



| | | | | | |
|------------------------------|------------------------------------|-----------------|---|--------------------------------------|----------|
| NOMBRE DE LA PRÁCTICA | INTRODUCCIÓN AL LENGUAJE C. | | | NO. | 1 |
| ASIGNATURA: | MÉTODOS NUMÉRICOS. | CARRERA: | INGENIERÍA EN SISTEMAS COMPUTACIONALES | DURACIÓN DE LA PRÁCTICA (HRS) | |

NOMBRE DEL ALUMNO: MAGALI VALENCIA CLEMENTE
GRUPO:3401

I. Competencia(s) específica(s):

- Aplica los tipos de errores para identificar la incertidumbre y limitaciones de los cálculos numéricos en una computadora.

Encuadre con CACEI: Registra el (los) atributo(s) de egreso y los criterios de desempeño que se evaluarán en esta práctica.

| No. atributo | Atributos de egreso del PE que impactan en la asignatura |
|--------------|---|
| 1 | Reconocer la necesidad permanente de conocimiento adicional y tener la habilidad para localizar, evaluar, integrar y aplicar este conocimiento adecuadamente. |

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro):

- Aulas de clase

III. Material empleado:

- Computadora personal.
- SO Linux.
- Dev C++.

IV. Desarrollo de la práctica:

Ejercicio 1

En este primer programa codificamos nuestro primer “Hola Mundo” en C programa.



```
C hola.c ×
C hola.c > ...
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5
6      printf("Hola \n");
7
8      system("Pause");
9      return 0;
10
11 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL bash - magali

```
● magali@PC2:~$ ls
Arreglo.c  Ejemplo.c  Ejer20.c  Prog      ejercicio2.c  hola.out
Ejemplo    Ejer20     Ejer20.out  cuadrado.c  hola.c
● magali@PC2:~$ ./hola.out
Hola
sh: 1: Pause: not found
○ magali@PC2:~$
```

Ejercicio 2

En este programa codificamos nuestro segundo texto de bienvenida en C.



```

C hola.c  C ejercicio2.c  C Ejer20.c  C cuadrado.c  C Texto.c  X
Prog > C Texto.c > main()
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main(){
4      printf("este es un texto \n");
5      printf("que fue impreso desde C");
6      printf("\n");
7      printf("Ya soy todo un programador");
8      printf("\n");
9      system("Pause");
10     return 0;
11 }

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
bash - Prog
● magali@PC2:~/Prog$ ls
Evaluacion.c  Evaluacion.out  Prueba  Prueba.c  Prueba.out  Texto.c
● magali@PC2:~/Prog$ gcc Texto.c -o Texto.out
● magali@PC2:~/Prog$ ./Texto.out
este es un texto
que fue impreso desde C
Ya soy todo un programador
sh: 1: Pause: not found
```

Ejercicio 3

Crea un programa que imprima una bienvenida con tu nombre.



```
home > magali > Prog > C 3.c > main()
```

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main(){
4      printf("Bienvenida\n");
5      printf("Magali");
6      printf("\n");
7      printf("Valencia Clemente");
8      printf("\n");
9      system("Pause");
10     return 0;
11 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS

```
● magali@PC5:~/Prog$ ./3
Bienvenida
Magali
Valencia Clemente
sh: 1: Pause: not found
○ magali@PC5:~/Prog$
```

Ejercicio 4

En este programa codificamos nuestra primera suma decimal en C



```
C Suma.c x C Ejemplo.c
Prog > C Suma.c > main()
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      //Quiero imprimir el numero 3:
6      printf("%d \n", 3);
7      //Quiero imprimir la suma de 3 mas 4:
8      printf("%d \n", 3+4);
9
10     printf("La suma de %d + %d es = %d\n", 3,4,3+4);
11     system("Pause");
12     return 0;
13 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
bash - Prog
● magali@PC2:~/Prog$ ls
Evaluacion.c Evaluacion.out Prueba Prueba.c Prueba.out Suma.c Texto.c Texto.out
● magali@PC2:~/Prog$ gcc Suma.c -o Suma.out
● magali@PC2:~/Prog$ ./Suma.out
3
7
La suma de 3 + 4 es = 7
sh: 1: Pause: not found
○ magali@PC2:~/Prog$
```

Ejercicio 5

En este programa codificamos 2 operaciones aritméticas en C++, para lo cual aplicamos algunas reglas de sintaxis numérica para hacer posible el funcionamiento de dicho programa.



```
C OperacionesAritmeticas.c ×
Prog > C OperacionesAritmeticas.c > main()
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main(){
4      //Quiero imprimir el numero 5;
5      printf("%d \n", 5);
6      //Quiero imprimir el numero 8;
7      printf("%d \n", 8);
8      printf("La suma de %d + %d es = %d\n", 5,8,5+8);
9      //Quiero imprimir el numero 78787;
10     printf("%d \n", 78787);
11     //Quiero imprimir el numero 3259;
12     printf("%d \n", 3259);
13     printf("La suma de %d + %d es = %d\n",78787,3259,78787+3259);
14     system("Pause");
15     return 0;
16 }
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
● magali@PC2:~$ ls
Arreglo.c  Ejemplo.c  Ejer20.c  Prog      ejercicio2.c  hola.out
Ejemplo    Ejer20     Ejer20.out  cuadrado.c  hola.c
● magali@PC2:~$ cd Prog
● magali@PC2:~/Prog$ gcc OperacionesAritmeticas.c -o OperacionesAritmeticas.out
● magali@PC2:~/Prog$ ./OperacionesAritmeticas.out
5
8
La suma de 5 + 8 es = 13
78787
3259
La suma de 78787 + 3259 es = 82046
sh: 1: Pause: not found
○ magali@PC2:~/Prog$
```

Lamina 3. Variables

Ejercicio 6

En este programa codificamos dos variables aritméticas con un valor definido en C++ de tipo entero, para lo cual aplicamos algunas reglas de sintaxis numérica para hacer posible el funcionamiento de dicho programa.



