



UNIVERSIDADE
FEDERAL DO CEARÁ
CAMPUS DE QUIXADÁ

Universidade Federal do Ceará - *campus* Quixadá

Projeto Integrado em Engenharia de Software I
Documento de Requisitos - Sistema de Gestão Financeira

Docente:

Jefferson Kennedy

Discentes:

Antônio Rewelli de Oliveira, 554047;
Giliardy Alves da Silva, 552752;
Lucas Ferreira Nobre, 554590;
Miqueias Bento da Silva, 553972.

Av. José de Freitas Queiroz, 5003, Quixadá - CE, 63902-580
14.09.2024

1. Introdução	3
O que é o projeto?	3
Escopo do projeto:	3
2. Requisitos funcionais	4
2.1 Gestão financeira	4
2.2 Controle e organização de dados	9
2.3 Visualização e relatórios	11
2.4 Funcionalidades adicionais	12
3. Modelagem de Entidade Relacional Estendido (EER)	13
Principais entidades:	14
Outras Entidades e Relações:	15
Interpretação Geral:	15
4. Atividades Realizadas	15
5. Conclusão e aprendizados	16

1. Introdução

Este documento especifica os requisitos do sistema *sistema de Gestão Financeira, Meu Bolso*, fornecendo informações necessárias para o projeto e implementação, assim como para a realização dos testes e homologação do sistema.

O que é o projeto?

O projeto consiste em uma aplicação desktop voltada para o gerenciamento financeiro pessoal. Ele busca simplificar o processo de controle financeiro, permitindo que o usuário registre e acompanhe seus ganhos e gastos mensais organizados por categorias. A aplicação também oferece geração de relatórios financeiros e dashboards interativos, que facilitam a visualização e compreensão das finanças pessoais. Além disso, funcionalidades como controle de investimentos pessoais e registro de metas financeiras – como economias para viagens e outros objetivos – são incluídas para auxiliar o usuário no planejamento e no alcance de metas financeiras.

Escopo do projeto:

- **Para onde vamos?**
 - Registro de ganhos mensais por categoria, como salário, freelancer, e outras fontes de renda.
 - Registro de gastos mensais por categoria, como alimentação, transporte, moradia, entre outros.
 - Geração de relatórios financeiros para análise mensal ou personalizada, facilitando a identificação de padrões de gastos.
 - Visualização de dashboards que resumem os gastos e rendimentos do mês, proporcionando uma visão clara das finanças.
 - Registro de valores destinados a investimentos pessoais (como forma de controle, sem funcionalidades de transação).
 - Registro de metas financeiras, como economizar para viagens ou aquisição de itens específicos, com acompanhamento de progresso.
- **Para onde não vamos?**
 - Não registra automaticamente os gastos (apenas entrada manual pelo usuário).
 - Não é uma plataforma de investimentos: o sistema não realiza transações, investimentos ou depósitos de dinheiro, servindo apenas para controle pessoal.

2. Requisitos funcionais

2.1 Gestão financeira

[RF001] CRUD receita

Descrição: Permite gerenciar receitas, como salários, bônus e outras fontes de renda. O usuário pode criar, visualizar, atualizar e excluir receitas. Dados envolvidos:

- **ID da transação:** identificador único da receita, tipo de dado INTEGER.
- **Descrição:** para identificar a receita registrada, deve ser do tipo VARCHAR(50) e não nula.
- **Valor:** o valor da receita a ser atribuída na transação, sendo do tipo DECIMAL.
- **Conta associada:** definir a conta de destino, é uma chave estrangeira do tipo INTEGER que referencia uma conta bancária do usuário.
- **Data:** data da transação, permitindo cadastro com até um mês anterior ou posterior à data atual, sendo do tipo DATE.
- **Categoria:** associar a uma categoria existente, é uma chave estrangeira do tipo INTEGER que referencia uma categoria de receita do usuário.
- **Comentário:** campo para informações adicionais descrevendo mais sobre a receita, é do tipo VARCHAR(200).
- **Repetição/Parcelamento:** configura que a receita é recorrente e deve ser repetida para os demais meses, podendo ainda ser parcelada ou não, sendo do tipo BOOLEAN.
- **Transação repetida:** referência a entidade com dados sobre a repetição da transação caso o campo de repetição seja verdadeira.

