

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



Primer Semestre 2025

Objetivos

Objetivo General

Desarrollar un sistema de gestión para un taller de reparación de vehículos que permita administrar de manera eficiente sus entidades. Se utilizarán estructuras de datos avanzadas para optimizar la organización y consulta de información en el sistema.

Objetivos Específicos

1. Utilizar estructura de datos como lista simple y lista doblemente enlazada para almacenar información de manera dinámica.
2. Desarrollar una interfaz gráfica amigable y funcional utilizando la librería GTK, permitiendo a los usuarios interactuar de manera intuitiva con el sistema.
3. Optimizar la gestión de repuestos mediante la implementación de un árbol AVL, asegurando búsquedas rápidas y eficientes.
4. Organizar y administrar los servicios de reparación utilizando un árbol binario, permitiendo diferentes tipos de recorridos para consultar información según las necesidades del usuario.
5. Estructurar la facturación del sistema mediante un árbol B de orden 5, garantizando una inserción y búsqueda eficiente de facturas.
6. Generar reportes visuales con Graphviz para representar gráficamente las estructuras de datos implementadas.

Estructuras a Utilizar

- **Lista simplemente enlazada:** Eficiente en inserciones y eliminaciones dinámicas sin requerir memoria contigua.
- **Lista doblemente enlazada:** Permite recorridos en ambas direcciones, facilitando accesos flexibles.
- **Árbol AVL:** Garantiza búsquedas, inserciones y eliminaciones rápidas gracias a su equilibrio automático.
- **Árbol Binario:** Organiza datos jerárquicamente y permite distintos tipos de recorridos eficientes.
- **Árbol B:** Optimiza la búsqueda y almacenamiento en estructuras grandes, ideal para bases de datos.

Descripción

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

Para mejorar el rendimiento y escalabilidad del sistema, en esta fase se han incorporado nuevas estructuras de datos que permiten una administración más eficiente de la información y para mejorar la experiencia del usuario, la interfaz gráfica se utilizará GTK, proporcionando un entorno visual intuitivo y adaptable a las necesidades de talleres de cualquier tamaño.

Gracias a estas mejoras, AutoGest Pro no solo simplifica la gestión operativa del taller, sino que también garantiza un rendimiento óptimo y una mayor escalabilidad del sistema.

Entidades

Usuarios

Son los clientes del taller y propietarios de los vehículos. Se almacenarán en una lista simplemente enlazada para facilitar la inserción y eliminación dinámica de registros. Sus atributos serán los siguientes:

- Id
- Nombres
- Apellidos
- Correo
- Edad
- Contrasenia

Validaciones: El ID y Correo deben ser únicos en el sistema para evitar duplicados.

Vehículos

Representan los automóviles registrados en el taller para su mantenimiento o reparación. Se almacenarán en una lista doblemente enlazada para permitir recorridos eficientes en ambos sentidos y facilitar la gestión de vehículos. Sus atributos serán los siguientes:

- Id
- Id_Usuario
- Marca
- Modelo
- Placa

Validaciones: El ID del vehículo debe ser único y se debe verificar que el usuario al que pertenece el vehículo exista en el sistema antes de registrarlo.

Repuestos

Son las piezas o componentes disponibles en el taller para la reparación de vehículos. Se almacenarán en un árbol AVL para garantizar que las búsquedas, inserciones y eliminaciones sean rápidas y eficientes. Sus atributos serán los siguientes:

- Id
- Repuesto
- Detalles
- Costo

Validaciones: El ID del repuesto debe ser único en el sistema.

Servicios

Son las tareas de mantenimiento o reparación realizadas en el taller. Se administrarán en un árbol binario, permitiendo ordenar y recorrer los servicios según diferentes criterios. Sus atributos serán los siguientes:

- Id
- Id_Repuesto
- Id_Vehiculo
- Detalles
- Costo

Validaciones: El ID del servicio debe ser único y se debe verificar que los ID de repuesto y vehículo existan en el sistema antes de registrar un servicio; de lo contrario, se mostrará un mensaje de error.

Facturas

Son los comprobantes generados por los servicios prestados a los clientes. Se almacenarán en un árbol B de orden 5, optimizando el acceso y gestión de grandes volúmenes de facturación.

- Id
- Id_Servicio
- Total

Validaciones: El ID de la factura debe ser único

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 admin y sus credenciales serán “admin@usac.com” y con contraseña “admin123”.

The wireframe shows a light gray rectangular form titled "Inicio de Sesión". Inside, there are two input fields: one labeled "Correo" and another for "Contraseña", both with placeholder text. Below them is a large green rectangular button labeled "Validar".

