

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Douglas Gomes de Paula

**Projeto e Implementação de um Aplicativo iOS
para Apoio à Jornada Acadêmica dos
Estudantes no PPGCO**

Uberlândia, Brasil

2024

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Douglas Gomes de Paula

**Projeto e Implementação de um Aplicativo iOS para
Apoio à Jornada Acadêmica dos Estudantes no PPGCO**

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 Ciência da Computação.

Orientador: Rafael Dias Araújo

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2024

Douglas Gomes de Paula

Projeto e Implementação de um Aplicativo iOS para Apoio à Jornada Acadêmica dos Estudantes no PPGCO

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 Ciência da Computação.

Trabalho aprovado. Uberlândia, Brasil, 22 de novembro de 2024:

Rafael Dias Araújo
Orientador

Bruno Augusto Nassif Travençolo
FACOM/UFU

Marcelo de Almeida Maia
FACOM/UFU

Uberlândia, Brasil
2024

Dedico este trabalho ao meu pai, Gilmar, pois sem seu apoio incondicional e sua dedicação, eu jamais seria quem sou ou teria chegado até aqui. A ele, toda minha gratidão e respeito.

Agradecimentos

Agradeço ao meu pai, Gilmar, por todo o esforço e dedicação em me manter estudando, aos meus amigos pelo apoio constante e palavras amigas que me motivaram nos momentos difíceis, e aos professores que, com seu conhecimento e paciência, foram fundamentais para o meu crescimento acadêmico.

Resumo

No ambiente acadêmico, a gestão do tempo pelos estudantes é uma dificuldade recorrente, podendo comprometer o bem-estar dos estudantes e impactar diretamente no seu desempenho. Em níveis avançados, como Mestrado e Doutorado, o baixo desempenho é uma das principais causas de evasão e uma forma de evitá-la é fazendo uma boa gestão de tempo. No Programa de Pós-graduação em Ciência da Computação na Universidade Federal de Uberlândia não há um sistema de informação integrado que auxilia essa jornada, fazendo com que estudantes e gestores recorram ao uso de planilhas ou outras ferramentas menos apropriadas. Dessa forma, uma aplicação móvel pode ser usada auxiliando na gestão das entregas, acesso ao progresso do curso, gestão das disciplinas, notas e informando em relação as etapas do programa. Este trabalho propõe o desenvolvimento de um aplicativo que permita ao estudante acompanhar a sua jornada no Programa de Pós-graduação em Ciência da Computação. Portanto, foi desenvolvido um protótipo e implementado um aplicativo iOS onde o estudante pode se informar e gerenciar, em certo nível, a sua jornada acadêmica. Dada a complexidade disso, este trabalho contempla a implementação sem integração com um servidor externo, pois esta etapa está sendo desenvolvida em outro trabalho de conclusão de curso. Sendo assim, ao final deste trabalho, haverá um protótipo de um aplicativo iOS, que ao ser integrado ao servidor, permitirá ao estudante ver seus créditos integralizados, acessar suas disciplinas e registrar suas notas, adicionar publicações, ver as regras para defesa da dissertação ou tese, acessar os marcos temporais do programa, enviar documentação obrigatória e receber recados da secretaria. Para a prototipação foi utilizado o Figma e para o desenvolvimento do aplicativo foi utilizado o Xcode.

Palavras-chave: PPGCO, Aplicativo, iOS, Mobile, Desenvolvimento.

Abstract

In the academic environment, time management is a recurring challenge for students, potentially compromising their well-being and directly impacting their performance. In advanced levels, such as Master's and Doctorate programs, low performance is one of the main causes of dropout, and a way to avoid it is through effective time management. In the Graduate Program in Computer Science at the Federal University of Uberlândia, there is no integrated information system to support this journey, causing students and administrators to rely on spreadsheets or other less suitable tools. In this way, a mobile application can assist in managing deliverables, accessing course progress, managing subjects, grades, and informing about program milestones. This work proposes the development of an application that allows students to monitor their journey in the Graduate Program in Computer Science. Therefore, a prototype was developed, and an iOS application was implemented where students can access information and manage their academic journey to a certain extent. Given the complexity, this work includes implementation without integration with an external server, as this stage is being developed in another capstone project. Thus, at the end of this work, there will be a prototype of an iOS application, which, when integrated with the server, will allow students to view their completed credits, access their subjects and record their grades, add publications, view the rules for thesis or dissertation defense, access program milestones, submit mandatory documentation, and receive messages from the program's office. Figma was used for prototyping and Xcode was used for application development.

Keywords: PPGCO, App, iOS, Mobile, Development.

Lista de ilustrações

Figura 1 – Padrão Arquitetural MVVM	18
Figura 2 – Exemplo da tela do Figma	22
Figura 3 – Exemplo da tela do Xcode	23
Figura 4 – Arquitetura do Aplicativo	24
Figura 5 – Caso de Uso Geral	25
Figura 6 – Fluxo de <i>Login</i>	32
Figura 7 – Fluxo de Cadastro	33
Figura 8 – Fluxo de Recuperação de Senha	34
Figura 9 – Tela Inicial	35
Figura 10 – Fluxo de Perfil	36
Figura 11 – Fluxo de Alteração de Senha	37
Figura 12 – Fluxo de Configuração	38
Figura 13 – Próxima entrega na tela inicial	39
Figura 14 – Fluxo de Ajuda	40
Figura 15 – Fluxo de Recados	41
Figura 16 – Fluxo de Disciplinas	42
Figura 17 – Fluxo de Publicações	43
Figura 18 – Fluxo de Dissertação	44
Figura 19 – Fluxo de Marcos Temporais	45
Figura 20 – Carregamento do botão	46
Figura 21 – Mensagem de erro	46
Figura 22 – Progresso de créditos	46
Figura 23 – Botões na Tela Principal	47
Figura 24 – Disciplinas que possuem navegação	47
Figura 25 – Toolbar da tela de Disciplinas	47
Figura 26 – Campo e-mail inválido	48
Figura 27 – Botões do menu lateral	49
Figura 28 – Atalhos na tela principal	49
Figura 29 – Tela principal e Menu lateral	50
Figura 30 – Campo com erro e botão desabilitado	50
Figura 31 – Atalho para enviar e-mail na tela de Ajuda	51

Lista de abreviaturas e siglas

API Interface de Programação de Aplicativo. [23](#), [54](#), [55](#)

CR Coeficiente de Rendimento Global. [13](#), [14](#)

FACOM Faculdade de Computação. [12](#)

FAQ Perguntas frequentes. [21](#), [30](#)

MVC *Model-View-Controller*. [17](#)

MVP *Model-View-Presenter*. [17](#)

MVVM *Model-View-ViewModel*. [17](#)

PPGCO Programa de Pós-graduação em Ciência da Computação. [11–13](#), [19–21](#), [30](#), [33](#), [42](#), [45](#), [51](#), [52](#), [54](#)

UFU Universidade Federal de Uberlândia. [51](#), [52](#), [55](#)

UI *User Interface*. [14](#), [21](#)

UX *User Experience*. [14](#), [21](#)

VIPER *View-Interactor-Presenter-Entity-Router*. [17](#)

Sumário

1	INTRODUÇÃO	11
1.1	Objetivos	12
1.2	Organização do trabalho	12
2	REVISÃO BIBLIOGRÁFICA	13
2.1	Programa de Pós-graduação em Ciência da Computação	13
2.2	UX/UI	14
2.2.1	Heurísticas de Nielsen	14
2.3	Aplicativo móvel	16
2.3.1	iOS	16
2.3.2	Swift	16
2.3.3	SwiftUI	17
2.4	Arquitetura de <i>Software</i>	17
2.4.1	Model-View-ViewModel	17
2.5	Trabalhos correlatos	18
3	MATERIAIS E MÉTODOS	20
3.1	Requisitos	20
3.2	Tecnologias utilizadas	21
3.2.1	Figma	21
3.2.2	Xcode	22
3.2.3	URLSession	23
3.3	Arquitetura	23
3.4	Casos de uso	24
3.4.1	UC01 - Cadastrar Usuário	25
3.4.2	UC02 - Autenticar Usuário	26
3.4.3	UC03 - Recuperar Senha	26
3.4.4	UC04 - Cadastrar/Editar Disciplinas	27
3.4.5	UC05 - Cadastrar/Editar Publicações	27
3.4.6	UC06 - Visualizar Recados	27
3.4.7	UC07 - Visualizar Créditos Integralizados/Faltantes	28
3.4.8	UC08 - Visualizar de Marcos Temporais	28
3.4.9	UC09 - Enviar Documentação Comprobatória	28
3.4.10	UC10 - Visualizar Informações sobre a Dissertação/Defesa	29
3.4.11	UC11 - Visualizar/Editar Dados Pessoais	29
3.4.12	UC12 - Ativar/Desativar Notificações sobre Marcos Temporais	29

3.4.13	UC13 - Acessar o FAQ do PPGCO	30
4	RESULTADOS E DISCUSSÕES	31
4.1	Login	31
4.2	Cadastro	32
4.3	Recuperação de Senha	34
4.4	Tela Inicial	35
4.5	Perfil	36
4.6	Alterar Senha	36
4.7	Configuração	37
4.8	Ajuda	40
4.9	Recados	40
4.10	Disciplinas	41
4.11	Publicações	42
4.12	Dissertação/Tese	43
4.13	Marcos Temporais	44
4.14	Validação das Heurísticas de Nielsen	45
4.14.1	Visibilidade do status do sistema	45
4.14.2	Compatibilidade entre o sistema e o mundo real	46
4.14.3	Controle e liberdade para o usuário	47
4.14.4	Consistência e padronização	48
4.14.5	Prevenção de erros	48
4.14.6	Reconhecimento em vez de memorização	48
4.14.7	Eficiência na usabilidade	49
4.14.8	Estética e design minimalista	49
4.14.9	Ajude os usuários a reconhecerem, diagnosticarem e recuperarem-se de erros	50
4.14.10	Ajuda e documentação	51
4.15	Implantação	51
4.16	Discussão	52
5	CONCLUSÃO	54
5.1	Trabalhos futuros	54
5.1.1	Integração com o servidor	54
5.1.2	Implementação de uma aplicação Android	55
5.1.3	Melhorias no aplicativo	55
	REFERÊNCIAS	56

1 Introdução

No ambiente acadêmico, a gestão do tempo pelos estudantes é uma dificuldade recorrente. Segundo [Soares et al. \(2023\)](#), a má administração do tempo pode elevar os níveis de estresse e comprometer o bem-estar dos estudantes, impactando diretamente o desempenho acadêmico. Esse problema afeta alunos em todos os níveis de ensino, já que, quanto mais elevado o estresse, menor tende a ser o rendimento acadêmico ([MIRANDA et al., 2022](#)). Em níveis avançados de estudos, como Mestrado e Doutorado, o baixo desempenho é uma das principais causas de evasão ([AUGUSTO et al., 2019](#)).

Para superar esse desafio é importante adquirir habilidades de gestão de tempo, pois segundo [MacCann, Fogarty e Roberts \(2012\)](#), a capacidade de planejar e organizar as atividades é um fator crucial para um bom desempenho acadêmico, embora uma parcela significativa da população não saiba como organizar seus horários e compromissos ([BARBOSA, 2018](#)). No nível de pós-graduação, a situação piora quando o estudante precisa se adequar a novas exigências acadêmicas e administrativas, precisando recorrer a regulamentos extensos para entender as normas do programa, gerenciar as entregas, os prazos e muitas das vezes conciliar com a vida profissional.

No contexto do [Programa de Pós-graduação em Ciência da Computação \(PPGCO\)](#), não há um sistema de informação integrado que permite que estudantes, orientadores, coordenadores e secretários acompanhem a jornada acadêmica dos estudantes matriculados nos cursos de mestrado e doutorado. A dinâmica do programa de pós-graduação é pautado por extensos regulamentos (tanto internos quanto externos) e, atualmente, a jornada acadêmica dos estudantes é controlada por planilhas. Embora esses métodos sejam úteis, eles apresentam limitações na organização, atualização e acessibilidade das informações, o que dificulta o gerenciamento da jornada acadêmica tanto para o aluno quanto para os gestores do programa. Dada esta situação, projetar e implementar uma aplicação que integre e controle essas informações é uma solução eficaz para melhorar a eficiência do programa, permitindo que o tempo gasto na gestão dessas informações possa ser utilizado de forma mais significativa para atividades de aprendizagem.

Considerando os desafios que os estudantes enfrentam além da jornada acadêmica, acessar as informações de qualquer lugar é uma necessidade crucial a ser atendida por essa solução, logo um aplicativo para *smartphone* é o mais indicado. Segundo [Conexis \(2018\)](#), 97% dos estudantes acessam a internet através do celular e, para 18% dos alunos, ele é o único dispositivo de acesso a rede.

Dessa forma, este trabalho se baseia no Regulamento do Programa de Pós-graduação em Ciência da Computação ([FACOM, 2021](#)) para implementar um aplicativo com fun-

cionalidades auxiliem os alunos de Mestrado e Doutorado do [PPGCO](#) em sua jornada acadêmica. Devido a familiaridade com o desenvolvimento iOS, este trabalho está focado nesta plataforma específica, porém um aplicativo Android poderá ser desenvolvido futuramente utilizando o mesmo protótipo, que foi desenvolvido aplicando as Heurísticas de Nielsen ([NIELSEN, 2024](#)) para propiciar uma melhor usabilidade para os usuários.

1.1 Objetivos

O objetivo deste trabalho é projetar e implementar uma aplicação iOS que atenda às necessidades dos alunos do [PPGCO](#), ajudando a gerenciar suas entregas e se informar do andamento da sua jornada acadêmica, em termos do cumprimento dos requisitos para término do curso.

Para realizar o trabalho, faz-se necessário os seguintes objetivos específicos:

- Levantamento e análise dos requisitos com base no regulamento interno do [PPGCO/Faculdade de Computação \(FACOM\)](#).
- Construção do protótipo das telas, baseando-se nas Heurísticas de Nielsen.
- Implementação das telas e preparo para integração com um sistema administrativo.

