

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 avanzado para un taller de reparación de vehículos que administre eficientemente sus entidades, integrando estructuras de datos complejas para optimizar organización, seguridad y consulta de información.

Objetivos Específicos

1. Implementar un grafo no dirigido para modelar relaciones entre vehículos y repuestos, facilitando análisis de uso.
2. Utilizar compresión Huffman para generar y comprimir reportes en texto plano de entidades (Vehículos, Repuestos, Servicios).
3. Aplicar Blockchain para almacenar y asegurar la información de usuarios, reemplazando la lista simplemente enlazada.
4. Reemplazar el árbol B de orden 5 por un Árbol de Merkle para gestionar facturas, garantizando integridad con atributos adicionales.
5. Mantener estructuras existentes (lista doblemente enlazada, árbol AVL, árbol binario) para vehículos, repuestos y servicios, optimizando su uso.
6. Generar reportes visuales con Graphviz para representar gráficamente las estructuras implementadas.

Estructuras a Utilizar

- **Grafo no dirigido:** Modela relaciones entre vehículos y repuestos; eficiente para consultas relacionales.
- **Compresión Huffman:** Reduce el tamaño de reportes en texto plano, optimizando almacenamiento.
- **Blockchain:** Almacena usuarios con seguridad e inmutabilidad (index, timestamp, ID, nombre, correo, hash).
- **Árbol de Merkle:** Gestiona facturas, verifica integridad.
- **Lista doblemente enlazada:** Almacena vehículos; permite recorridos bidireccionales.
- **Árbol AVL:** Organiza repuestos; asegura búsquedas rápidas y balanceadas.
- **Árbol Binario:** Administra servicios; soporta recorridos jerárquicos eficientes.

Descripción

AutoGest Pro es un sistema integral de gestión diseñado específicamente para talleres de reparación de vehículos. Este software optimiza las operaciones diarias, desde el registro seguro de usuarios hasta la gestión de vehículos, repuestos, servicios y facturas, asegurando eficiencia y organización. En esta fase, se incorporan estructuras avanzadas como grafo no dirigido, compresión Huffman, Blockchain y Árbol de Merkle, mejorando la seguridad, el análisis relacional y el almacenamiento de datos. La interfaz gráfica, desarrollada con GTK, ofrece un entorno intuitivo y adaptable, mientras que las nuevas funcionalidades garantizan un rendimiento óptimo, escalabilidad y verificación de integridad para talleres de cualquier tamaño.

Entidades

Usuarios

Son los clientes del taller y propietarios de los vehículos. En esta fase, se almacenarán en una estructura de Blockchain (Explicación en el apartado de Blockchain) para garantizar seguridad, inmutabilidad y trazabilidad de sus registros. Para el atributo Contraseña deberá ser encriptada y se utilizará la encriptación SHA-256.

- ID
- Nombres
- Apellidos
- Correo
- Edad
- Contraseña

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 registrarla.

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. En esta fase, se almacenarán en un Árbol de Merkle, reemplazando el árbol B de orden 5, para garantizar la integridad y verificación eficiente de los datos, optimizando la gestión de facturación.

- ID
- ID_Servicio
- Total
- Fecha
- Método de Pago

Validaciones: El ID de la factura debe ser único en el sistema.



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”.

Inicio de Sesión

Correo

Contraseña

Validar

Rol Administrador

Carga Masiva

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

Menu Carga masiva

Vehiculos

Cargar

Sugerencia 1

Carga Masiva Usuario

Usuario	Cargar
Vehiculos	Cargar
Repuestos	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"  
  }  
]
```

Vehículos

```
[  
  {  
    "ID": 1,  
    "ID_Usuario": 1,  
    "Marca": "Honda Civic",  
    "Modelo": 2005,  
    "Placa": "XZJ7H9K"  
  }  
]
```

Repuestos

