

**Pró-Reitoria Acadêmica  
Curso de Ciências da Computação  
Trabalho de Disciplina de Laboratório de Banco de  
Dados**

**CRIANDO UM MODELO DER PARA UM BANCO DE DADOS  
DO CAMPO FINANCEIRO – CONTROLE DE DOCUMENTOS E  
PROVISIONAMENTO**

**Autor(es): Ana Beatriz Gonçalves, Brenda Mykaelle e  
Paulo Higa  
Orientador: Prof. Jefferson Rodrigues**

**ANA BEATRIZ GONÇALVES, BRENDA MYKAELLE E PAULO HIGA**

**CRIANDO UM MODELO DER PARA UM BANCO DE DADOS DO CAMPO  
FINANCEIRO – CONTROLE DE DOCUMENTOS E PROVISIONAMENTO**

Documento apresentado ao Curso de graduação de Bacharelado em Ciência da Computação da Universidade Católica de Brasília, como requisito parcial para obtenção da aprovação nas disciplinas de Laboratório de Banco de Dados.

Orientador: Prof. Jefferson Rodrigues

**Brasília  
2025**

HIGA, Paulo – aluno; MYKAELLE, Brenda – aluna;  
BEATRIZ, Ana – aluna.

CRIANDO UM MODELO DER PARA UM BANCO DE  
DADOS DO CAMPO FINANCEIRO – CONTROLE DE  
DOCUMENTOS E PROVISIONAMENTO. 29 de Agosto  
de 2025.

Universidade Católica de Brasília, 29 de Agosto de 2025.  
Orientação: Prof. Jefferson Rodrigo.

## RESUMO

Referência: HIGA, Paulo; MYKAELLE, Brenda; BEATRIZ, Ana. Criando um modelo der para um banco de dados do campo financeiro – controle de documentos e provisionamento, 2024. nr p. Bacharelado em Ciência da Computação – UCB – Universidade Católica de Brasília, Taguatinga – DF, 2025.

Este documento descreve a estrutura do banco de dados para a etapa inicial de um Sistema de Gestão de Provisionamento Fiscal Automatizado. O projeto trata da complexidade da gestão manual de documentos fiscais, faturas e despesas, oferecendo uma solução sólida e escalável. A análise de requisitos funcionais e não funcionais, além da criação dos modelos conceitual, lógico e físico com ceitos do MER e DER fez parte da metodologia de desenvolvimento. Como resultado, temos um banco de dados relacional com 21 tabelas, projetadas para assegurar a integridade dos dados e a proteção das informações. A arquitetura executa o ciclo de vida de uma despesa, desde a importação automática do documento fiscal até a criação de lançamentos contábeis e gerenciamento de pagamentos, empregando conceitos avançados de modelagem, como entidades fortes, fracas, associativas e especialização. O modelo final atua como uma base robusta para a construção das camadas de back-end e front-end da Aplicação a ser desenvolvida e completada, promovendo a automação de processos, a transparência e a conformidade fiscal.

Palavras-chave: Provisionamento, Entidades, Tabelas, Modelo DER, Fiscal.

## **ABSTRACT**

*This document presents the database architecture for the initial phase of an Automated Fiscal Provisioning Management System. The project addresses the challenges of manual management of fiscal documents, invoices, and expenses by proposing a robust and scalable solution. The development methodology included the analysis of functional and non-functional requirements, followed by the design of conceptual, logical, and physical DER and MER based models. The outcome is a comprehensive relational database with 21 tables, structured to ensure data integrity, security, and scalability. The architecture implements the lifecycle of an expense, from the automatic import of fiscal documents to the generation of accounting entries and payment control. It leverages advanced modeling concepts such as strong, weak, associative, and specialized entities. The final model serves as a solid foundation for the development of the application's back-end and front-end layers, enabling process automation, financial transparency, and fiscal compliance.*

*Keywords: Provisioning, Entities, Tables, DER Model, Fiscal.*

## LISTA DE TABELAS

Tabela 1. Cardinalidades..... **Erro! Indicador não definido.**

## SUMÁRIO

|  |           |
|--|-----------|
| <b>RESUMO.....</b>   | <b>4</b>  |
| <b>ABSTRACT.....</b>   | <b>5</b>  |
| <b>LISTA DE TABELAS .....</b>  | <b>6</b>  |
| <b>1 INTRODUÇÃO.....</b>   | <b>8</b>  |
| 1.1 MOTIVAÇÃO .....  | 8         |
| 1.2 DIAGNÓSTICO DA EMPRESA/TEMA.....   | 9         |
| <b>2 OBJETIVOS.....</b>  | <b>10</b> |
| 2.1 OBJETIVO GERAL .....   | 10        |
| 2.2 OBJETIVOS ESPECÍFICOS .....  | 10        |
| <b>3 DESENVOLVIMENTO .....</b>   | <b>11</b> |
| 3.1 ENTENDENDO REQUISITOS, REFLEXÕES .....   | 11        |
| <b>4 REQUISITOS FUNCIONAIS.....</b>  | <b>13</b> |
| <b>5 REQUISITOS NÃO-FUNCIONAIS.....</b>  | <b>15</b> |
| <b>6 REGRAS DE NEGÓCIO.....</b>  | <b>17</b> |
| <b>7 O MODELO DER DE PETER CHEN .....</b>  | <b>19</b> |
| 7.1 CARDINALIDADES.....  | 20        |
| <b>8 EXPLICANDO O BANCO DE DADOS CRIADO NO MYSQL WORKBENCH DE ACORDO COM OS CRITÉRIOS E CONCEITOS .....</b>          | <b>22</b> |
| <b>9 AS TABELAS   RELACIONAMENTOS E ENTIDADES PRINCIPAIS – DIAGRAMA MODELO DER CONCEITUAL, LÓGICO E FÍSICO .....</b> | <b>24</b> |
| <b>10 OS TIPOS DE ATRIBUTOS E AS CHAVES ESTRANGEIRAS DO BANCO .....</b>  | <b>26</b> |
| <b>11 CONCLUSÃO.....</b>   | <b>32</b> |
| 11.1 MELHORIAS FUTURAS.....  | 33        |
| <b>REFERÊNCIAS .....</b>   | <b>34</b> |
| <b>GLOSSÁRIO .....</b>   | <b>35</b> |