Rol Administrador

Carga Masiva

El usuario administrador podrá realizar carga masiva de las siguientes entidades “USUARIOS”, “VEHICULOS” y “REPUESTOS”. El programa debe ser claro a qué entidad se realizará la carga masiva.

The wireframe shows a light gray rectangular form titled "Menu Carga masiva". It features a dropdown menu with "Vehiculos" selected. Below it is a green rectangular button labeled "Cargar".

Sugerencia 1

The wireframe shows a light gray rectangular form titled "Carga Masiva Usuario". It lists three entities: "Usuario", "Vehiculos", and "Repuestos", each next to a green rectangular button labeled "Cargar".

Sugerencia 2

Cada entidad se apoyará por medio de un archivo JSON para la carga masiva, estos vendrán en diferentes archivos.

Usuario

```
[
  {
    "ID": 1,
    "Nombres": "Carlos Alberto",
    "Apellidos": "Gomez Martinez",
    "Correo": "carlos.alberto@usac.com",
    "Edad": 20,
    "Contrasenia": "CarlosMartinez"
  },
  {
    "ID": 2,
    "Nombres": "Ana Lucia",
    "Apellidos": "Martinez Lopez",
    "Correo": "ana.lucia@usac.com",
    "Edad": 20,
    "Contrasenia": "AnaLopez"
  }
]
```

Vehículos

```
[
  {
    "ID": 1,
    "ID_Usuario": 1,
    "Marca": "Honda Civic",
    "Modelo": 2005,
    "Placa": "XZJ7H9K"
  },
  {
    "ID": 2,
    "ID_Usuario": 2,
    "Marca": "Toyota Corolla",
    "Modelo": 2010,
    "Placa": "JKS8T5A"
  }
]
```

Repuestos



```
[  
  {  
    "ID": 1,  
    "Repuesto": "Filtro de aceite",  
    "Detalles": "Filtra impurezas del aceite del motor.",  
    "Costo": 25.50  
  },  
  {  
    "ID": 2,  
    "Repuesto": "Bujías",  
    "Detalles": "Se usa para la mezcla de aire y combustible.",  
    "Costo": 15.75  
  }  
]
```

Gestión de Entidades

El usuario administrador podrá realizar las siguientes acciones VISUALIZAR y ELIMINAR para las entidades USUARIOS, VEHICULOS.

Editar de Usuario

Id	1
Id_Usuario	1
Marca	Honda Civic
Modelo	2008
Placa	PZDS562
<button>Buscar</button>	<button>Eliminar</button>

Sugerencia 1



Actualización de Repuestos

El usuario administrador también tiene la tarea de actualizar la información de los repuestos cuando sea necesario.

Actualizacion de Repuestos

Id		Buscar
Repuesto	Llantas	
Detalles	Michelin	
Costo	200	

Actualizar

Sugerencia 1

Visualización de Repuestos

El administrador también podrá visualizar los repuestos en los distintos órdenes PRE-ORDEN, IN-ORDEN, POST-ORDEN.

Visualización de Repuestos

Post-Orden	▼		
Id	Repuesto	Detalles	Costo

Sugerencia 1



Generar Servicio

El usuario administrador 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	
ID_Repuesto	
Id_Vehiculo	
Detalles	
Costo	

Guardar

Generación de Factura

Cuando se genere un servicio el programa generará una factura automáticamente con los atributos mencionados anteriormente (sección de entidades).

Control de Logueo

El administrador podrá generar el log de entradas y salidas de los usuarios en el sistema y será exportado en formato JSON con los siguiente atributos: usuario, entrada y salida.

```
[
  {
    "usuario": "juanperez@usac.com",
    "entrada": "2025-02-2 12:00:00.00",
    "salida": "2025-02-27 12:30:00.00"
  },
  {
    "usuario": "andragomez@usac.com",
    "entrada": "2025-02-27 13:00:00.00",
    "salida": "2025-02-27 13:45:00.00"
  }
]
```



