**Sumário**

* **Introdução, objetivo, situação problemática e modelo de negócio página (2)**
* **Diagrama de entidade e relacionamento (DER), página (3)**

[**https://drive.google.com/file/d/1o6ZtDjBi1ZFFoUeXKu3e0\_-R4J7zySes/view?usp=sharing**](https://drive.google.com/file/d/1o6ZtDjBi1ZFFoUeXKu3e0_-R4J7zySes/view?usp=sharing)

* **Relatórios criados e ferramentas utilizadas, pagina (5)**
* **Views, página (6)**

[**https://drive.google.com/file/d/1CreIAycHENSLyOcIC-U0J6ikpzW9AfVS/view?usp=sharing**](https://drive.google.com/file/d/1CreIAycHENSLyOcIC-U0J6ikpzW9AfVS/view?usp=sharing)

* **Functions, página (9)** [**https://drive.google.com/file/d/1XEre5b5bhmWV7W2A8Wb9V0YOaMrR25ef/view?usp=sharing**](https://drive.google.com/file/d/1XEre5b5bhmWV7W2A8Wb9V0YOaMrR25ef/view?usp=sharing)
* **Stored procedure, página (11)** [**https://drive.google.com/file/d/1K1PKlrWa225WTcSROajoav56mLBRPz1K/view?usp=sharing**](https://drive.google.com/file/d/1K1PKlrWa225WTcSROajoav56mLBRPz1K/view?usp=sharing)
* **Triggers, página (13)** [**https://drive.google.com/file/d/1IShJVEGBk5lg0loP8XiHuAFY803tSX1y/view?usp=sharing**](https://drive.google.com/file/d/1IShJVEGBk5lg0loP8XiHuAFY803tSX1y/view?usp=sharing)
* **Funções criadas, página (15)** [**https://drive.google.com/file/d/1yd3dlU9DxTnmYdfTFzmeRINEIm4wmeOv/view?usp=sharing**](https://drive.google.com/file/d/1yd3dlU9DxTnmYdfTFzmeRINEIm4wmeOv/view?usp=sharing)

**Introdução**

Este projeto de banco de dados tem o titulo “**Clinica Lavier**”. Foi criado para armazenar informações de pacientes, médicos e consultas, tanto para cadastro como para controle de agendamentos, armazenamento de dados e informações passadas. Com este sistema de gerenciamento é possível evitar erros operacionais, evitar agendas duplicadas, e deixar o atendimento pratico e assertivo.

**Objetivo**

Desenvolver um banco de dados relacional capaz de armazenar, organizar e gerenciar os dados operacionais da Clínica Lavier. O sistema permitirá controlar informações de pacientes, médicos e suas respectivas consultas, garantindo maior agilidade, segurança e facilidade na manipulação dos dados. Este sistema é interfuncional pois além de proporcionar praticidade no controle do atendimento, o administrativo é beneficiado por organizar dados sobre pacientes e médicos, parte contábil consegue manter uma exatidão na quantidade de consultas realizadas por cada medico para assim fazerem os pagamentos devidos e a área analítica consegue ter uma visão ampla de como estão os atendimentos, volume de consultas, entrada e saída de pacientes para assim darem feedback para melhorias.

**Situação problemática**

Atualmente, a Clínica Lavier realiza o controle de seus atendimentos e cadastros de forma manual, o que pode gerar retrabalho, perda de informações e falta de organização. Isso compromete a eficiência no atendimento aos pacientes, na agenda médica e no acompanhamento dos históricos clínicos.

**Modelo de negócio**

O modelo de negócio da Clínica Lavier é baseado na prestação de serviços médicos em diversas especialidades. A clínica realiza cadastros de pacientes, consultas com diferentes médicos, diagnósticos e acompanhamento clínico. O sistema foi modelado para permitir; Cadastro de pacientes, cadastro de médicos e suas especialidades, registro das consultas médicas, contendo sintomas e diagnósticos de uma forma completamente pratica, correta e segura. Assim ganharemos tempo na inclusão ou exclusão de pacientes, faremos análises mais eficientes e rápidas, evitará perdas de informações cadastrais, entre outros...

