

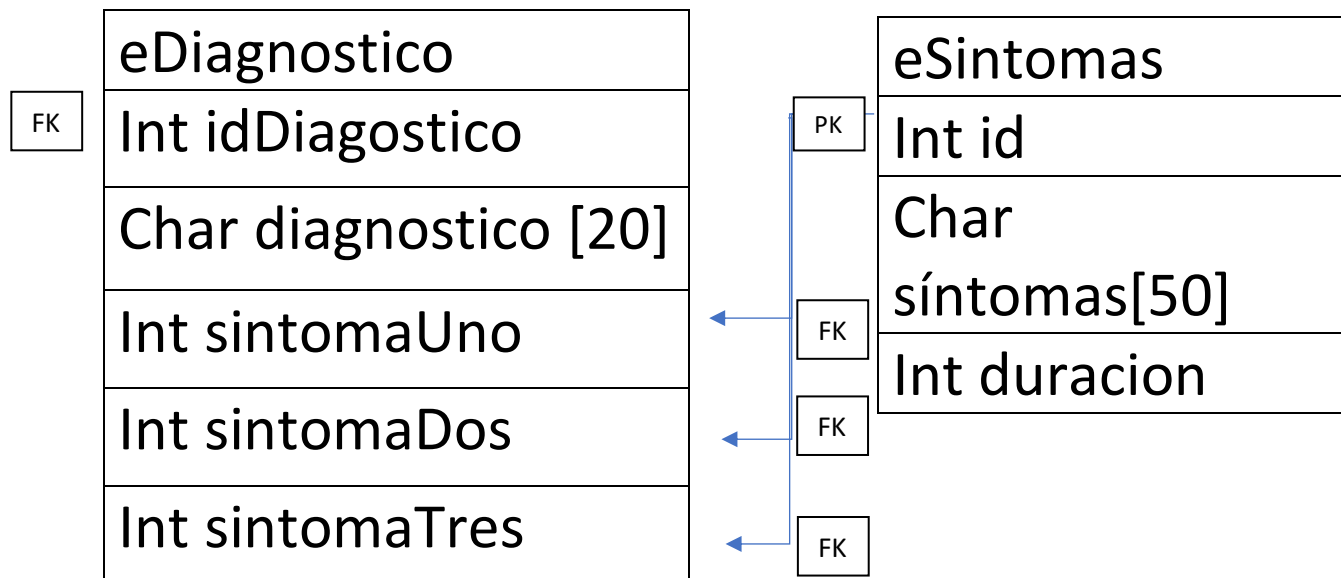
Alumno: Ignacio Javier Medici

Profesor: Scarafilo German

Materia: Laboratorio I

Curso: 1°C

Año: 2022



La primera estructura que agregue se refiere a los síntomas que tiene cada uno de los diagnósticos.

Los datos hardcodeados son:

0	Sin síntomas	0 días
1	Fiebre	5 días
2	Tos	14 días
3	Malestar	7 días
4	Perdida de gusto o olfato	14 días
5	Cansancio	14 días
6	Sudor frio	15 días
7	Dolor torácico	15 días
8	Vomito	15 días

La estructura agregada se relaciona mediante una llave primaria que es un id del síntoma, y lo relaciono con la estructura eDiagnostico con 3 llaves foráneas.

```
#ifndef CONSULTAS_H_
#define CONSULTAS_H_

#include "utn_funciones.h"
#include "fecha.h"
#include "especialidad.h"
#include "medico.h"
#include "diagnostico.h"
#include "medicamento.h"
#include "síntomas.h"

#define OCUPADO 1
#define LIBRE 0

typedef struct{
    int idConsulta;
    char nombrePaciente[50];
    eFecha fecha;
```

```

        int idMedico;
        int idDiagnostico;
        int estado;
    } eConsulta;
    /// @brief Buscar una consulta con el estado LIBRE
    ///
    /// @param listaConsultas
    /// @param tam
    /// @return Retorna la posicion del lugar libre
    int isEmptyArray(eConsulta* listaConsultas, int
tam);
    /// @brief Inicia todos los estados de la lista en
    LIBRE
    ///
    /// @param listaConsultas
    /// @param tam
    /// @return Caso error (-1) Caso correcto (0)
    int initArrayStatus(eConsulta* listaConsultas, int
tam);
    /// @brief Busca y devuelve el id de la consulta
    ///
    /// @param listaConsultas
    /// @param tam
    /// @param id
    /// @return Caso error (-1) Caso correcto (0)
    int BuscarConsultaPorId(eConsulta* listaConsultas,
int tam, int id);
    /// @brief carga una consulta
    ///
    /// @return Caso error (NULL) Caso correcto
    (eConsulta)
    eConsulta cargarConsulta();
    /// @brief lista todas las consultas
    ///
    /// @param listaConsulta
    /// @param tam
    void listarConsultas(eConsulta* listaConsulta, int
tam);
    /// @brief lista todas las consultas con datos del
    medico
    ///
    /// @param listaConsulta
    /// @param tam
    /// @param listaMedico