1.2 Organização do trabalho

O trabalho é organizado como se segue. O Capítulo 2 contém a revisão bibliográfica e os trabalhos correlatos que inspiraram este trabalho; o Capítulo 3 contém os requisitos, as tecnologias utilizadas no desenvolvimento, a arquitetura da aplicação e os casos de uso; o Capítulo 4 contém a aplicação desenvolvida, apresentando suas telas e funcionalidades, as etapas para a implantação e discussão sobre o desenvolvimento do trabalho; e o Capítulo 5 contém as considerações finais e trabalhos futuros.

2 Revisão Bibliográfica

Neste capítulo serão abordados todos os tópicos necessários para embasar o projeto e o desenvolvimento do aplicativo proposto por este trabalho.

2.1 Programa de Pós-graduação em Ciência da Computação

O Programa de Pós-graduação em Ciência da Computação ([PPGCO](#)) é um programa avançado para graduados em Ciência da Computação e/ou áreas semelhantes, projetado para que os alunos aprofundem e ampliem os conhecimentos adquiridos ao longo da graduação e descubram novos tópicos especializados que estão constantemente evoluindo no campo da computação. Segundo [Facom \(2021\)](#), o objetivo do programa é qualificar e formar novos docentes, pesquisadores e profissionais nas diversas áreas relacionadas a computação, oferecendo cursos nas modalidades de Mestrado e Doutorado.

O [PPGCO](#) estabelece prazos claros para a conclusão dos cursos. O mestrado tem como duração mínima 12 meses e duração máxima de 24 meses, podendo estender a duração para 30 meses em situações excepcionais, de acordo com o Art. 27 ([FACOM, 2021](#)). Já o doutorado tem duração mínima de 24 meses e duração máxima de 48 meses, podendo ter a duração máxima de até 60 meses em casos excepcionais, de acordo com o Art. 28 ([FACOM, 2021](#)).

Para integralizar o curso são exigidos créditos que variam de acordo com a titulação. No mestrado, os alunos precisam integralizar 44 créditos, sendo 25 em disciplinas e 18 relacionados à defesa da dissertação, de acordo com o Art. 30 ([FACOM, 2021](#)). No doutorado, o número de créditos pode variar entre 42 e 52 dependendo da área de formação anterior do aluno, de acordo com Art. 31 e Art. 32 ([FACOM, 2021](#)).

O sistema de avaliação do aproveitamento dos alunos pode ser visto no Art. 30 ([FACOM, 2021](#)), utilizando conceitos de “A” a “E”, onde “A” representa um aproveitamento total e “E” representa reprovação. Através dessas notas, ao final de cada período letivo, é feita uma média ponderada de todos os níveis de conceitos atribuídos até então, sendo os pesos o número de créditos das disciplinas e atribuindo valores aos níveis: “A” vale 4 pontos por crédito, “B” vale 3 pontos por crédito, “C” vale 2 pontos por crédito, “D” vale 1 pontos por crédito e “E” vale 0 pontos. Essa média é chamada de [Coeficiente de Rendimento Global \(CR\)](#).

Os Art. 47 a 49 ([FACOM, 2021](#)) parametriza quais critérios o aluno deve cumprir para poder fazer a defesa da dissertação de mestrado. Ele deve estar matriculado na disciplina de Dissertação de Mestrado, ter sido aprovado em exame de proficiência em

inglês, ter **CR** de pelo menos 2,5 e ter cumprido 26 créditos. Além disso, o aluno também precisa apresentar produção científica, como pedidos de patentes ou publicações, conforme exigências do programa e da CAPES.

Os Art. 59 a 61 (FACOM, 2021) regulamenta a defesa da tese de doutorado seguindo uma estrutura semelhante à do mestrado. Depois de integralizar os créditos em disciplinas, o aluno deve estar matriculado na disciplina de Tese de Doutorado, ter sido aprovado no exame de proficiência em inglês e no exame de qualificação, além de cumprir com os requisitos de produção científica estabelecidos pelo programa.

2.2 UX/UI

No desenvolvimento de um aplicativo há uma série de etapas importantes antes de chegar de fato na implementação do código fonte. Uma dessas etapas envolve o desenvolvimento da Experiência do Usuário, conhecida como *User Experience (UX)*, e da Interface do Usuário, conhecida como *User Interface (UI)*, ambas cruciais no design de aplicativos móveis a fim de garantir a satisfação e usabilidade do usuário. A **UI** é responsável por garantir um visual agradável, assertivo e interativo para o usuário, enquanto a **UX** se preocupa em como a **UI** com todos os seus componentes e possibilidades de interação pode afetar a experiência geral do usuário ao utilizar o aplicativo.

No desenvolvimento da **UI**, existe uma série de ferramentas para desenvolvimento de protótipos que serão utilizados no desenvolvimentos de aplicativos e, atualmente, as ferramentas **Sketch** e **Figma** são as mais utilizadas. Essas ferramentas permitem aos designers experimentar várias ideias e iterações, tendo controle sobre o *design* de componentes, tipografias e a documentação de todo o protótipo, facilitando todo o processo de criação de uma interface atraente e funcional.

Já no que diz respeito a **UX**, também existe uma série de métodos utilizados pelos designers para chegar na experiência desejada para o aplicativo móvel como: personas, análise competitiva, entrevista com partes interessadas (*stakeholders interviews*), *brainstorming*, *storyboards*, fluxo de usuário (*user flow*), análise heurística (como as Heurísticas de Nielsen (NIELSEN, 2024)), entre outras.

Em resumo, **UI** e **UX** são complementares no design de aplicativos móveis que, com o uso de ferramentas adequadas, desempenham um papel muito importante no desenvolvimento de soluções modernas que atendam as necessidades e expectativas dos usuários.

2.2.1 Heurísticas de Nielsen

Como citado anteriormente, há uma série de métodos utilizados pelos designers para garantir que o aplicativo projetado tenha a experiência desejada, isto é, projetar

interfaces que não gere inseguranças durante a interação do usuário e que garanta sua utilização de forma simples, eficiente e sem precisar muitas instruções. Considerando esse objetivo, Jacob Nielsen criou dez heurísticas que auxiliam no projeto de uma boa interface que ofereça uma ótima experiência de uso, sendo elas:

1. **Visibilidade do status do sistema:** a interface deve sempre manter o usuário atualizado em relação ao seu status através de *feedback* dentro de um tempo apropriado. Segundo Nielsen (2024), quando o *status* é conhecido pelo usuário, é possível entender o impacto que suas ações causam no sistema, permitindo que ele aprenda com as suas ações anteriores e preveja o que acontecerá em uma próxima ação; interações previsíveis geram confiança.
2. **Compatibilidade entre o sistema e o mundo real:** use elementos do contexto do usuário, como palavras, frases, símbolos, imagens, etc. O *design* deve ser familiar para o usuário, seguindo convenções do mundo real para que pareça natural e lógico. Segundo Nielsen (2024), é mais fácil para o usuário ter uma experiência intuitiva quando o *design* segue convenções do mundo real, tornando mais fácil aprender e lembrar como a interface funciona.
3. **Controle e liberdade para o usuário:** o usuário deve ter controle sobre suas ações e a liberdade de desfazê-las se a anterior for feita por engano. Segundo Nielsen (2024), o usuário tem a sensação de liberdade e confiança quando poder desistir de um processo, ou desfazer uma ação, se assim decidirem. A interface deve ter clareza nas formas de controle que o usuário tem sobre ela.
4. **Consistência e padronização:** não pode haver ambiguidade entre elementos da interface, isto é, o usuário não deve ter dúvidas sobre a usabilidade de um mesmo elemento em situações diferentes. Segundo Nielsen (2024), deixar de manter a consistência pode forçar os usuários a aprender algo novo sempre, tornando a usabilidade um processo exaustivo e uma interface deve utilizar de padrões pra tornar a experiência fácil e leve.
5. **Prevenção de erros:** é importante ter mensagens de erro efetivas, mas é mais importante ainda evitar que os erros aconteçam. A interface deve evitar dar opções que leve o usuário a cometer um erro.
6. **Reconhecimento em vez de memorização:** a interface deve ensinar o usuário de tal forma que ele não precise lembrar dos seus elementos para tomar decisões, ter elementos que possibilitem recuperar seu significado sem ter que memorizar. Segundo Nielsen (2024), interfaces bem projetadas reduzem o esforço cognitivo dos usuários através do reconhecimento dos elementos presentes.

7. **Eficiência e flexibilidade de uso:** interfaces que permitem flexibilização de ações frequentes são mais atraentes. O usuário vai se sentir mais a vontade com interfaces que permitem que ele as utilize da maneira que é mais conveniente para ele, isso vai do menos experiente ao usuário mais experiente.
8. **Estética e *design* minimalista:** a interface deve manter apenas elementos relevantes para o usuário. De acordo com [Nielsen \(2024\)](#), é importante que os elementos visuais da interface apoiem os objetivos principais do usuário.
9. **Ajude os usuários a reconhecerem, diagnosticarem e recuperarem-se de erros:** todos os erros devem ser expressos em linguagem simples e possibilitar que o usuário saiba qual ação tomar em relação ao mesmo.
10. **Ajuda e documentação:** informações adicionais devem ser acessíveis, claras e concretas. O usuário deve conseguir encontrar e utilizar as informações de forma simples quando necessárias.

2.3 Aplicativo móvel

Aplicativo móvel, também conhecido como *app*, é um sistema feito para dispositivos móveis como *smartphones*, *tablets*, ou qualquer dispositivo que seja portátil. Atualmente há diversas empresas que produzem e vendem *smartphones* e *tablets* onde os *apps* podem ser utilizados e grande parte desses dispositivos fazem uso dos sistemas operacionais [Android](#) ou [iOS](#), que possuem lojas onde os desenvolvedores podem publicar esses aplicativos e os usuários podem baixá-los.

2.3.1 iOS

O [iOS](#) o sistema operacional do iPhone, dispositivo móvel da Apple. Segundo [Haas \(2024\)](#) no site Canal Tech, o iOS foi proposto por Steve Jobs como uma versão do macOS — o sistema operacional para computadores Mac — adaptada para celulares e com tecnologia multitoque.

2.3.2 Swift

De acordo com a documentação da [Apple \(2024a\)](#), Swift é uma linguagem de programação poderosa e intuitiva para todas as plataformas Apple. É uma linguagem moderna, segura que tem boa performance e possui código aberto, permitindo que vários desenvolvedores possam participar da comunidade, junto a [Swift.org](#), que trabalha para tornar a linguagem cada vez melhor.

2.3.3 SwiftUI

SwiftUI é a ferramenta de desenvolvimento declarativo de interfaces para aplicativos iOS. De acordo com a [Apple \(2024b\)](#), o SwiftUI ajuda o desenvolvedor a criar aplicativos com ótima aparência para todas as plataformas Apple, incluindo o iOS, utilizando o Swift e com pouco código. Com essa ferramenta é possível ter um controle avançado das animações dos elementos na tela, fluxos de dados simplificados, criar aplicativos espaciais voltados para realidade aumentada, entre várias outras vantagens, tudo utilizando de um paradigma declarativo utilizando Swift. O [Xcode](#), editor de código oficial para desenvolvimento para as plataformas Apple, possui um visualizador dedicado para o SwiftUI que atualiza conforme o desenvolvedor declara o código da interface.

2.4 Arquitetura de *Software*

A arquitetura de *software* é como o esqueleto de um sistema. Ela define como os componentes se relacionam, como as pastas e os códigos são organizados no projeto e como o sistema pode evoluir. No contexto de aplicativos móveis, a arquitetura de *software* é ainda mais crucial, pois impacta diretamente na escalabilidade, testabilidade e manutenção do aplicativo. Uma boa arquitetura de *software* inclui a separação de responsabilidades, projetando a forma como a lógica de negócios, a interface do usuário e o acesso aos dados serão implementados, buscando facilitar a manutenção e a colaboração entre os desenvolvedores do projeto. Também procura permitir que o aplicativo cresça sem comprometer o desempenho ou a estabilidade e a escrita de testes automatizados para garantir a qualidade do código e a prevenção de problemas.

Existem vários padrões arquiteturais no desenvolvimento iOS, sendo de responsabilidade do arquiteto de *software* definir qual é melhor para o projeto. Existem diversas arquiteturas, porém as mais utilizadas em desenvolvimento iOS são [Model-View-Controller \(MVC\)](#), [Model-View-Presenter \(MVP\)](#), [View-Interactor-Presenter-Entity-Router \(VIPER\)](#) e [Model-View-ViewModel \(MVVM\)](#). Contudo, neste trabalho não serão detalhadas as arquiteturas [MVC](#), [MVP](#) e [VIPER](#), pois não serão utilizadas. Neste trabalho será utilizado padrão arquitetural [MVVM](#) pela familiaridade com a arquitetura e por ser uma das mais usadas em SwiftUI ([COSME, 2024](#)).

2.4.1 Model-View-ViewModel

Segundo [Coodesh \(2024\)](#), o padrão [MVVM](#) foi apresentado por John Grossman em 2005, desde então passou a ser muito usado no desenvolvimento *mobile*. O objetivo principal desse padrão arquitetural é separar a camada de *View* (que controla as interfaces do sistema) da camada *Model* (que controla as entidades que modelam a regra de negócio), utilizando uma camada intermediária independente, isto é, não tem relação 1:1 com

apenas uma *View*. A camada chamada de *ViewModel* permite que outras *Views* também a observem sem precisar de ajustes para evitar a quebra do sistema, pois a sua existência independe das regras de funcionamento da *View*. Por ser uma camada intermediária, a *ViewModel* receberá requisições da *View* quando houver eventos por parte do usuário na interface do sistema, tratando esses eventos e interagindo para com a *Model*. A *ViewModel* também ficará observando a *Model* até que haja uma resposta e, ao recebê-la, devolverá a resposta ao evento anterior para que a *View* se atualize.

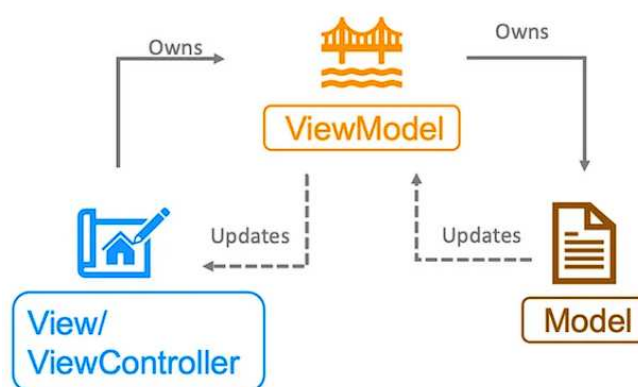


Figura 1 – Padrão Arquitetural MVVM.

