



República Bolivariana De Venezuela

Universidad Nacional Experimental De Guayana

Vicerrectorado Académico

Asignatura: Técnicas De Programación

Proyecto De Carrera: Ingeniería En Informática

### **De La Teoría A La Práctica: Proyecto Hipermercado**

Profesora:

Ingeniera Dubraska Roca

Elaborado Por:

Moirá Rojas | CI 32.503.797 | Sección 5

Diego Barrios | CI 31.444.804 | Sección 5

Gerald López | CI 31.074.565 | Sección 4

Jesús Mundarain | CI 31.384.887 | Sección 4

Miguel Rodríguez | CI 31.963.149 | Sección 4

Ciudad Guayana, Venezuela

24 de Febrero de 2025

## Índice

### Contenido

<b>Índice .....</b>	<b>2</b>
<b>Introducción .....</b>	<b>3</b>
<b>Ánàlisis del Planteamiento Realizado .....</b>	<b>5</b>
<b>Objetivo General.....</b>	<b>7</b>
<b>Objetivos Específicos .....</b>	<b>8</b>
<b>Proceso de Instalación y Configuración de la Plataforma .....</b>	<b>10</b>
<b>Tablas de Librerías, Funciones y/o Procedimientos Utilizados .....</b>	<b>17</b>
<b>Descripciones De Las Validaciones A Considerar .....</b>	<b>51</b>
<b>Diagrama De Flujo Del Algoritmo Principal.....</b>	<b>55</b>
<b>Pantallas De Cada Una De Las Corridas Del Programa.....</b>	<b>56</b>
<b>Manual De Usuario .....</b>	<b>72</b>
<b>Conclusiones .....</b>	<b>81</b>
<b>Referencias Consultadas .....</b>	<b>83</b>

## **Introducción**

En el marco de la materia de Técnicas de Programación I, se nos ha encomendado el desarrollo de un programa en Lenguaje C, diseñado específicamente para la gestión y control del Hipermercado Cachamay, ubicado en Puerto Ordaz. Este proyecto no solo representa un desafío técnico significativo, sino también una oportunidad para aplicar de manera práctica los conocimientos adquiridos a lo largo del curso.

El Hipermercado Cachamay, con el número de RIF: J-123456789 y número de teléfono 04140463588, cuenta con cinco cajas operativas responsables del cobro de todos los productos vendidos mensualmente. Nuestro programa tiene como objetivo principal optimizar la gestión de productos, clientes, facturas, ventas y personal, proporcionando una herramienta eficiente que facilite las operaciones diarias del hipermercado, adaptándose al estándar de programas manejados en este tipo de entornos

A través de este proyecto, se abordarán múltiples aspectos de la programación y el diseño de proyectos de este tipo, incluyendo la creación, modificación, eliminación y visualización de datos en diversos módulos, tales como productos, clientes, facturas, ventas, control de cajas y trabajadores. Además, se generarán reportes detallados que permitan un análisis exhaustivo de las operaciones del hipermercado.

El desarrollo de este programa no solo nos permitirá consolidar, desarrollar y evidenciar nuestras habilidades en programación, sino también nos hará comprender la importancia de las

soluciones tecnológicas en la gestión empresarial, contribuyendo así a mejorar la eficiencia y productividad del Hipermercado Cachamay.

El programa, desarrollado C con una interfaz gráfica amigable para el usuario, incluirá un sistema para manejar la conversión de precios en diferentes monedas (Bolívares, Pesos colombianos, Dólares, Euros) basada en la tasa del día, y permitirá la creación, modificación,

eliminación y visualización de datos relacionados con estos módulos. Adicionalmente, generará reportes como la cantidad de trabajadores del hipermercado, cantidad de clientes con compras realizadas, producto más vendido, totales de ventas diarias, semanales y mensuales, y el detalle de las últimas 10 facturas.

Se espera que el programa mejore la gestión y control de las operaciones del hipermercado, aumentando la eficiencia de la información manejada, lo que permitirá al cliente tener mucho más control y operatividad del Hipermercado Cachamay. Este proyecto representa una valiosa oportunidad para aplicar los conocimientos adquiridos en la materia de Técnicas de Programación I, consolidando nuestras habilidades en programación y diseño de sistemas, y se espera que tenga un impacto positivo significativo en la gestión del Hipermercado Cachamay, facilitando sus operaciones diarias.

## **Objetivo General**

Desarrollar un programa avanzado en lenguaje C con una interfaz gráfica intuitiva y fácil de usar para el control y manejo integral del Hipermercado Cachamay, ubicado en Puerto Ordaz. Este programa tiene como propósito principal optimizar y automatizar los procesos operativos y administrativos del hipermercado. Permitirá una gestión eficiente y precisa de los productos, clientes, facturas, ventas, cajas y trabajadores. Además, busca mejorar la calidad del servicio al cliente y la toma de decisiones estratégicas mediante la generación de reportes detallados y análisis de datos. Con esta herramienta, el Hipermercado Cachamay podrá mantener un control riguroso de su inventario, asegurar una experiencia de compra satisfactoria para sus clientes y facilitar la administración del personal y las operaciones diarias, mensuales y anuales del negocio. La implementación de este sistema integral contribuirá a un incremento en la eficiencia operativa y al fortalecimiento de la competitividad del hipermercado en el mercado local.

### **Objetivos Específicos**

1. **Diseñar e implementar un módulo para el manejo de productos** que permita crear, modificar, eliminar y mostrar información sobre los productos vendidos en el Hipermercado, incluyendo su código, nombre, cantidad disponible y precios en diferentes monedas (Bs, Pesos Colombianos, Dólares y Euros).
2. **Desarrollar un módulo de gestión de clientes** que permita registrar, modificar, eliminar y mostrar información detallada de los clientes, tales como nombre, apellido, teléfono, dirección y cédula, facilitando así un mejor seguimiento y servicio al cliente.
3. **Implementar un sistema de facturación** que genere facturas con toda la información necesaria, incluyendo identificación del Hipermercado, número de factura, fecha, datos del cliente, detalle de productos comprados, porcentaje de IVA y monto total a pagar.
4. **Desarrollar un módulo de control de ventas** que permita realizar consultas diarias y mensuales por caja, mostrando el total de ventas tanto diarias como mensuales realizadas en el Hipermercado, proporcionando reportes precisos para la toma de decisiones.
5. **Crear un sistema de gestión de cajas** que registre y controle la información de cada caja operativa, incluyendo número de caja, número del empleado responsable, nombre del responsable y totales facturados en Bs, Dólares, Pesos Colombianos y Euros.



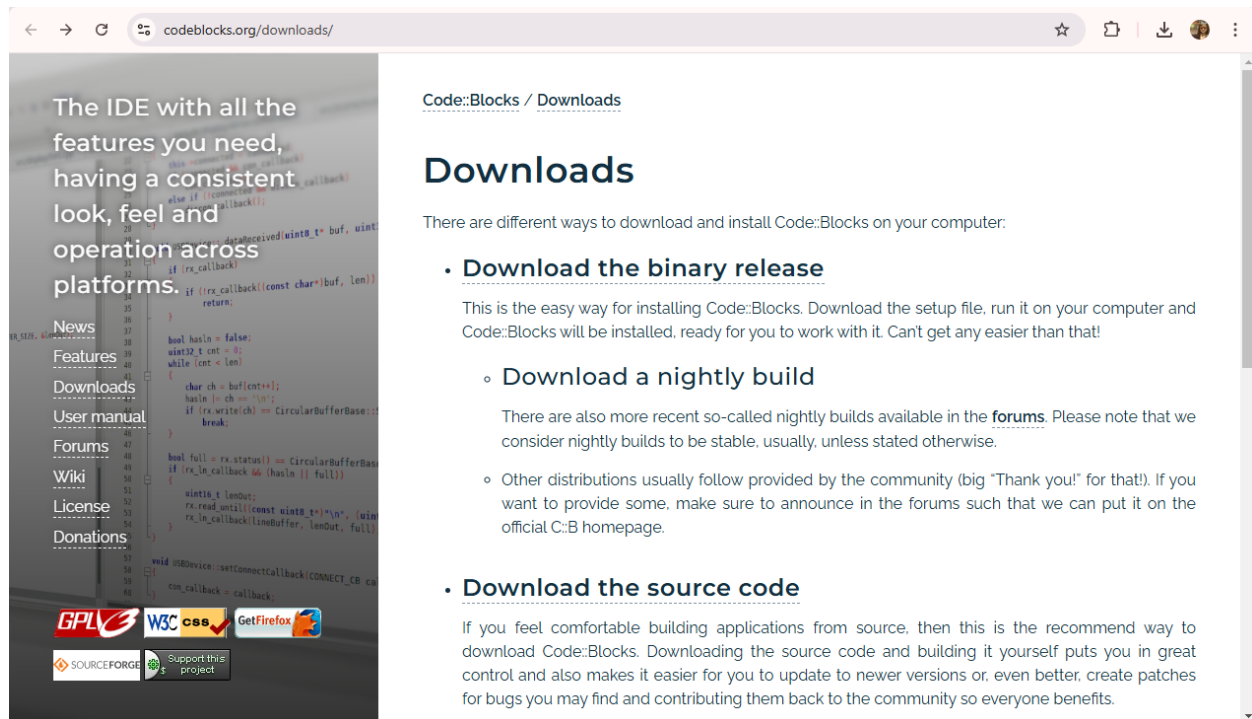
6. **Implementar un módulo para la gestión de trabajadores** que permita manejar la información de los empleados del Hipermercado, incluyendo nombre, apellido, cédula, número de empleado, teléfono y dirección, facilitando la administración del personal.
7. **Incorporar la funcionalidad para generar reportes** que incluyan la cantidad de trabajadores, cantidad de clientes con compras realizadas, producto más vendido, totales de ventas diarias, semanales y mensuales, y el detalle de las últimas 10 facturas, proporcionando así información clave para la gestión del Hipermercado.

## Proceso de Instalación y Configuración de la Plataforma

Para la elaboración de este proyecto hicimos uso de Code::Blocks, una plataforma de desarrollo integrada (IDE) para la programación en C, C++ y Fortran. El siguiente procedimiento describe los pasos seguidos para instalar y configurar la plataforma en un equipo con sistema operativo Windows, sin necesidad de instalar complementos adicionales.

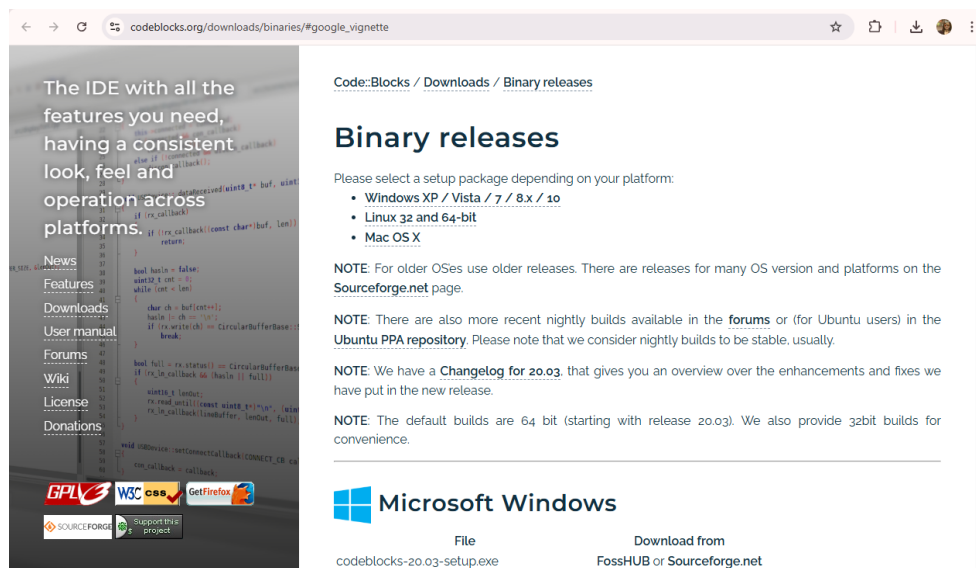
1. Para comenzar, se accedió al sitio web oficial de Code::Blocks:

(<https://www.codeblocks.org/downloads/>)



2. Ingresamos al vínculo “Download the binary release”, el cual nos dirigirá a “Binary Releases” ([https://www.codeblocks.org/downloads/binaries/#google\\_vignette](https://www.codeblocks.org/downloads/binaries/#google_vignette)). Ahí, bajamos dentro de la página hasta ver el apartado de Microsoft Windows. A

continuación, seleccionamos “codeblocks-20.03mingw-32bit-setup.exe” y vamos a descargarlo a FossHUB.