```

```

/// @param tamM
void listarConsultasConMedico(eConsulta*
listaConsulta, int tam, eMedico* listaMedico, int
tamM);
/// @brief lista todas las consultas con datos del
medico y su especialidad
///
/// @param listaConsulta
/// @param tam
/// @param listaMedico
/// @param tamM
/// @param listaEspecialidades
/// @param tamE
void listarConsultasMedicoEspecialidad(eConsulta*
listaConsulta,int tam,eMedico* listaMedico,int
tamM,eEspecialidad* listaEspecialidades,int tamE);
/// @brief lista una consulta con todos sus datos
///
/// @param unaConsulta
/// @param unaEspecialidad
/// @param unMedico
/// @param unDiagnostico
/// @param sintomas
void consulta_MostrarUna(eConsulta unaConsulta,
eEspecialidad unaEspecialidad, eMedico unMedico,
eDiagnostico unDiagnostico,eSintomas sintomas);
/// @brief lista todas las consultas con todos los
datos
///
/// @param listaConsultas
/// @param tam
/// @param listaMedicos
/// @param tamM
/// @param listaEspecialidades
/// @param tamE
/// @param listaDiagnosticos
/// @param tamD
/// @param listaSintomas
/// @param tamS
/// @return Caso error (-1) Caso correcto (0)
int consulta_MostrarTodas(eConsulta*
listaConsultas, int tam, eMedico* listaMedicos,
int tamM,eEspecialidad* listaEspecialidades,int

```

```

tamE,eDiagnostico* listaDiagnosticos, int
tamD,eSintomas* listaSintomas,int tamS);
/// @brief Da de alta una consulta dandole su
propio ID
///
/// @param listaConsulta
/// @param tam
/// @param listaMedico
/// @param tamM
/// @param listaEspecialidades
/// @param tamE
/// @return Caso error (-1) Caso correcto (0)
int consulta_Alta(eConsulta* listaConsulta, int
tam, eMedico* listaMedico, int tamM,
eEspecialidad* listaEspecialidades, int tamE);
/// @brief Estructura los datos que se desean
modificar
///
/// @param unaConsulta
/// @return Caso error (-1) Caso correcto (0)
eConsulta modificar_Conulta(eConsulta
unaConsulta);
/// @brief Modifica los datos dados en el alta
///
/// @param listaConsultas
/// @param tam
/// @param listaMedicos
/// @param tamM
/// @param listaEspecialidades
/// @param tamE
/// @param listaDiagnosticos
/// @param tamD
/// @param listaSintomas
/// @param tamS
/// @param idConsulta
/// @return Caso error (-1) Caso correcto (0)
int modificarConsultas(eConsulta*
listaConsultas,int tam,eMedico* listaMedicos,int
tamM,eEspecialidad* listaEspecialidades,int
tamE,eDiagnostico* listaDiagnosticos,int
tamD,eSintomas* listaSintomas,int tamS, int
idConsulta);
/// @brief Da de baja la consulta cambiando su
estado en LIBRE

```

```

///
/// @param listaConsultas
/// @param tam
/// @param listaMedicos
/// @param tamM
/// @param listaEspecialidades
/// @param tamE
/// @param listaDiagnosticos
/// @param tamD
/// @param listaSintomas
/// @param tamS
/// @param idConsulta
/// @return Caso error (-1) Caso correcto (0)
int darDeBajaConsulta(eConsulta*
listaConsultas,int tam,eMedico* listaMedicos,int
tamM,eEspecialidad* listaEspecialidades,int
tamE,eDiagnostico* listaDiagnosticos,int
tamD,eSintomas* listaSintomas,int tamS, int
idConsulta);
/// @brief Carga un diagnostico a una consulta
///
/// @param listaConsulta
/// @param tam
/// @param listaDiagnostico
/// @param tamD
/// @param listaMedicos
/// @param tamM
/// @param idConsulta
/// @return Caso error (-1) Caso correcto (0)
int cargarDiagnostico(eConsulta* listaConsulta,
int tam, eDiagnostico* listaDiagnostico,int
tamD,eMedico* listaMedicos, int tamM, int
idConsulta);
/// @brief ordena la lista de consultas por ID de
menor a mayor
///
/// @param listaConsultas
/// @param tam
/// @return Caso error (-1) Caso correcto (0)
int ordenarConsultasPorId(eConsulta*
listaConsultas, int tam);
/// @brief lista los medicos que tienen consultas
diagnosticadas
///

