



**Universidad Nacional Autónoma de México**  
**Facultad de ingeniería**



Fundamentos de programación

Proyecto Final. Cambio de una máquina expendedora.

Semestre 2022-1

Profesor: Marco Antonio Martínez Quintana

Nombre de la alumna: González Arroyo Paulina Montserrat

Fecha: 03/diciembre/2021

# ÍNDICE.

INTRODUCCIÓN.....	pág. 3
DIAGRAMA DE FLUJO.....	pág. 6
CÓDIGO.....	pág. 7
RESULTADOS DEL PROYECTO.....	pág. 8
TABLA DE RECURSOS INFORMÁTICOS [SOFTWARE Y HARDWARE] NECESARIOS.....	pág. 8
TABLA DE COSTOS ASOCIADOS AL PROYECTO.....	pág. 9
DIAGRAMA DE GANTT.....	pág. 9
URL DEL VIDEO.....	pág. 9
REPOSITORIO DE GITHUB DEL PROYECTO FINAL.....	pág. 9
MANUAL DE USUARIO DEL PROYECTO.....	pág. 10
CONCLUSIONES.....	pág. 15
REFERENCIA.....	pág. 17
GLOSARIO.....	pág. 18
ACRÓNIMOS O ABREVIATURAS UTILIZADAS.....	pág. 18

- **Introducción**

El presente proyecto final se basa en la problemática de una empresa que fabrica un modelo de máquinas expendedoras de refrescos la cual necesita un programa para estas máquinas que realice el cálculo de cuántas monedas cada tipo debe devolver. Para ello, en primer lugar, se introducirá por teclado la cantidad a devolver en monedas (múltiplo de 20 pesos, qué es la moneda más grande de la que se dispone), es decir, se tecleará 22 para 1 moneda de \$20 con 1 moneda de \$2. Este programa describirá en pantalla cuantas monedas de cada tipo hay que devolver teniendo en cuenta que:

- Se dispone de monedas de 20 pesos, 10 pesos, 5 pesos, 2 pesos y 1 peso.
- Siempre se dispone de las monedas necesarias de cada tipo.
- Se debe devolver el menor número de monedas posible, es decir, intentar devolver con las de mayor valor.

Ejemplos:

- ❖ Si se introduce la cantidad de 93 pesos, el programa debe imprimir: 4 monedas de 20 pesos, 1 moneda de 10 pesos, 0 monedas de 5 pesos, 1 moneda de 2 pesos y 1 moneda de 1 peso.
- ❖ Si se introduce la cantidad de 78 pesos, el programa debe imprimir: 3 monedas de 20 pesos, 1 moneda de 10 pesos, 1 monedas de 5 pesos, 1 moneda de 2 pesos y 1 moneda de 1 peso.

Este problema consiste en determinar cuántas monedas de cada tipo se necesitan para formar la cantidad que debe devolver la máquina expendedora, teniendo en cuenta que se debe utilizar el menor número posible de monedas de cada tipo. Esto supone que, hay un único dato de entrada (la cantidad a devolver), mientras que se tienen varios datos de salida (el número de monedas de cada tipo). Así:

- ❖ Datos de entrada: cantidad a devolver.
- ❖ Datos de salida: cuántas monedas de MXN\$ 20, cuántas monedas de MXN\$ 10, cuántas monedas de MXN\$ 5, cuántas monedas de MXN\$ 2, y cuantas monedas de un peso se necesitan.
- ❖ Proceso: determinar cuántas monedas de cada tipo hacen falta para formar la cantidad a devolver.

Una vez determinados los datos de entrada y de salida, habrá que pensar cómo es posible calcular el número de monedas de cada tipo que hacen falta. Un dato muy importante es el hecho de que se desea utilizar el menor número de monedas posible, lo que indica que se deben comenzar a emplear las monedas de mayor valor. Así, supongamos que la cantidad a devolver es MXN\$ 93. Si se divide esta cantidad por 20 (qué es el valor de la moneda de MXN\$ 20) se obtiene 4.65, Lo que significa que se necesitan cuatro monedas de MXN\$ 20 y todavía queda cantidad

por devolver. Es decir, ya se han devuelto MXN\$ 80 con cuatro monedas de MXN\$ 20 y quedan MXN\$ 13 ( $93 - 4 \cdot 20$ ) por devolver. Ahora si se divide MXN\$ 13 entre MXN\$ 10 (la moneda de 10 pesos), se obtiene 1.3, Por lo que se utiliza una moneda de MXN\$ 10 y todavía quedan MXN\$ 3 ( $13 - 10 \cdot 1$ ), El resultado es MXN\$ 3 por devolver, que al dividirlo por 2 (la moneda de MXN\$ 2), se obtiene 1.5, es decir una moneda de MXN\$ 2 y una moneda de un peso y ya no queda nada por devolver.

Este proceso puede resumirse de la siguiente forma:

- Dividir la cantidad a devolver por 20 para obtener el número de monedas de MXN\$ 20 a utilizar.
- Calcular lo que queda por devolver restando la cantidad a devolver menos el número de monedas de MXN\$ 20 utilizadas por 20 (que es su valor en pesos)
- Dividir la cantidad que queda por devolver por MXN\$ 10 para obtener el número de monedas de MXN\$ 10 a utilizar.
- Calcular lo que queda por devolver restando lo que quedaba por devolver menos el número de monedas de MXN\$ 10 utilizadas por 10 (que es su valor en pesos)
- Dividir la cantidad que queda por 5 para obtener el número de monedas de MXN\$ 5 a utilizar.
- Calcular lo que queda por devolver restando lo que quedaba por devolver menos el número de monedas de MXN\$ 5 utilizadas por 5 (que es su valor en pesos)
- Dividir la cantidad que queda por 2 para obtener el número de monedas de MXN\$ 2 a utilizar.
- Calcular lo que queda por devolver restando lo que quedaba por devolver menos el número de monedas de MXN\$ 2 utilizadas por 2 (que es su valor en pesos)
- Dividir la cantidad que queda por uno para obtener el número de monedas de un peso a utilizar.
- Calcular lo que queda por devolver restando lo que quedaba por devolver menos el número de monedas de un peso utilizadas por uno (que es su valor en pesos)

Por lo tanto, el programa deberá:

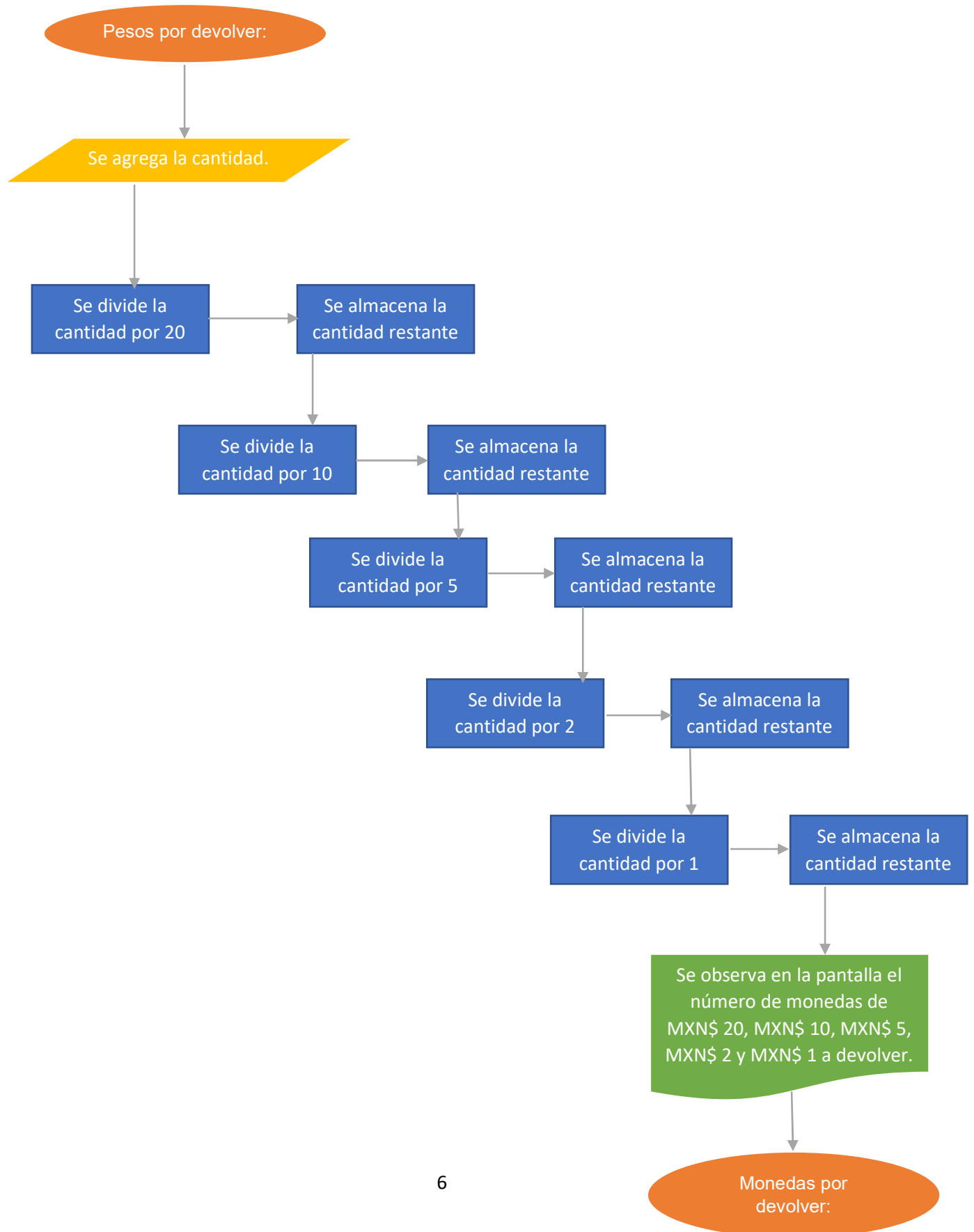
1. Leer la cantidad a devolver en pesos y almacenarlo en una variable pesos\_a\_devolver. copiar el contenido de pesos\_a\_devolver en la variable falta\_por\_devolver, ya que todavía queda todo por devolver.
2. Dividir la variable falta\_por\_devolver entre 20 y almacenar la parte interna de la división en la variable monedas\_\$20.
3. Obtener la cantidad que queda por devolver restando la variable falta\_por\_devolver menos la variable monedas\_\$20 por 20. Almacenar el resultado variable falta\_por\_devolver.

4. Dividir la variable `falta_por_devolver` entre 10 y almacenar la parte entera de la división en la variable `monedas_$10`.
5. Obtener la cantidad que queda por devolver restando la variable `falta_por_devolver` una variable `monedas_$10`.
6. Dividir la variable `falta_por_devolver` entre 5 y almacenar la parte entera de la división en la variable `monedas_$5`.
7. Obtener la cantidad que queda por devolver restando la variable `falta_por_devolver` una variable `monedas_$5`.
8. Dividir la variable `falta_por_devolver` entre 2 y almacenar la parte entera de la división en la variable `monedas_$2`.
9. Obtener la cantidad que queda por devolver restando la variable `falta_por_devolver` una variable `monedas_$2`.
10. Dividir la variable `falta_por_devolver` entre 1 y almacenar la parte entera de la división en la variable `monedas_$1`.
11. Escribir en pantalla el contenido de las variables `monedas_$20`, `monedas_$10`, `monedas_$5`, `monedas_$2`, `monedas_$1`.

Determinando el tipo de las variables anteriores:

- `Pesos_a_devolver` y `falta_por_devolver` contendrán la cantidad en pesos que queda por devolver inicialmente y después de cada cálculo, que son datos de tipo real y por lo que se deben declarar el tipo `float`.
- Las variables `monedas_$20`, `monedas_$10`, `monedas_$5`, `monedas_$2`, `monedas_$1` almacenan el número de monedas de cada tipo que se van a utilizar, es decir, datos de tipo entero, por lo que se deben declarar de tipo `int` (no superarán el rango máximo de este tipo)