Fonte: Hsin, 2022¹

2.5 Trabalhos correlatos

Nesta seção serão citados alguns trabalhos que serviram como inspiração para o desenvolvimento deste trabalho.

O aplicativo Controle Universitário (ANNA, 2021) foi desenvolvido para Android e permite que o usuário cadastre suas atividades durante o curso para ter maior controle das suas disciplinas, notas, faltas, provas e trabalhos. Ele também notifica o usuário sobre a proximidade de tarefas importantes, além de possuir uma visualização do progresso geral do curso através da sua grade curricular que pode ser carregada com base em um modelo criado. Apesar de ser desenvolvido para iOS, este trabalho permitirá o usuário marcar quais disciplinas ele está fazendo para ter um melhor controle assim como o trabalho citado e também notificará o usuário sobre datas importantes de acordo com os marcos temporais.

O MAAS, Sistema para Integração de Compromissos Acadêmicos, proposto por Silva e UEDA (2013), visa facilitar o gerenciamento de compromissos acadêmicos por meio de um aplicativo móvel e um sistema web. Nesse sistema, os professores cadastram

¹ Disponível em: <<https://medium.com/dbs-tech-blog/how-to-successfully-implement-model-view-viewmodel-mvvm-architecture-pattern-in-ios-app-8ef97657f180>>. Acesso em 07/08/2024.

as atividades, como provas e trabalhos, permitindo que os alunos acompanhem essas informações pelo aplicativo. Este trabalho tem a semelhança de ser o desenvolvimento da aplicação móvel que será usada pelos alunos do [PPGCO](#), e o sistema web desenvolvido em outro trabalho será usado pelos gestores do programa.

[Trello](#), [ClickUp](#), [Jira](#) e outros programas de gestão de projeto também poderiam ser usados para esta finalidade. Contudo, todo o controle e personalização da jornada do estudante ficaria sobre responsabilidade do próprio estudante.

3 Materiais e métodos

Este capítulo descreve o desenvolvimento do trabalho. A seção 3.1 mostra os requisitos levantados para desenvolvimento do aplicativo e a seção 3.2 as tecnologias utilizadas no desenvolvimento. Na seção 3.3 mostra a arquitetura e organização do código do aplicativo. Na seção 3.4 os casos de uso que o aplicativo implementa.

3.1 Requisitos

Através de reuniões, entre o orientador e os alunos envolvidos no projeto, discutiu-se o escopo da aplicação: quem usaria a aplicação? Quais dados seriam necessários? Qual plataforma seria desenvolvida (considerando o prazo)?. Ficou decidido que nesse primeiro momento, a aplicação móvel seria desenvolvida para iOS, seria destinada aos estudantes e a integração com o sistema Web administrativo seria feito em outro momento, dado que está sendo realizado em outro trabalho de conclusão de curso.

Através do estudo do regulamento interno do [PPGCO \(FACOM, 2021\)](#) foram levantados outros itens que entrariam nesse escopo e que modelam a jornada do estudante e como a aplicação deve ser em questão de requisitos funcionais:

1. Deve ser capaz de cadastrar o usuário por e-mail e matrícula.
2. Deve ser capaz de fazer a autenticação do usuário com e-mail e senha.
3. Deve ser capaz de recuperar a senha.
4. Deve ser capaz de cadastrar/editar disciplinas.
5. Deve ser capaz de cadastrar/editar publicações.
6. Deve ser capaz de receber recados.
7. Deve ser capaz de ver os créditos que faltam integralizar e os integralizados.
8. Deve ser capaz de ver os marcos temporais.
9. Deve ser capaz de enviar documentação comprobatória quando necessária.
10. Deve ser capaz de ver informações sobre a dissertação e defesa.
11. Deve ser capaz de ver informações pessoais.
12. Deve ser capaz de editar e-mail e lattes.

13. Deve ser capaz de alterar senha.
14. Deve ser capaz de ativar/desativar notificações sobre marcos temporais.
15. Deve ser capaz de acessar o [Perguntas frequentes \(FAQ\)](#) do PPGCO.

Os requisitos não funcionais do aplicativo, que complementam os requisitos funcionais, são:

1. Usabilidade:

- A interface do usuário deve seguir as Heurísticas de Nielsen.
- A interface deve ser consistente em todo o sistema, utilizando componentes padrões do iOS.

2. Confiabilidade:

- O sistema deve ser capaz de lidar com falhas, minimizando o impacto nos usuários.

3.2 Tecnologias utilizadas

A tecnologias que foram utilizadas para desenvolver este trabalho estão descritas abaixo. A seção 3.2.1 mostra o Figma, utilizado na prototipagem do aplicativo. A seção 3.2.2 mostra o Xcode, programa utilização no desenvolvimento do aplicativo. Na seção 3.2.3 mostra o URLSession, *framework* utilizado para que o aplicativo possa ser integrado ao servidor web futuramente.

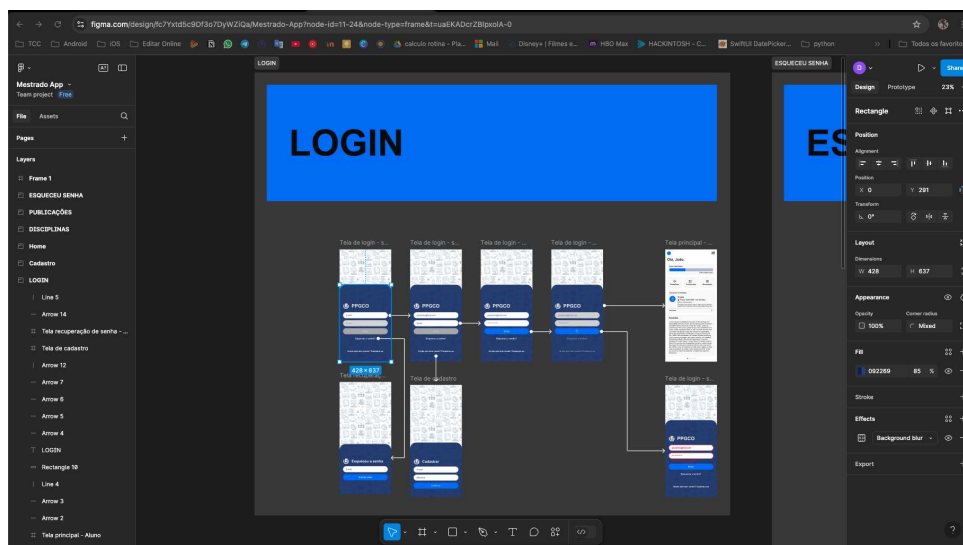
3.2.1 Figma

O [Figma](#) é um editor online que permite a criação de interfaces para websites, aplicativos e outros produtos digitais. A ferramenta oferece um arsenal de recursos, incluindo ferramentas de desenho vetorial, *layout* de elementos, prototipagem interativa, colaboração em tempo real, bibliotecas de componentes e integrações com outras ferramentas.

Diversos profissionais utilizam o Figma devido a sua versatilidade, como: Designers de UI/UX, Desenvolvedores, Gerentes de produto, entusiastas de design, entre outros. Há algumas características que fazem com que vários desses profissionais gostem de trabalhar com essa ferramenta, como sua acessibilidade que permite que ela seja utilizada em qualquer lugar, qualquer hora e por qualquer dispositivo. É possível colaborar com outras pessoas em tempo real, com uma interface simples e visualização dinâmica, podendo observar como cada pessoa modifica o projeto instantaneamente. A comunidade que utiliza

a ferramenta é muito grande, sendo assim há sempre alguém para discutir novas ideias ou pedir ajuda com alguma questão. Mas como toda ferramenta, há algumas questões que a impedem de ser perfeita, como a necessidade de estar sempre conectado a internet para ter funções colaborativas. Não possui recursos avançados de ilustração e edição de foto e, apesar de ter uma interface simples, também exige uma curva de aprendizado para utilizar todos os recursos.

Figura 2 – Exemplo da tela do Figma

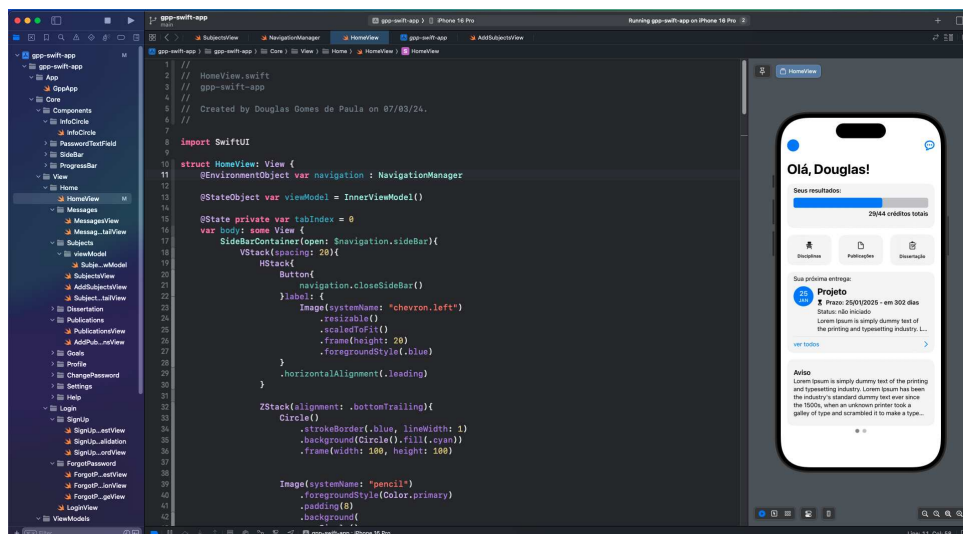


Fonte: Elaborada pelo Autor.

3.2.2 Xcode

Xcode é um conjunto de ferramentas utilizadas pelos desenvolvedores para criar aplicativos para as plataformas da Apple. No Xcode há um editor de código, ferramentas de visualização da interface da aplicação que está sendo desenvolvida, ferramentas de teste, e outras que podem ser encontradas na documentação da ferramenta (APPLE, 2024c). Para desenvolvimento dos aplicativos é necessário conhecer Objective-C ou Swift, linguagens de programação. Atualmente, o Swift é amplamente mais utilizado.

Figura 3 – Exemplo da tela do Xcode



Fonte: Elaborada pelo Autor.

3.2.3 URLSession

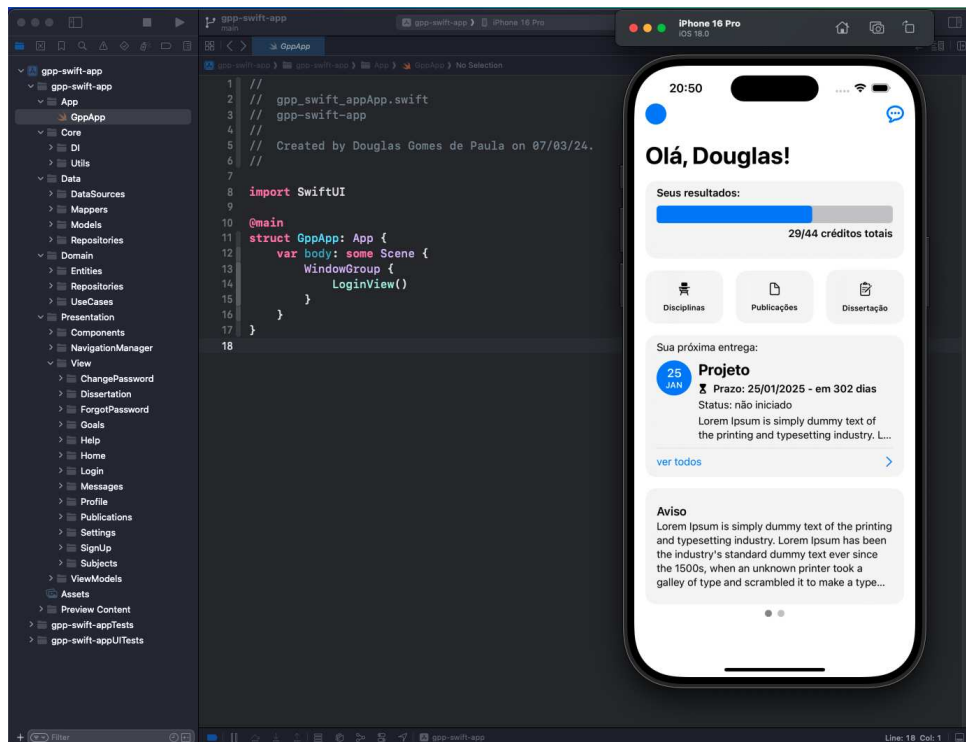
A *Interface de Programação de Aplicativo (API)* é o meio que possibilita a comunicação entre um cliente, uma aplicação que consumirá dados, e um servidor, uma aplicação que proverá dados. Neste trabalho será desenvolvida a aplicação cliente utilizando um *framework* de comunicação HTTP chamado [URLSession](#) para se conectar a aplicação servidor. O [URLSession](#) é uma biblioteca que facilita ao desenvolvedor iOS realizar o tráfego de dados entre um aplicativo e um servidor através de uma [API](#).

3.3 Arquitetura

A aplicação foi desenvolvida seguindo o conceito de *Clean Architecture*, proposto por Robert Cecil Martin ([MARTIN, 2017](#)), separando as preocupações, invertendo dependências internas e permitindo que o código seja desacoplado em relação a dependências externas. As vantagens de organizar o código dessa forma é obter uma aplicação robusta e escalável, facilitando a integração com um servidor externo (que será feita futuramente usando [URLSession](#)) e a inserção de testes automatizados (que não são temas deste trabalho).

A organização do código fonte está como na [Figura 4](#):

Figura 4 – Arquitetura do Aplicativo



Fonte: Elaborada pelo Autor.

