**CENTRO UNIVERSITÁRIO RUY BARBOSA**

ANDRÉ FELIPE XAVIER OLIVEIRA SANTOS

BRUNO GABRIEL FIGUEIREDO DA PAZ ASSIS

**BANCO DE DADOS: SISTEMA DE ATENDIMENTO AO PACIENTE**

Salvador/BA

2024

**CENTRO UNIVERSITÁRIO RUY BARBOSA**

ANDRÉ FELIPE XAVIER OLIVEIRA SANTOS

BRUNO GABRIEL FIGUEIREDO DA PAZ ASSIS

**BANCO DE DADOS: SISTEMA DE ATENDIMENTO AO PACIENTE**

Trabalho de banco de dados apresentado ao docente Heleno Cardoso, como parte dos requisitos para obtenção de nota.

Salvador/BA

2024

**SUMÁRIO**

[**1 INTRODUÇÃO** 4](#_Toc168434204)

[**2 MODELO CONCEITUAL** 5](#_Toc168434205)

[**3 MODELO LÓGICO** 6](#_Toc168434206)

[**3.1 Entidade Paciente:** 6](#_Toc168434207)

[**3.2 Entidade Medico:** 7](#_Toc168434208)

[**3.3 Entidade Especialidade:** 7](#_Toc168434209)

[**3.4 Entidade Atendimento Paciente:** 7](#_Toc168434210)

[**3.5 Entidade Tipo de atendimento:** 7](#_Toc168434211)

[**4 MODELO FÍSICO** 8](#_Toc168434212)

[**5 CONCLUSÃO** 14](#_Toc168434213)

[**6 REFERÊNCIAS BIBLIOGRAFICAS** 15](#_Toc168434214)

# **1 INTRODUÇÃO**

Este trabalho tem como objetivo desenvolver um modelo de banco de dados de um sistema de marcação de exames, que vai a partir da criação do modelo conceitual até o banco de dados físico.

# **2 MODELO CONCEITUAL**

O sistema de marcação de exames é projetado para gerenciar estas marcações, de modo a incluir informações referentes ao médico, paciente e ao próprio atendimento.

Ele tem como seu principal objetivo a organização e controle dessas informações de marcação de exame, de forma a garantir que os pacientes possam agendar seus exames eficientemente e que os médicos tenham acesso a estas informações para gerir sua agenda de uma melhor forma, otimizando os seus atendimentos.

Diagrama

Descrição gerada automaticamente

Imagem 1: Modelo conceitual (autoral)

Neste modelo conceitual temos algumas cardinalidades, são elas:

- Um paciente pode ter múltiplos atendimentos (relação 1 ... N)

- Um médico pode realizar múltiplos atendimentos (relação 1...N)

- Cada atendimento está associado a um único médico e a um único paciente (relação 1...1)

- Um médico terá apenas uma especialidade (relação 1...1)

- A mesma especialidade pode ter diversos médicos (relação 1...N)

- Um atendimento terá apenas um tipo (relação 1...1)

- Porém podemos ter diversos atendimentos com o mesmo tipo (relação 1...N)

# **3 MODELO LÓGICO**

Como modelo lógico temos uma representação de todas as entidades e cardinalidades que temos no nosso BD.

Diagrama

Descrição gerada automaticamente

Imagem 2: Modelo lógico (autoral)

Em relação as entidades e seus atributos, temos:

## **3.1 Entidade Paciente:**

Na entidade Paciente teremos um id AI (Auto Increment) como Primary Key que será usado para diferenciar um paciente cadastrado de outro. Além desse id, iremos ter dados como nome (varchar(100)), cpf (varchar(11)), telefone (varchar(11)) e email (varchar(50)).

Optamos por utilizar varchar no cpf e no telefone, mesmo eles sendo números, devido a eficiência de memória, pois como não iremos ter nenhuma operação matemática os envolvendo, não se faz necessário ser de um tipo number.

