**Banco de dados.**

Com base nas regras de negócio e nos requisitos pré-estabelecidos, foram criadas as tabelas para cada entidade disposta no banco de dados físico, sendo estas: Tabela User, Tabela Cliente, Tabela técnico, Tabela ADM, Tabela Chamado e Tabela Relatório.

As tabelas foram formalizadas e atendem as 3 primeiras formas normais, ou seja, não há atributos multivalorados, sem dependência da Primary Key ou de dependência transitiva. Após a formalização, foi feito o dicionário de dados de cada tabela, contendo as informações do nome do atributo, tipo de dados, tamanho do dado em bytes, obrigatoriedade, tipo de chave e descrição. A formalização e o dicionário de dados foram feitas para todas as tabelas, porém serão dispostas apenas algumas como exemplo, sendo as mais importantes:

**Tabela X – Tabela User.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela User** | | | | | |
| **Campo** | **Tipo de Dado** | **Obrigatório** | **Tamanho** | **Chave** | **Descrição** |
| id\_User | INT (IDENTITY) | Sim | 4 bytes | PK | Identificador único do usuário |
| email\_User | VARCHAR | Sim | 1 byte | UNIQUE | E-mail do usuário (não pode repetir) |
| senha\_User | VARCHAR | Sim | 1 byte | - | Senha do usuário |
| cargo\_User | VARCHAR | Sim | 1 byte | - | Cargo ocupado pelo usuário na empresa |
| departamento\_user | VARCHAR | Sim | 1 byte | - | Departamento ao qual o usuário pertence |
| nivelAcesso\_User | INT | Sim | 4 bytes | - | Nível de acesso do usuário no sistema |
| nome\_User | VARCHAR | Sim | 1 byte | - | Primeiro nome do usuário |
| sobrenome\_User | VARCHAR | Sim | 1 byte | - | Sobrenome do usuário |

Fonte: autoria própria

**Tabela X- Tabela Relatório.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela Relatorio** | | | | | |
| **Campo** | **Tipo de Dado** | **Obrigatório** | **Tamanho** | **Chave** | **Descrição** |
| id\_Relatorio | INT (IDENTITY) | Sim | 4 bytes | PK | Identificador único do relatório. Gerado automaticamente pelo banco. |
| id\_Cham | INT | Sim | 4 bytes | FK | Referência ao chamado relacionado (busca as informações do chamado) |
| relatorio | VARCHAR | Sim | 1 byte | - | Texto do relatório referente ao(s) chamado(s) solicitado(s). |
| id\_UsuarioCriador | INT | Sim | 4 bytes | FK | Referencia ao Id do Administrador que gera o relatório. |

Fonte: autoria própria.

**Tabela X – Tabela Chamado.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tabela Chamado** | | | | | |
| **Campo** | **Tipo de Dado** | **Obrigatório** | **Tamanho** | **Chave** | **Descrição** |
| id\_Cham | INT (IDENTITY) | Sim | 4 bytes | PK | Identificador do chamado. Gerado automaticamente pelo banco. |
| status\_Cham | VARCHAR | Sim | 1 byte | - | Status atual do chamado ( aberto, em andamento, fechado) |
| dataAbertura\_Cham | DATE (DDMMAAAA) | Sim | 3 bytes | - | Data de abertura do chamado no formato em dias, meses e anos |
| dataFechamento\_Cham | DATE (DDMMAAAA) | Não | 3 bytes | - | Data de fechamento do chamado no formato em dias, meses e anos |
| dataProblema | DATE (DDMMAAAA) | Sim | 3 bytes | - | Data de ínicio do problema do chamado no formato em dias, meses e anos |
| prioridade\_Cham | CHAR | Sim | 1 byte | - | Prioridade do chamado (A - alta, M - média, B - baixa) |
| categoria\_Cham | VARCHAR | Sim | 1 byte | - | Categoria do chamado (ex: software, hardware) |
| solucaoIA\_Cham | VARCHAR | Não | 1 byte | - | Solução sugerida pela inteligência artificial (se ela houver) |
| solucaoTec\_Cham | VARCHAR | Não | 1 byte | - | Solução proposta pelo técnico |
| descricao\_Cham | VARCHAR | Sim | 1 byte | - | Descrição do problema informado |
| solucaoFinal\_Cham | VARCHAR | Não | 1 byte | - | Solução final aplicada para resolver o chamado |
| tecResponsavel\_Cham | VARCHAR | Não | 1 byte | FK | Nome do técnico responsável pelo chamado |

Fonte: autoria própria.

