



Tecnicatura Superior
en Programación
Primer Cuatrimestre
Laboratorio I

Profesores

Germán Scarafilo
Ezequiel Taboada
Giovanni Lucchetta
Mariano Fernandez

Alumno

Lucas Fernando Gomez Carmona
DNI: 44594260

Fecha: 17/06/2022

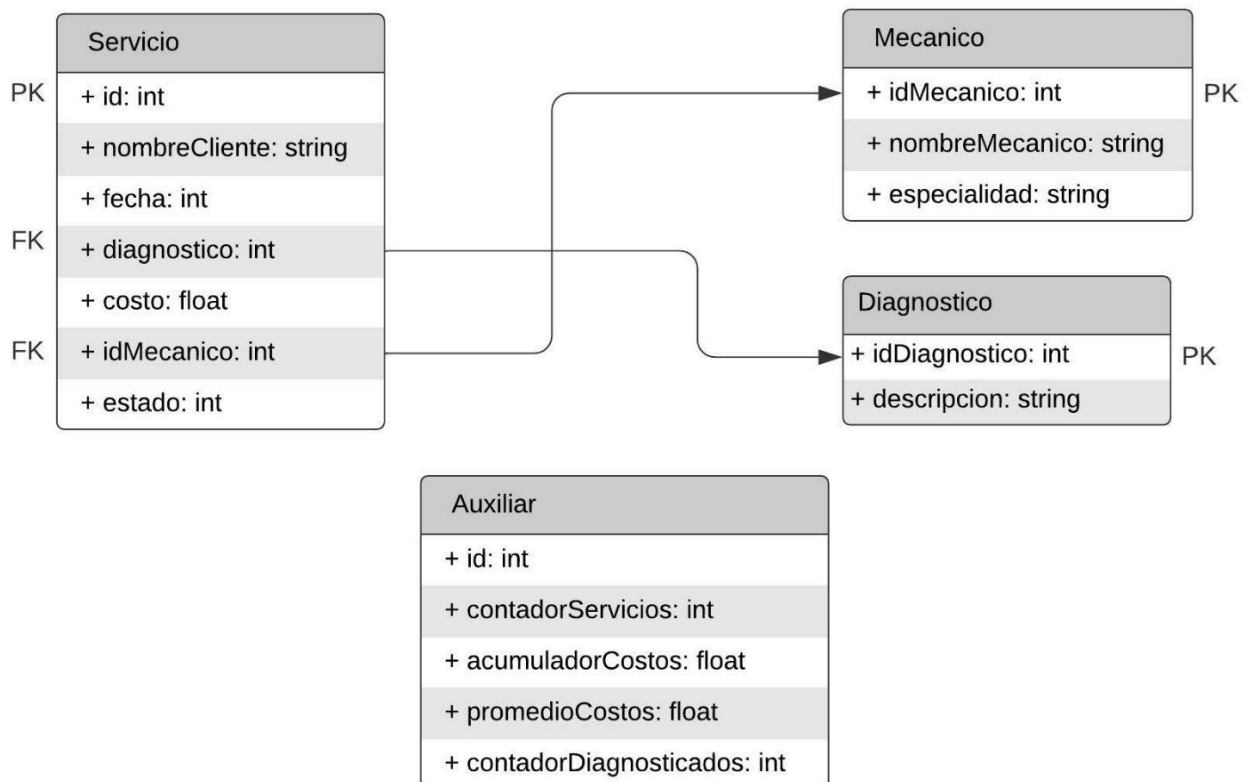
Estructura Diagnostico

Esta estructura posee un campo identificador del tipo int (idDiagnostico) la cual es la Primary Key. También tiene otro campo, en este caso del tipo string que es la descripción del diagnóstico.

En la estructura Servicio tiene una Foreign Key, el campo Diagnóstico de tipo int, nos permite acceder por medio del ID del diagnóstico a la estructura y poder hacer uso del campo "descripción".

Diagrama Primer Parcial Laboratorio I

Lucas Gomez Carmona



```

/// @brief
///
/// @param listaServicios
/// @param listaMecanicos
/// @param size
/// @return int 0 OK o -1 ERROR
int agregarServicio(Servicio listaServicios[], Mecanico
listaMecanicos[], int size);


/// @brief permite modificar una posicion del array lista
///
/// @param listaServicios array Servicio
/// @param listaMecanicos array Mecanico
/// @param size int
/// @return int 0 OK o -1 ERROR
int modificarServicio(Servicio listaServicios[], Mecanico
listaMecanicos[], int size);


/// @brief permite diagnosticar una posicion del array lista
///
/// @param listaServicios array Servicio
/// @param listaMecanicos array Mecanico
/// @param size int
/// @return int 0 OK o -1 ERROR
int diagnosticarServicio(Servicio listaServicios[], Mecanico
listaMecanicos[], Diagnostico listaDiagnosticos[], int size);


/// @brief Muestra mecanicos sin servicio
///
/// @param auxiliar
/// @param listaMecanicos
/// @return OK o ERROR
int mostrarMecaSinServicio(Auxiliar auxiliar[], Mecanico
listaMecanicos[]);

