Logotipo

O conteúdo gerado por IA pode estar incorreto.

**CENTRO UNIVERSITÁRIO CATÓLICA DO LESTE DE MINAS GERAIS**

**CURSOS DE CIÊNCIA DA COMPUTAÇÃO E ENGENHARIA DE SOFTWARE**

Daniel Caitano de Souza - A06172224

Eduardo Luís Fernandes - A06158242

Lucas Henrique Oliveira de Paula - A06172301

Mayan Patrick Alves Silveira - A06172341

Rafael Vitor Silva Sousa - A08114420

Thiago Dias Jardim - A06172229

**CONTROLE DE RESERVA DE AMBIENTES – HUB FABRI**

BANCO DE DADOS

Coronel Fabriciano/MG

maio de 2025

**SUMÁRIO**

1. DESCRIÇÃO DO PROJETO DO MINIMUNDO 4

1.1. REQUISITOS FUNCIONAIS: SISTEMA HUB 5

1.2. REQUISITOS NÃO-FUNCIONAIS: SISTEMA HUB 6

1.3. PROJETO DA INTERFACE 6

1.4. DIAGRAMA 8

2. INTRODUÇÃO 9

3. PADRÕES DE NOMECLATURA 9

3.1. PADRÃO DE CÓDIGO 9

3.2. PADRÃO DE NOMECLATURA DO BANCO DE DADOS 9

3.3. PADRÃO DE NOMECLATURA DE TABELAS 10

4. ESTRUTURA DE TABELAS 10

4.1. ENTIDADES IDENTIFICADAS 10

4.2. NOMES DAS TABELAS 10

4.3. NOMES DAS COLUNAS DAS TABELAS 11

4.4. ÍNDICES DAS COLUNAS DAS TABELAS 13

4.5. REPRESENTAÇÃO VISUAL DAS TABELAS 14

5. MODELAGEM 15

5.1 NORMALIZAÇÃO DE TABELA 15

5.2 BOAS PRÁTICAS 15

6. DICIONÁRIO DE DADOS E DIAGRAMA DO MODELO LÓGICO 16

6.1 INTRODUÇÃO 16

7. DICIONÁRIO DE DADOS 16

7.1. NOME E DESCRIÇÃO DAS TABELAS 16

7.2. LISTAGEM DOS ATRIBUTOS 17

7.3. RELACIONAMENTOS ENTRE AS TABELAS 20

8. DIAGRAMA MODELO LÓGICO 20

8.1. INTRODUÇÃO 20

8.2. DIAGRAMA ENTIDADE-RELACIONAMENTO DO MODELO LÓGICO 21

9. DIAGRAMA DO MODELO FÍSICO E PLANO DE TESTE 22

9.1. DIAGRAMA DO MODELO FÍSICO 22

9.2. PERGUNTAS EM SQL 22

**1. DESCRIÇÃO DO PROJETO DO MINIMUNDO**

Surgindo de uma decisão tomada em 2018, juntamente ao Estado, o “HUB Fabri” foi idealizado, sendo um espaço pensado para capacitação, suporte ao microempreendedor, *networking* e inovação. Tendo a missão de acelerar e sustentar o desenvolvimento socioeconômico do município pela melhoria do ambiente de negócios e do ecossistema de inovação.

Sendo uma peça fundamental para o fortalecimento do tecido empresarial do município e região, com papel de servir como um espaço de apoio e estímulo a pesquisa e inovação, além de atrair e receber investimentos para o município.

O espaço proporciona vários *insights*, alternativas de capacitação, informação para que este caráter da inovação seja mais embutido nos cidadãos. Também faz atração de empresas de base tecnológica e startups, conexões a fim de materializar seus projetos. Com parceria do Sebrae, cuja função é desmistificar a inovação para a classe empresarial e para os futuros empreendedores.

O “HUB Fabri” conta com o apoio do Poder Legislativo, Universidades, Empresas Âncoras; SEBRAE, SENAC, SENAI, SENAR, FIEMG, Associação Comercial, Sindcomércio e outras.

