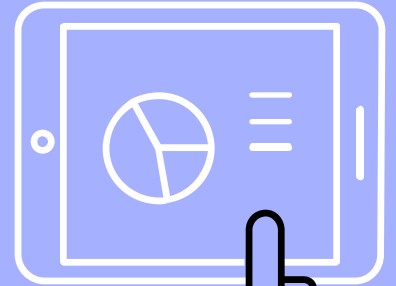
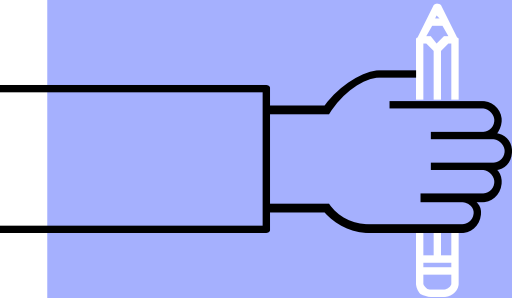
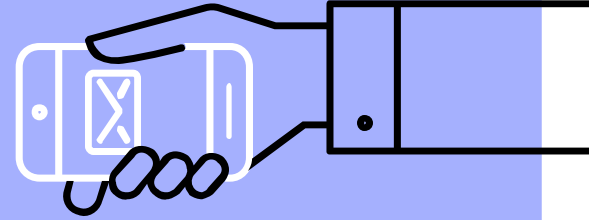
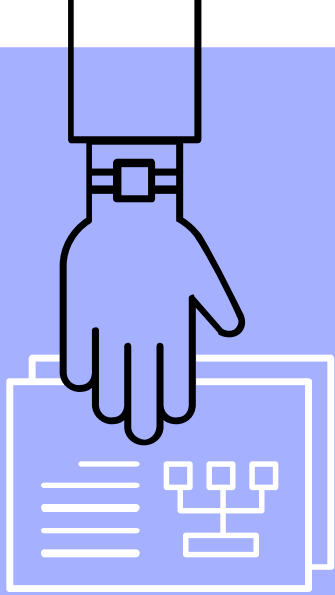


INTRODUCCIÓN AL LENGUAJE DE PROGRAMACIÓN C





Certifica a
JULIO HERNÁNDEZ

Por participar y aprobar el curso de

FUNDAMENTOS DE INGENIERÍA DE SOFTWARE

Christian Van Der Horst S
COO DE PLATZI



John Freddy Vega
CEO DE PLATZI

Certificación de aprobación online:

Aprobado el 22 de NOVIEMBRE de 2018

<https://platzi.com/@SunPixel>

Código: 1c924f5a-5402-487d-99f3-c29af66ab563



Certifica a
JULIO HERNÁNDEZ

Por participar y aprobar el

CURSO DE DESARROLLO WEB ONLINE

Christian Van Der Horst S
COO DE PLATZI



John Freddy Vega
CEO DE PLATZI

Certificación de aprobación online:

Aprobado el 28 de NOVIEMBRE de 2018

<https://platzi.com/@SunPixel>

Código: 44172583-ec40-4606-b2b6-253f72b06404

Programar es importante

Salida laboral importante. Además, su aprendizaje constituye una oportunidad al mejorar el razonamiento lógico formal.

NOMBRE DE LA EMPRESA: Banco de México
GIRO DE LA EMPRESA: Gubernamental / Bancario
PÁGINA WEB: www.banxico.org.mx
ÁREA: Dirección de Sistemas de Pagos
PUESTO: Analista de sistemas, especialidad en Requerimientos, Análisis y Diseño de Sistemas

REQUISITOS:

Escolaridad Mínima: Titulado o pasante en Ingeniería o licenciatura en informática, computación, sistemas o áreas afines
Promedio mínimo de 8.5
Cartilla del Servicio Militar liberada (Hombres)

CONOCIMIENTOS:

- Levantamiento de Requerimientos.
- Técnicas de modelado
- Conocimientos en lenguajes de programación C / C++, Java
- Sistemas operativos UNIX, Windows
- Análisis y diseño orientado a objetos usando UML
- Metodologías de desarrollo ágiles y RUP.
- Herramientas para análisis y diseño RATIONAL SOFTWARE QUEST
- Conocimiento de manejadores de Bases de datos
- Patrones de Diseño.
- Arquitecturas de Software: cliente-servidor, n capas, sistemas distribuidos
- Inglés técnico 80%

