

UNIVERSIDAD NACIONAL DE TRUJILLO
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
ESCUELA PROFESIONAL DE INFORMÁTICA



TEMA:

PROYECTO I UNIDAD – RECICLADORA MC

CURSO:

ESTRUCTURA DE DATOS

CICLO - SECCIÓN:

II - A

ESTUDIANTES:

MENDEZ CRUZ, ANGELY YAHAYRA

MENDEZ CRUZ, CIARA SOLANGE

DOCENTE:

CRUZ FLORIÁN, IRIS AUREA

TRUJILLO _ PERÚ

2020

ÍNDICE

I. RESUMEN	3
II. DESCRIPCIÓN DEL PROBLEMA Y/O CASO	4
III. PROPUESTA DE SOLUCIÓN	5
IV. DESARROLLO DE LA SOLUCIÓN	6
3.1 ANÁLISIS.....	6
ANÁLISIS DEL PROBLEMA (ENTRADA – PROCESO Y SALIDA)	6
3.2 DISEÑO	8
3.2.1 INTERFACES PRINCIPALES.....	8
3.3 IMPLEMENTACIÓN	12
PSEUCÓDIGO Y LIBRERÍAS	12
V. CONCLUSIONES	34
VI. RECOMENDACIONES	36
VII. REFERENCIAS.....	38

I. RESUMEN

En el presente proyecto se ha diseñado e implementado algoritmos de estructuras, arreglos unidimensionales, de cadenas, de estructuras condicionales, de estructuras repetitivas, la creación de funciones y librerías nuevas y el uso de funciones y librerías predeterminadas, también. Todo ello, permitió la creación del registro “Recicladora MC”. Este registro, proporciona una interfaz de bienvenida, de menú principal y secundaria en el cual mediante varias opciones los usuarios pueden acceder desde la opción -quienes somos- hasta la opción de registrar -residuos hospitalarios-.

El registro está diseñado para ser usado por personas, empresas, municipalidades y hospitales, en tres registros diferentes: residuos domiciliarios, residuos municipales, residuos hospitalarios, que están ubicados en la parte central de la interfaz.

Los algoritmos de estructuras en el programa, están representados por las estructuras: info_datos, varios_materiales, desechos, recicladores, info_datos_m, varios_materiales_m, desechos_m, municipalidades, info_datos_h, varios_materiales_h, desechos_h, hospitales, el uso de ellas permite la creación de una base datos, para el registro de diferentes tipos de residuos. Los algoritmos de estructuras condicionales en el programa son: if secuenciales y switch. También, empleamos la estructura repetitiva for y do-while, arreglos unidimensionales y cadenas. La creación de funciones, permitió no repetir acciones y no ocupar más espacio en memoria, sino reutilizar las funciones en varias partes del programa y las librerías nuevas creadas son: <TITULO.h>, <I-PRINCIPAL.h> , < I-SECUNDARIA.h> y el uso de funciones y librerías predeterminadas como: <iostream>, <conio.h>, <windows.h>, <string.h>, <stdio.h>, <stdlib.h>.

II. DESCRIPCIÓN DEL PROBLEMA Y/O CASO

Actualmente, son muchos los países que luchan por tomar medidas para limitar o disminuir sus emisiones contaminantes, desechos de los ciudadanos, y proteger el medio ambiente. Aún existen países y zonas que enfrentan graves problemas de excesiva contaminación y degradación de sus espacios naturales. Por ejemplo, Perú es uno de ellos.

Dentro del país, se encuentra la Región de La Libertad, y específicamente la provincia de Trujillo, en la cual existen diferentes mercados que abastecen de alimentos e insumos a los trujillanos, uno de ellos es el Mercado “La Hermelinda”, el cual emite una drástica contaminación de residuos, produce al menos 100 toneladas diarias de residuos sólidos que son arrojadas en los exteriores del centro de abastos y no en el botadero de El Milagro, lo que genera, además, una gran contaminación ambiental en el lugar, según El Comercio (2019). Esto podría generar un “alto riesgo sanitario” para la población por la alta proliferación de roedores y residuos sólidos, y la posibilidad de un brote de peste en la ciudad. Ante esta problemática nosotras, hemos considerado y tomado en cuenta, la importancia de conocer y practicar el Reciclaje, el cual brinda múltiples beneficios a favor de las personas y el medio ambiente.

El Reciclaje es una de las alternativas utilizadas para la reducción del volumen de residuos sólidos. Se trata de un proceso que consiste básicamente en volver a utilizar materiales que fueron desechados y que aún son aptos para elaborar otros productos o refabricar los mismos. Una forma de pensar con respecto a los residuos es la que propone la estrategia de las “3R”: reducir – reutilizar - reciclar.

Nosotras nos enfocamos en la R, de Reciclar, que se refiere a devolver al ciclo productivo los residuos. Reciclando convertimos a los materiales recuperados de los residuos en valorados recursos. A este proceso contribuyen, pequeñas y medianas recicladoras a favor del medio ambiente, comprando y valorando productos y/o materiales reutilizables, para convertirlos en nuevos con diversas utilidades.

III. PROPUESTA DE SOLUCIÓN

La propuesta de solución y tema objeto de este proyecto es la creación de una base de datos para una empresa Recicladora, con un nombre, por ejemplo, Recicladora MC.

Esta recicladora, es una empresa de reciclaje privada trujillana que busca reducir la cantidad de residuos enviados a disposición final favoreciendo la cadena entera de reciclaje: concientización, separación, recolección, transformación, comercialización.

La cual contribuye a la creación de puestos de trabajo y al desarrollo económico alrededor del negocio de la recolección y el procesamiento de los materiales, mediante la venta y compra de residuos como:

- Residuos En General
- Residuos Orgánicos
- Residuos Vidrios
- Residuos Plásticos Y Envases Metálicos
- Residuos Papel
- Residuos Peligrosos

A través de tener un registro, podrá la empresa conocer y obtener información del sector del reciclaje, que permita realizar diagnósticos, medir la evolución y el aporte a la sociedad, y capacitar a los asociados a las personas u organizaciones que reciclan, y así comenzar a divulgar la información mediante sus principales redes sociales, para invitar a la población trujillana, a tomar conciencia y cuidar el planeta, ya que sin duda las pequeñas acciones contribuyen a grandes cambios.

El objetivo principal del proyecto es que los ciudadanos tomen conciencia de la importancia de tratar los residuos de forma adecuada, minimizando la generación de residuos, re-usando mediante la compra y venta de varios residuos, desarrollo del negocio una manera sostenida y armónica., y reaprovechando aquellos que tienen valor comercial y pueden ser convertidos en nuevos productos.

III. DESARROLLO DE LA SOLUCIÓN

3.1 ANÁLISIS

3.1.1 ENTRADA

Obtuvimos datos de un dispositivo de entrada, en este caso, el teclado. Y lo almacenamos en variables.

- ♦ Para la función menú_principal: opcion (int), regresar (int)
- ♦ Para la función interfaz_secundaria: opcion_menu (int)
- ♦ Para la función menu_principal_domiciliarios: elegir (int), numero_recicladores (int), numero.dni (int).
- ♦ Para la función menu_principal_municipalidades: elegir_m (int), numero_municipalidades (int), numero_m_1.codigo (int).
- ♦ Para la función menu_principal_hospitalarios: elegir_h (int), numero_hospitales (int), numero_h.codigo_unico (int).