Também possui de uma enorme variedade de ambientes projetados para atender diferentes necessidades empresariais e tecnológicas, tendo *coworking* com 16 estações de trabalho; salas para; videoconferência, treinamento, reuniões, auditórios; laboratórios de inovação e informática e área de descompressão, contando com WI-FI gratuito para todos.

Lugar onde o indivíduo apresenta a sua necessidade e é criado um estudo de caso para suprir tal necessidade. “O Hub Fabri é uma espécie de ‘Upa do Empresário’” - Daniel Papa.

O espaço também é sede de serviços públicos, focados nos empresários e geração de empregos e renda na região. Dentre eles, a própria Secretaria de Desenvolvimento, Sala Mineira do Empreendedor, Portal do Emprego e Circuito Mata Atlântica. O “HUB Fabri” tem iniciativa de unir ações físicas e digitais para assegurar o sucesso dos empreendedores, incluindo uma agenda de cursos, oficinas e *workshops*.

**1.1. REQUISITOS FUNCIONAIS: SISTEMA HUB**

|  |  |
| --- | --- |
| CÓDIGO | FUNCIONALIDADE |
| [RF001] | O sistema deve permitir o gerenciamento das salas (informando a quantidade de pessoas e horários para conferir disponibilidade). |
| [RF002] | O sistema deve permitir o cadastro dos usuários (nome, e-mail, telefone). |
| [RF003] | O sistema deve permitir o cadastro dos usuários (CNPJ, Nome da empresa, nome do responsável). |
| [RF004] | O sistema deve possibilitar o a restrição de uso das salas na data e horário em que estas salas estiverem reservadas. |
| [RF005] | Para realizar a reserva de uma sala, o sistema deve verificar se esta sala já está reservada pelo mesmo usuário na mesma data e horário. |
| [RF006] | O sistema deve permitir o controle pelo ADM para registrar, aprovar ou reprovar a reserva do ambiente. |
| [RF007] | O sistema deve atualizar quando uma sala voltar a estar disponível. |
| [RF008] | O sistema deve definir diferentes perfis de usuários: Administrador, Usuário, [...] |
| [RF009] | O sistema deve possibilitar ao administrador alterar as informações das salas disponíveis e seu tempo de locação. |
| [RF010] | O usuário deve ser informado sobre as regras de uso do ambiente no ato da reserva. |

**1.2. REQUISITOS NÃO FUNCIONAIS: SISTEMA HUB**

|  |  |
| --- | --- |
| CÓDIGO | FUNCIONALIDADE |
| [RNF001] | O sistema deve funcionar em sistema operacional Windows, Linux ou MacOS. |
| [RNF002] | O sistema deve ser disponibilizado na web. |
| [RNF003] | O sistema deve encaminhar para o e-mail dos usuários as informações sobre a reserva. |

**1.3. PROJETO DA INTERFACE**

Interface gráfica do usuário, Aplicativo

O conteúdo gerado por IA pode estar incorreto.

Figura 1. Tela inicial/Login. Fonte: Mayan Patrick

Uma imagem contendo Interface gráfica do usuário

O conteúdo gerado por IA pode estar incorreto.

Figura 2. Fonte: Mayan Patrick

Gráfico

O conteúdo gerado por IA pode estar incorreto.

