

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Carlos Augusto Dantas Marquez

**Aspectos de usabilidade no Sistema SisFlow:
uma aplicação Web auxiliar para controle de
fluxo de estágios na FACOM**

Uberlândia, Brasil

2024

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Carlos Augusto Dantas Marquez

**Aspectos de usabilidade no Sistema SisFlow: uma
aplicação Web auxiliar para controle de fluxo de estágios
na FACOM**

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Sistemas de Informação.

Orientador: Renan Gonçalves Cattelan

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Sistemas de Informação

Uberlândia, Brasil

2024

Carlos Augusto Dantas Marquez

Aspectos de usabilidade no Sistema SisFlow: uma aplicação Web auxiliar para controle de fluxo de estágios na FACOM

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Sistemas de Informação.

Trabalho aprovado. Uberlândia, Brasil, 25 de novembro de 2024:

Renan Gonçalves Cattelan
Orientador

Prof. Dr. Alexsandro Santos Soares
Membro da Banca

Prof. Dr. Ronaldo Castro de Oliveira
Membro da Banca

Uberlândia, Brasil
2024

Dedico a conclusão de mais uma etapa da minha vida aos meus pais, que tanto me apoiaram durante a realização do curso e não mediram esforços para me auxiliar. Aos meus familiares e amigos de longa data, que sempre acreditaram em mim; a todas as pessoas incríveis que conheci nessa jornada pela UFU; aos docentes da FACOM, que tanto nos ensinaram ao longo dos anos de curso; a todos os professores que já fizeram parte da minha história, desde a pré-escola até hoje; a todos que já se foram desta vida; e finalmente, a mim mesmo, por não ter desistido de um ensino público, democrático e de qualidade.

Resumo

A criação de sistemas que automatizem o fluxo de estágio em universidades tem sido tema de muitos trabalhos técnicos. Este trabalho tem como objetivo dar continuidade e evoluir a proposta do Sistema SisFlow, que é basicamente uma solução Web para a automação do fluxo de estágios da Faculdade de Computação da Universidade Federal de Uberlândia. Porém, nesta nova versão do sistema, é dado foco em aspectos de Interação Humano-Computador, incorporando atributos de usabilidade e seguindo padrões e técnicas do processo de Design Centrado no Humano. Por meio do método de pesquisa exploratória inspirado na técnica de Avaliação Heurística, foi possível identificar inconsistências nas telas do sistema, que foram corrigidas e apresentadas de forma prática ao longo do texto. O sistema resultante busca, assim, melhorar a experiência do usuário e tornar os fluxos interativos mais intuitivos.

Palavras-chave: Usabilidade, Design Web, Heurísticas, Interface, Estágio.

Lista de ilustrações

Figura 1	– Diagrama de casos de uso do Sistema SisFlow (ROCHA, 2023). Pode-se observar que há 3 atores no sistema: Aluno, Professor Orientador e Coordenador de Estágio, com suas respectivas funcionalidades devidamente implementadas.	18
Figura 2	– Diagrama de transição de estados da solicitação de início de estágio (ROCHA, 2023). Note que há sete etapas até a iniciação do estágio, que envolvem os atores Aluno, Professor Orientador e Coordenador de Estágio, direta e indiretamente, a empresa contratante, representada por um Supervisor interno para prover assinatura e acompanhamento das atividades exercidas pelo estudante.	20
Figura 3	– Abstração do front-end, seus componentes e suas comunicações entre eles (ROCHA, 2023). Note que o componente de injeção <i>App</i> é responsável por gerenciar a base da <i>Single-Page Application</i>	22
Figura 4	– Diagrama de classes do back-end do Sistema SisFlow (ROCHA, 2023). A principal classe apresentada no diagrama é a do servidor Flask. . . .	24
Figura 5	– Abstração simplificada do Banco de dados, apresenta lista de tabelas agrupadas em categorias e seus respectivos relacionamentos (ROCHA, 2023).	26
Figura 6	– Primeira versão da tela inicial do sistema. Note que não há nenhum recurso de personalização, a acentuação não está correta para algumas palavras, a ordenação padrão é de acordo com a data e hora.	31
Figura 7	– Segunda versão da tela inicial do sistema. Observe que agora todos os textos estão com a devida acentuação e gramática coerente, neste momento há um filtro disponível em todos os campos da tabela.	32
Figura 8	– Versão <i>mobile</i> da tela inicial, note o agrupamento das linhas da tabela em bloco, para facilitar leitura e dar a noção de um bloco de informações pertencentes a um mesmo grupo.	33
Figura 9	– Tela de “Verificação da documentação” após padronização dos botões de “ <i>Download</i> ” e dos botões de “Voltar” à tela anterior, presentes respectivamente no canto direito e no canto superior esquerdo da página.	34
Figura 10	– Página de “Solicitação de início de estágio – concluída” após padronização nos botões de “ <i>Download</i> ” e nos botões de “Voltar” à página anterior, presentes respectivamente no canto direito e no canto superior esquerdo da página.	34

Figura 11 – Tela de “Aceite do aluno pelo orientador” na versão anterior do sistema, com os botões dispostos na seguinte ordem: “Aceitar”, “Recusar” e “Voltar”. Essa ordem original dificulta o fluxo natural de leitura, já que os usuários estão acostumados que o botão inferior à direita dê o sentido de continuidade.	35
Figura 12 – Tela de “Aceite do aluno pelo orientador” após alteração da ordem dos botões para: “Voltar”, “Recusar” e “Aceitar”. O que busca facilita o entendimento e trás uma interface mais previsível ao usuário.	35
Figura 13 – Tela de “Processo de assinaturas para início de estágio”, antes da alteração, note que o botão “Mostrar histórico” estava alinhado à esquerda, quebrando o fluxo de leitura e dando um aspecto desagradável à interface. Note também que o botão de “Voltar” estava no canto direito inferior da página, o que poderia fazer o usuário se frustrar ao quebrar a noção de continuidade padronizada na maioria dos sistemas atuais.	36
Figura 14 – Tela de “Processo de assinaturas para início de estágio” após a correção do alinhamento dos botões à direita e a inclusão do botão de “Voltar” com o ícone de seta no canto superior esquerdo da tela.	36
Figura 15 – Tela de “Solicitação de início de estágio - Verificação de documentação do aluno pela coordenação”, note que a descrição da área “Motivos” continha somente o texto “Selecione os motivos abaixo.” e uma breve explicação sobre o filtro. A interpretação do que seriam tais motivos, neste contexto, era vaga e subjetiva, carecendo de uma explicação mais elaborada.	37
Figura 16 – Tela de “Solicitação de início de estágio - Verificação de documentação do aluno pela coordenação”, note que a descrição da sessão “Motivos”, grifada em vermelho no canto superior esquerdo da figura, agora tem uma breve explicação “Selecione os motivos para eventual deferimento abaixo” e a mesma explicação sobre o funcionamento dos filtros. Essa simples alteração faz toda a diferença no entendimento da página.	38
Figura 17 – Tela de “Solicitação de início de estágio - Verificação de documentação do aluno pela coordenação”, note o espaço que os dados do aluno ocupavam na tela.	38
Figura 18 – Tela de “Solicitação de início de estágio - Verificação de documentação do aluno pela coordenação”, onde agora é possível colapsar os dados do aluno solicitante para facilitar a leitura das demais partes. Note também que, nesse ponto do sistema, os botões já estão todos padronizados.	39

Figura 19 – Tela de “Aceite de aluno pelo professor orientador”, contendo os botões “Aceitar” e “Recusar” antes da alteração cores. Note que, em relação à Figura 12 (na qual é a mesma tela mostrada), foi feita a alteração de retirar o botão “Voltar” da parte inferior direita e reposicionando, de forma padronizada, esse botão em todas as telas, na parte superior esquerda.	40
Figura 20 – Tela de “Aceite de aluno pelo professor orientador”, contendo os botões “Aceitar” e “Recusar” após a alteração cores e adição dos ícones. Note como o contraste facilita o entendimento e a noção de redundância de informações, tanto na cor, na disposição dos mesmos e também no ícone. Essa alteração tem o potencial de evitar erros e <i>missclicks</i> no uso cotidiano do sistema.	40
Figura 21 – Trechos de código responsáveis por implementar a busca condicional para o caso do Coordenador de Estágio estar logado no sistema. Observe a estrutura do código em visões, que foi previamente mencionada.	42
Figura 22 – Última versão da tela inicial do sistema, após implementar a busca para todos os campos da tabela de Solicitações.	44

Lista de tabelas

Tabela 1 – Tabela com as rotas do <i>Back-end</i> (ROCHA, 2023).	25
--	----

Lista de abreviaturas e siglas

API	<i>Application Programming Interface</i>
BCC	Bacharelado em Ciência da Computação
BSI	Bacharelado em Sistemas de Informação
FACOM	Faculdade de Computação
HTTP	<i>Hypertext Transfer Protocol</i>
IHC	Interação Humano-Computador Hypertext Preprocessor
ORM	<i>Object-Relational Mapping</i>
PHP	<i>Hypertext Preprocessor</i>
REST	<i>Representational State Transfer</i>
SPA	<i>Single-Page Application</i>
UFU	Universidade Federal de Uberlândia
UI	<i>User Interface</i>
UX	<i>User Experience</i>
UX/UI	<i>User Experience/User Interface</i>
Web	<i>World Wide Web</i>

Sumário

1	INTRODUÇÃO	11
1.1	Contextualização	11
1.2	Motivação	12
1.3	Objetivos	12
1.4	Contribuições	13
1.5	Organização do Texto	13
2	FUNDAMENTAÇÃO TEÓRICA	15
2.1	As 10 Heurísticas de Nielsen para Usabilidade de Interfaces de Usuário	15
2.2	As 113 Diretrizes de Design para Usabilidade de Homepages	17
2.3	O Sistema SisFlow	17
2.4	Arquitetura e Tecnologias Utilizadas	20
2.4.1	Front-end	21
2.4.2	Back-end	22
2.4.3	Base de dados	23
3	TRABALHOS RELACIONADOS	27
4	MÉTODO	29
5	DESENVOLVIMENTO	30
6	CONCLUSÃO	45
	REFERÊNCIAS	48

1 Introdução

1.1 Contextualização

O estágio é basicamente o primeiro contato do estudante com o mercado de trabalho em sua área de formação, um complemento significativo para o aprendizado. O aluno tem a oportunidade de aplicar conhecimentos técnicos adquiridos ao longo do curso em um ambiente profissional real, caracterizando-se pela conexão entre o ensino e a prática. O estágio pode ocorrer em empresas, organizações públicas ou privadas, e até na própria universidade, em projetos internos. Seja qual for a área escolhida para estagiar, o estágio irá proporcionar uma compreensão mais profunda das práticas, o que é difícil de se obter em uma disciplina teórica. As atividades do estágio devem estar relacionadas à Informática e ao que se espera de um egresso do curso.

São inúmeros os benefícios do estágio, tais como permitir ao aluno o desenvolvimento de habilidades interpessoais (comunicação, organização, trabalho em equipe, liderança, dentre outras), com impacto na empregabilidade, e também permite ao estudante explorar diferentes áreas de atuação dentro de sua formação acadêmica, auxiliando-o a definir, ou ao menos refinar, suas preferências profissionais antes de sua entrada formal no mercado de trabalho.

No caso da Faculdade de Computação (FACOM) da Universidade Federal de Uberlândia (UFU), o estágio pode ser obrigatório (que é uma disciplina obrigatória do curso) ou não obrigatório (que é uma atividade opcional que o aluno realiza para antecipar sua colocação no mercado de trabalho e que, além da experiência, gera horas complementares para os alunos).

O estágio supervisionado é uma atividade acadêmica de responsabilidade do Colegiado de Curso. Sua coordenação e execução são realizadas por um Coordenador de Estágio, que é indicado pelo colegiado, com um mandato de dois anos. Além do Coordenador de Estágio, também são atores o Aluno (discente regularmente matriculado no curso e que teve sua solicitação de estágio deferida), o Supervisor (interno da empresa onde o aluno realizará estágio, responsável pelo acompanhamento das atividades exercidas pelo estudante), o Professor Orientador (ou Orientador Acadêmico, necessariamente um docente da FACOM) e, por fim, o Setor de Estágios da UFU (conhecido como SESTA), que é um órgão da Diretoria de Ensino, da Pró-Reitoria de Graduação da UFU, responsável pela formalização e registro dos processos administrativos de estágios.

1.2 Motivação

Ainda no contexto acadêmico, é sabido que os professores exercem diversas tarefas em seu dia-a-dia – aulas, elaboração de material didático, correções de provas e trabalhos, dentre outras. Além dessas atividades de ensino, alguns professores ainda desempenham funções administrativas, como, por exemplo, o Coordenador de Estágio, que exerce o trabalho burocrático, muitas vezes manual, de controlar todo o fluxo de registro, acompanhamento, avaliação e conclusão dos estágios.

O Coordenador de Estágio é responsável por responder os e-mails dos alunos que fizeram uma solicitação de estágio, por verificar se tais alunos atendem aos requisitos definidos pelas normas de estágio e se, portanto, estão aptos para realizar o estágio. São também responsáveis por processar a documentação necessária, auxiliar na escolha de um Professor Orientador de estágio, por informar e gerenciar prazos e relatórios de conclusão.

Sendo assim, a criação de sistemas que automatizem o fluxo de estágio tem sido tema de muitos trabalhos técnicos. No contexto específico da FACOM, [Rocha \(2023\)](#) desenvolveu uma solução Web responsiva para a automatização das etapas de estágio a fim de tornar o fluxo de solicitação, deferimento e acompanhamento do estágio mais flexível, centralizado e facilitado para todos os atores envolvidos. A proposta de [Rocha \(2023\)](#) para o fluxo de estágios utilizou máquinas de estados com transições baseadas nas entradas do usuário e em eventos temporais para fornecer flexibilidade e manutenibilidade, a longo prazo, para futuras atualizações. O sistema resultante recebeu o nome de SisFlow.

1.3 Objetivos

O trabalho ora proposto tem por **objetivo geral** dar continuidade e evoluir a proposta do Sistema SisFlow original. Porém, com novo foco em aspectos de Interação Humano-Computador, incorporando atributos de usabilidade e seguindo padrões e técnicas do processo de Design Centrado no Humano (DCH). Pretende-se criar uma solução Web responsiva, com foco no front-end. Utilizando os conceitos de UX/UI, busca-se criar um sistema com uma interface gráfica mais intuitiva, visando automatizar procedimentos relacionados aos estágios da FACOM.

