



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS FLORIANÓPOLIS
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Maria Eduarda de Melo Hang

**Desenvolvimento de uma Progressive Web Application para o módulo de
matrícula da UFSC**

Florianópolis, Santa Catarina – Brasil

2022

Maria Eduarda de Melo Hang

**Desenvolvimento de uma Progressive Web Application para o módulo de
matrícula da UFSC**

Trabalho de Conclusão de Curso do Curso de Graduação em Ciência da Computação do Campus Florianópolis da Universidade Federal de Santa Catarina para a obtenção do título de Bacharel em Ciências da Computação.

Orientador: Prof. Ricardo Pereira e Silva, Dr.

Florianópolis, Santa Catarina – Brasil

2022

Maria Eduarda de Melo Hang

**Desenvolvimento de uma Progressive Web Application para o módulo de
matrícula da UFSC**

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de “Bacharel em Ciências da Computação” e aprovado em sua forma final pelo Curso de Graduação em Ciência da Computação.

Florianópolis, Santa Catarina – Brasil, .

Prof. Jean Everson Martina, Dr.
Coordenador do Curso

Banca Examinadora:

Prof. Ricardo Pereira e Silva, Dr.
Orientador

Prof. Raul Sidnei Wazlawick, Dr(a).
Avaliador
Universidade Federal de Santa Catarina –
UFSC

Profa. Christiane Anneliese Gresse Von
Wangenheim, Dra.
Avaliador(a)
Universidade Federal de Santa Catarina –
UFSC

AGRADECIMENTOS

Agradeço aos meus pais e amigos por terem me apoiado e ajudado nessa jornada difícil, sempre serei grata. Agradeço também ao meu time, Eldorado, e ao Laboratório Bridge por terem me incentivado e apoiado nesse projeto e ter dado uma infraestrutura incrível para trabalhar nessas aplicações.

RESUMO

Com o crescimento do uso de dispositivos móveis e acesso à internet no mundo, as instituições e empresas tentam atingir esse novo público através do desenvolvimento de aplicativos nativos ou *web* com a abordagem [Responsive Web Development \(RWD\)](#). As aplicações nativas possuem a vantagem de explorar ao máximo os recursos do dispositivo e trazer uma melhor experiência para o usuário, todavia, o custo de desenvolvimento e manutenção são altos, uma vez que é necessário desenvolver um sistema para cada plataforma e dar suporte para cada um deles. Para evitar esse problema, a [RWD](#) permite o desenvolvimento de um sistema independente da plataforma com o custo de perda na qualidade da experiência do usuário, uma vez que não pode aproveitar todos os recursos do dispositivo. Nesse cenário, a [Progressive Web Application \(PWA\)](#) surgiu para solucionar os problemas de ambas as abordagens, se aproveitando da independência de plataforma da [RWD](#) e dos recursos extras providos por aplicações nativas como as notificações *push* e possibilidade de instalação.

Diante desse contexto, o objetivo desse projeto foi desenvolver uma interface alternativa com [PWA](#) para o novo módulo de matrícula da [Universidade Federal de Santa Catarina \(UFSC\)](#), que está sendo desenvolvido pela [Secretaria de Tecnologia da Informação e Comunicação \(SeTIC\)](#) no modelo tradicional de aplicações web, para evitar a criação de duas interfaces (*desktop* e *mobile*), reduzir a quantidade de projetos que precisarão de manutenção e atingir os públicos-alvos do *mobile* e do *desktop* com uma única interface. Além disso, outra meta desse projeto era a melhoria na satisfação do usuário em relação à aplicação do [Sistema de Controle Acadêmico da Graduação \(CAGR\)](#).

Após a realização de um teste de usabilidade na interface proposta com 30 voluntários que representam o público-alvo do sistema, observou-se uma melhora no *score* do [System Usability Scale \(SUS\)](#) em relação ao módulo oficial e um aumento em duas respostas ímpares dos usuários no questionário [SUS](#). Além disso, constatou-se que as funcionalidades de notificações, acesso dos dados sem conexão com a rede e acesso ao status e posição da vaga em uma turma solicitada no pedido de matrícula foram bem avaliadas pela maioria dos voluntários desses testes. Em contrapartida, a instalação foi considerada menos útil por alguns participantes.

Palavras-chave: *PWA, Progressive Web Application, front-end*

ABSTRACT

With the growth of the use of mobile devices and internet access in the world, institutions and companies try to reach this new audience through the development of native or *web* applications with the **RWD** approach. Native applications have the advantage of exploiting the device's resources to the fullest and bringing a better user experience, however, the development and maintenance costs are high, since it is necessary to develop a system for each platform and provide support for each one of them. To avoid this problem, **RWD** allows the development of a platform-independent system at the cost of losing the quality of the user experience, since it cannot take advantage of all the device's resources. In this scenario, **PWA** emerged to solve the problems of both approaches, taking advantage of **RWD**'s platform independence and the extra resources provided by native applications such as *push* notifications and the possibility of installation.

Given this context, the objective of this project will be to develop an alternative interface with **PWA** for the new enrollment module at **UFSC**, which is being developed by **SeTIC** in the traditional model of web applications, to avoid creating two interfaces (*desktop* and *mobile*), reduce the number of projects that will need maintenance and reach the target audiences of *mobile* and *desktop* with a single interface. Furthermore, another goal of this project was to improve user satisfaction with the application of **CAGR**.

After carrying out a usability test on the proposed interface with 30 volunteers representing the system's target audience, an improvement was observed in the *score* of **SUS** in relation to the official module and an increase in two odd responses from users in the **SUS** questionnaire. In addition, it was found that the notification functionalities, data access without connection to the network and access to the status and position of the vacancy in a class requested in the enrollment request were well evaluated by the majority of the volunteers of these tests. On the other hand, the installation was considered less useful by some participants.

Keywords: **PWA**, *Progressive Web Application*, *front-end*

LISTA DE FIGURAS

Figura 1 – Panorama geral sobre o crescimento tecnológico em 2022	14
Figura 2 – Número de usuários na internet de 2012 até 2022	14
Figura 3 – Estrutura típica de uma aplicação web	18
Figura 4 – Utilização do JWT	20
Figura 5 – Interação entre os componentes do Model-View-Controller (MVC) . . .	21
Figura 6 – Arquitetura do Spring Security	24
Figura 7 – Market Share dos sistemas operacionais dos dispositivos móveis	25
Figura 8 – Service worker como proxy da aplicação	27
Figura 9 – Ciclo de vida de um service worker	28
Figura 10 – Manifesto do Instagram	29
Figura 11 – Atores envolvidos no envio de push messages e seus relacionamentos . .	31
Figura 12 – Diagrama de sequência contendo as etapas para efetuar o envio de push messages para um cliente	33
Figura 13 – Formato da resposta para o SUS	35
Figura 14 – Gráfico que associa os scores do SUS às notas na escala de letras	37
Figura 15 – Construção da grade de horários no MatrUFSC	38
Figura 16 – Construção da grade de horários no MatrUFSC2	39
Figura 17 – Construção da grade de horários no CAGR	40
Figura 18 – Busca de turmas no CAGR	40
Figura 19 – Matrícula	42
Figura 20 – Grade de horários	43
Figura 21 – Diagrama de casos de uso da Proof of Concept (PoC)	45
Figura 22 – Modelagem conceitual do banco de dados	46
Figura 23 – Diagrama de casos de uso da PWA	48
Figura 24 – Modelagem conceitual do banco de dados	49
Figura 25 – Fluxo ao registrar um aluno	50
Figura 26 – Tela para registrar o aluno	51
Figura 27 – Tela para registrar o aluno com erros de validação retornados pelo servidor	51
Figura 28 – Validação ao registrar um aluno	52
Figura 29 – Fluxo ao autenticar um aluno	53
Figura 30 – Tela para autenticar o aluno	54
Figura 31 – JSON Web Token (JWT) no Local Storage após uma autenticação bem-sucedida	54
Figura 32 – Botão para desautenticação ou sair	55
Figura 33 – Fluxo ao autorizar um aluno	56
Figura 34 – Fluxo ao criar e editar um pedido de matrícula	57
Figura 35 – Texto sobre conflitos e carga horária	58

Figura 36 – Grade de horários	58
Figura 37 – Campo de turmas com turmas selecionadas	59
Figura 38 – Buscando pelo nome do professor no campo de turmas	59
Figura 39 – Texto sobre a carga horária total quando está fora dos limites definidos	59
Figura 40 – Texto sobre a carga horária total quando está dentro dos limites definidos	59
Figura 41 – Conflito na grade de horários	60
Figura 42 – Fluxo ao visualizar grade de horários e espelho de matrícula com conexão com a rede	62
Figura 43 – Fluxo ao visualizar grade de horários e espelho de matrícula sem conexão com a rede	63
Figura 44 – Espelho de matrícula	63
Figura 45 – Exemplo de execução da stored procedure quando um aluno entra em uma turma	65
Figura 46 – Exemplo de execução da stored procedure quando um aluno sai de uma turma	66
Figura 47 – Resultado da requisição para o endpoint <i>/pedidoMatricula</i> guardado na cache	68
Figura 48 – Mensagem de alerta e botão de instalação e notificações removidos quando falta conexão	69
Figura 49 – Espelho de matrícula sem o botão de editar quando falta conexão	69
Figura 50 – Fluxo ao habilitar as notificações	70
Figura 51 – Fluxo ao desabilitar as notificações	71
Figura 52 – Botão para habilitar notificações	73
Figura 53 – Diagrama de atividades mostrando os passos para habilitar as notificações	73
Figura 54 – Notificação ao sair da fila de espera de uma turma	74
Figura 55 – Notificação ao entrar na fila de espera de uma turma	74
Figura 56 – Notificação ao sair da fila de espera de uma turma	75
Figura 57 – Notificação ao entrar na fila de espera de uma turma	76
Figura 58 – Botão de desativar notificações	76
Figura 59 – Notificação depois de habilitar as notificações	77
Figura 60 – Exemplo de execução da stored procedure quando um aluno entra em uma turma para determinar se deve ser notificado ou não	78
Figura 61 – Exemplo de execução da stored procedure quando um aluno sai de uma turma para determinar se deve ser notificado ou não	79
Figura 62 – Fluxo do envio de notificações após criar ou editar um pedido de matrícula	81
Figura 63 – Botão de instalação	83
Figura 64 – Etapas do teste de usabilidade que foram executadas pelos voluntários	84
Figura 65 – Primeira tarefa que foi proposta aos participantes do teste	85
Figura 66 – Navegadores usados pelos voluntários do teste	87

Figura 67 – Dispositivos usados pelos voluntários do teste	87
Figura 68 – Porcentagem de usuários que conseguiram concluir as atividades por tarefa	89
Figura 69 – Perguntas sobre o espelho de matrícula com status e posição da vaga .	90
Figura 70 – Respostas sobre a posição da vaga no espelho de matrícula	90
Figura 71 – Respostas sobre o status da vaga no espelho de matrícula	91
Figura 72 – Perguntas sobre as notificações	91
Figura 73 – Respostas sobre a utilidade das notificações	92
Figura 74 – Respostas sobre a clareza das mensagens nas notificações	92
Figura 75 – Perguntas sobre a instalação	93
Figura 76 – Respostas sobre a utilidade da instalação	93
Figura 77 – Respostas sobre o valor da possibilidade de instalação	94
Figura 78 – Alguns dos comentários sobre a instalação	94
Figura 79 – Perguntas sobre o espelho de matrícula	95
Figura 80 – Perguntas sobre a grade de horários e a interface da aplicação desenvol- vida quando há perda de conexão com a rede	95
Figura 81 – Respostas sobre a utilidade de ver as informações estando sem conexão com a rede	96
Figura 82 – Respostas sobre a clareza das informações no espelho de matrícula . . .	96
Figura 83 – Respostas sobre a clareza das informações na grade de horários	97
Figura 84 – Respostas sobre a facilidade de identificar a falta de conexão pela aplicação	97

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
BaaS	Backend-as-a-Service
CAAE	Certificado de Apresentação de Apreciação Ética
CAGR	Sistema de Controle Acadêmico da Graduação
CEPSH-UFSC	Comitê de Ética em Pesquisa com Seres Humanos da Universidade Federal de Santa Catarina
DoS	Denial-of-service attack
ECDSA	Elliptic Curve Digital Signature Algorithm
FCM	Firestore Cloud Messaging
HMAC	Hash-based message authentication code
HTTP	Hypertext Transfer Protocol
HTTPS	Hyper Text Transfer Protocol Secure
IAA	Índice de Aproveitamento Semestral Acumulado
JSON	JavaScript Object Notation
JWT	JSON Web Token
MVC	Model-View-Controller
PoC	Proof of Concept
PWA	Progressive Web Application
RSA	Rivest–Shamir–Adleman
RWD	Responsive Web Development
SDK	Software Development Kit
SeTIC	Secretaria de Tecnologia da Informação e Comunicação
SGBD	Sistema de gerenciamento de banco de dados
SIGAA	Sistema Integrado de Gestão de Atividades Acadêmicas
SUS	System Usability Scale
TCLE	Termo de Consentimento Livre e Esclarecido
UA	User agent
UFSC	Universidade Federal de Santa Catarina
UnB	Universidade de Brasília
URL	Uniform Resource Locator
VAPID	Voluntary Application Server Identification

SUMÁRIO

1	INTRODUÇÃO	13
1.1	JUSTIFICATIVA	15
1.2	OBJETIVOS	15
1.2.1	Objetivo Geral	15
1.2.2	Objetivos Específicos	15
1.3	MÉTODO DE PESQUISA	16
1.4	ESTRUTURA DO DOCUMENTO	17
2	FUNDAMENTAÇÃO TEÓRICA E TECNOLÓGICA	18
2.1	APLICAÇÕES WEB	18
2.2	JSON WEB TOKEN	18
2.2.1	Estrutura de um JWT	19
2.2.2	Como utilizar	20
2.3	MODEL-VIEW-CONTROLLER	21
2.4	FERRAMENTAS	22
2.4.1	React	22
2.4.2	Bold	22
2.4.3	Node.js e Express.js	22
2.4.4	Spring	23
2.4.4.1	Spring Boot	23
2.4.4.2	Spring Web MVC	23
2.4.4.3	Spring Security	23
2.5	APLICAÇÕES NATIVAS	24
2.6	APLICAÇÕES HÍBRIDAS	24
2.7	PROGRESSIVE WEB APPLICATION	25
2.8	SERVICE WORKER	26
2.8.1	Ciclo de vida	27
2.9	MANIFESTO	28
2.10	ARMAZENANDO DADOS NO CLIENTE - CACHE API, WEB STORAGE API E INDEXEDDB	29
2.11	NOTIFICAÇÕES NAS APLICAÇÕES WEB	30
2.11.1	Web Push API	30
2.11.2	Notification API	32
2.12	FIREBASE	34
2.12.1	Firebase Cloud Messaging	34
2.13	USABILIDADE E SYSTEM USABILITY SCALE	35
3	TRABALHOS RELACIONADOS	38
3.1	MATRUFSC E MATRUFSC2	38

3.2	SISTEMA DE CONTROLE ACADÊMICO DA GRADUAÇÃO	39
3.3	UNIVERSIDADE DE BRASÍLIA	40
3.4	ANÁLISE COMPARATIVA	41
4	DESENVOLVIMENTO	44
4.1	PROVA DE CONCEITO	44
4.1.1	Implementação	44
4.1.1.1	Vendo exercícios de outros usuários	44
4.1.1.2	Seguindo um usuário e parando de seguir	46
4.1.1.3	Salvando treinos	46
4.1.1.4	Compartilhar novos exercícios	46
4.1.1.5	Mandando notificações quando for criado um exercício	47
4.2	PWA DO MÓDULO DE MATRÍCULA	47
4.2.1	Registrando um aluno	50
4.2.1.1	Cliente	50
4.2.1.2	Servidor	52
4.2.2	Autenticando um aluno	52
4.2.2.1	Cliente	53
4.2.2.2	Servidor	54
4.2.2.3	Desautenticação do aluno	55
4.2.3	Autorizando um aluno	55
4.2.3.1	Cliente	55
4.2.3.2	Servidor	56
4.2.4	Criando ou editando um pedido de matrícula	57
4.2.4.1	Cliente	58
4.2.4.2	Servidor	60
4.2.5	Visualizando grade de horários e status da vaga através do espelho de matrícula	61
4.2.5.1	Cliente	62
4.2.5.2	Servidor	64
4.2.5.3	Sem conexão com a rede	64
4.2.5.3.1	<i>Service Worker</i>	64
4.2.5.3.2	<i>Cliente</i>	68
4.2.6	Notificações	69
4.2.6.1	Passos iniciais	70
4.2.6.2	Cliente	71
4.2.6.2.1	<i>Configurando a SDK</i>	71
4.2.6.2.2	<i>Habilitando as notificações</i>	72
4.2.6.2.3	<i>Recebendo as notificações</i>	73
4.2.6.2.4	<i>Desabilitando as notificações</i>	76

4.2.6.3	Servidor	76
4.2.6.3.1	<i>Inicializando a Firebase SDK Admin</i>	76
4.2.6.3.2	<i>Habilitando e desabilitando notificações</i>	77
4.2.6.3.3	<i>Determinando quais alunos devem receber as notificações</i>	77
4.2.6.3.4	<i>Enviando notificações</i>	78
4.2.7	Instalação	80
4.3	TESTE DE USABILIDADE	83
4.3.1	Tarefas	85
4.3.2	Questionários	85
4.3.3	Resultados	86
4.3.3.1	Utilidade das funcionalidades propostas	88
4.3.3.1.1	<i>Espelho de matrícula com status e posição da vaga</i>	89
4.3.3.1.2	<i>Notificações</i>	89
4.3.3.1.3	<i>Instalação</i>	90
4.3.3.1.4	<i>Visualização do espelho de matrícula e grade de horários independente da conexão com a rede</i>	91
5	CONCLUSÃO E TRABALHOS FUTUROS	98
	REFERÊNCIAS	100
	APÊNDICE A – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO	106
	APÊNDICE B – QUESTIONÁRIO SOBRE O MÓDULO DO SISTEMA DE CONTROLE ACADÊMICO DA GRADUAÇÃO	108
	APÊNDICE C – TAREFAS REALIZADAS PELOS VOLUNTÁRIOS	115
C.1	TAREFA 1	115
C.2	TAREFA 2	115
C.3	TAREFA 3	115
C.4	TAREFA 4	115
C.5	TAREFA 5	116
C.6	TAREFA 6	116
	APÊNDICE D – QUESTIONÁRIO SOBRE A INTERFACE PROPOSTA	118
	APÊNDICE E – CÓDIGOS FONTES DAS APLICAÇÕES DESENVOLVIDAS NO PROJETO	136
	APÊNDICE F – ARTIGO NO FORMATO SBC	137
	ANEXO A – REQUISITOS OFICIAIS RELACIONADOS AOS VETERANOS	157
	ANEXO B – PARECER DE APROVAÇÃO DO CEPESH-UFSC	179

1 INTRODUÇÃO

Desde 2021, a quantidade de usuários na internet teve um aumento de 192 milhões e atingiu um valor de 4,95 bilhões de usuários, representando cerca de 62,5% da população mundial (KEMP, 2022). Simultaneamente, a quantidade de usuários únicos com celular aumentou em 95 milhões e atingiu um valor de 5,31 bilhões, representando cerca de 67,1% da população mundial (KEMP, 2022). As Figuras 1 e 2 mostram esses crescimentos e o histórico de total de usuários na internet desde 2012 respectivamente.