Além do dicionário de dados de cada tabela, foi feito o dicionário de dados para cada relacionamento: os relacionamentos entre as entidades e o relacionamento entre os relacionamentos:

**Tabela X – Relacionamento entre entidades.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Dicionário de Dados dos Relacionamentos** | | | |
| **Tabela** | **Relacionamento** | **Nome do Relacionamento** | **Descrição** |
| tbl\_Cliente | tbl\_Chamado | Abrir | Tabela para abertura de um chamado feito pelo cliente. |
| tbl\_Tecnico | tbl\_Chamado | Solucionar | Tabela para solução do técnico de um chamado |
| tbl\_ADM | tbl\_Relatorio | Gerar | Tabela para gerar um relatório feito pelo ADM |
| tbl\_Chamado | Gerenciar | Tabela para gerenciamento de um relatório feito pelo ADM |
| tbl\_Chamado | tbl\_Tecnico | Solucionar | Tabela para solução do técnico de um chamado |
| tbl\_Cliente | Abrir | Tabela para abertura de um chamado feito pelo cliente. |
| tbl\_ADM | Gerar | Tabela para gerar um relatório feito pelo ADM |
| Gerenciar | Tabela para gerenciamento de um relatório feito pelo ADM |
| tbl\_Relatorio | tbl\_ADM | Gerar | Tabela para gerenciamento de um relatório feito por um ADM |
| tbl\_Categoria | tbl\_Chamado | Possuir | Tabela para adicionar categorias à um chamado. |

Fonte: autoria própria

**Tabela X – Relacionamento entre relacionamentos.**

|  |  |  |  |
| --- | --- | --- | --- |
| Dicionário de Dados dos Relacionamentos | | | |
| Relacionamento | Tabela 1 - FK | Tabela 2 - PK | Descrição |
| Abrir | Chamado | Cliente | Relacionamento que descreve qual cliente abre cada chamado |
| Solucionar | Chamado | Tecnico | Relacionamento que descreve qual tecnico soluciona cada chamado |
| Gerenciar | Chamado | ADM | Relacionamento que descreve qual ADM gerencia cada chamado |
| Gerar | Relatorio | ADM | Relacionamento que descreve qual ADM gera cada relatorio |
| Possuir | Chamado | Categoria | Relacionamento que descreve qual categoria possuirá cada chamado |

Fonte: autoria própria

Após o processo de criação de tabelas, foram desenvolvidos os três tipos diagrama de entidade-relacionamento: o conceitual descritivo, o conceitual e o lógico. O conceitual descritivo foi feito para cada tabela e cada relacionamento, sendo este:

**Entidade: User. -** **Atributos:** id\_User (Primary Key), nome\_User, sobrenome\_User, email\_User, senha\_User, cargo\_User, departamento\_User, nívelAcesso\_User, nome\_User

**Entidade Cliente - Atributos:** id\_User (Primary Key e Foreign Key de User).

**Entidade Técnico - Atributos:** id\_User (Primary Key e Foreign Key de User).

**Entidade ADM - Atributos:** id\_User (Primary Key e Foreign Key de User).

**Entidade Chamado -** **Atributos:** id\_Cham (Primary Key), dataProblema, tecResponsavel\_Cham (Foreign Key), solucaoFinal\_Cham, descricao\_Cham, solucaoTec\_Cham, categoria\_Cham, prioridade\_Cham, dataFechamento\_Cham, dataAbertura\_Cham, status\_Cham.

**Entidade** **Relatório -** **Atributos:** id\_Relatorio (Primary Key), id\_Cham (Foreign Key), relatorio, id\_UsuarioCriador, dataCriacao.

**Relacionamentos:**

**Abrir (entre Cliente e Chamado):**

Descrição: Um cliente pode abrir nenhum ou vários chamados;

Um chamado é aberto por um único cliente;

Cardinalidade: 1:N (Cliente - Chamado).

**Solucionar (entre Técnico e Chamado)**

**Descrição:** Um técnico pode solucionar nenhum ou vários chamados;

Um chamado é solucionado por um único técnico;

Cardinalidade: 1:N (Técnico - Chamado)

**Gerar (entre ADM e Relatório)**

**Descrição:** Um ADM pode gerar nenhum ou mais relatórios;

Um relatório é gerado por um ADM;

Cardinalidade: 1:N (ADM - Relatório).