### Diagrama de flujo:



## MaquinaExpendedora.c

```
#include <stdio.h>

/*Programa que calcula el cambio de una máquina expendedora*/

int main ()
{
    float pesos_a_devolver, falta_por_devolver;
    int monedas_$20, monedas_$10, monedas_$5, monedas_$2, monedas_$1;

    /*Lectura de datos: pesos_a_devolver */

    printf ("Pesos a devolver: ");
    scanf ("%f", &pesos_a_devolver);

    /* Proceso */

    falta_por_devolver=pesos_a_devolver;

    monedas_$20=falta_por_devolver/20;
    falta_por_devolver=falta_por_devolver-20*monedas_$20;

    monedas_$10=falta_por_devolver/10;
    falta_por_devolver=falta_por_devolver-10*monedas_$10;

    monedas_$5=falta_por_devolver/5;
    falta_por_devolver=falta_por_devolver-5*monedas_$5;

    monedas_$2=falta_por_devolver/2;
    falta_por_devolver=falta_por_devolver-2*monedas_$2;

    monedas_$1=falta_por_devolver/1;
    falta_por_devolver=falta_por_devolver-1*monedas_$1;

    /*Salida*/

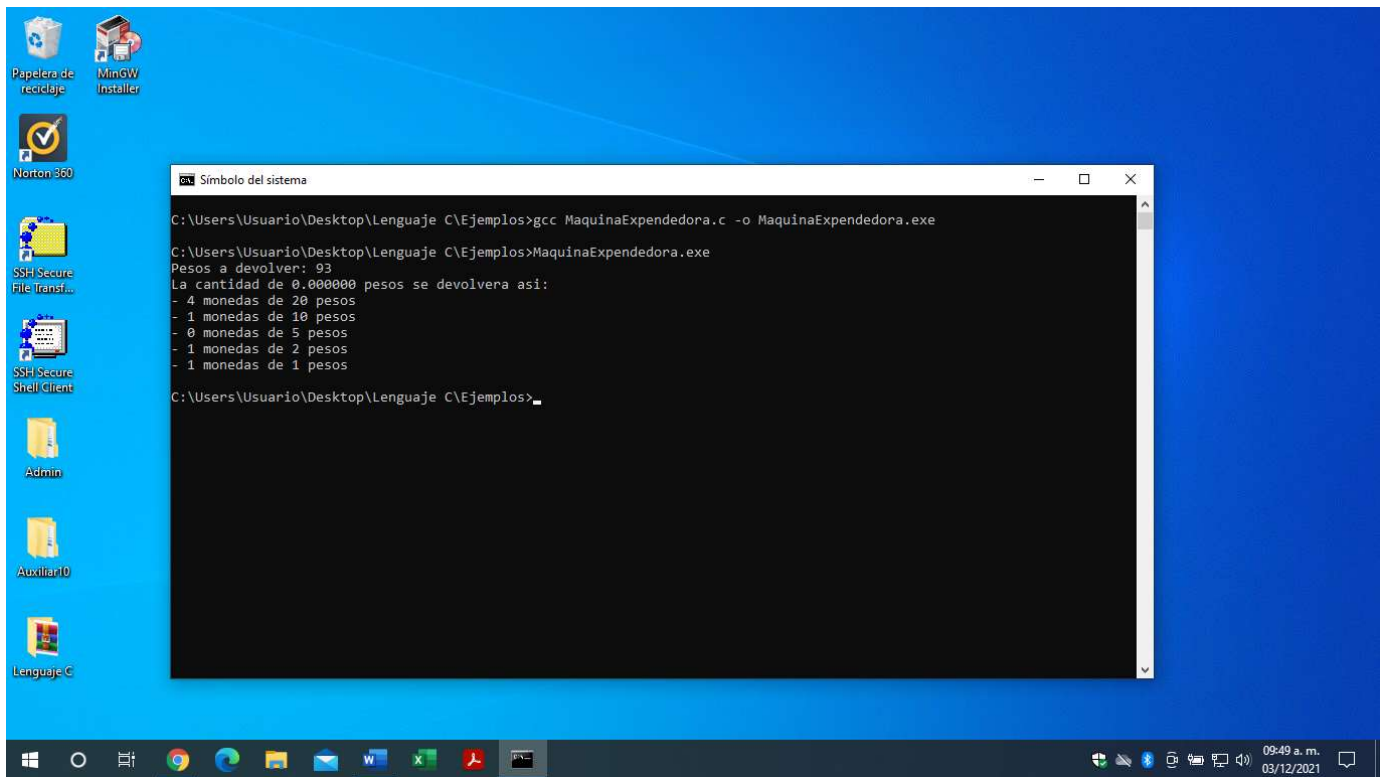
    printf ("La cantidad de %f pesos se devolvera asi:\n", "monedas_a_devolver");

    printf ("- %d monedas de 20 pesos\n", monedas_$20);
    printf ("- %d monedas de 10 pesos\n", monedas_$10);
    printf ("- %d monedas de 5 pesos\n", monedas_$5);
    printf ("- %d monedas de 2 pesos\n", monedas_$2);
    printf ("- %d monedas de 1 pesos\n", monedas_$1);

}
```

- **Resultados del proyecto**

Capturas de pantalla del funcionamiento de su proyecto



**Tabla de recursos informáticos [software y hardware] necesarios**

<b>SOFTWARE</b>	Microsoft Windows
	Google Chrome
	Google classroom
	Notepad++
	Youtube
	Microsoft Office Word
	Microsoft Office Excel
	Zoom
	Adobe Photoshop
<b>HARDWARE</b>	Monitores
	Teclado
	Mouse
	Módem
	Audífonos manos libres



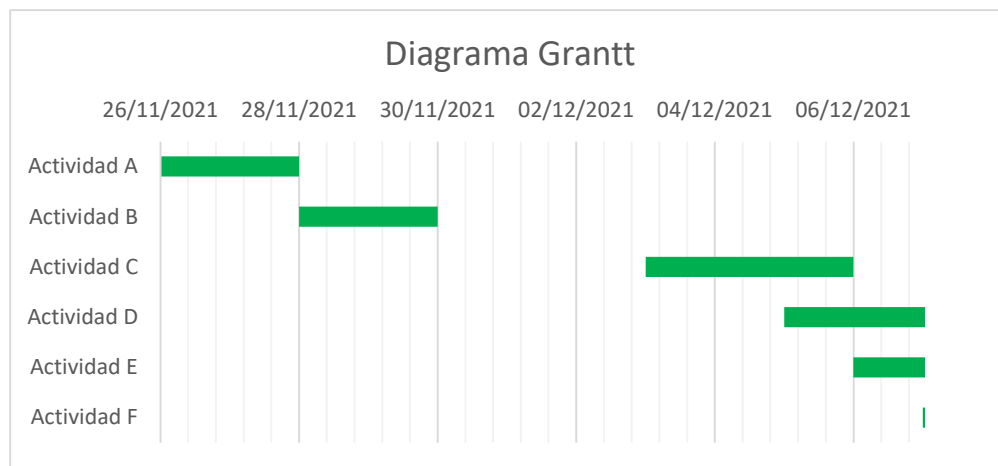
## Tabla de costos asociados al proyecto

<b>Tiempo requerido:</b> 26 de nov .2021: 2 horas 28 de nov. 2021: 2 horas 3 de dic. 2021: 3 horas 5 de diciembre: 3 horas	<b>SIN COSTO</b>
<b>Equipo de trabajo requerido:</b> Monitor, teclado, mouse, módem, audífonos manos libres.	<b>SIN COSTO</b>
<b>Conocimiento requerido:</b> Prácticas 0-12 de Fundamentos de programación	<b>SIN COSTO</b>

Costo final: Gratis.

El proyecto es gratuito.

## Diagrama de Gantt



## URL del video

[https://youtu.be/Fa4G\\_cmz38U](https://youtu.be/Fa4G_cmz38U)

## Repositorio de GitHub del Proyecto Final

<https://github.com/pausgr/ProyectoFinal>.

## Manual de usuario del proyecto

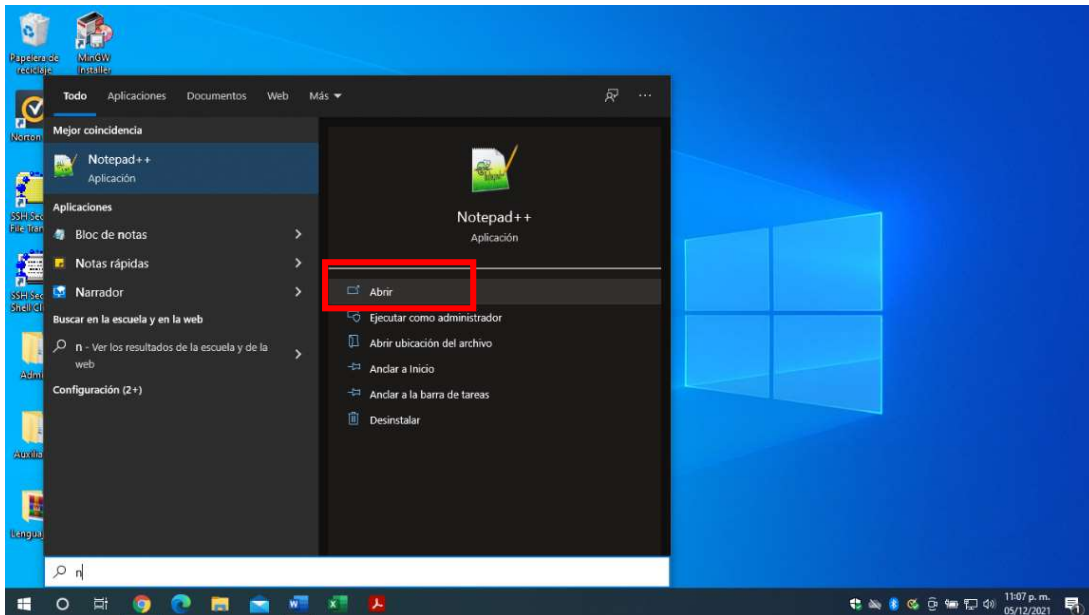
Este manual tiene como finalidad el manejo del programa de proyecto presente.

