P1 - Projeto prático de Engenharia de Software

Caríssimos(as),

Vamos fazer uma aplicação prática dos conceitos, técnicas e ferramentas de Engenharia de Software. Utilizaremos como base o desenvolvimento de uma aplicação para o controle de vacinação contra a COVID-19. Esta aplicação prática pode ser realizada em equipes de até 4 pessoas.

Pede-se:

Parte A: Requisitos de Software

Identificação dos Requisitos Funcionais;

Identificação dos Requisitos Não Funcionais;

Elaboração do Diagrama de Casos de Uso;

Elaboração das Narrativas dos Casos de Uso.

Parte B: Projeto de Interface

Uma reflexão dos princípios de interface e usabilidade utilizando como base a modelagem de requisitos realizada pela equipe e as heurísticas de Nielsen;

Um protótipo inicial da aplicação (ao menos 3 telas).

Parte C: Gestão da Configuração de Software

Criação de um repositório no GitHub;

Inclusão da equipe e do professor (usuário no GitHub: AndersonLuizBarbosa) como equipe nesse repositório;

Inclusão dos documentos de requisitos e do projeto de interface no repositório criado.

Parte D: Métricas de Software

Uma reflexão sobre quais métricas e indicadores são os mais adequados para esse tipo de desenvolvimento;

Cálculo dos Pontos de Função e preenchimento da planilha disponibilizada.

Parte E: Métodos Ágeis - SCRUM

A escolha de uma ferramenta (Jira Agile, PangoSCRUM, Trello, Monday entre outras);

A geração do backlog;

A geração de ao menos um sprint backlog;

A simulação de progresso e acompanhamento via BurnDown Chart.

Parte F: Testes de Software

Uma reflexão sobre quais os tipos de teste mais adequados para a verificação e validação dessa aplicação;

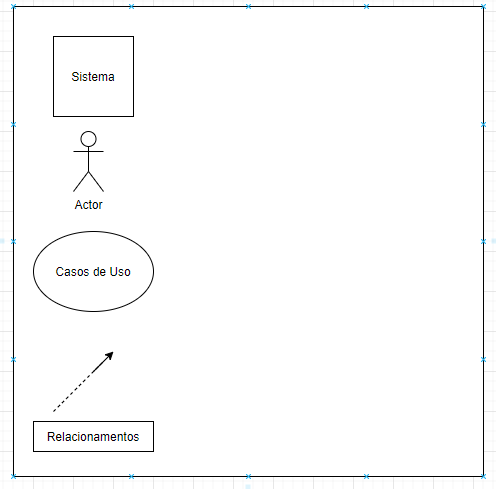
Elaboração de um plano de teste - utilizar como base a planilha disponibilizada.

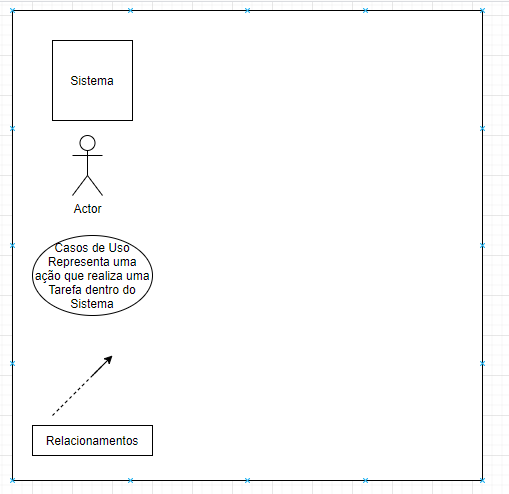
Ao final pede-se a elaboração de um vídeo (máximo de 5 minutos) para a apresentação da modelagem realizada