Apesar do número de usuários de celulares ter aumentado significativamente, os computadores ainda possuem cerca de 42,54% das visitas em sites, representando uma parcela significativa nos acessos, enquanto que os celulares possuem 54,98% (STATCOUNTER, 2022a). Com isso, as empresas que desenvolvem softwares possuem novos usuários a serem alcançados com diferentes formas de acessar a aplicação e o seguinte questionamento surge: como alcançar todos esses usuários com um único código e manter uma boa experiência do usuário ao utilizar a aplicação.

Como as aplicações web com a abordagem RWD acabam não trazendo a melhor experiência aos usuários de celulares, já que não possuem um desempenho superior aos aplicativos nativos e acesso a alguns recursos disponíveis nos dispositivos móveis, como notificações, localização, fotos e câmera (MALAVOLTA, 2016), essa abordagem pode se tornar inadequada para as empresas, apesar de conseguir atingir todos os usuários com um único código. Da mesma forma, as aplicações nativas, que possuem acesso aos recursos dos dispositivos e uma performance superior às aplicações web, podem custar caro caso a empresa queira desenvolver a aplicação para todas as plataformas disponíveis no mercado, visando atingir todos os usuários possíveis. Esse alto custo é causado pelos seguintes motivos (MALAVOLTA, 2016; CORRAL *et al.*, 2012):

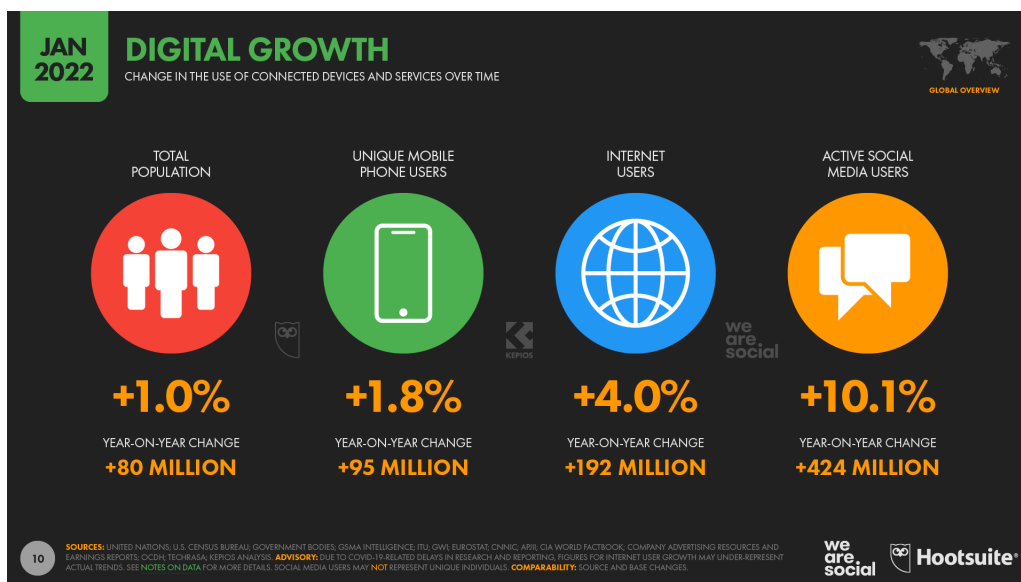
- intransponibilidade entre as plataformas, impedindo a reutilização de código e, por consequência, criando código redundante nos projetos;
- treinamentos sobre as plataformas e linguagens, podendo ser necessário aprender até sete linguagens diferentes (CHARLAND; LEROUX, 2011) caso haja necessidade por parte da empresa;
- manutenção e testes de projetos com código redundante.

Para contornar esses problemas, o conceito de PWA foi criado pela Google, unindo a experiência das aplicações nativas ao usuário com a independência de plataforma das aplicações web, permitindo que usuário utilize a aplicação sem conexão, receba notificações e instale a aplicação no celular (SAM RICHARD, 2020).

A meta do presente trabalho é o desenvolvimento um front-end PWA para o módulo de matrícula da Universidade Federal de Santa Catarina como uma alternativa para o que

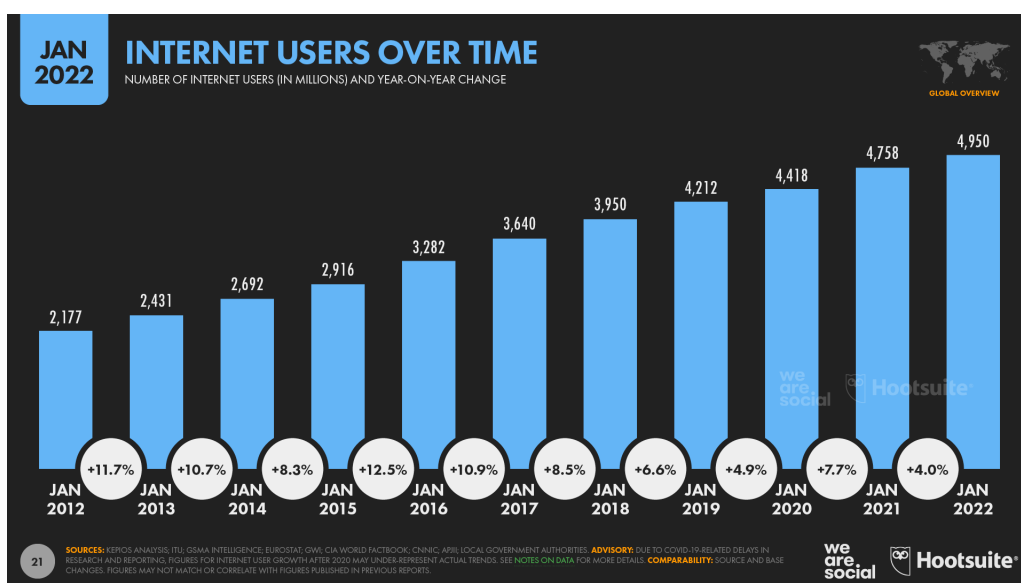
está sendo desenvolvido pela [SeTIC](#) no modelo web tradicional, evitando o desenvolvimento de aplicações nativas e web para atender todo os usuários e reduzir a quantidade de projetos que a universidade precisaria dar suporte. Além dessas contribuições, outra será melhorar a usabilidade desse sistema, tornando-o mais intuitivo aos usuários mais novos.

Figura 1 – Panorama geral sobre o crescimento tecnológico em 2022



Fonte: (KEMP, 2022)

Figura 2 – Número de usuários na internet de 2012 até 2022



Fonte: (KEMP, 2022)

1.1 JUSTIFICATIVA

O desenvolvimento de aplicações que queiram alcançar todos os usuários presentes no mercado, mantendo uma boa experiência para eles, torna-se custoso para as instituições devido a diversos fatores, como por exemplo: treinamento para compreender uma gama de tecnologias, produção de código e testes redundantes. Diante desse contexto, a [PWA](#) surgiu para diminuir os custos de desenvolvimento e manutenção das aplicações multiplataformas através da união dos recursos das aplicações nativas com a independência de plataforma das aplicações web, reduzindo a quantidade de código necessário para atender os usuários com dispositivos diferentes sem afetar tanto a experiência deles ao utilizar a aplicação.

Como [SeTIC](#) está desenvolvendo uma aplicação (back-end e front-end) nova para o módulo de matrícula da [UFSC](#) no modelo web tradicional, esse trabalho se propõe a produzir um front-end [PWA](#) alternativo para esse módulo de forma a atender todos os usuários e reduzir a quantidade de projetos que a universidade precisaria dar suporte. Além disso, esse projeto pretende melhorar a satisfação dos usuários do sistema e a experiência através dos recursos disponíveis da tecnologia, como notificações e ser instalável nos dispositivos.

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Esse trabalho tem como objetivo a aplicação das tecnologias da [PWA](#), que é recente no mercado, para solucionar um problema real, um módulo de matrícula defasado no âmbito tecnológico e estético, trazendo benefícios como um produto útil e um roadmap para desenvolvimento de [PWAs](#).

1.2.2 Objetivos Específicos

Os objetivos específicos para esse trabalho são:

- Desenvolver o cliente alternativo utilizando a biblioteca React e os componentes do Bold Design System ([BRIDGE, 2022](#)). O cliente deve permitir, quando houver suporte do sistema, que o aluno instale no dispositivo (celular, computador ou tablet), receba notificações, caso seja permitido, e veja as informações relacionadas ao pedido de matrícula, caso não tenha acesso à internet;
- Devido ao atraso no desenvolvimento do servidor oficial da [SeTIC](#), desenvolver o servidor com escopo reduzido em relação ao que esperado pelo documento de requisitos para o cliente consumir a [Application Programming Interface \(API\)](#). Esse servidor deve utilizar Spring Boot, Java 8 e Kotlin;

- Avaliar se houve melhora na satisfação dos usuários em relação ao cliente alternativo através de testes com questionários respondidos por voluntários;
- Produzir um roadmap de desenvolvido para PWAs para compartilhar o conhecimento obtido;
- O trabalho resultante deve seguir as normas de formatação de trabalhos de conclusão de curso da UFSC.

1.3 MÉTODO DE PESQUISA

Para atingir os objetivos descritos anteriormente, o método desta pesquisa será composto pelas seguintes etapas:

- Conhecer as aplicações que são utilizadas pelos alunos, sejam elas oficiais ou não para observar as principais funcionalidades propostas e como cada aplicação aborda e soluciona os problemas associados ao pedido de matrícula;
- Ler o documento de requisitos do módulo em desenvolvimento na UFSC, presente no anexo A, para identificar as funcionalidades que estão sendo propostas para o novo módulo de matrícula, focando apenas nas que estão associadas aos alunos e no registro do pedido de matrícula;
- Desenvolver uma PoC para estudar sobre as tecnologias associadas às PWAs, ganhando experiência e entendendo as limitações delas;
- Desenvolver um servidor com escopo extremamente reduzido para o cliente PWA consumir a API, visto que não foi possível utilizar o servidor da SeTIC devido aos atrasos no desenvolvimento. Esse servidor torna-se necessário por causa de um teste de usabilidade no cliente proposto;
- Desenvolver o cliente PWA, seguindo os requisitos que foram descritos no documento de requisitos, relacionados aos alunos e seu pedido de matrícula;
- Implantar o servidor e o cliente em uma plataforma gratuita para a realização de um teste de usabilidade;
- Elaborar o teste de usabilidade composto por um conjunto de tarefas que os usuários devem realizar e dois questionários para mensurar a satisfação da aplicação oficial do CAGR e da que está sendo proposta com o SUS visto que o foco desse projeto está em avaliar a satisfação dos voluntários;
- Submeter a pesquisa para o Comitê de Ética em Pesquisa com Seres Humanos da Universidade Federal de Santa Catarina (CEPSH-UFSC) para aprovação;

- Executar o teste de usabilidade e coletar os dados oriundos dos questionários;
- Realizar uma comparação quantitativa entre as duas aplicações para avaliar se houve uma melhora da satisfação dos voluntários na solução proposta através do cálculo do *score* do SUS;
- Verificar se o público-alvo considerou as novas funcionalidades (notificações, ver espelho de matrícula e grade de horários estando sem conexão e visualizar o status da vaga) como úteis.

1.4 ESTRUTURA DO DOCUMENTO

Este capítulo apresenta a justificativa, objetivos gerais e específicos e metodologia desse trabalho. O capítulo 2 discorre sobre os principais conceitos na construção de uma PWA e as tecnologias e bibliotecas que foram utilizadas para o desenvolvimento do cliente e dos experimentos iniciais. O capítulo 3 apresenta uma análise acerca dos trabalhos relacionados. O capítulo 4 descreve os experimentos iniciais que foram feitos para fins de estudo da PWA, o servidor, cliente e teste de usabilidade que foram desenvolvidos e os resultados obtidos do teste. Por fim, o capítulo 5 apresenta as conclusões e atividades a serem desenvolvidas nesse trabalho.

2 FUNDAMENTAÇÃO TEÓRICA E TECNOLÓGICA

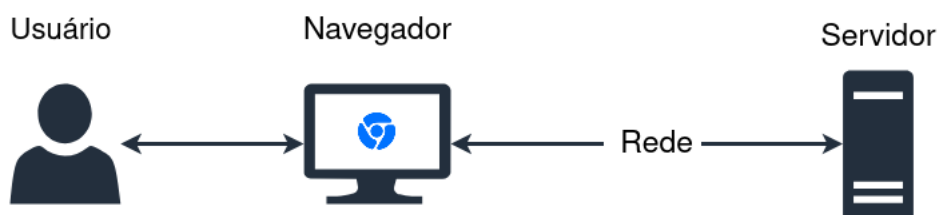
Nas seções seguintes serão descritos conceitos necessários para compreender a implementação que foi feita no decorrer do semestre visto que representa o arcabouço tecnológico para a solução proposta por esse projeto.

2.1 APLICAÇÕES WEB

Uma aplicação web é um software que roda em um navegador, permitindo que o usuário realize as tarefas precisando apenas de acesso à rede. Diferente das aplicações desktop, elas não precisam ser instaladas na máquina, ou seja, não ocupam espaço de memória no computador, e funcionam de forma independente ao sistema operacional, evitando problemas de aplicações dependentes de plataforma, como produção de código e testes redundantes para atender as diferentes plataformas.

A Figura 3 mostra a estrutura mais comum desse tipo de aplicação. Comumente, uma aplicação web pode ser dividida entre cliente, também conhecido como front-end, e servidor ou back-end. O cliente é responsável por interagir com o usuário, permitindo que o mesmo realize as tarefas e mostrando informações, e enviar requisições ao servidor para atender às necessidades do usuário. Enquanto isso, o servidor é responsável por receber essas requisições, processá-las e retornar uma resposta ao cliente. As requisições são enviadas pela rede através de protocolos de comunicação, como [Hypertext Transfer Protocol \(HTTP\)](#) e [Hyper Text Transfer Protocol Secure \(HTTPS\)](#).

Figura 3 – Estrutura típica de uma aplicação web



Fonte: Própria

2.2 JSON WEB TOKEN

O [JWT](#) é um padrão, definido pelo RFC 7519 ([JONES; BRADLEY; SAKIMURA, 2015](#)), cujo objetivo é transmitir ou armazenar de forma compacta e segura objetos JSON entre aplicações para autorização ou compartilhamento de informações ([LIMA, 2021](#)), como os dados do usuário logado. Além disso, [JWT](#) é assinado digitalmente com uma chave secreta com o algoritmo [Hash-based message authentication code \(HMAC\)](#) ou um par de chave pública e privada [Rivest-Shamir-Adleman \(RSA\)](#) ou [Elliptic Curve](#).

Digital Signature Algorithm (ECDSA), permitindo verificar a confiabilidade da informação (OKTA, 2022).

Sobre a autorização, o JWT é obtido após uma autenticação bem-sucedida e, nas requisições subsequentes ao servidor, esse token é adicionado ao cabeçalho das requisições para o servidor verificar a validade dele e se o usuário tem acesso a um determinado recurso ou rota (LIMA, 2021; OKTA, 2022).