## 1 INTRODUÇÃO

A computação e a tecnologia da informação (TI) passaram a ser essenciais para a competitividade e a eficácia das empresas contemporâneas. A informática não é apenas uma ferramenta de apoio; hoje, ela é o principal motor para otimizar processos, tomar decisões estratégicas baseadas em dados e gerenciar recursos de forma inteligente. Com a automação e digitalização, as empresas podem diminuir despesas operacionais, reduzir erros e direcionar suas equipes para tarefas que geram mais valor, fortalecendo sua posição em um mercado cada vez mais dinâmico.

Nesse cenário, os processos financeiros e fiscais, que costumam ser manuais e sujeitos a erros, constituem um desafio considerável para a saúde operacional de uma organização. Pesquisas indicam que mais de 60% dos erros de faturamento resultam da inserção manual de dados. Esses erros podem aumentar o custo de processamento por fatura em até 20%, impactar o fluxo de caixa e resultar em penalidades por atrasos nos pagamentos. Além de consumirem o tempo da equipe e prejudicarem a relação com fornecedores, os erros de entrada manual geram um risco significativo de multas e problemas em auditorias.

Este trabalho propõe a elaboração da arquitetura de um banco de dados para um sistema de gestão fiscal e contábil, que servirá como fundamento para uma solução de automação. Este projeto cria a base para um aplicativo que poderá administrar o ciclo completo de uma despesa. A arquitetura assegura a integridade dos dados e a proteção das informações por meio de um sólido modelo de dados relacional. A estrutura é projetada para proporcionar transparência e controle ao processo financeiro, desde a importação automática de documentos fiscais até a criação de lançamentos contábeis e gestão de pagamentos, auditorias e pesquisas, abordando diretamente as questões de erro e eficiência mencionadas anteriormente.

Para uma apresentação clara e sistemática, o presente estudo está dividido em oito capítulos principais, apresentando a ideia principal do Negócio, Requisitos, Regras, Modelos de Inspiração e muito mais.

### 1.1 MOTIVAÇÃO

A principal motivação para a automação sugerida por este trabalho decorre da ineficiência e dos perigos associados aos procedimentos fiscais financeiros manuais. Um dos principais motivos de erros operacionais é a gestão de notas fiscais e faturas de maneira convencional, que se baseia na inserção manual de dados. Pesquisas apontam que a digitação



manual de dados é responsável por mais de 60% dos erros de faturamento. Esses erros têm um efeito financeiro direto, podendo elevar o custo de processamento de uma fatura em até 20%, além de causarem atrasos nos pagamentos e multas. A ausência de um sistema centralizado e automatizado, além de consumir tempo precioso da equipe e gerar um risco significativo de multas em auditorias, compromete a visibilidade gerencial e a habilidade de fazer escolhas estratégicas.

## 1.2 DIAGNÓSTICO DA EMPRESA/TEMA

A empresa fictícia, intitulada PLANNERJAI (Planejamento e Inteligência Artificial). O Software/Site a ser desenvolvido tem como foco o campo de Finanças, conforme Tema sugerido em proposta pelo Professor da Universidade Católica de Brasília Jefferson.

A ideia principal que incentivou a escolha é a precariedade atual das ferramentas integradas aos Bancos nas Empresas. A falta de uma ferramenta eficaz, otimizada e capaz de diminuir erros humanos – que frequentemente ocorrem dentro das equipes integrantes gerenciadoras financeiras das grandes corporações – se apresenta como um problema, que afeta diretamente um dos pilares mais importantes de qualquer negócio, a “Capital”.

Vez que a solução foca na fraqueza da capacidade de gerenciamento financeiro adequado, buscamos implementar novas funcionalidades com base em caso de uso, e novas verificações feitas em cada ponto crítico da operação, com alertas que evitam erros do cotidiano. O Banco de Dados foi construído utilizando todos os campos entidades reais, a fim de possibilitar que os dados mais utilizados sejam compreendidos pelos usuários mais frequentes, sejam tratados e armazenados adequadamente, e que levantamentos para pesquisas e melhorias, bem como ajuda na definição de metas a longo, médio e curto prazo sejam possibilitadas através da ferramenta por parte dos Analistas de Negócio Financeiro e demais setores.

## 2 OBJETIVOS

Este capítulo tem como propósito fundamental definir os objetivos que guiaram a elaboração deste trabalho, estabelecendo as metas a serem alcançadas e os critérios de sucesso do projeto. A clareza dos objetivos assegura que todo o processo de análise e modelagem de dados seja conduzido de forma estratégica, alinhando as entregas técnicas às necessidades organizacionais apresentadas.

### 2.1 OBJETIVO GERAL

O objetivo geral deste trabalho é conceber e documentar a arquitetura de banco de dados para a primeira fase de um sistema de gestão fiscal e contábil, que suporte a automação do fluxo de provisionamento de despesas de uma organização.

### 2.2 OBJETIVOS ESPECÍFICOS

**Para atingir o objetivo geral, os seguintes objetivos específicos foram estabelecidos:**

- **Analisar e Documentar os Requisitos:** Identificar e formalizar os requisitos funcionais e não funcionais do sistema, bem como as regras de negócio que moldam o comportamento da aplicação e a integridade dos dados.
- **Elaborar o Modelo Conceitual e Lógico:** Criar o Diagrama Entidade-Relacionamento (DER) que represente as entidades do sistema, seus atributos e as relações de cardinalidade entre elas.
- **Desenvolver o Modelo Físico:** Gerar o script SQL completo para a criação das 21 tabelas do banco de dados, incluindo as chaves primárias, chaves estrangeiras, índices e restrições de integridade.
- **Produzir a Documentação Técnica:** Criar um dicionário de dados detalhado que descreva cada tabela, campo, tipo de dado e as restrições aplicadas, servindo como um guia completo para a implementação e manutenção do banco de dados.

