



UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI

Departamento de Computação - Faculdade de Ciências Exatas

Bacharelado em Sistemas de Informação

Raul Victor da Silva

MODERNIZAÇÃO E PADRONIZAÇÃO DIGITAL:

Proposta de um novo site para o Departamento de Computação da UFVJM

alinhado ao *Design System 3* do Governo Federal

Diamantina, MG

2025

Raul Victor da Silva

MODERNIZAÇÃO E PADRONIZAÇÃO DIGITAL:
Proposta de um novo site para o Departamento de Computação da UFVJM
alinhado ao Design System 3 do Governo Federal

Monografia apresentada ao curso de graduação em Sistemas de Informação da Universidade Federal dos Vales do Jequitinhonha e Mucuri, como parte dos requisitos exigidos para a obtenção do título de Bacharel em Sistemas de Informação.

Orientador: Prof. Alessandro Vivas Andrade

Diamantina, MG

2025



**MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI**

FOLHA DE APROVAÇÃO

Raul Victor da Silva,

**MODERNIZAÇÃO E PADRONIZAÇÃO DIGITAL: PROPOSTA DE UM NOVO SITE
PARA O DEPARTAMENTO DE COMPUTAÇÃO DA UFVJM ALINHADO AO DESIGN
SYSTEM 3 DO GOVERNO FEDERAL**

Trabalho de Conclusão de Curso apresentado
ao Curso de Sistemas de Informação da
Universidade Federal dos Vales do
Jequitinhonha e Mucuri, como requisito
parcial para conclusão do curso.

Orientador: Prof. Dr. Alessandro Vivas Andrade

Aprovado em 19 de novembro de 2025

BANCA EXAMINADORA

Prof. Dr. Alessandro Vivas Andrade

Faculdade de Ciências Exatas - DECOM - UFVJM

Profa. Dra. Luciana Pereira de Assis

Faculdade de Ciências Exatas - DECOM - UFVJM

Profa. Dra. Cinthya Rocha Tameirão

Faculdade de Ciências Exatas - DECOM - UFVJM



Documento assinado eletronicamente por **Luciana Pereira de Assis, Servidor(a)**, em 19/11/2025, às 15:59, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Alessandro Vivas Andrade, Servidor(a)**, em 24/11/2025, às 15:55, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



Documento assinado eletronicamente por **Cinthya Rocha Tameirão, Servidor(a)**, em 24/11/2025, às 16:34, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do [Decreto nº 8.539, de 8 de outubro de 2015](#).



A autenticidade deste documento pode ser conferida no site https://sei.ufvjm.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1945284** e o código CRC **C9A1E526**.

AGRADECIMENTOS

Chegar até aqui foi um processo longo, repleto de descobertas, desafios e amadurecimento. Durante os últimos 5 anos tive dúvidas, mudei caminhos, reencontrei propósitos e, acima de tudo, aprendi — sobre tecnologia, sobre pessoas e sobre mim mesmo.

Este trabalho de conclusão de curso representa não apenas o encerramento de um ciclo acadêmico, mas também um espelho de todo meu crescimento pessoal e profissional, fruto de experiências compartilhadas com pessoas que estimo profundamente.

Em primeiro lugar, expresso minha descomedida e eterna gratidão à minha mãe, Maria Euza, pelo amor incondicional, apoio e força que sempre me guiaram e sustentaram durante todos os momentos nessa caminhada. Agradeço, entre tantas coisas, por ter tido perseverança quando eu estava à beira da desistência, por ter compartilhado das minhas felicidades mesmo sem entendê-las, e por estar sempre comigo não importa o desafio.

Agradeço aos meus tios e tias, que também desempenharam papéis importantes ao oferecerem apoio nos momentos em que precisei. Aos amigos de longa data e aos que tive o prazer de conhecer ao longo do curso, minha gratidão pela companhia constante, pelos conselhos, pelas risadas e pelo apoio sincero ao longo da jornada. Um agradecimento especial ao Henrique, Erica, Túlio e Gabriel, que marcaram profundamente minha trajetória. A todos, o meu muito obrigado.

Minha sincera gratidão ao professor Alessandro Vivas, pela orientação atenta, pela generosidade no compartilhamento de conhecimento e pela confiança depositada neste trabalho. Também agradeço a todos os professores do curso pela contribuição em minha formação.

Expresso minha gratidão aos colegas e amigos da Dicom pelo companheirismo, pelas trocas enriquecedoras e, especialmente, à Helen, Larissa, Tiago e Estêvão, pelos nossos momentos de desespero e aprendizado.

Finalmente, agradeço a todas as pessoas que influenciaram e impulsionaram minha trajetória até a conclusão deste trabalho.

RESUMO

O site atual do Departamento de Computação (Decom) da Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM) encontra-se desalinhado das diretrizes da versão mais recente do Padrão Digital de Governo (DSGov), comprometendo a padronização, a experiência do usuário e a acessibilidade. Este trabalho propôs o desenvolvimento de um Mínimo Produto Viável (MVP) de um novo site para o Decom, com o objetivo central de modernizar sua presença digital, alinhando-a ao DSGov 3 e utilizando tecnologias modernas que garantem performance, flexibilidade e manutenibilidade. Para isso, foi adotada a metodologia D[AD]I Readaptada, que guiou a implementação de uma arquitetura desacoplada (*headless*). A solução técnica utiliza o CMS Wagtail (baseado em Django) no *backend* e um *frontend* em Next.js (baseado em React). O desenvolvimento superou desafios técnicos, como a incompatibilidade de componentes do DSGov com a renderização no lado do servidor (SSR), solucionada com o uso de *wrappers* e componentes *skeleton*. Os resultados validaram o sucesso da proposta. A aderência ao DSGov 3 foi confirmada, e a validação técnica com o Google Lighthouse (*desktop*), após otimizações, demonstrou pontuações de 98% em Performance, 94% em Acessibilidade e 100% em Boas Práticas. Notavelmente, o Cumulative Layout Shift (CLS) foi mitigado para 0.076. O feedback dos stakeholders foi positivo, validando a nova arquitetura da informação. O trabalho conclui que os objetivos foram alcançados, entregando uma solução robusta, embora limitada ao escopo de um MVP. Trabalhos futuros incluem a implantação do site em produção e a migração total de conteúdo, treinamento da equipe e extensão do sistema.

Palavras-chave: Padrão Digital de Governo, DSGov, Design System, Wagtail, Next.js, Arquitetura Desacoplada, Modernização de Sites, UFVJM.

ABSTRACT

The current website of the Computing Department (Decom) at the Federal University of Vales do Jequitinhonha e Mucuri (UFVJM) is misaligned with the guidelines of the latest version of the Digital Government Standard (DSGov), which compromises standardization, user experience, and accessibility. This work proposed the development of a Minimum Viable Product (MVP) for a new Decom website, with the central objective of modernizing its digital presence by aligning it with DSGov 3 and utilizing modern technologies to ensure performance, flexibility, and maintainability. To achieve this, a Readapted D[AD]I methodology was adopted, guiding the implementation of a decoupled (headless) architecture. The technical solution employs the Wagtail CMS (based on Django) for the backend and a Next.js (based on React) frontend. The development process overcame technical challenges, such as the incompatibility of DSGov components with server-side rendering (SSR), which was resolved through the use of wrappers and skeleton components. The results validated the proposal's success. Adherence to DSGov 3 was confirmed, and technical validation using Google Lighthouse (desktop), following optimizations, demonstrated scores of 98% in Performance, 94% in Accessibility, and 100% in Best Practices. Notably, the Cumulative Layout Shift (CLS) was mitigated to 0.076. Stakeholder feedback was positive, validating the new information architecture. This work concludes that its objectives were met, delivering a robust solution, albeit limited to the scope of an MVP. Future work includes the website deployment to production and the content total migration, team training and extension of the system.

Keywords: Digital Government Standard, DSGov, Design System, Wagtail, Next.js, Decoupled Architecture, Site modernization, UFVJM.

LISTA DE ABREVIATURAS E SIGLAS

API: *Application Programming Interface*

ARIA: *Accessible Rich Internet Applications*

CLS: *Cumulative Layout Shift*

CMS: *Content Management System*

CSS: *Cascading Style Sheets*

DADI: Definição, Arquitetura, Desenho e Implementação

Decom: Departamento de Computação

Dicom: Diretoria de Comunicação Social

DOM: *Document Object Model*

DSGov: *Design System* do Governo Federal

e-MAG: Modelo de Acessibilidade em Governo Eletrônico

e-PWG: Padrões Web em Governo Eletrônico

Facet: Faculdade de Ciências Exatas

HTML: *Hypertext Markup Language*

HTTP: *Hypertext Transfer Protocol*

IDG: Identidade Digital de Governo

JSON: *JavaScript Object Notation*

LCP: *Largest Contentful Paint*

MTV: *Model-Template-View*

MVP: *Minimum Viable Product*

ORM: *Object-Relational Mapper*

PDG: Padrão Digital de Governo

RUP: *Rational Unified Process*

SEO: *Search Engine Optimization*

SSR: *Server Side Rendering*

TBT: *Total Blocking Time*

TCC: Trabalho de Conclusão de Curso

UFVJM: Universidade Federal dos Vales do Jequitinhonha e Mucuri

URL: *Uniform Resource Locator*

UX: *User Experience*

WCAG: *Web Content Accessibility Guidelines*

WSGI: *Web Server Gateway Interface*

LISTA DE ILUSTRAÇÕES

Figura 1. Site desenvolvido utilizando o DSGov 4.....	16
Figura 2. Diagrama geral da metodologia D[AD]I.....	20
Figura 3. Nova estrutura do site do Decom.....	29
Figura 4. Proposta de capa para o site do Decom.....	30
Figura 5. Diagrama da arquitetura geral do sistema.....	32
Figura 6. Diagrama da arquitetura do backend.....	33
Figura 7. Diagrama da arquitetura geral de um app.....	34
Figura 8. Diagrama da arquitetura geral do frontend.....	34
Figura 9. Diagrama da Camada de Roteamento.....	35
Figura 10. Fluxo da requisição no frontend.....	36
Figura 11. Edição de página com StreamField no Wagtail.....	37
Figura 12. Representação em JSON de um bloco de parágrafo.....	37
Figura 13. Componente de carrossel.....	41
Figura 14. Parâmetros do componente de carrossel.....	41
Figura 15. Componente de listagem genérico.....	42
Figura 16. Parâmetros do componente de listagem genérico.....	42
Figura 17. Componente de Acesso Rápido.....	43
Figura 18. Parâmetros do componente de acesso rápido.....	44
Figura 19. Exemplo de componente de skeleton.....	45
Figura 20. Edição da TextPage com TinyMCE.....	47
Figura 21. Resultados do Google Lighthouse para o MVP.....	50
Figura 22. Pontuações das métricas de performance para o MVP.....	51
Figura 23. Página inicial do MVP em mobile.....	52
Figura 24. Resultados do Google Lighthouse para o Portal Institucional da UFVJM.....	52
Figura 25. Pontuações das métricas de performance para o Portal Institucional da UFVJM...	53

LISTA DE QUADROS

Quadro 1. Atividades da etapa de definição da D[AD]I.....	15
Quadro 2. Atividades da etapa de arquitetura e desenho da D[AD]I.....	15
Quadro 3. Atividades da etapa de implementação da D[AD]I.....	16
Quadro 4. Atividades da etapa de definição da D[AD]I Readaptada.....	19
Quadro 5. Atividades da etapa de arquitetura e desenho da D[AD]I Readaptada.....	20
Quadro 6. Definição da equipe e dos stakeholders.....	21
Quadro 7. Definição de público-alvo por ordem de relevância.....	22

SUMÁRIO

1 INTRODUÇÃO.....	10
1.1 Problema e proposta do trabalho.....	11
1.2 Objetivos.....	11
1.2.1 Objetivos específicos.....	11
1.2.2 Delimitações.....	12
1.3 Justificativa.....	12
2 REFERENCIAL TEÓRICO.....	12
2.1 Design Systems.....	13
2.2 Design de experiência do usuário.....	14
2.3 Padrão Digital de Governo.....	15
2.4 O site do Decom.....	17
2.5 Wagtail e Next.js.....	17
2.6 A metodologia D[AD]I.....	19
2.6.1 Etapa de definição.....	20
2.6.2 Etapa de arquitetura e desenho.....	21
2.6.3 Etapa de implementação.....	22
2.7 Trabalhos correlatos.....	22
3 METODOLOGIA.....	23
3.1 Seleção metodológica.....	24
3.2 D[AD]I Readaptada.....	24
3.2.1 Etapa de definição.....	25
3.2.2 Etapa de arquitetura e desenho.....	25
3.2.3 Etapa de implementação.....	26
4 DESENVOLVIMENTO.....	27
4.1 Fase de Planejamento.....	27
4.1.1 Definição.....	27
4.1.2 Arquitetura e desenho.....	29
4.2 Fase de Desenvolvimento.....	31

4.2.1 Materiais.....	31
4.2.2 Arquitetura do sistema.....	31
4.2.2.1 Arquitetura geral.....	32
4.2.2.2 Arquitetura do backend.....	33
4.2.2.3 Arquitetura do frontend.....	34
4.2.3 Construção de conteúdo dinâmico.....	36
4.2.4 Integração e complementação do DSGov 3.....	38
4.2.5 Implementação de recursos adicionais.....	45
4.2.6 Arquitetura de implantação.....	48
5 RESULTADOS E VALIDAÇÃO.....	49
5.1 Validação de aderência ao DSGov 3.....	49
5.2 Validação técnica.....	50
5.3 Validação metodológica e feedback dos Stakeholders.....	53
6 CONCLUSÃO.....	54
REFERÊNCIAS.....	56
APÊNDICES.....	59

1 INTRODUÇÃO

Com a consolidação da Internet como um importante meio de acesso à informação e à prestação de serviços, tornou-se indispensável que instituições públicas possuam uma presença digital sólida, eficiente e inclusiva. Segundo a Pesquisa Brasileira de Mídias e a Pesquisa Telefônica Regular Rodada 27, realizadas pela Secretaria de Comunicação Social da Presidência da República (Secom-PR), cerca de 49% dos entrevistados preferem a Internet como fonte primária ou secundária de informações e cerca de 43% dos entrevistados utilizam portais e páginas na internet como fonte de informações (Secom-PR, 2016; Secom-PR, 2022; Alves, 2017).

Em especial, os portais institucionais assumem um papel estratégico como canais oficiais de comunicação, prestação de contas e interação com a sociedade. Para cumprirem adequadamente essa função, tais plataformas devem atender a critérios essenciais de usabilidade, acessibilidade, coerência visual e eficiência técnica de modo a garantir que todas as pessoas, independentemente de suas condições ou limitações, possam usufruir de seus conteúdos e serviços de forma plena e segura (Brasil, 2025).

Nesse cenário, é fundamental que os ambientes digitais de instituições públicas sejam construídos com base em padrões modernos de design e desenvolvimento, capazes de promover uma experiência consistente, agradável e intuitiva aos usuários, além de assegurar conformidade com legislações específicas sobre acessibilidade digital e comunicação governamental. A adoção de diretrizes unificadas também permite consolidar a identidade visual do governo, reduzir custos com desenvolvimento e manutenção e promover maior confiabilidade e transparência nas interfaces públicas (Kholmatova, 2017; Garret, 2011).

Com o objetivo de padronizar e modernizar a experiência digital do cidadão, o Governo Federal Brasileiro tem investido, nos últimos anos, em políticas de transformação digital. Destacam-se, entre essas iniciativas, a criação do Padrão Digital de Governo (PDG) e a proposta do *Design System* do Governo Federal (DSGov), que estabelecem um conjunto de orientações técnicas, componentes reutilizáveis e princípios de design centrado no usuário para a construção de interfaces públicas digitais (Brasil, 2025). Essas diretrizes têm sido adotadas progressivamente por órgãos e instituições federais, promovendo maior coesão visual e funcionalidade nos serviços digitais oferecidos à população.

No âmbito da Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM), o portal institucional constitui-se como um dos principais meios de comunicação da universidade com a comunidade acadêmica e o público externo. Em especial, o Departamento

de Computação (Decom) da Faculdade de Ciências Exatas (Facet) da UFVJM possui um site em Wordpress de grande relevância para a comunidade que, atualmente, não está alinhado ao DSGov e roda sobre uma infraestrutura própria do departamento de forma independente do restante do portal institucional.

1.1 Problema e proposta do trabalho

Apesar da importância do site do Decom para a comunidade, sua estrutura e identidade visual ainda não estão plenamente em acordo com as diretrizes da última versão lançada do DSGov. Essa defasagem compromete não apenas a padronização com os demais órgãos públicos, mas também com outros sites da própria universidade que já seguem o padrão, mesmo que em uma versão antiga. Além disso, o site atual do departamento falta em recursos como buscas e filtros que facilitam a navegação e a busca por conteúdo. Aspectos fundamentais como a experiência do usuário e a acessibilidade se encontram em desacordo com o PDG.

Dessa forma, este trabalho tem como objetivo a proposta de um mínimo produto viável (MVP) de um novo site para o departamento, ajustado às diretrizes do DSGov 3 em termos de interface e conteúdo, e fundamentado na utilização de práticas e tecnologias modernas, escaláveis e seguras.

1.2 Objetivos

Diante do contexto de modernização digital e das diretrizes estabelecidas pelo Governo Federal para os portais públicos, este trabalho tem por objetivo desenvolver um MVP de um novo site para o departamento utilizando ferramentas modernas e implementando o Design System 3 do Governo Federal. Assim, melhorando a presença digital do Decom/UFVJM, por meio da aplicação prática de tecnologias atualizadas e dos padrões de design recomendados.

1.2.1 Objetivos específicos

- Estudar a estrutura e os princípios do DSGov 3 e do Padrão Digital de Governo;
- Compreender a arquitetura do Wagtail e sua integração com um *frontend* React desacoplado;
- Configurar um ambiente de desenvolvimento compatível com o Wagtail e React;

- Implementar o design proposto com base nos componentes do DSGov 3, adaptando-os à identidade do Decom/UFVJM;
- Validar a solução implementada por meio da análise de métricas de desempenho (*Web Vitals*) e da verificação da otimização técnica do MVP, considerando critérios de eficiência, responsividade e compatibilidade entre dispositivos.

1.2.2 Delimitações

Este trabalho se delimita ao escopo dos esforços necessários para o desenvolvimento e implantação do MVP do novo site para o Decom. Dadas as limitações de tempo decorrente da duração do semestre letivo e das disciplinas de Projeto Orientado I e II, não fazem parte do escopo deste trabalho, a integração com todos os sistemas institucionais da UFVJM, tais como sistemas acadêmicos ou administrativos, nem o desenvolvimento de funcionalidades além daquelas necessárias para estruturar, implantar e manter um *site* institucional moderno, acessível e aderente às diretrizes do Governo Federal.

