Programación I
Ingeniería en Informática

Trabajo Práctico de Laboratorio

Año 2024

- Bazán Cisneros, Lourdes - **MU:** 01514

Universidad Nacional de Catamarca
Facultad de Tecnología y Ciencias Aplicadas

Tabla de contenido

ENUNCIADO

Realizar un sistema de control de pasajes.

CONSIGNAS A RESOLVER

- Desarrollar el sistema haciendo uso del lenguaje de programación C.
- Implementar archivos y las estructuras que sean necesarias para llevar a cabo el sistema.

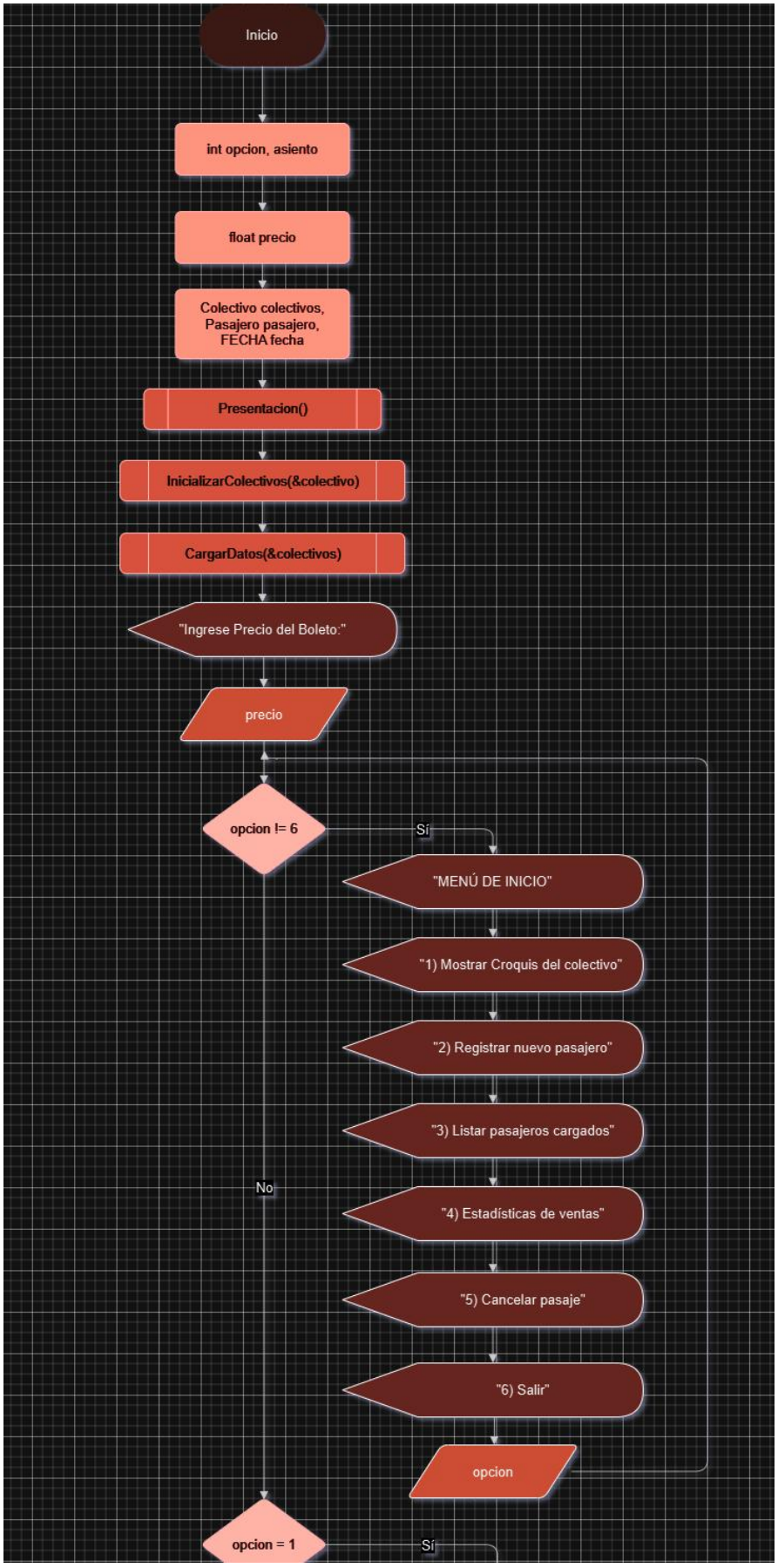
CASO DE ESTUDIO - RELEVAMIENTO DE INFORMACIÓN