## **3.2 Entidade Medico:**

Na entidade Medico teremos uma estrutura parecida com a entidade paciente, com um id AI como Primary Key, nome do médico também do tipo varchar, crm (varchar(11)) que é o registro do médico, o código da especialidade (int) sendo uma FK (Foreign Key), telefone e email.

O motivo do crm é o mesmo caso do cpf e telefone da entidade paciente e optamos por criar uma entidade de especialidade para que esses dados fossem organizados de uma maneira mais simples, não precisando digitar especialidade por especialidade a cada cadastro de médico e sim apenas o seu código.

## **3.3 Entidade Especialidade:**

Na entidade especialidade teremos um Código AI como Primary Key, uma descrição especialidade (varchar(45)) e um ie\_situacao (varchar(1)).

Este ie\_situacao irá adotar dois tipos de valores, ‘A’ quando ele estiver ativo e ‘I’ quando estiver inativo. Ele será uma forma simples de identificar na tabela se essa especialidade está ativa ou não.

## **3.4 Entidade Atendimento Paciente:**

Na entidade Atendimento Pacienteteremos como Primary Key o número do atendimento (Nr\_atendimento), sendo ele um inteiro AI. Está entidade terá três FK, sendo elas o id do paciente, o id do médico e o tipo de atendimento. Além disso, temos a data de entrada (dt\_entrada) e data de alta do paciente, sendo os dois do tipo datetime**.**

Nesta FK tipo de atendimento usamos a mesma lógica da especialidade na entidade médico. Outrossim, optamos por utilizar as datas de entrada e saída do paciente como um tipo datetime, pois ter a data e principalmente a hora que o paciente da a entrada e a saída irá ser importante para levantar dados de tempo médio de atendimento.

## **3.5 Entidade Tipo de atendimento:**

Fizemos esta entidade seguindo o mesmo conceito da entidade Especialidade. Ela irá ter um id AI PK do tipo int, uma descrição do tipo varchar e um ie situação para identificar se este tipo de atendimento está ativo ou não.

# **4 MODELO FÍSICO**

O modelo físico se refere a implementação do banco de dados em algum Sistema de Gerenciamento de Banco de Dados (SGBD). O escolhido por nossa equipe foi o MySQL.

Código para gerar o BD:

-- MySQL Workbench Forward Engineering

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='ONLY\_FULL\_GROUP\_BY,STRICT\_TRANS\_TABLES,NO\_ZERO\_IN\_DATE,NO\_ZERO\_DATE,ERROR\_FOR\_DIVISION\_BY\_ZERO,NO\_ENGINE\_SUBSTITUTION';

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

-- -----------------------------------------------------

-- Schema mydb

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;

USE `mydb` ;

-- -----------------------------------------------------

-- Table `mydb`.`Paciente`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `mydb`.`Paciente` (

`Id\_paciente` INT NOT NULL AUTO\_INCREMENT,

`dt\_nascimento` DATE NOT NULL,

`nm\_paciente` VARCHAR(100) NOT NULL,

`cpf` VARCHAR(11) NOT NULL,

`telefone` VARCHAR(11) NULL,

`email` VARCHAR(50) NULL,

PRIMARY KEY (`Id\_paciente`))

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`Especialidade`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `mydb`.`Especialidade` (

`cd\_especialidade` INT NOT NULL AUTO\_INCREMENT,

`especialidade` VARCHAR(45) NOT NULL,

`ie\_situacao` VARCHAR(1) NOT NULL,

PRIMARY KEY (`cd\_especialidade`),

UNIQUE INDEX `cd\_especialidade\_UNIQUE` (`cd\_especialidade` ASC) VISIBLE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`Medico`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `mydb`.`Medico` (