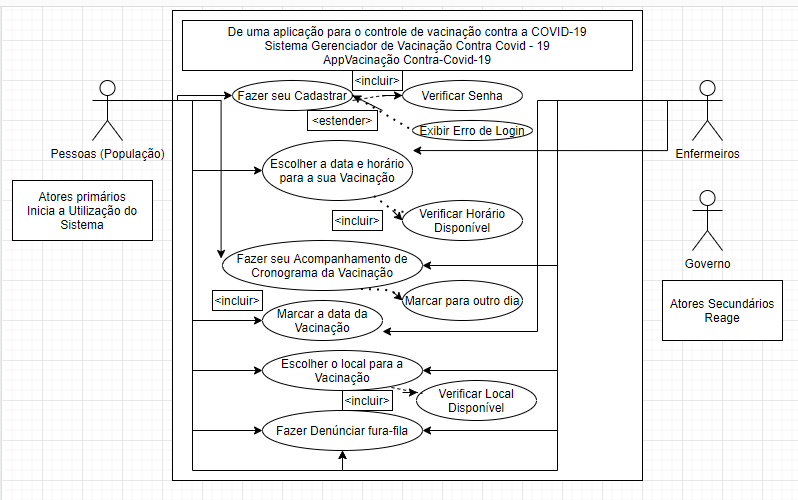
Ferramenta de Modelagem - Draw.IO

Diagrama de Caso de Uso UML

De uma aplicação para o controle de vacinação contra a COVID-19.







**Atividade 14 - Prática - Análise e Especificação de Requisitos**

Vamos fazer uma aplicação prática dos conceitos, técnicas e ferramentas para a Análise e Especificação de Requisitos. Utilizaremos como base o desenvolvimento de uma aplicação para o controle de vacinação contra a COVID-19. Pede-se:

**1 Identificação dos Requisitos Funcionais;**

Os **requisitos funcionais** descrevem o comportamento do sistema, seus **requisitos** para o funcionamento de cada item, ou seja, os **requisitos** que descrevem o trabalho que o sistema deve realizar. Eles **são** os cérebros do projeto porque descrevem as funções que o sistema deve ter.

Vacinações lançadas pelos pacientes: Os registros de vacinações lançados pelos pacientes ficarão com o estado de não verificado. Só após a validação do registro por um agente de saúde, que o registro de vacinação lançado passará a ter o estado de verificado. Isso é necessário para que não haja lançamento de vacinações não realizadas de fato no cartão de vacinação.

Vacinações lançadas pelos agentes de saúde: Os registros de vacinações lançados pelos agentes de saúde ficarão com o estado de verificado, não precisando serem validados posteriormente.

Acesso para consulta: Para o paciente acessar o Sistema de Controle de Vacinação são pedidos o CPF e a senha cadastrados por ele.

Alerta de vacinação: Ao realizar o login, os pacientes que tiverem vacinações precisando de reforço ou dose receberão na página inicial do sistema um alerta informando.

É necessário que seja possível cadastrar, editar e excluir as informações do usuário. Deverão ser cadastrados as seguintes dados: nome, município de nascimento, data de nascimento, nome da mãe e pai, endereço, ponto de referência, telefone, CEP, UF, bairro, cidade, raça/cor/etnia, UBS que frequenta, CPF, RG, e- mail e senha.

**2- Identificação dos Requisitos Não Funcionais;**

**Requisitos não funcionais são os requisitos** relacionados ao uso da aplicação em termos de desempenho, usabilidade, confiabilidade, segurança, disponibilidade, manutenção e tecnologias envolvidas. Estes **requisitos** dizem respeito a como **as** funcionalidades serão entregues ao usuário do software.

Modelo de impressão do Cartão de Vacinação: Apenas os registros de vacinações validadas aparecerão no modelo de impressão do cartão de vacinação.

Aviso para vacinação: Na consulta do calendário de vacinação, as campanhas de vacinas dos pacientes já vencidas ou que precisam da próxima dose serão destacadas.

Segurança

Somente usuários cadastrados terão acesso ao sistema

Desenvolvimento

Usabilidade

O sistema deverá ser de fácil interação para com todos os usuários.