```
C DecVariables.c •
Prog > C DecVariables.c > main()
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main(){
4      int unNumero = 3;
5      int otroNumero = 4;
6      //Quiero imprimir la suma de 3 mas 4
7      printf("El resultado de la suma es:");
8      printf("es:");
9      printf("%d", unNumero + otroNumero);
10
11     print("\n");
12     system("pause");
13     return 0;
14
15 }
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
magali@PC2:~$ ls
Arreglo.c  Ejemplo.c  Ejer20.c  Prog      ejercicio2.c  hola.out
Ejemplo    Ejer20     Ejer20.out  cuadrado.c  hola.c
magali@PC2:~$ cd Prog
magali@PC2:~/Prog$ gcc DecVariables.c -o DecVariables.out
magali@PC2:~/Prog$ ./DecVariables.out
El resultado de la suma es:es:7
sh: 1: Pause: not found
magali@PC2:~/Prog$
```

Ejercicio 7



```
home > magali > Prog > C 7.c > main()
1  #include <stdio.h>
2  int main () {
3  double r = 5;
4  double pi = 3.1416;
5  double area = pi * r * r;
6  printf("%f", area);
7  printf ("\n");
8
9  return 0;
10 }
```

Ejercicio 8

```
home > magali > Prog > C 8.c > main()
1  #include <stdio.h>
2  int main () {
3  double a = 3.1;
4  double A = 4.5;
5  printf("%f", a+A);
6  printf ("\n");
7
8  return 0;
9  }
10
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS

```
● magali@PC5:~/Prog$ ./8
7.600000
○ magali@PC5:~/Prog$
```

Ejercicio 9

En este programa codificamos nuestras variables aritméticas con un valor definido en C++ de tipo entero, para ello aplicamos algunas reglas de sintaxis numérica y de operaciones, para obtener los segundos de vida.



```
C Segundos.c X
Prog > C Segundos.c > main()
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      int min = 60;
6      int hora =60;
7      int dia =24;
8      int año =365*20;
9
10     printf("La cantidad de segundos que he vivido son:");
11
12     printf("%d",año*dia*min);
13     printf("\n");
14     system("Pause");
15     return 0;
16 }
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
● magali@PC2:~$ ls
Arreglo.c  Ejemplo.c  Ejer20.c  Prog      ejercicio2.c  hola.out
Ejemplo    Ejer20     Ejer20.out  cuadrado.c  hola.c
● magali@PC2:~$ cd Prog
● magali@PC2:~/Prog$ gcc Segundos.c -o Segundos.out
● magali@PC2:~/Prog$ ./Segundos.out
La cantidad de segundos que he vivido son:10512000
sh: 1: Pause: not found
○ magali@PC2:~/Prog$
```

Ejercicio 10

En este programa codificamos nuestras variables aritméticas con un valor definido en C++ de tipo entero, para lo cual aplicamos algunas reglas de sintaxis numérica y de operaciones, para obtener el resultado de nuestra multiplicación.



```
C Multiplicacion.c X
Prog > C Multiplicacion.c > main()
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      int V_uno=3;
6      int V_dos=4;
7      int V_tres=6;
8
9      printf("El resultado de la multiplicacion es:");
10     printf("%d", V_uno*V_dos*V_tres);
11     printf("\n");
12     system("Pause");
13     return 0;
14 }
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
● magali@PC2:~$ ls
Arreglo.c  Ejemplo.c  Ejer20.c  Prog      ejercicio2.c  hola.out
Ejemplo    Ejer20     Ejer20.out  cuadrado.c  hola.c
● magali@PC2:~$ cd Prog
● magali@PC2:~/Prog$ gcc Multiplicacion.c -o Multiplicacion.out
● magali@PC2:~/Prog$ ./Multiplicacion.out
El resultado de la multiplicacion es:72
sh: 1: Pause: not found
○ magali@PC2:~/Prog$
```

Ejercicio 11



```
home > magali > Prog > C 11.c > main()
1  #include <stdio.h>
2  int main () {
3      int una_variable;
4      una_variable = 3;
5      una_variable = una_variable + 5;
6      una_variable ++;
7
8      print("%d", una_variable + 5);
9
10     print("\n");
11
12     return 0;
13 }
```

Ejercicio 12

Crea un programa que calcule el área de un círculo que tiene 10 metros de diámetro.

```
home > magali > Prog > C 12.c > ...
1
2  #include <stdio.h>
3
4  int main(){
5      float pi = 3.1416;
6      int r=10;
7
8      printf("el area es: %f(%d*d) es de = %f\n",pi,r,pi*(r*r));
9      return 0;
10 }
```

Lamina 4. Operador_Incremento

Ejercicio 13

En este programa codificamos nuestras variables aritméticas con un valor definido en C++ de tipo entero, así como también algunos operadores aritméticos a usar.



```
C Multiplicacion.c  C Operadores.c X
Prog > C Operadores.c > main()
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      int x=5;
6      int y=10;
7      int z=++x*y--;
8
9      printf("x : %d\n", x);
10     printf("y : %d\n", y);
11     printf("z : %d\n", z);
12
13     system("Pause");
14     return 0;
15
16 }
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
● magali@PC2:~$ ls
Arreglo.c  Ejemplo.c  Ejer20.c  Prog      ejercicio2.c  hola.out
Ejemplo    Ejer20     Ejer20.out  cuadrado.c  hola.c
● magali@PC2:~$ cd Prog
● magali@PC2:~/Prog$ gcc Operadores.c -o Operadores.out
● magali@PC2:~/Prog$ ./Operadores.out
x : 6
y : 9
z : 60
sh: 1: Pause: not found
○ magali@PC2:~/Prog$
```

Lamina 6. Operadores lógicos y de relación

Ejercicio 14

En este programa codificamos nuestras variables aritméticas con un valor definido en C++ de tipo entero, así como también algunos operadores aritméticos a usar para resolver las tablas de verdad.



```
C OperadoresD.c X
Prog > C OperadoresD.c > main()
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      printf("*****AND*****");
6      printf("true && true : %d\n", (1&&1));
7      printf("true && false : %d\n", (1&&0));
8      printf("false && true : %d\n", (0&&1));
9      printf("false && false : %d\n", (0&&0));
10
11     printf("*****OR*****\n");
12     printf("true || true : %d\n", (1||1));
13     printf("true || false : %d\n", (1||0));
14     printf("false|| true : %d\n", (0||1));
15     printf("false|| false : %d\n", (0||0));
16
17     printf("*****XOR*****\n");
18     printf("true ^ true : %d\n", (1^1));
19     printf("true ^ false : %d\n", (1^0));
20     printf("false ^ true : %d\n", (0^1));
21     printf("false ^ false : %d\n", (0^0));
22     system("Pause");
23     return 0;
24
25 }
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
● magali@PC2:~$ ls
Arreglo.c Ejemplo Ejemplo.c Ejer20 Ejer20.c Ejer20.out Prog cuadrado.c ejercicio2.c hola.c hola.out
● magali@PC2:~$ cd Prog
● magali@PC2:~/Prog$ gcc OperadoresD.C -o OperadoresD.out
cc1plus: fatal error: OperadoresD.C: No such file or directory
compilation terminated.
● magali@PC2:~/Prog$ gcc OperadoresD.c -o OperadoresD.out
● magali@PC2:~/Prog$ ./OperadoresD.out
*****AND*****true && true : 1
true && false : 0
false && true : 0
false && false : 0
*****OR*****
true || true : 1
true || false : 0
false|| true : 0
false|| false : 0
*****XOR*****
true ^ true : 1
true ^ false : 0
false ^ true : 0
false ^ false : 0
sh: 1: Pause: not found
● magali@PC2:~/Prog$
```