Restrições:

- O ID da tabela deve ser preenchido automaticamente pelo sistema.
- O ID da transação sendo a chave primária, não poderá ser de valor nulo.
- A descrição não pode ser nula, pois deve ser preenchida para identificação facilitada pelo usuário a sua receita.
- O valor da receita não poderá ser valor negativo, e a moeda será referente ao real brasileiro.
- A data deve obrigatoriamente ser fornecida pelo usuário, sendo por default a data atual.
- A conta bancária para ser referenciada deve ter sido cadastrada no sistema e não poderá ser nula.
- A categoria para ser referenciada deve ter sido cadastrada no sistema ou utilizar as pré-cadastradas, e também não poderá ser nula.
- O campo de comentário é um campo opcional para o usuário.
- O campo repetição/parcelado marca a receita para repetição no banco de dados, e deve coletar dados adicionais para coletar mais informações sobre.
- Os dados de transação repetida não ficarão visíveis direto para o usuário, podendo ser preenchida após configurar o campo de repetição/parcelamento, ou seja, é um campo que pode ser vazio.

[RF002] CRUD despesa

Descrição: Permite gerenciar despesas, permitindo criar, visualizar, atualizar e excluir saídas de gastos. Dados envolvidos:

- **ID da transação:** identificador único da receita, tipo de dado INTEGER.
- **Descrição:** para identificar a receita registrada, deve ser do tipo VARCHAR(50) e não nula.
- **Valor:** o valor da receita a ser atribuída na transação, sendo do tipo DECIMAL.
- **Conta associada:** definir a conta de destino, é uma chave estrangeira do tipo INTEGER que referencia uma conta bancária do usuário.
- **Data:** data da transação, permitindo cadastro com até um mês anterior ou posterior à data atual, sendo do tipo DATE.
- **Categoria:** associar a uma categoria existente, é uma chave estrangeira do tipo INTEGER que referencia uma categoria de despesa do usuário.
- **Comentário:** campo para informações adicionais descrevendo mais sobre a despesa, é do tipo VARCHAR(200).
- **Repetição/Parcelamento:** configura que a despesa é recorrente e deve ser repetida para os demais meses, podendo ainda ser parcelada ou não, sendo do tipo BOOLEAN.
- **Transação repetida:** referência a entidade com dados sobre a repetição da transação caso o campo de repetição seja verdadeira.

Restrições:

- O ID da tabela deve ser preenchido automaticamente pelo sistema.
- O ID da transação sendo a chave primária, não poderá ser de valor nulo.
- A descrição não pode ser nula, pois deve ser preenchida para identificação facilitada pelo usuário a sua receita.
- O valor da despesa não poderá ser valor negativo, e a moeda será referente ao real brasileiro.
- A data deve obrigatoriamente ser fornecida pelo usuário, sendo por default a data atual.
- A conta bancária para ser referenciada deve ter sido cadastrada no sistema e não poderá ser nula.
- A categoria para ser referenciada deve ter sido cadastrada no sistema ou utilizar as pré-cadastradas, e também não poderá ser nula.
- O campo de comentário é um campo opcional para o usuário.
- O campo repetição/parcelado marca a despesa para repetição no banco de dados, e deve coletar dados adicionais para coletar mais informações sobre.
- Os dados de transação repetida não ficarão visíveis direto para o usuário, podendo ser preenchida após configurar o campo de repetição/parcelamento, ou seja, é um campo que pode ser vazio.

