

Plano de Projeto

Histórico da Revisão

Data	Versão	Descrição	Autor(es)
25/04/2025	Final	Revisão para a entrega final do plano de projeto	Rodrigo, Karl, Maicon

1. Identificação

- Nome do Projeto: ORCID++
- Equipe: clube de regatas do flamengo
- Data de criação do documento: 11/04/2025
- Membros da equipe:
Rodrigo de Freitas Lima – 12547510
Karl Cruz Altenhofen – 14585976
Maicon Chaves Marques - 14593530

2. Introdução

Em nosso projeto iremos desenvolver uma aplicação web, destinada a pesquisadores tendo como propósito melhorar a usabilidade do ORCID, utilizando sua API, pretendemos utilizar algumas partes já funcionais e desenvolver novas funcionalidades que melhoraram a usabilidade do site.

O modelo que será desenvolvido será o modelo cascata, já que a partir das análises realizados sobre o projeto, concluímos que haverá pouca mudança de requisitos, alta documentação e organização prévia do projeto.

O plano de projeto se dará nas etapas do modelo cascata: Definição de requisitos, Projeto de sistema e software, Implementação e teste unitário e Integração e teste do sistema.

O público-alvo deste projeto são os pesquisadores acadêmicos que utilizam o ORCID para gerenciar suas produções científicas, bem como as instituições de pesquisa que buscam melhorar a visibilidade e a interação com os perfis dos pesquisadores. O propósito do projeto é proporcionar uma experiência mais fluida e intuitiva ao navegar e interagir com os dados acadêmicos na plataforma ORCID. O uso do modelo cascata permitirá uma organização eficiente, com a entrega de resultados em cada fase, garantindo a integridade do sistema desde a concepção até a finalização.

3. Escopo do projeto

O projeto será uma reestruturação da página do ORCID, onde haverá mudança de design e adições de novas funcionalidades. Como acessar perfis de pesquisadores direto da citação em um artigo.

O objetivo é proporcionar uma melhor experiência de utilização para os stakeholders, facilitando a usabilidade do ORCID, e melhorando a experiência e produtividade com a plataforma.

Requisitos Técnicos:

- Linguagens de programação
 - HTML/CSS
 - JavaScript
- Frameworks:
 - React
 - Bootstrap
- Base de dados:
 - PostgreSQL
- API:
 - ORCID

Requisitos Funcionais:

- Criação e Gerenciamento de Perfil de Pesquisador:
 - O usuário pode criar um perfil único de identificação de pesquisador (com um ORCID iD) e gerenciar informações pessoais, como nome, afiliação institucional e área de atuação.
- Adição de Produção Intelectual:
 - O sistema permite que o usuário adicione e edite suas produções científicas (artigos, capítulos, apresentações, etc.).
- Gerenciamento de Contribuições com Identificadores Padrão:
 - O sistema permite associar identificadores únicos (como DOI, ISBN ou grant IDs) às produções cadastradas, garantindo padronização, rastreabilidade e interoperabilidade com outras plataformas acadêmicas.
- Visualização de Produções Acadêmicas:
 - Agrupar os trabalhos publicados em três categorias: Capítulo de livro, Conference paper e Journal article.
 - Permitir expandir e comprimir essas categorias para melhor navegação.
- Vinculação e Navegação por Perfis:
 - Permitir que ao clicar no nome de um pesquisador, o usuário seja redirecionado para o perfil ORCID daquele pesquisador.
- Gráficos de Evolução:
 - A evolução da quantidade de publicações ao longo do tempo.
 - A evolução da quantidade de publicações associadas a um projeto.
- Associação de Trabalhos a Projetos:
 - Permitir que trabalhos sejam vinculados a projetos específicos.
 - Mostrar quais publicações estão associadas a cada projeto.
- Links Adicionais em Trabalhos:

- Possibilitar a inclusão de múltiplos links dentro de um artigo (ex: link para versão publicada, versão preprint, repositórios etc.).
- Permitir adicionar links para páginas institucionais, como a página pessoal do pesquisador no site do ICMC.

