

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



**TEMA:**

**PROYECTO II 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

<b>I. RESUMEN .....</b>	<b>3</b>
<b>II. DESCRIPCIÓN DEL PROBLEMA Y/O CASO .....</b>	<b>4</b>
<b>III. PROPUESTA DE SOLUCIÓN .....</b>	<b>5</b>
<b>IV. DESARROLLO DE LA SOLUCIÓN .....</b>	<b>6</b>
<b>3.1 ANÁLISIS.....</b>	<b>6</b>
ANÁLISIS DEL PROBLEMA (ENTRADA – PROCESO Y SALIDA) .....	6
<b>3.2 DISEÑO .....</b>	<b>8</b>
<b>3.2.1 INTERFACES PRINCIPALES.....</b>	<b>8</b>
<b>3.3 IMPLEMENTACIÓN .....</b>	<b>12</b>
PSEUCÓDIGOS DE LAS FUNCIONES .....	12
<b>V. CONCLUSIONES .....</b>	<b>25</b>
<b>VI. RECOMENDACIONES .....</b>	<b>27</b>
<b>VII. REFERENCIAS.....</b>	<b>29</b>

## I. RESUMEN

En el presente proyecto se ha diseñado e implementado listas doblemente enlazadas lineales, punteros, arreglos unidimensionales, de cadenas, estructuras condicionales, 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, info\_desechos, desechos, recicladores, info\_ubicacion, info\_desechos\_m, desechos\_m, municipalidades, info\_datos\_h, info\_desechos\_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 creadas nuevas 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\_codigo\_m (int).
- ♦ Para la función menu\_principal\_hospitalarios: elegir\_h (int), numero\_hospitales (int), numero\_codigo\_unico (int).

##### 3.1.2 PROCESO

Los algoritmos de estructuras en el programa, están representados por las estructuras: info\_datos, info\_desechos, desechos, recicladores, info\_ubicacion, info\_desechos\_m, desechos\_m, municipalidades, info\_datos\_h, info\_desechos\_h, desechos\_h, hospitales, el uso de ellas permite la creación de una base datos, realizada en arreglos de listas, las cuales tienen apuntadores o punteros. Para la estructura domiciliarios los apuntadores: anterior, siguiente y primero. Para la estructura municipalidades los apuntadores: anterior\_m, siguiente\_m y primero\_m. Para la estructura hospitalarios los apuntadores: anterior\_h, siguiente\_h y primero\_h. 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\_municipales, ver\_hospitales, ver\_recicladores\_fi, ver\_hospitales\_fi, ver\_municipales\_fi; en las tres funciones primeras se mostrará en pantalla los datos personales de cada registro de residuos domiciliarios, municipales y hospitalarios de Inicio a fin y las tres últimas de Fin a Inicio.

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.