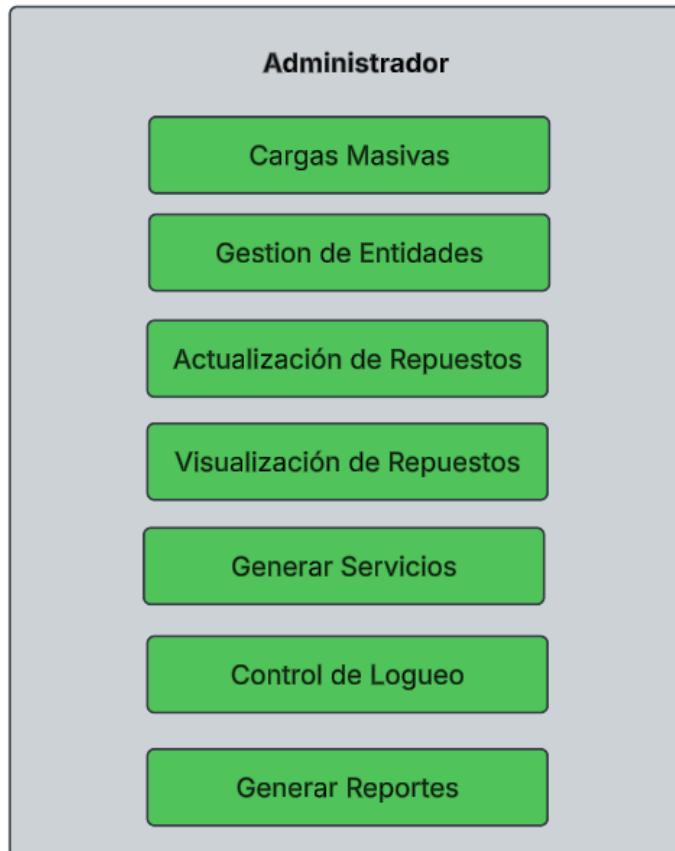
Generación de Reportes

Esta opción permitirá crear imágenes de los reportes que se explicaran más detalladamente en el apartado REPORTES utilizando Graphviz, estas imágenes deben estar en una carpeta llamada /Reportes. La ubicación de esta carpeta no tiene importancia pero destacando que todos los reportes que se generen deben estar dentro de esta.

```
.  
└── Fase2  
    ├── Fase2.csproj  
    ├── obj  
    ├── Program.cs  
    └── Reportes  
        └── source
```

Interfaz sugerida para el administrador

Se le presenta a continuación un menú que podría ayudarle como base para el diseño de su interfaz del menú del administrador.





Rol Usuario

Insertar Vehículos

El usuario será capaz de registrar sus propios vehículos en el sistema.

Registrar Vehiculo

Id	<input type="text"/>
Marca	<input type="text"/>
Modelo	<input type="text"/>
Placa	<input type="text"/>

Visualización de Servicios

Este podrá visualizar los servicios que ha realizado a sus vehículos, utilizando distintos tipos de filtrado los cuales serán PRE-ORDEN, POST-ORDEN y IN-ORDEN.

Visualización de Servicios

Post-Orden ▼

Id	Repuesto	Vehículo	Detalles	Costo



Visualización de Facturas

El usuario por medio de una tabla podrá visualizar las facturas que tiene pendientes a pagar.

Visualización de Facturas		
Id	Orden	Total

Sugerencia 1

Cancelar Facturas

Por medio del id de la factura el usuario podrá indicar que factura desea pagar. Una vez la factura sea pagada, este debe ser eliminada del sistema.

Pagar Factura		
Id		Buscar
Orden	1	
Total	100	
Pagar		

Sugerencia 1



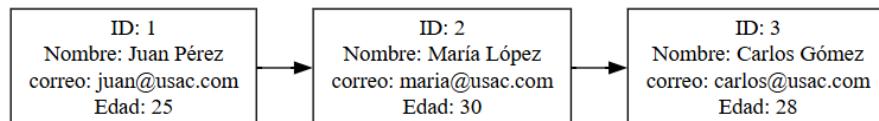
Interfaz Sugerida para el Usuario

Se le presenta a continuación un menú que podría ayudarle como base para el diseño de su interfaz del menú del administrador.

