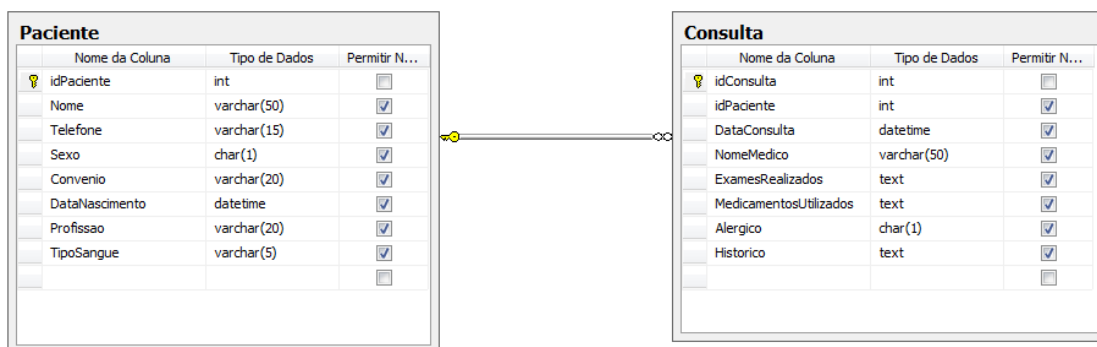


SISTEMA DE ATENDIMENTO PARA CLÍNICA MÉDICA

1. Descrição do Sistema

- O Sistema de Atendimento para Clínica Médica deverá ser capaz de cadastrar os Novos Pacientes e em seguida registrar a Consulta Médica durante o Atendimento ao Paciente previamente cadastrado;
- Um Paciente deve possuir um ID, seu Nome Completo, um Telefone celular para contato, identificação do Sexo (M = Masculino ou F = Feminino), um Convênio que deseja atendimento (Unimed, São Lucas, Saúde Bradesco ou Amil), a Data de Nascimento, Profissão que desempenha (Advogado, Analista de Sistemas, Biólogo, Comerciante, Engenheiro Civil, Jornalista, Outros), e o Tipo Sanguíneo (A+, B+, AB+, O+, A-, B-, AB-, ou O-);
- Durante o Atendimento o sistema deverá registrar o ID do Paciente, a Data da Realização do Atendimento, selecionar o nome de UM dos Médicos que atendem na Clínica (Dr. Alexandre, Dra. Cássia, Dr. Braga ou Dr. Rubens), selecionar os Exames que foram realizados no Paciente durante o Atendimento (Lista de Exames possíveis: Audiometria, Ecografia Tridimensional, Ecodopler, Eletrocardiograma, Exame Laboratorial Completo, Raio X do Torax, Raio X Abdômen, Ultra-sonografia), selecionar os Medicamentos que são utilizados pelo Paciente (Lista de Medicamentos possíveis: Amoxicilina, Aspirina, Complexo B, Diclofenaco Sódico, Dipirona, Insulina, Ibuprofeno, Nimesulida, Paracetamol, Polivitamínico), informar se o Paciente é Alérgico (Sim ou Não), e registrar o Histórico do Atendimento em um campo de texto livre.

2. Diagrama de Estrutura de Dados (DED)



3. Scripts SQL

Criar um NOVO Banco de Dados no SQL Server e dar o nome de **[dbAtendimento]**.

Script SQL para Tabela **Usuario**

```
CREATE TABLE [dbo].[Usuario](
    [IdUsuario] [int] NOT NULL,
    [Username] [varchar](20) NULL,
    [Password] [varchar](20) NULL,
    PRIMARY KEY CLUSTERED
(
    [IdUsuario] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO
```

Script SQL para Tabela **Paciente**

```
CREATE TABLE [dbo].[Paciente](
    [idPaciente] [int] NOT NULL,
    [Nome] [varchar](50) NULL,
    [Telefone] [varchar](15) NULL,
    [Sexo] [char](1) NULL,
    [Convenio] [varchar](20) NULL,
    [DataNascimento] [datetime] NULL,
    [Profissao] [varchar](20) NULL,
    [TipoSangue] [varchar](5) NULL,
    CONSTRAINT [PK_Paciente] PRIMARY KEY CLUSTERED
(
    [idPaciente] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO

SET ANSI_PADDING OFF
GO
```