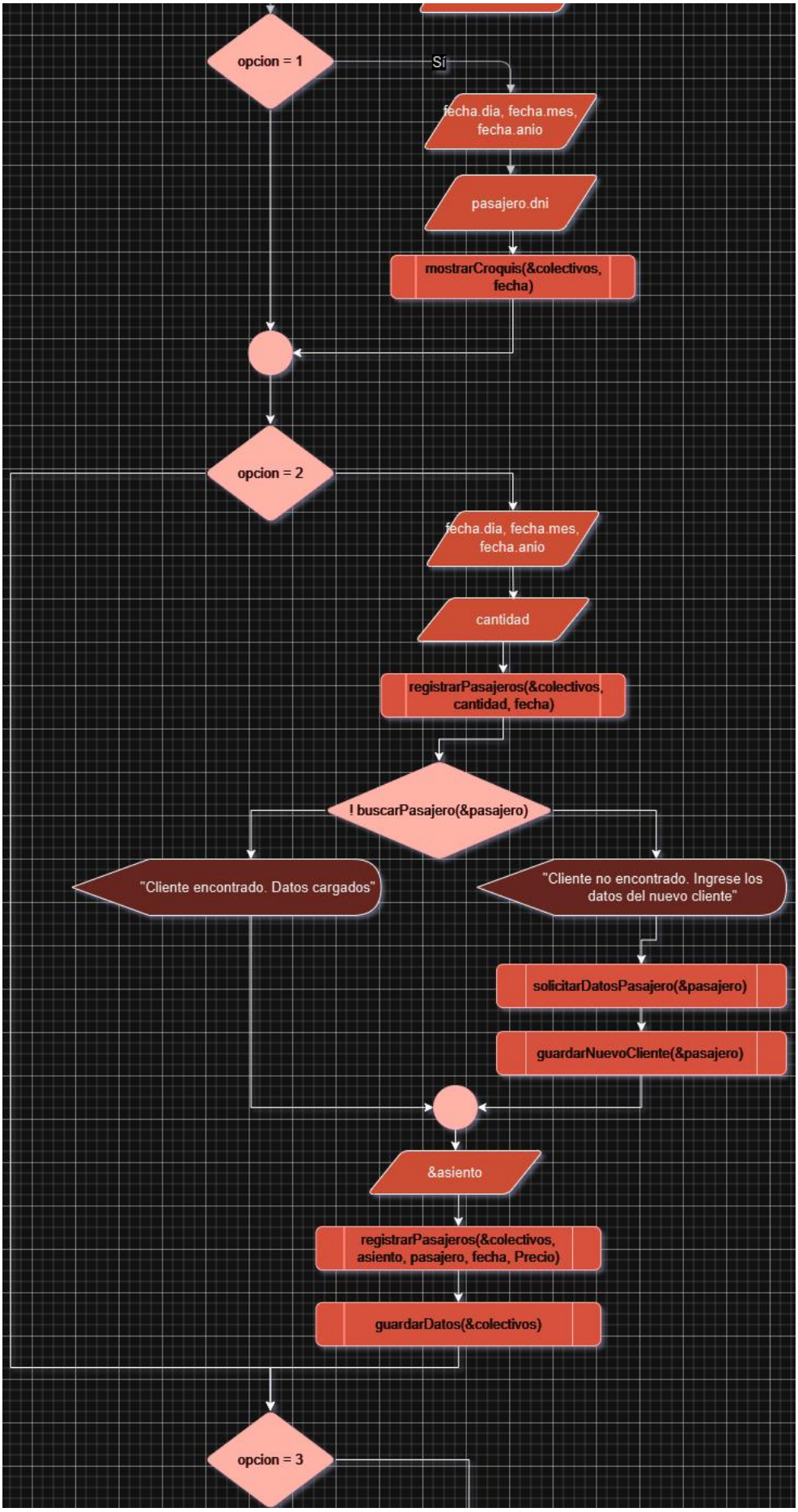
En una primera etapa de relevamiento se obtuvo la siguiente información:

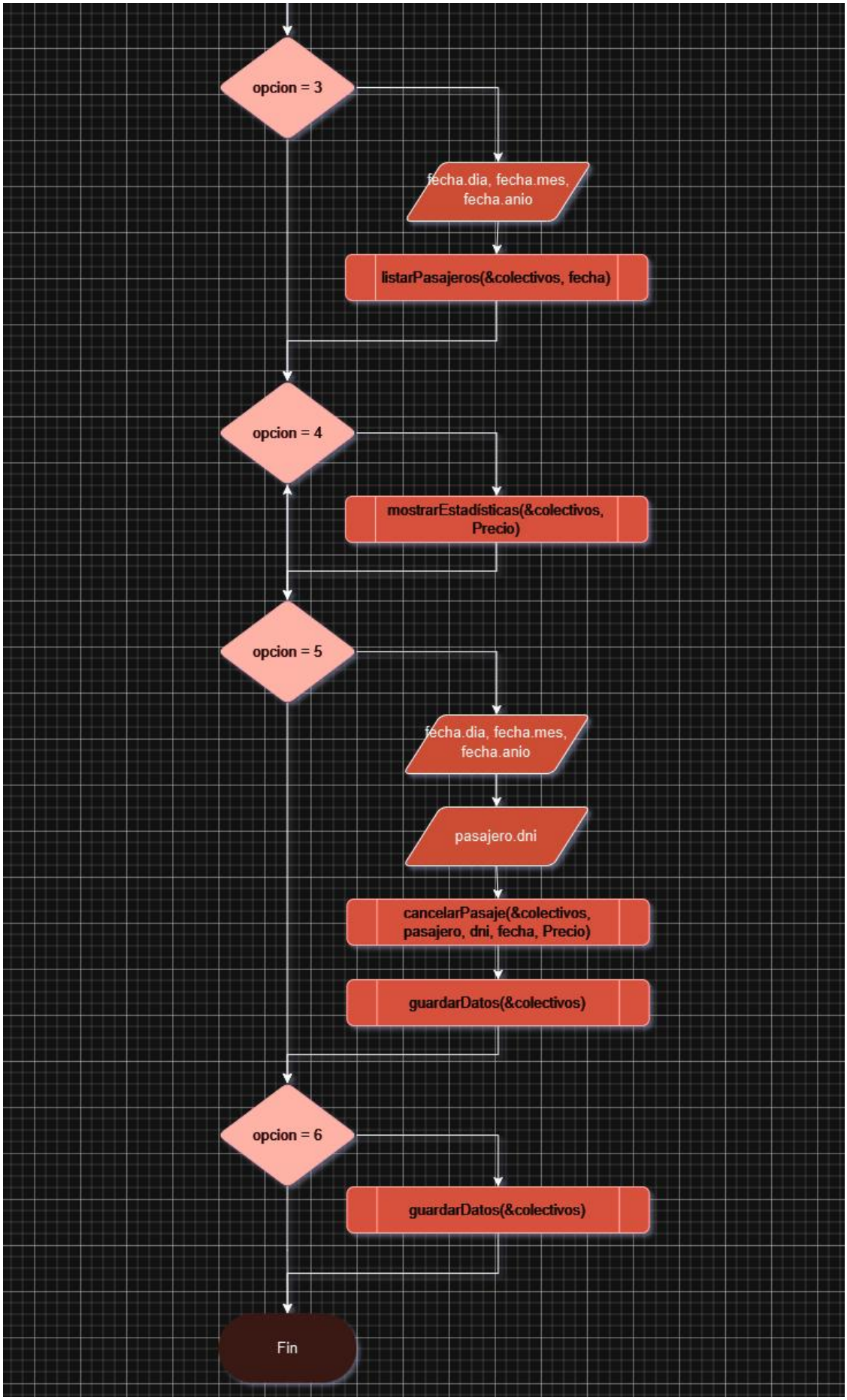
1) Se debe realizar un sistema de control de pasajes para una nueva empresa de transporte de pasajeros que llevará gente del centro de Catamarca al Aeropuerto Felipe Varela.
2) Se debe permitir elegir el lugar que ocuparán los pasajeros dentro del ómnibus.
3) Los asientos van del 1 al 20, siendo los impares ventanilla, y los pares pasillo.
4) La empresa solo cuenta con una unidad, el cual realiza 4 viajes por día, 2 de ida y 2 de vuelta.
5) Mostrar en pantalla un croquis del ómnibus para que el vendedor o el cliente pueda elegir su ubicación con mayor facilidad, donde se identifique los asientos que están ocupados y los que están disponibles.
6) Los vendedores tratarán de ubicar a los pasajeros que viajen solos juntos para evitar que queden asientos libres desparramados.
7) Se pueden vender pasajes con ilimitado tiempo de anticipación.
8) Para terminar el proceso de venta de pasaje se debe poder ingresar: Nombre, Dirección, Teléfono, DNI y Fecha de Nacimiento de cada pasajero.
9) Los mayores abonan el 100% del importe, los menores de 6 años el 50% y los menores de 2 años viajan gratis.
10) Los niños ocupan un asiento, en cambio los bebés deben ser cargados por sus padres sin ocupar un asiento.
11) Se permiten devoluciones con más de 48hs de anticipación. Para este caso el/los asientos volverán a estar disponibles.
12) Permitir hacer un listado del viaje que el operador precise, mostrando la siguiente información: Nro de asiento, DNI, Nombre, Teléfono y Edad.
13) Hacer estadísticas donde muestre la cantidad de asientos que se vendieron y los que no, con su respectivo porcentaje. Mostrar el porcentaje de mayores y menores.
14) Mostrar la recaudación por viaje y diaria según los solicite el operador.