2.2.1 Estrutura de um JWT

O JWT consiste de três partes separadas por um ponto final ("."), seguindo este formato:

`<header>.<payload>.<signature>`

O header ou cabeçalho identifica que é um JWT e qual algoritmo foi utilizado para realizar a assinatura através das chaves *typ* e *alg* respectivamente. O código ?? mostra um exemplo de header.

```
{  
  "alg": "HS256",  
  "typ": "JWT"  
}
```

Fonte: (OKTA, 2022)

Listing 2.2.1 – Exemplo de um header

O payload ou corpo, contém declarações sobre a entidade associada ao token e dados adicionais. Essas informações também são conhecidas como *claims*. As *claims* podem ser divididas em três grupos: registradas, públicas e privadas. As registradas são pré-definidas, cujo preenchimento não é obrigatório mas recomendado para a validação dos tokens, e seus valores são (OKTA, 2022):

- *sub* (*subject*): identificador da entidade a qual o token se refere (LIMA, 2021);
- *iss* (*issuer*): emissor do token;
- *exp* (*expiration*): timestamp de quando o token irá expirar;
- *iat* (*issued at*): timestamp de quando o token foi criado;
- *aud* (*audience*): destinatário do token, representa a aplicação que irá usá-lo.

As públicas são valores que foram registrados publicamente na IANA JSON Web Token Registry, enquanto que as privadas não são registradas. O código ?? mostra um exemplo de payload.

```
{  
  "sub": "1234567890",  
  "name": "John Doe",  
  "admin": true  
}
```

Fonte: (OKTA, 2022)

Listing 2.2.2 – Exemplo de um payload

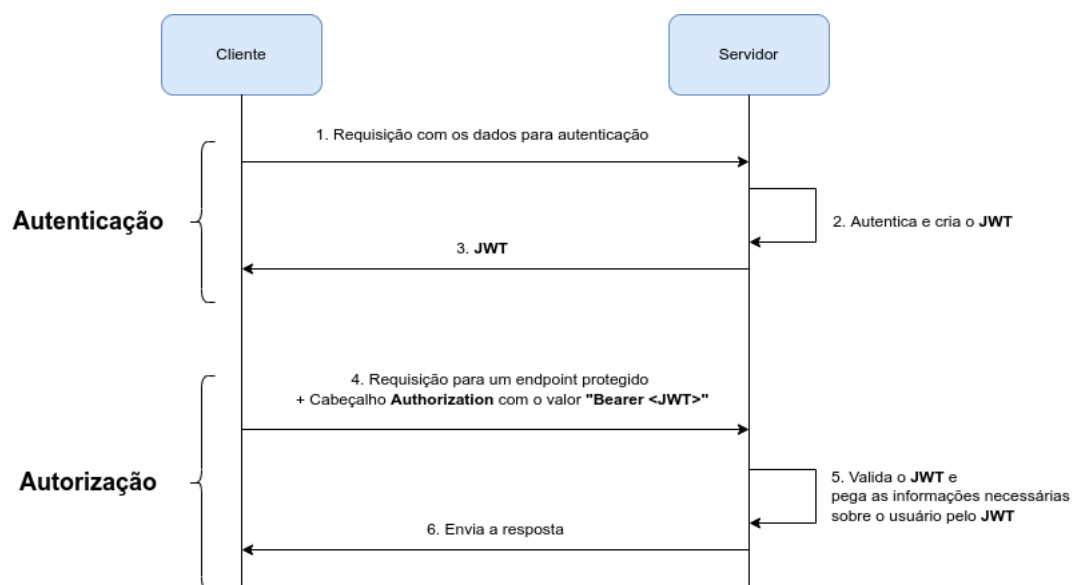
Por fim, a signature representa a assinatura para a aplicação verificar a integridade dos dados, ou seja, se ela não foi alterada antes de chegar no destino, e seu valor é definido da seguinte forma:

HMAC(<header>.<payload>, <segredo>) ou
RSA(<header>.<payload>, *chave_{Pública}*, *chave_{Privada}*) ou
ECDSA(<header>.<payload> , *chave_{Pública}*, *chave_{Privada}*)

2.2.2 Como utilizar

Depois da autenticação bem-sucedida, o **JWT** será entregue ao cliente que deverá inserir o token no cabeçalho **Authorization** com o valor no formato '*Bearer <jwt>*' para todas as requisições que forem acessar rotas ou recursos protegidos (OKTA, 2022). A figura 4 mostra o fluxo descrito para usar o **JWT**.

Figura 4 – Utilização do JWT

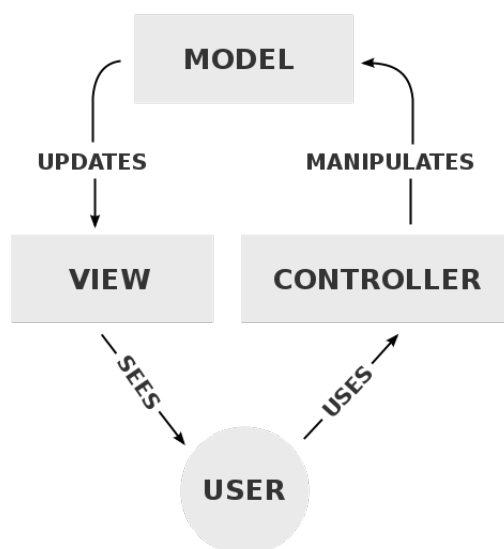


Fonte: Própria

2.3 MODEL-VIEW-CONTROLLER

O **MVC** é um padrão de projeto que foi proposto por Trygve Reenskaug na década de 1970 cujo foco é na divisão de responsabilidades nas seguintes camadas: Modelo (*Model*), Visualização (*View*) e Controle (*Controller*) ([REENSKAUG, 1979](#)). Sendo algumas das vantagens desse padrão: a separação das regras de negócio da apresentação dos dados, reutilização do código da camada de visualização para outros projetos com menos esforço e desenvolvimento paralelo para cada camada devido à baixa dependência entre as camadas. O Modelo é responsável pela implementação da lógica das regras de negócio e manipulação dos dados, ou seja, a interação com a camada de persistência. Por outro lado, a Visualização é responsável pela apresentação dos dados e interação com o usuário e o Controle é responsável por intermediar a comunicação entre as duas camadas anteriores, podendo ser uma atualização no modelo ou na visualização dos dados. A figura 5 mostra a interação entre as camadas.

Figura 5 – Interação entre os componentes do **MVC**



Fonte: ([WIKIPÉDIA, 2011](#))

Imaginando um cenário para exemplificar essa interação seria a autenticação de um usuário em um aplicação qualquer. Nesse caso, o usuário preencheria os dados e clicaria em um botão, presente na Visualização, para enviar os dados para o login. Em seguida, o Controle receberia a requisição, validaria os dados e solicitaria para o Modelo para autenticar, verificando se o usuário realmente existe e se as senhas são iguais. Por fim, a Visualização é atualizada com o resultado da autenticação, podendo liberar as outras telas ou mostrar um erro na autenticação.

2.4 FERRAMENTAS

Nesta seção, serão detalhadas todas as ferramentas que foram usadas para o desenvolvimento das aplicações desse projeto para introduzir conceitos e termos que serão utilizados no decorrer do texto.

2.4.1 React

Segundo um artigo da empresa Mozilla, a biblioteca React pode ser definida como (MOZILLA, 2022d):

Segundo seu slogan oficial, React é uma biblioteca para construção de interfaces de usuário. React não é um framework – nem mesmo é exclusivo para web. É utilizado com outras bibliotecas para renderização em certos ambientes. (...) Para construir para web, desenvolvedores usam React em conjunto com ReactDOM. React e ReactDOM são frequentemente discutidos nos mesmos espaços e utilizados para resolver os mesmos problemas como outros arcabouços (frameworks). Quando referimos React como "arcabouço"(framework) estamos trabalhando com o termo/entendimento coloquial. A meta primária do React é minimizar os erros que ocorrem quando os desenvolvedores estão construindo UIs (User Interface). Isto é devido ao uso de componentes - autocontidos, partes lógicas de códigos que descrevem uma parte da interface do usuário. Estes componentes são adicionados para criar uma UI completa e o React concentra muito do trabalho de renderizar, proporcionando que se concentre no projeto de UI.

2.4.2 Bold

O Bold é uma biblioteca open source com componentes React reutilizáveis e customizáveis seguindo o Bold Design System (BRIDGE, 2022) que foi desenvolvida pelo Laboratório Bridge. A biblioteca oferece diversos componentes como, por exemplo, botões, campos de texto, data e senha, modais, tabelas, etc. Além disso, disponibiliza o uso e customização de temas e paleta de cores da aplicação e ferramentas para a internacionalização do cliente. [Nesta página](#) é possível encontrar todos os componentes disponibilizados e o código fonte pode ser encontrado neste [link](#).

2.4.3 Node.js e Express.js

Node ou Node.js é um ambiente em tempo de execução que permite executar código na linguagem JavaScript fora do contexto de um navegador, sendo utilizado para desenvolver servidores em JavaScript (MOZILLA, 2022f). O Express ou Express.js é um framework do Node para simplificar e facilitar o desenvolvimento de servidores, permitindo gerenciar requisições HTTP e suas rotas, tratamento de exceções e adicionar novos processos de requisição por meio de "middleware" em qualquer ponto da "fila" de requisições (MOZILLA, 2022f).

2.4.4 Spring

O Spring é um framework que foi criado para facilitar o desenvolvimento de aplicações corporativas escritas em Java utilizando os conceitos de Inversão de Controle e Injeção de Dependências. Nesse framework, existem ferramentas para persistência de dados, segurança, desenvolvimento web, elaboração de testes de integração e unitários e micro-serviços (SPRING, 2022). Dentre essas ferramentas, foram utilizadas as seguintes para o desenvolvimento da aplicação proposta por esse projeto: Spring Boot, Spring Web MVC e Spring Security. Todas são descritas nas próximas seções.

2.4.4.1 Spring Boot

O Spring Boot é um projeto voltado para acelerar e facilitar, principalmente nas fases iniciais, o desenvolvimento das aplicações que utilizam o Spring Framework ao pré-definir certas configurações e componentes necessários para subir, do jeito mais rápido e simples possível, e implantar uma aplicação (SPRING, 2022; TREINAWEB, 2022). Além disso, caso as configurações padrão não atendam as necessidades do projeto, ela permite alterá-las através de anotações.

2.4.4.2 Spring Web MVC

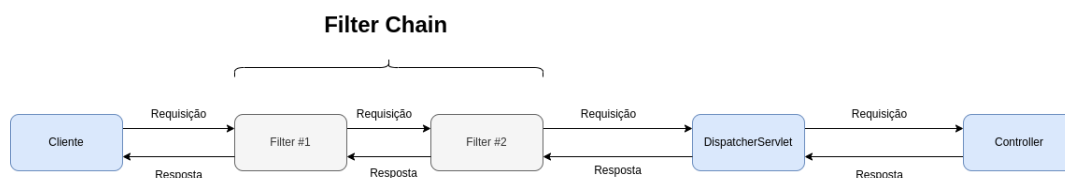
O Spring Web MVC oferece ferramentas para o desenvolvimento de um servidor web, permitindo que trate as requisições HTTP seguindo o modelo MVC. Ela permite mapear o tratamento das requisições para os métodos de um controller através das anotações como *@RequestMapping*, *@PostMapping* e *@GetMapping*. Todas as requisições são recebidas pelo *DispatchServlet* e serão delegadas conforme o mapeamento que foi comentado previamente. Além disso, essa ferramenta fica responsável por transformar os parâmetros recebidos pela [Uniform Resource Locator \(URL\)](#) ou o corpo de uma requisição POST na classe esperada pelo parâmetro e pela construção da resposta no formato JSON, caso o método possua um retorno, ou adicionando um código de erro.

2.4.4.3 Spring Security

O Spring Security é um projeto desenvolvido para a parte de autenticação e autorização das aplicações e permite integração com Spring Web MVC para proteger endpoints para usuários não autenticados, liberar endpoints que podem ser requisitados livremente. Além disso, essa ferramenta permite configurar a autorização e autenticação conforme a necessidade do protejo, como, por exemplo, a utilização de [JWT](#) para realizar a autorização de determinadas funcionalidades através da inserção de um filtro. A figura 6 mostra a arquitetura dessa ferramenta e o fluxo no tratamento de uma requisição. Nessa arquitetura, um *Filter* é uma entidade que modifica ou adapta a requisição e sua resposta antes de chegarem no *DispatchServlet* e, conseqüentemente, nos controllers e normalmente

são utilizados para realizar autenticação, auditoria, compressão e encriptação dos dados. Além disso, a *Filter Chain* representa uma corrente de filtros para determinar a ordem da aplicação dos mesmos e permite que um filtro pare a execução sequencial da corrente caso encontre um erro.

Figura 6 – Arquitetura do Spring Security



Fonte: Própria

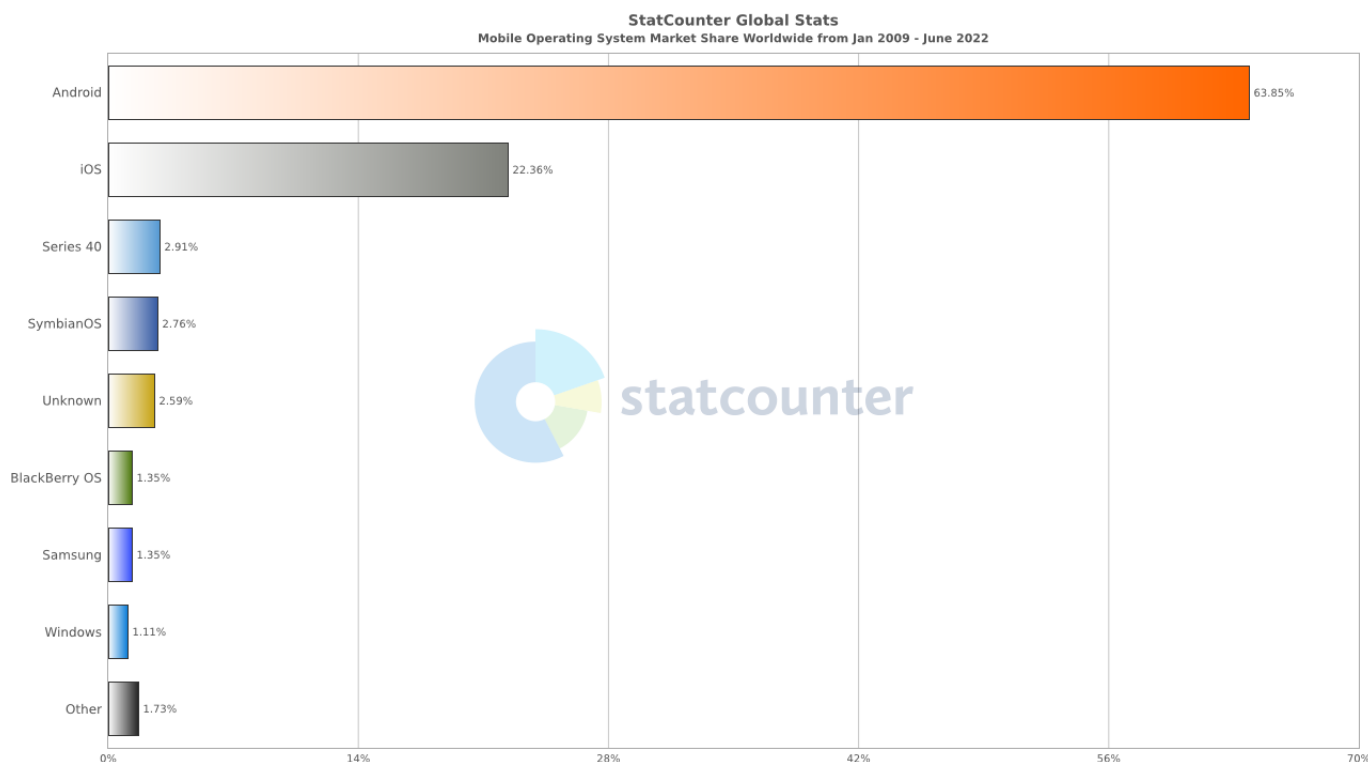
2.5 APLICAÇÕES NATIVAS

Uma aplicação nativa é um software desenvolvido para ser usado em uma plataforma ou dispositivo específico. Uma plataforma, que pode ser um software ou hardware, fornece a infraestrutura para uma aplicação ser desenvolvida e executada, além de oferecer ferramentas através de APIs para acessar os recursos. Alguns exemplos de plataformas seriam os sistemas operacionais Windows, Linux, iOS e Android. Essas APIs permitem que os desenvolvedores criem aplicações com experiência mais rica para o usuário, ao permitir acessar as câmeras, localização geográfica, sensores e outros recursos do dispositivo, e com uma performance superior (MALAVOLTA, 2016), algo que as aplicações web não oferecem por completo. Todavia, as aplicações nativas não permitem a transponibilidade de código entre as plataformas, ou seja, um código feito para o iOS não poderá ser utilizado para um Android e vice-versa, acarretando no desenvolvimento de diferentes aplicações que atendem às mesmas necessidades e que precisarão de testes iguais e desenvolvedores especializados para dar manutenção e elaborar melhorias na aplicação que não poderão ser utilizados em projetos de outras plataformas. E, como as empresas e instituições desejam atender todos os usuários com seus aplicativos e o market share dos sistemas operacionais dos dispositivos móveis é dividido, como pode ser visto na Figura 7, elas acabam enfrentando essas questões de intransponibilidade entre plataformas, que aumentam drasticamente o custo do desenvolvimento do software.