```

```

/// @param listaConsulta
/// @param tam
/// @param listaMedicos
/// @param tamM
/// @param listaDiagnostico
/// @param tamD
/// @param listaEspecialidad
/// @param tamE
/// @return Caso error (-1) Caso correcto (0)
int
listarMedicoConConsultaDiagnosticadas(eConsulta*
listaConsulta, int tam, eMedico* listaMedicos, int
tamM, eDiagnostico* listaDiagnostico, int
tamD, eEspecialidad* listaEspecialidad, int tamE);
/// @brief lista las consultas con las fechas mas
recientes
///
/// @param listaConsultas
/// @param tam
/// @return Caso error (-1) Caso correcto (0)
int listarConsultaFechaMasReciente(eConsulta*
listaConsultas, int tam);
/// @brief lista solo las consultas diagnosticadas
///
/// @param listaConsulta
/// @param tam
/// @param listaDiagnosticos
/// @param tamD
/// @return Caso error (-1) Caso correcto (0)
int listarConsultasDiagnosticadas(eConsulta*
listaConsulta, int tam, eDiagnostico*
listaDiagnosticos, int tamD);
/// @brief lista solo las consultas diagnosticadas
con COVID-19
///
/// @param listaConsulta
/// @param tam
/// @param listaDiagnosticos
/// @param tamD
/// @return Caso error (-1) Caso correcto (0)
int listarDiagnosticoCovid19(eConsulta*
listaConsulta, int tam, eDiagnostico*
listaDiagnosticos, int tamD);

```

```

/// @brief lista las consultas y especialidades
alfabeticamente
///
/// @param listaConsulta
/// @param tam
/// @param listaMedico
/// @param tamM
/// @param listaEspecialidades
/// @param tamE
/// @return Caso error (-1) Caso correcto (0)
int listarConsultasEspecialidadAlfa(eConsulta*
listaConsulta , int tam ,eMedico* listaMedico ,int
tamM ,eEspecialidad* listaEspecialidades, int
tamE);
/// @brief lista las consultas de abril hasta
julio
///
/// @param listaConsultas
/// @param tam
/// @param listaMedicos
/// @param tamM
/// @param listaDiagnosticos
/// @param tamD
/// @param listadoEspecialidades
/// @param tamE
/// @param idEspecialidad
/// @param idMedico
/// @return Caso error (-1) Caso correcto (0)
int listarConsultasAbrilJulio(eConsulta*
listaConsultas,int tam,eMedico* listaMedicos,int
tamM,eDiagnostico* listaDiagnosticos, int tamD,
eEspecialidad* listadoEspecialidades, int tamE,
int idEspecialidad, int idMedico);
/// @brief lista las consultas de abril a julio de
determinadas especialidades
///
/// @param listaConsultas
/// @param tam
/// @param listaMedicos
/// @param tamM
/// @param listaDiagnosticos
/// @param tamD
/// @param listadoEspecialidades
/// @param tamE

```



```

/// @return Caso error (-1) Caso correcto (0)
int
listarConsultasEspecialidadDeterminada (eConsulta*
listaConsultas, int tam, eMedico* listaMedicos, int
tamM, eDiagnostico* listaDiagnosticos, int tamD,
eEspecialidad* listadoEspecialidades, int tamE);
/// @brief lista las enfermedades menos
diagnosticadas por los medicos
///
/// @param listaConsultas
/// @param tam
/// @param listaDiagnosticos
/// @param tamD
/// @return Caso error (-1) Caso correcto (0)
int enfermedadMenosDiagnosticada (eConsulta*
listaConsultas, int tam, eDiagnostico*
listaDiagnosticos, int tamD);
/// @brief lista la especialidad mas estudiada por
los medicos
///
/// @param listaMedicos
/// @param tamM
/// @param listaEspecialidades
/// @param tamE
/// @return Caso error (-1) Caso correcto (0)
int especialidadMasEstudiada (eMedico*
listaMedicos, int tamM, eEspecialidad*
listaEspecialidades, int tamE);
/// @brief lista los sintomas mas comunes de cada
diagnostico
///
/// @param listaSintomas
/// @param tam
/// @return Caso error (-1) Caso correcto (0)
int SintomasMasComunes (eSintomas* listaSintomas,
int tam);
/// @brief Da 5 consultas con todos los datos para
facilitar el desarrollo de la aplicacion
///
/// @param listaConsulta
/// @param listaForzada
/// @param tam
void AltaForzada (eConsulta* listaConsulta,
eConsulta* listaForzada, int tam);

