

**UNIVERSIDAD TECNOLÓGICA DE SANTIAGO
DEPARTAMENTO DE INGENIERIA ELECTRONICA
TALLER PROGRAMACION 1
IET-725-001**

**UNIVERSIDAD TECNOLÓGICA DE SANTIAGO
(UTESA)**



ASIGNATURA

Taller De Programación

TEMA:

Ante Proyecto final

NOMBRES Y MATRICULAS

Elían A. Domínguez Romano

1180536

PROFESOR/A

Raúl A. Torivio

Santiago RD.

Fecha

15/12/2023



Programa para máquina expendedora de snacks

DESCRIPCIÓN DEL PROBLEMA

La ausencia de una máquina de snacks en un entorno determinado concurrido por personas, como una oficina, parques, escuelas o lugar público, puede generar diversas dificultades y frustraciones para las personas necesitan un bocadillo rápido para acortar el tiempo lo más posible o simplemente para el deleite y antojos de comer un snack por esta razón es que se necesita una máquina para satisfacer las necesidades alimenticias de las personas.

OBJETIVO DEL PROYECTO

La instalación de una máquina expendedora de snacks puede tener varios objetivos beneficiosos para distintos entornos. Aquí te presento algunos objetivos comunes por los cuales es necesario tener una máquina expendedora de snacks:

1-Satisfacer Necesidades Inmediatas: Proporcionar una solución rápida y conveniente para satisfacer las necesidades alimenticias inmediatas de las personas en entornos como oficinas, escuelas o lugares públicos.

2-Mejorar la Productividad: Facilitar el acceso a alimentos dentro del lugar de trabajo puede contribuir a mantener altos niveles de energía entre los empleados, lo que, a su vez, puede mejorar la productividad y el rendimiento laboral.

3-Promover un Ambiente Saludable: Ofrecer opciones saludables en una máquina expendedora puede fomentar hábitos alimenticios más saludables entre los individuos, contribuyendo a un ambiente general de bienestar.

4-Incrementar la Satisfacción del Usuario: Proporcionar opciones de snacks variadas y apetitosas puede mejorar la experiencia de los usuarios en el lugar, aumentando su satisfacción y comodidad.

5-Reducir la Necesidad de Salir del Entorno: Al ofrecer opciones de snacks dentro del lugar, se reduce la necesidad de que las personas salgan del entorno para buscar alimentos, lo que ahorra tiempo y mantiene a los individuos en el lugar deseado.

6-Ofrecer Variedad de Opciones: Las máquinas expendedoras pueden proporcionar una amplia variedad de opciones de snacks, adaptándose a diferentes preferencias y necesidades dietéticas, lo que aumenta la satisfacción del consumidor.

7-Apoyar la Conciliación Laboral: En entornos laborales, la disponibilidad de snacks puede contribuir a la conciliación laboral, permitiendo a los empleados disfrutar de pequeñas pausas para recargar energías de manera rápida y eficiente.

RESUMEN TÉCNICO

Este programa en C ha sido diseñado para gestionar eficientemente una máquina expendedora de snacks. Se basa en la implementación de dos estructuras de datos fundamentales: `struct Producto` que almacena información detallada sobre cada producto, y `struct Usuario` que gestiona los datos de los usuarios del sistema. Con el objetivo de mejorar la experiencia del usuario, se han incorporado funciones clave, como la capacidad de iniciar sesión, cambiar contraseñas y agregar nuevos productos a la lista. Además, se han implementado medidas de seguridad, como la verificación de contraseñas y la validación de operaciones para prevenir acciones no autorizadas.