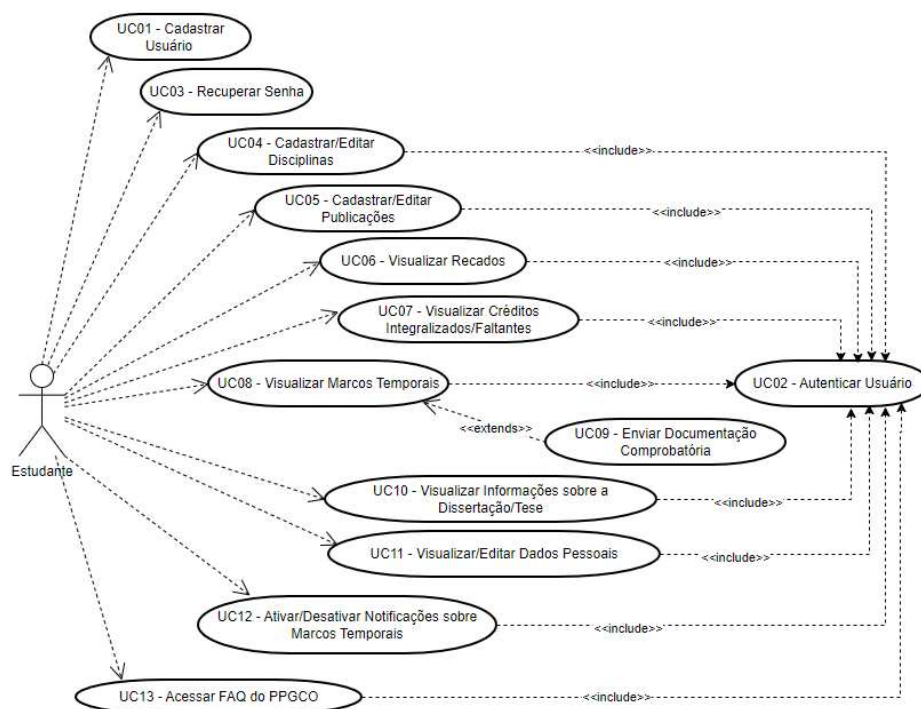
Cada pasta possui uma responsabilidade:

- **App:** O arquivo de entrada da aplicação se encontra nessa pasta.
- **Core:** Esta pasta concentra os arquivos necessários para gerenciar a injeção de dependência e arquivos com funções auxiliares.
- **Data:** Esta pasta concentra os modelos de dados, mapeadores, repositórios e fontes de dados.
- **Domain:** Esta pasta concentra entidades do domínio, repositórios que fazem interface para manipular as entidades e os casos de uso que contém a regra de negócio.
- **Presentation:** Esta pasta concentra a implementação das telas, os componentes, o controle de navegação e as *ViewModels*.

3.4 Casos de uso

Considerando os requisitos funcionais levantados, cada caso de uso descreve a interação entre o usuário e o sistema para realizar determinada funcionalidade. A Figura 5 ilustra o caso de uso geral.

Figura 5 – Caso de Uso Geral



Fonte: Elaborada pelo Autor.

3.4.1 UC01 - Cadastrar Usuário

Ator: Estudante

Descrição: O estudante deve ser capaz de se cadastrar no aplicativo fornecendo seu e-mail e número de matrícula.

Fluxo Principal:

1. O estudante acessa a tela de cadastro.
2. O sistema solicita o e-mail e a matrícula.
3. O estudante insere o e-mail e a matrícula e confirma.
4. O sistema envia um *token* de validação no e-mail inserido.
5. O estudante insere o *token* recebido no e-mail e confirma.
6. O sistema valida o *token* e redireciona para a tela de criação de senha.
7. O estudante insere a senha, insere a confirmação da senha e insere o link do *lattes* e confirma.

8. O sistema cria o usuário e redireciona para a tela de autenticação.

3.4.2 UC02 - Autenticar Usuário

Ator: Estudante

Descrição: O estudante deve ser capaz de fazer *login* no aplicativo usando e-mail e senha.

Fluxo Principal:

1. O estudante acessa a tela de login.
2. O sistema solicita o e-mail e senha.
3. O estudante insere os dados.
4. O sistema valida o e-mail e senha.
5. O sistema autentica o estudante e permite o acesso ao aplicativo.

3.4.3 UC03 - Recuperar Senha

Ator: Estudante

Descrição: O estudante deve ser capaz de recuperar a senha caso a tenha esquecido.

Fluxo Principal:

1. O estudante acessa a opção "Esqueci minha senha".
2. O sistema solicita o e-mail cadastrado.
3. O estudante insere o e-mail.
4. O sistema envia um *token* no e-mail.
5. O estudante insere o *token* recebido no e-mail e confirma.
6. O sistema valida o *token* e redireciona para a tela de criação de senha.
7. O estudante insere a senha, insere a confirmação da senha e confirma.
8. O sistema troca a senha e redireciona para a tela de autenticação.

3.4.4 UC04 - Cadastrar/Editar Disciplinas

Ator: Estudante

Descrição: O estudante deve ser capaz de cadastrar ou editar as disciplinas que está cursando.

Fluxo Principal:

1. O estudante acessa a seção de disciplinas.
2. O sistema apresenta uma lista de disciplinas.
3. O estudante escolhe se cadastrar em nova disciplina ou editar a nota de uma que já está cadastrado.
4. O estudante insere ou modifica as informações da disciplina.
5. O sistema salva as mudanças.

3.4.5 UC05 - Cadastrar/Editar Publicações

Ator: Estudante

Descrição: O estudante deve ser capaz de cadastrar ou editar suas publicações acadêmicas.

Fluxo Principal:

1. O estudante acessa a seção de publicações.
2. O sistema apresenta uma lista de publicações se existir.
3. O estudante escolhe cadastrar nova publicação ou editar uma existente.
4. O estudante insere ou modifica as informações da publicação.
5. O sistema salva as mudanças.

3.4.6 UC06 - Visualizar Recados

Ator: Estudante

Descrição: O estudante deve ser capaz de receber recados e notificações.

Fluxo Principal:

1. O sistema envia um recado ao estudante.
2. O estudante acessa a tela de recados.

3. O estudante visualiza o recado recebido.

3.4.7 UC07 - Visualizar Créditos Integralizados/Faltantes

Ator: Estudante

Descrição: O estudante deve ser capaz de ver os créditos integralizados e os que faltam para a conclusão.

Fluxo Principal:

1. O estudante acessa a seção de créditos.
2. O sistema exibe os créditos cursados e os que faltam.

3.4.8 UC08 - Visualizar de Marcos Temporais

Ator: Estudante

Descrição: O estudante deve ser capaz de visualizar os prazos e marcos importantes do curso.

Fluxo Principal:

1. O estudante acessa a seção de marcos temporais.
2. O sistema exibe os marcos e prazos associados ao curso do estudante.

3.4.9 UC09 - Enviar Documentação Comprobatória

Ator: Estudante

Descrição: O estudante deve ser capaz de enviar documentos comprobatórios quando necessário.

Fluxo Principal:

1. O estudante acessa a seção de envio de documentação.
2. O sistema solicita o *upload* do documento.
3. O estudante faz o *upload* do documento.
4. O sistema confirma o recebimento do documento.

3.4.10 UC10 - Visualizar Informações sobre a Dissertação/Defesa

Ator: Estudante

Descrição: O estudante deve ser capaz de visualizar as informações relacionadas à sua dissertação e defesa.

Fluxo Principal:

1. O estudante acessa a seção de dissertação/defesa.
2. O sistema exibe informações como datas e exigências.

3.4.11 UC11 - Visualizar/Editar Dados Pessoais

Ator: Estudante

Descrição: O estudante deve ser capaz de visualizar suas informações pessoais no aplicativo.

Fluxo Principal:

1. O estudante acessa a seção de perfil.
2. O sistema exibe opções para acessar informações pessoais.
3. O estudante acessa a opção perfil para visualizar/editar seu e-mail ou *lattes*.
4. O estudante acessa a opção de alterar senha para poder alterar sua senha.

3.4.12 UC12 - Ativar/Desativar Notificações sobre Marcos Temporais

Ator: Estudante

Descrição: O estudante deve ser capaz de ativar ou desativar as notificações de prazos importantes.

Fluxo Principal:

1. O estudante acessa a opção de configurações.
2. O estudante escolhe ativar ou desativar notificações.
3. O sistema ajusta as preferências de notificação.

3.4.13 UC13 - Acessar o FAQ do PPGCO

Ator: Estudante

Descrição: O estudante deve ser capaz de acessar as [FAQ](#) do [PPGCO](#).

Fluxo Principal:

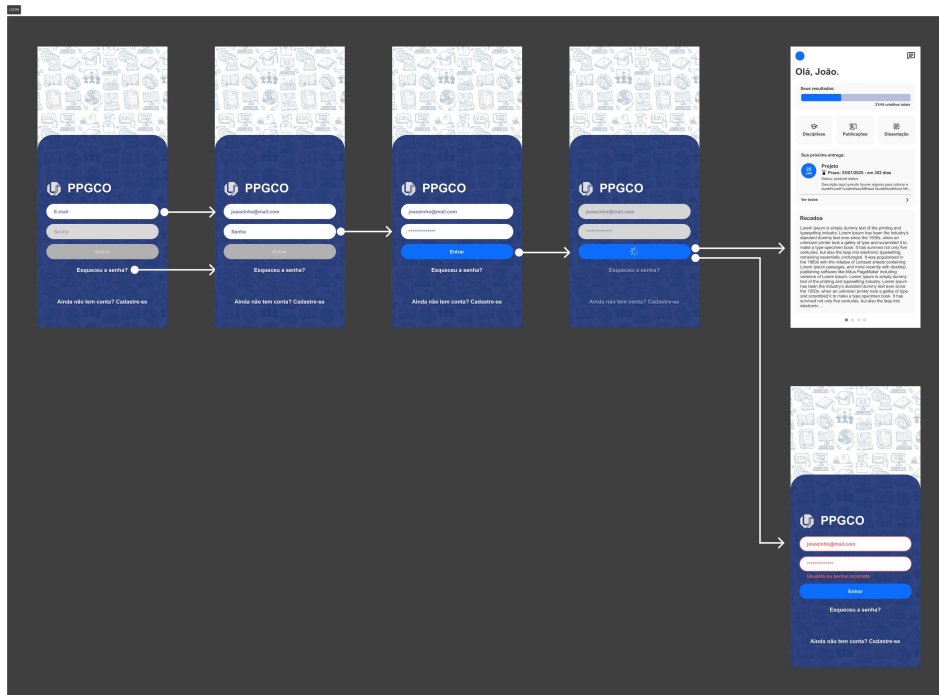
1. O estudante acessa a seção de Ajuda.
2. O sistema exibe as perguntas frequentes.

4 Resultados e Discussões

Este capítulo apresenta os fluxos que foram implementados no aplicativo iOS a fim de atender a proposta deste trabalho. Na seção 4.1 está o fluxo de *Login* com a autenticação do usuário e acesso para outros fluxos. Na seção 4.2 mostra o fluxo do Cadastro e a 4.3 o fluxo de Recuperação de Senha, ambas acessadas através da tela de Login. A seção 4.4 mostra Tela Inicial do aplicativo, onde é possível acessar todos os fluxos que dependem da autenticação para serem acessados pelo usuário. As seções 4.5, 4.6, 4.7 e 4.8 são os fluxos de Perfil, Alteração de Senha, Configurações e Ajuda, respectivamente, todas acessadas através do menu lateral acessado pelo canto superior esquerdo do aplicativo. O fluxo de Recados é mostrado na seção 4.9. Na seção 4.10 é mostrado o fluxo de Disciplinas, na seção 4.11 o fluxo de Publicações e na seção 4.12 a tela de Dissertação. O fluxo de Marcos Temporais é mostrado na seção 4.12. Para concluir o capítulo, na seção 4.14 está a validação das Heurísticas de Nielsen sobre a aplicação, na seção 4.15 os passos necessários para a implantação do aplicativo e na seção 4.16 trás uma discussão cerca da elaboração e implementação do aplicativo.

4.1 Login

A autenticação do usuário na tela de Login é feita através de e-mail e senha, como pode ser vista na Figura 6:

Figura 6 – Fluxo de *Login*

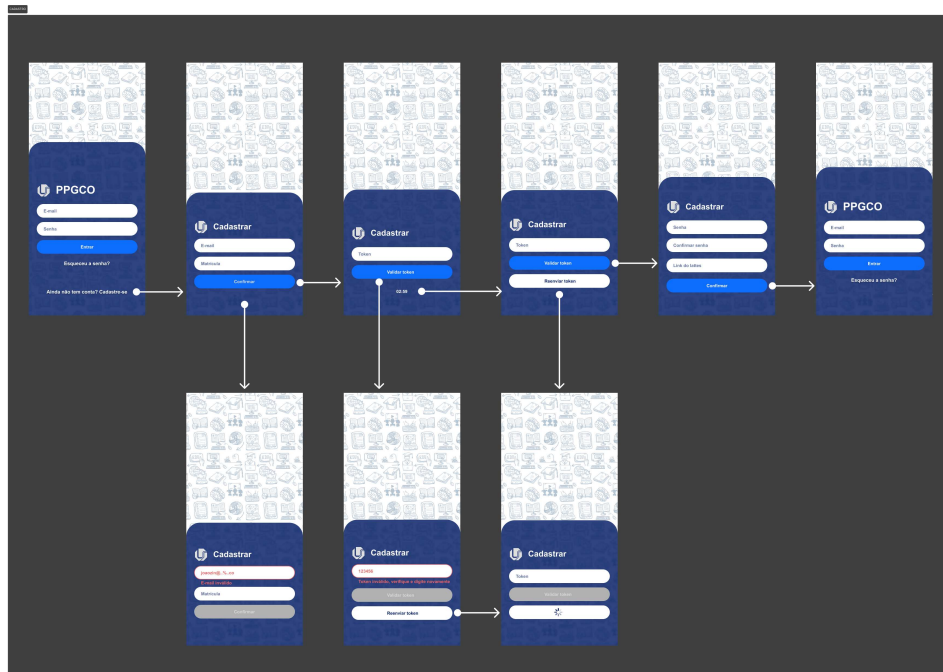
Fonte: Elaborada pelo Autor.

