**Equipe FullStack**

**Documentação do Banco de Dados para o Sistema de Agendamentos de Reuniões para a empresa SIATT**

1.Introdução  
Esse documento tem como objetivo armazenar informações sobre a modelagem do banco de dados, facilitar o entendimento de pessoas que possuem interesse no projeto e auxiliar membros da nossa equipe de desenvolvedores.  
O propósito deste sistema é centralizar informações de diferentes fontes a fim de manter um banco de dados capaz de agendar reuniões de diferentes tipos e diferentes níveis de permissão na SIATT, automatizando o processo e reduzindo o tempo gasto na atividade, tanto durante o agendamento quanto durante a preparação da reunião.  
Aqui é encontrado detalhes sobre as entidades que compõem o banco de dados e como eles se relacionam, atributos que compõem cada entidade, o detalhamento sobre seus relacionamentos, as regras de negócio que orientam sua interação, o esquema do modelo conceitual do banco de dados juntamente com o esquema do modelo lógico, o dicionário de dados presentes nesse modelo, código SQL do banco de dados e as especificações de termos para facilitar a compreensão da nomenclatura.

**Nota:** *Esta documentação está sujeita a alterações à medida que o projeto evolui. As mudanças feitas em qualquer parte desse esquema de modelagem de banco de dados serão prontamente atualizadas e postas à disposição. A versão mais atualizada estará disponível para qualquer pessoa que tenha o interesse em acessá-la.*

4. Regras de Negócio

- Usuários possuem nome, e-mail, nível de permissão e diretoria.  
- Reuniões podem ser online e offline, individual ou em grupo e em casos especiais, Hibridas.  
- Salas possuem nível de permissão, podem ser online e físicas.  
- Reuniões Hibridas possuem uma sala online e uma sala física.  
- Usuários apenas agenda a sala para reunião, ou convoca uma reunião através de um agendamento completo.

3. Entidades e Atributos

**Usuários:  
- id\_usuario:** INT, identificador do usuário (Primary Key). **- nome\_usuario:** VARCHAR(256), nome do usuário. **- email\_usuario:** VARCHAR(256), e-mail do usuário.  
- **senha\_usuario**: VARCHAR(256), senha de acesso do usuário.  
**- diretoria\_usuario:** VARCHAR(256), departamento do usuário.  
**- permissao\_usuario:** VARCHAR(256), nível de permissão do usuário (1, 2, 3).  
- **admin\_usuario**: BOOLEAN(DEFAULT), administrador responsável pelo usuário.

**Agendamentos:  
- id\_agendamento:** INT, identificador único de agendamento (Primary Key).  
**- id\_usuario:** INT, chave estrangeira que faz referência a id\_usuario da entidade Usuário.  
**- id\_sala:** INT, chave estrangeira que faz referência a id\_sala da entidade Salas  
**- id\_reuniao:** INT, chave estrangeira que faz referência a id\_reuniao da entidade Reunião. **- datetime\_início:** DATETIME, data e hora de início do agendamento. **- datetime\_final:** DATETIME, data e hora do final doagendamento.

**Reuniões:  
- id\_reuniao:** INT, identificador único de reunião (Primary Key)**.  
- id\_agendamento:** INT, chave estrangeira que faz referência a id\_agendamento da entidade Agendamento. **- id\_participante:** INT, chave estrangeira que faz referência a id\_participante da entidade Participante. **- tipo\_reuniao**: VARCHAR(256), indicador de tipo de reunião (Física, Virtual ou Hibrida)**.  
- titulo\_reuniao:** VARCHAR(256), nome atribuído a reunião.  
**- datetime\_inicio:** DATETIME, data e hora de início da reunião. **- datetime\_final:** DATETIME, data e hora do final da reunião. **- pauta\_reuniao:** VARCHAR(256), assunto que será abordado na reunião. **- responsável\_reuniao:** VARCHAR(256), usuário responsável pela reunião.

**Participantes:  
- id\_participante:** INT, identificador único de participantes (Primary Key).  
**- nome\_participante:** VARCHAR(256), nome dos participantes.  
**- email\_participante:** VARCHAR(256), e-mail dos participantes.  
**Salas:  
- id\_sala:** INT, identificador único de sala (Primary Key).  
- **id\_sala\_fisica:** INT, chave estrangeira que faz referência a id\_sala\_fisica da entidade Sala Física.  
**- id\_sala\_virtual:** INT, chave estrangeira que faz referência a id\_sala\_virtual da entidade Sala Virtual.- **id\_sala\_hibrida:** INT, chave estrangeira que faz referência a id\_sala\_hibrida da entidade Sala Hibrida. **- permissao\_sala:** BOOLEAN, nível de permissão que a sala exige.  
**- tipo\_sala:** VARCHAR(256), indicador de tipo de sala (Física, Virtual ou Hibrida).