The IDE with all the features you need, having a consistent look, feel and operation across platforms.

News  
Features  
Downloads  
User manual  
Forums  
Wiki  
License  
Donations

Code::Blocks / Downloads / Binary releases

## Binary releases

Please select a setup package depending on your platform:

- Windows XP / Vista / 7 / 8.x / 10
- Linux 32 and 64-bit
- Mac OS X

**NOTE:** For older OSes use older releases. There are releases for many OS version and platforms on the [Sourceforge.net](#) page.

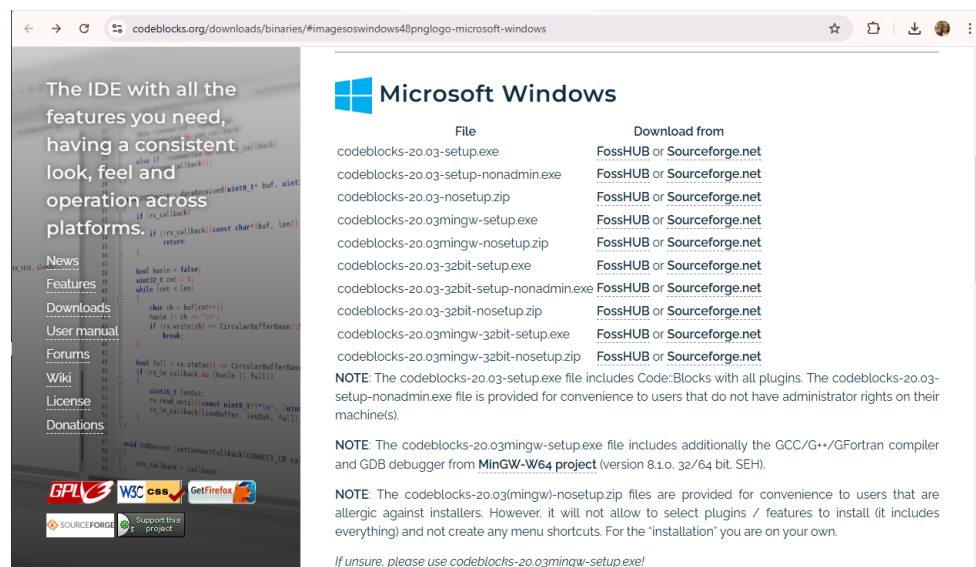
**NOTE:** There are also more recent nightly builds available in the [forums](#) (or for Ubuntu users) in the [Ubuntu PPA repository](#). Please note that we consider nightly builds to be stable, usually.

**NOTE:** We have a [Changelog for 20.03](#), that gives you an overview over the enhancements and fixes we have put in the new release.

**NOTE:** The default builds are 64 bit (starting with release 20.03). We also provide 32bit builds for convenience.

### Microsoft Windows

File	Download from
codeblocks-20.03-setup.exe	FossHUB or Sourceforge.net



The IDE with all the features you need, having a consistent look, feel and operation across platforms.

News  
Features  
Downloads  
User manual  
Forums  
Wiki  
License  
Donations

codeblocks.org/downloads/binaries/#imagesoswindows48pnglogo-microsoft-windows

## Microsoft Windows

File	Download from
codeblocks-20.03-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03-setup-nonadmin.exe	FossHUB or Sourceforge.net
codeblocks-20.03-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03mingw-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03mingw-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-setup-nonadmin.exe	FossHUB or Sourceforge.net
codeblocks-20.03-32bit-nosetup.zip	FossHUB or Sourceforge.net
codeblocks-20.03mingw-32bit-setup.exe	FossHUB or Sourceforge.net
codeblocks-20.03mingw-32bit-nosetup.zip	FossHUB or Sourceforge.net

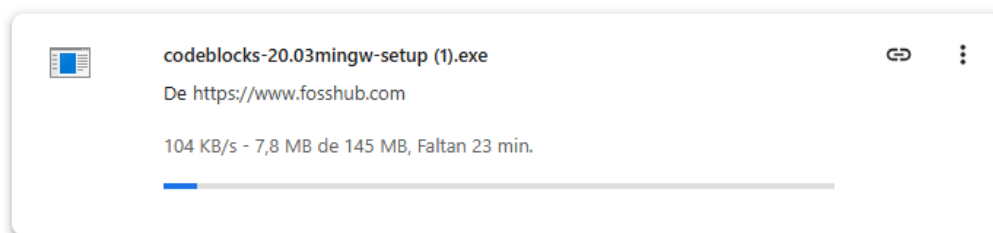
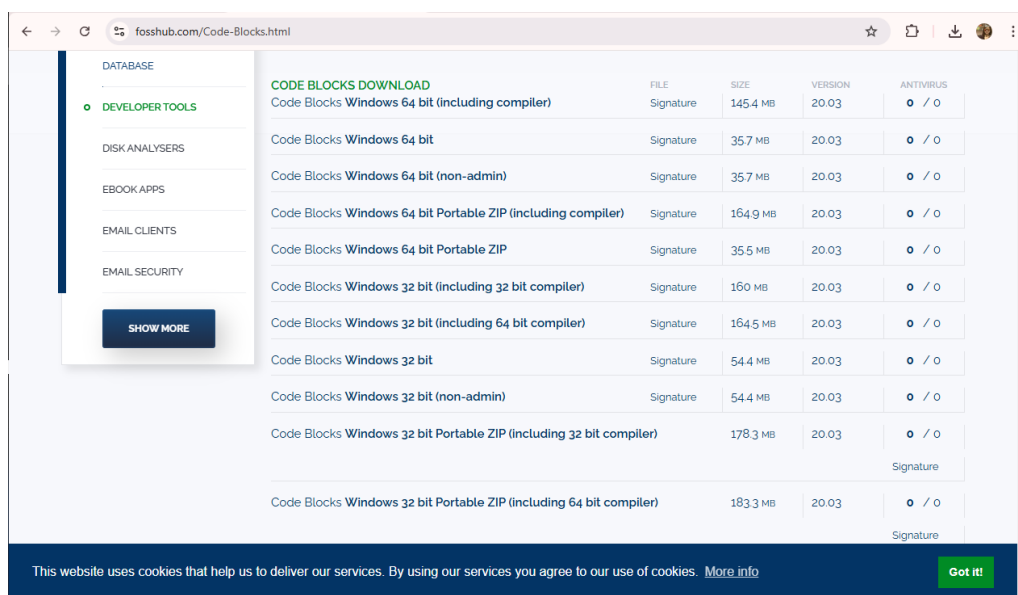
**NOTE:** The codeblocks-20.03-setup.exe file includes Code::Blocks with all plugins. The codeblocks-20.03-setup-nonadmin.exe file is provided for convenience to users that do not have administrator rights on their machine(s).

**NOTE:** The codeblocks-20.03mingw-setup.exe file includes additionally the GCC/G++/GFortran compiler and GDB debugger from [MinGW-W64 project](#) (version 8.1.0, 32/64 bit, SEH).

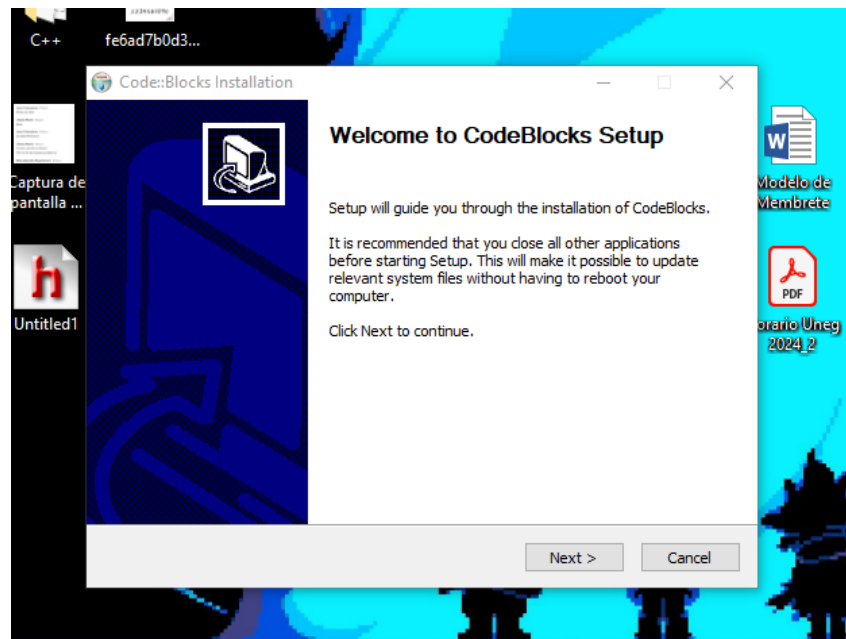
**NOTE:** The codeblocks-20.03(mingw)-nosetup.zip files are provided for convenience to users that are allergic against installers. However, it will not allow to select plugins / features to install (it includes everything) and not create any menu shortcuts. For the “installation” you are on your own.

*If unsure, please use codeblocks-20.03mingw-setup.exe!*

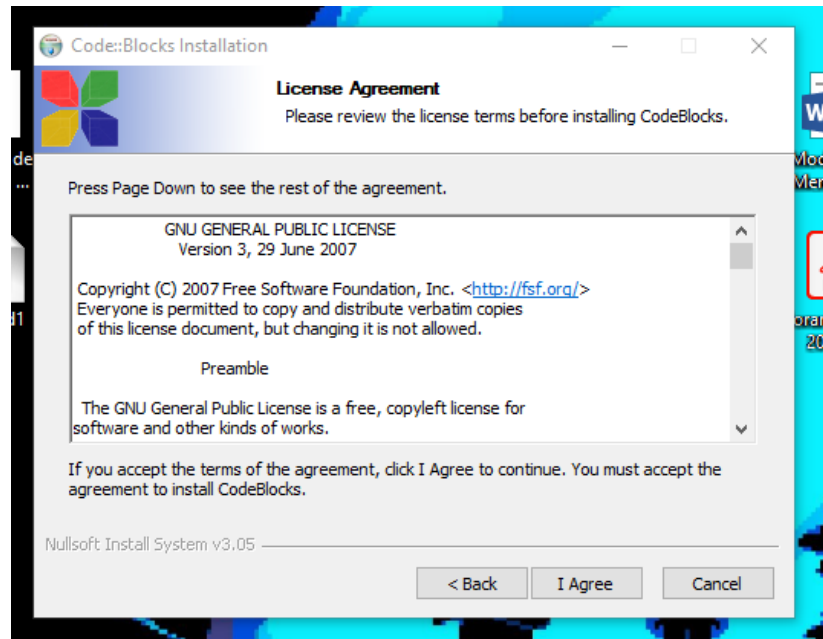
3. Dentro de FossHUB, hacemos click en la primera opción, la cual es “Code Blocks Windows 64 bit (Including Compiler)”;
- posterior a esto, la descarga empezará en tres segundos y sólo tendremos que esperar a que se descargue para proseguir.



