



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño



Universidad Autónoma de Baja California



Facultad de Ingeniería, Arquitectura y Diseño F.I.A.D.

Campus Ensenada

Ramón Mejía Martínez 372099

Grupo: 932

Actividad 9 3/4

Curp

Programación Estructurada

Ensenada, Baja California a 22 de octubre de 2023.



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Ingeniero en Software y tecnologías emergentes

Materia: Programación Estructurada / Clave 36276

Alumno: Ramón Mejía Martínez

Matrícula: 372099

Maestro: Pedro Núñez Yépiz

Actividad No. : 9 3/4

Tema - Unidad : Librerías en c, Cadenas, Funciones.

Ensenada Baja California a 22 de octubre del 2023.



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

1. Introducción

En esta actividad se realizará la implementación en c del sistema para generar la curp completamente validada.

2. Competencia

Esta actividad tiene como objetivo el manejo cadenas, librerías y funciones, esto para poder implementar un sistema con muchas condiciones ayudándose de las funciones para modularizar el código.

3. Fundamentos

Librerías en C.

Las librerías en c son archivos o bloques de código con funciones especiales que podemos incluir en nuestro código importando sus funcionalidades.

Para la creación de nuestra propia librería podemos crear un archivo.h que contendrá todas las funciones que deseemos, este archivo puede contener los prototipos de las funciones e incluir los cuerpos en otro archivo también puede contener tanto prototipos como cuerpos.

Para

importar estas librerías en nuestros archivos.c se hace de la siguiente manera:

```
#include "(ruta de la librería)"
```

Usando la palabra reservada `#include` con la que se agregan otras librerías, entre comillas escribimos la ruta de nuestra librería, ahora tendremos todas las funciones existentes en el archivo.h para nuestro archivo.c.

Para Visual Studio Code podemos escribir una descripción de la función con comentarios en el archivo.h y se mostrara en nuestro archivo.c para mejor entendimiento del código.

```
op; int validNum(char *txt, int limi, int lims)
    Valida un numero entero
sys Parametros:
pri -txt Cadena a mostrar antes de leer la varia
pri -limi Limite inferior aceptado
pri -lims Limite superior aceptado
op=validNum("Escoge una opcion: ",1,2);
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Cadenas.

En C no existen la variable tipo cadenas. Existen

- arreglos de caracteres
- punteros a carácter

Cuando se almacena una cadena de caracteres debe estar terminada por el carácter cuyo código es 0 ('\0') ya que es lo que esperan encontrar las funciones de cadenas.

Las funciones de cadenas esperan una dirección de memoria (char *); como el nombre de un arreglo es la dirección donde se almacena el arreglo, las funciones de cadenas pueden recibir tanto un array como un puntero.

Para declarar una cadena se hace de la siguiente manera:

```
char identificador[10];
```

Usando la palabra reservada char, identificador y su dimensión, si la cadena lee un contenido menor a 10 por ejemplo "Casa" esto se ubicara de la siguiente forma.

0	1	2	3	4	5	6	7	8	9
'C'	'a'	's'	'a'	'\0'					

Las cadenas se pueden manejar de distintas formas ya sea como arreglos o con ciertas funciones, pero para esta practica se manejarán como arreglos exceptuando para leer la cadena.

Para leer una cadena se usa la función gets(identificador);

Para imprimir se puede usar scanf("%s",identificador); o imprimiendo carácter por carácter, para esto es necesario un estructura de control repetitiva:

```
for(i=0;identificador[i]!='\0';i++)  
{  
    printf("%c",identificador[i]);  
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Funciones.

Las funciones suelen encapsular una operación más o menos compleja de la que se deriva un resultado. Para ejecutar esta operación, las funciones pueden precisar la invocación de otras funciones (o incluso de ellas mismas como es el caso de las funciones recursivas).

Las funciones en un programa son entidades que dado un conjunto de datos (los parámetros), se les encarga realizar una tarea muy concreta y se espera hasta obtener el resultado. Lo idóneo es dividir tareas complejas en porciones más simples que se implementan como funciones. La división y agrupación de tareas en funciones es uno de los aspectos más importantes en el diseño de un programa.