Após inserir e-mail e senha e clicar no botão “Entrar”, a aplicação fará uma requisição no servidor, enviando o e-mail e a senha do usuário, e receberá uma resposta onde apenas dois resultados são possíveis: e-mail e senha válidos e o usuário é redirecionado para a tela inicial do aplicativo, ou e-mail ou senha inválidos e uma mensagem de erro é mostrada para o usuário. A validação do e-mail e senha do usuário pelo servidor é escopo de outro trabalho.

4.2 Cadastro

Na tela de *login*, ao clicar em “Ainda não tem conta? Cadastre-se” o usuário é redirecionado para o fluxo de cadastro conforme imagem abaixo:

Figura 7 – Fluxo de Cadastro



Fonte: Elaborada pelo Autor.

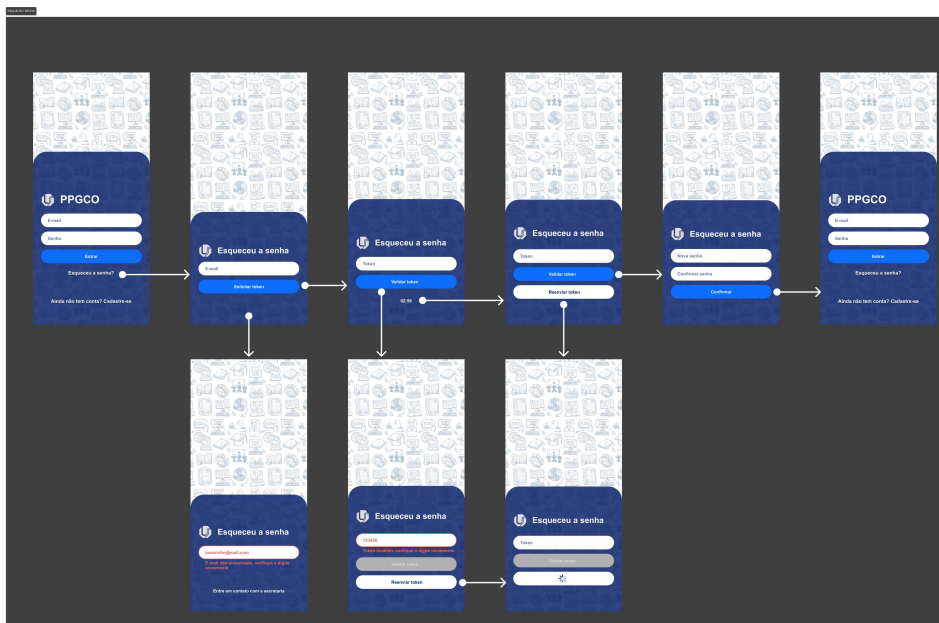
Os passos para se cadastrar são:

- Estar previamente cadastrado no sistema pela gestão do PPGCO;
- Preencher os campos e-mail e matrícula com os dados que foram cadastrados pela gestão;
- Ao receber o *token* no e-mail, preencher no campo de *token* e clicar em “Validar token”;
 - Caso o *token* não seja recebido no e-mail, o usuário precisa aguardar o tempo zerar para que o botão “Reenviar token” seja habilitado e ele reenvie o token;
 - Caso o *token* seja inválido uma mensagem de erro será exibida e o botão “Reenviar token” é habilitado mesmo que o tempo ainda não tenha zerado;
- Com o *token* validado, o usuário é redirecionado para a tela de criação de senha e inserção do link do *lattes*;
- Com o sucesso do cadastro o usuário volta para a tela de login onde deve usar o e-mail e a senha criada para se autenticar no aplicativo.

4.3 Recuperação de Senha

O fluxo é acessado pela tela de *Login* clicando em “Esqueci a senha?” como na figura abaixo:

Figura 8 – Fluxo de Recuperação de Senha



Fonte: Elaborada pelo Autor.

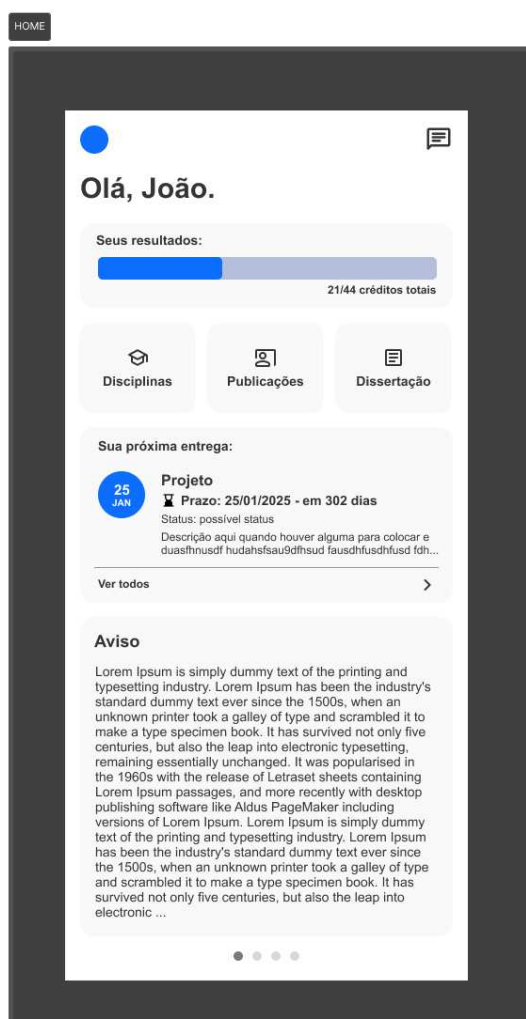
Os passos para recuperar a senha são:

- O usuário insere o e-mail de cadastro e clica em “Solicitar token”;
 - Se o e-mail não for encontrado uma mensagem de erro é exibida e um texto clicável indica o meio para entrar em contato com a secretaria.
- Ao receber o *token* no e-mail, preencher no campo *token* e clicar em “Validar token”;
 - Caso o *token* não seja recebido no e-mail, o usuário precisa aguardar o tempo zerar para que o botão “Reenviar token” seja habilitado e ele reenvie o token;
 - Caso o *token* seja inválido uma mensagem de erro será exibida e o botão “Reenviar token” é habilitado mesmo que o tempo ainda não tenha zerado;
- Com o *token* validado, o usuário é redirecionado para a tela de criação de da nova senha;
- Com o sucesso do cadastro da nova senha, o usuário é redirecionado de volta para a tela de login onde deve usar o e-mail e a nova senha criada para se autenticar no aplicativo.

4.4 Tela Inicial

Após se autenticar, o usuário acessa a tela inicial do aplicativo. A tela inicial é o ponto de entrada para todos os outros fluxos que necessitam autenticação do usuário e pode ser vista abaixo:

Figura 9 – Tela Inicial



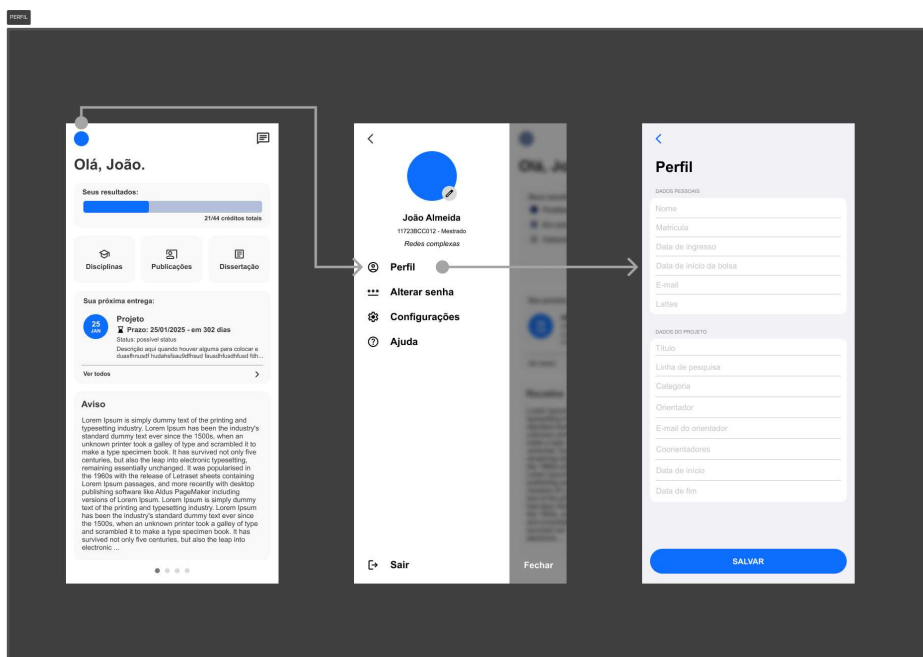
Fonte: Elaborada pelo Autor.

Nesta tela, o usuário acessa o menu lateral no lado esquerdo e a lista de todos os recados no lado direito. O usuário também vê a mensagem de saudação com o seu nome e uma caixa com os resultados dos seus créditos integralizados e os créditos que ainda faltam concluir. Ele tem acesso para a tela de disciplinas (seção 4.10), para a tela de publicações (seção 4.11) e para a tela de dissertação (seção 4.12). Há também uma caixa com o marco mais próximo a ser entregue e uma opção para ver todos os marcos temporais. No final da tela há um carrossel com os recados mais recentes e os que ainda não foram lidos.

4.5 Perfil

O fluxo do perfil é acessado pela tela inicial clicando no botão no canto superior esquerdo como mostra a figura abaixo:

Figura 10 – Fluxo de Perfil



Fonte: Elaborada pelo Autor.

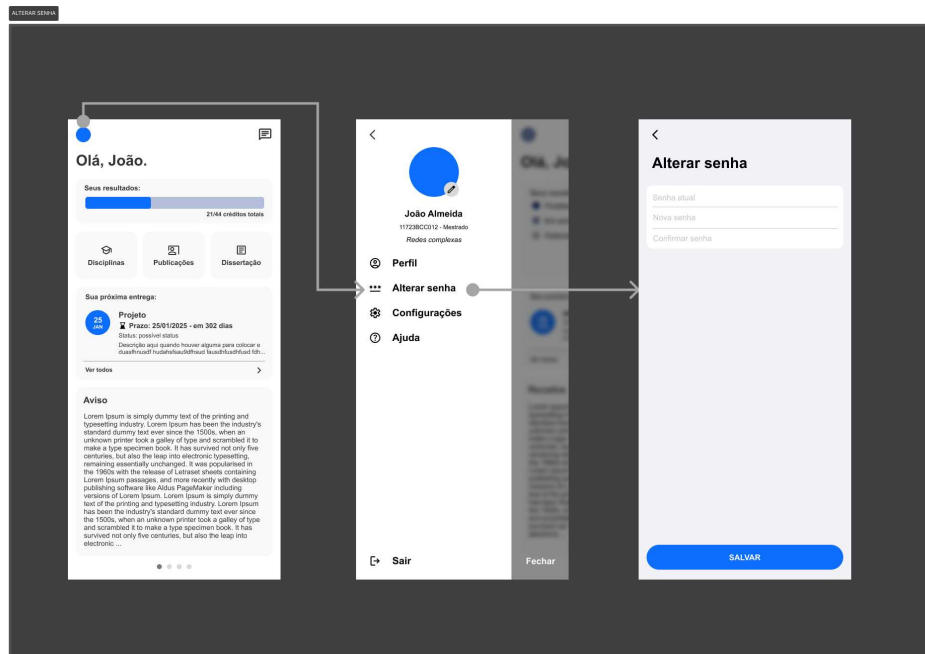
O menu lateral é aberto com várias informações e botões. Nele é mostrado a foto do usuário e opção para alterá-la. Os dados nome do usuário, matrícula, curso (Mestrado ou Doutorado) e linha de pesquisa também são mostrados. Além das informações há os botões para os fluxos: perfil, alteração de senha, configurações, ajuda e sair. Para fechar o menu lateral o usuário pode clicar na área fora do menu, na seta no canto superior esquerdo ou em “Fechar” no canto inferior direito.

Para o fluxo de perfil o usuário clica em “Perfil” e é redirecionado para a tela com os dados do usuário, onde consegue visualizar e editar os dados. Um ícone de cadeado será exibido ao lado do campo quando ele não puder ser editado.

4.6 Alterar Senha

O fluxo de alteração de senha é acessado pela tela inicial clicando no botão no canto superior esquerdo como mostra a figura abaixo:

Figura 11 – Fluxo de Alteração de Senha



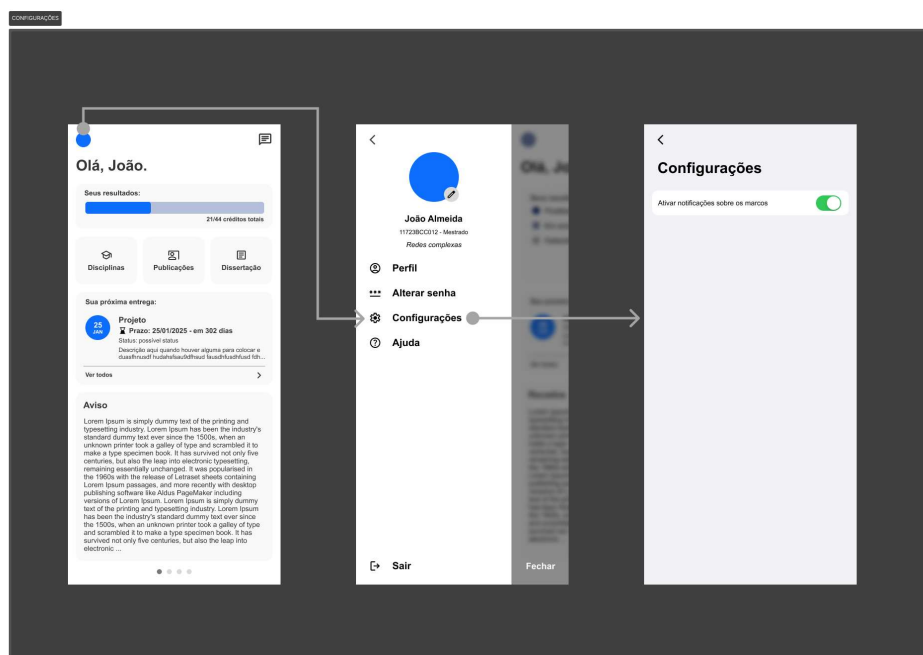
Fonte: Elaborada pelo Autor.