EXPERIENCIA:

Un año como analista programador y/o en diseño de sistemas

HABILIDADES:

- Buenas relaciones interpersonales
- Buena redacción
- Interés por trabajar en equipo
- Capacidad para trabajar bajo presión
- Interés por el orden y la calidad
- Inventiva / Originalidad
- Razonamiento analítico y sistémico
- Orientación al servicio
- Apertura al cambio

ACTIVIDADES PRINCIPALES A DESARROLLAR:

- Establecer y formalizar las expectativas del desarrollo o proyecto tomando en consideración las necesidades y características solicitadas por los usuarios finales y los interesados involucrados. Apoyar a las demás disciplinas en el entendimiento global de los requerimientos a través de la documentación generada
- Establecer y analizar los requerimientos del sistema con el usuario y formalizar el alcance del proyecto.
- Realizar el análisis de los requerimientos del sistema para proponer alternativas de arquitectura y diseño considerando la infraestructura con la que se cuenta.
- Diseñar la mejor solución posible a las necesidades planteadas por el usuario y/o área de requerimientos, que sirva como base para su implantación
- Integrar a los responsables de la Implementación del sistema para enriquecer el diseño
- Garantizar el cumplimiento de la normatividad de desarrollo que aplique al puesto, asegurándose de construir los documentos necesarios.

Levantamiento de Requerimientos.
Técnicas de modelado
Conocimientos en lenguajes de programación C / C++, Java
Sistemas operativos UNIX, Windows
Análisis y diseño orientado a objetos usando UML

Indentación | Orden y comentarios

```
int main (){  
    if(x!=6){  
        switch(x){  
  
            case 1: printf("Suma");  
            break;  
  
            case 2: printf("Resta");  
            break;  
  
            default: printf("Opcion incorrecta");  
            break;  
  
        }  
    }  
}
```



Indentación | Orden y comentarios

```
int main ()
{
    if(x!=6)
    {
        switch(x)
        {
            case 1: printf("Suma");
                    break;

            case 2: printf("Resta");
                    break;

            default: printf("Opcion incorrecta");
                    break;
        }
    }
}
```



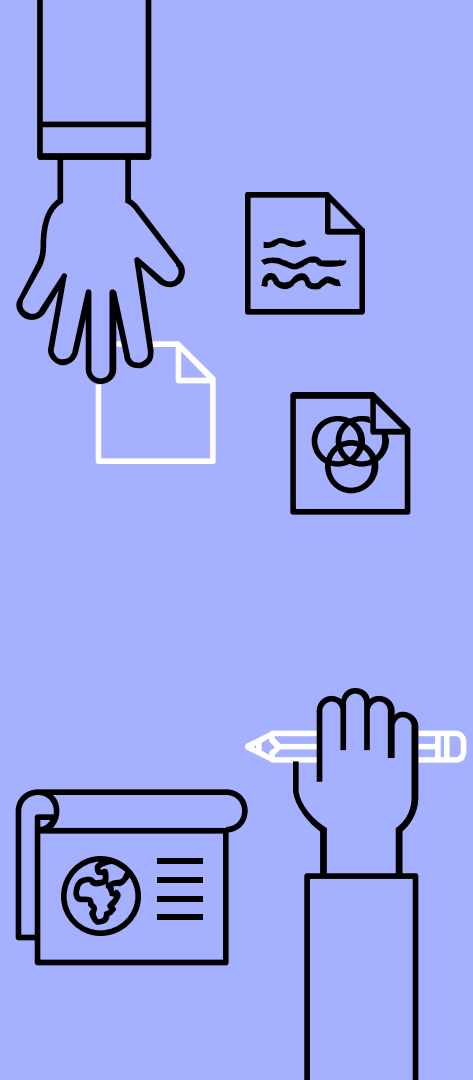
Lenguaje C | Características

También conocido como “Lenguaje de programación de sistemas” desarrollado en el año 1972

Para simplificar el funcionamiento de el lenguaje C tiene incluidas librerías de funciones.

```
2 |  
3 |  
4 | #include <stdio.h>  
5 | #include <iostream>  
6 |  
13 |  
14 | printf("");  
15 |
```