```

```

/// @brief muestra los mecanicos que no tengan ningun servicio
asignado
///
/// @param listaServicios array Servicio
/// @param listaMecanicos array Mecanico
/// @param auxiliar array Auxiliar
/// @param size int
/// @return int 0 OK o -1 ERROR
int mostrarC(Servicio listaServicios[], Mecanico
listaMecanicos[], Auxiliar auxiliar[], int size);


/// @brief Muestra los servicios diagnosticados
///
/// @param listaServicios
/// @param listaDiagnosticos
/// @param size
/// @return OK o ERROR
int mostrarServiciosDiagnosticados(Servicio
listaServicios[], Diagnostico listaDiagnosticos[], int size);


/// @brief muestra los servicios ordenados por fecha desde el
mas antiguo al mas reciente
///
/// @param listaServicios array Servicio
/// @param size int
/// @return int 0 OK o -1 ERROR
int mostrarPorFecha(Servicio listaServicios[], Diagnostico
listaDiagnosticos[], int size);


/// @brief realiza el calculo de promedio de cotizacion de los
mecanicos
///
/// @param auxiliar
/// @param listaMecanicos
/// @return OK o ERROR
int cotizacionPromedioMecanico(Auxiliar auxiliar[], Mecanico
listaMecanicos[]);

```

```

/// @brief Inicializa la estructura auxiliar
///
/// @param listaMecanicos
/// @param auxiliar
/// @return OK o ERROR
int inicializarAux(Mecanico listaMecanicos[], Auxiliar*
auxiliar[]);

/// @brief muestra el promedio de cotizacion por mecanico
///
/// @param listaServicios array Servicio
/// @param listaMecanicos array Mecanico
/// @param auxiliar array Auxiliar
/// @param size int
/// @return int 0 OK o -1 ERROR
int mostrarPromedioMecanicos(Servicio listaServicios[],
Mecanico listaMecanicos[],Auxiliar auxiliar[], int size);

/// @brief Muestra los mecanicos del punto G
///
/// @param listaMecanicos
/// @param idAux
/// @param auxiliar
/// @param contadorMayor
/// @return OK o ERROR
int mostrarMecG(Mecanico listaMecanicos[], int idAux, Auxiliar
auxiliar[], int contadorMayor);

/// @brief muestra el o los mecanicos que mas servicios
diagnosticaron
///
/// @param listaServicios array Servicio
/// @param listaMecanicos array Mecanico
/// @param auxiliar array Auxiliar
/// @param size int
/// @return int 0 OK o -1 ERROR
int mostrarG(Servicio listaServicios[], Mecanico
listaMecanicos[], Auxiliar auxiliar[], int size);

```

```

/// @brief muestra los servicios ordenados alfabeticamente por
especialidad del mecanico
///
/// @param listaServicios array Servicio
/// @param listaMecanicos array Mecanico
/// @param size int
/// @return int 0 OK o -1 ERROR
int mostrarH(Servicio listaServicios[], Mecanico
listaMecanicos[], int size);


/// @brief Muestra el porcentaje de cada mecanico
///
/// @param auxiliar
/// @param listaMecanicos
/// @param contadorDiagnosticados
/// @return OK o ERROR
int mostrarPorcentaje(Auxiliar auxiliar[], Mecanico
listaMecanicos[], int contadorDiagnosticados);


/// @brief muestra el porcentaje del total que atiende cada
mecanico
///
/// @param listaServicios array Servicio
/// @param listaMecanicos array Mecanico
/// @param auxiliar array Auxiliar
/// @param size int
/// @return int 0 OK o -1 ERROR
int mostrarJ(Servicio listaServicios[], Mecanico
listaMecanicos[], Auxiliar auxiliar[], int size);


/// @brief Contador de diagnosticos
///
/// @param size
/// @param listaServicios
/// @param auxiliar
/// @return OK o ERROR
int contadorDiag(int size, Servicio listaServicios[],Auxiliar*
auxiliar[]);

```

```

/// @brief Ordena los diagnosticos por especialidad
///
/// @param auxiliar
/// @param auxiliarD
/// @return OK o ERROR
int ordenarDiag(Auxiliar* auxiliar[], Auxiliar auxiliarD[]);

/// @brief muestra la lista correspondiente a la parte 4 del
parcial
///
/// @param listaServicios
/// @param auxiliar
/// @param listaDiagnosticos
/// @param size
/// @return OK o ERROR
int mostrarParteCuatro(Servicio listaServicios[], Auxiliar
auxiliar[], Diagnostico listaDiagnosticos[], int size);

/// @brief Inicializa todas las posiciones del array
listaServicios
///
/// @param listaServicios array Servicio
/// @param size int
/// @return int 0 OK -1 ERROR
int validarInicializado(Servicio listaServicios[], int size);

/// @brief consulta la disponibilidad del array listaServicios
///
/// @param listaServicios array Servicio
/// @param size int
/// @return int 0 SI 1 NO
int consultarDisponibilidad(Servicio listaServicios[], int
size);