**DER (Diagrama Entidade-Relacionamento)**

**Lista de tabelas, colunas, tipos de dados e chaves.**

Entidades, Campos e relacionamento.

As entidades são:

**Pacientes:** que armazena os dados dos pacientes

**Médicos:** Contém informações dos profissionais da saúde

**Consulta**: Registra os atendimentos realizados, relacionando pacientes e médicos.

**Dentro de cada entidade temos os campos, tipo de dados e a chave, respectivamente;**

“**Paciente**” =

**id\_paciente,** tipo de dados **INT(inteiro)** sendo **PRIMARY KEY**.

**nome** tipo de dados **VARCHAR**

**data\_nascimento** tipo de dados **DATE**

**cpf** tipo de dados **VARCHAR**

**telefone** tipo de dados **VARCHAR**

“**Médicos**” =

**id\_medico** tipo de dados  **INT** sendo **PRIMARY KEY**

**nome** tipo de dados  **VARCHAR**

**especialidade** tipo de dados **VARCHAR**

**crm** tipo de dados **VARCHAR**

**telefone** tipo de dados **VARCHAR**

“**Consultas**” =

**id\_consulta** tipo de dados **INT PRIMARY KEY**

**id\_paciente** tipo de dados  **INT**

**id\_medico** tipo de dados **INT**

**sintomas** tipo de dados **TEXT**

**diagnostico** tipo de dados **TEXT**

**FOREIGN KEY** (id\_paciente) **REFERENCES** pacientes(id\_paciente),

**FOREIGN KEY** (id\_medico) **REFERENCES** medicos(id\_medico)

**RELACIONAMENTO**

Um Paciente para varias consultas (um para muitos)

Uma consulta para um paciente (um para um)

Um medico para varias consultas (um para muitos).

**Neste link está o script.sql completo com a criação de todas as tabelas, todas as inserções, views, functions, stored procedure, triggers e as funções criadas. Este banco de dados recebe o titulo + extensão de “CLINICA\_LAVIER.SQL"**

[**https://drive.google.com/file/d/1aKGXGUA2VGUUOacP6GmdPotfGOhaSRad/view?usp=sharing**](https://drive.google.com/file/d/1aKGXGUA2VGUUOacP6GmdPotfGOhaSRad/view?usp=sharing)

**Neste link está somente as tabelas e inserções do banco de dados:** [**https://drive.google.com/file/d/1o6ZtDjBi1ZFFoUeXKu3e0\_-R4J7zySes/view?usp=sharing**](https://drive.google.com/file/d/1o6ZtDjBi1ZFFoUeXKu3e0_-R4J7zySes/view?usp=sharing)

**Relatórios criados.**

**Lista de Pacientes Cadastrados:**  
Relatório com nome, CPF, telefone e data de nascimento dos pacientes.

**Lista de Médicos e Suas Especialidades:**  
Relatório que apresenta os médicos cadastrados, com suas especialidades e CRM.

**Consultas Realizadas:**  
Relatório contendo ID da consulta, nome do paciente, nome do médico, sintomas e diagnóstico.

**Consultas por Médico:**  
Permite visualizar todas as consultas feitas por determinado médico, ajudando na organização da agenda e histórico clínico.

**Consultas por Paciente:**  
Relatório que exibe o histórico de consultas de um paciente específico, com seus sintomas e diagnósticos.

Entre outros...

**Ferramentas utilizadas**

As ferramentas utilizadas para criação do projeto foram; MySQL workbench, google drive e word.