[RF003] CRUD transferência

Descrição: Gerencia transferências entre contas bancárias do usuário, possibilitando sua criação, visualização, atualização e exclusão. Dados envolvidos:

- **ID da transferência:** identificador único da transferência, sendo do tipo INTEGER.
- **Valor:** valor que deverá ser transferido de conta bancária para outra, sendo do tipo DECIMAL.
- **Descrição:** para identificar a transferência registrada, deve ser do tipo VARCHAR(50).
- **Conta origem:** conta bancária do usuário na qual deverá ser descontado o valor da transferência, sendo uma chave estrangeira de uma conta bancária.
- **Conta destino:** conta bancária do usuário na qual será debitado o valor da transferência, sendo uma chave estrangeira de uma conta bancária.

Restrições:

- O campo descrição é opcional, sendo por default “Transferência”.
- Para realizar a transferência o usuário deverá possuir no mínimo duas contas bancárias cadastradas no sistema.
- A conta de Origem e Conta de Destino deverão ser contas diferentes e não poderão ser nulas.
- A Conta de Origem deverá possuir o valor a ser transferido para a Conta de Destino.

[RF004] CRUD categorias

Descrição: Gerencia categorias de receitas e despesas, possibilitando criação, visualização, atualização e exclusão. Dados envolvidos:

- **ID da categoria:** identificador único da tabela, sendo do tipo INTEGER
- **Tipo:** especificar se a categoria é para receita ou despesa, sendo do tipo ENUM.
- **Nome:** definir um nome representativo, do tipo VARCHAR(50).
- **Cor:** Selecionar uma cor para identificação visual, sendo do tipo VARCHAR(7).

Restrições:

- O ID deve ser preenchido automaticamente pelo sistema no banco de dados.
- O tipo da categoria só poderá ser do tipo RECEITA ou DESPESA, sendo definido pelo sistema, e não pode ser nulo.
- O nome identificador da categoria não poderá ser nulo.
- A cor deverá ser no padrão hexadecimal, guardando apenas o código da cor.
- Por questão de integridade de dados, para excluir uma categoria não deve haver receitas ou despesas que faça referência a categoria, sendo necessário o usuário excluir ou modificar a receita ou despesa com a categoria antes de excluir a categoria.

[RF005] Selecionar categoria de despesas

Descrição: Permite ao usuário escolher uma categoria ao registrar uma nova despesa, organizando os dados.

Dados envolvidos: Dados da categoria a ser selecionada e dados da transação.

Restrição: Somente categorias cadastradas previamente podem ser selecionadas, não sendo permitido manter o campo com valor nulo.

[RF006] Selecionar categoria de receitas

Descrição: Permite ao usuário escolher uma categoria ao registrar uma nova receita, organizando os dados.

Dados envolvidos: Dados da categoria a ser selecionada e dados da transação.

Restrição: Somente categorias cadastradas previamente podem ser selecionadas, não sendo permitido manter o campo com valor nulo.

[RF007] CRUD contas bancárias

Descrição: Permite gerenciar contas bancárias, com operações de criação, atualização, visualização e exclusão.

- **Saldo:** Saldo da conta do cliente, sendo do tipo DECIMAL.
- **Nome do banco:** Preferencialmente o nome do banco do usuário, sendo do tipo VARCHAR(20).
- **Categoria:** Define o tipo de conta do usuário como corrente, poupança, dinheiro ou investimento, é do tipo ENUM.

Restrições:

- O valor do saldo é calculado a partir de um valor inicial informado pelo usuário, com o valor das despesas e receitas destinadas à conta bancária.
- O nome do banco não poderá ser nulo.
- O valor do campo categoria será por default preenchido na categoria de conta corrente.

[RF008] CRUD cartão de crédito

Descrição: Permite gerenciar informações de cartões de crédito, incluindo limite e prazos.

- **Limite:** Valor do limite de crédito do cartão, sendo do tipo DECIMAL.
- **Nome:** Nome de identificação do cartão, do tipo VARCHAR(20).
- **Cor:** Seleção de uma cor para identificação visual, sendo do tipo VARCHAR(7).
- **Datas de fechamento:** Data configurada para o fechamento da fatura no mês, do tipo DATE.
- **Data de vencimento:** Data configurada para o vencimento da fatura no mês, do tipo DATE.

