



GISS - Gestão Integrada de Serviços de Saúde

RELATÓRIO

Miguel Gregório, 35388;
Tiago Canhão, 35660;
João Lopes, 37952;

PL3, Base De Dados
2017/2018

1.Introdução

Com este projeto, pretende-se criar um software de Gestão Integrada de Serviços de Saúde. Pretende-se também que todos os serviços de saúde estejam interligados com a respetiva área clinica, corpo clinico, etc. Também permitirá marcar consultas aos utentes, entre outras funcionalidades que serão descritas ao longo deste relatório.

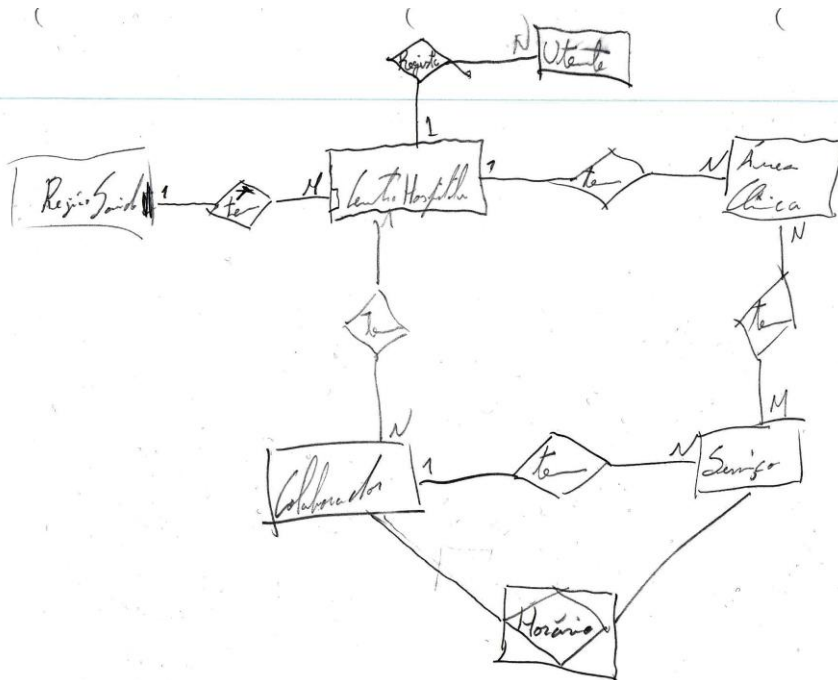
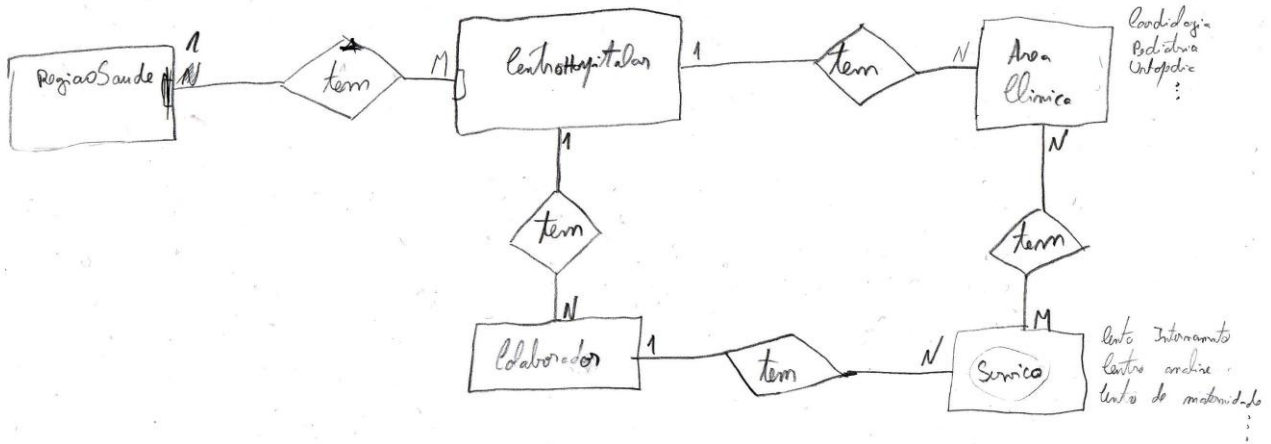
Para a concretização deste projeto foram utilizados os seguintes programas:

- SQL Server Management Studio : foi usado para criar a base de dados, executar os scripts para criar tabelas, e adicionar dados iniciais a essas tabelas;
- NetBeans IDE : foi usado para criar a interface gráfica das aplicações. Foi onde foram declaradas as instruções SQL bem como as respetivas restrições.