Figura 3. Fonte: Mayan Patrick

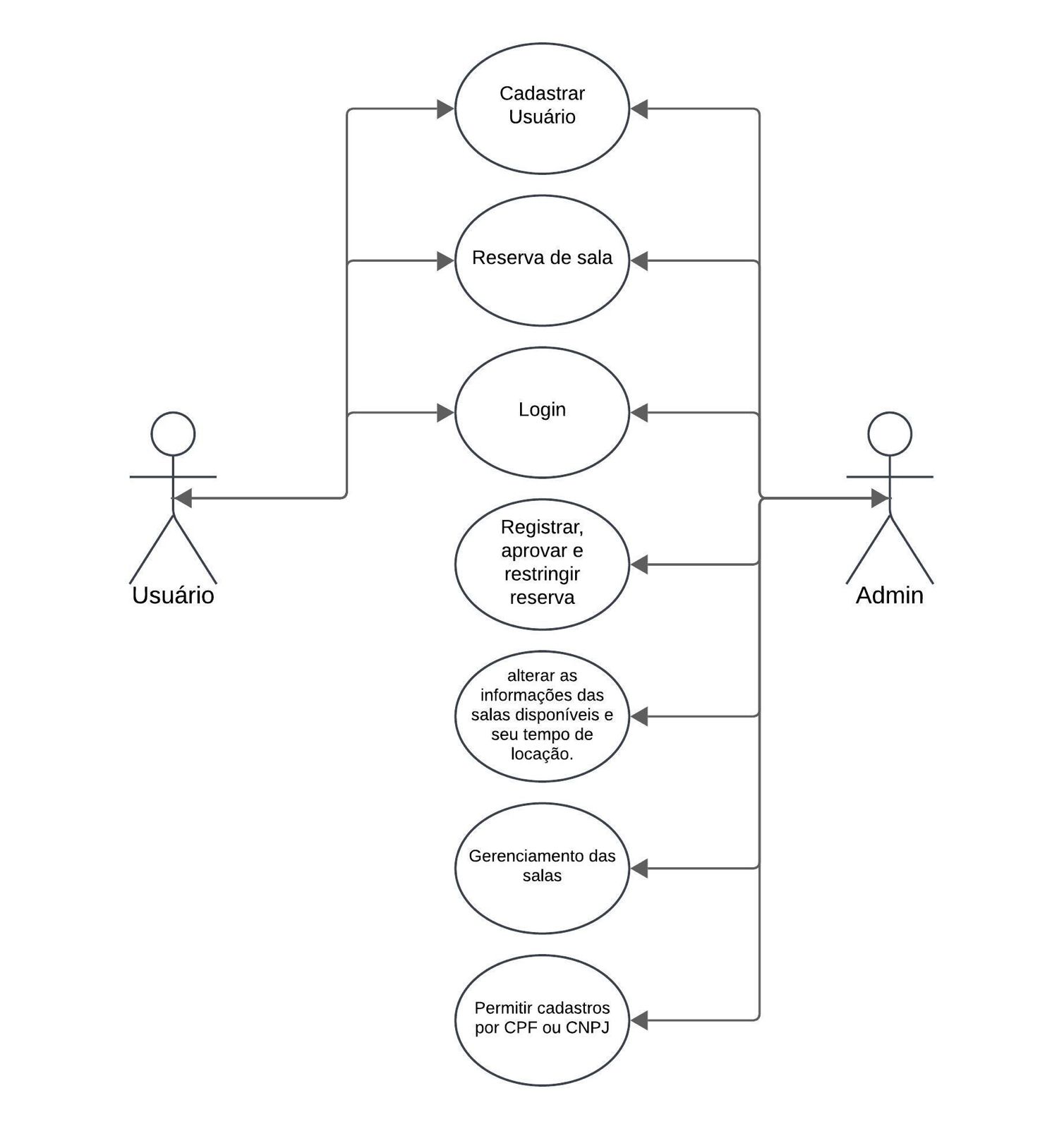
**1.4 DIAGRAMA DE CASO DE USO**

Figura 4. Fonte:

2. INTRODUÇÃO

Objetivo: Este documento tem como objetivo estabelecer padrões de nomenclatura, boas práticas e diretrizes para a modelagem e documentação do banco de dados do projeto, garantindo uniformidade, legibilidade e facilidade de manutenção.

Escopo: Aplica-se a todos os bancos de dados relacionais e não relacionais utilizados no projeto.

Público-alvo: Desenvolvedores, analistas de dados, DBAs e equipes de documentação.

3. PADRÕES DE NOMENCLATURA

3.1. padrão de código

Nesse projeto será usado o padrão ***snake\_case*** onde todas as letras devem ser minúsculas e cada palavra é separada por um sublinhado (\_) .

Texto

O conteúdo gerado por IA pode estar incorreto.