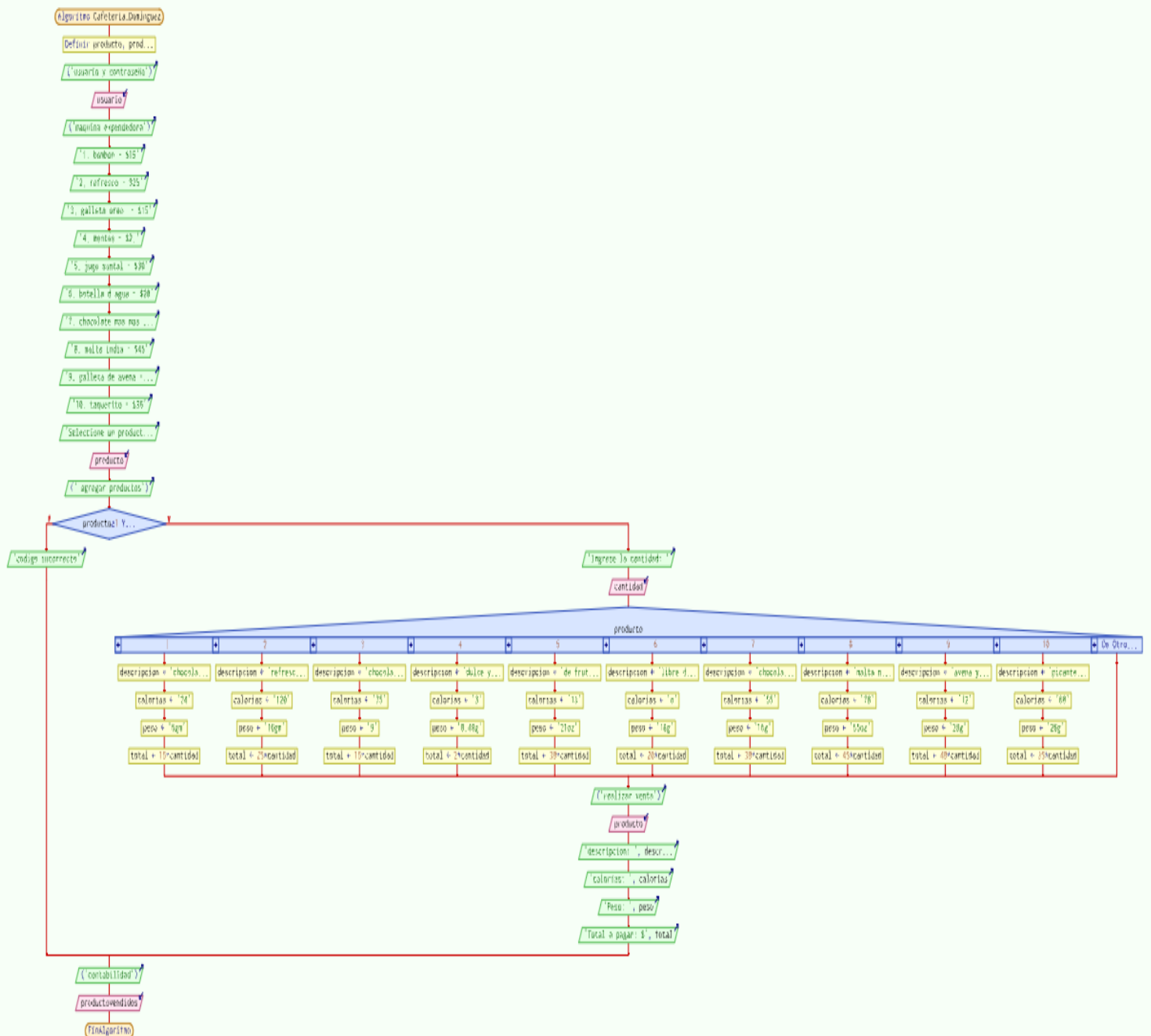
El programa también destaca por su capacidad para cargar y guardar datos desde y hacia archivos de texto, garantizando la persistencia de información sobre productos y usuarios entre diferentes ejecuciones. La interfaz de usuario se realiza a través de la consola, proporcionando una interacción intuitiva al solicitar la información necesaria para llevar a cabo diversas acciones, como realizar ventas, mostrar la contabilidad y visualizar la lista de productos disponibles. La eficiencia y escalabilidad del programa se ven respaldadas por controles que evitan exceder límites, como el número máximo de productos disponibles. En resumen, esta implementación aborda integralmente las necesidades de una máquina expendedora de snacks, incorporando elementos de seguridad, facilidad de uso y gestión eficiente de datos.