Como **objetivos específicos**, tem-se:

- Estudo do Sistema SisFlow, com entendimento de sua máquina de estados, funcionamento geral, ambiente operacional e arquitetura de software;
- Levantamento das heurísticas de design a serem aplicadas na inspeção de usabilidade;

- Inspeção de usabilidade das telas do Sistema SisFlow, com identificação de possíveis violações das heurísticas de design escolhidas e proposta de eventuais correções; e
- Documentação dos resultados.

1.4 Contribuições

Com o SisFlow em funcionamento, todo o fluxo do estágio passará a estar concentrado em uma única plataforma. Anteriormente, esse processo era manual e repetitivo, envolvendo constantemente lidar com dúvidas de alunos, assinaturas (onde muitas vezes havia re-trabalho, já que as informações e o processo como um todo não estavam tão claros aos estudantes, e a ordem correta de assinatura), autorizações e restrições de prazos. Com a conclusão deste trabalho, espera-se que o processo se tornará mais simples e direto, onde as etapas necessárias para registro, acompanhamento e conclusão dos registros de estágio poderão ser concluídas com alguns cliques e interações entre os atores do sistema. Assim, a ferramenta resultante permitirá que todo o fluxo do estágio, desde a solicitação até a conclusão e envio de relatórios, esteja centralizado por meio da Aplicação Web Auxiliar. Isso facilitará a vida tanto dos professores quanto dos alunos, pois, com as informações concentradas, eles poderão trabalhar de forma independente e mais coesa, com lembretes de prazos e documentação de mais fácil acesso.

Além dos objetivos citados acima, o resultado esperado desta monografia é aplicar conceitos vistos ao longo do curso, principalmente na disciplina de Interação Humano Computador, para descrever e mostrar na prática ao público geral como funcionam estes conceitos e como sua aplicação em um sistema pode melhorar significativamente o uso do mesmo. O estudo de usabilidade de um sistema é fundamental porque garante que ele seja fácil de entender, eficiente de usar e satisfaça as necessidades dos usuários. Quando um sistema tem usabilidade, os usuários conseguem realizar suas tarefas sem frustrações, erros são reduzidos, e a produtividade aumenta. Ainda, sistemas bem projetados promovem uma experiência positiva, que pode levar a uma maior aceitação e retenção de usuários, bem como a menores custos de treinamento e suporte, já que o sistema se torna intuitivo e com o uso corriqueiro o usuário não apresentará dificuldades significativas, ao se comparar com outro sistema que não tem estas preocupações.

1.5 Organização do Texto

O restante desta monografia está organizado da seguinte forma: no Capítulo 2, é apresentada a fundamentação teórica e técnica para a proposta; no Capítulo 3, são discutidos trabalhos relacionados; no Capítulo 4, é descrito o método utilizado no desenvolvimento do projeto; no Capítulo 5, é detalhado o desenvolvimento e seus principais

resultados; por fim, no Capítulo 6, são tecidas as considerações finais sobre o trabalho desenvolvido.

2 Fundamentação Teórica

De acordo com Nielsen ([NIELSEN, 1994b](#)), a usabilidade é essencial para o sucesso de um sistema Web, pois melhora a experiência do usuário, o que o faz se sentir mais confortável e entusiasmado ao usar o sistema. Dito isso, Nielsen desenvolveu 10 heurísticas de usabilidade fundamentais para o design e também 113 diretrizes para construção dos sites, das quais, algumas serão utilizadas ao longo da monografia. Elas servem como diretrizes fundamentais para o design de interfaces de usuário (UI) e também para avaliar a experiência do usuário (UX), seja em *websites*, aplicativos *mobile*, sistemas interativos para computador, sistemas Web.

2.1 As 10 Heurísticas de Nielsen para Usabilidade de Interfaces de Usuário

As 10 Heurísticas de Usabilidade para o Design de Interfaces de Usuário ([NIELSEN, 1994a](#)) figuram como uma das contribuições mais significativas para o campo de usabilidade de interfaces. Jakob Nielsen é pioneiro nessa área e o conjunto de heurísticas que ele definiu foi elaborado a partir da análise de uma série de interfaces de usuário, identificando problemas recorrentes que afetavam a usabilidade:

1. **Visibilidade do Estado do Sistema:** Permite que o usuário tenha um feedback visual do que está acontecendo no sistema, como uma mensagem de sucesso no salvamento de documento. Isso ajuda o mesmo a ter clareza sobre o que está acontecendo o que reduz a sensação de incerteza ao interagir com a interface.
2. **Correspondência entre o Sistema e o Mundo Real:** Consiste em usar elementos que o ser humano já conhece, como por exemplo o ícone da lixeira para excluir ou apagar algum arquivo. Isso facilita o entendimento, pois o sistema se comunica em termos que o usuário já conhece e associa à determinada função.
3. **Controle e Liberdade do Usuário:** Oferece ao usuário opções de editar informações, desfazer e corrigir erros. Isso faz o mesmo se sentir mais seguro ao explorar todo o âmbito da interface, ao se saber que tem controle para desfazer algo indesejado.
4. **Consistência e Padrões:** Mantém a disposição de botões, formulários e outros elementos visuais, a fim de criar certa intimidade entre o usuário e o sistema. Elementos semelhantes devem parecer e funcionar de maneira parecida, essa padronização facilita a navegação e também reduz a carga cognitiva do usuário.

5. **Prevenção de Erros:** Fornece ajuda onde usuários costumam errar, por exemplo, um campo validador de CPF, um campo que complete os dados de moradia ao digitar o CEP. Também podem ser mensagens de alerta de ações irreversíveis, que guiam o usuário para uma navegação mais segura e minimiza suas prováveis frustrações ao se utilizar um sistema.
6. **Reconhecimento em vez de Lembrança:** Enfatiza a importância de manter as opções, funcionalidades e caminhos visíveis e reconhecíveis para os usuários, em vez de exigir que eles se lembrem de informações anteriores. Basicamente a interface tem de ser intuitiva, para ajudar o usuário a focar em suas tarefas sem o esforço de lembrar de todos os detalhes da interface.
7. **Flexibilidade e Eficiência de Uso:** Usuários novos têm um caminho mais claro e guiado, mesmo que um pouco mais demorado, não pode deve ser confuso ou complexo. Porém, permite que usuários mais avançados tenham maior controle com atalhos, personalizações e outras opções mais avançadas.
8. **Estética de Design Minimalista:** Torna a interface visualmente atraente, limpa e prioriza o conteúdo mais relevante do sistema, permitindo que o usuário se concentre nas tarefas principais e facilita a navegação sem sobrecarregar o usuário com informações, muitas das vezes desnecessárias.
9. **Ajuda aos Usuários a Reconhecer, Diagnosticar e Recuperar-se de Erros:** Com mensagens de erro claras e orientações relevantes para ajudar os usuários a corrigirem um problema. Isso acaba a reduzir a frustração do usuário e permite uma recuperação mais rápida e eficiente.
10. **Ajuda e Documentação:** É uma forma de fornecer suporte aos usuários com uma documentação clara e acessível, orientada para a resolução de problemas comuns e específicos, na qual pode ser realizada por tutoriais, FAQs ou sistemas de ajuda integrados ao próprio sistema. Esse recurso é muito importante para garantir que o usuário tenha autonomia na resolução de dúvidas.

Todas essas heurísticas são cruciais nas fases de planejamento e desenvolvimento do sistema, porque elas oferecem um guia prático para criar interfaces que atendam às necessidades e expectativas dos usuários, melhorando a usabilidade, a acessibilidade e a satisfação geral. Ao seguir essas diretrizes, os designers podem minimizar problemas de usabilidade, reduzindo a frustração dos usuários e promovendo uma interação mais intuitiva, eficaz e agradável com o sistema.

A criação e documentação desses conceitos foi um marco fundamental para a área de usabilidade, já que passou a se ter uma estrutura sólida para avaliar, validar e melhorar

a usabilidade de sistemas de *software* e, como explicado na próxima seção, também na Web.

2.2 As 113 Diretrizes de Design para Usabilidade de Homepages

Enquanto as 10 Heurísticas de Usabilidade para o Design de Interfaces de Usuário são um conjunto de princípios gerais para avaliar a usabilidade de interfaces de usuário, as 113 Diretrizes de Design para Usabilidade de Homepages (NIELSEN, 2001) são especializadas para o domínio de Web design e fornecem recomendações mais específicas e práticas para se projetar interfaces Web. Essas diretrizes abrangem uma ampla gama de tópicos que variam desde aspectos de usabilidade e acessibilidade até questões técnicas de design.

Dentre os principais grupos de conceitos abordados estão: propósito, conteúdo, navegação, busca, design visual, tratamento de dados e convenções para design Web. Tais diretrizes foram estudadas ao longo de décadas de pesquisa e prática em usabilidade e design de interfaces, sendo publicadas no período de 1990 até o início dos anos 2000 por Jakob Nielsen e sua equipe. As diretrizes foram compiladas em várias publicações, como livros, revistas, artigos científicos e também disponibilizadas online.

Mais recentemente, avanços na tecnologia e mudanças nas expectativas dos usuários têm influenciado o desenvolvimento de diretrizes de Web design. Alguns dos avanços mais significativos incluem: o design responsivo e questões de desempenho; o design adaptativo e personalizado, com base no comportamento e preferências do usuário; a interação natural, que incluem as interfaces baseadas em gestos, voz, dentre outros; a realidade aumentada; acessibilidade avançada, que conta com a criação de novas diretrizes e padrões para garantir que os sites sejam acessíveis a todos os usuários, independentemente de suas habilidades ou limitações; e desenvolvimento sustentável, onde considerações ambientais tem se tornado uma preocupação, levando a iniciativas que reduzem o consumo de energia e os impactos ambientais associados ao desenvolvimento e uso de websites.

2.3 O Sistema SisFlow

Como já exposto no Capítulo 1, este trabalho estende o trabalho de (ROCHA, 2023). O Sistema SisFlow é uma aplicação Web que visa auxiliar o fluxo de estágio da FACOM. Em sua primeira versão, o foco foi para a arquitetura, componentes e fluxos, visando automatizar uma solução inicial e torná-la funcional. Rocha (2023) focou em desenvolver o fluxo principal do sistema, com ênfase na estruturação do back-end e da base de dados, bem como no desenho da máquina de estados correspondente.

Os fluxos de solicitação são os pontos principais da aplicação como um todo, as

operações de solicitação são mais críticas da solução, pois utilizam de todas as funcionalidades mais importantes do sistema, incluindo a máquina de estados, o agendador de eventos e o servidor de e-mails.

Na Figura 1, é representado um diagrama de casos de uso, que auxilia na descrição e entendimento do processo e do fluxo principal do sistema.

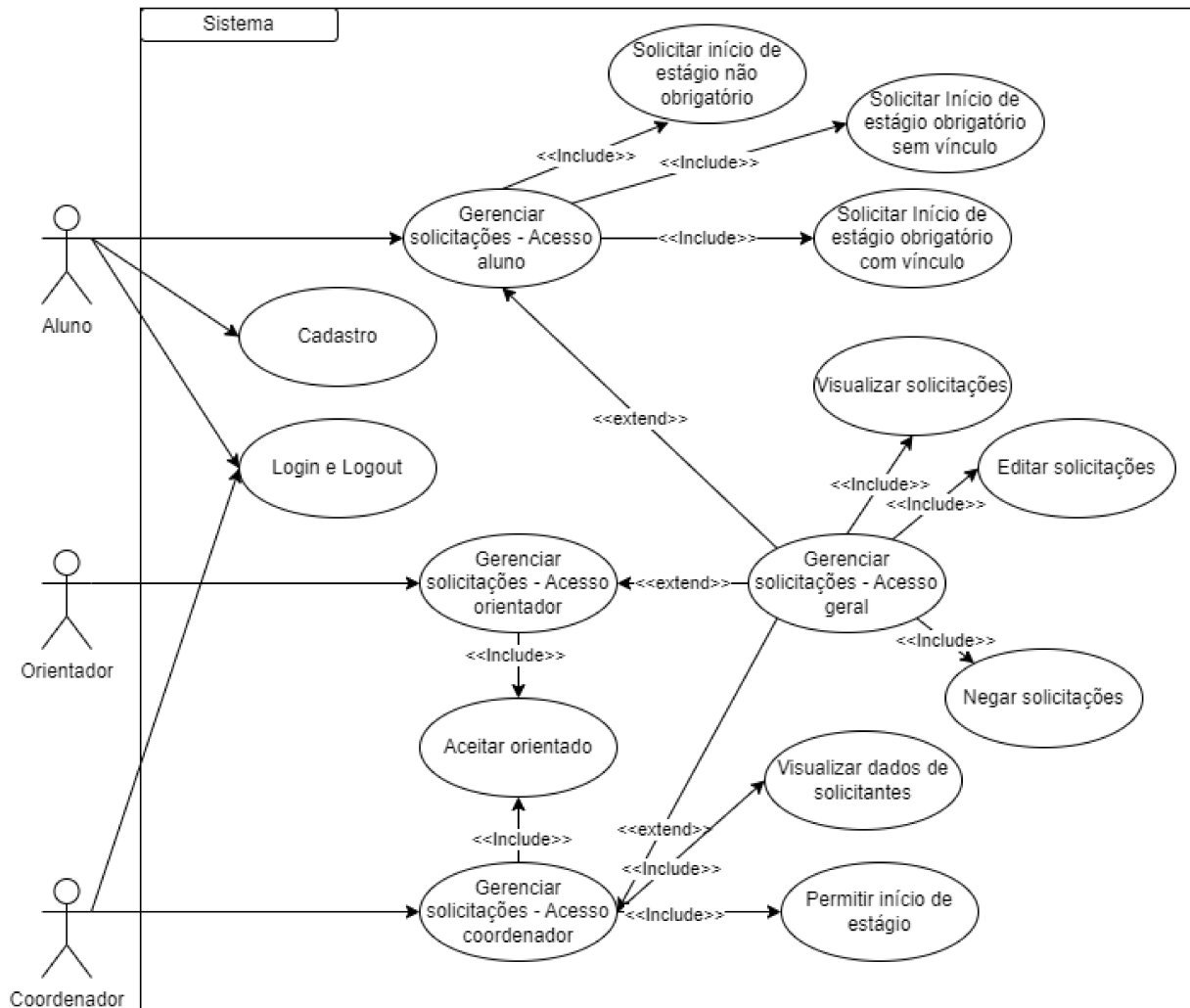


Figura 1 – Diagrama de casos de uso do Sistema SisFlow (ROCHA, 2023). Pode-se observar que há 3 atores no sistema: Aluno, Professor Orientador e Coordenador de Estágio, com suas respectivas funcionalidades devidamente implementadas.

