**Check List - Vídeo Locadora Imperial**

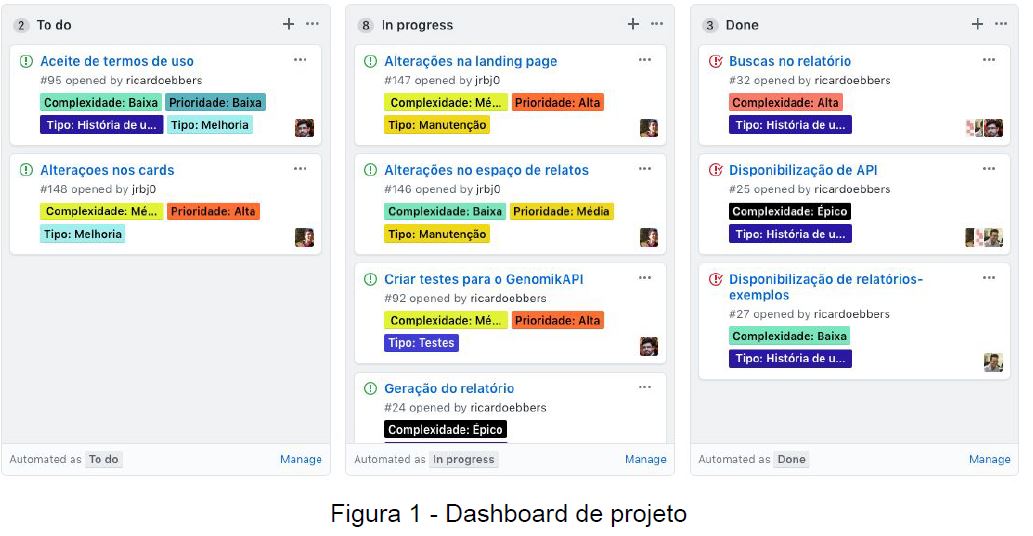
* **Documentação**
* **Análise**
* **Projeto**
* **Implementação**
* **Implantação**
* **Modelo de ciclo de vida:** o ciclo de desenvolvimento do software é iterativo.

## **Ciclo de vida do aplicativo em um projeto Agile.**

1. **Criação do Repositório do Projeto:**
2. **Introdução :** Contendo a contextualização do domínio do negócio ao qual o projeto está inserido, o problema e motivação (mercadológica, digamos assim) para o investimento no projeto (você vão fazer uma venda do projeto pra mim, então, tem que estar claro que existe uma oportunidade aqui) e apresentação da proposta.
3. **Gerência de Configuração e Ambiente:** Descrição das ferramentas e ambientes de desenvolvimento utilizados, com os respectivos links (i.e. ferramentas de desenvolvimento, linguagens, bancos de dados, bibliotecas, frameworks, repositórios de gerenciamento de versões e mudanças, servidores de implantação, entre outras).
4. **Características da aplicação :** Elucidação das características da aplicação, suas funcionalidades, diferenciais, etc.
5. **Documento de Requisitos:** identificação, análise, priorização, e eventual evolução dos objetivos.

* Introdução
* Descrição do Propósito do Sistema
* Descrição do Domínio
* Requisitos de Usuário

1. **Histórias do Usuário (HU):** deverão ser cadastradas e gerenciadas como "issues" no issue tracker do repositório no GitHub e devem ter um ciclo de vida a ser definido pelo próprio time.
2. **Máquina de estados: (i.e. To Do -> Doing -> Testing -> Done).**

****

1. **Processo de Desenvolvimento do Projeto:** Para a execução do projeto, o time deverá especificar e documentar explicita e claramente qual o processo de desenvolvimento utilizado no projeto.

* **Visão de Análise e Projeto (Arquitetura) :** Análise e Projeto do sistema a ser desenvolvido. podem ser utilizados especificações de casos de uso (para os mais 15% mais complexos) e diagramas para apoiar o projeto.
* **Visão de Implantação :** Visão e instruções de implantação do sistema. Um release notes com os erros, falhas e faltas conhecidos também é bem visto. Site onde ele está implantado e/ou um vídeo de demonstração promovendo o produto (pode estar no youtube ou outro serviço semelhante).

● **Visão de Uso :** Guia de uso da solução do ponto de vista de todos os stakeholders identificados.

● **Revisão do Projeto :** Descrição do processo de desenvolvimento, principais problemas e tomadas de decisão em relação ao projeto. Atribuição das atividades e técnicas de gerenciamento, monitoramento e controle.

● **Revisão Individual :** Lições aprendidas do ponto de vista individual dos membros do time em relação a execução do projeto em si.

1. **Postmortem do projeto:** Resumo do que funcionou e do que não funcionou durante a execução do projeto, e planos para melhorar o que não funcionou, lições aprendidas, etc.

1) Período: data de início - data de término da iteração

2) O que estava planejado?

a) atividades e pra quem

3) O que foi feito?

a) atividades e por quem

4) O que não foi feito?

a) atividades e por quem, quais os impedimentos

5) O que está planejado para a próxima iteração?

a) atividades e pra quem

6) Lições aprendidas (Post Mortem / Rationale)

1. **Código da aplicação:** O código da aplicação deverá estar versionado no respositório. A linguagem e demais ferramentas ou frameworks são de responsabilidade do time. Como sugestão, é indicado utilizar frameworks Web Ágeis como, por exemplo, Ruby on Rails, Grails, Spring Boot, Django, Flask, Zendo, entre outros.
2. **Testes funcionais:** Deverão ser produzidos testes funcionais automatizado seguindo a metodologia TDD, de acordo com a linguagem escolhida para o desenvolvimento do projeto. O time irá selecionar, se for o caso, quais HUs deverão ter testes funcionais produzidos para as mesmas.
3. **Testes de aceitação:** Deverão ser produzidos testes de aceitação automatizados seguindo a metodologia BDD. Como sugestão de ferramenta tem o Cucumber que funciona para algumas linguagens.
4. **Implantação da aplicação:** A versão final e funcional da aplicação deverá ser entregue implantada em algum serviço de nuvem, seja como PaaS (i.e Heroku) ou IaaS (i.e. AWS, Azure, Oracle Cloud, Google Cloud).
5. **Relatório Final do Projeto:** Eventualmente, na execução de projetos, podemos nos deparar com a necessidade de documentar ações e decisões por meio de relatórios técnicos ou ainda, realizar apresentações para discutir os resultados alcançados até então.

|  |  |  |
| --- | --- | --- |
| **Tarefa** | **Responsável** | **Situação** |
| Criação do Repositório do Projeto | Lilian | OK |
| Introdução | Rogério |  |
| Gerência de Configuração e Ambiente | Rogério |  |
| Características da aplicação | Rogério |  |
| Documento de Requisitos | Rafaela |  |
| Diagrama de entidade relacionamento e Diagrama lógico do banco de dados | Mônica | OK |
| Product Backlog:  Histórias do Usuário (HU) | Mônica | OK |
| Máquina de estados | Todos |  |
| Processo de Desenvolvimento do Projeto: | Todos |  |
| Postmortem do projeto | Osmário |  |
| Código da aplicação | Lilian |  |
| Banco de dados da aplicação | Lilian |  |
| Testes funcionais (TDD) | Lilian |  |
| Testes de aceitação (BDD) | Mônica |  |
| Implantação da aplicação | Lilian |  |
| Relatório Final do Projeto | Osmário |  |
| Verificar, ao final de tudo, o que vai ser entregue ao professor. | Mônica |  |