Ejercicio 15



En este programa codificamos nuestras variables aritméticas con un valor definido en C++ de tipo entero, así como también algunos operadores aritméticos a usar para resolver las tablas de verdad.

```
C logicos.c •
Prog > C logicos.c > main()
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      int w =9;
6      int x =3;
7      int y =7;
8      int z =2;
9
10     printf("Enunciado 1: %d\n", ((x<y)&(w>z)));
11     printf("Enunciado 2: %d\n", ((x>=w)^(z=y)));
12     printf("Enunciado 3: %d\n", ((y<=x)^(x!=w)));
13     printf("Enunciado 4: %d\n", ((w==9)^(x==3)));
14     printf("Enunciado 5: %d\n", ((y>z)&(z=x)));
15     printf("Enunciado 6: %d\n", (! w !=9));
16
17     system("Pause");
18     return 0;
19 }
```

Ejercicio 16

```
home > magali > Prog > 16.cpp > ...
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      int x=5;
6      int y=10;
7      int z=++x*y--;
8
9      printf("x : %d\n", x);
10     printf("y : %d\n", y);
11     printf("z : %d\n", z);
12
13     system("Pause");
14     return 0;
15 }
16
```



Ejercicio 17

```
C Triangulo.c X
Prog > C Triangulo.c > main()
1  #include <stdio.h>
2  #include <math.h>
3  #include <stdlib.h>
4  #define PI 3.14159265
5
6  int main (){
7      double hip=100;
8      double ang=45;
9      double cat_op;
10     double cat_ad;
11     double AT;
12     double AC;
13     cat_op=sin(ang*PI/180)*100;
14     printf("Dimension del cateto opuesto: %3f \n", cat_op);
15     cat_ad=cos(ang*PI/180)*100;
16     printf("Dimension del cateto adyacente: %3f \n", cat_ad);
17     AT=((cat_op*cat_ad)/2);
18     printf("\n El area de un triangulo es: %3f \n", AT);
19     AC=(AT/2);
20     printf("\n El area del cuadrado es: %3f \n", AC);
21
22     system("Pause");
23 }
24

C Triangulo.c C Operaciones.c ●
Prog > C Operaciones.c > main()
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main (){
5      double a=1;
6      double b=3;
7      double c=5;
8      double d=30;
9      double e=23;
10
11     double z;
12
13     z=((a/b)+(b/c)+(a/d)/(e/d));
14     printf("Resultado: %f \n", z);
15
16     system("Pause");
17     return 0;
18
19 }
```



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
bash - Prog + - [] ... ^ X

magali@PC2:~$ ls
Arreglo.c Ejemplo Ejemplo.c Ejer20 Ejer20.c Ejer20.out Prog cuadrado.c ejercicio2.c hola.c hola.out
magali@PC2:~$ cd Prog
magali@PC2:~/Prog$ gcc Operaciones.c -o Operaciones.out
magali@PC2:~/Prog$ ./Operaciones.out
Resultado; 0.976812
sh: 1: Pause: not found
magali@PC2:~/Prog$
```

Ejercicio 18

```
home > magali > Prog > 18.cpp > main()
1
2 #include <stdio.h>
3 #include <stdlib.h>
4
5 int main(){
6     int antigüedad;
7     int sueldo;
8     int x;
9     printf("Años de antigüedad:\n");
10     scanf("%d", &antigüedad);
11
12     if (antigüedad >=5){
13         printf("Sueldo:\n");
14         scanf("%d", &sueldo);
15         x=(sueldo*100)/10;
16         if (x >=1000){
17             printf("Aceptado, Felicidades\n");
18         }
19     }else{
20         printf("No Aceptado\n");
21     }
22 }
23
24
```

Ejercicio 19

En este ejercicio se muestra como declarar una cadena.



```
C Cadena.c X
Prog > C Cadena.c > main()
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      char cad[10];
6      printf("Introduce una palabra \n");
7      gets (cad);
8      printf("\n %s",cad);
9
10     return 0;
11 }
```

Ejecución.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  COMMENTS

● magali@PC5:~/Prog$ ./Cadena
Introduce una palabra
Magali
○ Magali@magali@PC5:~/Prog$
```

Ejercicio 2. Cadenas

```
C Cadena.c  C cad2  X
Prog > C cad2
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      char cad[]="Es una cadena";
6      printf("%s \n", cad);
7
8      return 0;
9  }
```



Ejecución

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  COMMENTS

● magali@PC5:~/Prog$ gcc cad2.c -o cad
● magali@PC5:~/Prog$ ./cad
  Es una cadena
○ magali@PC5:~/Prog$
```

Ejercicio 21. Strlen

```
C strlen.c  X  C EjemploSt.c  ●
Prog > C strlen.c > main()
  2  #include <stdlib.h>
  3  # include <string.h>
  4  # include <stdlib.h>
  5
  6  int main(){
  7  int len;
  8  char cad[]="Es una cadena";
  9  len = strlen(cad);
 10  printf("La longitud de:'%s\' es:%d\n", cad, len);
 11
 12  return 0;
 13  }
```

Ejecución



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS

● magali@PC5:~/Prog$ ./st
La longitud de:'Es una cadena' es:13
○ magali@PC5:~/Prog$
```

Ejercicio 22. Escribe un programa que reciba una palabra por teclado

```
C EjemploSt.c C Palabrat.c X
Prog > C Palabrat.c > main()
1  #include <stdio.h>
2  #include<string.h>
3  #include <stdlib.h>
4
5  int main(){
6      int len;
7      char cad[10];
8      printf("Ingrese una palabra \n");
9      scanf("%s", cad);
10     len = strlen(cad);
11     for(int i = 1; i<= len; i++){
12         for( int b = 1; b <= len; b++){
13             printf("*\t");
14         }
15         printf("\n");
16     }
17     printf("\n");
18
19     return 0;
20 }
21
```

Ejecución



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  COMMENTS
● magali@PC5:~/Prog$ ./pal
Ingrese una palabra
magali
*      *      *      *      *      *
*      *      *      *      *      *
*      *      *      *      *      *
*      *      *      *      *      *
*      *      *      *      *      *
*      *      *      *      *      *

○ magali@PC5:~/Prog$
```

Ejercicio 23 Strcpy

```
C EjemploStc X
Prog > C EjemploStc > main()
1  #include <stdio.h>
2  #include<string.h>
3  #include <stdlib.h>
4
5  int main(){
6      int len;
7      char origen[]="Origen";
8      char destino[7];
9      strcpy(destino, origen);
10     printf("destino: %s", destino);
11
12     return 0;
13
14 }
```

Ejecución

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  COMMENTS
● magali@PC5:~/Prog$ ./eje
○ destino: Origenmagali@PC5:~/Prog$
```