Script SQL para Tabela **Consulta**

```
CREATE TABLE [dbo].[Consulta](
    [idConsulta] [int] NOT NULL,
    [idPaciente] [int] NULL,
    [DataConsulta] [datetime] NULL,
    [NomeMedico] [varchar](50) NULL,
    [ExamesRealizados] [text] NULL,
```

```
[MedicamentosUtilizados] [text] NULL,  
[Alergico] [char](1) NULL,  
[Historico] [text] NULL,  
CONSTRAINT [PK_Consulta] PRIMARY KEY CLUSTERED  
(  
    [idConsulta] ASC  
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY  
= OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]  
  
GO  
  
SET ANSI_PADDING OFF  
GO  
  
ALTER TABLE [dbo].[Consulta] WITH CHECK ADD CONSTRAINT  
[FK_Consulta_Paciente] FOREIGN KEY([idPaciente])  
REFERENCES [dbo].[Paciente] ([idPaciente])  
GO  
  
ALTER TABLE [dbo].[Consulta] CHECK CONSTRAINT [FK_Consulta_Paciente]  
GO
```

4. Telas do Sistema

- **1º Passo:** Criar um NOVO Projeto no Visual Studio ([ProjetoClinicaMedica](#));
- **2º Passo:** Criar o Banco de Dados ([dbAtendimento](#)) que deverá ser salvo na raiz do projeto;
- **3º Passo:** Organizar o Projeto em três pacotes (pastas ou namespaces):
 - 1) Pacote [conexao](#) contendo a classe Conexao;
 - 2) Pacote [resources](#) contendo as imagens utilizadas no projeto (as Imagens são de livre escolha do aluno);
 - 3) Pacote [visao](#) contendo os seguintes Formulários:
 - a. Formulário SPLASH ([FormSplash](#)) para abertura do Sistema.
 - b. Formulário de LOGIN ([FormLogin](#)) para autenticar o acesso ao Sistema de Atendimento para Clínica Médica.
 - c. Formulário Principal ([FormPrincipal](#)) contendo um componente **MenuStrip** contendo as opções para acesso aos Formulários de Paciente e Atendimento, e opção de **Sair** do sistema.

- d. Formulário para cadastro de Pacientes (**FormPaciente**) contendo os seguintes componentes:

- **Label** - um para cada campo da Tabela Paciente;
- **TextBox** - para os campos: ID, Nome, Telefone;
- **RadioButton** - para os campos: Sexo; Profissao, TipoSangue;
- **ComboBox** - para o campo Convenio;
- **MonthCalendar** - para o campo DataNascimento;
- **Button** – para implementação dos métodos SQL: btnInserir, btnSelecionar, btnExcluir, btnAtualizar e btnLimpar;
- **DataGridView** – para listar os Pacientes Cadastrados

- e. Formulário para registro dos Atendimentos (**FormAtendimento**) contendo os seguintes componentes:

- **Label** - um para cada campo da Tabela Consulta;
- **TextBox** - para os campos: ID da Consulta e ID do Paciente, e Historico (propriedade Multiline = true); e para o Paciente deverá ser recuperado o Nome e a Idade (a ser calculada com base na DataNascimento);
- **RadioButton** - para o campo: Alergico;
- **ComboBox** - para o campo NomeMedico;
- **MonthCalendar** - para o campo DataConsulta;
- **CheckedListBox** – para os campos: ExamesRealizados e MedicamentosUtilizados;
- **Button** – para implementação dos métodos SQL: btnInserir, btnSelecionar, btnExcluir, btnAtualizar e btnLimpar;
- **DataGridView** – para listar os Atendimentos Registrados

- **3º Passo:** Salvar o Projeto (compactado em zip ou rar) e enviar para o email professor (joao.moraes@unilins.edu.br) – Não esquecer de remover as pastas BIN e OBJ antes de compactar e enviar!