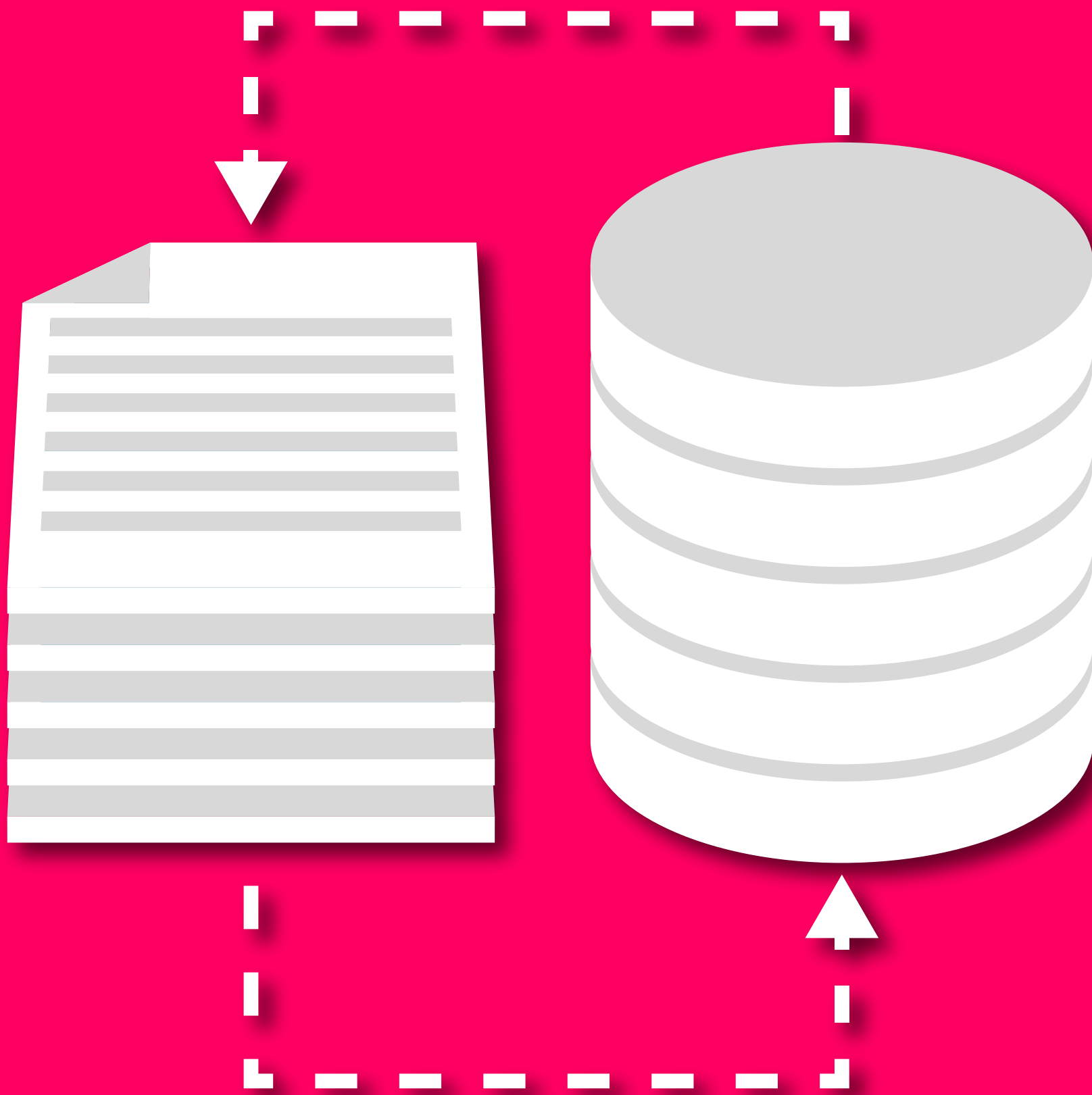




El futuro digital  
es de todos

MinTIC



# transacciones en la base de datos

## Taller 01



Universidad de Caldas

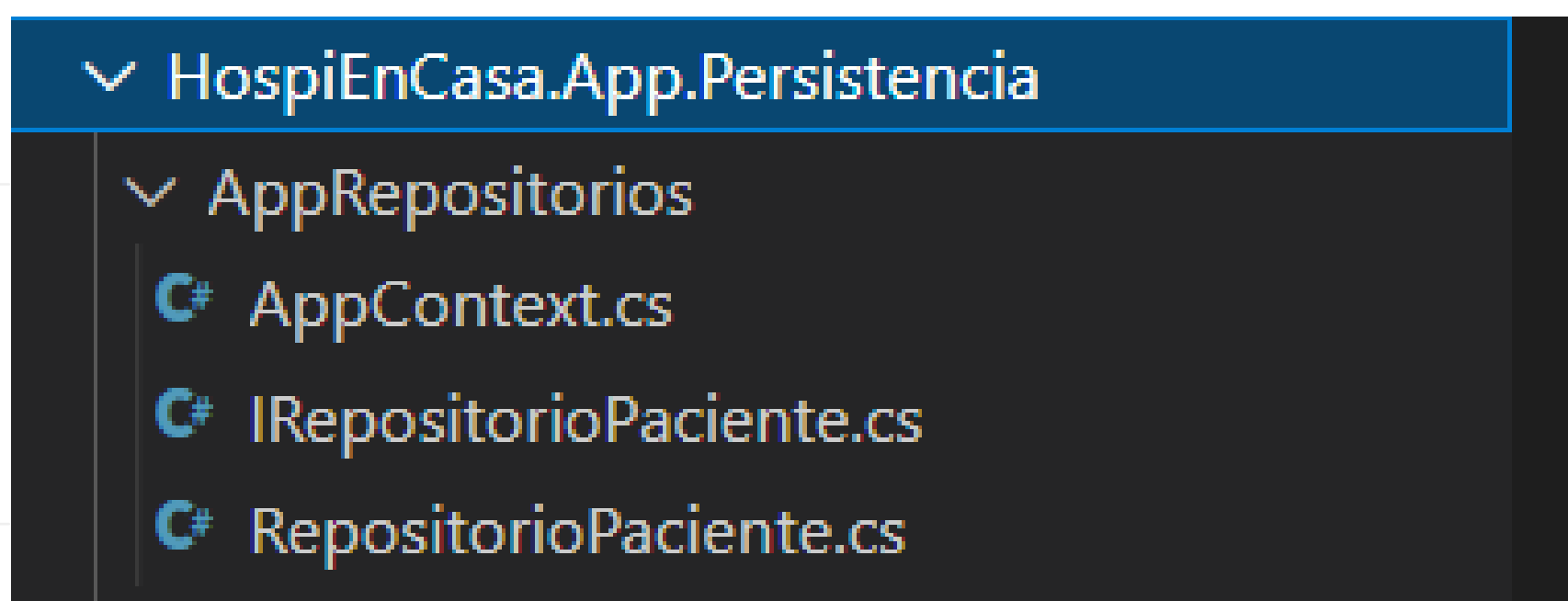
# Objetivo

Este taller tiene como objetivo aplicar los conceptos sobre transacciones en entidades utilizando repositorios. El taller se divide en dos partes: la primera parte, consiste en realizar las transacciones básicas similares a las explicadas para la entidad Paciente en el video práctico. En la segunda parte se explica el manejo de entidades relacionadas con otras entidades.