3.1.2 PROCESO

Los algoritmos de estructuras en el programa, están representados por las estructuras: info_datos, varios_materiales, desechos, recicladores, info_datos_m, varios_materiales_m, desechos_m, municipalidades, info_datos_h, varios_materiales_h, desechos_h, hospitales, el uso de ellas permite la creación de una base datos, para el registro de diferentes tipos de residuos. Los algoritmos de estructuras condicionales en el programa son: if secuenciales y switch. También, empleamos la estructura repetitiva for y do-while, arreglos unidimensionales y cadenas.

Las librerías nuevas creadas son: <TITULO.h>, <I-PRINCIPAL.h> , < I-SECUNDARIA.h>

También se hará uso de funciones y librerías predeterminadas del lenguaje de programación C++ como: <iostream>, <conio.h>, <windows.h>, <string.h>, <stdio.h>, <stdlib.h>.

3.1.3 SALIDA

Mostramos valores en un dispositivo de salida, en este caso, la pantalla.

- ♦ Para la funciones ver_recicladores, ver_recicladores_m, ver_hospitales se mostrará en pantalla los datos personales de cada registro de residuos domiciliarios, municipales y hospitalarios.

3.2 DISEÑO.

3.2.1 INTERFACES PRINCIPALES (CAPTURA DE PANTALLAS)

INICIO: Muestra el nombre de la recicladora e imagen central.



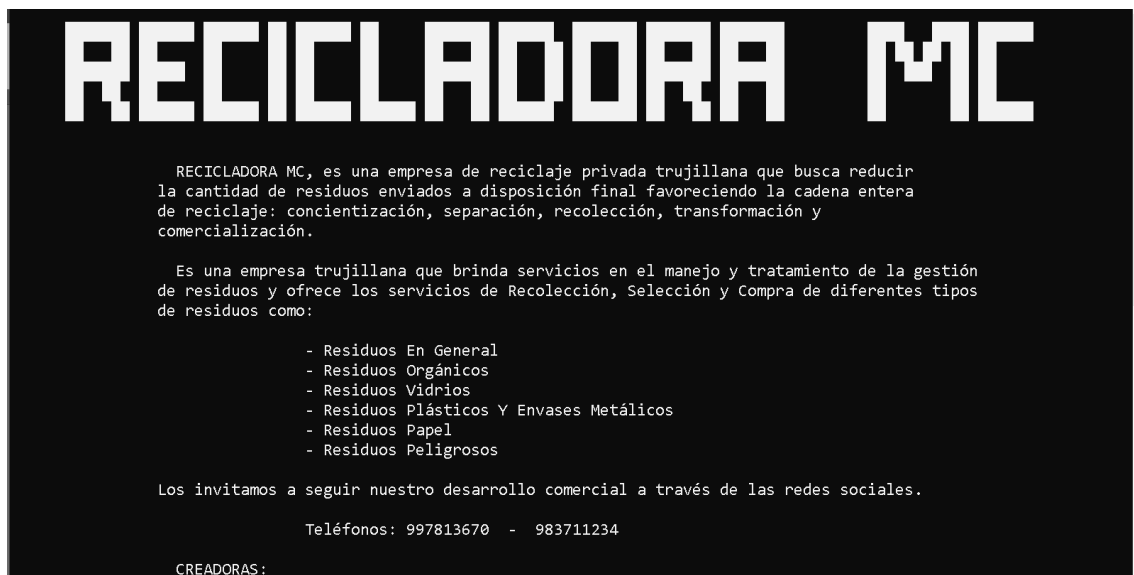
I-PRINCIPAL: Muestra el menú, quienes somos y salir.



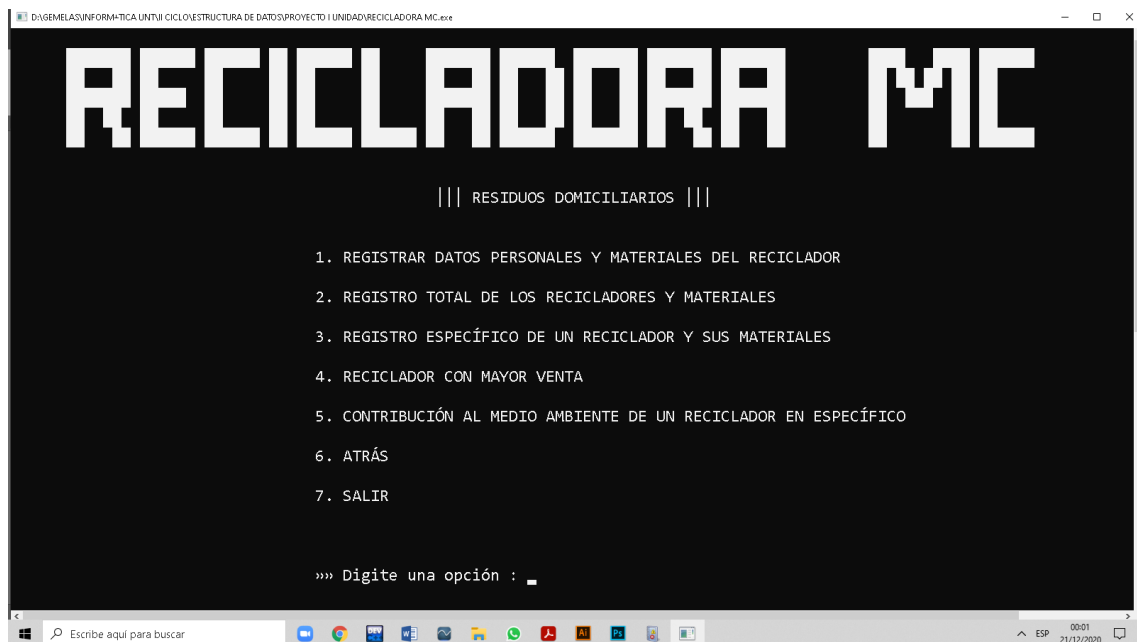
I-SECUNDARIA: Muestra las tres opciones de RESIDUOS.



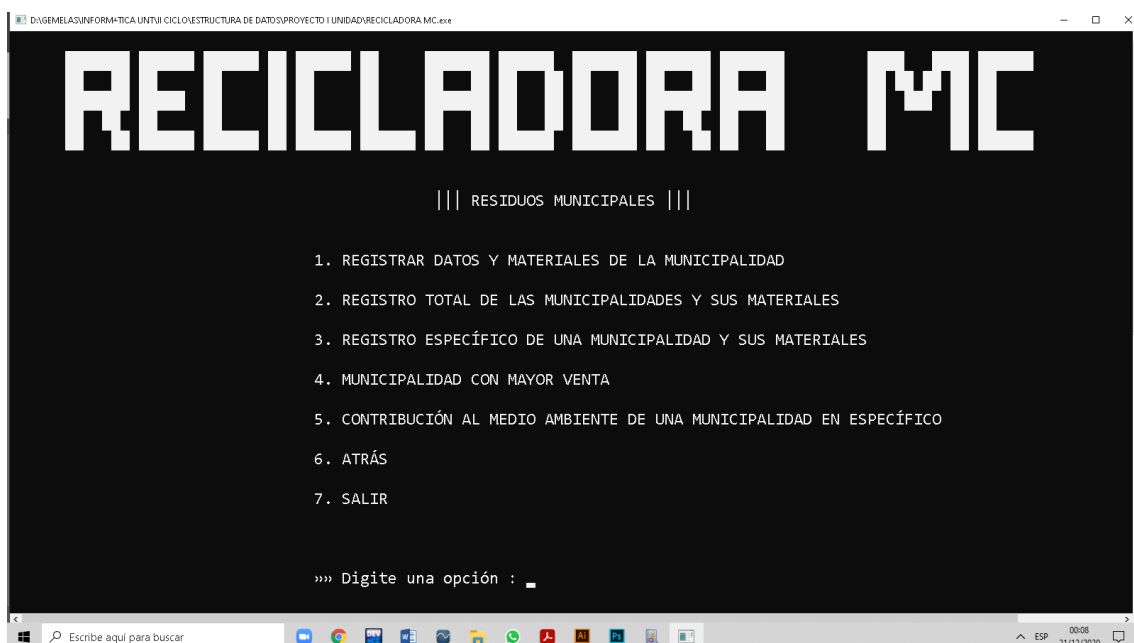
QUIENES SOMOS: Muestra la descripción de la recicladora, creadoras y número de contacto.