Requisitos Não Funcionais:

- Requisitos de Produto:
 - Navegação intuitiva: A interface deve ser simples e de fácil entendimento para qualquer usuário.
 - Responsividade: O sistema deve funcionar corretamente em diferentes tamanhos de tela (computadores, tablets e celulares).
 - Compatibilidade com navegadores: O sistema deve operar de forma consistente nos principais navegadores (Chrome, Firefox, Safari, Edge).
 - Desempenho: O carregamento e interação com os dados deve ser rápido, com respostas em até 2 segundos.
 - Acessibilidade: A interface deve seguir boas práticas de acessibilidade para permitir uso por pessoas com deficiência.
- Requisitos Organizacionais:
 - Padrões de desenvolvimento: O código deve seguir boas práticas de organização e manutenção, com uso de React, Bootstrap e versionamento com Git.
 - Documentação técnica: Todo o sistema deve ser devidamente documentado.
- Requisitos Externos:
 - Integração com API do ORCID: O sistema deve consumir a API pública do ORCID para ler e atualizar informações.

4. Equipe e infraestrutura

Membros da Equipe:

- Rodrigo de Freitas Lima – Desenvolvimento
- Karl Cruz Altenhofen – Organização e Comunicação
- Maicon Chaves Marques – Design e Experiência do Usuário

Estrutura de Trabalho: A equipe será composta por três pessoas, e as etapas do projeto serão realizadas em conjunto, com desenvolvimento simultâneo, para otimizar o tempo e o fluxo de trabalho.

Infraestrutura:

A infraestrutura é composta por três computadores pessoais dos membros da equipe.

Ferramentas de Desenvolvimento:

- Visual Studio Code: Para edição de código.
- Node.js: Para execução e desenvolvimento de aplicações JavaScript.
- pgAdmin: Para gerenciamento do banco de dados PostgreSQL.
- ChatGPT e Claude: Ferramentas de apoio para obtenção de respostas rápidas e sugestões em problemas técnicos.
- Lovable: Ferramenta para prototipagem e testes de usabilidade.

Materiais de Apoio: Utilização dos PDFs da disciplina disponíveis no Edisciplinas, que servem como referência para a implementação das funcionalidades e melhores práticas do projeto.

5. Acompanhamento do projeto

Será realizada uma reunião semanal com todos os membros do grupo, afim de desenvolver as etapas do modelos cascata.

O acompanhamento será realizado da seguinte forma, a cada etapa do modelo cascata, Definição de requisitos, Projeto de sistema e software, Implementação e teste unitário e Integração e teste do sistema o sistema será mostrado ao cliente, e coletaremos suas opiniões e exigências para o projeto.

6. Cronograma e Marcos do projeto

Cronograma:

- 1) Definição de requisitos
 - 1.1) Documento de requisitos (Entrega: 02/05)
 - 1.2) Diagrama de casos de uso (Entrega: 09/05)
- 2) Projeto de sistema e software
 - 2.1) Diagrama de classes (Entrega: 16/05)
 - 2.2) Diagrama de atividades (Entrega: 23/05)
 - 2.3) Diagrama de sequência (Entrega: 30/05)
- 3) Implementação e teste unitário
 - 3.1) MVP (Entrega: 06/06)
 - 3.2) Resultados dos teste unitários (Entrega: 13/06)
- 4) Integração e teste do sistema
 - 4.1) Resultados do teste de sistema (Entrega: 20/06)
 - 4.2) Entrega do produto final (Entrega: 20/06)

Todos os entregáveis serão colocados no github.com

7. Gerência de Riscos

Riscos:

- Falta de Tempo (risco médio): Pode ser reduzido por uso de cronograma e fazer antecipadamente.
- Problemas com Infraestrutura (risco baixo): Exemplos: queda de internet/energia ou perda de computadores pessoais. Reduzido por realizar antecipadamente e o uso da infraestrutura da USP como backup.
- Incapacidade técnica (risco baixo): Reduzido por existência de conhecimento técnico previamente adquirido. Caso falte capacidade para implementação de uma feature, buscaremos aprender ou utilizar meios alternativos para implementá-la.
- Não atender às especificações dos requisitos (risco médio): Reduzido por constantes verificações do documento de requisitos para garantir que o projeto não falte um requisito ou se desvie.
- Não atender às expectativas do cliente (risco baixo): Reduzido pelas entrevistas cuidadosamente realizadas e com foco em atender as necessidades dos clientes.

