



**ANHANGUERA**

**Nome: Luis Inacio - RA: 171**

**PORTFÓLIO - ROTEIRO DE AULA PRÁTICA  
PROGRAMAÇÃO E DESENVOLVIMENTO DE BANCO DE DADOS**

**Nome: Luis Inacio - RA: 171**

**PORTFÓLIO - ROTEIRO DE AULA PRÁTICA  
PROGRAMAÇÃO E DESENVOLVIMENTO DE BANCO DE DADOS**

- - Criação da estrutura de um banco de dados (tabelas) com a linguagem SQL por meio de um diagrama entidade-relacionamento pré-definido. - Inserir dados no banco de dados criado. - Consultar os dados armazenados por meio da criação de uma visão (View). - Criar um relatório no final da atividade.

Professor: Anderson Emidio de  
Macedo Goncalves

Etiopia

2024

## SUMÁRIO

<b>1. INTRODUÇÃO.....</b>	<b>4</b>
<b>2. OBJETIVOS GERAIS.....</b>	<b>4</b>
<b>3. PREPARANDO AMBIENTE.....</b>	<b>5</b>
3.1 Instalação MySQL Server e Workbench.....	5
<b>4. MÉTODO.....</b>	<b>7</b>
4.1 Criando a base de dados (Loja).....	7
4.2 Criando as tabelas.....	7
4.3 Criação do script para inserir dados.....	9
4.4 Comandos de consulta (DQL).....	9
4.5 Visualizando a consulta (DQL).....	10
<b>5. CONCLUSÕES.....</b>	<b>10</b>
<b>6. REFERÊNCIAS.....</b>	<b>11</b>

## 1. INTRODUÇÃO

Neste portfólio, apresento uma atividade prática em Programação e Desenvolvimento de Banco de Dados, focada na criação de um sistema de gerenciamento utilizando MySQL Server e MySQL Workbench. O trabalho é estruturado em três etapas principais:

**Criação da Base de Dados:** Desenvolvi a base "Loja" a partir de um diagrama entidade-relacionamento, utilizando comandos DDL e definindo estrutura adequada, incluindo chaves primárias com auto-incremento.

**Inserção de Dados:** Criei um script "inserir.sql" para inserir dados em todas as tabelas da base, garantindo que o sistema estivesse devidamente populado.

**Consultas aos Dados:** Desenvolvi o script "consulta.sql", com uma VIEW que retorna informações sobre contas pendentes de pagamento.

A escolha das ferramentas MySQL Community Server e MySQL Workbench reflete a eficiência e robustez que oferecem. Este portfólio não só documenta as etapas do projeto, mas também serve como um recurso de aprendizado prático sobre bancos de dados e o uso da linguagem SQL.

