**Projeto Aplicado – Relatório Final**

|  |  |
| --- | --- |
| **Nome do Aluno** | **Albert Martins Vieira** |
| **Título do Trabalho** |  |
| **Curso** | MBA em Engenharia de Software com Métodos Ágeis |
| **Linha de Especialização** | Modelo de criação e evolução de produto de software com métodos ágeis |
| **Orientador** | **Ítalo Rodrigues Castro** |
| **Data** |  |

**INTRODUÇÃO**

1. Apresentação do Desafio e da Solução:
   1. Setor do mercado e a justificativa de tal seleção;
   2. Características e restrições de escopo do Desafio;
   3. Oportunidade vislumbrada que motivou o desenvolvimento da Solução.

**a. Setor de mercado**

O setor de mercado em que minha proposta se encaixa é política e projetos públicos de governo. O setor político atual apresenta momento instável no país, muitos políticos sendo investigados por desvios de verbas em projetos, propostas em época de eleição não se convertem em projetos finalizados e a população cada dia mais interessada em fiscalizar e cobrar estas promessas. A solução proposta visa inovar através do desenvolvimento de um software que centralize e facilite o acesso as informações de propostas de candidatos e possibilite o acompanhamento da execução destas propostas pelo candidato eleito.

**b. Características e restrições**

Com estudo aprofundado sobre política e projetos públicos de governo, foram levantadas algumas características essenciais para o início do desenvolvimento da solução, que visão ser comprovados através de protótipos e MVPs para que sua evolução entre em uma melhoria contínua. Entre as características temos:

- Centralizar as informações com foco na facilidade de acesso.

- Desenvolvimento de um website, onde a população poderá acessar as principais informações sobre os projetos dos partidos por candidatos.

- Os projetos poderão ser acessados por área específica, como por exemplo, saúde, educação, etc.

- Disponibilização de uma interface onde a população possa comparar as propostas por temas e candidatos.

- Possibilidade de envio de notificações para o usuário sobre atualizações de projetos selecionados.

- Utilização de gráficos para notificar o percentual de propostas por áreas e sobre a evolução dos projetos pelo governo eleito.

Também se observou restrições iniciais para o escopo que poderão ser modificados com os resultados das validações do MVP, tais como:

- Controle de intensões de votos da população.

- Histórico sobre a relação de candidatos e percentual de votos por eleição.

- Foco que não seja sobre as propostas e projetos políticos e informações dos candidatos.

**c. Oportunidade vislumbrada**

Após a realização de estudos sobre propostas políticas de governo, observam-se no país que as informações sobre propostas e projetos políticos de governo são pouco divulgadas, trazendo dificuldades no acesso dessas informações pela população. Além disso, durante o mandato do governo a população não consegue ter um acompanhamento da execução desses projetos de uma forma eficiente, onde as principais informações não estão em base de dados centralizada e também de forma que possa ser acessada de qualquer lugar.

Com isso a criação do software visa melhorar o acesso a informação, trazendo a população as propostas dos candidatos e acompanhamento da execução destas pelo governo eleito, auxiliando na formação política dos cidadãos.

Sendo assim, para verificar a viabilidade do projeto proposto frente a oportunidade, realizou-se uma pesquisa utilizando a ferramenta SurveyMonkey, onde verificou-se a viabilidade do projeto proposto no desafio.

Pesquisa desk:

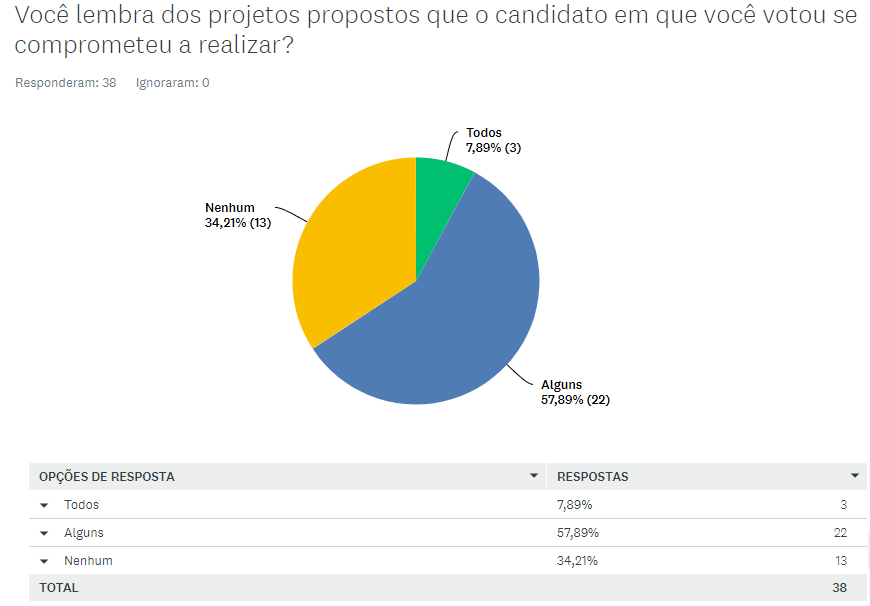


Gráfico 1 - Propostas do Candidato

Fonte: acervo pessoal

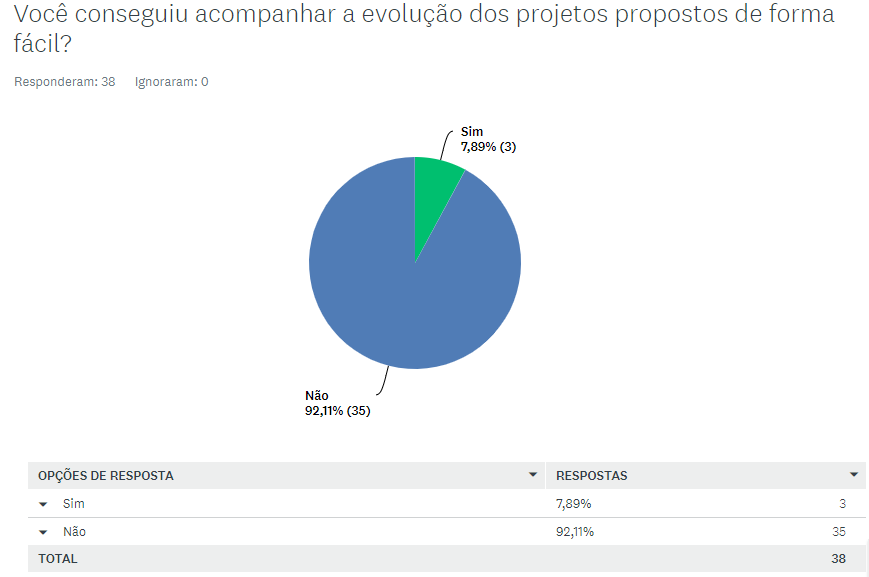


Gráfico 2 - Evolução das propostas do candidato eleito

Fonte: acervo pessoal

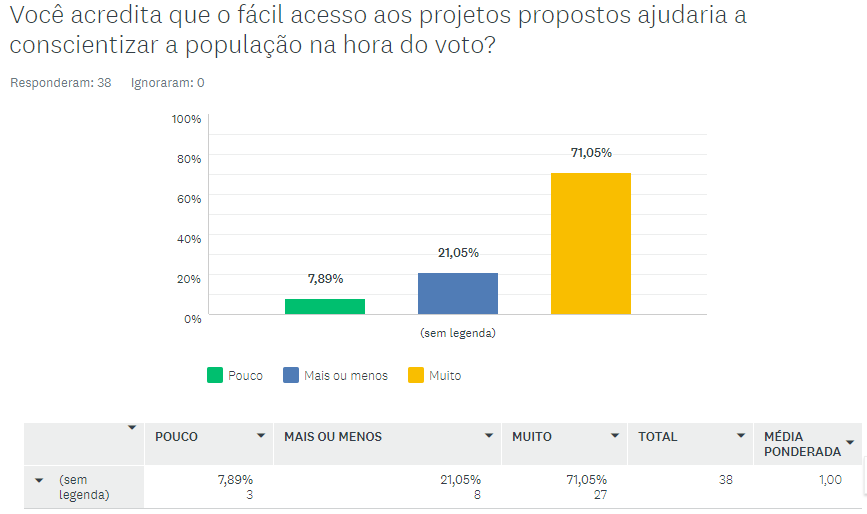


Gráfico 3 - Fácil acesso a informação de propostas

Fonte: acervo pessoal

1. Identificação da(s) pessoa(s) envolvida(s) no desafio:

As pessoas envolvidas foram levantadas através da criação do mapa de empatia, a fim de proporcionar um conhecimento profundo do cliente/usuário e criação de personas para identificação do cliente ideal do produto a ser desenvolvido. Através dos resultados destas ferramentas, obteve-se insumos para criação da matriz de stakeholders.

Principais Envolvidos:

Usuários, políticos, partidos, TSE, governo e fornecedores.