1.3 Justificativa

O *site* do Decom/UFVJM representa um canal estratégico de comunicação institucional, acadêmica e científica. É por meio dele que a comunidade interna e externa tem acesso a informações essenciais sobre o curso de Bacharelado em Sistemas de Informação, projetos, eventos, produção científica e ações do departamento. Nesse sentido, sua atualização e adequação aos padrões atuais de acessibilidade e usabilidade é fundamental para garantir uma presença digital condizente com a relevância acadêmica do Decom.

A versão atualmente disponível não segue as diretrizes do DSGov, criado com o objetivo de padronizar, modernizar e tornar mais acessíveis as interfaces digitais de órgãos e instituições públicas. Ao adotar o DSGov 3, versão mais recente oficialmente lançada e consolidada, este trabalho contribui para ajustar o site do Decom às práticas recomendadas pelo Governo Federal.

Além disso, a escolha por tecnologias atuais, com ampla utilização e suporte, garante não apenas robustez e escalabilidade, mas também facilidade de manutenção e evolução futura do sistema. O desenvolvimento do novo site do Decom reforça o caráter aplicado do projeto, ampliando seu impacto para além do âmbito teórico, ao mesmo tempo em que demonstra a viabilidade prática da adoção do DSGov 3 em contextos universitários, servindo também como referência para novos projetos, ou como base para novas implementações.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta os fundamentos teóricos que sustentam o trabalho desenvolvido, abordando os principais conceitos relacionados a *design systems*, experiência do usuário (UX), desenvolvimento Web e sistemas de gerenciamento de conteúdo (CMS). Também são discutidas as diretrizes do Padrão Digital de Governo (PDG), iniciativa do Governo do Brasil voltada à padronização da identidade visual e da experiência digital nos serviços públicos on-line. São brevemente contextualizados o portal institucional da UFVJM e o estado atual do site do Decom. Por fim, são apresentadas outras definições chave para este trabalho incluindo conceitos relacionados à *development stack* (isto é, o conjunto de ferramentas, linguagens, *frameworks*, e tecnologias utilizadas no desenvolvimento da aplicação).

2.1 Design Systems

Conforme destacado por Kholmatova (2017), não há um conceito padronizado para o termo “*design system*”, o que possibilita sua aplicação em diferentes contextos. No entanto, segundo a autora, um *design system* pode ser compreendido como um conjunto de padrões e práticas estruturados, com o objetivo de orientar o desenvolvimento de um determinado produto (Kholmatova, 2017). De acordo com o Serviço Brasileiro de Apoio às Micro e Pequenas Empresas (Sebrae, 2022), um *design system* é um recurso composto por um conjunto padronizado de regras, elementos e princípios que promovem a consistência do design e facilitam a comunicação entre equipes multidisciplinares. Além disso, esse sistema orienta o desenvolvimento das interfaces de uma organização, garantindo uniformidade e eficiência no processo de criação.

De acordo com Frost (2016), os guias de estilo são documentos que guardam a definição dos materiais de design, fornecendo diretrizes de usabilidade, orientações e restrições, e representam um dos elementos centrais de um *design system*. Kholmatova (2017) corrobora essa visão ao destacar que os guias de estilo se concentram principalmente em aspectos como cores, tipografia e demais elementos visuais. Além disso, os guias de estilo abrangem diversos elementos essenciais para a padronização, como grades de design (*design grids*), paletas de cores, tipografia e diretrizes para o uso de logomarcas, além de especificações para seções e elementos individuais da interface (Garret, 2011).

Frost (2016) enfatiza que existem diferentes tipos de guias de estilo que interagem entre si e englobam múltiplas documentações sobre aspectos diferentes, sendo os voltados

especificamente para a Web apenas uma parte de um ecossistema mais amplo, além disso, também cita que dentre os benefícios trazidos por esses guias estão a promoção da coesão e consistência entre interfaces, a construção de uma linguagem comum entre equipes e a consolidação das decisões de design adotadas.

Ainda segundo Frost (2016), os guias de estilo compõem os *design systems*, mas se distinguem deles em escopo: enquanto os guias oferecem a documentação dos padrões, os *design systems* integram essa documentação com os componentes funcionais e demais mecanismos que permitem a sua operacionalização.

Dessa forma, no contexto deste trabalho, um *design system* pode ser conceituado como um conjunto de diretrizes, elementos, princípios e componentes funcionais que são aplicados na criação de interfaces consistentes e eficientes. Entre sua documentação essencial, destacam-se os guias de estilo, que estabelecem a documentação sobre padrões visuais e funcionais a serem seguidos.

2.2 Design de experiência do usuário

Além de estabelecer diretrizes visuais e funcionais, a eficiência de um *design system* também deve ser considerada. Ao abordar esse aspecto, Kholmatova (2017) avalia o quanto as diretrizes em um *design system* contribuem para os objetivos do produto. Segundo a autora, um *design system* eficiente é aquele que equilibra custo-benefício no design, ao mesmo tempo em que garante uma experiência satisfatória para o usuário, em concordância com a finalidade do produto (Kholmatova, 2017).

A experiência do usuário (*User Experience – UX*) está diretamente ligada à aplicação de um *design system* e à sua eficiência, pois influencia a forma como os usuários interagem com as interfaces digitais. Conforme Lynch (2016), a experiência do usuário é um conceito abrangente que envolve ações e tomadas de decisão que impactam a forma como os indivíduos interagem com um site. O autor destaca que uma boa UX possibilita o acesso e o uso de recursos que agregam valor aos usuários, sendo um fator essencial para o sucesso de uma empresa.

Garrett (2011) complementa essa abordagem ao afirmar que a experiência do usuário vai além dos aspectos estéticos e funcionais isolados, inserindo-os em um contexto real de uso. O autor destaca que, em produtos simples, como cadeiras e mesas, os requisitos para uma boa experiência estão embutidos na própria definição do produto. Em contrapartida, produtos

mais complexos exigem uma definição separada entre o que é o produto e qual é a experiência que ele deve proporcionar, conforme o seguinte trecho:

But the manufacturers of chairs and tables tend not to employ user experience designers. In these simple cases, the requirements to deliver a successful user experience are built into the definition of the product itself: In some sense, a chair you can't sit on isn't a chair at all. With more complex products, though, the requirements to deliver a successful user experience are independent of the definition of the product (GARRET, 2011, p. 9).

Dessa forma, entende-se que a experiência do usuário se apresenta como base na estruturação de um *design system*, sendo um dos principais direcionadores de suas diretrizes e componentes. Um *design system* bem elaborado não apenas define aspectos visuais e técnicos, mas também traduz princípios de UX em padrões reutilizáveis que orientam decisões de design e desenvolvimento. Isso garante que a experiência proporcionada ao usuário final seja consistente, intuitiva e concordante aos objetivos do produto.

2.3 Padrão Digital de Governo

No contexto governamental, o artigo 1º da Portaria nº 7.508, de 22 de novembro de 2022, do Ministério das Comunicações/Secretaria Especial de Comunicação Social, dispõe que o Padrão Digital de Governo (PDG) define as diretrizes de *layout* para o portal gov.br e suas atualizações (Brasil, 2022).

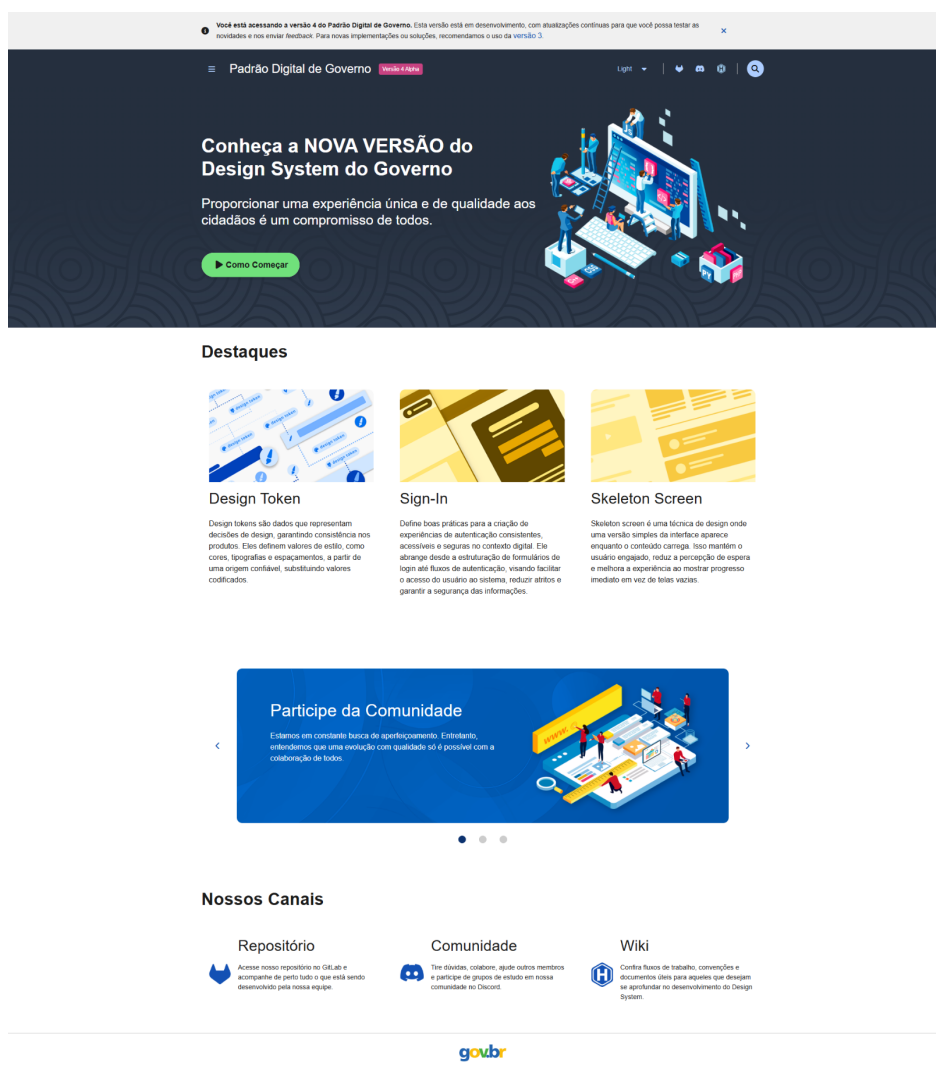
Segundo o Governo do Brasil (2025), o PDG corresponde ao *design system* oficial do governo federal, estabelecendo padrões de interface que devem ser seguidos por *designers* e desenvolvedores na construção de produtos digitais governamentais. O PDG surge da necessidade de padronizar a experiência do usuário ao acessar serviços e produtos digitais do governo, promovendo maior usabilidade, acessibilidade e eficiência.

Com a adoção desse padrão, busca-se reduzir as dificuldades de navegação entre diferentes sistemas governamentais, garantindo uma identidade visual unificada e uma interação mais intuitiva para os cidadãos (Brasil, 2025). Esse conjunto de diretrizes revela a conformidade do PDG às boas práticas de UX, colocando a experiência do usuário como um elemento central na formulação das interfaces públicas.

O PDG representa a evolução da antiga Identidade Digital de Governo (IDG) e baseia-se no *Design System* do Governo Brasileiro (DSGov), uma iniciativa voltada à unificação da experiência do usuário no acesso aos serviços e produtos digitais do governo federal. Como *design system*, o PDG é descrito pelo Governo do Brasil (2025) como um

documento dinâmico, sujeito a grandes revisões e que atua como referência técnica e visual para o desenvolvimento de soluções digitais governamentais. Atualmente, a versão mais recente do DSGov é a 4.0¹, conforme a **Figura 1**, que ainda se encontra em fase *alpha* de desenvolvimento, o que significa que está em constante aprimoramento e ainda não está completa.

Figura 1. Site desenvolvido utilizando o DSGov 4



Fonte: SERPRO (2025)

Por meio da disponibilização de uma base unificada de componentes reutilizáveis, documentação técnica, diretrizes e exemplos de implementação, o PDG facilita o trabalho de equipes de design e desenvolvimento ao passo em que promove a interoperabilidade e a manutenção mais simples dos sistemas.

¹ **SERPRO.** Padrão Digital de Governo. Acesso em: 27 de março de 2025. Disponível em: <https://next-ds.estaleiro.serpro.gov.br/home>

Um dos pontos fundamentais do PDG é a acessibilidade digital, que segue o Modelo de Acessibilidade em Governo Eletrônico (e-MAG) e os padrões internacionais do WCAG (*Web Content Accessibility Guidelines*), garantindo que os serviços públicos estejam disponíveis a todos os cidadãos, inclusive àqueles portadores de deficiência (Brasil, 2025).

No entanto, por ainda estar em fase *alpha*, a versão 4.0 do DSGov apresenta a necessidade de atualizações constantes e a possível ocorrência de mudanças durante o desenvolvimento, o que pode impactar diretamente a adoção por parte de instituições públicas. Por esse motivo, o DSGov 3.6.2, última versão estável oficialmente lançada, ainda é recomendada para novas implementações e representa o estado da arte (oficialmente lançado) quanto a esse esforço.

2.4 O site do Decom

Em 2017, foi proposta uma nova estruturação do portal institucional da Universidade Federal dos Vales do Jequitinhonha e Mucuri (UFVJM) com a implementação da IDG como forma de atender à legislação vigente naquele tempo (Alves, 2017). O portal, sob responsabilidade da Diretoria de Comunicação Social (Dicom/UFVJM), é um dos principais meios de comunicação oficiais da universidade e desempenha um importante papel na disseminação de informações institucionais.

Apesar de sua importância, o projeto Novo Portal proposto por Alves (2017) ainda está em curso de execução e grande parte dos sites da universidade ainda não fazem parte da nova infraestrutura, alguns também não são gerenciados pela Dicom e tampouco atendem ao IDG ou DSGov.

O site atual do Decom constitui um exemplo de site ainda não incorporado ao projeto Novo Portal. Apesar disso, ele disponibiliza uma variedade de páginas e documentos relacionados à vida acadêmica e administrativa, incluindo a divulgação de trabalhos de conclusão de curso, notícias, orientações para estágios, grupos de pesquisa e demais documentações institucionais (UFVJM, 2025).

O site exerce papel relevante tanto para a comunidade interna quanto para o público externo, funcionando como uma vitrine digital do departamento e oferecendo conteúdos exclusivos, como trabalhos de conclusão de curso já publicados. Atualmente, a plataforma é desenvolvida em Wordpress e hospedada na infraestrutura própria do departamento.

2.5 Wagtail e Next.js

Segundo a Oracle (2025), um sistema de gerenciamento de conteúdo (CMS) é uma plataforma que oferece suporte à criação, edição, organização e publicação de conteúdo digital, além de incluir recursos de controle de acesso, permitindo que até mesmo equipes pequenas e não técnicas possam gerenciar o conteúdo de sites com autonomia. Em sua forma desacoplada, o CMS concentra-se na gestão e estruturação do conteúdo, enquanto o *frontend* pode ser desenvolvido de maneira independente, adaptando-se a diferentes ambientes e dispositivos. Essa arquitetura proporciona maior flexibilidade, escalabilidade e modernização das interfaces, além de facilitar integrações com outras tecnologias e serviços.

No projeto do Novo Portal da UFVJM, optou-se pela utilização do Plone como CMS (Alves, 2017). O Plone é um sistema de código aberto robusto e seguro, baseado no servidor de aplicações Zope, também *open source* (Aspeli, 2011), amplamente utilizado por instituições públicas e educacionais. Entretanto, apesar de sua consolidação, a versão mais recente do Plone (6.1.1) (Plone, 2025), acompanhada do Volto como interface *frontend*, não dispõe de uma implementação oficial compatível com o DSGov em sua versão 3, o que impõe desafios para sua adoção em novos projetos que demandam alinhamento imediato a esse padrão.

Diante desse contexto, surge como alternativa o Wagtail, CMS moderno construído sobre o Django, um *framework open source* em Python para desenvolvimento de aplicações Web (Django software foundation, 2025). Criado em 2003, o Django adota a arquitetura Model-Template-View (MTV) e busca oferecer um desenvolvimento ágil, seguro e escalável, baseado no princípio “*batteries included*” (isto é, trazendo recursos nativos como autenticação, administração e ORM — *Object-Relational Mapper*, responsável por mapear dados de banco de dados em objetos) (Prisma, 2025; Holovaty; Kaplan-Moss, 2009).

Ao herdar essa estrutura, o Wagtail alia flexibilidade e robustez à facilidade de manutenção, destacando-se por seu painel administrativo amigável, pelo suporte a modelos de conteúdo estruturados e por sua aderência a arquiteturas desacopladas.

Para a camada de apresentação ao usuário, a alternativa mais satisfatória é o Next.js, *framework* que estende o React, biblioteca JavaScript criada pelo Facebook em 2013 e voltada à construção de interfaces de usuário baseadas em componentes reutilizáveis (Meta, 2025). O React popularizou conceitos como o *Virtual DOM* (*Document Object Model* — modelo virtual de objetos de página estruturadas em árvore) e a renderização declarativa, favorecendo a criação de interfaces responsivas e manuteníveis.

O Next.js, por sua vez, agrega funcionalidades avançadas, como *Server-Side Rendering* (SSR) e roteamento automático, características que o tornam especialmente indicado para aplicações que exigem desempenho, escalabilidade e otimização para mecanismos de busca (SEO) (Next.js, 2025).

Na prática, o SSR consiste na geração do HTML completo da página diretamente no servidor a cada requisição, enviando-o já renderizado ao navegador do usuário (Next.js, 2025). Essa abordagem melhora a performance percebida, facilita a indexação por buscadores e garante que o conteúdo esteja disponível imediatamente, antes mesmo da execução do JavaScript no cliente.

A utilização do SSR em conjunto com o Next.js e Wagtail possibilita aliar a praticidade da gestão de conteúdo com o desempenho de aplicações modernas baseadas em React, resultando em um ambiente robusto, acessível e otimizado para a Web contemporânea.

A preferência pelo Wagtail e Next.js se fundamenta em uma análise comparativa com outras alternativas disponíveis no ecossistema de CMS e *frameworks* de *frontend*.

O Plone, apesar de sua robustez e tradição em instituições públicas e educacionais, apresenta atualmente limitações quanto à implementação do DSGov 3, já que sua principal interface *frontend*, o Volto, não oferece uma implementação oficial a esse padrão. Dessa forma, sua adoção exigiria um esforço de adaptação maior que apenas utilizar o Next.js, o que resultaria em maior complexidade de manutenção a longo prazo.

Outros CMS amplamente utilizados, como o WordPress, embora contem com grande adoção e disponibilidade de plugins, apresentam desvantagens no contexto deste trabalho, como maior suscetibilidade a vulnerabilidades decorrentes da instalação de extensões de terceiros (Hostinger, 2024).

