Recuperatorio Primer Parcial Laboratorio I

Alumno: Santiago Iannello

DNI: 44195364

División: 1” B”

Docentes: Scarafilo, Lucchetta, Taboada, Fernandez

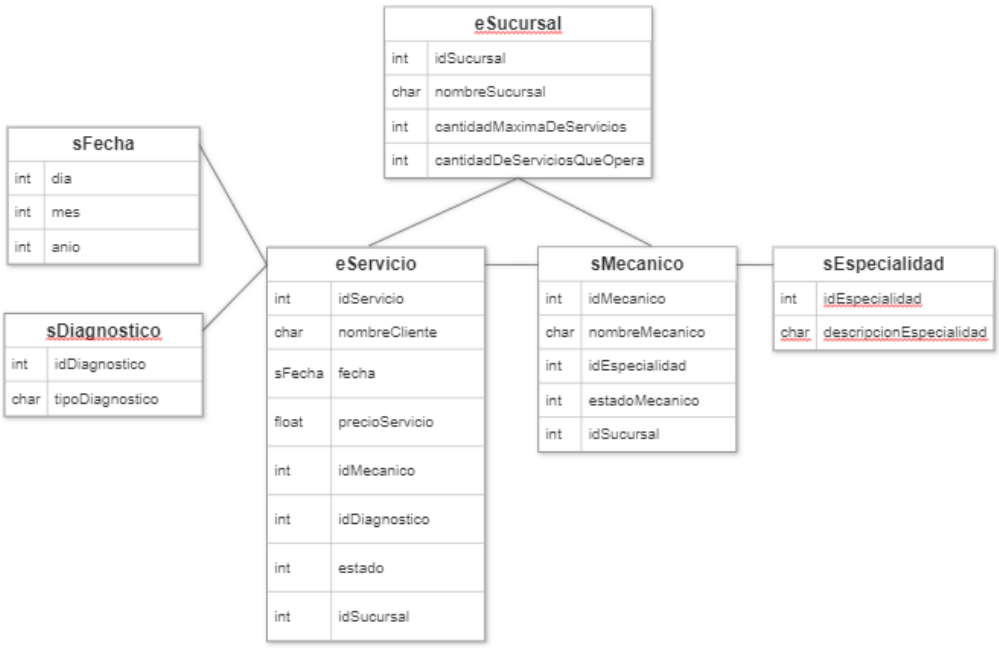
Estructura agregada

```
typedef struct
{
	int idSucursal;
	char nombreSucursal[21];
	int cantidadMaximaDeServicios;
	int cantidadDeServiciosQueOpera;
}eSucursal;
```

La estructura **eSucursal** registra las sucursales posibles para un servicio y un mecánico. La misma consta de 4 campos, los cuales son:

- idSucursal: Guarda el ID de cada sucursal, para identificarlas.
- nombreSucursal: Guarda el nombre con el que se reconoce a cada sucursal.
- cantidadMaximaDeServicios: Guarda la cantidad máxima de servicios que pueden estar operando a la vez en una sucursal.
- cantidadDeServiciosQueOpera: Guarda la cantidad de servicios que están operando a la vez en una sucursal, para corroborar que no se exceda el máximo.

La estructura **eSucursal** se relaciona con las estructuras **eServicio** y **sMecanico** mediante el campo “idSucursal”. Esto para registrar que servicios y que mecánicos están operando en cada sucursal. En **eSucursal**, el campo “idSucursal” es la Primary Key (PK), mientras que los campos con el mismo nombre (“idSucursal”), en las estructuras de servicios y mecánicos son las Foreign Key (FK).



Informe anexado:

void *ListarMecanicosYServiciosDeCadaSucursal*(eServicio listaServicios[], **int** sizeServicios, sMecanico listaMecanicos[], **int** sizeMecanicos, eSucursal listaSucursales[], **int** sizeSucursales);

- @brief Muestra los mecánicos y servicios que operan en cada sucursal
- @param listaServicios: Recibe la lista de servicios que cargó el usuario.
- @param sizeServicios: Recibe el tamaño de la lista de servicios(TAM).
- @param listaMecanicos: Recibe la lista de mecanicos(hardcodeada).
- @param sizeMecanicos: Recibe el tamaño de la lista de mecanicos(T).
- @param listaSucursales: Recibe la lista de sucursales(hardcodeada).
- @param sizeSucursales: Recibe el tamaño de la lista de sucursales(S1).

