

Itens de um Plano de Projeto

Histórico da Revisão

Data	Versão	Descrição	Autores
11/04/2025	1.0	Primeira Reunião de Planejamento	Julia Cavallio Orlando Gabriel de Andrade Abreu Isabela Beatriz Sousa Nunes Farias Antônio Carlos de Almeida Micheli Neto Nicolas Carreiro Rodrigues
24/04/2025	1.1	Segunda Reunião de Planejamento	
14/05/2025	1.2	Definimos Requisitos	
20/05/2025	1.3	Revisão do planejamento para alinhamento com o prazo	

1. Identificação

- a. Nome do Projeto: Gaviões da Fiel
- b. Equipe:
 - Julia Cavallio Orlando - 14758721
 - Gabriel de Andrade Abreu - 14571362
 - Isabela Beatriz Sousa Nunes Farias - 13823833
 - Antônio Carlos de Almeida Micheli Neto - 14559013
 - Nicolas Carreiro Rodrigues - 14600801

2. Introdução

Objetivos: Este plano tem como objetivo descrever de forma clara e estruturada os principais elementos que nortearão o desenvolvimento do projeto: escopo, requisitos, cronograma, recursos necessários, riscos e estratégias de validação. Além disso, define os papéis da equipe envolvida e as etapas do processo de construção da solução.

Público alvo: O público-alvo deste documento são os membros da equipe de desenvolvimento, professores e pesquisadores usuários da plataforma ORCID, além de possíveis orientadores e avaliadores acadêmicos interessados no andamento e na organização do projeto.

Propósito: O projeto visa desenvolver uma aplicação complementar ou proposta de melhoria para a plataforma ORCID, com foco na realidade dos pesquisadores brasileiros. Entre os propósitos estão: melhorar a usabilidade, interface mais bonita e intuitiva, fazer uma filtragem dos dados, integração entre pesquisadores, implementar métricas do pesquisador e integração entre pesquisadores (facilitar o vínculo com coautores).

Evolução do plano de projeto: Este plano será atualizado conforme o projeto evoluir, especialmente nas fases de levantamento de requisitos, definição de funcionalidades e testes com usuários. Alterações serão feitas para refletir decisões de design, mudanças de escopo ou melhorias identificadas ao longo do processo. Nesse sentido, faremos reuniões semanais com o intuito de avaliar a progressão do projeto.

Método de desenvolvimento: Adotaremos a metodologia ágil Scrum, com ciclos curtos de desenvolvimento (sprints), reuniões frequentes de acompanhamento e entregas incrementais. Isso permitirá que o projeto evolua de forma contínua, com validações reais ao longo do processo e rápida adaptação às necessidades dos usuários. O foco será deixar o uso da plataforma original mais intuitivo, ampliar a integração com sistemas já utilizados pelos pesquisadores brasileiros e garantir que o resultado final atenda tanto à comunidade acadêmica quanto ao público geral interessado em ciência.

3. Escopo do projeto

O projeto tem como objetivo o desenvolvimento de uma aplicação complementar ou proposta de extensão à plataforma ORCID, com foco na realidade dos pesquisadores brasileiros e na ampliação da comunicação da ciência com a sociedade. A proposta surgiu a partir de entrevistas com professores universitários que utilizam a plataforma e identificaram limitações importantes em sua experiência de uso.

Importância para os stakeholders

Para pesquisadores, especialmente brasileiros, o projeto busca tornar o processo de registro, vínculo e divulgação científica mais fluido, completo e integrado à realidade nacional, como o uso de plataformas internacionais como openAlex. Para a sociedade, a aplicação visa apresentar a produção científica de forma mais acessível e compreensível, promovendo a valorização da ciência. Já para instituições acadêmicas e financeiras, o projeto oferece maior visibilidade e transparência das contribuições dos pesquisadores.

