Recuperatorio Laboratorio

a)

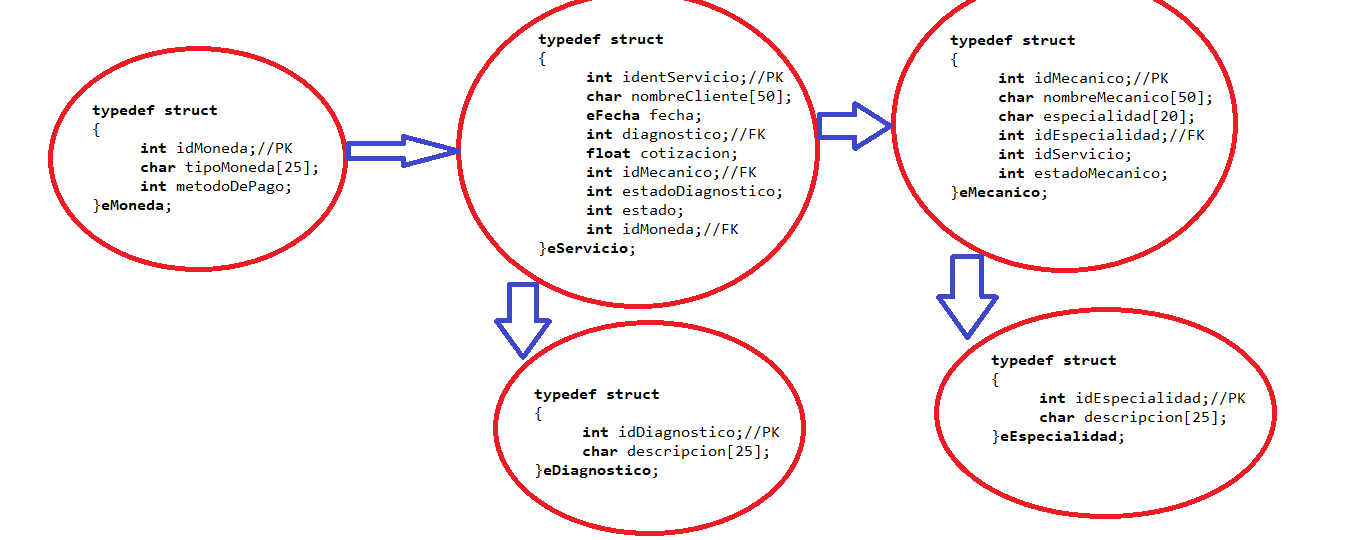
Alumno: Ismael Alejandro Frias

DNI: 44253413

Materia: Recuperatorio 1er Parcial Laboratorio

División: B

b)



C)

Agregar la siguiente estructura a su modelo de datos.

Tema 1: Diagnóstico (identificador de diagnóstico, descripción)

Tema 2: Especialidad (identificador de especialidad, descripción)

Establecer las relaciones necesarias con la estructura anterior y hardcodear datos.

Refactorizar el proyecto en función a la estructura agregada.

1. ALTA SERVICIO: (TEMA 1) (Obligatoria) Se ingresan los datos de un servicio. El id deberá ser autoincremental. No se deberá diagnosticar al momento de crear el servicio.

2. MODIFICAR SERVICIO: (TEMA 1) (TEMA 2)(Obligatoria) Se ingresará el id del

servicio, permitiendo en un submenú modificar:

● Nombre del cliente

● Fecha del servicio

● Mecánico

3. CANCELAR SERVICIO: Se ingresará el id del servicio y se realizará una baja lógica del mismo.

4. DIAGNOSTICAR: (TEMA 1) (TEMA 2)(Obligatoria) dado el ingreso del id de un mecánico, listar los servicios QUE NO ESTÉN DIAGNOSTICADOS. Se deberá ingresar el id del servicio y elegir el diagnóstico

indicado. Se deberá asignar también la cotización del mismo.

5. LISTAR:

a. Todos los mecánicos. (TEMA 2) (Obligatoria)

b. Todos los servicios. (TEMA 1)(Obligatoria)

c. Todos los mecánicos a los cuales no se les haya asignado ningún servicio. Se deberá agregar la

descripción de la especialidad del mecánico. (TEMA 2) (Obligatoria)

d. Listado de servicios ordenados por fecha, desde el más antiguo hasta el más reciente. (TEMA 1)

e. Para una fecha determinada se necesita saber cuál fue la suma de cotizaciones en concepto de cambio de

aceite. (TEMA 1)

f. Cotización promedio de cada uno de los mecánicos. (TEMA 2)

g. Determinar cuál/cuáles son los mecánicos que diagnosticaron más servicios.

h. Listado de servicios ordenados por especialidad del mecánico alfabéticamente (TEMA 2)

i. Listado de todos los servicios entre el mes de Marzo y Mayo del 2022 para una especialidad determinada.

j. Porcentaje de servicio diagnosticados que atiende cada mecánico en función del total de diagnósticos.