**Sala Física:  
- id\_sala\_fisica:** INT, identificador único de sala física (Primary Key). **- nome\_sala\_fisica:** VARCHAR(256), nome da sala física. **- permissao\_sala\_fisica:** VARCHAR(256), nível de permissão que a sala física exige.  
**- vagas\_sala\_fisica:** INT, quantidade de vagas disponíveis na sala física.

**Sala Virtual:   
- id\_sala\_virtual:** INT, identificador único de sala virtual (Primary Key).  
**- nome\_sala\_virtual:** VARCHAR(256), nome da sala virtual. **- link\_sala\_virtual:** VARCHAR(256), link de acesso a sala virtual.  
**- permissao\_sala\_virtual:** VARCHAR(256), nível de permissão que a sala exige.

**Sala Hibrida:  
- id\_sala\_hibrida:** INT, identificador único de sala hibrida (Primary Key). **- nome\_sala\_hibrida:** VARCHAR(256), nome da sala hibrida. **- id\_sala\_virtual:** INT, chave estrangeira que faz referência a id\_sala\_virtual da entidade Sala Virtual. **- id\_sala\_fisica:** INT, chave estrangeira que faz referência a id\_sala\_fisica da entidade Sala Física.

4. Relacionamentos

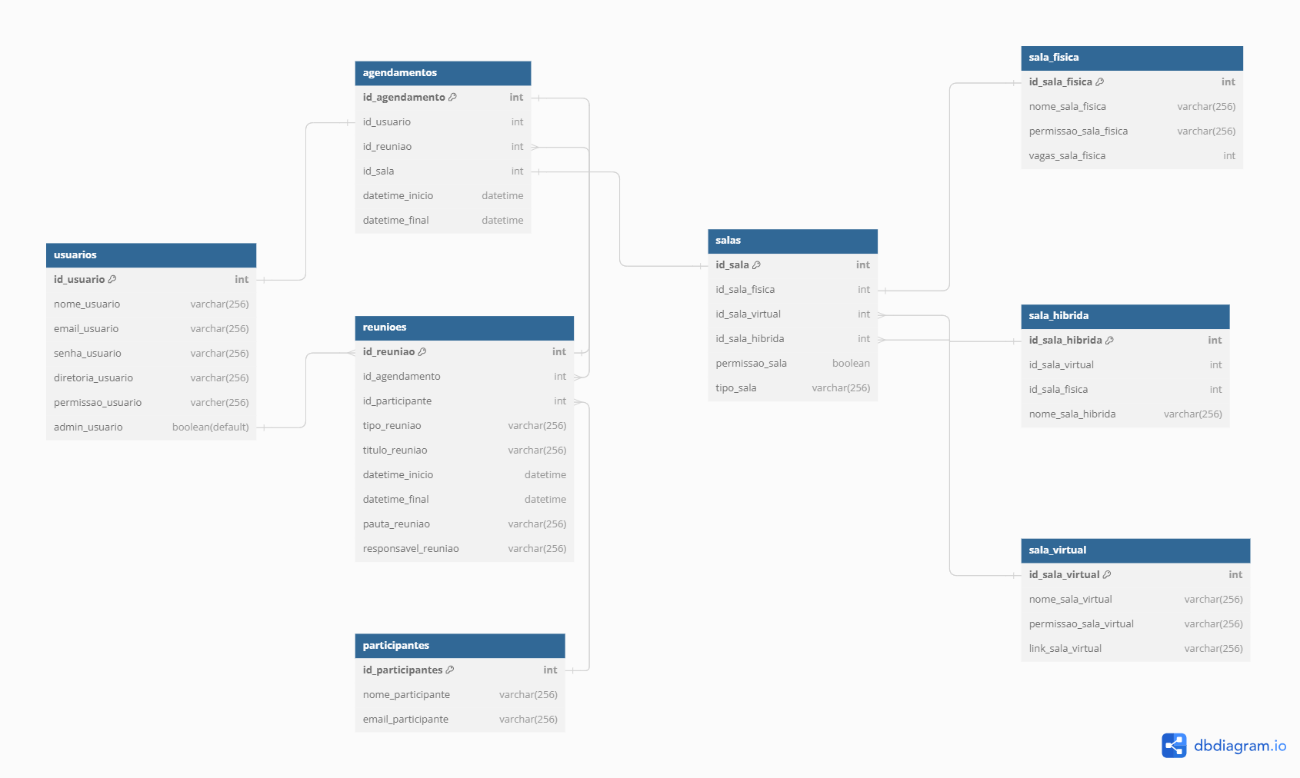
- Um **Usuário** pode ou não, criar vários **Agendamentos**, mas cada **Agendamento** só pertence a um **Usuário**.  
- Um **Agendamento** , pode ou não, estar associado a uma **Sala Física**, e uma **Sala Física** pode não ter **Agendamentos**, ou ter vários **Agendamentos**.  
- Um **Agendamento**, pode ou não, estar associado a uma **Sala Virtual**, e uma **Sala Virtual** pode não ter **Agendamentos**, ou ter vários **Agendamentos**.  
- Um **Agendamento** , pode ou não, estar associado a uma **Sala Hibrida**, e uma **Sala Hibrida** pode não ter **Agendamentos**, ou ter um **Agendamentos**, desde que contenha uma **Sala Virtual** e uma **Sala Física**.  
- Uma **Reunião** está associada a um **Agendamento**, mas um **Agendamento** não precisa conter uma **Reunião**.  
- Em uma **Reunião** deve haver no mínimo um **Usuário**.  
- Uma **Reunião** pode ou não ter **Participantes**, sendo em uma **Sala Física** um número de **Participantes** limitado, em **Sala Virtual** ilimitado e **Sala Hibrida** parcialmente ilimitado (Caso a **Sala Física** contida na **Reunião** atinja capacidade máxima).