No que se refere ao *frontend*, bibliotecas e *frameworks* como Vue.js ou Angular também poderiam ser empregados. Contudo, segundo a pesquisa *State of JavaScript 2024*, o React figura consistentemente entre as bibliotecas mais utilizadas e com maior índice de satisfação entre desenvolvedores. O Next.js, por sua vez, também figura como um dos *frameworks* mais utilizados na pesquisa (State of JavaScript, 2024).

Assim, a combinação Wagtail e Next.js representa uma solução equilibrada que une segurança, escalabilidade e facilidade de manutenção, ao mesmo tempo em que possibilita a aderência ao DSGov 3 e oferece um caminho mais sustentável para o desenvolvimento e a manutenção do site do Decom/UFVJM, o tornando flexível o suficiente para quaisquer mudanças no futuro.

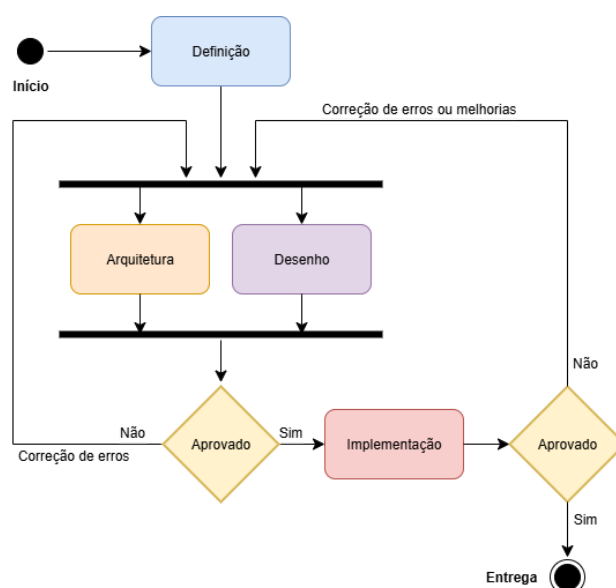
2.6 A metodologia D[AD]I

Alves (2017) propôs, em seu estudo, uma adaptação da metodologia DADI (Definição, Arquitetura, Desenho e Implementação) às particularidades do projeto Novo Portal da UFVJM. A versão ajustada, denominada pelo autor D[AD]I, resulta da unificação das etapas de Arquitetura e Desenho em um único momento metodológico, de modo a otimizar o processo e adequá-lo às demandas específicas do projeto em questão (Alves, 2017).

O projeto Novo Portal caracteriza-se por sua ampla abrangência e permanece em desenvolvimento até os dias atuais, contemplando a reestruturação do Portal Institucional da UFVJM e a integração de seus sites permanentes e temporários. Conforme relatado por Alves (2017), o levantamento realizado à época identificou 111 sites a serem reorganizados, o que evidencia a magnitude e a complexidade da iniciativa.

Nesse contexto, a adaptação feita por Alves (2017) na metodologia DADI original resultou em três etapas: Definição, Arquitetura e Desenho, e Implementação, conforme a **Figura 2** a seguir.

Figura 2. Diagrama geral da metodologia D[AD]I



Fonte: Adaptado de Alves (2017)

2.6.1 Etapa de definição

Conforme descrito por Alves (2017), a etapa de Definição abrange o levantamento sistemático de informações sobre os sites existentes, contemplando tanto sua estrutura quanto os conteúdos disponibilizados. Nesse momento, também são identificadas novas demandas informacionais, que passam a compor o escopo do site a ser desenvolvido, assegurando que o

produto final abarque não apenas o estado atual, mas também as necessidades emergentes da instituição. As atividades desta etapa, conforme Alves (2017), estão descritas no **Quadro 1** a seguir.

Quadro 1. Atividades da etapa de definição da D[AD]I

Atividade	Descrição
D-1 Reunião inicial	Discussões iniciais do projeto com setor. Coleta de informações e registro de usuários do <i>site</i>
D-2 Definir o público-alvo	Definição de público-alvo
D-3 Identificar a equipe	Identificação dos usuários e seus papéis administrativos do <i>site</i>
D-4 Identificar os <i>Stakeholders</i>	Identificação dos <i>Stakeholders</i> do projeto
D-5 Análise de legislação	Identificar e analisar possíveis disposições legais sobre o conteúdo do <i>site</i>
D-6 Identificar <i>links</i> atuais	Mapeamento do conteúdo já existente
D-7 <i>Benchmarking</i>	Mapeamento de referências e boas práticas de outras universidades para o conteúdo do <i>site</i>
D-8 Identificar arquivos	Mapeamento de arquivos já existentes
D-9 Fechar requisitos	Apresentação do conteúdo levantado ao setor para apreciação
D-10 Validar requisitos	Validação dos requisitos pela equipe de gestão do <i>site</i>

Fonte: Adaptado de Alves (2017)

2.6.2 Etapa de arquitetura e desenho

Nesta etapa, Alves (2017) destaca que as informações coletadas na etapa de definição são utilizadas para a proposta de uma nova distribuição do conteúdo. O **Quadro 2** apresenta as atividades desta etapa.

Quadro 2. Atividades da etapa de arquitetura e desenho da D[AD]I

Atividade	Descrição
AD-1 Listar o conteúdo esperado	Modelar a estrutura de páginas do novo <i>site</i> baseada nas informações coletadas anteriormente
AD-2 Realizar <i>Card Sorting</i>	Realização de <i>Card Sorting</i> com os representantes do público-alvo
AD-3 Proposta de organização	Proposta de organização interna de arquivos, categorias de arquivos e temporalidade
AD-4 Construção coletiva da capa	Design participativo da capa do <i>site</i> com a equipe responsável pelo setor utilizando os elementos disponíveis

Atividade	Descrição
AD-5 Criar banco de imagens	Preparar uma seleção de mídias para a criação do novo <i>site</i>
D-6 Criação de ícones e ilustrações	Criar ícones e ilustrações necessárias
AD-7 Finalizar proposta	Apresentação da proposta final e realização de ajustes necessários

Fonte: Adaptado de Alves (2017)

2.6.3 Etapa de implementação

Na etapa de implementação, Alves (2017) considera a criação dos *sites* e inserção do conteúdo previamente preparado. Verificações quanto à propriedade intelectual do conteúdo e a validação da documentação produzida devem ser realizadas antes do início das atividades desta etapa, que são apresentadas no **Quadro 3**.

Quadro 3. Atividades da etapa de implementação da D[AD]I

Atividade	Descrição
I-1 Criar sítio do órgão	Criação e configuração do <i>site</i> com inclusão dos usuários e seus papéis
I-2 Construção do sítio	Criação dos elementos e estrutura do <i>site</i>
I-3 Alimentação do conteúdo	Inserção do conteúdo das páginas pelos membros do setor com supervisão e treinamento
I-4 Treinamento da equipe	Treinamento da equipe de gestão do <i>site</i> que ocorre juntamente à atividade I-3
I-5 Testes e checagem	Verificação e avaliação da estrutura, conteúdo e sua escrita, e atendimento às especificações
I-6 Ajustes pós testes	Correções aos problemas identificados na atividade anterior e realização de novos testes
I-7 Geração do arquivo do sítio	Arquivação e preparação de documentações finais do <i>site</i>
I-8 Entrega formal do sítio	Após todas as atividades, formaliza-se a entrega do <i>site</i>

Fonte: Adaptado de Alves (2017)

2.7 Trabalhos correlatos

O trabalho de Alves (2017), descreve o processo de reestruturação do portal da UFVJM com base na Identidade Digital de Governo (IDG) e na adoção do CMS Plone. A proposta envolveu a substituição do antigo ambiente em Joomla! 1.5 por uma solução mais segura, acessível e escalável, utilizando tecnologias como Docker, além da aplicação da

metodologia D[AD]I (Definição, Arquitetura e Desenho e Implementação) e de práticas participativas com os usuários. Alves (2017) também incorporou recomendações dos Padrões Web em Governo Eletrônico (e-PWG), direcionando o projeto às diretrizes de comunicação digital vigentes à época.

Apesar de sua grande relevância, o estudo concentra-se na implementação da IDG em sua versão anterior ao atual DSGov 3, e utiliza uma versão também obsoleta do CMS Plone, o Plone 4. Dessa forma, não contempla conceitos mais recentes propostos pelo DSGov e, por utilizar o Plone em uma versão datada, atualmente também pode apresentar vulnerabilidades. Em contraste, este trabalho propõe a conciliação com o DSGov 3 na construção de um novo site para o Decom utilizando ferramentas modernas, o que representa um avanço em termos de modernização tecnológica, aderência aos padrões atuais de usabilidade e padronização de interfaces digitais na UFVJM.

O trabalho de Oliveira (2021), por sua vez, tem como objetivo principal incorporar práticas acessíveis ao ambiente digital da universidade. Oliveira (2021) propõe a elaboração de uma cartilha orientativa baseada no Modelo de Acessibilidade em Governo Eletrônico (eMAG), vinculando sua aplicação às atividades do projeto Novo Portal UFVJM, em seu trabalho, também estruturado com base na IDG e no CMS Plone.

O estudo reforça a importância da padronização e da acessibilidade digital em portais públicos, amparando-se em diretrizes legais e técnicas, como as recomendações da WCAG. No entanto, assim como o trabalho de Alves (2017), sua abordagem está ancorada em diretrizes anteriores àquelas propostas pelo DSGov 3. Nesse sentido, o presente projeto se diferencia por incorporar essa nova abordagem, centrada na organização e reutilização de componentes, na aplicação de *design tokens* e na integração sistêmica de princípios de acessibilidade e padronização. Trata-se, portanto, de uma aplicação evolutiva dos estudos anteriores realizados no mesmo contexto institucional.

O trabalho de Gomes (2023), propõe uma reestruturação da interface da plataforma Lattes com o objetivo de melhorar sua usabilidade. Utilizando como base os padrões estabelecidos pelo DSGov, Gomes (2023) adota a metodologia *Double Diamond*, passando pelas etapas de descoberta, definição, desenvolvimento e entrega. O estudo envolveu levantamento bibliográfico, elaboração de protótipos e testes, focando em critérios como eficácia, eficiência e satisfação.

Embora não trate da integração do DSGov a um CMS específico, o trabalho demonstra o potencial do *design system* como ferramenta para modernização de interfaces públicas e melhoria da experiência do usuário em ambientes governamentais. Dessa forma,

contribui para reforçar a relevância do uso de *design systems* como estratégia de padronização visual, usabilidade e acessibilidade, objetivos que também orientam o presente trabalho ao propor a integração do DSGov 3 ao site do Decom.

3 METODOLOGIA

Neste capítulo são apresentados os procedimentos metodológicos que nortearam a construção do MVP do novo site do Decom/UFVJM, incluindo a definição das etapas de trabalho, os materiais empregados e os critérios adotados em cada fase. A escolha da metodologia leva em consideração a natureza aplicada do projeto, a necessidade de nivelamento aos padrões do Governo Federal e as especificidades do contexto institucional em que a solução será implantada.

3.1 Seleção metodológica

Diversas metodologias de desenvolvimento de *software* e projetos Web podem ser aplicadas em contextos acadêmicos e institucionais. Entre as mais difundidas estão os modelos tradicionais sequenciais, como o Modelo em Cascata, caracterizado pela rigidez e pela execução linear das etapas de levantamento de requisitos, análise, implementação e testes (Pressman, 2010). Outra abordagem consolidada é o RUP (*Rational Unified Process*), que estrutura o desenvolvimento em fases iterativas e incrementais, mas cuja complexidade pode representar sobrecarga para projetos de menor porte (Jacobson; Booch; Rumbaugh, 1999).

No campo do desenvolvimento Web contemporâneo, ganham destaque ainda as metodologias ágeis, como o *Scrum*, que favorece entregas contínuas, adaptação a mudanças e colaboração próxima entre equipe técnica e *stakeholders* (Schwaber; Sutherland, 2020). Tais práticas, embora eficazes em ambientes corporativos ou em equipes de maior escala, para projetos menores, podem acabar adicionando mais complexidade.

Diante desse panorama, optou-se pela adoção de uma adaptação da metodologia Definição, Arquitetura e Desenho e Implementação (D[AD]I), proposta por Alves (2017) no contexto do projeto Novo Portal da UFVJM. Diferentemente das abordagens genéricas mencionadas, a D[AD]I foi adaptada especificamente para guiar o desenvolvimento do projeto Novo Portal, contemplando desde o diagnóstico inicial até a implementação. A escolha, portanto, justifica-se por sua aderência temática e por já ter sido validada em contexto análogo ao presente trabalho.

Entretanto, considerando que o escopo deste estudo é mais restrito — limitado ao desenvolvimento de um MVP do *site* do Decom/UFVJM — e que há restrições temporais próprias deste trabalho de conclusão de curso, a metodologia será novamente adaptada. A proposta mantém os princípios fundamentais da D[AD]I, mas simplifica atividades e entregáveis, buscando nivelar rigor metodológico e viabilidade prática.

3.2 D[AD]I Readaptada

A metodologia D[AD]I Readaptada, conforme aplicada neste trabalho, organiza-se em três fases principais, derivadas do modelo original de Alves (2017): (1) Definição, (2) Arquitetura e Desenho, e (3) Implementação. Embora os nomes das fases se mantenham, as atividades contidas em cada uma foram ajustadas para atender às demandas específicas do projeto do novo *site* do Decom e ao cronograma acadêmico e da equipe responsável pelo departamento. A seguir, detalha-se cada uma dessas etapas e os procedimentos adotados.

3.2.1 Etapa de definição

A etapa de definição inaugura o processo de desenvolvimento iniciando a coleta de informações e, nesta readaptação metodológica, foi reestruturada para garantir uma maior agilidade. A principal modificação consistiu na fusão de atividades correlatas. Com o objetivo de simplificar o mapeamento dos envolvidos, as tarefas de identificação de *stakeholders* e da equipe foram unificadas, uma vez que, neste contexto, os grupos de interesse e os executores apresentam uma interseção. Similarmente, visando a uma análise de conteúdo mais coesa, o levantamento de links e arquivos do *site* atual foi consolidado em “levantamento de conteúdo legado” como uma única frente de trabalho.

Outra alteração relevante foi a remoção da “Reunião inicial” como uma atividade obrigatória desta fase. A justificativa para essa supressão está na natureza dinâmica do projeto, na qual as reuniões de alinhamento podem ser realizadas conforme a necessidade e não como um evento pontual ou restrito ao começo dos trabalhos, o objetivo é reduzir a dependência de encontros para a realização das atividades planejadas, assim, as reuniões podem ser úteis para validar os resultados ou para coleta de *feedback*. As atividades que compõem a versão adaptada desta etapa são apresentadas no **Quadro 4**.

Quadro 4. Atividades da etapa de definição da D[AD]I Readaptada

Atividade	Descrição
-----------	-----------

D-1 Identificar equipe	Definição dos <i>stakeholders</i> e levantamento de usuários do site e seus papéis.
D-2 Definir o público-alvo	Definição de público-alvo
D-3 Identificar conteúdo legado	Identificação de links, páginas e documentos já existentes
D-4 Análise de legislação	Identificar e analisar possíveis disposições legais sobre o conteúdo do <i>site</i>
D-5 <i>Benchmarking</i>	Mapeamento de referências e boas práticas de outras universidades para o conteúdo do <i>site</i>
D-6 Fechar requisitos	Apresentação do conteúdo levantado ao setor para apreciação
D-7 Validar requisitos	Validação dos requisitos pela equipe de gestão do <i>site</i>

Fonte: Autor, adaptado de Alves (2017)

3.2.2 *Etapa de arquitetura e desenho*

A segunda fase da metodologia, a arquitetura e desenho, é responsável por traduzir os requisitos levantados em uma estrutura de navegação e em um projeto visual para o site.

A primeira alteração foi o corte da atividade de *Card Sorting*. A decisão baseia-se na baixa complexidade esperada para a arquitetura de informação do *site* do Decom, o que permite uma definição estrutural mais direta a partir da análise dos requisitos.

Em seguida, a abordagem para a criação da página principal foi modificada: a “Construção coletiva da capa” foi substituída pela “Proposta de capa”. Neste novo fluxo, em vez de uma dinâmica totalmente colaborativa, o desenvolvedor elabora e apresenta uma proposta de *layout*, que é então refinada com base no *feedback* dos interessados. Essa abordagem otimiza o tempo e se mostra mais eficiente para a escala deste trabalho.

Por fim, as tarefas de “Criar banco de imagens” e “Criar ícones e ilustrações” foram unificadas na atividade “Produção de Ativos Visuais”, centralizando a criação da identidade gráfica do projeto e garantindo maior coesão entre os elementos. O **Quadro 5** detalha as atividades desta etapa.

Quadro 5. Atividades da etapa de arquitetura e desenho da D[AD]I Readaptada

Atividade	Descrição
AD-1 Listar o conteúdo esperado	Modelar a estrutura de páginas do novo <i>site</i> baseada nas informações coletadas anteriormente
AD-2 Proposta de organização	Proposta de organização interna de arquivos, categorias de arquivos e temporalidade
Atividade	Descrição

AD-3 Proposta de capa	Proposta de <i>layout</i> para página inicial do <i>site</i>
AD-4 Produção de ativos visuais	Preparar imagens, ícones e outros ativos visuais necessários
AD-5 Finalizar proposta	Apresentação da proposta final e realização de ajustes necessários

Fonte: Autor, adaptado de Alves (2017)

3.2.3 *Etapa de implementação*

A terceira e última fase da metodologia, a implementação, corresponde à materialização do projeto, convertendo o *design* e a arquitetura definidos em um *site* funcional e navegável.

Para esta etapa, optou-se por manter a maioria das atividades propostas no modelo D[AD]I original de Alves (2017), a condução desta etapa excluiu apenas o "treinamento da equipe" que supera o escopo deste trabalho. Contudo, para alinhar a metodologia à natureza deste projeto, algumas atividades foram ressignificadas. A documentação exigida na "geração do arquivo do sítio" e a "entrega formal", por exemplo, foram unificadas na produção e apresentação deste trabalho.

De forma semelhante, a abordagem de validação foi repensada: as atividades de “testes e checagem” e “ajustes pós-testes” deixaram de ser etapas finais para se tornarem um processo contínuo e iterativo, executado ao longo de todo o desenvolvimento para proporcionar maior agilidade e qualidade.

4 DESENVOLVIMENTO

O desenvolvimento deste trabalho seguiu um roteiro dividido em duas fases principais, em conformidade com a metodologia D[AD]I Readaptada. A fase de Planejamento compreendeu as etapas de Definição e Arquitetura e Desenho, dedicando-se à especificação e ao design da solução. Em seguida, a fase de Desenvolvimento focou-se na construção do projeto, abrangendo as atividades da etapa de Implementação.

4.1 Fase de Planejamento

Nesta fase inicial, o foco foi o levantamento de requisitos, a compreensão do domínio do problema e a definição da arquitetura informacional da solução.