Parte 4:

Agregar la siguiente consulta:

(TEMA 1): El top 3 de desperfectos más diagnosticados. (agregar 3 diagnósticos más)

(TEMA 2):El top 3 de especialidades más estudiadas por los mecánicos.

e) Enlace de drive con el video:

<https://drive.google.com/file/d/1lvvue3pLBird6m8TUUzUFzAndZmPlEk2/view?usp=sharing>

d)

Servicio.h:

/// @brief inicializa los servicios en libre

/// @param miServicios

/// @param sizeServicios

/// @return retorna si se pudo hacer

int inicializarServicios(eServicio miServicios[], int sizeServicios);

/// @brief busca espacio libre en el array

/// @param miServicios

/// @param sizeServicios

/// @return retorn el indice donde hay un espacio libre

int buscarLugarLibre(eServicio miServicios[], int sizeServicios);

/// @brief carga un cliente

/// @param miServicios

/// @param sizeServicios

/// @param contadorIDs

/// @return retorna un entero si la carga fue o no exitosa

int agregarServicio(eServicio miServicios[], int sizeServicios, int contadorIDs);

/// @brief busca un id ingresado por el usuario en el array

/// @param miServicios

/// @param sizeServicios

/// @return retorna el ID si es que lo encontro

int buscarServicioPorID(eServicio miServicios[], int sizeServicios);

/// @brief modifica el nombre del cliente ingresado por ID

/// @param miServicios

/// @param sizeServicios

/// @param idServicio

/// @return retonar un entero si la modificacion fue exitosa

int ModificarNombreDelCliente(eServicio miServicios[], int sizeServicios, int idServicio);

/// @brief modifica la fecha del servicio ingresado por ID

/// @param miServicios

/// @param sizeServicios

/// @param idServicio

/// @return retonar un entero si la modificacion fue exitosa

int ModificarFechaDelServicio(eServicio miServicios[], int sizeServicios, int idServicio);

/// @brief da de baja el servicio ingresado por ID

/// @param miServicios

/// @param sizeServicios

/// @param idServicio

/// @return retonar un entero si la baja fue exitosa

int cancelarServicio(eServicio miServicios[], int sizeServicios, int idServicio);

/// @brief diagnostica un servicio que se haya diagnosticado antes

/// @param miServicios

/// @param sizeServicios

/// @param idServicio

/// @param miDiagnosticos

/// @param idMecanico

/// @param miMecanicos

/// @param moneda

/// @return retorna si la diagnosticacion fue exitosa o no

int diagnosticarServicio(eServicio miServicios[], int sizeServicios, int idServicio, eDiagnostico miDiagnosticos[], int idMecanico, eMecanico miMecanicos[], eMoneda moneda[]);

/// @brief muestra la lista de todos los servicios

/// @param miServicios

/// @param sizeServicios

void mostrarListaServicios(eServicio miServicios[], int sizeServicios);

/// @brief busca un servicio no diagnosticado mediante el ID ingresado

/// @param miServicios

/// @param sizeServicios

/// @return retorna un entero si lo encontro

int buscarServicioNoDiagnosticado(eServicio miServicios[], int sizeServicios);

/// @brief muestra una lista de los servicios que no se han diagnosticado aún

/// @param miServicios

/// @param sizeServicios

void mostrarListaServiciosNoDiagnosticados(eServicio miServicios[], int sizeServicios);

/// @brief inicializa todos los diagnosticos en "no diagnosticaos" por defecto

/// @param miServicios

/// @param sizeServicios

void inicializarDiagnostico(eServicio miServicios[], int sizeServicios);

/// @brief muestra una lista de los servicios diagnosticados

/// @param miServicios

/// @param sizeServicios

void mostrarListaServiciosDiagnosticados(eServicio miServicios[], int sizeServicios);

/// @brief busca un servicio diagnosticaso mediante el ID ingresado

/// @param miServicios

/// @param sizeServicios

/// @param miMecanicos

/// @param sizeMecanicos

/// @return retorna el ID si lo encontro

int buscarServicioDiagnosticado(eServicio miServicios[], int sizeServicios, eMecanico miMecanicos[], int sizeMecanicos);

/// @brief muesta una lista de los servicios ordenados por fecha de menor a mayor

/// @param miServicios

/// @param sizeServicios

void listadoDeServiciosOrdenadosPorFecha(eServicio miServicios[], int sizeServicios);

/// @brief muestra una lista de las cotizacion total por cambio de aceite dada una fecha ingresada