**Matrizes e particularidades dos envolvidos:**

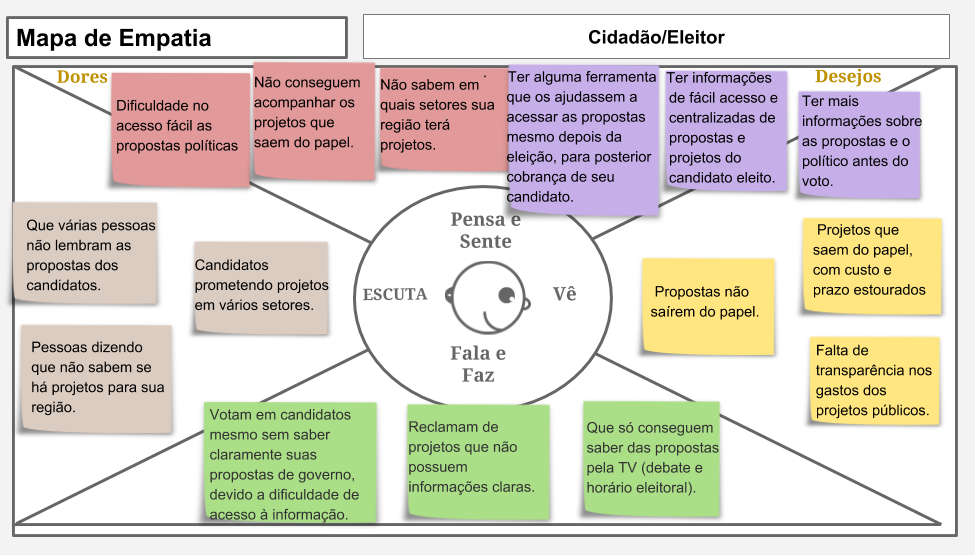
**Mapa de Empatia:**

Figura 1 – Mapa de Empatia

Fonte: acervo pessoal

**Personas:**

**1 – Gabriel Moraes**





Figura 2 – Persona 1

Fonte: acervo pessoal

**2 – Helen Nascimento**



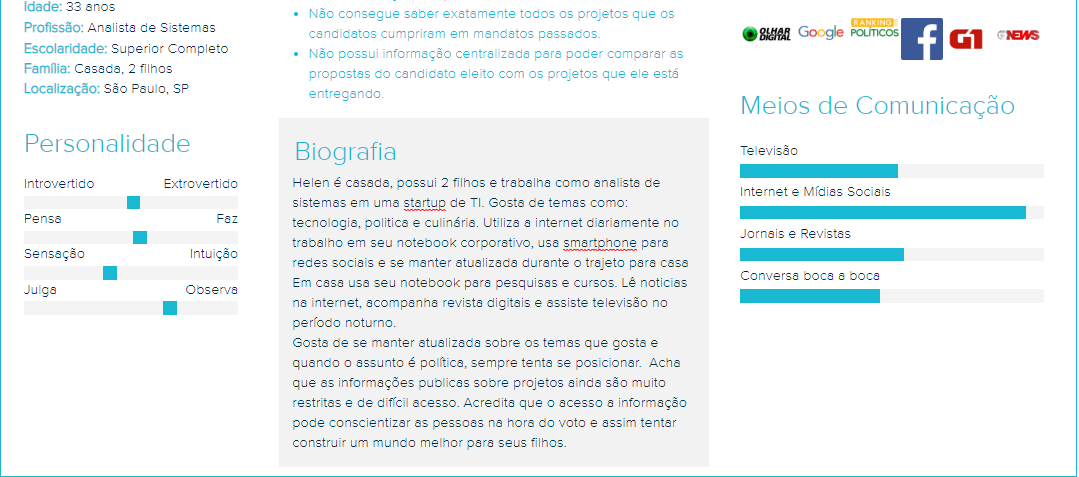


Figura 3 – Persona 2

Fonte: acervo pessoal

**Matriz de Stakeholders:**

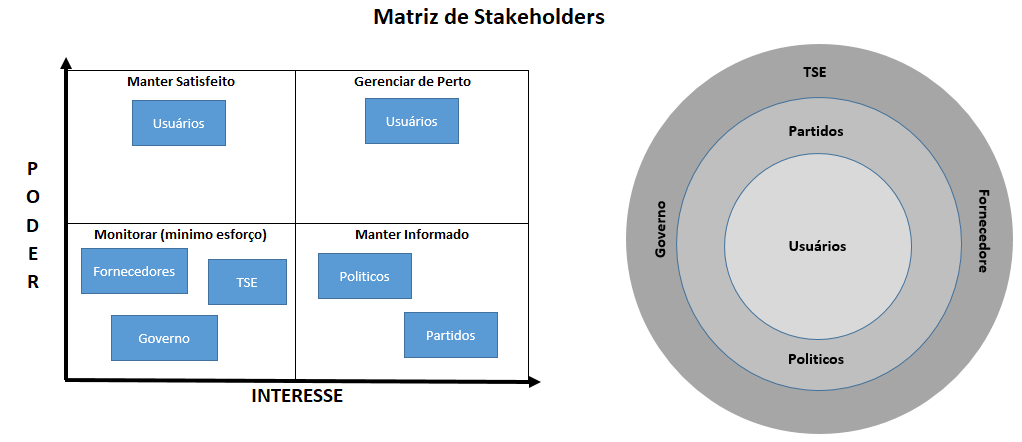


Figura 4 – Matriz de Stakeholders

Fonte: acervo pessoal

1. Construção da proposta de solução:
   1. Requisitos da construção do protótipo/MVP, com descrição do experimento e as métricas de validação;
   2. Indicadores econômico-financeiros do projeto.

**a. Requisitos para o MVP e descrição do experimento e métricas**

Como atividade prévia do levantamento dos requisitos e para maior entendimento das necessidades do usuário e de que forma a solução proposta consegue gerar valor, realizou-se a criação do Bussiness Model Canvas e Canvas de proposta de valor, através de seus resultados foi obtido os insumos para definição dos requisitos para o MVP.

Como principais requisitos temos:

- Criação da arquitetura da solução em formato Website.

- Apresentação das propostas dos candidatos a presidente e governador do estado de São Paulo para a próxima eleição.

- Gestão de candidatos.

- Gestão de propostas.

- Pesquisa por tema de proposta e candidatos.

- Comparação de propostas e projetos por tema e candidato.

- Apresentar informações sobre projetos realizados no último mandato deste candidato.

Com os requisitos já definidos, foram realizados experimentos com os seguintes objetivos:

- Receber feedback do grupo de usuários sobre a relevância das informações apresentadas no site como as propostas e projetos políticos.

- Verificar a necessidade de informações mais aprofundadas sobre os partidos e candidatos.

- Entender o melhor formato para comparação de propostas anteriores X projetos realizados no mandato do candidato.

- Verificar a utilidade de uma funcionalidade de notificação de projetos por região ou tema.

Serão utilizadas as seguintes métricas para validação dos experimentos no MVP:

- Medição da satisfação do usuário referente a gestão de candidatos.

- Medição da satisfação do usuário referente a gestão de propostas.

- Medição de satisfação no módulo de comparação de propostas X projetos do candidato eleito.

- Quantidade de pesquisas realizadas através dos filtros de busca.

- Quantidade de usuários que cadastraram seus e-mails e telefones para recebimento de notificações.

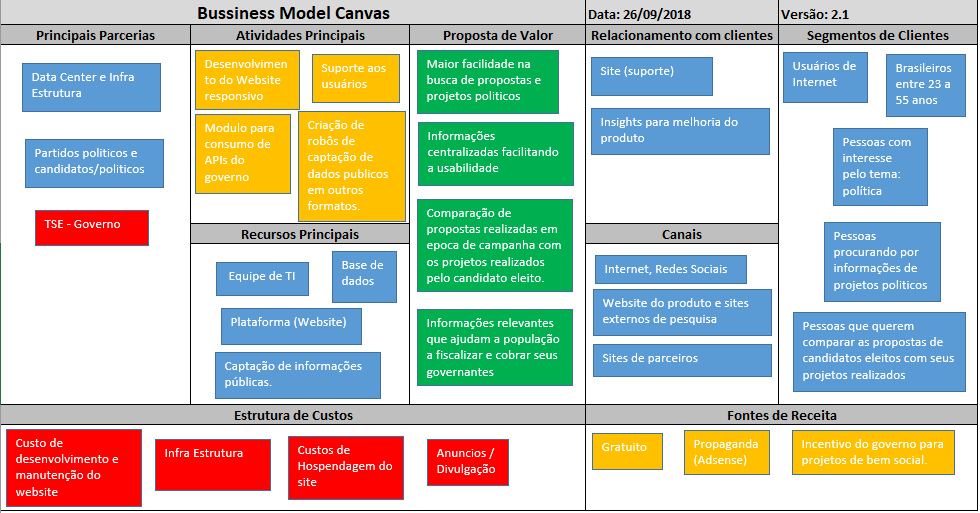


Figura 5 – Bussiness Model Canvas

Fonte: acervo pessoal

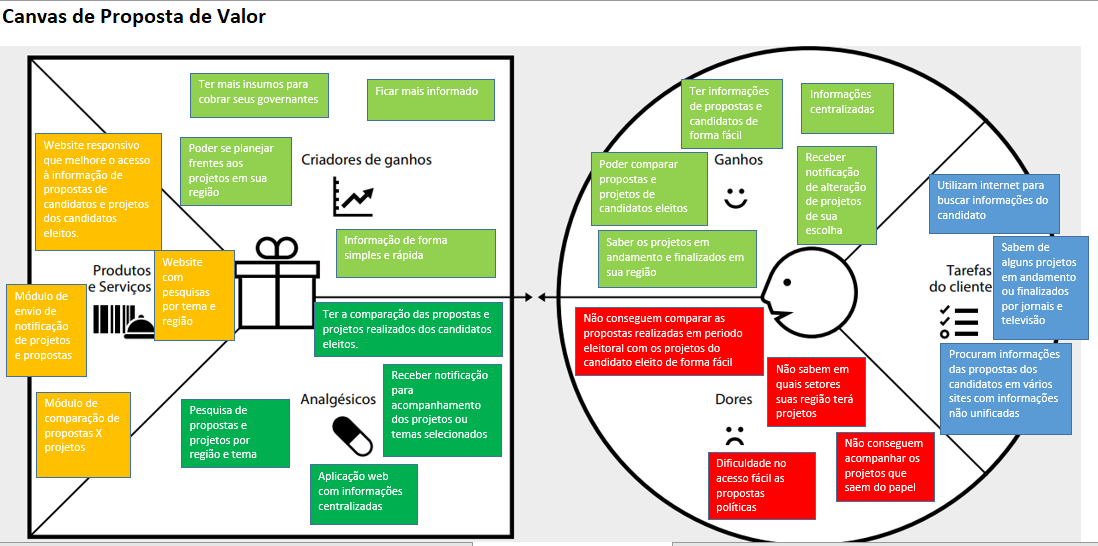


Figura 6 – Canvas Proposta de valor

Fonte: acervo pessoal

**b. Indicadores econômico-financeiros do projeto**

Com o objetivo de disponibilizar a informação gratuita ao usuário final, os indicadores econômicos-financeiros do projeto foram levantados através de pesquisas de rentabilização neste formato de negócio e reuniões e entrevistas com empresas/startups que possuem negócios atuando nos mesmos moldes. Com isto chegou-se aos principais indicadores:

- Rentabilidade com propagandas.