### 3 DESENVOLVIMENTO

#### 3.1 ENTENDENDO REQUISITOS, REFLEXÕES

→ **Tratando-se primariamente de Contas Correntes e Contas de Investimento, utilizadas por Empresas de médio-grande porte para pagamentos e provisionamentos (processo de registro de gastos futuros à serem aprovados pelos Gerentes):**

Empresas podem trabalhar com dois tipos de Fluxo, seguindo as entidades:

- **Conta corrente** (onde se guarda o dinheiro para pagamentos).
- **Conta Investimento** (onde se guarda dinheiro de rendimento, a ser investido em novos Serviços e contratos).

#### **Fluxo 1 – Pagamento direto pela Conta Corrente**

1. Nota fiscal emitida – recebida do fornecedor.
2. Provisionamento – valor registrado no ERP.
3. Aprovação – gestores validam e aprovam a despesa.
4. Encaminhada à conta corrente – a despesa é vinculada à conta de pagamentos.
5. Pagamento efetuado – o ERP (sistema) gera o arquivo CNAB, envia ao banco e o banco “lê” o arquivo e agenda os pagamentos. Muitas vezes ainda precisa de autorização/aprovação dentro do internet banking pelos responsáveis cadastrados.

**Fluxo 2 – Pagamento via Conta Investimento**

1. Nota fiscal emitida – recebida do fornecedor.
2. Provisionamento – valor registrado no ERP.
3. Aprovação – gestores validam e aprovam a despesa.
4. Resgate da conta investimento – o financeiro solicita o resgate do valor aplicado para a conta corrente (pode ser automático ou manual, dependendo da política da empresa e do banco).
5. Pagamento efetuado pela conta corrente – com o valor já transferido da conta investimento, o sistema gera o arquivo CNAB, envia ao banco e o banco “lê” o arquivo e agenda os pagamentos. Muitas vezes ainda precisa de autorização/aprovação dentro do internet banking pelos responsáveis cadastrados.

## 4 REQUISITOS FUNCIONAIS

**O programa a ser projetado seguindo o Banco de Dados construído será capaz de:**

**[RF001]** → Cadastrar Empresa Cliente.

**[RF002]** → Permitir o cadastro dos usuários com campos obrigatórios como Nome, E-mail(institucional), Senha, Departamento, Role e CPF.

**[RF003]**→ Guardar informações dos Fornecedores.

**[RF004]**→ Guardar os dados necessários para o fluxo das Notas(Documentos) Fiscais Buscadas no CPF/CNPJ da empresa no banco de dados(seja da conta Investimento ou Empresarial) de acordo com as Entidades criadas.

**[RF005]** → Permitir Integração Futuro à ferramentas de Gerenciamento de usuários e departamentos (Como o ACTIVE DIRECTORY do Windows), com campos como user\_login com limite de 20 caracteres para não ocorrer divergência entre versões do Servidor Windows).

**[RF006]**→ Permitir inclusão automatizada de notas do tipo NF-e, NFS-e e CT-e, (porque o XML facilita a leitura automática), com base nos padrões da SEFAZ Brasileira e Nacional.

**[RF007]**→ Permitir inclusão parcialmente automatizada de Documentos do tipo Boleto, Contratos, Impostos, Despesas operacionais e Pagamentos diversos.

**[RF008]**→ Diferenciar, no Back-end, obrigatoriedades complementares dos campos obrigatórios de acordo com tipo de nota.

**[RF009]**→ Diferenciar, no Back-end, obrigatoriedades complementares dos campos obrigatórios de acordo com país de origem.

**[RF010]**→ Permitir pesquisas com facilidade para Levantamentos de Gastos.

**[RF011]** → Conter Restrições de segurança em alterações.

**[RF012]** → Verificar status do pagamento.

**[RF013]** → Verificar retorno do Banco.

**[RF014]** → Salvar Log dos usuários e alterações feitas através de auditorias, baseados em data e horário.

**[RF015]** → Promover restrições baseadas na role(função) dos usuários.

**[RF016]** → Restringir usuários a partir de um número de login sessão simultâneo(Na camada de aplicação).

## **5 REQUISITOS NÃO-FUNCIONAIS**

### **Requisitos inspirados em Regras ISO(Organização Internacional de Normalização):**

#### **[RNF001] → Performance – Consultas**

O banco deve responder a consultas simples (busca por documento fiscal por CNPJ, data ou chave de acesso) em até 10 segundos para bases com até 1 milhão de registros na tabela documentos\_fiscais.

#### **[RNF002] → Performance – Relatórios**

Relatórios consolidados (ex.: total de despesas por fornecedor em um período) devem ser processados em até 30 segundos em bases com até 1 milhões de registros.

#### **[RNF003] → Escalabilidade**

O banco deve suportar até 1 milhão de documentos fiscais em 5 anos, sem necessidade de reestruturação do modelo físico.

#### **[RNF004] → Disponibilidade**

O sistema de banco de dados deve estar disponível 99,5% do tempo, considerando janelas de manutenção planejadas.

#### **[RNF005] → Backup e Recuperação**

- Deve ser realizado backup completo diário com retenção de 30 dias.
- Deve haver backup incremental a cada 6 horas.
- O tempo máximo para restauração completa (RTO) não deve ultrapassar 1 hora.
- A perda máxima aceitável de dados (RPO) é de 15 minutos.

**[RNF006] → Segurança - Armazenamento de Senhas**

Todas as senhas devem ser armazenadas utilizando hash seguro (ex.: bcrypt, Argon2).