2.6 APLICAÇÕES HÍBRIDAS

As aplicações híbridas são uma solução desenvolvida para resolver o problema da dependência de plataforma das aplicações nativas, obrigando as instituições a gastarem mais tempo e dinheiro para desenvolver o mesmo código, em relação às regras de negócio,

Figura 7 – Market Share dos sistemas operacionais dos dispositivos móveis



Fonte: (STATCOUNTER, 2022b)

para plataformas diferentes (MALAVOLTA, 2016). De forma resumida, elas são aplicações web, ou seja, utilizam tecnologias como HTML, CSS e Javascript, que rodam um container nativo, que permite que elas acessem determinadas funcionalidades do dispositivo, como geolocalização e câmera, e são disponibilizadas nas apps stores das plataformas para os usuários baixarem (MALAVOLTA, 2016). Devido à camada de abstração provida pelo container, a performance dessas aplicações tende a cair um pouco em relação às outras (MALAVOLTA, 2016). Alguns dos frameworks para o desenvolvimento dessas aplicações são: React Native, Ionic, Flutter, Cordova e Xamarin.

2.7 PROGRESSIVE WEB APPLICATION

Como foi descrito previamente na Introdução, o mercado encontra-se dividido entre dispositivos móveis, que se fragmenta em diversos sistemas operacionais, como pode ser visto na Figura 7, e computadores ou desktops, e as aplicações web possuem restrições às ferramentas disponíveis que poderiam ser utilizadas para enriquecer a experiência do usuário, todavia necessitam de apenas um projeto desenvolvido para atingir todos os usuários. Por outro lado, as aplicações nativas oferecem essas ferramentas com o custo das empresas precisarem de um projeto para cada sistema operacional que desejam atingir

no mercado. Diante desse cenário, a empresa Google desenvolveu o conceito de **PWAs** que são aplicações que unem as vantagens das nativas, ser instalável e acessar recursos do dispositivo como câmera e localização geográfica, e web, um único código para atingir todos os usuários.

Para uma aplicação ser considerada uma **PWA**, ela deve ser (MOZILLA, 2022g; OSMANI, 2015):

- **Progressiva**: deve funcionar no nível mais básico para os usuários que utilizarem navegadores antigos e no mais avançado para navegadores mais novos em relação aos recursos disponíveis;
- **Responsiva**: deve se adaptar a qualquer formato de tela, seja de um desktop ou dispositivo móvel;
- **Independente de conexão**: deve funcionar sem conexão de rede ou com uma conexão ruim;
- **Detectável pelos motores de busca (Search engines)**: deve oferecer metadados para que os motores de buscas encontrem a aplicação através da web;
- **Instalável**: deve permitir que o usuário adicione a aplicação na tela inicial, deixando um ícone clicável;
- **Acessível via link**: deve permitir acessar a aplicação via link sem necessitar de uma instalação, como acontece nas aplicações nativas, facilitando o compartilhamento entre usuários;
- **Engajável**: deve mandar notificações quando atualizações ou conteúdos novos surgirem mesmo quando o usuário não estiver utilizando a aplicação;
- **Segura**: deve garantir que a conexão entre usuário, aplicação e servidor seja segura contra ataque de terceiros.

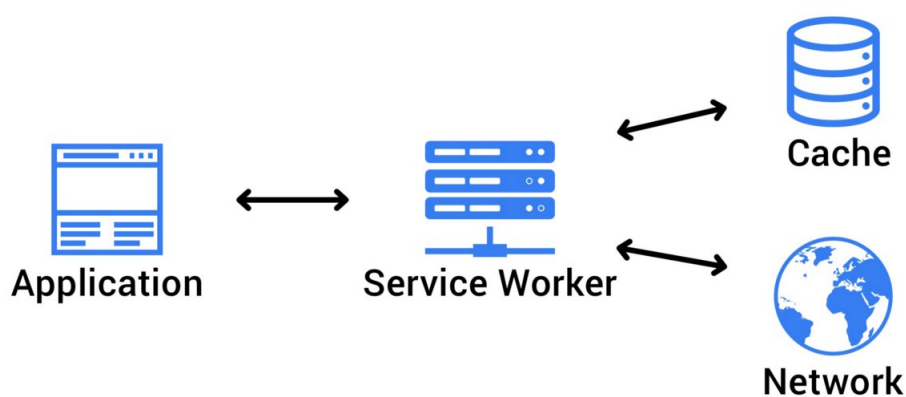
Para garantir essas propriedades, as **PWAs** utilizam os seguintes recursos: Service Workers, Cache API, Manifesto, Web Push API e Notification API. Outras ferramentas que podem ser utilizadas para armazenar dados no cliente são Web Storage API e IndexedDB. Todos esses recursos serão explicados nas seções seguintes.

2.8 SERVICE WORKER

O service worker é um script que o navegador executa em segundo plano, não afetando a interação do usuário com o cliente e possibilitando funcionalidades novas às aplicações web que apenas as aplicações nativas tinham acesso (GAUNT, 2021). A respeito dessas novas funcionalidades, existem as notificações push, uma notificação que é

enviada para os dispositivos para engajar os usuários, sincronização em segundo plano, possibilitando enviar dados quando o usuário retomar a conexão com o servidor (ARCHIBALD, 2019), interceptação das requisições feitas pelo cliente e acesso às APIs da Cache e IndexedDB. A Figura 8 mostra o service worker como um proxy da aplicação, podendo interceptar as requisições do cliente e retornar com dados da cache do navegador ou do próprio servidor.

Figura 8 – Service worker como proxy da aplicação



Fonte: (OLIVEIRA, 2019)

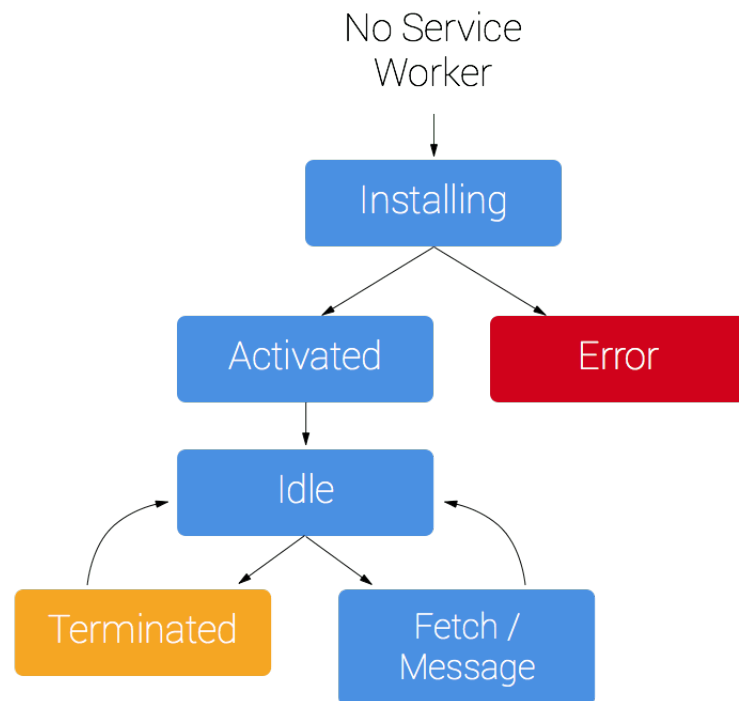
Para ter acesso aos recursos citados anteriormente com um service worker, é necessário atender aos seguintes pré-requisitos: suporte por parte do navegador, já que nem todos os navegadores deram suporte por ser uma tecnologia recente, e o servidor em produção deve utilizar o **HTTPS**, já que é necessário garantir que o service worker não foi adulterado através de um ataque Man-in-the-middle já que a sua capacidade em interceptar requisições e alterar as respostas delas pode causar estragos em mãos erradas (GAUNT, 2021).

2.8.1 Ciclo de vida

Um service worker possui um ciclo de vida que pode ser separado nas seguintes etapas: registro, instalação, ativação, em espera, tratando eventos e encerrado (GAUNT, 2021). A Figura 9 mostra esse ciclo através de um diagrama.

A etapa de registro consiste em registrar o service worker no navegador para iniciar a etapa de instalação. Na fase de instalação, o script do service worker será baixado pelo navegador e depois instalado, é nesse momento que os recursos estáticos são armazenados na cache e caso ocorra uma falha ao baixar esses arquivos estáticos, a instalação falhará.

Figura 9 – Ciclo de vida de um service worker



Fonte: (GAUNT, 2021)

Em seguida, o service worker passará para a etapa de ativação e ela é utilizada para a atualização das caches antigas e, por fim, o service worker ficará entre os estados encerrado, para economizar energia, e tratando eventos (GAUNT, 2021).

2.9 MANIFESTO

O manifesto é um arquivo no formato [JavaScript Object Notation \(JSON\)](#) que contém metadados relacionados à aplicação web para permitir a instalação dela no dispositivo, assim como uma aplicação nativa, sem precisar de uma app store (MOZILLA, 2022; PETE LEPAGE THOMAS STEINER, 2022). Nesse arquivo, podemos encontrar as seguintes informações:

- Nome da aplicação através da chave *name*;
- Ícones da aplicação através da chave *icons*;
- [URL](#) que o usuário acessará ao inicializar a aplicação através da chave *start_url*;
- Forma de disponibilização da aplicação através da chave *display*, podendo ser *full-screen* (tela cheia), *standalone* (assim como uma aplicação nativa, sem barra de pesquisa e navegação), *minimal-ui* (uma aplicação nativa com barra de navegação para atualizar e voltar) e *browser*;

- Cor de fundo enquanto a aplicação é carregada através da chave *background_color*;
- Atalhos para facilitar determinadas atividades do usuário através de um clique com a chave *shortcuts*.


A Figura 10 mostra o manifesto do Instagram, em que é possível ver algumas chaves que foi descritas anteriormente, como os ícones, nome da aplicação, URL inicial, etc.

Figura 10 – Manifesto do Instagram

App Manifest
<https://www.instagram.com/data/manifest.json>

Identity	
name	Instagram
short_name	Instagram

Presentation	
dir	auto
lang	
start_url	https://www.instagram.com/?utm_source=pwa_homescreen
display	standalone
orientation	portrait
theme_color	<input type="radio"/> #FFFFFF
background_color	<input type="radio"/> #FFFFFF
scope	https://www.instagram.com/
id	https://www.instagram.com/?utm_source=pwa_homescreen

Icons	
144x144	
image/png	

Fonte: Própria

2.10 ARMAZENANDO DADOS NO CLIENTE - CACHE API, WEB STORAGE API E INDEXEDDB

A Cache e IndexedDB são APIs para persistência de dados no cliente, sendo usadas geralmente para melhorar o desempenho das aplicações e enriquecer a experiência

do usuário com funcionalidades quando existe problema de conexão (LEPAGE, 2022a). A Cache API é utilizada para armazenar e recuperar as apenas requisições GET e as suas respostas (MOZILLA, 2022c; LEPAGE, 2022a) armazenando em pares na forma *Requisição/Resposta*, e é normalmente usada para guardar recursos estáticos da aplicação. Por outro lado, a IndexedDB é utilizada para guardar outros dados provindos do servidor, sendo um banco de dados NoSQL no modelo chave-valor (MOZILLA, 2022e).

Seguindo o mesmo modelo de armazenamento de chave-valor, a Web Storage API permite que as aplicações armazenem informações no navegador, podendo ser usado para guardar informações sobre o usuário e suas preferências. Existem dois tipos de armazenamento (MOZILLA, 2022a):

- `sessionStorage` mantém as informações armazenadas por origem e permanece disponível enquanto há uma sessão aberta no navegador (mesmo a página sendo recarregada). Caso o browser seja fechado a sessão será limpa e as informações serão perdidas;
- `localStorage` mesmo que o navegador seja fechado, os dados permanecem armazenados.

2.11 NOTIFICAÇÕES NAS APLICAÇÕES WEB

Para compreender o funcionamento das notificações nas aplicações web, é necessário conhecer e entender o fluxo das seguintes Web APIs: Push e Notification. Nas subseções seguintes, serão explicadas cada uma delas com mais detalhes.

2.11.1 Web Push API

Primeiramente, é importante compreender os seguintes termos (THOMSON; DAMAGGIO; RAYMOR, 2016; THOMSON; CACERES; BEVERLOO, 2022):

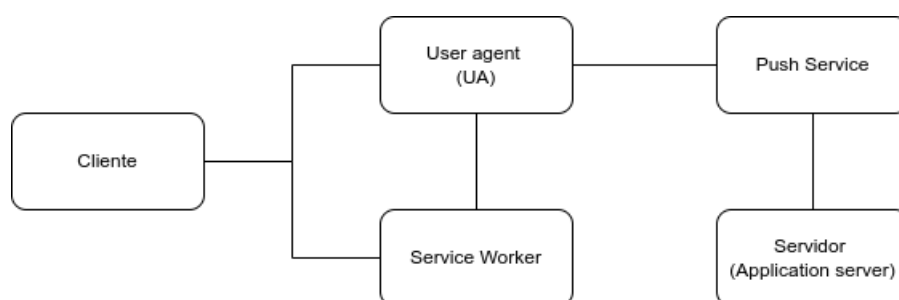
- **Application server**: uma aplicação que deseja enviar push messages para um cliente, que é uma aplicação web;
- **User agent (UA)** ou agente de usuário: um dispositivo e um software que recebe as push messages;
- **Push message**: representa uma mensagem que foi enviada de um application server para um UA para atualizar uma determinada informação ou notificar o usuário para engajá-lo;
- **Push service**: um serviço que garante uma entrega confiável e eficiente ao UA e cria as push message subscriptions;
- **Push message subscription**: um contexto de entrega de mensagens entre um UA e o push service;

- **Voluntary Application Server Identification (VAPID)**: Um método que permite o application server se identificar ao solicitar o envio de notificações para um push service para desenvolver o próprio provedor de push messages (THOMSON; BEVERLOO, 2017). Nesse método, algumas informações presentes são: um par de chave pública e privada e forma de contato com o provedor, como um e-mail.

A Web Push API pode ser descrita como: "A API Push torna possível que aplicações web recebam mensagens enviadas a elas de um servidor, independentemente de aplicação estar ou não em primeiro plano, ou até mesmo carregada, em um agente de usuário. Isso permite que desenvolvedores entreguem notificações e atualizações assíncronas a usuários que optarem por elas, resultando num melhor engajamento com conteúdo novo oportuno."(MOZILLA, 2022b).

A figura 11 mostra todos os atores envolvidos nesse envio das mensagens e seus relacionamentos.

Figura 11 – Atores envolvidos no envio de push messages e seus relacionamentos



Fonte: Própria

Para configurar o servidor e o cliente para enviarem e receberem as push messages respectivamente, recomenda-se seguir o fluxo que está sendo mostrado na figura 12. Esse processo pode ser dividido nas seguintes etapas: registro e ativação do service worker, subscrição do cliente para as push messages e compartilhamento da subscrição com o servidor ou application server. Sobre a primeira etapa, ela não será descrita com mais detalhes porque já foi explicada na seção 2.8.

Na segunda etapa, ocorrerá a subscrição do cliente através do método *subscribe* do *Push Manager* (THOMSON; CACERES; BEVERLOO, 2022). Primeiramente, esse método solicitará a permissão do usuário para enviar notificações caso ainda não tenham sido permitidas e apenas prosseguirá se o usuário der a devida permissão. É recomendado possuir a chave pública da VAPID, também conhecida como *applicationServerKey*, para o push service conseguir filtrar as notificações que vierem para o cliente que está solicitando a subscrição, visto que outros servidores podem enviar notificações sem que o cliente tenha solicitado a inscrição, evitar Denial-of-service attacks (DoSs) no serviço e possuir o mínimo

de informações para entrar em contato, como o e-mail, caso ocorra algum problema o envio das push messages (THOMSON; BEVERLOO, 2017). Em alguns push services, a *applicationServerKey* é obrigatória (THOMSON; CACERES; BEVERLOO, 2022) para realizar a subscrição. Além disso, a API solicitará

Para servidor conseguir enviar as push messages, é necessário enviar todas as informações de uma push message subscription, que são: *endpoint*, chave pública do cliente (*p256dh*) e segredo de autenticação (*auth*). O *endpoint* é uma URL absoluta que deverá ser exposta pelo push service para o application server solicitar o envio de push messages e deve identificar unicamente a subscrição associada (THOMSON; CACERES; BEVERLOO, 2022). As duas últimas informações são utilizadas para a encriptação de dados ou *payload* de uma push message. Todas essas informações devem ser enviadas por um canal de comunicação seguro, como HTTPS, para evitar que aplicações não-autorizadas enviem push messages que poderão ser aceitas pelo UA (THOMSON, 2017). A encriptação é utilizada para evitar que clientes não-autorizados tenham acesso aos dados de uma push message caso ela tenha sido enviada erroneamente (THOMSON, 2017).

Depois de compartilhar os dados da subscrição, um servidor poderá solicitar o envio das push messages para o push service, que chegará ao service worker do cliente como um evento *push*, como pode ser visto na figura 12. Com isso, o service worker precisa adicionar um *event listener* para o evento *push* para chamar a função de *callback*, que poderá mostrar uma notificação, atualizar os dados da cache, etc.