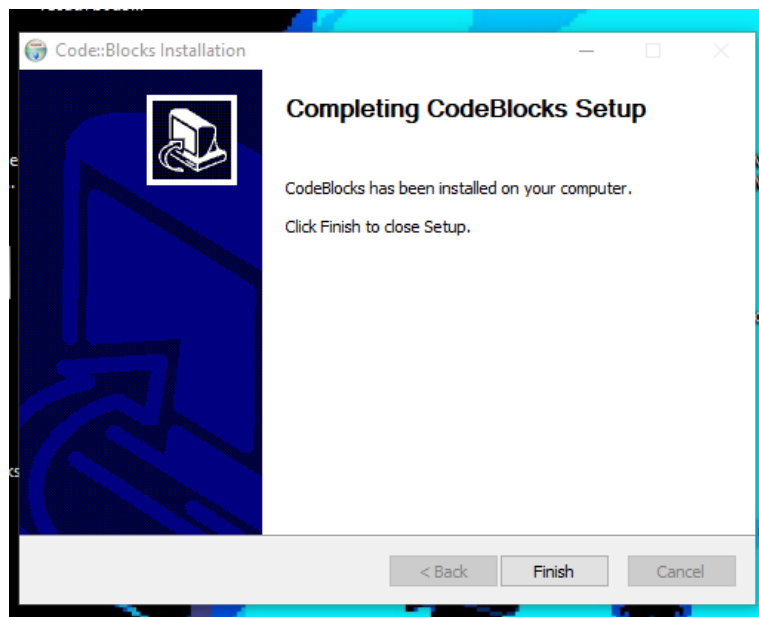
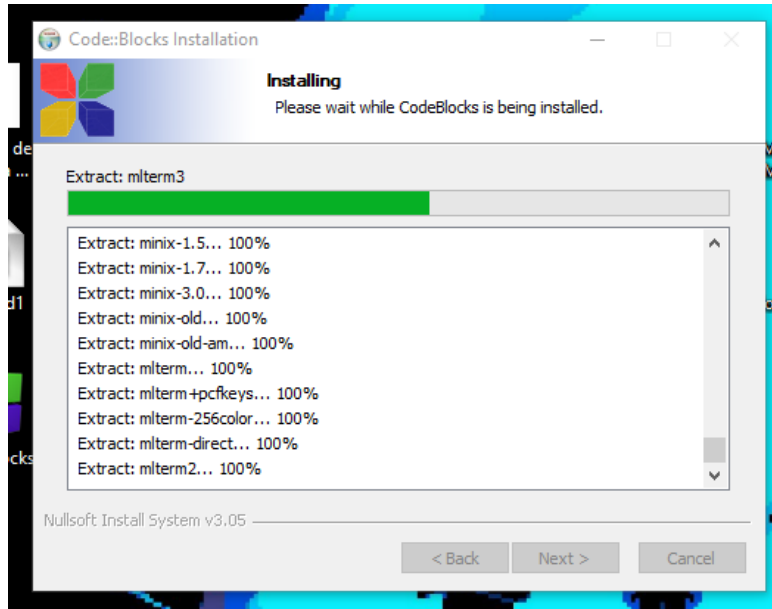
4. Una vez descargado, ejecutamos el set up descargado, el cual nos llevará a una pantalla de bienvenida, donde se nos indica el inicio del proceso de instalación



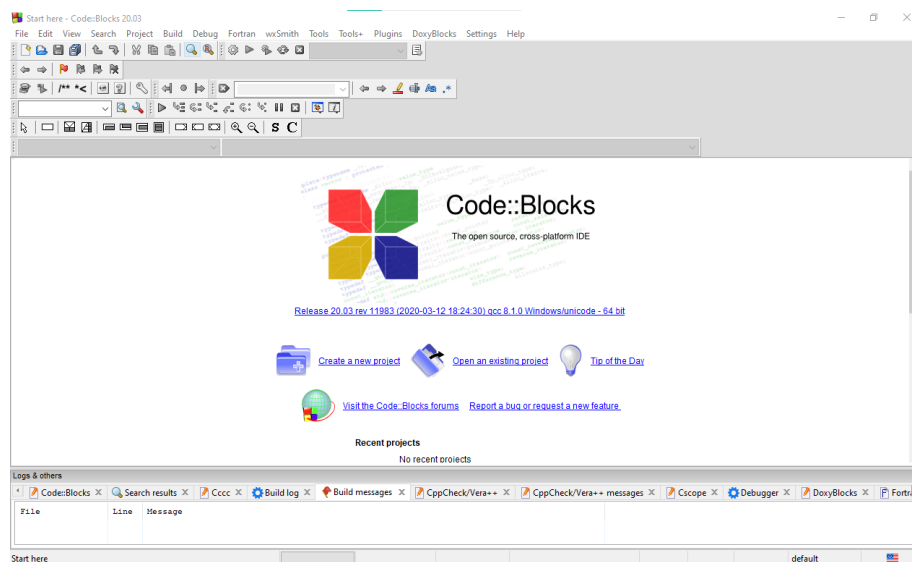
5. Al presionar “Next” en la pantalla anterior, nos llevará a la pestaña de acuerdo de licencia. Para proseguir, elegimos “I Agree” para aceptar, lo que nos llevará a la siguiente pantalla.



6. Al salir de la pantalla anterior, ahora tendremos que elegir componentes, los elegimos todos y continuamos. Posterior a eso, se mostrará una pantalla donde se nos pide que seleccionemos la ruta de instalación, en la cual dejaremos la ruta predeterminada proporcionada por el instalador. Luego de eso, llegamos a la pantalla de instalación, en la cual cliquearemos "Install" para comenzar la instalación. Una vez completada la instalación, cliqueamos en "Finish" para cerrar el asistente de instalación.



7. Se ejecutó Code::Blocks por primera vez desde el acceso directo creado en el escritorio o desde el menú de inicio.





## Tablas de Librerías, Funciones y/o Procedimientos Utilizados

### a. Librerías.

Nombre	Descripción	Funciones Predefinidas Utilizadas
<b>stdio.h</b>	Biblioteca estándar de entrada y salida en C.	printf, scanf, sprintf, fgets, getchar, fopen, fprintf, fclose
<b>stdlib.h</b>	Biblioteca estándar de utilidades en C.	system
<b>string.h</b>	Biblioteca estándar para manejo de cadenas de caracteres en C.	strlen, strcpy, strcmp, strcspn
<b>time.h</b>	Biblioteca estándar para manipulación de fechas y horas en C.	time, localtime, strftime

### b. Funciones y procedimientos.

#### *Funciones Y Módulos Principales*

##### 1. **Procedimiento:** void mostrar\_menu\_principal()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables Locales:**
  - int opcion;
- **Descripción:** Muestra el menú principal de la aplicación, lee la opción seleccionada y contiene una estructura condicional “switch” que llama a las funciones correspondientes a dicha opción, todo eso dentro de una estructura repetitiva “while” para repetir el ciclo hasta que la opción seleccionada sea correspondiente a la opción de salida, limpiando la pantalla en cada iteración.

**2. Procedimiento:** void modulo\_productos();

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int opcion\_producto;
- **Descripción:** Muestra el menú del módulo de productos, lee la opción seleccionada y contiene una estructura condicional “switch” que llama a las funciones correspondientes a dicha opción, todo eso dentro de una estructura repetitiva “while” para repetir el ciclo hasta que la opción seleccionada sea correspondiente a la opción de salida, limpiando la pantalla en cada iteración.

**3. Procedimiento:** void modulo\_clientes();

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int opcion\_cliente;
- **Descripción:** Muestra el menú del módulo de clientes, lee la opción seleccionada y contiene una estructura condicional “switch” que llama a las funciones correspondientes a dicha opción, todo eso dentro de una estructura repetitiva “while” para repetir el ciclo

hasta que la opción seleccionada sea correspondiente a la opción de salida, limpiando la pantalla en cada iteración.

**4. Procedimiento:** void modulo\_trabajadores();

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int opcion\_trabajador();
- **Descripción:** Muestra el menú del módulo de trabajadores, lee la opción seleccionada y contiene una estructura condicional “switch” que llama a las funciones correspondientes a dicha opción, todo eso dentro de una estructura repetitiva “while” para repetir el ciclo hasta que la opción seleccionada sea correspondiente a la opción de salida, limpiando la pantalla en cada iteración.

**5. Procedimiento:** void modulo\_cajas();

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int opcion\_caja;
- **Descripción:** Muestra el menú del módulo de cajas, lee la opción seleccionada y contiene una estructura condicional “switch” que llama a las funciones correspondientes a dicha

opción, todo eso dentro de una estructura repetitiva “while” para repetir el ciclo hasta que la opción seleccionada sea correspondiente a la opción de salida, limpiando la pantalla en cada iteración.

**6. Procedimiento:** void modulo\_facturas();

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int opcion\_factura;
- **Descripción:** Muestra el menú del módulo de facturas, lee la opción seleccionada y contiene una estructura condicional “switch” que llama a las funciones correspondientes a dicha opción, todo eso dentro de una estructura repetitiva “while” para repetir el ciclo hasta que la opción seleccionada sea correspondiente a la opción de salida, limpiando la pantalla en cada iteración.

**7. Procedimiento:** void modulo\_ventaas();

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int opcion\_ventas;

- **Descripción:** Muestra el menú del módulo de ventas, lee la opción seleccionada y contiene una estructura condicional “switch” que llama a las funciones correspondientes a dicha opción, todo eso dentro de una estructura repetitiva “while” para repetir el ciclo hasta que la opción seleccionada sea correspondiente a la opción de salida, limpiando la pantalla en cada iteración.

**8. Procedimiento:** void mostrar\_ayuda\_usuario();

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:** Ninguna
- **Descripción:** Muestra un manual de usuario integrado dentro del programa en una salida de texto sencilla.

### *Funciones De Productos*

**9. Procedimiento:** void crear\_producto();

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int cantidad\_agregar;
  - int i;
  - Producto nuevo\_producto;

- char regresar\_menu;
- **Descripción:** Esta función permite la creación de nuevos productos. Primero verifica si se ha alcanzado el límite máximo de productos. Si no, solicita la cantidad de productos que se desean agregar y verifica que sea una cantidad válida. Luego, itera según la cantidad indicada, solicitando detalles de cada producto y verificando que no se repitan los códigos. Calcula los precios en otras monedas y agrega el producto a la lista. Finalmente, pregunta al usuario si desea regresar al menú principal.

#### 10. Procedimiento: void calcular\_precios\_monedas(Producto \*producto)

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Producto \*producto (puntero a una estructura Producto)
- **Variables locales:** Ninguna
- **Descripción:** Esta función recibe un puntero a una estructura Producto y calcula los precios del producto en distintas monedas (USD, COP y EUR) basándose en la tasa de cambio correspondiente. El precio en bolívares (Bs) es tomado como referencia y se divide por las tasas de cambio definidas para obtener los precios en dólares (USD), pesos colombianos (COP) y euros (EUR).

Vamos ahora con la función modificar\_producto():

#### 11. Procedimiento: void modificar\_producto()

- **Tipo de Retorno:** void

- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int codigo\_modificar;
  - int index;
  - int cantidad;
  - float precio\_bs;
  - char regresar\_menu;
- **Descripción:** Esta función permite modificar un producto existente en la lista. Verifica si hay productos registrados y solicita el código del producto a modificar. Si el producto existe, permite cambiar el nombre, la cantidad disponible y el precio en bolívares. Los precios en otras monedas son recalculados si el precio en bolívares se modifica. Finalmente, pregunta al usuario si desea regresar al menú principal.

Finalmente, vamos con la función eliminar\_producto():

## 12. Procedimiento: void eliminar\_producto()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**

- int codigo\_eliminar;
  - int index;
  - char regresar\_menu;
- **Descripción:** Esta función permite eliminar un producto existente en la lista. Verifica si hay productos registrados y solicita el código del producto a eliminar. Si el producto existe, desplaza los productos siguientes una posición hacia atrás para eliminar el producto de la lista. Finalmente, pregunta al usuario si desea regresar al menú principal.

### 13. Procedimiento: void mostrar\_producto()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int codigo\_mostrar;
  - int index;
  - char regresar\_menu;
- **Descripción:** Esta función permite mostrar los detalles de un producto específico. Verifica si hay productos registrados y solicita el código del producto a mostrar. Si el producto existe, muestra sus detalles (código, nombre, cantidad disponible y precios en



distintas monedas) en la consola. Finalmente, pregunta al usuario si desea regresar al menú principal.

**14. Procedimiento:** void mostrar\_todos\_productos()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - char regresar\_menu;
- **Descripción:** Esta función muestra una lista de todos los productos registrados. Verifica si hay productos disponibles y, si los hay, limpia la consola y muestra una tabla con los detalles de cada producto (código, nombre, cantidad disponible y precios en distintas monedas). Finalmente, pregunta al usuario si desea regresar al menú principal.

*Funciones De Clientes*

**15. Procedimiento:** void crear\_cliente()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int cantidad\_agregar;

- int i;
- Cliente nuevo\_cliente;
- char regresar\_menu;
- **Descripción:** Esta función permite la creación de nuevos clientes. Primero verifica si se ha alcanzado el límite máximo de clientes. Si no, solicita la cantidad de clientes que se desean agregar y verifica que sea una cantidad válida. Luego, itera según la cantidad indicada, solicitando detalles de cada cliente y verificando que no se repitan las cédulas. Agrega el cliente a la lista. Finalmente, pregunta al usuario si desea regresar al menú principal.

#### 16. Procedimiento: void modificar\_cliente()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int cedula\_modificar;
  - int index;
  - char regresar\_menu;
- **Descripción:** Esta función permite modificar un cliente existente en la lista. Verifica si hay clientes registrados y solicita la cédula del cliente a modificar. Si el cliente existe,

permite cambiar el nombre, apellido, teléfono y dirección del cliente. Finalmente, pregunta al usuario si desea regresar al menú principal.

