

**INSTITUTO FEDERAL DE RONDÔNIA**  
**CAMPUS PORTO VELHO CALAMA**  
**CURSO SUPERIOR DE TECNOLOGIA EM ANÁLISE E**  
**DESENVOLVIMENTO DE SISTEMAS**

---

**LUIS HENRIQUE BERGONZINI SOUZA**

**FLUX: UMA PROPOSTA DE SISTEMA WEB PARA ACOMPANHAMENTO DE**  
**PROJETOS**

**Porto Velho/RO**  
**2025**

**LUIS HENRIQUE BERGONZINI SOUZA**

**FLUX: UMA PROPOSTA DE SISTEMA WEB PARA ACOMPANHAMENTO DE  
PROJETOS**

Artigo entregue como Trabalho de Conclusão de Curso ao Instituto Federal de Educação, Ciência e Tecnologia de Rondônia (IFRO), *Campus* Porto Velho Calama, como requisito parcial para obtenção do grau de Tecnólogo, junto ao Curso Superior de Tecnologia em Análise e Desenvolvimento de Sistemas.

Orientador: Prof. Leandro Ferrarezi Valiante

**Porto Velho/RO  
2025**

Ficha catalográfica elaborada pelo Sistema Gerador de Ficha Catalográfica do IFRO.

S729f

Souza, Luis Henrique Bergonzini.  
Flux: Uma proposta de sistema web para acompanhamento de projetos. / Luis Henrique Bergonzini Souza. - Porto Velho, 2025.  
22 f. : il.

Orientador(a): Prof. Me. Leandro Ferrarezi Valiante.

Trabalho de Conclusão de Curso (Superior de Tecnologia em Análise e Desenvolvimento de Sistemas) – Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO, Porto Velho, 2025.

1. Gestão de projetos. 2. Administração pública. 3. Sistemas web.  
I. Valiante, Leandro Ferrarezi (orient.). II. Instituto Federal de Educação, Ciência e Tecnologia de Rondônia - IFRO. III. Título.

CDD: 005.13

**Bibliotecário(a) Responsável:** Evandro Silva de Sousa, CRB-11-956

**LUIS HENRIQUE BERGONZINI SOUZA**

**FLUX: UMA PROPOSTA DE SISTEMA WEB PARA ACOMPANHAMENTO DE  
PROJETOS**

A banca examinadora, abaixo listada, aprova o Trabalho de Conclusão de Curso “FLUX: UMA PROPOSTA DE SISTEMA WEB PARA ACOMPANHAMENTO DE PROJETOS” elaborado por “LUIS HENRIQUE BERGONZINI SOUZA” como requisito parcial para obtenção do grau de Tecnólogo em Análise e Desenvolvimento de Sistemas, pelo Instituto Federal de Educação, Ciência e Tecnologia de Rondônia.

Porto Velho/RO, 28/07/2025

**Comissão Examinadora**

---

**Prof. Leandro Ferrarezi Valiante - IFRO**  
(Orientador)

---

**Prof. Fernando Dall’Igna - IFRO**

---

**Prof. Celso Guedes Gomes - IFRO**

## FLUX: UMA PROPOSTA DE SISTEMA WEB PARA ACOMPANHAMENTO DE PROJETOS

**RESUMO:** No cenário atual da gestão pública e privada, o acompanhamento de projetos através de planilhas eletrônicas apresenta limitações críticas, como dificuldades de colaboração, riscos à integridade dos dados e falta de integração com ferramentas analíticas. Diante desses desafios, este trabalho apresenta o desenvolvimento do sistema “Flux”, oferecendo como uma solução de software que substitui processos manuais por uma plataforma integrada de gestão estratégica. Desenvolvido com PHP, CodeIgniter e MySQL, o sistema proporciona controle de acesso diferenciado e gestão segura das informações. A metodologia combinou análise documental com abordagem ágil, incluindo modelagem relacional e desenvolvimento iterativo validado com usuários. Destaques incluem visualização hierárquica de projetos, atualização automática de status e mecanismos de aprovação *workflow*. A implementação do novo sistema eliminou efetivamente os riscos de modificações indevidas do modelo anterior. A interface intuitiva, com seus recursos de tabelas dinâmicas e múltiplos filtros, resultou em uma maior facilidade de utilização pelos usuários e em um gerenciamento mais eficiente das alterações por parte dos administradores.

**PALAVRAS-CHAVE:** gestão de projetos. administração pública. sistemas web.

**ABSTRACT:** In the current scenario of public and private management, project monitoring through spreadsheets presents critical limitations such as collaboration difficulties, data integrity risks, and lack of integration with analytical tools. Facing these challenges, this work presents the development of the “Flux” system as a software solution that replaces manual processes with an integrated strategic management platform. Developed with PHP, CodeIgniter and MySQL, the system provides differentiated access control and secure information management. The methodology combined document analysis with an agile approach, including relational modeling and iterative development validated with users. Key features include hierarchical project visualization, automatic status updates, and workflow approval mechanisms. The implementation of the new system effectively eliminated the risks of improper modifications present in the previous model. The intuitive interface, with its dynamic tables and multiple filters, resulted in greater ease of use for end users and more efficient change management for administrators.