Funcionalidades:

Cadastra, altera e consulta os dados das unidades no sistema.

Manter Pessoas: Cadastra, altera e consulta os dados dos pacientes no sistema.

Consultar Cadastro da Pessoa: Consulta os dados dos pacientes no sistema.

Consultar Cartão de Vacinação: Consulta os dados do cartão de vacinação no sistema.

Registrar Vacinação: Realiza o registro da vacina aplicada no cartão de vacinação.

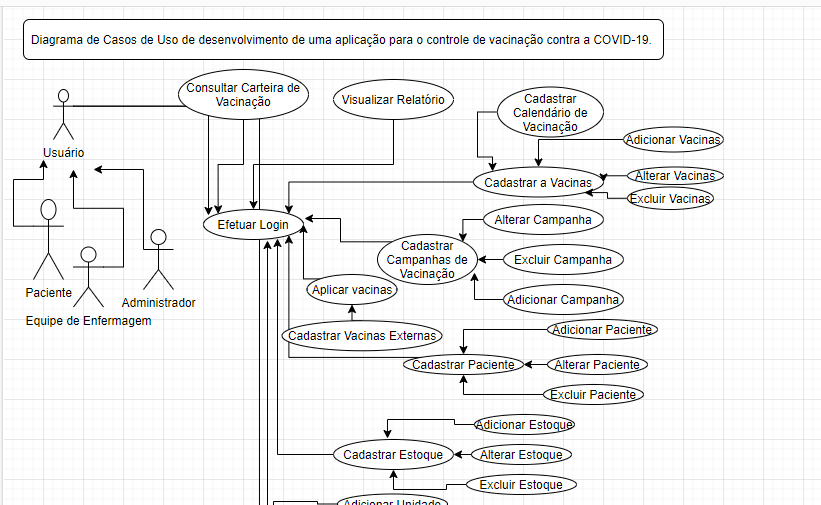
Validar Vacinação: Valida o registro da vacina lançada pelo paciente no cartão de vacinação.

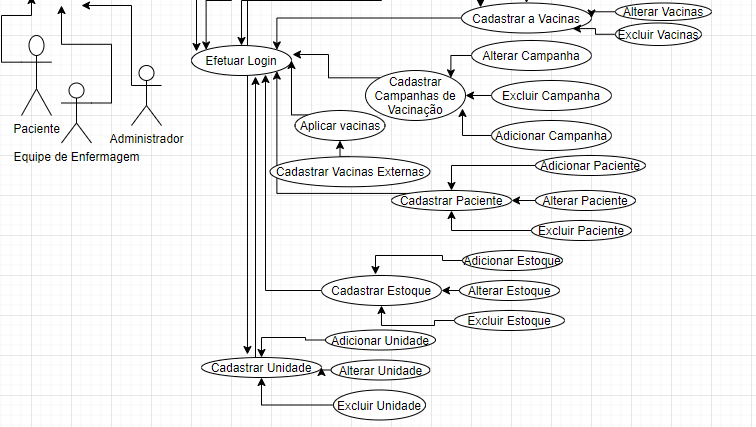
Emitir Cartão de Vacinação: Realiza a impressão das informações dos cartões de vacinação no sistema.

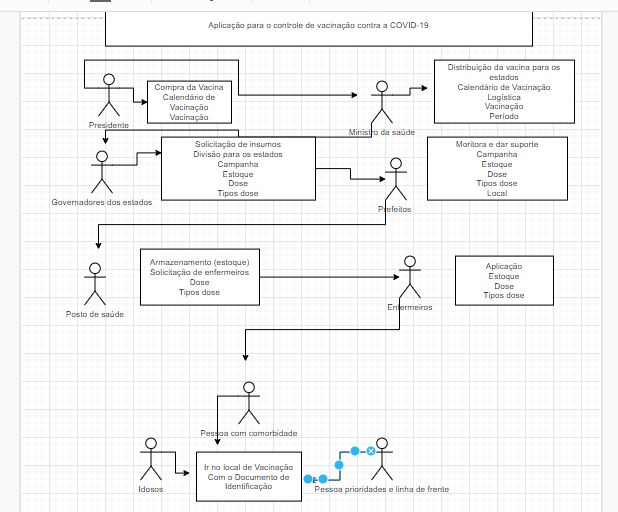
Consultar Calendário de Vacinação: Exibe o calendário de vacinação nacional do ano da consulta