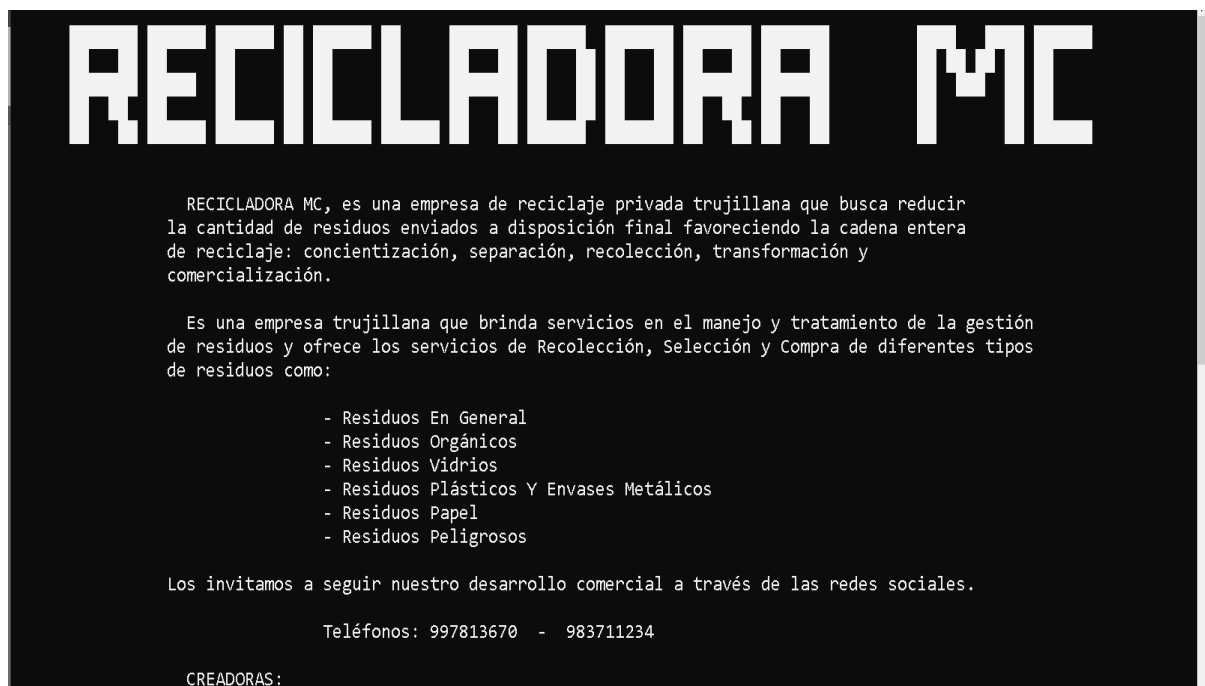




**I-SECUNDARIA: Muestra las tres opciones de RESIDUOS.**



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



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

```
RECICLADORA MC

||| RESIDUOS DOMICILIARIOS |||

1. REGISTRAR DATOS PERSONALES Y TIPO DE DESECHO DEL RECICLADOR
2. REGISTRO TOTAL DE LOS RECICLADORES Y TIPO DE DESECHO - DE INICIO A FIN
3. REGISTRO TOTAL DE LOS RECICLADORES Y TIPO DE DESECHO - DE FIN A INICIO
4. REGISTRO ESPECÍFICO DE UN RECICLADOR Y TIPO DE DESECHO
5. RECICLADOR CON MAYOR VENTA
6. RECICLADOR CON MENOR VENTA
7. AGREGAR NUEVO RECICLADOR AL REGISTRO
8. ELIMINAR DATOS DE UN RECICLADOR EN ESPECÍFICO
9. CONTRIBUCIÓN AL MEDIO AMBIENTE DE UN RECICLADOR EN ESPECÍFICO