**Gerencia (entre ADM e Chamado)**

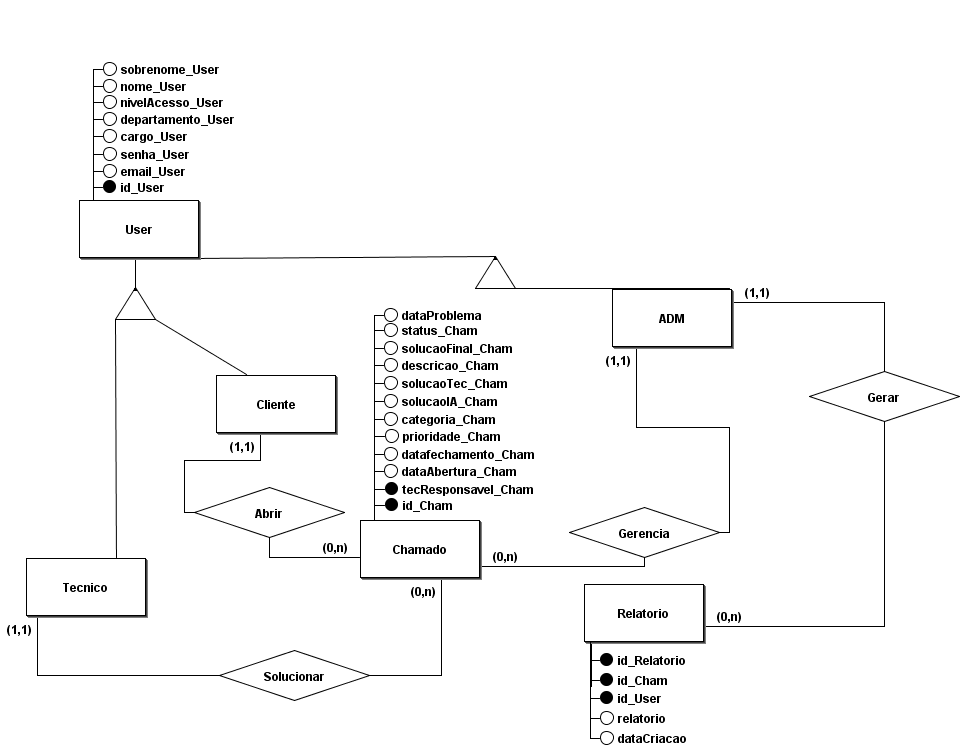
**Descrição:** Um ADM pode gerenciar nenhum ou mais chamados;

Cada ADM pode gerenciar um chamado;

Cardinalidade: 1:1 (ADM - Chamado).

O DER conceitual visual deve seguir as informações à risca do conceitual descritivo, e após a elaboração do mesmo no aplicativo brModelo, obteve-se tal imagem:

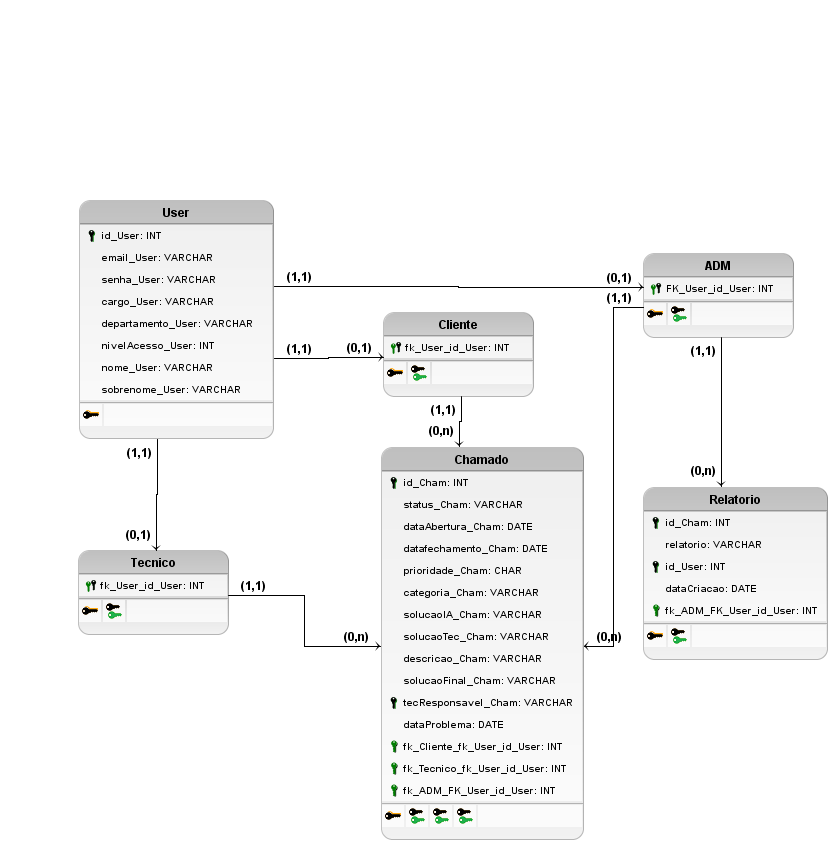
**Figura X – Diagrama de Entidade-Relacionamento Conceitual Visual.**

****

Fonte: autoria própria.