**KEYWORDS:** project management. public administration. web systems.

## 1 INTRODUÇÃO

No cenário atual da gestão de projetos, a eficiência no monitoramento e acompanhamento de atividades é fundamental para o sucesso das iniciativas estratégicas de qualquer organização. Conforme destacado por Kerzner (2025), a falta de ferramentas adequadas pode levar a atrasos, retrabalhos e falhas na comunicação, comprometendo os resultados esperados. Esse desafio foi identificado no Núcleo de Monitoramento da Coordenadoria de Gestão Estratégica (COGE) da Superintendência Estadual de Tecnologia da Informação e Comunicação (SETIC), onde o acompanhamento de projetos era realizado por meio de planilhas no *Google Sheets*. Embora essa solução permitisse um nível básico de organização, estudos como os de Dinsmore e Cabannis-Brewin (2014) apontam que o uso excessivo de planilhas em processos complexos de gestão pode introduzir erros, dificultar a colaboração e limitar a escalabilidade.

A utilização de planilhas eletrônicas, apesar de sua ampla adoção, apresenta limitações intrínsecas quando aplicada a processos dinâmicos de gestão de projetos. Como observado por Few (2006), planilhas com estruturas muito personalizadas, como mesclagem de células e formatações visuais complexas, tornam-se inadequadas para integração com ferramentas de *Business Intelligence* (BI), exigindo manipulação manual dos dados. Essa foi uma das principais dificuldades enfrentadas pela equipe, já que os relatórios no *Power BI* demandavam uma reorganização constante das informações, consumindo tempo e aumentando o risco de inconsistências. Além disso, a ausência de um controle de acesso robusto, conforme discutido por Stair e Reynolds (2011), gerava preocupações quanto à segurança e integridade das informações.

Diante desses desafios, a literatura em gestão de projetos e sistemas de informação reforça a importância de soluções integradas. Segundo Pressman e Maxim (2021), sistemas web desenvolvidos sob medida para necessidades específicas podem otimizar processos, reduzir retrabalhos e melhorar a qualidade dos dados. Foi nesse contexto que surgiu o projeto “Flux”, uma iniciativa alinhada às recomendações de Sommerville (2011) sobre a importância de sistemas customizados para resolver problemas operacionais críticos. O objetivo foi substituir o processo manual por uma plataforma centralizada, garantindo não apenas a atualização ágil das informações, mas também sua pronta disponibilidade para análise.

Além dos desafios operacionais já mencionados, a escolha pelo desenvolvimento de um sistema próprio, em vez da aquisição de soluções comerciais prontas, justifica-se por vários fatores críticos. Em primeiro lugar, os *softwares* disponíveis no mercado, como *Microsoft Project*, *Jira* ou *Asana*, frequentemente exigem licenças custosas, o que pode representar uma dificuldade para órgãos públicos como a SETIC, conforme discutido por Whittington et al. (2019) em sua análise sobre viabilidade de

soluções tecnológicas no setor público. Além disso, esses sistemas muitas vezes possuem módulos excessivamente complexos, que demandam um longo período de adaptação dos usuários, como observado por Nielsen (1994) em seus estudos sobre usabilidade em software empresarial. Essa complexidade desnecessária pode dificultar a adoção pela equipe, especialmente quando apenas funcionalidades específicas são requeridas.

Outro aspecto relevante é a escalabilidade: um sistema customizado como o “Flux” permite a incorporação progressiva de demandas futuras, como a inclusão de evidências de entregas, indicadores personalizados ou integração com outros sistemas internos, algo que soluções genéricas não oferecem com a mesma flexibilidade, conforme destacado por Fowler (2012) em seu trabalho sobre arquitetura de software adaptável.

Por fim, o controle total sobre o banco de dados facilita não apenas a otimização do armazenamento e consultas, mas também a geração de relatórios e *dashboards* no BI sem as limitações impostas por formatos proprietários, alinhando-se às recomendações de Kimball e Ross (2013) para projetos de *data warehousing* ágeis. Dessa forma, o desenvolvimento de uma solução sob medida mostrou-se não apenas viável, mas estratégico, atendendo às necessidades imediatas e futuras de organizações com custo e complexidade reduzidos.

## 1.1 Objetivos

### 1.1.1 Objetivo geral

Desenvolver um sistema *web* centralizado para gestão de projetos que substitua o processo manual baseado em planilhas eletrônicas.

### 1.1.2 Objetivos específicos

- (i) Modelar a estrutura do banco de dados do sistema;
- (ii) Elaborar um esquema de permissões para controle de acesso ao sistema;
- (iii) Desenvolver interfaces com filtros dinâmicos para manipulação do sistema.

## 2 REFERENCIAL TEÓRICO

Neste capítulo serão apresentados os fundamentos teóricos que embasam as escolhas tecnológicas realizadas para o desenvolvimento do sistema “Flux”. Por meio

de uma revisão da literatura especializada, foram analisadas as características, vantagens e usos práticos de cada tecnologia escolhida, justificando sua seleção para atender aos requisitos específicos do projeto.

Como linguagem de programação *backend*, foi utilizado o PHP (*Hypertext Preprocessor*), uma das tecnologias mais consolidadas e amplamente adotadas no desenvolvimento web. Criada originalmente por Rasmus Lerdorf em 1994, a linguagem evoluiu significativamente ao longo dos anos, incorporando características de linguagens modernas enquanto mantinha sua essência focada na *web*. A versão 8.1 foi selecionada por oferecer melhorias significativas de performance, incluindo a implementação do compilador JIT (*Just-In-Time*), cuja eficácia foi comprovada por Popov (2020), demonstrando aceleração de até 3 vezes na execução de determinados tipos de código.