Las partes de las funciones son: Prototipo, se encuentra antes de la función main y es la definición de las funciones las cuales llevan, tipo de dato que devolverán, nombre y parámetros de entrada.

Ejemplo Prototipo:

```
int suma(int a,int b);
```

El int inicial es el tipo de dato que regresa, suma el nombre, a y b son los parámetros de los cuales se debe especificar qué tipo de datos son

Otra parte es la llamada a la función la cual se hace en la función main o dentro de otra función.

Ejemplo Llamada:

```
resultado=suma(5,2);
```

```
resultado=suma(cal1,cal2);
```

Y por último el cuerpo de la función, esto va después de la función main, se debe poner el prototipo exactamente igual a excepción de que se quitan los puntos y comas y debajo se crea el bloque que contendrá el código del proceso que hará la función. Ejemplo cuerpo de la función:



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
int suma(int a,int b)
```

```
{
```

```
    int r;
```

```
    r=a+b;
```

```
    return r;
```

```
}
```

4. Procedimiento

1.

CURP

CADENAS.

INSTRUCCIONES:

1.- Realiza un programa en C que utilice una librería propia

(**Funciones de validar cadenas y numeros**)

2.- Realiza reporte de práctica (**librerías en c, Cadenas, Funciones**)

3.- Sube a Blackboard, programa, librería, reporte de practica y PDF anexo con capturas del código

(4 Archivos /1 .cpp / 1 .h , / 2 PDF)

4.- Sube a GitHub en tu repositorio los 3 documentos y poner enlace en BlackBoard

NOTA: No se puede utilizar ninguna función de la librería <string.h> solo funciones propias que se encuentren en la librería.

2.

CURP

Realiza un programa que sirva para generar el **CURP** de una persona.

El programa debe pedir los datos al usuario, generar, almacenar en una cadena y desplegar el CURP.

El programa deberá repetirse cuantas veces desee el usuario

Nota: el programa deberá estar **100% validado**, de datos entrada así como las reglas que e deben cumplir al generar el curp



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

5. Resultados y conclusiones:

Main:

La función principal se usó para el menú del curp.

```
int main()
{
    int op;
    do
    {
        system("cls");
        printf("Curp\n\n");
        printf("1) Generar curp\n");
        printf("2) Salir\n");
        op=validNum("Escoge una opcion: ",1,2);
        if(op==1)
        {
            curp();
        }
        printf("\n");
        system("pause");
    }while(op!=2);
    return 0;
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Función curp:

- No recibe ni devuelve nada.
- Llama a la función correspondiente para leer los datos necesarios.

```
void curp(void)
{
    srand(time(NULL));
    //datos
    char rnd[2];
    char curp[19];
    char name[STRING], appat[STRING], apmat[STRING];
    char anio[4], mes[2], dia[2], estado[2], sex;
    char s;

    s=datos(name, appat, apmat, anio, mes, dia, estado);
    if(s==1)
    {
        sex='H';
    }
    else
    {
        sex='M';
    }
}
```




Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

- Genera un aproximado de la curp aun sin validar para después validarla.

```
curp[0]=appat[0];
curp[1]=vocalInterna(appat);
curp[2]=apmat[0];
curp[3]=name[0];
curp[4]=anio[2];
curp[5]=anio[3];
curp[6]=mes[0];
curp[7]=mes[1];
curp[8]=dia[0];
curp[9]=dia[1];
curp[10]=sex;
curp[11]=estado[0];
curp[12]=estado[1];
curp[13]=consInterna(appat);
curp[14]=consInterna(apmat);
curp[15]=consInterna(name);
curp[16]=curp16(anio);
itoa(rand()%10,rnd,10);
curp[17]=rnd[0];
curp[18]='\0';|
/*
system("cls");
printf("\n");
printf("Curp: ");
strprint(curp);
*/
validCurp(curp,name,appat,apmat);

system("cls");
printf("Curp: ");
printf("%s",curp);
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Función curp16:

- Recibe el año de nacimiento.
- Según el año de nacimiento devuelve el carácter correspondiente.