Restrições:

- Os campos de Limite, Nome, Data de Fechamento, Data de Vencimento e Cor não poderão ser vazios, sendo todos obrigatoriamente preenchidos pelo usuário no cadastro do cartão.
- O valor do limite do cartão não poderá ser negativo.
- A data de vencimento deve ser maior que a data de fechamento.

[RF009] CRUD despesas fixas

Descrição: Possibilita ao usuário facilitar o gerenciamento de despesas recorrentes, como aluguel ou contas mensais. Envolve os seguintes dados:

- **Identificador de transação:** ID da despesa referente que irá se repetir, sendo uma chave estrangeira, ou seja, o tipo do ID que é INTEGER.
- **Data início:** Data que define o início da repetição, sendo do tipo DATE.
- **Conta:** A conta referente a transação e na qual deve ser alterado o saldo em razão do valor da receita.

Restrições:

- Todos os campos não poderão ter seu valor como nulo.

[RF0010] CRUD receitas fixas

Descrição: Possibilita ao usuário facilitar o gerenciamento de receitas recorrentes, como salários mensais. Envolve os seguintes dados:

- **Identificador de transação:** ID da receita referente que irá se repetir, sendo uma chave estrangeira, ou seja, o tipo do ID que é INTEGER.
- **Data início:** Data que define o início da repetição, sendo do tipo DATE.
- **Conta:** A conta referente a transação e na qual deve ser alterado o saldo em razão do valor da receita.

Restrições:

- Todos os campos não poderão ter seu valor como nulo.

[RF0011] CRUD despesas parceladas

Descrição: Possibilita ao usuário dividir despesas em parcelas, definindo a quantidade de parcelas que o valor vai ser dividido. Envolve os seguintes dados:

- **Identificador de transação:** ID da despesa referente que irá se repetir, sendo uma chave estrangeira, ou seja, o tipo do ID que é INTEGER.

- **Quantidade de repetição:** Define a quantidade de repetições da despesa deve ser acrescentada no histórico de transações futuras, sendo do tipo INT.
- **Data início:** Data que define o início da repetição, sendo do tipo DATE.
- **Data final:** Data da última parcela a ser cadastrada no banco de dados, sendo calculada em razão da Data Inicial com a Quantidade de Repetição informada pelo usuário.
- **Valor da parcela:** O valor da parcela, será o valor total da transação, pegue da transação pai, e calculando a partir da Quantidade de Parcela informada pelo usuário.

Restrições:

- Todos os campos não poderão ter seu valor como nulo.

[RF0012] CRUD receitas parceladas

Descrição: Possibilita ao usuário dividir receitas em parcelas, definindo a quantidade de parcelas que o valor vai ser dividido. Envolve os seguintes dados:

- **Identificador de transação:** ID da receita referente que irá se repetir, sendo uma chave estrangeira, ou seja, o tipo do ID que é INTEGER.
- **Quantidade de repetição:** Define a quantidade de repetições da receita deve ser acrescentada no histórico de transações futuras, sendo do tipo INT.
- **Data início:** Data que define o início da repetição, sendo do tipo DATE.
- **Data final:** Data da última parcela a ser cadastrada no banco de dados, sendo calculada em razão da Data Inicial com a Quantidade de Repetição informada pelo usuário.
- **Valor da parcela:** O valor da parcela, será o valor total da transação, pegue da transação pai, e calculando a partir da Quantidade de Parcela informada pelo usuário.

Restrições:

- Todos os campos não poderão ter seu valor como nulo.

2.2 Controle e organização de dados

[RF0012] CRUD orçamento

Descrição: Permite definir e gerenciar orçamentos mensais para transações do tipo de despesas de uma determinada categoria definida pelo usuário, alertando quando o limite for alcançado ou estiver próximo de ser atingido. Os dados envolvidos são:

- **Identificador de orçamento:** ID do orçamento, o tipo do ID é INTEGER.
- **Valor Total:** Quanto o usuário planeja gastar, sendo do tipo DECIMAL.