/// @param miServicios

/// @param sizeServicios

void fechaCotizacionesCambioDeAceite(eServicio miServicios[], int sizeServicios);

/// @brief muestra una lista de la cotizacion promedio por mecanico

/// @param miServicios

/// @param sizeServicios

void cotizacionPromedioPorMecanico(eServicio miServicios[], int sizeServicios);

/// @brief muestra cual/cuales son los mecanicos con mas servicios

/// @param miMecanicos

/// @param sizeMecanicos

/// @param miServicios

/// @param sizeServicios

void mecanicosConMasServicios(eMecanico miMecanicos[], int sizeMecanicos, eServicio miServicios[], int sizeServicios);

/// @brief muestra una lista de servicios ordenados por especialidad

/// @param miServicios

/// @param sizeServicios

/// @param miMecanicos

/// @param sizeMecanicos

void serviciosOrdenadosPorEspecialidadAlfabeticamente(eServicio miServicios[], int sizeServicios, eMecanico miMecanicos[], int sizeMecanicos);

/// @brief muestra una lista de los servicios diagnosticados entre marzo y mayo de 2022 si es que los hay

/// @param miServicios

/// @param sizeServicios

/// @param miEspecialidad

void serviciosEntreMarzoYMayo(eServicio miServicios[], int sizeServicios, eEspecialidad miEspecialidad[]);

/// @brief muestra una lista del porcentaje de diagnosticos por mecanico

/// @param miServicios

/// @param sizeServicios

void porcentajeDeDiagnosticosPorMecanicos(eServicio miServicios[], int sizeServicios);

Mecanico.h:

/// @brief muestra una lista de mecanicos

/// @param miMecanicos

/// @param miEspecialidad

void listaDeMecanicos(eMecanico miMecanicos[], eEspecialidad miEspecialidad[]);

/// @brief carga los mecanicos hardcodeados

/// @param miMecanicos

void cargarMecanicos(eMecanico miMecanicos[]);

/// @brief busca un mecanico mediante el ID ingresado por el usuario

/// @param miMecanicos

/// @param sizeMecanicos

/// @return retorna el ID del mecanico si lo encontro

int buscarMecanicoPorID(eMecanico miMecanicos[], int sizeMecanicos);

/// @brief modifica el ID del mecanico

/// @param miMecanicos

/// @param sizeMecanicos

/// @param miServicios

/// @param sizeServicios

/// @param miEspecialidad

/// @return retorna un entero si la modificacion fue exitosa

int ModificarMecanico(eMecanico miMecanicos[], int sizeMecanicos, eServicio miServicios[], int sizeServicios, eEspecialidad miEspecialidad[]);

/// @brief muestra una lista de mecanicos que no se les haya asigando ningun diagnostico

/// @param miMecanicos

/// @param sizeMecanicos

/// @param miEspecialidad

void listaDeMecanicosSinServicio(eMecanico miMecanicos[], int sizeMecanicos, eEspecialidad miEspecialidad[]);

Diagnostico.h:

/// @brief carga los diagnosticos hardcodeados

/// @param miDiagnosticos

void cargarDiagnosticos(eDiagnostico miDiagnosticos[]);

/// @brief muestra una lista de diagnosticos

/// @param miDiagnosticos

void listaDeDiagnosticos(eDiagnostico miDiagnosticos[]);

/// @brief muestra un top 3 de desperfectos mas diagnosticados

/// @param miDiagnosticos

/// @param miServicios

/// @param sizeServicios

void desperfectosMasDiagnosticados(eDiagnostico miDiagnosticos[], eServicio miServicios[], int sizeServicios);

Especialidad.h:

/// @brief carga las especialidad harcodedas

/// @param miEspecialidad

void cargarEspecialidad(eEspecialidad miEspecialidad[]);

/// @brief muestra una lista de especialidades

/// @param miEspecialidad

void listaDeEspecialidades(eEspecialidad miEspecialidad[]);

/// @brief muestra un top de especialidades mas estudiadas por mecanico

/// @param miEspecialidad

/// @param miServicios

/// @param sizeServicios

/// @param miMecanicos

void especialidadesMasEstudidasPorMecanico(eEspecialidad miEspecialidad[], eServicio miServicios[], int sizeServicios, eMecanico miMecanicos[]);

Moneda.h:

/// @brief carga los tipos de monedas hardcodeadas

/// @param moneda

void cargarMonedas(eMoneda moneda[]);

/// @brief muestra los servicios con su moneda, especialidad, moneda y metodo de pago

/// @param moneda

/// @param miServicios

/// @param sizeServicios

/// @param miMecanicos

/// @param miDiagnosticos

void servciosMonedaYEspecialidad(eMoneda moneda[], eServicio miServicios[], int sizeServicios, eMecanico miMecanicos[], eDiagnostico miDiagnosticos[]);