`id\_medico` INT NOT NULL AUTO\_INCREMENT,

`crm` VARCHAR(11) NOT NULL,

`Nm\_medico` VARCHAR(100) NOT NULL,

`cd\_especialidade` INT NOT NULL,

`telefone` VARCHAR(45) NULL,

`email` VARCHAR(100) NULL,

PRIMARY KEY (`id\_medico`),

UNIQUE INDEX `crm\_UNIQUE` (`crm` ASC) VISIBLE,

INDEX `cd\_especialidade\_idx` (`cd\_especialidade` ASC) VISIBLE,

CONSTRAINT `cd\_especialidade`

FOREIGN KEY (`cd\_especialidade`)

REFERENCES `mydb`.`Especialidade` (`cd\_especialidade`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`tipo\_atendimento`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `mydb`.`tipo\_atendimento` (

`idtipo\_atendimento` VARCHAR(2) NOT NULL,

`Desc\_tipo\_atendimento` VARCHAR(120) NOT NULL,

`ie\_situacao` VARCHAR(1) NOT NULL,

PRIMARY KEY (`idtipo\_atendimento`),

UNIQUE INDEX `idtipo\_atendimento\_UNIQUE` (`idtipo\_atendimento` ASC) VISIBLE)

ENGINE = InnoDB;

-- -----------------------------------------------------

-- Table `mydb`.`Atendimento\_paciente`

-- -----------------------------------------------------

CREATE TABLE IF NOT EXISTS `mydb`.`Atendimento\_paciente` (

`Nr\_sequencia` INT NOT NULL,

`Nr\_atendimento` INT NOT NULL AUTO\_INCREMENT,

`id\_paciente` INT NOT NULL,

`id\_medico` INT NOT NULL,

`Dt\_entrada` DATETIME NOT NULL,

`Dt\_alta` DATETIME NOT NULL,

`tipo\_atendimento` VARCHAR(2) NOT NULL,

PRIMARY KEY (`Nr\_sequencia`),

INDEX `id\_paciente\_idx` (`id\_paciente` ASC) VISIBLE,

INDEX `id\_medico\_idx` (`id\_medico` ASC) VISIBLE,

INDEX `idtipo\_atendimento\_idx` (`tipo\_atendimento` ASC) VISIBLE,

CONSTRAINT `id\_paciente`

FOREIGN KEY (`id\_paciente`)

REFERENCES `mydb`.`Paciente` (`Id\_paciente`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `id\_medico`

FOREIGN KEY (`id\_medico`)

REFERENCES `mydb`.`Medico` (`id\_medico`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `idtipo\_atendimento`

FOREIGN KEY (`tipo\_atendimento`)

REFERENCES `mydb`.`tipo\_atendimento` (`idtipo\_atendimento`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

SET SQL\_MODE=@OLD\_SQL\_MODE;

SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

Este código foi gerado pelo próprio mysql usando a ferramenta Forward Engineer ou o atalho Ctrl + G.

Interface gráfica do usuário, Aplicativo

Descrição gerada automaticamente

Imagem 3: Ferramenta Forward Enginner (autoral)

# **5 CONCLUSÃO**

Neste trabalho, desenvolvemos um banco de dados para um sistema de marcação de exames, passando pelas etapas de modelo conceitual, lógico e físico. A realização deste projeto permitiu o aprofundamento em técnicas de modelagem de banco de dados, compreensão das relações entre entidades e implementação prática no MySQL. Este aprendizado será valioso para futuros projetos de banco de dados.

# **6 REFERÊNCIAS BIBLIOGRAFICAS**

Aws. Qual é a diferença entre um modelo de dados lógico e um modelo de dados físico?. **AWS**. Disponível em: < <https://aws.amazon.com/pt/compare/the-difference-between-logical-and-physical-data-model/>>. Acesso em: 02/06/2024.

LOPES, Bergson. Modelo Conceitual de Dados - Aprenda a utilizar os principais mecanismos de abstração. **BLR DATA**. Disponível em: < https://www.blrdata.com.br/single-post/2016/03/19/modelo-conceitual-de-dados-aprenda-a-utilizar-os-principais-mecanismos-de-abstra%C3%A7%C3%A3o>. Acesso em: 02/06/2024.