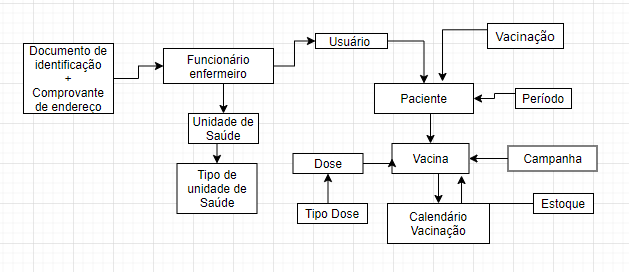
**3- Elaboração do Diagrama de Casos de Uso;**

**4 - Elaboração das Narrativas dos Casos de Uso.**









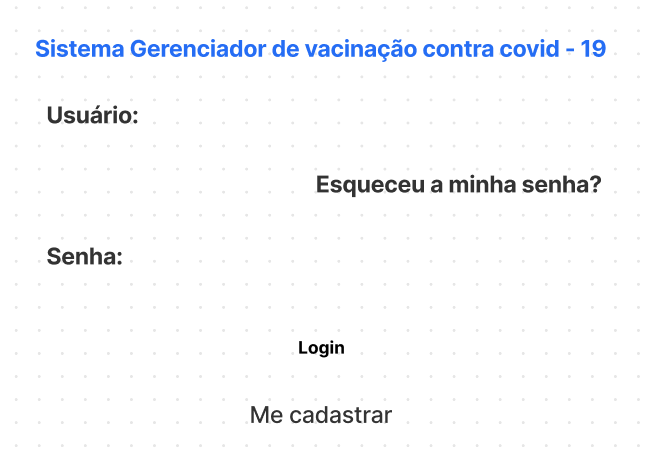
**Atividade 15 - Prática - Projeto de Interface**

1- Uma reflexão dos princípios de interface e usabilidade utilizando como base a modelagem de requisitos realizada pela equipe e as heurísticas de Nielsen;

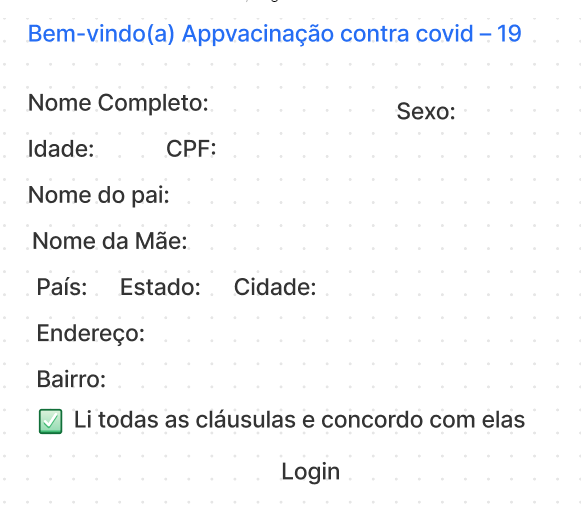
2 -Um protótipo inicial da aplicação (ao menos 3 telas).