Para este programa utilizaremos todos lo aprendido en el aula y como funciones las estructuras, bucles, ficheros, etc.

UNIVERSIDAD TECNOLÓGICA DE SANTIAGO
DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA
TALLER PROGRAMACIÓN 1
IET-725-001

DISEÑO ESQUEMÁTICO

- Diagrama de flujo



UNIVERSIDAD TECNOLOGICA DE SANTIAGO
DEPARTAMENTO DE INGENIERIA ELECTRONICA
TALLER PROGRAMACION 1
IET-725-001

Código fuente:

```
/*-----|
| Autor: Elian A. Dominguez      Matricula: 1-18-0536  |
| Fecha: 15/12/2023             Version: 1.83.1 |
|-----|
| Descripcion del Programa: Este es un programa en C      |
| simulando una maquina de snack donde se tienen 10 productos |
| a seleccionar, dando como resultado el precio, una descripcion |
| el peso y las calorías                                     |
|-----*/

// Incluir E/S y Librerías Standard
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <stdbool.h>

struct Producto {
    char nombre[50];
    char descripcion[100];
    float precio;
    float peso;
    int calorías;
    int cantidad;
};

struct Usuario {
    char nombre[50];
    char contraseña[50];
};

bool buscarUsuario(struct Usuario usuarios[], int numUsuarios, const char* nombre) {
    for (int i = 0; i < numUsuarios; i++) {
        if (strcmp(usuarios[i].nombre, nombre) == 0) {
            return true; // El usuario existe
        }
    }
    return false; // El usuario no existe
}

void cargarProductos(struct Producto productos[], int* numProductos);

void agregarUsuario(struct Usuario usuarios[], int* numUsuarios, const char* nombre, const
char* contraseña) {
    strcpy(usuarios[*numUsuarios].nombre, nombre);
```

UNIVERSIDAD TECNOLÓGICA DE SANTIAGO
DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA
TALLER PROGRAMACIÓN 1
IET-725-001

```
        strcpy(usuarios[*numUsuarios].contrasena, contrasena);
        (*numUsuarios)++;
    }

void guardarUsuarios(struct Usuario usuarios[], int numUsuarios) {
    FILE* archivo = fopen("usuarios.txt", "w");

    if (archivo == NULL) {
        printf("No se pudo abrir el archivo de usuarios.\n\n");
        return;
    }

    for (int i = 0; i < numUsuarios; i++) {
        fprintf(archivo, "Nombre: %s\n", usuarios[i].nombre);
        fprintf(archivo, "Contrasena: %s\n", usuarios[i].contrasena);
        fprintf(archivo, "\n");
    }

    fclose(archivo);
}

void cargarUsuarios(struct Usuario usuarios[], int* numUsuarios) {
    FILE* archivo = fopen("usuarios.txt", "r");

    if (archivo == NULL) {
        return;
    }

    char linea[200];

    while (fgets(linea, sizeof(linea), archivo)) {
        sscanf(linea, "Nombre: %[^\\n]", usuarios[*numUsuarios].nombre);
        fgets(linea, sizeof(linea), archivo);
        sscanf(linea, "Contrasena: %[^\\n]", usuarios[*numUsuarios].contrasena);
        fgets(linea, sizeof(linea), archivo);
        fgets(linea, sizeof(linea), archivo); // Leer la línea en blanco
        (*numUsuarios)++;
    }

    fclose(archivo);
}

void mostrarMenu();
void iniciarSesion(struct Usuario usuarios[], int* numUsuarios, struct Usuario* usuario);
void cambiarContrasena(struct Usuario* usuario);
void agregarProducto(struct Producto productos[], int* numProductos);
void mostrarContabilidad(struct Producto productos[], int numProductos);
```