**17. Procedimiento:** void eliminar\_cliente()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int cedula\_eliminar;
  - int index;
  - char regresar\_menu;
- **Descripción:** Esta función permite eliminar un cliente existente en la lista. Verifica si hay clientes registrados y solicita la cédula del cliente a eliminar. Si el cliente existe, desplaza los clientes siguientes una posición hacia atrás para eliminar al cliente de la lista. Finalmente, pregunta al usuario si desea regresar al menú principal.

**18. Procedimiento:** void mostrar\_cliente()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**

- int cedula\_mostrar;
- int index;
- char regresar\_menu;
- **Descripción:** Esta función permite mostrar los detalles de un cliente específico. Verifica si hay clientes registrados y solicita la cédula del cliente a mostrar. Si el cliente existe, muestra sus detalles (cédula, nombre, apellido, teléfono y dirección) en la consola. Finalmente, pregunta al usuario si desea regresar al menú principal.

**19. Procedimiento:** void mostrar\_todos\_clientes()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - char regresar\_menu;
- **Descripción:** Esta función muestra una lista de todos los clientes registrados. Verifica si hay clientes disponibles y, si los hay, limpia la consola y muestra una tabla con los detalles de cada cliente (cédula, nombre, apellido, teléfono y dirección). Finalmente, pregunta al usuario si desea regresar al menú principal.

*Funciones De Trabajadores*

**20. Procedimiento:** void crear\_trabajador()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int cantidad\_agregar;
  - int i;
  - Trabajador nuevo\_trabajador;
  - char regresar\_menu;
- **Descripción:** Esta función permite la creación de nuevos trabajadores. Primero verifica si se ha alcanzado el límite máximo de trabajadores. Si no, solicita la cantidad de trabajadores que se desean agregar y verifica que sea una cantidad válida. Luego, itera según la cantidad indicada, solicitando detalles de cada trabajador y verificando que no se repitan los números de empleado. Agrega al trabajador a la lista. Finalmente, pregunta al usuario si desea regresar al menú principal.

## 21. Procedimiento: void modificar\_trabajador()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int numero\_empleado\_modificar;

- int index;
- int cedula;
- char regresar\_menu;
- **Descripción:** Esta función permite modificar un trabajador existente en la lista. Verifica si hay trabajadores registrados y solicita el número de empleado del trabajador a modificar. Si el trabajador existe, permite cambiar el nombre, apellido, teléfono, dirección y cédula del trabajador. Finalmente, pregunta al usuario si desea regresar al menú principal.

## 22. Procedimiento: void eliminar\_trabajador()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int numero\_empleado\_eliminar;
  - int index;
  - char regresar\_menu;
- **Descripción:** Esta función permite eliminar un trabajador existente en la lista. Verifica si hay trabajadores registrados y solicita el número de empleado del trabajador a eliminar. Si el trabajador existe, desplaza a los trabajadores siguientes una posición hacia atrás para

eliminar al trabajador de la lista. Finalmente, pregunta al usuario si desea regresar al menú principal.

**23. Procedimiento:** void mostrar\_trabajador()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int numero\_empleado\_mostrar;
  - int index;
  - char regresar\_menu;
- **Descripción:** Esta función permite mostrar los detalles de un trabajador específico. Verifica si hay trabajadores registrados y solicita el número de empleado del trabajador a mostrar. Si el trabajador existe, muestra sus detalles (número de empleado, cédula, nombre, apellido, teléfono y dirección) en la consola. Finalmente, pregunta al usuario si desea regresar al menú principal.

**24. Procedimiento:** void mostrar\_todos\_trabajadores()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**

- char regresar\_menu;
- **Descripción:** Esta función muestra una lista de todos los trabajadores registrados.  
Verifica si hay trabajadores disponibles y, si los hay, limpia la consola y muestra una tabla con los detalles de cada trabajador (número de empleado, cédula, nombre, apellido, teléfono y dirección). Finalmente, pregunta al usuario si desea regresar al menú principal.

### *Funciones De Cajas*

#### **25. Procedimiento:** void crear\_caja()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int cantidad\_agregar;
  - int i;
  - Caja nueva\_caja;
  - char regresar\_menu;
- **Descripción:** Esta función permite la creación de nuevas cajas. Primero verifica si se ha alcanzado el límite máximo de cajas. Si no, solicita la cantidad de cajas que se desean agregar y verifica que sea una cantidad válida. Luego, itera según la cantidad indicada, solicitando detalles de cada caja y verificando que no se repitan los números de caja.



Asocia la caja a un trabajador responsable y establece los totales facturados en distintas monedas a 0. Agrega la caja a la lista. Finalmente, pregunta al usuario si desea regresar al menú principal.

**26. Procedimiento:** void modificar\_caja()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int numero\_caja\_modificar;
  - int index;
  - int nuevo\_numero\_empleado;
  - char regresar\_menu;
- **Descripción:** Esta función permite modificar una caja existente en la lista. Verifica si hay cajas registradas y solicita el número de caja a modificar. Si la caja existe, permite cambiar el número de empleado responsable y actualiza el nombre del trabajador responsable. Finalmente, pregunta al usuario si desea regresar al menú principal.

**27. Procedimiento:** void eliminar\_caja()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno

- **Variables locales:**
  - `int numero_caja_eliminar;`
  - `int index;`
  - `char regresar_menu;`
- **Descripción:** Esta función permite eliminar una caja existente en la lista. Verifica si hay cajas registradas y solicita el número de caja a eliminar. Si la caja existe, desplaza las cajas siguientes una posición hacia atrás para eliminar la caja de la lista. Finalmente, pregunta al usuario si desea regresar al menú principal.

**28. Procedimiento:** `void mostrar_caja()`

- **Tipo de Retorno:** `void`
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - `int numero_caja_mostrar;`
  - `int index;`
  - `char regresar_menu;`
- **Descripción:** Esta función permite mostrar los detalles de una caja específica. Verifica si hay cajas registradas y solicita el número de caja a mostrar. Si la caja existe, muestra sus detalles (número de caja, número de empleado responsable, nombre del empleado

responsable y totales facturados en distintas monedas) en la consola. Finalmente, pregunta al usuario si desea regresar al menú principal.

**29. Procedimiento:** void mostrar\_todas\_cajas()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - char regresar\_menu;
- **Descripción:** Esta función muestra una lista de todas las cajas registradas. Verifica si hay cajas disponibles y, si las hay, limpia la consola y muestra una tabla con los detalles de cada caja (número de caja, número de empleado responsable, nombre del empleado responsable y totales facturados en distintas monedas). Finalmente, pregunta al usuario si desea regresar al menú principal.

*Funciones De Facturas*

**30. Procedimiento:** void crear\_factura()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - Factura nueva\_factura;

- `int numero_caja_factura;`
  - `int caja_index;`
  - `int cedula_cliente;`
  - `int cliente_index;`
  - `float subtotal_factura;`
  - `char agregar_producto;`
  - `char prompt_message[100];`
  - `int codigo_producto;`
  - `int producto_index;`
  - `int cantidad_producto_factura;`
  - `float iva;`
  - `char regresar_menu;`
- **Descripción:** Esta función crea una nueva factura. Verifica la existencia de cajas, trabajadores, clientes y productos antes de proceder. Solicita al usuario seleccionar una caja y un cliente, luego permite agregar productos a la factura, verificando disponibilidad de stock. Calcula el subtotal y el IVA, asigna la fecha actual a la factura, y actualiza el total facturado en la caja seleccionada. Finalmente, muestra los detalles de la factura generada y pregunta al usuario si desea regresar al menú principal.

**31. Procedimiento:** void mostrar\_factura()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int numero\_factura\_mostrar;
  - int index;
  - char regresar\_menu;
- **Descripción:** Esta función permite mostrar los detalles de una factura específica. Verifica si hay facturas generadas y solicita el número de factura a mostrar. Si la factura existe, muestra sus detalles en la consola. Finalmente, pregunta al usuario si desea regresar al menú principal.

**32. Procedimiento:** void mostrar\_factura\_detalle(Factura \*factura)

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Factura \*factura (puntero a una estructura Factura)
- **Variables locales:**
  - struct tm \*info\_tiempo;
  - char buffer\_tiempo[80];

- `time_t tiempo_unix;`
- `float subtotal_productos;`
- **Descripción:** Esta función muestra los detalles de una factura específica. Formatea la fecha y hora de la factura, muestra los datos del hipermercado, número de factura, fecha, número de caja, y los datos del cliente. Luego, lista los productos incluidos en la factura con sus cantidades, descripciones, precios unitarios y subtotales. Finalmente, muestra el subtotal de los productos, el IVA y el total a pagar.

### 33. Procedimiento: `void mostrar_ultimas_facturas()`

- **Tipo de Retorno:** `void`
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - `int i;`
  - `char regresar_menu;`
- **Descripción:** Esta función muestra las últimas 10 facturas generadas. Si no hay facturas generadas, muestra un mensaje indicando que no hay facturas. Si hay menos de 10 facturas, las muestra todas; de lo contrario, muestra las últimas 10 facturas. Muestra los detalles de cada factura utilizando la función `mostrar_factura_detalle()`. Finalmente, pregunta al usuario si desea regresar al menú principal.

**34. Procedimiento:** void consultar\_ventas\_diarias\_caja()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int numero\_caja\_consultar;
  - int caja\_index;
  - time\_t tiempo\_actual;
  - struct tm \*fecha\_actual\_tm;
  - int dia\_actual;
  - int mes\_actual;
  - int ano\_actual;
  - float total\_ventas\_bs;
  - int cantidad\_facturas\_caja;
  - char reporte\_diario\_caja[2000];
  - char regresar\_menu;
- **Descripción:** Esta función consulta las ventas diarias de una caja específica. Verifica si hay cajas y facturas registradas. Solicita al usuario el número de caja a consultar y

verifica su existencia. Luego, recorre las facturas del día actual para esa caja y acumula el total de ventas y la cantidad de facturas emitidas. Finalmente, imprime el reporte en consola, guarda el reporte en un archivo TXT y pregunta al usuario si desea regresar al menú principal.

**35. Procedimiento:** void consultar\_ventas\_mensuales\_caja()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int numero\_caja\_consultar;
  - int caja\_index;
  - time\_t tiempo\_actual;
  - struct tm \*fecha\_actual\_tm;
  - int mes\_actual;
  - int ano\_actual;
  - float total\_ventas\_bs;
  - int cantidad\_facturas\_caja;
  - char reporte\_mensual\_caja[2000];



- char regresar\_menu;
- **Descripción:** Esta función consulta las ventas mensuales de una caja específica. Verifica si hay cajas y facturas registradas. Solicita al usuario el número de caja a consultar y verifica su existencia. Luego, recorre las facturas del mes actual para esa caja y acumula el total de ventas y la cantidad de facturas emitidas. Finalmente, imprime el reporte en consola, guarda el reporte en un archivo TXT y pregunta al usuario si desea regresar al menú principal.

### 36. Procedimiento: void mostrar\_total\_ventas\_diarias()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - time\_t tiempo\_actual;
  - struct tm \*fecha\_actual\_tm;
  - int dia\_actual;
  - int mes\_actual;
  - int ano\_actual;
  - float total\_ventas\_bs;
  - char reporte\_total\_diario[2000];

- char regresar\_menu;
- **Descripción:** Esta función muestra el total de ventas diarias del hipermercado. Verifica si hay facturas generadas. Recorre las facturas del día actual y acumula el total de ventas. Finalmente, imprime el reporte en consola, guarda el reporte en un archivo TXT y pregunta al usuario si desea regresar al menú principal.

**37. Procedimiento:** void mostrar\_total\_ventas\_mensuales()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - time\_t tiempo\_actual;
  - struct tm \*fecha\_actual\_tm;
  - int mes\_actual;
  - int ano\_actual;
  - float total\_ventas\_bs;
  - char reporte\_total\_mensual[2000];
  - char regresar\_menu;

- **Descripción:** Esta función muestra el total de ventas mensuales del hipermercado.

Verifica si hay facturas generadas. Recorre las facturas del mes actual y acumula el total de ventas. Finalmente, imprime el reporte en consola, guarda el reporte en un archivo TXT y pregunta al usuario si desea regresar al menú principal.