**[RNF007] → Segurança – Dados Sensíveis** Dados como CPF, CNPJ, dados bancários e chaves de acesso fiscais devem ser armazenados com criptografia em repouso (AES-256 ou equivalente).

**[RNF008] → Integridade dos Dados** O banco deve rejeitar documentos fiscais duplicados via chave de acesso única.

- O banco deve rejeitar documentos fiscais duplicados via chave de acesso única.
- Exclusões de registros em entidades principais (empresas, documentos\_fiscais, provisionamentos) devem ser feitas por inativação lógica (soft delete), preservando histórico.

**[RNF009] → Auditoria**

Todas as operações de inclusão, alteração e exclusão em tabelas críticas (documentos\_fiscais, provisionamentos, contas\_pagar) devem ser registradas em tabela de auditoria, com usuário, data, operação e valores alterados.

**[RNF010] → Conformidade Fiscal**

O armazenamento de documentos fiscais deve garantir integridade por meio de hash SHA-256, evitando adulteração de XMLs e anexos.



## **6 REGRAS DE NEGÓCIO**

**Regras de negócio são diretrizes ou instruções precisas que definem ou restringem a estrutura, o comportamento e as operações de uma empresa para atingir seus objetivos:**

### **[RN001] → Identificação Única de Documentos**

Cada documento fiscal é identificado de forma única por sua chave de acesso. O banco de dados impede a inserção de um documento com uma chave de acesso duplicada.

### **[RN002] → Identificação Única de Entidades**

Uma empresa, um fornecedor ou um usuário não podem ter um CNPJ/CPF ou e-mail duplicado garantindo a unicidade de cada entidade principal no sistema.

### **[RN003] → Workflow de Provisionamento**

Cada Documento Fiscal deve gerar uma e apenas uma Provisão. Por sua vez, cada Provisão deve gerar uma e apenas uma Conta a Pagar.

### **[RN004] → Regras de Aprovação**

As despesas de provisão devem ser aprovadas por um ou mais usuários, seguindo um fluxo que pode ser definido por limites de valor, categoria ou centro de custo

### **[RN005] → Rastreabilidade Contábil**

Cada Lançamento Contábil deve estar obrigatoriamente vinculado a um Provisionamento, garantindo que toda movimentação contábil possa ser rastreada até sua origem.

### **[RN006] → Proibição de Relações Inválidas**

O sistema impede que uma empresa seja sua própria matriz, garantindo a integridade da hierarquia empresarial.

**[RN007] → Cálculo Automático**

O valor líquido de um documento fiscal é sempre calculado de forma automática, sendo a diferença entre o valor total e o valor dos impostos.

**[RN008] → Organização de Custos**

Os custos de uma despesa podem ser alocados em vários Centros de Custo ou Projetos, e essa alocação é registrada para fins de controle e análise.

## 7 O MODELO DER DE PETER CHEN

O Modelo Entidade Relacionamento, conhecido como ER, ou MER), foi criado em 1976 por Peter Chen, Cientista da Computação nascido em TAIWAN. O Modelo e se trata de um modelo conceitual usado em diversas áreas da engenharia de software e relacionadas, usado para melhor ilustrar e descrever negócios, utilizando de conceitos como objetos (as Entidades) envolvidas em um domínio de negócios, com características (atributos) e seu relacionamento. O modelo, quando bem-construído, é capaz de refletir diretamente na estrutura de um Banco de dados do negócio a ser desenvolvido, ajudando na compreensão da lógica e atuação de cada entidade no fluxo e contexto daquela realidade espelhada.

No entanto, possuímos algumas diferenças entre as possibilidades de representação através do MER, que por sua é representando de forma Textual, e a Representação visual dos modelos em Diagramas Conceituais, Lógicos e Físicos, conhecida como DER (Diagrama Entidade-Relacionamento).

Um, dentre os vários outros conceitos criados pelo Cientista, foi o conceito de cardinalidade, a chave para a compreensão visual dos modelos essenciais às etapas de desenvolvimento do projeto. A Cardinalidade diz respeito a quantidade de ocorrências que uma Entidade-objeto pode ter em um relacionamento com uma outra Entidade - usando uma simples analogia de algo corriqueiro, ao comparecer a qualquer Panificadora para fazer um pedido, um Cliente pode comprar muitos Produtos, Pão, Café, Bolo, mas aquele produto em específico, sendo comprado pelo cliente, está sendo comprado somente por uma única pessoa, a qual irá pagar por aquele produto.

## 7.1 CARDINALIDADES

**A notação utilizada no modelo é a (min, max), que indicam:**

**Mínimo(min):** a quantidade mínima de vezes que uma Entidade deve participar do relacionamento (Pode ser 0 ou 1)

**Máximo(máx.):** a quantidade máxima de vezes que pode participar (1 ou N (que representa vários)).

Como no exemplo usado anteriormente, para ser cliente da panificadora, a entidade (Pessoa) precisa comprar no mínimo um produto da loja, não tendo limite para quantos itens irá comprar (1, N).

Um pedido que foi realizado pelo cliente, no estoque da loja, será relacionado somente à ele. A entidade (Produto) é comprada 1 vez para existir e é associada (ou relacionada) à exatamente um único cliente (Pessoa) (1:1).