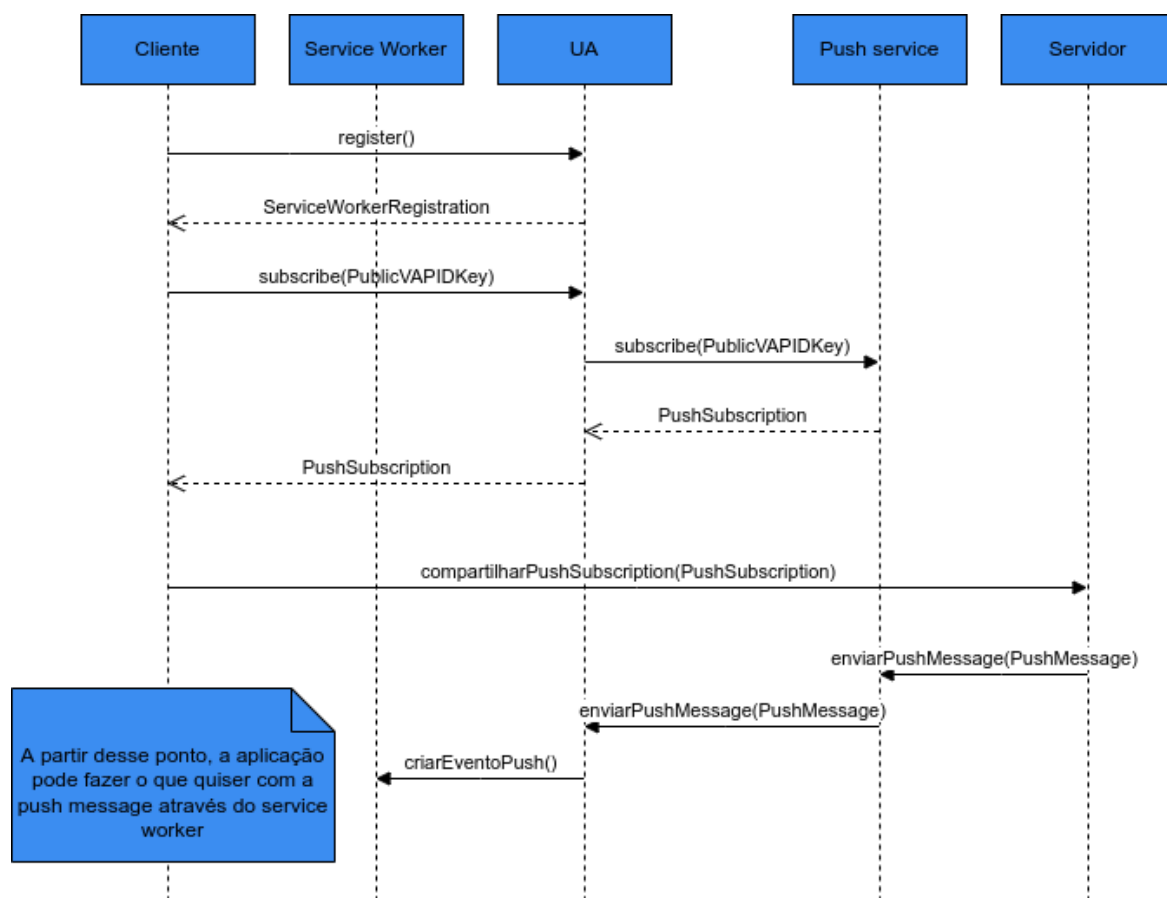
Além disso, essa API permite que o service worker realize uma ação quando a subscrição for atualizada, já que ela pode possuir um tempo de expiração, revogada ou perdida (THOMSON; CACERES; BEVERLOO, 2022). Para isso funcionar, é necessário adicionar um *event listener* ao evento *pushsubscriptionchange*. A função de callback receberá como parâmetro um objeto que implementa a interface *PushSubscriptionChangeEvent*, possuindo a antiga subscrição (*oldSubscription*) e a nova (*newSubscription*). Os dados devem ser compartilhados com o servidor caso seja uma atualização ou removidos caso a subscrição tenha sido revogada ou perdida (THOMSON; CACERES; BEVERLOO, 2022).

2.11.2 Notification API

A Notification API permite que aplicações web mostrem notificações no dispositivo, seja um computador, celular ou tablet, para o usuário final mesmo que a aplicação não esteja na aba ativa ou o navegador não esteja no primeiro plano (MOZILLA, 2022h). Alguns navegadores podem não dar suporte para essa API, logo é recomendado avaliar se existe esse suporte no navegador do usuário.

Além disso, o método *requestPermission* permite que o cliente solicite a permissão para as notificações ao usuário, e recomenda-se fortemente a solicitar essa permissão antes de notificar o usuário, alguns navegadores poderão barrar as notificações caso a permissão

Figura 12 – Diagrama de sequência contendo as etapas para efetuar o envio de push messages para um cliente



Fonte: Própria. Baseado em (THOMSON; CACERES; BEVERLOO, 2022)

não tenha sido concedida (MOZILLA, 2022h). Para criar uma nova notificação, basta utilizar o construtor da *Notification*, que permite os seguintes parâmetros (MOZILLA, 2022h):

- *title*: o título da notificação, ou seja, o texto que será mostrado no topo da notificação. Esse parâmetro é obrigatório;
- *options*: um objeto que permite definir o corpo, ícone, imagem, configurar as vibrações e ações que o usuário pode realizar, entre outros. Para saber mais detalhes sobre as opções, leia [esta documentação](#). Esse parâmetro é opcional.

No caso do service worker, é necessário obter o registro, também conhecido como *ServiceWorkerRegistration* que pode ser acessado através do *self.registration* no script do service worker, e chamar o método *showNotification*, que terá os mesmos parâmetros do construtor da *Notification* (MOZILLA, 2022i). Ao adicionar esse método ao callback do evento *push*, que foi explicado na seção 2.11.1, será possível mostrar as notificações ao

usuário mesmo quando a aplicação não estiver rodando em primeiro plano ou nem estiver aberta.

2.12 FIREBASE

O Firebase é um [Backend-as-a-Service \(BaaS\)](#) desenvolvido pela Google para prover funcionalidades típicas de uma aplicação, como autenticação, autorização, armazenamento de dados, coleta de métricas e serviço de notificações, para facilitar e acelerar o desenvolvimento dos clientes de uma aplicação, já que o [BaaS](#) garante as funcionalidades de um servidor ou boa parte delas ([ANDRADE, 2020](#)). Existem diversas ferramentas que são oferecidas pela instituição, mas, para esse projeto, será utilizada apenas a [Firebase Cloud Messaging \(FCM\)](#), que será descrita na próxima seção.

2.12.1 Firebase Cloud Messaging

Segundo a própria instituição, a [FCM](#) pode ser definida como ([FIREBASE, 2020](#)):

O Firebase Cloud Messaging (FCM) é uma solução de mensagens multiplataforma que permite enviar mensagens de forma confiável e sem custo.

Usando o FCM, você pode notificar um aplicativo cliente de que novos e-mails ou outros dados estão disponíveis para sincronização. Você pode enviar mensagens de notificação para estimular o reengajamento e a retenção do usuário. Para casos de uso como mensagens instantâneas, uma mensagem pode transferir uma carga útil de até 4.000 bytes para um aplicativo cliente.

Com essa ferramenta, é possível desenvolver uma aplicação com notificações e push messages para todo tipo de dispositivo de forma mais simples e sem precisar se preocupar com o Web Push Protocol e Web Push [API](#) e Notification [API](#).

No [FCM](#), temos dois estados para a aplicação: em primeiro plano, quando está com o foco, e em segundo plano, quando está oculta atrás de outras janelas ou fechada e as mensagens podem vir de fontes diferentes dependendo do estado da aplicação. No caso da biblioteca disponibilizada para o JavaScript, as mensagens em primeiro plano serão recebidas pelo método *onMessage*, enquanto que as em segundo plano pelo *onBackgroundMessage*. Além disso, existem dois tipos de mensagem que podem ser enviadas para o cliente ([FIREBASE, 2022b](#)):

- Mensagens de notificação: são tratadas automaticamente pela [Software Development Kit \(SDK\)](#) do [FCM](#), virando notificações no dispositivo do usuário, e possuem valores-chave reservados (título, corpo, ícone, etc) e um payload de dados opcional que permite inserir pares de chave-valor personalizados;
- Mensagens de dados: não são tratadas automaticamente pela [SDK](#), cabendo à aplicação processá-las, e possuem apenas pares de valores-chave personalizados sem nomes de chave reservados;

No caso das mensagens de notificação, há um comportamento diferente dependendo do estado da aplicação. Caso esteja em primeiro plano, a aplicação receberá um objeto contendo os valores-chave reservados e o payload de dados, caso contrário, apenas receberá os valores-chave reservados, sendo que o payload de dados só estará disponível quando o usuário clicar na notificação (FIREBASE, 2022b).

2.13 USABILIDADE E SYSTEM USABILITY SCALE

Conforme a NBR 9241-11 (ABNT, 2002), o termo usabilidade pode ser definido como: "Medida na qual um produto pode ser usado por usuários específicos para alcançar objetivos específicos com eficácia, eficiência e satisfação em um contexto específico de uso.". E os termos eficácia, eficiência e satisfação podem ser definidas como (ABNT, 2002):

7.2 eficácia: Acurácia e completude com as quais usuários alcançam objetivos específicos. 7.3 eficiência: Recursos gastos em relação à acurácia e abrangência com as quais usuários atingem objetivos. 7.4 satisfação: Ausência do desconforto e presença de atitudes positivas para com o uso de um produto.

Para ter essa medida, são utilizados testes de usabilidade que são compostos por tarefas típicas que deverão ser realizadas por usuários representativos, ou seja, não serão pessoas envolvidas no desenvolvimento do próprio sistema, em conjunto com as 10 perguntas do SUS. O SUS foi desenvolvido pelo John Brooke em 1986 e é um método simples e rápido para averiguar a satisfação dos usuários do sistema independente da tecnologia seguindo os três pilares definidos pela NBR 9241-11 (TEIXEIRA, 2015; BROOKE, 1995; SAURO, 2011). Em relação às perguntas, elas seguem um modelo de afirmação, como "Eu acho que gostaria de usar esse sistema com frequência." (TEIXEIRA, 2015), e o usuário deve responder seguindo a Escala Likert, cujos extremos são 1 (discordo fortemente) e 5 (concordo fortemente), para avaliar o nível de concordância. A figura 13 mostra o formato da resposta.

Figura 13 – Formato da resposta para o SUS

Strongly Disagree 1	2	3	4	Strongly Agree 5
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Fonte: (SAURO, 2011)

As perguntas são as seguintes (TEIXEIRA, 2015):

1. Eu acho que gostaria de usar esse sistema com frequência;
2. Eu acho o sistema desnecessariamente complexo;

3. Eu achei o sistema fácil de usar;
4. Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema;
5. Eu acho que as várias funções do sistema estão muito bem integradas;
6. Eu acho que o sistema apresenta muita inconsistência;
7. Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente;
8. Eu achei o sistema atrapalhado de usar;
9. Eu me senti confiante ao usar o sistema;
10. Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.

Depois que os usuários responderem todas as perguntas, é necessário calcular a pontuação final, conhecida como *score*, e ela é calculada da seguinte forma ([TEIXEIRA, 2015](#)):

1. Para as respostas ímpares (1, 3, 5, 7, 9), subtraia 1 da pontuação que o usuário respondeu;
2. Para as respostas pares (2, 4, 6, 8, 10), subtraia a resposta de 5. Ou seja, se o usuário respondeu 2, contabilize 3. Se o usuário respondeu 4, contabilize 1;
3. Some todos os valores das dez perguntas, e multiplique por 2.5. Essa é sua pontuação final, que pode ir de 0 a 100.

Um exemplo do cálculo do score pode ser encontrado na Tabela 2. No caso de ter mais de um usuário, é possível calcular o score para cada usuário e depois pegar o score médio para atribuir a pontuação final da aplicação. A pontuação média é de 68 ([SAURO, 2011](#)) e caso a aplicação não atinja esse valor, ela pode estar passando por problemas sérios de usabilidade ([TEIXEIRA, 2015](#)). A figura 14 mostra um gráfico que associa os scores do SUS às notas na escala de letras, sendo *F* a pior e *A* a melhor.

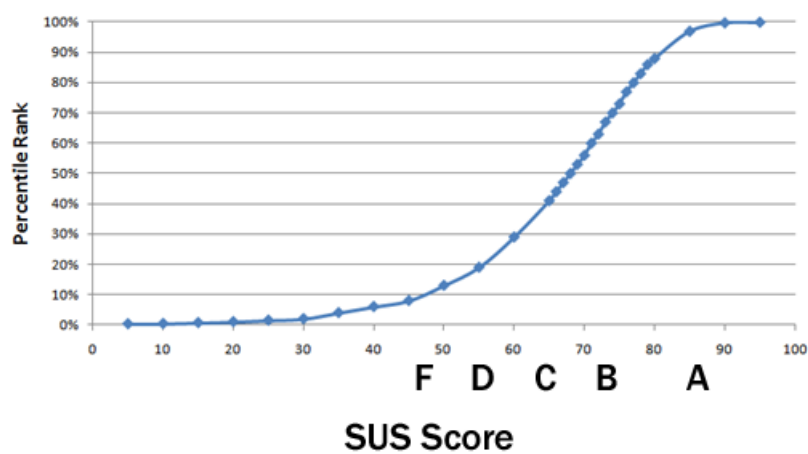
No contexto desse projeto, a avaliação da usabilidade será feita apenas sob o aspecto da satisfação dos usuários em relação aos sistemas e aos requisitos funcionais previamente estabelecidos, como as notificações e visualização da posição e status em uma turma matriculada. Ademais, os outros aspectos como eficácia e eficiência não foram considerados relevantes para essa avaliação visto que ela estava voltada para verificar o impacto da nova abordagem na realização das matrículas assim como a reação aos feedbacks que o sistema atual não fornece.

Perguntas	Resposta do usuário 1	Cálculo para o score
Pergunta 1	5	$5 - 1 = 4$
Pergunta 2	1	$5 - 1 = 4$
Pergunta 3	4	$4 - 1 = 3$
Pergunta 4	2	$5 - 2 = 3$
Pergunta 5	5	$5 - 1 = 4$
Pergunta 6	1	$5 - 1 = 4$
Pergunta 7	3	$3 - 1 = 2$
Pergunta 8	2	$5 - 2 = 3$
Pergunta 9	5	$5 - 1 = 4$
Pergunta 10	3	$5 - 3 = 2$
TOTAL		33
SCORE		$33 * 2,5 = 82,5$

Tabela 2 – Exemplo de cálculo do score para 1 usuário

Fonte: (BARBOZA, 2019)

Figura 14 – Gráfico que associa os scores do SUS às notas na escala de letras



Fonte: (SAURO, 2011)

3 TRABALHOS RELACIONADOS

Nesse capítulo são mostrados trabalhos relacionados que realizam a mesma função que será feita por esse projeto.

3.1 MATRUFSC E MATRUFSC2

Os sites [MatrUFSC](#) e [MatrUFSC2](#), que é uma evolução do MatrUFSC, são aplicações web que foram desenvolvidas por alunos do Curso de Ciência da Computação da UFSC para permitir que os alunos planejem diversas grades de horário para o próximo semestre ([MATRUFSC2, 2022b](#)), uma vez que o módulo de matrícula oficial apenas fica disponível durante a etapa de matrícula. Essas aplicações permitem que o aluno veja todos os horários disponíveis, quantidade de vagas livres e ocupadas, e os professores para uma determinada matéria no semestre especificado. Além disso, há um campo de busca que permite pesquisar as disciplinas por nome e código, permite filtrar os dados por semestre e campus e há uma identificação de cada turma através das cores para facilitar a visualização da grade de horários. Apesar disso, elas não possuem uma integração com o módulo oficial. As Figuras 15 e 16 mostram as interfaces para o MatrUFSC e MatrUFSC2 respectivamente.

Figura 15 – Construção da grade de horários no MatrUFSC

The screenshot shows the MatrUFSC web application interface. At the top, there are navigation links and a 'Sobre' link. Below that, there are dropdown menus for 'campus: Florianópolis', '2022-1', and 'Plano 1'. There are also buttons for 'abrir', 'salvar', and a search field. The main content area is a table with columns for 'Código', 'Turma', 'Período', and 'Combinções'. Below this is a grid for selecting classes, with columns for 'Segunda', 'Terça', 'Quarta', 'Quinta', 'Sexta', and 'Sábado'. To the right of the grid is a list of available classes with columns for 'Turma', 'Vagas Ocupadas', and 'Professores'. At the bottom, there is a footer with a disclaimer and a 'Como usar' link.