**38. Procedimiento:** void mostrar\_producto\_mas\_vendido()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno

- **Variables locales:**

- int ventas\_productos[MAX\_PRODUCTOS];
- int producto\_mas\_vendido\_index;
- int cantidad\_mas\_vendida;
- char reporte\_producto\_mas\_vendido[2000];
- char regresar\_menu;

- **Descripción:** Esta función determina cuál es el producto más vendido basado en las facturas generadas. Recorre todas las facturas y cuenta la cantidad de productos vendidos. Encuentra el producto más vendido y genera un reporte con la cantidad total vendida. Imprime el reporte en consola, guarda el reporte en un archivo TXT y pregunta al usuario si desea regresar al menú principal.

**39. Procedimiento:** void mostrar\_cantidad\_trabajadores\_hipermercado()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - char regresar\_menu;
- **Descripción:** Esta función muestra la cantidad total de trabajadores registrados en el hipermercado. Imprime la cantidad en consola y pregunta al usuario si desea regresar al menú principal.

**40. Procedimiento:** void mostrar\_cantidad\_clientes\_con\_compras()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - int clientes\_con\_compra[MAX\_CLIENTES];
  - int cantidad\_clientes\_con\_compras;
  - char reporte\_clientes\_con\_compra[2000];
  - char regresar\_menu;

- **Descripción:** Esta función muestra la cantidad de clientes que han realizado compras basadas en las facturas generadas. Recorre todas las facturas y marca los clientes que han realizado compras. Genera un reporte con la cantidad total de clientes con compras. Imprime el reporte en consola, guarda el reporte en un archivo TXT y pregunta al usuario si desea regresar al menú principal.

### *Funciones Auxiliares*

#### **41. Función:** int obtener\_entero(const char \*mensaje)

- **Tipo de Retorno:** int
- **Parámetros que Recibe:** const char \*mensaje (puntero a una cadena de caracteres)
- **Variables locales:**
  - int valor;
- **Descripción:** Esta función muestra un mensaje y obtiene un número entero del usuario. Verifica que la entrada sea válida y limpia el buffer de entrada antes de retornar el valor.

#### **42. Función:** float obtener\_flotante(const char \*mensaje)

- **Tipo de Retorno:** float
- **Parámetros que Recibe:** const char \*mensaje (puntero a una cadena de caracteres)
- **Variables locales:**
  - float valor;

- **Descripción:** Esta función muestra un mensaje y obtiene un número flotante del usuario.

Verifica que la entrada sea válida y limpia el buffer de entrada antes de retornar el valor.

**43. Función:** void obtener\_cadena(const char \*mensaje, char \*cadena, int longitud\_maxima)

- **Tipo de Retorno:** void
- **Parámetros que Recibe:**
  - const char \*mensaje (puntero a una cadena de caracteres)
  - char \*cadena (puntero a una cadena de caracteres)
  - int longitud\_maxima (longitud máxima de la cadena)
- **Variables locales:** Ninguna
- **Descripción:** Esta función muestra un mensaje y obtiene una cadena de caracteres del usuario. Elimina el salto de línea al final de la entrada y maneja posibles errores de lectura.

**44. Función:** void presionar\_enter\_para\_continuar()

- **Tipo de Retorno:** void
- **Parámetros que Recibe:** Ninguno
- **Variables locales:** Ninguna

- **Descripción:** Esta función muestra un mensaje indicando que se debe presionar ENTER para continuar. Espera a que el usuario presione ENTER.

**45. Función:** int buscar\_producto\_por\_codigo(int codigo)

- **Tipo de Retorno:** int
- **Parámetros que Recibe:** int codigo (código del producto a buscar)
- **Variables locales:**
  - int i;
- **Descripción:** Esta función busca un producto por su código en el array de productos. Retorna el índice del producto si lo encuentra, o -1 si no lo encuentra.

**46. Función:** int buscar\_cliente\_por\_cedula(int cedula)

- **Tipo de Retorno:** int
- **Parámetros que Recibe:** int cedula (cédula del cliente a buscar)
- **Variables locales:**
  - int i;
- **Descripción:** Esta función busca un cliente por su cédula en el array de clientes. Retorna el índice del cliente si lo encuentra, o -1 si no lo encuentra.

**47. Función:** int buscar\_trabajador\_por\_numero\_empleado(int numero\_empleado)

- **Tipo de Retorno:** int
- **Parámetros que Recibe:** int numero\_empleado (número de empleado del trabajador a buscar)
- **Variables locales:**
  - int i;
- **Descripción:** Esta función busca un trabajador por su número de empleado en el array de trabajadores. Retorna el índice del trabajador si lo encuentra, o -1 si no lo encuentra.

**48. Función:** int buscar\_caja\_por\_numero(int numero\_caja)

- **Tipo de Retorno:** int
- **Parámetros que Recibe:** int numero\_caja (número de la caja a buscar)
- **Variables locales:**
  - int i;
- **Descripción:** Esta función busca una caja por su número en el array de cajas. Retorna el índice de la caja si la encuentra, o -1 si no la encuentra.

**49. Función:** int buscar\_factura\_por\_numero(int numero\_factura)

- **Tipo de Retorno:** int
- **Parámetros que Recibe:** int numero\_factura (número de la factura a buscar)



- **Variables locales:**

- `int i;`

- **Descripción:** Esta función busca una factura por su número en el array de facturas.

Retorna el índice de la factura si la encuentra, o -1 si no la encuentra.

**50. Procedimiento:** `void limpiar_consola()`

- **Tipo de Retorno:** void

- **Parámetros que Recibe:** Ninguno

- **Variables locales:** Ninguna

- **Descripción:** Esta función limpia la consola. Funciona en sistemas operativos Windows y Unix.

**51. Procedimiento:** `void guardar_reporte_txt(const char *nombre_archivo, const char *contenido)`

- **Tipo de Retorno:** void

- **Parámetros que Recibe:**

- `const char *nombre_archivo` (nombre del archivo donde se guardará el reporte)

- `const char *contenido` (contenido del reporte)

- **Variables locales:**

- FILE \*archivo\_reporte;
- **Descripción:** Esta función guarda un reporte en un archivo de texto. Si ocurre un error al abrir el archivo, muestra un mensaje de error.

**52. Función:** char\* obtener\_fecha\_hora\_actual()

- **Tipo de Retorno:** char\*
- **Parámetros que Recibe:** Ninguno
- **Variables locales:**
  - time\_t tiempo\_actual\_unix;
  - struct tm\* tiempo\_info;
- **Descripción:** Esta función obtiene la fecha y hora actual en formato YYYY-MM-DD HH:MM:SS y la retorna como una cadena de caracteres. Reserva memoria para la cadena antes de retornarla.

## **Descripciones De Las Validaciones A Considerar**

En el Proyecto: Hipermecado han sido implementadas ciertas funciones con el fin de que realicen validaciones, esto para garantizar el manejo óptimo de los datos requeridos y prevenir cualquier resultado indeseado. Las validaciones que a continuación se muestran están descritas más técnicamente en el apartado de “Tabla De Librerías, Funciones Y Procedimientos Utilizados”, donde están ubicados entre las funciones cincuenta y uno a cincuenta y nueve.

### **1. Validación de Enteros:**

- Función: `int validar_entero(char input[])`
- Descripción: Se asegura de que la cadena de caracteres ingresada por el usuario represente un número entero positivo. Si la cadena contiene algún carácter no numérico, la función devuelve -1 para indicar un error.
- Uso: Esta validación se utiliza para validar códigos de productos, cantidades de productos, números de empleados, entre otros.

### **2. Validación de Flotantes:**

Función: ``float validar_flotante(char input[])``

- Descripción: Verifica que la cadena de caracteres ingresada represente un número flotante positivo. La función permite un punto decimal y devuelve -1.0 en caso de error.
- Uso: Se utiliza para validar precios en Bolívares y otras cantidades que pueden ser decimales.

### 3. Validación de Códigos de Producto:

- Función: ``int validar_codigo_producto(char input[])``
- Descripción: Asegura que el código del producto ingresado sea un entero positivo. Llama a ``validar_entero`` y devuelve el código si es válido, o -1 si no lo es.
- Uso: Se utiliza al agregar y modificar productos para garantizar que los códigos sean válidos.

### 4. Validación de Cantidades de Producto:

- Función: ``int validar_cantidad_producto(char input[])``
- Descripción: Garantiza que la cantidad de productos ingresada sea un entero positivo. Llama a ``validar_entero`` y devuelve la cantidad si es válida, o -1 si no lo es.
- Uso: Se utiliza al agregar y modificar productos y al ingresar la cantidad de productos en las facturas.

### 5. Validación de Precios en Bolívars:

- Función: ``float validar_precio_bs(char input[])``
- Descripción: Verifica que el precio en Bolívars ingresado sea un número flotante positivo. Llama a ``validar_flotante`` y devuelve el precio si es válido, o -1.0 si no lo es.
- Uso: Se utiliza al agregar y modificar productos para asegurar que los precios sean válidos.

## 6. Validación de Números de Empleado:

- Función: ``int validar_numero_empleado(char input[])``
- Descripción: Asegura que el número de empleado ingresado sea un entero positivo.  
Llama a ``validar_entero`` y devuelve el número de empleado si es válido, o -1 si no lo es.
- Uso: Se utiliza al agregar y modificar trabajadores para garantizar que los números de empleado sean válidos.

## 7. Validación de Cédulas:

- Función: ``char* validar_cedula(char input[])``
- Descripción: Asegura que la cédula ingresada tenga el formato correcto (ej: V-12345678 o E-1234567). Verifica que el primer carácter sea 'V' o 'E', seguido de un guion y una serie de dígitos. Devuelve la cédula si es válida, o NULL si no lo es.
- Uso: Se utiliza al agregar y modificar clientes y trabajadores para garantizar que las cédulas sean válidas.

## 8. Validación de Teléfonos:

- Función: ``char* validar_telefono(char input[])``
- Descripción: Verifica que el teléfono ingresado tenga el formato correcto (ej: 0414-1234567). Comprueba que los primeros cuatro caracteres y los últimos siete sean dígitos, separados por un guion. Devuelve el teléfono si es válido, o NULL si no lo es.

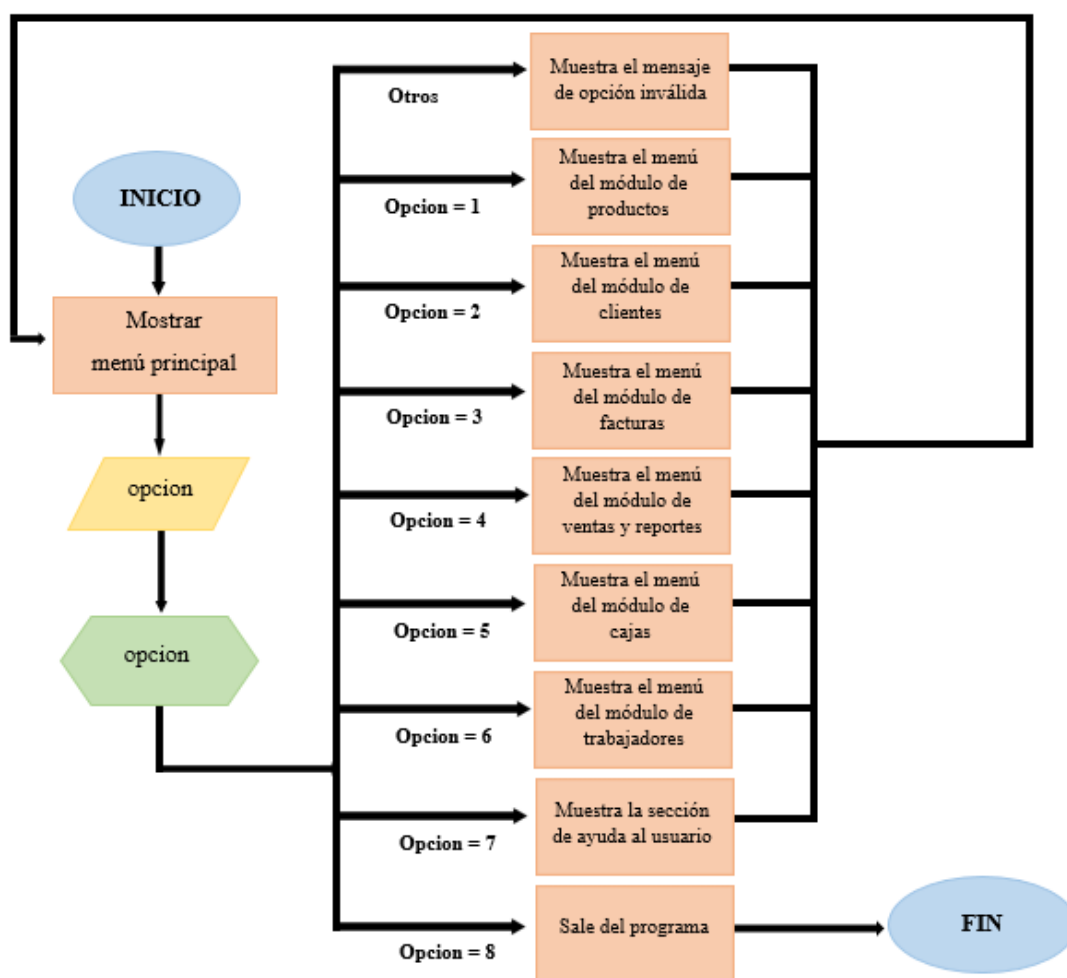
- Uso: Se utiliza al agregar y modificar clientes y trabajadores para asegurar que los números de teléfono sean válidos.