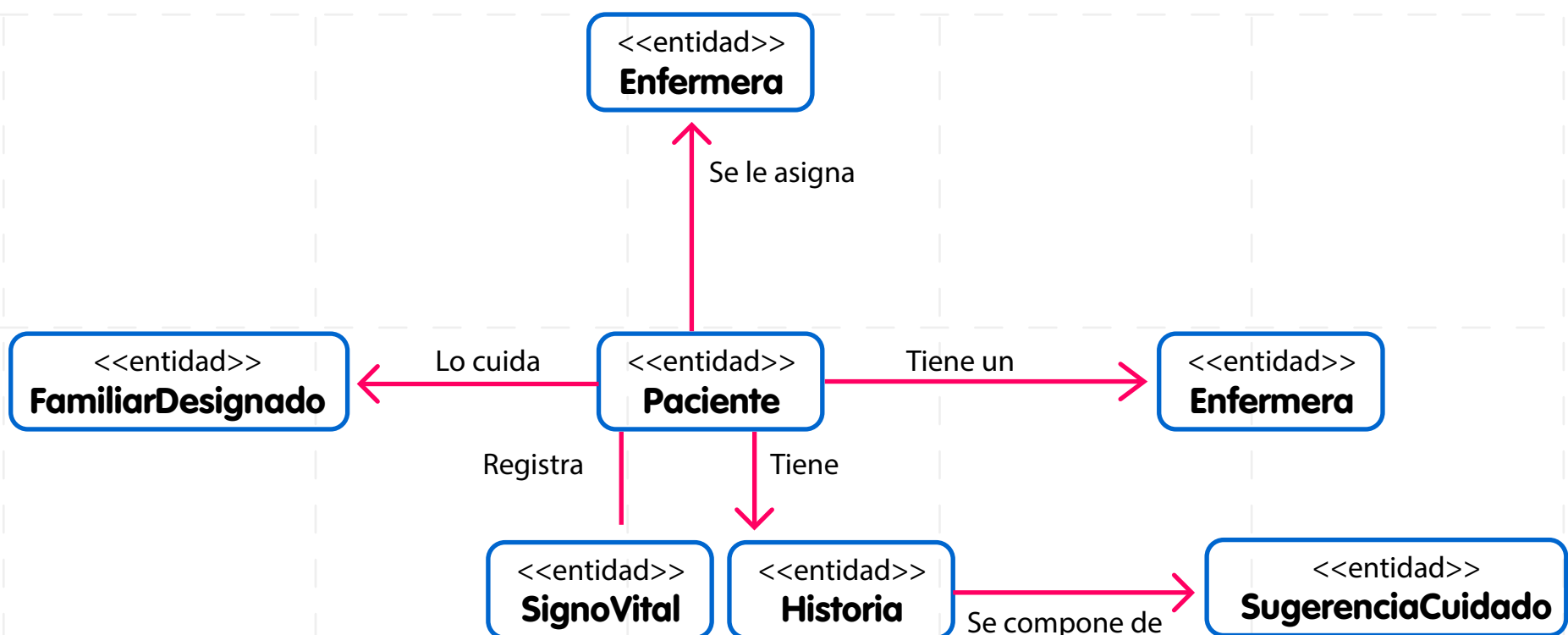


## Primera Parte

En el proyecto Hospital en Casa hemos implementado el contexto para todas las entidades y el repositorio para la entidad paciente.



Sin embargo, de acuerdo con nuestro modelo de clases de entidad además de Paciente se tienen las siguientes entidades: Enfermera, Médico y Familiar Designado.

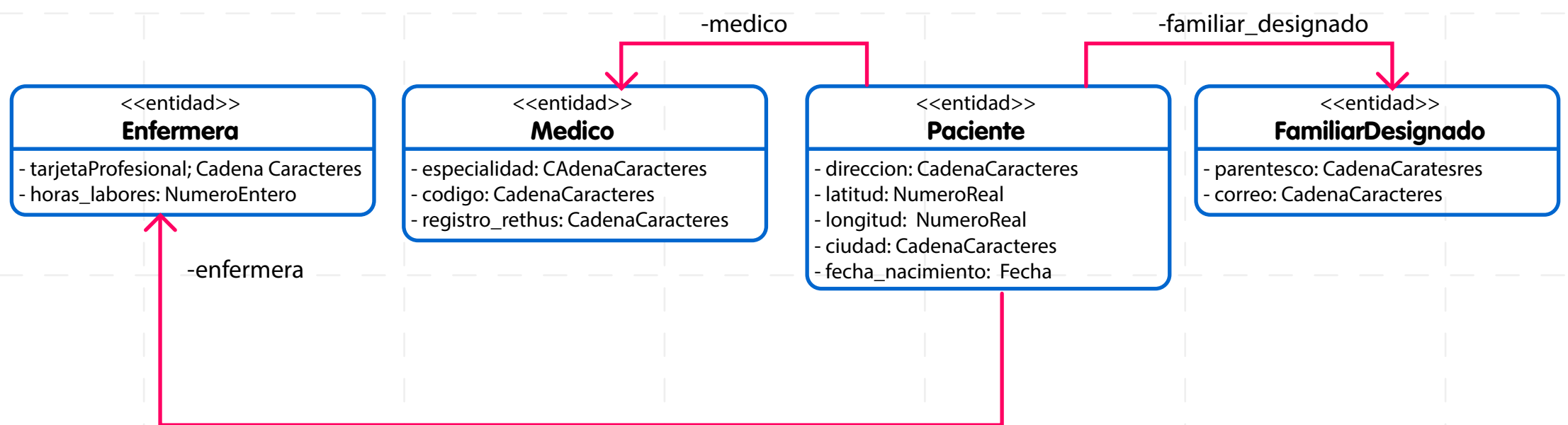


## ¿Qué se debe hacer?

- Realizar la implementación de los repositorios para las entidades Faltantes (Interfaz y repositorio)
- Implementar métodos de prueba en el proyecto de tipo consola.

## Segunda Parte

Hasta ahora hemos implementado las transacciones más comunes para cada una de las entidades del dominio del problema. Sin embargo, existen otra serie de transacciones que son específicas del dominio del problema y que también deben implementarse. Analicemos las siguientes entidades y sus relaciones.



De acuerdo con el modelo anterior (parcial), En Paciente, no solo se deben manejar sus datos generales (nombre, apellidos, etc.). Tambien se debe permitir:

- La asignación de un médico
- La asignación de una enfermera
- La asignación de un familiar asignado al paciente

Para implementarlas debemos tener en cuenta que serían funciones adicionales para el repositorio de pacientes, por lo tanto, deben adicionarse primero en la interfaz y luego implementarse en el repositorio.