Ejercicio 24

Escribe un programa que reciba por teclado dos palabra y cada una de ellas las almacene en un arreglo.



```
Prog > C Palabras.c > main()
1  # include <stdio.h>
2  # include <string.h>
3
4  int main(){
5
6      char palabra1[20];
7      char palabra2[20];
8      char temp[20];
9
10     printf("ingrese la primera palabra");
11     scanf("%s", palabra1);
12
13     printf("ingrese la segunda palabra");
14     scanf("%s", palabra2);
15
16     printf("Antes del intercambio:\n");
17     printf("palabra1: %s\n", palabra1);
18     printf("palabra2: %s\n", palabra2);
19
20     strcpy(temp, palabra1);
21     strcpy(palabra1,palabra2);
22     strcpy(palabra2,temp);
23
24     printf("Despues del intercambio: \n");
25     printf("palabra1: %s\n",palabra1);
26     printf("palabra2: %s\n",palabra2);
27
28     return 0;
29 }
```

Ejecución

```
Matriz.c      apuntadores4.c
● magali@PC5:~/Prog$ gcc Palabras.c -o pal
● magali@PC5:~/Prog$ ./pal
ingrese la primera palabramagali
ingrese la segunda palabravalencia
Antes del intercambio:
palabra1: magali
palabra2: valencia
Despues del intercambio:
palabra1: valencia
palabra2: magali
○ magali@PC5:~/Prog$
```

Ejercicio 25. Strcat.



```
C strat.c  X  C parln.c
Prog > C strat.c > main()
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  int main(){
6  int len;
7  char origen[]="brisas";
8  char destino[11] = "para";
9  strcat(destino, origen);
10 printf("destino: %s ", destino);
11 system("Pause");
12 return 0;
13 }
14
15
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL ...

```
● magali@PC5:~/Prog$ ./str
sh: 1: Pause: not found
○ destino: parabrisas magali@PC5:~/Prog$
```

Ejercicio 26. Escribe un programa que reciba por teclado dos palabras. Y concatene N veces la segunda palabra a la primera palabra. Donde N es la longitud de la primera palabra.



```
C EjercicioU.c ●
Prog > C EjercicioU.c > main()
1  #include <stdio.h>
2  #include <string.h>
3
4  int main(){
5      char palabra1[20];
6      char palabra2[20];
7
8      printf("ingrese la primera palabra:");
9      scanf("%s", palabra1);
10
11     printf("ingrese la segunda palabra:");
12     scanf("%s", palabra2);
13
14     int longitud= strlen(palabra1);
15
16     printf("la longitud de la primera palabra es: %d\n",longitud);
17     printf("la concatenacion de la primera palabra es %d veces a la primera palabra es:",
18     longitud);
19
20     for(int i=0; i<longitud; i++){
21         printf("%s",palabra2);
22     }
23     printf("\n");
24     return 0;
25 }
```

Ejecución

```
● magali@PC5:~/Prog$ ./eje
ingrese la primera palabra:magali
ingrese la segunda palabra:valencia
la longitud de la primera palabra es: 6
la concatenacion de la primera palabra es 6 veces a la primera palabra es:valenciavalenciavalenciavalencia
valencia
```

Ejercicio 27. Strcmp.



```
Prog > C strcmp.c > main()
1  # include <stdio.h>
2  # include <string.h>
3
4  int main(){
5  int res;
6  char str1[]="brisas";
7  char str2[] = "para";
8  res = strcmp( str1, str2 );
9  printf("resultado: %d ", res);
10
11 return 0;
12 }
13
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL ... bash - Prog + -

```
● magali@PC5:~/Prog$ ./str
○ magali@PC5:~/Prog$
```

Ejercicio 28. Palabra mayor o menor.



```
Prog > C mayor.c > main()
1  #include <stdio.h>
2  #include <string.h>
3
4  int main(){
5  int res;
6  char palabra1[10];
7  char palabra2[10];
8  int comparacion;
9
10 printf("ingrese la primera palabra: \n");
11 scanf("%s", palabra1);
12
13 printf("ingrese la segunda palabra: \n");
14 scanf("%s", palabra2);
15
16 comparacion = strcmp(palabra1,palabra2);
17
18 if(comparacion == 0){
19     printf("ambas palabras son iguales \n");
20
21 }else if(comparacion < 0){
22     printf("la palabra %s es mayor \n ", palabra1, palabra2 );
23
24 }else{
25     printf("la palabra es %s es menor \n", palabra1, palabra2 );
26 }
27
28 return 0;
29
30 }
```

Ejecución

Ejercicio 29. Palíndroma.



```
Prog > C Poli1.c > main()
1  #include <stdio.h>
2  #include <string.h>
3
4  #define MAX 100
5
6  int main()
7  {
8      char palabra[MAX];
9      int i, longitud, es_palindromo = 1;
10
11     printf("Ingresa una palabra: ");
12     scanf("%s", palabra);
13
14     longitud = strlen(palabra);
15
16     for( i = 0; i < longitud / 2; i++){
17         if(palabra[i] != palabra[longitud - i - 1]){
18             es_palindromo = 0;
19             break;
20         }
21     }
22     if(es_palindromo){
23         printf("La palabra es un palindromo.\n");
24     }else{
25         printf("la palabra no es un palindromo.\n");
26     }
27     return 0;
28
29 }
```

Ejecución.

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL COMMENTS bash - Prog + - [ ] [X]
EjercicioDos.c      Poli1
EjercicioDos.out    Poli1.c
EjercicioStrcmp.c   Polindroma
EjercicioU          Polindroma.c
EjercicioU.c        Posiciones.c
Estructuras1.c      Prueba
Evaluacion.c         Prueba.c
Evaluacion.out       Prueba.out
Fraccion.c           Segundos.c
● magali@PC5:~/Prog$ gcc Poli1.c -o pol
● magali@PC5:~/Prog$ ./pol
Ingresa una palabra: salas
La palabra es un palindromo.
```



Ejercicio 30.

```
Prog > C Contador.c > ...
1  #include <ctype.h>
2  #include <string.h>
3  #include <stdio.h>
4
5  #define MAX 100
6
7  int main()
8  {
9      char palabra[MAX];
10     int letras[26]={0}; //inicializa el arreglo en 0
11     int i, longitud;
12
13     printf("ingrese una palabra: ");
14     scanf("%s", palabra);
15
16     longitud = strlen(palabra);
17
18     for (i = 0; i < longitud; i++){
19         if(isalpha(palabra[i])){//verifica si el caracter es una letra
20             letras[tolower(palabra[i]) - 'a']++; //Aumenta el contador de
21         }
22     }
23
24     for (i = 0; i < 26; i++){
25         if (letras[i] > 0){
26             printf("La letra %c aparece %d veces.\n", 'a' + i, letras[i])
27         }
28     }
29
30     return 0;
31 }
32 }
```

Ejecución.