Más información en:
[https://www.ecured.cu/C_\(lenguaje_de_programaci%C3%B3n\)](https://www.ecured.cu/C_(lenguaje_de_programaci%C3%B3n))



Lenguaje C | Características

Incluye sentencias como if, else, for, do y while

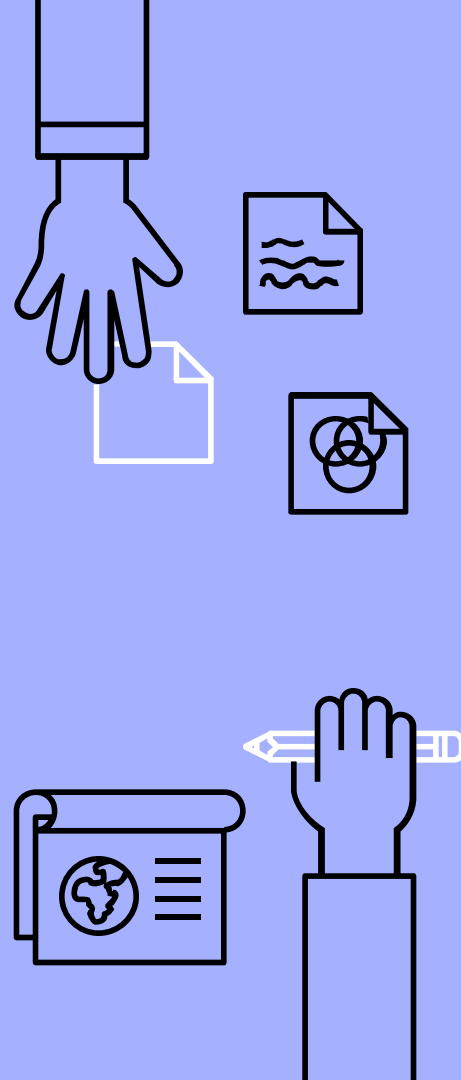
```
for (i=0; i<10; i++)  
{  
}
```

Cada instrucción que pasemos a poner en C va seguida de un punto y coma para decirle al compilador que hasta ahí llega la instrucción

```
13  
14     printf("");  
15
```

Más información en:

[https://www.ecured.cu/C_\(lenguaje_de_programaci%C3%B3n\)](https://www.ecured.cu/C_(lenguaje_de_programaci%C3%B3n))



En dónde programar

Dev c++ o codeblocks

Para windows

Xcode o un editor de textos y luego con la terminal

Para Mac

```
368  int lcd_create_  
369  {  
370      memset(emp  
371      int i = 0;  
372      int tmp;  
  
373  
374      tmp = per  
375      printf("p  
376      for(i = 7
```


Estructura general de un programa en C

Se definen las librerías que se utilizarán

(Encabezado)

```
3  
4 #include < >  
5  
6
```

Declaramos la función principal como entero

```
9  
10 int main ()  
11 {  
12
```

Al final de la función principal se regresa un valor 0, para finalizar el programa

```
14  
15 return 0;  
16  
17 }
```



Librería standard input output (Stdio.h)

printf

```
18  
19  
20     printf("Hola mundo");  
21  
22  
23     printf("Ingresa un numero ==> ");  
24  
25     printf("\t\t ***MENU*** \n");
```