VIEWS

**1- view informacoes\_consultas**

**CREATE VIEW informacoes\_consulta AS**

**SELECT**

**c.id\_consulta,**

**p.nome AS nome\_paciente,**

**p.data\_nascimento,**

**p.cpf,**

**p.telefone AS telefone\_paciente,**

**m.nome AS nome\_medico,**

**m.especialidade,**

**m.crm,**

**m.telefone AS telefone\_medico,**

**c.sintomas,**

**c.diagnostico**

**FROM**

**consultas c**

**JOIN**

**pacientes p ON c. id\_paciente = p.id\_paciente**

**JOIN**

**medicos m ON c.id\_medico = m.id\_medico;**

**SELECT \* FROM informacoes\_consulta;**

**CREATE VIEW vw\_consultas\_especialidade AS**

**SELECT**

**m.especialidade,**

**COUNT(c.id\_consulta) AS total\_consultas**

**FROM consultas c**

**JOIN medicos m ON c.id\_medico = m.id\_medico**

**GROUP BY m.especialidade;**

**Objetivo:** objetivo é relacionar as três tabelas me trazendo o nome do paciente, com qual medico ele passou, o sintomas e diagnostico.

Esta view tem como objetivo puxar os dados completo da consulta, desde o id , nome do doutor na qual atendeu o paciente com todos os dados de ambos incluindo os sintomas e diagnostico do paciente. Neste caso as três tabelas estão se relacionando (**Consultas, medico, paciente).**

**2-View lista\_pacientes**

**CREATE VIEW lista\_pacientes AS**

**SELECT id\_paciente, nome, cpf, telefone**

**FROM pacientes;**

**Objetivo:** Esta view o objetivo é fornecer uma lista resumida dos pacientes cadastrados, mostrando apenas as informações essenciais para identificação e contato, como CPF, NOME e telefone.

**3-View lista\_medicos**

**CREATE VIEW lista\_medicos AS**

**SELECT id\_medico, nome, especialidade, crm, telefone**

**FROM medicos;**

**Objetivo:** View criada para a finalidade de identificação de forma mais rápida referente as informações dos especialistas, incluindo CRM, NOME, Telefone e especialidade do medico.

**4-View medico\_especialidade**

**CREATE VIEW medico\_especialidade AS**

**SELECT nome, especialidade**

**FROM medicos;**

**Objetivo:** Exibir uma relação simplificada dos médicos e suas especialidades, útil para consultas rápidas sem necessidade de outros dados.

**5-View consultas\_inf\_basicas**

**CREATE VIEW consultas\_inf\_basicas AS**

**SELECT id\_consulta, id\_paciente, id\_medico**

**FROM consultas;**

**Objetivo:** nesta view o objetivo é ter uma visão básica das consultas realizadas, mostrando a identificação da consulta e os envolvidos “paciente e médico”.

FUNCTION

**1- function calcular\_idade.**

**DELIMITER $$**

**CREATE FUNCTION calcular\_idade(data\_nasc DATE)**

**RETURNS INT**

**DETERMINISTIC**

**BEGIN**

**RETURN TIMESTAMPDIFF(YEAR, data\_nasc, CURDATE());**

**END$$**

**DELIMITER ;**

**Objetivo:** Esta função foi criada para facilitar a visualização em relacação a idade do paciente. Ela recebe a data de nascimento como entrada e devolve um numero inteiro como idade. Facilita no dia a dia para verificação e busca mais detalhada sobre a vida do paciente. A TABELA utilizada foi a de “**PACIENTES**”.

**2- function total\_consultas\_paciente.**

**DELIMITER $$**

**CREATE FUNCTION total\_consultas\_paciente(paciente\_id INT)**

**RETURNS INT**

**DETERMINISTIC**

**BEGIN**

**DECLARE total INT;**

**SELECT COUNT(\*) INTO total**

**FROM consultas**

**WHERE id\_paciente = paciente\_id;**

**RETURN total;**

**END $$**

**DELIMITER ;**

**Objetivo:** Esta função tem como objetivo retornar o **número total de consultas realizadas por um paciente específico**, utilizando o seu ID como parâmetro. Desta forma conseguimos chegar aos números de forma prática.