Para o fluxo de alteração de senha o usuário clica em “Alterar senha” e é redirecionado para a tela com os campos “Senha atual”, “Nova senha” e “Confirmar senha”. A senha atual é a senha que ele usa para se autenticar no aplicativo. A nova senha é a senha que ele deseja que seja a nova senha de autenticação e a senha de confirmação é a mesma senha digitada no campo anterior, cujo objetivo é validar se ele tem certeza da nova senha.

4.7 Configuração

A tela de configuração possui apenas um *switch* para o usuário ativar ou desativar as notificações em relação aos marcos temporais.

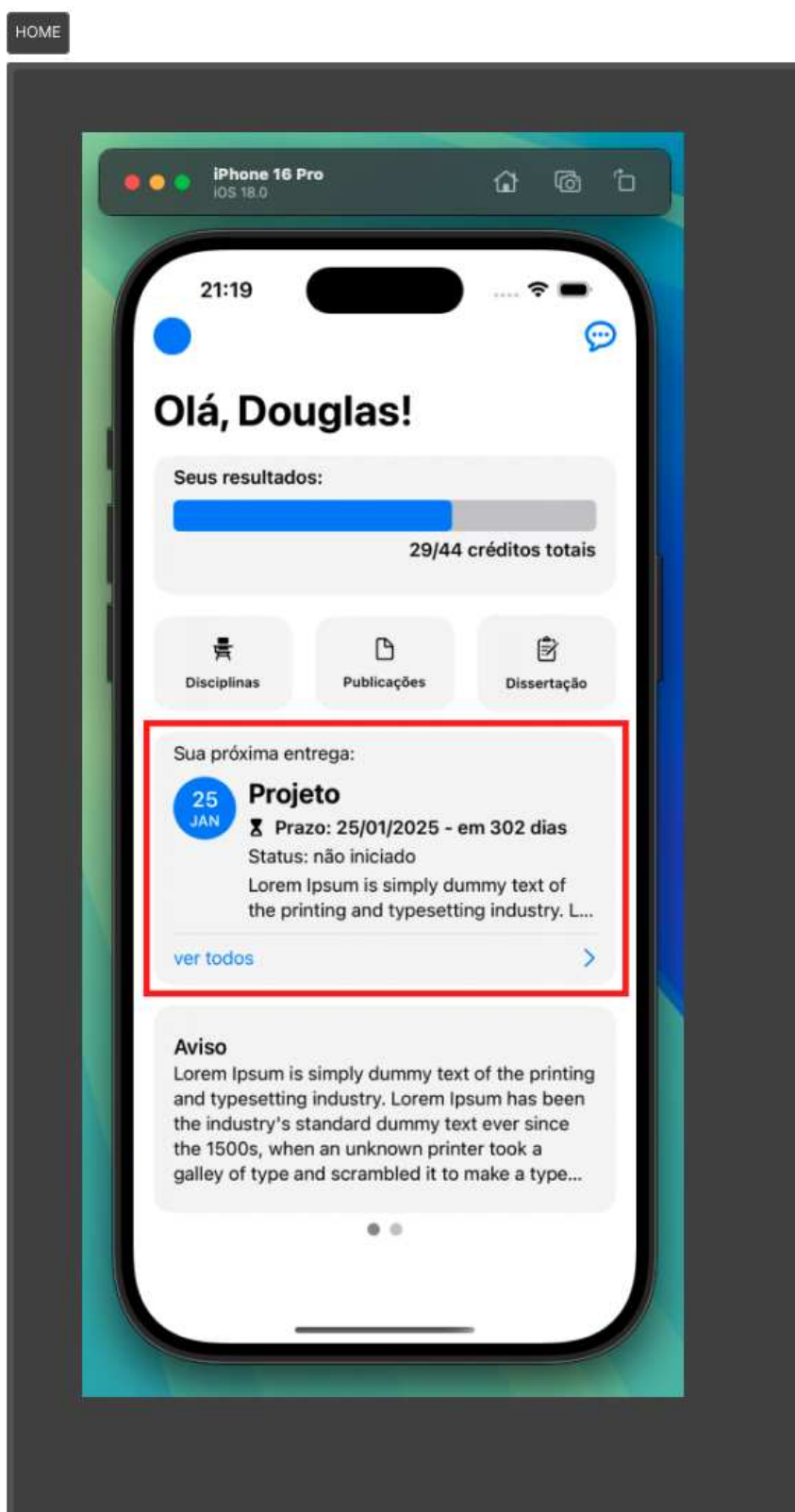
Figura 12 – Fluxo de Configuração



Fonte: Elaborada pelo Autor.

Ao deixá-lo ativo, o aplicativo avisará sempre a data de conclusão de um marco estiver próximo, ajudando com que o usuário se lembre das suas entregas. A entrega mais próxima é exibida na tela inicial na caixa com o título “Sua próxima entrega”, como na imagem abaixo:

Figura 13 – Próxima entrega na tela inicial

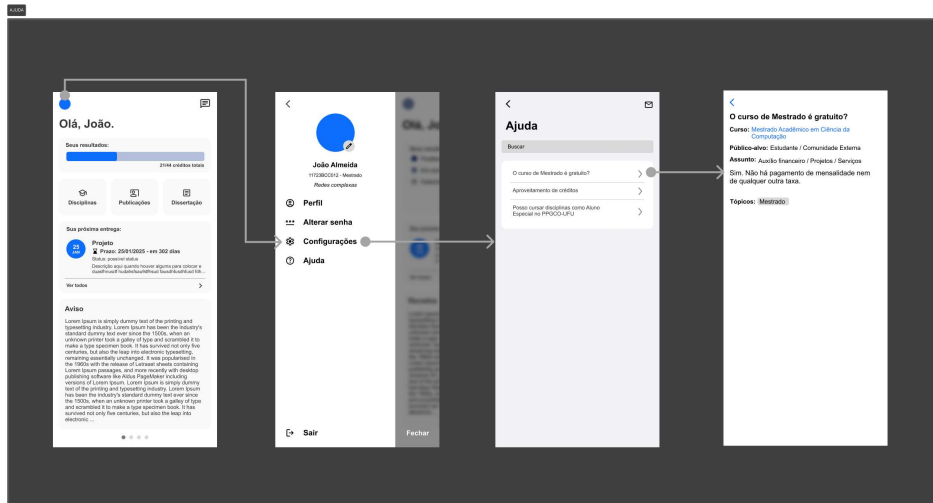


Fonte: Elaborada pelo Autor.

4.8 Ajuda

O fluxo de ajuda é acessado pela tela inicial clicando no botão no canto superior esquerdo como mostra a figura abaixo:

Figura 14 – Fluxo de Ajuda



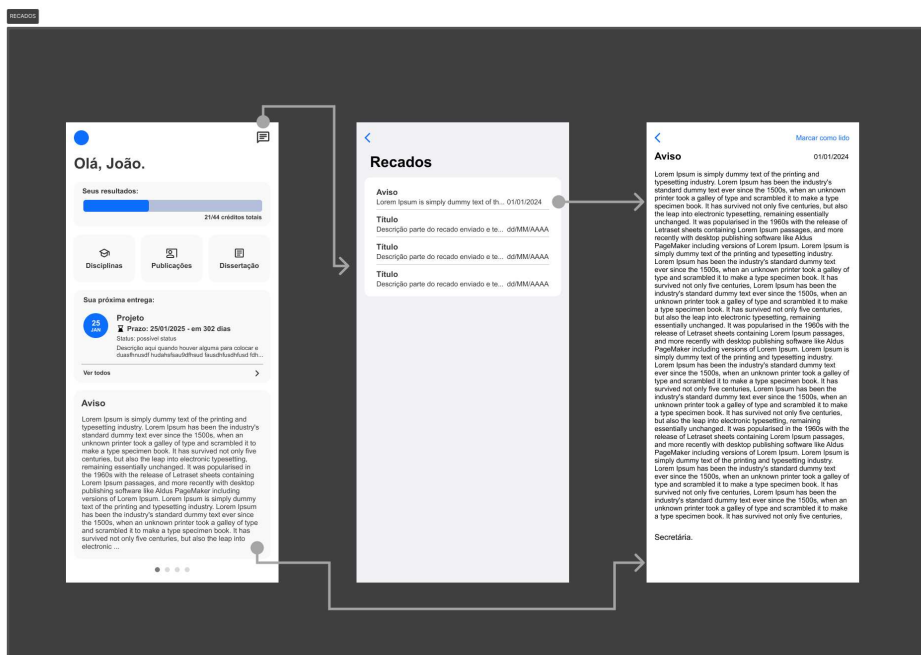
Fonte: Elaborada pelo Autor.

Uma lista com as perguntas mais frequentes é exibida. Ao selecionar uma das perguntas, o usuário é redirecionado para a tela que contém a resposta e links em palavras chaves.

4.9 Recados

Os recados já podem ser vistos desde a tela inicial no final da tela, ou acessar a lista de recados para ver todos os recados que já foram recebidos. Veja abaixo:

Figura 15 – Fluxo de Recados



Fonte: Elaborada pelo Autor.

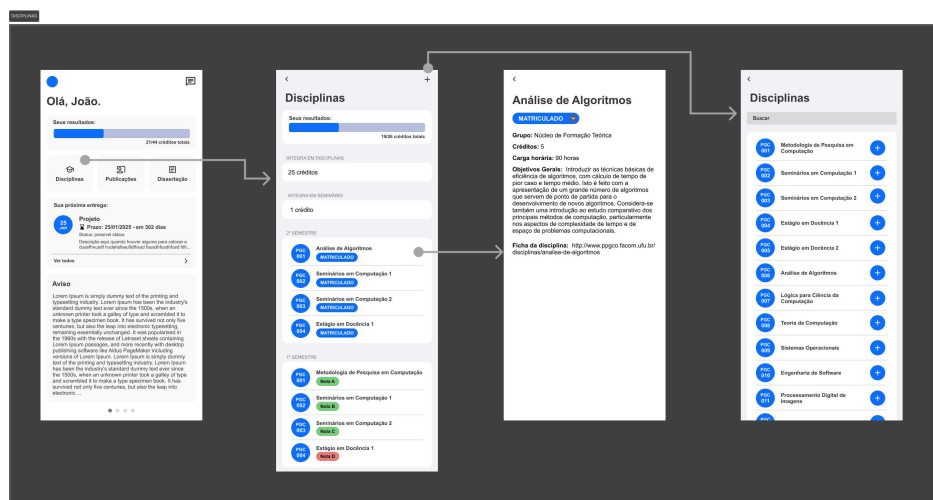
Ao clicar no ícone de recados no canto superior direito, o usuário é redirecionado para a lista com todos os recados (lidos e não lidos). Ao clicar em um recado, ele é aberto e o usuário consegue vê-lo por completo e pode marcá-lo como lido ou não lido.

No final da tela inicial, o carrossel exibe os recados mais recentes não lidos. Ao clicar em um recado do carrossel, o usuário é redirecionado para o recado completo, podendo marcá-lo como lido.

4.10 Disciplinas

O fluxo de disciplinas é acessado pela tela inicial como mostra a figura abaixo:

Figura 16 – Fluxo de Disciplinas



Fonte: Elaborada pelo Autor.

Ao clicar no botão “Disciplinas”, o usuário é redirecionado pra tela de Disciplinas onde consegue ver quantos créditos ele pode integralizar e quantos já integralizou com este módulo. A cada semestre ele se matricula em disciplinas e isso pode ser visualizado na segunda tela, onde contém no primeiro semestre disciplinas com notas e no segundo as disciplinas apenas matriculadas.

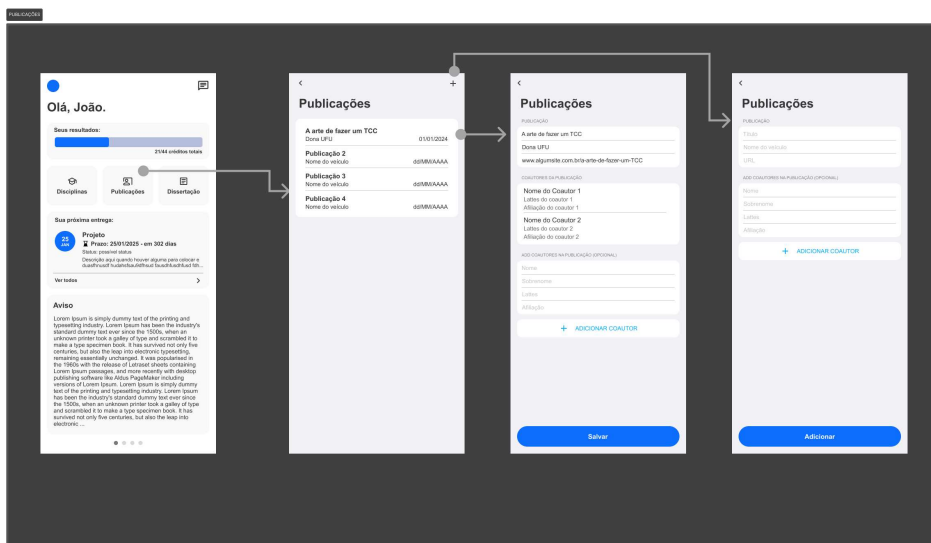
Ao clicar no botão “+” é aberta a tela de matricula de disciplinas. Nessa primeira versão a matrícula será apenas para controle do usuário no aplicativo, não significando que ele está matriculado na disciplina no sistema do PPGCO.

Na tela de disciplinas, ao clicar em uma disciplina na listagem dos semestres, isto é, disciplina com nota ou com status “matriculado”, o usuário é redirecionado para a tela de detalhe dessa disciplina, onde ele consegue ver mais informações da disciplina e/ou editar o status dela (com a nota, por exemplo). Isso pode ser visto na terceira tela do fluxo na figura 16.

4.11 Publicações

O fluxo de publicações é acessado pela tela inicial como mostra a figura abaixo:

Figura 17 – Fluxo de Publicações



Fonte: Elaborada pelo Autor.