Requerimientos:

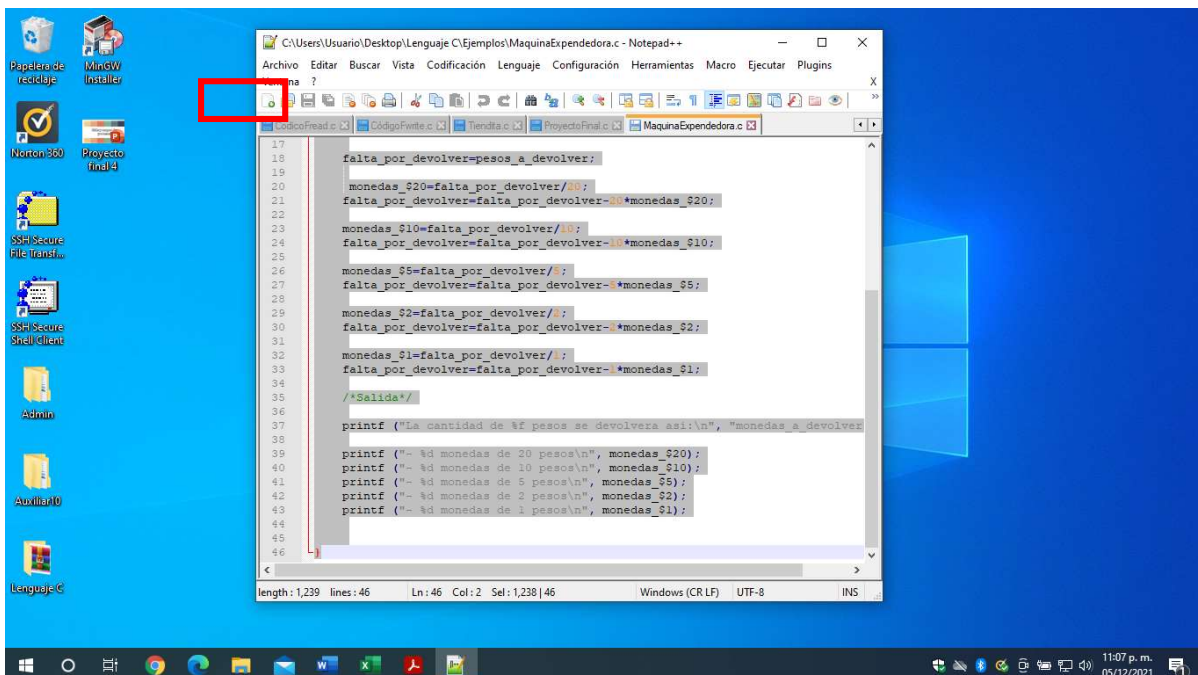
- MingGW
- Notepad++

Acceso.

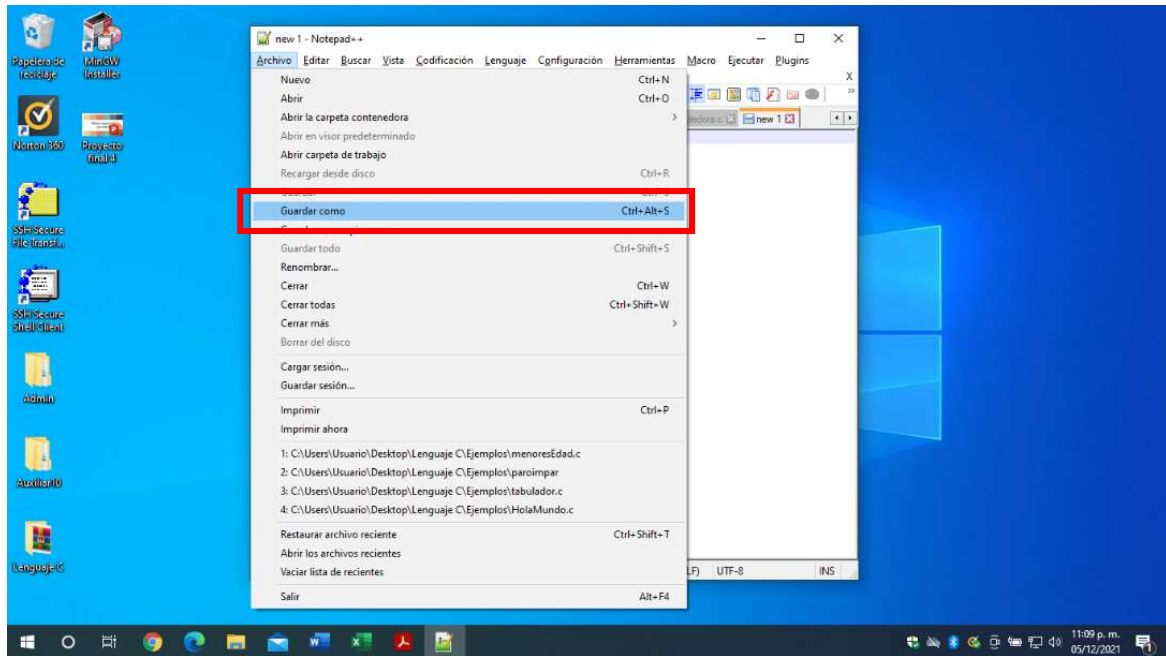
1. Ingresar a “Notepad++”



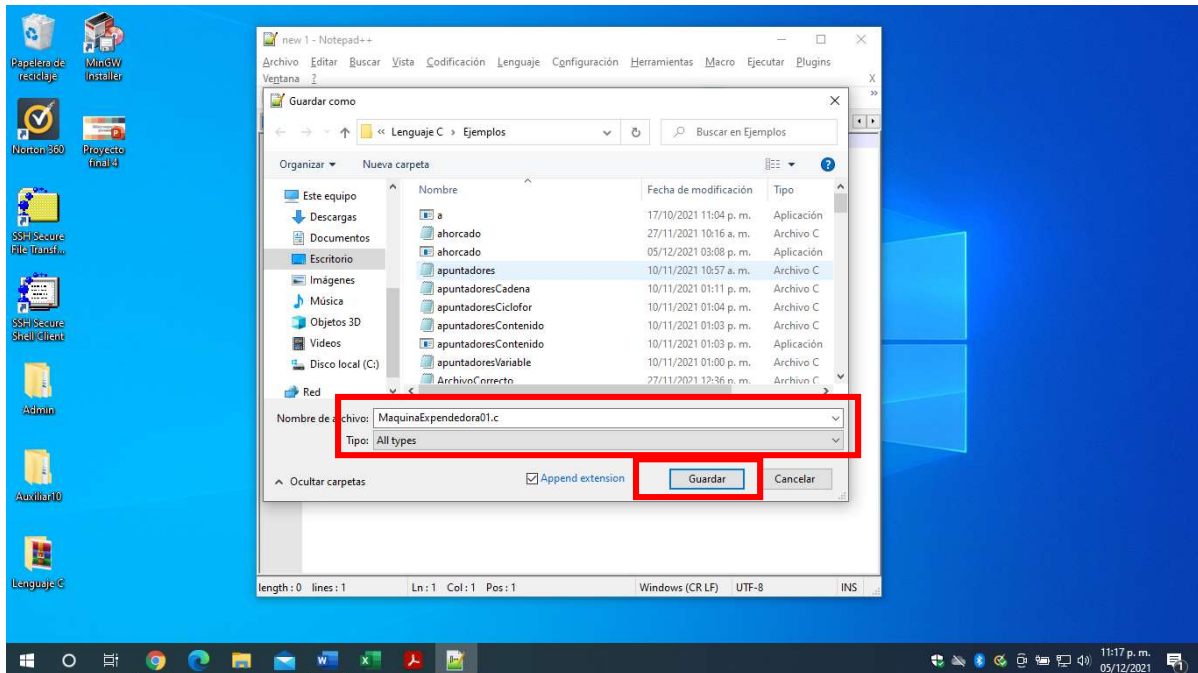
2. Abrir una nueva nota.