- **Valor Gasto:** Total já consumido no mês pelo usuário na categoria definida, sendo do tipo DECIMAL.
- **Valor Restante:** Saldo disponível do total que o usuário ainda pode utilizar, sendo o tipo DECIMAL.
- **Categoria:** Categoria que será estimado o orçamento, sendo uma chave estrangeira referenciando a categoria, ou seja, do tipo LONG.

Restrições:

- A quantidade de orçamentos que podem ser criados é a mesma de categorias cadastradas no sistema.
- Cada orçamento só poderá possuir uma categoria.
- O campo de Valor Total deve obrigatoriamente ser preenchido pelo usuário, não podendo ser nulo.
- Os campos de Valor Gasto e Valor Restante devem ser alterados pelo sistema automaticamente ao ser registrado uma despesa com a mesma categoria do orçamento.
- Categoria deve ser um campo obrigatório, não podendo ser nulo.

[RF0013] CRUD metas

Descrição: Permite definir e gerenciar metas financeiras, associando transações ou categorias específicas para acompanhar o progresso em direção ao valor definido.

Dados envolvidos:

- **Valor Total:** O valor que o usuário deseja atingir na meta, do tipo DECIMAL.
- **Valor Atual:** O progresso atual acumulado em direção à meta, do tipo DECIMAL.
- **Descrição:** Identificação para o objetivo na meta definida pelo usuário, do tipo VARCHAR(50).
- **Imagem de capa:** Imagem da capa da meta definida e editada pelo usuário, sendo do tipo BYTEA (armazenamento de imagens no postgresql).

Restrições:

- Cada meta deve possuir um valor total definido pelo usuário, não podendo ser nulo.
- O campo Valor Atual será atualizado automaticamente conforme as transações associadas sejam registradas.
- A descrição deverá ser obrigatoriamente preenchida para identificação da meta pelo usuário.
- A foto pode ser nula, na qual o usuário não é obrigado a fornecer uma imagem de capa.

2.3 Visualização e relatórios

[RF0014] Visualizar dashboards

Descrição: Permite exibir um painel com informações consolidadas das transações de receitas e despesas, separado por tipo de categoria e conta bancária.

Dados envolvidos:

- **Transações:** Informações consolidadas das transações, gerando um valor total e mostrando dados conforme os diferentes tipos de dashboards.
- **Mês de visualização:** informa o mês que o usuário quer ver os dados em dashboards, sendo apenas do tipo INT com o número do mês.

Restrições:

- O dashboard deve apresentar gráficos e tabelas que sejam gerados com dados divididos por mês.
- O mês pode ser alterado pelo usuário, mas os dados serão sempre em relação ao selecionado pelo mesmo.

[RF0015] Visualizar histórico de transações

Descrição: Permite consultar o histórico de transações registradas no sistema, filtrando por período, categoria e tipo de transação. Dados envolvidos:

- **Data:** Data em que a transação foi registrada, do tipo DATE.
- **Categoria:** Categoria associada à transação, do tipo LONG.
- **Valor:** Valor da transação, do tipo DECIMAL.
- **Tipo:** Tipo da transação (receita ou despesa), do tipo ENUM.

Restrições:

- As transações devem ser exibidas em ordem cronológica, da mais recente para a mais antiga.
- Os filtros de pesquisa por período, categoria e tipo de transação são obrigatórios para consultas específicas.

[RF0016] Filtrar dashboards

Descrição: Permite ao usuário aplicar filtros no dashboard para personalizar a exibição das informações, filtrando por categorias, datas, e conta. Dados envolvidos:

- **Período:** Intervalo de tempo (diário, semanal, mensal).

- **Categoria:** Seleção de categorias específicas para análise.
- **Tipo de transação:** Receita, despesa ou ambos.

Restrições:

- Pelo menos um filtro deve ser aplicado para exibir dados no dashboard.

2.4 Funcionalidades adicionais

[RF0017] Notificação de status de orçamento

Descrição: Notifica o usuário quando o status de um orçamento muda sendo atingido o limite, como ao atingir um limite definido.