**Sistema Gerenciador de vacinação contra covid – 19**

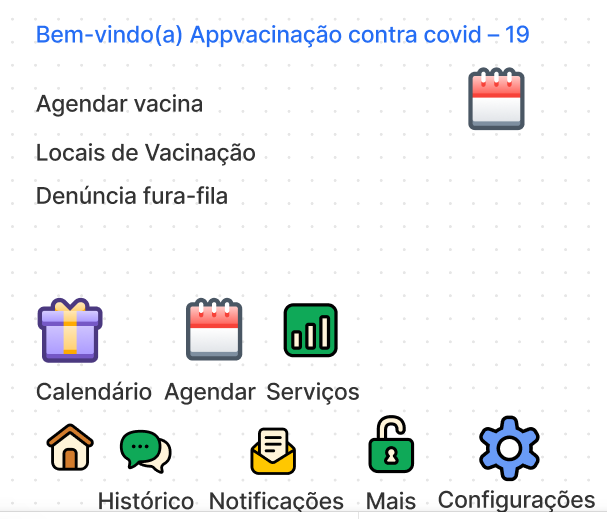
**1ª Tela inicial do App vacinação contra covid – 19**

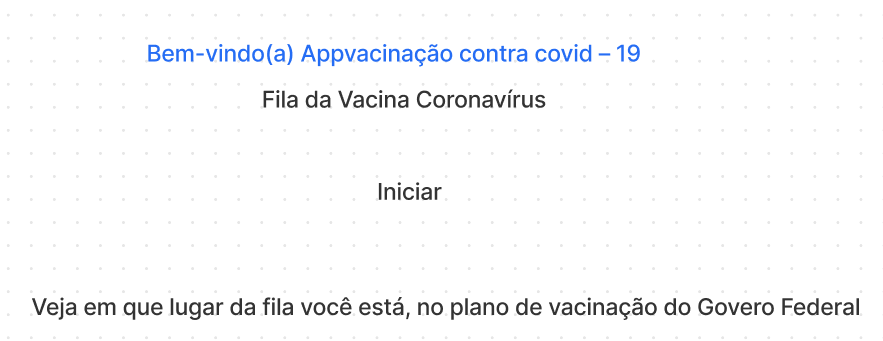


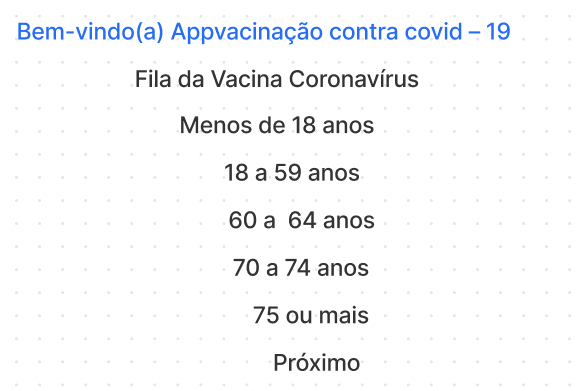
**2ª Tela de cadastramento no site**

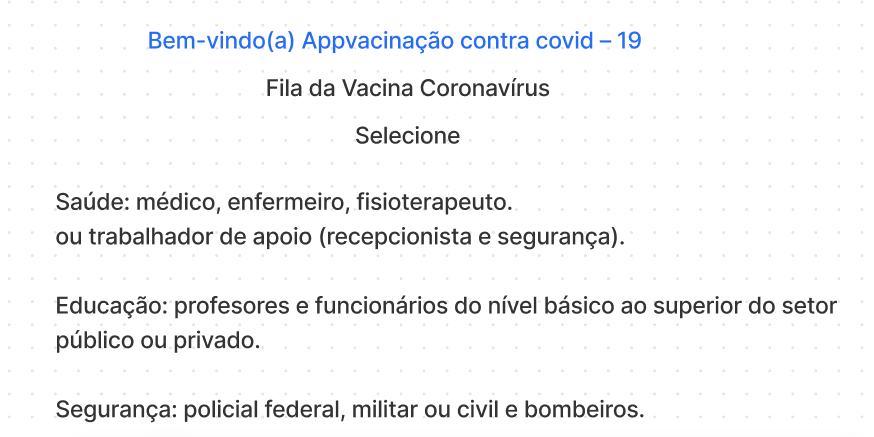


**3ª Tela já cadastrado - Tela para consultar e agendamento para a vacinação.**







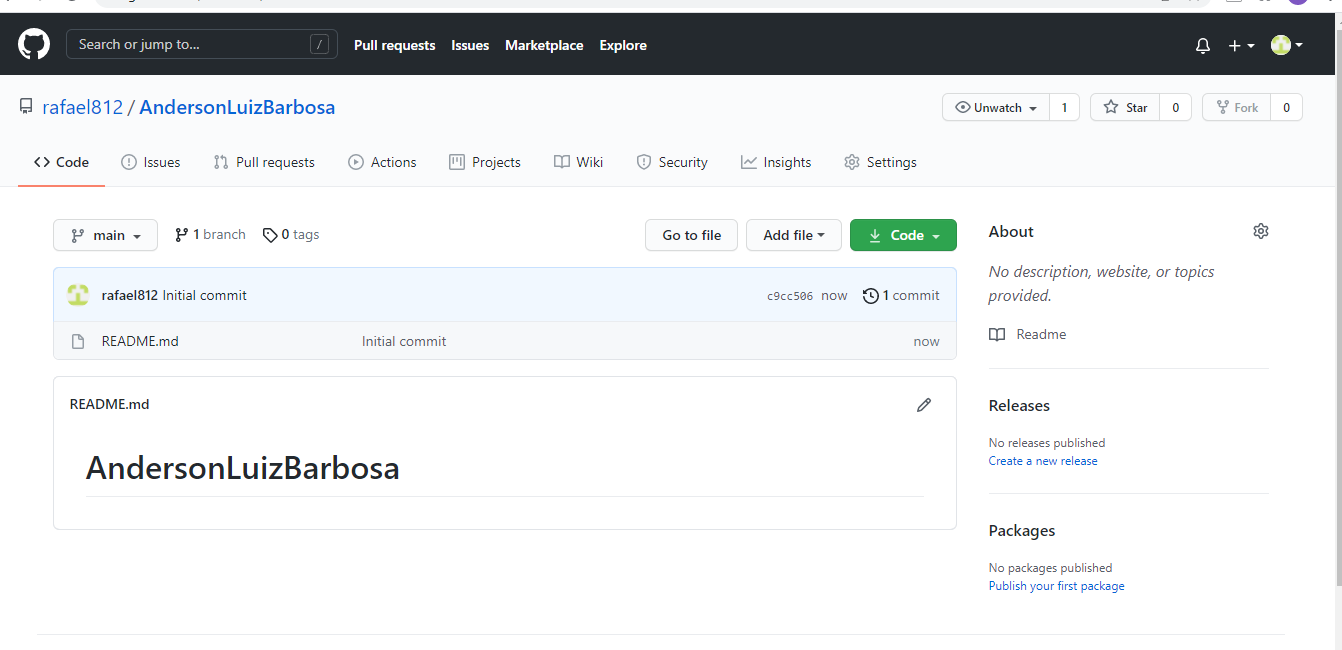


**Atividade 16 - Prática - Gestão de Configuração de Software**

1- Criação de um repositório no GitHub;

2- Inclusão da equipe e do professor (usuário no GitHub: AndersonLuizBarbosa) como equipe nesse repositório;

Inclusão dos documentos de requisitos e do projeto de interface no repositório criado.



**Rafael812/AndersonLuizBarbosa**

**Não sei mexer GitHub**

**Não sei Criação de um repositório no GitHub**

**Atividade 17 - Prática - Métricas de Software**

1- Uma reflexão sobre quais métricas e indicadores são os mais adequados para esse tipo de desenvolvimento;

2 - Cálculo dos Pontos de Função e preenchimento da planilha disponibilizada.