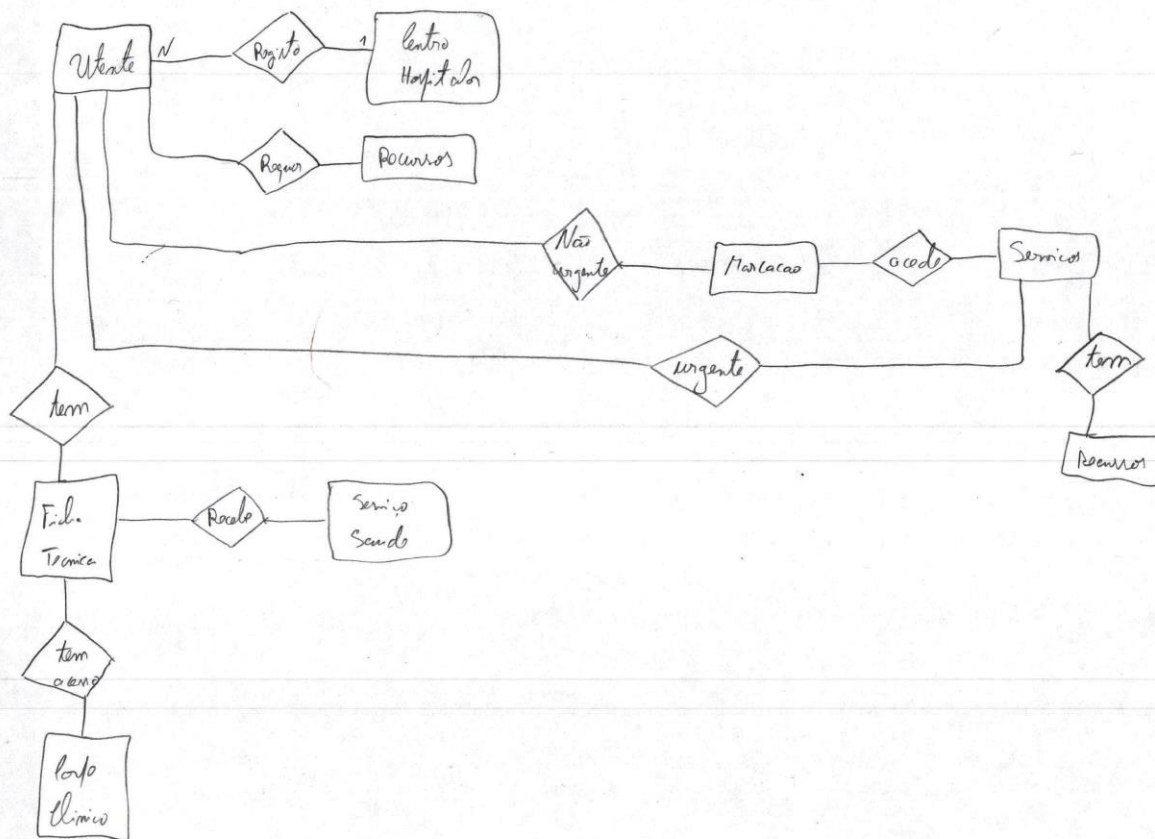
2.Modelo de Dados

2.1 - 1ª Fase da modelação de dados

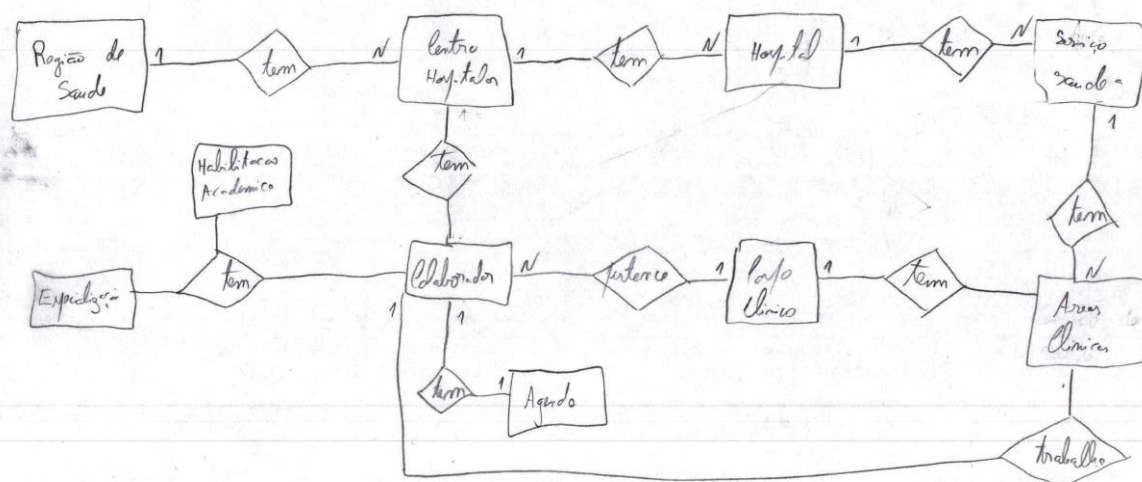
27/04/2018



2.2 – 2ª Fase da modelação de dados



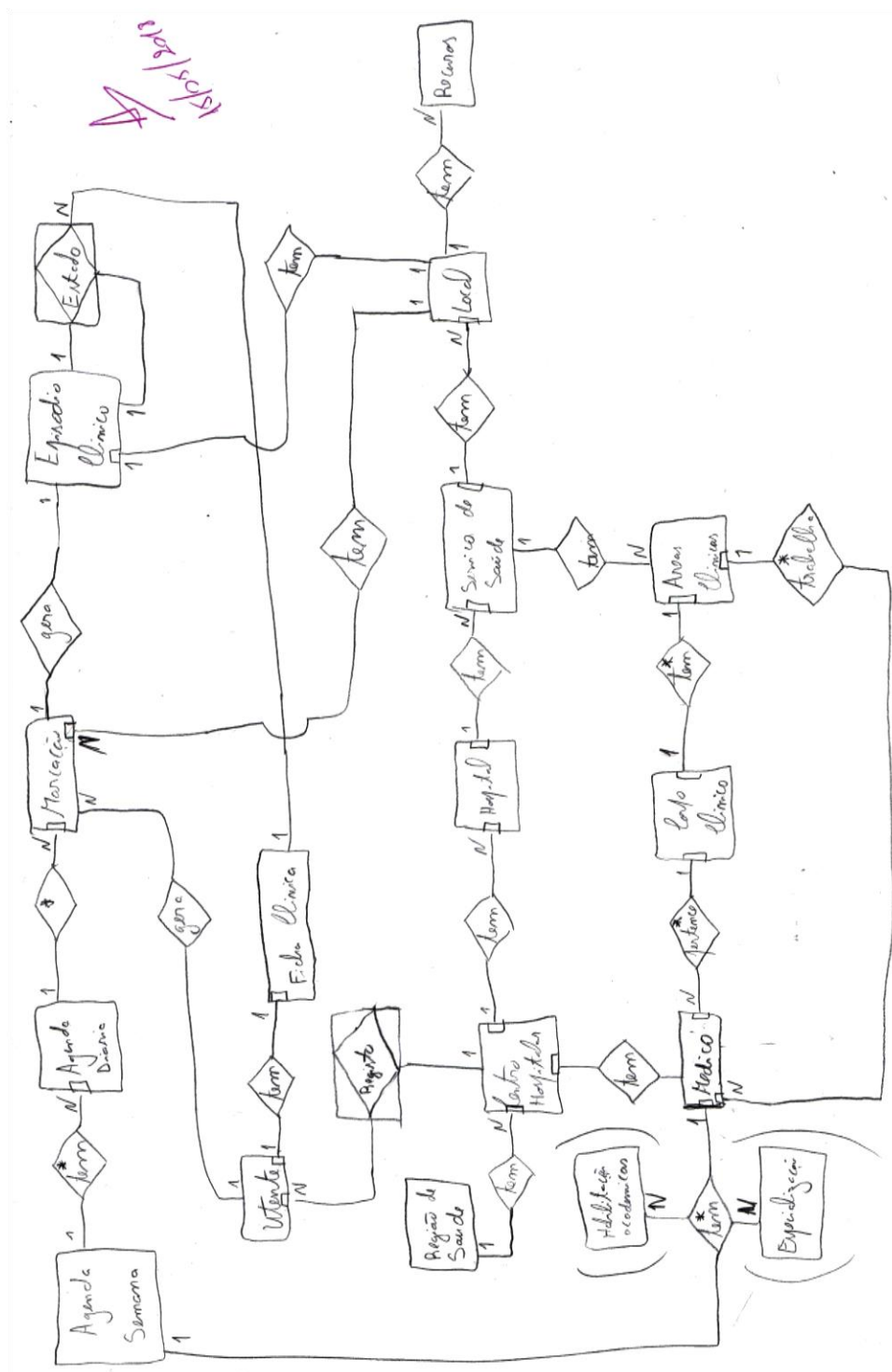
8/5/2018



8/5/2018

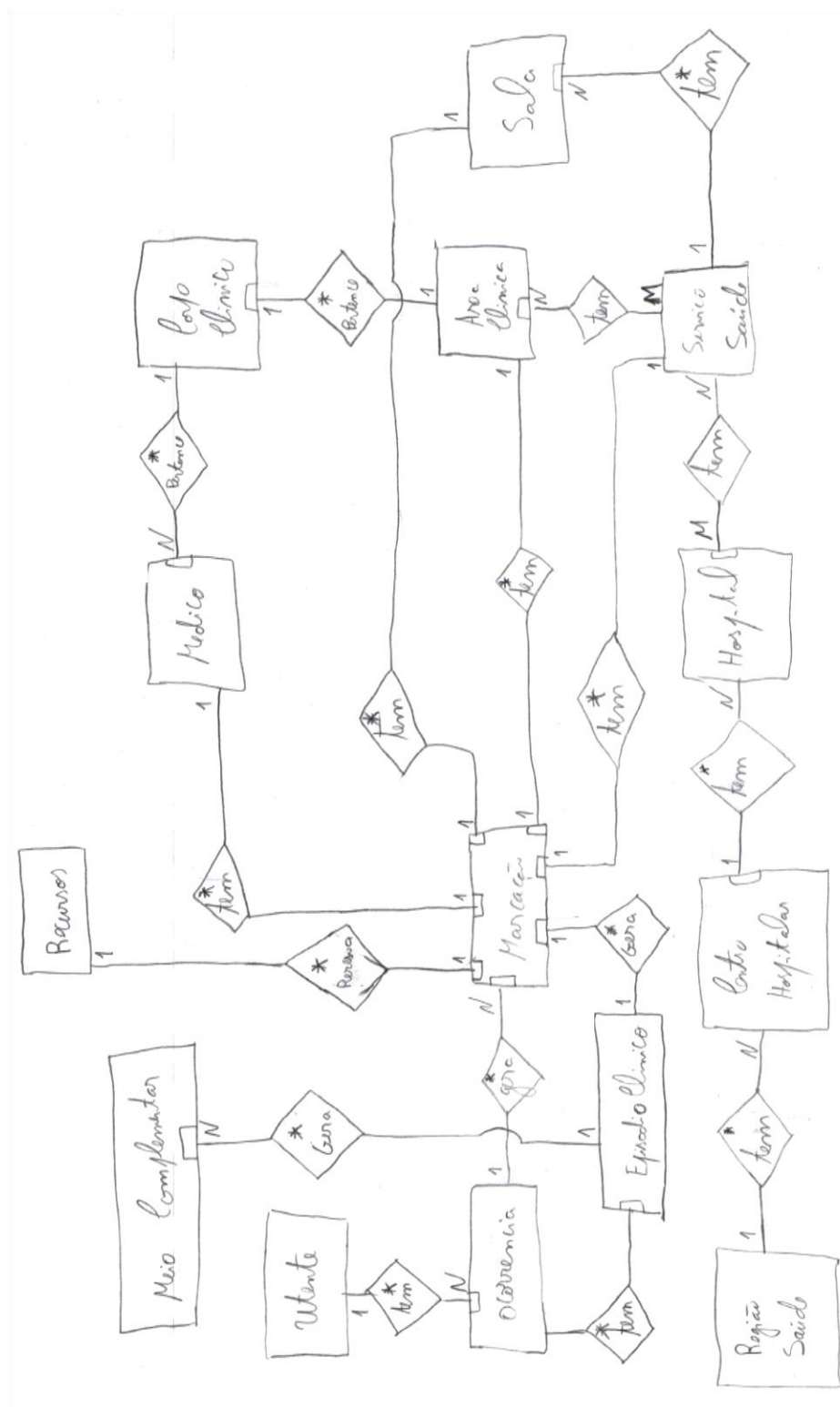
2.3 – 3ª Fase da modelação de dados

Este modelo foi o ultimo a ser desenvolvido antes de o grupo começar a fazer os scripts e a adicionar dados às tabelas. Durante esse processo o grupo deparou-se com algumas inconformidades e portanto foi desenvolvido um quarto e ultimo modelo de dados que consideramos o modelo final (4ª fase de modelação de dados _ Final).



2.4 – 4ª Fase da modelação de dados_Final

Durante a execução dos scripts, o grupo deu-se conta que ainda existiam dados redundantes ou sem importância para o resultado final, e portanto essas correções deram origem a este ultimo modelo de dados.



2.4.1 – Dados Iniciais da Base de Dados

Os atributos a **azul** são **chaves primarias**, os atributos a **verde** são **chaves estrangeiras**, e a **vermelho** são **chaves compostas**.

Utente(**UtenteID**, Nome, BI)

UtenteID	Nome	BI
1	João	12345
2	Rita	54321
3	Pedro	87654
4	Joana	34856

Ocorrencia(**OcorrenciaID**, Tipo, **UtenteID**, **DescricaoID**)

OcorrenciaID	Tipo	UtenteID	Descricao
1	Urgência	3	Irritação no pé
2	Normal	3	Dor no peito
3	Normal	2	Dor de Cabeça
4	Urgência	4	Braço Partido

Marcação(**MarcacaoID**, **OcorrenciaID**, **ServicoSaudeID**, **AreaClinicaID**, **MedicoID**, Dia, Hora, **SalaID**, **RecursosID**)

MarcacaoID	OcorrenciaID	ServicoSaudeID	AreaClinicaID	MedicoID	Dia	Hora	SalaID	RecursosID
1	1	5	3 (ortopedia)	2	2015/03/24	22:00	4	5
2	1	8	4(dermatologia)	1	2015/03/24	22:30	2	3
3	3	3	1 (Psiquiatria)	4	2016/08/02	14:00	3	2
4	3	7	2 (Neurologia)	3	2016/08/02	15:00	1	4
5	2	4	6(Pediatria)	5	2015/12/25	18:00	5	1

Episodio Clinico(**EpisodioClinicoID**, **MarcacaoID**, **OcorrenciaID**, **Descricao**)

EpisodioClinicoID	MarcacaoID	OcorrenciaID	Descricao