```
Fraccion.out      Suma.c      fac1.c
● magali@PC5:~/Prog$ gcc Contador.c -o Contador
● magali@PC5:~/Prog$ ./Contador
ingrese una palabra: magali
La letra a aparece 2 veces.
La letra g aparece 1 veces.
La letra i aparece 1 veces.
La letra l aparece 1 veces.
La letra m aparece 1 veces.
○ magali@PC5:~/Prog$
```



Ejercicio 31 Funciones

```
Prog > C funciones1.c > promedio(float, float)
1  #include <stdio.h>
2  #include <string.h>
3
4  float promedio (float a, float b);
5
6  int main(){
7      float a = 5, b=10, prom;
8      prom=promedio (a,b);
9      printf("El promedio es: %2.1f\n",prom);
10     return 0;
11 }
12
13 float promedio(float a, float b){
14     float prom;
15     a = a + 3;
16     b = b + 3;
17     prom=(a+b)/2;
18
19     return prom;
20 }
```

Ejecución.

```
PROBLEMS  OUTPUT  TERMINAL  ...
LongitudC.c  apuntadores3  strcat.c
LongitudC.out  apuntadores3.c  strcmp.c
Matriz  apuntadores4.c  strlen.c
Matriz.c  apuntadores5.c  tarea
Multiplicacion.c  apuntadores6.c  tarea.c
Multiplicacion.out  arq.c  tarea2
● magali@PC5:~/Prog$ gcc funciones1.c -o fu
● magali@PC5:~/Prog$ ./fu
El promedio es: 10.5
○ magali@PC5:~/Prog$
```

Ejercicio 32 Palabra Invertida.



```
Prog > C invertida.c > ...
1  # include <stdio.h>
2  # include<string.h>
3
4  int main(){
5      char cad[100];
6      char tem;
7      printf("Dijite una palabra: \n");
8      scanf("%s", cad);
9
10     int i, mitad, len;
11     len=strlen(cad);
12     mitad=len/2;
13     printf("Palabra original:%s\n",cad);
14     for(i=0;i<mitad;i++){
15         tem=cad[i];
16         cad[i]=cad[len-i-1];
17         cad[len-i-1]=tem;
18     }
19     printf("Palabra invertida:%s\n",cad);
20     return 0;
21 }
22 }
```

Ejecución

```
● magali@PC5:~/Prog$ gcc invertida.c -o inv
● magali@PC5:~/Prog$ ./inv
Dijite una palabra:
magali
Palabra original:magali
Palabra invertida:ilagam
○ magali@PC5:~/Prog$
```

Ejercicio 33

Escribir una función que tome como parámetros las longitudes de los tres lados de un triángulo (a, b, c) y devuelva el área del triángulo.



```
Prog > C Triangulo.c > main()
1  #include <string.h>
2  #include <stdio.h>
3  #include <math.h>
4
5
6  int main(){
7      float a, b, c, p, area;
8      printf("Ingresa la longitud del lado a: ");
9      scanf("%f", &a);
10     printf("Ingresa la longitud del lado b: ");
11     scanf("%f", &b);
12     printf("Ingresa la longitud del lado c: ");
13     scanf("%f", &c);
14
15     p = (a + b + c) / 2;
16     area = sqrt(p * (p - a) * (p - b) * (p - c));
17
18     printf("El area del triangulo es: %f\n", area);
19     return 0;
20
21 }
```

Ejecución.

```
14: command not found
● magali@PC5:~/Prog$ gcc Triangulo.c -o Triangulo -lm
● magali@PC5:~/Prog$ ./Triangulo
Ingresa la longitud del lado a: 10
Ingresa la longitud del lado b: 20
Ingresa la longitud del lado c: 15
El area del triangulo es: 72.618439
● magali@PC5:~/Prog$
```

Ejercicio 34

Calcular la potencia de forma recursiva



```
Prog > C recursiva.c > main(void)
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4
5  int main(void){
6  int x,y int max ;
7  x = 2;
8  y = 3;
9  max = potencia(x,y);
10 printf("La potencia es: %d ", max);
11
12 return 0;
13 }
14 int potencia (int a, int b){
15 if (b < 1)
16 return 1;
17 return a * potencia (a, b-1);
18 }
```

Ejercicio 35. Factorial

```
Prog > C factorial.c > main(void)
1  #include <stdio.h>
2
3  unsigned long long factorial(unsigned long long numero) {
4      // Si hemos llegado a 1, detenemos la recursión
5      if (numero <= 1)
6          return 1;
7      return numero * factorial(numero - 1); // Restar 1
8  }
9
10 int main(void) {
11     unsigned long long numero = 5;
12     unsigned long long factorialDelNumero = factorial(numero);
13     printf("El factorial de %llu es %llu", numero, factorialDe
14 }
```

Ejecución.



```
Palabrat.c          reloj1.c
Poli1               reloj1.out
Poli1.c            st
Polindroma          str
Polindroma.c        strcad.c
Posiciones.c        strcad.out
Prueba             strcat.c
Prueba.c           strcmp.c
Prueba.out         strlen.c
Segundos.c         tarea
Segundos.out       tarea.c
Suma.c             tarea2
Suma.out           triangulo
● magali@PC5:~/Prog$ gcc factorial.c -o fac
● magali@PC5:~/Prog$ ./fac
○ magali@PC5:~/Prog$
```

Ejercicio 36

```
Prog > C HorasT.c > main()
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      int HST;
6      int PH;
7      int SS;
8      int SSE;
9
10     printf("Horas semanales trabajadas:\n");
11     scanf("%d", &HST);
12     printf("Pago por hora:\n");
13     scanf("%d", &PH);
14     SS=(HST * PH);
15     if(HST <=40){
16         printf("El sueldo semanal es: $ %d \n", SS);
17     }else if (HST >40){
18         SSE=(1.5*(HST-40)* PH);
19         printf("El sueldo extra es: $ %d \n", SSE);
20         SS=(40 * PH + (PH * 1.5 *(HST-40)));
21         printf("Sueldo total con extra: $ %d \n", SS);
22     }
23
24     return 0;
25 }
```

Ejecución



```
El sueldo semanal es: $ 400
● magali@PC5:~/Prog$ ./HorasT
Horas semanales trabajadas:
30
Pago por hora:
200
El sueldo semanal es: $ 6000
○ magali@PC5:~/Prog$
```

Ejercicio 38.

```
Prog > C apuntadores2.c > ...
1  #include <stdio.h>
2
3  int*p, y;
4
5  void func()
6  {
7      int x=40;
8      p=&x;
9      y=*p;//correcto
10     *p=23;//correcto
11 }
12 int main (void)
13 {
14     func();
15     y=*p;//incorrecto
16     *p=25;//incorrecto
17     printf("El valor de y es %d \nEl valor de *p es %d\nEl
18     valor de p es %p",y,*p,p);
19 }
```

Ejecución.