```

```
/// @brief consulta el estado de un servicio ingresando el id
///
/// @param listaServicios array Servicio
/// @param size int
/// @param id int
/// @return -1 LIBRE 0 OCUPADO
int consultarEstado(Servicio listaServicios[], int size, int id);
```

```
/// @brief consulta el estado de la lista
///
/// @param listaServicios array Servicio
/// @param size int
/// @return int 0 vacia 1 llena
int estadoLista(Servicio listaServicios[], int size);
```

```
/// @brief inicializa todas las posiciones del array lista
///
/// @param listaServicios array Servicio
/// @param size int
/// @return 0 OK
int inicializarServicios(Servicio listaServicios[], int size);
```

```
/// @brief permite eliminar un servicio del array lista
///
/// @param listaServicios array Servicio
/// @param size int
/// @return int 0 OK o -1 ERROR
int eliminarServicio(Servicio listaServicios[], int size);
```

```
/// @brief Muestra todos los servicios
///
/// @param listaServicios
/// @param size
/// @return OK o ERROR
int mostrarServicios(Servicio listaServicios[], int size);
```



```

/// @brief muestra todos los servicios de una fecha
determinada por el usuario
///
/// @param listaServicios array Servicio
/// @param size int
/// @return int 0 OK o -1 ERROR
int mostrarPuntoE(Servicio listaServicios[], int size);

/// @brief muestra todos los servicios entre marzo y mayo del
2022
///
/// @param listaServicios array Servicio
/// @param size int
/// @return int 0 OK o -1 ERROR
int mostrarI(Servicio listaServicios[], int size);

/// @brief muestra todos los servicios de la especialidad
suspension
///
/// @param listaServicios
/// @param size
/// @return OK o ERROR
int mostrarSuspension(Servicio listaServicios[], int size);

/// @brief Cuenta los diagnosticados totales
///
/// @param listaServicios
/// @param size
/// @return OK o ERROR
int contarDiagnosticados(Servicio listaServicios[], int size);

/// @brief muestra en pantalla los mecanicos
///
/// @param listaMecanicos array Mecanico
/// @return int 0 OK o -1 ERROR
int mostrarMecanicos(Mecanico listaMecanicos[]);

```

```

/// @brief Muestra todos los servicios diagnosticados
///
/// @param listaDiagnosticos
/// @return OK o ERROR
int mostrarDiagnosticos(Diagnostico listaDiagnosticos[]);

/// @brief valida cadena de caracteres
///
/// @param B
/// @return int
int ValidateString (char *B);

/// @brief pide una cadena de caracteres
///
/// @param message
/// @param size
/// @return char
char getString(char message[], int size);

/// @brief lee el stdin hasta que encuentra un '\n' o hasta que
haya
/// copiado en cadena un maximo de 'longitud' -1 caracteres
/// @param pResultado puntero al espacio de memoria donde se
copiara
/// donde se copiara la cadena obtenida
/// @param longitud define el tamaño de la cadena
/// @return retorna 0 (exito) si se obtiene una cadena si no
-1 (error)
int myGets(char* cadena, int longitud);

/// @brief
///
/// @param cadena
/// @return
int esNumerica(char* cadena);

```

```

/// @brief get an integer
/// @param pResultado int
/// @return return 0 SUCCESS or -1 ERROR
int getInt(int* pResultado);

/// @brief get an integer and validate it.
///
/// @param pResultado int
/// @param mensaje char
/// @param mensajeError char
/// @param minimo int
/// @param maximo int
/// @param reintentos int
/// @return return 0 SUCCESS or -1 ERROR
int utn_getNumero(int* pResultado, char* mensaje, char*
mensajeError, int minimo, int maximo, int reintentos);

/// @brief validate a float
/// @return 0 SUCCESS or 1 ERROR
int validarFloat(float* input, char mensaje[]);

/// @brief validate an alphanumeric
///
/// @param variable char
/// @return 0 SUCCESS or -1 ERROR
int validarAlfanumerico(char variable[]);

/// @brief get an alphanumeric
///
/// @param nombre char
/// @param mensaje char
void PedirAlfanumerico(char* nombre[], char mensaje[]);

```

ENLACE AL VIDEO DE LA DEFENSA EN DRIVE

**[https://drive.google.com/file/d/1-HZ4zECMaP0jex83k
yjs9hD1nB04vV4N/view?usp=sharing](https://drive.google.com/file/d/1-HZ4zECMaP0jex83kyjs9hD1nB04vV4N/view?usp=sharing)**