```
char curp16(char anio[])
{
    if(anio[0]=='1')
    {
        return anio[2];
    }
    else
    {
        switch(anio[2])
        {
            case '0':
                return 'A';
                break;
            case '1':
                return 'B';
                break;
            case '2':
                return 'C';
                break;
        }
    }
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Función datos:

- Recibe todas las cadenas para guardar los datos.
- Lee y valida llamando a otras funciones el nombre, fecha de nacimiento, estado y sexo.
- Aquí se valida si la persona no tiene algún apellido.

```
int datos(char name[],char appat[],char apmat[],char anio[],char mes[],char dia[],char estado[])
{
    int nap;
    int sex;
    system("cls");
    printf(" Datos\n\n");
    printf("Ingresa tu nombre: ");
    validStrCurp(name);
    strM(name);

    printf("\nTiene apellido paterno?\n");
    printf("1) Si\n");
    printf("2) No\n");
    nap=validNum("Escoge una opcion: ",1,2);
    if(nap==1)
    {
        printf("\nIngresa tu apellidido paterno: ");
        validStrCurp(appat);
        strM(appat);
    }
    else
    {
        appat[0]='X';
        appat[1]='\0';
    }

    printf("\nTiene apellido materno?\n");
    printf("1) Si\n");
```

- Devuelve la elección del sexo.

```
system("cls");
printf("Fecha de nacimiento\n\n");
fecha(anio,mes,dia);

Est(estados);

system("cls");
sex=validNum("\nIngresa tu sexo\n1) Masculino\n2) Femenino\nElige una opcion: ",1,2);

return sex;
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Función validStrCurp:

- Recibe la cadena a validar (nombre y apellidos).
- Lee la cadena y verifica si coincide con algún carácter valido.
- Solo permite letras, espacios, algunos símbolos, no permite iniciar ni terminar con espacio ni que haya dos espacios seguidos,

```
void validStrCurp(char str[])
{
    int i,valido,band,espacio;
    do
    {
        fflush(stdin);
        gets(str);
        band=1;
        espacio=1;
        i=0;
        do
        {
            valido=0;
            //fflush(stdin);
            //c=getch();
            if(str[i]>='A') ...
            if(str[i]>='a') ...
            if(str[i]==-91) //ñ ...
            if(str[i]==-92) //Ñ ...
            if(simbolo(str[i])) ...
            if(dieresis(str[i]))
            {
                valido++;
            }
        }
    }
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

```
        if(str[i]=='\0')
        {
            if(str[i-1]!=' ')
            {
                valido++;
                band=0;
            }
            else
            {
                printf("\nla cadena no puede acabar con espacio\n");
                printf("\nVuela a ingresar la cadena: ");
                espacio=0;
            }
        }
        if(valido!=0)
        {
            i++;
        }
        else
        {
            if(espacio)
            {
                printf("\nla cadena tiene caracteres invalidos\n");
                printf("\nVuela a ingresar la cadena: ");
            }
            band=0;
        }
    }while(band);

}while(valido==0);
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Función simbolo:

- Recibe un carácter.
- Devuelve si es un símbolo valido para el curp.

```
int simbolo(char c1)
{
    switch(c1)
    {
        case 39:
        case 45:
        case 47:
        case 46:
            return 1;
            break;
        default:
            return 0;
    }
}
```

Función dieresis:

- Recibe un carácter.
- Devuelve si es una vocal con diéresis o no.

```
int dieresis(char c)
{
    switch(c)
    {
        case -114: //a
        case -45:  //e
        case -40:  //i
        case -103: //o
        case -124: //a
        case -119: //e
        case -117: //i
        case -108: //o
            return 1;
            break;
        default:
            return 0;
    }
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Función fecha:

- Recibe las cadenas para la fecha.
- Lee y valida la fecha de nacimiento del usuario.

```
void fecha(char anio[],char mes[],char dia[])
{
    int born[3];
    bool leap=false;
    born[2]=validNum("\nIngresa tu año de nacimiento: ",1900,2023);
    itoa(born[2],anio,10);

    born[1]=validNum("\nIngresa tu mes de nacimiento: ",1,12);
    itoa(born[1],mes,10);
    if(born[1]>=10)
    {
        itoa(born[1],mes,10);
    }
    else
    {
        itoa(born[1],mes,10);
        mes[1]=mes[0];
        mes[0]='0';
    }

    if(born[1]==2) //mes
    {
        if(born[2]%4==0) //año
        {
            leap=true;
        }
    }
    born[0]=validNum("\nIngresa tu día de nacimiento: ",1,diaValid(born[1],leap));
    if(born[0]>=10) ...
    else ...
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Función diaValid:

- Recibe el mes y si el año es bisiesto.
- Devuelve el límite máximo que se puede elegir para el día.

```
int diaValid(int mes,bool leap)
{
    switch(mes)
    {
        case 1:
        case 3:
        case 5:
        case 7:
        case 8:
        case 10:
        case 12:
            return 31;
            break;
        case 4:
        case 6:
        case 9:
        case 11:
            return 30;
            break;
        case 2:
            if(leap==true)
            {
                return 29;
            }
            else
            {
                return 28;
            }
            break;
    }
}
```




Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Función Est:

- Recibe la cadena para el estado.
- Asigna las letras correspondientes a la cadena estado según la elección.

```
void Est(char estado[])
{
    int E;
    char estList[33][3]={
        "AS", "BC", "BS", "CC", "CS", "CH", "CM", "CL", "CM", "DG",
        "GT", "GR", "HG", "JC", "EM", "MN", "MS", "NT", "NL", "OC",
        "PL", "QT", "QR", "SP", "SL", "SR", "TC", "TS", "TL", "VZ",
        "YN", "ZS", "NE"
    };

    system("cls");
    printf("Estado\n\n");
    printf("1) Aguascalientes\n");
    printf("2) Baja California\n");
    printf("3) Baja California Sur\n");
    printf("4) Campeche\n");
    printf("5) Coahuila de Zaragoza\n");
    printf("6) Colima\n");
    printf("7) Durango\n");
    printf("8) Guanajuato\n");
    printf("9) Guerrero\n");
    printf("10) Hidalgo\n");
    printf("11) Jalisco\n");
    printf("12) Mexico\n");
    printf("13) Michoacan\n");
    printf("14) Morelos\n");
    printf("15) Nayarit\n");
    printf("16) Nuevo Leon\n");
    printf("17) Oaxaca\n");
    printf("18) Pinar del Rio\n");
    printf("19) Queretaro\n");
    printf("20) Quintana Roo\n");
    printf("21) San Luis Potosi\n");
    printf("22) Sinaloa\n");
    printf("23) Sonora\n");
    printf("24) Tabasco\n");
    printf("25) Tamaulipas\n");
    printf("26) Tlaxcala\n");
    printf("27) Veracruz\n");
    printf("28) Yucatan\n");
    printf("29) Zacatecas\n");
    printf("30) Chihuahua\n");
    printf("31) Coahuila de Zaragoza\n");
    printf("32) Zacatecas\n");
    printf("33) Extranjero\n");
    E=validNum("Escoge tu estado: ",1,33);

    estado[0]=estList[E-1][0];
    estado[1]=estList[E-1][1];
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Función validCurp:

- Recibe el nombre y la aproximación de la curp.
- Valida todo el nombre, el uso de diéresis y símbolos, y las palabras inconvenientes

```
void validCurp(char curp[],char name[],char appat[],char apmat[])
{
    validName(curp,name,appat,apmat);

    //diéresis
    if(dieresis(curp[0]))
    {
        curp[0]='X';
    }
    if(dieresis(curp[2]))
    {
        curp[2]='X';
    }
    if(dieresis(curp[3]))
    {
        curp[3]='X';
    }
    system("pause");

    validSimbolo(curp);

    if(validPalabraInconven(curp))
    {
        curp[1]='X';
    }
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Función validName:

- Recibe la curp aproximada y las cadenas del nombre.