Basicamente, o usuário inicia o fluxo ao fazer a solicitação de início de estágio no front-end, esta solicitação é enviada ao back-end através de um script de envio de requisições, a instância da classe *Server* do Flask recebe essa solicitação e a encaminha ao recurso de solicitações, que por sua vez fica responsável por fazer a formatação dos argumentos recebidos e a validação de cada argumento de acordo com o seu tipo. Com estes parâmetros já formatados, o recurso realiza uma chamada ao repositório de solicitações, que executa a leitura dos pontos relevantes e continua o processamento da solicitação.

Com todos estes dados tratados e prontos, a máquina de estados começa a trabalhar, alterando os estados de cada solicitação de acordo com o tipo de transição que foi realizada, e então é feita a leitura dos dados no novo estado. Após o carregamento do novo estado na memória, será realizado o processamento de agendamento dos próximos eventos de solicitação do usuário, considerando as suas transições que são agendadas. Após isso, os dados da solicitação persistem com seu novo estado e de possíveis novos eventos agendados. Por fim caso a transição ocorrida possua envio de e-mails associados, estes serão enviados de forma assíncrona para não impedir o retorno da solicitação do usuário.

A máquina de estados é posicionada então no fluxo de solicitações e é relativa às entradas do usuário e eventos temporais, sua abstração ocorre no banco de dados onde são categorizados na categoria de Solicitações e Transições, a partir de suas instâncias, é possível criar um fluxo no qual os usuários poderão interagir. Um ponto de destaque é o início desse fluxo, já que é o mais trabalhoso para os atores envolvidos no processo de estágios, pois ele envolve diversas etapas e eventuais deferimentos e validações de documentos, além da escolha de Professor Orientador pelo aluno, do aceite de orientação pelo docente e do processo de assinaturas.

Na Figura 2, é apresentado um diagrama de transições de estados baseado no fluxo principal de dados do sistema, o processo se inicia logo após a solicitação do aluno, evento que irá posicionar a máquina de estados em seu primeiro estado, responsável por renderizar uma página dinâmica que permite a coleta de documentação do aluno. Em seguida, essa documentação passa por uma validação pela coordenação de estágio no segundo estado, por meio de uma página manual que permite verificar a aptidão do aluno para o procedimento. Após a aprovação da documentação, a máquina avança para o terceiro estado, permitindo ao discente solicitar orientação de um dos professores cadastrados na base de dados.

Ao solicitar a orientação, a máquina irá para o seu quarto estado, que fica na espera do aceite do discente pelo Professor Orientador escolhido, após este aceite, a máquina transita para o quinto estado, onde ocorre o envio de documentação essencial para início do estágio. Após a conclusão desse processo de assinaturas, o aluno se torna apto para iniciar o estágio, caso ocorram eventuais problemas em quaisquer estados, é possível realizar cancelamentos ou indeferimentos da solicitação para o perfil de usuário específico, o que coloca a máquina em um estado de exceção.

O projeto apresentado será uma segunda versão do Sistema SisFlow, agora com foco em revisar e aperfeiçoar seu front-end com o emprego de conceitos de Interação Humano-Computador e focando em usabilidade, a fim de torná-lo um sistema mais fácil e agradável de se utilizar. As tecnologias utilizadas na primeira versão do trabalho foram mantidas, já que a primeira etapa foi feita pensando na manutenibilidade, o que tornou mais fácil dar continuidade ao desenvolvimento do sistema. Ao fim do projeto, espera-se

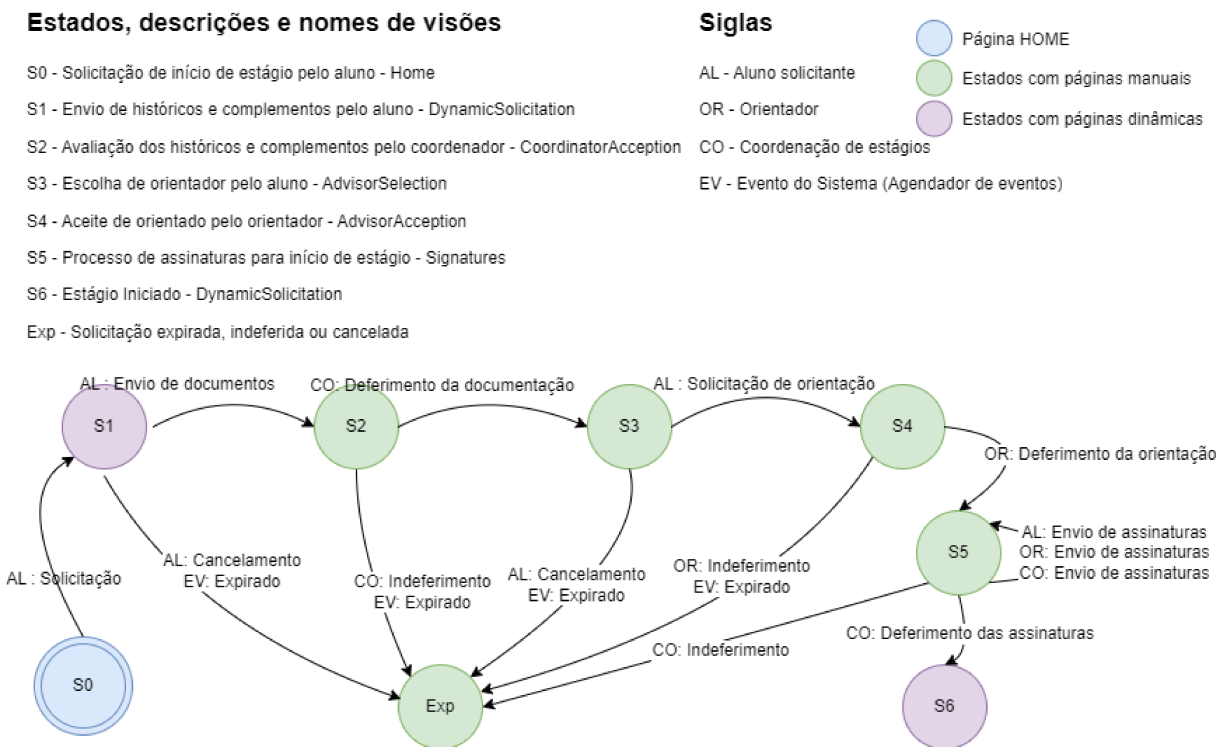


Figura 2 – Diagrama de transição de estados da solicitação de início de estágio (ROCHA, 2023). Note que há sete etapas até a iniciação do estágio, que envolvem os atores Aluno, Professor Orientador e Coordenador de Estágio, direta e indiretamente, a empresa contratante, representada por um Supervisor interno para prover assinatura e acompanhamento das atividades exercidas pelo estudante.

que esta monografia se torne um conglomerado de informações relevantes sobre o sistema em si e também uma visão mais detalhada sobre seus aspectos de usabilidade, com a aplicação prática de diferentes conceitos estudados durante o curso e com exemplos de como tais conceitos alteram o sistema na prática.

2.4 Arquitetura e Tecnologias Utilizadas

Em linhas gerais, a aplicação é arquitetada em três camadas desacopladas, sendo elas o front-end, o back-end e a base de dados. Para o back-end, foi utilizado Python (Python Software Foundation, 2024), juntamente com o framework Flask (Flask Project, 2024), o qual cria um servidor que implementa a API REST (Roy Fielding et al., 2000). Para estabelecer a conexão entre o sistema e a base de dados, utiliza-se o conjunto de ferramentas SQLAlchemy (SQLAlchemy Team, 2024), uma biblioteca que permite o mapeamento entre objetos de linguagem e instâncias das tabelas do banco de dados utilizando ORM (MDN Web Docs, 2024) (uma ferramenta de suma importância para a transformação de *Big Data* em informações úteis para os negócios). Para o front-end, foi escolhido

o framework Vue.js (Vue.js Team, 2024), pois ele facilita a criação de uma *Single Page Application* e seu subsequente reuso. Não foi necessário utilizar técnicas e métodos de prototipação, já que o sistema original já apresentava uma interface funcional. Porém, foram aplicadas técnicas para melhoria da usabilidade, seguindo regras e padrões de design, como as 10 heurísticas de Nielsen e as 113 diretrizes de Nielsen para design Web anteriormente citadas.

A seguir, serão detalhadas as principais tecnologias e ferramentas utilizadas no desenvolvimento deste trabalho.

2.4.1 Front-end

Na camada de Front-end, ou visual, estão implementadas todas as interfaces onde os usuários irão interagir com o sistema, desenvolvida utilizando o *framework* Vue.js ¹, que facilita o desenvolvimento e reuso que uma SPA oferece. O *framework* contempla, em seu conjunto de ferramentas, as tecnologias fundamentais de soluções front-end, tais como HTML (W3C, 2024b), CSS (W3C, 2024a) e JavaScript (Mozilla Developer Network (MDN), 2024), que são organizadas pelo gerenciador de pacotes npm (npm, Inc., 2024).

A estrutura de módulos e componentes disponibilizada pelo *framework* possui a maioria das dependências e módulos necessários para o desenvolvimento da aplicação. Com essa estrutura, fica mais fácil adicionar novos módulos. A aplicação foi construída a partir da junção de componentes, em um conjunto denominado “visão”. As visões, ou *views*, são renderizadas no navegador e selecionadas com base na atual rota do site, conforme definido nos conceitos de uma *Single Page Application*. Na Figura 3, é apresentada uma abstração geral da camada.

Na visão dos usuários, quando a página é alterada, existe a percepção que foi renderizada uma nova interface, porém, é uma mudança feita pelo Vue.js que renderiza novos componentes, reposicionando os antigos da mesma página com o uso de scripts da biblioteca, que por sua vez abstraem e simulam a mudança das denominadas visões, sem a necessidade do recarregamento total da página, o que evita novas requisições para o servidor back-end, isso gera uma melhora significativa no desempenho e facilita o reuso de componentes.

Além dessa construção, em todos os componentes são incluídas seções de HTML, CSS e JavaScript centralizadas, estas seções fazem parte do escopo do objeto ou das visões, que definem o seu funcionamento, incluindo a descrição de sua interface e implementação da lógica utilizada em cada uma. Este é um padrão não só deste *framework* em específico, mas de alguns outros, pois assim, facilita-se a construção de novos componentes e desaclopa as funcionalidades de uso. As funcionalidades mais gerais do sistema, são posi-

¹ O *framework* Vue.js na versão 3.2 pode ser verificado em <<https://blog.vuejs.org/>>

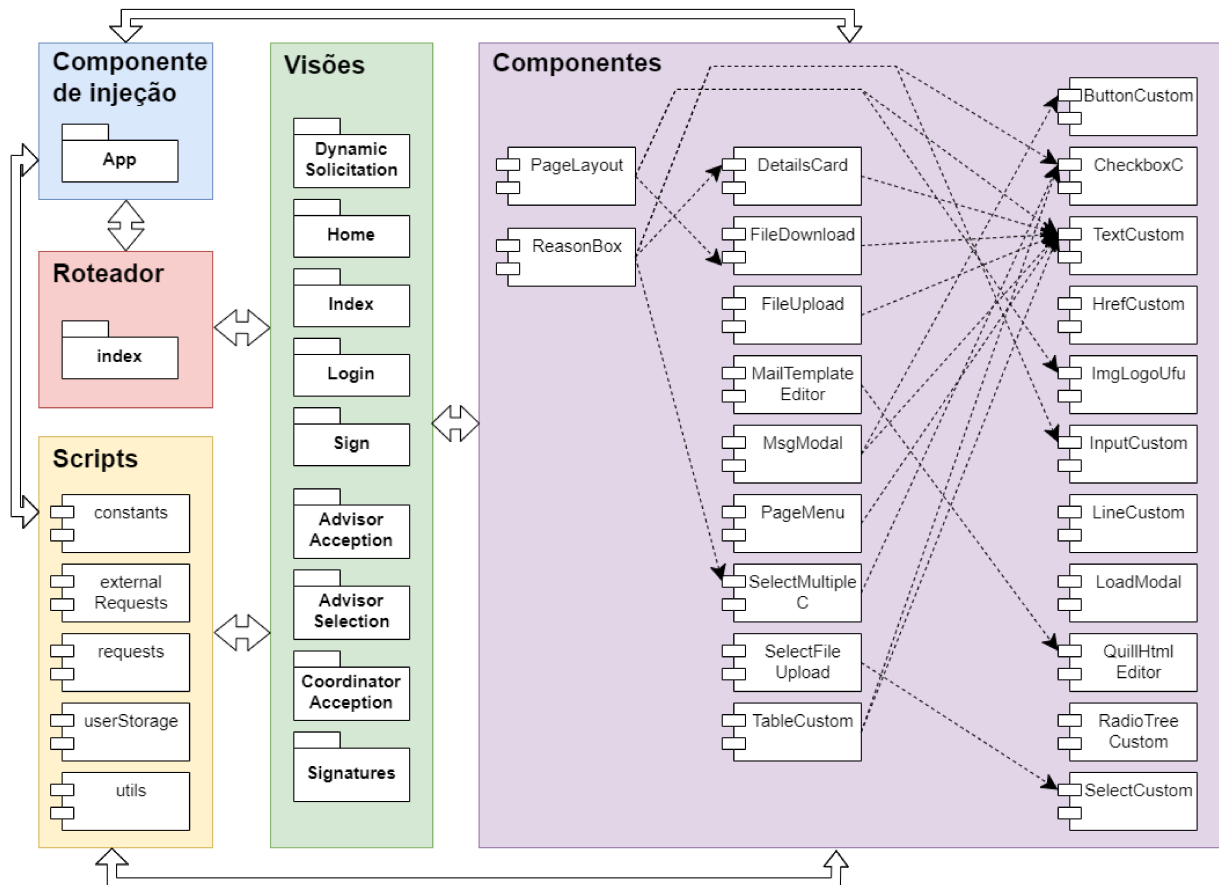


Figura 3 – Abstração do front-end, seus componentes e suas comunicações entre eles (ROCHA, 2023). Note que o componente de injeção *App* é responsável por gerenciar a base da *Single-Page Application*.

cionadas nos scripts para permitir o seu uso em toda a aplicação, neste contexto pode-se citar o script *requests* que é responsável pelas rotas de comunicação com o servidor back-end. Neste sistema há dois tipos de visões, as dinâmicas que são geradas dinamicamente com base no banco de dados e as estáticas, que por sua vez são complexas demais para serem criadas dinamicamente.