10. ATRÁS
11. SALIR

»» Digite una opción :
```

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

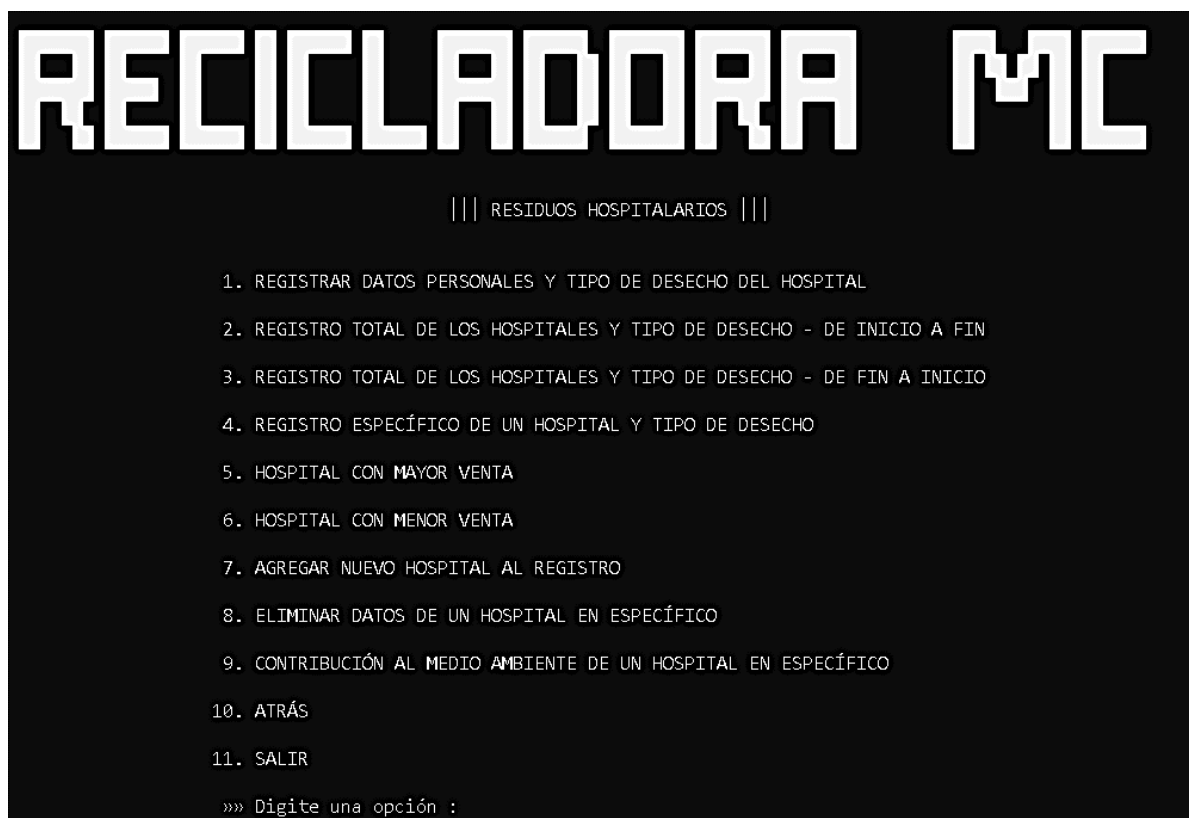
```
RECICLADORA MC

||| RESIDUOS MUNICIPALES |||

1. REGISTRAR DATOS Y TIPO DE DESECHO DE LA MUNICIPALIDAD
2. REGISTRO TOTAL DE LAS MUNICIPALIDADES Y TIPO DE DESECHO - DE INICIO A FIN
3. REGISTRO TOTAL DE LAS MUNICIPALIDADES Y TIPO DE DESECHO - DE FIN A INICIO
4. REGISTRO ESPECÍFICO DE UNA MUNICIPALIDAD Y TIPO DE DESECHO
5. MUNICIPALIDAD CON MAYOR VENTA
6. MUNICIPALIDAD CON MENOR VENTA
7. AGREGAR NUEVA MUNICIPALIDAD AL REGISTRO
8. ELIMINAR DATOS UNA MUNICIPALIDAD EN ESPECÍFICO
9. CONTRIBUCIÓN AL MEDIO AMBIENTE DE UNA MUNICIPALIDAD EN ESPECÍFICO
10. ATRÁS
11. SALIR

»» Digite una opción :
```