- Rentabilidade com disponibilização de serviços com informações de projetos políticos.

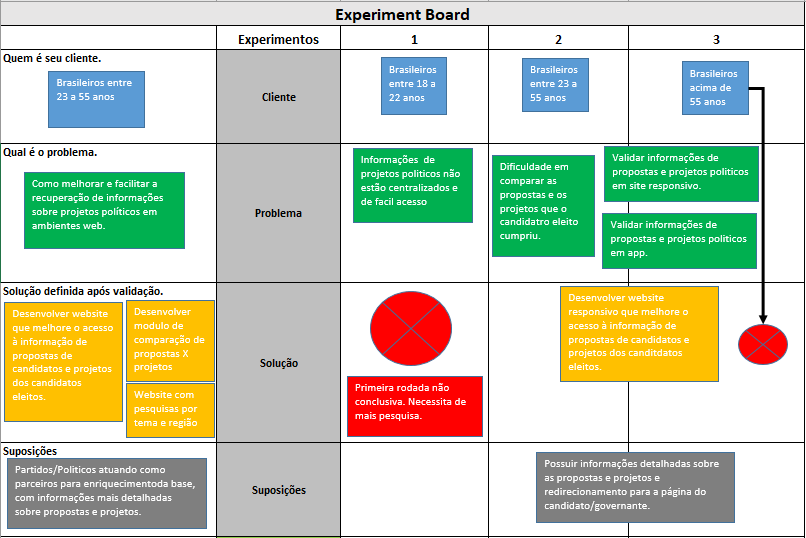
- Rentabilidade com disponibilização de serviços com informações de propostas de candidatos.

- Incentivos governamentais para projetos de bem social.

**DESCRIÇÃO DO DESENVOLVIMENTO DO TRABALHO**

1. Definição do produto:
   1. Definição do problema
   2. Definição de proposta de solução
   3. Apresentação da prototipação.

Com a imersão no dia a dia do nosso usuário ideal e utilizando a ferramenta “Experiment Board” conseguiu-se entender, testar e validar seus problemas e suas necessidades, desenvolvendo uma solução alinhada ao que realmente cria valor ao usuário final.



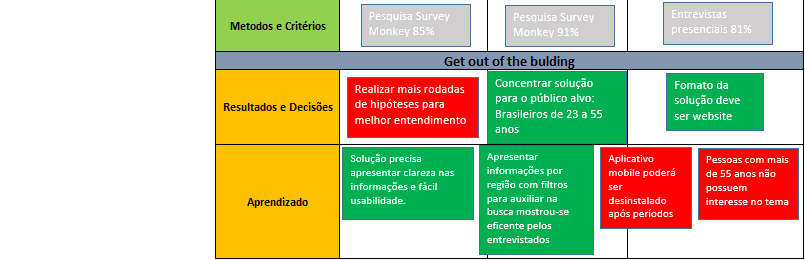


Figura 7 – Experiment Board

Fonte: acervo pessoal

**a. Problema**

Como melhorar e facilitar a recuperação de informações sobre projetos políticos em ambientes web.

**b. Solução**

Desenvolver website que melhore o acesso à informação de propostas de candidatos e projetos dos candidatos eleitos, disponibilizando mecanismos de pesquisas e comparação.

**c. Protótipo**

1. Definição do processo de codificação:
   1. Definição dos princípios de codificação adotados;
   2. Definição das ferramentas de validação da qualidade do código.

**a. Definição dos princípios de codificação adotados**

Antes de iniciar a codificação do MVP, foram alinhados os princípios e técnicas que permearam durante todo o desenvolvimento, condensando-as no contexto de desenvolvimento ágil para tornar os processos e a codificação mais eficiente.

Para o desenvolvimento do MVP foram utilizados padrões de HTML, CSS, Javascript e Java, alinhados com os princípios definidos previamente. Entre os princípios estão:

**Arquitetura incremental**

Uma premissa para a definição arquitetural, foi escolher uma arquitetura que possibilitasse a evolução do produto à medida que o desenvolvimento fosse progredindo e o MVP fosse evoluindo. Tendo como princípio a resposta a mudanças e software em funcionamento difundidos na cultura ágil.

**Versionamento de código**

Um dos princípios fundamentais adotados foi a utilização de um sistema de controle de versão, juntamente com o processo de versionamento, com objetivo de dar maior segurança ao projeto, gerenciar o histórico de mudança, rastrear alterações e garantir estabilidade ao website que está sendo desenvolvido.

A ferramenta utilizada foi o Git juntamente com o Github para realizar este controle. Foram versionados todos artefados produzidos como: Código fonte do protótipo, código de teste, pesquisas e documentações.

As seguintes boas práticas de versionamento foram seguidas:

- Commits com mensagens significativas.

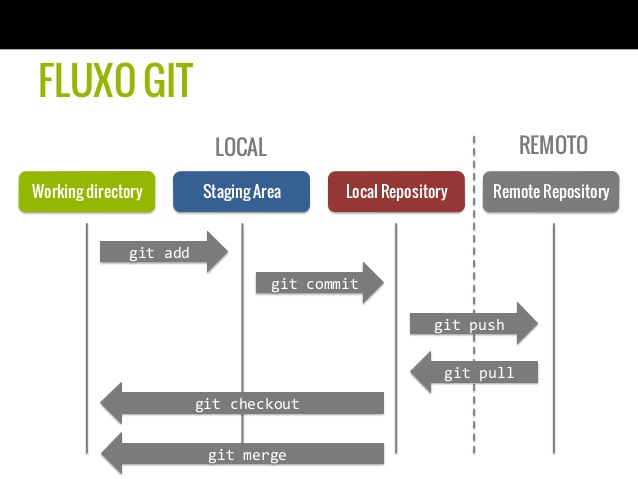
- Commits pequenos e diários.

- Repositório atualizado.

- Branch máster sempre funcional.