2.4.2 Back-end

A camada de back-end faz a gestão dos dados da aplicação, ela considera o fluxo completo das solicitações do usuário, além de aplicar as regras de segurança, configurações da API REST e modelagem dos objetos ORM a fim de facilitar a interação entre o banco de dados e front-end.

Como citado anteriormente, a linguagem escolhida foi Python² juntamente com o

² Para mais detalhes sobre a linguagem Python na versão 3.10 verifique o site <<https://www.python.org/>>

framework Flask³. Como também já mencionado, para conectar o back-end da aplicação com o banco de dados, foi utilizado o conjunto de ferramentas SQLAlchemy⁴, que é uma biblioteca que permite o mapeamento entre objetos da linguagem e instâncias das tabelas do banco de dados ao se utilizar ORM.

A camada tem várias funcionalidades para englobar abstrações de máquinas de estados, segurança da API baseada na autenticação por pares de chaves públicas e privadas, *threads* para os servidores de e-mail e agendadores de eventos. Essas funcionalidades são encapsuladas em diferentes classes dentro da aplicação, que são representadas por um diagrama de classes conforme ilustrado na Figura 4.

É importante citar também as rotas da API REST fornecidas, cada uma processa e/ou retorna dados com uma finalidade específica, que buscam realizar funcionalidades para usuários. As rotas atuam no processamento e persistência de dados, considerando operações de escrita, leitura, atualização/edição e remoção dos dados. Nesta subseção, elas são apresentadas a seguir, na Tabela 1, incluindo a categoria, operação HTTP, rota e qual componente a utiliza na camada de front-end.

2.4.3 Base de dados

O banco de dados relacional escolhido é o MySQL (Oracle Corporation, 2024) na camada de persistência de dados, projetado para ser flexível a mudanças no fluxo do sistema. Foi escolhido o MySQL por sua popularidade e familiaridade; ao todo, o sistema conta com quarenta e três tabelas para a persistência dos dados, pois grande parte da complexidade foi abstraída na camada. Na abstração presente na Figura 5, pode-se visualizar uma lista com os nomes das tabelas presentes no esquema do banco de dados, bem como suas relações com as demais tabelas.

O enfoque da Figura 5 é fornecer um descritivo breve de como é possível realizar alterações nas funcionalidades do sistema, bem como a mudança de fluxos das máquinas de estados considerando mudanças em suas tabelas. Os scripts e dados iniciais para criação das tabelas estão presentes no repositório back-end, para serem criados automaticamente a primeira vez que a aplicação é executada, desde que as ferramentas estejam previamente instaladas e configuradas. Isso torna a aplicação simples de se iniciar em novos servidores.

O sistema primeiro verifica se o esquema existe, caso não exista ele inicia uma transação e cria todas as tabelas necessárias, já com dados iniciais que geram os fluxos da máquina de estados, o nome do esquema utilizado pode ser facilmente alterado ao modificar o campo respectivo nas variáveis de ambiente.

³ Verifique mais detalhes sobre o *framework* Flask 2.2.2 em <<https://flask.palletsprojects.com/>>

⁴ Verifique mais detalhes sobre o conjunto de ferramentas SQL SQLAlchemy 2.0 em <<https://www.sqlalchemy.org/>>

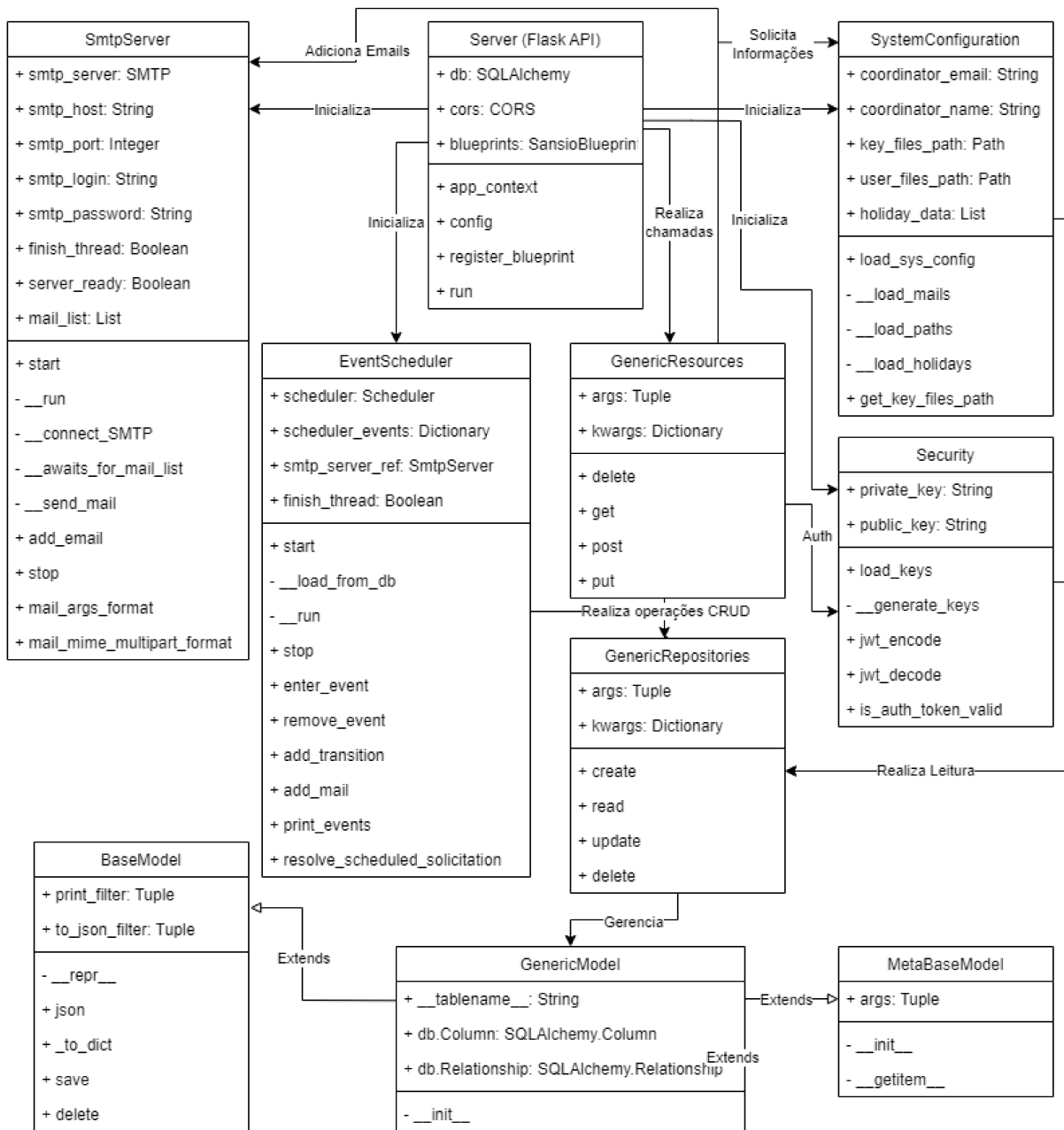


Figura 4 – Diagrama de classes do back-end do Sistema SisFlow (ROCHA, 2023). A principal classe apresentada no diagrama é a do servidor Flask.

Tabela 1 – Tabela com as rotas do *Back-end* (ROCHA, 2023).

Categoria	Verbo HTTP	Rota	Componentes do <i>Front-end</i> que utilizam
Arquivos	Post	/file	FileUpload, SelectFileUpload(Indiretamente), DynamicSolicitation(Indiretamente), Signatures(Indiretamente)
	Get	/file	FileDownload, DynamicSolicitation(Indiretamente), CoordinatorAcception(Indiretamente), Signatures(Indiretamente)
Autenticação	Get	/sign	Sign
	Post	/login	App, Login(Indiretamente)
	Post	/sign	Sign
	Put	/sign	Sign
Envio de E-mails	Post	/sendmail	CoordinatorAcception
Orientadores	Get	/advisors	AdvisorSelection
Página Dinâmica	Get	/dynamicpage	
Solicitação	Get	/solicitation	AdvisorAcception, AdvisorSelection, CoordinatorAcception, DynamicSolicitation, Signatures
	Post	/solicitation	AdvisorAcception, AdvisorSelection, CoordinatorAcception, DynamicSolicitation, Signatures
	Put	/solicitation	Home
Solicitação de Orientadores	Patch	/solicitation/advisor	AdvisorAcception
	Put	/solicitation/advisor	AdvisorSelection
Solicitações	Get	/solicitations/coordinator	Home
	Get	/solicitations/advisor	Home
	Get	/solicitations/student	Home
Tabela de Motivos	Get	/reasons	ReasonBox, CoordinatorAcception(Indiretamente)
Transições de Solicitações	Get	/solicitation/transitions	

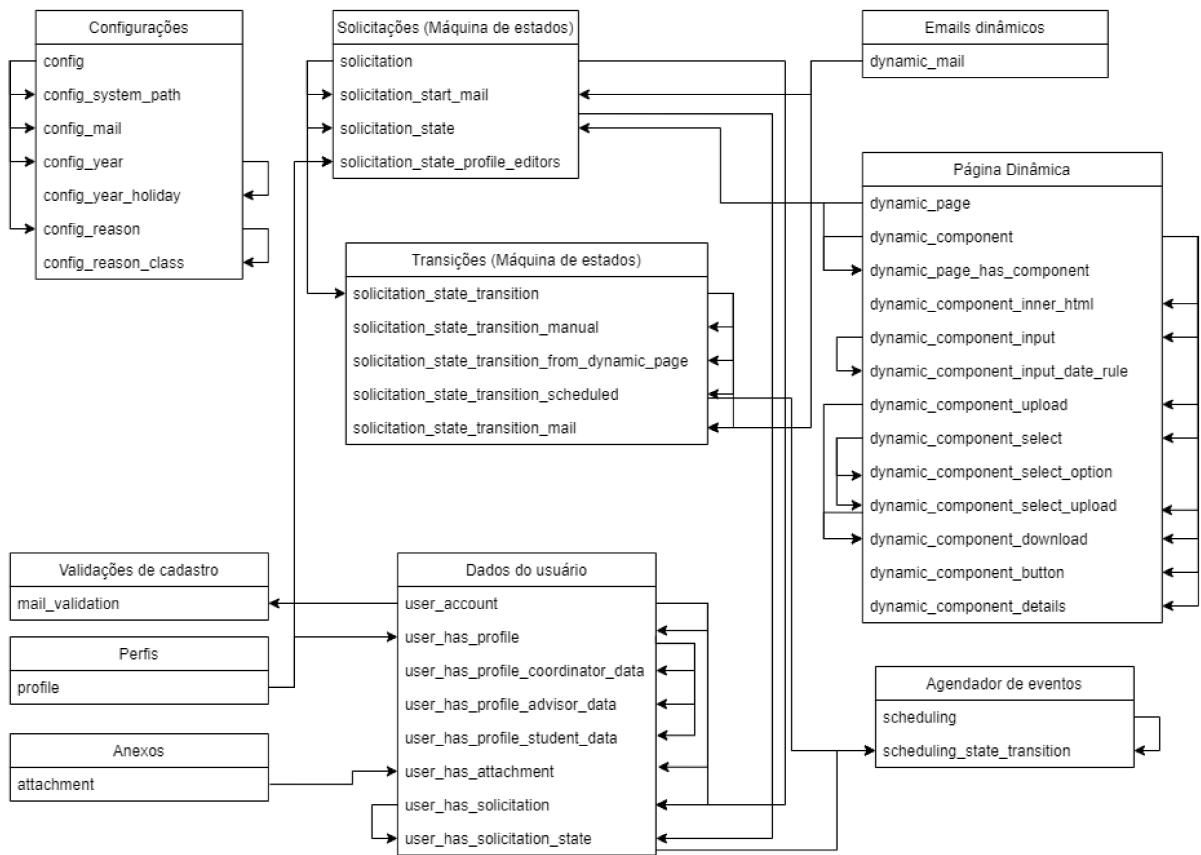


Figura 5 – Abstração simplificada do Banco de dados, apresenta lista de tabelas agrupadas em categorias e seus respectivos relacionamentos (ROCHA, 2023).

3 Trabalhos Relacionados

Diversos outros trabalhos encontrados na literatura, muitos deles em nível de Trabalho de Conclusão de Curso, abordam a temática de sistemas para automatização do fluxo de estágio em universidades e institutos educacionais. Tais projetos buscam contribuir para a melhoria dos processos acadêmicos e demonstram a importância da automação e otimização dos fluxos de estágio supervisionado.

Um exemplo é a monografia escrita por [Leoncio \(2018\)](#), que apresenta um sistema para controle de Estágio no Departamento Acadêmico de Informática da Universidade Tecnológica Federal do Paraná, Campus Ponta Grossa, que oferece recursos para o gerenciamento dos estágios.

O SIGESTAGIOS é uma plataforma Web para sistematizar e automatizar as etapas do estágio supervisionado no âmbito da Universidade Estadual do Piauí ([SOUZA; ROCHA; BALUZ, 2019](#); [BALUZ; ROCHA; SOUZA, 2020](#)). Assim como o trabalho ora apresentado, o SIGESTAGIOS atende atores como Alunos, Professores Orientadores, Coordenação de Estágio e empresas, utilizando, inclusive, o mesmo *framework* para desenvolver o front-end (Vue.js). A plataforma SIGESTAGIOS foi pensada primordialmente para mitigar erros e redundâncias no preenchimento de dados e também facilitar a gestão de documentos, centralizando-a. Assim, evita-se gastos desnecessários com papel e tempo, tanto dos alunos quanto dos demais agentes envolvidos no processo.

Já no trabalho de [Seles \(2018\)](#), foi feito o desenvolvimento do Sistema Integrado de Gestão Acadêmica, que acabou substituindo o sistema anterior que utilizavam, o qual se tornou obsoleto. Eles programaram um sistema baseado nas linguagens HTML, PHP e MySQL, o que agilizou o processo com uma melhor análise, informações acessadas mais rapidamente e também possibilitou a adaptação e evolução, fundamentais no desenvolvimento de um sistema coeso.

Na monografia de [Almeida \(2015\)](#), foi observado que o Coordenador de Estágio, além de suas atividades como professor e pesquisador, teria de gastar um bom tempo respondendo a e-mails, fazendo telefonemas e assinando documentos. Foram usadas tecnologias como CSS, Bootstrap, Apache e uma Metodologia Ágil XP adaptada. Para o desenvolvimento, o autor utilizou conceitos de Interação Humano-Computador, como a criação de Histórias de Usuários.