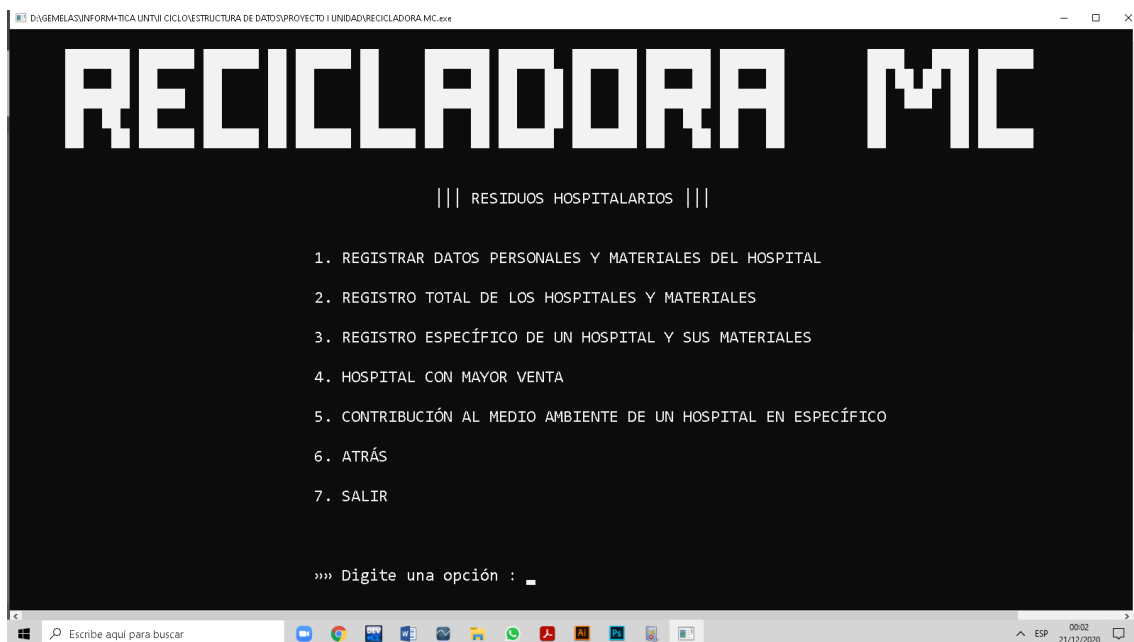
RESIDUOS DOMICILIARIOS: Muestra las opciones del menú del registro para domicilios.



RESIDUOS MUNICIPALES: Muestra las opciones del menú del registro para municipalidades.



RESIDUOS HOSPITALARIOS: Muestra las opciones del menú del registro para hospitales.



3.3 IMPLEMENTACIÓN.

- PSEUDOCÓDIGO

1. RECICLADORA MC

```
//MENU DOMICILIARIOS
struct info_datos
{
}
struct solo_material
{
}
struct varios_materiales
{
struct      solo_material S [30]
}
struct desechos
{
string      desechos
}
struct recicladores
{
struct      info_datos direccion
struct      varios_materiales P
struct      desechos D
}
reciclador [30], numero
//struct recicladores numero
string desechos [6] =
{
"EN GENERAL GRIS)", "ORGANICOS (MORADO)",      "VIDRIOS (VERDE",
"PLASTICOS Y ENVASES METALICOS AMARILLO)",      "PAPEL (AZUL)",
"PELIGROSOS (ROJO"
}
SubProceso TITULO_A ()
cout << "\t\t\t\t\t || REGISTRAR DATOS PERSONALES Y MATERIALES DEL
RECICLADOR || \n\n"
FinSubProceso
SubProceso RETORNA_PRINCIPAL ()
cin >> regresar
Borrar Pantalla
Si ( regresar == 1 )
SOLO_TITULO
menu_principal_domiciliarios
FinSi
FinSubProceso
{
reciclador[p].P.precio_total = 0
Para int i = 0 Hasta s Con Paso 1
SOLO_TITULO
titulo_a
```



```

Borrar Pantalla
registrar_materiales i , reciclador[i].P.cifra_materiales, 0 , 8
2:
SOLO_TITULO
titulo_a
cout << "\n\t\t\t\t\t| REGISTRO DE MATERIALES RECICLADOS DE
"<<reciclador[i].nombres<<" "<<reciclador[i].apellidos<<" | \n\n"
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 5
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 7
cin >> reciclador[i].P.cifra_materiales
reciclador[i].D.desechos = "\n\n\t\t\t\t\t -| DESECHOS ORGANICOS MORADO |- \n"
Borrar Pantalla
registrar_materiales i , reciclador[i].P.cifra_materiales, 1 , 5
3:
SOLO_TITULO
titulo_a
cout << "\n\t\t\t\t\t| REGISTRO DE MATERIALES RECICLADOS DE
"<<reciclador[i].nombres<<" "<<reciclador[i].apellidos<<" | \n\n"
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 2
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 7
cin >> reciclador[i].P.cifra_materiales
reciclador[i].D.desechos = "\n\n\t\t\t\t\t -| DESECHOS VIDRIOS VERDE |- \n"
Borrar Pantalla
registrar_materiales i , reciclador[i].P.cifra_materiales, 2 , 2
4:
SOLO_TITULO
titulo_a
cout << "\n\t\t\t\t\t| REGISTRO DE MATERIALES RECICLADOS DE
"<<reciclador[i].nombres<<" "<<reciclador[i].apellidos<<" | \n\n"
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 6
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 7
cin >> reciclador[i].P.cifra_materiales
reciclador[i].D.desechos = "\n\n\t\t\t\t\t -| DESECHOS PLASTICOS Y ENVASES
METALICOS AMARILLO |- \n"
Borrar Pantalla
registrar_materiales i , reciclador[i].P.cifra_materiales, 3 , 6
5:
SOLO_TITULO
titulo_a
cout << "\n\t\t\t\t\t| REGISTRO DE MATERIALES RECICLADOS DE
"<<reciclador[i].nombres<<" "<<reciclador[i].apellidos<<" | \n\n"
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 1
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 15
cin >> reciclador[i].P.cifra_materiales
reciclador[i].D.desechos = "\n\n\t\t\t\t\t -| DESECHOS PAPEL AZUL |- \n"
Borrar Pantalla
registrar_materiales i , reciclador[i].P.cifra_materiales, 4 , 1
6:
SOLO_TITULO
titulo_a
cout << "\n\t\t\t\t\t| REGISTRO DE MATERIALES RECICLADOS DE
"<<reciclador[i].nombres<<" "<<reciclador[i].apellidos<<" | \n\n"
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 4
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 15