Requisitos Funcionais

1. **Espaço dedicado à divulgação científica**, com linguagem acessível ao público geral.
2. **Organizar os dados**, de modo que o usuário possa filtrar pesquisas por tema, relevância e outras métricas.
3. **Perfil acadêmico com modelo inspirado em redes profissionais**
4. **Integração ampliada com bases de dados internacionais** (ex: ORCID e openAlex).
5. **Organização de perfil com seções bem definidas e explicativas.**

Requisitos Não Funcionais

1. **Interface intuitiva**, tanto para pesquisadores quanto para não especialistas.
2. **Compatibilidade com diferentes dispositivos e navegadores.**
3. **Baixa curva de aprendizagem**, a fim de tornar a nossa aplicação mais intuitiva para o usuário.
4. **Segurança e privacidade**, especialmente no compartilhamento de dados sensíveis.

Requisitos Técnicos

1. Front-end responsivo (web)
2. Manipulação de dados acadêmicos e exportação em formatos padrão (ex: XML).

Fora do Escopo

Sistema próprio de análise de métricas complexas (como índice h ou fator de impacto): Estamos calculando essas métricas baseado nos nossos dados.

4. Equipe e infraestrutura

Equipe

A equipe de desenvolvimento será composta por 5 membros que atuarão em papéis definidos segundo a metodologia ágil **Scrum**:

- **Product Owner (PO):**

Responsável: Julia Cavallio Orlando

Responsável por representar os interesses dos usuários e stakeholders. Define as prioridades do backlog, garantindo que a equipe esteja sempre focada no que entrega maior valor. Também atua como ponte entre os entrevistados e o time técnico.

- **Scrum Master:**

Responsável: Gabriel de Andrade Abreu

Responsável por assegurar que os princípios e práticas do Scrum sejam seguidos. Remove impedimentos, facilita reuniões (como as dailies e reviews) e apoia a equipe a manter seu ritmo de trabalho. Devido ao porte reduzido do time, também participa ativamente do desenvolvimento, contribuindo com implementações, revisões de código e testes para acelerar as entregas.

- **Desenvolvedores (3 membros):**

Responsáveis: Isabela Beatriz Sousa Nunes Farias, Antônio Carlos de Almeida Micheli Neto e Nicolas Carreiro Rodrigues

Responsáveis por projetar, codificar, testar e entregar as funcionalidades do sistema. Trabalharão tanto no front-end quanto no back-end, com foco na responsividade, integração com APIs e experiência do usuário.

Infraestrutura

Para garantir a organização, eficiência e comunicação da equipe, serão utilizadas as seguintes ferramentas e recursos:

Ferramentas de Desenvolvimento

- **VS Code:** Editor de código principal da equipe.
- **Git + GitHub:** Controle de versão e repositório remoto para colaboração no código-fonte.

Gerenciamento de Projeto

- **Google Meet / Presencialmente:** Reuniões semanais de planejamento, revisão e acompanhamento do projeto.
- **Google Drive / Docs:** Documentação compartilhada, controle de versões dos documentos e relatórios.

Outros Recursos

- **Notebook pessoal de cada membro:** Equipamento básico utilizado na implementação e testes.
- **Entrevistas e feedbacks de usuários reais (professores ou colegas de sala):** Recurso essencial para validação contínua do produto.

5. Acompanhamento do projeto

Atividade	Frequência	Responsáveis	Descrição
Sprint Planning (Planejamento da Sprint)	Semanal	Toda a equipe, liderado pelo Scrum Master e PO	Definição das tarefas que serão realizadas durante a semana com base no backlog e prioridades do projeto.
Desenvolvimento assíncrono	Contínuo	Desenvolvedores	Atividades de codificação, testes, revisão de código e documentação, feitas de forma remota e flexível durante a semana.
Sprint Review (Revisão da Sprint)	Semanal	Toda a equipe + Cliente (professores convidados)	Apresentação das funcionalidades desenvolvidas, coleta de feedback e avaliação do progresso.
Sprint Retrospective (Retrospectiva)	Semanal	Toda a equipe	Reflexão sobre o que funcionou bem, o que pode melhorar e ações para a próxima sprint.
Reuniões com stakeholders/clientes	Na semana de testes	Product Owner + Cliente (professores e se não der com colegas que fazem o mesmo trabalho)	Reuniões específicas para validar funcionalidades, ouvir sugestões e alinhar expectativas com os usuários reais.