```
OptLogicos.c          cad2
● magali@PC5:~/Prog$ gcc apuntadores2.c -o ap
● magali@PC5:~/Prog$ ./ap
El valor de y es 23
El valor de *p es 25
○ El valor de p es 0x7ffe9183acf4magali@PC5:~/Prog$
```

Ejercicio 39. Apuntadores, paso de parámetros de referencia.



```
Prog > C apunadores3.c > ...
1  #include <stdio.h>
2
3  int main(void)
4  {
5      int x=2;
6      int y=5;
7      printf("Antes x=%d, y=%d\n",x,y);
8      intercambio(&x,&y);
9      printf("Despues x=%d, y=%d\n",x,y);
10 }
11 void intercambio(int*a,int*b){
12     int temp;
13     temp=*b;
14     *b=*a;
15     *a=temp;
16 }
```

Ejercicio 40. Apunadores

```
Prog > C apunadores4.c > ...
1  #include <stdio.h>
2
3  int main(){
4      char cadena[10];
5      printf("Un int ocupa %d bytes", sizeof(int));
6      printf("Un char ocupa %d bytes", sizeof(char));
7      printf("Un float ocupa %d bytes", sizeof(float));
8      printf("Un double ocupa %d bytes", sizeof(double));
9      printf("Cadena ocupa %d bytes", sizeof(cadena));
10 }
11
12
```

Ejercicio 41. Apunadores 5.



```
Prog > C apunadores5.c > main(void)
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(void)
5  {
6      int i,n;
7      char *buffer;
8
9      printf("Teclea la longitud de la cadena");
10     scanf("%d",&i);
11
12     buffer=(char*)malloc((i+1)*sizeof(char));
13     if(buffer==NULL)exit(1);
14
15     for(n=0; n<i; n++)
16         buffer[n]=rand()%26+'a';
17     buffer[i]='\0';
18
19     printf("Random string: %s\n",
20     buffer);
21     free(buffer);
22
23 }
```

Ejecución.

| PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | COMMENTS |
|--|--------|---------------|----------|-----------|
| Multiplicacion.out | | arreglo.c | | tarea2 |
| OpLogicos.c | | arreglo1.c | | triangulo |
| Operaciones.c | | cad | | |
| ● magali@PC5:~/Prog\$ gcc apunadores5.c -o apu | | | | |
| ● magali@PC5:~/Prog\$./apu | | | | |
| Teclea la longitud de la cadena5 | | | | |
| Random string: nwlrb | | | | |
| ○ magali@PC5:~/Prog\$ | | | | |

Ejercicio 42. Crea un arreglo entero de tamaño x, en donde x es ingresado por teclado. Llena todos los elementos del arreglo con datos ingresados por el usuario. Muestra los valore.



```
Prog > C apunadores6.c > ...
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      int x;
6      printf("Ingrese el tamaño del arreglo: ");
7      scanf("%d", &x);
8
9      int arr[x];
10     printf("Ingrese %d numeros para llenar el arreglo:\n",x);
11     for(int i=0; i<x; i++){
12         scanf("%d",&arr[i]);
13     }
14     printf("Los vlores ingresados son: ");
15     for(int i=0; i<x; i++){
16         printf("%d",arr[i]);
17     }
18     return 0;
19 }
```

Ejecución.

| PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | COMMENTS |
|--|------------|---------------|----------|----------|
| Multiplicacion.c | arqu.c | tarea.c | | |
| Multiplicacion.out | arreglo.c | tarea2 | | |
| OpLogicos.c | arreglo1.c | triangulo | | |
| Operaciones.c | cad | | | |
| ● magali@PC5:~/Prog\$ gcc apunadores6.c -o apu | | | | |
| ● magali@PC5:~/Prog\$./apu | | | | |
| Ingrese el tamaño del arreglo: 6 | | | | |
| Ingrese 6 numeros para llenar el arreglo: | | | | |
| 10 | | | | |
| 6 | | | | |
| 7 | | | | |
| 9 | | | | |
| 6 | | | | |
| 4 | | | | |
| ● magali@PC5:~/Prog\$ | | | | |

Ejercicio 43. Crea un arreglo de tipo char de tamaño x, en donde x es ingresado por teclado. Llena elemento por elemento del arreglo con letras ingresados por el usuario. Muestra el arreglo impreso en forma inversa. Todo debe ser manejado con apuntadore

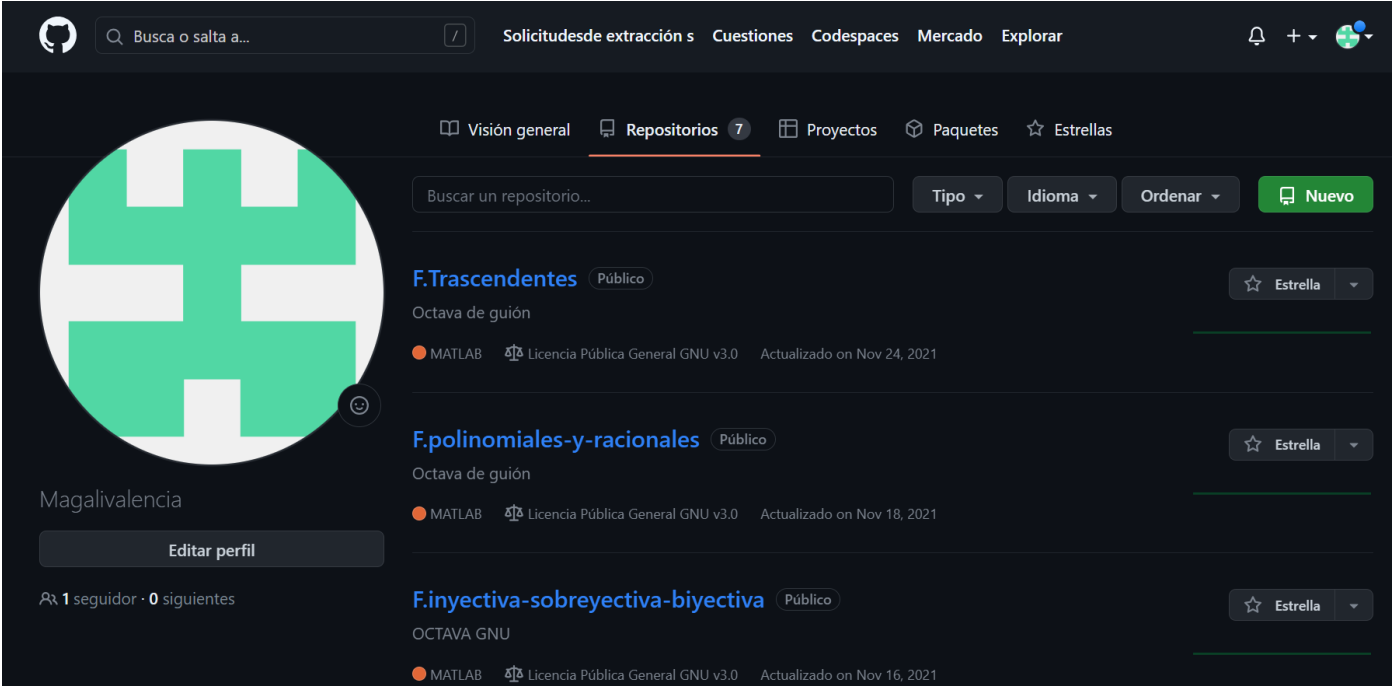