**Stored procedure**

**1- Stored procedure adicionar\_paciente.**

**DELIMITER $$**

**CREATE PROCEDURE adicionar\_paciente(**

**IN p\_nome VARCHAR(100),**

**IN p\_data\_nascimento DATE,**

**IN p\_cpf VARCHAR(14),**

**IN p\_telefone VARCHAR(20)**

**)**

**BEGIN**

**INSERT INTO pacientes (nome, data\_nascimento, cpf, telefone)**

**VALUES (p\_nome, p\_data\_nascimento, p\_cpf, p\_telefone);**

**END$$**

**DELIMITER ;**

**Objetivo** : foi criada para **inserir um novo paciente**  na tabela, de forma prática e segura. Ela recebe como parâmetros os dados básicos necessários (nome, data de nascimento, CPF e telefone) e realiza a inserção no banco. Objetivo é para facilitar no dia a dia a inclusão de novos pacientes, levando em conta o movimento na qual a clinica pode ter, este procedimento reduzirá o tempo de trabalho e agilizará o processo.

**2- Stored procedure listar\_pacientes.**

**DELIMITER $$**

**CREATE PROCEDURE listar\_pacientes()**

**BEGIN**

**SELECT**

**id\_paciente,**

**nome,**

**cpf,**

**telefone**

**FROM**

**pacientes;**

**END $$**

**DELIMITER ;**

**Objetivo:** Essa Stored procedure tem o objetivo de fornecer uma **lista simples dos pacientes cadastrados na clínica**, facilitando a consulta de informações básicas, como nome, CPF e telefone, para fins administrativos ou operacionais.

**Triggers**

**1- Trigger padronizar\_nome\_medico**

**DELIMITER $$**

**CREATE TRIGGER padronizar\_nome\_medico**

**BEFORE INSERT ON medicos**

**FOR EACH ROW**

**BEGIN**

**SET NEW.nome = UPPER(NEW.nome);**

**END $$**

**DELIMITER ;**

**Objetivo:**  Com esta trigger todos os médicos estarão garantidos de serem incluídos no banco de dados com letra maiúscula, para melhor visualização e diferenciação.

**2- Trigger preencher\_telefone\_paciente**

**DELIMITER $$**

**CREATE TRIGGER preencher\_telefone\_paciente**

**BEFORE INSERT ON pacientes**

**FOR EACH ROW**

**BEGIN**

**IF NEW.telefone IS NULL OR NEW.telefone = '' THEN**

**SET NEW.telefone = 'Sem telefone';**

**END IF;**

**END $$**

**DELIMITER ;**

**Objetivo:** Nesta trigger se alguém tentar cadastrar um paciente sem número de telefone, terá o texto “ sem telefone” , objetivo principal é termos no mínimo um contato telefônico atualizado em cada cadastro de paciente.

**Funções criadas.**

1-

**SELECT**

**consultas.id\_paciente,**

**pacientes.nome AS nome\_paciente,**

**medicos.nome AS nome\_medico ,**

**medicos.especialidade,**

**consultas.sintomas AS sintomas\_paciente,**

**consultas.diagnostico AS diagnostico\_paciente**

**FROM consultas**

**JOIN pacientes ON consultas.id\_paciente = pacientes.id\_paciente**

**JOIN medicos ON consultas.id\_medico = medicos.id\_medico;**

**Objetivo:** é fazer as tabelas se relacionarem entre sí, me trazendo o id e nome do paciente, seguido do nome do medico e sua especialidade juntamente com o sintoma e diagnostico de cada paciente cadastrado na clinica Lavier.

2-

**SELECT id\_consulta, id\_paciente, sintomas, diagnostico**

**FROM consultas**

**WHERE id\_medico = 3;**

**Objetivo:** Selecionar a consulta feitas pelo medico id numero 3, certamente podendo modificar o id\_medico para conseguir consultar o atendimento de qualquer outro médico que seja.