



Universidad Autónoma
Gabriel René Moreno

INF312 - BASE DE DATOS I (SA)

UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO

FACULTAD DE INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN Y

TELECOMUNICACIONES

BASE DE DATOS HOSPITALARIA

Leonardo H. Añez Vladimirovna
toboroichi98@outlook.com

Mauricio Elian Delgadillo
elianklk@gmail.com

10 de Diciembre 2018

El presente proyecto esta disponible en github, como codigo libre, junto con todo el material de este informe: <https://github.com/toborochi/Personal-Projects/tree/master/DB>

GitHub



Introducción

El proyecto abordara el problema de registro de un sistema hospitalario a travez de las tres etapas de diseño: Conceptual, Lógico y Físico.

Planteamiento

Esta base de Datos contendra datos demograficos del paciente, informacion de facturacion y seguros, asi como su historial medico. Permite realizar un seguimiento de los registros tanto de pacientes hospitalizados como ambulatorios. Se podra verificar la habitacion que ocupa actualmente un paciente hospitalizado, y los medicos podran identificar facilmente a sus pacientes, y determinar que enfermeras estan asignadas para ayudarlos en una determinada sala.

La base de datos requerirá muchos tipos diferentes de datos. Toda la información de contacto de los pacientes, como la dirección, el número de teléfono y el correo electrónico, será necesaria, así como toda la información de prescripción. Toda la información de la cita se registrará y guardará para que se extraiga cuando sea necesario. Si un paciente es un paciente interno, entonces se registrará la habitación en la que reside actualmente. También se almacenará información de facturación y seguro del paciente. También habrá datos almacenados para cada médico en el hospital, describiendo su especialidad, y supervisores. También habrá información sobre las enfermeras y la sala de las que están a cargo durante sus fechas y horarios de turnos especificados. Las enfermeras son responsables del cuidado de todos los pacientes que residen en sus salas y habitaciones designadas, y tendrán que informar a su supervisor de vez en cuando.

Para facilitar el proceso de elaboración del Diagrama, tendremos las siguientes asunciones:

Asunciones

- Todos los pacientes deben tener una póliza de seguro para ser tratados en el hospital.
- Un paciente puede someterse a muchos tratamientos en el hospital.
- Un paciente puede o no recibir recetas de los médicos en el hospital.
- Todas las enfermeras y médicos tienen supervisores.
- Los pacientes recibirán cada costo de tratamiento bajo pagos individuales.
- Los pacientes pueden realizar los pagos en cuotas múltiples.
- Las enfermeras son asignadas a una sala durante su turno.
- Los pacientes hospitalizados se registran en una habitación dentro de una sala apropiada.
- Todos los médicos del hospital se especializan en un campo médico determinado.