Fluxo do versionamento:



\*\*\*\*\* Colocar print do github que estará o projeto e todos os artefatos

**Padronização**

A padronização adota para a codificação do protótipo e dos testes é flexível e seguem os padrões e boas práticas utilizados na comunidade. Como a comunidade java é bem difundida no mercado, não houve problemas para encontrar recomendações e documentação para auxílio da implementação.

\*\*\*\* Inserir print do eclipse com organização das classes.

**Reutilização de código**

Todos os componentes, métodos e serviços desenvolvidos que podem se repetir foram criados de forma que possam ser reutilizados. Outro ponto foi a utilização de frameworks de mercado, para minimizar a recriação de algo já existente e utilizado na comunidade e facilitando a otimização do desenvolvimento.

Alguns frameworks/bibliotecas que foram reutilizados são:

Bootstrap – Framework de código aberto utilizado na criação do protótipo/MVP na parte de front-end. As vantagens de sua utilização é o suporte a web design responsivo e extensa documentação.

Faker – Biblioteca open-source para simulação de dados e auxílios na inserção de dados nos campos e testes com dados gerados randomicamente.

**b. Definição das ferramentas de validação da qualidade do código.**

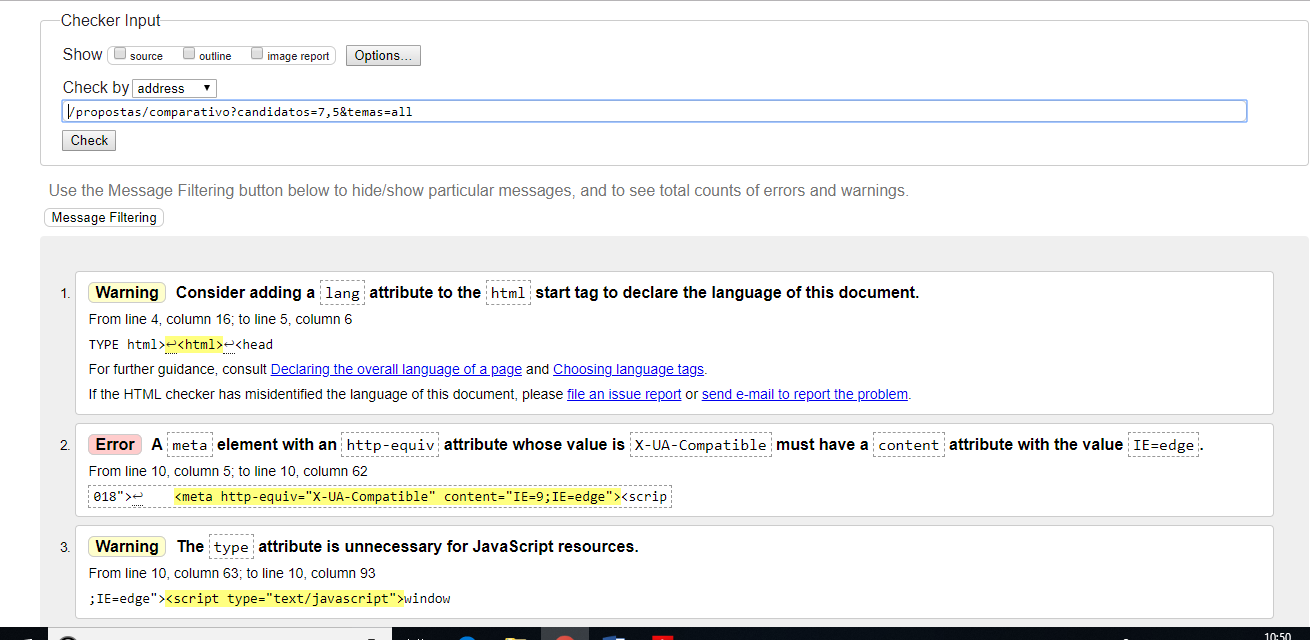
Prezando pela qualidade da aplicação, foram definidas algumas ferramentas de validação de código, preocupando-se com a garantia da qualidade desde as fases iniciais do desenvolvimento. Dentre as ferramentas estão:

Junit - Framework open-source utilizado na criação dos testes unitários. Entre suas facilidades estão: criação rápida de código de teste, comunidade ativa e possibilidade de asserções dos resultados esperados.

\*\*\*Inserir trecho de um teste unitário via junit – Exemplo:

<http://dtidigital.com.br/blog/testes-unitarios-utilizando-mock/>

CSS HTML Validator – Editor de HTML que auxilia na validação e detecção de erros comuns no HTML e CSS da página. Foi utilizado para ajudar a detectar atributos, comentários e códigos obsoletos que poderiam afetar a otimização do website.



Verificar para inserir no validador protótipo final

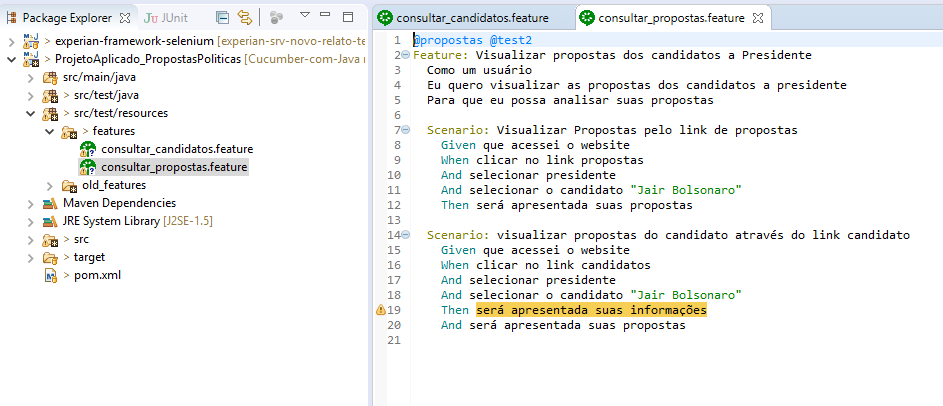
1. Definições de processo da qualidade:
   1. Definição do modelo de especificação de testes;
   2. Definição da infra-estrutura de testes automatizados;
   3. Definição do processo de automação de testes.