scanf

```
29  
30     scanf("%d",&variable);  
31
```



Tipos de variables

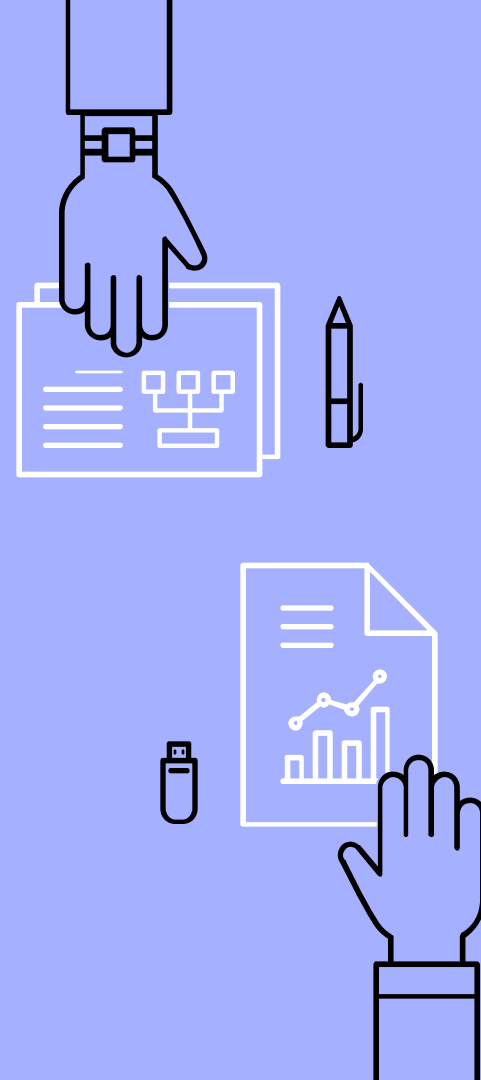
Números enteros

Declaración

```
28     int x;  
29  
30     int x= 10;  
31  
32     int x,y,z;  
33  
34     int x,y=10,z;  
35
```

Impresión y escaneo

```
37     printf(" %d ",x);  
38  
39     printf(" %d %d %d ",x,y,z);  
40  
scanf("%d",&x);
```



Tipos de variables

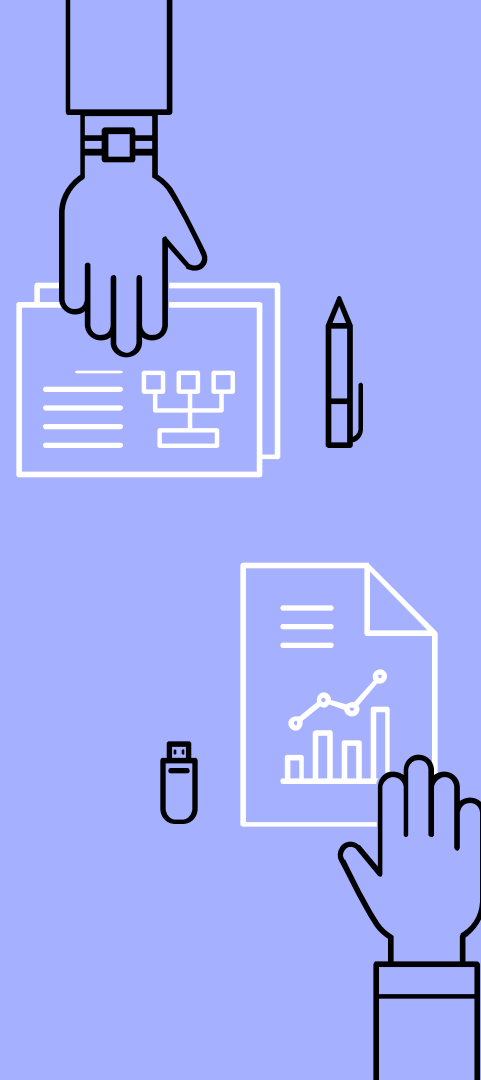
Números decimal

Declaración

```
float x;
```

Impresión y escaneo

```
printf(" %f ",x);  
printf(" %.2f ",x);  
scanf("%f",&x);
```



Tipos de variables

Caracteres

Declaración

```
char q;
```

```
char q='s';
```

Impresión y escaneo

```
printf(" %c ",q);
```

```
printf(" %c ",q);
```

```
scanf("%c",&q);
```