Há, porém, outras combinações dessas cardinalidades como (0:N), (N:M), como a seguir:

**Notação:** (0, N)

**Leitura:** Pode não participar, mas pode participar de vários.

**Exemplo:** Um cliente pode nunca ter feito pedidos, ou ter muitos.

**Notação:** (1, N)

**Leitura:** Deve participar de pelo menos 1, até muitos.

**Exemplo:** Um aluno deve estar matriculado em pelo menos 1 disciplina, mas pode cursar várias.

**Notação:** (0, 1)

**Leitura:** Participação opcional e no máximo uma vez.

**Exemplo:** Um funcionário pode ou não ter um crachá.

**Notação:** (1, 1)

**Leitura:** Obrigatório e exatamente uma vez.

**Exemplo:** Todo pedido deve ser feito por um cliente (e apenas um).

**Notação:** (N, M) ou (min, N)  $\leftrightarrow$  (min, M)

**Leitura:** Muitos para muitos. Cada ocorrência de uma entidade pode se relacionar com várias da outra, e vice-versa.

**Exemplo:** Um aluno pode cursar várias disciplinas, e cada disciplina pode ter vários alunos.

## 8 EXPLICANDO O BANCO DE DADOS CRIADO NO MYSQL WORKBENCH DE ACORDO COM OS CRITÉRIOS E CONCEITOS

### Entidades Fortes -

**empresas:** a empresa cliente.

**roles:** função para perfis de usuário da empresa.

**usuarios:** usuários da empresa.

**fornecedores:** quem emite os documentos.

**empresa\_fornecedor:** permite que um fornecedor esteja associado a mais de uma empresa cadastrada.

**centros\_custo:** os departamentos da empresa.

**categorias\_contabeis:** o plano de contas, categorias definidas pela empresa.

**projetos:** iniciativas com prazo.

**documentos\_fiscais:** documentos da empresa que o software irá lidar.

**provisionamentos:** a “promessa” de despesa.

**contas\_pagar:** o compromisso financeiro.

**lancamentos\_contabeis:** registro contábil.

**anexos:** os arquivos que podem ser anexados nas etapas do fluxo.

**workflows\_aprovacao:** as regras de aprovação definidas pela empresa para gastor que podem precisar de aprovação de usuários com roles específicas.

**remessas\_cnab:** arquivos de pagamento enviados ao banco.

### Entidades Fracas - (dependentes)

**empresa\_contatos:** não existe sem a empresa.

**user\_sessions:** não existe sem o usuário.

**retornos\_cnab:** não existe sem a remessa\_cnab correspondente.

### Entidades Associativas -

**documento\_centro\_custo:** conecta documentos\_fiscais a centros\_custo.

**provisionamento\_projetos:** conecta provisionamentos a projetos.

**documento\_anexos:** conecta documentos\_fiscais a anexos.

**aprovacoes:** conecta provisionamentos a usuários

### Entidades de Especialização -

**fornecedores:** atua como uma superclasse que pode ser de dois tipos especializados: Pessoa Jurídica ou Pessoa Física. Essa diferenciação é feita através do *campo tipo\_pessoa* existente na tabela.

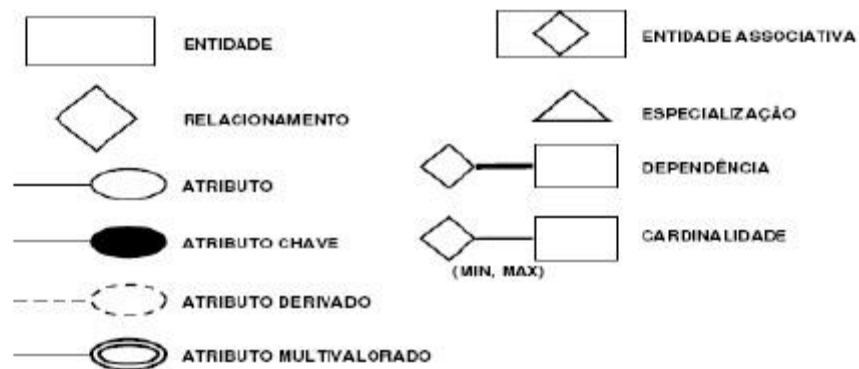
### Cardinalidades apresentadas no Banco:

| Entidade           | Relacionamento         | Cardinalidade | Entidade              |
|--------------------|------------------------|---------------|-----------------------|
| Empresas           | gerenciam              | 1:N           | Usuários              |
| Empresas           | definem                | 1:N           | Fornecedores          |
| Empresas           | definem                | 1:N           | Centros de Custo      |
| Empresas           | definem                | 1:N           | Projetos              |
| Fornecedores       | emitem                 | 1:N           | Documentos Fiscais    |
| Documentos Fiscais | geram                  | 1:1           | Provisionamentos      |
| Provisionamentos   | criam                  | 1:N           | Lançamentos Contábeis |
| Provisionamentos   | indicam                | 1:1           | Contas a Pagar        |
| Documentos Fiscais | se associa a           | N:M           | Centros de Custo      |
| Documentos Fiscais | contem                 | N:M           | Anexos                |
| Provisionamentos   | tem custo alocado para | N:M           | Projetos              |
| Provisionamentos   | requer aprovação de    | N:M           | Usuários              |
| Remessas CNAB      | contém                 | 1:N           | Contas a Pagar        |
| Remessas CNAB      | gera                   | 1:N           | Retornos CNAB         |
| Empresas           | possuem                | 1:N           | Empresa Contatos      |
| Usuários           | possuem                | 1:N           | User session          |

## 9 AS TABELAS | RELACIONAMENTOS E ENTIDADES PRINCIPAIS – DIAGRAMA MODELO DER CONCEITUAL, LÓGICO E FÍSICO

Ao criar as representações gráficas, seguem-se as seguintes Notações:

### Notação Peter Chen



Citado anteriormente, o Modelo conceitual é um Diagrama cujo foco é conceitual, isto é, ele conceitua e apresenta atributos principais da lógica do fluxo do sistema, atributos das Entidades do tipo UNIQUE ou Identificadores do negócio.

Um Identificador de Negócio (UNIQUE) é um campo da Tabela (Entidade) criada, como Cnpj ou chave\_acesso. O Cnpj é o que uma pessoa de negócios usaria para identificar um registro (de uma Empresa), em nosso contexto. O Identificador é UNIQUE é o identificador que tem significado no mundo real.