## **9. Validación de Fechas:**

- Función: ``char* validar_fecha(char input[])``
- Descripción: Asegura que la fecha ingresada tenga el formato correcto (DD/MM/AAAA). Verifica que los días estén entre 1 y 31 y los meses entre 1 y 12. Devuelve la fecha si es válida, o NULL si no lo es.
- Uso: Se utiliza al agregar facturas para garantizar que las fechas sean válidas.

Estas validaciones previenen errores y aseguran que las operaciones realizadas con los datos sean correctas y precisas.

### Diagrama De Flujo Del Algoritmo Principal



## Pantallas De Cada Una De Las Corridas Del Programa

Menú principal:

```
=====
                        Hipermercado Cachamay - MENU PRINCIPAL
=====
RIF: J-123456789 - Telefono: 0414-0463588
=====
1. Modulo Productos
2. Modulo Clientes
3. Modulo Facturas
4. Modulo Ventas y Reportes
5. Modulo Cajas
6. Modulo Trabajadores
7. Ayuda al Usuario
8. Salir
=====
Seleccione una opcion:
```

1. Módulo productos:

```
===== Modulo Productos =====
1. Crear Producto
2. Modificar Producto
3. Eliminar Producto
4. Mostrar Producto
5. Mostrar Todos los Productos
6. Regresar al Menu Principal
=====
Seleccione una opcion:
```

1.1 Crear producto:



```
===== Modulo Productos =====  
1. Crear Producto  
2. Modificar Producto  
3. Eliminar Producto  
4. Mostrar Producto  
5. Mostrar Todos los Productos  
6. Regresar al Menu Principal  
=====  
Seleccione una opcion: 1  
Limite de productos: 100  
¿Cuantos productos desea agregar?: 5
```

```
===== Nuevo Producto (1 de 5) =====  
Ingrese el codigo del producto (Ejemplo: 101): 1111  
Ingrese el nombre del producto: Mouse chafa  
Ingrese la cantidad disponible: 50  
Ingrese el precio en Bolivares (Bs): 300  
Producto 'Mouse chafa' agregado exitosamente.  
Presione ENTER para continuar...
```

```
===== Nuevo Producto (5 de 5) =====  
Ingrese el codigo del producto (Ejemplo: 101): 5555  
Ingrese el nombre del producto: Resma de papel tamano carta  
Ingrese la cantidad disponible: 10  
Ingrese el precio en Bolivares (Bs): 4000  
Producto 'Resma de papel tamano carta' agregado exitosamente.  
Presione ENTER para continuar...  
¿Desea regresar al menu principal? (s/n):
```

## 1.2 Modificar producto:

```

===== Modulo Productos =====
1. Crear Producto
2. Modificar Producto
3. Eliminar Producto
4. Mostrar Producto
5. Mostrar Todos los Productos
6. Regresar al Menu Principal
=====
Seleccione una opcion: 2
Ingrese el codigo del producto a modificar: 222

```

```

===== Modificar Producto =====
Producto actual: Headset HP (Codigo: 222)
Ingrese el nuevo nombre del producto (dejar en blanco para no modificar):
Ingrese la nueva cantidad disponible (ingrese -1 para no modificar): -1
Ingrese el nuevo precio en Bolivares (ingrese -1 para no modificar): 5500
Producto modificado exitosamente.
Presione ENTER para continuar...
¿Desea regresar al menu principal? (s/n):

```

### 1.3 Eliminar producto:

```

===== Modulo Productos =====
1. Crear Producto
2. Modificar Producto
3. Eliminar Producto
4. Mostrar Producto
5. Mostrar Todos los Productos
6. Regresar al Menu Principal
=====
Seleccione una opcion: 3
Ingrese el codigo del producto a eliminar: 1111
Producto con codigo 1111 eliminado exitosamente.
Presione ENTER para continuar...
¿Desea regresar al menu principal? (s/n):

```

### 1.4 Mostrar Producto:

```

===== Modulo Productos =====
1. Crear Producto
2. Modificar Producto
3. Eliminar Producto
4. Mostrar Producto
5. Mostrar Todos los Productos
6. Regresar al Menu Principal
=====
Seleccione una opcion: 4
Ingrese el codigo del producto a mostrar: 222

```

```

===== Detalles del Producto =====
Codigo: 222
Nombre:
Cantidad Disponible: 20
Precio (Bs): 5500.00
Precio (USD): 157.14
Precio (COP): 550000.00
Precio (EUR): 137.50
=====
Presione ENTER para continuar...
¿Desea regresar al menu principal? (s/n):

```

### 1.5 Mostrar todos los productos:

```

===== Listado de Productos =====
=====
Codigo    Nombre                Cantidad    Precio Bs    Precio USD    Precio COP    Precio EUR
=====
222      Headset                20          5500.00      157.14        550000.00     137.50
3333     Carpeta marron         40           20.00         0.57         2000.00        0.50
444      Control TV             50          1000.00       28.57        100000.00     25.00
5555     Resma de papel tamaño carta 10          4000.00      114.29        400000.00     100.00
=====
Presione ENTER para continuar...
¿Desea regresar al menu principal? (s/n):

```

## 2. Módulo clientes:

```

===== Modulo Clientes =====
1. Crear Cliente
2. Modificar Cliente
3. Eliminar Cliente
4. Mostrar Cliente
5. Mostrar Todos los Clientes
6. Regresar al Menu Principal
=====
Seleccione una opcion:

```

## 2.2 Crear cliente:

```

===== Nuevo Cliente (1 de 3) =====
Ingrese la cedula del cliente (Ejemplo: 12345678): 32503797
Ingrese el nombre del cliente: Moira
Ingrese el apellido del cliente: Rojas
Ingrese el telefono del cliente (Ejemplo: 04141234567): 04266939566
Ingrese la direccion del cliente: Villa Colombia
Cliente 'Moira Rojas' agregado exitosamente.
Presione ENTER para continuar...

```

```

===== Nuevo Cliente (3 de 3) =====
Ingrese la cedula del cliente (Ejemplo: 12345678): 30200100
Ingrese el nombre del cliente: Gerald
Ingrese el apellido del cliente: Lopez
Ingrese el telefono del cliente (Ejemplo: 04141234567): 04120009999
Ingrese la direccion del cliente: Alta Vista
Cliente 'Gerald Lopez' agregado exitosamente.
Presione ENTER para continuar...
¿Desea regresar al menu principal? (s/n):

```

## 2.3 Modificar cliente:

```

===== Modulo Clientes =====
1. Crear Cliente
2. Modificar Cliente
3. Eliminar Cliente
4. Mostrar Cliente
5. Mostrar Todos los Clientes
6. Regresar al Menu Principal
=====
Seleccione una opcion: 2
Ingrese la cedula del cliente a modificar: 32503797

```

```

===== Modificar Cliente =====
Cliente actual: Moira Rojas (Cedula: 32503797)
Ingrese el nuevo nombre del cliente (dejar en blanco para no modificar): Alejandra
Ingrese el nuevo apellido del cliente (dejar en blanco para no modificar): Romero
Ingrese el nuevo telefono del cliente (dejar en blanco para no modificar): 04246939566
Ingrese la nueva direccion del cliente (dejar en blanco para no modificar): Villa Brasil
Cliente modificado exitosamente.
Presione ENTER para continuar...
¿Desea regresar al menu principal? (s/n):

```

## 2.4 Mostrar cliente:

```

===== Modulo Clientes =====
1. Crear Cliente
2. Modificar Cliente
3. Eliminar Cliente
4. Mostrar Cliente
5. Mostrar Todos los Clientes
6. Regresar al Menu Principal
=====
Seleccione una opcion: 4
Ingrese la cedula del cliente a mostrar: 32503797

```

```

===== Detalles del Cliente =====
Cedula: 32503797
Nombre: Alejandra
Apellido: Romero
Telefono: 04246939566
Direccion: Villa Brasil
=====
Presione ENTER para continuar...
¿Desea regresar al menu principal? (s/n):

```

## 2.5 Mostrar todos los clientes:

```

===== Listado de Clientes =====
=====
Cedula          Nombre          Apellido          Telefono          Direccion
=====
32503797        Alejandra        Romero            04246939566      Villa Brasil
32456789        Diego            Barrios           04140987654      Villa Asia
30200100        Gerald          Lopez             04120009999      Alta Vista
=====
Presione ENTER para continuar...

```

## 3. Módulo facturas:

```

===== Modulo Facturas =====
1. Crear Factura
2. Mostrar Factura
3. Mostrar Ultimas 10 Facturas
4. Regresar al Menu Principal
=====
Seleccione una opcion:

```

### 3.1 Crear factura:

```

===== Listado de Cajas =====
=====
Numero de Caja  Empleado Responsable  Nombre Empleado  Total Bs Facturado  Total USD Facturado  Total COP Facturado  Total EUR Facturado
=====
3                3                    Poe              6380.00             0.00                 0.00                 0.00
2                2                    Eris             0.00                 0.00                 0.00                 0.00
4                4                    Fernaco          0.00                 0.00                 0.00                 0.00
=====
Presione ENTER para continuar...
¿Desea regresar al menu principal? (s/n): n
1. Ingrese el NUMERO DE CAJA para esta factura: 2
   Caja 2 seleccionada. Cajero responsable: Eris

--- 2. DATOS DEL CLIENTE ---
Ingrese la CEDULA del cliente:
Cedula del Cliente: 32503797
Cliente: Moira Rojas

--- 3. AGREGAR PRODUCTO A LA FACTURA ---
Ingrese el CODIGO del producto:
Codigo del Producto (o 0 para TERMINAR): 1
¡PRODUCTO NO ENCONTRADO! No existe producto con codigo 1. Intente de nuevo.

--- 3. AGREGAR PRODUCTO A LA FACTURA ---
Ingrese el CODIGO del producto:
Codigo del Producto (o 0 para TERMINAR): 222
Cantidad a facturar de 'Headset': 1
Producto 'Headset' agregado a la factura.
¿Agregar OTRO producto a la factura? (s/n):

```

### 3.2 Mostrar factura:

```

===== Modulo Facturas =====
1. Crear Factura
2. Mostrar Factura
3. Mostrar Ultimas 10 Facturas
4. Regresar al Menu Principal
=====
Seleccione una opcion: 2
Ingrese el numero de factura a mostrar: 1

```

```

===== FACTURA GENERADA =====
Hipermercado Cachamay
RIF: J-123456789 - Telefono: 0414-0463588
=====
Factura No: 000004
Fecha: 23/02/2025 19:48:34
Caja No: 2 - Empleado: Eris (Empleado No: 2)
----- Datos del Cliente -----
Nombre: Moira Rojas
Cedula: 32503797
----- Productos -----
Cantidad  Descripcion                Precio Unitario Subtotal
-----
1          Headset                  5500.00      5500.00
-----
Subtotal:                5500.00 Bs
IVA (16%):                880.00 Bs
=====
Total a Pagar:            6380.00 Bs
=====
Presione ENTER para continuar...¿Desea regresar al menu principal? (s/n):

```

### 3.3 Mostrar ultimas 10 facturas:

```

===== Modulo Facturas =====
1. Crear Factura
2. Mostrar Factura
3. Mostrar Ultimas 10 Facturas
4. Regresar al Menu Principal
=====
Seleccione una opcion: 3

```

```

===== Ultimas 10 Facturas Generadas =====
----- Factura No: 000002 -----
Hipermercado Cachamay
RIF: J-123456789 - Telefono: 0414-0463588
=====
Factura No: 000002
Fecha: 23/02/2025 19:47:24
Caja No: 3 - Empleado: Poe (Empleado No: 3)
----- Datos del Cliente -----
Nombre: Moira Rojas
Cedula: 32503797
----- Productos -----
Cantidad  Descripcion                Precio Unitario Subtotal
-----
1          Headset                  5500.00      5500.00
-----
Subtotal:                5500.00 Bs
IVA (16%):                880.00 Bs
=====
Total a Pagar:            6380.00 Bs
=====
----- Factura No: 000004 -----
Hipermercado Cachamay
RIF: J-123456789 - Telefono: 0414-0463588
=====
Factura No: 000004
Fecha: 23/02/2025 19:48:34
Caja No: 2 - Empleado: Eris (Empleado No: 2)
----- Datos del Cliente -----
Nombre: Moira Rojas
Cedula: 32503797
----- Productos -----
Cantidad  Descripcion                Precio Unitario Subtotal
-----
1          Headset                  5500.00      5500.00
-----
Subtotal:                5500.00 Bs
IVA (16%):                880.00 Bs
=====
Total a Pagar:            6380.00 Bs
=====
Presione ENTER para continuar...