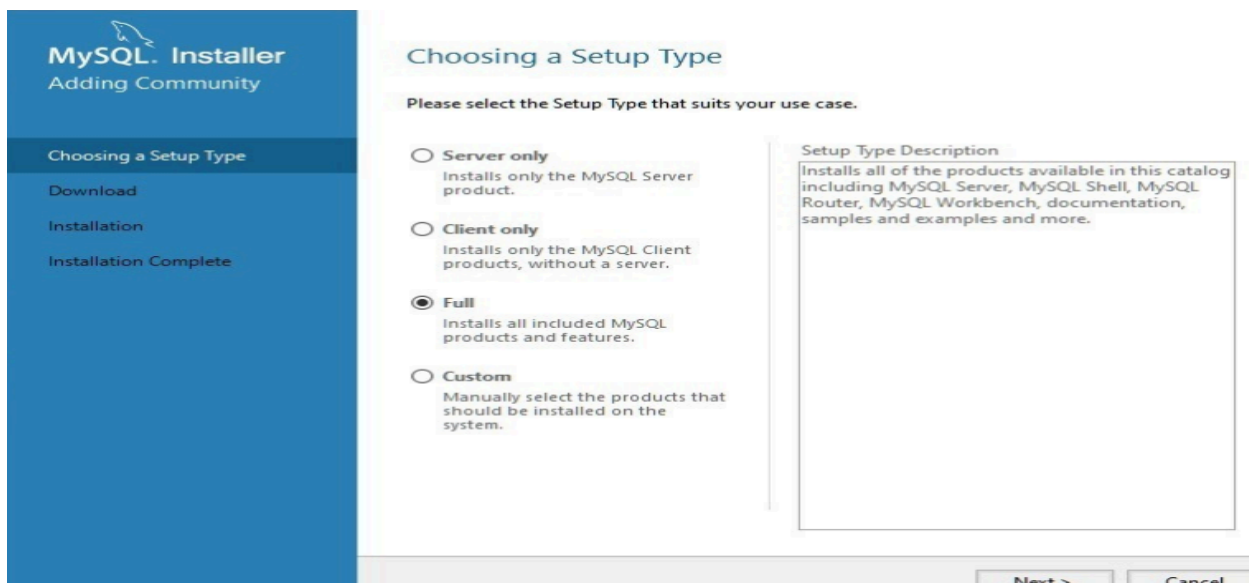
## 2. OBJETIVOS GERAIS

1. - Instalar o sistema MySQL Community Server e MySQL Workbench.
2. - Elaborar o modelo físico (implementação da estrutura) do banco de dados proposto pelo DER no software MySQL Workbench.
3. - Criar o script "inserir.sql" para inserir dados em todas as tabelas criadas.
4. - Elaborar o script "consulta.sql" que irá conter uma visão que retornará todas as contas que ainda não foram pagas.