Lo primero que debe hacerse es definir **la firma** del método que se va a implementar en la interfaz que define el repositorio (Contrato). En el siguiente ejemplo está definida la firma para el método que permite asignar médico.

```

namespace HospiEnCasa.App.Persistencia
{
    public interface IRepositoryPaciente
    {
        IEnumerable<Paciente> GetAllPacientes();
        Paciente AddPaciente(Paciente paciente);
        Paciente UpdatePaciente(Paciente paciente);
        void DeletePaciente(int idPaciente);
        Paciente GetPaciente(int idPaciente);
        Medico AsignarMedico(int idPaciente, int idMedico);
    }
}
  
```

Este método recibe como parámetros los identificadores del paciente y del médico que se le asignará y retorna los datos del médico asignado si se logró la asignación o null en caso contrario. A continuación, se muestra la implementación del método, lo primero que se hace es buscar al paciente y en caso de encontrarlo se debe buscar al médico, si este también existe se le asigna al paciente y se actualiza la base de datos.

```
namespace HospiEnCasa.App.Persistencia
{
    // Código ya implementado
    Medico IRepositoryPaciente.AsignarMedico(int idPaciente, int idMedico)
    {
        var pacienteEncontrado = _appContext.Pacientes.FirstOrDefault(p => p.Id == idPaciente);
        if (pacienteEncontrado != null)
        {
            var medicoEncontrado = _appContext.Medicos.FirstOrDefault(m => m.Id == idMedico);
            if (medicoEncontrado != null)
            {
                pacienteEncontrado.Medico = medicoEncontrado;
                _appContext.SaveChanges();
            }
            return medicoEncontrado;
        }
        return null;
    }
}
```

Solo queda implementar un método en el main para probar lo que hicimos.

```
private static void AsignarMedico()
{
    var medico = _repoPaciente.AsignarMedico(1,2);
    Console.WriteLine(medico.Nombre+" "+medico.Apellidos);
}
```

Y en el Main

```
static void Main(string[] args)
{
    Console.WriteLine("Hello World EF!");
    //AddPaciente();
    //BuscarPaciente(1);
    AsignarMedico();
}
```

# ¿Qué sucede en la base de datos?

Si bien desde el punto de vista de la aplicación, el modelo esta dividido en entidades que heredan de la clase persona, si consultamos la base de datos encontramos que solo se ha creado una sola tabla Personas donde se unieron todos los atributos de todas las entidades.

```
CREATE TABLE [dbo].[Personas](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Nombre] [nvarchar](max) NULL,
    [Apellidos] [nvarchar](max) NULL,
    [NumeroTelefono] [nvarchar](max) NULL,
    [Genero] [int] NOT NULL,
    [Ciudad] [nvarchar](max) NULL,
    [Codigo] [nvarchar](max) NULL,
    [Correo] [nvarchar](max) NULL,
    [Direccion] [nvarchar](max) NULL,
    [Discriminator] [nvarchar](max) NOT NULL,
    [Enfermerald] [int] NULL,
    [Especialidad] [nvarchar](max) NULL,
    [FamiliarId] [int] NULL,
    [FechaNacimiento] [datetime2](7) NULL,
    [HistorialId] [int] NULL,
    [HorasLaborales] [int] NULL,
    [Latitud] [real] NULL,
    [Longitud] [real] NULL,
    [Medicoid] [int] NULL,
    [Parentesco] [nvarchar](max) NULL,
    [RegistroRethus] [nvarchar](max) NULL,
    [TarjetaProfesional] [nvarchar](max) NULL,
)
```

Los campos resaltados con amarillo se denominan llaves y se tienen de dos tipos:

**Llave primaria:** por ejemplo `[Id] [int] IDENTITY(1,1) NOT NULL`, se llama clave primaria o clave principal a un campo o a una combinación de campos que identifica de forma única a cada fila de una tabla.

**Llave secundaria:** por ejemplo:

[Enfermerald] [int] NULL,  
[FamiliarId] [int] NULL,  
[Medicoid] [int] NULL,

Una clave foránea o clave ajena (o Foreign Key FK) es una referencia entre dos tablas, por ejemplo, en un paciente se asigna en este campo el identificador del médico asignado. Una vez ejecutemos el programa, se puede verificar que efectivamente se ha actualizado el paciente con la llave foránea del médico asignado.

Id	Nombre	Apellidos	NumeroTelefono	Direccion	Discriminator	Longitud	MedicoID
1	Nicolay	Perez	3001645	Calle 4 No 7-4	Paciente	5.07062	2
2	Nataly	Arruble	300400	Calle 9 No 9-10	Medico	NULL	NULL

## ¿Qué se debe hacer?

Implementar los métodos para el repositorio que permitan

- Asignar Enfermera
- Asignar Familiar designado
- Asignar Historia





Universidad de Caldas