Observaciones: Como los datos fueron obtenidos en una primera etapa, pueden estar desordenados, imprecisos, algunos pueden ser irrelevantes o también pueden faltar. Use su criterio cuando corresponda y de ser necesario continúe el relevamiento con los docentes de la cátedra.

DIAGRAMA DE FLUJO




opcion = 1
Sí
fecha.dia, fecha.mes, fecha.anio
pasajero.dni
mostrarCroquis(&colectivos, fecha)
opcion = 2
fecha.dia, fecha.mes, fecha.anio
cantidad
registrarPasajeros(&colectivos, cantidad, fecha)
! buscarPasajero(&pasajero)
"Cliente encontrado. Datos cargados"
"Cliente no encontrado. Ingrese los datos del nuevo cliente"
solicitarDatosPasajero(&pasajero)
guardarNuevoCliente(&pasajero)
&asiento
registrarPasajeros(&colectivos, asiento, pasajero, fecha, Precio)
guardarDatos(&colectivos)
opcion = 3


opcion = 3
fecha.dia, fecha.mes, fecha.anio
listarPasajeros(&colectivos, fecha)
opcion = 4
mostrarEstadísticas(&colectivos, Precio)
opcion = 5
fecha.dia, fecha.mes, fecha.anio
pasajero.dni
cancelarPasaje(&colectivos, pasajero, dni, fecha, Precio)
guardarDatos(&colectivos)
opcion = 6
guardarDatos(&colectivos)
Fin

PRUEBA DE TESTEO

- ***ESTRUCTURAS DE DATOS***

```
typedef struct {
   int dia;
   int mes;
   int anio;
} FECHA;

typedef struct {
   char nombre[50];
   char direccion[100];
   char telefono[15];
   char dni[10];
   char fechaNacimiento[11];
   int edad;
} Pasajero;

typedef struct {
   int numero;
   int ocupado;
   Pasajero pasajero;
} Asiento;

typedef struct {
   FECHA fecha;
   Asiento asientos[20];
   int asientosOcupados;
} RegistroViaje;

typedef struct {
   RegistroViaje viajes[365];
   int totalViajes;
   float recaudacion;
} Colectivo;

typedef struct {
   int mes;
   int anio;
   int asientosVendidos;
   float recaudacion;
} EstadisticaMensual;
```

- ***FUNCIONES DEL SISTEMA***

  - **PRESENTACIÓN DEL SISTEMA**

```
void presentacion() {
    printf("**********************************************************************
*******\n");
    printf("***\t\t\t\t\t\t\t\t\t  ***\n");
    printf("***\t\t\t\t\t\t\t\t\t  ***\n");
    printf("***\t\tBIENVENIDO A LA EMPRESA DE TRANSPORTES CATAMARCA\t  ***\n");
    printf("***\t\t\t\t\t\t\t\t\t  ***\n");
    printf("***\t\t\t\t\t\t\t\t\t  ***\n");
```

```
    printf("**********************************************************************
*******\n");
    system("PAUSE");
}
```

- **INICIALIZACIÓN DEL SISTEMA**

```
    void inicializarColectivos(Colectivo *colectivos) {
      colectivos->totalViajes = 0;
      colectivos->recaudacion = 0.0;
    }
```