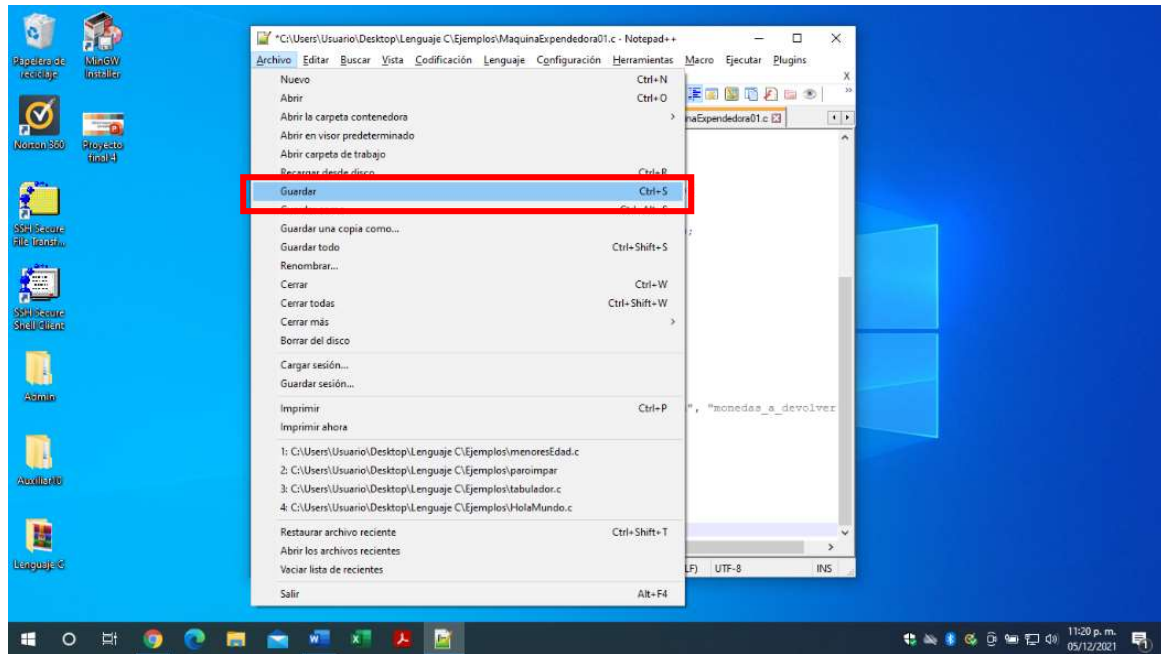
3. Seleccionar “Archivo”, y “guardar como”, o el su caso “Ctrl+Alt+S”.



4. Guardar el documento con la terminación “.c” y en Tipo poner “All types”.



5. Escribir el siguiente código para después guardarlo con la tecla “Ctrl+S”



```
#include <stdio.h>

/*Programa que calcula el cambio de una máquina expendedora*/

int main ()
{
    float pesos_a_devolver, falta_por_devolver;
    int monedas_$20, monedas_$10, monedas_$5, monedas_$2, monedas_$1;

    /*Lectura de datos: pesos_a_devolver */

    printf ("Pesos a devolver: ");
    scanf ("%f", &pesos_a_devolver);

    /* Proceso */

    falta_por_devolver=pesos_a_devolver;

    monedas_$20=falta_por_devolver/20;
    falta_por_devolver=falta_por_devolver-20*monedas_$20;

    monedas_$10=falta_por_devolver/10;
    falta_por_devolver=falta_por_devolver-10*monedas_$10;

    monedas_$5=falta_por_devolver/5;
    falta_por_devolver=falta_por_devolver-5*monedas_$5;

    monedas_$2=falta_por_devolver/2;
    falta_por_devolver=falta_por_devolver-2*monedas_$2;

    monedas_$1=falta_por_devolver/1;
    falta_por_devolver=falta_por_devolver-1*monedas_$1;

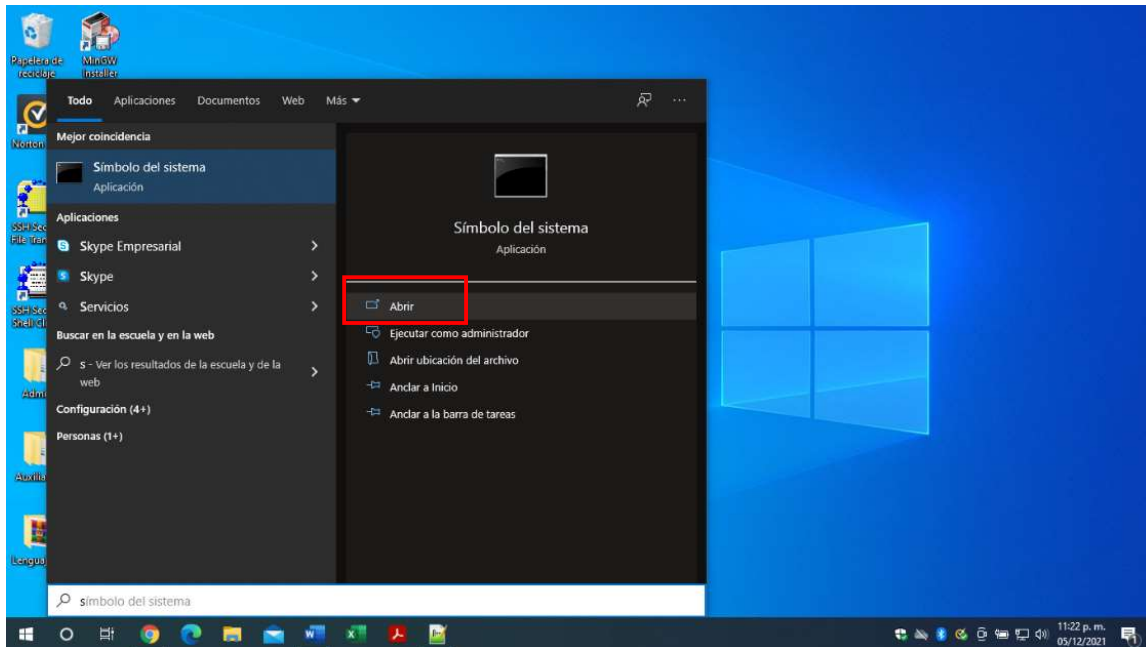
    /*Salida*/

    printf ("La cantidad de %f pesos se devolvera asi:\n",
"monedas_a_devolver");

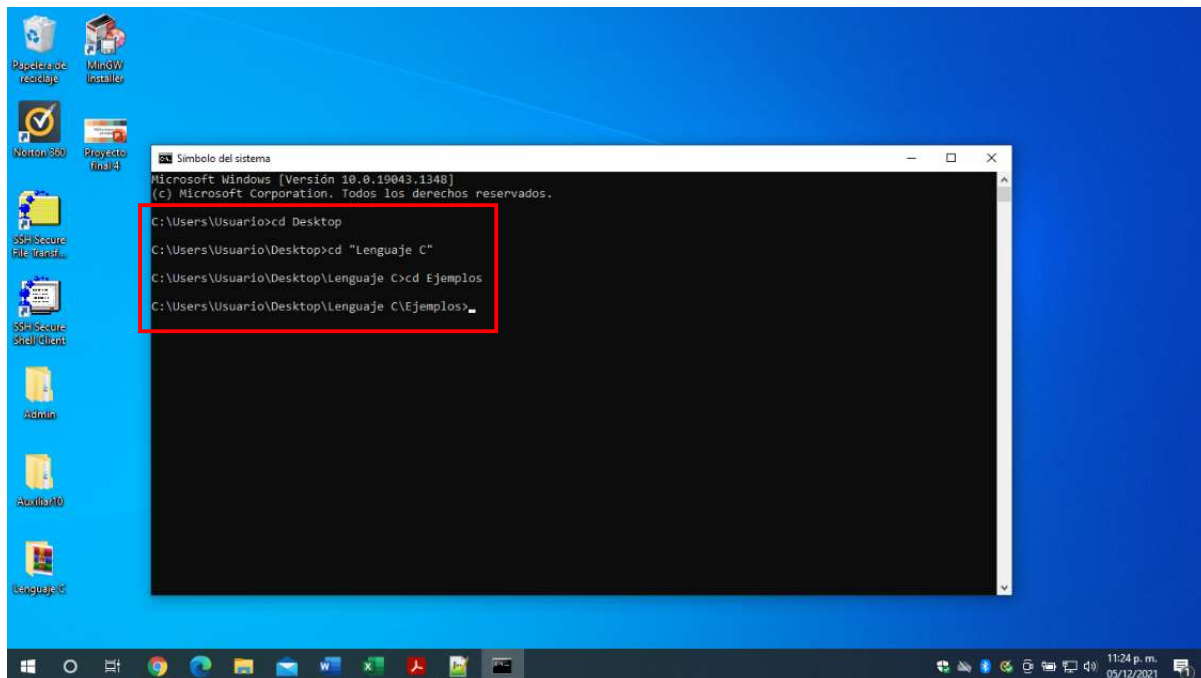
    printf ("- %d monedas de 20 pesos\n", monedas_$20);
    printf ("- %d monedas de 10 pesos\n", monedas_$10);
    printf ("- %d monedas de 5 pesos\n", monedas_$5);
    printf ("- %d monedas de 2 pesos\n", monedas_$2);
    printf ("- %d monedas de 1 pesos\n", monedas_$1);

}
```