4.1.1 Definição

A etapa de Definição, primeira fase da metodologia D[AD]I Readaptada, teve como objetivo estabelecer os fundamentos do projeto. A execução de suas atividades resultou na produção de artefatos essenciais, incluindo a formalização de pessoal da equipe e dos stakeholders (**Quadro 6**) e a caracterização do público-alvo (**Quadro 7**). Adicionalmente, foi realizado um mapeamento do conteúdo legado, apresentado de forma resumida no **Quadro 8** e detalhado no **Apêndice A**, e conduzida uma análise de *benchmarking*, cujos resultados estão consolidados no **Apêndice B**.

Quadro 6. Definição da equipe e dos stakeholders

Grupo	Quantidade de pessoas
<i>Stakeholders</i>	3
Equipe	19

Fonte: Elaborado pelo autor

Quadro 7. Definição de público-alvo por ordem de relevância

Público	Descrição
Estudantes internos	Estudantes da UFVJM que já conhecem, mesmo que parcialmente, a universidade e seu funcionamento;
Docentes, técnicos administrativos e comunidade interna	Pessoas que têm contato com informações da universidade diretamente de suas fontes, e que têm interesse em alguma divulgação de conteúdo.
Possíveis estudantes	Pessoas que não possuem vínculo com a universidade e não conhecem seu funcionamento ou estrutura, mas desejam ingressar em algum curso da instituição. Pode incluir outras instituições que se envolvam no processo de ingresso de estudantes em universidades.
Comunidade em geral	Pessoas com interesse em informações sobre o departamento e seus processos. Pode incluir pessoas com interesse em vagas de emprego.

Fonte: Elaborado pelo autor

Quadro 8. Resumo do mapeamento de conteúdo legado

Conteúdo	Quantidade
Páginas	11
Links externos	2
Arquivos	17
Repetições	Um link externo e um arquivo

Fonte: Elaborado pelo autor

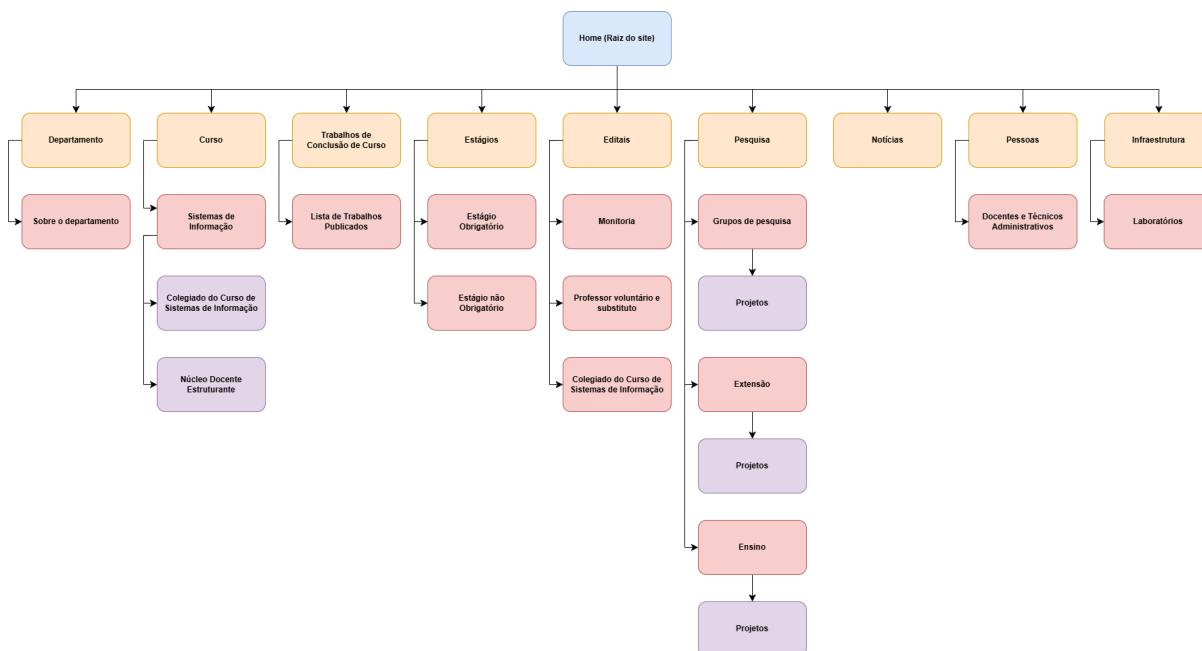
A definição do público-alvo, evidenciou a necessidade de focar o conteúdo do site e sua apresentação nos interesses do público de estudantes e docentes, técnicos administrativos e colaboradores, sendo esses os públicos-alvo de maior relevância para o site. Em paralelo, o mapeamento do conteúdo legado demonstrou um volume de informações relativamente enxuto, com 11 páginas e 17 arquivos a serem migrados ou reestruturados, não levando em conta arquivos de trabalhos de conclusão de curso. Essa baixa complexidade informacional corroborou a decisão metodológica de suprimir a atividade de *Card Sorting* presente na metodologia D[AD]I original, permitindo uma definição mais direta da arquitetura do site.

Por fim, durante a análise de legislação e o *benchmarking* foram definidos os contornos de design e os requisitos não-funcionais do projeto. A análise legal não evidenciou nenhuma legislação já presente no *site*, apenas a necessidade de ajuste ao DSGov 3 conforme já mencionado anteriormente no capítulo de Referencial Teórico, por outro lado a análise comparativa com outras instituições durante o *benchmarking* proveu referências e boas práticas que informaram as decisões tomadas na etapa seguinte, de Arquitetura e Desenho.

4.1.2 *Arquitetura e desenho*

A etapa de Arquitetura e Desenho, fundamentada nas informações coletadas na fase anterior, concentrou-se na elaboração de uma nova estrutura navegacional para o site. O objetivo foi otimizar a organização do conteúdo e facilitar sua descoberta pelos usuários. A arquitetura de informação resultante, detalhada na **Figura 3**, foi organizada em três níveis hierárquicos. O primeiro nível corresponde à raiz do site, onde se encontra a página inicial. O segundo nível ramifica-se em oito categorias principais de conteúdo: Cursos, Infraestrutura, Editais, Pessoas, Estágios, Grupos de Pesquisa, Trabalhos de Conclusão de Curso e Contatos. Por fim, o terceiro nível aprofunda a hierarquia com páginas específicas sobre assuntos subordinados à categoria pai.

Figura 3. Nova estrutura do site do Decom



Fonte: Elaborado pelo autor

Adicionalmente à estrutura navegacional, esta etapa incluiu a elaboração de uma proposta de layout para a página inicial (capa), e a reestruturação de conteúdo das demais páginas. Com base no conteúdo existente e nos novos requisitos, foi desenvolvida uma interface que simplifica a apresentação das informações. O resultado da capa, que pode ser visualizado na **Figura 4**, foca em uma experiência de usuário mais direta e voltada às necessidades do público-alvo.

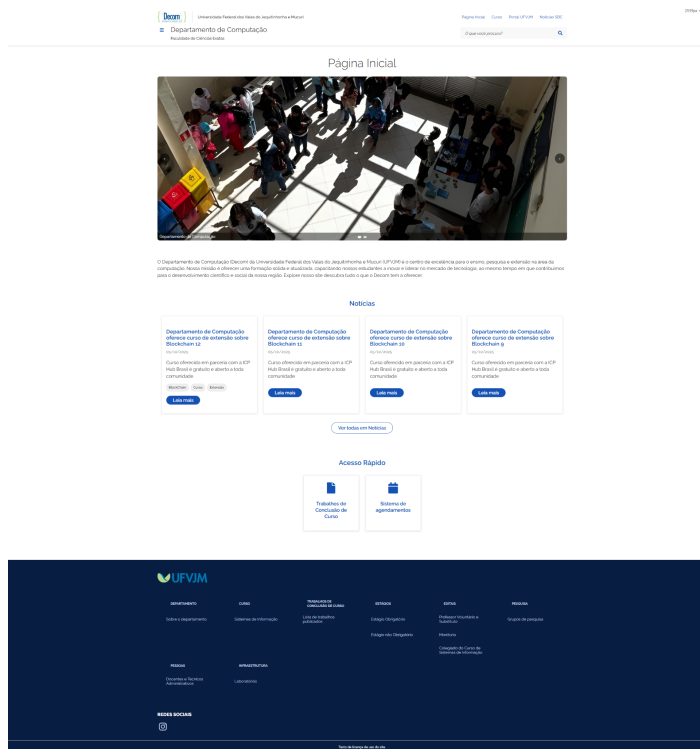
A nova página inicial possui um layout simples, iniciando com um carrossel de imagens de altura ajustável no topo, e um breve texto sobre o departamento. Logo abaixo, uma seção exibe as últimas notícias publicadas no site. A página também é complementada pelos elementos globais de cabeçalho e rodapé.

O conteúdo das demais páginas foi repensado. Os textos legados foram reescritos e redistribuídos para garantir uma separação de assuntos mais lógica e intuitiva, facilitando a navegação.

A atividade de produção de ativos visuais, apesar de mantida na metodologia D[AD]I Readaptada, não representou grandes esforços durante a fase de planejamento, isso porque a maior parte dos ícones utilizados no site são provenientes do próprio DSGov 3. As imagens utilizadas no carrossel da página inicial foram extraídas do Instagram do departamento, onde se encontram publicadas, porém, neste trabalho, foram editadas para preservar a identidade das pessoas nelas presentes. Além disso, uma imagem temática abordando a tecnologia de

blockchain foi criada, com utilização de inteligência artificial, para ilustrar uma notícia de exemplo, escrita a partir de uma publicação real² no atual site do departamento.

Figura 4. Proposta de capa para o site do Decom



Fonte: Elaborado pelo autor

Ainda nesta etapa, foram realizadas reuniões com os *stakeholders* do projeto, nas quais foram formalizadas as definições da etapa anterior e apresentadas, ajustadas e validadas as novas propostas, incluindo a nova estrutura e a capa do site.

4.2 Fase de Desenvolvimento

Este capítulo aborda a execução técnica do projeto: a fase de desenvolvimento. Neste momento, o foco se volta para a codificação, a componentização e a integração que deram vida ao MVP do novo site. Serão detalhadas as tecnologias e atividades de implementação das partes do sistema, com ênfase na utilização do DSGov 3 como pilar para a construção da interface, e as decisões de arquitetura que garantiram um produto funcional e voltado às definições da fase de planejamento.

² Disponível em: decom.ufvjm.edu.br/dc2020/

4.2.1 Materiais

Para o desenvolvimento do sistema, foram utilizadas diferentes ferramentas e tecnologias, a seguir são apresentadas as principais:

- Docker versão 28.4.0: ferramenta de containerização de aplicações;
- Node.js versão 22.20: ambiente de execução de código JavaScript;
- Python versão 3.12: linguagem de programação;
- TypeScript versão 5: linguagem de programação;
- Next.js versão 14.2.3: *framework* de aplicações baseado em React;
- Django versão 5.2.6: *framework* de aplicações Web;
- Wagtail versão 7.1.1: CMS baseado em Django;
- Nginx versão 1.28.0: servidor Web;
- Postgres versão 15: sistema de banco de dados.

A utilização dessas tecnologias é detalhada nos capítulos seguintes.

4.2.2 Arquitetura do sistema

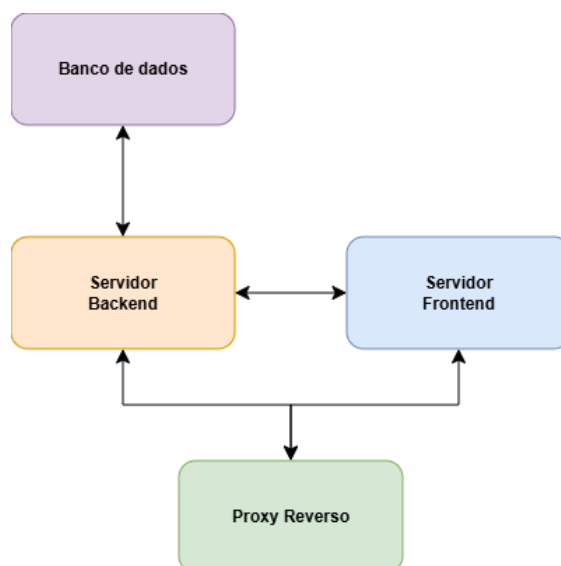
Este capítulo detalha a estrutura arquitetônica do produto, projetada para ser moderna, desacoplada, escalável e de fácil manutenção. A arquitetura foi dividida em três camadas principais de análise: a arquitetura geral do sistema, a arquitetura do *backend* e a arquitetura do *frontend*.

4.2.2.1 Arquitetura geral

A solução adota o padrão de arquitetura Cliente-Servidor com uma abordagem desacoplada (*Headless*). Nesse modelo, o *backend* e o *frontend* são desenvolvidos como duas aplicações independentes que se comunicam por meio de uma *Application Programming Interface* (API, isto é, uma interface de programação de aplicações) (Tanenbaum; Wetherall, 2011). Essa separação oferece flexibilidade para que cada parte do sistema evolua de forma autônoma e permite que o mesmo conteúdo do *backend* seja consumido por diferentes *frontend* no futuro (como aplicativos móveis ou outros portais). Essa arquitetura também aumenta a modularidade do sistema à medida que cada parte, seja *backend* ou *frontend*, pode ser substituída ou modificada sem prejuízo de outras partes.

O fluxo de comunicação é orquestrado por um *proxy* reverso, que atua como um intermediário entre o usuário e os servidores da aplicação. Conforme ilustrado na **Figura 5**, o *proxy* reverso é responsável por direcionar as requisições para o serviço apropriado:

Figura 5. Diagrama da arquitetura geral do sistema

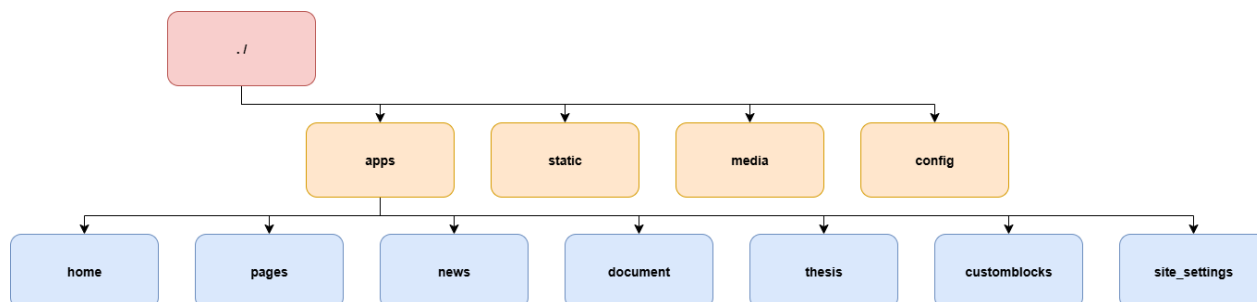


Fonte: Elaborado pelo autor

Neste modelo, o processo de acesso a uma página se inicia quando o usuário, através de seu navegador, faz uma requisição. Essa requisição é primeiramente recebida pelo *proxy* reverso, que a repassa ao servidor *frontend*. Para construir a página, o *frontend* busca as informações dinâmicas necessárias fazendo chamadas de API ao servidor *backend*. O *backend* consulta o banco de dados, processa os dados solicitados e os envia de volta para o *frontend* em formato JSON (*JavaScript Object Notation*). Finalmente, o servidor *frontend* utiliza essas informações para renderizar a página, que é então enviada ao usuário, passando novamente pelo *proxy* reverso. O *proxy* reverso atua como um ponto de entrada único para o sistema, permitindo que os clientes acessem todos os serviços através de uma mesma combinação de *host* e porta, enquanto ele faz o roteamento correto das requisições para os diferentes serviços no servidor.

4.2.2.2 Arquitetura do backend

O *backend* foi desenvolvido com o CMS Wagtail, que é construído sobre o *framework* Django. A estrutura do projeto foi organizada de forma modular, conforme a **Figura 6**, promovendo a separação de responsabilidades e facilitando a manutenção.

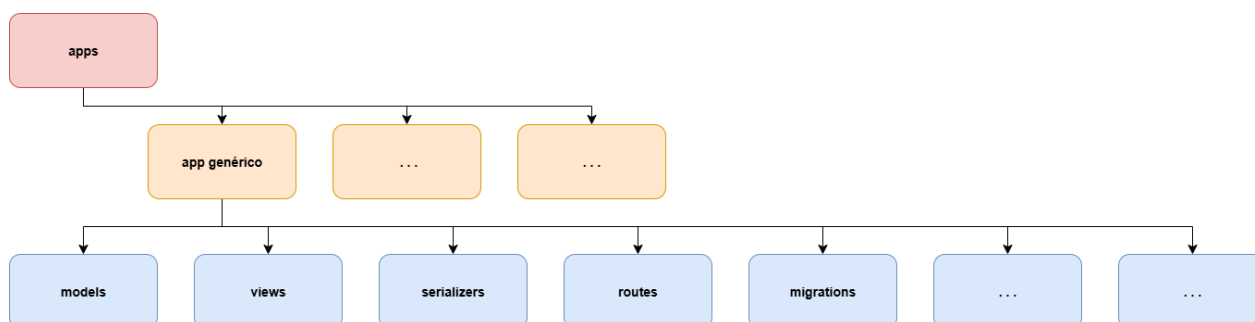
Figura 6. Diagrama da arquitetura do *backend*

Fonte: Elaborado pelo autor

O projeto está organizado segundo a estrutura convencional do Django, que promove a modularidade e a clareza. Na pasta `config`, ficam as configurações globais, como variáveis de ambiente, URLs principais e as definições do servidor WSGI (*Web Server Gateway Interface* — servidor usado pelo Django). A modularidade é alcançada através da pasta `apps`, que agrupa cada funcionalidade do site — como notícias, páginas e documentos — em suas próprias aplicações autocontidas. Por fim, os arquivos de usuários (imagens e documentos enviados via Wagtail) são armazenados na pasta `media`, enquanto os arquivos estáticos da aplicação, como CSS e JavaScript, residem na pasta `static`.

Ainda, para cada pasta representando uma funcionalidade (*app*) na pasta `apps`, a estrutura interna segue um padrão, representado na **Figura 7**, para organizar a lógica da API.