Código	Turma	Período	Combinções	Créditos por semana	
<input checked="" type="checkbox"/>	INE5423	06208	2022-1	Banco de Dados I *CIÊNCIAS DA COMPUTAÇÃO	12
<input checked="" type="checkbox"/>	INE5414	04208	2022-1	Redes de Computadores I *CIÊNCIAS DA COMPUTAÇÃO	12
<input checked="" type="checkbox"/>	MTM3121	02205	2022-1	Álgebra Linear *ENGENHARIA CIVIL	12

Horário	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
07:30						
08:20						
09:10						
10:10	INE5414	MTM3121	INE5414	MTM3121		
11:00	INE5414	MTM3121	INE5414	MTM3121		
13:30		INE5423		INE5423		
14:20		INE5423		INE5423		
15:10						
16:20						
17:10						
18:30						
19:20						
20:20						
21:10						

Turma	Vagas Ocupadas	Professores
<input checked="" type="checkbox"/> 02201	(0)/60	Marcio Rodolfo Fernandes
<input checked="" type="checkbox"/> 02203	(0)/60	
<input checked="" type="checkbox"/> 02230	(0)/45	
<input checked="" type="checkbox"/> 02205	(0)/30	
<input checked="" type="checkbox"/> 02212	(0)/60	Professor Não Informado
<input checked="" type="checkbox"/> 02215	(0)/45	
<input checked="" type="checkbox"/> 02216	(0)/45	
<input checked="" type="checkbox"/> 02220	(0)/45	
<input checked="" type="checkbox"/> 02236	(0)/45	
<input checked="" type="checkbox"/> 03225	(0)/40	
<input checked="" type="checkbox"/> 03226	(0)/30	
<input checked="" type="checkbox"/> 03227	(0)/20	
<input checked="" type="checkbox"/> agrupar turmas com horários iguais		

Não se esqueça de fazer sua matrícula no [CAGR!](#) Este aplicativo não tem nenhum vínculo com a UFSC. [Como usar](#)
 github.com/pet-comp-ufsc/capiim. versão 2.5.13. banco de dados atualizado em 06/03/22 - 12:00

Fonte: Própria

Figura 16 – Construção da grade de horários no MatrUFSC2

Semestre: Campus: Disciplina:

Código	Turma	Horas-Aula	Período	<	1/1	>
<input checked="" type="checkbox"/> INE5429 - Seguranca em Computacao *CIENCIAS DA COMPUTACAO	07208	4	2018-1			
<input checked="" type="checkbox"/> INE5414 - Redes de Computadores I *CIENCIAS DA COMPUTACAO	04208	4	2018-1			
<input checked="" type="checkbox"/> INE5422 - Redes de Computadores II *CIENCIAS DA COMPUTACAO	05208	4	2018-1			
<input checked="" type="checkbox"/> MTM3421 - Algebra Linear I *MATEMATICA - Bacharelado	03223	6	2018-1			
Total:		18				

	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
07:30	MTM3421		MTM3421			
08:20	MTM3421		MTM3421			
09:10	MTM3421		MTM3421			
10:10	INE5414		INE5414			
11:00	INE5414		INE5414			
13:30			INE5422		INE5422	
14:20			INE5422		INE5422	
15:10						
16:20		INE5429		INE5429		
17:10		INE5429		INE5429		
18:30						
19:20						

	Turma	Vagas	Professores
<input type="checkbox"/>	03222	0/60	
<input checked="" type="checkbox"/>	03223	0/60	

Agrupar turmas com horários iguais

[Gerenciador de combinações](#)

Fonte: Própria

3.2 SISTEMA DE CONTROLE ACADÊMICO DA GRADUAÇÃO

O **CAGR** é uma aplicação web contendo diversas funcionalidades importantes para os alunos da **UFSC**, como a realização da matrícula. A Figura 17 mostra a interface para essa funcionalidade. A partir dessa figura, torna-se evidente que essa interface está defasada e apresenta alguns problemas:

- Na aba de **Disciplinas Solicitadas**, os valores colocados não possuem uma estrutura clara para o usuário entender do que eles se tratam;
- Na aba de turmas, que está acima da **Disciplinas Solicitadas**, também não possui uma estrutura clara para que o usuário entenda os dados que são passados após dar um scroll;
- A partir dessa interface, não é possível saber qual o professor de cada turma, sendo necessário buscar no módulo de **Cadastro de Turmas** do **CAGR**, cuja interface

pode ser vista na Figura 18.

Figura 17 – Construção da grade de horários no CAGR

Semestre: **Etapa: 1**
 Matricula:
 Curso:
 Local:
 Plano 1: Plano 2: Plano 3:

Data:
 Situação:
 Currículo:

Ano e Semestre da provável Formatura

	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
07:30						
08:20		INES410		INES410		
09:10		INES410		INES410		
10:10		INES411		INES411	INES411	
11:00		INES411		INES411	INES411	
13:30						
14:20						
15:10		INES408		INES408		
16:20		INES408		INES408		
17:10		INES408		INES408		
18:30						
19:20						
20:20						
21:10						

Disciplinas Solicitadas

INES408 03208A 6
 INES410 03208B 4
 INES411 03208B 6

16 Aulas (Min/Max 12/28)

Fonte: (MATRUFSC2, 2022a)

Figura 18 – Busca de turmas no CAGR

Cadastro de Turmas

Semestre: 2021
 Departamento: TODOS
 Curso: EPSCIPLO | CIÊNCIAS DA COMPUTAÇÃO
 Disciplina: Organização e Arquitetura de Computadores
 Professor:
 Horários: TODOS OS DIAS | TODOS OS HORÁRIOS

2 resultados foram encontrados

Ementa	Currículo	Disciplina	Turma	Nome da Disciplina	Horas Aula	Vagas Oportunidades	Vagas Ocupadas	Alunos Especiais	Saldo Vagas	Pedidos sem vaga	Horários	Professor
1	INES607	02238A	02238A	Organização e Arquitetura de Computadores *SISTEMAS DE INFORMAÇÃO (noturno)	72	33	0	0	33		4.1830-2 / CTC-CTC210 5.1830-2 / CTC-CTC210	Cristina Menhardt
2	INES607	02238B	02238B	Organização e Arquitetura de Computadores *SISTEMAS DE INFORMAÇÃO (noturno)	72	30	0	0	30		4.1830-2 / CTC-CTC305 5.1830-2 / CTC-CTC305	Mateus Grelier da Silva

Fonte: Própria

3.3 UNIVERSIDADE DE BRASÍLIA

Em (RODRIGUES, 2020), foi proposta uma nova interface responsiva para o Sistema Integrado de Gestão de Atividades Acadêmicas (SIGAA) da Universidade de Brasília (UnB), uma aplicação usada por instituições de ensino superior para gerenciar as atividades referentes à vida acadêmica dos discentes, devido aos problemas de usabilidade encontrados após a realização de uma pesquisa virtual com os alunos. Dentre os problemas citados, estão a falta de responsividade para aplicativos móveis, informações ilegíveis

sobre os horários das turmas (tanto que os alunos desenvolveram uma extensão no navegador para converter) e uma interface extremamente ultrapassada. Além disso, os alunos comentaram que apenas usam o [SIGAA](#) porque são obrigados, senão nem usariam. Em vista disso, é possível observar que esses mesmos problemas são encontrados pelo sistema de matrícula presente no [CAGR](#) da [UFSC](#), sendo evidenciados pela criação de ferramentas como o [MatrUFSC](#) e [MatrUFSC 2](#) e manuais e tutoriais para os alunos ([SECRETARIA INTEGRADA DE GRADUAÇÃO \(SIG\) DO CENTRO DE CIÊNCIAS, 2022](#); [CURSO DE CIÊNCIAS DA COMPUTAÇÃO DA UFSC, 2022](#); [CURSO DE GRADUAÇÃO EM LETRAS – LÍNGUA PORTUGUESA E LITERATURAS DE LÍNGUA PORTUGUESA DA UFSC, 2022](#); [CURSOS DE GRADUAÇÃO EM EDUCAÇÃO FÍSICA DA UFSC, 2022](#)). Nessa interface, foram desenvolvidas mais funcionalidades do que as foram propostas por esse projeto como uma agenda para anotar as atividades, aba de links úteis para a biblioteca central e o restaurante universitário, acesso ao cronograma e ementa de cada disciplina do semestre cursado, entre outras. Em relação às funcionalidades semelhantes, como a realização do pedido de matrícula e visualização da grade de horários, as figuras [19](#) e [20](#) mostram as telas propostas.

3.4 ANÁLISE COMPARATIVA

Em relação ao [MatrUFSC](#) e [MatrUFSC 2](#), por serem aplicações não-oficiais, elas não permitem que o aluno envie a grade de horário que foi feita para o módulo oficial de matrícula da [UFSC](#), logo o usuário precisará copiar manualmente a grade que foi feita nessas aplicações para a oficial, causando um retrabalho que poderia ser evitado. Além disso, elas não possuem funcionalidades no caso de perda de conexão e notificações que melhorariam a experiência do usuário, como por exemplo: notificar ao iniciar a etapa de matrícula para lembrar o usuário de realizar os planejamentos das grades e fazer a matrícula no site oficial ou permitir o acesso às grades mesmo sem conexão com o servidor. Diante disso, esse projeto se diferencia porque teria acesso à [API](#) do módulo de matrícula oficial, se o atraso não tivesse ocorrido, permitindo que o aluno salvasse as grades sem precisar passar manualmente depois e por utilizar os recursos que uma [PWA](#), como as notificações, avisando o aluno sobre o início da etapa de matrícula, instabilidade, permitindo que o usuário instale no computador ou no celular e use como se fosse um aplicativo, além do acesso das grades independente do estado de conexão. Todavia, não foi possível inserir opções de filtragem dos dados e identificação das turmas solicitadas através de cores para facilitar a visualização na grade de horários.

Em relação ao módulo no [CAGR](#), essa aplicação não contém funcionalidades no caso de perda de conexão, notificações e instabilidade assim como os [MatrUFSC](#) e [MatrUFSC 2](#), possui uma interface tecnologicamente e esteticamente defasada e não mostra informações como o professor responsável por ministrar. Diante desse contexto, esse projeto desenvolveu uma interface mais intuitiva e com tecnologias mais atuais no

Figura 19 – Matrícula

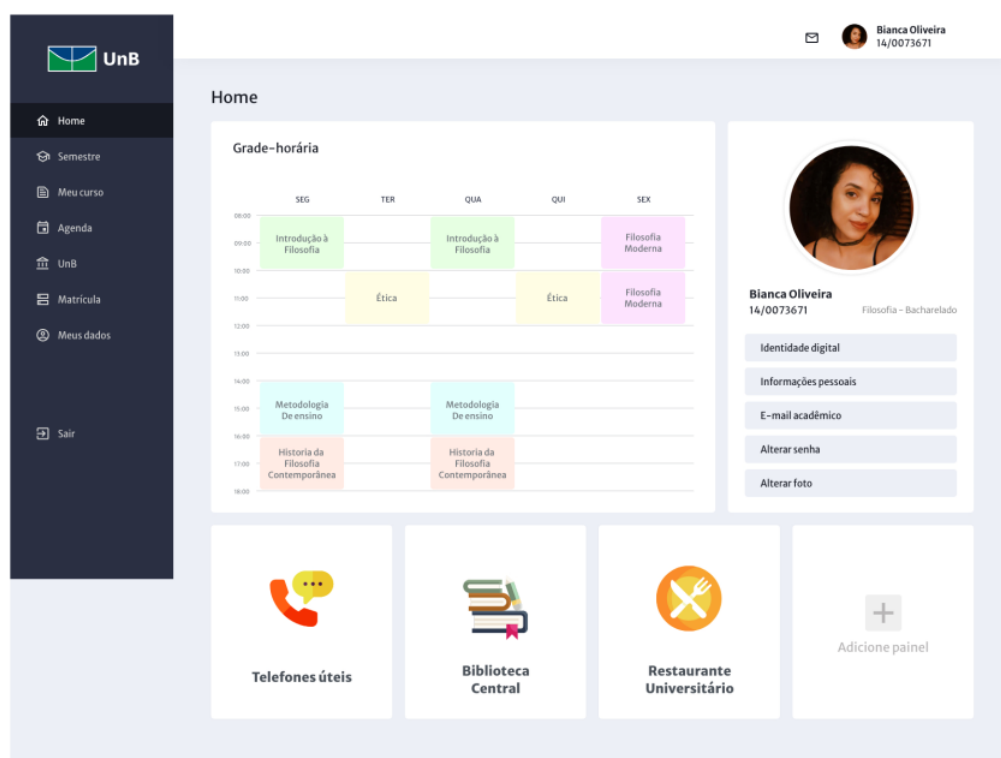


Fonte: (RODRIGUES, 2020)

mercado, que mostra mais informações para auxiliar na tomada de decisão do aluno e utiliza os recursos associados a uma PWA como as notificações para atualizar um aluno sobre a entrada ou saída da fila de espera de uma turma que estava esperando para se matricular, permitindo tomar uma decisão imediatamente e ver o espelho de matrícula e grade de horários estando sem conexão com a rede.

Por fim, em relação ao trabalho proposto em (RODRIGUES, 2020), esse projeto se

Figura 20 – Grade de horários



Fonte: (RODRIGUES, 2020)

distingue pelo fato de disponibilizar os recursos das aplicações nativas, como as notificações e instalação, para os usuários que utilizam os navegadores também e não obrigar os alunos a instalarem o sistema, dando a liberdade de escolha.

4 DESENVOLVIMENTO

4.1 PROVA DE CONCEITO

Durante esse semestre, foi desenvolvida uma [PoC](#) para colocar em prática os conceitos estudados sobre a [PWA](#) e suas funcionalidades. A aplicação representa uma rede social para ver, compartilhar e salvar exercícios físicos e as suas funcionalidades são:

1. O usuário poderá ver os exercícios que foram compartilhados por outros usuário;
2. O usuário poderá seguir ou deixar de seguir outro usuário;
3. O usuário poderá salvar os exercícios favoritos e acessá-los independente do estado de conexão com o servidor;
4. O usuário poderá compartilhar novos exercícios;
5. Se um usuário compartilhar um novo exercício, todos os usuários que o seguem devem receber uma notificação sobre esse novo exercício.

A figura 21 mostra as funcionalidades descritas através de uma diagrama de casos de uso. Além disso, não foi realizada uma análise de requisitos e nem utilizado uma metodologia de desenvolvimento de software porque o foco da prova de conceito era o ganho de experiência e aprendizado, logo, era importante simplificar o processo ao máximo.

4.1.1 Implementação

O código da implementação se encontra no [repositório do GitHub](#), contendo as informações necessárias para rodar a aplicação corretamente. Caso ocorra algum problema nas variáveis de ambiente, [baixe estes arquivos](#) para usá-los na hora de configurá-las.

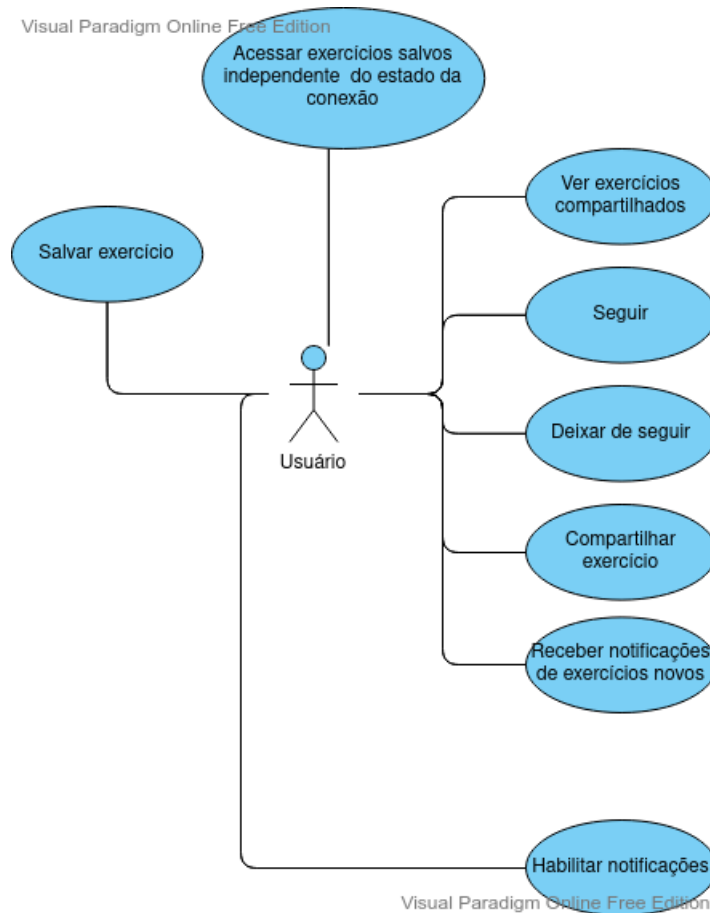
Em relação às tecnologias e bibliotecas empregadas, foi utilizado o [Bold Design System](#) e o React para o cliente e Express.js para o servidor devido à familiaridade da acadêmica com as tecnologias e simplificação no desenvolvimento, respectivamente. Para persistir as informações, foi utilizado o [Sistema de gerenciamento de banco de dados \(SGBD\)](#) PostgreSQL com um banco de dados relacional. A Figura 22 mostra a modelagem conceitual desse banco.

Nas subseções seguintes, serão apresentados pseudocódigos para as regras de negócio desenvolvidas na ordem que foram listadas.

4.1.1.1 Vendo exercícios de outros usuários

Para garantir que a visualização dos exercícios aconteça independente do estado da conexão com o servidor, a IndexedDB foi utilizada para persistir os dados. O cliente sempre fará uma requisição ao servidor para solicitar os exercícios mais recentes e caso a

Figura 21 – Diagrama de casos de uso da PoC



Fonte: Própria

requisição seja um sucesso, mostrará os exercícios na tela e salvará os dados na IndexedDB, caso contrário, mostrará os exercícios salvos da IndexedDB.