- **MOSTRAR CROQUIS DEL COLECTIVO**

```
void mostrarCroquis(Colectivo *colectivos, FECHA fecha) {
  int idx = buscarRegistroViaje(colectivos, fecha);
  if (idx == -1) {
    printf("No hay registros para la fecha %d/%d/%d\n", fecha.dia, fecha.mes, fecha.anio);
    return;
  }
  RegistroViaje *registro = &colectivos->viajes[idx];
  printf("Croquis del Colectivo para la fecha %d/%d/%d:\n", fecha.dia, fecha.mes, fecha.anio);
  int i;
  for (i = 0; i < 20; i++) {
    if (registro->asientos[i].ocupado) {
      printf("[X] ");
    } else {
      printf("[%d] ", registro->asientos[i].numero);
    }
    if ((i + 1) % 4 == 0) printf("\n");
  }
}
```

- **REGISTRO DE PASAJEROS**

```
void registrarPasajero(Colectivo *colectivos, int asiento, Pasajero pasajero, FECHA fecha, float
Precio) {
  EstadisticaMensual estadisticasMensuales[12] = {0};
  calcularEstadisticasMensuales(colectivos, estadisticasMensuales,Precio);
  int idx = buscarRegistroViaje(colectivos, fecha);
  if (idx == -1) {
    idx = colectivos->totalViajes;
    inicializarRegistroViaje(&colectivos->viajes[idx], fecha);
    colectivos->totalViajes++;
  }
  RegistroViaje *registro = &colectivos->viajes[idx];
  if (asiento < 1 || asiento > 20 || registro->asientos[asiento - 1].ocupado) {
    printf("Asiento no disponible para la fecha %d/%d/%d.\n", fecha.dia, fecha.mes, fecha.anio);
    return;
  }
  registro->asientos[asiento - 1].pasajero = pasajero;
  registro->asientos[asiento - 1].ocupado = 1;
  registro->asientosOcupados++;
  colectivos->recaudacion += calcularPrecio(pasajero.edad, Precio);
  printf("Pasajero registrado exitosamente en el asiento %d para la fecha %d/%d/%d.\n",
asiento, fecha.dia, fecha.mes, fecha.anio);

  FILE *archivo = fopen("estadisticas_mensuales.bin", "wb");
```

```
    if (archivo == NULL) {
        printf("Error al abrir el archivo para guardar las estad%csticas mensuales.\n",161);
        return;
    }
    fwrite(estadisticasMensuales, sizeof(EstadisticaMensual), 1, archivo);
    fclose(archivo);
    printf("Estad%csticas mensuales guardadas exitosamente.\n",161);
}
```

- **CÁLCULO DEL PRECIO (YA SEA ADULTO, MENOR O BEBÉ)**

```
float calcularPrecio(int edad, float Precio) {
    if (edad < 2) return Precio = 0;
    if (edad < 6) return Precio/2;  // Precio reducido
    return Precio;  // Precio completo
}
```

- **LISTAR PASAJEROS**

```
void listarPasajeros(Colectivo *colectivos, FECHA fecha) {
    int idx = buscarRegistroViaje(colectivos, fecha);
    if (idx == -1) {
        printf("No hay registros para la fecha %d/%d/%d\n", fecha.dia, fecha.mes, fecha.anio);
        return;
    }
    RegistroViaje *registro = &colectivos->viajes[idx];
    printf("Lista de Pasajeros para la fecha %d/%d/%d:\n", fecha.dia, fecha.mes, fecha.anio);
    int i;
    for (i = 0; i < 20; i++) {
        if (registro->asientos[i].ocupado) {
            printf("\t\t\tAsiento %d:\n%s,\tDNI: %s,\tTelefono: %s,\tEdad: %d\n",
                registro->asientos[i].numero,
                registro->asientos[i].pasajero.nombre,
                registro->asientos[i].pasajero.dni,
                registro->asientos[i].pasajero.telefono,
                registro->asientos[i].pasajero.edad);
        }
    }
}
```