Repositorio de GitHub con el trabajo:

https://github.com/SantiSTC/recuperatorio_laboratorio_1.git

Drive con el video explicativo:

https://drive.google.com/file/d/1N_KhJ7jWQ3wT2oBnx7F7rc4wdmlkHS8Z/view?usp=sharing

Prototipo de las funciones del programa:

```
/// @brief Permite modificar el nombre, la fecha o el mecanico encargado
de cada uno de los servicios ingresados.
/// @param listaServicios Recibe la lista de servicios que cargó el
usuario.
/// @param sizeServicios Recibe el tamaño de la lista de servicios(TAM).
/// @param listaMecanicos Recibe la lista de mecanicos(hardcodeada).
/// @param sizeMecanicos Recibe el tamaño de la lista de mecanicos(T).
/// @return Retorna -1 o 1 para corroborar que pudo funcionar
correctamente e informarlo al usuario.
```

- **int** ModificarServicio(eServicio listaServicios[], **int** sizeServicios, sMecanico listaMecanicos[], **int** sizeMecanicos);

```
/// @brief Permite asignar un diagnostico a cada uno de los servicios
ingresados, ademas se le asigna la cotizacion a este mismo.
/// @param listaServicios Recibe la lista de servicios que cargó el
usuario.
/// @param sizeServicios Recibe el tamaño de la lista de servicios(TAM).
/// @param listaMecanicos Recibe la lista de mecanicos(hardcodeada).
/// @param sizeMecanicos Recibe el tamaño de la lista de mecanicos(T).
/// @param listaDiagnosticos Recibe la lista de
diagnosticos(hardcodeada).
/// @param sizeDiagnosticos Recibe el tamaño de la lista de
diagnosticos(SIZE).
/// @return Retorna -1 o 1 para corroborar que pudo funcionar
correctamente e informarlo al usuario.
```

- **int** DiagnosticarServicio(eServicio listaServicios[], **int** sizeServicios, sMecanico listaMecanicos[], **int** sizeMecanicos, sDiagnostico listaDiagnosticos[], **int** sizeDiagnosticos, eSucursal listaSucursales[], **int** sizeSucursales);

```
/// @brief Cambia el estado de un mecanico a OCUPADO.
/// @param listaMecanicos Recibe la lista de mecanicos(hardcodeada).
/// @param sizeMecanicos Recibe el tamaño de la lista de mecanicos(T).
/// @param idAuxMecanico Recibe un ID ingresado por el usuario que indica
el ID del mecanico sobre el que se hará la modificacion.
```

- **void** AsignarOcupadoAMecanico(sMecanico listaMecanicos[], **int** sizeMecanicos, **int** idAuxMecanico, **int** idAuxSucursal);

```
/// @brief Permite calcular el dinero promedio que le ha ingresado a cada
mecanico.
/// @param listaServicios Recibe la lista de servicios que cargó el
usuario.
/// @param sizeServicios Recibe el tamaño de la lista de servicios(TAM).
/// @param listaMecanicos Recibe la lista de mecanicos(hardcodeada).
/// @param sizeMecanicos Recibe el tamaño de la lista de mecanicos(T).
```

- **void** CalcularCotizacionPromedioPorMecanico(eServicio listaServicios[], **int** sizeServicios, sMecanico listaMecanicos[], **int** sizeMecanicos);

```
/// @brief Muestra al mecanico que mas diagnosticos tiene asignados.
/// @param listaServicios Recibe la lista de servicios que cargó el
usuario.
/// @param sizeServicios Recibe el tamaño de la lista de servicios(TAM).
/// @param listaMecanicos Recibe la lista de mecanicos(hardcodeada).
/// @param sizeMecanicos Recibe el tamaño de la lista de mecanicos(T).
```

- **void** MecanicosConMasDiagnosticos(eServicio listaServicios[], **int** sizeServicios, sMecanico listaMecanicos[], **int** sizeMecanicos);

```
/// @brief Ordena la lista de servicios, alfabeticamente, segun la
especialidad del mecanico encargado.
/// @param listaServicios Recibe la lista de servicios que cargó el
usuario.
/// @param sizeServicios Recibe el tamaño de la lista de servicios(TAM).
/// @param listaMecanicos Recibe la lista de mecanicos(hardcodeada).
/// @param sizeMecanicos Recibe el tamaño de la lista de mecanicos(T).
```