```

```

cin >> reciclador[i].P.cifra_materiales
reciclador[i].D.desechos = "\n\n\t\t\t\t -| DESECHOS PELIGROSOS ROJO |- \n"
Borrar Pantalla
registrar_materiales i , reciclador[i].P.cifra_materiales, 5 , 4
FinSegun
FinPara
FinSubProceso
SubProceso VER_MATERIALES ( NUMERO_RECICLADORES)
cout << reciclador[numero_recicladore]s.D.desechos
cout << "\n\t\t\t\t MATERIALES\t CANTIDAD\t PRECIO - VENTA\tSUBTOTAL\n "
Para int i=0 Hasta reciclador[numero_recicladore]s.P.cifra_materiales Con
Paso 1
cout << "\n\t\t\t\t
"<<reciclador[numero_recicladore]s.P.S[i].nombre_material <<"\t "
cout << reciclador[numero_recicladore]s.P.S[i].cantidad <<" \t\t "
cout << reciclador[numero_recicladore]s.P.S[i].precio_venta <<" \t\t\t"
cout << reciclador[numero_recicladore]s.P.S[i].sub_total
FinPara
cout << "\n\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t- " =cout << "\n\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t- " -1
FinSubProceso
SubProceso VER_RECICLADORES ( NUMERO_RECICLADORES )
Si ( numero_recicladore]s > 0 )
Para int i = 0 Hasta numero_recicladore]s Con Paso 1
cout <<"\n\n\t\t\t\t\t -| DATOS PERSONALES DEL RECICLADOR " << i+1 << " |- \n"
cout <<"\n\t\t\t\t Nombres y apellidos : " <<reciclador[i].nombress<<"
"<<reciclador[i].apellidos
cout <<"\n\t\t\t\t Direcci\xA2n : " <<
reciclador[i].direccion.calle<<" "<<reciclador[i].direccion.numero<<"
"<<reciclador[i].direccion.urbanizacion
ver_materiales i
FinPara
FinSi
FinSubProceso
SubProceso MAYOR_VENTA ( NUMERO_RECICLADORES )
mayor=reciclador[0].P.precio_total
Para int i=0 Hasta numero_recicladore]s Con Paso 1
cout <<"\n\t\t\t\t\t | - DATOS PERSONALES DEL RECICLADOR " << i+1 << " - |
\n"
cout <<"\n\t\t\t\t\t Nombres y apellidos : " <<reciclador[i].nombress<<"
"<<reciclador[i].apellidos
cout <<"\n\t\t\t\t\t TOTAL DE VENTA : " <<
reciclador[i].P.precio_total <<"\n\n"
Si (reciclador[i].P.precio_total>mayor)
mayor=reciclador[i].P.precio_total
valor=i
FinSi
FinPara
cout << "\n\n\t\t\t\t\t | ~ MAYOR VENTA S/. ~ | \n "
cout << "\n\t\t\t\t\t Nombres y apellidos : "
<<reciclador[valor].nombress<<" "<<reciclador[valor].apellidos
FinSubProceso
SubProceso CONTRIBUCION ( )
SOLO_TITULO
Para int i=0 Hasta numero_recicladore]s Con Paso 1
Si ( reciclador[i].dni == numero.dni )

```



```

cout << "\t\t\t\t 7. SALIR                                     \n\n\n"
cin >> elegir
Borrar Pantalla
Segun  elegir Hacer
1:
SOLO_TITULO
Hacer
titulo_a
cin >> numero_recicladores
Borrar Pantalla
Hasta Que ( numero_recicladores <1)
regis_recicladores  numero_recicladores
SOLO_TITULO
titulo_a
retorna_principal
2:
SOLO_TITULO
cout << "\t\t\t\t || REGISTRO TOTAL DE LOS RECICLADORES Y MATERIALES || \n\n"
ver_recicladores  numero_recicladores
retorna_principal
3:
SOLO_TITULO
cin >> numero.dni
Borrar Pantalla
dni_un_reciclador  numero_recicladores
retorna_principal
4:
SOLO_TITULO
cout << "\t\t\t\t\t || RECICLADOR CON MAYOR VENTA || \n\n\n"
mayor_venta  numero_recicladores
retorna_principal
5:
SOLO_TITULO
cin >> numero.dni
Borrar Pantalla
contribucion
retorna_principal
6:
interfaz_secundaria
FinSegun
Hasta Que ( (elegir <1) || (elegir > 7) )
FinSubProceso
SubProceso RES_DOMICILIOS ( )
menu_principal_domiciliarios
FinSubProceso
//MEN◆ MUNICIPALES
struct info_datos_m
}
struct solo_material_m
}
struct varios_materiales_m
struct      solo_material_m S [30]
}
struct desechos_m
string      desechos
}
}

```

```

struct municipalidades
struct      info_datos_m ubicacion
struct      varios_materiales_m P
struct      desechos_m D
}
municipalidad [30], numero_m, numero_m_1
//struct recicladores numero
string desechos_m [6] =
"EN GENERAL GRIS)", "ORGANICOS (MORADO)",      "VIDRIOS (VERDE",
"PLASTICOS Y ENVASES METALICOS AMARILLO)",      "PAPEL (AZUL)", "PELIGROSOS
(ROJO"
}
SubProceso TITULO_A_M ()
cout << "\t\t\t\t\t || REGISTRAR DATOS Y MATERIALES DE LA MUNICIPALIDAD ||
\n\n"
FinSubProceso
SubProceso RETORNA_PRINCIPAL_M ()
cin >> regresar_m
Borrar Pantalla
Si ( regresar_m == 1 )
SOLO_TITULO
menu_principal_municipalidades
FinSi
FinSubProceso
municipalidad[p].P.precio_total = 0
Para int i = 0 Hasta s Con Paso 1
SOLO_TITULO
titulo_a_m
cout << "\n\t\t\t\t\t| REGISTRO DE MATERIALES RECICLADOS DE LA
"<<municipalidad[p].denominacion <<" | \n\n"
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), color
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 7
cout << "\n\t\t\t\t\t ~ Digite el nombre del material reciclado: "
cin.getlinemunicipalidad[p].P.S[i].nombre_material, 30 , '\n'
cout << "\t\t\t\t\t ~ Digite la cantidad Kg                : "
cin >> municipalidad[p].P.S[i].cantidad
cout << "\t\t\t\t\t ~ Digite el precio - venta S/.          : "
cin >> municipalidad[p].P.S[i].precio_venta
Borrar Pantalla
municipalidad[p].P.S[i].sub_total = municipalidad[p].P.S[i].cantidad *
municipalidad[p].P.S[i].precio_venta
municipalidad[p].P.precio_total = municipalidad[p].P.precio_total +
municipalidad[p].P.S[i].sub_total
FinPara
}
SubProceso REGIS_RECICLADORES_M ( N_C )
Para int i = 0 Hasta n_c Con Paso 1
SOLO_TITULO
titulo_a_m
cout << "\n\t\t\t\t\t\t\t | REGISTRO DE MUNICIPALIDADES | \n"
cout << "\n\t\t\t\t\t\t\t|- Datos Municipales -| \n\n"
cout << "\t\t\t\t\t\t\t~ Digite el n\xA3mero del CODIGO de la Municipalidad: "
cin >> municipalidad[i].codigo
cout << "\t\t\t\t\t\t\t~ Digite la Denominaci\xA2n de la Municipalidad : "
cin.getline municipalidad[i].denominacion,50,'\n'
cout << "\n\n\t\t\t\t\t\t\t|- Ubicaci\xA2n de la Municipalidad -| \n\n"
cout << "\t\t\t\t\t\t\t~ Digite el distrito al que pertenece      : "
cin.getline municipalidad[i].ubicacion.distrito,30,'\n'