No trabalho de [Perini e Yamamoto \(2007\)](#), os autores identificaram o mesmo problema da falta de um sistema para o gerenciamento das atividades na disciplina de Estágio Supervisionado Curricular Obrigatório, é importante citar que este trabalho teve início em dois mil e sete, então nota-se que não é de hoje que as pessoas se preocupam com a auto-

mação de processos dentro das instituições de ensino. Por sua vez, nota-se uma diferença significativa nas tecnologias utilizadas e disponíveis na época e também na arquitetura e metodologia escolhidas, no caso, utilizaram o modelo espiral, que integra o Modelo de Prototipação e o Modelo Sequencial, seguindo o paradigma Orientado a Objetos.

Na monografia de [Miranda e Paula \(2018\)](#), os autores também implementam o fluxo documental do estágio. Uma semelhança com o sistema aqui descrito é que ele também se preocupou com a portabilidade/acessibilidade, já que o sistema denominado de Sistema de Gerenciamento de Estágios foi desenvolvido para Web, o que torna-o acessível através de qualquer dispositivo com acesso à Internet por meio de um navegador.

Outro sistema que pode ser citado como trabalho correlato é um da própria Faculdade de Computação da UFU, em que [PEREIRA \(2023\)](#) automatizou uma solução que tomava tempo da coordenação e, apesar de não ser um sistema para gerenciamento de estágio, tem algumas semelhanças com o Sistema SisFlow. [PEREIRA \(2023\)](#) automatizou o envio de e-mails de alerta para os discentes com base na sua situação acadêmica de acordo com normas de graduação, auxiliando os usuários a se atentarem com prazos e limites que devem ser cumpridos.

Embora sistemas desse tipo não sejam incomuns, é importante ressaltar que cada trabalho possui suas particularidades, regras e metas específicas, de acordo com a universidade de origem. O diferencial desta monografia, é o foco na usabilidade e conceitos da Interação Humano Computador, a fim de promover um sistema que seja intuitivo, acessível e centrado nas necessidades dos usuários. Ao enfatizar esses aspectos, busca-se oferecer uma experiência de uso agradável e eficiente, reduzindo a curva de aprendizado e aumentando a satisfação e o engajamento dos usuários. Embora o custo financeiro não seja uma preocupação central neste Trabalho de Conclusão de Curso, é importante lembrar que, em contextos reais, um sistema bem projetado pode reduzir custos a longo prazo, minimizando a necessidade de suporte e retrabalho devido a falhas de usabilidade e problemas na interação.

4 Método

Nesta monografia, o método de pesquisa utilizado foi o método exploratório, que pode ser definido como um método capaz de melhorar a definição ou design de um fenômeno ainda desconhecido, sendo importante em estágios iniciais, quando se busca principalmente a familiaridade com algum objeto ou disciplina (WAZLAWICK, 2009).

Ainda de acordo com Wazlawick (2009), este método não possui uma teoria formal nem medidas de avaliação bem definidas e, geralmente, a pesquisa é conduzida por meio de estudos de caso, análises e comparações. A finalidade do método neste trabalho é expor ao público o produto em si de modo a promover a reflexão e novas hipóteses, comparando-o com a versão anterior do sistema. Por meio de estudos de caso, observação e comparação com a versão anterior, busca-se confirmar a hipótese de que uma abordagem centrada na Interação Humano-Computador é fundamental para melhorar um produto e garantir sua aceitação pelo público. É, portanto, um trabalho de evolução incremental, por meio do aperfeiçoamento de um produto/sistema existente.

O método de pesquisa foi complementado com o uso das heurísticas de usabilidade descritas nas Seções 2.1 e 2.2, a partir com o emprego do método de avaliação de usabilidade, mais especificamente como uma inspeção de usabilidade inspirada na técnica de Avaliação Heurística (NIELSEN, 1992).

Em linha com os objetivos específicos, o desenvolvimento deste trabalho se deu em três principais etapas:

1. Primeiramente, o estudo do sistema já existente e sua arquitetura;
2. Revisão das telas e fluxos de informação para incorporação dos aspectos de usabilidade (heurísticas); e
3. A sugestão e implementação das correções e mudanças necessárias.

A primeira etapa já foi descrita na Seção 2.3, enquanto as demais etapas, que constituem a principal contribuição do trabalho, serão apresentadas no capítulo seguinte.

5 Desenvolvimento

A aplicação é mantida em dois repositórios distintos^{1,2}, separando a interface visual das regras de negócio e do banco de dados. Essa divisão visa ao desacoplamento das diferentes camadas tecnológicas, organizadas entre front-end e back-end. Com essa estrutura, atualizações podem ser feitas de maneira independente, permitindo que equipes trabalhem simultaneamente e de forma mais eficiente. Além disso, facilita a consistência no uso de ferramentas de *deploy*, pois é possível configurar a infraestrutura para disponibilizar apenas os recursos necessários para cada parte do sistema, seja a interface de usuário ou a lógica de negócios.

Primeiramente, foi feita uma análise no fluxograma do sistema, para entender como ele funciona e analisar quais melhorias deveriam ser feitas a fim de facilitar o uso e melhorar a experiência do usuário. É importante ressaltar que toda mudança em um sistema já construído deve ser feita de maneira metódica, a fim de não torná-lo mais complexo, não fazer re-trabalho e não estragar uma funcionalidade ou fluxo antes presente, todos os detalhes de arquitetura importam neste primeiro momento.

Após entender como o sistema se comporta em seu uso cotidiano, foram feitas as revisões das telas, para tentar identificar alguma inconsistência no design e/ou violações nos princípios de usabilidade.

Começando na tela inicial do sistema, disponível na Figura 6, foram encontrados alguns problemas, como nomes sem a devida acentuação, e o ícone e logo do sistema sem a funcionalidade de voltar à página *home*.

Após corrigir esses problemas, notou-se que o Coordenador de Estágio, agente fundamental no uso do sistema, não possuía nenhum recurso de controle e personalização do sistema, como um filtro ou uma busca, o que dificultaria seu uso. No caso da base de testes usada, tem-se poucos usuários cadastrados, o que não gera problemas de poluição visual na tela, porém, no dia a dia, o número estimado de alunos realizando o processo de estágio conta com mais de trezentos simultaneamente, dispostos em uma tabela única, o que pode dificultar o trabalho do Coordenador de Estágio, que nesse caso exerce o papel de administrador do sistema.

Na segunda versão do sistema, representada pela Figura 7, foram corrigidos os problemas encontrados, garantindo a quarta heurística de design descrita por (NIELSEN, 1994b), que é a Consistência e padrões, já que foram corrigidas todas as palavras e adici-

¹ *Front-end* disponível em <<https://github.com/carlosadnsm/sisflow-frontend>>

² *Back-end* e código da base de dados disponível em <<https://github.com/carlosadnsm/sisflow-backend>>

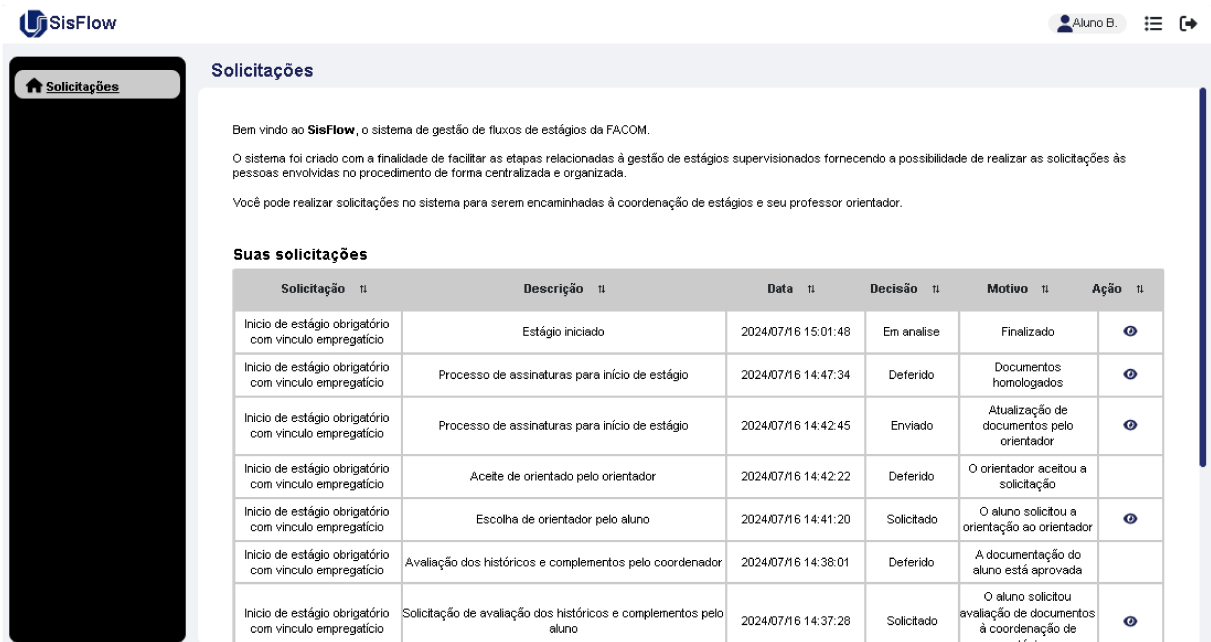


Figura 6 – Primeira versão da tela inicial do sistema. Note que não há nenhum recurso de personalização, a acentuação não está correta para algumas palavras, a ordenação padrão é de acordo com a data e hora.

onado a todas as páginas a funcionalidade do ícone do sistema no canto superior esquerdo funcionar como um *hiperlink* para a página inicial do sistema. Ao adicionar os filtros para a ordenação de todos os campos, é garantida uma maior flexibilidade e eficiência do uso, que se relaciona com a sétima heurística de usabilidade, flexibilidade e eficiência de uso. Os filtros de agrupamento funcionam como aceleradores, permitindo a personalização de maneira rápida e eficiente.

A personalização dos filtros foi feita somente para a versão de *desktop*, já que, na versão *mobile*, o uso não será corriqueiro no caso do administrador do sistema, o principal afetado pela mudança. Foi uma decisão de design pragmática, já que ao usar o sistema de um *smartphone* ou *tablet*, a tabela se colapsa e suas colunas agrupam-se em blocos de linhas para facilitar a navegação e leitura; o mesmo pode ser visualizado na Figura 8.

Correções de consistência, como alinhamento de botões, padronização dos mesmos, padronizações de cores, podem ser vistas como simples estética, mas são fundamentais para facilitar o entendimento, pelo usuário, do modelo mental do sistema. O usuário não deve se preocupar com o significado de ícones, símbolos, links, mas sim o desenvolvedor deve programar o sistema pensando em uma interface previsível e amigável, oferecendo um sistema com baixa curva de aprendizado para novos usuários, com um modelo mental claro, o que também remete à quarta heurística de (NIELSEN, 1994a). Nas Figuras 9 e 10, podemos ver exemplos de padronização, nos botões de *download* e nos botões de voltar à página anterior. Note que todos os botões dos sistema foram padronizados de forma



Bem vindo ao **SisFlow**, o sistema de gestão de fluxos de estágios da FACOM.

O sistema foi criado com a finalidade de facilitar as etapas relacionadas à gestão de estágios supervisionados fornecendo a possibilidade de realizar as solicitações às pessoas envolvidas no procedimento de forma centralizada e organizada.

Você pode realizar solicitações no sistema para serem encaminhadas à coordenação de estágios e seu professor orientador.

Suas solicitações

Solicitação	Descrição	Data	Decisão	Motivo	Ação
Início de estágio obrigatório com vínculo empregatício	Estágio iniciado	2024/07/16 15:01:48	Em análise	Finalizado	🔍
Início de estágio obrigatório com vínculo empregatício	Processo de assinaturas para início de estágio	2024/07/16 14:47:34	Deferido	Documentos homologados	🔍
Início de estágio obrigatório com vínculo empregatício	Processo de assinaturas para início de estágio	2024/07/16 14:42:45	Enviado	Atualização de documentos pelo orientador	🔍
Início de estágio obrigatório com vínculo empregatício	Aceite de orientado pelo orientador	2024/07/16 14:42:22	Deferido	O orientador aceitou a solicitação	
Início de estágio obrigatório com vínculo empregatício	Escolha de orientador pelo aluno	2024/07/16 14:41:20	Solicitado	O aluno solicitou a orientação ao orientador	🔍
Início de estágio obrigatório com vínculo empregatício	Avaliação dos históricos e complementos pelo coordenador	2024/07/16 14:38:01	Deferido	A documentação do aluno está aprovada	
Início de estágio obrigatório com vínculo empregatício	Solicitação de avaliação dos históricos e complementos pelo aluno	2024/07/16 14:37:28	Solicitado	O aluno solicitou avaliação de documentos à coordenação de estágios	🔍

Figura 7 – Segunda versão da tela inicial do sistema. Observe que agora todos os textos estão com a devida acentuação e gramática coerente, neste momento há um filtro disponível em todos os campos da tabela.

verbosa, para o usuário se acostumar com as ações que podem ser tomadas ao clique. Estas padronizações foram feitas em todas as telas do sistema, sendo que as duas telas apresentadas foram utilizadas somente com a finalidade de demonstração.

Repare que houve uma preocupação em manter todos os botões contendo verbos em seus rótulos, para dar sentido de ação e garantir que o usuário entenda claramente o que acontecerá ao clicar em cada um deles. Essa escolha de vocabulário não só facilita a compreensão das funcionalidades, mas também alinha a interface com o modelo mental do usuário, que geralmente espera que as interfaces indiquem explicitamente quais ações estão disponíveis. Essa abordagem permite que o sistema comunique, de forma intuitiva, o que cada comando realizará, minimizando confusões e erros ao longo do uso.

A opção por rótulos verbosos atende ainda à heurística de Correspondência entre o sistema e o mundo real, na qual a linguagem do sistema é alinhada ao dia a dia do usuário, ajudando-o a identificar rapidamente as opções e ações que precisa tomar. Além disso, o uso de verbos contribui para a visibilidade do estado do sistema, pois reforça as ações que o sistema realizará, aumentando a previsibilidade e transparência das interações. Essa consistência na escolha de rótulos ao longo de toda a aplicação também favorece a Consistência e Padrões, garantindo que, ao se deparar com novos botões ou opções, o usuário já tenha uma ideia clara das ações que pode esperar.