```

#### 4. Módulo ventas y reportes:



```

===== Modulo Ventas y Reportes =====
1. Consultar Ventas Diarias por Caja
2. Consultar Ventas Mensuales por Caja
3. Mostrar Total Ventas Diarias
4. Mostrar Total Ventas Mensuales
5. Mostrar Producto Mas Vendido
6. Mostrar Cantidad de Trabajadores del Hipermercado
7. Mostrar Cantidad de Clientes con Compras Realizadas
8. Regresar al Menu Principal
=====
Seleccione una opcion: █

```

#### 4.1 Consultar ventas diarias por caja:

```

===== Listado de Cajas =====
Numero de Caja Empleado Responsable Nombre Empleado Total Bs Facturado Total USD Facturado Total COP Facturado Total EUR Facturado
-----
3 3 Poe 6380.00 0.00 0.00 0.00
2 2 Eris 6380.00 0.00 0.00 0.00
4 4 Fernaco 0.00 0.00 0.00 0.00
=====
Presione ENTER para continuar...
¿Desea regresar al menu principal? (s/n): n
Ingrese el numero de caja para consultar ventas diarias: 3

```

```

===== Reporte Ventas Diarias Caja No 3 - Fecha: 23/02/2025 =====
Caja No: 3 - Empleado Responsable: Poe
Total Ventas del Dia: 6380.00 Bs
Cantidad de Facturas Emitidas: 1
=====
Presione ENTER para continuar...
Reporte de ventas diarias por caja guardado en 'reporte_ventas_diarias_caja.txt'
Presione ENTER para continuar...
¿Desea regresar al menu principal? (s/n): n

```

#### 4.2 Consultar ventas mensuales por caja:

```

===== Listado de Cajas =====
Numero de Caja Empleado Responsable Nombre Empleado Total Bs Facturado Total USD Facturado Total COP Facturado Total EUR Facturado
-----
3 3 Poe 6380.00 0.00 0.00 0.00
2 2 Eris 6380.00 0.00 0.00 0.00
4 4 Fernaco 0.00 0.00 0.00 0.00
=====
Presione ENTER para continuar...
¿Desea regresar al menu principal? (s/n): n
Ingrese el numero de caja para consultar ventas mensuales: 2

```

```

===== Reporte Ventas Mensuales Caja No 2 - Mes: 02/2025 =====
Caja No: 2 - Empleado Responsable: Eris
Total Ventas del Mes: 6380.00 Bs
Cantidad de Facturas Emitidas: 1
=====
Presione ENTER para continuar...
Reporte de ventas mensuales por caja guardado en 'reporte_ventas_mensuales_caja.txt'
Presione ENTER para continuar...
¿Desea regresar al menu principal? (s/n):

```

#### 4.3 Mostrar total de ventas diarias:

```

===== Reporte Total Ventas Diarias - Fecha: 23/02/2025 =====
Total Ventas Diarias del Hipermercado: 12760.00 Bs
=====
Presione ENTER para continuar...
Reporte de total de ventas diarias guardado en 'reporte_total_ventas_diarias.txt'
Presione ENTER para continuar...
¿Desea regresar al menu principal? (s/n):

```

#### 4.4 Mostrar total de ventas mensuales:

```

===== Reporte Total Ventas Mensuales - Mes: 02/2025 =====
Total Ventas Mensuales del Hipermercado: 12760.00 Bs
=====
Presione ENTER para continuar...
Reporte de total de ventas mensuales guardado en 'reporte_total_ventas_mensuales.txt'
Presione ENTER para continuar...
¿Desea regresar al menu principal? (s/n):

```

#### 4.5 Mostrar producto más vendido:

```

===== Reporte Producto Mas Vendido =====
Producto mas vendido: Headset (Codigo: 222)
Cantidad total vendida: 2 unidades
Presione ENTER para continuar...
Reporte de producto mas vendido guardado en 'reporte_producto_mas_vendido.txt'
Presione ENTER para continuar...
¿Desea regresar al menu principal? (s/n):

```

#### 4.6 Mostrar total de trabajadores registrados:

```

===== Cantidad de Trabajadores del Hipermercado Cachamay =====
Cantidad total de trabajadores registrados: 4
=====
Presione ENTER para continuar...

```

#### 4.7 Mostrar cantidad de clientes con comprar realizadas:

```
===== Reporte Clientes con Compras Realizadas =====
Cantidad de clientes con compras en Hipermercado Cachamay: 1
=====
Presione ENTER para continuar...
Reporte de clientes con compras guardado en 'reporte_clientes_con_compras.txt'
Presione ENTER para continuar...
¿Desea regresar al menu principal? (s/n):
```

```
===== Reporte Clientes con Compras Realizadas =====
Cantidad de clientes con compras en Hipermercado Cachamay: 1
=====
Presione ENTER para continuar...
```

#### 5. Módulo de cajas:

```
===== Modulo Cajas =====
1. Crear Caja
2. Modificar Caja
3. Eliminar Caja
4. Mostrar Caja
5. Mostrar Todas las Cajas
6. Regresar al Menu Principal
=====
Seleccione una opcion:
```

##### 5.1 Crear caja:

```
===== Modulo Cajas =====
1. Crear Caja
2. Modificar Caja
3. Eliminar Caja
4. Mostrar Caja
5. Mostrar Todas las Cajas
6. Regresar al Menu Principal
=====
Seleccione una opcion: 1
Limite de cajas: 5
¿Cuántas cajas desea agregar?: 2
```

```

===== Nueva Caja (1 de 2) =====
Ingrese el numero de caja (Ejemplo: 1): 1
Ingrese el numero de empleado responsable de la caja: 1
Caja numero 1 agregada exitosamente.
Presione ENTER para continuar...

```

## 5.2 Modificar caja:

```

===== Modulo Cajas =====
1. Crear Caja
2. Modificar Caja
3. Eliminar Caja
4. Mostrar Caja
5. Mostrar Todas las Cajas
6. Regresar al Menu Principal
=====
Seleccione una opcion: 2
Ingrese el numero de caja a modificar: 3

```

```

===== Modificar Caja =====
Caja actual: Numero 3, Responsable: Poe (Numero de Empleado: 3)
Ingrese el nuevo numero de empleado responsable de la caja (ingrese -1 para no modificar): -1
Caja modificada exitosamente.
Presione ENTER para continuar...
¿Desea regresar al menu principal? (s/n):

```

## 5.3 Eliminar caja:

```

===== Modulo Cajas =====
1. Crear Caja
2. Modificar Caja
3. Eliminar Caja
4. Mostrar Caja
5. Mostrar Todas las Cajas
6. Regresar al Menu Principal
=====
Seleccione una opcion: 3
Ingrese el numero de caja a eliminar: 1
Caja con numero 1 eliminada exitosamente.
Presione ENTER para continuar...

```

## 5.4 Mostrar caja:

```

===== Modulo Cajas =====
1. Crear Caja
2. Modificar Caja
3. Eliminar Caja
4. Mostrar Caja
5. Mostrar Todas las Cajas
6. Regresar al Menu Principal
=====
Seleccione una opcion: 4
Ingrese el numero de caja a mostrar: 2

```

```

===== Detalles de la Caja =====
Numero de Caja: 2
Numero de Empleado Responsable: 2
Nombre del Empleado Responsable: Eris
Total Facturado (Bs): 0.00
Total Facturado (USD): 0.00
Total Facturado (COP): 0.00
Total Facturado (EUR): 0.00
=====
Presione ENTER para continuar...

```

### 5.5 Mostrar todas las cajas:

```

===== Listado de Cajas =====

```

Numero de Caja	Empleado Responsable	Nombre Empleado	Total Bs Facturado	Total USD Facturado	Total COP Facturado	Total EUR Facturado
3	3	Poe	0.00	0.00	0.00	0.00
2	2	Eris	0.00	0.00	0.00	0.00
4	4	Fernaco	0.00	0.00	0.00	0.00

```

=====
Presione ENTER para continuar...

```

### 6. Módulo trabajadores:

```

===== Modulo Trabajadores =====
1. Crear Trabajador
2. Modificar Trabajador
3. Eliminar Trabajador
4. Mostrar Trabajador
5. Mostrar Todos los Trabajadores
6. Regresar al Menu Principal
=====
Seleccione una opcion: 1
Limite de trabajadores: 20
¿Cuantos trabajadores desea agregar?: 3

```

#### 6.1 Crear trabajador:

```

===== Modulo Trabajadores =====
1. Crear Trabajador
2. Modificar Trabajador
3. Eliminar Trabajador
4. Mostrar Trabajador
5. Mostrar Todos los Trabajadores
6. Regresar al Menu Principal
=====
Seleccione una opcion: 1
Limite de trabajadores: 20
¿Cuantos trabajadores desea agregar?: 3

```

```

===== Nuevo Trabajador (1 de 3) =====
Ingrese el numero de empleado (Ejemplo: 1): 1
Ingrese la cedula del trabajador (Ejemplo: 12345678): 30000000
Ingrese el nombre del trabajador: Damian
Ingrese el apellido del trabajador: Fox
Ingrese el telefono del trabajador (Ejemplo: 04141234567): 04261111111
Ingrese la direccion del trabajador: La economica
Trabajador 'Damian Fox' agregado exitosamente.
Presione ENTER para continuar...

```

## 6.2 Modificar trabajador:

```

===== Modulo Trabajadores =====
1. Crear Trabajador
2. Modificar Trabajador
3. Eliminar Trabajador
4. Mostrar Trabajador
5. Mostrar Todos los Trabajadores
6. Regresar al Menu Principal
=====
Seleccione una opcion: 2
Ingrese el numero de empleado del trabajador a modificar: 3

```

```

===== Modificar Trabajador =====
Trabajador actual: Poe Verne Verne (Numero de Empleado: 3)
Ingrese el nuevo nombre del trabajador (dejar en blanco para no modificar): Poe
Ingrese el nuevo apellido del trabajador (dejar en blanco para no modificar): Verne
Ingrese el nuevo telefono del trabajador (dejar en blanco para no modificar): 04140001113
Ingrese la nueva direccion del trabajador (dejar en blanco para no modificar): Villa Alianza
Ingrese la nueva cedula del trabajador (ingrese -1 para no modificar): -1
Trabajador modificado exitosamente.
Presione ENTER para continuar...
¿Desea regresar al menu principal? (s/n):

```

### 6.3 Eliminar trabajador:

```
===== Modulo Trabajadores =====
1. Crear Trabajador
2. Modificar Trabajador
3. Eliminar Trabajador
4. Mostrar Trabajador
5. Mostrar Todos los Trabajadores
6. Regresar al Menu Principal
=====
Seleccione una opcion: 3
Ingrese el numero de empleado del trabajador a eliminar: 2
Trabajador con numero de empleado 2 eliminado exitosamente.
Presione ENTER para continuar...
```

### 6.4 Mostrar trabajador:

```
===== Modulo Trabajadores =====
1. Crear Trabajador
2. Modificar Trabajador
3. Eliminar Trabajador
4. Mostrar Trabajador
5. Mostrar Todos los Trabajadores
6. Regresar al Menu Principal
=====
Seleccione una opcion: 4
Ingrese el numero de empleado del trabajador a mostrar: 1
```

```
===== Detalles del Trabajador =====
Numero de Empleado: 1
Cedula: 30000000
Nombre: Damian
Apellido: Fox
Telefono: 04261111111
Direccion: La economica
=====
Presione ENTER para continuar...
```

### 6.5 Mostrar todos los trabajadores:

```

===== Listado de Trabajadores =====
=====
Numero de Empleado  Cedula      Nombre      Apellido      Telefono      Direccion
=====
1                   30000000    Damian      Fox           04261111111  La economica
3                   28000000    Poe         Verne         04140001113  Villa Alianza
=====
Presione ENTER para continuar...