Já postei na planilha do excel

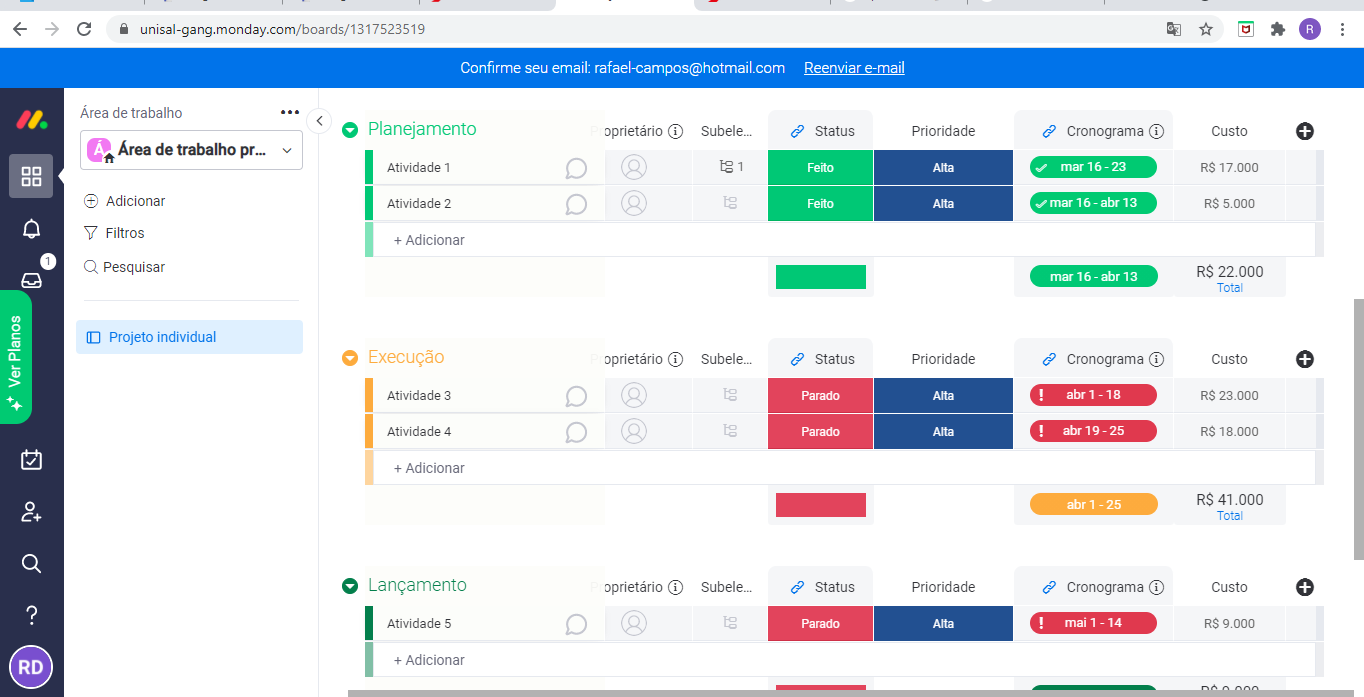
**Atividade 18 - Prática - Métodos Ágeis**

1 - A escolha de uma ferramenta (Jira Agile, PangoSCRUM, Trello, Monday entre outras);