5. Modelo Conceitual

Diagrama

Descrição gerada automaticamente

6. Modelo Lógico



7. Código SQL para criação de tabelas.

create database api4;

drop database api4;

use api4;

create table usuarios (

id\_usuario int primary key auto\_increment,

nome\_usuario varchar(256) not null,

email\_usuario varchar(256) not null,

senha\_usuario varchar(256) not null,

diretoria\_usuario varchar(256) not null,

permissao\_usuario varchar(256) not null,

admin\_usuario boolean default 0

);

create table reunioes (

id\_reuniao int primary key auto\_increment,

id\_agendamento int not null,

# foreign key (id\_agendamento) references agendamentos (id\_agendamentos),

id\_participante int not null,

# foreign key (id\_participante) references participantes (id\_participantes),

tipo\_reuniao varchar(256) not null,

titulo\_reuniao varchar (256) not null,

datetime\_inicio datetime not null,

datetime\_final datetime not null,

pauta\_reuniao varchar(256) not null,

responsavel\_reuniao varchar(256) not null

);

create table agendamentos (

id\_agendamento int primary key auto\_increment,

id\_usuario int not null,

# foreign key (id\_usuario) references usuarios (id\_usuario),

id\_reuniao int not null,

# foreign key (id\_reuniao) references reunioes (id\_reuniao),

id\_sala int not null,

# foreign key (id\_sala) references salas (id\_sala),

datetime\_inicio datetime not null,

datetime\_final datetime not null

);

create table participantes (

id\_participantes int primary key auto\_increment,

nome\_participante varchar(256) not null,

email\_participante varchar (256) not null

);

create table salas (

id\_sala int primary key auto\_increment not null,

id\_sala\_fisica int not null,

# foreign key (id\_sala\_fisica) references sala\_fisica (id\_sala\_fisica),

id\_sala\_virtual int not null,

# foreign key (id\_sala\_virtual) references sala\_virtual (id\_sala\_virtual),

id\_sala\_hibrida int not null,

# foreign key (id\_sala\_hibrida) references sala\_hibrida (id\_sala\_hibrida),

permissao\_sala boolean not null,

tipo\_sala varchar(256) not null

);

create table sala\_fisica (

id\_sala\_fisica int primary key not null,

nome\_sala\_fisica varchar(256) not null,

permissao\_sala\_fisica varchar(256) not null,

vagas\_sala\_fisica int not null

);

create table sala\_virtual (

id\_sala\_virtual int primary key not null,

nome\_sala\_virtual varchar(256) not null,

permissao\_sala\_virtual varchar(256) not null,

link\_sala\_virtual varchar(256) not null

);

create table sala\_hibrida (

id\_sala\_hibrida int primary key not null,

id\_sala\_virtual int not null,

# foreign key (id\_sala\_virtual) references sala\_virtual (id\_sala\_virtual),

id\_sala\_fisica int not null,

# foreign key (id\_sala\_fisica) references sala\_fisica (id\_sala\_fisica),

nome\_sala\_hibrida varchar(256) not null

);