Diseño Conceptual

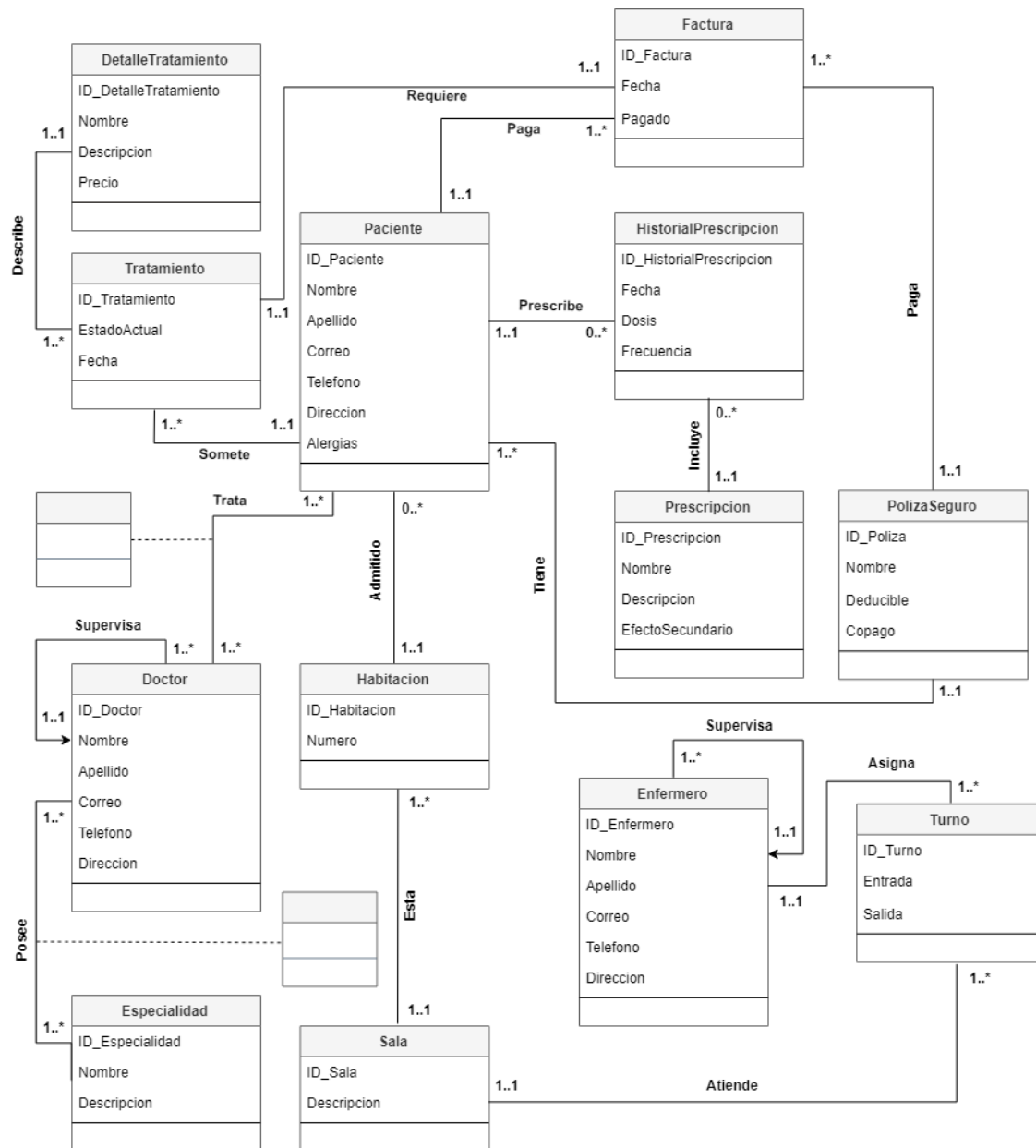


Figura 1: M.C. de la Base de Datos Hospitalaria

Diseño Lógico

En esta etapa se presenta el código realizado en Microsoft SQL Server. A continuación las implementaciones de las tablas:

Diseño Físico

Detalle Tratamiento

```
1 create table DetalleTratamiento(  
2     ID_DetalleTratamiento int not null,  
3     Nombre varchar(50) not null,  
4     Descripcion varchar(200) not null,  
5     Precio float not null,  
6     primary key(ID_DetalleTratamiento));
```

Sala

```
1 create table Sala(  
2     ID_Sala int not null,  
3     Descripcion varchar(200) not null,  
4     primary key(ID_Sala));
```

Habitacion

```
1 create table Habitacion(  
2     ID_Habitacion int not null,  
3     Numero int not null,  
4     SalaS int not null,  
5     primary key(ID_Habitacion),  
6     Foreign key(SalaS) references Sala(ID_Sala)  
7     On delete cascade  
8     On update cascade,);
```

Poliza Seguro

```
1 create table PolizaSeguro(  
2     ID_Poliza int not null,  
3     Nombre varchar(50) not null,  
4     Deducible float not null,  
5     Copago float not null,  
6     primary key(ID_Poliza),);
```

Especialidad

```
1 create table Especialidad(  
2     ID_Especialidad int not null,  
3     Nombre varchar(50) not null,  
4     Descripcion varchar(200) not null,  
5     primary key(ID_Especialidad),);
```

Paciente

```
1 create table Paciente(  
2     ID_Paciente int not null,  
3     Nombre varchar(50) not null,  
4     Apellido varchar(50) not null,  
5     Correo varchar(50),  
6     Telefono int ,  
7     Direccion varchar(50) not null,  
8     Alergia varchar(50) not null,  
9     ID_PolizaS int not null,  
10    ID_HabitacionS int not null  
11    primary key(ID_Paciente),  
12    foreign key(ID_PolizaS) references PolizaSeguro(ID_Poliza)  
13    On delete cascade  
14    On update cascade,  
15    foreign key(ID_HabitacionS) references Habitacion(ID_Habitacion)  
16    On delete cascade  
17    On update cascade);
```

Factura

```
1 create table Factura(  
2     ID_Factura int not null,  
3     Fecha date not null,  
4     Pagado float not null,  
5     ID_PacienteF int not null,  
6     ID_PolizaF int not null,  
7     Primary key(ID_Factura),  
8     foreign key(ID_PacienteF) references Paciente(ID_Paciente)  
9     on delete no action  
10    On update no action,  
11    foreign key(ID_PolizaF) references PolizaSeguro(ID_Poliza)  
12    On delete no action  
13    On update no action);
```