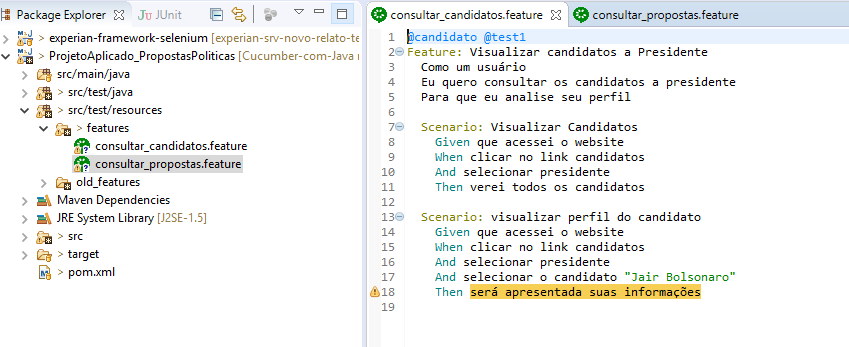
Como parte das atividades da equipe de desenvolvimentos do projeto, o processo de qualidade e testes seguiram a metodologia ágil e foram definidos e executados dentro das respectivas sprints correntes. Conforme as estórias do usuário eram priorizadas na planning, as atividades de testes eram criadas e executadas, tendo como premissa a qualidade da entrega para finalização da Sprint.

**a. Definição do modelo de especificação de testes**

Como modelos de especificação de testes, foi adotado a técnica de BDD - Behavior Driven Development (Desenvolvimento Guiado a Comportamento), nesta técnica o foco é no comportamento da aplicação e comumente utilizada para testes.

Os testes foram focados na navegação e experiência do usuário através do protótipo/mvp. Todos os testes foram organizados e separados através das features e seus cenários. A seguir alguns testes que foram executados para o projeto:





**b. Definição da infra-estrutura de testes automatizados**

A definição da infra-estrutura dos testes automatizados foi tomada de forma que o reaproveitamento de tecnologia e linguagem de código fosse possível, tendo sinergia com o desenvolvimento da aplicação e agilidade no processo.

A linguagem definida para automação dos testes foi Java. Utilizou-se frameworks de mercado para auxiliar na criação dos testes tais como:

Selenium-Webdriver – Framework difundido no mercado responsável pela interação e manipulação do HTML da página.

Cucumber – Framework utilizado para execução dos testes de aceitação e e parser da linguagem de negócio Gherkin.

Eclipse – IDE utilizada para o desenvolvimento dos testes em Java.

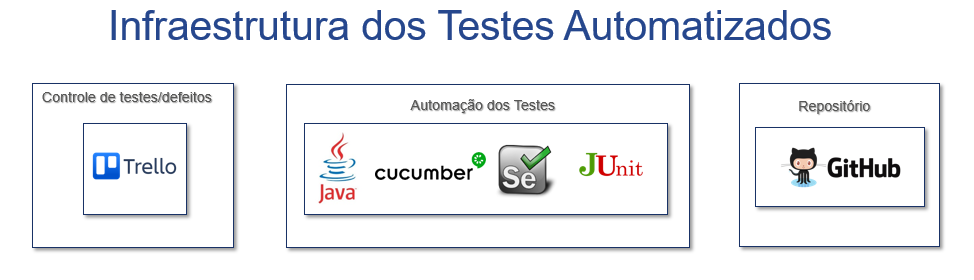


Figura 7 – Infraestrutura dos testes

Fonte: acervo pessoal

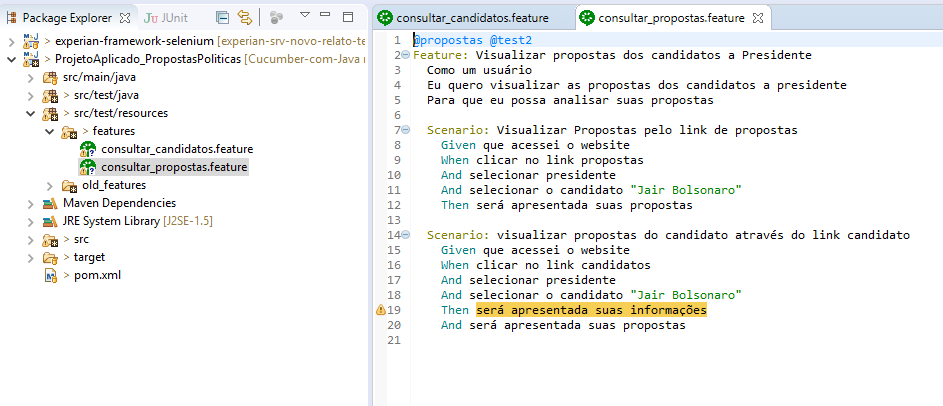
**C. Definição do processo de automação de testes.**

Para o processo de automação de teste foi utilizado o conceito shift-left, na qual os testes são executados desde o início do ciclo de desenvolvimento da solução, melhorando a qualidade e detectando erros mais cedo. De forma prática, após a definição dos primeiros protótipos de tela, foram definidos os possíveis elementos da página e seus identificadores (ID), com estas informações em mão mais as regras de negócio, os testes já começaram a ser desenvolvidos em paralelo com o desenvolvimento do MVP. Com a entrega dos primeiros componentes, os primeiros testes já foram executados em ambientes locais de forma automatizada.

Algumas premissas foram adotadas na automação de testes, afim de padronizar a escrita e facilitar os testes de comportamento do website, tais como:

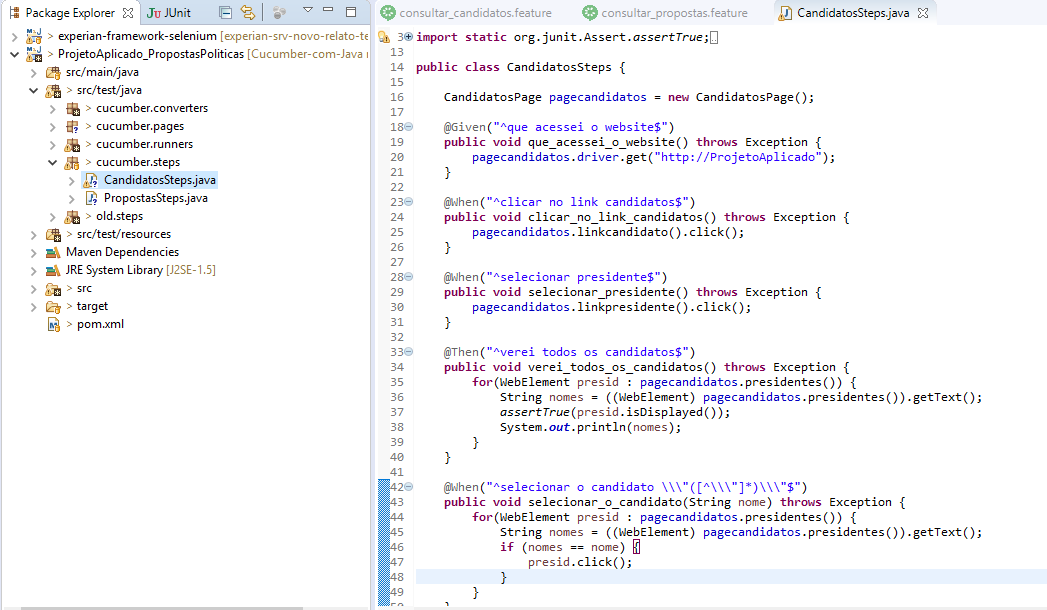
Camada de Negócio - Cucumber:

Antes de automatizar os testes, o arquivo de feature (camada de negócio) era escrito e atrelado a história de usuário correspondente.