```

## Manual De Usuario

Bienvenido al sistema de gestión del Hipermercado Cachamay. Este sistema te permitirá administrar eficientemente productos, clientes, trabajadores, cajas y facturas, así como consultar ventas y generar reportes detallados. Este manual te guiará paso a paso en el uso del sistema.

Al iniciar el sistema, se te presentará el menú principal con las siguientes opciones:

1. Módulo Productos
2. Módulo Clientes
3. Módulo Facturas
4. Módulo Ventas y Reportes
5. Módulo Cajas
6. Módulo Trabajadores
7. Ayuda al Usuario
8. Salir

Para seleccionar una de esas, introduce el número que corresponda a la opción mediante el teclado, y selecciona enter. A continuación, las indicaciones pertinentes de cada módulo:

### Módulo de Productos



En este módulo, usted podrá controlar los productos del Hipermercado. Las opciones que tiene disponibles son las siguientes:

### 1. Crear Producto:

1.1 Selecciona "Crear Producto" desde el menú de productos.

1.2 Ingresa la información solicitada: código, nombre, cantidad disponible y precio en bolívares.

1.3 El sistema calculará automáticamente los precios en otras monedas (USD, COP y EUR).

1.4 El producto se agregará al sistema.

### 2. Modificar Producto

2.1 Selecciona "Modificar Producto" desde el menú de productos.

2.2 Ingresa el código del producto a modificar.

2.3 Actualiza la información que desees (nombre, cantidad, precio).

2.4 El sistema recalculará los precios en otras monedas si el precio en bolívares se modifica.

### 3. Eliminar Producto

3.1 Selecciona "Eliminar Producto" desde el menú de productos.

3.2 Ingresa el código del producto a eliminar.

3.3 El producto se eliminará del sistema.

#### 4. Mostrar Producto

4.1 Selecciona "Mostrar Producto" desde el menú de productos.

4.2 Ingresa el código del producto a mostrar.

4.3 El sistema mostrará los detalles del producto.

#### 5. Mostrar Todos los Productos

5.1 Selecciona "Mostrar Todos los Productos" desde el menú de productos.

5.2 El sistema mostrará una lista de todos los productos registrados.

**Módulo De Clientes:** Desde aquí, usted podría gestionar los clientes del hipermercado. Las opciones disponibles son:

#### 1. Crear Cliente

1.1 Selecciona "Crear Cliente" desde el menú de clientes.

1.2 Ingresa la información solicitada: cédula, nombre, apellido, teléfono y dirección.

1.3 El cliente se agregará al sistema.

#### 2. Modificar Cliente

2.1 Selecciona "Modificar Cliente" desde el menú de clientes.

2.2 Ingresa la cédula del cliente a modificar.

2.3 Actualiza la información que desees (nombre, apellido, teléfono, dirección).

#### 3. Eliminar Cliente

3.1 Selecciona "Eliminar Cliente" desde el menú de clientes.

3.2 Ingresa la cédula del cliente a eliminar.

3.3 El cliente se eliminará del sistema.

#### 4. Mostrar Cliente

4.1 Selecciona "Mostrar Cliente" desde el menú de clientes.

4.2 Ingresa la cédula del cliente a mostrar.

4.3 El sistema mostrará los detalles del cliente.

#### 5. Mostrar Todos los Clientes

5.1 Selecciona "Mostrar Todos los Clientes" desde el menú de clientes.

5.2 El sistema mostrará una lista de todos los clientes registrados.

### **Módulo De Facturas**

Desde aquí usted podrá gestionar las facturas del hipermercado. Las opciones disponibles son:

#### 1. Crear Factura

1.1 Selecciona "Crear Factura" desde el menú de facturas.

1.2 Ingresa la información solicitada: número de caja, cédula del cliente y productos con sus cantidades.

1.3 El sistema generará la factura y actualizará los inventarios y totales facturados.

#### 2. Mostrar Factura

2.1 Selecciona "Mostrar Factura" desde el menú de facturas.

2.2 Ingresa el número de la factura a mostrar.

2.3 El sistema mostrará los detalles de la factura.

### 3. Mostrar Últimas 10 Facturas

3.1 Selecciona "Mostrar Últimas 10 Facturas" desde el menú de facturas.

3.2 El sistema mostrará una lista de las últimas 10 facturas generadas.

## **Módulo De Ventas Y Reportes**

Desde aquí usted podrá consultar ventas y generar reportes. Las opciones disponibles son:

### 1. Consultar Ventas Diarias por Caja

1.1 Selecciona "Consultar Ventas Diarias por Caja" desde el menú de ventas y reportes.

1.2 Ingresar el número de caja.

1.3 El sistema mostrará un reporte de las ventas diarias de la caja seleccionada.

### 2. Consultar Ventas Mensuales por Caja

2.1 Selecciona "Consultar Ventas Mensuales por Caja" desde el menú de ventas y reportes.

2.2 Ingresar el número de caja.

2.3 El sistema mostrará un reporte de las ventas mensuales de la caja seleccionada.

### 3. Mostrar Total Ventas Diarias

3.1 Selecciona "Mostrar Total Ventas Diarias" desde el menú de ventas y reportes.

3.2 El sistema mostrará un reporte del total de ventas diarias de todo el hipermercado.

### 4. Mostrar Total Ventas Mensuales

4.1 Selecciona "Mostrar Total Ventas Mensuales" desde el menú de ventas y reportes.

4.2 El sistema mostrará un reporte del total de ventas mensuales de todo el hipermercado.

5. Mostrar Producto Más Vendido

5.1 Selecciona "Mostrar Producto Más Vendido" desde el menú de ventas y reportes.

5.2 El sistema mostrará un reporte del producto más vendido en el hipermercado.

6. Mostrar Cantidad de Trabajadores del Hipermercado

6.1 Selecciona "Mostrar Cantidad de Trabajadores del Hipermercado" desde el menú de ventas y reportes.

6.2 El sistema mostrará el total de trabajadores registrados en el hipermercado.

7. Mostrar Cantidad de Clientes con Compras Realizadas

7.1 Selecciona "Mostrar Cantidad de Clientes con Compras Realizadas" desde el menú de ventas y reportes.

7.2 El sistema mostrará el total de clientes que han realizado compras en el hipermercado.

## **Módulo Cajas**

Desde aquí usted podrá gestionar las cajas del hipermercado. Las opciones disponibles son:

1. Crear Caja

1.1 Selecciona "Crear Caja" desde el menú de cajas.

1.2 Ingresa la información solicitada: número de caja y número de empleado responsable.

1.3 La caja se agregará al sistema.

2. Modificar Caja

2.1 Selecciona "Modificar Caja" desde el menú de cajas.

2.2 Ingresar el número de caja a modificar.

2.3 Actualizar el número de empleado responsable.

### 3. Eliminar Caja

3.1 Seleccionar "Eliminar Caja" desde el menú de cajas.

3.2 Ingresar el número de caja a eliminar.

3.3 La caja se eliminará del sistema.

### 4. Mostrar Caja

4.1 Seleccionar "Mostrar Caja" desde el menú de cajas.

4.2 Ingresar el número de caja a mostrar.

4.3 El sistema mostrará los detalles de la caja.

### 5. Mostrar Todas las Cajas

5.1 Seleccionar "Mostrar Todas las Cajas" desde el menú de cajas.

5.2 El sistema mostrará una lista de todas las cajas registradas.

## **Módulo Trabajadores**

Desde aquí usted podrá gestionar los trabajadores del hipermercado. Las opciones disponibles son:

### 1. Crear Trabajador

1.1 Seleccionar "Crear Trabajador" desde el menú de trabajadores.

1.2 Ingresar la información solicitada: número de empleado, cédula, nombre, apellido, teléfono y dirección.

1.3 El trabajador se agregará al sistema.

## 2. Modificar Trabajador

2.1 Selecciona "Modificar Trabajador" desde el menú de trabajadores.

2.2 Ingresa el número de empleado del trabajador a modificar.

2.3 Actualiza la información que desees (nombre, apellido, teléfono, dirección, cédula).

## 3. Eliminar Trabajador

3.1 Selecciona "Eliminar Trabajador" desde el menú de trabajadores.

3.2 Ingresa el número de empleado del trabajador a eliminar.

3.3 El trabajador se eliminará del sistema.

## 4. Mostrar Trabajador

4.1 Selecciona "Mostrar Trabajador" desde el menú de trabajadores.

4.2 Ingresa el número de empleado del trabajador a mostrar.

4.3 El sistema mostrará los detalles del trabajador.

## 5. Mostrar Todos los Trabajadores

5.1 Selecciona "Mostrar Todos los Trabajadores" desde el menú de trabajadores.

5.2 El sistema mostrará una lista de todos los trabajadores registrados.

## **Ayuda al Usuario**

En este apartado, usted contará con una especie de manual de usuario electrónico, el cual le permitirá aclarar dudas rápidas desde el interior del programa.

## **Salir**

Al seleccionar esta opción, usted saldrá del programa y este se cerrará



## **Conclusiones**

El desarrollo del sistema de gestión para el Hipermercado Cachamay ha permitido consolidar un entorno eficiente para la administración de productos, clientes, trabajadores, cajas y facturas. Este proyecto, diseñado en Lenguaje C con interfaz gráfica, ha cumplido con los requerimientos planteados al inicio, abarcando todas las áreas críticas para el control y manejo del hipermercado.

Se logró desarrollar y consolidar un sistema que incluye módulos para la gestión de productos, clientes, trabajadores, cajas y facturas, donde cada módulo permite crear, modificar, eliminar y mostrar datos, facilitando así la operatividad del hipermercado. El sistema también permite la generación de diversos reportes, incluyendo la cantidad de trabajadores, clientes con compras realizadas, producto más vendido y totales de ventas diarias y mensuales. Estos reportes proporcionan información valiosa para la toma de decisiones gerenciales.

Igualmente, se incorporó la funcionalidad de cálculo de precios en diferentes monedas (bolívars, dólares, pesos colombianos y euros), lo cual es esencial para un negocio que opera en una economía dinámica.

El módulo de facturación permite la generación de facturas detalladas, facilitando así el proceso de cobro y la gestión de inventarios y se implementaron consultas de ventas diarias y mensuales por caja, así como el total de ventas del hipermercado, lo cual optimiza el seguimiento y control de las operaciones diarias.

Durante el desarrollo del proyecto, se utilizó una metodología ágil que permitió la adaptación constante a los requisitos y cambios sugeridos. Se trabajó en equipo, lo que facilitó la distribución de tareas y la colaboración efectiva entre los miembros. La implementación de validaciones y pruebas exhaustivas garantizó la calidad y funcionamiento del sistema.

Sabemos que nuestras habilidades deben ampliarse y procurar siempre cumplir con las expectativas planteadas, las cuales intentamos cubrir en la mayor medida posible para nosotros. Como estudiantes y futuros profesionales en esta materia, este proyecto nos ayudó a forjar una visión más precisa sobre nuestras destrezas y debilidades, a la vez que hemos aprendido nuevas maneras de trabajar para futuros proyectos.

En conclusión, el proyecto ha sido exitoso en cumplir con los objetivos planteados, proporcionando al Hipermercado Cachamay un sistema robusto y eficiente para la gestión de sus operaciones diarias. La colaboración y el esfuerzo del equipo de desarrollo han sido fundamentales para alcanzar este logro, y se espera que el sistema contribuya significativamente al crecimiento y optimización del hipermercado, tal y como ha influido en nuestro crecimiento como programadores.

### Referencias Consultadas

Cerrada Somolinos, J., & Collado Machuca, M. (2015). *Fundamentos De Programación*.

Editorial Universitaria Ramón Areces.

Joyanes Aguilar, L., & Zahonero Martínez, I. (2005). *Programación en C: Metodología, algoritmos y estructura de datos (2da ed.)*. McGrawHill.

García Carballeira, F., et al. (2002). *El lenguaje de programación C: diseño e implementación de programas*.

Taboada, Alejandro [Programación ATS] (2 de Abril de 2018) *Programación en C++* [Listado de videos] Youtube

[https://youtube.com/playlist?list=PLWtYZ2ejMVJmUTNE2QVaCd1y\\_6GslOeZ6&si=sJdEyxPhQ0SvnU31](https://youtube.com/playlist?list=PLWtYZ2ejMVJmUTNE2QVaCd1y_6GslOeZ6&si=sJdEyxPhQ0SvnU31)