```
Prog > C apun7.c > ...
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5
6      int x,i;
7      int*arr;
8
9      printf("ingrese el tamaño de arreglo: ");
10     scanf("%d",&x);
11
12     arr=(int*)malloc(x* sizeof(int));
13
14     for(i=0; i<x; i++){
15         printf("Ingrese el valor para el elemento %d: ",i);
16         scanf("%d",&arr[i]);
17     }
18
19     printf("los valores ingresados son\n");
20     for(i=0; i<x; i++){
21         printf("%d: ",arr[i]);
22     }
23     free(arr);
24     return 0;
25 }
```

| PROBLEMS | OUTPUT | DEBUG CONSOLE | TERMINAL | COMMENTS |
|----------|--------|---------------|--|----------|
| | | | Ingrese el valor para el elemento 0: 4 Ingrese el valor para el elemento 1: 5 Ingrese el valor para el elemento 2: 8 los valores ingresados son | |
| | | | ● magali@PC5:~/Prog\$./apu ingrese el tamaño de arreglo: 3 Ingrese el valor para el elemento 0: 2 Ingrese el valor para el elemento 1: 6 Ingrese el valor para el elemento 2: 8 los valores ingresados son | |
| | | | ○ 2: 6: 8: magali@PC5:~/Prog\$ | |



Para subir los códigos a Github, lo primero que tenemos que realizar es crear una cuenta en Github, una vez que creamos nuestra cuenta, tendremos que crear un nuevo repositorio y pondremos el nombre que deseemos.



The screenshot shows a GitHub profile for a user named 'Magalivalencia'. The profile has a green and white cross-like avatar and is followed by 1 person. The 'Repositorios' (Repositories) tab is selected, showing 7 repositories. Three repositories are visible:

- F.Trascendentes** (Público): Octava de guión, MATLAB, Licencia Pública General GNU v3.0, Actualizado on Nov 24, 2021.
- F.polinomiales-y-rationales** (Público): Octava de guión, MATLAB, Licencia Pública General GNU v3.0, Actualizado on Nov 18, 2021.
- F.inyectiva-sobreyectiva-biyectiva** (Público): OCTAVA GNU, MATLAB, Licencia Pública General GNU v3.0, Actualizado on Nov 16, 2021.

Crear un nuevo repositorio

Un repositorio contiene todos los archivos de proyecto, incluido el historial de revisiones. ¿Ya tiene un repositorio de proyectos en otro lugar? [Importar un repositorio](#).

Dueño *



Magalivalencia ▾



Nombre del repositorio *

Los grandes nombres de repositorios son cortos y memorables. ¿Necesitas inspiración? ¿Qué tal **jubilant-octo-dollop**?

Descripción (opcional)



Público

Cualquier persona en Internet puede ver este repositorio. Tú eliges quién puede comprometerse.



Privado

Usted elige quién puede ver y confirmar este repositorio.

Inicialice este repositorio con:



Agregar un archivo README

Aquí es donde puede escribir una descripción larga para su proyecto. [Obtén más información sobre los archivos README](#).


Agregar .gitignore

Plantilla .gitignore: Ninguno ▾

Para crear el repositorio tenemos que seleccionar si lo queremos público o privado, en este caso es público para que puedan acceder.

Después de crearlo nos vamos a ajustes de developer y crearemos un nuevo token de acceso, como se muestra en la imagen.





Solicitudesde extracción sCuestionesCodespacesMercadoExplorar

Configuración / Configuración del desarrollador

Aplicaciones de GitHubAplicaciones OAuthTokens de acceso personalFichas de grano finoTokens (clásico)


Tokens de acceso personal (clásico)

Generar nuevo token

¿Necesita un token de API para scripts o pruebas? [Genera un token de acceso personal](#) para acceder rápidamente a la [API de GitHub](#).

Los tokens de acceso personal (clásicos) funcionan como tokens de acceso OAuth ordinarios. Ellos se puede usar en lugar de una contraseña para Git sobre HTTPS, o se puede usar para autenticarse en la API a través de la autenticación básica.

© 2023 GitHub, Inc. [Letra chica](#) [Privacidad](#) [Seguridad](#) [Estado](#) [Docs](#) [Póngase en contacto con GitHub](#) [Precios](#) [API](#) [Adiestramiento](#) [Blog](#) [Acerca de](#)



Confirmar acceso

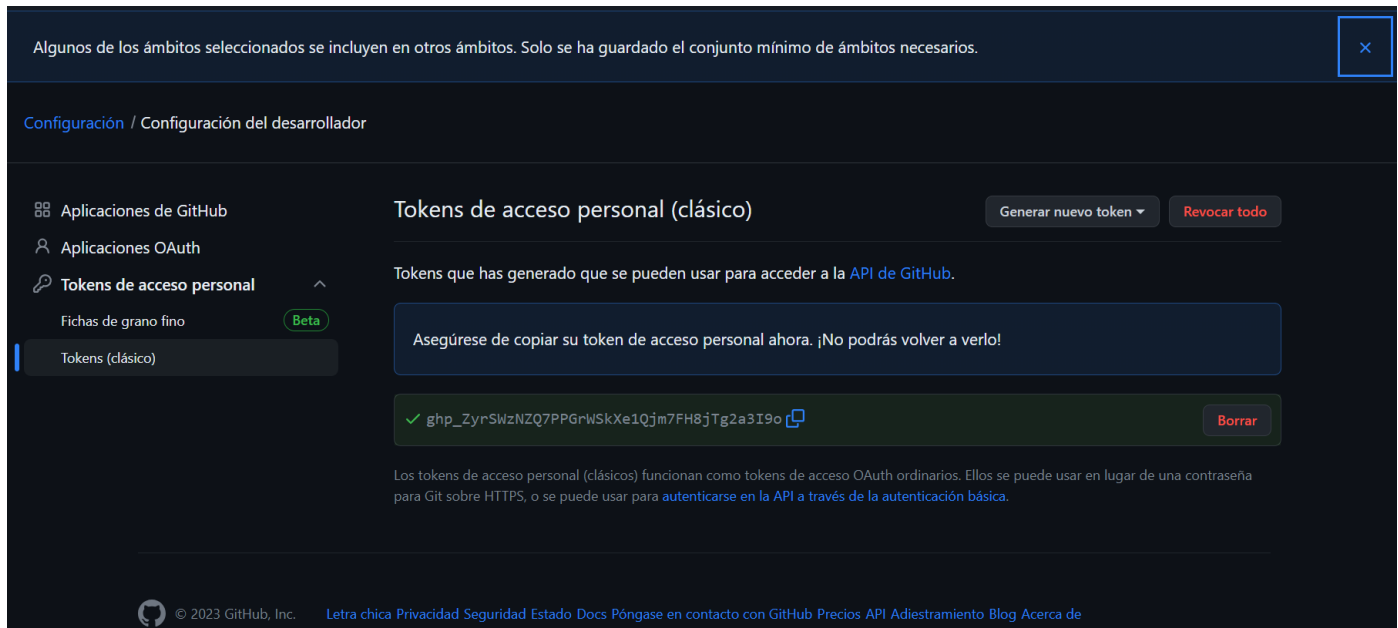
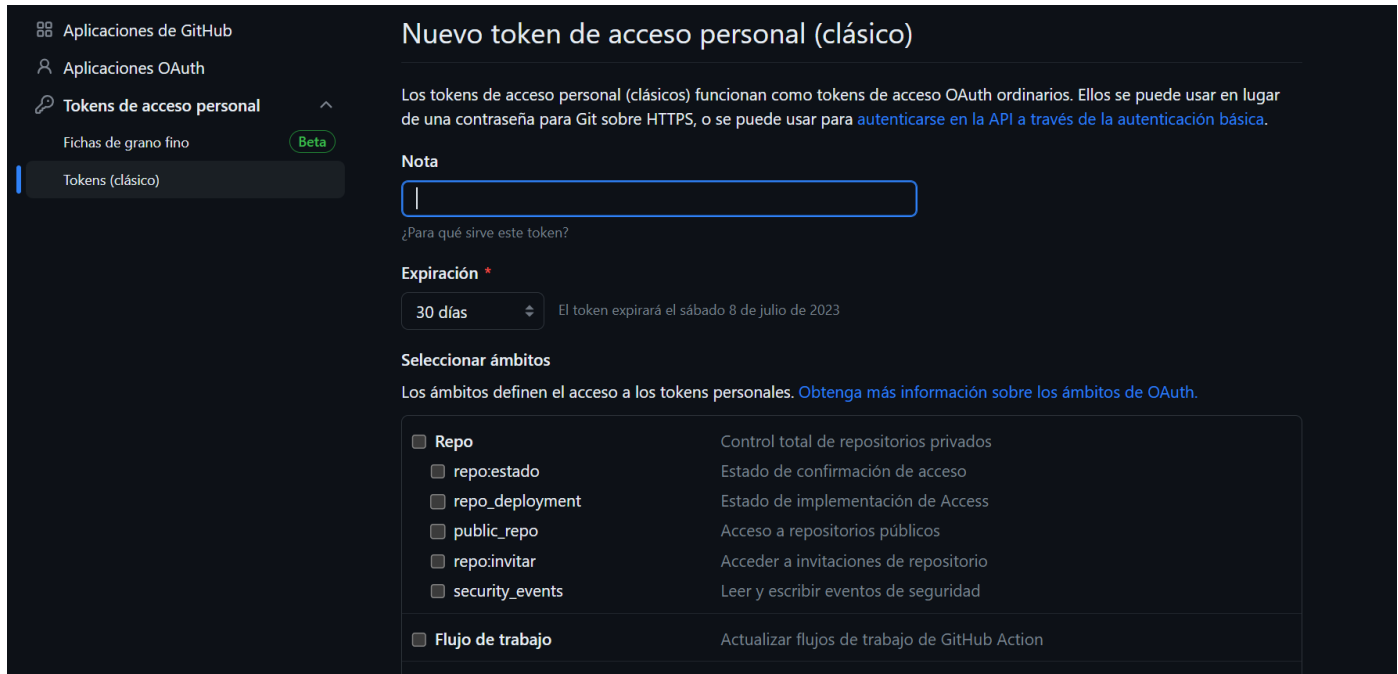
Contraseña

[¿Olvidó su contraseña?](#)

Confirmar

Propina: Estás entrando en **modo sudo**. Una vez que haya realizado una acción protegida por sudo, solo se le pedirá que vuelva a autenticarse después de unas horas de inactividad.

[Letra chica](#) [Privacidad](#) [Seguridad](#) [Póngase en contacto con GitHub](#)



Cuando terminemos de crearlo, ahora si podremos subir nuestros archivos, para asegurarnos de que Github se instaló y configuro correctamente, se puede verificar con los siguientes comandos.