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



### 3.3 IMPLEMENTACIÓN.

- **PSEUDOCÓDIGOS DE LAS FUNCIONES DEL PROGRAMA**

#### 1. FUNCIÓN REGISTRAR RECICLADORES

**//Registra recicladores y tipo de residuos**

**Inicio** regis\_recicladores

**Para** i =0 **hasta** i < n\_c **con paso** 1 **hacer** // n\_c=numero de recicladores

**Mostrar** (REGISTRO DE RECICLADORES)

**Mostrar** (RECICLADOR N°, i +1)

**Mostrar** (Datos Personales)

**Mostrar** (Digite el número de DNI del reciclador)

**Hacer**

**Leer** (nuevo (dni))

**Mientras** nuevo(dni) < 10000000 **o** nuevo(dni) > 99999999

**Mostrar** (Digite los nombres del reciclador)

**Leer** (nuevo (nombres))

**Mostrar** (Digite los apellidos del reciclador)

**Leer** (nuevo (apellidos))

**Mostrar** (Dirección de Vivienda)

**Mostrar** (Digite los nombres de la calle)

**Leer** (nuevo (direccion.calle))

**Mostrar** (Digite el número de la casa)

**Leer** (nuevo (direccion.numero))

**Mostrar** (Digite la urbanización o distrito)

**Leer** (nuevo (direccion.urbanizacion))

**Mostrar** (Registro de Desechos)

**Mostrar** (1. DESECHOS EN GENERAL (GRIS))

**Mostrar** (2. ORGÁNICOS (MORADO))

**Mostrar** (3. VIDRIOS (VERDE))

**Mostrar** (4. PLÁSTICOS Y ENVASES METÁLICOS (AMARILLO))

**Mostrar** (5. PAPEL (AZUL))

**Mostrar** (6. DESECHOS PELIGROSOS (ROJO))

## Según opc hacer

- 1: Mostrar** (REGISTRO DE DESECHOS, desechos [0], DE, nuevo (nombres), nuevo (apellidos))  
nuevo (D.desechos) ← DESECHOS EN GENERAL (GRIS)  
**Mostrar** (Digite la cantidad (Kg))  
**Leer** (nuevo (P.cantidad\_kg))  
Digite el precio - venta (\$/.)  
**Leer** (nuevo (P.precio\_venta\_kg))  
nuevo (P.precio\_total) ← nuevo (P.cantidad\_kg) x nuevo (P.precio\_venta\_kg)  
**break**
- 2: Mostrar** (REGISTRO DE DESECHOS, desechos [1], DE, nuevo (nombres), nuevo (apellidos))  
nuevo (D.desechos) ← ORGÁNICOS (MORADO)  
**Mostrar** (Digite la cantidad (Kg))  
**Leer** (nuevo (P.cantidad\_kg))  
Digite el precio - venta (\$/.)  
**Leer** (nuevo (P.precio\_venta\_kg))  
nuevo (P.precio\_total) ← nuevo (P.cantidad\_kg) x nuevo (P.precio\_venta\_kg)  
**break**
- 3: Mostrar** (REGISTRO DE DESECHOS, desechos [2], DE, nuevo (nombres), nuevo (apellidos))  
nuevo (D.desechos) ← VIDRIOS (VERDE)  
**Mostrar** (Digite la cantidad (Kg))  
**Leer** (nuevo (P.cantidad\_kg))  
Digite el precio - venta (\$/.)  
**Leer** (nuevo (P.precio\_venta\_kg))  
nuevo (P.precio\_total) ← nuevo (P.cantidad\_kg) x nuevo (P.precio\_venta\_kg)  
**break**
- 4: Mostrar** (REGISTRO DE DESECHOS, desechos [3], DE, nuevo (nombres), nuevo (apellidos))  
nuevo (D.desechos) ← PLÁSTICOS Y ENVASES METÁLICOS (AMARILLO)  
**Mostrar** (Digite la cantidad (Kg))  
**Leer** (nuevo (P.cantidad\_kg))  
Digite el precio - venta (\$/.)  
**Leer** (nuevo (P.precio\_venta\_kg))  
nuevo (P.precio\_total) ← nuevo (P.cantidad\_kg) x nuevo (P.precio\_venta\_kg)  
**break**

- 5:** **Mostrar** (REGISTRO DE DESECHOS, desechos [4], DE, nuevo (nombres), nuevo (apellidos))  
nuevo (D.desechos) ← PAPEL (AZUL)  
**Mostrar** (Digite la cantidad (Kg))  
**Leer** (nuevo (P.cantidad\_kg))  
Digite el precio - venta (\$/.)  
**Leer** (nuevo (P.precio\_venta\_kg))  
nuevo (P.precio\_total) ← nuevo (P.cantidad\_kg) x nuevo (P.precio\_venta\_kg)  
**break**
- 6:** **Mostrar** (REGISTRO DE DESECHOS, desechos [5], DE, nuevo (nombres), nuevo (apellidos))  
nuevo (D.desechos) ← DESECHOS PELIGROSOS (ROJO)  
**Mostrar** (Digite la cantidad (Kg))  
**Leer** (nuevo (P.cantidad\_kg))  
Digite el precio - venta (\$/.)  
**Leer** (nuevo (P.precio\_venta\_kg))  
nuevo (P.precio\_total) ← nuevo (P.cantidad\_kg) x nuevo (P.precio\_venta\_kg)  
**break**

//Actualizar reciclador

nuevo (siguiente) ← NULL

**Si** primero<> NULL **entonces**

nuevo\_1 ← primero

**Mientras** nuevo\_1 (siguiente) <> NULL **hacer**

nuevo\_1 ← nuevo\_1 (siguiente)

nuevo\_1 (siguiente) ← nuevo

nuevo (anterior)= nuevo\_1

**De lo contrario**

primero ← nuevo

nuevo (anterior) ← NULL

**Fin** regis\_recicladores

## 2. FUNCIÓN VER RECICLADORES – INICIO A FIN

*// Función muestra los recicladores de Inicio a Fin*

**Inicio** ver\_recicladores (**Entero** numero\_recicladores)

**float** precio\_global = 0, Kg\_global = 0

ver\_r ← primero

**Para** i = 0 **hasta** i < numero\_recicladores **con paso** 1 **hacer**

**Mostrar** (DATOS PERSONALES DEL RECICLADOR, numero\_recicladores - i)

**Mostrar** (Nombres y apellidos: ver\_r (nombres), ver\_r (apellidos))

**Mostrar** (Nº DNI: ver\_r (dni))

**Mostrar** (Dirección:)

**Mostrar** (ver\_r(direccion.calle, ver\_r(direccion.numero), ver\_r(direccion.urbanizacion))

**Mostrar** (ver\_r (D.desechos))

**Mostrar** (CANTIDAD (Kg.), PRECIO - VENTA(\$/.), TOTAL(\$/.))

**Mostrar** (ver\_r (P.cantidad\_kg), ver\_r (P.precio\_venta\_kg), ver\_r (P.precio\_total)

Kg\_global ← Kg\_global + ver\_r (P.cantidad\_kg)

precio\_global ← precio\_global + ver\_r (P.precio\_total)

ver\_r ← ver\_r (siguiente)