A documentação oficial do PHP PHP Group (2020) destaca que a versão 8 introduziu um sistema de tipos mais rigoroso, com suporte a tipos de união e o operador *nullsafe*, recursos que contribuem para a escrita de código mais seguro e menos propenso a erros. A maturidade do ecossistema PHP, com mais de 100 mil pacotes disponíveis através do gerenciador de dependências *Composer*, foi outro fator determinante, permitindo a integração rápida de bibliotecas especializadas para diversas finalidades.

Para o armazenamento e gerenciamento de dados, foi adotado o *MySQL*, um sistema gerenciador de banco de dados (SGBD) relacional *open-source* que se destaca por sua confiabilidade, performance e facilidade de uso. Conforme evidenciado na documentação oficial MySQL (2023), este sistema gerenciador de banco de dados apresenta características especialmente adequadas para aplicações web de médio porte, oferecendo um equilíbrio ideal entre recursos avançados e simplicidade de operação.

O desenvolvimento web moderno divide-se em duas camadas principais: o *front-end*, responsável pela interface e interação com o usuário, e o *back-end*, que processa a lógica de negócios e gerencia os dados. Enquanto o *front-end* utiliza tecnologias como HTML, CSS e JavaScript para criar interfaces dinâmicas e responsivas, o *back-end* emprega linguagens como PHP para processamento de requisições, acesso a bancos de dados e implementação da lógica do sistema. Essa separação de responsabilidades, conforme destacado por Fielding (2000), permite maior escalabilidade, manutenibilidade e especialização de cada componente.

A interface do usuário foi construída utilizando as tecnologias fundamentais do desenvolvimento web *front-end*: HTML5, JavaScript e o *framework* Bootstrap 5. O HTML5, em sua versão mais recente, foi empregado seguindo as melhores práticas de semântica e acessibilidade definidas pelo consórcio W3C World Wide Web Consortium (2023). Como observa Pilgrim (2010), os novos elementos semânticos in-

trozidos pelo *HTML5*, como `<header>`, `<nav>`, `<section>` e `<article>`, não apenas melhoram a organização estrutural do código, mas também potencializam a acessibilidade e o SEO (*Search Engine Optimization*, ou Otimização para Mecanismos de Busca). Para o estilo visual e layout responsivo, a adoção do *framework* Bootstrap 5 se mostrou estratégica, pois conforme sua documentação oficial Bootstrap Team (2021), oferece um sistema de *grid* avançado, componentes pré-estilizados e utilitários de design que aceleram o desenvolvimento enquanto mantêm consistência visual em diferentes dispositivos. A metodologia de componentes do Bootstrap foi combinada com a abordagem BEM (*Block, Element, Modifier*) para CSS, seguindo as diretrizes propostas por Yandex (2013), resultando em um código *front-end* organizado. A interatividade foi implementada com *JavaScript* puro, utilizando o padrão AJAX conforme descrito originalmente por Garrett et al. (2005), permitindo atualizações assíncronas da interface sem necessidade de recarregamento completo da página, com o intuito de melhorar a experiência do usuário.

Como *framework back-end*, foi selecionado o CodeIgniter 4, uma evolução do tradicional *framework* PHP que mantém a filosofia de simplicidade e alto desempenho. De acordo com sua documentação oficial (CodeIgniter Foundation (2022)), o CodeIgniter se destaca por ser rápido e leve porque não exige muitos recursos do servidor, demonstrando eficiência notável em operações CRUD (*Create, Read, Update, Delete*, ou Criar, Ler, Atualizar, Excluir) básicas - um fator relevante para um sistema que precisa responder rapidamente a múltiplas requisições concorrentes. A arquitetura MVC (*Model-View-Controller*, ou Modelo-Visão-Controle) foi implementada seguindo os padrões estabelecidos por Fowler (2012), promovendo uma clara separação de preocupações que facilita a manutenção e evolução do sistema.

Para o sistema de autenticação e segurança, a integração do módulo “SHIELD” incorporou as melhores práticas contemporâneas de segurança descritas pela OWASP OWASP Foundation (2023), incluindo autenticação multi-fator e proteção contra ataques CSRF (*Cross-Site Request Forgery*, ou Falsificação de Solicitação Entre Sites) e XSS (*Cross-Site Scripting*, ou Script Entre Sites), alinhando-se também com as recomendações de Sandhu (1998) sobre segurança em aplicações web. O controle de acesso baseado em funções (RBAC - *Role-Based Access Control*, ou Controle de Acesso Baseado em Funções) seguiu o modelo proposto por Sandhu (1998), garantindo que cada usuário tenha exatamente os privilégios necessários para suas atividades.

### 3 METODOLOGIA

Este capítulo descreve o processo de desenvolvimento do sistema de monitoramento de projetos, desde a identificação das necessidades até a implementação da solução. A metodologia adotada foi baseada em um ciclo iterativo, envolvendo levantamento de requisitos, escolha de tecnologias, desenvolvimento, testes e implantação, sempre com foco em resolver os problemas identificados no uso da planilha original.

Esta pesquisa, segundo a tipologia proposta por Wazlawick (2020), classifica-se como primária em sua natureza, por gerar conhecimento aplicado através do desenvolvimento do sistema Flux como solução inovadora para os desafios de gestão de projetos. Quanto aos objetivos, configura-se como pesquisa de design, pois seu propósito central é projetar uma alternativa tecnológica que supere as limitações dos processos manuais baseados em planilhas, materializando “como as coisas poderiam ser” na prática organizacional.