Dados envolvidos:

- **Orçamento:** Orçamento relacionado à notificação, do tipo LONG.
- **Mensagem:** Conteúdo da notificação, do tipo TEXT.
- **Data de Geração:** Data e hora da notificação, do tipo TIMESTAMP.

Restrições:

- Notificações só serão exibidas na tela inicial e não serão armazenadas permanentemente.
- A notificação será removida automaticamente após ser visualizada pelo usuário.
- A notificação só deverá ser gerada quando o usuário atingir o limite ou faltar 20% para atingir o total.

[RF0018] Notificação de progresso na meta

Descrição: Notifica o usuário quando o progresso de uma meta alcança um marco importante ou é concluído.

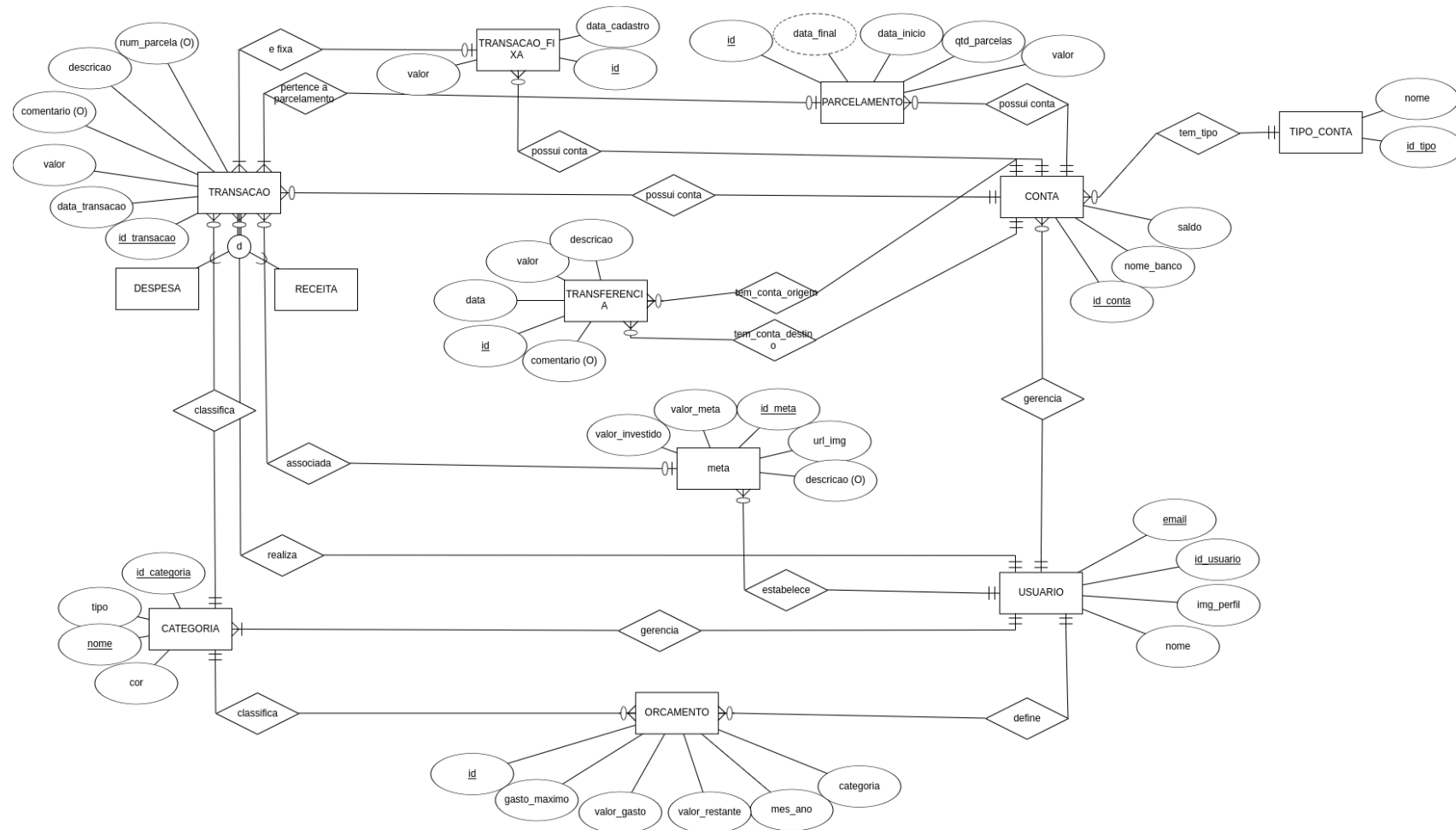
Dados envolvidos:

- **Meta:** Meta relacionada à notificação, do tipo LONG.
- **Mensagem:** Descrição do progresso alcançado, do tipo TEXT.
- **Data de Geração:** Data e hora da notificação, do tipo TIMESTAMP.

Restrições:

- Apenas metas com Data Limite definida podem gerar notificações de progresso.
- As notificações não serão armazenadas após sua exibição.
- A notificação só deverá ser gerada quando o usuário completar ou faltar 20% para atingir o total da meta definida

3. Modelagem de Entidade Relacional Estendido (EER)



Principais entidades:

- **TRANSACAO:** Representa uma operação financeira, que pode ser classificada como **RECEITA** (entrada) ou **DESPESA** (saída).
 - Relacionada com **CONTA:** Indica a conta envolvida na transação.
 - Relacionada com **CATEGORIA:** Define a categoria da receita ou despesa. Exemplos de categorias de despesa: alimentação, salário, conta de luz e etc.
 - Relacionado com **TRANSACAO_FIXA:** define se a transação é fixa.
 - Relacionado com **PARCELAMENTO:** define se a transação faz parte de um parcelamento.
- **TRANSACAO_FIXA:** Um registro que indica o valor de uma transação que deverá se repetir mensalmente na mesma data.
 - Relacionado a **TRANSACAO:** uma transação que for fixa deve possuir uma chave primária para o registro da transação fixa.
 - Relacionado a **CONTA:** a transação fixa deve pertencer a associada a uma conta (de onde entrará uma receita ou sairá uma despesa mensalmente).
- **PARCELAMENTO:** Um registro que indica um parcelamento e o valor total das parcelas.
 - Relacionado a **TRANSACAO:** Uma transação de despesa poderá ser uma parcela associada a um parcelamento, nesse caso a transação deverá conter também a informação do número da parcela que a transação representa.
 - Relacionado a **CONTA:** O parcelamento deve pertencer a uma conta (de onde sairá a despesa mensal).
- **TRANSFERÊNCIA:** é um registro que representa uma troca de dinheiro entre as próprias contas do usuário, logo não se caracteriza como receita ou despesa, já que não afetará o saldo final do usuário.
 - Relacionado com **CONTA:** Está relacionado com duas contas, uma de saída do dinheiro e a outra de entrada.
- **CONTA:** Representa uma conta bancária.
 - Relacionada com **TRANSACAO:** Indica as transações realizadas na conta.
 - Relacionada com **USUARIO:** Indica o usuário associado à conta.
 - Relacionado com **TIPO_CONTA:** Indica o tipo de conta (investimentos, corrente ou etc.).
- **CATEGORIA:** Define diferentes categorias de despesas ou receitas.
 - Relacionada com **TRANSACAO:** Classifica as transações por categoria.
 - Relacionado com **ORÇAMENTO:** um orçamento é atrelado a uma categoria, pois define um valor máximo estipulado que poderá ser gasto naquela categoria. O valor máximo é definido pelo usuário e cabe a ele se não irá exceder o valor por ele estipulado.
- **ORCAMENTO:** Representa um plano financeiro mensal para uma determinada categoria, definindo o valor máximo estipulado de gasto para aquela categoria.
 - Relacionado com **CATEGORIA:** pois ele estipula um gasto máximo para uma categoria.
 - Relacionado com **USUARIO:** Indica o usuário responsável pelo orçamento.
- **META:** Representa uma meta de poupança financeira que o usuário deseja guardar. Uma meta possui uma descrição dada pelo usuário de acordo com seu objetivo, por exemplo: 'Comprar a moto', 'viagem para a Suíça' ou etc. Também possui um valor,

que é o saldo separado para aquela meta que o usuário deseja reservar. O dinheiro reservado para uma meta é descontado do saldo de uma conta.