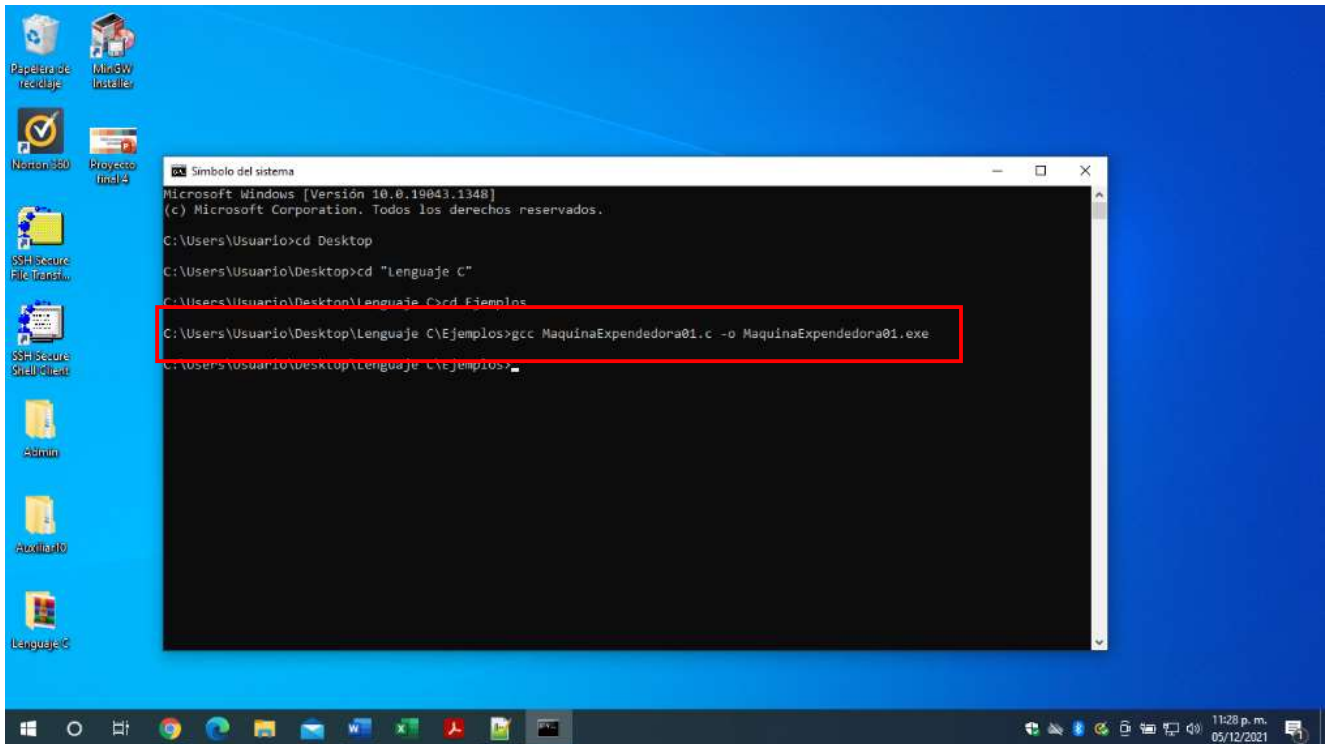
6. Abrir “Símbolo de Sistema” desde tu ordenador.