Metodologicamente, neste trabalho adota-se uma abordagem híbrida: inicia-se com pesquisa documental para análise crítica dos artefatos existentes (planilhas, relatórios e fluxos de trabalho), seguida de pesquisa-ação durante a implementação do sistema, onde o pesquisador atua ativamente no ambiente real para iterar e validar a solução. Essa combinação, conforme Wazlawick (2020), é típica de estudos em Computação que buscam equilibrar rigor teórico e aplicabilidade prática, garantindo que o *software* desenvolvido (Flux) responda tanto às demandas imediatas quanto a critérios acadêmicos de validação.

Antes do desenvolvimento, foi realizada uma revisão da literatura sobre sistemas de gestão de projetos e as limitações de planilhas eletrônicas. Conforme destacado por Reschenhofer e Matthes (2015), sistemas baseados em planilhas frequentemente apresentam problemas de integridade de dados e falta de controle de acesso, tornando-se inadequados para gestão de projetos em médio e grande porte. Essa fundamentação confirmou a necessidade de migrar para uma solução que não promovesse esse tipo de problemas.

Como ambiente de investigação inicial, foi utilizada uma demanda real do núcleo de monitoramento de projetos da Coordenadoria de Gestão Estratégica (COGE), pertencente à SETIC. O acompanhamento era realizado por meio de uma planilha eletrônica que, apesar de funcional, apresentava limitações como controle de acesso insuficiente, falta de centralização para relatórios (BI) e dificuldades na atualização colaborativa dos dados.

Para entender melhor essas necessidades, foram realizadas reuniões com a equipe responsável pelo monitoramento, nas quais foram discutidos os principais pontos de melhoria. Além disso, foi realizada uma análise detalhada da estrutura da plani-

---

lha existente, mapeando seus campos, fluxos de trabalho e gargalos. Com base nesse levantamento, definiu-se que o sistema precisaria:

- a) Controlar acessos diferenciados (administradores x usuários comuns);
- b) Facilitar a visualização e filtragem dos projetos em tabelas dinâmicas;
- c) Permitir solicitações de atualização de forma organizada e rastreável; e
- d) Garantir integridade dos dados, evitando inconsistências como as que ocorreriam na planilha.

O desenvolvimento seguiu um fluxo iterativo, com protótipos sendo validados pela equipe de monitoramento antes da implementação final. Inicialmente, foram criadas as telas básicas do sistema, reproduzindo a organização da planilha, mas com melhorias na usabilidade, como filtros dinâmicos e ordenação de colunas.

Ao longo do processo, foram realizados testes contínuos, tanto pela equipe de desenvolvimento quanto por usuários-chave, para identificar e corrigir possíveis falhas. Esses testes incluíram:

- a) Validação de permissões (garantindo que usuários não-admin não acessassem funções restritas);
- b) Testes de carga (verificando o comportamento do sistema com múltiplos acessos simultâneos);
- c) Checagem de integridade de dados (assegurando que as atualizações não corrompessem informações existentes).

Após a fase de testes, o sistema foi disponibilizado em produção, substituindo gradualmente a planilha antiga. Nos primeiros dias de uso, foi realizado um acompanhamento próximo para resolver eventuais problemas durante a fase de cadastro dos projetos. Essas informações foram utilizadas para ajustes finais, como a inclusão de novos campos de dados que não haviam sido previstos inicialmente, modificações para melhor usabilidade e correção de problemas encontrados.

#### **4 DESENVOLVIMENTO**

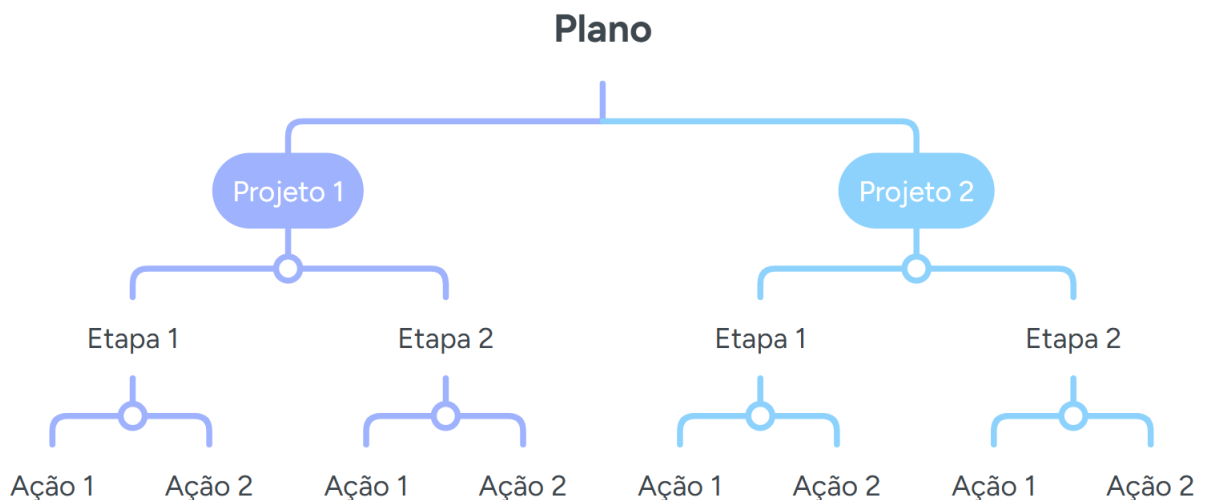
O desenvolvimento do sistema de monitoramento de projetos partiu de uma análise profunda da estrutura organizacional já consolidada na planilha eletrônica anteriormente utilizada pela equipe. A solução foi construída considerando a hierarquia existente de Planos > Projetos > Etapas > Ações, mas com significativas melhorias na forma de visualização, manipulação e controle dos dados. O processo de desenvolvimento seguiu uma abordagem iterativa, com constantes validações junto aos usuá-

rios finais para garantir que o sistema atendesse plenamente às necessidades operacionais.