UNIVERSIDAD TECNOLOGICA DE SANTIAGO
DEPARTAMENTO DE INGENIERIA ELECTRONICA
TALLER PROGRAMACION 1
IET-725-001

```
void mostrarProductos(struct Producto productos[], int numProductos);
void realizarVenta(struct Producto productos[], int numProductos);
void guardarProductos(struct Producto productos[], int numProductos);
void cargarProductos(struct Producto productos[], int* numProductos); // Agrega este
prototipo
```

```
int main() {
    int opcion;
    int numUsuarios = 0;
    struct Usuario usuarios[10]; // Puedes ajustar el tamaño según tus necesidades
    struct Usuario usuario;
    int numProductos = 0;
    struct Producto productos[17];
```

```
    cargarUsuarios(usuarios, &numUsuarios);
    cargarProductos(productos, &numProductos);
```

```
    while (1) {
        mostrarMenu();
        printf("Ingrese una opcion: ");
        scanf("%d", &opcion);
        getchar(); // Vaciar buffer de entrada
```

```
        switch (opcion) {
            case 1:
                iniciarSesion(usuarios, &numUsuarios, &usuario);
                break;
            case 2:
                cambiarContrasena(&usuario);
                break;
            case 3:
                if (strlen(usuario.nombre) == 0) {
                    printf("Debe iniciar sesion primero.\n\n");
                } else {
                    agregarProducto(productos, &numProductos);
                }
                break;
            case 4:
                if (strlen(usuario.nombre) == 0) {
                    printf("Debe iniciar sesion primero.\n\n");
                } else {
                    mostrarContabilidad(productos, numProductos);
                }
                break;
            case 5:
                mostrarProductos(productos, numProductos);
                break;
```

UNIVERSIDAD TECNOLOGICA DE SANTIAGO
DEPARTAMENTO DE INGENIERIA ELECTRONICA
TALLER PROGRAMACION 1
IET-725-001

```
        case 6:
            realizarVenta(productos, numProductos);
            break;
        case 7:
            guardarProductos(productos, numProductos);
            printf("Se han guardado exitosamente los datos.\n\n");
            break;
        case 8:
            guardarUsuarios(usuarios, numUsuarios);
            printf("Se han guardado exitosamente los usuarios.\n");
            printf("Hasta luego.\n");
            return 0;
        default:
            printf("Opcion invalida. Intente de nuevo.\n\n");
    }
}

return 0;
}

void mostrarMenu() {
    printf("---- Menu ----\n");
    printf("1. Iniciar sesion\n");
    printf("2. Cambiar contrasena\n");
    printf("3. Agregar producto\n");
    printf("4. Mostrar contabilidad\n");
    printf("5. Mostrar productos\n");
    printf("6. Realizar venta\n");
    printf("7. Guardar productos\n");
    printf("8. Salir\n\n");
}

void iniciarSesion(struct Usuario usuarios[], int* numUsuarios, struct Usuario* usuario) {
    char nombre[50];
    char contrasena[50];

    printf("Ingrese su nombre de usuario: ");
    fgets(nombre, sizeof(nombre), stdin);
    nombre[strcspn(nombre, "\n")] = '\0'; // Eliminar el salto de linea

    printf("Ingrese su contrasena: ");
    fgets(contrasena, sizeof(contrasena), stdin);
    contrasena[strcspn(contrasena, "\n")] = '\0'; // Eliminar el salto de linea

    // Verificar si el usuario existe
    if (buscarUsuario(usuarios, *numUsuarios, nombre)) {
        // Si el usuario existe, verificar la contraseña
    }
}
```