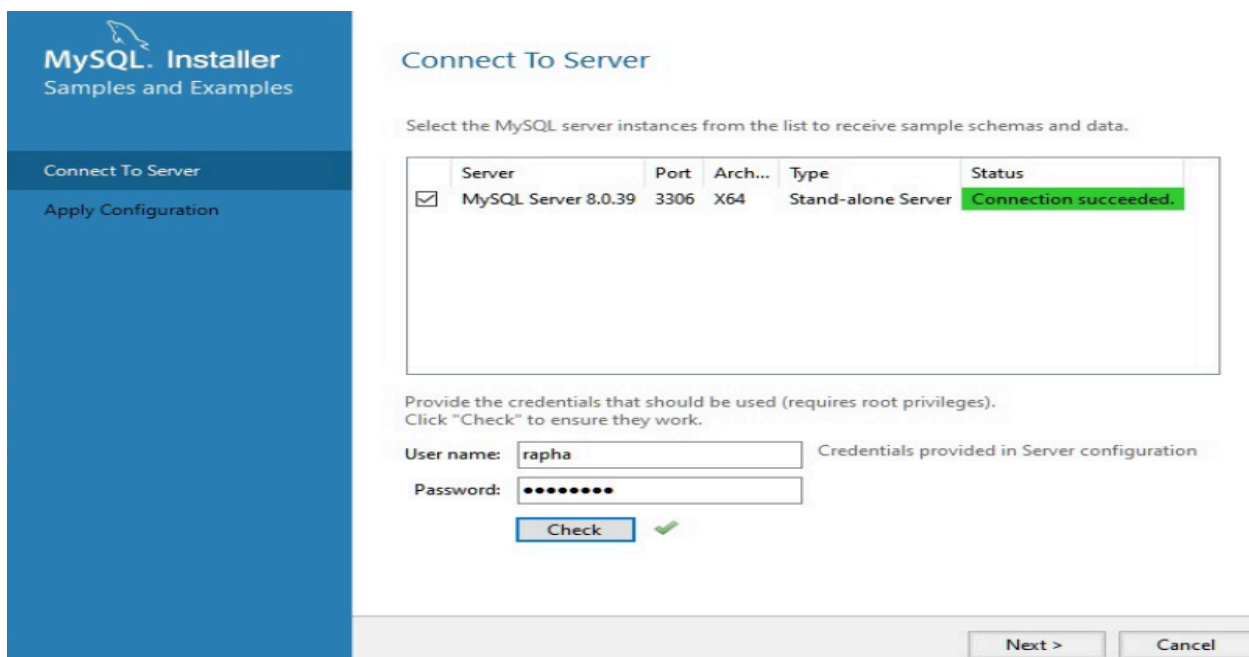
### 3. PREPARANDO AMBIENTE

#### 3.1 Instalação MySQL Server e Workbench

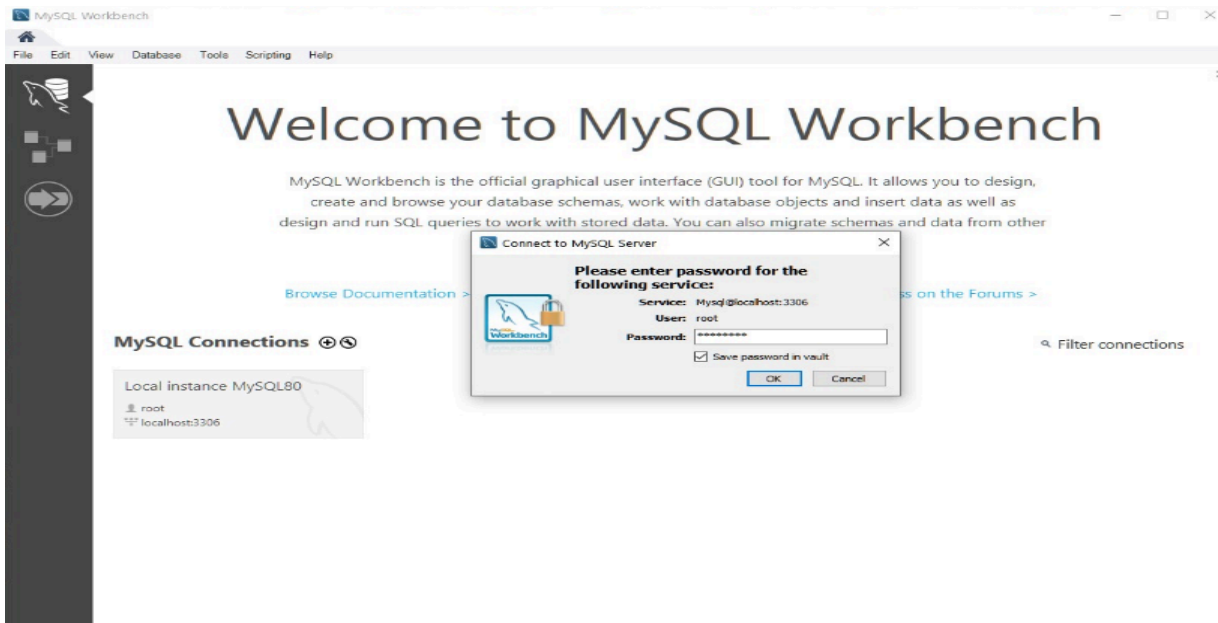
Baixe o MysqlWorkbench no site oficial “<https://dev.mysql.com/downloads/workbench/>”, faça a instalação completa (Full).



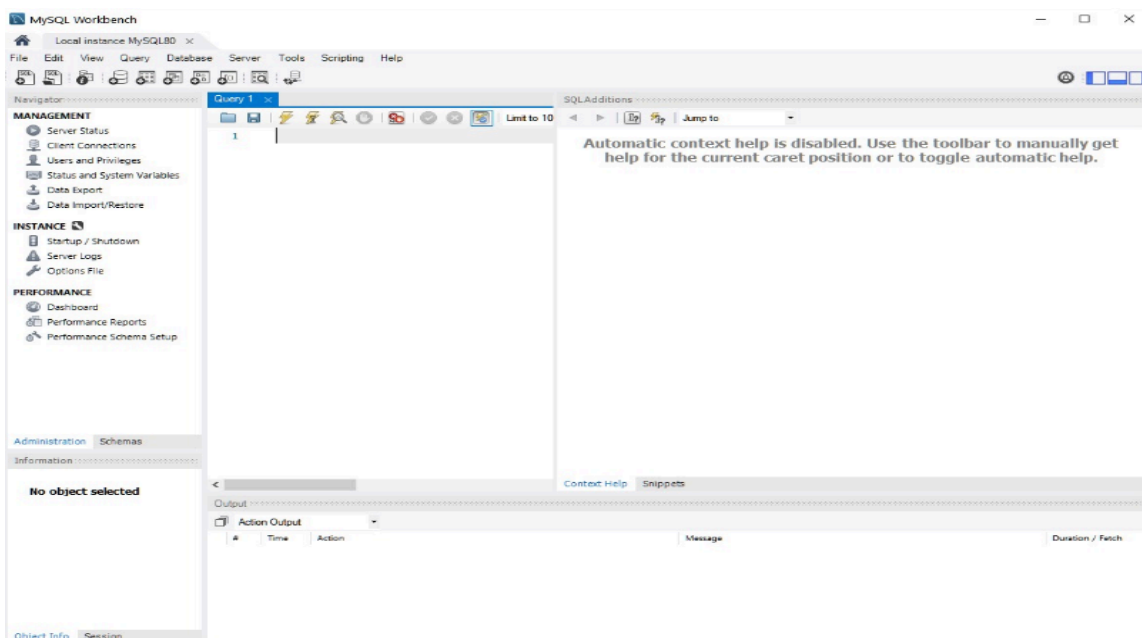
Configure o servidor com um usuario e senha para poder acessar posteriormente:



Ao iniciar o MySQL Workbench clique no sinal de (+) em my connections e crie uma nova conexão, no nosso caso vou dar o nome de Rede, assim que criar ele ira solicitar o login e a senha que voce definiu anteriormente:



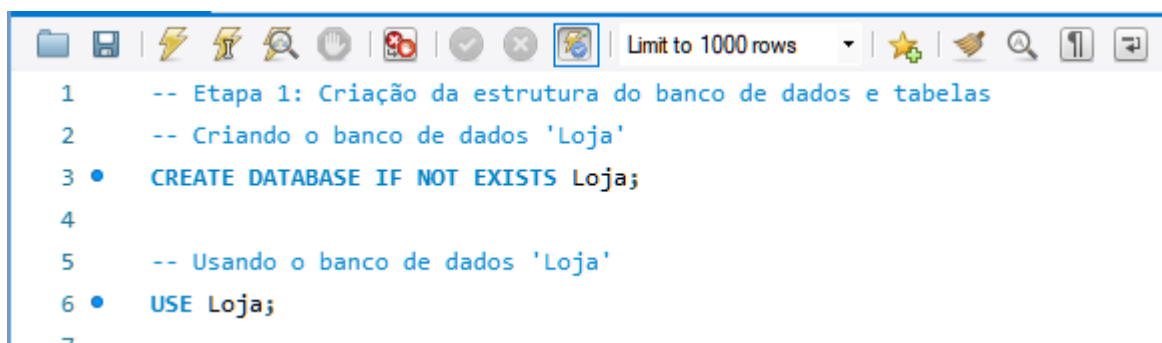
Ao término da instalação e configuração o software deve apresentar essa tela:



## 4. MÉTODO

### 4.1 Criando a base de dados (Loja)

Nessa parte temos a demonstração do primeiro script e já sendo a etapa 1.



```

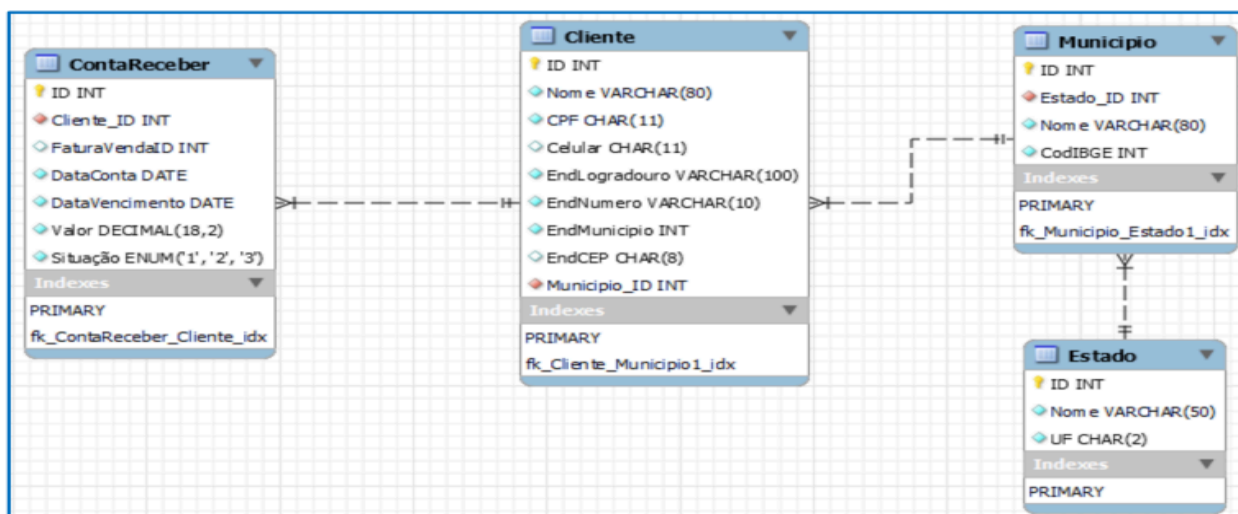
1  -- Etapa 1: Criação da estrutura do banco de dados e tabelas
2  -- Criando o banco de dados 'Loja'
3  • CREATE DATABASE IF NOT EXISTS Loja;
4
5  -- Usando o banco de dados 'Loja'
6  • USE Loja;
7
  