1	3	3	Encaminhamento neurologia
2	4	3	Foi receitado Brufen
3	5	2	Será operado ao coração
4	1	1	Análise ao sangue
5	2	1	Eczema

Sala(**SalaID**, **ServicoSaudeID**)

SalaID	ServicoSaudeID
1	1
2	1
3	3
4	2
5	2
6	4

Meios Complementares(**MeioComplementarID**, **EpisodioClinicoID**, Nome, Resultado)

MeioComplementarID	EpisodioClinicoID	Nome	Resultado
1	3	Operação	Óbito
2	4	Análise	Nada a relatar

Medico(**MedicoID**, Nome, **CorpoClinicoID**)

MedicoID	Nome	CorpoClinicoID
1	Alberto	2 Dermatologia
2	Ana	1 Ortopedia
3	Miguel	4 Neurologia
4	Andre	3 Psiquiatria
5	Patricia	6 Med. Dentaria
6	Antonio	2 Dermatologia
7	Tiago	1 Ortopedia

Area Clinica(**AreaClinicaID**, Nome)

AreaClinicaID	Nome
1	Psiquiatria
2	Neurologia
3	Ortopedia
4	Dermatologia
5	Cardiologia
6	Pediatria
7	Urologia

Corpo Clinico(CorpoClinicoID, AreaClinicaID)

CorpoClinicoID	AreaClinicaID
1	3 ortopedia
2	4 dermatologia
3	1 psiquiatria
4	2 neurologia
5	6 pediatria
6	5 cardiologia
7	7 urologia

Servico de Saude(ServicoSaudeID, Nome)

ServicoSaudeID	Nome
1	Centro Saude
2	Centro Internamento
3	Centro Tratamento Crónico
4	Centro Operatorio
5	Centro de Analises
6	Centro Maternidade
7	Centro Farmaceutico
8	Centro Urgencia

Atribuicao Servico Area(ServicoSaudeID, AreaClinicaID)

ServicoSaudeID	AreaClinicaID
1	6
1	5
2	1
2	2
3	7
3	1
4	7
4	6
4	5
5	7
5	3
6	6
6	4
7	2
8	5
8	4

Regiao Saude(**RegiaoSaudeID**, Nome)

RegiaoID	Nome
1	Regiao Centro
2	Regiao Norte

Centro Hospitalar(**CentroHospitalarID**, **RegiaoID**, Nome)

CentroHospitalarID	RegiaoID	Nome
1	1	CH Cova da Beira
2	1	CH Beira Interior
3	2	CH Alto Douro

Hospital(**HospitalID**, **CentroHospitalarID**, Nome)

HospitalID	CentroHospitalarID	Nome
1	1	Santa Maria
2	1	Sousa Martins
3	3	São Joao
4	2	Trindade

Atribuicao Hospital(**HospitalID**, **ServicoSaudeID**)

HospitalID	ServicoSaudeID
1	1
1	2
1	4
2	7
2	8
2	3
3	4
3	5
3	8
4	1
4	3

Recursos(RecursosID, Nome, Validade)

RecursosID	Nome	Validade
1	cama	2015-03-26
2	vacina	2016-12-25
3	toalha	2017-07-21
4	seringa	2017-01-02
5	creme	2019-11-30

