

Universidad San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ciencias y sistemas
Estructuras de Datos

Ingenieros:

- Ing. Edgar Ornelis
- Ing. Álvaro Hernández
- Ing. Luis Espino

Auxiliares:

- Steven Mejía
- Luis Garcia
- Marcos Itzep



AutoGest Pro

Fase 1

Objetivos

Objetivo General

Desarrollar un sistema de gestión para un taller de reparación de vehículos que permita administrar de manera eficiente los vehículos, los repuestos, los servicios y el flujo de facturación, aplicando estructuras de datos como listas, pilas, colas y matrices dispersas en el lenguaje de programación C#.

Objetivos Específicos

1. Utilizar punteros con unsafe code para gestionar el registro de vehículos en el taller.
2. Utilizar Graphviz para generar reportes visuales de las estructuras de datos implementadas (lista de vehículos, pila de órdenes, cola de atención y matriz de repuestos).

Descripción

AutoGest Pro es un sistema integral de gestión diseñado específicamente para talleres de reparación de vehículos. Este software simplifica y optimiza todas las operaciones diarias del taller, desde el registro de vehículos hasta la gestión de servicios y el control de repuestos.

El sistema permite registrar cada vehículo que ingresa al taller, junto con los datos del propietario y las características del automóvil. Además, facilita la creación y seguimiento de servicios, asegurando que cada reparación se realice en el orden correcto y con los repuestos adecuados. Con su interfaz intuitiva y amigable, el sistema es fácil de usar y se adapta a las necesidades de talleres de cualquier tamaño.

Entidades

Usuarios

Son los clientes del taller y propietarios de los vehículos. En el sistema, serán registrados como usuarios y almacenados en una lista simplemente enlazada. Sus atributos serán los siguientes:

- ID
- Nombres
- Apellidos
- Correo
- Contraseña

Vehículos

Representan los automóviles registrados en el taller para su mantenimiento o reparación. Se almacenarán en una lista doblemente enlazada. Sus atributos serán los siguientes:

- ID
- ID_Usuario
- Marca
- Modelo
- Placa

Repuestos

Son las piezas o componentes disponibles en el taller para la reparación de vehículos. Se almacenarán en una lista circular. Sus atributos serán los siguientes:

- ID
- Repuesto
- Detalles
- Costo

Servicios

Son las tareas de mantenimiento o reparación realizadas en el taller. Se administrarán en una cola, garantizando un orden de atención. Sus atributos serán los siguientes:

- ID
- Id_Repuesto
- Id_Vehiculo
- Detalles
- Costo

Facturas

Son los comprobantes generados por los servicios prestados a los clientes. Se almacenarán en una pila. Sus atributos serán los siguientes:

- ID
- ID_Orden
- Total

Bitácora

Registra la relación entre los vehículos y los repuestos utilizados en los servicios. Se administrará mediante una matriz dispersa y se actualizará automáticamente con cada servicio generado.

Interfaces del Sistema

Inicio de Sesión

Se tendrá una ventana de inicio de sesión en el cual los usuarios registrados podrán acceder a sus respectivas cuentas, cabe destacar que se tendrá un usuario root y sus credenciales serán "root@gmail.com" y con contraseña "root123".

Funcionalidades del Usuario Root

Carga Masiva

El usuario root podrá realizar carga masiva de las siguientes entidades "Usuarios", "Vehículos" y "Repuestos". El programa debe ser claro a qué entidad se realizará la carga masiva.



El archivo que contendrá los datos estará en formato JSON. Y la estructura de estos será la siguiente:

```
## Usuario

[
  {
    "ID": 1,
    "Nombres": "Juan",
    "Apellidos": "Pérez",
    "Correo": "juan.perez@mail.com",
    "Contrasenia": "123456"
  },

```

```
{
  "ID": 2,
  "Nombres": "María",
  "Apellidos": "Gómez",
  "Correo": "maria.gomez@mail.com",
  "Contraseña": "password123"
}
```

Vehículos

```
[
  {
    "ID": 1,
    "ID_Usuario": 1,
    "Marca": "Toyota",
    "Modelo": "Corolla",
    "Placa": "ABC123"
  },
  {
    "ID": 2,
    "ID_Usuario": 2,
    "Marca": "Ford",
    "Modelo": "Focus",
    "Placa": "XYZ456"
  }
]
```

Repuestos

```
[
  {
    "ID": 1,
    "Repuesto": "Filtro de aceite",
    "Detalles": "Filtro de aceite para motor 1.8L",
    "Costo": 15.75
  },
  {
    "ID": 2,
    "Repuesto": "Bujías",
    "Detalles": "Juego de bujías para motor 2.0L",
    "Costo": 30.50
  }
]
```

Ingreso Manual

El usuario root podrá realizar el ingreso de forma manual de las siguientes entidades “Usuarios”, “Vehículos”, “Repuestos” y “Servicios”.