```
[  
  {  
    "ID": 1,  
    "Repuesto": "Filtro de aceite",  
    "Detalles": "Filtra impurezas del aceite del motor.",  
    "Costo": 25.50  
  }  
]
```



Inserción de Usuarios

Realizará la inserción respetando siempre las reglas de blockchain.

Insertar Usuario

Id	<input type="text"/>
Nombres	<input type="text"/>
Apellidos	<input type="text"/>
Correo	<input type="text"/>
Edad	<input type="text"/>
Contrasenia	<input type="text"/>

Insertar

Sugerencia 1

Visualización de Usuarios

Este lo podrá realizar por medio del ID del usuario.

Insertar Usuario

Id	<input type="text"/>
Nombres	<input type="text"/>
Apellidos	<input type="text"/>
Correo	<input type="text"/>
Edad	<input type="text"/>
Contrasenia	<input type="text"/>

Buscar

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

Esta funcionalidad es esencial, ya que integra múltiples operaciones críticas en el sistema. Al generar un servicio, se ejecutarán las siguientes acciones:

- Inserción de un nodo en el árbol binario de servicios con los datos correspondientes.
- Creación automática de una factura con los atributos definidos en el apartado de Entidades.
- Establecimiento de una relación en el grafo no dirigido entre el ID del vehículo y el ID del repuesto utilizado (detallado en el apartado de Grafos).

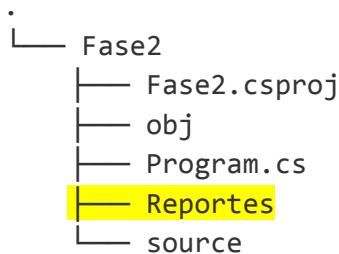
Crear Servicio

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



Generación de Reportes

Esta opción permitirá crear imágenes de los reportes que se explicarán 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.



Generar Backup

Esta funcionalidad permite crear una copia de seguridad de las entidades del sistema para su almacenamiento y recuperación. Se realizarán las siguientes acciones:

1. Generación de un archivo en formato JSON con la entidad Usuarios (almacenada en Blockchain), sin aplicar compresión.
2. Se debe comprimir la información por medio del método Huffman y almacenarlos dentro de un archivo de texto plano con extensión .edd para las siguientes entidades VEHÍCULOS y REPUESTOS.
- **Nota:** La compresión mediante el método Huffman se aplicará únicamente a los archivos .edd de Vehículos y Repuestos, optimizando su tamaño. El archivo JSON de Usuarios no será comprimido.

Cargar Backup

Esta funcionalidad se ejecuta al iniciar el programa y permite restaurar las entidades del sistema desde los archivos de backup, asegurando su integridad y consistencia. Se realizarán las siguientes acciones:

- Carga del archivo JSON de Usuarios en la estructura Blockchain, validando que la cadena no esté corrupta mediante la verificación de los hashes y el campo PREVIOUS HASH de cada bloque.
- Descompresión de los archivos .edd de Vehículos y Repuestos, cargando los datos descomprimidos automáticamente en sus estructuras respectivas: lista doblemente enlazada para Vehículos y árbol AVL para Repuestos.
- Validación de consistencia: Se verificará que la cantidad de vehículos y repuestos cargados coincida con los registros previos al backup. Si se detecta corrupción en el Blockchain o discrepancias en los datos, se mostrará un mensaje de error y no se procederá con la carga.



Rol Usuario

Visualización de Vehículos

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

Visualización de Vehículos				
Id	Id_Usuario	Marca	Modelo	Placa

Sugerencia 1

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				
<input type="button" value="Post-Orden"/> ▼				
Id	Repuesto	Vehiculo	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

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.

user@usac.com
Visualizar Vehículo
Visualización de Servicios
Visualización de Facturas
Cerrar Sesión

Blockchain

Es una cadena de bloques que opera como una lista enlazada, donde cada bloque se inserta al inicio. Cada bloque contiene:

- **INDEX:** Número del bloque en la cadena, comenzando en 0 para el bloque génesis e incrementándose en 1 por cada nuevo usuario (1, 2, 3, etc.).
- **TIMESTAMP:** Fecha y hora de creación del bloque, en formato (DD-MM-YY::HH:MM:SS).
- **DATA:** Información del usuario (ID, Nombres, Apellidos, Correo, Edad, Contraseña).
- **NONCE:** Número entero que se itera desde 0, incrementándose en 1, hasta cumplir la prueba de trabajo.
- **PREVIOUS HASH:** Hash del bloque anterior; para el bloque génesis, su valor es "0000". Actúa como apuntador para detectar alteraciones.
- **HASH:** Identificador único del bloque, generado con SHA-256 aplicado a la concatenación de INDEX + TIMESTAMP + DATA + NONCE + PREVIOUS HASH (sin espacios ni saltos de línea).

Prueba de trabajo:

Se busca un hash válido con un prefijo de 4 ceros (ej. "0000"). El NONCE se itera desde 0 hasta encontrar un hash que cumpla esta condición, asegurando la integridad y dificultad computacional del bloque.

Generación de bloques:

Un nuevo bloque se genera cada vez que se registra un usuario en el sistema, almacenando en DATA los atributos del usuario correspondiente.

Grafo No Dirigido

Es una estructura que modela las relaciones entre vehículos y repuestos utilizados en el taller, permitiendo analizar patrones de uso de manera eficiente. Cada grafo contiene:

- **Nodos:** Representan los vehículos (identificados por su ID) y los repuestos (identificados por su ID).
- **Aristas:** Indican una relación entre un vehículo y un repuesto, estableciendo que dicho repuesto fue utilizado en el mantenimiento o reparación del vehículo. No tienen dirección, ya que la relación es bidireccional.

Generación de relaciones:

Una nueva arista se genera automáticamente cada vez que se registra un servicio en el sistema (ver apartado Generar Servicio), vinculando el ID del vehículo con el ID del repuesto asociado al servicio.



Reportes

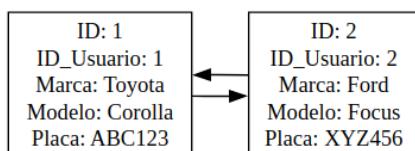
- Usuarios