**Mostrar** (RESULTADOS FINALES DE LOS, numero\_recicladores, RECICLADORES)

**Mostrar** (CANTIDAD GLOBAL (Kg.), TOTAL GLOBAL (\$/.))

**Mostrar** (Kg\_global, precio\_global)

**Fin** ver\_recicladores

### 3. FUNCIÓN VER RECICLADORES – FIN A INICIO

*// Función muestra los recicladores de Fin a Inicio*

**Inicio** ver\_recicladores\_fi

**float** precio\_global = 0, Kg\_global = 0

a ← primero

**Mientras** a (siguiente) <> NULL **hacer**

    a ← a (siguiente)

**Mientras** a <> NULL **hacer**

**Para** i =0 **hasta** i < numero\_recicladores **con** paso 1 **hacer**

**Mostrar** (DATOS PERSONALES DEL RECICLADOR, numero\_recicladores - i)

**Mostrar** (Nombres y apellidos: a (nombres), a (apellidos))

**Mostrar** (Nº DNI: a (dni))

**Mostrar** (Dirección:)

**Mostrar** (a (direccion.calle) a (direccion.numero), a (direccion.urbanizacion))

**Mostrar** (a (D.desechos))

**Mostrar** (CANTIDAD (Kg.), PRECIO - VENTA(\$/.), TOTAL(\$/.))

**Mostrar** (a (P.cantidad\_kg), a (P.precio\_venta\_kg), a (P.precio\_total))

        Kg\_global ← Kg\_global + a (P.cantidad\_kg)

        precio\_global ← precio\_global + a (P.precio\_total)

        a ← a (anterior)