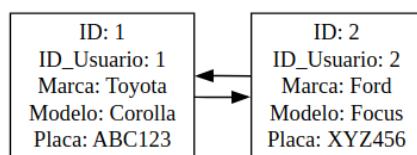


Reportes

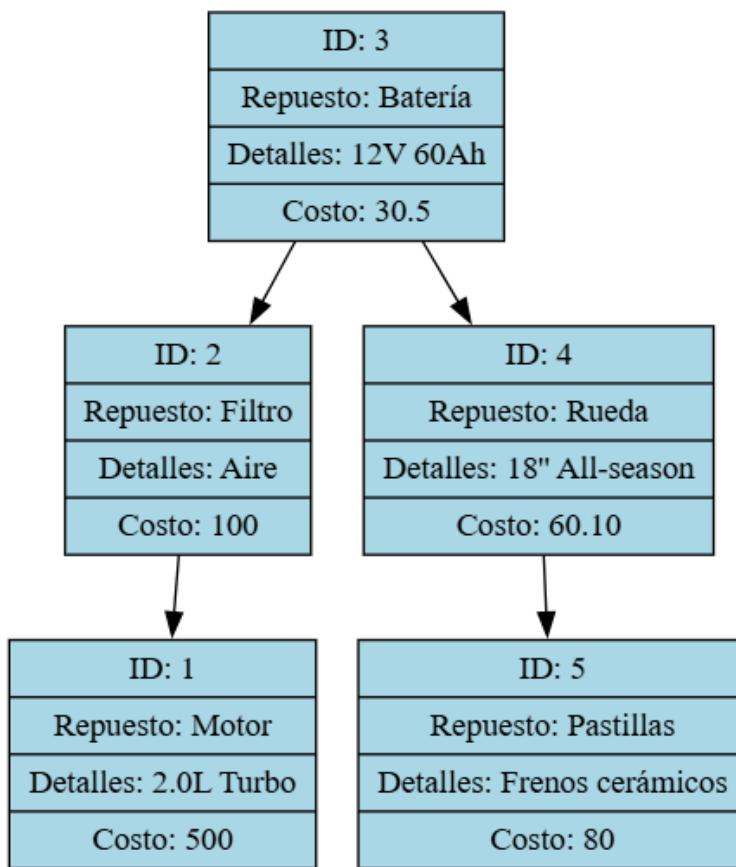
- **Usuarios**



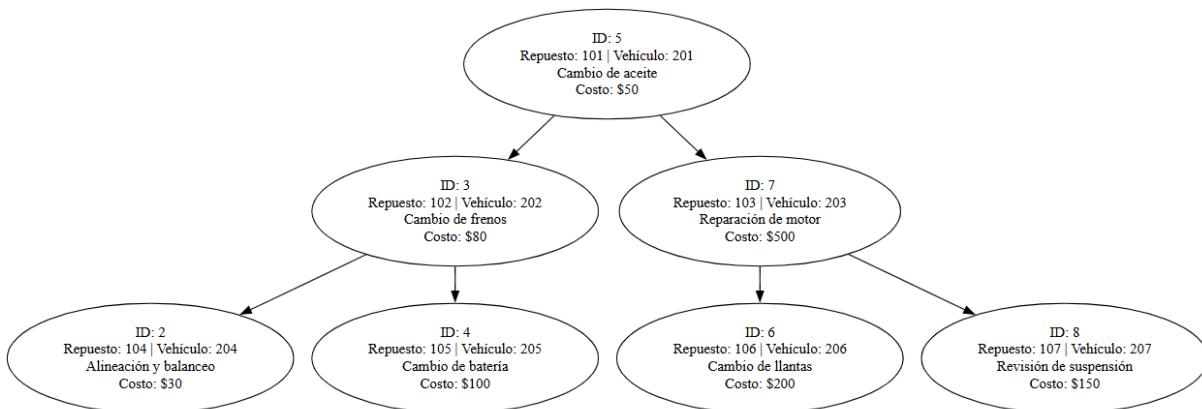
- **Vehículos**



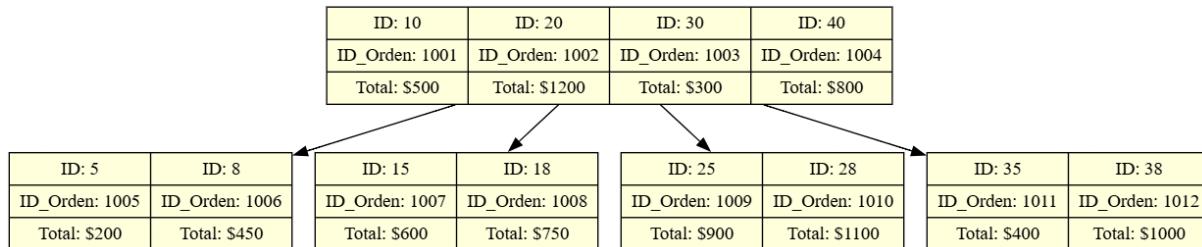
- **Repuestos**



- **Servicios**



- **Facturación**



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 **admin@usac.com** y su contraseña será **admin123**. Este usuario ya debe de estar en el sistema, por lo cual no se puede ni insertar manualmente ni cargarlo por medio del archivo JSON.
- Sistema Operativo: Linux (distribución libre).
- IDE: Libre.
- 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: **28/03/2025 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: MarItzep

Entregables

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