UNIVERSIDAD TECNOLOGICA DE SANTIAGO
DEPARTAMENTO DE INGENIERIA ELECTRONICA
TALLER PROGRAMACION 1
IET-725-001

```
for (int i = 0; i < *numUsuarios; i++) {
    if (strcmp(usuarios[i].nombre, nombre) == 0 && strcmp(usuarios[i].contrasena,
contrasena) == 0) {
        // Iniciar sesión
        strcpy(usuario->nombre, nombre);
        strcpy(usuario->contrasena, contrasena);
        printf("Sesion iniciada correctamente.\n\n");
        return;
    }
}
// Si la contraseña es incorrecta
printf("Contrasena incorrecta.\n\n");
} else {
    // Si el usuario no existe, ofrecer crear uno nuevo
    char opcion;
    printf("El usuario no existe. ¿Desea crear un nuevo usuario? (S/N): ");
    scanf(" %c", &opcion);
    if (opcion == 'S' || opcion == 's') {
        // Agregar el nuevo usuario
        agregarUsuario(usuarios, numUsuarios, nombre, contrasena);
        strcpy(usuario->nombre, nombre);
        strcpy(usuario->contrasena, contrasena);
        printf("Nuevo usuario creado y sesion iniciada correctamente.\n\n");
    } else {
        printf("Operacion cancelada.\n\n");
    }
}
}

// Resto del código sin cambios

void cambiarContrasena(struct Usuario* usuario) {
    char contrasenaAntigua[50];

    if (strlen(usuario->nombre) == 0) {
        printf("Debe iniciar sesion primero.\n\n");
        return;
    }

    printf("Ingrese su contrasena actual: ");
    fgets(contrasenaAntigua, sizeof(contrasenaAntigua), stdin);
    contrasenaAntigua[strcspn(contrasenaAntigua, "\n")] = '\0'; // Eliminar el salto de linea

    if (strcmp(usuario->contrasena, contrasenaAntigua) == 0) {
        char nuevaContrasena[50];

        printf("Ingrese su nueva contrasena: ");
```

UNIVERSIDAD TECNOLOGICA DE SANTIAGO
DEPARTAMENTO DE INGENIERIA ELECTRONICA
TALLER PROGRAMACION 1
IET-725-001

```
fgets(nuevaContrasena, sizeof(nuevaContrasena), stdin);
nuevaContrasena[strcspn(nuevaContrasena, "\n")] = '\0'; // Eliminar el salto de linea

strcpy(usuario->contrasena, nuevaContrasena);
printf("Contrasena cambiada exitosamente.\n\n");
} else {
    printf("Contrasena incorrecta.\n\n");
}
}

void agregarProducto(struct Producto productos[], int* numProductos) {
    if (*numProductos == 17) {
        printf("Ya se ha alcanzado el maximo de productos.\n\n");
        return;
    }

    struct Producto producto;

    printf("Ingrese el nombre del producto: ");
    fgets(producto.nombre, sizeof(producto.nombre), stdin);
    producto.nombre[strcspn(producto.nombre, "\n")] = '\0'; // Eliminar el salto de linea

    printf("Ingrese la descripcion del producto: ");
    fgets(producto.descripcion, sizeof(producto.descripcion), stdin);
    producto.descripcion[strcspn(producto.descripcion, "\n")] = '\0'; // Eliminar el salto de linea

    printf("Ingrese el precio del producto: ");
    scanf("%f", &producto.precio);
    getchar(); // Vaciar buffer de entrada

    printf("Ingrese el peso del producto: ");
    scanf("%f", &producto.peso);
    getchar(); // Vaciar buffer de entrada

    printf("Ingrese las calorías del producto: ");
    scanf("%d", &producto.calorias);
    getchar(); // Vaciar buffer de entrada

    printf("Ingrese la cantidad disponible del producto: ");
    scanf("%d", &producto.cantidad);
    getchar(); // Vaciar buffer de entrada

    productos[*numProductos] = producto;
    (*numProductos)++;

    printf("Producto agregado correctamente.\n\n");
}
```