```

### 4.2 Criando as tabelas (clientes, contaReceber, estado e municipio)

Respeitando as seguintes regras:

- As chaves primárias devem ser colocadas todas como auto-incremento.
- Respeite os relacionamentos, tipos, precisões e restrições de não nulo.
- O campo “Situação” da tabela “ContaReceber” deve ser do tipo ENUM e possuir apenas os valores 1, 2 ou 3, sendo: 1 – Conta registrada, 2 – Conta cancelada, 3 – Conta paga.

Aproveitando o script de criação da base de dados vamos escrever os comandos para criação de nossas tabelas e suas respectivas colunas, como na imagem abaixo



Após escrever nosso script de criação ele deve ficar dessa forma:

```

1  -- Etapa 1: Criação da estrutura do banco de dados e tabelas
2  -- Criando o banco de dados 'Loja'
3  • CREATE DATABASE IF NOT EXISTS Loja;
4
5  -- Usando o banco de dados 'Loja'
6  • USE Loja;
7
8  -- Criando a tabela 'Estado'
9  • CREATE TABLE IF NOT EXISTS Estado (
10     ID INT AUTO_INCREMENT PRIMARY KEY,
11     Nome VARCHAR(50) NOT NULL,
12     UF CHAR(2) NOT NULL
13 );
14
15 -- Criando a tabela 'Municipio'
16 • CREATE TABLE IF NOT EXISTS Municipio (
17     ID INT AUTO_INCREMENT PRIMARY KEY,
18     Estado_ID INT NOT NULL,
19     Nome VARCHAR(80) NOT NULL,
20     CodIBGE INT NOT NULL,
21     INDEX fk_Municipio_Estado_idx (Estado_ID),
22     FOREIGN KEY (Estado_ID) REFERENCES Estado(ID)
23 );
24
25 -- Criando a tabela 'Cliente'
26 • CREATE TABLE IF NOT EXISTS Cliente (
27     ID INT AUTO_INCREMENT PRIMARY KEY,
28
29     Nome VARCHAR(100) NOT NULL,
30     CPF CHAR(11) NOT NULL,
31     Celular CHAR(11),
32     EndLogradouro VARCHAR(100),
33     EndNumero VARCHAR(10),
34     EndCEP CHAR(8),
35     Municipio_ID INT NOT NULL,
36     INDEX fk_Cliente_Municipio_idx (Municipio_ID),
37     FOREIGN KEY (Municipio_ID) REFERENCES Municipio(ID)
38 );
39
40 -- Criando a tabela 'ContaReceber'
41 • CREATE TABLE IF NOT EXISTS ContaReceber (
42     ID INT AUTO_INCREMENT PRIMARY KEY,
43     Cliente_ID INT NOT NULL,
44     FaturaVendaID INT NOT NULL,
45     DataConta DATE NOT NULL,
46     DataVencimento DATE NOT NULL,
47     Valor DECIMAL(18,2) NOT NULL,
48     Situacao ENUM('1', '2', '3') NOT NULL,
49     INDEX fk_ContaReceber_Cliente_idx (Cliente_ID),
50     FOREIGN KEY (Cliente_ID) REFERENCES Cliente(ID)