Figura 4. Imagem demonstrativa do padrão snake\_case

3.2. padrão de nomeclatura do banco de dados

O nome do Banco de Dados será precedido por “bd” seguido pelo seu nome.

Até o momento, nosso projeto terá somente um banco de dados, que se chamará “bd\_hub\_principal”.

3.3. padrão de nomeclatura de tabelas

Todos os nomes de Tabelas deverão ser precedidos pelo prefixo “tb” e deverão ser escritos no plural (caso o nome tenha um).

Exemplo:

Uma tabela que vai abrigar os registros de usuários se chamará: “tb\_usuarios”.

4. ESTRUTURA DE TABELAS

Nesse campo serão demonstradas as entidades identificadas e a estrutura de tabelas, contendo as relações de chaves primárias, chaves estrangeiras e constraints.

4.1. ENTIDADES IDENTIFICADAS

Em nosso projeto foram identificadas as seguintes entidades:

* Usuário;
* Empresa;
* Administrador;
* Ambiente;
* Reserva;

4.2. NOMES DAS TABELAS

Tomando como base as entidades identificadas deverão ser criadas as seguintes tabelas:

* Tabela tb\_usuarios;
* Tabela tb\_empresas;
* Tabela tb\_administradores;
* Tabela tb\_ambientes;
* Tabela tb\_reservas;
* Tabela tb\_pendencias;
* Tabela tb\_status;

4.3. nomes das COLUNAS DAS TABELAS

O que está grifado em Amarelo representa Chave Primária, e o que está escrito em Verde representa uma Chave Estrangeira.

* Tabela tb\_usuarios:
* id – INT, PRIMARY KEY
* nome – VARCHAR(250), NOT NULL
* email – VARCHAR(250), NOT NULL, UNIQUE
* telefone – VARCHAR(20), NOT NULL, UNIQUE
* Tabela tb\_empresas:
* nome fantasia – VARCHAR(250), NOT NULL
* cnpj – INT, PRIMARY KEY
* telefone – VARCHAR(20), NOT NULL, UNIQUE
* area\_atuacao – VARCHAR(250), NOT NULL
* Tabela tb\_administradores:
* id – INT, PRIMARY KEY
* nome – VARCHAR(250), NOT NULL
* email – VARCHAR(250), NOT NULL, UNIQUE
* telefone – VARCHAR(20), NOT NULL, UNIQUE
* Tabela tb\_ambientes:
* id – INT, PRIMARY KEY
* nome – VARCHAR(250), NOT NULL
* ocupação máxima - INT
* tipo – VARCHAR(50)
* disponibilidade – VARCHAR(50)
* Tabela tb\_reservas:
* ambiente\_id-INT, FOREIGN KEY

**Chave estrangeira que referencia a coluna “id” ta tabela “tb\_ambientes”**

* id – INT, PRIMARY KEY

**(Nesse caso “ambiente\_id” e “id” formam uma chave primária composta)**

* usuario\_id – FOREIGN KEY

**Chave estrangeira que referencia a coluna “id” da tabela “tb\_usuarios”**

* data\_reserva – DATE, NOT NULL
* horario\_inicio – TIME, NOT NULL
* horario\_fim – TIME, NOT NULL
* quantidade\_pessoas – INT
* finalidade – VARCHAR(25 0)
* Tabela tb\_status:
* reserva\_id – INT, FOREIGN KEY

**Chave estrangeira que referencia a coluna “id” da tabela “tb\_reservas”**

* id\_status – INT, PRIMARY KEY

**(Nesse caso há uma chave primária composta)**

* status\_aprovacao – VARCHAR(50)
* administrador\_id – FOREIGN KEY

**Chave estrangeira que referencia a coluna “id” da tabela “tb\_administradores”**

4.4. ÍNDICES DAS COLUNAS DAS TABELAS

Os índices têm como objetivo facilitar a busca dentro das tabelas do Banco de Dados ao procurar de forma direta, dados inseridos nas colunas, economizando tempo e processamento.