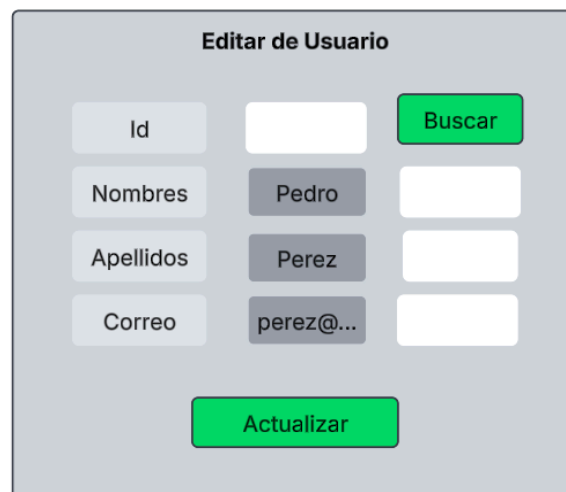


Formulario de Ingreso de Usuario. El formulario tiene un título "Ingreso de Usuario" en la parte superior. Debajo del título, hay cinco campos de entrada de texto, cada uno con un label a su izquierda: "Id", "Nombres", "Apellidos", "Correo" y "Contraseña". Los campos "Id", "Nombres", "Apellidos" y "Correo" son blancos, mientras que el campo "Contraseña" es gris. Debajo de los campos, hay un botón verde con el texto "Guardar".

Gestión de Usuarios

Podrá también ver, eliminar y editar la información de los usuarios que están registrados en el sistema. El programa debe validar que el usuario exista y en caso de no mostrar un error.

- **Ver usuario:** Mostrará la información del usuario incluyendo sus vehículos.
- **Editar Usuario:** Por medio del Id se podrá modificar Nombres, Apellidos y el Correo.
- **Eliminar Usuario:** Por medio del Id se podrá eliminar al usuario.



Formulario de Editar de Usuario. El formulario tiene un título "Editar de Usuario" en la parte superior. Debajo del título, hay un campo de entrada de texto con el label "Id" y un botón verde "Buscar" a su derecha. Debajo de esto, hay tres filas de campos de entrada de texto, cada una con un label a su izquierda: "Nombres", "Apellidos" y "Correo". Los campos "Nombres" y "Apellidos" tienen valores predefinidos "Pedro" y "Perez" respectivamente, y el campo "Correo" tiene el valor "perez@...". Debajo de los campos, hay un botón verde con el texto "Actualizar".

Generar Servicio

El usuario root podrá generar un servicio y sucesivamente generar una factura para el usuario, solicitando los siguientes datos, si no existe ya sea el repuesto o el vehículo debe mostrar un error:

Crear Servicio

ID	<input type="text"/>
Id_Repuesto	<input type="text"/>
Id_Vehiculo	<input type="text"/>
Detalles	<input type="text"/>
Costo	<input type="text"/>

Guardar

Generación de Factura

Cuando se genere un servicio el programa automáticamente realizará las siguientes acciones:

- Creará una factura con la siguiente información, este se almacenará en la pila y los datos de este serán:
 - ID
 - ID_Orden
 - Total (Costo del servicio + Costo del repuesto)
- Se insertará el detalle de la orden en la matriz dispersa.

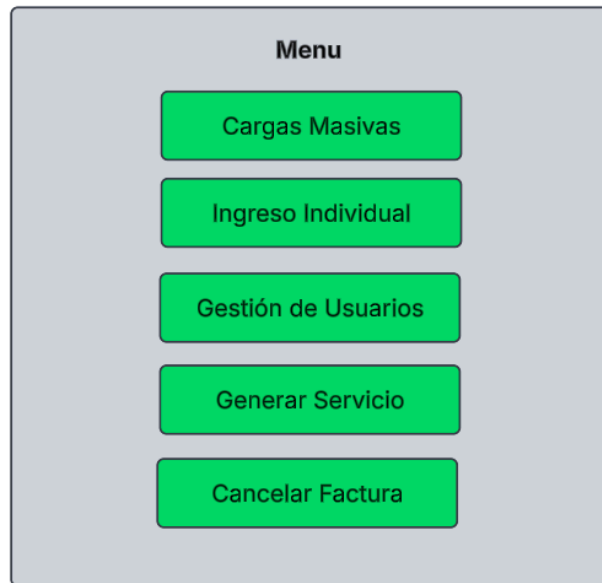
Cancelar Factura

Esta opción le informará al sistema que el usuario ya pagó por el servicio realizado y tendrá que obtenerlo de la pila. Debe mostrar los datos de la factura.