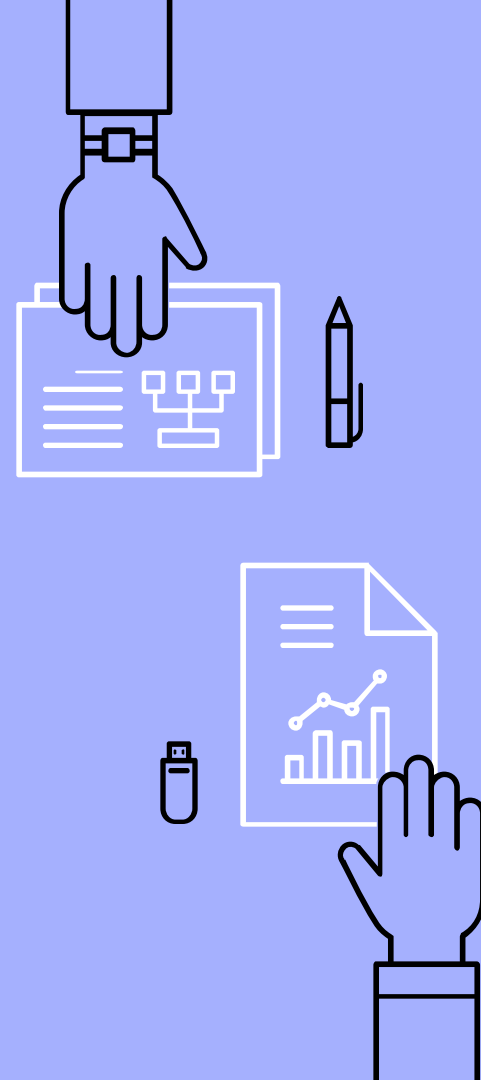
Variables con arreglos

```
x=10;
```

```
int x[5];
```

```
x[0]=1;  
x[1]=2;  
x[2]=3;  
x[3]=4;  
x[4]=5;
```

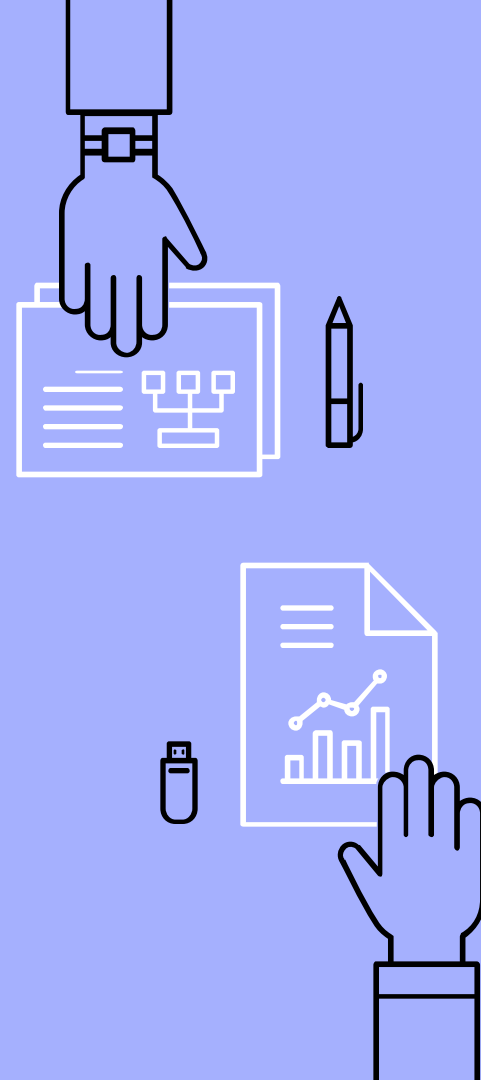
```
int x[5]  
for(i=0; i<5; i++){  
    printf("Ingresa un numero ==> ");  
    scanf("%d",&x[i]);  
}
```



Sentencias if

```
if(x==0){  
    printf("X es cero");  
}  
  
else if (x==1){  
    printf("X es uno");  
}  
  
else {  
    printf("X no es cero ni uno");  
}
```

```
if(x<0){}  
if(x<=0){}  
if(opcion=='s'){}  
if (x<0 && x<5){}  
if (x=0 || x=1){}
```