#### 4.1 Estrutura Hierárquica e Organização dos Dados

A base do sistema manteve a estrutura hierárquica já conhecida pelos usuários, composta por quatro níveis principais, conforme demonstrado na Figura 1. No nível mais abrangente estão os “Planos”, que representam conjuntos estratégicos contendo diversos “Projetos” relacionados. Cada projeto, por sua vez, é dividido em “Etapas” que representam fases de execução, sendo estas compostas por “Ações” específicas que correspondem às tarefas operacionais. Essa organização permitiu uma transição suave dos dados da planilha para o sistema, minimizando a curva de aprendizado dos usuários. A principal melhoria neste aspecto foi a implementação de mecanismos que garantem a integridade referencial entre os níveis, eliminando problemas comuns nas planilhas como inconsistências nos relacionamentos entre os dados.

Figura 1 – Estrutura Hierárquica



Fonte: O próprio autor.

#### 4.2 Página de Visão Geral

A primeira interface desenvolvida foi a página de “Visão Geral”, conforme ilustrado na Figura 2, criada para oferecer uma visão unificada de todo o ecossistema de projetos. Diferentemente da planilha original que exigia constantes rolagens e alternância entre abas, esta página apresenta todas as “Ações” do sistema em uma única tabela, com informações contextuais sobre a Etapa, Projeto e Plano a que cada uma pertence. A implementação utilizou o *plugin* DataTables, que além da paginação automática, permite ordenação por qualquer coluna e filtros rápidos. Um dos diferenciais

desta página são os múltiplos filtros dinâmicos que permitem aos usuários visualizar os dados a partir de diferentes perspectivas, como filtrar todas as ações de um determinado projeto ou visualizar apenas as ações com status “Em atraso”.

Figura 2 – Página Visão Geral

Visão Geral

Plano: Todos | Projeto: Todos | Etapa: Todos | Ação: Todas

Responsáveis: Digite o responsável | Status: Todos | Priorização: Todos | Período: dd/mm/aaaa

Limpar | Filtrar

Visão Geral dos Projetos | Configurar Campos

10 resultados por página

Priorização	Plano	Projeto	Etapa	Ação	Responsáveis	Entrega Estimada	Data Inicial	Data Final	Status
☆	Plano de Transformação Digital	ADA		Acompanhar a construir planilha do domínio	Isabella Carvalho	-	-	-	Não iniciado
☆	Plano de Transformação Digital	ADA		Correções do resultado da versão teste		16/07/2025	26/06/2025	-	Atrasado

Fonte: O próprio autor.

### 4.3 Páginas de Visualização Detalhada

Complementando a “Visão Geral”, o sistema oferece uma abordagem de navegação vertical através das páginas de “Visualização Detalhada” conforme ilustrado na Figura 3. Esta funcionalidade foi desenvolvida para permitir uma imersão progressiva nos diferentes níveis hierárquicos, mantendo sempre o contexto da navegação. Ao acessar um “Plano” específico, por exemplo, o usuário pode visualizar todos os projetos vinculados, e ao selecionar um “Projeto”, é direcionado para a listagem de suas etapas.

Figura 3 – Página Visão Detalhada

The screenshot shows a web interface for 'Planos'. At the top, there is a breadcrumb 'Visão Detalhada / Planos'. Below it, the title 'Planos' is displayed. The main content area contains a search form with two input fields: 'Nome' (containing 'Filtrar por nome') and 'Sigla' (containing 'Filtrar por sigla'). To the right of these fields are two buttons: 'Limpar' (with a trash icon) and 'Filtrar' (with a funnel icon). Below the search form is a section titled 'Lista de Planos'. It includes a dropdown menu for 'Mostrar' set to '10' and the text 'registros por página'. Below this is a table with the following data:

Nome	Sigla	Descrição	Progresso	Opções
Plano de Transformação Digital	PTD	Define ações para modernizar os serviços públicos por meio da digitalização, promovendo maior eficiência, transparência e acessibilidade. No contexto da SETIC, o PTD tem como objetivo impulsionar a inovação no Governo do Estado de Rondônia, garantindo a oferta de serviços digitais centrados no cidadão, integrados, seguros e alinhados às diretrizes de governo digital.	35%	
Plano Diretor de Tecnologia da Informação	PDTIC	Instrumento estratégico que orienta o planejamento, a gestão e os investimentos em tecnologia da informação e comunicação no âmbito da administração pública, visando as ações tecnológicas da superintendência ao Planejamento Estratégico do Governo do Estado de Rondônia, promovendo a oferta de serviços digitais modernos, seguros, integrados, adaptáveis e inclusivos, posicionando-se como referência em soluções digitais de governo no estado.	22%	

At the bottom of the table, it says 'Mostrando 1 a 2 de 2 registros'. To the right, there are navigation buttons: 'Anterior', '1' (highlighted), and 'Próxima'.

Fonte: O próprio autor.

A implementação técnica utilizou rotas dinâmicas no CodeIgniter, como `/planos/{id}/projetos`, garantindo uma URL semântica e de fácil compreensão. Cada nível possui sua própria interface de CRUD (*Create, Read, Update, Delete*, ou Criar, Ler, Atualizar, Excluir), mas com a particularidade de preservar o relacionamento hierárquico. Quando um usuário adiciona uma nova etapa, por exemplo, o sistema automaticamente a associa ao projeto de origem da navegação, eliminando os erros de vinculação que eram comuns na planilha.