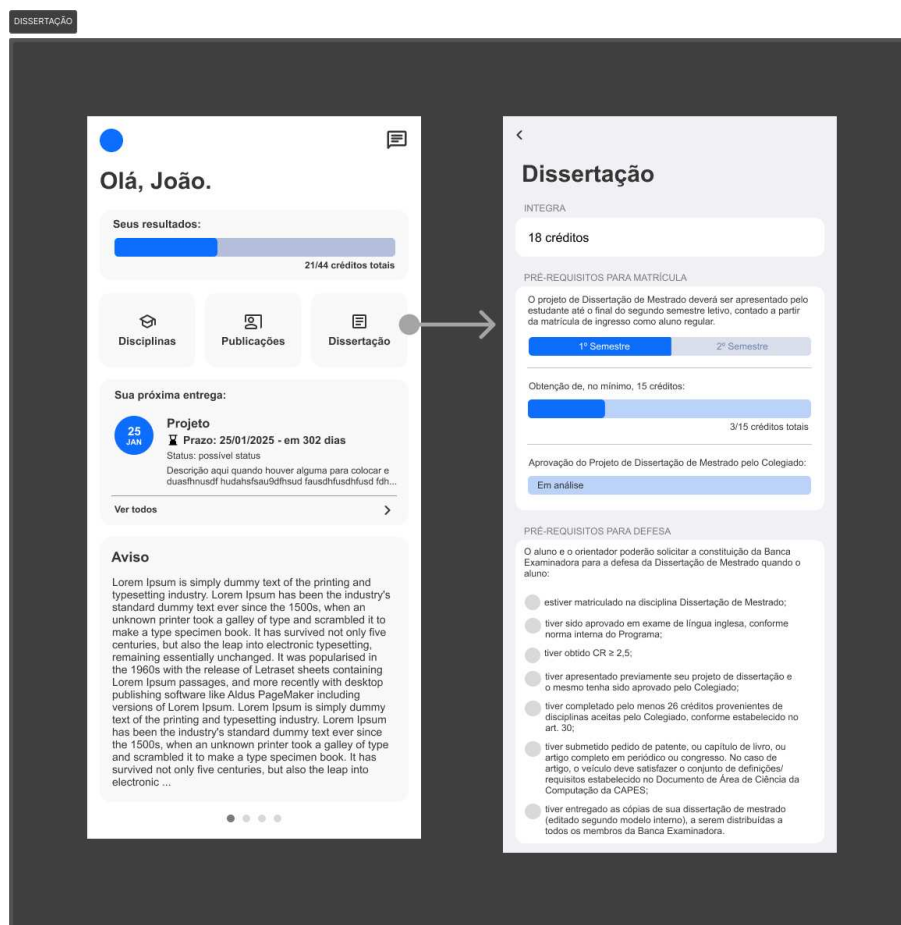
Ao clicar no botão “publicações”, o usuário é redirecionado para uma tela com a lista de publicações. Ao clicar em um item da lista ele é redirecionado para a tela com os detalhes dessa publicação, onde ele consegue alterar as informações.

Ao clicar no botão “+” é aberta a tela de adição da publicação, onde o usuário preenche as informações e ao salvar ela passará a aparecer na tela de listagem de publicações.

4.12 Dissertação/Tese

Ao clicar no botão “Dissertação” (para alunos de Doutorado aparecerá como “Tese”) na tela inicial, o usuário é redirecionado para a tela com os detalhes e critérios sobre a defesa da dissertação, como mostra a figura abaixo:

Figura 18 – Fluxo de Dissertação



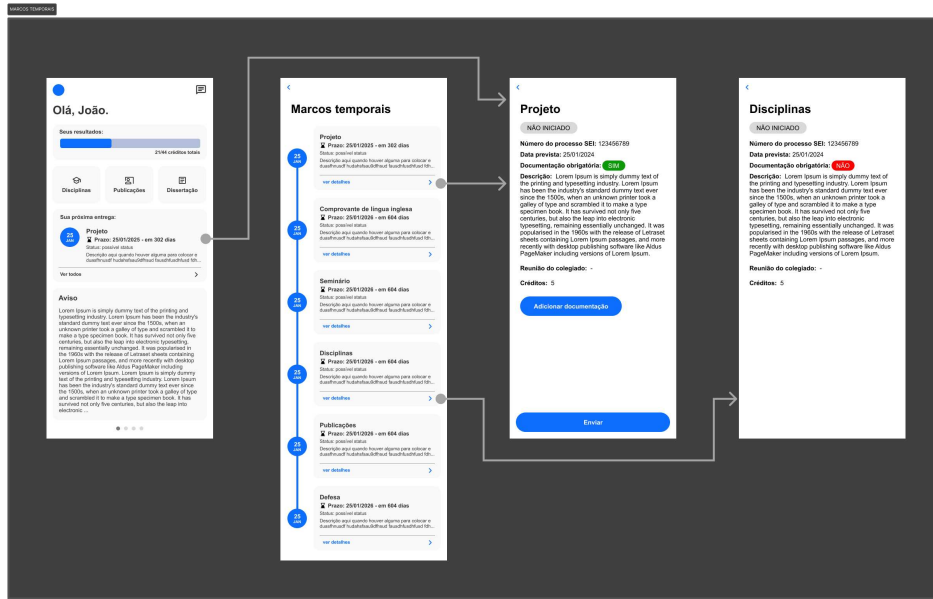
Fonte: Elaborada pelo Autor.

Como mostra a figura, há pré-requisitos para que o aluno se matricule na dissertação. O aplicativo auxilia o estudante a ter uma visão simplificada desses critérios. O aplicativo também mostra os pré-requisitos para a defesa da dissertação, assim o estudante sabe exatamente como funciona essa parte do curso.

4.13 Marcos Temporais

O estudante de Mestrado/Doutorado precisa cumprir com alguns requisitos para concluir o curso. Considerando esses requisitos e seus prazos, foi criada uma linha do tempo e cada uma dessas tarefas se tornou um marco temporal nessa linha. Eles são acessados através da tela inicial do aplicativo. Na tela inicial do aplicativo há uma caixa com o título “Sua próxima entrega”, nela o próximo marco é exibido ao usuário como mostra a primeira tela da figura abaixo:

Figura 19 – Fluxo de Marcos Temporais



Fonte: Elaborada pelo Autor.

Caso o usuário esteja com o *switch* de notificações sobre os marcos temporais ativados na tela de Configurações, conforme mostrado na seção 4.7, o usuário será notificado 7 dias antes da data do marco e depois novamente 2 horas antes do horário da entrega.

Na tela inicial é possível que o usuário acesse os detalhes do marco temporal mais próximo clicando sobre o mesmo e sendo redirecionado para a tela de detalhes, a terceira tela do fluxo na figura 19. Caso haja documentação a ser enviada, um botão de anexo da documentação é exibido e ele pode enviar a documentação para a gestão do PPGCO.

4.14 Validação das Heurísticas de Nielsen

Considerando a apresentação de cada fluxo do aplicativo nas seções anteriores, é possível ressaltar elementos que validam as Heurísticas de Nielsen (NIELSEN, 2024).

4.14.1 Visibilidade do status do sistema

Elementos que demonstram ao usuário o status do sistema foram incluídos, como o caso da animação de carregamento exibido nos botões quando é necessário que os usuários aguardem algum processamento, como na Figura 20.

Figura 20 – Carregamento do botão



Fonte: Elaborada pelo Autor.

Além disso, nas mensagens de erro que podem ser exibidas após uma requisição retornada do servidor, como na Figura 21:

Figura 21 – Mensagem de erro



Fonte: Elaborada pelo Autor.

No progresso da jornada do estudante na tela principal, como na Figura 22:

Figura 22 – Progresso de créditos



Fonte: Elaborada pelo Autor.

4.14.2 Compatibilidade entre o sistema e o mundo real

Usando padrões reconhecíveis e conceitos que já são conhecidos pelo usuário, como Disciplinas, Publicações e Dissertação na Figura 23:

Figura 23 – Botões na Tela Principal



Fonte: Elaborada pelo Autor.

Além dos gestos e navegação que são funcionalidades padronizadas do iOS que o usuário já é familiarizado.

4.14.3 Controle e liberdade para o usuário

O usuário consegue navegar para novas telas clicando sobre os itens que possuem o ícone “>” ao lado direito (comportamento padrão de navegação em aplicações iOS), como na Figura 24:

Figura 24 – Disciplinas que possuem navegação



Fonte: Elaborada pelo Autor.

E voltar clicando em “<”, como na Figura 25:

Figura 25 – Toolbar da tela de Disciplinas



Fonte: Elaborada pelo Autor.

4.14.4 Consistência e padronização

Cada ícone e texto utilizado dentro da aplicação possui o mesmo significado em qualquer tela do sistema, assim como a forma de navegar para uma nova tela e voltar para a anterior e o comportamento dos componentes também são consistentes.

4.14.5 Prevenção de erros

Campos inválidos desabilitam o botão e exibem uma mensagem no campo, a fim de evitar que o usuário cometa um erro enviando um dado que não é esperado pelo servidor, como na Figura 26:

Figura 26 – Campo e-mail inválido



Fonte: Elaborada pelo Autor.

4.14.6 Reconhecimento em vez de memorização

Cada botão do sistema possui um rótulo consistente com seu significado e isso permite que o usuário consiga reconhecer cada função a partir deste rótulo, sem a necessidade de memorizar qual a função do botão e para onde ele leva, como na Figura 27:

Figura 27 – Botões do menu lateral

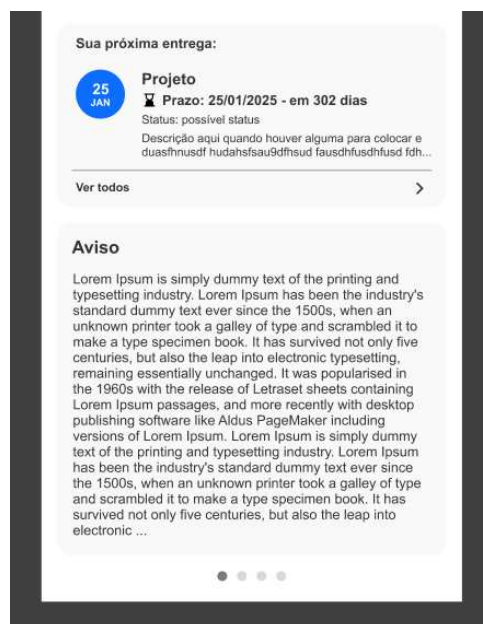


Fonte: Elaborada pelo Autor.

4.14.7 Eficiência na usabilidade

Existem atalhos para minimizar etapas e aumentar a eficiência do estudante ao usar o aplicativo, como a caixa de próximas entregas e quadro de recados na tela principal, como na Figura 28:

Figura 28 – Atalhos na tela principal



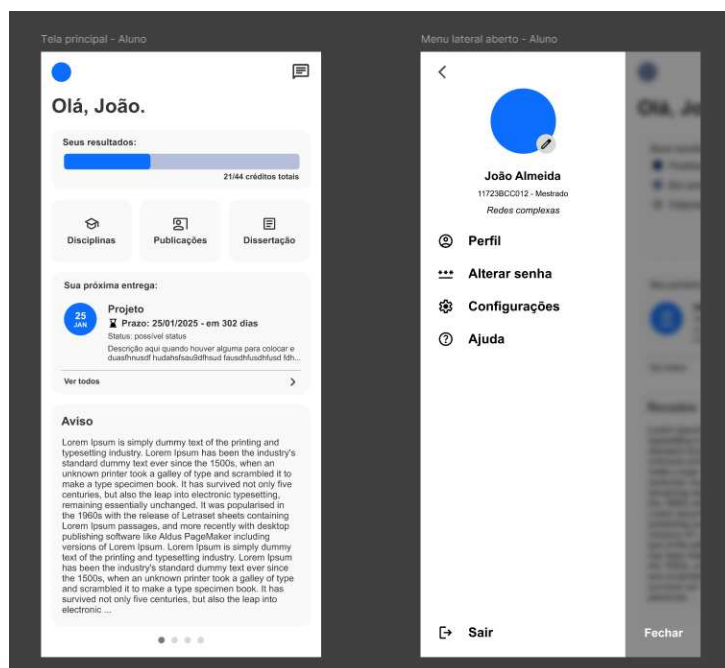
Fonte: Elaborada pelo Autor.

4.14.8 Estética e design minimalista

Foram usados elementos simples e cores sóbrias, a fim de manter a visibilidade limpa e o foco no que interessa ao usuário. Apenas as informações e fluxos mais impor-

tantes ficaram na tela principal e os atalhos para outros fluxos foram adicionados ao menu lateral, como na Figura 29:

Figura 29 – Tela principal e Menu lateral



Fonte: Elaborada pelo Autor.

4.14.9 Ajude os usuários a reconhecerem, diagnosticarem e recuperarem-se de erros

Mensagens são mostradas quando os campos são inválidos e se necessário é exibido um texto clicável orientando contato com os gestores do programa. A validação do campo que exibe a mensagem de erro pode ser feita tanto localmente quanto em uma resposta de requisição no servidor. Em casos de erros que impeçam o usuário de prosseguir, o botão também é desabilitado, como na segunda tela da Figura 30:

Figura 30 – Campo com erro e botão desabilitado

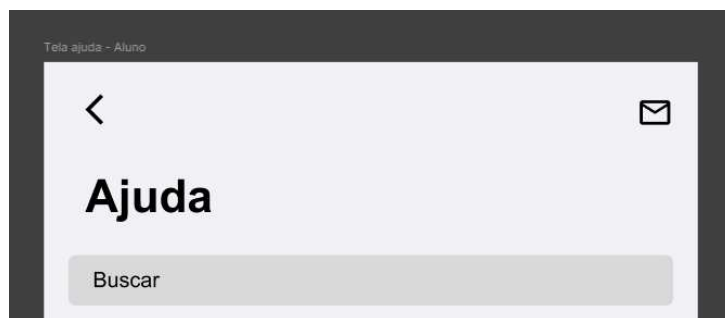


Fonte: Elaborada pelo Autor.

4.14.10 Ajuda e documentação

É possível encontrar ajuda através do menu lateral, como na Figura 14 e, caso as perguntas e respostas listadas não sejam suficiente, é possível entrar em contato com os gestores do programa clicando no ícone de e-mail como na Figura 31:

Figura 31 – Atalho para enviar e-mail na tela de Ajuda



Fonte: Elaborada pelo Autor.