2.4.2 – Funcionamento e Descrição da Organização

- Um utente chegará ao hospital e registrará a sua ocorrência (ex.: Dor de cabeça). Aí, será gerada uma ocorrência, que poderá ser do tipo normal, ou urgência, caso o utente entre de urgência com esse sintoma.
- De seguida será marcada uma primeira marcação, referente a essa ocorrência (Dor de cabeça). A marcação necessitará do ID da ocorrência, da área clínica e do serviço de saúde referentes ao sintoma do utente, o ID da sala onde se dará o futuro episódio clínico, o dia e a hora em que será o episódio clínico, e o ID do recurso que será necessário.
- Caso seja necessário, e se justifique (ou seja, caso o problema do utente ainda não tenha ficado resolvido), será marcado um outro episódio clínico (uma nova marcação), dentro da mesma ocorrência (Dor de cabeça).
- O episódio clínico será a concretização da marcação efetuada anteriormente. Nos dados do episódio clínico o medido introduzirá o ID da marcação a que corresponde o episódio clínico, e a descrição da consulta (evolução do estado do paciente, o que foi receitado, o que será feito, etc).
- Os recursos serão divididos por ID, onde cada recurso tem um nome, e a sua validade.
- Cada médico tem um ID, e o ID do corpo clínico ao qual ele pertence.
- Os corpos clínicos são divididos por ID, onde cada corpo clínico pertence a uma área clínica.
- Cada serviço de saúde terá um nome e um ID. Foi feita uma tabela de atribuição onde se atribui as diversas áreas clínicas aos serviços de saúde respetivos, pois um serviço de saúde poderá ter várias áreas clínicas.
- Existem também várias salas, cada uma com um ID próprio. Cada sala está associada a um serviço de saúde.
- Existem várias regiões de saúde. Cada região de saúde tem vários Centros Hospitalares, e cada Centro Hospitalar tem vários Hospitais. Cada serviço de saúde poderá (ou não) estar presente em vários hospitais, tal como um hospital terá vários serviços diferentes.
- Do episódio clínico podem resultar meios complementares (operações, análises, etc). Cada meio complementar é identificado por um ID, e terá um nome, e a que episódio clínico pertence.

3. Aplicação

3.1 - Decomposição e distribuição de tarefas

- Elaboração do modelo de dados (modelação) – Miguel Gregório, Tiago Canhão
- Criação das Scripts (criar bd, criar tabelas e restrições, inserir dados) – João Lopes, Tiago Canhão;
- Criação das aplicações (design) – João Lopes, Tiago Canhão, Miguel Gregório;
- Inserir manipulação de dados na aplicação – João Lopes, Tiago Canhão, Miguel Gregório;
- Criação do relatório – Miguel Gregório.

3.2 - Acesso à base de dados

3.2.1 – Ligação NetBeans – DataBase

```
import java.sql.*;

public class Db {
    static private Connection con;

    public static Connection getCon() throws Exception{
        if (con == null){
            Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
            con = DriverManager.getConnection("jdbc:sqlserver://localhost:1433;databaseName=Projeto;user=sa;password=sqlserver");
        }
        return con;
    }

    public static void setData(String s) throws Exception{
        Db.getCon().createStatement().executeUpdate(s);
    }

    public static ResultSet getData(String s) throws Exception{
        return Db.getCon().createStatement().executeQuery(s);
    }
}
```

3.2.2 – Exemplo de consulta de dados na DataBase

Este excerto vai buscar a ocorrenciaID à tabela marcação quando a marcacaoID é igual à marcação escrita pelo utilizador na TextField.

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
    int i = 0;
    String s = null;
    try{
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        Connection con = DriverManager.getConnection("jdbc:sqlserver://localhost:1433;databaseName=Projeto;user=sa;password=sqlserver");
        String query = "select OcorrenciaID from Marcacao where MarcacaoID = "+jTextField1.getText()+"";
        Statement st = con.createStatement();
        ResultSet rs = st.executeQuery(query);
        while(rs.next()){
            i = rs.getInt("OcorrenciaID");
        }
        jLabel7.setText(i + "");
    }
    catch(Exception e){
        JOptionPane.showMessageDialog(null, e);
    }
    // TODO add your handling code here:
}
```

3.2.3 – Exemplo de Inserção de dados na DataBase

Este excerto irá consultar a tabela das Ocorrências e guardar o ID da ocorrência mais alta. De seguida irá criar uma nova linha na tabela com o ID seguinte e com os dados inseridos nas TextFields.