```

```

cout << "\t\t\t\t\t~ Digite la provincia al que pertenece : "
cin.getline municipalidad[i].ubicacion.provincia,30,'\n'
cout << "\t\t\t\t\t~ Digite el departamento al que pertenece: "
cin.getline municipalidad[i].ubicacion.departamento,30,'\n'
Borrar Pantalla
SOLO_TITULO
titulo_a_m
cout << "\n\t\t\t\t\t| REGISTRO DE MATERIALES RECICLADOS DE LA
"<<municipalidad[i].denominacion<<" | \n\n"
cout << "\t\t\t\t\t\t 1. DESECHOS EN GENERAL GRIS \n\n"
cout << "\t\t\t\t\t\t 3. VIDRIOS VERDE \n\n"
cout << "\t\t\t\t\t\t 5. PAPEL AZUL \n\n"
cout << "\t\t\t\t\t\t 6. DESECHOS PELIGROSOS ROJO \n\n"
cin >> opc_m
Borrar Pantalla
Segun      opc_m Hacer
1:
SOLO_TITULO
titulo_a_m
cout << "\n\t\t\t\t\t| REGISTRO DE MATERIALES RECICLADOS DE LA
"<<municipalidad[i].denominacion<<" | \n\n"
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 8
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 7
cin >> municipalidad[i].P.cifra_materiales
municipalidad[i].D.desechos = "\n\n\t\t\t\t\t -| DESECHOS EN GENERAL GRIS |- \n"
Borrar Pantalla
registrar_materiales_m i , municipalidad[i].P.cifra_materiales, 0 , 8
2:
SOLO_TITULO
titulo_a_m
cout << "\n\t\t\t\t\t| REGISTRO DE MATERIALES RECICLADOS DE LA
"<<municipalidad[i].denominacion<<" | \n\n"
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 5
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 7
cin >> municipalidad[i].P.cifra_materiales
municipalidad[i].D.desechos = "\n\n\t\t\t\t\t -| DESECHOS ORGANICOS MORADO |-
\n"
Borrar Pantalla
registrar_materiales_m i , municipalidad[i].P.cifra_materiales, 1 , 5
3:
SOLO_TITULO
titulo_a_m
cout << "\n\t\t\t\t\t| REGISTRO DE MATERIALES RECICLADOS DE LA
"<<municipalidad[i].denominacion<<" | \n\n"
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 2
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 7
cin >> municipalidad[i].P.cifra_materiales
municipalidad[i].D.desechos = "\n\n\t\t\t\t\t -| DESECHOS VIDRIOS VERDE |- \n"
Borrar Pantalla
registrar_materiales_m i , municipalidad[i].P.cifra_materiales, 2 , 2
4:
SOLO_TITULO
titulo_a_m
cout << "\n\t\t\t\t\t| REGISTRO DE MATERIALES RECICLADOS DE LA
"<<municipalidad[i].denominacion<<" | \n\n"
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 6
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 7

```

[illegible]

[illegible]

```
cout << municipalidad[i].D.desechos
cout << "\n\t\t\t\t MATERIALES\t CANTIDAD\t PRECIO - VENTA\tSUBTOTAL\n "
Para int j=0 Hasta municipalidad[i].P.cifra_materiales Con Paso 1
cout << "\n\t\t\t\t "<<municipalidad[i].P.S[j].nombre_material <<"\t "
cout << municipalidad[i].P.S[j].cantidad <<" \t\t "
cout << municipalidad[i].P.S[j].precio_venta <<" \t\t\t"
cout << municipalidad[i].P.S[j].sub_total
FinPara
cout << "\n\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t- " =cout << "\n\t\t\t\t\t\t\t\t\t\t\t\t\t\t\t- " -1
FinSi
FinPara
FinSubProceso
SubProceso MENU_PRINCIPAL_MUNICIPALIDADES ()
Hacer
cout << "\t\t\t\t 1. REGISTRAR DATOS Y MATERIALES DE LA MUNICIPALIDAD
        \n\n"
cout << "\t\t\t\t 2. REGISTRO TOTAL DE LAS MUNICIPALIDADES Y SUS MATERIALES
        \n\n"
cout << "\t\t\t\t 4. MUNICIPALIDAD CON MAYOR VENTA
        \n\n"
cout << "\t\t\t\t 7. SALIR
        \n\n\n"
cin >> elegir_m
Borrar Pantalla
Segun elegir_m Hacer
1:
SOLO_TITULO
Hacer
titulo_a_m
cin >> numero_municipalidades
Borrar Pantalla
Hasta Que ( numero_municipalidades < 1 )
regis_recicladores_m numero_municipalidades
SOLO_TITULO
titulo_a_m
retorna_principal_m
2:
SOLO_TITULO
cout << "\t\t\t\t || REGISTRO TOTAL DE LAS MUNICIPALIDADES Y SUS MATERIALES
|| \n\n"
ver_recicladores_m numero_municipalidades
retorna_principal_m
3:
SOLO_TITULO
cin >> numero_m_1.codigo
Borrar Pantalla
dni_un_reciclador_m numero_municipalidades
retorna_principal_m
4:
SOLO_TITULO
cout << "\t\t\t\t\t\t || MUNICIPALIDAD CON MAYOR VENTA || \n\n\n"
mayor_venta_m numero_municipalidades
retorna_principal_m
5:
SOLO_TITULO
cin >> numero_m.codigo
Borrar Pantalla
```

```

contribucion_m numero_municipalidades
retorna_principal_m
6:
interfaz_secundaria
FinSegun
Hasta Que ( (elegir_m <1) || (elegir_m > 7) )
FinSubProceso
SubProceso RES_MUNICIPALIDADES ( )
menu_principal_municipalidades
FinSubProceso
//MEN HOSPITALARIOS
struct info_datos_h
}
struct solo_material_h
}
struct varios_materiales_h
struct solo_material_h S [30]
}
struct desechos_h
string desechos_h
}
struct hospitales
struct info_datos_h direccion
struct varios_materiales_h P
struct desechos_h D
}
hospital [30], numero_h
//struct recicladores numero
string desechos_h [6] =
"EN GENERAL GRIS)", "ORGANICOS (MORADO)", "VIDRIOS (VERDE",
"PLASTICOS Y ENVASES METALICOS AMARILLO)", "PAPEL (AZUL)", "PELIGROSOS
(ROJO"
}
SubProceso TITULO_A_H ( )
cout << "\t\t\t\t\t || REGISTRAR DATOS PERSONALES Y MATERIALES DEL HOSPITAL
|| \n\n"
FinSubProceso
SubProceso RETORNA_PRINCIPAL_H ( )
cin >> regresar_h
Borrar Pantalla
Si ( regresar_h == 1 )
SOLO_TITULO
menu_principal_hospitalarios
FinSi
FinSubProceso
hospital[p].P.precio_total_h = 0
Para int i = 0 Hasta s Con Paso 1
SOLO_TITULO
titulo_a_h
cout << "\n\t\t\t\t\t| REGISTRO DE MATERIALES RECICLADOS DE
"<<hospital[p].nombres_h<<" "<<hospital[p].modelo_h<<" | \n\n"
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), color
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 7
cout << "\n\t\t\t\t\t~~ Digite el nombre del material reciclado: "
cin.getlinehospital[p].P.S[i].nombre_material_h, 30 , '\n'
cout << "\t\t\t\t\t~~ Digite la cantidad Kg : "