Figura 7. Diagrama da arquitetura geral de um *app*



Fonte: Elaborado pelo autor

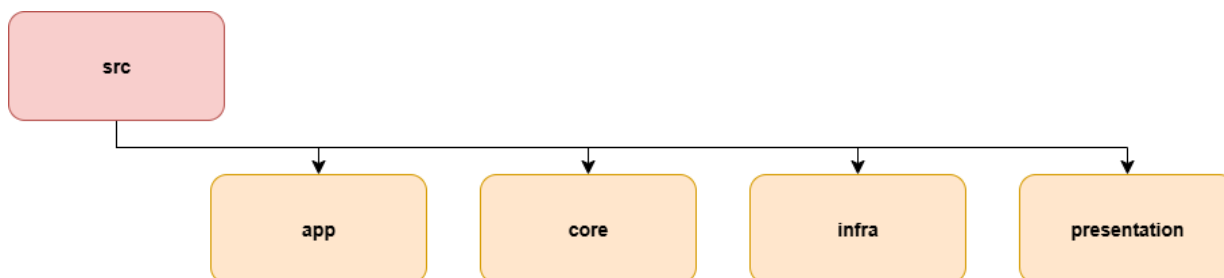
Nesta estrutura, o fluxo de uma requisição de API é claramente definido pela estrutura de pastas. Primeiramente, o diretório *routes* mapeia a URL de uma requisição para a *view* apropriada. A *view*, por sua vez, contém a lógica de negócio, interagindo com os *models* (que representam as tabelas do banco de dados). Antes de enviar a resposta, os *serializers* entram em ação, convertendo os dados dos modelos para o formato JSON e garantindo que apenas os campos necessários sejam expostos. Por fim, a pasta *migrations* gerencia o versionamento do banco de dados, refletindo as alterações feitas nos modelos. Essas pastas não representam a estrutura completa de todos os *apps*, mas sim uma estrutura padrão e comum entre eles.

4.2.2.3 Arquitetura do *frontend*

O *frontend* foi construído com Next.js e segue uma arquitetura em camadas para separar as responsabilidades e desacoplar a lógica de negócio da interface do usuário.

O fluxo de uma requisição através da arquitetura do *frontend* pode ser entendido como uma jornada sequencial através de suas quatro camadas principais, refletidas na estrutura de pastas do diretório *src*, detalhada na **Figura 8**.

Figura 8. Diagrama da arquitetura geral do *frontend*

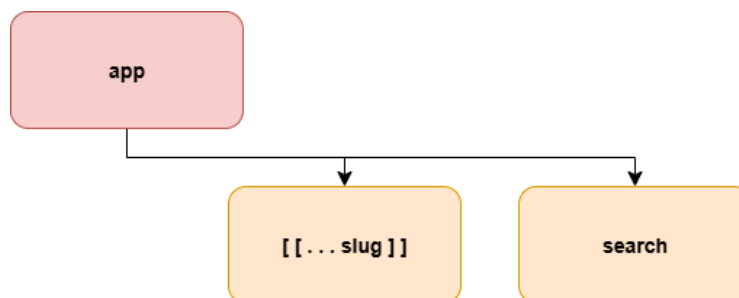


Fonte: Elaborado pelo autor

O fluxo começa na camada de roteamento (*app*), mostrada em detalhe na **Figura 9**, que serve como ponto de entrada. Quando um usuário acessa uma URL, o sistema de

roteamento do Next.js (*App Router*) captura a requisição. A rota dinâmica `[[...slug]]` é a peça chave, atuando como uma rota "catch-all", isto é, uma rota genérica que intercepta a maioria das requisições. Esta camada extrai o caminho da URL e inicia o processo de busca de conteúdo.

Figura 9. Diagrama da Camada de Roteamento



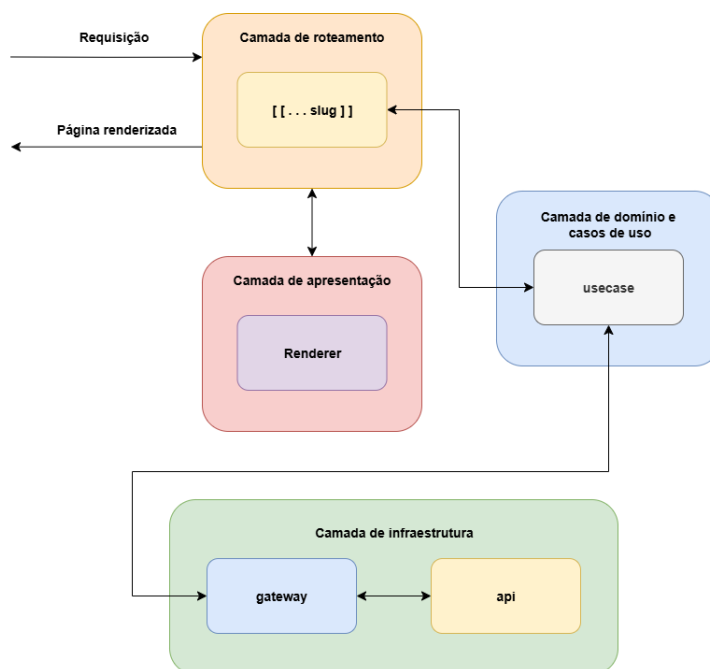
Fonte: Elaborado pelo autor

A partir deste ponto, a requisição é passada para a camada de domínio e casos de uso (pasta core). A rota aciona um caso de uso (*usecase*) específico, como "buscar página por *slug*". Este caso de uso contém a lógica de negócio pura e não sabe de onde os dados virão; ele apenas executa a regra de negócio e depende de uma abstração, a interface *gateway*, para obter os dados necessários.

A responsabilidade de buscar os dados concretos recai sobre a camada de infraestrutura (pasta infra). A implementação do *gateway* nesta camada, por exemplo *WagtailPageGateway*, realiza a chamada HTTP para a API do Wagtail. A lógica de baixo nível para a comunicação, como o uso de funções de rede e a conversão de JSON, está encapsulada no diretório api. Uma vez que a resposta da API é recebida, a camada de infraestrutura a transforma em uma entidade (uma estrutura de dados definida na camada core) e a retorna para o caso de uso.

Com os dados em mãos, o fluxo retorna para a camada app, que agora possui a informação necessária para renderizar a página. Neste ponto, entra a camada de apresentação (pasta presentation). A rota seleciona o renderizador apropriado para o tipo de conteúdo recebido. Este renderizador funciona como um orquestrador, recebendo os dados processados e distribuindo-os entre componentes menores e reutilizáveis — como *Header*, *Footer* e o renderizador de blocos — para construir a interface visual completa. Esses componentes puros são responsáveis apenas pela aparência, garantindo que a lógica de negócio permaneça isolada. Ao final, a página HTML montada é enviada de volta ao usuário. A representação visual deste fluxo pode ser conferida na **Figura 10**, a seguir.

Figura 10. Fluxo da requisição no *frontend*



Fonte: Elaborado pelo autor

4.2.3 Construção de conteúdo dinâmico

Um dos principais requisitos do projeto é oferecer flexibilidade na criação e gestão do conteúdo, permitindo que os administradores do site pudessem construir páginas com *layouts* variados sem a necessidade de intervenção técnica.

No *backend*, a flexibilidade foi alcançada através do uso intensivo do recurso *StreamField* do Wagtail. Diferente de um campo de texto rico tradicional, que oferece um único editor de conteúdo, o *StreamField* permite que uma página seja construída como uma sequência de diferentes "blocos" de conteúdo (Wagtail, 2025). Cada bloco é um tipo de conteúdo específico, como um parágrafo de texto, uma imagem, um vídeo incorporado ou uma citação destacada.

O *StreamField* permitiu que modelos de página fossem definidos com um corpo de conteúdo (*body*) do tipo *StreamField*, contendo uma lista de blocos disponíveis para o editor. Dessa forma, ao editar uma página, o administrador pode adicionar, remover e reordenar esses blocos livremente (**Fig. 11**), montando o *layout* da página de acordo com sua necessidade. A vantagem dessa abordagem é transformar a gestão de conteúdo de uma simples edição de texto para uma construção de fato de páginas, dando autonomia para a equipe de gestão do site.

Figura 11. Edição de página com *StreamField* no Wagtail



Fonte: Elaborado pelo autor

No frontend, o desafio era consumir essa estrutura de blocos recebida via API como um array de objetos JSON, e traduzi-la visualmente para o usuário. Cada objeto no *array* representa um bloco e contém seu tipo (*type*), seus atributos (*attributes*) e blocos filhos (*children*) aninhados até um bloco base de texto (“*type*”: “*text*”) (**Fig. 12**), ou seja, o conteúdo em si.

Figura 12. Representação em JSON de um bloco de parágrafo



Fonte: Elaborado pelo autor

Para lidar com isso, foi implementado um componente especializado na camada de apresentação, o *BlockContentRenderer*. Este componente é responsável por iterar sobre o *array* de blocos recebido da API e, para cada bloco, utilizar o valor da propriedade *type* para determinar qual componente React específico deve ser renderizado. Foi criado um mapeamento que associa cada tipo de bloco a um componente visual correspondente (por exemplo, o bloco de tipo “p” é mapeado para um componente que renderiza um parágrafo formatado, enquanto o bloco “img” para uma *tag* HTML de imagem).

Mesmo páginas que não utilizam *StreamField*, mas sim um editor de texto rico convencional, são serializadas pelo *backend* para que suas representações em JSON sigam a mesma estrutura de blocos. Assim, o *frontend* recebe sempre uma estrutura de dados familiar e consegue tratá-la sem complexidade adicional.

Dessa forma, quando dados retornam da API para a rota “*catch-all*”, já apresentada anteriormente, a rota chama um renderizador com base no tipo de página, e este renderizador utiliza o *BlockContentRenderer* para renderizar o conteúdo da página dinamicamente.

Esse mecanismo de renderização dinâmica é a âncora da arquitetura desacoplada do projeto. Ele permite que novos tipos de blocos possam ser adicionados no *backend* no futuro, exigindo apenas o mapeamento deste bloco a um novo componente visual correspondente no *frontend*, sem a necessidade de alterar a lógica de busca de dados ou definir estaticamente a estrutura das páginas. Isso resulta em um sistema altamente extensível e flexível.

4.2.4 Integração e complementação do DSGov 3

A adesão ao Padrão Digital de Governo é desde o início um dos eixos deste projeto. A implementação do DSGov 3 no *frontend* acelerou o desenvolvimento ao fornecer uma base consistente de componentes acessíveis e reutilizáveis por meio da biblioteca React Components - GOVBR-DS³. No entanto, a aplicação prática revelou alguns entraves relacionados à estratégia de renderização das páginas, além da necessidade de não apenas consumir o *design system*, mas também de complementá-lo para atender às especificidades do site do Decom, levantados na fase de planejamento.

A base da interface do usuário foi construída utilizando os componentes nativos oferecidos pela biblioteca do DSGov 3 para React. Elementos estruturais globais, como o cabeçalho (*Header*), o rodapé (*Footer*) e o sistema de navegação por migalhas de pão (*Breadcrumbs*), foram integrados diretamente. O uso desses componentes garantiu que o site

³ Disponível em: <https://gitlab.com/govbr-ds/bibliotecas/react-components>

já nascesse com características essenciais de responsividade e acessibilidade, além de seguir a identidade visual do Governo Federal. Componentes de interação, como botões, e de apresentação de conteúdo, como os cards utilizados na listagem de notícias, também foram aproveitados, o que resultou em uma economia de tempo de desenvolvimento e na garantia de uma experiência de usuário consistente e familiar para quem navega em outros portais governamentais.

O uso dos componentes da biblioteca React Components - GOVBR-DS, no entanto, introduziu um grande ponto de atenção no que tange à estratégia de renderização das páginas. Inicialmente, foi considerada a aplicação da renderização no lado do servidor (SSR – *Server-Side Rendering*) em conjunto com a referida biblioteca. A adoção do SSR visava, principalmente, aprimorar a otimização para mecanismos de busca e, ao mesmo tempo, habilitar o armazenamento de páginas em *cache*.

Contudo, emergiu uma incompatibilidade, decorrente das diferenças entre o ambiente de execução do servidor (Node.js) e o do navegador, uma vez que o primeiro não dispõe de certas funcionalidades exclusivas do segundo. Constatou-se que determinados componentes da biblioteca GOVBR-DS fazem uso de recursos presentes apenas no ambiente do navegador, o que resultava em erros durante o processo de renderização via SSR.

Apesar da ocorrência de erros no servidor, as páginas eram carregadas normalmente no navegador devido a um mecanismo de recuperação do *framework* Next.js. No entanto, como essas falhas impediam a conclusão bem-sucedida do processo de SSR, todas as vantagens associadas a essa abordagem, como o SEO aprimorado e o *cache*, eram eliminadas.

A solução desenvolvida para contornar essa limitação foi a criação de componentes de invólucro (*wrappers*). Esses componentes atuam como intermediários, responsáveis por selecionar dinamicamente qual elemento será renderizado com base no ambiente de execução. Em ambientes de servidor, os invólucros renderizam elementos alternativos e seguros, como tags HTML nativas ou componentes que não dependem de recursos do navegador. Por outro lado, no ambiente do navegador, os mesmos invólucros renderizam diretamente os componentes originais e interativos da biblioteca GOVBR-DS, durante um processo chamado de hidratação (*hydration*).

No decorrer do desenvolvimento, foram avaliadas bibliotecas alternativas que implementam o DSGov, como a *react-dsgov*⁴, mantida pelo Tesouro Nacional. Contudo, a instalação deste pacote por meio do gerenciador npm resultou na emissão de múltiplos alertas

⁴ Disponível em: <https://github.com/tesouro/react-dsgov>

de vulnerabilidade de segurança. Diante desse risco, a alternativa foi descartada, optando-se por prosseguir com a solução previamente definida que utiliza componentes de invólucro.

Adicionalmente, foi identificado um segundo obstáculo de ordem técnica: a incompatibilidade entre a biblioteca GOVBR-DS e a versão estável mais recente do *framework* Next.js, a versão 15. Para solucionar essa questão e assegurar o correto funcionamento da aplicação, foi necessário realizar o *downgrade* do Next.js para a sua versão estável anterior, a 14.

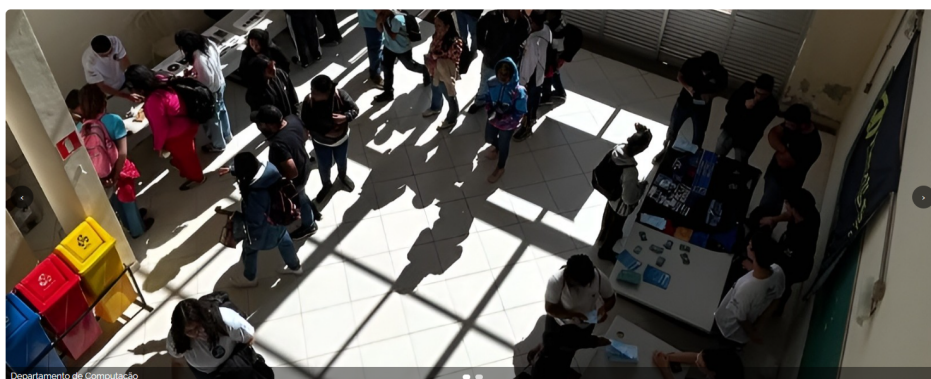
Apesar da cobertura do DSGov 3, certas funcionalidades previstas para o novo site do Decom demandaram componentes que não fazem parte do escopo padrão do *design system*, e não tiveram implementações prontas na biblioteca GOVBR-DS. Nesses casos, a estratégia foi desenvolver componentes customizados, porém seguindo os princípios e os *design tokens* (cores, tipografia, espaçamento) do DSGov para manter a coesão visual.

A necessidade de desenvolver componentes personalizados foi identificada durante a fase de planejamento, a partir da análise do conteúdo preexistente no sistema legado. Nessa análise, constatou-se a presença de elementos dinâmicos, como carrosséis de imagens e listagens de conteúdo (notícias, publicações), que não possuíam equivalentes diretos na biblioteca GOVBR-DS. Diante dessa demanda, um dos principais componentes desenvolvidos foi um carrossel de imagens (**Fig. 13**), projetado para a página inicial com o objetivo de exibir banners e destaques visuais.

A implementação do componente no *frontend* foi realizada com base em requisitos de responsividade e usabilidade. Ele foi projetado para operar de forma fluida tanto em computadores, com navegação via mouse, quanto em dispositivos móveis, com interações por toque. Adicionalmente, foram incorporadas funcionalidades para aprimorar a experiência do usuário e a acessibilidade, como a pausa da transição automática de slides ao posicionar o cursor sobre o componente e o suporte à navegação completa via teclado (tecla Tab).

No *backend*, o carrossel foi implementado como um bloco de conteúdo personalizável, oferecendo ao administrador do *site* controle sobre alguns parâmetros. Entre as opções de customização, incluem-se a opção para ativar ou desativar a transição automática das imagens, o ajuste da velocidade da transição, a capacidade de associar um *link* a cada imagem — tornando-a clicável — e a configuração da proporção visual. Por padrão, o componente ocupa toda a largura do contêiner da página (cerca de 1480 pixels em monitores grandes), e sua altura é ajustada como um percentual dessa largura. Para obter uma proporção de 16:9, por exemplo, a altura é configurada como 56% da largura total.

Figura 13. Componente de carrossel



Fonte: Elaborado pelo autor

Figura 14. Parâmetros do componente de carrossel

Fonte: Elaborado pelo autor

Para a exibição de coleções de conteúdo, como notícias e documentos, foi desenvolvido um componente de listagem genérico (**Fig. 15**). No *frontend*, este componente recebe uma lista de páginas e as renderiza dinamicamente em cartões (*cards*) individuais. Cada cartão apresenta informações essenciais como título clicável, descrição, data de publicação e um botão de acesso ao conteúdo completo, por redundância. Esses cartões são organizados em um *layout* de grade (*grid*) responsivo, que se adapta automaticamente à quantidade de itens e ao espaço disponível na tela, permitindo diferentes configurações

visuais. Para assegurar a coesão com a identidade visual do restante do site, a construção deste componente utilizou elementos base da biblioteca GOVBR-DS.

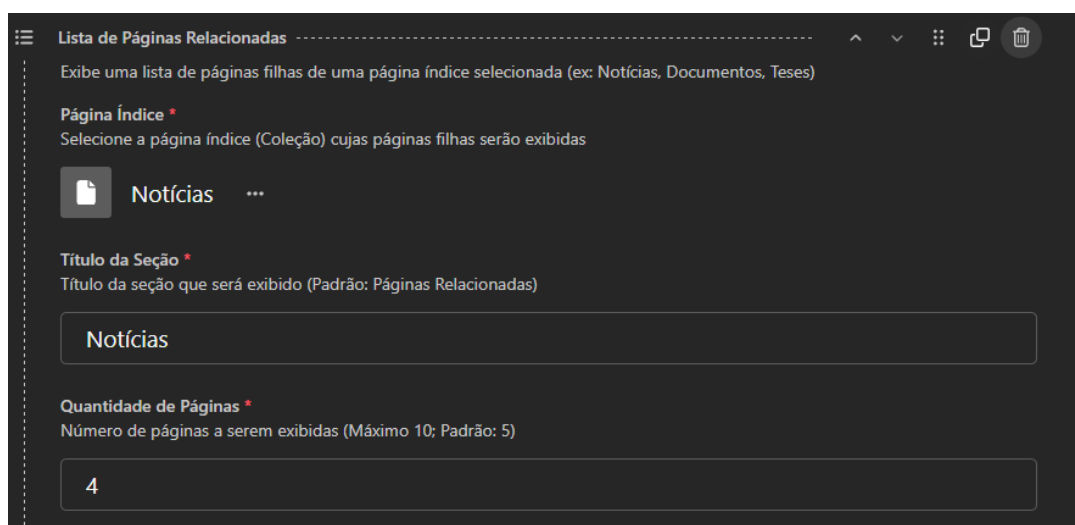
Já no *backend*, o componente foi implementado como um bloco de conteúdo flexível. Sua configuração principal permite que o administrador do *site* selecione uma "página índice", que funciona como um agregador para outras páginas. A partir dessa seleção, o sistema busca e retorna ao *frontend* todas as páginas-filhas publicadas sob esse índice. O bloco de configuração também inclui campos adicionais para personalização, como a definição de um título para a seção e um seletor para limitar a quantidade de itens a serem exibidos.