- **MOSTRAR ESTADÍSTICAS**

```
void mostrarEstadisticas(Colectivo *colectivos,float Precio) {
    EstadisticaMensual estadisticasMensuales[12] = {0};
    calcularEstadisticasMensuales(colectivos, estadisticasMensuales, Precio);

    printf("Estad%csticas del Colectivo:\n",161);
    printf("-------------------------\n");
    printf("Total de Viajes: %d\n", colectivos->totalViajes);
    printf("Recaudaci%cn Total: %.2f\n",162, colectivos->recaudacion);
    printf("-------------------------\n");
    printf("Estad%csticas Mensuales:\n",161);
    int i;

    for(i = 0; i <= 11; i++) {
        if (estadisticasMensuales[i].asientosVendidos > 0) {
            printf("Mes: %d/%d\n", estadisticasMensuales[i].mes, estadisticasMensuales[i].anio);
```

```
            printf("  Asientos Vendidos: %d\n", estadisticasMensuales[i].asientosVendidos);
            printf("  Recaudaci%cn: %.2f\n",162, estadisticasMensuales[i].recaudacion);
        }
    }
}
```

- **SOLICITUD DE DATOS A PASAJEROS**

```
void solicitarDatosPasajero(Pasajero *pasajero) {
    printf("Ingrese nombre: ");
    scanf(" %[^\n]", pasajero->nombre);
    printf("Ingrese direcci%cn: ", 162);
    scanf(" %[^\n]", pasajero->direccion);
    printf("Ingrese telefono: ");
    scanf(" %[^\n]", pasajero->telefono);
    printf("Ingrese fecha de nacimiento (dd/mm/yyyy): ");
    scanf(" %[^\n]", pasajero->fechaNacimiento);
    printf("Ingrese edad: ");
    scanf("%d", &pasajero->edad);
}
```

- **GUARDAR DATOS EN ARCHIVO**

```
void guardarDatos(Colectivo *colectivos) {
    FILE *archivo = fopen("datos_viajes.dat", "wb");
    if (archivo == NULL) {
        printf("No se pudo abrir el archivo para guardar.\n");
        return;
    }
    fwrite(colectivos, sizeof(Colectivo), 1, archivo);
    fclose(archivo);
    printf("Datos de viajes guardados exitosamente.\n");
}
```

- **CARGAR DATOS EN ARCHIVO**

```
void cargarDatos(Colectivo *colectivos) {
    FILE *archivo = fopen("datos_viajes.dat", "rb");
    if (archivo == NULL) {
        printf("No se pudo abrir el archivo para cargar.\n");
        return;
    }
    fread(colectivos, sizeof(Colectivo), 1, archivo);
    fclose(archivo);
    printf("Datos de viajes cargados exitosamente.\n");
}
```

- **BUSCAR PASAJEROS EN ARCHIVO**

```
int buscarPasajero(Pasajero *pasajero) {
    FILE *archivo = fopen("clientes.dat", "rb");
    if (archivo == NULL) {
        printf("No se pudo abrir el archivo de clientes.\n");
        return 0;
    }
    Pasajero temp;
```

```
    while (fread(&temp, sizeof(Pasajero), 1, archivo)) {
        if (strcmp(temp.dni, pasajero->dni) == 0) {
            *pasajero = temp;
            fclose(archivo);
            return 1; // Cliente encontrado
        }
    }
    fclose(archivo);
    return 0; // Cliente no encontrado
}
```

- **GUARDAR NUEVO CLIENTE EN ARCHIVO**

```
void guardarNuevoCliente(Pasajero *pasajero) {
    FILE *archivo = fopen("clientes.dat", "ab");
    if (archivo == NULL) {
        printf("No se pudo abrir el archivo para guardar el cliente.\n");
        return;
    }
    fwrite(pasajero, sizeof(Pasajero), 1, archivo);
    fclose(archivo);
    printf("Nuevo cliente guardado exitosamente.\n");
}
```

- **BUSCAR REGISTRO DE VIAJE**

```
int buscarRegistroViaje(Colectivo *colectivos, FECHA fecha) {
    int i;
    for (i = 0; i < colectivos->totalViajes; i++) {
        if (colectivos->viajes[i].fecha.dia == fecha.dia &&
            colectivos->viajes[i].fecha.mes == fecha.mes &&
            colectivos->viajes[i].fecha.anio == fecha.anio) {
            return i;
        }
    }
    return -1;
}
```

- **INICIALIZACIÓN REGISTRO DE VIAJE**

```
void inicializarRegistroViaje(RegistroViaje *registro, FECHA fecha) {
    registro->fecha = fecha;
    registro->asientosOcupados = 0;
    int i;
    for (i = 0; i < 20; i++) {
        registro->asientos[i].numero = i + 1;
        registro->asientos[i].ocupado = 0;
    }
}
```