#### 4.4 Controle de Acessos e Fluxo de Aprovação

Um dos avanços mais significativos em relação à planilha original foi a implementação de um sistema de controle de acessos e fluxo de aprovação. O módulo "SHIELD" do CodeIgniter foi integrado para gerenciar a autenticação e autorização dos usuários, que são divididos em dois grupos principais: "Administradores" e "Usuários Comuns". Os "Administradores" possuem acesso completo a todas as funcionalidades do sistema, podendo criar, editar e excluir itens em qualquer nível hierárquico diretamente. Já os "Usuários Comuns" podem visualizar todos os dados, mas as alterações são realizadas através de um fluxo formal de solicitações que requer aprovação. Esta abordagem resolveu um dos principais problemas da planilha, onde qualquer usuário com acesso poderia fazer modificações sem qualquer tipo de controle ou registro.

## 4.5 Módulo de Solicitações

O módulo de solicitações foi desenvolvido para gerenciar o ciclo de vida das alterações propostas por usuários não administradores. Na visão do solicitante, foi implementada a página “Minhas Solicitações”, demonstrada a seguir na Figura 4, que apresenta um histórico completo com o *status* (Pendente, Aprovada ou Rejeitada) das solicitações, data de envio e, quando aplicável, justificativas para a decisão tomada.

Figura 4 – Página de Minhas Solicitações

Fonte: O próprio autor.

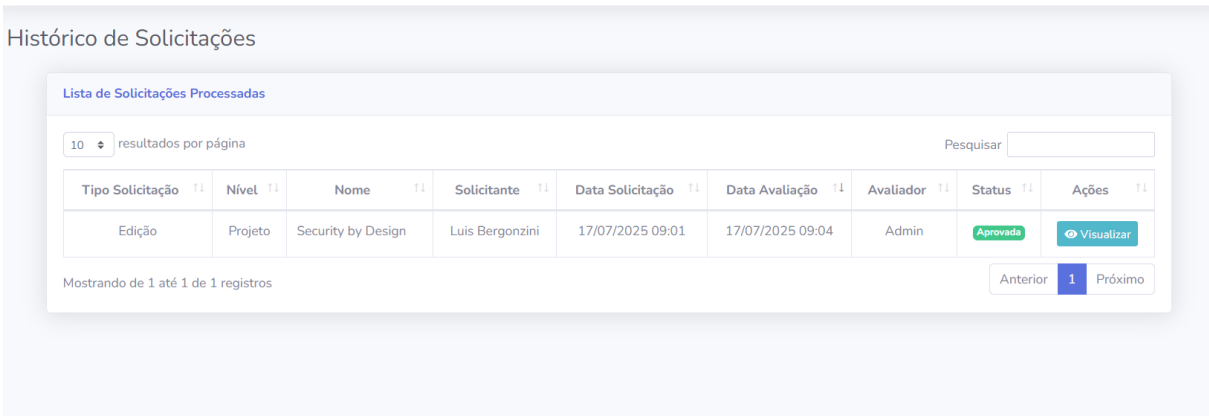
Para os administradores, foi criada a página de “Solicitações Pendentes”, que concentra todas as alterações aguardando análise onde os administradores poderão avaliar as solicitações, conforme ilustrado na Figura 5.

Figura 5 – Página de Solicitações

Fonte: O próprio autor.

Além disso, foi criada também a interface de “Histórico de Solicitações”, com registro completo de todas as avaliações já realizadas pelos administradores, permitindo visualizar os detalhes de cada visualização, conforme demonstrado na Figura 6 a seguir.

Figura 6 – Histórico de Solicitações



The screenshot shows a web interface titled "Histórico de Solicitações". At the top right, there is a user profile for "Admin" with a notification bell icon. Below the title, there is a section "Lista de Solicitações Processadas". This section includes a dropdown menu set to "10" for "resultados por página" and a search box labeled "Pesquisar". Below this is a table with the following data:

Tipo Solicitação	Nível	Nome	Solicitante	Data Solicitação	Data Avaliação	Avaliador	Status	Ações
Edição	Projeto	Security by Design	Luis Bergonzini	17/07/2025 09:01	17/07/2025 09:04	Admin	Aprovada	Visualizar

At the bottom of the table area, it says "Mostrando de 1 até 1 de 1 registros". To the right of this, there are navigation buttons: "Anterior", "1" (highlighted), and "Próximo".

Fonte: O próprio autor.

O processo de avaliação foi cuidadosamente projetado para apresentar ao administrador uma comparação lado a lado entre os dados atuais do sistema e as alterações propostas, facilitando a tomada de decisão. Esta funcionalidade foi implementada através de um modal dinâmico que carrega os dados via AJAX (*Asynchronous JavaScript and XML*, ou JavaScript e XML Assíncronos), apresentando as diferenças de forma clara e intuitiva.

#### 4.6 Módulo de Gerenciamento de Usuários

Para completar a estrutura de controle de acessos, foi desenvolvido um módulo exclusivo para administradores gerenciarem os usuários do sistema conforme ilustrado na Figura 7 a seguir.