Outra alteração feita com o objetivo de manter a consistência com padrões e ajudar na minimização de erros foi a alteração e padronização de botões de ação positiva como

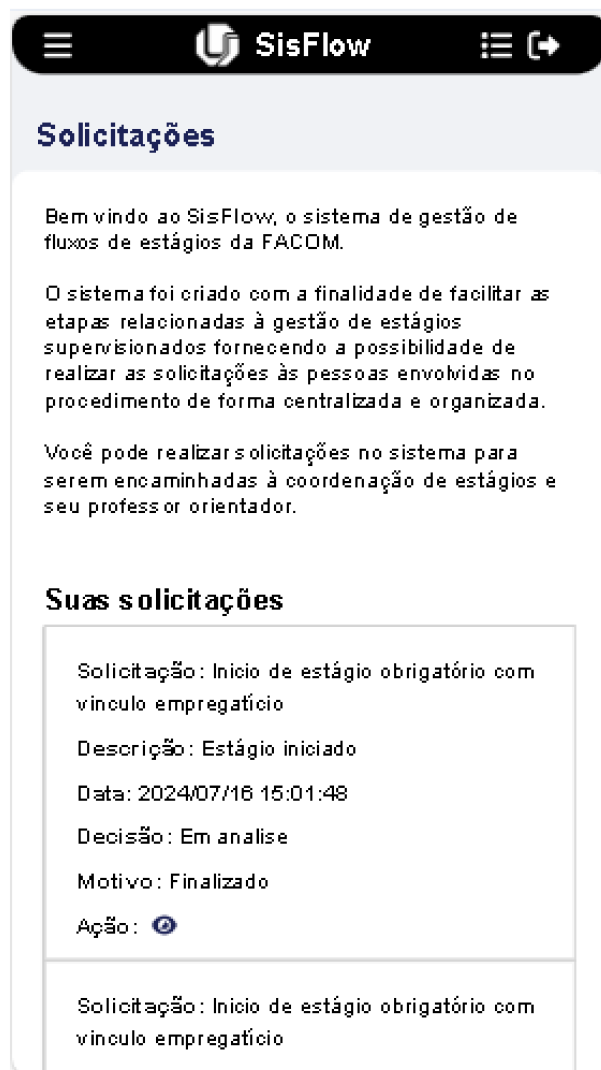


Figura 8 – Versão *mobile* da tela inicial, note o agrupamento das linhas da tabela em bloco, para facilitar leitura e dar a noção de um bloco de informações pertencentes a um mesmo grupo.

“Aceitar” e “Confirmar” para o canto direito da página, pois isso corresponde ao fluxo natural de leitura e à forma como os usuários processam as informações e tomam decisões. Nas Figuras 11 e 12, é possível observar um exemplo dessa mudança, onde os botões de ação positiva foram realocados para o canto direito da interface. Essa alteração não apenas reforça o alinhamento com o modelo mental dos usuários, mas também contribui para um design mais intuitivo, que facilita a navegação, minimiza erros e melhora a experiência geral de uso.

Na tela de Assinaturas para início de estágio, também foi feita a padronização dos botões de *Download* e para Mostrar histórico à direita e também do botão “Voltar” no canto superior esquerdo, o que pode ser visualizado nas Figuras 13 e 14.

Ao se comprometer a fazer um sistema que foca em usabilidade, qualquer detalhe conta para ajudar na semântica. O sistema conta com uma tela de verificação de do-



Figura 9 – Tela de “Verificação da documentação” após padronização dos botões de “Download” e dos botões de “Voltar” à tela anterior, presentes respectivamente no canto direito e no canto superior esquerdo da página.



Figura 10 – Página de “Solicitação de início de estágio – concluída” após padronização nos botões de “Download” e nos botões de “Voltar” à página anterior, presentes respectivamente no canto direito e no canto superior esquerdo da página.

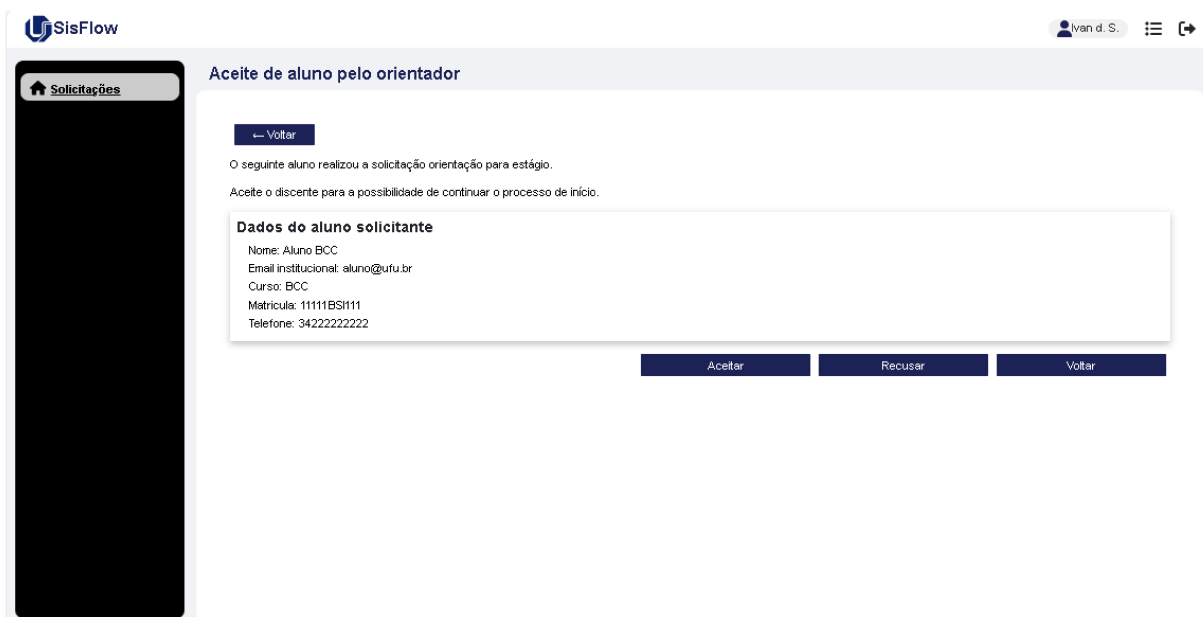


Figura 11 – Tela de “Aceite do aluno pelo orientador” na versão anterior do sistema, com os botões dispostos na seguinte ordem: “Aceitar”, “Recusar” e “Voltar”. Essa ordem original dificulta o fluxo natural de leitura, já que os usuários estão acostumados que o botão inferior à direita dê o sentido de continuidade.

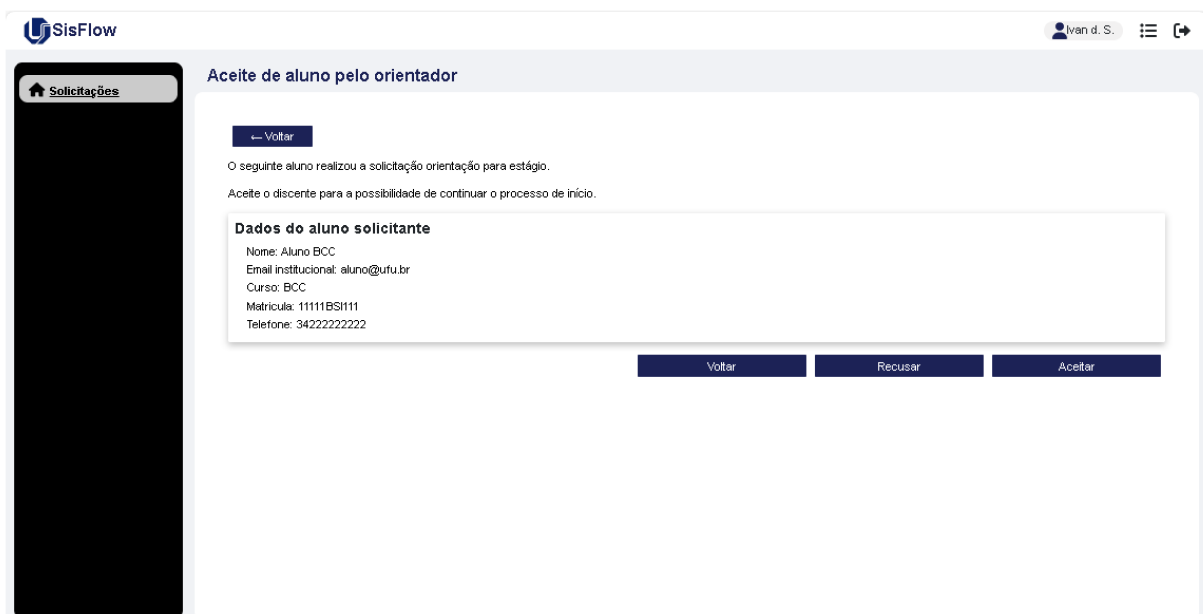


Figura 12 – Tela de “Aceite do aluno pelo orientador” após alteração da ordem dos botões para: “Voltar”, “Recusar” e “Aceitar”. O que busca facilitar o entendimento e trás uma interface mais previsível ao usuário.



Figura 13 – Tela de “Processo de assinaturas para início de estágio”, antes da alteração, note que o botão “Mostrar histórico” estava alinhado à esquerda, quebrando o fluxo de leitura e dando um aspecto desagradável à interface. Note também que o botão de “Voltar” estava no canto direito inferior da página, o que poderia fazer o usuário se frustrar ao quebrar a noção de continuidade padronizada na maioria dos sistemas atuais.

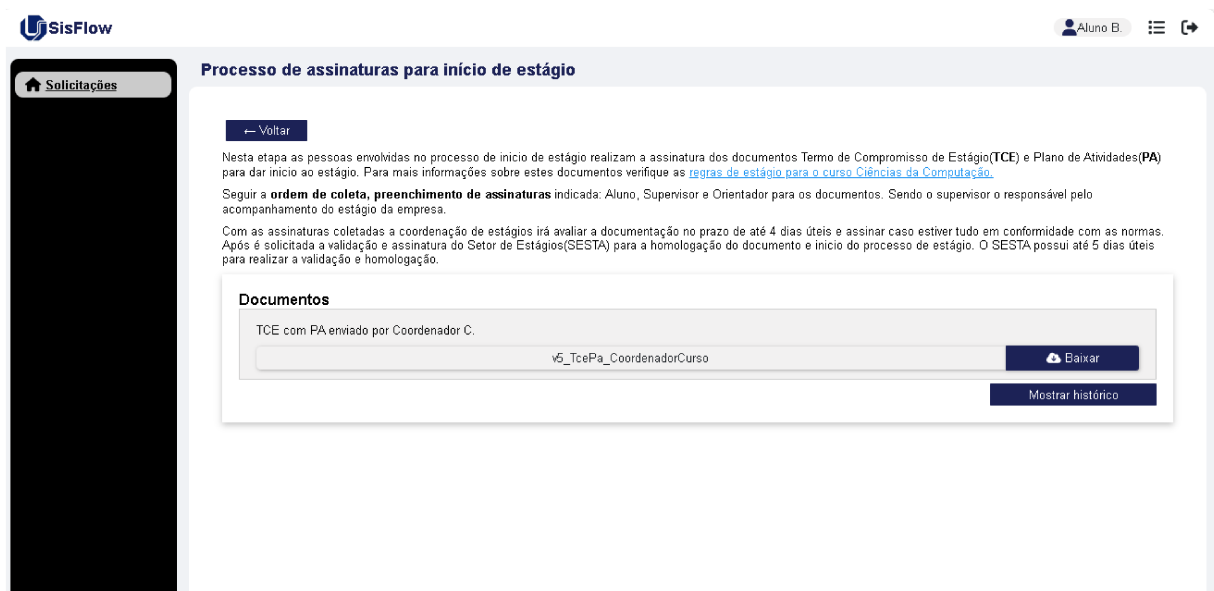


Figura 14 – Tela de “Processo de assinaturas para início de estágio” após a correção do alinhamento dos botões à direita e a inclusão do botão de “Voltar” com o ícone de seta no canto superior esquerdo da tela.

cumentação do aluno pela coordenação, onde o coordenador pode deferir ou indeferir o início do estágio. No final desta tela, o coordenador pode gerar um e-mail automático e até personalizá-lo, descrevendo os motivos em caso de indeferimento. Porém, foi percebido que, ao passar rapidamente pela área “Motivos” na interface original, não ficava claro para o usuário o que seriam tais motivos. Então, uma simples alteração na descrição (vocabulário do modelo mental) permitiu adicionar semântica e evitar erros, como pode ser observado nas Figuras 15 e 16.



Figura 15 – Tela de “Solicitação de início de estágio - Verificação de documentação do aluno pela coordenação”, note que a descrição da área “Motivos” continha somente o texto “Selecione os motivos abaixo.” e uma breve explicação sobre o filtro. A interpretação do que seriam tais motivos, neste contexto, era vaga e subjetiva, carecendo de uma explicação mais elaborada.

Nesta mesma tela, foi verificado que os dados do aluno estavam ocupando um grande espaço de tela, o que vai contra a teoria de Gestalt, no artigo (OFICIAL, 2021) podemos ver que no contexto de design e desperdício de tela, a teoria da Gestalt pode ser aplicada para evitar o uso ineficiente do espaço visual. Como exemplo, ao organizar elementos em uma interface, é importante considerar leis como a proximidade (elementos próximos são percebidos como pertencentes a um mesmo grupo) e a similaridade (elementos semelhantes são percebidos juntos). Porém, era importante mantê-los, então foi adicionada a funcionalidade de colapsar estes dados ao clique, sendo observado o ganho de espaço de tela nas Figuras 17 e 18.

Na maioria dos sistemas, temos botões de ação positiva e negativa, como “Aceitar” e “Recusar”, “Voltar” e “Continuar”, entre outras variantes. Neste sistema, não é diferente e, para evitar erros, optou-se pelo acréscimo de cores como forma de redundância de informação, reforçando mais uma vez a semântica e o modelo mental do sistema



Figura 16 – Tela de “Solicitação de início de estágio - Verificação de documentação do aluno pela coordenação”, note que a descrição da sessão “Motivos”, grifada em vermelho no canto superior esquerdo da figura, agora tem uma breve explicação “Selecione os motivos para eventual deferimento abaixo” e a mesma explicação sobre o funcionamento dos filtros. Essa simples alteração faz toda a diferença no entendimento da página.



Figura 17 – Tela de “Solicitação de início de estágio - Verificação de documentação do aluno pela coordenação”, note o espaço que os dados do aluno ocupavam na tela.

