int\t\t%d\n",sizeof(int));
  printf("float\t\t%d\n",sizeof(float));
  printf("double\t\t%d\n",sizeof(double));

  return 0;
}
```

```c
#include <stdio.h>

int main(void)
{
  printf("Tamaños de tipos de datos\n");
  printf("char\t\t%d\n",sizeof(char));
  printf("short\t\t%d\n",sizeof(short));
  printf("int\t\t%d\n",sizeof(int));
  printf("float\t\t%d\n",sizeof(float));
  printf("double\t\t%d\n",sizeof(double));

  return 0;
}
```

```
Tamaños de tipos de datos
char      1
short     2
int       4
float     4
double    8
```

Palabras reservadas

Palabras reservadas

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Palabras reservadas

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Palabras reservadas

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Palabras reservadas

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Palabras reservadas

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Palabras reservadas

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Palabras reservadas

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	volatile
do	if	static	while

Palabras reservadas

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	~~goto~~	sizeof	volatile
do	if	static	while

Consultas

claudiojpaz@gmail.com

Horario de Consulta: Miércoles 17:00-19:00hs
Of.5 Ed.Salcedo