**Mostrar** (RESULTADOS FINALES DE LOS, numero\_recicladores, RECICLADORES)

**Mostrar** (CANTIDAD GLOBAL (Kg.), TOTAL GLOBAL (\$/.))

**Mostrar** (Kg\_global, precio\_global)

**Fin** ver\_recicladores\_fi



#### 4. FUNCIÓN AGREGAR NUEVO RECICLADOR

##### //Agregar un nuevo reciclador al registro

**Inicio** anadir\_nuevo\_reciclador

**Mostrar** (Digite 'D' si agregara DESPUES de la posición)

**Mostrar** (Digite 'A' si agregara DESPUES de la posición)

**Hacer**

**Leer** (deter)

**Mientras** deter <> 'A' **y** deter <> 'a' **y** deter <> 'D' **y** deter <> 'd'

**Mostrar** (Datos Personales)

**Mostrar** (Digite el número de DNI del reciclador)

**Leer** (q (dni))

**Mostrar** (Digite los nombres del reciclador)

**Leer** (q (nombres))

**Mostrar** (Digite los apellidos del reciclador)

**Leer** (q (apellidos))

**Mostrar** (Digite los apellidos del reciclador)

**Leer** (q (apellidos))

**Mostrar** (Dirección de Vivienda)

**Mostrar** (Digite los nombres de la calle)

**Leer** (q (direccion.calle))

**Mostrar** (Digite el número de la casa)

**Leer** (q (direccion.numero))

**Mostrar** (Digite la urbanización o distrito)

**Leer** (q (direccion.urbanizacion))

**Mostrar** (Registro de Desechos)

**Mostrar** (1. DESECHOS EN GENERAL (GRIS))

**Mostrar** (2. ORGÁNICOS (MORADO))

**Mostrar** (3. VIDRIOS (VERDE))

**Mostrar** (4. PLÁSTICOS Y ENVASES METÁLICOS (AMARILLO))

**Mostrar** (5. PAPEL (AZUL))

**Mostrar** (6. DESECHOS PELIGROSOS (ROJO))

**Mostrar** (Digite una opción)

## Según opc hacer

- 1:**     **Mostrar** (REGISTRO DE DESECHOS, desechos [0], DE, q (nombres), q (apellidos))  
          q (D.desechos) ← DESECHOS EN GENERAL (GRIS)  
          **Mostrar** (Digite la cantidad (Kg))  
          **Leer** (q (P.cantidad\_kg))  
          Digite el precio - venta (\$/.)  
          **Leer** (q (P.precio\_venta\_kg))  
          q (P.precio\_total) ← q (P.cantidad\_kg) x q (P.precio\_venta\_kg)  
          **break**
- 2:**     **Mostrar** (REGISTRO DE DESECHOS, desechos [1], DE, q (nombres), q (apellidos))  
          q (D.desechos) ← ORGÁNICOS (MORADO)  
          **Mostrar** (Digite la cantidad (Kg))  
          **Leer** (q (P.cantidad\_kg))  
          Digite el precio - venta (\$/.)  
          **Leer** (q (P.precio\_venta\_kg))  
          q (P.precio\_total) ← q (P.cantidad\_kg) x q (P.precio\_venta\_kg)  
          **break**
- 3:**     **Mostrar** (REGISTRO DE DESECHOS, desechos [2], DE, q (nombres), q (apellidos))  
          q (D.desechos) ← VIDRIOS (VERDE)  
          **Mostrar** (Digite la cantidad (Kg))  
          **Leer** (q (P.cantidad\_kg))  
          Digite el precio - venta (\$/.)  
          **Leer** (q (P.precio\_venta\_kg))  
          q (P.precio\_total) ← q (P.cantidad\_kg) x q (P.precio\_venta\_kg)  
          **break**
- 4:**     **Mostrar** (REGISTRO DE DESECHOS, desechos [3], DE, q (nombres), q (apellidos))  
          q (D.desechos) ← PLÁSTICOS Y ENVASES METÁLICOS (AMARILLO)  
          **Mostrar** (Digite la cantidad (Kg))  
          **Leer** (q (P.cantidad\_kg))  
          Digite el precio - venta (\$/.)  
          **Leer** (q (P.precio\_venta\_kg))  
          q (P.precio\_total) ← q (P.cantidad\_kg) x q (P.precio\_venta\_kg)  
          **break**

**5:**    **Mostrar** (REGISTRO DE DESECHOS, desechos [4], DE, q (nombres), q (apellidos))  
q (D.desechos) ← PAPEL (AZUL)  
**Mostrar** (Digite la cantidad (Kg))  
**Leer** (q (P.cantidad\_kg))  
Digite el precio - venta (\$/.)  
**Leer** (q (P.precio\_venta\_kg))  
q (P.precio\_total) ← q (P.cantidad\_kg) × q (P.precio\_venta\_kg)  
**break**

**6:**    **Mostrar** (REGISTRO DE DESECHOS, desechos [5], DE, q (nombres), q (apellidos))  
q (D.desechos) ← DESECHOS PELIGROSOS (ROJO)  
**Mostrar** (Digite la cantidad (Kg))  
**Leer** (q (P.cantidad\_kg))  
Digite el precio - venta (\$/.)  
**Leer** (q (P.precio\_venta\_kg))  
q (P.precio\_total) ← q (P.cantidad\_kg) × q (P.precio\_venta\_kg)  
**break**

*//Actualización de la lista con los nuevos datos del reciclador ANTES o DESPUÉS de la posición indicada*

**Si** pos > 0 **y** pos <= numero\_recicladores + 1 **entonces**

**Leer** (pos)

**Si** deter = 'A' o deter = 'a' **y** pos = 1 **entonces** *//Añadir antes del primer reciclador*

        q (siguiente) ← primero

        q (anterior) ← NULL

        q (siguiente(anterior)) ← q

        primero ← q

**De lo contrario**

**Si** deter = 'D' o deter = 'd' **y** pos = 1 **entonces** *//Añadir después del primer reciclador*

            q (siguiente) ← primero (siguiente)

            q (anterior) ← primero

            q (siguiente(anterior)) ← q

            primero (siguiente) ← q

**De lo contrario**

            t ← primero

**Para**  $i = 1$  **hasta** primero  $\neq$  NULL **con paso** 1 **hacer**

*// Para pos  $\neq$  del primero añade antes*

**Si** deter = 'A' **o** deter = 'a' **y**  $i = pos - 1$  **entonces**

q (siguiente)  $\leftarrow$  t (siguiente)

q (anterior)  $\leftarrow$  t

q (siguiente(anterior))  $\leftarrow$  q

t (siguiente)  $\leftarrow$  q

break

*// Para pos  $\neq$  del primero añade después*

**Si** deter = 'D' **o** deter = 'd' **y**  $i = pos$  **entonces**

q (siguiente)  $\leftarrow$  t (siguiente)

q (anterior)  $\leftarrow$  t

**Si** q (siguiente)  $\neq$  NULL **entonces**

q (siguiente(anterior))  $\leftarrow$  q

t (siguiente)  $\leftarrow$  q

break

t  $\leftarrow$  t (siguiente)