```

```

cin >> hospital[p].P.S[i].cantidad_h
cout << "\t\t\t\t\t ~~ Digite el precio - venta S/.          : "
cin >> hospital[p].P.S[i].precio_venta_h
Borrar Pantalla
hospital[p].P.S[i].sub_total_h = hospital[p].P.S[i].cantidad_h *
hospital[p].P.S[i].precio_venta_h
hospital[p].P.precio_total_h = hospital[p].P.precio_total_h +
hospital[p].P.S[i].sub_total_h
FinPara
}
SubProceso REGIS_HOSPITALES ( N_C )
Para int i = 0 Hasta n_c Con Paso 1
SOLO_TITULO
titulo_a_h
cout << "\n\t\t\t\t\t | REGISTRO DE HOSPITALES | \n"
cout << "\n\t\t\t\t\t|- Datos Hospitalarios -| \n\n"
cout << "\t\t\t\t\t~~ Digite el n\xA3mero de CODIGO UNICO del hospital: "
cin >> hospital[i].codigo_unico
cout << "\t\t\t\t\t~~ Digite el nombre del hospital          : "
cin.getline hospital[i].nombres_h,30,'\n'
cout << "\t\t\t\t\t~~ Digite el modelo al que pertenece el hospital: "
cin.getline hospital[i].modelo_h,30,'\n'
cout << "\n\n\t\t\t\t\t|- Ubicaci\xA2n del Hospital -| \n\n"
cout << "\t\t\t\t\t~~ Digite el nombre del distrito      : "
cin.getline hospital[i].direccion.districto_h,30,'\n'
cout << "\t\t\t\t\t~~ Digite el nombre de la provincia : "
cin.getline hospital[i].direccion.provincia_h,30,'\n'
cout << "\t\t\t\t\t~~ Digite el nombre del departamento: "
cin.getline hospital[i].direccion.departamento_h,30,'\n'
Borrar Pantalla
SOLO_TITULO
titulo_a_h
cout << "\n\t\t\t\t\t| REGISTRO DE MATERIALES RECICLADOS DE
"<<hospital[i].nombres_h<<" "<<hospital[i].modelo_h<<" | \n\n\n"
cout << "\t\t\t\t\t\t 1. DESECHOS EN GENERAL GRIS \n\n"
cout << "\t\t\t\t\t\t 3. VIDRIOS VERDE \n\n"
cout << "\t\t\t\t\t\t 5. PAPEL AZUL \n\n"
cout << "\t\t\t\t\t\t 6. DESECHOS PELIGROSOS ROJO \n\n"
cin >> opc_h
Borrar Pantalla
Segun opc_h Hacer
1:
SOLO_TITULO
titulo_a_h
cout << "\n\t\t\t\t\t| REGISTRO DE MATERIALES RECICLADOS DE
"<<hospital[i].nombres_h<<" "<<hospital[i].modelo_h<<" | \n\n"
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 8
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 7
cin >> hospital[i].P.cifra_materiales_h
hospital[i].D.desechos_h = "\n\n\t\t\t\t\t -| DESECHOS EN GENERAL GRIS |- \n"
Borrar Pantalla
registrar_materiales_h i , hospital[i].P.cifra_materiales_h, 0 , 8

```



```

2:
SOLO_TITULO
titulo_a_h
cout << "\n\t\t\t\t\t| REGISTRO DE MATERIALES RECICLADOS DE
"<<hospital[i].nombres_h<<" "<<hospital[i].modelo_h<<" | \n\n"
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 5
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 7
cin >> hospital[i].P.cifra_materiales_h
hospital[i].D.desechos_h = "\n\n\t\t\t\t\t -| DESECHOS ORGANICOS MORADO |- \n"
Borrar Pantalla
registrar_materiales_h i , hospital[i].P.cifra_materiales_h, 1 , 5
3:
SOLO_TITULO
titulo_a_h
cout << "\n\t\t\t\t\t| REGISTRO DE MATERIALES RECICLADOS DE
"<<hospital[i].nombres_h<<" "<<hospital[i].modelo_h<<" | \n\n"
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 2
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 7
cin >> hospital[i].P.cifra_materiales_h
hospital[i].D.desechos_h = "\n\n\t\t\t\t\t -| DESECHOS VIDRIOS VERDE |- \n"
Borrar Pantalla
registrar_materiales_h i , hospital[i].P.cifra_materiales_h, 2 , 2
4:
SOLO_TITULO
titulo_a_h
cout << "\n\t\t\t\t\t| REGISTRO DE MATERIALES RECICLADOS DE
"<<hospital[i].nombres_h<<" "<<hospital[i].modelo_h<<" | \n\n"
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 6
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 7
cin >> hospital[i].P.cifra_materiales_h
hospital[i].D.desechos_h = "\n\n\t\t\t\t\t -| DESECHOS PLASTICOS Y ENVASES
METALICOS AMARILLO |- \n"
Borrar Pantalla
registrar_materiales_h i , hospital[i].P.cifra_materiales_h, 3 , 6
5:
SOLO_TITULO
titulo_a_h
cout << "\n\t\t\t\t\t| REGISTRO DE MATERIALES RECICLADOS DE
"<<hospital[i].nombres_h<<" "<<hospital[i].modelo_h<<" | \n\n"
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 1
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 15
cin >> hospital[i].P.cifra_materiales_h
hospital[i].D.desechos_h = "\n\n\t\t\t\t\t -| DESECHOS PAPEL AZUL |- \n"
Borrar Pantalla
registrar_materiales_h i , hospital[i].P.cifra_materiales_h, 4 , 1
6:
SOLO_TITULO
titulo_a_h
cout << "\n\t\t\t\t\t| REGISTRO DE MATERIALES RECICLADOS DE
"<<hospital[i].nombres_h<<" "<<hospital[i].modelo_h<<" | \n\n"
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 4
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 15
cin >> hospital[i].P.cifra_materiales_h
hospital[i].D.desechos_h = "\n\n\t\t\t\t\t -| DESECHOS PELIGROSOS ROJO |- \n"
Borrar Pantalla
registrar_materiales_h i , hospital[i].P.cifra_materiales_h, 5 , 4
FinSegun
FinPara

```

[illegible]

```

FinPara
FinSubProceso
SubProceso MENU_PRINCIPAL_HOSPITALARIOS ( )
Hacer
cout << "\t\t\t\t 1. REGISTRAR DATOS PERSONALES Y MATERIALES DEL HOSPITAL
        \n\n"
cout << "\t\t\t\t 2. REGISTRO TOTAL DE LOS HOSPITALES Y MATERIALES
        \n\n"
cout << "\t\t\t\t 4. HOSPITAL CON MAYOR VENTA
        \n\n"
cout << "\t\t\t\t 7. SALIR
        \n\n\n"
cin >> elegir_h
Borrar Pantalla
Segun  elegir_h Hacer
1:
SOLO_TITULO
Hacer
titulo_a_h
cin >> numero_hospitales
Borrar Pantalla
Hasta Que ( numero_hospitales <1)
regis_hospitales  numero_hospitales
SOLO_TITULO
titulo_a_h
retorna_principal_h
2:
SOLO_TITULO
cout << "\t\t\t\t || REGISTRO TOTAL DE LOS HOSPITALES Y MATERIALES || \n\n"
ver_hospitales  numero_hospitales
retorna_principal_h
3:
SOLO_TITULO
cin >> numero_h.codigo_unico
Borrar Pantalla
codigo_un_hospital  numero_hospitales
retorna_principal_h
4:
SOLO_TITULO
cout << "\t\t\t\t\t || HOSPITAL CON MAYOR VENTA || \n\n\n"
mayor_venta_h  numero_hospitales
retorna_principal_h
5:
SOLO_TITULO
cin >> numero_h.codigo_unico
Borrar Pantalla
contribucion_h  numero_hospitales
retorna_principal_h
6:
interfaz_secundaria
FinSegun
Hasta Que ( (elegir_h <1) || (elegir_h > 7) )
FinSubProceso
SubProceso RES_HOSPITALARIOS ( )
menu_principal_hospitalarios