- **void** OrdenarListadosAlfabeticamentePorEspecialidadDelMecanico(eServicio listaServicios[], **int** sizeServicios, sMecanico listaMecanicos[], **int** sizeMecanicos, sEspecialidad listaEspecialidades[], **int** sizeEspecialidad);

```
/// @brief Muestra una lista de los servicios con fechas entre Marzo y
Mayo de 2022, para una especialidad determinada(ingresada por el
usuario).
/// @param listaServicios Recibe la lista de servicios que cargó el
usuario.
/// @param sizeServicios Recibe el tamaño de la lista de servicios(TAM).
/// @param listaDiagnosticos Recibe la lista de
diagnosticos(hardcodeada).
/// @param sizeDiagnosticos Recibe el tamaño de la lista de
diagnosticos(SIZE).
```

- **void** ListadoDeServiciosEntreMarzoYMayo2022(eServicio listaServicios[], **int** sizeServicios, sDiagnostico listaDiagnosticos[], **int** sizeDiagnosticos);

```
/// @brief Calcula el porcentaje de diagnosticos que atendió cada
mecanico en funcion al total de servicios diagnosticados.
/// @param listaServicios Recibe la lista de servicios que cargó el
usuario.
/// @param sizeServicios Recibe el tamaño de la lista de servicios(TAM).
/// @param listaMecanicos Recibe la lista de mecanicos(hardcodeada).
/// @param sizeMecanicos Recibe el tamaño de la lista de mecanicos(T).
```

- **void** CalcularPorcentajeDeDiagnosticosPorMecanico(eServicio listaServicios[], **int** sizeServicios, sMecanico listaMecanicos[], **int** sizeMecanicos);

/// @brief Permite ingresar a los listados, mostrando las 10 opciones disponibles.
/// @param listaServicios Recibe la lista de servicios que cargó el usuario.
/// @param sizeServicios Recibe el tamaño de la lista de servicios(TAM).
/// @param listaMecanicos Recibe la lista de mecanicos(hardcodeada).
/// @param sizeMecanicos Recibe el tamaño de la lista de mecanicos(T).
/// @return Retorna -1 o 1 para corroborar que pudo funcionar correctamente e informarlo al usuario.

- **int** ListarServicios(eServicio listaServicios[], **int** sizeServicios, sMecanico listaMecanicos[], **int** sizeMecanicos, sDiagnostico listaDiagnosticos[], **int** sizeDiagnosticos, sAuxiliar listaAuxiliar[], **int** sizeAuxiliar, sEspecialidad listaEspecialidades[], **int** sizeEspecialidades, eSucursal listaSucursales[], **int** sizeSucursales);

/// @brief Calcula los 3 desperfectos mas elegidos a la hora de diagnosticar.
/// @param listaServicios Recibe la lista de servicios que cargó el usuario.
/// @param sizeServicios Recibe el tamaño de la lista de servicios(TAM).
/// @param listaDiagnosticos Recibe la lista de diagnosticos(hardcodeada).
/// @param sizeDiagnosticos Recibe el tamaño de la lista de diagnosticos(SIZE).
/// @param listaAuxiliar Recibe una lista auxiliar para poder realizar un ordenamiento.
/// @param sizeAuxiliar Recibe el tamaño de la lista auxiliar(SIZE).

- **void** CalcularTop3DesperfectosMasDiagnosticados(eServicio listaServicios[], **int** sizeServicios, sDiagnostico listaDiagnosticos[], **int** sizeDiagnosticos, sAuxiliar listaAuxiliar[], **int** sizeAuxiliar);

/// @brief Calcula las 3 especialidades que mas aparecen entre los mecanicos.
/// @param listaEspecialidades Recibe la lista de especialidades posibles que puede tener un mecanico.
/// @param sizeEspecialidades Recibe el tamaño de la lista de especialidades(T1).
/// @param listaMecanicos Recibe la lista de mecanicos(hardcodeada).
/// @param sizeMecanicos Recibe el tamaño de la lista de mecanicos(T).
/// @param listaAuxiliar Recibe una lista auxiliar para poder realizar un ordenamiento.
/// @param sizeAuxiliar Recibe el tamaño de la lista auxiliar(SIZE).