Facturacion

Id	1
Id_Orden	1
Total	200

Interfaz Sugerida

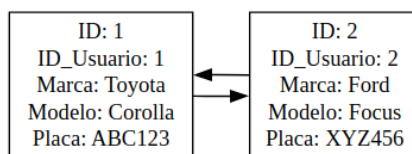


Reportes

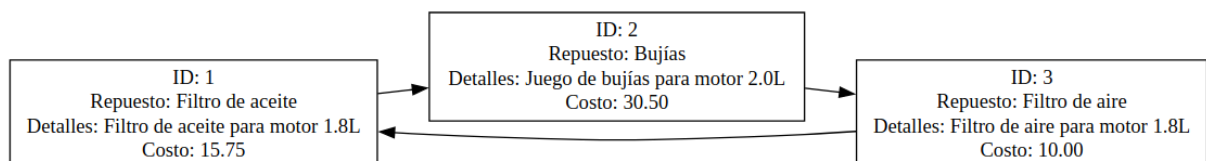
- Usuarios**



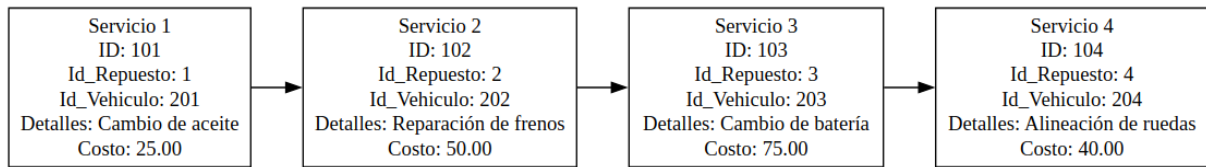
- Vehículos**



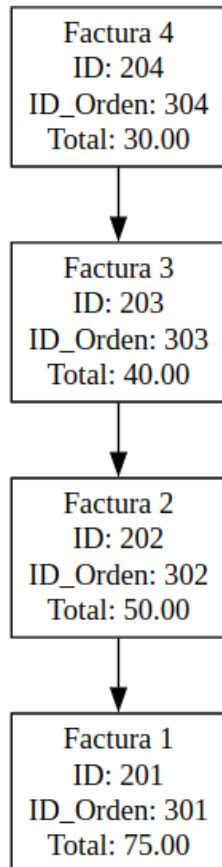
- Repuestos**



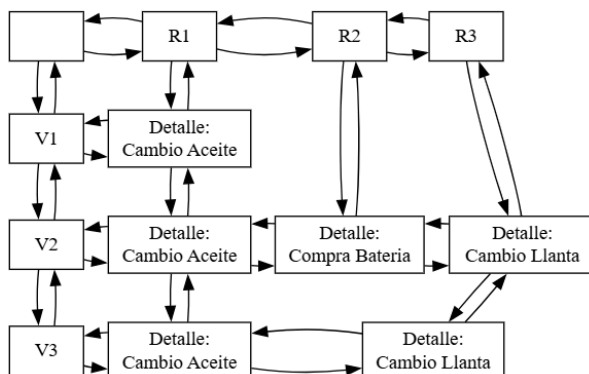
- **Servicios**



- **Facturación**



- **Bitácora**



- **Top 5 vehículos con más servicios**
- **Top 5 vehículos más antiguos**

Observaciones

- Lenguaje de programación a utilizar: **C#**
- Las interfaces presentadas en el enunciado solo son guías para la comprensión del estudiante.
- Para la creación de interfaces debe utilizar la librería GTK.
- El nombre del usuario administrador será **root@gmail.com** y su contraseña será **root123**.
- Sistema Operativo: Linux (distribución libre).
- IDE: Libre.
- Para esta fase solo se deberá utilizar punteros con unsafe code.
- Herramienta para desarrollo de reportes gráficos: **Graphviz**.
- Durante la calificación se harán preguntas para validar que el estudiante realizó el proyecto, de no responder correctamente se anulará la nota obtenida en la o las secciones en la que se aplique tal concepto.
- Cada estudiante deberá crear un repositorio de github con el nombre [EDD]1S2025_carnet, agrupando cada fase con el nombre de **Fase#** dentro del mismo repositorio.
- Apartado de entrega en la plataforma UEDI: Fecha y hora de entrega: **28/02/2025 23:59 PM**
- Las copias serán penalizadas con una nota de 0 y castigadas según lo indique el reglamento.
- En la calificación cada estudiante deberá llevar su laptop para presentar el proyecto.
- Se debe agregar al auxiliar al repositorio como colaborador. Usuario de github:
Sección A: Facundo-25
Sección B: Lu1s16
Sección C: Marltzep

Entregables

- Link a repositorio
 - Código fuente
 - Manual de Usuario en **Markdown**
 - Manual Técnico en **Markdown**