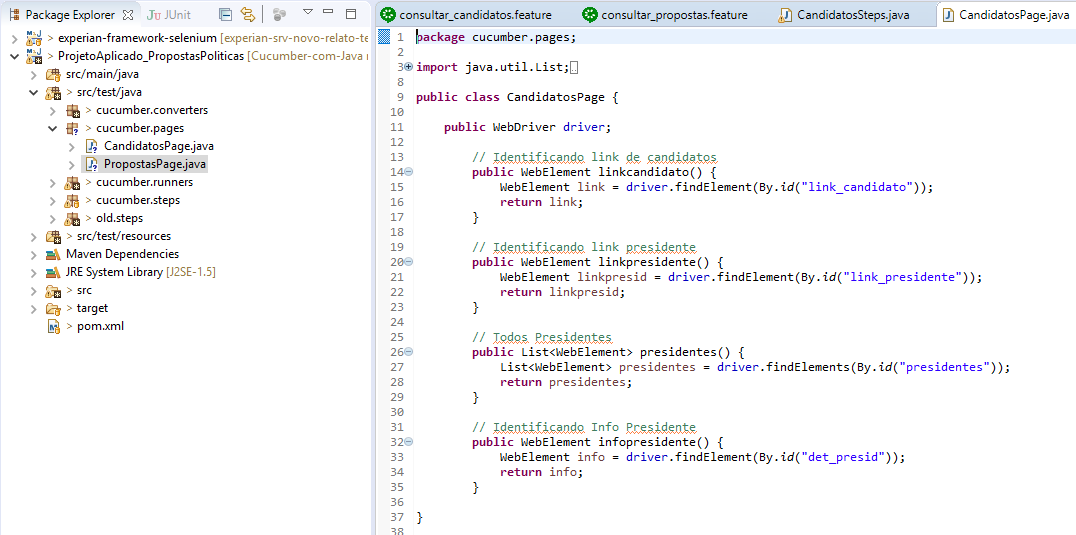
Camada de Código – Step Definition:

Para cada linha do feature file foi desenvolvido o código de teste nesta camada, realizando as ações desejadas.



Repositório de elementos – Page Object:

Camada responsável pela identificação dos elementos da página (Html).



Com estes processos e premissas definidos, foi possível ter reutilização e padronização no código de teste, maior agilidade na criação da automação e entrega com maior qualidade desde o início do desenvolvimento.

1. Definição de dimensionamento:
   1. Definição do processo de estimativas do time;
   2. Apresentação do Dimensionamento preliminar do escopo / backlog;
   3. Apresentação do Dimensionamento de time.

**a. Definição do processo de estimativas do time**

Todo o processo de estimativa de tamanho das estórias foi baseado na metodologia ágil “Scrum”. A ferramenta utilizada foi o “Scrum Poker”, no qual as histórias de usuário priorizadas para a Sprint eram estimadas pelo time. Neste processo, todos os integrantes do time tinham que expressar simultaneamente sua estimativa de tamanho para determinado requisito. Os tamanhos extremos eram justificados, e após as explicações todos integrantes estimavam novamente, definindo o “story point” daquele requisito. Desta forma todos os requisitos/ histórias de usuário foram estimadas e identificadas se conseguiriam ser atendidos dentro da Sprints planejadas.



Figura 5 – Scrum poker – Baralho utilizado para estimar as histórias de usuário

Fonte: <https://en.wikipedia.org/wiki/Planning_poker>

**b. Apresentação do dimensionamento preliminar do escopo / backlog;**

Com a maioria das estórias definidas, mesmo que podendo sofrer mudanças, foi realizado um dimensionamento inicial do escopo do backlog para análise de previsibilidade de data término, utilizando-se os pontos estimados das estórias priorizadas e similaridades entre as histórias do backlog e possíveis complexidades. Através deste estudo chegou-se ao número de 5 sprints com duração de 3 semanas cada, para entrega do MVP da solução.

Com data de início das atividades da Sprint no começo do mês de Outubro a previsão de finalização ficou entre a segunda e a terceira semana de Janeiro.

A baixo temos a representação do escopo preliminar com seu dimensionamento em “Story Points”.

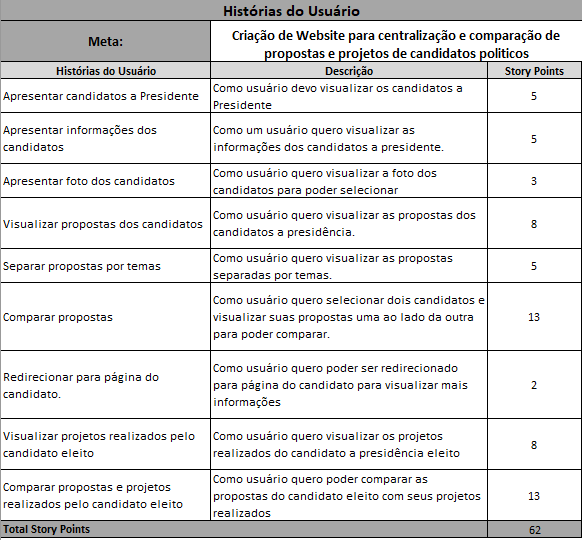


Figura 7 – Histórias estimadas com scrum poker

Fonte: acervo pessoal

**c. Apresentação do Dimensionamento de time.**

1. Definição de Devops:
   1. Definição do processo de entrega contínua;
   2. Definição da integração de processos técnicos.

**a. Definição do processo de entrega contínua:**

O processo de entrega contínua foi pensado de forma que conseguisse ser realizado em ciclos curtos e com frequência. Desta forma, os processos utilizados no desenvolvimento, armazenamento de dados e testes foram mapeados em formato que possibilitasse este ciclo contínuo.

Os principais processos estabelecidos que contribuíram para a entrega contínua foram:

- Controle de versão:

Todos artefatos criados eram versionados diariamente no github, fazendo com que os códigos ficassem estáveis, padronizados e pudessem ser retornados a um ponto estável, caso algo não funcionasse como o planejado.

- Entregas curtas:

As funcionalidades para o MVP foram quebradas em tarefas menores, para que fosse possível realizar entregas mais rápidas e facilitar os testes e validação do entregável.