- **void** CalcularTop3EspecialidadesMasEstudiadas(sEspecialidad listaEspecialidades[], **int** sizeEspecialidades, sMecanico listaMecanicos[], **int** sizeMecanicos, sAuxiliar listaAuxiliar[], **int** sizeAuxiliar);

```
/// @brief Ordena la lista de mecanicos alfabeticamente, por su
especialidad.
/// @param listaMecanicos Recibe la lista de mecanicos(hardcodeada).
/// @param sizeMecanicos Recibe el tamaño de la lista de mecanicos(T).
```
- **void** OrdenarMecanicosAlfabeticamentePorEspecialidad(sMecanico listaMecanicos[], **int** sizeMecanicos, sEspecialidad listaEspecialidades[], **int** sizeEspecialidad);

```
//Filtro inventado para la esctructura inventada(Recuperatorio)
/// @brief Muestra los mecanicos y servicios que operan en cada sucursal
/// @param listaServicios Recibe la lista de servicios que cargó el
usuario.
/// @param sizeServicios Recibe el tamaño de la lista de servicios(TAM).
/// @param listaMecanicos Recibe la lista de mecanicos(hardcodeada).
/// @param sizeMecanicos Recibe el tamaño de la lista de mecanicos(T).
/// @param listaSucursales Recibe la lista de sucursales(hardcodeada).
/// @param sizeSucursales Recibe el tamaño de la lista de sucursales(S1).
```
- **void** ListarMecanicosYServiciosDeCadaSucursal(eServicio listaServicios[], **int** sizeServicios, sMecanico listaMecanicos[], **int** sizeMecanicos, eSucursal listaSucursales[], **int** sizeSucursales);

```
/// @brief Permite generar un ID autoincremental para cada servicio.
/// @return Retorna el ID generado.
```
- **int** ObtenerID();

```
/// @brief Permite la carga de un servicio, obtiene su ID, pide el nombre
y la fecha, y marca el espacio como "OCUPADO" en el array.
/// @return El servicio generado.
```
- eServicio CargarServicio();

```
/// @brief Busca el primer espacio libre dentro del array para crear un
nuevo servicio.
/// @param listaServicios Recibe la lista de servicios que cargó el
usuario.
/// @param sizeServicios Recibe el tamaño de la lista de servicios(TAM).
/// @return Retorna 1 si encontró el espacio para crear el servicio, -1
si no pudo encontrarlo.
```
- **int** BuscarEspacio(eServicio listaServicios[], **int** sizeServicios);

```
/// @brief Carga un servicio a la lista.
/// @param listaServicios Recibe la lista de servicios que cargó el
usuario.
/// @param sizeServicios Recibe el tamaño de la lista de servicios(TAM).
/// @return Retorna 1 si pudo cargar el servicio, -1 si no pudo hacerlo.
```
- **int** CargarListaServicios(eServicio listaServicios[], **int** sizeServicios);

```
/// @brief Permite eliminar un servicio del sistema.
```

/// @param listaServicios Recibe la lista de servicios que cargó el usuario.
/// @param sizeServicios Recibe el tamaño de la lista de servicios(TAM).
/// @return Retorna -1 o 1 para corroborar que pudo funcionar correctamente e informarlo al usuario.

- **int** EliminarServicio(eServicio listaServicios[], **int** sizeServicios);

/// @brief
/// @param listaServicios Recibe la lista de servicios que cargó el usuario.
/// @param sizeServicios Recibe el tamaño de la lista de servicios(TAM).

- **void** InicializarServicio(eServicio listaServicios[], **int** sizeServicios);

/// @brief Muestra la lista de todos los servicios ingresados.
/// @param listaServicios Recibe la lista de servicios que cargó el usuario.
/// @param sizeServicios Recibe el tamaño de la lista de servicios(TAM).

- **void** MostrarListaServicios(eServicio listaServicios[], **int** sizeServicios);

/// @brief Muestra un servicio en concreto.
/// @param unServicio Recibe un solo servicio para mostrar.

- **void** MostrarUnServicio(eServicio unServicio);

/// @brief Ordena y muestra los servicios ordenados por su fecha.
/// @param listaServicios Recibe la lista de servicios que cargó el usuario.
/// @param sizeServicios Recibe el tamaño de la lista de servicios(TAM).

- **void** ServiciosOrdenadosPorFecha(eServicio listaServicios[], **int** sizeServicios);

/// @brief Permite ingresar una fecha determinada, y conocer el dinero que ingresó en tal fecha.
/// @param listaServicios Recibe la lista de servicios que cargó el usuario.
/// @param sizeServicios Recibe el tamaño de la lista de servicios(TAM).