Com o DER Conceitual em mãos, foi gerado o modelo lógico, facilitando uma posterior criação do banco de dados físicos, utilizando o SQL Server:

**Figura X –Diagrama de Entidade-Relacionamento Lógico.**



Fonte: autoria própria

Todas as etapas anteriores são primordiais para a criação de um código em linguagem SQL, visto que realizando esses processos, haverá uma padronização e formalização do banco de dados, evitando que haja erros no processo de criação ou de execução. Quando um defeito é identificado de forma precoce em uma dar etapas anteriores, é simples ajustar o que estava equivocado e seguir o processo de uma forma linear.

Por fim, foi desenvolvido a parte física do banco de dados: o código em linguagem SQL Server. O principal objetivo do script é criar as tabelas necessárias para o funcionamento do sistema, definindo suas colunas, tipos de dados e relacionamentos, garantindo uma estrutura eficiente e organizada para o armazenamento das informações. Inicialmente, é criada a tabela Usuario contendo os campos id, email, senha, cargo, departamento, nível de acesso, nome e sobrenome. O campo id atua como chave primária e como ponte entre a tabela Usuario e as demais tabelas especializadas, além disso, duas triggers são definidas:

* trg\_InserirUsuario: insere automaticamente os dados do usuário na tabela correspondente ao seu tipo (Cliente, Técnico ou Administrador) após um INSERT na tabela Usuario.
* trg\_Usuario\_PadronizarCargo: padroniza automaticamente os textos dos campos de cargos, garantindo consistência na entrada de dados.

Em seguida, são criadas as tabelas Cliente, Tecnico e Administrador, cada uma contendo apenas o campo id\_user, que é chave primária da respectiva tabela e chave estrangeira referenciando o id da tabela Usuario. Essa separação por tipo de usuário permite, futuramente, o armazenamento de dados específicos e exclusivos para cada categoria de usuário.

Posteriormente, é criada a tabela Chamado, com os campos status, data\_abertura, data\_fechamento, data\_problema, prioridade, categoria, descricao, solucao\_ia, solucao\_tecnico, solucao\_final e tecnico\_responsavel, sendo este último uma chave estrangeira que referencia o campo id\_user da tabela Tecnico. A tabela Chamado conta com restrições (constraints) para garantir a integridade dos dados:

* CK\_Status\_Chamado: valida que apenas valores de status permitidos sejam inseridos.
* CK\_Prioridade\_Chamado: valida os valores de prioridade permitidos.

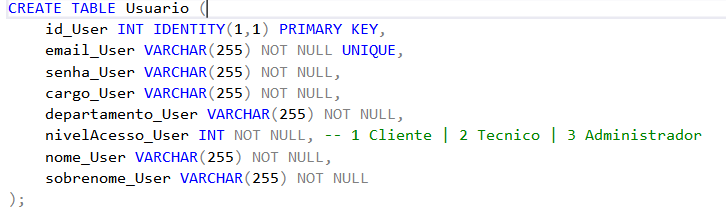
Além disso, foi implementada a trigger trg\_NormalizaStatusChamado, que padroniza automaticamente os textos inseridos nos campos status e prioridade. Por exemplo, se o usuário inserir "EM anDAMENTO", o sistema normaliza e armazena como "Em andamento. Por fim, é criada a tabela Relatorio, composta pelos campos id\_relatorio, id\_chamado, relatorio, data\_criacao e id\_usuario\_criador. Desses, dois são chaves estrangeiras:

* id\_chamado referencia o campo id\_chamado da tabela Chamado;
* id\_usuario\_criador referencia o campo id\_user da tabela Usuario.

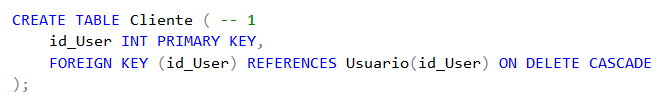
Para garantir que apenas usuários do tipo **Administrador** possam criar relatórios, foi definida a trigger trg\_ValidarRelatorioADM. Essa trigger verifica se o valor de id\_usuario\_criador corresponde a um usuário da tabela Usuario que esteja registrado como Administrador, bloqueando a inserção caso contrário. A seguir estarão imagens do código SQL, todas com autoria própria