7. Ingresar a la carpeta donde se guardó la nota, implementando “cd”.



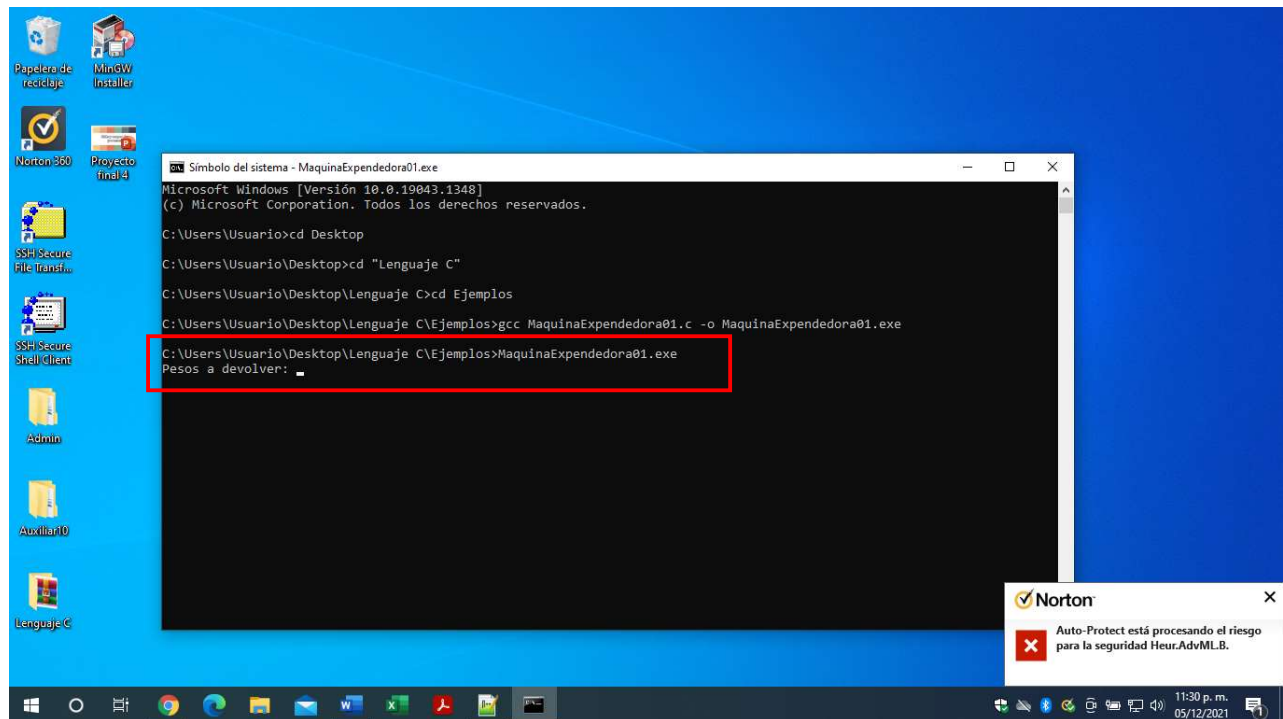
8. Una vez estando dentro de la carpeta escribir: **gcc “Nombre\_de\_la\_nota.c” -o “Nombre\_de\_la\_nota.exe”**



```
Microsoft Windows [Versión 10.0.19043.1348]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Usuario>cd Desktop
C:\Users\Usuario\Desktop>cd "Lenguaje C"
C:\Users\Usuario\Desktop\Lenguaje C>cd Ejemplos
C:\Users\Usuario\Desktop\Lenguaje C\Ejemplos>gcc MaquinaExpendedora01.c -o MaquinaExpendedora01.exe
C:\Users\Usuario\Desktop\Lenguaje C\Ejemplos>
```

9. Escribir lo siguiente: **Nombre\_de\_la\_nota.exe**

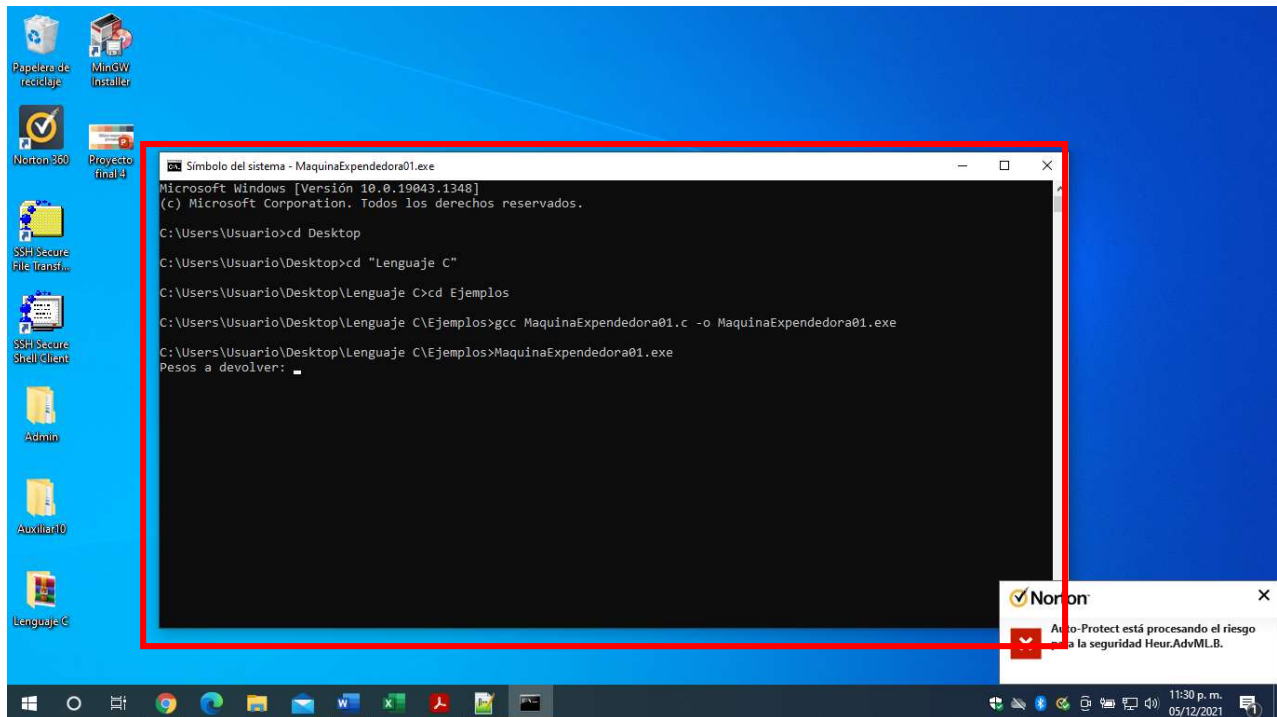


```
Microsoft Windows [Versión 10.0.19043.1348]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Usuario>cd Desktop
C:\Users\Usuario\Desktop>cd "Lenguaje C"
C:\Users\Usuario\Desktop\Lenguaje C>cd Ejemplos
C:\Users\Usuario\Desktop\Lenguaje C\Ejemplos>gcc MaquinaExpendedora01.c -o MaquinaExpendedora01.exe
C:\Users\Usuario\Desktop\Lenguaje C\Ejemplos>MaquinaExpendedora01.exe
Pesos a devolver: 
```



10. El programa ha sido ejecutado con éxito si se ve de la siguiente manera:



**¡Fin del Manual de usuario!**

## **Conclusiones**

Describir cómo aplicaron los temas de la clase en su proyecto final y por qué no aplicaron los temas faltantes (si los hay)

Se aplicó los temas

- Practica 0: La práctica consistió en conocer la página web del laboratorio y la estructura y contenido de Sistema de Gestión de Calidad.
- Practica 1: Se trató de realizar búsquedas especializadas, realizar visitas virtuales y conocer diferentes proveedores de servicios, todo a través de la web.
- Practica 2: identificar al sistema operativo como una parte esencial de un sistema de cómputo. Explorar un sistema operativo GNU/Linux con el fin de conocer y utilizar los comandos básicos.
- Practica 3: elaborar algoritmos correctos y eficientes en una solución de problemas siguiendo las etapas de análisis y diseño pertenecientes al ciclo de vida del software.
- Practica 4: elaborar diagramas de flujo que representen soluciones algorítmicas vistas como una serie de acciones que comprendan un proceso.
- Practica 5: elaborar pseudocódigos que representen soluciones algorítmicas empleando la sintaxis y semántica adecuadas.