O Modelo Lógico, por sua vez usa Identificadores Técnicos. Identificadores Técnicos são as Primary Keys ou Id's, chaves artificiais criadas pelo desenvolvedor para ser um identificador único simples e eficiente para o banco de dados. Ele não tem um significado fora daquele contexto.

Já no Modelo Físico, a implementação final, é feita uma junção do modelo lógico, de forma que o Banco é transformado em um código executável, sua representação final inclui todos os detalhes técnicos do banco de dados: tipos de dados (VARCHAR, INT, DECIMAL),



limites (VARCHAR (255)), e as restrições de integridade (NOT NULL, UNIQUE, ON DELETE CASCADE), bem como as Primary e Foreign Keys.

**Os Modelos CONCEITUAL, LÓGICO e FÍSICO foram criados para melhor explicação do Banco de Dados para o Negócio construído e serão anexados juntos em PDF na Entrega da Atividade.**

**Modelos ARQUIVOS**

## 10 OS TIPOS DE ATRIBUTOS E AS CHAVES ESTRANGEIRAS DO BANCO

Ao modelarmos o banco de dados, selecionamos os tipos de dados para cada atributo, considerando a integridade e a flexibilidade. Utilizando INT AUTO\_INCREMENT para os identificadores únicos das tabelas, assegurando chaves primárias simples e eficazes. Para campos textuais curtos, como nomes, códigos e siglas (nome, codigo, uf), foram empregados o tipo VARCHAR com tamanhos apropriados. Para textos extensos, como descrições, observações ou conteúdo de XML (descricao, observacoes, xml\_content), utilizamos os tipos TEXT ou LONGTEXT, possibilitando o armazenamento de grandes quantidades de texto sem restringir a informação.

Para valores monetários, como orçamentos, valores de documentos ou provisionamentos (valor\_total, valor\_impostos, valor\_provisionado), foram utilizados DECIMAL(15,2) para evitar problemas de arredondamento e garantir precisão financeira. Para datas e horários, como emissão, vencimento ou criação de registros (data\_emissao, created\_at, updated\_at), utilizamos DATE ou TIMESTAMP, garantindo rastreabilidade com base no tempo.

Os atributos que possuem listas fixas de opções, como tipo\_documento, status\_processamento, tipo\_pessoa e status de aprovação, empregamos o ENUM, garantindo que somente valores válidos sejam inseridos. Utilizei JSON para campos que contêm informações estruturadas, porém flexíveis, como permissoes\_base ou dados\_bancarios, o que possibilita a adição ou remoção de dados sem a necessidade de modificar a estrutura da tabela. Por último, para atributos que indicam valores binários de verdadeiro ou falso (como ativo, principal, is\_active), utilizei o tipo BOOLEAN, o que torna a leitura do código mais clara e eficiente.

Além disso, utilizei CHAR para campos de tamanho fixo, como CEP e CPF, assegurando a padronização do tamanho e a otimização do armazenamento. E para hashes de arquivos (hash\_arquivo), optei pelo tipo VARCHAR(64), que é adequado para armazenar um SHA-256.

### Atributos Simples:

Atributos simples (ou atômicos) são características de uma entidade que não podem ser subdivididas em partes menores para fins de representação.

Campo **razao\_social** da tabela **empresas**: Será preenchido como "Empresa de Software".

Campo **valor\_total** da tabela **documentos\_fiscais**: Será preenchido como "1500.25".

Campo **data\_emissao** da tabela **documentos\_fiscais**: Será preenchido como "2025-08-28".

Campo **email** da tabela **usuarios**: Será preenchido como "joao.silva@empresa.com".

### Atributos Compostos:

Estes atributos, embora armazenados em colunas separadas, formam um único conceito.

Campo de **Endereço** na tabela **fornecedores**: Um único endereço será preenchido nas colunas `endereco_logradouro` como "Rua das Flores", `endereco_numero` como "123", e `endereco_cidade` como "São Paulo".

### Atributos Multivalorados:

Esses atributos podem ter vários valores para um único registro, e sua estrutura no banco de dados lida com essa complexidade.

**Dados Bancários** na tabela **fornecedores**: O campo `dados_bancarios`, do tipo JSON, pode ser preenchido para armazenar múltiplas contas de um mesmo fornecedor, como "[{ "banco": "001", "agencia": "1234", "conta": "5678" }, { "banco": "237", "agencia": "5678", "conta": "1234" }]".

**Contatos** na tabela **empresas**: A tabela `empresa_contatos` resolve o problema de ter vários contatos. Você terá um registro principal para o e-mail "contato@empresa.com.br", e um segundo registro para o telefone "(61) 9999-8888", ambos vinculados à mesma empresa.

## Chaves Primárias e Compostas

As chaves são os identificadores únicos.

**Chave Primária:** Os campos “id” de uma tabela como Empresas será preenchido com um valor único como "1", "2", "3" e assim por diante.

**Chave Composta:** Na tabela **documento\_centro\_custo**, a chave composta que une o documento\_fiscal\_id e o centro\_custo\_id será preenchida com valores como ("1", "10") ou ("1", "12"), indicando que o mesmo documento pode ser rateado para diferentes centros de custo.