```
void validName(char curp[],char name[],char appat[],char apmat[])
```

- Comprueba si el nombre tiene espacios, si los tiene y el nombre coincide con alguno que no es valido para redactar el curp entonces se crea una nueva cadena para guardar el nombre sin el nombre invalido, se repite varias veces este proceso para nombres que lleven varios nombres inválidos consecutivos.
- Se hace lo mismo para los apellidos.

```
if(espacio(name))
{
    r=3;
    if(compName(name))
    {
        i=0;
        while(name[i]!=' ')
        {
            i++;
        }
        i++;
        k=0;
        for(j=i;name[j]!='\0';j++)
        {
            name2[k]=name[j];
            k++;
        }
        name2[k]='\0';
        r=0;
    }
    if(espacio(name2)) ...
    if(espacio(name3)) ...

    if(r==0) ...
    else
    {
        if(r==1) ...
        else
        {
            if(r==2)
            {
                curp[3]=name4[0];
                curp[15]=consInterna(name4);
            }
        }
    }
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

- Según el nombre usado asigna los caracteres correspondientes a la curp.

```
if(r==0)
{
    curp[0]=appat2[0];
    curp[1]=vocalInterna(appat2);
    curp[13]=consInterna(appat2);
}
else
{
    if(r==1)
    {
        curp[0]=appat3[0];
        curp[1]=vocalInterna(appat3);
        curp[13]=consInterna(appat3);
    }
    else
    {
        if(r==2)
        {
            curp[0]=appat4[0];
            curp[1]=vocalInterna(appat4);
            curp[13]=consInterna(appat4);
        }
    }
}
```

Función espacio:

- Recibe una cadena.
- Regresa si tiene algún espacio o no.

```
int espacio(char str[])
{
    int i;
    for(i=0;str[i]!='\0';i++)
    {
        if(str[i]==' ')
        {
            return 1;
        }
    }
    return 0;
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Función compName:

Función compAp:

La función compAp es igual, la diferencia es que no agrega a la lista los nombres.

- Recibe la cadena para el nombre.
- Compara si el nombre coincide con algún nombre invalido.
- Regresa si coincidió o no.

```
int compName(char name[])
{
    int i,j,suma;
    char no_validName[27][strlen(name)]={"MARIA","MA","MA.","M.","M","JOSE","J.","J","DA","DAS","DE","DEL","DER",
    "DI","DIE","DD","Y","EL","LA","LOS","LAS","LE","LES","MAC","MC","VAN","VON"};

    for(j=0;j<27;j++)
    {
        suma=0;
        for(i=0;name[i]!=' ';i++)
        {
            if(name[i]==no_validName[j][i])
            {
                suma++;
            }
        }

        if(suma==i)
        {
            suma++;
            return 1;
        }
    }
    return 0;
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Función vocalInterna:

- Recibe el apellido paterno.
- Busca el primer vocal interna del apellido paterno, toma los símbolos como vocales ya que para la curp son usados como vocales y consonantes.
- Regresa el carácter correspondiente según la primer vocal o símbolo que encuentre.

```
char vocalInterna(char appat[])
{
    int i;
    for(i=1; appat[i]!='\0'; i++)
    {
        if(simbolo(appat[i])) ...
        if(appat[i]=='A') ...
        if(appat[i]=='E') ...
        if(appat[i]=='I') ...
        if(appat[i]=='O') ...
        if(appat[i]=='U')
        {
            return appat[i];
        }

        if(dieresis(appat[i]))
        {
            return 'X';
        }
    }
    return 'X';
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Función consInterna:

- Recibe una cadena.
- Regresa la primera consonante que encontró, si no encuentra una consonante devuelve X.