Relação de índices nas tabelas:

* Tabela tb\_usuarios:
* Colunas: nome, email
* (idx\_usuarios\_nome), (idx\_usuarios\_email)
* Tabela tb\_empresas:
* Colunas: nome\_fantasia
* (idx\_empresas\_nome\_fantasia)
* Tabela tb\_administradores:
* Colunas: nome, email
* (idx\_administradores\_nome), (idx\_administradores\_email)
* Tabela tb\_ambientes:
* Colunas: nome, tipo, ocupacao
* (idx\_ambientes\_nome),(idx\_ambientes\_tipo), (idx\_ambientes\_ocupacao\_maxima)
* Tabela tb\_reservas:
* Colunas: data\_ reservas, horario\_inicio, horario\_fim
* (idx\_reservas\_data\_agendamento),(idx\_reservas\_horario\_inicio), (idx\_reservas\_horario\_fim)
* Tabela tb\_status:
* Colunas: status\_aprovacao
* (idx\_status\_status\_aprovacao)

4.5. REPRESENTAÇÃO VISUAL DAS TABELAS

|  |  |  |  |
| --- | --- | --- | --- |
| id | nome | email | telefone |
|  |  |  |  |

Figura 5. Representação visual da tabela “tb\_usuarios”

|  |  |  |  |
| --- | --- | --- | --- |
| cnpj | nome\_fantasia | telefone | area\_atuacao |
|  |  |  |  |

Figura 6. Representação visual da tabela “tb\_empresas”

|  |  |  |  |
| --- | --- | --- | --- |
| id | nome | email | telefone |
|  |  |  |  |

Figura 7. Representação visual da tabela “tb\_administradores”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | nome | ocupacao | tipo | disponibilidade |
|  |  |  |  |  |

Figura 8. Representação visual da tabela “tb\_ambientes”

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ambiente\_id | id | usuario\_cpf | data\_  reserva | horario\_  inicio | horario\_  fim | quantidade\_  pessoas | finalidade |
|  |  |  |  |  |  |  |  |

Figura 9. Representação visual da tabela “tb\_reservas”

|  |  |  |  |
| --- | --- | --- | --- |
| reserva\_id | id\_status | status\_aprovacao | administrador\_id |
|  |  |  |  |

Figura 10. Representação visual da tabela “tb\_status”

5. Modelagem

5.1 Normalização de tabela.

Neste projeto serão usados os seguintes NS de normalização que são: 1NF 2NF 3NF somente o 4 e 5 não serão utilizados.

5.2 Boas PRÁTICAS.

* Utilização de índices para facilitar nas consultas;
* Uso de nomes significativos e bem descritos para tabelas e colunas, para que os outros desenvolvedores tenho facilidade na hora de identificar as tabelas;
* Uso de constraints para que haja uma restrição que deve ser seguida para alguns campos das tabelas;
* Use o SQL de forma que facilite tanto o entendimento quanto a manipulação das tabelas e colunas;
* Uso de diagramas de entidade-relacionamento (ER) para representar as entidades e os relacionamentos entre elas. Isso ajuda a visualizar as dependências entre os dados;
* Definir claramente as **entidades**, **atributos** e **relacionamentos**;
* Use as **cardinalidades** corretamente para garantir que os relacionamentos sejam bem definidos (1:1, 1:N, N:M).

**6. DICIONÁRIO DE DADOS E DIAGRAMA DO MODELO LÓGICO**

**6.1. INTRODUÇÃO**  
  
Um dicionário de banco de dados, também conhecido como catálogo de dados ou repositório de metadados, é uma coleção organizada de informações que descrevem a estrutura, os relacionamentos e as características dos dados armazenados em um sistema de gerenciamento de banco de dados (SGBD). Ele serve como uma referência essencial para desenvolvedores, administradores de banco de dados e usuários, facilitando a compreensão e a utilização eficaz dos dados.  
O dicionário de banco de dados é fundamental para a integridade e a eficiência do sistema, pois fornece um entendimento claro sobre como os dados estão organizados e como podem ser acessados. Além disso, ele desempenha um papel crucial na documentação e na governança de dados, ajudando a garantir que os dados sejam utilizados de maneira consistente e conforme as políticas estabelecidas.

