



ATIVIDADE PRÁTICA DA DISCIPLINA GERÊNCIA DE CONFIGURAÇÃO E EVOLUÇÃO

ENGENHARIA DE SOFTWARE

LEONAM UIRLEY DO NASCIMENTO CASSEMIRO – 4672144
PROF^a. MARIANE G B FERNANDES

História de Usuário:

O empresário Felipe Fernandes precisa realizar a automatização do sistema de sua startup AGROPETS. Atualmente, o empresário disponibiliza em sua startup produtos para agricultura, pecuária e pets no geral. Porém não tem nenhum software para realizar as seguintes funções: pedidos realizados por clientes; tipos de produtos por sua categoria; quantidade de produtos em estoque; valor unitário de produto; dados do cliente (nome, CPF, e-mail, telefone de contato, endereço, histórico de compras efetuadas e devoluções/trocas); impressão de notas fiscais das compras realizadas por cliente; Gastos mensais com funcionários; Gastos mensais básicos (energia e água); e os lucros da empresa (mensal e anual). Além disso, Felipe precisará ter neste software dois tipos de login, um administrativo (terá acesso a todos os dados de sua startup e dos clientes) e outro login para seus funcionários (sem o demonstrativo de rendimentos que a startup ganha por dia/mês/ano e gastos gerais da empresa). Seu desafio é pensar como irá desenvolver **ficticiamente** o software que atenda a demanda do empresário Felipe para automatizar a startup AGROPETS.

A partir da **HISTÓRIA DE USUÁRIO** responda as seguintes perguntas:

1. “A identificação dos riscos pode ser um processo do time, em que ele se reúne para fazer um brainstorm dos possíveis riscos. Alternativamente, os gerentes de projetos podem identificar os riscos com base em sua experiência do que deu errado em projetos anteriores” (SOMMERVILLE, 2019). Com base nisto, faça uma lista fictícia **com 5 erros do projeto de software AGROPETS** pode apresentar. (25 pontos)

ERROS DO PROJETO	DESCRIÇÃO
Problemas de desempenho	O software pode apresentar lentidão e travamentos frequentes
Falhas na comunicação com dispositivos externos	Se precisar se comunicar com impressoras, leitores de código ou outros dispositivos, problemas de compatibilidade podem surgir
Falta de escalabilidade	O software pode não ser capaz de aguentar o volume de dados e transações
Problemas de segurança	Falhas na implementação de medidas de segurança podem comprometer os dados dos clientes e da empresa.
Deficiência na geração de relatórios	O sistema pode não fornecer relatórios suficientes sobre histórico de vendas, compras, produtos etc.

2. Além dos riscos exibidos na Tabela 1, identifique pelo menos seis outros possíveis **riscos de gerência da equipe do projeto** que poderiam surgir no projeto de software **AGROPETS**. Fazer uma tabela conforme a Tabela 1. (25 pontos)