- **CÁLCULO DE ESTADÍSTICAS MENSUALES**

```
void calcularEstadisticasMensuales(Colectivo *colectivos, EstadisticaMensual
estadisticasMensuales[],float Precio) {
    int i;
    for (i = 0; i < colectivos->totalViajes; i++) {
```

```
      RegistroViaje *registro = &colectivos->viajes[i];
      int mes = registro->fecha.mes - 1;
      int anio = registro->fecha.anio;

      estadisticasMensuales[mes].mes = registro->fecha.mes;
      estadisticasMensuales[mes].anio = anio;
      estadisticasMensuales[mes].asientosVendidos += registro->asientosOcupados;
      int j;
      for (j = 0; j < 20; j++) {
        if (registro->asientos[j].ocupado) {
          estadisticasMensuales[mes].recaudacion += calcularPrecio(registro-
>asientos[j].pasajero.edad,Precio);
        }
      }
    }
}
```

- **CANCELACIÓN DE VIAJE**

```
void cancelarPasaje(Colectivo *colectivos, char *dni, FECHA fecha,float Precio){
  EstadisticaMensual estadisticasMensuales[12] = {0};
  calcularEstadisticasMensuales(colectivos, estadisticasMensuales,Precio);
  int idx = buscarRegistroViaje(colectivos, fecha);
  if(idx == -1 ){
    printf("No hay registros para la fecha %d/%d/%d",fecha.dia,fecha.mes,fecha.anio);
    return;
  }
  RegistroViaje *registro = &colectivos->viajes[idx];
  int i;
  for(i = 0; i<20 ; i++){
    if(registro->asientos[i].ocupado && strcmp(registro->asientos[i].pasajero.dni, dni) == 0 ){
      registro->asientos[i].ocupado = 0;
      registro->asientosOcupados--;
      printf("Pasaje cancelado para el Dni %s para la fecha %d/%d/%d\n",dni,
fecha.dia,fecha.mes,fecha.anio);
      colectivos->recaudacion -= calcularPrecio(registro->asientos[i].pasajero.edad,Precio);
      colectivos->totalViajes--;
      return;

    }
  }
  printf("No se encontro el pasajero con dni %s para la fecha %d/%d/%d",dni,
fecha.dia,fecha.mes,fecha.anio);


  FILE *archivo = fopen("estadisticas_mensuales.bin", "wb");
  if (archivo == NULL) {
    printf("Error al abrir el archivo para guardar las estad%csticas mensuales.\n",161);
    return;
  }
  fwrite(estadisticasMensuales, sizeof(EstadisticaMensual), 1, archivo);
  fclose(archivo);
  printf("Estad%csticas mensuales guardadas exitosamente.\n",161);
}
```

MANUAL DE USUARIO

- ***INTRODUCCIÓN***

  - **PROPÓSITO DEL SISTEMA**

    El sistema de control de pasajes permite a una empresa de transporte gestionar la venta de pasajes, asignar asientos a los pasajeros, y mantener registros de las ventas y estadísticas de ocupación.

  - **ALCANCE DEL SISTEMA**

    El sistema está diseñado para gestionar los pasajes de una única unidad de transporte que realiza 4 viajes diarios, desde el centro de Catamarca hasta el aeropuerto Felipe Varela, y viceversa.

- ***REQUISITOS DEL SISTEMA***

  - **HARDWARE**

    Computadora con capacidad de ejecutar programas en lenguaje C.

  - **SOFTWARE**

    Sistema operativo compatible con compiladores de C (Windows, MacOS, Linux, entre otros)

- ***INSTALACIÓN DEL SISTEMA***

PASO N°1: Descargar el código fuente.
PASO N°2: Compilar el código fuente.
PASO N°3: Ejecutar el código fuente.

- ***USO DEL SISTEMA***

  - **INICIO DEL PROGRAMA**

    Al ejecutar el programa, muestra un mensaje de presentación al sistema, seguido de inicializar el ómnibus y cargar los datos previamente guardar, en el caso que existan.
    El siguiente paso es seleccionar alguna opción del menú principal.

  - **OPCIONES DEL MENÚ**

    - **MOSTRAR CROQUIS DEL COLECTIVO**
    - **REGISTRAR UN NUEVO PASAJERO**
    - **LISTAR PASAJEROS**
    - **MOSTRAR ESTADÍSTICAS DE VENTAS**
    - **CANCELAR PASAJE**
    - **SALIR DEL SISTEMA**