```

```

#endif /* CONSULTAS_H_ */
#ifndef ESPECIALIDAD_H_
#define ESPECIALIDAD_H_
#include "utn_funciones.h"

#define TAM_E 3

typedef struct{
    int id;
    char especialidad[20];
    int idDiagnostico;
}eEspecialidad;

/// @brief Lista las especialidades ingresadas
///
/// @param listaEspecialidad eEspecialidad*
/// @param tam int
void listarEspecialidades(eEspecialidad* ,int );
/// @brief Busca la posicion del id buscado por
parametro
///
/// @param listaEspecialidad eEspecialidad*
/// @param tam int
/// @param idEspecialidad int
/// @return retorno la posicion buscada
int buscarIdEspecialidad(eEspecialidad* ,int
,int);

#endif /* ESPECIALIDAD_H_ */

#ifndef MEDICO_H_
#define MEDICO_H_
#include "utn_funciones.h"
#include "especialidad.h"

#define TAM_M 5

typedef struct{
    int id;
    char nombreMedico[50];

```

```

        int idEspecialidad;
    } eMedico;

    /// @brief Lista los medicos ingresados
    ///
    /// @param lista eMedico*
    /// @param tamM int
    /// @param listaEspecialidad eEspecialidad*
    /// @param tamE int
    void listarMedicos(eMedico* lista, int tamM,
    eEspecialidad* listaEspecialidad, int tamE);
    /// @brief Busca la posicion del id buscado por
    parametro
    ///
    /// @param lista
    /// @param tamM
    /// @param idMedico
    /// @return retorno la posicion del medico en la
    lista
    int idMedico(eMedico* lista, int tamM, int
    idMedico);

    #endif /* MEDICO_H */
    #ifndef SINTOMAS_H
    #define SINTOMAS_H

    #include "utn_funciones.h"

    #define TAM_S 10

    typedef struct{
        int id;
        char sintomas[50];
        int duracion;
    } eSintomas;

    /// @brief lista los sintomas ingresados
    ///
    /// @param listaSintomas eSintomas*
    /// @param tamS int
    void listarSintomas(eSintomas* listaSintomas, int
    tamS);

```

```

/// @brief Busca la posicion del id buscado por
parametro
///
/// @param listaSintomas eSintomas*
/// @param tamM int
/// @param idsintoma int
/// @return retorno la posicion del sintoma en la
lista
int buscarSintomaPorId(eSintomas* listaSintomas,
int tamS, int idSintoma);

#endif /* SINTOMAS_H_ */
#ifndef MEDICAMENTO_H_
#define MEDICAMENTO_H_
#include "utn_funciones.h"

#define TAM_ME 3
typedef struct{
    int id;
    char medicamento[20];
    int tiempoDeUso;
} eMedicamento;

/// @brief Lista los medicamentos ingresados
///
/// @param listaMedicamento eMedicamento*
/// @param tamMe int
void listarMedicamento(eMedicamento*
listaMedicamento, int tamMe);

/// @brief Busca la posicion del id buscado por
parametro
///
/// @param listaMedicamento eMedicamento*
/// @param tamMe int
/// @param idMedicamento int
/// @return retorno la posicion del sintoma en la
lista
int buscarMedicamentoPorId(eMedicamento*
listaMedicamento, int tamMe, int idMedicamento);

#endif /* MEDICAMENTO_H_ */

```

```

#ifndef DIAGNOSTICO_H_
#define DIAGNOSTICO_H_
#include "utn_funciones.h"

#define TAM_D 4

typedef struct{
    int idDiag;
    char diagnositco[20];
    int sintomaUno;
    int sintomaDos;
    int sintomaTres;
} eDiagnostico;

/// @brief lista los diagnosticos ingresados
///
/// @param listaDiagnosticos eDiagnostico*
/// @param tamD int
void listarDiagnostico(eDiagnostico*
listaDiagnosticos,int tamD);

/// @brief Busca la posicion del id buscado por
parametro
///
/// @param listaDiagnosticos eDiagnostico*
/// @param tam int
/// @param idDiagnostico int
/// @return retorno la posicion del sintoma en la
lista
int buscarIdDiagnostico(eDiagnostico*
listaDiagnosticos,int tam,int idDiagnostico);

#endif /* DIAGNOSTICO_H_ */

```