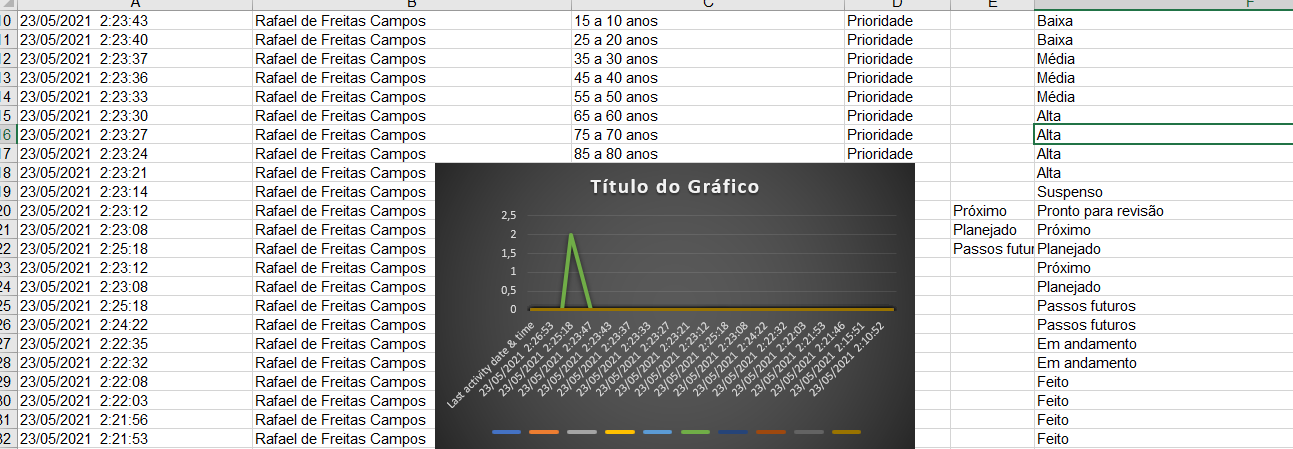
2 - A geração do backlog;

3 - A geração de ao menos um sprint backlog;

4- A simulação de progresso e acompanhamento via BurnDown Chart.







**Atividade 19 - Prática - Teste de Software**

1- Uma reflexão sobre quais os tipos de teste mais adequados para a verificação e validação dessa aplicação;

2- Elaboração de um plano de teste - utilizar como base a planilha disponibilizada.

1. Teste de Configuração

Testa se o software funciona no hardware a ser instalado.

2. Teste de Instalação

Testa se o software instala como planejado, em diferentes hardwares e sob diferentes condições, como pouco espaço de memória, interrupções de rede, interrupções na instalação etc.

3. Teste de Integridade

Testa a resistência do software à falhas (robustez).

4. Teste de Segurança

Testa se o sistema e os dados são acessados de maneira segura, apenas pelo autor das ações.

5. Teste Funcional

Testa os requisitos funcionais, as funções e os casos de uso. “A aplicação faz o que deveria fazer?”

6. Teste de Unidade

Testa um componente isolado ou classe do sistema.

7. Teste de Integração

Testa se um ou mais componentes combinados funcionam de maneira satisfatória. Há quem diga que o teste de integração é composto por vários testes de unidade.

8. Teste de Volume

Testa o comportamento do sistema operando com o volume “normal” de dados e transações envolvendo o banco de dados durante um longo período de tempo.

9. Teste de Performance

O teste de performance se divide em 3 tipos:

Teste de carga:

Testa o software sob as condições normais de uso. Ex.: tempo de resposta, número de transações por minuto, usuários simultâneos etc.

Teste de stress

Testa o software sob condições extremas de uso. Grande volume de transações e usuários simultâneos. Picos excessivos de carga em curtos períodos de tempo.

Teste de estabilidade

Testa se o sistema se mantém funcionando de maneira satisfatória após um período de uso.

10. Teste de Usabilidade

Teste focado na experiência do usuário, consistência da interface, layout, acesso às funcionalidades etc.

11. Testes de Caixa Branca e Caixa Preta

Basicamente, teste de caixa branca envolve o código e o de caixa-preta, não.

12. Teste de Regressão

Reteste de um sistema ou componente para verificar se alguma modificação recente causou algum efeito indesejado, além de, certificar se o sistema ainda atende os requisitos.

13. Teste de Manutenção

Testa se a mudança de ambiente não interferiu no funcionamento do sistema.