```
git --version  
git config --global user.name git config  
git config --global user.email
```

En caso de no tenerlos los podemos instalar de la siguiente manera: Abre la terminal en Ubuntu. Puedes hacerlo presionando Ctrl+Alt+T. Actualiza la lista de paquetes de tu sistema ejecutando el siguiente comando: `sudo apt update` Instala github → `sudo apt install git` Asigna tus credenciales → `git config --global user.name "Nombre"`
`git config --global user.email tu@email.com`

Abre la terminal y navega hasta el directorio que contiene los archivos que deseas subir. Puedes usar el comando `cd` para cambiar de directorio.

Inicializa un nuevo repositorio Git en el directorio ejecutando el siguiente comando:

Añade los archivos que deseas subir al repositorio Git. Puedes agregar todos los archivos en el directorio actual usando el siguiente comando:


Realiza un commit de los archivos agregados con un mensaje descriptivo. Ejecuta el siguiente comando:

```
magali@PC5:~/Prog$ git init
hint: Using 'master' as the name for the initial branch. This default branch name
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/magali/Prog/.git/
magali@PC5:~/Prog$ git add
Nothing specified, nothing added.
hint: Maybe you wanted to say 'git add .'
hint: Turn this message off by running
hint: "git config advice.addEmptyPathspec false"
magali@PC5:~/Prog$ git commit -m "Cargando Archivos"
Author identity unknown
```

Enlaza tu repositorio local con el repositorio remoto de GitHub. Utiliza la URL del repositorio en formato HTTPS o SSH, dependiendo de tu preferencia utilizando el token de acceso creado anteriormente.

El link agregando antes tu token.



✓ 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 

Borrar

Crea una nueva rama.

Luego, cambia a la nueva rama ejecutando:

Por último, sube los archivos al repositorio remoto ejecutando el siguiente comando:

Verifica que se hayan subido correctamente a tu repositorio.

Ya podemos observar que se muestra la rama con los archivos.