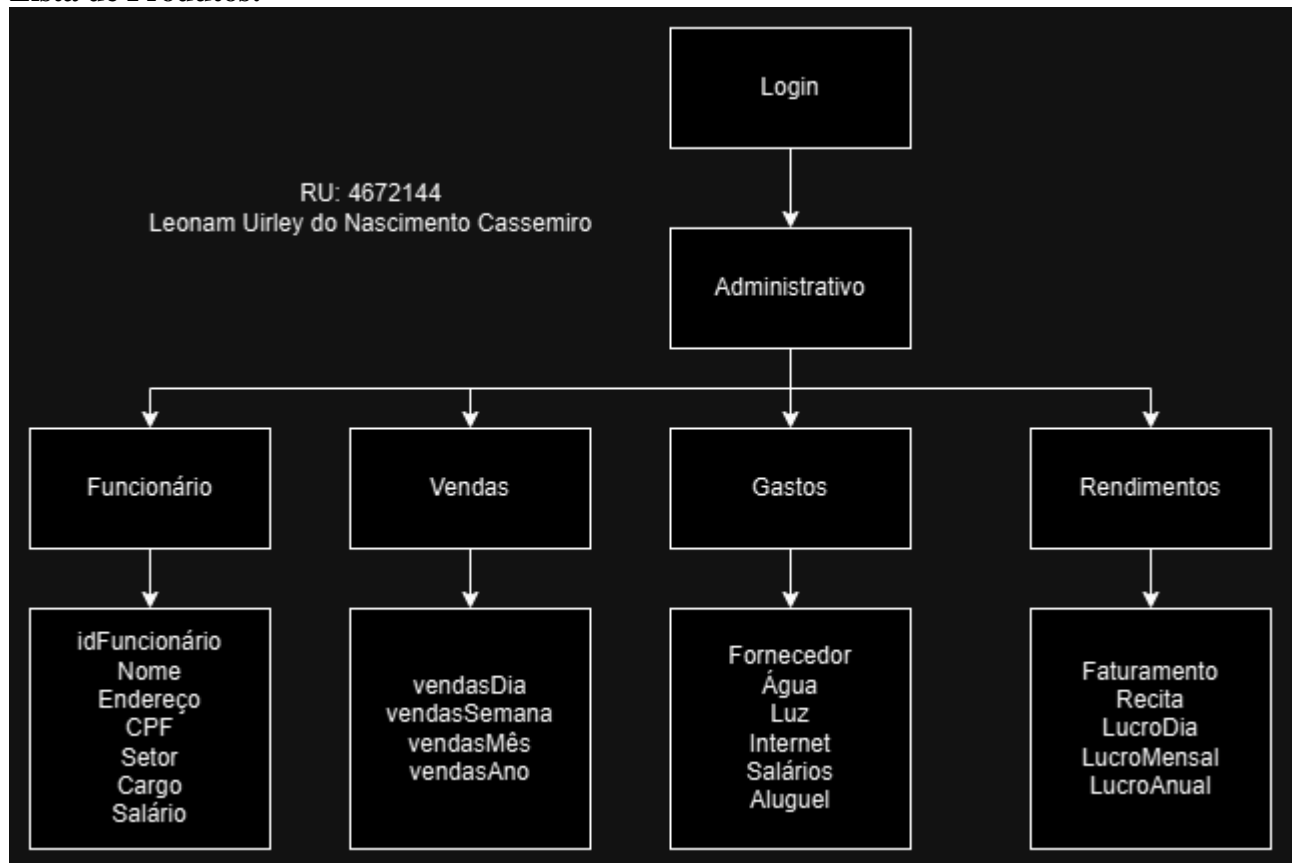
RISCO DE EQUIPE	DESCRIÇÃO
Falta de experiência da equipe	Equipe com falta de experiência no setor de atuação.
Comunicação ineficaz	Falta de clareza na comunicação, ausência de uma falta de feedback e colaboração.
Dificuldade de encontrar e reter talentos	Escassez de profissionais qualificados na área de desenvolvimento de software.

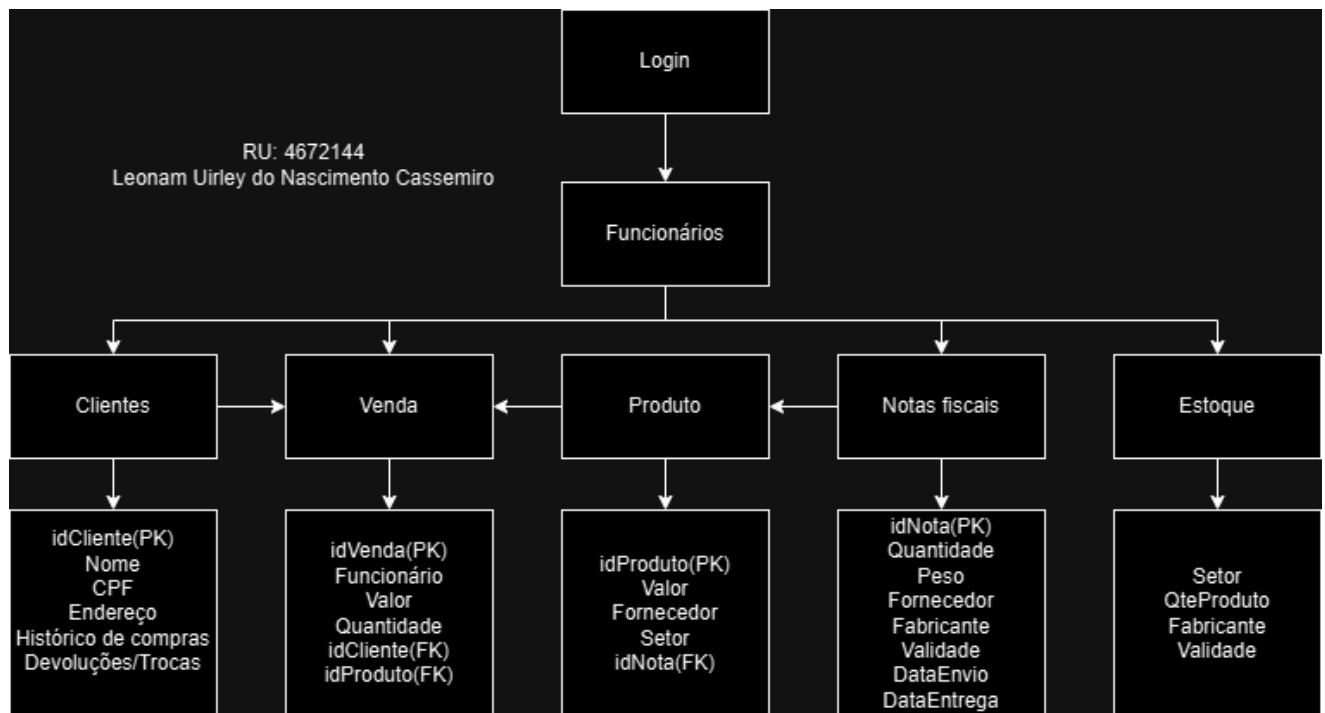
Falta de comprometimento	Falta e engajamento e motivação da equipe.
Conflitos interpessoais	Desacordos, falta de respeito e discussões entre os membros da equipe.