Figura 15. Componente de listagem genérico



Fonte: Elaborado pelo autor

Figura 16. Parâmetros do componente de listagem genérico



Fonte: Elaborado pelo autor

Adicionalmente aos componentes supracitados, também foi desenvolvido um componente de "itens de acesso rápido". A criação desta funcionalidade partiu da necessidade identificada de expor *links* de alta relevância de forma mais visual e direta nas páginas. Para atender a esse requisito, o componente de *frontend* foi projetado para receber uma coleção de itens, cada um contendo um título, um *link* e um ícone associado. Esses itens são, então, renderizados como cartões individuais e organizados em um *layout* de grade responsiva.

No *backend*, o bloco de conteúdo correspondente oferece opções de personalização. Similarmente ao bloco de listagem genérica, é possível definir um título customizado para a seção. Além disso, o bloco permite a configuração individual de cada cartão, viabilizando a atribuição de uma classe de ícone específica da biblioteca FontAwesome 4 para cada item de acesso rápido. Cada cartão pode ainda receber um link para uma página interna, usando o seletor de páginas do próprio Wagtail, ou uma URL externa, usada no caso de *sites* externos ou arquivos.

Figura 17. Componente de Acesso Rápido



Fonte: Elaborado pelo autor

Figura 18. Parâmetros do componente de acesso rápido

Acesso Rápido

Seção com cartões de acesso rápido em layout responsivo (grade em desktop, lista em mobile)

Título da Seção
Título da seção (opcional) (Padrão: 'Acesso Rápido')

Acesso Rápido

Cartões
Adicione cartões de acesso rápido (mínimo 1, máximo 12)

Cartão de Acesso Rápido
Um cartão individual com título, ícone e link (interno ou externo)

Título *
Título do cartão de acesso rápido

Trabalhos de Conclusão de Curso

Ícone FontAwesome *
Nome do ícone FontAwesome 4 (ex: 'fa-home', 'fa-user', 'fa-file'. Padrão 'fa-link'). Apenas 'fa-*'

fa-file

Link *
Escolha uma página interna ou URL externa

Página Interna *
Trabalhos de Conclusão de Curso

Fonte: Elaborado pelo autor

Ainda, com o objetivo de otimizar a performance percebida e garantir a estabilidade visual das páginas durante o carregamento de dados, foi adotada a estratégia de implementação de componentes de *skeleton*, componentes visuais que tem a função de imitar a estrutura básica do conteúdo até que este esteja pronto para ser exibido. Essa abordagem visa mitigar o *Cumulative Layout Shift* (CLS), uma métrica criada pela Google que quantifica a instabilidade do *layout* de uma página, a qual ocorre quando elementos visuais mudam inesperadamente de posição após a renderização inicial (MDN Web Docs, 2025).

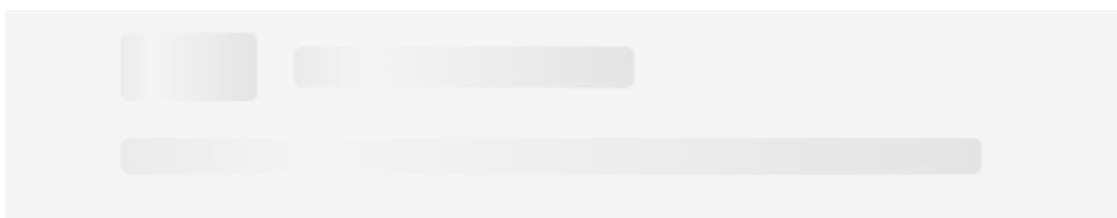
Esse fenômeno foi particularmente notável em componentes globais da interface, como o cabeçalho, o rodapé e os *breadcrumbs*. Para solucionar o problema, foram desenvolvidas versões *skeleton* para esses componentes. Eles funcionam como estruturas de marcação (*placeholders*) estáticas e leves, que ocupam previamente o espaço similar ao que será preenchido pelo componente final.

Mesmo que as informações necessárias para os componentes globais, citados anteriormente, estejam disponíveis durante a renderização no servidor, a impossibilidade de

utilizar os componentes finais, da biblioteca GOVBR-DS, durante o SSR, faz com que seja necessária a renderização de componentes alternativos, nesse caso, os componentes *skeleton*. Dessa forma, a página é renderizada no servidor com *skeletons*, porém ainda carrega em seu HTML todas as informações que estariam presentes nos componentes finais, mesmo que não os mostre visualmente, assim, a indexação de conteúdo por mecanismos de busca não é afetada.

Após o carregamento da página no navegador do usuário, os componentes *skeleton* são trocados pelos componentes finais, mantendo similares as proporções e tamanhos, dessa forma a página sofre menos mudanças bruscas no *layout*.

Figura 19. Exemplo de componente de *skeleton*



Fonte: Elaborado pelo autor

4.2.5 Implementação de recursos adicionais

Além das funcionalidades estruturais e da integração com o DSGov, o desenvolvimento do projeto contemplou a criação de tipos de conteúdo especializados, visando atender às necessidades de um *site* acadêmico e aprimorar a experiência de edição dos administradores. Identificou-se a necessidade de um tratamento especial para certos conteúdos, sendo inadequado tratá-los como uma página genérica. Dessa forma, foram modelados novos tipos de páginas para permitir a definição de metadados essenciais à filtragem e busca. Concretamente, implementaram-se modelos específicos como NewsPage, DocumentPage e ThesisPage. Cada um desses modelos possui campos próprios e estruturados — como "data de publicação" para notícias ou "autor" e "ano" para trabalhos de conclusão de curso — garantindo, assim, a consistência e a integridade dos dados.

Para a organização dos conteúdos específicos, foi adotado o padrão de "páginas de índice". Essa abordagem consistiu na criação de páginas de coleção correspondentes a cada tipo de conteúdo, como NewsIndexPage, DocumentsIndexPage e ThesesIndexPage. Essas páginas funcionam como agregadores automáticos, listando todas as páginas-filhas publicadas do seu tipo específico.

A adoção dessa estrutura hierárquica apresenta vantagens em duas frentes: primeiramente, otimiza a organização do conteúdo no painel administrativo do Wagtail; em segundo lugar, simplifica a exposição dos dados via API, permitindo que o *frontend* solicite coleções completas — como as notícias do departamento ou TCCs publicados em 2024 — a partir de um único *endpoint*, isto é, caminho de comunicação com a API.

Essa arquitetura de *backend* foi diretamente espelhada no *frontend*. Diferentemente de páginas genéricas, as páginas de coleção apresentam uma estrutura de listagem que não representa um bloco de conteúdo específico no *backend*; dessa forma, um renderizador de página genérica não seria capaz de processar o conteúdo dessas páginas. Assim, foi necessário definir renderizadores específicos para cada tipo de coleção, como o `NewsIndexRenderer`, que exibe a listagem de notícias com seus filtros, e o `NewsDetailRenderer`, responsável por renderizar o conteúdo detalhado de um item individual.

Além das implementações acima, também foi implementado um novo modelo de página capaz de oferecer uma edição mais avançada de texto, o modelo `TextPage`. Embora o *StreamField* ofereça máxima flexibilidade para a estruturação de páginas complexas, foi constatado que, para conteúdos mais simples e predominantemente textuais, um editor de texto rico tradicional traria mais opções de formatação para o usuário administrador. Para suprir essa demanda de usabilidade, o editor padrão do Wagtail foi substituído pelo TinyMCE.

A integração foi viabilizada por meio de uma aplicação Django customizada, `wagtail_tinymce`, desenvolvida especificamente para este projeto. Esta aplicação utiliza os *hooks* do Wagtail para injetar os *scripts* necessários e substituir o *widget* padrão do campo `RichTextField`.

Essa modificação proporcionou aos editores de conteúdo uma barra de ferramentas mais completa e familiar, com funcionalidades avançadas, como a criação de tabelas e formatações de texto mais granulares. Esse aprimoramento resulta em uma arquitetura de edição balanceada, que combina a flexibilidade do *StreamField* em páginas complexas com o poder de um editor de texto avançado em contextos onde este é mais adequado.

Apesar de sua implementação, o uso do modelo `TextPage` é desencorajado por duas razões principais: uma técnica e uma filosófica.

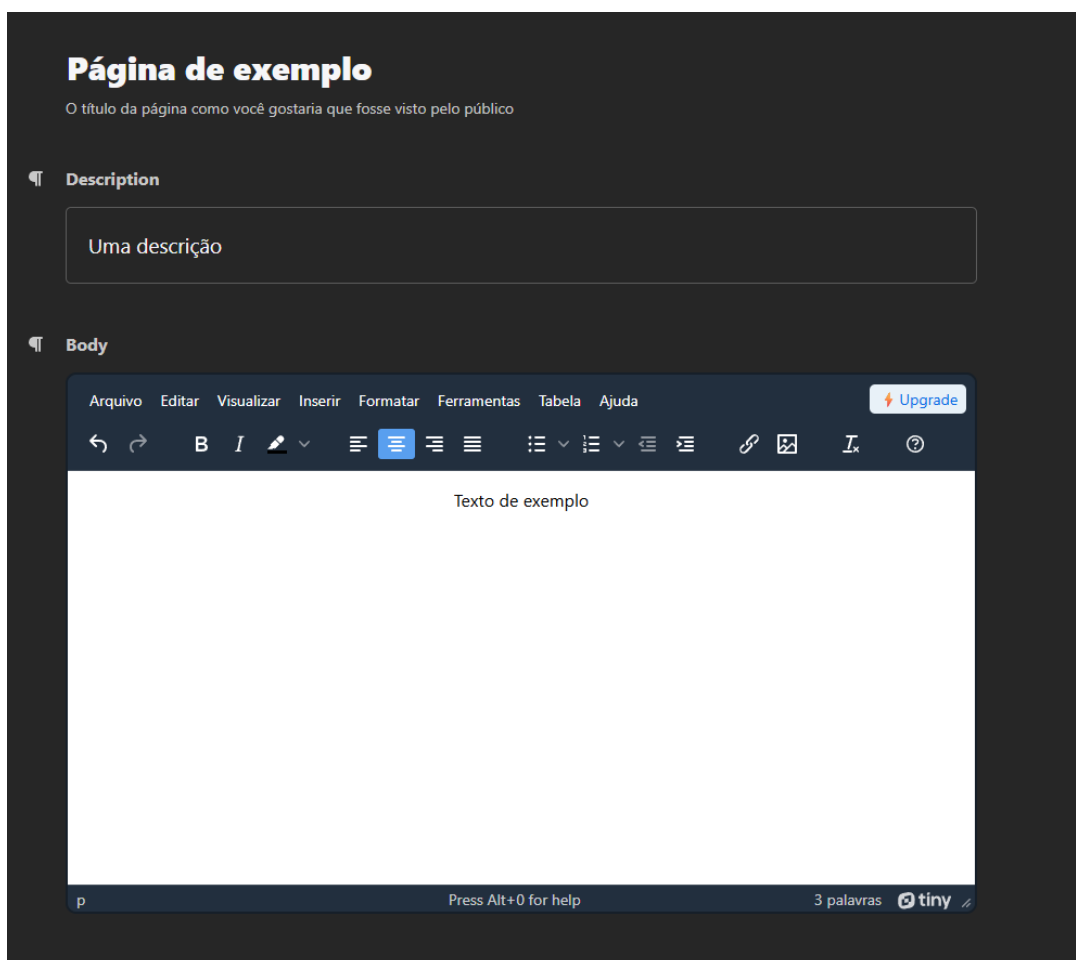
Primeiramente, do ponto de vista técnico, o editor TinyMCE, embora potente, pode gerar HTML não semântico. Isso significa que a marcação produzida pode não ter o valor semântico adequado para tecnologias assistivas, como leitores de tela, ou para a indexação por mecanismos de busca. Isso foi constatado durante testes com tabelas, nos quais o editor gerou células de cabeçalho com a tag `<td>` em vez da semanticamente correta `<th>`. Embora o

TinyMCE incluía uma interface de edição de código-fonte que possibilita o ajuste manual, essa funcionalidade representa um entrave para administradores sem capacidade técnica para edição de código.

Em segundo lugar, do ponto de vista filosófico, o uso de editores avançados como o TinyMCE pode ser considerado uma contradição ao "*Zen of Wagtail*". Esta é a filosofia de *design* que guia o desenvolvimento do CMS, e um de seus princípios fundamentais envolve a definição clara de responsabilidades. Segundo essa filosofia, não é papel do editor de conteúdo decidir sobre como o conteúdo deve ser apresentado; sua responsabilidade é apenas prover o conteúdo (Wagtail, 2025).

Ainda assim, a TextPage com TinyMCE (Fig. 20) foi mantida como uma alternativa para casos em que *StreamField* e o editor de texto rico padrão não atendam às necessidades específicas do conteúdo.

Figura 20. Edição da TextPage com TinyMCE



Fonte: Elaborado pelo autor

Por fim, a arquitetura de informação do projeto exigia elementos puramente estruturais, além das páginas de conteúdo, para organizar a hierarquia do site de forma lógica.

Para esse fim, foi implementado o modelo `FolderPage`, que funciona como um "contentor" ou "pasta" na árvore de páginas do Wagtail, agrupando páginas relacionadas sob um mesmo caminho de URL (ex: `/cursos/`).

Diferentemente de outros tipos de página, a `FolderPage` não possui conteúdo próprio para exibição. Para evitar que o usuário se depare com uma página vazia e para melhorar a experiência de navegação, o modelo inclui um campo-chave que permite ao administrador do site selecionar uma página filha como visualização padrão. Dessa forma, quando um usuário tenta visualizar diretamente uma `FolderPage`, o sistema mostra automaticamente o conteúdo da página filha designada, garantindo uma navegação fluida e intuitiva.

4.2.6 *Arquitetura de implantação*

A arquitetura de implantação, projetada para garantir que a solução seja portátil, escalável e de fácil manutenção, foi efetivamente implementada na infraestrutura do Decom. Esta seção detalha o processo e a arquitetura de implantação, que utiliza as ferramentas de containerização e *proxy* reverso listadas nos materiais (**seção 4.2.1**).

A solução é totalmente containerizada usando Docker. Como evidenciado pelos arquivos *Dockerfile* presentes nos repositórios do *backend* e do *frontend*, cada aplicação (o *frontend* Next.js e o *backend* Wagtail) é encapsulada em seu próprio contêiner. Isso garante um ambiente de execução isolado e consistente, eliminando problemas de dependência entre o servidor e a aplicação. Além disso, a infraestrutura também conta com banco de dados e *proxy* reverso containerizados.

Considerando que a maior parte da inserção de conteúdo foi realizada em ambiente de desenvolvimento, é necessário fazer a migração do banco de dados para o ambiente de produção. Para isso, gera-se um arquivo de *dump* do banco de dados de desenvolvimento usando o comando `pg_dump -U [usuario] -F p [nome_do_banco] > [nome_do_arquivo].sql` no terminal do container correspondente e, após a inicialização do contêiner de banco de dados em produção, o arquivo deve ser usado para populá-lo ao utilizar o comando `psql -U [usuario] [novo_banco_de_dados] < [nome_do_arquivo].sql`. Importante notar que para que o processo funcione, o banco de dados em produção deve estar vazio e nenhuma migração deve ter sido aplicada pelo *backend*, ou seja, o container do banco de dados deve ser iniciado individualmente, e somente após a migração do banco o restante dos containers podem ser iniciados. Os arquivos de mídia do *site* também precisam ser migrados para o container de

backend em produção, iniciando-o individualmente para possibilitar o acesso ao sistema de arquivos do container.

O Nginx, por sua vez, assume o papel de *Proxy* Reverso, conforme ilustrado no diagrama de arquitetura geral (**Figura 5**). Em um cenário de produção, esses contêineres (*frontend*, *backend* e *Nginx*) são orquestrados por docker-compose.

Nessa arquitetura, o Nginx atua como o ponto de entrada único para todo o tráfego do site. Ele é configurado para servir o *frontend*, direcionando requisições pela raiz do *site* (ex: /, /cursos, /noticias) para o contêiner do Next.js. O Nginx também é responsável por servir eficientemente os arquivos estáticos do *frontend* (CSS, JavaScript, imagens) que são gerados no build da aplicação. Concomitantemente, ele é configurado para rotear para o *backend* as requisições destinadas a caminhos específicos, como /api/ (para a API de conteúdo do Wagtail), /admin/ (para o painel administrativo) e /media/ (para os arquivos e imagens enviados pelos usuários), encaminhando-as diretamente para o contêiner do Wagtail.

Essa abordagem implementa o modelo de arquitetura desacoplada em nível de infraestrutura, além de também oferecer benefícios de segurança (ocultando os servidores de aplicação da internet pública) e performance (permitindo o cache de respostas estáticas pelo Nginx).

5 RESULTADOS E VALIDAÇÃO

Este capítulo apresenta os resultados obtidos com o desenvolvimento do MVP e detalha o subsequente processo de avaliação da solução. O foco é demonstrar, por meio de análises técnicas e qualitativas, que os objetivos propostos na seção 1.2 deste trabalho foram alcançados. A validação foi estruturada em três pilares principais. O primeiro foi a verificação da aderência ao DSGov 3, analisando a conformidade visual e estrutural com o Padrão Digital de Governo. O segundo pilar consistiu na validação técnica, focada em performance e otimização, que incluiu a análise de métricas de desempenho (*Web Vitals*), acessibilidade, responsividade e compatibilidade, em resposta direta ao objetivo específico do projeto. Por fim, o terceiro pilar compreendeu a validação metodológica e funcional, baseada na análise do *feedback* dos stakeholders sobre a funcionalidade do MVP.