4.15 Implantação

O aplicativo ainda precisa seguir um processo de implantação para que esteja pronto para uso em um ambiente de produção. Ressalta-se que essa etapa ficou fora do escopo deste trabalho, visto que há a dependência de outro trabalho de conclusão de curso em que a parte administrativa é construída. Para conseguir implantar o aplicativo, é necessário seguir as seguintes etapas:

- Discutir e refinar as *features* do aplicativo a fim de ter uma maior assertividade na ferramenta para com os estudantes do [PPGCO](#);
- Integrar o aplicativo com o servidor web implementado em outro trabalho de conclusão de curso da [Universidade Federal de Uberlândia \(UFU\)](#);
- Criar uma conta Apple e inscrevê-la no [Apple Developer Program](#);
- Configurar o aplicativo no [Apple Store Connect](#), pois ela é a plataforma de gerenciamento de aplicativos da Apple;
- Extrair capturas de tela do aplicativo conforme diretrizes da Apple para a loja;
- Configurar o Xcode com os [certificados](#) necessários para distribuição do aplicativo;
- Revisar o [App Review Guidelines](#) para garantir que todas das *features* do aplicativo estão aderentes as diretrizes da Apple;
- Enviar o aplicativo para revisão de publicação da Apple.

4.16 Discussão

Este trabalho apresentou a implementação de um aplicativo iOS para apoiar a jornada acadêmica dos estudantes do PPGCO da UFU. Os resultados apresentam indícios que o aplicativo cumpre o objetivo de melhorar o gerenciamento das atividades e prazos acadêmicos através de um acesso intuitivo e organizado às informações para uma melhor progresso no curso. No entanto, ressalta-se que não houve validação com usuários reais.

Uma aplicação como essa destaca a importância do uso da tecnologia no ambiente acadêmico, onde os estudantes enfrentam demandas crescentes de organização e adaptação aos regulamentos dos seus cursos. O uso de uma aplicação *mobile*, projetado com heurísticas de usabilidade e uma arquitetura robusta, contribui para a criação de um sistema eficiente que pode melhorar o desempenho acadêmico ao reduzir o tempo gasto na gestão de informações, alinhando-se a estudos anteriores que sugerem que ferramentas organizacionais melhoram o rendimento dos alunos (SOARES et al., 2023).

Ao se comparar com os trabalhos correlatos, como o aplicativo Controle Universitário (ANNA, 2021) e o MAAS (SILVA; UEDA, 2013), este trabalho tem uma abordagem mais específica, buscando atender diretamente às necessidades dos alunos do PPGCO com funcionalidades alinhadas ao seu regulamento interno (FACOM, 2021). Enquanto o Controle Universitário fornece funcionalidades que abrange alunos de graduação, e o MAAS facilita o gerenciamento de compromissos acadêmicos por meio de uma plataforma web, o aplicativo proposto se diferencia pela sua adaptação ao regulamento interno do PPGCO, além de implementar as Heurísticas de Nielsen (NIELSEN, 2024) para garantir uma melhor experiência de uso.

Algumas limitações precisam ser consideradas. Primeiramente, o aplicativo não está integrado a um servidor externo, pois este servidor ainda está em desenvolvimento em outro trabalho de conclusão de curso e só terá a sua integração completa após a finalização do mesmo. Essa integração permitirá a autenticação dos usuários e sincronização dos dados com o sistema web que está sendo desenvolvido. Além disso, o aplicativo foi desenvolvido exclusivamente para a plataforma iOS, restringindo o alcance apenas aos usuários que utilizam dispositivos da Apple. Essas limitações indicam que ainda há esforços a serem feitos para que o aplicativo chegue até os estudantes, além da necessidade de implementação de uma aplicação Android para alcançar a abrangência total do público alvo.

Observações inesperadas surgiram no meio do processo de desenvolvimento, especialmente em relação as funcionalidades que exigiram um alto grau de personalização para atender os requisitos do regulamento interno do PPGCO. Para assegurar que as funcionalidades fossem intuitivas e facilmente acessíveis, foi necessário estudar e adaptar vários pontos do regulamento interno. Como exemplo, os marcos temporais, que são uma

abstração dos requisitos do regulamento interno que são necessários para a conclusão da jornada de Mestrado e Doutorado. Outra dificuldade foi garantir que, além dos casos de uso, as Heurísticas de Nielsen (NIELSEN, 2024) fossem atendidas completamente.

Pode-se observar que todos os casos de uso foram satisfeitos através das figuras mostradas nos resultados. O caso de uso de cadastro é apresentado na Figura 7, o caso de uso de autenticação é apresentado na Figura 6, o caso de uso de recuperação de senha é apresentado na Figura 8, o caso de uso de disciplinas é apresentado na Figura 16, o caso de uso de publicações é apresentado na Figura 17, o caso de uso de recados é apresentado na Figura 15, o caso de uso de visualização de créditos é apresentado na Figura 9, o caso de uso de marcos temporais e envio de documentação comprobatória é apresentado na Figura 19, o caso de uso de dissertação/tese é apresentado na Figura 18, o caso de uso de dados pessoais é apresentado na Figura 10, o caso de uso de notificações é apresentado na Figura 12 e, por fim, o caso de uso do FAQ é apresentado na Figura 14.

5 Conclusão

Neste trabalho, foi desenvolvido um aplicativo iOS para auxiliar o estudante em sua jornada no PPGCO. Com ele, será possível o acesso e gestão das etapas da jornada acadêmica dos estudantes direto de seus *smartphones*, facilitando o acesso aos seus dados, como notas de disciplinas, recados da gestão do programa, prazos dos marcos temporais, entre outras funcionalidades.

O desenvolvimento se deu através do ambiente de desenvolvimento Xcode utilizando a linguagem Swift. Para realizar a integração com o servidor, quando este estiver pronto, será utilizado o URLSession para fazer a comunicação via API. A implementação da aplicação se manteve na utilização de recursos nativos da plataforma da Apple, visando usar recursos atualizados e de uso comum no mercado de desenvolvimento *mobile*.

Para o desenvolvimento da aplicação, foi adotado um design minimalista e funcional, seguindo as Heurísticas de Nielsen (NIELSEN, 2024), o que garantiu uma experiência de usuário clara e eficiente. A aplicação cumpre o seu papel ao proporcionar uma visualização clara do progresso do estudante, como pode ser visto na Figura 9, e facilita a gestão de tempo com seus outros recursos mostrados em resultados.

Ao final do trabalho, espera-se não apenas fornecer uma aplicação para melhorar o acompanhamento do estudante na jornada do PPGCO, mas também contribuir para o avanço do conhecimento na área de desenvolvimento de aplicativos aplicados à educação, inspirando futuras pesquisas e desenvolvimentos de novas aplicações na área. Portanto, o objetivo deste trabalho não foi apenas desenvolver um aplicativo para os estudantes, mas também melhorar continuamente o processo educacional e aumentar a eficiência e a qualidade do ensino superior facilitando o acesso à informação.

5.1 Trabalhos futuros

Esta seção mostra possíveis trabalhos futuros, sendo a seção 5.1.1 apresenta a integração com o servidor, a seção 5.1.2 a implementação de uma versão do aplicativo para Android e a seção 5.1.3 possíveis melhorias no aplicativo.

5.1.1 Integração com o servidor

Esta aplicação foi desenvolvida considerando a implementação de um servidor, ou seja, ele necessita de um servidor para funcionar através de uma API. A implementação deste servidor é tema de outro trabalho de conclusão de curso, e assim que finalizado abre a possibilidade da integração desta aplicação com o mesmo.

Sem um servidor, as funcionalidades da aplicação não poderão ser usadas, como cadastro e autenticação do aluno, visualização de créditos integrados, recados da gestão do programa, visualização de disciplinas feitas e disponíveis para fazer, assim como suas notas, entre outras funcionalidades. Considerando isto, há a pendência de integrar esta aplicação com o servidor resultado de outro trabalho antes de disponibilizar uma versão para que estudantes testem a experiência do aplicativo.

5.1.2 Implementação de uma aplicação Android

Considerando que este trabalho desenvolveu um aplicativo iOS, um trabalho a ser feito é desenvolver a mesma aplicação seguindo o protótipo feito no Figma, mas para a plataforma Android. Com essa implementação, a aplicação passa a ter potencial de alcançar todos os estudantes da universidade.

5.1.3 Melhorias no aplicativo

- Implementação de tela para que o estudante consiga inserir uma disciplina que não esteja pré-programada pela gestão;
- Integração dos marcos temporais no calendário do iPhone.
- Integração com o sistema oficial da [UFU](#) para que os dados, como notas de disciplinas, sejam importados automaticamente;
- Integração com bibliotecas de coleta de dados que capturem *bugs* a fim de poder melhorar a aplicação continuamente;
- Como o aplicativo é voltado para os estudantes, há a possibilidade de criar o usuário administrativo, para que gestores do programa também consigam gerir os estudantes através do aplicativo;
- Integração com a [API](#) da Plataforma Lattes para recuperar as publicações.

Referências

ANNA, V. G. S. **Controle Universitário: Um Aplicativo de controle dos estudos e do andamento do graduando ao longo do curso superior**. 2021. Disponível em: <<https://www.monografias.ufop.br/handle/35400000/3464>>. Citado 2 vezes nas páginas 18 e 52.

APPLE. **Swift**. 2024. Swift.org. Disponível em: <<https://www.swift.org/documentation/>>. Acesso em: 02 fev. 2024. Citado na página 16.

_____. **SwiftUI**. 2024. Apple Developer Documentation. Disponível em: <<https://developer.apple.com/documentation/swiftui/>>. Acesso em: 02 fev. 2024. Citado na página 17.

_____. **Xcode**. 2024. Apple Developer Documentation. Disponível em: <<https://developer.apple.com/documentation/xcode>>. Acesso em: 02 fev. 2024. Citado na página 22.

AUGUSTO, R.; AMBIEL, M.; RAISSA, A.; COSTA, L.; DEYVIS, A.; ARAÚJO, S.; CAMILA, J.; CAMILO, C.; ROMANIN, S.; RESUMO, Z. **Motivos de evasão na pós-graduação no Brasil: um instrumento de medida**. 2019. Disponível em: <<https://doi.org/10.5380/psi.v24i1.62231>>. Citado na página 11.

BARBOSA, C. **A tríade do tempo**. Buzz Editora, 2018. ISBN 9788593156618. Disponível em: <<https://books.google.com.br/books?id=0lBdDwAAQBAJ>>. Citado na página 11.

CONEXIS. **Celular é o principal meio de acesso à Internet dos estudantes no Brasil**. 2018. Acessado dia 10/09/2024 às 18:37. Disponível em: <<https://conexis.org.br/celular-e-o-principal-meio-de-acesso-a-internet-dos-estudantes-no-brasil/>>. Citado na página 11.

COODESH. **O que é arquitetura MVVM?** 2024. Coodesh. Disponível em: <<https://coodesh.com/blog/dicionario/o-que-e-arquitetura-mvvm/>>. Acesso em: 12 nov. 2024. Citado na página 17.

COSME, G. **Documentação de Arquiteturas de Software em SwiftUI**. 2024. Medium. Disponível em: <<https://medium.com/@gugacosme/documenta%C3%A7%C3%A3o-de-arquiteturas-de-software-em-swiftui-f556674b651f#:~:text=A%20arquitetura%20MVVM%20%C3%A9%20uma,interfaces%20de%20usu%C3%A1rio%20em%20SwiftUI.>>> Acesso em: 12 nov. 2024. Citado na página 17.

FACOM, P. P. **Regulamento do Programa de Pós-graduação em Ciência da Computação**. 2021. Acessado dia 10/09/2024 às 19:08. Disponível em: <<https://ppgco.facom.ufu.br/legislacoes/regulamento-interno-do-ppgco>>. Citado 5 vezes nas páginas 11, 13, 14, 20 e 52.

HAAS, G. **O que é iOS?** 2024. Canal Tech. Disponível em: <<https://canaltech.com.br/ios/o-que-e-ios/>>. Acesso em: 02 fev. 2024. Citado na página 16.

- MACCANN, C.; FOGARTY, G. J.; ROBERTS, R. D. Strategies for success in education: Time management is more important for part-time than full-time community college students. **Learning and Individual Differences**, v. 22, n. 5, p. 618–623, 2012. ISSN 1041-6080. Adult Learning. Disponível em: <<https://doi.org/10.1016/j.lindif.2011.09.015>>. Citado na página 11.
- MARTIN, R. C. **Clean architecture**. [S.l.]: Prentice Hall, 2017. Citado na página 23.
- MIRANDA, G. J.; SILVA, S. M. C. da; SILVA, L. B. e; PEGORARO, R. F.; PEREIRA, J. M. Dificuldades, preocupações e estresse na pós-graduação. **Revista Gestão Universitária na América Latina - GUAL**, Universidade Federal de Santa Catarina (UFSC), v. 15, p. 24–43, 8 2022. Disponível em: <<https://doi.org/10.5007/1983-4535.2022.e83913>>. Citado na página 11.
- NIELSEN, J. **10 Usability Heuristics for User Interface Design**. 2024. NNGroup. Disponível em: <<https://www.nngroup.com/articles/ten-usability-heuristics/>>. Acesso em: 02 fev. 2024. Citado 8 vezes nas páginas 12, 14, 15, 16, 45, 52, 53 e 54.
- SILVA, V. M.; UEDA, L. **MAAS: sistema para integração de compromissos acadêmicos empregando tecnologia móvel**. 2013. Disponível em: <<http://repositorio.utfpr.edu.br/jspui/handle/1/9276>>. Citado 2 vezes nas páginas 18 e 52.
- SOARES, A. B.; ALVES, P. R. S. da S.; JARDIM, M. E. de M.; MEDEIROS, C. A. C. de; RIBEIRO, R. Gestión del tempo en la rutina universitaria: resultados de una intervención. **Ciencias Psicológicas**, Universidad Católica del Uruguay, v. 17, 2023. ISSN 16884221. Disponível em: <<https://doi.org/10.22235/cp.v17i2.2845>>. Citado 2 vezes nas páginas 11 e 52.