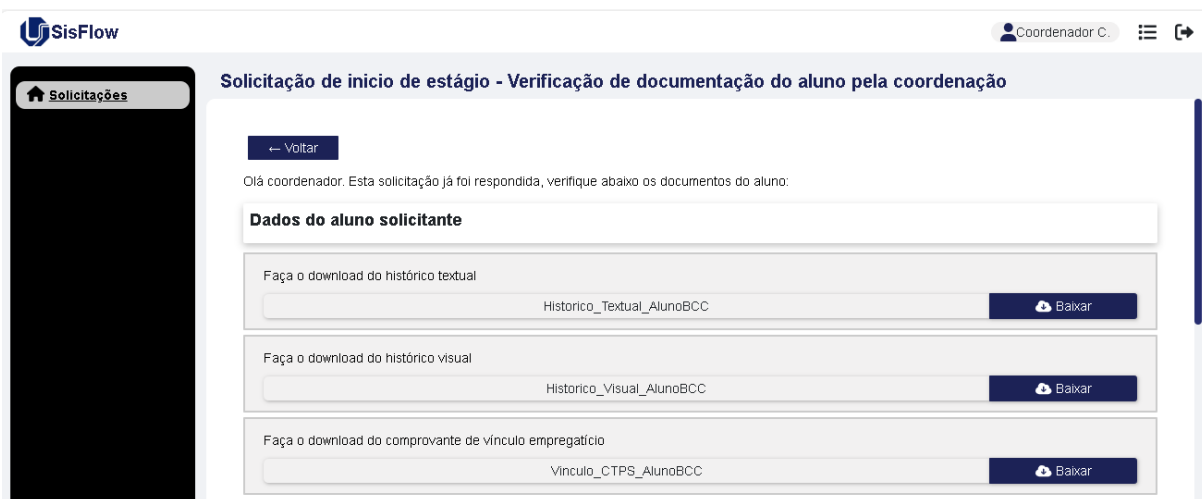


Figura 18 – Tela de “Solicitação de início de estágio - Verificação de documentação do aluno pela coordenação”, onde agora é possível colapsar os dados do aluno solicitante para facilitar a leitura das demais partes. Note também que, nesse ponto do sistema, os botões já estão todos padronizados.

para o usuário. Inicialmente, todos os botões seguiam o padrão neutro, na cor azul escura, o que deixa o sistema bonito e minimalista. Porém, foram criadas mais duas cores, verde e vermelho, na mesma paleta (consistência), a fim de reforçar o *affect* positivo e negativo, já que o verde refere-se ao sentimento de positividade, crescimento, confiança e estabilidade, enquanto o vermelho refere-se a atenção, urgência, perigo e alerta. Essas duas cores, quando combinadas e usadas de forma estratégica, podem melhorar significativamente a experiência do usuário e também a eficácia do design do site, evitando erros e *missclicks* (clique acidental, geralmente no lugar errado). O uso desse contraste pode ser visto nas Figuras 19 e 20.

De acordo com (VESPUCCI, 2009), estima-se que no Brasil o daltonismo afeta cerca de dez por cento dos homens e um por cento das mulheres, uma quantidade significativa de pessoas. O daltonismo é provocado por um gene recessivo ligado ao cromossomo X, por isso as mulheres são menos afetadas, já que seus dois cromossomos X devem ser afetados para que ela se torne daltônica. Com isso em vista, além das cores, também foram adicionados ícones de confirmação e rejeição, adicionando mais uma camada semântica que favorece a acessibilidade do sistema. Tais alterações também podem ser visualizadas nas Figuras 19 e 20. Novamente, ressaltamos que essas melhorias no design dos botões foram incorporadas em todo o sistema e que a tela em questão foi usada somente como exemplo.

Por fim, notou-se que o sistema ainda restringia um pouco o seu uso para o ator principal do sistema, o Coordenador de Estágio da FACOM, mesmo que tenha sido incluído o filtro para todos os campos da tabela, faltava algo para acelerar mais ainda seu



Figura 19 – Tela de “Aceite de aluno pelo professor orientador”, contendo os botões “Aceitar” e “Recusar” antes da alteração cores. Note que, em relação à Figura 12 (na qual é a mesma tela mostrada), foi feita a alteração de retirar o botão “Voltar” da parte inferior direita e reposicionando, de forma padronizada, esse botão em todas as telas, na parte superior esquerda.



Figura 20 – Tela de “Aceite de aluno pelo professor orientador”, contendo os botões “Aceitar” e “Recusar” após a alteração cores e adição dos ícones. Note como o contraste facilita o entendimento e a noção de redundância de informações, tanto na cor, na disposição dos mesmos e também no ícone. Essa alteração tem o potencial de evitar erros e *missclicks* no uso cotidiano do sistema.

uso. (CHOO, 2003) sugere que a busca é uma atividade intencional, na qual os indivíduos mobilizam recursos para satisfazer suas necessidades cognitivas e operacionais, gerando novos conhecimentos ou suportando decisões importantes no contexto organizacional. O modelo de busca e uso da informação de Choo é amplamente adotado para compreender o comportamento de usuários ao acessar e utilizar informações. Ele descreve como as pessoas identificam uma necessidade de informações, buscam dados para preenchê-la e aplicam o conhecimento adquirido, influenciadas por fatores emocionais e situacionais. O modelo é valioso na IHC, pois esclarece como sistemas devem ser projetados para apoiar melhor as etapas da busca de informações, facilitando a navegação, usabilidade e a tomada de decisão dos usuários ao longo de suas jornadas informacionais.

A título de ilustração, na Figura 21, foram separados trechos de código para detalhar melhor como foi feita a programação do sistema, permitindo comparações entre a abordagem original e a atual. Anteriormente, o sistema fazia uma verificação de qual perfil de usuário estava logado e assim carregava a sua respectiva tabela. No estado atual do sistema, além desta verificação também é feita uma verificação se o campo de busca está vazio. Para evitar problemas, foi definido no *return* do *data* o termo *searchTerm* recebe o valor ‘’. Além destas alterações, foi necessário estilizar a barra de busca através de CSS e criar dentro do *computed* uma atualização condicional da tabela “*filteredData*” somente se o usuário digitar algo no campo de busca. O resultado do código pode ser visualizado na Figura 22.

É importante destacar que a busca ignora qualquer tipo de acentuação e não importa se o usuário escreve em minúsculo ou maiúsculo. Esta última funcionalidade implementada abrangeu várias das 10 heurísticas de Nielsen, tais como:

- **A visibilidade de estado do sistema:** A verificação condicional e a atualização da tabela “*filteredData*” informam claramente ao usuário o estado da busca e como os resultados são filtrados. Isso garante que o usuário entenda o que está acontecendo enquanto utiliza o sistema.
- **Correspondência entre o sistema e o mundo real:** O sistema ignora acentuação e diferenciação entre maiúsculas e minúsculas, o que facilita o uso da linguagem natural do usuário, permitindo uma experiência mais intuitiva. O comportamento da busca reflete uma maneira de pesquisa natural e esperada pelos usuários.
- **Controle e liberdade do usuário:** O usuário tem controle sobre a busca, podendo definir os termos a serem pesquisados, e o sistema permite realizar modificações sem dificuldades. Se o campo de busca estiver vazio, a busca é limpa, permitindo ao usuário recomeçar a pesquisa de forma fácil.
- **Prevenção de erros:** A verificação se o campo de busca está vazio ajuda a evitar

The image shows a code editor with a file explorer on the left and three code snippets. The file explorer shows a project structure with folders like 'sisflow-backend', 'sisflow-frontend', 'node_modules', 'public', 'src', 'assets', 'components', 'js', 'router', and 'views'. The 'views' folder is expanded, showing files like 'Advisor', 'Coordinator', 'InternshipBeginning', 'Student', 'DynamicSolicitationView.vue', 'HomeView.vue', 'IndexView.vue', 'LoginView.vue', 'SignView.vue', and 'App.vue'. The 'HomeView.vue' file is selected.

```

<div class="pageContentRow" v-if="this.coordinatorSolTable['content'].length > 0">
  <input
    v-model="searchTerm"
    type="text"
    placeholder="Buscar por qualquer campo da tabela"
    class="search-bar"
  />
  <div v-if="displayedTableData['content'].length > 0">
    <TextCustom
      customFontSize="title"
      margin="20px 0px 5px 0px"
      display="block"
    >
      Solicitações
    </TextCustom>
    <TableCustom class="tableC" :tableData="this.displayedTableData"/>
  </div>
  <div v-else>
    <p>Nenhum resultado encontrado.</p>
  </div>
</div>

```

```

data() {
  return {
    userProfiles: null,
    searchTerm: '',
    coordinatorSolTable: {
      'titles': [ 'Aluno', 'Orientador', 'Solicitação', 'Descrição', 'Data', 'Decisão', 'Motivo', 'Ação' ],
      'colTypes': [ 'string', 'string', 'string', 'string', 'string', 'string', 'string', 'iconfunction' ],
      'colWidths': [ '11%', '10%', '15%', '25%', '10%', '11%', '11%', '7%' ],
      'content': []
    },
  };
}

```

```

computed: {
  filteredData() {
    const search = this.searchTerm
      .toLowerCase()
      .normalize('NFD')
      .replace(/[\u0300-\u036f]/g, '');
    return this.coordinatorSolTable.content.filter((row) => {
      return Object.values(row).some((value) => {
        value
          .toString()
          .toLowerCase()
          .normalize('NFD')
          .replace(/[\u0300-\u036f]/g, '')
          .includes(search)
      });
    });
  },
  displayedTableData() {
    return this.searchTerm ? { ...this.coordinatorSolTable, content: this.filteredData } : this.coordinatorSolTable;
  },
}
mounted() {},

```

Figura 21 – Trechos de código responsáveis por implementar a busca condicional para o caso do Coordenador de Estágio estar logado no sistema. Observe a estrutura do código em visões, que foi previamente mencionada.

erros no processamento da pesquisa, minimizando problemas relacionados ao input inválido.

- **Reconhecimento em vez de recordação:** A barra de busca, que apresenta uma forma de acessar rapidamente os dados sem a necessidade de memorizar detalhes específicos, contribui para que o usuário se concentre na pesquisa, em vez de lembrar de termos exatos.
- **Eficiência e flexibilidade de uso:** O filtro de busca condicional e a atualização da tabela conforme o usuário digita, mesmo que a entrada seja insensível a maiúsculas/minúsculas e acentuação, aumenta a eficiência, tornando a busca mais rápida e flexível para diferentes tipos de usuários (novos ou experientes).

Essas seis heurísticas foram diretamente aplicadas, enquanto que outras, como “estética e design minimalista” e “ajuda e documentação”, foram utilizadas indiretamente.

Também é possível perceber que o trecho de código apresentado contempla algumas das diretrizes para Web design, como:

- **Facilidade de uso do campo de busca:** A implementação do campo de busca que aceita entradas sem diferenciação de acentuação ou maiúsculas/minúsculas facilita o uso para todos os usuários, independentemente do conhecimento técnico ou cuidado com os detalhes.
- **Clareza visual e feedback imediato:** O design da barra de busca, junto com a atualização condicional da tabela “filteredData”, proporciona um feedback imediato sobre os resultados da busca. Isso é importante para garantir que o usuário entenda instantaneamente os efeitos de suas ações.
- **Acessibilidade:** Considerações como a funcionalidade de busca sem diferenciar acentuação e maiúsculas/minúsculas tornam a aplicação mais acessível, já que há menos necessidade de o usuário inserir dados perfeitamente formatados.
- **Consistência:** A implementação segue padrões consistentes de navegação e interação, tornando a busca uma funcionalidade que se comporta de maneira previsível e familiar para o usuário.

Este capítulo procurou detalhar os problemas de usabilidade encontrados na interface original do Sistema SisFlow e, de forma correspondente, a implementação de melhorias para saná-las. A metodologia utilizada na pesquisa foi inspirada no método de inspeção de usabilidade, a partir da adaptação da técnica de Avaliação Heurística (NIELSEN, 1992), é importante citar que o método implica na escolha de um conjunto de

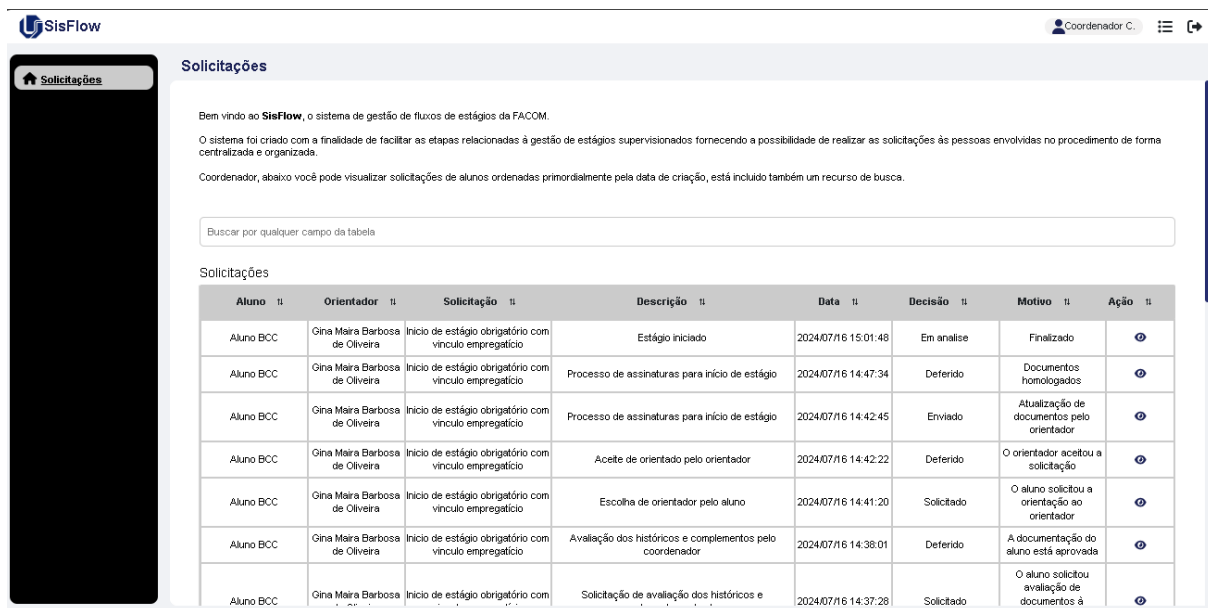


Figura 22 – Última versão da tela inicial do sistema, após implementar a busca para todos os campos da tabela de Solicitações.

heurísticas e que, no caso, foi escolhido o conjunto das 10 heurísticas do Nielsen, complementado pelas 113 diretrizes. Como diferencial, optou-se por não classificar a gravidade dos problemas encontrados e, ao invés de apenas listá-los e sugerir correções, como seria feito na técnica original, tais correções foram efetivamente implementadas, visando a continuidade do desenvolvimento do sistema, sua evolução e eventual implantação.