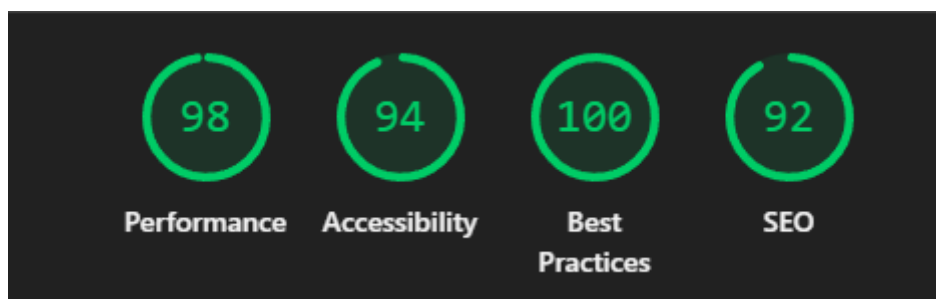
5.1 Validação de aderência ao DSGov 3

Um dos principais objetivos deste trabalho era garantir o alinhamento do novo *site* ao Padrão Digital de Governo. A validação dessa aderência foi conduzida por meio de inspeção visual e análise estrutural dos componentes. A inspeção visual confirmou que os elementos globais da interface, como o cabeçalho (*Header*), o rodapé (*Footer*) e o menu, estão em plena conformidade com as diretrizes do DSGov 3. Da mesma forma, a paleta de cores, a tipografia (*design tokens*) e os espaçamentos seguem o padrão, garantindo a identidade visual governamental esperada. Adicionalmente, a análise de componentes demonstrou que os elementos customizados — como o Carrossel, a Listagem de Notícias e o Acesso Rápido (descritos na **seção 4.2.4**) — apresentam variações em relação ao DSGov, pois foram desenvolvidos especificamente para este projeto por falta de implementação padrão da biblioteca `@govbr-ds/react-components`. Entretanto, a coesão visual e funcional da aplicação, como um todo, permanece satisfatória. O alinhamento do site às diretrizes do DSGov, então, pode ser considerada um sucesso mesmo com as variações implementadas em componentes customizados.

5.2 Validação técnica

Em linha com os objetivos específicos do projeto, foi realizada uma validação técnica focada na eficiência do MVP. Para esta análise, foram utilizados relatórios automatizados gerados pela ferramenta *Google Lighthouse*, que avalia o desempenho com base nos *Web Vitals*, explicados anteriormente, além de métricas de acessibilidade, boas práticas de otimização técnica e SEO.

Figura 21. Resultados do *Google Lighthouse* para o MVP



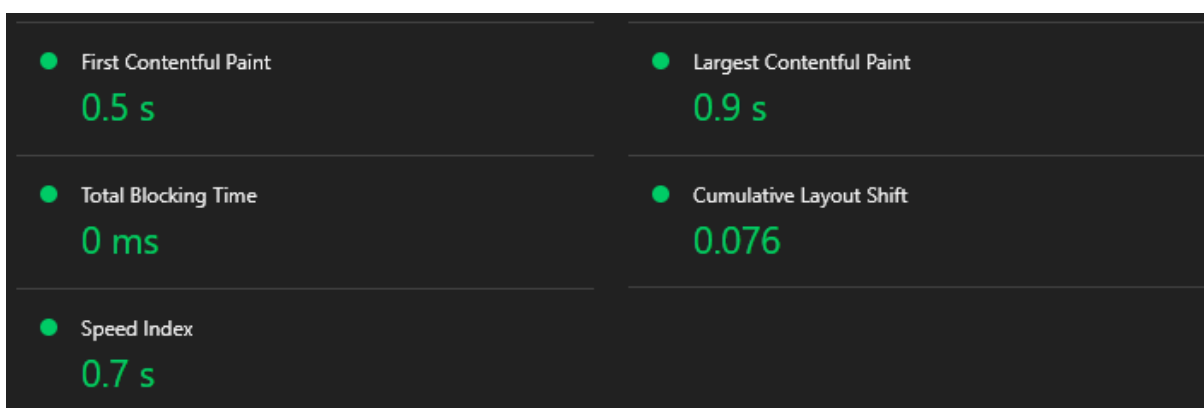
Fonte: Elaborado pelo autor

Em linha com os objetivos específicos, foi realizada uma validação técnica focada na eficiência do MVP. Para esta análise, foram utilizados relatórios automatizados gerados pela

ferramenta *Google Lighthouse*, que avalia o desempenho, acessibilidade, boas práticas e SEO (*Web Vitals*). Os testes foram executados na página inicial em ambiente de desenvolvimento, simulando condições de acesso *desktop*, após um ciclo de otimização focado nos problemas de *layout* identificados anteriormente.

Os resultados demonstraram uma melhoria substancial. A pontuação de performance (98%) é considerada excelente e atesta que os desafios técnicos impostos pela biblioteca *@govbr-ds/react-components* (detalhados na **seção 4.2.4**) foram superados com sucesso. O principal destaque desta otimização foi a mitigação bem-sucedida do *Cumulative Layout Shift* (CLS), que foi reduzido de um problemático 0.687 para 0.076. Este novo valor é classificado como "Bom" pelos padrões do *Web Vitals* (abaixo de 0.1) e valida o sucesso das correções (Google, 2025).

Figura 22. Pontuações das métricas de performance para o MVP



Fonte: Elaborado pelo autor

A análise anterior havia identificado que os “saltos” no *layout* eram causados pelo descasamento de tamanho entre os *skeletons* e os componentes finais (Carrossel e Coleção de Notícias), além de imagens sem dimensões explícitas. A correção envolveu o ajuste dos *placeholders* para que tivessem dimensões fixas (como *min-height* ou *aspect-ratio*) similares às dos componentes dinâmicos que os substituíam. Esta correção valida plenamente a estratégia de *skeleton* (descrita na **seção 4.2.4**) como a solução correta para os desafios de hidratação e CLS. As demais métricas, como *Largest Contentful Paint* (LCP) de 0.9 segundos e *Total Blocking Time* (TBT) de 0ms, demonstram excelente carregamento e responsividade à interação.

Nas demais categorias, os resultados mantiveram-se excelentes: Acessibilidade (94%), validando o alinhamento ao PDG e o uso dos componentes DSGov (e-MAG, ARIA); Boas

Práticas (100%), demonstrando a qualidade técnica da arquitetura; e SEO (92%), confirmando a eficácia da geração de meta *tags* dinâmicas (seção 4.2.2.3).

Finalmente, a verificação foi complementada por testes manuais de Responsividade e Compatibilidade, que constataram a adaptação fluida do *layout* e o correto funcionamento dos componentes customizados em dispositivos de toque.

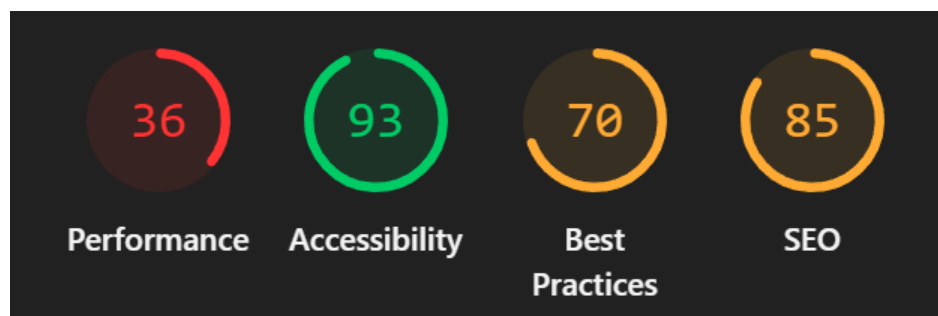
Figura 23. Página inicial do MVP em *mobile*



Fonte: Elaborado pelo autor

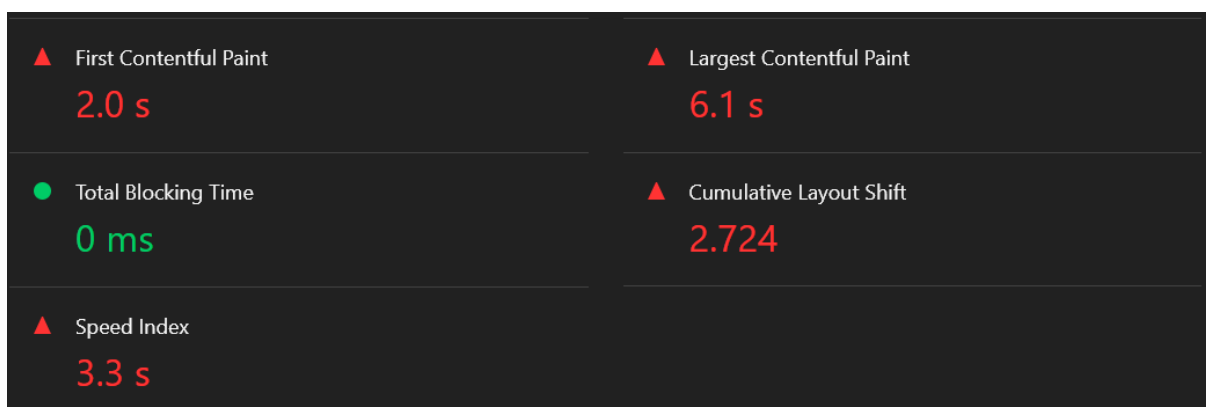
Para fornecer um contexto claro sobre o impacto da modernização proposta, também foram realizados os testes para a página inicial do Portal Institucional da UFVJM conforme a seguir.

Figura 24. Resultados do *Google Lighthouse* para o Portal Institucional da UFVJM



Fonte: Elaborado pelo autor

Figura 25. Pontuações das métricas de performance para o Portal Institucional da UFVJM



Fonte: Elaborado pelo autor

O Portal Institucional da UFVJM (**Figura 24**), que utiliza a arquitetura anterior baseada em Plone e IDG 2.0 (Alves, 2017) obteve uma pontuação de performance de 36%, indicando grandes gargalos. Essa performance é explicada pelas métricas de *Web Vitals* (**Figura 25**), que mostram um LCP de 6.1 segundos e, notavelmente, um CLS de 2.724. Este valor de CLS é classificado como "Ruim" (acima de 0.25) e indica uma instabilidade visual severa, prejudicando a experiência do usuário.

Em nítido contraste, a análise dos resultados do MVP proposto neste trabalho (**Figura 21**) demonstra a superioridade técnica da nova arquitetura. O MVP alcançou uma pontuação de performance de 98%, um resultado excelente que atesta a eficácia da solução e a superação dos desafios técnicos encontrados (detalhados na **seção 4.2.4**).

5.3 Validação metodológica e *feedback* dos Stakeholders

Conforme a metodologia D[AD]I Readaptada (seção 3.2), o processo de desenvolvimento incluiu validação com os *stakeholders* do projeto do novo site do Decom. Neste projeto, a validação com os *stakeholders* do Decom se concentrou em uma reunião formal realizada após a conclusão do MVP.

Nesta reunião, foi apresentado o protótipo funcional do novo *site*. A avaliação geral foi positiva, com os *stakeholders* validando a aderência da solução aos requisitos levantados na fase de definição e à proposta visual da fase de arquitetura e desenho. Aspectos como a nova arquitetura da informação (**Figura 3**), a interface alinhada ao DSGov 3 e a clareza geral da navegação foram bem recebidos.

As sugestões fornecidas durante a reunião concentraram-se principalmente em ajustes e refinamentos de conteúdo, como a atualização de textos e imagens específicas, sem demandar alterações funcionais significativas no MVP.

6 CONCLUSÃO

Este trabalho propôs o desenvolvimento de um Mínimo Produto Viável (MVP) para o novo *site* do Departamento de Computação (Decom) da UFVJM. O objetivo central foi modernizar a presença digital do departamento, alinhando-a à versão 3 do Padrão Digital de Governo (DSGov 3) e utilizando tecnologias modernas que garantissem performance, flexibilidade e manutenibilidade.

Para alcançar este objetivo, foi adotada a metodologia DADI Readaptada, que se mostrou eficaz para o escopo do projeto, permitindo um levantamento de requisitos focado e uma fase de planejamento e arquitetura ágil. A solução técnica implementou uma arquitetura desacoplada (headless), utilizando o CMS Wagtail no backend e um frontend em Next.js (seção 4.2.2).

Os resultados apresentados no Capítulo 5 validaram o sucesso da proposta. A aderência ao DSGov 3 foi confirmada (seção 5.1) e a validação técnica (seção 5.2) demonstrou a excelência do MVP, com pontuações de performance de 91%, acessibilidade de 94% e, notavelmente, um Cumulative Layout Shift (CLS) de 0.076, comprovando a eficácia das estratégias de otimização (seção 4.2.4). O feedback positivo dos stakeholders (seção 5.3) validou a nova arquitetura da informação e o alinhamento da solução às expectativas do departamento.

Com base nisso, pode-se afirmar que os objetivos específicos do trabalho foram alcançados. O estudo aprofundado do DSGov 3 (**seções 2.3 e 4.2.4**) e a compreensão da arquitetura Wagtail/Next.js (**seção 4.2.2**) fundamentaram a implementação. O ambiente de arquitetura desacoplada foi configurado com sucesso utilizando Docker e Nginx (**seção 4.2.6**), sobre o qual o *design* alinhado ao DSGov 3 foi implementado e validado (**seção 5.1**). Finalmente, a própria validação da solução (**seção 5.2**) confirmou a eficácia do projeto ao apresentar métricas de *Web Vitals* excelentes. Os resultados positivos do projeto validaram o MVP, desde sua concepção até a implantação final.

Como trabalhos futuros, propõe-se a implantação do site em uma infraestrutura de produção, seguida da migração completa do conteúdo legado, especialmente o histórico de Trabalhos de Conclusão de Curso e documentos. Paralelamente, será necessário realizar a capacitação dos servidores e membros do Decom para a gestão de conteúdo no Wagtail. A longo prazo, recomenda-se a evolução e manutenção contínua, com a implementação de novas funcionalidades não contempladas no MVP e a eventual migração para futuras versões do DSGov (como a 4.0, quando estável).

Este trabalho contribui para a modernização digital da UFVJM, oferecendo uma solução robusta, performática e alinhada aos padrões nacionais, que pode servir de referência para a atualização de outros portais institucionais.

REFERÊNCIAS

ALVES, E. C. Reconstrução do portal institucional da UFVJM. 2017. Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação) – Universidade Federal dos Vales do Jequitinhonha e Mucuri, Diamantina, 2017.

ASPÉLI, M. Plone 4 professional development: build robust, content-centric web applications with Plone 4. Birmingham: Packt Publishing, 2011.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. The Unified Modeling Language User Guide. Reading, MA: Addison-Wesley, 1998.

BRASIL. Modelo de Acessibilidade em Governo Eletrônico. Brasília, 2025. Disponível em: <https://www.gov.br/governodigital/pt-br/acessibilidade-e-usuario/acessibilidade-digital/modelo-de-acessibilidade>. Acesso em: 27 jun. 2025.

BRASIL. Padrão Digital de Governo. Brasília, 2025. Disponível em: <https://www.gov.br/ds/introducao/sobre>. Acesso em: 27 jun. 2025.

BRASIL. Portaria nº 7.508, de 22 de novembro de 2022. Diário Oficial da União: seção 1, Brasília, DF, 2022.

DJANGO SOFTWARE FOUNDATION. About the Django Software Foundation. Disponível em: <https://www.djangoproject.com/foundation>. Acesso em: 26 ago. 2025.

FROST, B. Atomic Design. [S.l.]: CreateSpace Independent Publishing Platform, 2016.

GARRETT, J. J. The elements of user experience: user-centered design for the web and beyond. 2. ed. Berkeley: New Riders, 2011.

GOMES, A. A. da S. Proposta de uma nova interface do Currículo Lattes a partir de padrões de interfaces do Design System do Governo Federal Brasileiro. 2023. Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação) – Universidade Federal do Ceará, Fortaleza, 2023.

HOLOVATY, A.; KAPLAN-MOSS, J. The definitive guide to Django: Web development done right. 2. ed. New York: Apress, 2009.

HOSTINGER. Os 10 problemas de segurança WordPress mais comuns e como resolvê-los. Disponível em: https://www.hostinger.com/br/tutoriais/problemas-de-seguranca-wordpress/#4_Plugins_Temas_e_Software_Desatualizados. Acesso em: 26 ago. 2025.

KHOLMATOVA, A. Design systems. New York: Smashing Magazine, 2017.

LYNCH, P. J.; HORTON, S. Web style guide: basic design principles for creating web sites. 4. ed. New Haven: Yale University Press, 2016.

MDN WEB DOCS. Cumulative Layout Shift (CLS) - Glossary | MDN. Disponível em: <https://developer.mozilla.org/en-US/docs/Glossary/CLS>.

META. React Versions – React. Disponível em: <https://react.dev/versions#initial-commit>. Acesso em: 26 ago. 2025.

NEXT.JS. Rendering: Server-Side Rendering. Disponível em: <https://nextjs.org/docs/pages/building-your-application/rendering/server-side-rendering>. Acesso em: 26 ago. 2025.

OLIVEIRA, I. J. S. Implementação de padrões de acessibilidade no portal institucional da UFVJM: aplicabilidade das recomendações do Modelo de Acessibilidade em Governo Eletrônico integrado às diretrizes da Identidade Padrão de Comunicação Digital do Poder Executivo Federal. 2021. Trabalho de Conclusão de Curso (Graduação em Sistemas de Informação) – Universidade Federal dos Vales do Jequitinhonha e Mucuri, Diamantina, 2021.

ORACLE. O que é um sistema de gerenciamento de conteúdo (CMS)? 2025. Disponível em: <https://www.oracle.com/br/content-management/what-is-cms/>. Acesso em: 27 jun. 2025.

PLONE. Plone 6.1.1 released. Por: REES, Maurits Van. 2025. Disponível em: <https://community.plone.org/t/plone-6-1-1-released/21627>. Acesso em: 27 jun. 2025.

PRESSMAN, Roger S. Software Engineering: A Practitioner's Approach. 7. ed. New York: McGraw-Hill, 2010.

PRISMA. What is an ORM – Object-Relational Mapper. Disponível em: <https://www.prisma.io/dataguide/types/relational/what-is-an-orm#what-is-an-orm>. Acesso em: 26 ago. 2025.

SCHWABER, Ken; SUTHERLAND, Jeff. The Scrum Guide: The Definitive Guide to Scrum: The Rules of the Game. [S.l.]: Scrum.org, 2020. Disponível em: <https://scrumguides.org/docs/scrumguide/v2020/2020-Scrum-Guide-US.pdf>. Acesso em: 24 out. 2025.

SEBRAE. O que é Design System. 2022. Disponível em: <https://sebrae.com.br/sites/PortalSebrae/artigos/o-que-e-design-system,2e9ef4498f6a2810VgnVCM100000d701210aRCRD>. Acesso em: 27 jun. 2025.