UNIVERSIDAD TECNOLOGICA DE SANTIAGO
DEPARTAMENTO DE INGENIERIA ELECTRONICA
TALLER PROGRAMACION 1
IET-725-001

```
void mostrarContabilidad(struct Producto productos[], int numProductos) {
    int cantidadVendida = 0;
    float montoTotal = 0;

    for (int i = 0; i < numProductos; i++) {
        cantidadVendida += productos[i].cantidad - 1;
        montoTotal += productos[i].precio * (productos[i].cantidad - 1);
    }

    printf("Cantidad de productos vendidos: %d\n", cantidadVendida);
    printf("Monto total: %.2f\n\n", montoTotal);
}

void mostrarProductos(struct Producto productos[], int numProductos) {
    printf("---- Productos Disponibles ----\n");

    for (int i = 0; i < numProductos; i++) {
        printf("Nombre: %s\n", productos[i].nombre);
        printf("Descripcion: %s\n", productos[i].descripcion);
        printf("Precio: %.2f\n", productos[i].precio);
        printf("Peso: %.2f\n", productos[i].peso);
        printf("Calorias: %d\n", productos[i].calorias);
        printf("Cantidad: %d\n\n", productos[i].cantidad);
    }
}

void realizarVenta(struct Producto productos[], int numProductos) {
    if (numProductos == 0) {
        printf("No hay productos disponibles para la venta.\n\n");
        return;
    }

    int opcion;
    int cantidad;
    float total = 0;

    printf("---- Realizar Venta ----\n");
    mostrarProductos(productos, numProductos);

    printf("Seleccione el producto a vender (1-%d): ", numProductos);
    scanf("%d", &opcion);

    if (opcion < 1 || opcion > numProductos) {
        printf("Opcion invalida.\n\n");
        return;
    }
}
```

UNIVERSIDAD TECNOLÓGICA DE SANTIAGO
DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA
TALLER PROGRAMACIÓN 1
IET-725-001

```
printf("Ingrese la cantidad a vender: ");
scanf("%d", &cantidad);

if (cantidad <= 0 || cantidad > productos[opcion - 1].cantidad) {
    printf("Cantidad invalida.\n\n");
    return;
}

total = cantidad * productos[opcion - 1].precio;

printf("Total a pagar: %.2f\n", total);

// Proceso de pago
float pago;
printf("Ingrese el monto recibido (billetes de 100 y 50): ");
scanf("%f", &pago);

if (pago < total) {
    printf("Monto insuficiente.\n\n");
    return;
}

// Calcular cambio
float cambio = pago - total;
printf("Cambio a devolver: %.2f\n", cambio);

// Actualizar la cantidad de productos después de la venta
productos[opcion - 1].cantidad -= cantidad;

printf("Venta realizada exitosamente.\n\n");
}

void guardarProductos(struct Producto productos[], int numProductos) {
    FILE* archivo = fopen("productos.txt", "w");

    if (archivo == NULL) {
        printf("No se pudo abrir el archivo de productos.\n\n");
        return;
    }

    for (int i = 0; i < numProductos; i++) {
        fprintf(archivo, "Nombre: %s\n", productos[i].nombre);
        fprintf(archivo, "Descripcion: %s\n", productos[i].descripcion);
        fprintf(archivo, "Precio: %.2f\n", productos[i].precio);
        fprintf(archivo, "Peso: %.2f\n", productos[i].peso);
        fprintf(archivo, "Calorias: %d\n", productos[i].calorias);
    }
}
```

UNIVERSIDAD TECNOLOGICA DE SANTIAGO
DEPARTAMENTO DE INGENIERIA ELECTRONICA
TALLER PROGRAMACION 1
IET-725-001

```
        fprintf(archivo, "Cantidad: %d\n\n", productos[i].cantidad);
    }

    fclose(archivo);
}

void cargarProductos(struct Producto productos[], int* numProductos) {
    FILE* archivo = fopen("productos.txt", "r");

    if (archivo == NULL) {
        return;
    }

    char linea[200];

    while (fgets(linea, sizeof(linea), archivo)) {
        sscanf(linea, "Nombre: %[^\\n]", productos[*numProductos].nombre);
        fgets(linea, sizeof(linea), archivo);
        sscanf(linea, "Descripcion: %[^\\n]", productos[*numProductos].descripcion);
        fgets(linea, sizeof(linea), archivo);
        sscanf(linea, "Precio: %f", &productos[*numProductos].precio);
        fgets(linea, sizeof(linea), archivo);
        sscanf(linea, "Peso: %f", &productos[*numProductos].peso);
        fgets(linea, sizeof(linea), archivo);
        sscanf(linea, "Calorias: %d", &productos[*numProductos].calorias);
        fgets(linea, sizeof(linea), archivo);
        sscanf(linea, "Cantidad: %d", &productos[*numProductos].cantidad);
        fgets(linea, sizeof(linea), archivo);

        (*numProductos)++;
    }

    fclose(archivo);
}
```