Acompanhamento do Projeto

- **Pelo time:** O progresso será acompanhado por meio da organização de issues no github e nas atas de reunião. O Scrum Master será responsável por garantir a fluidez das tarefas e identificar impedimentos.
- **Pelo cliente (professores entrevistados ou colegas):** Serão convidados a participar das Sprint Reviews para ver as entregas e dar feedback sobre o projeto.
- **Documentação:** As decisões, reuniões e mudanças serão registradas em um Google Doc compartilhado com todos os membros da equipe.

6. Cronograma e Marcos do projeto

PROPOSTA DE CRONOGRAMA:

Período	Etapa	Atividades Principais	Marco (Milestone)	Entregáveis
Semana 1	Iniciação e Planejamento	Definição do escopo, organização da equipe, cronograma, escolha das ferramentas	Início oficial do projeto	Documento inicial de planejamento
Semana 2	Levantamento de Requisitos e diagrama de caso de uso	Consolidação dos problemas e sugestões, definição de funcionalidades	Requisitos definidos	Documento de requisitos
Semana 3	Design e Arquitetura e desenvolvimento Inicial	Prototipação das interfaces, definição da arquitetura do sistema. Implementação de buscas de pesquisadores e da visualização de um perfil	Funcionalidades iniciais implementadas e protótipo aprovado pelo cliente	Protótipo de média fidelidade
Semana 4	MVP (Produto Mínimo Viável)	Realizar a montagem do front-end inicial, contemplando estrutura, responsividade e telas principais; e implementar funcionalidades básicas da interface, como filtragem de dados, integração entre pesquisadores (facilitando vínculos entre coautores)	Entrega do MVP	MVP navegável com principais funções

		e apresentação de métricas de desempenho.		
Semana 5	Testes e Validações	Testes com usuários, coleta de feedback, ajustes de usabilidade e funcionalidades	Validação com usuários reais	Relatório de feedback + ajustes realizados
Semana 6	Iterações finais	Refino da interface, inclusão de novas integrações, melhorias nas métricas e dados do perfil	Versão beta finalizada	Aplicação quase completa, pronta para entrega
Semana 7	Documentação e Apresentação	Finalização da documentação, gravação de vídeo demonstrativo e preparação da apresentação	Encerramento do projeto	Documentação final, vídeo de demonstração, apresentação

Marcos Importantes (Milestones)

- **Semana 1:** Documento inicial de planejamento
- **Semana 2:** Levantamento de Requisitos
- **Semana 3:** Design e Arquitetura e desenvolvimento Inicial
- **Semana 4:** Entrega do **MVP** com interface básica e fluxo inicial.
- **Semana 5:** Feedback real de usuários (professores entrevistados).
- **Semana 6:** Entrega da versão beta do projeto
- **Semana 7:** Entrega final com documentação e versão funcional da aplicação.

7. Gerência de Riscos

Risco	Causa	Impacto	Estratégia de mitigação
Dependência de APIS externas (Integração)	Abstrair integrações em camadas de serviço, monitorar versões, manter fallback.	Alto	Abstrair integrações em camadas de serviço, monitorar versões, manter fallback.
Crescimento de escopo (Scope Creep)	Roadmap ambicioso vs. 8 semanas de execução	Alto	Backlog priorizado, critérios de aceitação claros, mudanças aprovadas pelo PO.
Capacidade da equipe	Somente 3 devs full-stack	Moderado	Pair programming, documentação mínima.
Qualidade / aceitação do usuário (Usabilidade)	Falta de testes e validações insuficientes	Alto	Sessões de teste na Semana 4, prototipagem contínua, métricas de usabilidade.
Engajamento de stakeholders	Professores podem faltar às Sprint Reviews	Moderado	Agendar com antecedência e chamar colegas para ajudar caso não dê.

8. Testes do produto

A validação do produto será contínua e adaptada à metodologia ágil adotada. Serão aplicados diferentes tipos de teste para garantir qualidade e aderência às necessidades dos usuários:

- **Testes Funcionais:** Verificar se cada funcionalidade implementada cumpre seu propósito.
- **Testes de Usabilidade:** Aplicados com professores e usuários reais para avaliar a experiência e acessibilidade.
- **Testes de Integração:** Verificar a comunicação correta entre os módulos e com APIs externas
- **Testes de Regressão:** Garantir que funcionalidades antigas continuam funcionando após novas implementações.
- **Critérios de Aceitação:** Cada funcionalidade no backlog será acompanhada de critérios claros e objetivos, definidos pelo PO, para que possa ser considerada “pronta” após implementação e testes.