```
private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {  
    try{  
        int i = 0;  
        String s = null;  
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");  
        Connection con = DriverManager.getConnection("jdbc:sqlserver://localhost:1433;databaseName=Projeto;user=sa;password=sqlserver");  
        String query = "INSERT INTO Ocorrencia (OcorrenciaID, Tipo, UtenteID, Descricao) values (?, ?, ?, ?)";  
        PreparedStatement pst = con.prepareStatement(query);  
        ResultSet rs = Db.getData("select Max(OcorrenciaID) from Ocorrencia");  
        while(rs.next()){  
            i = rs.getInt(1);  
            i++;  
            s = String.valueOf(i);  
        }  
        pst.setString(1, s);  
        pst.setString(2, jTextField2.getText());  
        pst.setString(3, jTextField1.getText());  
        pst.setString(4, jTextArea1.getText());  
        pst.executeUpdate();  
        JOptionPane.showMessageDialog(null, "Inserido com sucesso");  
    }  
    catch(Exception e){  
        JOptionPane.showMessageDialog(null, e);  
    }  
    // TODO add your handling code here:  
}
```

3.3 – Funcionalidade

A Aplicação funciona do seguinte modo:

- 1) A aplicação inicializa mostrando o interface da figura 1 – menu principal. Poder-se-á escolher-se cada um dos três tópicos.



Figura 1

- 2) Optando por clicar no separador marcações, o utilizador terá de optar por uma das opções (figura 2): Visualizar o Horário semanal de um médico, Visualizar o Horário semanal de uma sala, e efetuar agendamentos de consultas (marcação).



Figura 2

- 3) Se o utilizador clicar em “visualizar horário semanal”, terá um espaço para colocar o ID do médico do qual pretende consultar o horário. Terá também de seleccionar o dia em que pretende começar a ver esse horário. Ao premir “Pesquisar”, aparecerá na tabela as marcações que o médico tem marcadas na data inserida, e nos seis dias seguintes. No caso da figura 3, o médico 2 tem uma marcação no dia 24/03/2015 e outra no dia 25/03/2015. Caso o médico não tenha nenhuma marcação nas datas seleccionadas, a tabela aparecerá vazia.

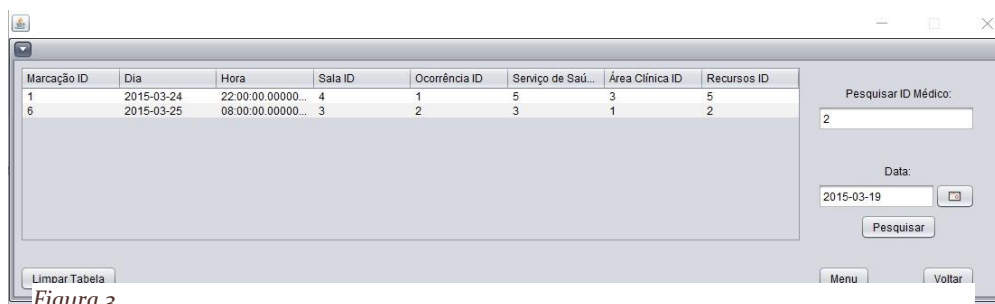


Figura 3

- 4) Muito semelhante a 3), a opção “Visualizar horário Sala” permitirá consultar o horário de uma determinada sala (Figura 4).

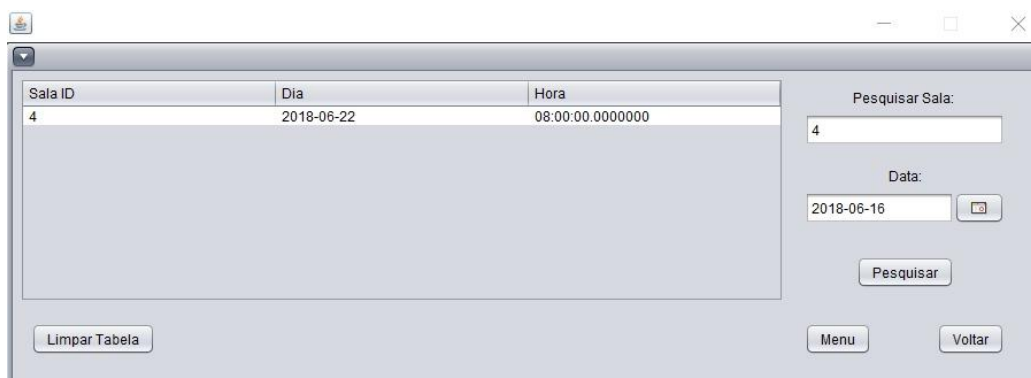


Figura 4

- 5) Na Opção “Agendamento”, Apresenta-se o interface da figura 5.
- À esquerda, é possível registar uma ocorrência. Essa ocorrência pode ser do tipo normal ou urgência. De seguida colocar-se-á o ID do utente, e a descrição da ocorrência. E finalmente ao clicar no botão “criar”, criará então uma nova ocorrência com os dados inseridos. Caso o ID do utente não exista, o programa irá emitir uma mensagem de erro.
- À direita, é possível agendar uma marcação de um episódio clinico. Para isso será necessário inserir o ID da ocorrência a que se refere a marcação, a sala, o médico, a área clinica, entre outros. Caso alguns dos dados não existam, ou não exista uma correspondência entre eles (ex.: aquele serviço de saúde não possui aquela área clinica, ou aquele médico não é daquela área clinica), o programa

emite uma mensagem de erro ao utilizador ao premir no botão “Criar Marcação”. Caso contrario o programa cria a marcação normalmente.

The screenshot shows two side-by-side windows. The left window, titled 'Ocorrência', contains a dropdown menu for 'Tipo' set to 'Normal', a text field for 'Utente ID' with the value '2', and a text area for 'Descrição' containing the word 'Teste'. At the bottom right of this window is a button labeled 'Criar'. The right window, titled 'Marcação', contains several text fields: 'Ocorrência ID' (4), 'Serviço de Saúde ID' (2), 'Data' (018-06-15), 'Área Clínica ID' (1), 'Hora' (09:30), 'Médico ID' (5), 'Sala' (4), and 'Recursos ID' (3). At the bottom right of this window is a button labeled 'Criar Marcação'. At the bottom of the entire interface are two buttons: 'Menu' on the left and 'Voltar' on the right.

Figura 5

- 6) Voltando ao menu principal, e seleccionando a opção “Episódio Clínico”, o utilizador é direcionado para o interface da figura 6.
 À esquerda, o utilizador (que à partida será um médico) irá colocar o ID da marcação que está a acontecer no momento. Ao premir o botão “Atualizar”, aparecerá em baixo o número da ocorrência correspondente. Se o ID da marcação não existir, dará uma mensagem de erro. De seguida, ao clicar no botão “Tabela”, aparecerão à direita os episódios clínicos anteriores da mesma ocorrência. Também aparecerão os respetivos meios complementares (operações, análises, etc., e respectivos resultados) caso existam. De seguida, o médico adicionará uma descrição ao progresso da consulta atual.

The screenshot shows a single window titled 'Registo de Progresso da Consulta'. On the left side, there is a text field for 'Marcação ID' with the value '4', a button 'Atualizar', a text field for 'Ocorrência' with the value '3', a button 'Tabela', and a text area for 'Descrição' containing the word 'Teste'. At the bottom right of this section is a button labeled 'Adicionar'. On the right side, there are two tables. The top table is titled 'Consultas anteriores' and has two columns: 'Episódio Clínico ID' and 'Descrição'. It contains four rows of data. The bottom table is titled 'Ver Meios Complementares' and has three columns: 'Episódio Clínico ID', 'Nome', and 'Resultado'. It contains one row of data. At the bottom of the window are two buttons: 'Limpar Tabelas' on the left and 'Voltar' on the right.

Episódio Clínico ID	Descrição
1	Encaminhamento Neurologia
2	Foi recetado Brufen
7	eTeste
8	eTeste

Episódio Clínico ID	Nome	Resultado
2	ee	rr

Figura 6

- 7) Finalmente, caso o utilizador, no menu principal, selecione a opção “Meios Complementares”, será direcionado para um interface onde é possível registar um meio complementar. Será pedido ao utilizador o ID do episódio clínico à qual corresponde o meio complementar que irá registar. Será também pedido ao utilizador que indique o nome (análise, operação, etc.), e o resultado desse mesmo episódio.

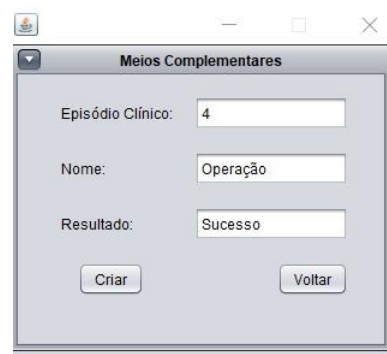
A imagem mostra uma janela de software intitulada "Meios Complementares". No topo, há uma barra de título com ícones de minimizar, maximizar e fechar. O corpo da janela contém três campos de entrada rotulados "Episódio Clínico:", "Nome:" e "Resultado:". Os campos "Episódio Clínico:" e "Resultado:" contêm os valores "4" e "Sucesso", respetivamente. O campo "Nome:" contém o valor "Operação". Na base da janela, há dois botões: "Criar" à esquerda e "Voltar" à direita.

Figura 7

- 8) Em todas as fases existe um botão de “voltar” e/ou “Menu”, para que o utilizador consiga navegar na aplicação sem ter de a fechar. Existem também botões para limpar tabelas.

4. Conclusões

Pensamos que no geral este projeto foi bem conseguido, pois conseguimos elaborar tanto o modelo de dados proposto, como a aplicação com as funcionalidades pretendidas. O grupo conseguiu também criar soluções para situações onde o enunciado não transmite uma ideia clara do que é pretendido.

Na opinião do grupo, a manipulação de dados no formato Data dos diferentes programas utilizados (SQL server e NetBeans) foi uma das tarefas que criaram mais dificuldades.

Com mais tempo e mais conhecimento, a tabela e restantes funcionalidades do assunto “Recursos”, poderia ser mais bem conseguida, apesar de estar a funcionar.

5. Epilogo

O grupo de trabalho está de acordo quanto ao facto se esta ser uma cadeira útil e bastante importante neste curso. Contudo, seriam de esperar aulas mais práticas e mais diversificadas, e não tão fechadas e dentro do mesmo tema/assunto.

Outro assunto onde o grupo está também de acordo é que a cotação deste projeto não reflete a dedicação e empenho que ele exige. Apesar de o professor perguntar aos alunos se estavam de acordo com a cotação atribuída ao projeto, os alunos não sabiam o grau de exigência que o projeto teria.

5. Apêndice

5.1 – Script Criar DataBase

```
USE master

IF ( EXISTS( SELECT * FROM [dbo].[sysdatabases] Where name = 'Projeto') )
Begin
    DROP DATABASE Projeto
end

IF (NOT EXISTS( SELECT * FROM [dbo].[sysdatabases] Where name = 'Projeto') )
Begin

    CREATE DATABASE Projeto
    ON
        ( NAME = 'Projeto_dat',
          FILENAME = 'C:\BD1718\Projetodat.mdf',
          -- FILENAME = N'e:\Microsoft SQL Server\MSSQL\data\PJdat.mdf',

          SIZE = 10,
          MAXSIZE = 200,
          FILEGROWTH = 5 )
    LOG ON
        ( NAME = 'Projeto_log',
          FILENAME = 'C:\BD1718\Projetolog.ldf',
          -- FILENAME = N'e:\Microsoft SQL Server\MSSQL\data\PJlog.ldf',

          SIZE = 5MB,
          MAXSIZE = 100MB,
          FILEGROWTH = 5MB )
End
```

5.2 – Script Criar Tabelas

```
USE Projeto

if not exists (select * from dbo.sysobjects
               where id = object_id(N'[dbo].[Utente]') )
begin
    CREATE TABLE Utente (
        UtenteID int NOT NULL
            CHECK (UtenteID >= 1),                -- constraint type: check
        Nome nvarchar(100) NOT NULL,
        BI nvarchar(30) NOT NULL,