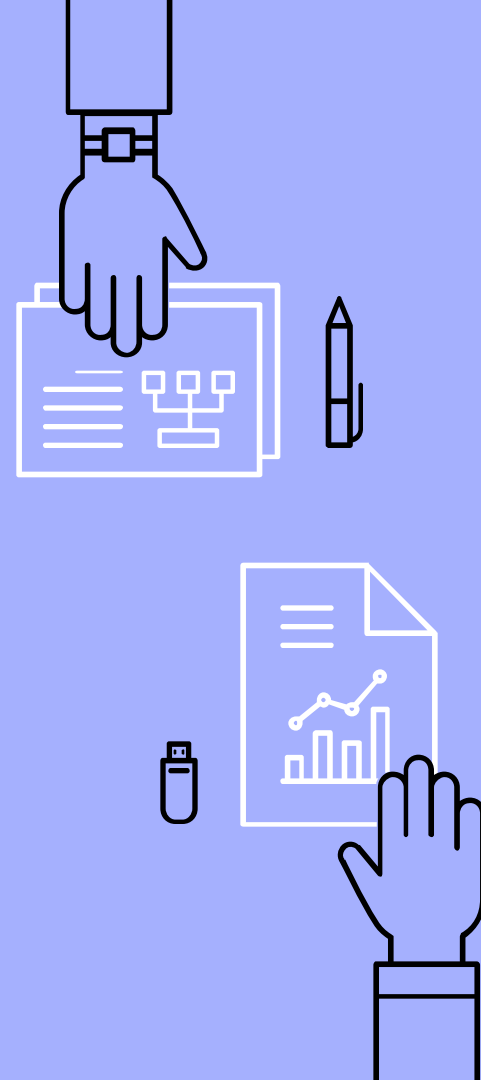


Sentencias for y while

```
for (int i=0; i<10; i++)  
{  
    printf("hola ");  
}
```

```
for (int i=0; i*2<10; i+=2)  
{  
    printf("hola ");  
}
```

```
x=0;  
while (x<10){  
    printf("hola ");  
    x++;  
}
```



Sentencias while y do while

```
x=11;  
  
do  
{  
  
printf("Hola");  
x+=1;  
  
} while (x<10)
```

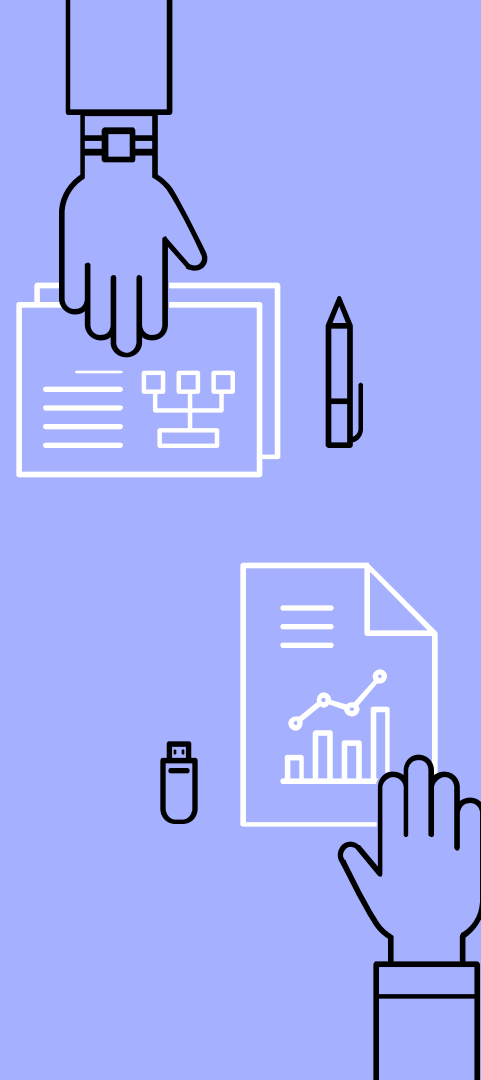
```
x=11;  
  
while (x<10)  
{  
  
printf("Hola");  
x+=1;  
  
}
```



Sentencia switch | Menú

```
Menu:  
1)Suma  
2)Resta  
Elige una opcion
```

```
int main(){  
    int x;  
  
    printf("Menu:");  
    printf("\n1)Suma");  
    printf("\n2)Resta");  
    printf("\nElige una opcion ");  
    scanf("%d",&x);  
  
    return 0;  
}
```



Sentencia switch | Menú

Menu

1)Suma
2)Resta

Elige una opcion==>

```
int main(){  
    int x;  
  
    printf("\t\t***Menu***\n\n\t1)Suma\n\t2)Resta\n\nElige una opcion==> ");  
    scanf("%d",&x);  
  
    return 0;  
}
```



Sentencia switch

```
switch(x){  
    case 1: printf("Suma");  
            break;  
    case 2: printf("Resta");  
            break;  
    default: printf("Opcion incorrecta");  
            break;  
}
```





“

*Medir los avances de
programación por líneas
de código es como medir
el progreso de la
construcción de aviones
por peso.*