**7. DICIONÁRIO DE DADOS**

**7.1. NOME E DESCRIÇÃO DAS TABELAS**

* **Tabela tb\_usuarios:**

Essa tabela contém informações sobre os usuários que faram agendamentos das salas.

* **Tabela tb\_empresas:**

Essa tabela contém informações sobre as empresas que faram agendamentos das salas.

* **Tabela: tb\_administradores:**

Essa tabela contém informações sobre os responsáveis pela admissão do agendamento das salas.

* **Tabela tb\_salas:**

Essa tabela foi feita para guardar detalhadamente as informações sobre cada sala dentro do Hub.

* **Tabela tb\_reservas:**

Essa tabela é responsável para registra as informações dos horários e datas em que as salas estão disponíveis para o agendamento.

* **Tabela tb\_status:**

Essa tabela tem como objetivo conter as informações de validação ao agendamento da sala.

**7.2. LISTAGEM DOS ATRIBUTOS**

**Tabela tb\_usuarios:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Campo | Tipo de dados | Chave Primaria? | Chave Estrangeira? | Restrição | Descrição |
| id | Inteiro | Sim | Não | Chave primaria | Identificador único para usuário |
| nome | Texto | Não | Não | Não nulo | Nome do usuário |
| email | Texto | Não | Não | Não nulo, único | Email do usuário |
| telefone | Texto | Não | Não | Não nulo, único | Contato do usuário |

**Tabela tb\_empresas:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Campo | Tipo de Dados | Chave Primaria? | Chave Estrangeira? | Restrições | Descrição |
| nome\_fantasia | Texto | Não | Não | Não nulo | Nome da empresa |
| cnpj | Inteiro | Sim | Não | Chave Primaria, Único | Identificador único da empresa |
| telefone | Texto | Não | Não | Não nulo, único | Forma de contato com a empresa |
| area\_atuacao | Texto | Não | Não | Não nulo | Área de atuação da empresa |

**Tabela tb\_administradores:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Campo | Tipo de Dados | Chave Primária? | Chave Estrangeira? | Restrições | Descrição |
| id | Inteiro | Sim | Não | Chave Primaria | Identificador único para administradores |
| nome | texto | Não | Não | Não Nulo | Nome do Administradores |
| email | texto | Não | Não | Não Nulo, único | Email Administradores |
| telefone | texto | Não | Não | Não Nulo, único | Contado do administrador |

**Tabela tb\_ambientes:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Campo | Tipo de Dados | Chave Primária? | Chave Estrangeira? | Restrições | Descrição |
| id | Inteiro | sim | Não | Chave Primaria | Identificador único para sala |
| nome | texto | Não | Não | Não Nulo | Nome da sala |
| Ocupação máxima | inteiro | Não | Não | Não Nulo, único | Ditar o máximo que podem usar a sala ao mesmo tempo |
| tipo | texto | Não | Não |  | Funcionalidades da sala |
| disponibilidade | texto | Não | Não |  | Identificar se a sala está disponível ou não |

**Tabela tb\_reservas:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Campo | Tipo de Dados | Chave Primária? | Chave Estrangeira? | Restrições | Descrição |
| id\_  agendamento | Inteiro | Sim | Não | Chave composta | Identificador único para agendamento |
| ambientes\_  id | texto | Sim | Sim | Chave composta | Identificador único para sala |
| Usuário\_  id | inteiro | Não | Sim | Não Nulo, único | Identificador único para usuário |
| data\_  agendamento | data | Não | Não | Não nulo | Data para o agendamento da sala |
| horário | Inteiro | Não | Não | Não nulo | Horário do agendamento da sala |
| quantidade\_  pessoas | Inteiro | Não | Não | Não nulo | Armazenar a quantidade de pessoas que estarão presentes |
| finalidade | texto | Não | Não | Não nulo | Ver para oque a sala será utilizada |