**Codigos SQL para a criação das tabelas:**

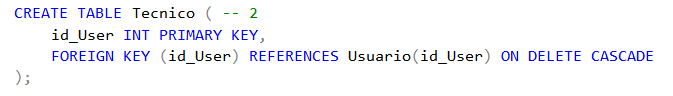
**Figura X – Tabela Usuario.**



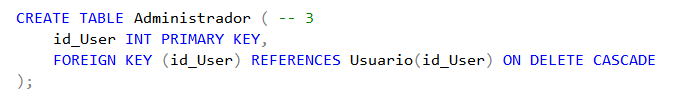
**Figura X – Tabela Cliente.**



**Figura X – Tabela Tecnico.**



**Figura X – Tabela Administrador.**



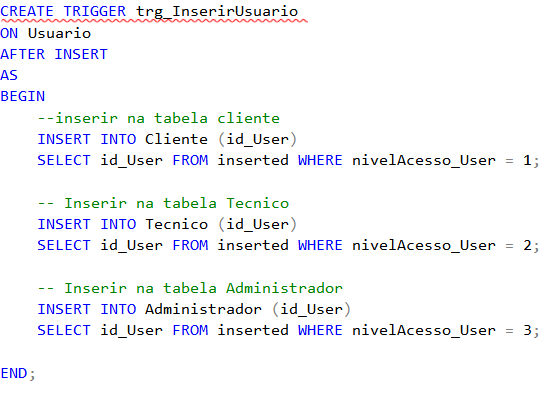
**Figura X – INSERT para povoar as tabelas.**



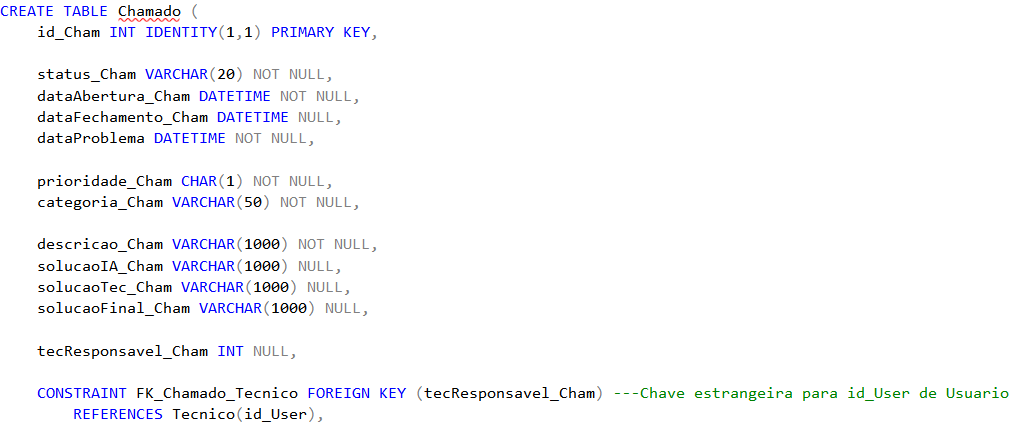
**Figura X – Resultado da seleção: SELECT \* FROM Usuario.**



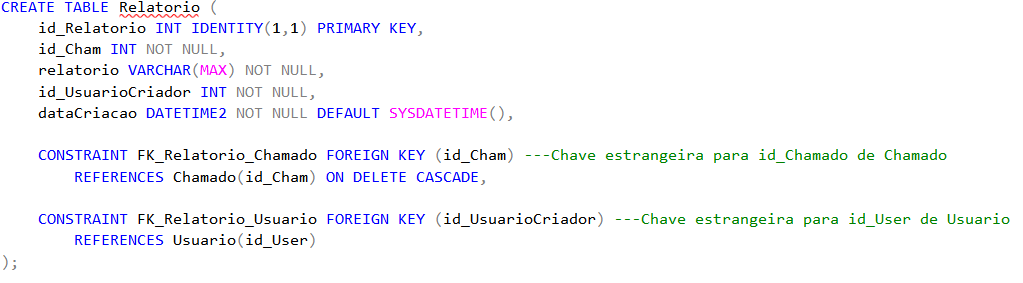
**Figura X – Tabela TRIGGER para a inserção automática**



**Figura X – Tabela Chamado**



**Figura X – Tabela Relatorio**



No modelo conceitual visual, o atributo da tabela relatório era id\_User, porém foi criado outro nome (id\_UsuarioCriador) referenciando a PK id\_User, melhorando a manutenibilidade do script. A preocupação com a integridade dos dados e a separação por papéis de usuários reforça o compromisso com a escalabilidade e a clareza do sistema. Com essa estrutura sólida, o sistema está preparado para sustentar as funcionalidades previstas e permitir futuras evoluções com consistência e eficiência.