numero\_recicladores  $\leftarrow$  numero\_recicladores + 1

**Mostrar** (El reciclador ha sido agregado al registro)

**Fin** anadir\_nuevo\_reciclador

## 5. FUNCIÓN BUSCA EL DNI DE UN RECICLADOR Y MUESTRA SUS DATOS

**Función** dni\_un\_reciclador (Entero num\_dni\_recicla) // Muestra el registro de un reciclador y desecho según dni.

**Leer** (num\_dni\_recicla)

$i \leftarrow 1$  //  $i$ =iterador

$k \leftarrow 0$  //  $k$ =bandera

dni\_r  $\leftarrow$  primero

**Si** primero  $\neq$  NULL

**Hacer**

**Si** dni\_r(dni)  $\leftarrow$  num\_dni\_recicla // evalúa la igualdad entre el dni ingresado con los registrados.

**Mostrar** DATOS PERSONALES DEL RECICLADOR  $i$

**Mostrar** Nombres y apellidos : dni\_r ( nombres) dni\_r (apellidos)

**Mostrar** N° DNI: dni\_r (dni)

**Mostrar** Dirección: dni\_r ( direccion.calle) dni\_r( direccion.numero) dni\_r (direccion.urbanizacion )

**Mostrar** dni\_r (D.desechos)

**Mostrar** CANTIDAD (Kg.) PRECIO - VENTA(\$/.) TOTAL(\$/.)

**Mostrar** dni\_r ( P.cantidad\_kg )

**Mostrar** dni\_r ( P.precio\_venta\_kg)

**Mostrar** dni\_r (P.precio\_total)

$k \leftarrow 1$

dni\_r  $\leftarrow$  dni\_r (siguiente) //actualiza al siguiente nodo

$i \leftarrow i+1$

**Mientras** ((dni\_r  $\neq$  NULL) y ( $k = 0$ ))

**Si** ( $k = 0$ )//evalúa si no existe ningún dni igual al ingresado en el registro.

**Mostrar** El reciclador no fue encontrado.

**FIN**

## 6. FUNCIÓN MAYOR VENTA DE TODOS LOS RECICLADORES

**Función** mayor\_venta ( **Entero** numero\_recicladores ) **//Función que muestra la mayor venta**

i ← 1 **// i=iterador**

k ← 0 **//k=bandera**

**Flotante** mayor

may\_venta\_r ← primero

may\_venta\_reciclador ← primero

mayor ← may\_venta\_r ( P.precio\_total )

**Para** i ← 0 **Hasta** i < numero\_recicladores **Con paso 1 Hacer**

**Si** may\_venta\_r ( P.precio\_total ) > mayor

mayor ← may\_venta\_r ( P.precio\_total )

may\_venta\_r ← may\_venta\_r ( siguiente ) **//actualiza al siguiente nodo**

**Si** primero <> NULL

**Hacer**

**Si** may\_venta\_reciclador ( P.precio\_total ) = mayor **//busca el precio total y muestra datos**

**Mostrar** DATOS PERSONALES DEL RECICLADOR i

**Mostrar** Nombres y apellidos : may\_venta\_reciclador ( nombres) may\_venta\_reciclador (apellidos)

**Mostrar** N° DNI: may\_venta\_reciclador (dni )

**Mostrar** Dirección: may\_venta\_reciclador ( direccion.calle) may\_venta\_reciclador ( direccion.numero)

**Mostrar** may\_venta\_reciclador (direccion.urbanizacion )

**Mostrar** may\_venta\_reciclador (D.desechos)

**Mostrar** CANTIDAD (Kg.) PRECIO - VENTA(\$/.) TOTAL(\$/.)

**Mostrar** may\_venta\_reciclador ( P.cantidad\_kg )

**Mostrar** may\_venta\_reciclador ( P.precio\_venta\_kg)

**Mostrar** may\_venta\_reciclador (P.precio\_total)

k ← 1

may\_venta\_reciclador ← may\_venta\_reciclador (siguiente) **//actualiza al siguiente nodo**

i ← i+1

**Mientras** ((may\_venta\_reciclador <> NULL) y (k = 0))

**FIN**

## 7. FUNCIÓN MENOR VENTA DE TODOS LOS RECICLADORES

**Función** menor\_venta ( **Entero** numero\_recicladores ) // **Función que muestra la menor venta**

$i \leftarrow 1$  //  $i = \text{iterador}$

$k \leftarrow 0$  //  $k = \text{bandera}$

**Flotante** menor

men\_venta\_r  $\leftarrow$  primero

men\_venta\_reciclador  $\leftarrow$  primero

menor  $\leftarrow$  may\_venta\_r ( P.precio\_total )