**Tabela tb\_status:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Campo | Tipo de Dados | Chave Primária? | Chave Estrangeira? | Restrições | Descrição |
| agendamento\_id | Inteiro | Sim | Sim | Chave Primaria ,Não Nulo, Único | Identificador único da agendamento |
| id\_status | Inteiro | Sim | Não | Chave Primaria,  Único | Identificador único da status |
| status\_aprovação | texto | Não | Não | Não Nulo, único | Aprovar ou desaprovar a utilização da sala |
| administrador\_id | Inteiro | Sim | Sim | Chave Primaria ,Não Nulo, Único | Identificador único da administrador |

**7.3 RELACIONAMENTOS ENTRE AS TABELAS**

Usuário se relaciona com a tabela agendamento para agendar uma sala.

Empresa se relaciona com a tabela agendamento para agendar uma sala.

Administrador se relaciona com a tabela status para aprovar ou desaprovar o agendamento de alguma sala.

A tabela sala está relacionada a tabela agendamento para ser confirmar ser a sala em questão será utilizada em que hora e data.

Agendamento se relaciona com a tabela sala para averiguar se está disponível ou ocupada no momento.

Status se relaciona tanto com a tabela agendamento quanto para a tabela administrador para que com as informações da tabela agendamento que vai averiguar se a sala está disponível, e a tabela administrador que vai dar a aprovação da utilização da sala.

**8. DIAGRAMA MODELO LÓGICO**

**8.1. INTRODUÇÃO**

O diagrama de modelo lógico é uma representação visual que descreve a estrutura de um banco de dados em um nível de abstração intermediário, entre o modelo conceitual e o modelo físico. Ele é utilizado para ilustrar como os dados são organizados, as relações entre diferentes entidades e as regras de integridade que governam esses dados.

No diagrama de modelo lógico, as entidades (ou tabelas) são representadas por retângulos, enquanto os atributos (ou colunas) são listados dentro dessas entidades. As relações entre as entidades são indicadas por linhas que conectam os retângulos, frequentemente acompanhadas de notações que especificam a cardinalidade (por exemplo, um-para-um, um-para-muitos) e a opcionalidade (se a relação é obrigatória ou opcional).

Diagrama, Esquemático

O conteúdo gerado por IA pode estar incorreto.**8.2 DIAGRAMA ENTIDADE-RELACIONAMENTO DO MODELO LÓGICO**

Figura 11. Diagrama Entidade-Relacionamento do Modelo Lógico

**9. DIAGRAMA DO MODELO FÍSICO E PLANO DE TESTE**

**9.1. DIAGRAMA DO MODELO FÍSICO**

Diagrama

O conteúdo gerado por IA pode estar incorreto.

Figura 12. Diagrama do modelo físico

**9.2. PERGUNTAS EM SQL**

1. Quais usuários reservaram ambientes mais de uma vez em 2024?

* Tabelas envolvidas: tb\_reservas, tb\_usuarios
* Tipo de consulta: SELECT + JOIN + GROUP BY + HAVING
* SQL:

*SELECT tb\_usuarios.nome, COUNT(tb\_reservas.id) AS total\_reservas*

*FROM tb\_usuarios*

*JOIN tb\_reservas ON tb\_usuarios.id = tb\_reservas.usuario\_id*

*WHERE YEAR(tb\_reservas.data\_reserva) = 2024*

*GROUP BY tb\_usuarios.id, tb\_usuarios.nome*

*HAVING COUNT(tb\_reservas.id) > 1;*



Figura 13. Exemplo visual pergunta 1

1. Quantas reservas foram feitas por tipo de ambiente?

* Tabelas envolvidas: tb\_reservas, tb\_ambientes
* Tipo de consulta: SELECT + JOIN + GROUP BY
* SQL:

*SELECT tb\_ambientes.tipo, COUNT(tb\_reservas.id) AS total\_reservas*

*FROM tb\_ambientes*

*JOIN tb\_reservas ON tb\_ambientes.id = tb\_reservas.ambiente\_id*

*GROUP BY tb\_ambientes.tipo;*

Tabela

O conteúdo gerado por IA pode estar incorreto.