Esta funcionalidade resolveu uma limitação crítica da planilha original, que não permitia nenhum tipo de gestão de acessos individualizada. A implementação consiste em uma página que lista todos os usuários cadastrados, com informações básicas como nome, e-mail e grupo de permissão (admin ou usuário comum). A interface inclui um sistema de filtros que permite buscar usuários por nome ou e-mail, além de um seletor para filtrar por grupo de permissão. Para cada usuário listado, os administradores podem editar suas informações, alterar seu grupo de permissão ou removê-lo do sistema quando necessário. A exclusão de usuários segue um processo seguro com confirmação em duas etapas para prevenir remoções acidentais. Todo o módulo foi

Figura 7 – Página Gerenciar Usuários

Gerenciamento de Usuários

Nome  Grupo  Data de Criação

Lista de Usuários

Exibir  resultados por página

ID	Nome	Email	Grupo	Status	Data de Criação	Ações
1	Luis Bergonzini	luishenriquebergsouza@outlook.com	User	Ativo	02/05/2025 04:48	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
2	Isabella Carvalho	isabellactlemess@gmail.com	User	Ativo	07/05/2025 11:08	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
3	Ediane Egert	edianegalvao@setic.ro.gov.br	User	Ativo	07/05/2025 11:12	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
4	Teste	usuarioteste@teste.com	User	Ativo	07/05/2025 18:01	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>
5	Admin	usuarioadmin@teste.com	Admin	Ativo	07/05/2025 18:02	<input type="button" value="Editar"/> <input type="button" value="Excluir"/>

Mostrando de 1 até 5 de 5 registros

Fonte: O próprio autor.

desenvolvido com validações para garantir a integridade dos dados, como verificação de e-mails únicos e impedimento de remoção do último administrador do sistema.

#### 4.7 Funcionalidades Avançadas de Monitoramento

Para aprimorar a capacidade de acompanhamento dos projetos, foram implementadas funcionalidades avançadas de monitoramento em todos os níveis hierárquicos. No nível de “Planos”, “Projetos” e “Etapas”, foi adicionada uma coluna de progresso que calcula automaticamente a porcentagem de “Ações” concluídas em relação ao total. Este cálculo é realizado através de uma função no *back-end* que consulta o banco de dados em tempo real, garantindo que a informação esteja sempre atualizada. A visualização do progresso foi enriquecida com uma barra gráfica colorida (verde para itens com bom progresso, amarelo para médio e vermelho para atrasados) acompanhada do percentual numérico, seguindo princípios de visualização de dados para facilitar a interpretação rápida como indicado na Figura 8 a seguir.

Uma melhoria fundamental no sistema foi a automatização do campo de status, que superou a limitação da planilha original que dependia de atualizações manuais propensas a erros. O mecanismo implementado calcula dinamicamente o *status* com base em três parâmetros temporais essenciais: (i) a data estimada para entrega, definida no planejamento inicial; (ii) a data de início real, informada pelo usuário quando a ação efetivamente começa; e (iii) a data de conclusão real, registrada quando a ação é finalizada. Esta lógica opera através de um sistema de verificação em duas cama-

Figura 8 – Barra gráfica de progresso

Eixo	Responsáveis	Data Fim	Progresso	Opções
PLANEJAMENTO ESTRATÉGICO E GOVERNANÇA (PEG)	Nenhum responsável	-	44%	
PLANEJAMENTO ESTRATÉGICO E GOVERNANÇA (PEG)	Nenhum responsável	-	57%	
PLANEJAMENTO ESTRATÉGICO E GOVERNANÇA (PEG)	Nenhum responsável	29/05/2025	100%	8 de 8 ações finalizadas

Fonte: O próprio autor.

das: uma atualização imediata, quando qualquer uma dessas datas é modificada pelo usuário, complementada por uma verificação periódica (a cada 5 minutos) durante o acesso às páginas relevantes do sistema.

O sistema define quatro estados possíveis: “Não iniciado” (quando a data de início real não foi registrada e a data atual é anterior à estimada), “Em andamento” (quando há data de início real mas não de conclusão), “Finalizado” (quando a data de fim real está preenchida) e “Atrasado” (quando a data atual ultrapassa a data estimada de entrega sem registro de data de conclusão real). Cada *status* é representado visualmente por cores e ícones distintivos (vermelho para atrasos, verde para concluído, azul para não iniciados e cinza para paralisados), seguindo princípios do design de interfaces. Esta solução eliminou as inconsistências crônicas do modelo anterior, onde a dependência de atualizações manuais frequentemente resultava em status desatualizados que comprometiam a qualidade do monitoramento.

Estas implementações aprimoraram o processo monitoramento, permitindo que os gestores identifiquem rapidamente pontos de atenção e tomem decisões baseadas em dados atualizados. A atualização dinâmica de *status* criou um sistema de alerta

precoce para potenciais atrasos, algo que era extremamente trabalhoso de manter manualmente na planilha original. A experiência do usuário foi cuidadosamente considerada em cada etapa, garantindo que as informações mais críticas sejam imediatamente visíveis, enquanto detalhes adicionais permanecem acessíveis através de interações simples.

## 5 CONCLUSÃO

O desenvolvimento do sistema “Flux” representou uma solução que contorna os desafios de monitoramento de projetos, substituindo com vantagens o uso de planilhas eletrônicas que, apesar de amplamente utilizadas, apresentam limitações significativas em termos de controle de acesso, integridade dos dados e colaboração entre equipes, conforme já alertavam Kerzner (2025) e Dinsmore e Cabannis-Brewin (2014). A implementação desta plataforma demonstrou na prática como é possível transformar um processo manual e propenso a erros em um sistema organizado, seguro e eficiente, cumprindo assim o objetivo central deste trabalho. A escolha por tecnologias como PHP 8.1, CodeIgniter 4 e MySQL mostrou-se acertada, permitindo criar uma solução que incorpora desde os princípios de usabilidade defendidos por Nielsen (1994), até as melhores práticas de arquitetura de software adaptável sugeridas por Fowler (2012).