        CONSTRAINT PK_UtenteID PRIMARY KEY (UtenteID) -- constraint type: primary key
        /*CONSTRAINT U_Nome UNIQUE (Nome)             -- constraint type: unique*/
    );
end

-- .....

if not exists (select * from dbo.sysobjects
               where id = object_id(N'[dbo].[AreaClinica]') )
```

```

begin
    CREATE TABLE AreaClinica (
        AreaClinicaID int NOT NULL
        CONSTRAINT PK_AreaClinicaID PRIMARY KEY (AreaClinicaID)
        CHECK (AreaClinicaID >= 1),
        Nome nvarchar(100) NOT NULL,
    );
end

-- .....

if not exists (select * from dbo.sysobjects
                where id = object_id(N'[dbo].[CorpoClinico]')) )
begin
    CREATE TABLE CorpoClinico (
        CorpoClinicoID int NOT NULL
        CONSTRAINT PK_CorpoClinicoID PRIMARY KEY (CorpoClinicoID)
        CHECK (CorpoClinicoID >= 1),
        AreaClinicaID int NOT NULL
        CONSTRAINT FK_AreaClinicaID FOREIGN KEY (AreaClinicaID)
        REFERENCES AreaClinica(AreaClinicaID)
    );
end

-- .....

if not exists (select * from dbo.sysobjects
                where id = object_id(N'[dbo].[Medico]')) )
begin
    CREATE TABLE Medico (
        MedicoID int NOT NULL
        CONSTRAINT PK_MedicoID PRIMARY KEY (MedicoID)
        CHECK (MedicoID >= 1),
        Nome nvarchar(100) NOT NULL,
        CorpoClinicoID int NOT NULL
        CONSTRAINT FK_CorpoClinicoID FOREIGN KEY (CorpoClinicoID)
        REFERENCES CorpoClinico(CorpoClinicoID)
    );
end

-- .....

if not exists (select * from dbo.sysobjects
                where id = object_id(N'[dbo].[RegiaoSaude]')) )
begin
    CREATE TABLE RegiaoSaude (
        RegiaoID int NOT NULL
        CONSTRAINT PK_RegiaoID PRIMARY KEY (RegiaoID)
        CHECK (RegiaoID >= 1),
        Nome nvarchar(100) NOT NULL
    );
end

-- .....

if not exists (select * from dbo.sysobjects
                where id = object_id(N'[dbo].[CentroHospitalar]')) )
begin
    CREATE TABLE CentroHospitalar (
        CentroHospitalarID int NOT NULL
        CONSTRAINT PK_CentroHospitalarID PRIMARY KEY (CentroHospitalarID)

```

```

        CHECK (CentroHospitalarID >= 1),
        RegiaoID int NOT NULL
        CONSTRAINT FK_RegiaoID FOREIGN KEY (RegiaoID)
        REFERENCES RegiaoSaude(RegiaoID),
        Nome nvarchar(100) NOT NULL
    );
end

-- .....
if not exists (select * from dbo.sysobjects
               where id = object_id(N'[dbo].[Hospital]')) )
begin
    CREATE TABLE Hospital (
        HospitalID int NOT NULL
        CONSTRAINT PK_HospitalID PRIMARY KEY (HospitalID)
        CHECK (HospitalID >= 1),
        CentroHospitalarID int NOT NULL
        CONSTRAINT FK_CentroHospitalarID FOREIGN KEY (CentroHospitalarID)
        REFERENCES CentroHospitalar(CentroHospitalarID),
        Nome nvarchar(100) NOT NULL
    );
end

-- .....
if not exists (select * from dbo.sysobjects
               where id = object_id(N'[dbo].[ServicoSaude]')) )
begin
    CREATE TABLE ServicoSaude (
        ServicoSaudeID int NOT NULL
        CONSTRAINT PK_ServicoSaudeID PRIMARY KEY (ServicoSaudeID)
        CHECK (ServicoSaudeID >= 1),
        Nome nvarchar(100) NOT NULL
    );
end

-- .....
if not exists (select * from dbo.sysobjects
               where id = object_id(N'[dbo].[Sala]')) )
begin
    CREATE TABLE Sala (
        SalaID int NOT NULL
        CONSTRAINT PK_SalaID PRIMARY KEY (SalaID)
        CHECK (SalaID >= 1),
        ServicoSaudeID int NOT NULL
        CONSTRAINT FK_ServicoSaudeID FOREIGN KEY (ServicoSaudeID)
        REFERENCES ServicoSaude(ServicoSaudeID)
    );
end

-- .....
if not exists (select * from dbo.sysobjects
               where id = object_id(N'[dbo].[Recursos]')) )
begin
    CREATE TABLE Recursos (
        RecursosID int NOT NULL
        CONSTRAINT PK_RecursosID PRIMARY KEY (RecursosID)
        CHECK (RecursosID >= 1),
        Nome nvarchar(100) NOT NULL,
        Validade Date NOT NULL
    );
end

```

```

-- .....

if not exists (select * from dbo.sysobjects
               where id = object_id(N'[dbo].[Ocorrencia]')) )
begin
    CREATE TABLE Ocorrencia (
        OcorrenciaID int NOT NULL
        CONSTRAINT PK_OcorrenciaID PRIMARY KEY (OcorrenciaID)
        CHECK (OcorrenciaID >= 1),
        Tipo nvarchar(10) NOT NULL,
        UtenteID int NOT NULL
        CONSTRAINT FK_UtenteID FOREIGN KEY (UtenteID)
        REFERENCES Utente(UtenteID),
        Descricao nvarchar(500) NOT NULL
    );
end

-- .....

if not exists (select * from dbo.sysobjects
               where id = object_id(N'[dbo].[Marcacao]')) )
begin
    CREATE TABLE Marcacao (
        MarcacaoID int NOT NULL
        CONSTRAINT PK_MarcacaoID PRIMARY KEY (MarcacaoID)
        CHECK (MarcacaoID >= 1),
        OcorrenciaID int NOT NULL
        CONSTRAINT FK_OcorrenciaID FOREIGN KEY (OcorrenciaID)
        REFERENCES Ocorrencia(OcorrenciaID),
        ServicoSaudeID int NOT NULL
        CONSTRAINT SK_ServicoSaudeID FOREIGN KEY (ServicoSaudeID)
        REFERENCES ServicoSaude(ServicoSaudeID),
        AreaClinicaID int NOT NULL
        CONSTRAINT SK_AreaClinicaID FOREIGN KEY (AreaClinicaID)
        REFERENCES AreaClinica(AreaClinicaID),
        MedicoID int NOT NULL
        CONSTRAINT FK_MedicoID FOREIGN KEY (MedicoID)
        REFERENCES Medico(MedicoID),
        Dia Date NOT NULL,
        Hora Time NOT NULL,
        SalaID int NOT NULL
        CONSTRAINT FK_SalaID FOREIGN KEY (SalaID)
        REFERENCES Sala(SalaID),
        RecursosID int NOT NULL
        CONSTRAINT FK_RecursosID FOREIGN KEY (RecursosID)
        REFERENCES Recursos(RecursosID)
    );
end

-- .....

if not exists (select * from dbo.sysobjects
               where id = object_id(N'[dbo].[EpisodioClinico]')) )
begin
    CREATE TABLE EpisodioClinico (
        EpisodioClinicoID int NOT NULL
        CONSTRAINT PK_EpisodioClinicoID PRIMARY KEY (EpisodioClinicoID)
        CHECK (EpisodioClinicoID >= 1),
        MarcacaoID int NOT NULL
        CONSTRAINT FK_MarcacaoID FOREIGN KEY (MarcacaoID)
        REFERENCES Marcacao(MarcacaoID),
        OcorrenciaID int NOT NULL
        CONSTRAINT SK_OcorrenciaID FOREIGN KEY (OcorrenciaID)

```