3. De acordo como que foi estudado na Aula prática 02, você precisará fazer **um Plano de Gerência de Configuração** para o projeto de software **AGROPETS**. (50 pontos)

Nome do Projeto: AGROPETS System

Lista de Produtos:





Itens de Configuração:

- *Código-fonte do sistema:* Este item abrange todos os arquivos de código que compõem o software AGROPETS.
- *Banco de dados:* O controle de versão do banco de dados garante a consistência dos dados, permitindo a recuperação de dados em caso de falhas e o acompanhamento das mudanças na estrutura do banco de dados.
- *Diagramas de classe:* Os diagramas de classe fornecem uma representação visual da estrutura do sistema, ilustrando as classes, seus atributos e as relações entre elas.
- *Documentos de requisitos:* Esses documentos detalham os requisitos funcionais e não funcionais do sistema, definindo o que o software deve fazer e como deve se comportar.
- *Manual do usuário:* É crucial controlar as versões do manual para garantir que as informações estejam atualizadas e consistentes com a versão do software em uso.

Baselines do Projeto:

Requisitos Funcionais	
RF01	Permitir login dos funcionários
RF02	Permitir login administrativo
RF03	Cadastrar produtos
RF04	Controlar estoque de produtos
RF05	Registrar valor unitário de cada produto
RF06	Cadastrar clientes com informações
RF07	Registrar histórico de compras, devoluções e trocas de cada item
RF08	Gerar e imprimir notas fiscais para as compras

RF09	Registrar gastos mensais com funcionários
RF10	Registrar gastos mensais com energia e água
RF11	Calcular e exibir lucros da empresa (mensal e anual)
RF12	Restringir acesso de funcionários a dados financeiros da empresa

Requisitos Não Funcionais	
RFN01	O sistema deve ser fácil de usar, com uma interface intuitiva para facilitar a interação do usuário
RFN02	As senhas de login devem ser armazenadas de forma segura, utilizando criptografia para proteger os dados dos usuários
RFN03	O sistema deve ser responsivo e ter um tempo de resposta rápido para as operações, garantindo uma boa experiência para o usuário.
RFN04	O sistema deve ser compatível com diferentes sistemas operacionais, permitindo a sua utilização em diversas plataformas.
RFN05	O sistema deve ser confiável e estar disponível para uso durante o horário comercial da empresa.
RFN06	O sistema deve ser escalável para suportar o crescimento futuro da empresa, como o aumento do número de produtos, clientes e transações.
RFN07	O sistema deve ter mecanismos de backup e recuperação de dados para garantir a segurança das informações em caso de falhas.
RFN08	O sistema deve ser desenvolvido utilizando boas práticas de programação para garantir a qualidade do código e facilitar a manutenção.

Auditoria de Configuração:

Checklist de auditoria de configuração	
Descrição	Ação
A visão do projeto está definida e documentada?	(x) sim () não
O backup do produto está criado e priorizado?	(x)sim ()não
As sprints foram planejadas com base no backlog?	(x)sim ()não
As daily meetings estão sendo realizadas diariamente?	()sim (x)não
O código-fonte está sendo versionado adequadamente?	(x)sim ()não
Os testes estão sendo realizados durante a sprint?	()sim (x)não

A documentação do projeto está sendo realizada?	<input type="radio"/> sim <input checked="" type="radio"/> não
As funcionalidades desenvolvidas na sprint foram demonstradas e revisadas?	<input type="radio"/> sim <input checked="" type="radio"/> não
A retrospectiva da sprint foi realizada para identificar pontos de melhoria?	<input checked="" type="radio"/> sim <input type="radio"/> não
Os itens de configuração estão sendo gerenciados e controlados?	<input checked="" type="radio"/> sim <input type="radio"/> não

Ferramentas Utilizadas:

- O controle de versão e melhorias no código fonte foram usadas com *GitHub*.
- A modelagem e documentação do projeto foram usadas com *Draw.io*.
- O gerenciamento do projeto foi usado o *Trello*.
- A comunicação entre os colaboradores foi usado o *Microsoft Teams*.
- Para testes do projeto foi usado o *Selenium*.