Manual de usuario:

1. Inicio del Programa:

- Iniciar las variables y estructuras de datos.
- Cargar usuarios y productos desde archivos si existen.

2. Menú Principal:

- Muestra el menú con varias opciones.
- Espera la entrada del usuario para seleccionar una opción.

3. Selección de Opción:

- Si la opción es:

1(Iniciar Sesión):

- Solicitar nombre de usuario y contraseña.
- Verificar la existencia del usuario y la validez de la contraseña.
- Iniciar sesión o crear un nuevo usuario según sea necesario.

2(Cambiar Contraseña):

- Solicitar contraseña actual y nueva contraseña.
- Verificar la contraseña actual.
- Cambiar la contraseña si la verificación es exitosa.

3(Agregar Producto):

- Solicitar información del nuevo producto.
- Agregar el producto a la lista.

4(Mostrar Contabilidad):

- Calcular la cantidad total de productos vendidos y el monto total obtenido.
- Mostrar los resultados.

5 (Mostrar Productos):

- Mostrar la información de todos los productos disponibles.

6(Realizar Venta):

- Mostrar la lista de productos disponibles.
- Solicitar la selección del producto y la cantidad a vender.
- Calcular el total a pagar.
- Solicitar el monto recibido y calcular el cambio.
- Actualizar la cantidad de productos después de la venta.

7(Guardar Productos):

- Guardar la información de los productos en un archivo.

8 (Salir):

- Guardar la información de usuarios y productos en archivos.
- Terminar el programa.

4. Fin del Programa:

- Finalizar la ejecución del programa.

Conclusiones:

El programa para la máquina expendedora de snacks presenta una solución eficiente y conveniente para satisfacer las necesidades alimenticias de personas en entornos diversos, como oficinas, escuelas o lugares públicos. Al abordar la ausencia de opciones de snacks en estos lugares, el programa se propone mejorar la experiencia de los usuarios y promover un ambiente más saludable. El objetivo del proyecto incluye proporcionar una solución inmediata para las necesidades alimenticias, mejorar la productividad en entornos laborales, fomentar hábitos alimenticios saludables y aumentar la satisfacción del usuario. Además, se busca reducir la necesidad de salir del entorno para buscar alimentos, ofreciendo variedad de opciones y apoyando la conciliación laboral.

Desde el punto de vista técnico, el programa permite la selección y compra de hasta 18 productos diferentes, cada uno con su descripción, peso y contenido calórico específico. La interfaz es sencilla, utilizando números para seleccionar productos y aceptando billetes de 50 y 100 pesos, así como monedas. El resumen técnico destaca la eficiencia del programa para agilizar los procesos de compra y pago. El manual de usuario proporciona instrucciones claras sobre el inicio del programa, la navegación a través del menú principal y la ejecución de diversas opciones, como iniciar sesión, cambiar contraseña, agregar productos, mostrar contabilidad, realizar ventas y guardar información. Este enfoque paso a paso facilita la comprensión y el uso efectivo del programa.

Para finalizar el programa para la máquina expendedora de snacks combina eficazmente funcionalidades técnicas con objetivos prácticos, abordando las necesidades alimenticias de manera rápida y eficiente en diversos entornos, lo que contribuye a la comodidad y satisfacción de los usuarios.