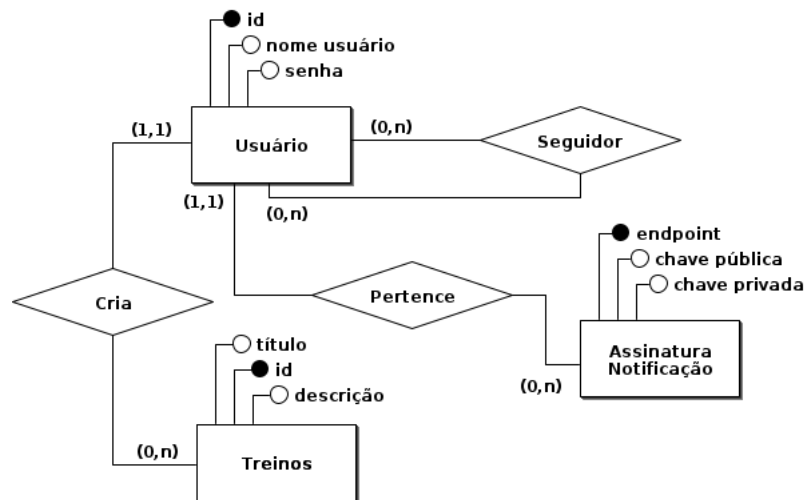
O algoritmo 1 mostra o passo a passo para desenvolver essa funcionalidade. Não foi criado um algoritmo para a parte do servidor porque consiste apenas pegar todos os exercícios que não pertencem ao usuário logado.

Algorithm 1: Vendo exercícios - cliente

```

Data:  $id_{usuário}$ 
exercícios = requisiçãoExercícios( $id_{usuário}$ );
if a requisição for um sucesso then
  | mostrarExercícios(exercícios);
  | indexedDB.salvarExercícios(exercícios);
else
  | exercíciosindexedDB = indexedDB.pegarExercíciosSalvos();
  | mostrarExercícios(exercíciosindexedDB);
end
  
```

Figura 22 – Modelagem conceitual do banco de dados



Fonte: Própria

4.1.1.2 Seguindo um usuário e parando de seguir

Ao seguir um usuário, o cliente primeiro envia os identificadores dos usuários envolvidos ao servidor para salvar no banco de dados. Em seguida, é necessário verificar se o usuário já concedeu ou concederá permissão para utilizar as notificações e se essa permissão for garantida, o cliente fará uma inscrição no serviço de notificações, que é oferecido pelo próprio navegador, e enviará a subscrição para o servidor, que guardará para usar na regra 5.

O algoritmo 2 mostra o passo a passo para desenvolver essa funcionalidade. Não foi criado um algoritmo para a parte do servidor porque consiste apenas em receber os dados através de uma requisição e salvá-los no banco de dados.

4.1.1.3 Salvando treinos

Assim como na regra 1, foi utilizado a IndexedDB para salvar os exercícios que forem escolhidos pelo usuário. Nenhum algoritmo foi criado devido à simplicidade, já que consiste em apenas salvar o exercício que o usuário escolheu e trazer todos os exercícios que foram salvos da IndexedDB.

4.1.1.4 Compartilhar novos exercícios

Ao criar um novo exercício, o cliente enviará os dados necessários para o servidor que salvará no banco de dados. Nenhum algoritmo foi desenvolvido para essa funcionalidade

Algorithm 2: Seguindo um usuário - cliente

```

Data:  $id_{seguidor}$  e  $id_{seguido}$ 
Function VerificarPermissoes( $id_{seguidor}$ ):
  if navegador possui suporte para notificações then
    if usuário não tomou uma decisão sobre permitir as notificações then
      if solicitarPermissão() retornou que foi garantida then
        | EnviarSubscricao( $id_{seguidor}$ );
      end
    else
      if usuário já permitiu permissão para notificações then
        | EnviarSubscricao( $id_{seguidor}$ );
      end
    end
  end
Function EnviarSubscricao( $id_{seguidor}$ ):
  if service worker está ativo then
    |  $subscrição_{push}$  = inscreverParaServiçoPush();
    | Enviar para o servidor a  $subscrição_{push}$  e  $id_{seguidor}$  para salvar no banco;
  end
Function SeguirUsuario( $id_{seguidor}$ ,  $id_{seguido}$ ):
  | Enviar para o servidor os dados  $id_{seguidor}$  e  $id_{seguido}$  para salvar no banco;
  | VerificarPermissoes( $id_{seguidor}$ );

```

devido à simplicidade, uma vez que o cliente apenas precisará pegar os dados de um formulário e enviar esses dados, enquanto que o servidor deve armazenar essas dados no banco.

4.1.1.5 Mandando notificações quando for criado um exercício

Após a execução dos passos da regra 4, o servidor deverá buscar todos os seguidores do usuário que criou o exercício e os dados para realizar as notificações de cada seguidor e, por fim, enviar as notificações com os dados do novo exercício. O cliente de cada seguidor tratará a notificação recebida pelo servidor e exibirá uma notificação na tela para o usuário de que um novo exercício foi criado.

O algoritmo 3 mostra o passo a passo para desenvolver essa funcionalidade. Não foi criado um algoritmo para a parte do cliente porque consiste apenas em extrair os dados do evento de notificação e mostrar uma notificação na tela do usuário.

4.2 PWA DO MÓDULO DE MATRÍCULA

Como descrito anteriormente, a aplicação foi desenvolvida para disponibilizar as seguintes funcionalidades aos alunos:

- Submissão e edição do pedido de matrícula;

Algorithm 3: Enviando notificações sobre um exercício novo - servidor

```

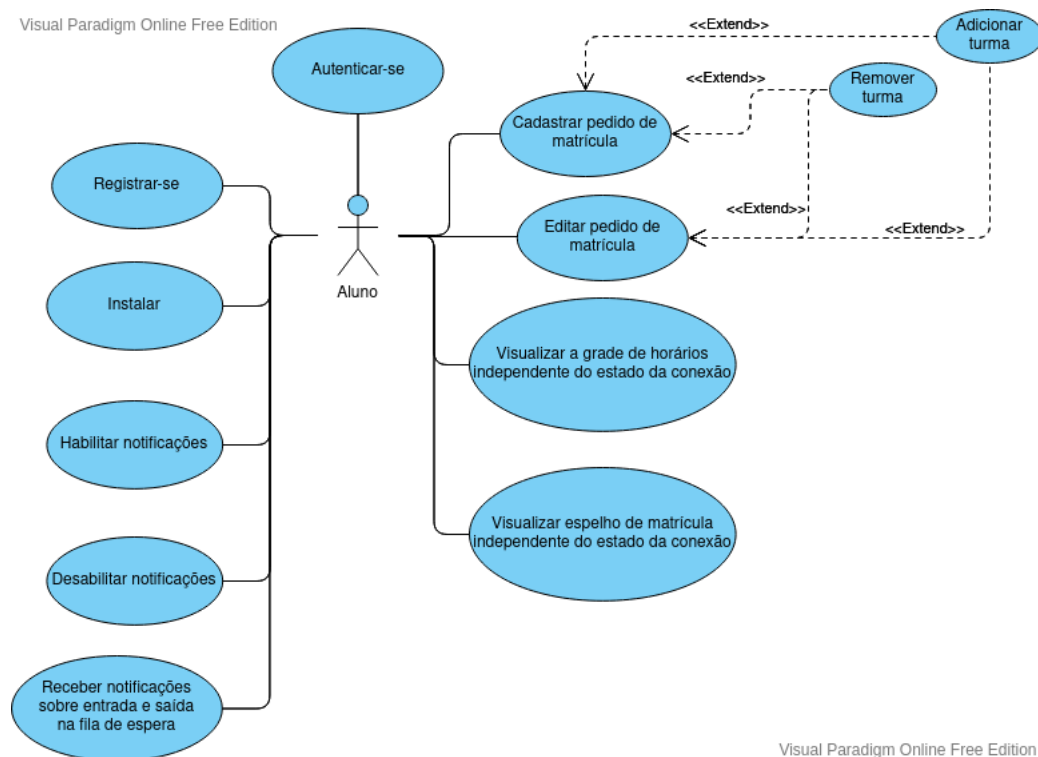
Data:  $id_{criador}$ ,  $título_{exercício}$ 
 $ids_{seguidores} = buscarIdsSeguidoresPeloId(id_{criador});$ 
 $nome_{criador} = buscarNomePeloId(id_{criador});$ 
foreach  $id_{seguidor} \in ids_{seguidores}$  do
  |  $dados_{notificação} = buscarDadosNotificação(id_{seguidor});$ 
  |  $enviarNotificação(dados_{notificação}, título_{exercício}, nome_{criador})$ 
end

```

- Visualização da grade de horários e do status da vaga em cada turma solicitada, se está fora (vaga possível) ou dentro de uma fila de espera e a posição do aluno, independente do estado de conexão do dispositivo;
- Possibilidade de instalação no computador, celular ou tablet e outros dispositivos;
- Envio de notificações caso o aluno entre ou saía de uma fila de espera, se for concedida a permissão.

Essas funcionalidades foram baseadas no documento de requisitos presentes no anexo A. A figura 23 mostra as funcionalidades descritas através de uma diagrama de casos de uso.

Figura 23 – Diagrama de casos de uso da PWA

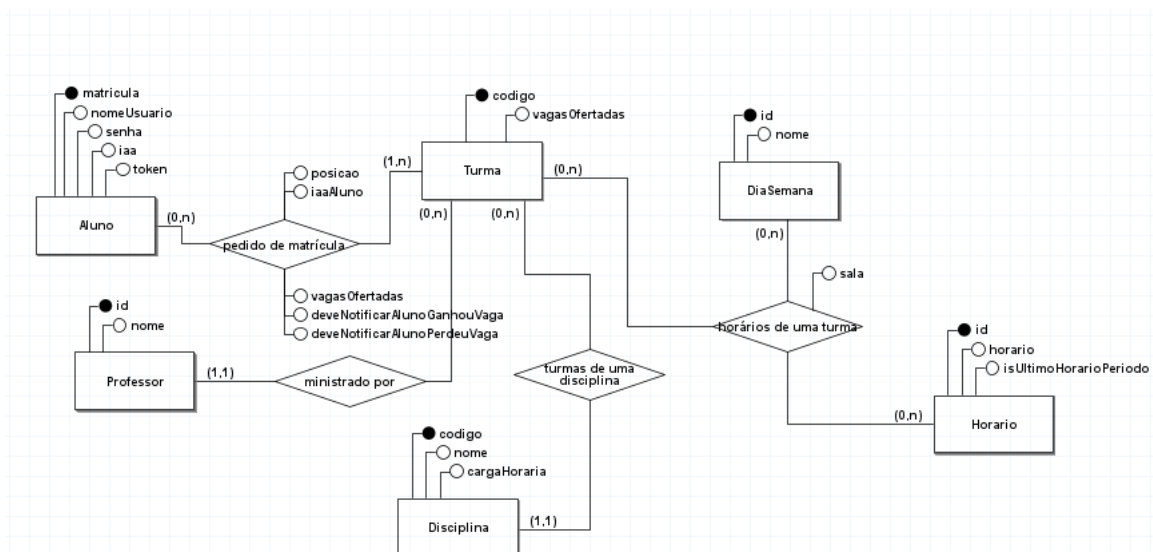


Fonte: Própria. Baseado nos requisitos presentes no anexo A

Como ocorreram atrasos no desenvolvimento do servidor do módulo de matrícula pela [SeTIC](#), um servidor com escopo extremamente reduzido foi implementado para permitir a realização do teste de usabilidade. Nesse servidor, não há preocupação com questões de segurança para evitar mais aumento de escopo no projeto e atrasos no desenvolvimento da aplicação proposta. Além disso, a posição de um aluno em uma turma é unicamente determinado pelo [Índice de Aproveitamento Semestral Acumulado \(IAA\)](#) para manter o servidor o mais simples possível, mas funcional para a realização do teste de usabilidade. Com esse atraso, as seguintes funcionalidades complementares foram necessárias: registro, autenticação e autorização de um aluno no servidor. Os requisitos que foram usados como base dessa aplicação podem ser vistos com mais detalhes no anexo [A](#). Vale ressaltar que nem todos os requisitos descritos nesse documento foram contemplados pela autora devido ao tempo disponível para desenvolver e o aumento de escopo causado pelo atraso no servidor.

Para o desenvolvimento do servidor, foi utilizado o framework Spring Boot, Java 8 e Kotlin. Já para o cliente, foi utilizado o React e o Bold Design System ([BRIDGE, 2022](#)), desenvolvido pelo Laboratório Bridge. Para persistir as informações, foi utilizado o [SGBD PostgreSQL](#) com um banco de dados relacional. A figura [24](#) mostra a modelagem conceitual desse banco, sendo que algumas colunas foram replicadas nos relacionamentos para evitar a realização de joins nas stored procedures, que serão mostradas na subseção [4.2.4](#), do banco. Todas essas escolhas foram feitas devido à familiaridade da acadêmica com essas tecnologias e ferramentas. Para a criação inicial do projeto do cliente e servidor, foi utilizada a ferramenta [create-react-app para PWA](#) e o [Spring Initializr](#) respectivamente.

Figura 24 – Modelagem conceitual do banco de dados



Fonte: Própria

Todo o código desenvolvido para o cliente e servidor pode ser encontrado nos

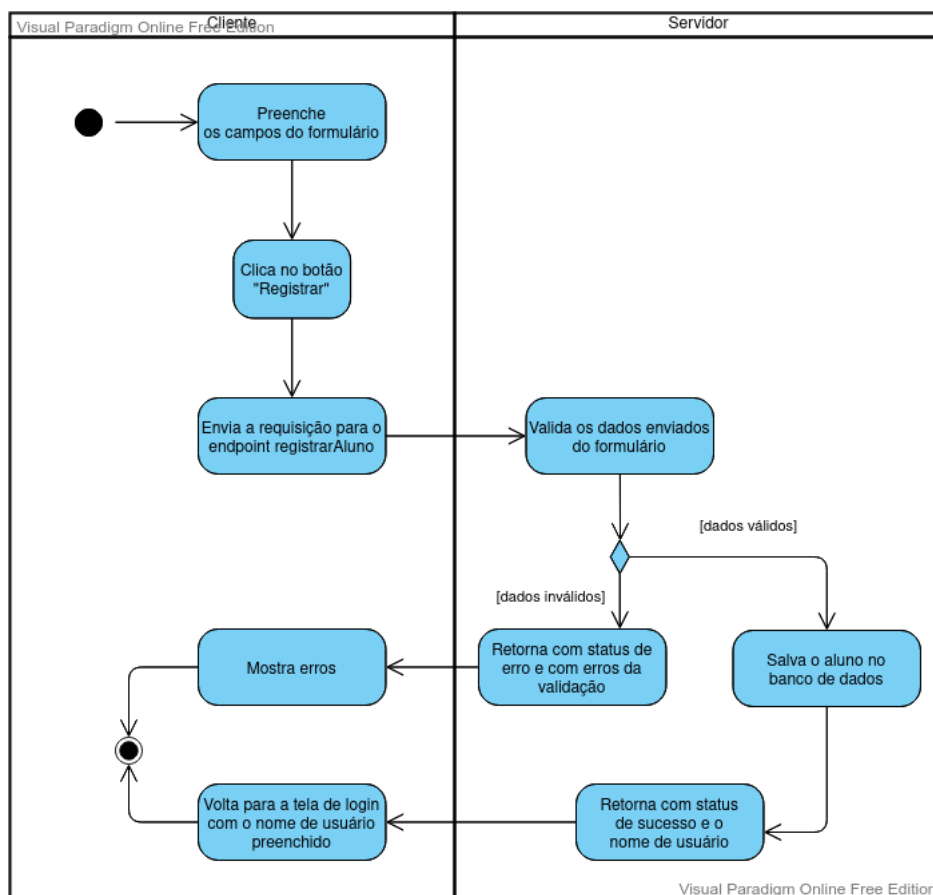
seguintes repositórios do GitHub: [cliente](#) e [servidor](#). Essa separação foi necessária para distribuir as duas aplicações na plataforma do Heroku.

Nas próximas subseções, cada funcionalidade será descrita em detalhes para explicar como ocorreu o desenvolvimento da mesma tanto no servidor quanto no cliente.

4.2.1 Registrando um aluno

Nessa funcionalidade, foi necessário implementar tanto no servidor quanto no cliente. O fluxo dessa funcionalidade pode ser visto na figura 25 através de um diagrama de atividades.

Figura 25 – Fluxo ao registrar um aluno



Fonte: Própria


Os detalhes de implementação no cliente e servidor serão explicados nas seções seguintes.

4.2.1.1 Cliente

No cliente, foi necessário desenvolver uma tela, que pode ser vista na figura 26, que contém um formulário para preencher as informações necessárias que são o nome de

usuário e a senha. Existem dois botões nessa tela, um para voltar à tela de login com o título "Voltar", e o outro para submeter os dados ao servidor com o título "Registrar". Ao clicar no botão de registrar, uma requisição POST para o endpoint `/registrarAluno` é enviada ao servidor com o corpo contendo as informações preenchidas no formato **JSON**.