Figura 14. Resultado da consulta SQL da pergunta 2

1. Quais ambientes estão com status "pendente"?

* Tabelas envolvidas: tb\_ambientes, tb\_status
* Tipo de consulta:SELECT + JOIN + WHERE
* SQL:

*SELECT tb\_ambientes.nome, tb\_status.status\_aprovacao*

*FROM tb\_ambientes*

*JOIN tb\_status ON tb\_ambientes.id = tb\_status.ambiente\_id*

*WHERE tb\_status.status\_aprovacao = 'pendente';*

Interface gráfica do usuário, Texto

O conteúdo gerado por IA pode estar incorreto.

Figura 15. Resultado da consulta SQL da pergunta 3

1. Qual a média de pessoas por tipo de ambiente reservado em abril de 2024?

* Tabelas envolvidas: tb\_reservas, tb\_ambientes
* Tipo de consulta: SELECT + JOIN + WHERE + GROUP BY
* SQL:

*SELECT tb\_ambientes.tipo, AVG(tb\_reservas.quantidade\_pessoas) AS media\_pessoas*

*FROM tb\_reservas*

*JOIN tb\_ambientes ON tb\_reservas.ambiente\_id = tb\_ambientes.id*

*WHERE MONTH(tb\_reservas.data\_reserva) = 4 AND YEAR(tb\_reservas.data\_reserva) = 2024*

*GROUP BY tb\_ambientes.tipo;*

Tabela

O conteúdo gerado por IA pode estar incorreto.

Figura 16. Exemplo visual da consulta SQL da pergunta 4

1. Quais administradores aprovaram mais de uma reserva?

* Tabelas envolvidas: tb\_status, tb\_administradores
* Tipo de consulta: SELECT + JOIN + GROUP BY + HAVING
* SQL:

*SELECT tb\_administradores.nome, COUNT(tb\_status.id\_status) AS total\_aprovacoes*

*FROM tb\_status*

*JOIN tb\_administradores ON tb\_status.administrador\_id = tb\_administradores.id*

*WHERE tb\_status.status\_aprovacao = 'aprovado'*

*GROUP BY tb\_administradores.id, tb\_administradores.nome*

*HAVING COUNT(tb\_status.id\_status) > 1;*

Texto

O conteúdo gerado por IA pode estar incorreto.

Figura 17. Exemplo visual da consulta SQL da pergunta 5

1. Quantas reservas foram realizadas em cada dia de abril de 2024?

* Tabelas envolvidas: tb\_reservas
* Tipo de consulta: SELECT + WHERE + GROUP BY
* SQL:

*SELECT tb\_reservas.data\_reserva, COUNT(tb\_reservas.id) AS total\_reservas*

*FROM tb\_reservas*

*WHERE MONTH(tb\_reservas.data\_reserva) = 4 AND YEAR(tb\_reservas.data\_reserva) = 2024*

*GROUP BY tb\_reservas.data\_reserva*

*ORDER BY tb\_reservas.data\_reserva;*

Tabela

O conteúdo gerado por IA pode estar incorreto.

Figura 18. Exemplo visual da consulta SQL da pergunta 6

1. Quais usuários fizeram reservas no auditório em maio de 2024?

* Tabelas envolvidas: tb\_reservas, tb\_usuarios, tb\\_ambientes
* Tipo de consulta: SELECT + JOIN + WHERE
* SQL:

*SELECT tb\_usuarios.nome, tb\_reservas.data\_reserva, tb\_reservas.horario\_inicio, tb\_reservas.horario\_fim*

*FROM tb\_reservas*

*JOIN tb\_usuarios ON tb\_reservas.usuario\_id = tb\_usuarios.id*

*JOIN tb\_ambientes ON tb\_reservas.ambiente\_id = tb\_ambientes.id*

*WHERE tb\_ambientes.tipo = 'auditorio'*

*AND MONTH(tb\_reservas.data\_reserva) = 5*

*AND YEAR(tb\_reservas.data\_reserva) = 2024;*



Figura 19. Exemplo visual da consulta SQL da pergunta 7