8. Testes do produto

Testes funcionais:

- Teste Unitário (automático): Verifica as funcionalidade implementada de forma individual; realizados por meio de uma comparação entre o output esperado e o output dos casos de teste.
- Teste de Integração (automático): Garante que os diferentes módulos do sistema se comuniquem de forma esperada e trabalhem em conjunto corretamente; realizados por meio de uma comparação entre o output esperado e o output dos casos de teste.
- Teste de Sistema (manual): Checa o sistema como um todo; testado pelos membros da equipe, verificando se cada funcionalidade implementada no software funciona como esperado sob o olhar do usuário.

Testes não funcionais:

- Teste de Usabilidade (manual): Avalia se os usuários conseguem navegar e entender as funcionalidades com facilidade, baseado em feedback. Além disso, garante a funcionalidade do sistema em diferentes plataformas (celular, tablet, desktop); entrar no sistema por diversas plataformas utilizando dispositivos pessoais e verificar se a aparência está agradável e intuitiva.
- Teste de Aceitação (manual): Verifica se o sistema, como um todo, atende aos requisitos descritos na especificação e às expectativas do cliente; seguir a jornada do usuário enquanto checa as especificações.

9. Gerenciamento de configuração de software

O gerenciamento de configuração de software será realizado utilizando o GitHub como plataforma de versionamento. Todos os entregáveis do projeto serão disponibilizados no repositório do GitHub, no link <https://github.com/EngSoft2025/orcid-project-clube-de-regatas-do-flamengo>, com as entregas realizadas conforme as datas estabelecidas no cronograma do projeto.

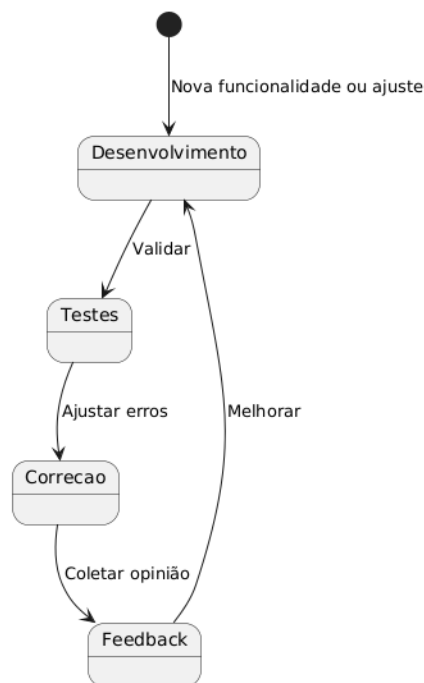
10. Plano de Manutenção de software

1. Desenvolvimento de novas funcionalidades ou alteração em funcionalidade existente: Implementação ou aprimoramento contínuo de funcionalidades com base nas necessidades dos usuários e demandas do sistema.

2. Testes: Realização de testes unitários e de integração para garantir a qualidade e o funcionamento das novas funcionalidades.

3. Correção de erros: Identificação e correção de quaisquer erros ou falhas reportadas durante o uso do sistema.

4. Registro de feedback: Coleta de feedback dos usuários para avaliar a eficácia das mudanças e identificar áreas para melhorias adicionais.



Referências

SOMMERVILLE, Ian. Engenharia de software. 9. ed. São Paulo: Pearson Education do Brasil, 2011. Disponível em: <https://www.facom.ufu.br/~william/Disciplinas%202018-2/BSI-GSI030-EngenhariaSoftware/Livro/engenhariaSoftwareSommerville.pdf>. Acesso em: 24 abr. 2025.