9. Gerenciamento de Configuração de Software

Escopo – Este processo aplica-se a todo o ciclo de vida do repositório:

<https://github.com/EngSoft2025/orcid-project-gavioes-da-fiel>

9.1 Repositório oficial

Item	Valor
URL	https://github.com/EngSoft2025/orcid-project-gavioes-da-fiel
Visibilidade	Pública
Branches persistentes	main, dev

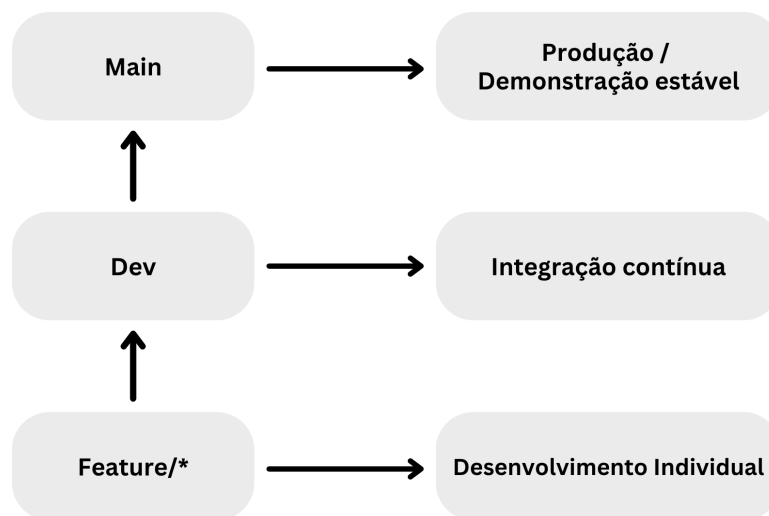
9.2 Convenções de nomenclatura – ramificações e artefatos

Tipo de IC¹	Padrão	Exemplo real
Branch principal	main (produção)	main
Branch de integração	dev	dev
Branches de funcionalidade	features/<autor> ou feature/<id>-<descr>	feature/dashboard

Commit	<frase no imperativo> + detalhes (Closes #<issue>)	adicionar busca por ORCID ID
---------------	---	---------------------------------

¹ IC = Item de Configuração. Cada branch ou tag deve apontar para uma issue ou PR correspondente, garantindo rastreabilidade.

9.3 Fluxo de Branching (compatível com o layout atual)



1. **Branches de funcionalidade** (`feature/<id>-<descr>`) são criadas a partir de `dev`.
2. Concluída a implementação, abre-se PR para `dev`.
3. Após validação e testes, `dev` é mergeado em `main` (sem fast-forward)

9.5 Gerenciamento de Dependências

- **Back-end:** `requirements.txt` (Python)
- **Front-end:** `package.json` (Node)

10. Plano de Manutenção de software

Após a entrega final, será estabelecido um plano de manutenção com foco na evolução contínua do sistema:

- **Manutenção Corretiva:** Correção de bugs reportados por usuários ou identificados em testes.
- **Manutenção Evolutiva:** Inclusão de novas funcionalidades com base em feedback de usuários.
- **Manutenção Preventiva:** Refatoração de código, melhorias de performance e segurança.
- **Manutenção Adaptativa:** Ajustes para compatibilidade com novas versões de navegadores, APIs e bibliotecas.



Papéis:

- **Usuários:** Reportar feedbacks e sugestões via formulário ou e-mail.
- **PO:** Avaliar e priorizar mudanças.
- **Scrum Master:** Garantir fluxo contínuo e Implementar, testar e atualizar o sistema.
- **Desenvolvedores:** Implementar, testar e atualizar o sistema.

Referências

SOMMERVILLE, Ian. Engenharia de software. 9. ed. São Paulo: Pearson Education do Brasil, 2011.

ISOTANI, Seiji. Materiais de aula. São Paulo: Universidade de São Paulo, 2025.