6 Conclusão

Estudos que contemplem os aspectos de usabilidade de um sistema são de suma importância para torná-lo mais intuitivos, amigáveis e simples de usar, minimizando erros que podem ser previstos sob a ótica da Interação Humano-Computador. A usabilidade, como um aspecto central na concepção de sistemas, visa não apenas melhorar a interação, mas também proporcionar uma experiência satisfatória, eficiente e agradável. Quando os princípios de IHC são aplicados corretamente, o sistema não só se torna mais acessível, mas também ajuda os usuários a realizarem suas tarefas de maneira mais rápida e sem frustrações.

Ao longo deste trabalho, foram utilizados dois conjuntos consagrados de regras de design: as 10 Heurísticas de Nielsen para Usabilidade de Interfaces de Usuário, e as 113 Diretrizes de Design para Usabilidade de Homepages.

De modo geral, a heurística de Nielsen mais presente foi a quarta, que trata da Consistência e Padrões. Essa heurística foi amplamente aplicada no desenvolvimento do sistema, proporcionando uma experiência uniforme em todas as telas e funcionalidades, o que facilita a navegação e reduz a carga cognitiva dos usuários. A consistência foi alcançada ao garantir que o comportamento do sistema e a interface seguissem um padrão claro e previsível, o que é crucial para que o usuário não precise relembrar como realizar ações em diferentes partes do sistema.

Além disso, exemplos da primeira heurística, que trata da Visibilidade do Status do Sistema, também estiveram presentes na página inicial do sistema. A interface foi desenhada para informar ao usuário de forma clara e contínua sobre o que estava acontecendo no sistema, se havia processos em andamento ou se alguma ação era necessária, proporcionando maior segurança e controle ao usuário durante sua navegação.

A Correspondência entre o Sistema e o Mundo Real, descrita na segunda heurística, foi aplicada por meio do uso de ícones que representam ações e estados do sistema de maneira intuitiva. Isso permitiu uma comunicação visual que facilita a compreensão do sistema, dispensando que o usuário precisasse memorizar jargões técnicos, tornando a interface mais amigável e próxima ao seu cotidiano.

A Liberdade de Controle do Usuário, mencionada na terceira heurística, foi garantida pelo sistema ao permitir que o usuário tenha autonomia para editar informações em determinadas áreas. Essa flexibilidade possibilitou ao usuário maior controle sobre as interações, permitindo correções rápidas e ajustes necessários durante o uso, o que contribui diretamente para uma experiência mais personalizada e sem frustrações.

A quinta heurística, descrita por Prevenção de Erros foi alcançada com a implementação da busca, ao se fazer uma verificação se o campo de busca está vazio, o ajuda a se evitar erros no processamento da busca.

O Reconhecimento ao invés de Lembrança, presente na sexta heurística, foi obtido ao alinhar todos os botões para a direita, alterar os botões de ação positiva e negativa para as respectivas cores verde e vermelho, ao manter ícones comumente usados como os de nuvem para *download*, a seta para trás para voltar, adicionar a funcionalidade de voltar para todas às páginas e a funcionalidade de voltar para a *home* clicando no ícone superior esquerdo do sistema, dentre outras funcionalidades.

A sétima heurística, de Flexibilidade e Eficiência de Uso, foi contemplada com a inclusão do filtro e busca para o Coordenador de Estágio, um dos atores principais do sistema que não fará seu uso corriqueiramente, mas sim um uso avançado.

A oitava heurística, de Estética e Design Minimalista, foi contemplada ao não exagerar demais nas cores, ao se buscar sempre um bom contraste e sem informações que sobrecarreguem a carga cognitiva do usuário.

A penúltima heurística, descrita pela Ajuda aos Usuários a Reconhecer, Diagnosticar e Recuperar-se de Erros, foi contemplada ao se permitir edição nos arquivos enviados para o sistema.

Por conseguinte, a décima e última heurística, de Ajuda e Documentação, foi contemplada com o check-list de documentação e também indiretamente, com a escrita desta monografia, já que ela complementa a documentação produzida originalmente por Rocha (2023), tornando-se um conglomerado de informações do sistema, sua arquitetura, implementações e funcionamento prático, a fim de servir como base para trabalhos futuros e também como um manual informativo para os usuários.

No caso deste trabalho, diversas melhorias foram implementadas para corrigir e aprimorar o sistema. A arquitetura foi revista, ajustando a forma como os dados são apresentados e filtrados, e melhorando aspectos de usabilidade. A Lei de Fitts (FITTS, 1954), que descreve a relação entre a distância e o tamanho dos objetos de interação, foi atendida ao ajustar o layout e os botões de forma a garantir que a interação fosse mais rápida e natural. A Lei de Hick (HICK, 1952), que sugere que o tempo de tomada de decisão aumenta com o número de opções, foi considerada ao simplificar os menus e filtros, proporcionando uma navegação mais ágil e intuitiva, visando reduzir a sobrecarga cognitiva do usuário.

Com a implementação final do sistema, o resultado é um sistema bem estruturado, que apresenta um fluxo de informações intuitivo, um layout agradável e uma experiência de uso fluida. O sistema está quase pronto para ser utilizado na FACOM, o que tornará o processo de automação do fluxo de estágio mais flexível, centralizado e acessível para todos

os envolvidos, desde professores até alunos. Embora a aplicação já esteja funcionando de forma eficiente, ainda há algumas áreas que poderiam ser aprimoradas para otimizar ainda mais a experiência do usuário. Por exemplo, a integração com outras plataformas de gestão acadêmica poderia ser considerada, e também a implementação do sistema e seus componentes em nuvem para utilização de todos os atores.

A implementação resultante mantém a máquina de estados original desenvolvida por (ROCHA, 2023), projetada e aqui adaptada para simplificar a criação e a execução dos fluxos interativos na aplicação. O trabalho buscou não apenas aprimorar o sistema, mas também explicar e documentar as telas da solução Web para a automação do fluxo de estágio, buscando torná-lo mais flexível, centralizado e facilitado para todos os atores envolvidos. O foco foi garantir que as funcionalidades fossem adaptadas para uma utilização mais intuitiva e eficaz, considerando sempre as necessidades dos usuários e os objetivos do sistema. No futuro, seria interessante explorar melhorias adicionais, como a implementação de funcionalidades de personalização do perfil do usuário.

Embora o Sistema SisFlow tenha alcançado melhorias significativas na automação do fluxo de estágios e na experiência do usuário, há várias áreas que podem ser exploradas em trabalhos futuros para continuar aprimorando o sistema. Como por exemplo a integração com o sistema da UFU, investigar e implementar medidas adicionais de segurança e privacidade de dados para proteger as informações sensíveis dos usuários, migrar o sistema para uma estrutura em nuvem e mudar a lógica de algumas funções, a fim de melhorar o desempenho.

Em resumo, o sistema desenvolvido representa um avanço significativo na automatização do fluxo de registro e acompanhamento de estágio, com uma interface que, ao seguir os princípios da Interação Humano-Computador e as heurísticas de Nielsen, se torna mais eficaz e fácil de usar. O sistema busca atender não apenas às necessidades dos usuários, mas também proporcionar uma experiência mais intuitiva, o que é essencial para qualquer aplicação que busque atender adequadamente seus usuários.

Referências

- ALMEIDA, T. W. d. G. **Desenvolvimento do sistema de controle de estágio do IF Sertão PE**. 117 p. Trabalho de Conclusão de Curso — Instituto Federal de Educação, Ciência e Tecnologia do Sertão Pernambucano, 2015. Citado na página 27.
- BALUZ, R.; ROCHA, F.; SOUZA, I. Plataforma web para gestão dos estágios supervisionados nos cursos de graduação. **Revista Eletrônica de Iniciação Científica em Computação**, v. 18, n. 2, p. 12, jul. 2020. Disponível em: <<https://sol.sbc.org.br/journals/index.php/reic/article/view/1720>>. Citado na página 27.
- CHOO, C. W. **A organização do conhecimento: busca e uso da informação nas organizações**. São Paulo: Saraiva, 2003. 425 p. Citado na página 41.
- FITTS, P. M. The information capacity of the human motor system in controlling the amplitude of movement. **Journal of Experimental Psychology**, v. 47, n. 6, p. 381–391, 1954. Citado na página 46.
- Flask Project. **Flask: Web Development, One Drop at a Time**. [S.l.], 2024. Acesso em: 21 dez. 2024. Disponível em: <<https://flask.palletsprojects.com/>>. Citado na página 20.
- HICK, W. E. On the rate of gain of information. **Quarterly Journal of Experimental Psychology**, v. 4, n. 1, p. 11–26, 1952. Citado na página 46.
- LEONCIO, D. **Sistema para controle de estágio supervisionado do departamento acadêmico de informática**. 55 p. Trabalho de Conclusão de Curso — Universidade Tecnológica Federal do Paraná, 2018. Citado na página 27.
- MDN Web Docs. **Object-Relational Mapping (ORM)**. [S.l.], 2024. Acesso em: 21 dez. 2024. Disponível em: <<https://developer.mozilla.org/en-US/docs/Glossary/ORM>>. Citado na página 20.
- MIRANDA, M. Z.; PAULA, F. S. de. **SGE – Sistema de Gerenciamento DE Estágios**. 117 p. Trabalho de Conclusão de Curso — Universidade Estadual do Mato Grosso do Sul, 2018. Citado na página 28.
- Mozilla Developer Network (MDN). **JavaScript: High-level, Dynamic, Untyped, and Interpreted Programming Language**. [S.l.], 2024. Acesso em: 21 dez. 2024. Disponível em: <<https://developer.mozilla.org/en-US/docs/Web/JavaScript>>. Citado na página 21.
- NIELSEN, J. Finding usability problems through heuristic evaluation. In: **Proceedings of the SIGCHI Conference on Human Factors in Computing Systems**. New York, NY, USA: Association for Computing Machinery, 1992. (CHI '92), p. 373–380. ISBN 0897915135. Disponível em: <<https://doi.org/10.1145/142750.142834>>. Citado 2 vezes nas páginas 29 e 43.

_____. **10 Usability Heuristics for User Interface Design**. 1994. <<https://www.nngroup.com/articles/ten-usability-heuristics/>>. Acesso em 22/03/2024. Acesso em: 13 jul. 2024. Citado 2 vezes nas páginas 15 e 31.

_____. **Usability Engineering**. San Francisco: Morgan Kaufmann, 1994. ISBN 978-0125184069. Citado 2 vezes nas páginas 15 e 30.

_____. **113 Design Guidelines for Homepage Usability**. 2001. <<https://www.nngroup.com/articles/113-design-guidelines-homepage-usability/>>. Acesso em 22/04/2024. Acesso em: 22 abr. 2024. Citado na página 17.

npm, Inc. **npm: Node Package Manager**. [S.l.], 2024. Acesso em: 21 dez. 2024. Disponível em: <<https://www.npmjs.com>>. Citado na página 21.

OFICIAL, D. O que é gestalt e como usar no design. **Designe**, 2021. Acesso em: 20 out. 2024. Disponível em: <<https://designe.com.br/o-guia-do-designer-para-a-teoria-gestalt/>>. Acesso em: 20 outubro 2024. Citado na página 37.

Oracle Corporation. **MySQL: The world's most popular open source database**. [S.l.], 2024. Acesso em: 21 dez. 2024. Disponível em: <<https://www.mysql.com>>. Citado na página 23.

PEREIRA, E. C. d. O. **Sistema de alerta de limites e prazos acadêmicos para discentes de graduação da FACOM**. 50 p. Trabalho de Conclusão de Curso — Universidade Federal de Uberlândia, 2023. Citado na página 28.

PERINI, L. C.; YAMAMOTO, L. M. Sistema de gerenciamento de estágio supervisionado-módulo administrativo. In: **10o Encontro de Atividades Científicas da UNOPAR**. [S.l.: s.n.], 2007. p. 8. Citado na página 27.

Python Software Foundation. **Python Programming Language**. [S.l.], 2024. Acesso em: 21 dez. 2024. Disponível em: <<https://www.python.org>>. Citado na página 20.

ROCHA, V. C. **Uma aplicação Web para auxiliar o fluxo de estágios da FACOM: Detalhamento de sua arquitetura, dos fluxos e da máquina de estados**. 117 p. Trabalho de Conclusão de Curso — Universidade Federal de Uberlândia, 2023. Citado 12 vezes nas páginas 5, 8, 12, 17, 18, 20, 22, 24, 25, 26, 46 e 47.

Roy Fielding et al. **REST API: Representational State Transfer**. [S.l.], 2000. Conceito de APIs RESTful. Acesso em: 21 dez. 2024. Disponível em: <<https://restfulapi.net>>. Citado na página 20.

SELES, J. C. **Desenvolvimento de sistema para gestão de estágio**. 114 p. Trabalho de Conclusão de Curso — Centro Paula Souza, 2018. Citado na página 27.

SOUZA, I. B. S. de; ROCHA, F. das C.; BALUZ, R. A. R. S. Sigestagios: Sistema web para controle de estágios supervisionados no ensino superior. **Revista Eletrônica de Sistemas de Informação e Gestão Tecnológica**, v. 10, n. 1, p. 10, 2019. Citado na página 27.

SQLAlchemy Team. **SQLAlchemy: The Database Toolkit for Python**. [S.l.], 2024. Acesso em: 21 dez. 2024. Disponível em: <<https://www.sqlalchemy.org>>. Citado na página 20.

VESPUCCI, K. M. Daltônicos ao volante. **Revista Trânsito da CET-SP**, p. 1–7, 2009. Citado na página 39.

Vue.js Team. **Vue.js: The Progressive JavaScript Framework**. [S.l.], 2024. Acesso em: 21 dez. 2024. Disponível em: <<https://vuejs.org>>. Citado na página 21.

W3C. **CSS: Cascading Style Sheets**. [S.l.], 2024. Acesso em: 21 dez. 2024. Disponível em: <<https://www.w3.org/Style/CSS/>>. Citado na página 21.

_____. **HTML: HyperText Markup Language**. [S.l.], 2024. Acesso em: 21 dez. 2024. Disponível em: <<https://html.spec.whatwg.org/>>. Citado na página 21.

WAZLAWICK, R. S. **Metodologia de pesquisa para ciência da computação**. [S.l.]: Elsevier, Rio de Janeiro, 2009. v. 2. 168 p. ISBN 978-8535277838. Citado na página 29.