- Relacionado com **USUARIO**: uma meta pertence a um usuário.
- Relacionado a **TRANSACAO**: uma meta possui um histórico de transações feitas para aquela meta.
- **USUARIO**: Representa um usuário do sistema.
 - Relacionado com **CONTA**: Indica as contas do usuário.
 - Relacionado com **ORCAMENTO**: Indica os orçamentos do usuário.
 - Relacionado com **META**: Indica as metas do usuário

Outras Entidades e Relações:

Além das entidades principais, a modelagem inclui outras entidades que representam:

- **REGISTRO**: Pode ser um registro histórico de uma transação ou outra informação relevante.
- **ARQUIVO**: Pode armazenar documentos relacionados às transações.
- **AGENDA**: Pode ser utilizada para agendar pagamentos ou outras tarefas financeiras.
- **ALERTAS**: Podem ser configurados para notificar o usuário sobre eventos financeiros importantes.

Interpretação Geral:

A modelagem EER sugere um sistema de gestão financeira completo, capaz de:

- **Registrar todas as transações financeiras**: Pagamentos, recebimentos, transferências, etc.
- **Classificar as transações por categorias**: Despesas, receitas, investimentos, etc.
- **Gerenciar orçamentos**: Criar, acompanhar e analisar orçamentos.
- **Associar transações a contas e usuários**: Permitir um controle individualizado das finanças.
- **Gerar relatórios**: Fornecer informações detalhadas sobre as finanças do usuário.

4. Atividades Realizadas

Atividade	Descrição	Responsável
Levantamento de requisitos com stakeholders	Levantamento de requisitos com stakeholders.	Antônio Rewelli, Lucas Ferreira,
Levantamento de requisitos com benchmarking	Levantamento de requisitos por meio de aplicações já utilizadas	Miqueias Bento, Antônio Rewelli, Giliardy Alves, Lucas Ferreira.

Modelagem de Dados, versão inicial	Modelagem de dados feita inicialmente, servindo como um esboço para a versão final.	Lucas Ferreira, Antônio Rewelli, Miqueias Bento.
Modelagem de Dados, versão final	Modelagem final do projeto, com todas as entidades, atributos e relacionamentos adicionados.	Lucas Ferreira, Miqueias Bento
Descrição dos requisitos	Descrição detalhada dos requisitos funcionais do sistema, com informações sobre a descrição, dados e restrições.	Miqueias Bento, Giliardy Alves.

5. Conclusão e aprendizados

A atividade dessa entrega teve forte ligação com as atividades realizadas para a disciplina de Projeto Integrado em Engenharia de Software I, o que nos permitiu aprofundarmos nossos conhecimentos sobre coleta de requisitos através das práticas de entrevistas semi-estruturadas, formulários e benchmarking. Além disso, praticamos organização e gerenciamento de processos, pois usamos o scrum para gerenciar nossas atividades.

Em banco de dados aprendemos a projetar um banco de dados através do diagrama ER, utilizando os seus conceitos de alto nível, como: herança, atributos derivados e etc. Também tomamos conhecimento da importância de detalhar as entidades e alinhar o entendimento sobre os campos das entidades e relacionamentos com os membros da equipe, a fim de manter uma comunicação clara e objetiva e, conseqüentemente, evitar que cada um dos membros tenha um entendimento diferente sobre os requisitos ou modelagem do banco de dados. A respeito dos relacionamentos, aprendemos que é de suma importância explicitar a cardinalidade, já que a cardinalidade nos guiará na produção futura do diagrama relacional, e também ter nomes que descrevam claramente o sentido do relacionamento, a fim de evitar ambiguidades.

Ademais, aprimoramos nossa capacidade de trabalho em equipe, gerenciamento de time, divisão de tarefas complexas em tarefas menores e distribuição de responsabilidades entre os membros da equipe.