```

```

FinPara
FinSubProceso
SubProceso MENU_PRINCIPAL_HOSPITALARIOS ( )
Hacer
cout << "\t\t\t\t 1. REGISTRAR DATOS PERSONALES Y MATERIALES DEL HOSPITAL
        \n\n"
cout << "\t\t\t\t 2. REGISTRO TOTAL DE LOS HOSPITALES Y MATERIALES
        \n\n"
cout << "\t\t\t\t 4. HOSPITAL CON MAYOR VENTA
        \n\n"
cout << "\t\t\t\t 7. SALIR
        \n\n\n"
cin  >> elegir_h
Borrar Pantalla
Segun  elegir_h Hacer
1:
SOLO_TITULO
Hacer
titulo_a_h
cin  >> numero_hospitales
Borrar Pantalla
Hasta Que ( numero_hospitales <1)
regis_hospitales  numero_hospitales
SOLO_TITULO
titulo_a_h
retorna_principal_h
2:
SOLO_TITULO
cout << "\t\t\t\t || REGISTRO TOTAL DE LOS HOSPITALES Y MATERIALES || \n\n"
ver_hospitales  numero_hospitales
retorna_principal_h
3:
SOLO_TITULO
cin  >> numero_h.codigo_unico
Borrar Pantalla
codigo_un_hospital  numero_hospitales
retorna_principal_h
4:
SOLO_TITULO
cout << "\t\t\t\t || HOSPITAL CON MAYOR VENTA || \n\n\n"
mayor_venta_h  numero_hospitales
retorna_principal_h
5:
SOLO_TITULO
cin  >> numero_h.codigo_unico
Borrar Pantalla
contribucion_h  numero_hospitales
retorna_principal_h
6:
interfaz_secundaria
FinSegun
Hasta Que ( (elegir_h <1) || (elegir_h > 7) )
FinSubProceso
SubProceso RES_HOSPITALARIOS ( )
menu_principal_hospitalarios

```

```

FinSubProceso
//MEN PRINCIPAL
SubProceso DESCRIPCION_SOMOS ()
cout <<" \n\n\t\t RECICLADORA MC, es una empresa de reciclaje privada
trujillana que busca reducir "
cout <<" \n\t\tla cantidad de revariableReservadaduos enviados a
dispovariableReservadaci\xA2n final favoreciendo la cadena entera "
cout <<" \n\t\tde reciclaje: concientizaci\xA2n, separaci\xA2n,
recolecci\xA2n, transformaci\xA2n y"
cout <<" \n\t\tcomercializaci\xA2n."
cout <<" \n\n\t\t Es una empresa trujillana que brinda servicios en el manejo
y tratamiento de la gesti\xA2n "
cout <<" \n\t\tde revariableReservadaduos y ofrece los servicios de
Recolecci\xA2n, Selecci\xA2n y Compra de diferentes tipos "
cout <<" \n\t\tde revariableReservadaduos como: "
cout <<" \n\n\t\t      - RevariableReservadaduos En General "
cout <<" \n\t\t      - RevariableReservadaduos Org\xA0nicos "
cout <<" \n\t\t      - RevariableReservadaduos Vidrios "
cout <<" \n\t\t      - RevariableReservadaduos Pl\xA0sticos Y Envases
Met\xA0licos "
cout <<" \n\t\t      - RevariableReservadaduos Papel "
cout <<" \n\t\t      - RevariableReservadaduos Peligrosos "
cout <<" \n\n\t\tLos invitamos a seguir nuestro desarrollo comercial a
trav\x82s de las redes sociales. "
cout <<" \n\n\t\t CREADORAS:"
cout <<" \n\n\t\t      - Angely Mendez Cruz "
cout <<" \n\t\t      - Ciara Mendez Cruz "
FinSubProceso
SubProceso ERFAZ_SECUNDARIA ()
SOLO_TITULO
I_SECUNDARIA
Hacer
cin >> opcion_menu
Hasta Que ( (opcion_menu <1) || (opcion_menu > 4) )
Borrar Pantalla
Segun opcion_menu Hacer
1//domiciliarios:
SOLO_TITULO
RES_DOMICILIOS
2//municipales:
SOLO_TITULO
RES_MUNICIPALIDADES
3//hospitalarios:
SOLO_TITULO
RES_HOSPITALARIOS
4:
menu_principal
FinSegun
FinSubProceso
SubProceso MENU_PRINCIPAL ()
//Interfaz principal

```

```

SOLO_TITULO
I_PRINCIPAL
Hacer
cin >> opcion
Hasta Que ( ( opcion < 1 ) || ( opcion > 3 ) )
Borrar Pantalla
Segun opcion Hacer
1 //menu:
//Interfaz secundaria
interfaz_secundaria
2 //quienes somos:
SOLO_TITULO
descripcion_somos
cin >> regresar
Borrar Pantalla
Si ( regresar == 1 )
SOLO_TITULO
menu_principal
FinSi
FinSegun
FinSubProceso
Algoritmo nombre
mode con: cols=160 lines=65
//interfaz inicio
I_TITULO
cout << "C A R G A N D O ..."
4000Esperar 4 segundos
Borrar Pantalla
menu_principal
Esperar Tecla
FinAlgoritmo

```

2. LIBRERÍAS:

- TITULO.H

```
SubProceso COLOR( A, B)
SetConsoleTextAttributeConsole, nuevo_color
FinSubProceso
SubProceso GOTOXY( X, Y)
hcon = GetStdHandleSTD_OUTPUT_HANDLE
COORD dwPos
dwPos.X = x
dwPos.Y= y
SetConsoleCursorPovvariableReservadationhcon,dwPos
FinSubProceso
SubProceso LETRAP( X)
FinSubProceso
SubProceso LETRAR( X)
FinSubProceso
SubProceso LETRAE( X)
FinSubProceso
SubProceso LETRAC( X)
FinSubProceso
SubProceso LETRAI( X)
FinSubProceso
SubProceso LETRAL( X)
FinSubProceso
SubProceso LETRAA( X)
FinSubProceso
SubProceso LETRAD( X)
FinSubProceso
SubProceso LETRAO( X)
FinSubProceso
SubProceso LETRAM( X)
FinSubProceso
SubProceso DIBUJO_RECICLAJE( X)
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 2
cout<<"          " <<endl
cout<<"
{
cout<<"
{
FinSubProceso
"<<endl
cout<<"          "
}
SubProceso I_TITULO()
color 0F
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 15
letraR6
letraE14
letraC21
letraI28
letraC31
```

```
letraL38
letraA45
letraD53
letraO61
letraR69
letraA77
letraM93
letraC105
Para int i=0 Hasta 80 Con Paso 1
dibujo_reciclajei
FinPara
Borrar Pantalla
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 15
letraR6
letraE14
letraC21
letraI28
letraC31
letraL38
letraA45
letraD53
letraO61
letraR69
letraA77
letraM93
letraC105
dibujo_reciclaje35
FinSubProceso
SubProceso SOLO_TITULO()
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 15
letraR6
letraE14
letraC21
letraI28
letraC31
letraL38
letraA45
letraD53
letraO61
letraR69
letraA77
letraM93
letraC105
FinSubProceso
```