```
char consInterna(char str[])
{
    int i;
    for(i=1;str[i]!='\0';i++)
    {
        if(str[i]!='A')
        {
            if(str[i]!='E')
            {
                if(str[i]!='I')
                {
                    if(str[i]!='O')
                    {
                        if(str[i]!='U')
                        {
                            if(!dieresis(str[i]))
                            {
                                return str[i];
                            }
                        }
                    }
                }
            }
        }
    }
    return 'X';
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Función validSimbolo:

- Recibe la curp.
- Si encuentra un símbolo valido lo reemplaza por una X.

```
void validSimbolo(char curp[])
{
    int i;
    for(i=0;curp[i]!='\0';i++)
    {
        if(simbolo(curp[i]))
        {
            curp[i]='X';
        }
    }
}
```




Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Función validPalabraInconven:

- Recibe la curp.
- Esta es la última validación que se hace a la curp.
- Define todas las palabras inconvenientes y las compara con los primeros 4 caracteres de la curp.
- Regresa si coincide con alguna palabra inconveniente. Si coincide se reemplaza el segundo carácter de la curp por X en la función validCurp.

```
int validPalabraInconven(char curp[])
{
    int i,j;
    int suma;
    char inconv[81][5]={
        "BACA","LOCO","BAKA","LOKA","BUEI","LOKO","BUEY","MAME","CACA","MAMO",
        "CACO","MEAR","CAGA","MEAS","CAGO","MEON","CAKA","MIAR","CAKO","MION",
        "COGE","MOCO","COGI","MOKO","COJA","MULA","COJE","MULO","COJI","NACA",
        "COJO","NACO","COLA","PEDA","CULO","PEDO","FALO","PENE","FETO","PIPI",
        "GETA","PITO","GUEI","POPO","GUEY","PUTA","JETA","PUTO","JOTO","QULO",
        "KACA","RATA","KACO","ROBA","KAGA","ROBE","KAGO","ROBO","KAKA","RUIN",
        "KAKO","SENO","KOGI","TETA","KOGI","VACA","KOJA","VAGA","KOJE","VAGO",
        "KOJI","VAKA","KOJO","VUEI","KOLA","VUEY","KULO","WUEI","LILO","WUEY",
        "LOCA"
    };

    for(i=0;i<81;i++)
    {
        suma=0;
        for(j=0;j<4;j++)
        {
            if(curp[j]==inconv[i][j])
            {
                suma++;
            }
        }
        if(suma==j)
        {
            return 1;
        }
    }
    return 0;
}
```



Universidad Autónoma de Baja California

Facultad de Ingeniería Arquitectura y Diseño

Conclusión:

El sistema para la generación de la curp puede parecer simple a primera visita pero tiene muchas excepciones que hace que la curp vaya siendo modificada en su cada vez que se valida un punto, el uso de librerías propias nos ayuda a evitar reescribir código y agilizar un poco la creación de nuevo código, el uso de las funciones nos ayuda a modularizar el programa, al ser un código tan largo nos ayuda igual que las librerías a reutilizar líneas de código y a separar el problema en problemas más sencillos.

En cuanto al proceso de la creación del programa, salió bien a excepción del uso de una función con punteros, que al parecer hacia que se desbordara la memoria, esto causo que se hiciera más procesos en lugar de aprovechar lo que se hacía en la función. Para la validación del nombre se podría optimizar bastante el código.

6. Anexos

PDF:

https://drive.google.com/file/d/1Uy-nxvHldz3CNfD54vclqAmGw2nWfY_/view?usp=sharing

Repositorio:

<https://github.com/Muners24/Programacion-Estructurada>



7. REFERENCIAS

Diseño de algoritmos y su codificación en lenguaje C

Corona, M.A. y Ancona, M.A. (2011)..

España: McGraw-Hill.

ISBN: 9786071505712

Programación estructurada a fondo:implementación de algoritmos en C

:Pearson Educación.Sznajdleder, P. A. (2017)..

Buenos Aires,Argentina: Alfaomega

Como programar en C/C++

H.M. Deitel/ P.J. Deitel

Segunda edición

Editorial: Prentice Hall.

ISBN:9688804711

Programación en C.Metodología, estructura de datos y objetos

Joyanes, L. y Zahonero, I. (2001)..

España:McGraw-Hill.

ISBN: 8448130138

De Computadores, P., & Olave, T. A. (s/f). *Algoritmos de Búsqueda y Ordenamiento*.

Utfsm.cl. Recuperado el 9 de octubre de 2023, de <https://www.inf.utfsm.cl/~noell/IWI-131-p1/Tema8b.pdf>