## Lista Completa de Chaves Estrangeiras (FKs)

### Tabelas – campos - FK(Foreign Key) - Tabela referenciada...

#### **empresas:**

**empresa\_matriz\_id** → FK que referencia **id** na própria tabela **empresas**.

#### **usuarios:**

**empresa\_id** → FK que referencia **id** na tabela **empresas**;

**role\_id**: → FK que referencia **id** na tabela **roles**.

#### **empresa\_fornecedor:**

**empresa\_id** → FK que referencia **id** na tabela **empresas**.

**fornecedor\_id** → FK que referencia **id** na tabela **fornecedores**.

#### **centros\_custo:**

**empresa\_id** → FK que referencia **id** na tabela **empresas**;

**centro\_pai\_id** → FK que referencia **id** na própria tabela **centros\_custo**.

#### **categorias\_contabeis:**

**empresa\_id** → FK que referencia **id** na tabela **empresas**.

#### **projetos:**

**empresa\_id** → FK que referencia **id** na tabela **empresas**;

**responsavel\_id** → FK que referencia **id** na tabela **usuarios**.

#### **anexos:**

**uploaded\_by** → FK que referencia **id** na tabela **usuarios**.

#### **documentos\_fiscais:**

**empresa\_id** → FK que referencia **id** na tabela **empresas**;

**fornecedor\_id** → FK que referencia **id** na tabela **fornecedores**;

**usuario\_criacao\_id** → FK que referencia **id** na tabela **usuarios**.

#### **provisionamentos:**

**documento\_fiscal\_id** → FK que referencia **id** na tabela **documentos\_fiscais**;

**empresa\_id** → FK que referencia **id** na tabela **empresas**;

**fornecedor\_id** → FK que referencia **id** na tabela **fornecedores**;

**categoria\_contabil\_id** → FK que referencia **id** na tabela **categorias\_contabeis**;

**usuario\_criacao\_id** → FK que referencia **id** na tabela **usuários**;  
**usuario\_aprovacao\_id** → FK que referencia **id** na tabela **usuarios**.

#### **contas\_pagar:**

**provisionamento\_id** → FK que referencia **id** na tabela **provisionamentos**;  
**empresa\_id** → FK que referencia **id** na tabela **empresas**;  
**fornecedor\_id** → FK que referencia **id** na tabela **fornecedores**;  
**remessa\_id** → FK que referencia **id** na tabela **remessas\_cnab**.

#### **lancamentos\_contabeis:**

**provisionamento\_id** → FK que referencia **id** na tabela **provisionamentos**;  
**empresa\_id** → FK que referencia **id** na tabela **empresas**;  
**centro\_custo\_id** → FK que referencia **id** na tabela **centros\_custo**;  
**usuario\_lancamento\_id** → FK que referencia **id** na tabela **usuarios**.

#### **documento\_centro\_custo:**

**documento\_fiscal\_id** → FK que referencia **id** na tabela **documentos\_fiscais**;  
**centro\_custo\_id** → FK que referencia **id** na tabela **centros\_custo**;  
**usuario\_criacao\_id** → FK que referencia **id** na tabela **usuarios**.

#### **provisionamento\_projetos:**

**provisionamento\_id** → FK que referencia **id** na tabela **provisionamentos**;  
**projeto\_id** → FK que referencia **id** na tabela **projetos**;  
**usuario\_criacao\_id** → FK que referencia **id** na tabela **usuarios**.

#### **documento\_anexos:**

**documento\_fiscal\_id** → FK que referencia **id** na tabela **documentos\_fiscais**;  
**anexo\_id** → FK que referencia **id** na tabela **anexos**.

#### **aprovacoes:**

**provisionamento\_id** → FK que referencia **id** na tabela **provisionamentos**;  
**workflow\_id** → FK que referencia **id** na tabela **workflows\_aprovacao**;  
**usuario\_aprovador\_id** → FK que referencia **id** na tabela **usuarios**.

#### **empresa\_contatos:**

**empresa\_id** → FK que referencia **id** na tabela **empresas**.

**user\_sessions:**

**user\_id** → FK que referencia **id** na tabela **usuários**;

**empresa\_id** → FK que referencia **id** na tabela **empresas**.

**remessas\_cnab:**

**empresa\_id** → FK que referencia **id** na tabela **empresas**;

**usuario\_geracao\_id** → FK que referencia **id** na tabela **usuarios**.

**retornos\_cnab:**

**remessa\_id** → FK que referencia **id** na tabela **remessas\_cnab**;

**usuario\_processamento\_id** → FK que referencia **id** na tabela **usuarios**.

## 11 CONCLUSÃO

Este trabalho concluiu com sucesso a primeira fase do projeto de desenvolvimento de um sistema de gestão fiscal e contábil, alcançando plenamente os objetivos propostos. A arquitetura de banco de dados, que é o pilar de sustentação da aplicação, foi concebida e documentada de maneira abrangente e precisa. Os resultados obtidos demonstram uma abordagem metodológica rigorosa, que começou com a análise aprofundada dos problemas e se encerrou com a criação de um blueprint técnico e funcional para a solução.

Todos os objetivos específicos foram atendidos de forma satisfatória. Foram identificados e formalizados os requisitos funcionais e não funcionais do sistema, o que incluiu a definição de regras de negócio essenciais. A partir desses requisitos, foram elaborados os modelos conceitual, lógico e físico, que traduzem a lógica de negócio em uma estrutura de banco de dados relacional. A entrega do script SQL completo e de um dicionário de dados detalhado valida a conclusão de cada etapa do projeto.

Como resultado, a organização agora possui uma arquitetura de dados robusta, escalável e otimizada para a automação de processos críticos. O modelo, composto por 21 tabelas, utiliza de forma eficaz conceitos como entidades fortes e fracas, relacionamentos 1:1, 1:N e N:M, e especialização, para garantir a integridade e a consistência das informações. Essa estrutura está pronta para mitigar os riscos associados a erros manuais, proporcionando maior segurança, auditabilidade e transparência em todas as operações financeiras.

Em suma, este projeto estabeleceu a fundação necessária para as próximas etapas de desenvolvimento a serem desenvolvidos na matéria de Laboratório de Banco de Dados. O sucesso na construção do banco de dados assegura que a aplicação completa, a ser desenvolvida, terá um alicerce sólido para suportar a complexidade do negócio, promovendo a eficiência



### 11.1 MELHORIAS FUTURAS

Para melhorias futuras do banco de dados, algumas melhorias são importantes. Começamos integrando ferramentas robustas de gestão, como o Active Directory do Windows, que organiza usuários e recursos da rede, e o Oracle Database, um sistema poderoso para armazenar dados da empresa com segurança.