**Para**  $i \leftarrow 0$  **Hasta**  $i < \text{numero\_recicladores}$  **Con paso 1 Hacer**

**Si** men\_venta\_r ( P.precio\_total )  $<$  menor

menor  $\leftarrow$  men\_venta\_r ( P.precio\_total )

men\_venta\_r  $\leftarrow$  men\_venta\_r ( siguiente ) // **actualiza al siguiente nodo**

**Si** primero  $\neq$  NULL

**Hacer**

**Si** men\_venta\_reciclador ( P.precio\_total ) = menor // **busca el precio total y muestra datos**

**Mostrar** DATOS PERSONALES DEL RECICLADOR  $i$

**Mostrar** Nombres y apellidos : men\_venta\_reciclador ( nombres ) men\_venta\_reciclador ( apellidos )

**Mostrar** Nº DNI: men\_venta\_reciclador ( dni )

**Mostrar** Dirección: men\_venta\_reciclador ( direccion.calle ) men\_venta\_reciclador ( direccion.numero )

**Mostrar** men\_venta\_reciclador ( direccion.urbanizacion )

**Mostrar** men\_venta\_reciclador ( D.desechos )

**Mostrar** CANTIDAD (Kg.) PRECIO - VENTA(\$/.) TOTAL(\$/. )

**Mostrar** men\_venta\_reciclador ( P.cantidad\_kg )

**Mostrar** men\_venta\_reciclador ( P.precio\_venta\_kg )

**Mostrar** men\_venta\_reciclador ( P.precio\_total )

$k \leftarrow 1$

men\_venta\_reciclador  $\leftarrow$  men\_venta\_reciclador ( siguiente ) // **actualiza al siguiente nodo**

$i \leftarrow i + 1$

**Mientras** ((may\_venta\_reciclador  $\neq$  NULL) y (  $k = 0$  ))

**FIN**

## 8. FUNCIÓN ELIMINA UN RECICLADOR EN ESPECIFICO POR SU DNI

**Función** eliminar\_reciclador ( **Entero** num\_dni\_eliminar\_recicla) // **Función que elimina un reciclador de la lista**

$i \leftarrow 1$  // **i=iterador**

$k \leftarrow 0$  // **k=bandera**

**Si** primero  $\neq$  NULL

eliminar\_r  $\leftarrow$  primero;

**Mientras** eliminar\_r  $\neq$  NULL

**Si** eliminar\_r ( dni ) = num\_dni\_eliminar\_recicla // **busca el dni en la lista**

**Si** eliminar\_r = primero

primero  $\leftarrow$  primero (siguiente)

**delete** (eliminar\_r) // **elimina el nodo que contiene el dni**

eliminar\_r  $\leftarrow$  primero;

$k \leftarrow 1$

**De lo contrario**

eliminar\_r (anterior(siguiente)) = eliminar\_r (siguiente)

**Si** eliminar\_r (siguiente)  $\neq$  NULL

eliminar\_r (siguiente(anterior))  $\leftarrow$  eliminar\_r (anterior)

eliminar\_r\_m  $\leftarrow$  eliminar\_r

eliminar\_r  $\leftarrow$  eliminar\_r (siguiente)

**delete**(eliminar\_r\_m) // **elimina el nodo que contiene el dni**

$k \leftarrow 1$

**De lo contrario**

eliminar\_r  $\leftarrow$  eliminar\_r (siguiente) // **actualiza al siguiente nodo**

**Si**  $k = 0$  // **evalúa si no existe ningún dni igual al ingresado en el registro.**

**Mostrar** El número de DNI del reciclador no fue encontrado.

**De lo contrario**

**Mostrar** El reciclador ha sido eliminado.

numero\_recicladores  $\leftarrow$  numero\_recicladores - 1 // **actualiza # de recicladores**



## 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 segunda unidad, como listas doblemente enlazadas simples, 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/iHfpwznwvjts/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=2ahUKEwiyrtGqqq3sAhUGmVkKHfeVAm8QFjAKegQIAxAC&url=https%3A%2F%2Fwww.editorialpatria.com.mx%2Fpdf%2F9786074384154.pdf&usg=AOvVaw2FoIqncneS032eQder37se>