**Requisitos:**

* Cadastros:
  + *Team:*
    - *Time de trabalho (setores, usuários e projetos próprios);*
    - *Funciona como se fosse o código da loja, todo registro de cadastro e lançamento será associado a um Team;*
  + **Setores:**
    - Campos: nome, estado (normal, bloqueado e excluído), [team];
  + **Usuários:**
    - Campos: nome, sigla, setor e estado (normal, bloqueado e excluído), [team];
    - Usuário deverá realizar login ao acessar o programa. Após login, todos os lançamentos que ele fizer ficarão associados ao próprio;
  + **Projetos:**
    - Campos: nome, estado (normal, excluído), [team];
    - Produtos que equipe desenvolve;
  + Iterações:
    - Campos: nome, data inicia, data final, estado, [team];
    - Período de trabalho de fluxo não contínuo;
  + Fluxo:
    - Campos: Nome, Tipo (Todo, Doing e Done), Usuários Liberados;
    - Representa as etapas que um requisito ou bug passará até estar disponível para produção;
    - Fases verticais da iteração;
    - Tipos do fluxo básico:
      * ToDo: tarefas em fase inicial, estão prontas para serem trabalhadas. Deve existir apenas uma na iteração;
      * Doing: tarefas que estão sendo trabalhadas;
      * Done: tarefas prontas para a próxima fase;
* Lançamentos:
  + **Requisitos;**
    - Campos: protocolo, descrição, motivação, projeto (algum já existente ou nenhum), versão, valor agregado (1-10), prioridade (baixa, média ou alta), clientes solicitantes (consulta), data de lançamento, data última alteração, usuário criou, usuário pendente, estado (novo, verificado, desenvolvimento, pronto), anotações,
    - Quem pode criar: qualquer usuário;
  + **Bugs**;
    - Campos: protocolo, descrição, projeto, versão, clientes reclamantes, data de lançamento, criticidade (critico, alto, médio, baixo), data última alteração, usuário criou, usuário pendente, estado (novo, verificado, corrigindo, pronto), anotações;
    - Quem pode criar: qualquer usuário;
  + Impedimento:
    - Campos: protocolo, protocolo relacionado, descrição, usuário criou, usuário pendente, estado (aberto, verificado e resolvido), [team];
    - Somente quem criou pode modificar o estado do impedimento para “resolvido”;
    - Deve estar relacionado a um requisito ou bug;
    - Quem pode criar: desenvolvedores e testadores;
* Cards:
  + Item que compõem a iteração;
  + Um Card representa uma tarefa a ser concluída;
  + Pode ser referente a um Requisito ou um Bug, fazendo uma referência do protocolo no card;
  + É o item que ficará visível aos usuários no board;
  + Estimativa: utilizar matriz [*Fibonacci-Like* + *T-Shirt Size*](http://blog.plataformatec.com.br/2019/09/relacao-de-story-points-com-o-tempo-de-desenvolvimento-lead-time/) *por enquanto;*
    - *Complexidade: PP, P, M, G e GG;*
    - *Incerteza: PP, P, M, G e GG;*
    - *Critérios:*
      * *Complexidade de desenvolvimento, complexidade de testes, dependência de terceiros, incertezas relacionadas a história, dificuldade para por em produção, etc.;*
* Acompanhamento:
  + Um usuário pode marcar um protocolo para acompanhar;
  + Protocolos que estão sendo acompanhados devem aparecer no dashboard do usuário;
  + Para parar de acompanhar ou aparecer no dashboard, usuário deverá fazer isso manualmente;
* Relatórios:
  + Posição Iteração;
    - Protocolo, Tipo (requisito ou bug), Descrição e Estado;
  + Métricas:
    - Thoughput: dia, semana, mês;
    - Lead-Time;
  + CFD;

**Critérios:**

1. Product backlog separado do iteration backlog;
2. Multiplas iterações;
3. Incluir colunas no board;
4. Incluir raias no board;
5. Free

**Ferramentas analisadas:**

* Jira
* Azure Devops
* Kanban Ink
* Kanban Tool
* GitHub
* Trello