- **void** CotizacionesEnFechaDeterminada(eServicio listaServicios[], **int** sizeServicios);

/// @brief Muestra los servicios que no fueron diagnosticados.
/// @param listaServicios Recibe la lista de servicios que cargó el usuario.
/// @param sizeServicios Recibe el tamaño de la lista de servicios(TAM).

- **void** MostrarServiciosNoDiagnosticados(eServicio listaServicios[], **int** sizeServicios);

/// @brief Muestra un diagnostico en concreto.
/// @param unDiagnostico Recibe un diagnostico para mostrar.

```
- void MostrarUnDiagnostico(sDiagnostico unDiagnostico);

/// @brief Muestra la lista de los diagnosticos.
/// @param listaDiagnosticos Recibe la lista de
diagnosticos(hardcodeada).
/// @param sizeDiagnosticos Recibe el tamaño de la lista de
diagnosticos(SIZE).
- void MostrarListaDeDiagnosticos(sDiagnostico listaDiagnosticos[],
  int sizeDiagnosticos);
/// @brief Ordena las especialidades en orden alfabetico
/// @param listaEspecialidades Recibe la lista de
especialidades(hardcodeada).
/// @param sizeEspecialidad Recibe el tamaño de la lista de
especialidades(T1).
- void OrdenarEspecialidadesAlfabeticamente(sEspecialidad
  listaEspecialidades[], int sizeEspecialidad);

/// @brief Pide el ingreso de un numero entero.
/// @return Retorna el numero ingresado.
- int PedirEntero();

/// @brief Pide el ingreso de un numero flotante.
/// @return Retorna el numero ingresado.
- float PedirFlotante();

/// @brief Pide el ingreso de una cadena de caracteres.
/// @param cadenaCaracteres Recibe una cadena de caracteres.
- void PedirCadenaDeCaracteres(char cadenaCaracteres[]);

/// @brief Pide el nombre de un cliente.
/// @param cadenaCaracteres Recibe una cadena de caracteres.
- void PedirNombreCliente(char cadenaCaracteres[]);

/// @brief Pide el dia de una fecha.
/// @return Retorna el numero ingresado.
- int PedirDia();

/// @brief Pide el mes de una fecha.
/// @return Retorna el numero ingresado.
- int PedirMes();

/// @brief Pide el año de una fecha.
/// @return Retorna el numero ingresado.
- int PedirAnio();

/// @brief Despliega el menu de los listados disponibles, y pide que se
elija uno.
/// @return Retorna el numero elegido para la opcion correspondiente.
- int MenuListados();
```

```
/// @brief Despliega el menu principal y pide que se elija una opcion.
/// @return Retorna el numero elegido para la opcion correspondiente.
```
- **int** MenuPrincipal();

```
/// @brief Muestra un mecanico en concreto.
/// @param unMecanico Recibe la informacion de UN mecanico.
```
- **void** MostrarUnMecanico(sMecanico unMecanico);

```
/// @brief Muestra la lista de mecanicos.
/// @param listaMecanicos Recibe la lista de mecanicos(hardcodeada).
/// @param sizeMecanicos Recibe el tamaño de la lista de mecanicos(T).
```
- **void** MostrarListaMecanicos(sMecanico listaMecanicos[], **int** sizeMecanicos);

```
/// @brief Muestra los mecanicos a los que NO se les asigno ningun
servicio.
/// @param listaMecanicos Recibe la lista de mecanicos(hardcodeada).
/// @param sizeMecanicos Recibe el tamaño de la lista de mecanicos(T).
```
- **void** MostrarMecanicosDesocupados(sMecanico listaMecanicos[], **int** sizeMecanico);

```
/// @brief Muestra los mecanicos a los que SI se les asigno un servicio.
/// @param listaMecanicos Recibe la lista de mecanicos(hardcodeada).
/// @param sizeMecanicos Recibe el tamaño de la lista de mecanicos(T).
```
- **void** MostrarMecanicosOcupados(sMecanico listaMecanicos[], **int** sizeMecanico);

```
/// @brief  Muestra una sucursal.
/// @param unaSucursal Recibe la informacion de UNA sucursal.
```
- **void** MostrarUnaSucursal(eSucursal unaSucursal);

```
/// @brief Muestra la lista de sucursales.
/// @param listaSucursales Recibe la lista de sucursales(hardcodeada).
/// @param sizeSucursales Recibe el tamaño de la lista de sucursales(S1).
```
- **void** MostrarListaSucursales(eSucursal listaSucursales[], **int** sizeSucursales);