- **I_PRINCIPAL.h**

```

SubProceso DIBUJO_MENU()
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 9
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 15
cout<<" ~ 1 ~ "<<endl
FinSubProceso
SubProceso DIBUJO QUIENES SOMOS()
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 6
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 15
cout<<" ~ 2 ~ "<<endl
FinSubProceso
SubProceso DIBUJO_SALIDA()
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 2
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 15
cout<<" ~ 3 ~ "<<endl
cout<<" SALIR "<<endl
FinSubProceso
SubProceso I_PRINCIPAL()
color 0F
dibujo_menu
dibujo_quienes_somos
dibujo_salida
FinSubProceso

```

- **I_SECUNDARIA.h**

```

uvariableReservadang namespace std
SubProceso DIBUJO_REVARIABLESERVADA_DOMICILIARIOS()
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 6
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 15
cout<<" ~ 1 ~ "<<endl
cout<<" REvariableReservadaDUOS DOMICILIARIOS "<<endl
FinSubProceso
SubProceso DIBUJO_REVARIABLESERVADA_MUNICIPALES()
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 2
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 15
cout<<" ~ 2 ~ "<<endl
cout<<" REvariableReservadaDUOS MUNICIPALES "<<endl
FinSubProceso
SubProceso DIBUJO_REVARIABLESERVADA_HOSPITALES()
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 4
SetConsoleTextAttribute GetStdHandle (STD_OUTPUT_HANDLE), 15
cout<<" ~ 3 ~ "<<endl
cout<<" REvariableReservadaDUOS HOSPITALARIOS "<<endl
FinSubProceso
SubProceso I_SECUNDARIA()
color 0F
dibujo_revariableReservada_domiciliarios
dibujo_revariableReservada_municipales
dibujo_revariableReservada_hospitales
FinSubProcesoFinSubProceso

```

V. CONCLUSIONES

1. La elaboración del registro para la empresa Recicladora MC, nos ayudó a incrementar nuestros conocimientos en el programa Dev C++, al igual que los conocimientos básicos totales del lenguaje de programación C++ y codificación adquiridos durante las clases de los diversos temas obtenidos durante la primera unidad, como algoritmos de estructuras, arreglos unidimensionales, de cadenas, de estructuras condicionales, de estructuras repetitivas. Además, reforzó nuestro interés por descubrir y aprender distintas librerías, funciones y comandos que simplifican los procedimientos y hacen que sea más simple las capacidades y el funcionamiento para su elaboración.
2. Mediante el programa realizado, se pudo obtener un registro y al ser usado por una empresa de reciclaje las personas pueden adquirir un registro, podrá la empresa conocer y obtener información del sector del reciclaje, que permita realizar diagnósticos, medir la evolución y el aporte a la sociedad, y capacitar a los asociados a las personas u organizaciones que reciclan, y así comenzar a divulgar la información mediante sus principales redes sociales, para invitar a la población trujillana, a tomar conciencia y cuidar el planeta, ya que sin duda las pequeñas acciones contribuyen a grandes cambios.
3. Una de las principales barreras que se enfrenta para la elaboración de un programa, es el desconocimiento de algunas funciones y/o estructuras del lenguaje C++, pues dado las diferentes alternativas de originalidad de cada integrante del equipo, la implementación de dichas alternativas origina una investigación y aprendizaje pleno, el cual involucraría extender los conocimientos fuera de lo aprendido, para evitar confusión y escasez de características dentro del programa.
4. La implementación del código fue de forma estructurada y de fácil comprensión para que, en futuros usos del mismo, sea accesible interpretar el funcionamiento

y que la inclusión de nuevas funcionalidades no sea una tarea traumática. Un claro ejemplo son los comentarios y pautas de escritura dentro del código.

5. El trabajo en equipo fue muy gratificante y a la vez nos da una amplia visión de los diferentes conocimientos y aportes que cada una de las dos integrantes ofrece. Fue importante que las integrantes del equipo estuvieran involucradas con la planeación y elaboración del programa y permitió verificar que esté como lo planeado. Finalmente, logramos que todas nuestras ideas, se complementen y con sinergia se elaboró el programa.

VI. RECOMENDACIONES

1. El programa debe ser correctamente instalado para no tener problemas al compilar y ejecutar, es decir en la carpeta que alberga el código Dev C++, es necesario que se encuentre las librerías, ejecutables, etc. del programa. Además, los usuarios deben tener conocimientos básicos en computación.
2. Establecer animaciones de sonidos atractivos, mediante cada cambio de opciones de la interfaz de Inicio, además de realizar las correcciones en el código para su implementación inmediata y así evitar posibles anomalías dentro del programa.
3. Diseñar y llevar a cabo la ejecución de imágenes o gif (imágenes en movimiento), en este caso del logo de reciclaje, y para fortalecer el entendimiento del programa y atracción ante el público, esto generará visualmente una comprensión de cada interfaz, además de la ubicación de cada registro para domicilios, municipalidades y hospitales.
4. Crear una opción llamada “seleccionar idioma”, esta opción permitirá que desarrollemos el programa en el idioma inglés o quechua, claro está teniendo apoyo y soporte de expertos profesionales o buenos conocimientos en ese aspecto.
5. Realizar otras temáticas para darle variedad al programa, es decir, según fechas importantes como navidad, año nuevo chino, Halloween, etc. ambientar las interfaces y enriquecer mucho su contenido. Un ejemplo, en navidad, colocar colores característicos de esta fecha, como rojo, verde, amarillo, sonidos navideños, además la figura animada principal puede tener un gorro navideño representativo.
6. Implementar una página web de la Recicladora MC, en el cual se pueda observar el registro de información de cada persona o institución, pues es muy común publicar los resultados sobre cuánto reciclamos.

7. Optimizar el espacio de memoria, teniendo en cuenta conocimientos y aprendizajes del lenguaje del código utilizado en el programa, el compilador, e incluso el mismo procesador, eso permitirá cambiar el código y reordenar instrucciones para mejorar el rendimiento.

VII. REFERENCIAS

Joyanes, L. (2007). *Estructura de Datos en C++*. Cuarta Edición. McGRAW-HILL/INTERAMERICANA DE ESPAÑA, S. A. U.

Cruz, J. (2012). *Proyecto de la materia: Estructura de Datos*. PREZI.
<https://prezi.com/ihfpwznwjvts/proyecto-de-la-materia-de-estructura-de-datos/>

Vicente, B., Roldán, M. (2016). *Estructuras de Datos*. Universidad de Málaga, Dpto. Lenguajes y CC. Computación E.T.S.I. Informática.

IBM - PC. *Código ASCII. American Standard Code for Information Interchange* (*Código Estadounidense Estándar para el Intercambio de Información*).
<https://elcodigoascii.com.ar/>

Juganaru, M. (2014). *Introducción a la programación*. PRIMERA EDICIÓN EBOOK. Grupo Editorial Patria, Mexico.
<https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=&cad=rja&uact=8&ved=2ahUKEwiyrGqqq3sAhUGmVkKHfeVAm8QFjAKegQIAxAC&url=https%3A%2F%2Fwww.editorialpatria.com.mx%2Fpdf%2F9786074384154.pdf&usg=AOvVaw2FoIqncneS032eQder37se>