```

        REFERENCES Ocorrencia(OcorrenciaID),
        Descricao nvarchar(500) NOT NULL
    );
end

-- .....

if not exists (select * from dbo.sysobjects
               where id = object_id(N'[dbo].[MeiosComplementares]')) )
begin
    CREATE TABLE MeiosComplementares (
        MeiosComplementaresID int NOT NULL
        CONSTRAINT PK_MeiosComplementaresID PRIMARY KEY (MeiosComplementaresID)
        CHECK (MeiosComplementaresID >= 1),
        EpisodioClinicoID int NOT NULL
        CONSTRAINT FK_EpisodioClinicoID FOREIGN KEY (EpisodioClinicoID)
        REFERENCES EpisodioClinico(EpisodioClinicoID),
        Nome nvarchar(20) NOT NULL,
        Resultado nvarchar(500) NOT NULL
    );
end

-- .....

if not exists (select * from dbo.sysobjects
               where id = object_id(N'[dbo].[AtribuicaoServicoArea]')) )
begin
    CREATE TABLE AtribuicaoServicoArea (
        ServicoSaudeID int
        CONSTRAINT nn_ServicoSaudeID NOT NULL,
        AreaClinicaID int
        CONSTRAINT nn_AreaClinicaID NOT NULL,
        -- nn -> NOT NULL

        CONSTRAINT PK_AtribuicaoServicoArea
        PRIMARY KEY (ServicoSaudeID, AreaClinicaID),           -- constraint
type: primary key

        CONSTRAINT UK_ServicoSaudeID FOREIGN KEY (ServicoSaudeID)
        REFERENCES ServicoSaude(ServicoSaudeID),

        CONSTRAINT UK_AreaClinicaID FOREIGN KEY (AreaClinicaID)
        REFERENCES AreaClinica(AreaClinicaID)
    );
end

-- .....

if not exists (select * from dbo.sysobjects
               where id = object_id(N'[dbo].[AtribuicaoHospitalServico]')) )
begin
    CREATE TABLE AtribuicaoHospitalServico (
        HospitalID int
        CONSTRAINT nn_HospitalID NOT NULL,
        ServicoSaudeID int
        CONSTRAINT nn_ServicoSaudeID NOT NULL,
        -- nn -> NOT NULL

        CONSTRAINT PK_AtribuicaoHospitalServico
        PRIMARY KEY (HospitalID, ServicoSaudeID),           -- constraint type:
primary key

```

```
        CONSTRAINT FK_HospitalID FOREIGN KEY (HospitalID)
        REFERENCES Hospital(HospitalID),

        CONSTRAINT TK_ServicoSaudeID FOREIGN KEY (ServicoSaudeID)
        REFERENCES ServicoSaude(ServicoSaudeID)

    );
end
```

5.3 – Script Inserir Dados Nas tabelas

```
USE Projeto

INSERT INTO Utente(UtenteID, Nome, BI)
Values (1, 'João', 12345);

INSERT INTO Utente(UtenteID, Nome, BI)
Values (2, 'Rita', 54321);

INSERT INTO Utente(UtenteID, Nome, BI)
Values (3, 'Pedro', 87654);

INSERT INTO Utente(UtenteID, Nome, BI)
Values (4, 'Joana', 34856);


INSERT INTO AreaClinica(AreaClinicaID, Nome)
Values (1, 'Psiquiatria');

INSERT INTO AreaClinica(AreaClinicaID, Nome)
Values (2, 'Neurologia');

INSERT INTO AreaClinica(AreaClinicaID, Nome)
Values (3, 'Ortopedia');

INSERT INTO AreaClinica(AreaClinicaID, Nome)
Values (4, 'Dermatologia');

INSERT INTO AreaClinica(AreaClinicaID, Nome)
Values (5, 'Cardiologia');

INSERT INTO AreaClinica(AreaClinicaID, Nome)
Values (6, 'Pediatria');

INSERT INTO AreaClinica(AreaClinicaID, Nome)
Values (7, 'Urologia');


INSERT INTO CorpoClinico(CorpoClinicoID, AreaClinicaID)
Values (1, 3);

INSERT INTO CorpoClinico(CorpoClinicoID, AreaClinicaID)
Values (2, 4);
```

```
INSERT INTO CorpoClinico(CorpoClinicoID, AreaClinicaID)
Values (3, 1);
```

```
INSERT INTO CorpoClinico(CorpoClinicoID, AreaClinicaID)
Values (4, 2);
```

```
INSERT INTO CorpoClinico(CorpoClinicoID, AreaClinicaID)
Values (5, 6);
```

```
INSERT INTO CorpoClinico(CorpoClinicoID, AreaClinicaID)
Values (6, 5);
```

```
INSERT INTO CorpoClinico(CorpoClinicoID, AreaClinicaID)
Values (7, 7);
```

```
INSERT INTO Medico(MedicoID, Nome, CorpoClinicoID)
Values (1, 'Alberto', 2);
```

```
INSERT INTO Medico(MedicoID, Nome, CorpoClinicoID)
Values (2, 'Ana', 1);
```

```
INSERT INTO Medico(MedicoID, Nome, CorpoClinicoID)
Values (3, 'Miguel', 4);
```

```
INSERT INTO Medico(MedicoID, Nome, CorpoClinicoID)
Values (4, 'André', 3);
```

```
INSERT INTO Medico(MedicoID, Nome, CorpoClinicoID)
Values (5, 'Patrícia', 6);
```

```
INSERT INTO Medico(MedicoID, Nome, CorpoClinicoID)
Values (6, 'António', 2);
```

```
INSERT INTO Medico(MedicoID, Nome, CorpoClinicoID)
Values (7, 'Tiago', 1);
```

```
INSERT INTO RegiaoSaude(RegiaoID, Nome)
Values (1, 'Região Centro');
```

```
INSERT INTO RegiaoSaude(RegiaoID, Nome)
Values (2, 'Região Norte');
```

```
INSERT INTO CentroHospitalar(CentroHospitalarID, RegiaoID, Nome)
Values (1, 1, 'CH Cova da Beira');
```

```
INSERT INTO CentroHospitalar(CentroHospitalarID, RegiaoID, Nome)
Values (2, 1, 'CH Beira Interior');
```

```
INSERT INTO CentroHospitalar(CentroHospitalarID, RegiaoID, Nome)
Values (3, 2, 'CH Alto Douro');
```



```
INSERT INTO Hospital(HospitalID, CentroHospitalarID, Nome)
Values (1, 1, 'Santa Maria');
```

```
INSERT INTO Hospital(HospitalID, CentroHospitalarID, Nome)
Values (2, 1, 'Sousa Martins');
```

```
INSERT INTO Hospital(HospitalID, CentroHospitalarID, Nome)
Values (3, 3, 'São João');
```

```
INSERT INTO Hospital(HospitalID, CentroHospitalarID, Nome)
Values (4, 2, 'Trindade');
```

```
INSERT INTO ServicoSaude(ServicoSaudeID, Nome)
Values (1, 'Centro de Saúde');
```

```
INSERT INTO ServicoSaude(ServicoSaudeID, Nome)
Values (2, 'Centro de Internamento');
```

```
INSERT INTO ServicoSaude(ServicoSaudeID, Nome)
Values (3, 'Centro de Tratamento Crónico');
```

```
INSERT INTO ServicoSaude(ServicoSaudeID, Nome)
Values (4, 'Centro Operatório');
```

```
INSERT INTO ServicoSaude(ServicoSaudeID, Nome)
Values (5, 'Centro de Análises');
```

```
INSERT INTO ServicoSaude(ServicoSaudeID, Nome)
Values (6, 'Centro de Maternidade');
```

```
INSERT INTO ServicoSaude(ServicoSaudeID, Nome)
Values (7, 'Centro Farmacêutico');
```

```
INSERT INTO ServicoSaude(ServicoSaudeID, Nome)
Values (8, 'Centro de Urgência');
```