Prescripcion

```
1 create table Prescripcion(  
2     ID_Prescripcion int not null,  
3     Nombre varchar(50) not null,  
4     Descripcion varchar(200) not null,  
5     EfectoSecundario varchar(200) not null,  
6     primary key(ID_Prescripcion),);
```

Tratamiento

```
1 create table Tratamiento(  
2     ID_Tratamiento int not null,  
3     EstadoActual varchar(50) not null,  
4     Fecha varchar(8) not null,  
5     ID_PacienteD int not null,  
6     ID_FacturaD int not null,  
7     ID_DetalleTratamientoD int not null,  
8     primary key(ID_Tratamiento),  
9     Foreign key(ID_PacienteD) references Paciente(ID_Paciente)  
10    On delete cascade  
11    On update cascade,  
12    Foreign key(ID_FacturaD) references Factura(ID_Factura)  
13    On delete cascade  
14    On update cascade,  
15    Foreign key(ID_DetalleTratamientoD) references DetalleTratamiento(  
16    ID_DetalleTratamiento)  
17    On delete cascade  
18    On update cascade);
```

Enfermero

```
1 create table Enfermero(  
2     ID_Enfermero int not null,  
3     Nombre varchar(50) not null,  
4     Apellido varchar(50) not null,  
5     Correo varchar(50),  
6     Telefono int ,  
7     Direccion varchar(50) not null,  
8     ID_EnfermeroS int null,  
9     primary key(ID_Enfermero),  
10    );  
11  
12 alter table Enfermero  
13 add foreign key (ID_EnfermeroS) references Enfermero(ID_Enfermero)  
14 on update no action  
15 on delete no action;
```

Trata

```
1 create table Trata(  
2     ID_DoctorT int not null,  
3     ID_PacienteT int not null,  
4     primary key (ID_DoctorT, ID_PacienteT),  
5     Foreign key (ID_DoctorT) references Doctor(ID_Doctor)  
6     on update no action  
7     on delete no action,  
8     Foreign key (ID_PacienteT) references Paciente(ID_Paciente)  
9     on update no action  
10    on delete no action);
```

Doctor

```
1 create table Doctor(  
2     ID_Doctor int not null,  
3     Nombre varchar(50) not null,  
4     Apellido varchar(50) not null,  
5     Correo varchar(50),  
6     Telefono int ,  
7     Direccion varchar(50) not null,  
8     ID_DoctorS int ,  
9     primary key(ID_Doctor),  
10 );  
11  
12  
13 alter table Doctor  
14 add foreign key (ID_DoctorS) references Doctor(ID_Doctor)  
15 on update no action  
16 on delete no action;
```

Posee

```
1 create table Posee(  
2     ID_DoctorP int not null,  
3     ID_EspecialidadP int not null,  
4     primary key (ID_DoctorP, ID_EspecialidadP),  
5     Foreign key (ID_DoctorP) references Doctor(ID_Doctor)  
6     on update no action  
7     on delete no action,  
8     Foreign key (ID_EspecialidadP) references Especialidad(ID_Especialidad)  
9     on update no action  
10    on delete no action);
```

Turno

```
1 create table Turno(  
2     ID_Turno int not null,  
3     Entrada date not null,  
4     Salida varchar(8) not null,  
5     ID_SalaT int not null,  
6     ID_EnfermeroT int not null,  
7     primary key(ID_Turno),  
8     Foreign key(ID_SalaT) references Sala(ID_Sala)  
9     On delete cascade  
10    On update cascade,  
11    Foreign key(ID_EnfermeroT) references Enfermero(ID_Enfermero)  
12    On delete cascade  
13    On update cascade);
```


Historial Prescripcion

```
1 create table HistorialPrescripcion(  
2     ID_HistorialPrescripcion int not null,  
3     Fecha date not null,  
4     Dosis int not null,  
5     Frecuencia int not null,  
6     ID_PacienteH int not null,  
7     ID_PrescripcionH int not null,  
8     primary key(ID_HistorialPrescripcion),  
9     Foreign key(ID_PacienteH) references Paciente(ID_Paciente)  
10    On delete cascade  
11    On update cascade,  
12    Foreign key(ID_PrescripcionH) references Prescripcion(ID_Prescripcion)  
13    On delete cascade  
14    On update cascade);
```