

PROPOSTA DE PROJETO

Objetivo

Este projeto tem caráter puramente acadêmico e educacional. Deve ser executado, preferencialmente, por equipes com 4 a 5 membros para tratar de diversos aspectos da Engenharia de Software (Documentação, Análise, Projeto, Implementação, Implantação), especificamente do **Ciclo de Vida do Software**.

Para isso, é disponibilizada a documentação dos requisitos de uma Vídeo Locadora chamada Imperial.

Prazos

A entrega do projeto se dará por meio de envio de email informando o repositório do time. O time deverá ter um release formalmente feito no repositório, até a data final do prazo.

Data final do prazo:

Entrega de artefatos

O projeto é constituído da solução construída de acordo com a proposta e todos os artefatos que a compõem (i.e. código fonte, imagens, arquivos de configuração, scripts, documentações de apoio, entre outras), versionada e compartilhadas por meio de um repositório no GitHub, a rastreabilidade do ciclo de vida, um relatório técnico final contendo a contextualização, escopo, tomadas de decisão, documentação da arquitetura da solução, projeto de implantação, etc. e a solução implantada em algum serviço de PaaS (Platform as a Service) como o Heroku ou a partir de uma solução de IaaS (Infrastructure as a Service) como, por exemplo, AWS, Azure, Oracle Cloud ou Google Cloud.

Lista de artefatos avaliativos

Repositório do projeto

O projeto deve estar salvo, documentado e com todo seu ciclo de vida versionado e rastreado em um repositório do GitHub a ser criado pelo time. O link do repositório será

informado em seção específica de Apresentação do Projeto em relatório técnico a ser entregue e que também estará salvo no próprio repositório.

Página de readme/inicial com nome do projeto, justificativa do projeto, descrição do produto, especificação do papel de cada membro da equipe (com informações de contato como email) e demais informações que o time achar pertinentes.

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, selecionando as melhores práticas e atividades dos modelos de processo apresentados em sala. É permitido utilizar atividades e/ou boas práticas de outros modelos não discutidos, desde que claramente documentados.

Rastreabilidade do Ciclo de Vida

O time deverá, a partir do documento de requisitos disponibilizado, promover a identificação, análise, priorização, e eventual evolução dos objetivos do time em relação ao projeto em forma de Histórias do Usuário (HU). As mesmas 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, de acordo com o processo de desenvolvimento de software especificado pelo time. Para isso, as HU deverão ter uma "máquina de estados" indicando o seu estado no momento presente (i.e. To Do -> Doing -> Testing -> Done). Um exemplo de visão de rastreabilidade de projeto é apresentado na Figura 1.

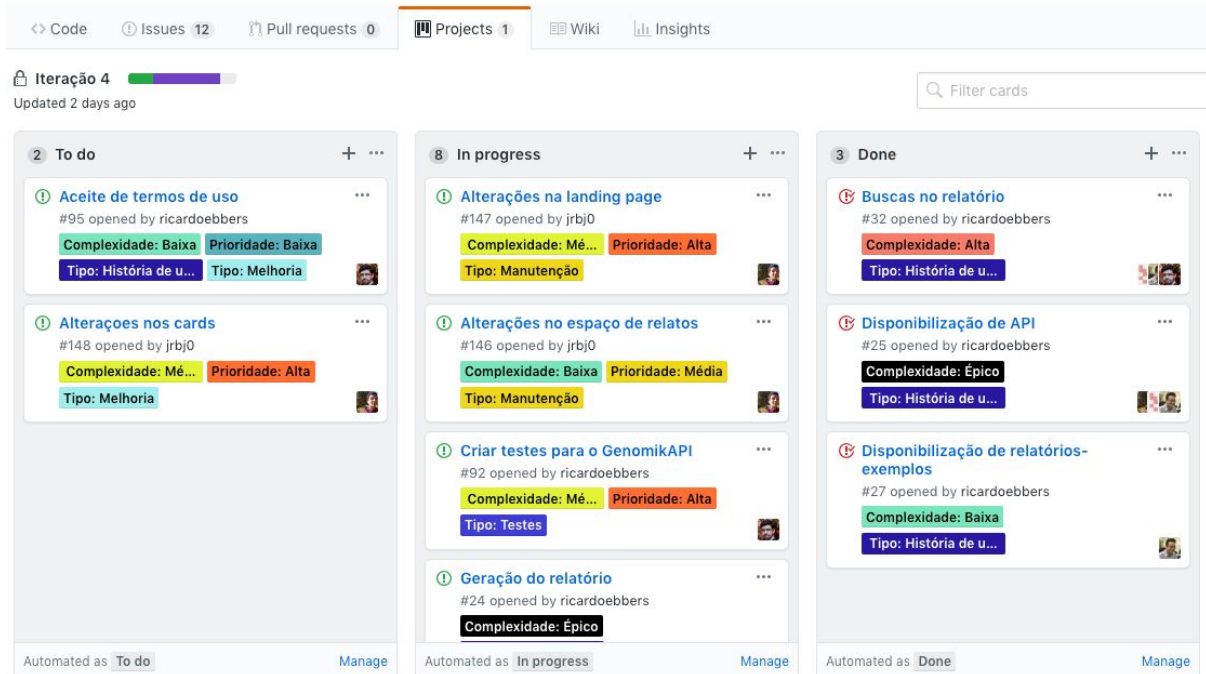


Figura 1 - Dashboard de projeto

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)

Código da aplicação

O código da aplicação deverá estar versionado no repositório. 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.

Testes funcionais

Deverão ser produzidos testes funcionais automatizados 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.

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. O time irá selecionar, se for o caso, quais HUs deverão ter testes funcionais produzidos para as mesmas.

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).

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. Neste contexto, considere seguir estes [guidelines](#) para apoiar a construção do seu relatório técnico ou ajudar a planejar a sua apresentação oral.

O projeto deve compreender, não limitado a, os seguintes itens ou tópicos:

- **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.
- **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).
- **Características da aplicação:** Elicitação das características da aplicação, suas funcionalidades, diferenciais, etc.
- **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.

Considerações de propriedade intelectual do projeto

Se você está pensando em trabalhar em um projeto que pode ter futuras aplicações comerciais, por favor, tenha em mente o seguinte:

- De acordo com a política da UFPE, a propriedade intelectual criada por alunos pertence exclusivamente a eles.
- De acordo com a política da disciplina, protótipos do projeto serão de acesso livre no GitHub durante a disciplina, mas não precisa necessariamente ter uma licença aberta.
- De acordo com a política da disciplina, nenhum NDA cobrirá qualquer aspecto de um projeto de disciplina. Todos os aspectos das -- ideias de projetos, storyboards, código fonte, casos de teste, as imagens de GUI, revisões de código -- serão publicamente visíveis e acessíveis durante todo o período da disciplina (e possivelmente depois, por questões históricas e de referência). O que os estudantes escolhem fazer com esses produtos após o final da aula é exclusivamente problema deles.
- Maiores informações sobre propriedade intelectual produzida na UFPE podem ser obtida na [Positiva - Diretoria de Inovação da UFPE](#).
- Em especial, veja o [manual de propriedade intelectual da Positiva](#).