```
INSERT INTO Sala(SalaID, ServicoSaudeID)
Values (1, 1);
```

```
INSERT INTO Sala(SalaID, ServicoSaudeID)
Values (2, 1);
```

```
INSERT INTO Sala(SalaID, ServicoSaudeID)
Values (3, 3);
```

```
INSERT INTO Sala(SalaID, ServicoSaudeID)
Values (4, 2);
```

```
INSERT INTO Sala(SalaID, ServicoSaudeID)
Values (5, 2);
```

```
INSERT INTO Sala(SalaID, ServicoSaudeID)
Values (6, 4);
```

```
INSERT INTO Recursos(RecursosID, Nome, Validade)
Values (1, 'Cama', '2015-03-26');
```

```
INSERT INTO Recursos(RecursosID, Nome, Validade)
Values (2, 'Vacina', '2016-12-25');
```

```
INSERT INTO Recursos(RecursosID, Nome, Validade)
Values (3, 'Toalha', '2017-07-21');
```

```
INSERT INTO Recursos(RecursosID, Nome, Validade)
Values (4, 'Seringas', '2017-01-02');
```

```
INSERT INTO Recursos(RecursosID, Nome, Validade)
Values (5, 'Creme', '2019-11-30');
```

```
INSERT INTO Ocorrencia(OcorrenciaID, Tipo, UtenteID, Descricao)
Values (1, 'Urgência', 3, 'Irritação no pé');
```

```
INSERT INTO Ocorrencia(OcorrenciaID, Tipo, UtenteID, Descricao)
Values (2, 'Normal', 3, 'Dor no peito');
```

```
INSERT INTO Ocorrencia(OcorrenciaID, Tipo, UtenteID, Descricao)
Values (3, 'Normal', 2, 'Dor de cabeça');
```

```
INSERT INTO Ocorrencia(OcorrenciaID, Tipo, UtenteID, Descricao)
Values (4, 'Urgência', 4, 'Braço partido');
```

```
INSERT INTO Marcacao(MarcacaoID, OcorrenciaID, ServicoSaudeID, AreaClinicaID,
MedicoID, Dia, Hora, SalaID, RecursosID)
Values (1, 1, 5, 3, 2, '2015-03-24', '22:00', 4, 5);
```

```
INSERT INTO Marcacao(MarcacaoID, OcorrenciaID, ServicoSaudeID, AreaClinicaID,
MedicoID, Dia, Hora, SalaID, RecursosID)
Values (2, 1, 8, 4, 1, '2015-03-24', '22:00', 2, 3);
```

```
INSERT INTO Marcacao(MarcacaoID, OcorrenciaID, ServicoSaudeID, AreaClinicaID,
MedicoID, Dia, Hora, SalaID, RecursosID)
Values (3, 3, 3, 1, 4, '2016-08-02', '14:00', 3, 2);
```

```
INSERT INTO Marcacao(MarcacaoID, OcorrenciaID, ServicoSaudeID, AreaClinicaID,
MedicoID, Dia, Hora, SalaID, RecursosID)
Values (4, 3, 7, 2, 3, '2016-08-02', '15:00', 1, 4);
```

```
INSERT INTO Marcacao(MarcacaoID, OcorrenciaID, ServicoSaudeID, AreaClinicaID,
MedicoID, Dia, Hora, SalaID, RecursosID)
Values (5, 2, 4, 6, 5, '2015-12-25', '18:00', 5, 1);
```

```
INSERT INTO EpisodioClinico(EpisodioClinicoID, MarcacaoID, OcorrenciaID, Descricao)
Values (1, 3, 3, 'Encaminhamento Neurologia');
```

```

INSERT INTO EpisodioClinico(EpisodioClinicoID, MarcacaoID, OcorrenciaID, Descricao)
Values (2, 4, 3, 'Foi receitado Brufen');

INSERT INTO EpisodioClinico(EpisodioClinicoID, MarcacaoID, OcorrenciaID, Descricao)
Values (3, 5, 2, 'Será operado ao coração');

INSERT INTO EpisodioClinico(EpisodioClinicoID, MarcacaoID, OcorrenciaID, Descricao)
Values (4, 1, 1, 'Análises ao sangue');

INSERT INTO EpisodioClinico(EpisodioClinicoID, MarcacaoID, OcorrenciaID, Descricao)
Values (5, 2, 1, 'Eczema');


INSERT INTO MeiosComplementares(MeiosComplementaresID, EpisodioClinicoID, Nome,
Resultado)
Values (1, 3, 'Operação', 'Óbito');

INSERT INTO MeiosComplementares(MeiosComplementaresID, EpisodioClinicoID, Nome,
Resultado)
Values (2, 4, 'Análise', 'Nada a relatar');


INSERT INTO AtribuicaoServicoArea(ServicoSaudeID, AreaClinicaID)
Values (1, 6);

INSERT INTO AtribuicaoServicoArea(ServicoSaudeID, AreaClinicaID)
Values (1, 5);

INSERT INTO AtribuicaoServicoArea(ServicoSaudeID, AreaClinicaID)
Values (2, 1);

INSERT INTO AtribuicaoServicoArea(ServicoSaudeID, AreaClinicaID)
Values (2, 2);

INSERT INTO AtribuicaoServicoArea(ServicoSaudeID, AreaClinicaID)
Values (3, 7);

INSERT INTO AtribuicaoServicoArea(ServicoSaudeID, AreaClinicaID)
Values (3, 1);

INSERT INTO AtribuicaoServicoArea(ServicoSaudeID, AreaClinicaID)
Values (4, 7);

INSERT INTO AtribuicaoServicoArea(ServicoSaudeID, AreaClinicaID)
Values (4, 6);

INSERT INTO AtribuicaoServicoArea(ServicoSaudeID, AreaClinicaID)
Values (4, 5);

INSERT INTO AtribuicaoServicoArea(ServicoSaudeID, AreaClinicaID)
Values (5, 7);

INSERT INTO AtribuicaoServicoArea(ServicoSaudeID, AreaClinicaID)
Values (5, 3);

INSERT INTO AtribuicaoServicoArea(ServicoSaudeID, AreaClinicaID)
Values (6, 6);

```

```

INSERT INTO AtribuicaoServicoArea(ServicoSaudeID, AreaClinicaID)
Values (6, 4);

INSERT INTO AtribuicaoServicoArea(ServicoSaudeID, AreaClinicaID)
Values (7, 2);

INSERT INTO AtribuicaoServicoArea(ServicoSaudeID, AreaClinicaID)
Values (8, 5);

INSERT INTO AtribuicaoServicoArea(ServicoSaudeID, AreaClinicaID)
Values (8, 4);


INSERT INTO AtribuicaoHospitalServico(HospitalID, ServicoSaudeID)
Values (1, 1);

INSERT INTO AtribuicaoHospitalServico(HospitalID, ServicoSaudeID)
Values (1, 2);

INSERT INTO AtribuicaoHospitalServico(HospitalID, ServicoSaudeID)
Values (1, 4);

INSERT INTO AtribuicaoHospitalServico(HospitalID, ServicoSaudeID)
Values (2, 7);

INSERT INTO AtribuicaoHospitalServico(HospitalID, ServicoSaudeID)
Values (2, 8);

INSERT INTO AtribuicaoHospitalServico(HospitalID, ServicoSaudeID)
Values (2, 3);

INSERT INTO AtribuicaoHospitalServico(HospitalID, ServicoSaudeID)
Values (3, 4);

INSERT INTO AtribuicaoHospitalServico(HospitalID, ServicoSaudeID)
Values (3, 5);

INSERT INTO AtribuicaoHospitalServico(HospitalID, ServicoSaudeID)
Values (3, 8);

INSERT INTO AtribuicaoHospitalServico(HospitalID, ServicoSaudeID)
Values (4, 1);

INSERT INTO AtribuicaoHospitalServico(HospitalID, ServicoSaudeID)
Values (4, 3);

```