```



### 4.3 Criação do script para inserir dados

Já com nossas tabelas e colunas devidamente criadas vamos inserir dados nelas:

```

52  -- Script para Inserção de dados nas tabelas
53  -- Inserindo dados na tabela 'Estado'
54  ● INSERT INTO Estado (Nome, UF) VALUES
55    ('São Paulo', 'SP'),
56    ('Rio de Janeiro', 'RJ'),
57    ('Minas Gerais', 'MG');
58
59  -- Inserindo dados na tabela 'Município'
60  ● INSERT INTO Município (Estado_ID, Nome, CodIBGE) VALUES
61    (1, 'São Paulo', 3550308),
62    (2, 'Rio de Janeiro', 3304557),
63    (3, 'Belo Horizonte', 3106200);
64
65  -- Inserindo dados na tabela 'Cliente'
66  ● INSERT INTO Cliente (Nome, CPF, Celular, EndLogradouro, EndNumero, EndCEP, Município_ID) VALUES
67    ('João Silva', '12345678901', '11987654321', 'Rua A', '100', '01001000', 1),
68    ('Maria Oliveira', '98765432100', '21987654321', 'Rua B', '200', '20040001', 2),
69    ('Carlos Pereira', '55555555555', '31987654321', 'Rua C', '300', '30120001', 3);
70
71  -- Inserindo dados na tabela 'ContaReceber'
72  ● INSERT INTO ContaReceber (Cliente_ID, FaturaVendaID, DataConta, DataVencimento, Valor, Situacao) VALUES
73    (1, 1001, '2024-08-01', '2024-09-01', 150.00, '1'),
74    (2, 1002, '2024-08-05', '2024-09-05', 250.00, '2'),
75    (3, 1003, '2024-08-10', '2024-09-10', 300.00, '3');

```

### 4.4 Comandos de consulta (DQL)

Nessa etapa vamos elaborar um script para consulta que irá conter uma visão (VIEW) que retornará todas as contas que ainda não foram pagas.

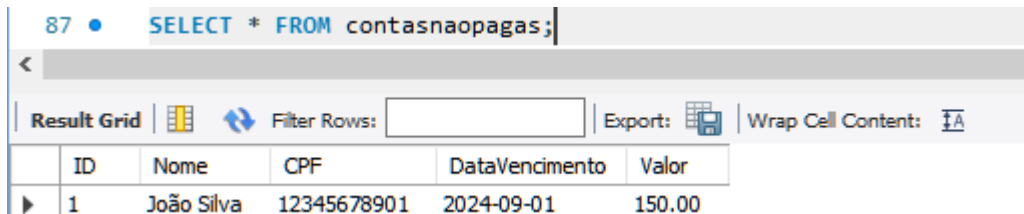
```

79  -- Criando a visão 'ContasNaoPagas' para contas não pagas
80  ● CREATE VIEW Contasnaopagas AS
81    SELECT CR.ID, C.Nome, C.CPF, CR.DataVencimento, CR.Valor
82    FROM ContaReceber CR
83    JOIN Cliente C ON CR.Cliente_ID = C.ID
84    WHERE CR.Situacao = '1';

```

#### 4.5 Comandos de consulta (DQL)

Agora que temos nossa View (contasnaopagas) podemos consultar seu retorno com o comando ilustrado abaixo:



The screenshot shows a SQL query editor interface. At the top, a query bar contains the text `SELECT * FROM contasnaopagas;`. Below the query bar is a toolbar with options like 'Result Grid', 'Filter Rows', 'Export', and 'Wrap Cell Content'. The 'Result Grid' is active, displaying a table with the following data:

	ID	Nome	CPF	DataVencimento	Valor
▶	1	João Silva	12345678901	2024-09-01	150.00

### 5. CONCLUSÃO

Neste portfólio, tive a oportunidade de explorar profundamente o processo de criação e gestão de bancos de dados usando MySQL. Desde a instalação do MySQL e do Workbench até a execução de consultas SQL, cada etapa foi uma experiência de aprendizado significativa.

Comecei pela instalação do MySQL Community Server e configurei o Workbench, que me ofereceu uma interface muito útil para trabalhar com o banco de dados. Isso me deu confiança para avançar para a criação da base de dados chamada "Loja", onde pude aplicar os conceitos da modelagem de dados através de um Diagrama (DER). Ao criar as tabelas, aprendi a definir chaves primárias como auto-incremento e a escolher tipos de dados adequados, como ENUM para o campo "Situação" na tabela "ContaReceber". Essa atenção aos detalhes garantiu a integridade dos dados e me fez perceber a importância de uma boa estruturação.

A fase de inserção de dados, por meio do script `inserir.sql`, foi muito prática. Pude ver como os registros interagiam entre si e como garantir que os dados fossem consistentes, ao trabalhar na criação de uma visão com o script `consulta.sql`, compreendi o valor de simplificar consultas complexas. A visão que criei facilitou a visualização das contas ainda não pagas, demonstrando como diferentes elementos no banco de dados podem ser conectados de maneira eficaz.

## 6R. REFERÊNCIAS

### **MySQL Workbench**

MySQL. (2023). MySQL Workbench. Disponível em:

<https://www.mysql.com/products/workbench/> Acesso em: 14 ago. 2024.