SECOM – SECRETARIA DE COMUNICAÇÃO SOCIAL DA PRESIDÊNCIA DA REPÚBLICA. Pesquisa Brasileira de Mídias. Brasília: Secom/PR, 2016.

SECOM – SECRETARIA DE COMUNICAÇÃO SOCIAL DA PRESIDÊNCIA DA REPÚBLICA. Pesquisa Telefônica Regular – Rodada 27. Brasília: Secom/PR, 2022.

SERPRO – SERVIÇO FEDERAL DE PROCESSAMENTO DE DADOS. Padrão Digital de Governo. Brasília, 2025. Disponível em: <https://next-ds.estaleiro.serpro.gov.br/introducao/sobre>. Acesso em: 27 jun. 2025.

STATE OF JAVASCRIPT. State of JavaScript 2024. Disponível em: <https://stateofjs.com/en-US>. Acesso em: 26 ago. 2025.

TANENBAUM, Andrew S.; WETHERALL, David. Redes de computadores. 5. ed. São Paulo: Pearson Prentice Hall, 2011.

UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI. Departamento de Computação – Decom. Disponível em: <http://www.decom.ufvjm.edu.br/dc2020/>. Acesso em: 26 ago. 2025.

WAGTAIL. Create a portfolio page. Wagtail Documentation. 2025. Disponível em: https://docs.wagtail.org/en/stable/tutorial/create_portfolio_page.html. Acesso em: 24 out. 2025.

WAGTAIL. The Zen of Wagtail. Wagtail Documentation, v6.1.3. 2025. Disponível em: https://docs.wagtail.org/en/v6.1.3/getting_started/the_zen_of_wagtail.html. Acesso em: 24 out. 2025.

APÊNDICES

APÊNDICE A - IDENTIFICAÇÃO DE CONTEÚDO LEGADO

Nome	Formato	Descrição	Destino (não preencher)
DECOM			
• I_ Página Inicial	Página	Notícias e Agenda	
• I_ Cursos	Página	Página do Curso de Sistemas de Informação	
• I_ Colegiado do Curso de Sistemas de Informação	Página	Página apresentando o Colegiado do Curso de Sistemas de Informação	
• I_ UFVIM	Link externo	Link para o site da UFVIM	
• I_ Portaria FACET nº 001/2025, de 28 de fevereiro de 2025	Página	Arquivo da Portaria	
• I_ Documentos	Página	Página que contém arquivos PDF dos diversos documentos relacionados às atividades do Decom	
• I_ Projeto pedagógico do Curso de Sistemas de Informação – 2008	Arquivo	Arquivo do Projeto pedagógico	
• I_ Diretrizes para a defesa pública de trabalho de conclusão de curso (tcc) no curso de sistemas de informação	Arquivo	Arquivo contendo informações sobre a defesa de TCC no curso de Sistemas de Informação	
• I_ Regimento do Colegiado do Curso de Sistemas de Informação	Arquivo	Arquivo do Regimento do Colegiado do Curso de Sistemas de Informação	
• I_ Regulamento dos cursos de graduação da UFVIM	Link externo	Link para página da Prograd	
• I_ Manual de normalização: MONOGRAFIAS, DISSERTAÇÕES E TESES – 3ª ed.	Link externo	Link para o site da UFVIM (Link quebrado)	
• I_ Normas de Utilização dos laboratórios do Departamento de Computação	Arquivo	Arquivo de normas de utilização dos laboratórios do Decom	
• I_ Formulário de solicitação para servidores externos ao DECOM	Arquivo	Arquivo do formulário de requisição de uso de Laboratório por servidores externos ao Decom	
• I_ Editais de professor voluntário, monitoria e representantes do colegiado	Arquivo	Arquivos dos editais de professor voluntário, monitoria e representantes do colegiado	
• I_ Servidores			
• I_ Administrativos	Página	Página informativa sobre os servidores técnicos administrativo do Decom	
• I_ Docentes	Página	Página informativa sobre os servidores docentes do Decom	
• I_ Estágios	Página	Página sobre estágio no curso de Sistemas de Informação	
• I_ Fluxograma	Arquivo	Arquivo com fluxograma do processo de registro de estágio obrigatório	
• I_ Passo a passo descritivo para a formalização do Estágio	Arquivo	Arquivo com passo a passo para formalização de estágio obrigatório	
• I_ Termo de Aceite de Orientação	Arquivo	Arquivo do termo de aceite do professor orientador	
• I_ Plano de Atividades	Arquivo	Arquivo do Plano de Atividades	
• I_ Relatório Final contendo avaliação de Supervisor	Arquivo	Arquivo do Relatório Final	
• I_ Aproveitamento de atividade profissional	Arquivo	Arquivo de Aproveitamento de Atividade Profissional	
• I_ Vínculo de estágio EE a disciplina de ECS	Arquivo	Arquivo de Vínculo de Estágio não Obrigatório a disciplina de Estágio Obrigatório	
• I_ Termo de compromisso	Link externo	Link para o arquivo do Termo de compromisso no site da Prograd	
• I_ Orientações aos alunos sobre processo e documentos de estágio	Arquivo	Arquivo do termo de aceite do professor orientador	
• I_ Aceite do orientador de estágio	Arquivo	Arquivo do termo de aceite do professor orientador (Repete)	
• I_ Passo a passo do EEC	Arquivo	Arquivo com Orientações para realização de estágio não obrigatório	
• I_ Plano de Estágio	Arquivo	Arquivo do Plano de Atividades de estágio não obrigatório	
• I_ Relatório parcial de estágio	Arquivo	Arquivo do Relatório Semestral de Estágio	
• I_ Rescisão de estágio	Arquivo	Arquivo de Rescisão de Estágio	
• I_ Estágio presencial na pandemia Anexo I	Arquivo	Arquivo de Ciência e Concordância para realização de estágio durante a pandemia	
• I_ Estágio presencial na pandemia Anexo II	Arquivo	Arquivo de Declaração da instituição concedente para realização de estágio durante a pandemia	
• I_ Termo de compromisso	Link externo	Link para o arquivo do Termo de compromisso no site da Prograd (Repete)	
• I_ Grupos de pesquisa	Página	Página sobre os grupos de pesquisa existentes no Decom	
• I_ Trabalhos de Conclusão de Curso	Página	Página contendo todos os trabalhos de conclusão de curso já publicados desde 2013	
• I_ Contatos	Página	Página contendo os responsáveis e os contatos do Decom, Coordenação do curso de Sistemas e Secretária do Decom	

APÊNDICE B - BENCHMARKING

Nº	Instituição	URL	CMS*	IDG*	URL do conteúdo de interesse	Benchmarking	Motivo de interesse
1	UNIVERSIDADE FEDERAL DE MATO GROSSO (UFMT)	https://www.ufmt.br/	Outros	Não	https://www.ufmt.br/instituto-faculdade/facc	Positivo	Histórico; Cursos de extensão; Página da Atleta; Acesso rápido
2	UNIVERSIDADE DE BRASÍLIA (UNB)	https://unb.br/	Joomla	Não	http://fci.unb.br	Comum	
3	UNIVERSIDADE FEDERAL DE SERGIPE (UFS)	https://www.ufs.br/	Outros	Não	https://www.sigaa.ufs.br/sigaa/public/d/epartamento/portal.jsf?_af=pt_BR&id=93	Negativo	Página do departamento no Sigaa
4	UNIVERSIDADE FEDERAL DO AMAZONAS (UFAM)	https://ufam.edu.br/	Joomla	Sim	https://icompa.ufam.edu.br/site	Comum	
5	(UFOP)	https://ufop.br/	Drupal	Não	http://www3.decom.ufop.br/decom/inicib/	Positivo	Página do colegiado
6	UNIVERSIDADE FEDERAL DE SÃO JOÃO DEL REI (UFSJ)	https://www.ufsj.edu.br/	Outros	Não	https://dcomp.ufsj.edu.br	Negativo	Site indisponível
7	UNIVERSIDADE FEDERAL DO ACRE (UFAC)	https://www.ufac.br/	Plone	Sim	http://csi.ufac.br	Comum	
8	UNIVERSIDADE FEDERAL DE PELOTAS (UFPEL)	https://portal.ufpel.edu.br/	Wordpress	Não	https://wp.ufpel.edu.br/cdtec	Comum	
9	UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO (UNIRIO)	https://www.unirio.br/	Plone	Não	https://bsi.uniriotec.br	Positivo	Página do Enade; Página de colação de grau
10	UNIVERSIDADE FEDERAL DE MATO GROSSO DO SUL (UFMS)	https://www.ufms.br/	Wordpress	Não	https://www.facom.ufms.br	Positivo	Página de oportunidades
11	UNIVERSIDADE FEDERAL DE RORAIMA (UFRR)	https://ufrr.br/	Wordpress	Sim	https://ufrr.br/igeo	Comum	
12	UNIVERSIDADE FEDERAL DO AMAPÁ (UNIFAP)	https://www.unifap.br/	Wordpress	Não	https://www2.unifap.br/dcet	Comum	
13	UNIVERSIDADE FEDERAL DE SÃO CARLOS (UFSCAR)	https://www.ufscar.br/	Plone	Não	https://www.dc.ufscar.br	Positivo	Linha do tempo
14	UNIVERSIDADE FEDERAL DO MARANHÃO (UFMA)	https://portal.badrao.ufma.br/site	Plone	Sim	-	Negativo	Não foram encontrados sites similares
15	UNIVERSIDADE FEDERAL DE LAVRAS (UFLA)	https://ufla.br/	Outros	Sim	https://dccc.ufla.br	Positivo	Página de formulários; Página de reserva de laboratórios
16	UNIVERSIDADE FEDERAL DE ITAUBÁ (UNIFEI)	https://www.unifei.edu.br/	Wordpress	Não	https://www.unifei.edu.br	Negativo	Site indisponível
17	UNIVERSIDADE FEDERAL DE CIÊNCIAS DA SAÚDE DE PORTO ALEGRE (UFCSA)	https://ufcsa.edu.br/	Joomla	Não	https://ufcsa.edu.br/estude-na-ufcsa/graduacao/conheca-os-cursos/psicologia	Comum	
18	UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ (UTFPR)	https://www.utfpr.edu.br/	Plone	Não	https://www.utfpr.edu.br/cursos/graduacao/bacharelado/ciencia-da-computacao	Comum	
19	UNIVERSIDADE FEDERAL RURAL DO RIO DE JANEIRO (UFRRJ)	https://portal.ufrrj.br/	Wordpress	Não	-	Negativo	Não foram encontrados sites similares
20	UNIVERSIDADE FEDERAL DOS VALES DO JEQUITINHONHA E MUCURI (UFVJM)	https://portal.ufvjm.edu.br/	Plone	Sim	-	Comum	Site da UFMJM
21	UNIVERSIDADE FEDERAL DO RECONCANO DA BAHIA (UFRRB)	https://ufrrb.edu.br/portal/	Joomla	Sim	https://www.ufrrb.edu.br/cetec	Positivo	Página de Empresas Jôniores
22	UNIVERSIDADE FEDERAL DE UBERLÂNDIA (UFU)	https://ufu.br/	Drupal	Sim	https://facom.ufu.br	Comum	
23	UNIVERSIDADE FEDERAL DE SÃO PAULO (UNIFESP)	https://www.unifesp.br/	Joomla	Não	-	Negativo	Não foram encontrados sites similares
24	UNIVERSIDADE FEDERAL DO ABC (UFABC)	https://www.ufabc.edu.br/	Joomla	Sim	https://bcc.ufabc.edu.br	Comum	
25	UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE (UFRRN)	https://ufrrn.br/	Wordpress	Não	https://sigaa.ufrrn.br/sigaa/public/departamento/portal.jsf?id=56	Negativo	Página do departamento no Sigaa
26	UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL (UFRGS)	http://www.ufrgs.br/ufrgs/inicial	Plone	Não	https://www.inf.ufrgs.br/site	Negativo	Sem página do departamento
27	UNIVERSIDADE FEDERAL DO PIAUÍ (UFPI)	https://www.ufpi.br/	Joomla	Sim	-	Negativo	Site indisponível
28	UNIVERSIDADE FEDERAL DO VALE DO SÃO FRANCISCO (UNIVASF)	https://portais.univasf.edu.br/	Plone	Sim	www.univasf.edu.br/~ccompa/	Comum	
29	UNIVERSIDADE FEDERAL DE CAMPINA GRANDE (UFCG)	https://portal.ufcg.edu.br/	Outros	Sim	https://www.computacao.ufcg.edu.br/graduaaduaacao	Negativo	Apenas menu de navegação lateral
30	UNIVERSIDADE FEDERAL DO OESTE DO PARÁ (UFOPA)	https://www.ufopa.edu.br/ufopa/	Outros	Não	https://leg.ufopa.edu.br/leg	Positivo	Página sobre laboratórios

59	UNIVERSIDADE FEDERAL DO OESTE DA BAHIA (UFOP)	https://ufob.edu.br/	Plone	Sim	https://ufob.edu.br/ensino/graduacao/biologia	Negativo	Página única e poucas informações
60	UNIVERSIDADE FEDERAL DO CARIRI (UFCA)	https://www.ufca.edu.br/	Wordpress	Não	https://www.ufca.edu.br/cursos/graduacao/ciencia-da-computacao		Páginas de Normas acadêmicas, Atos regulatórios e Ementas
61	UNIVERSIDADE FEDERAL DO SUL DA BAHIA (UFSB)	https://ufsb.edu.br/	Joomla	Sim	https://ufsb.edu.br/hac-sia/graduacao/licenciatura-interdisciplinar-em-matematica-e-computacao-e-suas-tecnologias	Comum	
62	UNIVERSIDADE FEDERAL RURAL DA AMAZÔNIA (UFRA)	https://novo.ufra.edu.br/	Joomla	Sim	https://icibe.ufra.edu.br/index.php?lang=pt	Comum	
63	UNIVERSIDADE DA INTEGRAÇÃO INTERNACIONAL DA LUSOFONIA AFRO-BRASILEIRA (UNILAB)	https://unilab.edu.br/	Wordpress	Não	https://unilab.edu.br/engenharia-de-computacao	Comum	
64	UNIVERSIDADE FEDERAL DE CATALÃO (UFCA)	https://catalogo.ufg.br/	Outros	Não	https://dcc.catalogo.ufg.br	Positivo	Galeria de Fotos
65	UNIVERSIDADE FEDERAL DE JATAÍ (UFJ)	https://portalufj.jatai.ufg.br/	Outros	Não	https://computacao.jatai.ufg.br	Negativo	Pouco conteúdo
66	UNIVERSIDADE FEDERAL DE RONDONÓPOLIS (UFR)	https://ufr.edu.br/	Wordpress	Sim	https://ufr.edu.br/cursos/si	Comum	
67	UNIVERSIDADE FEDERAL DO AGRESTE DE PERNAMBUCO (UFAPE)	http://ufape.edu.br/	Drupal	Não	https://ufape.edu.br/CCBCC/CAMUS	Negativo	Página única com muito texto. Site do curso indisponível
68	UNIVERSIDADE FEDERAL DO DELTA DO PARNÁIBA (UFDPA)	https://ufdpar.edu.br/	Plone	Sim	https://sigaa.ufpi.br/sigaa/public/curso/portal.js?id=74246&lc=pt_BR	Negativo	Site com informações completas no sigaa
69	UNIVERSIDADE FEDERAL DO NORTE DO TOCANTINS (UFNT)	https://ufnt.edu.br/	Wordpress	Sim	https://ufnt.edu.br/agronomia-2	Comum	
70	UNIVERSIDADE DE SÃO PAULO (USP)	https://www5.usp.br/	Wordpress	Não	https://www.ime.usp.br/dcc	Positivo	Página de próximas defesas de TCC

APÊNDICE C - DEFINIÇÃO DE PÚBLICO ALVO

Prioridade	Público	Descrição
3	ESTUDANTES EM POTENCIAL E SEU UNIVERSO	Estudantes que desejam ingressar na Universidade, seus familiares, cursinhos, colégios, pessoas e instituições envolvidas na escolha do curso, que ainda não possuem vínculo com a Instituição, portanto, ainda não conhecem sua história, sua estrutura, os cursos e serviços que oferece.
1	ESTUDANTES INTERNOS	Estudantes que já ingressaram na Instituição, já sabem parcialmente sobre o funcionamento da Universidade, sua estrutura, um pouco da sua história. Os estudantes podem ser também professores, técnicos administrativos ou colaboradores terceirizados.
2	PROFESSORES, TÉCNICOS ADMINISTRATIVOS E COMUNIDADE INTERNA	Possuem acesso a muita informação sobre a Instituição, direto com suas fontes. Os professores ainda usam o Portal como forma de divulgar seus trabalhos.
4	COMUNIDADE EM GERAL	Pessoas que possuem interesse em serviços e informações gerais sobre a UFVJM, interessados também em oportunidades de trabalho e de negócios junto à Instituição.

APÊNDICE D - DEFINIÇÃO DE EQUIPE E STAKEHOLDERS

Nome	E-Mail	Grupo	Vínculo
Alan Fernando Santos de Ávila	alanavila@ufvjm.edu.br	Stakeholder	TAE
Alessandro Vivas Andrade	alessandrovivas@ufvjm.edu.br	Stakeholder	Professor
André Luiz Covre	andre.covre@ufvjm.edu.br		Professor
Áthila Rocha Trindade	athila.trindade@ufvjm.edu.br		Professor
Caroline Queiroz Santos	caroline.queiroz@ufvjm.edu.br		Professor
Cinthya Rocha Tameirão	cinthya.tameirao@ufvjm.edu.br	Stakeholder	Professor
Claudia Beatriz Berti	claudiabberti@ufvjm.edu.br		Professor
Eduardo Pelli	pellie@ufvjm.edu.br		Professor
Erinaldo Barbosa da Silva	erinaldo.silva@ufvjm.edu.br		Professor
Evandro das Mercês Amorim	evandro.amorim@ufvjm.edu.br		TAE
George Henrique Merino Rodolfo	george.rodolfo@ufvjm.edu.br		Professor
Geruza de Fátima Tomé	geruza.sabino@ufvjm.edu.br		Professor
Kattiana Fernandes Constantino	kattiana.constantino@ufvjm.edu.br		Professor
Leonardo Lana de Carvalho	leonardolana.carvalho@ufvjm.edu.br		Professor
Luciana Pereira de Assis	lpassis@ufvjm.edu.br		Professor
Marcelo Ferreira Rego	marcelofr@ufvjm.edu.br		Professor
Marcus Vinicius Carvalho Guelpele	marcus.guelpele@ufvjm.edu.br		Professor
Oscar Keiji Eguchi	oscar.eguchi@ufvjm.edu.br		TAE
Rafael Santin	rafael.santin@ufvjm.edu.br		Professor

APÊNDICE E - REPOSITÓRIO COM CÓDIGO FONTE

Repositório DecomDSGov no Github. Disponível em:

<https://github.com/Raul772/DecomDSGov>