- Practica 6: la práctica con esto de laborar programas en lenguaje C utilizando las instrucciones de control de tipo secuencia, para realizar la declaración de variables de diferentes tipos de datos, así como efectuar llamadas a funciones externas de entrada y salida para asignar y mostrar valores de variables y expresiones.
- Practica 7: elaborar programas en lenguaje C que incluyen las estructuras de selección if, if-else, switch y ternaria (o condicional) para la resolución de problemas básicos.
- Practica 8: elaborar programas en lenguaje sea para la resolución de problemas básicos que incluyen las estructuras de repetición.
- Practica 9: utilizar arreglos de una dimensión en la elaboración de programas que resuelvan problemas que requieran agrupar datos del mismo tipo alineándolos en un vector o lista.
- Practica 10: la práctica se trató de utilizar arreglos de 2 dimensiones en la elaboración de programas que resuelvan problemas que requieren agrupar datos del mismo tipo en estructuras que utilicen 2 índices.
- Practica 11: esta práctica constó de elaborar programas en lenguaje C donde la solución del problema se divide en funciones. Distinguir lo que es el prototipo o firma de una función y la implementación de ella, así como manipular parámetros tanto en la función principal como en otras.
- Practica 12: esta práctica con todo elaborar programas en lenguaje que requieran el uso de archivos de texto plano en la resolución de problemas entendiendo a los archivos como un elemento de almacenamiento secundario.

En la práctica presente se llevó a cabo el uso de la realización de búsquedas específicas, la visita de diferentes proveedores de servicios a través de la web pertenecientes a la práctica 1, se elaboró algoritmos correctos y eficientes para la solución de problemas además del seguimiento de las etapas análisis y diseño pertenecientes a la práctica 2 y 3, se elaboró un diagrama de flujo con el fin de comprendieron proceso perteneciente a la práctica 4, también se implementó el uso de pseudocódigos para representar soluciones algorítmicas pertenecientes a la práctica número 5, de igual manera se elaboró un programa en lenguaje C con la declaración de variables de diferentes tipos de datos mostrando valores de variables y expresiones pertenecientes a la práctica número 6, en la práctica número 7 se incluyó la elaboración de programas en lenguaje sé que incluían estructuras de selección y su respectiva evaluación, la práctica número 8 se tomó como base en la resolución del problema de estructura de repetición, la práctica número 9 fue de gran utilidad para la elaboración del agrupamiento de datos del mismo tipo alineados en una lista, la práctica número 10 no se tomó en cuenta ya que no era requerido en este proyecto, la práctica número 11 sirvió para manipular parámetros tanto en la función principal como en otras finalizando con la práctica número 12 la cual sirvió



de guía para entender a los archivos como un elemento de almacenamiento secundario.

Como conclusión personal puedo destacar que el uso del lenguaje C, implica conocimientos básicos muy sencillos que gracias a ellos se puede hacer el desarrollo de algún programa en este caso un proyecto para alguna empresa o industria aplicado a la vida diaria el cual puede tener un fin de programar y facilitar la resolución de problemas sencillos como pueden ser juegos o máquinas para principiantes.

De igual manera me gustaría destacar que este tipo de proyectos a largo plazo pueden ser de mucha ayuda ya que nos da inicios de cómo se fueron estructurando algunas calculadoras o máquinas que hoy en día usamos en nuestra vida cotidiana, ya que para todos esos sistemas operativos o aparatos inteligentes existe un detrás de, en el cual muchísimas personas elaboraron, crearon y desarrollaron el prototipo gracias a una problemática causal. ahora sé que en un futuro pueda implementar el lenguaje C, y más adelante otros tipos de lenguajes que me ayuden a programar y a destacarme como ingeniero industrial para estar un paso más adelante de mis colegas de la escuela y a nivel profesional. el problema actual de mi proyecto lo desarrollé gracias aquí siempre me ha dado curiosidad el hecho de pagar un boleto de estacionamiento en alguna plaza así que gracias a este curso de fundamentos de programación logré hacer algo similar e incluso en un futuro podré desarrollar programas más sofisticados en los que la problemática sí aparte de toda una sociedad y este programa ayude y facilite la vida de toda una comunidad entera. Tuve una buena perspectiva de esta materia y me ayudó bastante para las bases que requiero para hacer egresada de ingeniería industrial. Me gustó bastante este proyecto y espero seguir aprendiendo y fortaleciendo esta rama de la ingeniería para aportar a mi escuela a mi comunidad y el mundo entero.

## Referencia.

1. Mollá, R. Problemas resueltos en lenguaje C. Universidad Politécnica de Valencia. (S.F.) Recuperado de: [https://gdocu.upv.es/alfresco/service/api/node/content/workspace/SpacesStore/13c264ae-843c-4790-9fd3-5a32913cb421/TOC\\_0683\\_04\\_01.pdf?quest=true](https://gdocu.upv.es/alfresco/service/api/node/content/workspace/SpacesStore/13c264ae-843c-4790-9fd3-5a32913cb421/TOC_0683_04_01.pdf?quest=true)
2. S.A. [Saber programas]. (31 de julio del 2017) *Cómo crear un DIAGRAMA de GANTT en Excel [Cronograma usando los gráficos]*. [Video]. YouTube. Recuperado de: <https://youtu.be/chR6kx4btDQ>

## Glosario.

- **Hardware:** Es la parte que puedes ver y tocar de los dispositivos. Es decir, todos los componentes de su estructura física como pantallas y teclados.
- **Software:** son los programas informáticos que hacen posible la ejecución de tareas específicas dentro de un computador. Por ejemplo, los sistemas operativos, aplicaciones, navegadores web, juegos o programas.
- **Diagrama de Gantt:** es una herramienta gráfica cuyo objetivo es exponer el tiempo de dedicación previsto para diferentes tareas o actividades a lo largo de un tiempo total determinado. A pesar de esto, el diagrama de Gantt no indica las relaciones existentes entre actividades.
- **Ordenador:** Máquina electrónica capaz de almacenar información y tratarla automáticamente mediante operaciones matemáticas y lógicas controladas por programas informáticos.

## Acrónimos o abreviaturas utilizadas.

- **Stdio.h:** standard input-output header" (cabecera estándar E/S)
- **Int main:** es una firma legal y un tipo de devolución para **main**.
- **Float:** es una forma de notación científica usada en los computadores con la cual se pueden representar números reales extremadamente grandes y pequeños de una manera muy eficiente y compacta, y con la que se pueden realizar operaciones aritméticas.
- **Printf:** (que deriva su nombre de "print formatted") imprime un mensaje por pantalla utilizando una "cadena de formato" que incluye las instrucciones para mezclar múltiples cadenas en la cadena final a mostrar por pantalla.
- **Scanf:** permite leer varios tipos de datos de una sola vez, tales como enteros, números decimales o cadenas de caracteres.