Em seguida, automatizamos as notas fiscais, inclusive as menos frequentes, com triggers para agilizar processos e reduzir erros. E, por fim, planejamos o futuro com um back-end eficiente, camadas de aplicação bem estruturadas (como restrições e verificações adicionais de integridade dos dados) e automação com inteligência artificial para analisar dados, prever tendências e otimizar decisões.

Ademais, páginas de Simulação financeira poderiam ser incluídas, bem como pesquisas diretas do Banco de Dados de forma interna e sugestiva junto da Inteligência Artificial.

## REFERÊNCIAS

Artigo sobre Automação e Escalabilidade (Tecnologia):

GOMES, J. V. Escalabilidade e Manutenção: O Uso do JSON e Chaves Compostas na Modelagem de Dados Moderna. Dev Magazine, [S.l.], 5 jul. 2024. Disponível em:

<https://www.devmagazine.com.br/modelagem-avancada>. Acesso em: 28 set. 2025.

CONTABILIZEI. Modelo de nota fiscal (NF): de serviço, eletrônica, simples. Contabilizei – contabilidade online. Disponível em: <https://www.contabilizei.com.br/contabilidade-online/modelo-de-nota-fiscal/>

.Acesso em: 29 ago. 2025.

LAUREANO, Marcos A. P.; MORAES, Paulo E. S. Segurança como estratégia de gestão da informação. Revista Economia & Tecnologia, ISSN 1415-451X, v. 8, fasc. 3, p. 38-44, 2005.

NFE.IO. Dados para emissão de nota fiscal: conheça os necessários! NFE.io. Disponível em:

<https://nfe.io/blog/nota-fiscal/dados-necessarios-para-emissao-de-nota-fiscal/> .Acesso em: 29 ago. 2025.

OMIE. Quais são as informações obrigatórias para emitir uma Nota Fiscal de Produto (NF-e)? Ajuda

Omie. Disponível em: <https://ajuda.omie.com.br/pt-BR/articles/498846-quais-sao-as-informacoes-obrigatorias-para-emitir-uma-nota-fiscal-de-produto-nf-e> . Acesso em: 29 ago. 2025.

Relatório sobre Integridade de Dados (3NF/OLTP):

SILVA, R. A. Arquitetura de Banco de Dados para Alta Integridade: Por Que o 3NF é Crucial em Sistemas Financeiros. TechLead Consultoria, [S.l.], 15 mar. 2024. Disponível em:

<https://www.techlead.com.br/artigos/integridade-3nf>. Acesso em: 28 set. 2025.

Reportagem sobre Custos de Erro (Motivação): FINANCE TODAY EDITORIAL. O Preço da Digitação:

Como Erros Manuais Aumentam o Custo por Fatura em 20%. Finance Today, [S.l.], 10 fev. 2023.

Disponível em: <https://www.financetoday.com/custo-erros-manuais>. Acesso em: 28 set. 2025.

WIKIPÉDIA. Modelo entidade relacionamento. Wikipédia, a enciclopédia livre. Disponível em:

[https://pt.wikipedia.org/wiki/Modelo\\_entidade\\_relacionamento](https://pt.wikipedia.org/wiki/Modelo_entidade_relacionamento) . Acesso em: 29 ago. 2025.

## **GLOSSÁRIO**

### **AES-256**

Algoritmo de criptografia simétrica que opera em blocos de 128 bits, usando chave de 256 bits, realizando 14 rodadas de processamento.

### **ARGON2**

Função de hash para senhas, vencedora do PHC 2015, que utiliza memória alta para dificultar ataques de força bruta; variantes: Argon2d, Argon2i e Argon2id.

### **BCRYPT**

Função de hash para senhas baseada no Blowfish, com salting e fator de custo ajustável, resistente a ataques de força bruta.

### **CNAB**

Padrão da FEBRABAN para troca automática de informações financeiras entre empresas e bancos, com arquivos de remessa (.REM) e retorno (.RET).

### **ENTIDADE ASSOCIATIVA**

Entidade que representa relacionamentos complexos, geralmente N:N, permitindo armazenar atributos adicionais da relação entre outras entidades.

### **ENTIDADE ESPECIALIZADA**

Entidade que representa uma subcategoria de uma entidade mais geral, herdando atributos e podendo ter atributos ou relacionamentos próprios.

### **ENTIDADE FRACA**

Entidade que depende de uma entidade forte para existir e não possui chave primária própria; usa chave parcial combinada com a chave da entidade forte.

**ENTIDADE FORTE**

Entidade que possui chave primária própria e existe independentemente de outras entidades, representada por um retângulo único em diagramas ER.

**HASH SHA-256**

Função de hash que gera saída fixa de 256 bits, usada para integridade, autenticação e blockchain.

**ISO (ORGANIZAÇÃO INTERNACIONAL DE NORMALIZAÇÃO)**

Organização não governamental que cria normas internacionais para padronizar produtos, serviços e processos, promovendo qualidade, segurança e eficiência.

**NOTAS FISCAIS ELETRÔNICAS (NF-E, NFS-E, CT-E) E XML**

Documentos digitais que registram vendas de produtos (NF-e), serviços (NFS-e) e transporte de cargas (CT-e); XML é o formato oficial e legal.

**PRIMARY E FOREIGN KEY**

Primary key identifica unicamente cada registro; foreign key referencia a chave primária de outra tabela, garantindo integridade referencial.

**RATEADO**

Dividido proporcionalmente ou distribuído de forma justa entre partes, aplicado a lucros, prejuízos ou seguros.

**RAZÃO SOCIAL DE UMA EMPRESA**

Nome legal e oficial de uma empresa, registrado formalmente e usado em documentos legais, contratos e registros governamentais.