Os resultados alcançados superaram as expectativas iniciais em vários aspectos. O sistema de controle de acesso baseado em papéis eliminou os riscos de modificações indevidas que existiam no modelo anterior, enquanto a interface intuitiva com tabelas dinâmicas e múltiplos filtros favoreceu uma maior produtividade para os usuários. A estrutura relacional garantiu a integridade dos dados, resolvendo um dos principais problemas do modelo baseado em planilhas.

Para o futuro, o sistema “Flux” possui um grande potencial de evolução, com ao menos duas implementações especialmente promissoras:

- A possibilidade de integração com sistemas de autenticação de terceiros, de forma a trazer benefícios importantes como login único e maior segurança no gerenciamento de acessos, seguindo rigorosamente os padrões estabelecidos pela LGPD e pelas normas de TI governamentais;
- A criação de um módulo avançado de indicadores estratégicos, que permitirá o acompanhamento detalhado do alinhamento entre os projetos e os objetivos institucionais, com painéis analíticos flexíveis e personalizáveis.

Esta experiência demonstrou a viabilidade e os benefícios de soluções customizadas no setor público, especialmente quando existem necessidades específicas

que justificam o desenvolvimento próprio em vez da aquisição de sistemas comerciais genéricos, como bem argumentou Whittington et al. (2019). O “Flux” não apenas apresenta uma alternativa para resolução de problemas imediatos de gestão de projetos, mas estabelece uma base tecnológica que permitirá a incorporação progressiva de novas funcionalidades, acompanhando assim a evolução das demandas da SETIC e demonstrando como a tecnologia, quando bem aplicada, pode ser um aliado para a modernização da administração pública.

## REFERÊNCIAS

- BOOTSTRAP TEAM. **Bootstrap 5 Documentation**. 2021. Disponível em: <https://getbootstrap.com/docs/5.0/getting-started/introduction/>. Acesso em: 18 jun. 2025.
- CODEIGNITER FOUNDATION. **CodeIgniter 4 User Guide**. 2022. Disponível em: [https://codeigniter.com/user\\_guide/index.html](https://codeigniter.com/user_guide/index.html). Acesso em: 1 jul. 2025.
- DINSMORE, Paul C; CABANNIS-BREWIN, Jeannette. **AMA-Manual de Gerenciamento de Projetos 2ed**. Brasport, 2014.
- FEW, Stephen. **Information dashboard design: The effective visual communication of data**. O'Reilly Media, Inc., 2006.
- FIELDING, Roy Thomas. **Architectural styles and the design of network-based software architectures**. University of California, Irvine, 2000.
- FOWLER, Martin. **Patterns of enterprise application architecture**. Addison-Wesley, 2012.
- GARRETT, Jesse James et al. **Ajax: A new approach to web applications**. San Francisco, CA, USA, 2005.
- KERZNER, Harold. **Project management: a systems approach to planning, scheduling, and controlling**. John Wiley & Sons, 2025.
- KIMBALL, Ralph; ROSS, Margy. **The data warehouse toolkit: The definitive guide to dimensional modeling**. John Wiley & Sons, 2013.
- MYSQL. **MySQL ACID Compliance Documentation**. 2023. Acesso em: 5 jun. 2025.
- NIELSEN, Jakob. **Usability engineering**. Morgan Kaufmann, 1994.
- OWASP FOUNDATION. **OWASP Top Ten**. 2023. Disponível em: <https://owasp.org/www-project-top-ten/>. Acesso em: 8 jul. 2025.
- PHP GROUP. **PHP 8.0 Release Announcement**. 2020. Disponível em: <https://www.php.net/releases/8.0/en.php>. Acesso em: 30 maio 2025.
- PILGRIM, Mark. **HTML5: Up and Running**. O'Reilly Media, 2010. ISBN 978-0-596-80602-6. Disponível em: <https://www.oreilly.com/library/view/html5-up-and/9781449399339/>.
- POPOV, Nikita. **PHP RFC: JIT**. 2020. Disponível em: <https://wiki.php.net/rfc/jit>. Acesso em: 22 maio 2025.
- PRESSMAN, Roger S; MAXIM, Bruce R. **Engenharia de software-9**. McGraw Hill Brasil, 2021.

RESCHENHOFER, Thomas; MATTHES, Florian. **An empirical study on spreadsheet shortcomings from an information systems perspective**. Springer International Publishing, 2015. P. 50–61.

SANDHU, Ravi S. **Role-based access control**. 1998.

SOMMERVILLE, Ian. **Software engineering (ed.)** America: Pearson Education Inc, 2011.

STAIR, Ralph; REYNOLDS, George. **Principles of Information Systems**. Melbourne, Australia: Cengage Learning, 2011.

WAZLAWICK, Raul Sidnei. **Metodologia de Pesquisa para Ciência da Computação**. 3. ed. Rio de Janeiro: Editora Campus, 2020.

WHITTINGTON, Richard et al. **Exploring strategy**. Pearson UK, 2019.

WORLD WIDE WEB CONSORTIUM. **HTML5 Recommendation**. 2023. Disponível em: <https://www.w3.org/TR/html52/>. Acesso em: 12 jun. 2025.

YANDEX. **BEM Methodology**. 2013. Disponível em: <https://en.bem.info/methodology/>. Acesso em: 25 jun. 2025.