```

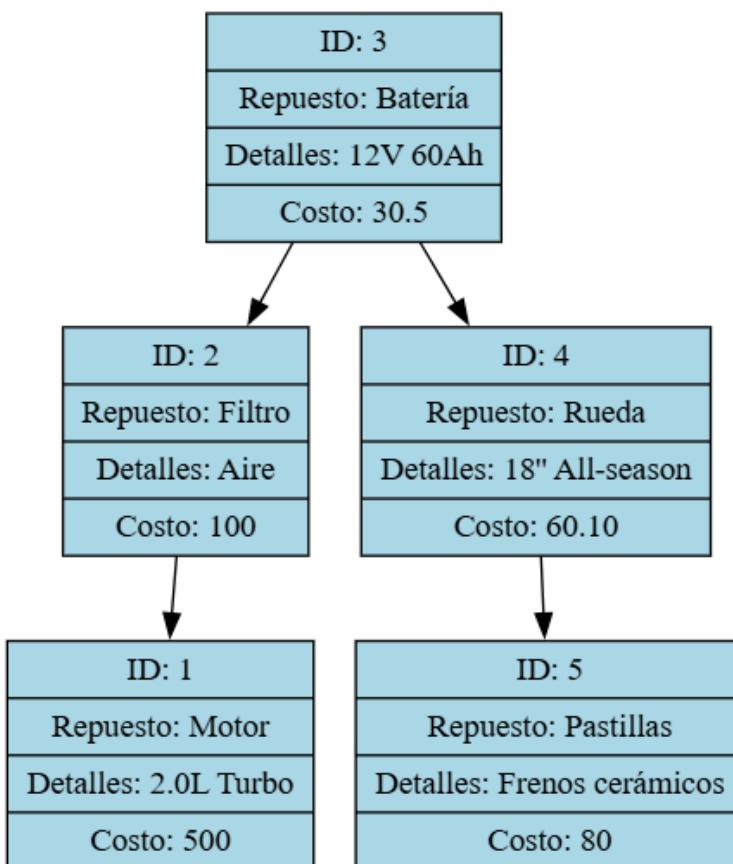
INDEX: 0
TIMESTAMP: 27-03-25::10:00:00
DATA: {ID: 1, Nombres: Juan, Apellidos: Pérez, Correo: juan@usac.com, Edad: 25, Contraseña: pass123}
NONCE: 542
PREVIOUS HASH: 0000
HASH: 0000a1b2c3d4

```

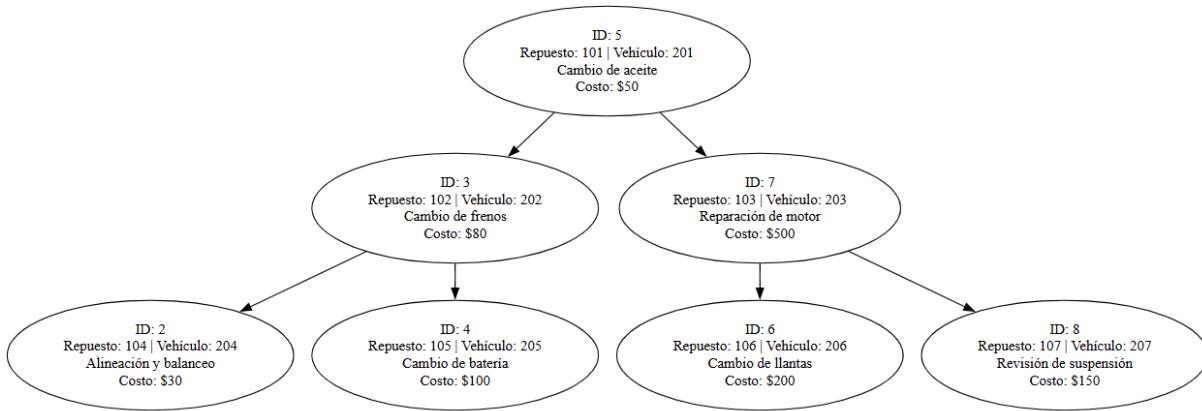
- Vehículos



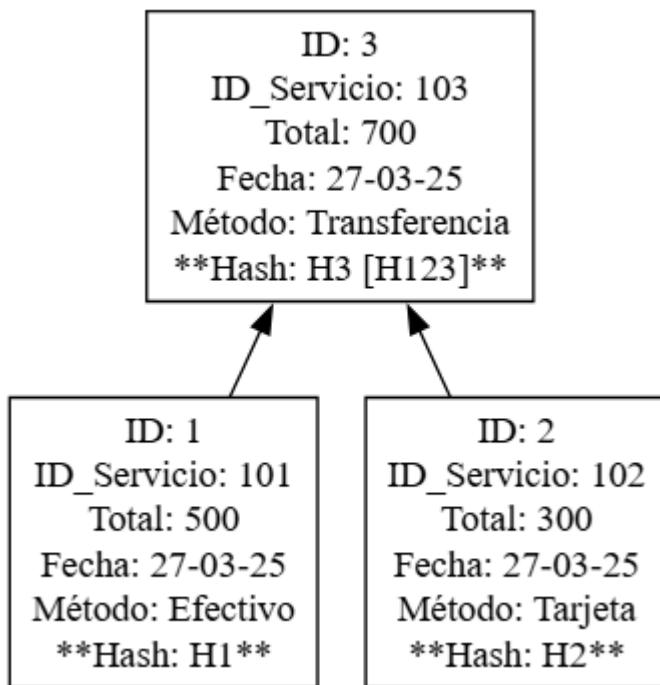
- Repuestos



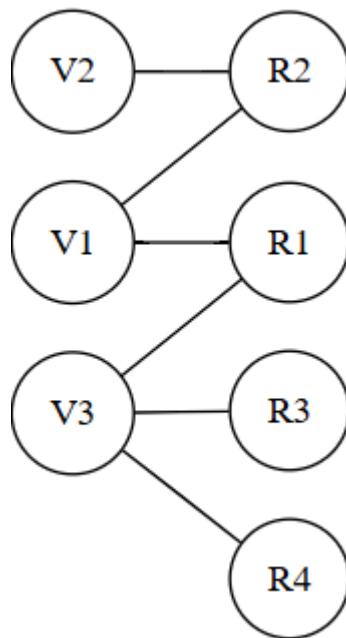
- **Servicios(Aplicar recorrido IN-ORDEN)**



- **Facturación**



Grafo no dirigido



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"  
  }]
```

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: **1/05/2025
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**