- Desenvolvimento com foco na qualidade:

Utilizou-se ferramentas para validação da qualidade do código HTML gerado para os protótipos navegáveis, validando sua estrutura, formato e sintaxe.

- Testes para garantia da qualidade:

Testes foram automatizados e definidos de acordo com as histórias priorizadas e entregues. Testes de código, usabilidade e integração dos componentes do protótipo foram realizados.

Este ciclo foi repetido em todas as sprints e melhorado continuamente de acordo com os ajustes que se fizeram necessários, garantido que as entregas fossem contínuas e com qualidade.

**b. Definição da integração de processos técnicos**

Foi utilizado como premissa para integração de processos técnicos, ferramentas que facilitassem, automatizassem as entregas e que ajudasse a reduzir o risco de erros nas tarefas manuais.

Para a entrega do MVP, iniciou-se com o mínimo necessário para que fosse possível realizar as entregas utilizando integração contínua (CI), identificando pontos de evolução para fases posteriores.

O ferramental definido para integração dos processos técnicos iniciou-se com controlador de versão, script de compilação automatizado, checkins frequentes e testes automatizados e de usabilidade para manter o processo de compilação de forma ágil.

Ferramentas utilizadas para suportar a integração dos processos e melhoria contínua como um todo:

\*\*\*Inserir print das ferramentas e imagem devops

**DESCRIÇÃO DO TRABALHO FINAL**

1. Modelo de Desenvolvimento Ágil:
   1. Apresentação das principais características do processo;
   2. Apresentação das práticas adotadas;
   3. Descrição do tamanho do time;
   4. Apresentação de outras características do projeto;
   5. Apresentação de documentos que formalizam as principais decisões tomadas.

**a. Apresentação das principais características do processo:**

Como já mencionado em definições anteriores, a metodologia ágil foi utilizada em todo o ciclo de desenvolvimento do projeto aplicado, tendo os processos voltados ao framework “Scrum”. As principais características do processo foram a entrega das histórias de maior valor, feedback constante das entregas e melhoria dos pontos não satisfatórios identificados, experimentação de modelos e ferramentas de prototipação para melhor definição de como seguir com o desenvolvimento do projeto aplicado, adaptação as mudanças de decisões, buscar a melhoria contínua e simplicidade nos processos com intuito de ser mais ágil.

**b. Apresentação das práticas adotadas:**

Foram adotadas as práticas ágeis já difundidas na comunidade e no framework “Scrum”, combinando com algumas práticas do framework Kanban. Dentre as práticas utilizadas destacam-se as:

Definição de requisitos como histórias de usuário:

Após as realizações de pesquisas e entrevistas com futuros usuários, os requisitos foram definidos no formato de histórias de usuário, sendo priorizadas de acordo com a relevância identificada nestas pesquisas. Para gerenciamento e controle das histórias, notas e comentários de sua evolução, foi utilizado o quadro Kanban representado pela ferramenta Trello.

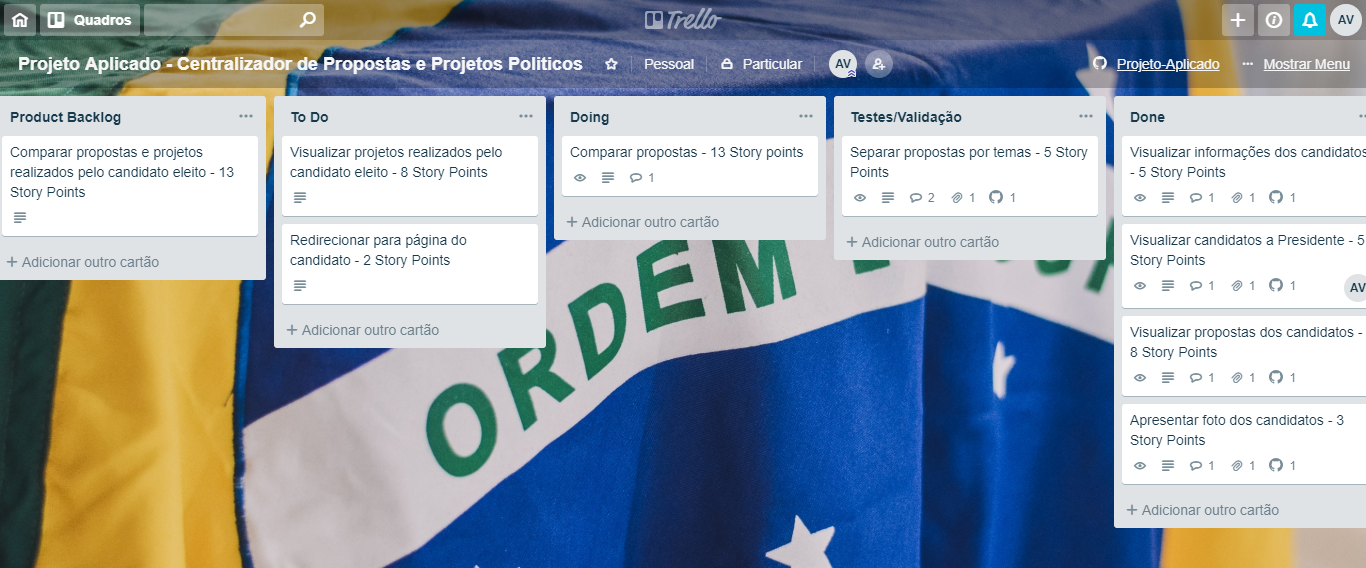


Figura 7 – Quadro Kanban - Trello

Fonte: conta pessoal - <https://trello.com>

Estimativa das atividades:

As histórias de usuário foram estimadas utilizando “Scrum Poker”, conforme abordado e exemplificado no item 7 do relatório.

Entregas de funcionalidades dividas por Sprint:

Após as estimativas das histórias de usuário, eram analisadas quantas histórias conseguiriam ser entregues dentro de uma Sprint, que definiu-se ter período de 3 semanas. Com isso foi possível ter entregas pequenas e rápidas, e ter uma previsão de entrega final do projeto conforme quadro abaixo:

Retrospectiva

**c. Descrição do tamanho do time:**

**RESULTADOS**

1. Descrição e análise dos resultados alcançados:
   1. Resultados positivos encontrados (caso existam). Explique;
   2. Resultados negativos encontrados (caso existam). Explique.

**CONCLUSÃO**

1. Apresentação da conclusão:
   1. Principais contribuições que seu projeto gera aos envolvidos;
   2. Inovações, particularidades ou vantagens que o projeto/resultado possui em relação a similares;
   3. Limitações do projeto;
   4. Próximos passos necessários para que o projeto evolua/se desenvolva.