Figura 26 – Tela para registrar o aluno



Matrícula UFSC

← Voltar

Registrando um aluno

Usuário *
Digite o seu usuário

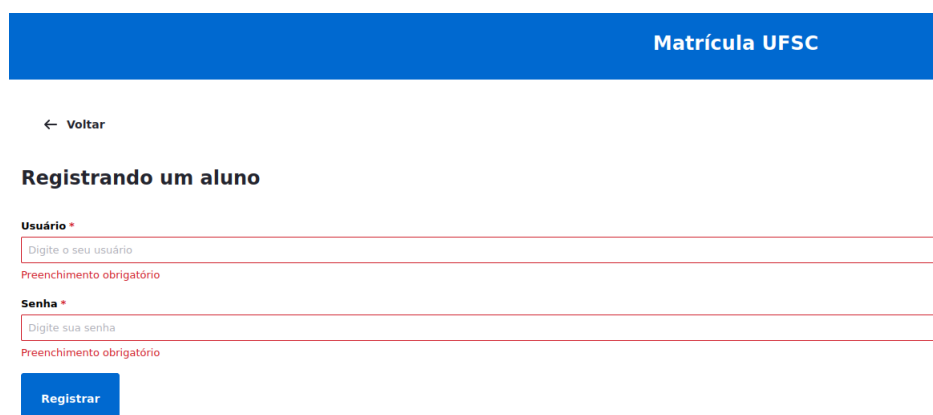
Senha *
Digite sua senha

Registrar

Fonte: Própria

Se não houver nenhum erro de validação por parte do servidor, o usuário será redirecionado para a tela de login com o campo do nome do usuário já preenchido. Caso contrário, os erros serão mostrados nos campos como mostra a figura 27.

Figura 27 – Tela para registrar o aluno com erros de validação retornados pelo servidor



Matrícula UFSC

← Voltar

Registrando um aluno

Usuário *
Digite o seu usuário
Preenchimento obrigatório

Senha *
Digite sua senha
Preenchimento obrigatório

Registrar

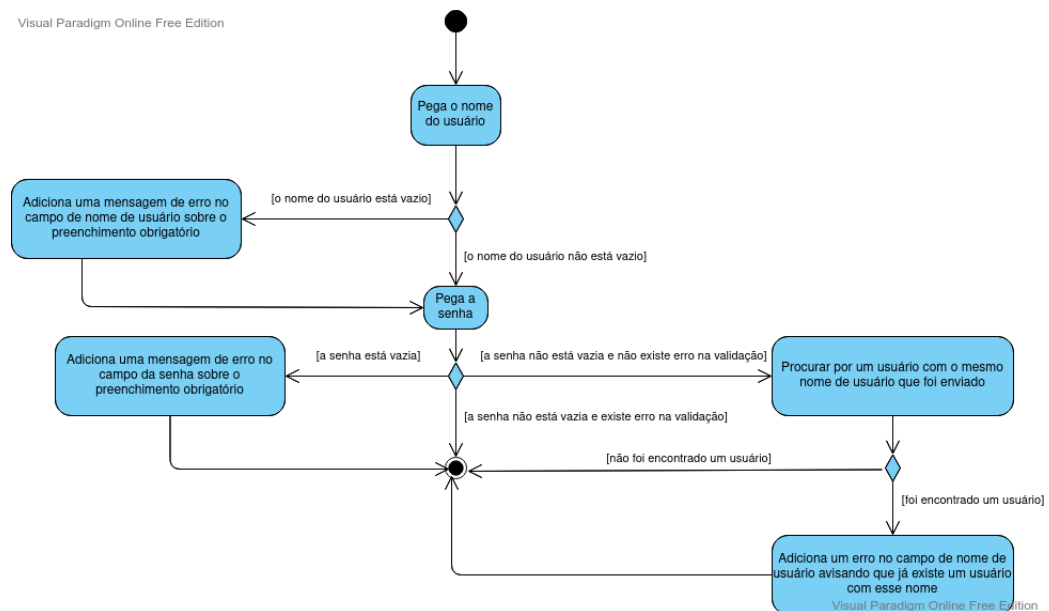
Fonte: Própria

4.2.1.2 Servidor

No servidor, foi necessário criar um controller, *AlunoController*, para receber as requisições e criar o método *registrarAluno* para o endpoint */registrarAluno*.

Antes de adicionar um registro da tabela de alunos, uma validação é feita nos dados recebidos para garantir que a senha e o nome do usuário não estejam vazios e que o nome de usuário não esteja sendo utilizado por outro aluno pela classe *RegistrarAlunoInputValidator*. Caso ocorra algum erro na validação, o servidor retornará os erros e o código de erro 400 (*Bad Request*) ao cliente. Caso contrário, um novo registro na tabela de alunos será adicionado, sendo que a senha passará por uma função hash para não guardá-la como texto simples e o *IAA* será um número inteiro aleatório entre 0 e 9 gerado pelo servidor. A figura 28 mostra todo o processo de validação através de um diagrama.

Figura 28 – Validação ao registrar um aluno



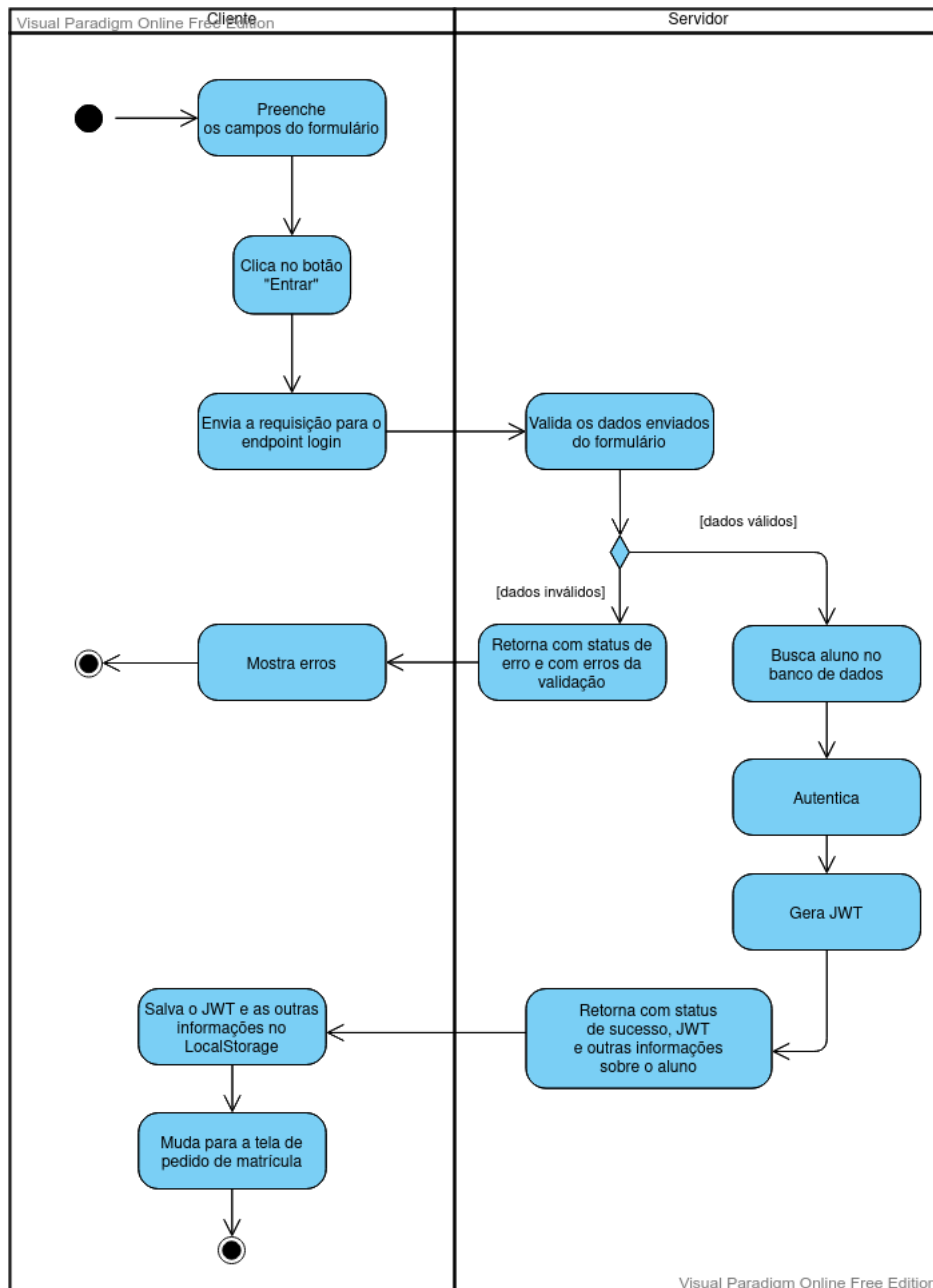
Fonte: Própria

4.2.2 Autenticando um aluno

A autenticação é responsável por permitir que o aluno envie seu nome de usuário e senha para identificar-se e, posteriormente, ter acesso às outras funcionalidades implementadas através da autorização. Nessa funcionalidade, foi necessário implementar tanto no servidor quanto no cliente. O fluxo dessa funcionalidade pode ser visto na figura 29 através de um diagrama de atividades.

Os detalhes de implementação no cliente e servidor serão explicados nas seções seguintes.

Figura 29 – Fluxo ao autenticar um aluno



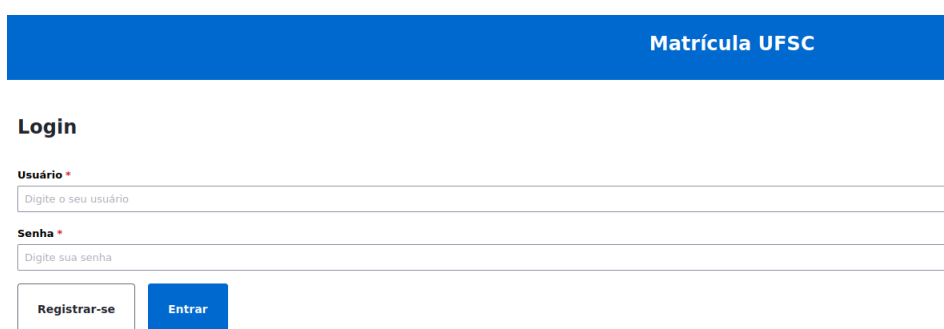
Fonte: Própria

4.2.2.1 Cliente

No cliente, foi necessário desenvolver uma tela, que pode ser vista na figura 30, que contém um formulário para preencher as informações necessárias que são o nome de usuário e a senha. Existem dois botões nessa tela, um para registrar com o título 'Registrar-se', e o outro para submeter os dados ao servidor com o título 'Entrar'. Ao clicar no botão de enviar, uma requisição POST para o endpoint `/login` é enviada ao servidor com o corpo contendo as informações preenchidas no formato `JSON`. Se não houver nenhum erro de

validação por parte do servidor, o usuário será redirecionado para a tela do pedido de matrícula e guardará o **JWT** no Local Storage do navegador para autorizar as próximas requisições. Vale ressaltar que essa prática não é correta porque um atacante poderia acessar o Local Storage, pegar o **JWT** para realizar as requisições ao servidor e fingir ser um aluno, todavia, como as questões de segurança não são o foco desse projeto, optou-se por essa abordagem para simplificar o desenvolvimento. A figura 31 mostra um exemplo de **JWT** guardado no Local Storage. Se existirem erros na validação, eles serão mostrados nos campos assim como na funcionalidade anterior.

Figura 30 – Tela para autenticar o aluno



Fonte: Própria

Figura 31 – **JWT** no Local Storage após uma autenticação bem-sucedida

Fonte: Própria

4.2.2.2 Servidor

No servidor, foi necessário adicionar um novo método ao *AlunoController* para receber as requisições de autenticação no endpoint */login*.

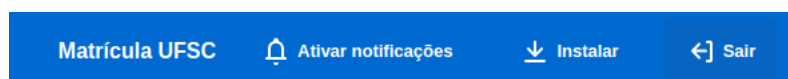
Antes de autenticar, uma validação é feita para garantir que o nome de usuário e senha estão preenchidos pela classe *LoginInputValidator*. Caso não ocorra um erro de validação, a autenticação é feita pela classe *AuthenticationService* disponibilizada pela Spring Security, buscando o usuário pelo nome e verificando se a senha no banco de dados bate com a senha recebida depois de ser computada pela função hash. Se ocorrer um erro na autenticação, o servidor adicionará uma mensagem de erro 'Usuário e/ou senha inválido(s)' e retornará o código de erro 400 (*Bad Request*) ao cliente. Senão, um **JWT**

é gerado contendo o *subject* com a matrícula do aluno e o *issued at* com a data na hora da criação, assinado usando o algoritmo [HMAC](#) com SHA-256 e retornado ao cliente. Toda a geração do [JWT](#) pode ser visto na classe *JWTService*. A data de expiração não foi colocada porque seria necessário desenvolver uma funcionalidade de *refresh token* no servidor e cliente, ampliando ainda mais o escopo.

4.2.2.3 Desautenticação do aluno

Para permitir que diversos usuários utilizem o cliente no mesmo navegador, foi desenvolvida funcionalidade de desautenticação, que cancela a autenticação efetuada. Um botão com o título 'Sair' foi desenvolvido que, ao ser clicado, removerá todos os dados do Local Storage do navegador associados ao aluno, como o [JWT](#). A figura 32 mostra esse botão.

Figura 32 – Botão para desautenticação ou sair



Fonte: Própria

4.2.3 Autorizando um aluno

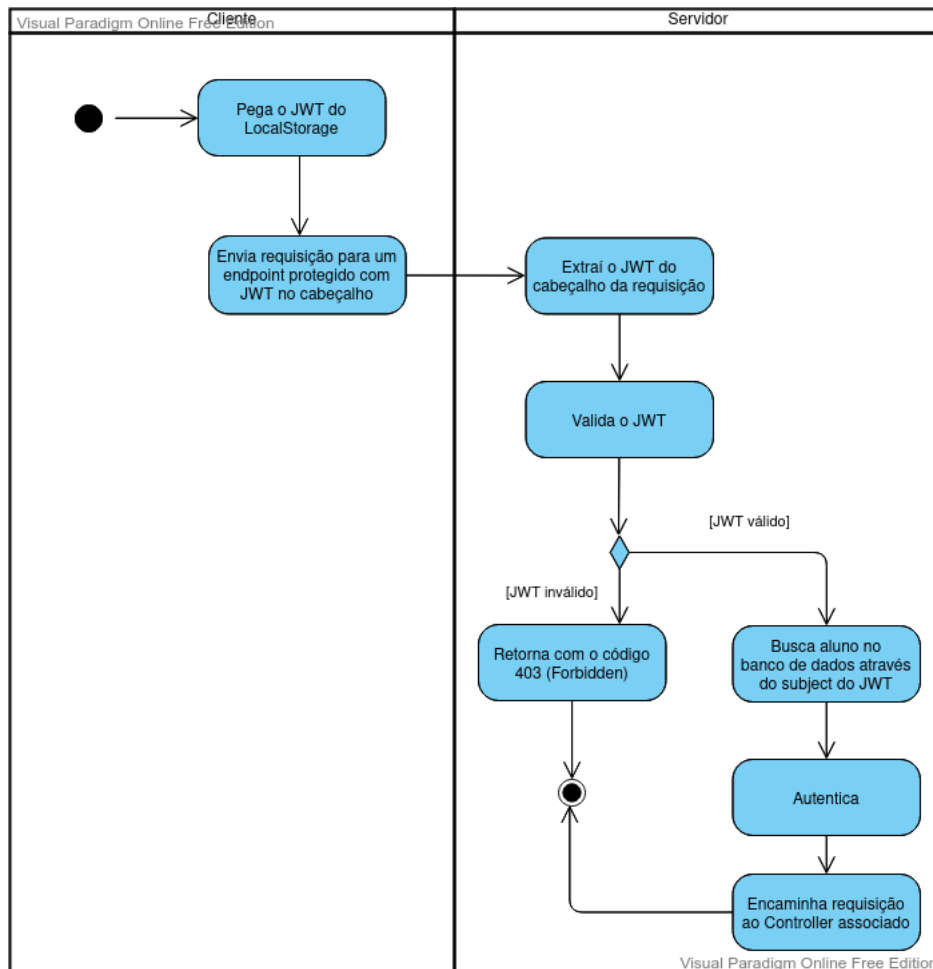
A autorização é responsável por garantir que apenas um aluno autenticado tenha acesso às funcionalidades de criar, editar e visualizar o pedido de matrícula. Nessa funcionalidade, foi necessário implementar tanto no servidor quanto no cliente. O fluxo dessa funcionalidade pode ser visto na figura 33 através de um diagrama de atividades.

Os detalhes de implementação no cliente e servidor serão explicados nas seções seguintes.

4.2.3.1 Cliente

Como apenas os endpoints de autenticação (*/login*) e registro de aluno (*/registrarAluno*) estão disponíveis para usuários não-autenticados, é necessária uma autorização para os outros endpoints. Para que o cliente tenha acesso, é necessário enviar o [JWT](#) no cabeçalho *Authorization* de todas as requisições seguindo o formato '*Bearer <jwt>*' e esse token é extraído do Local Storage do navegador. Toda esse processo pode ser encontrado no arquivo *fetch.ts*.

Figura 33 – Fluxo ao autorizar um aluno



Fonte: Própria

4.2.3.2 Servidor

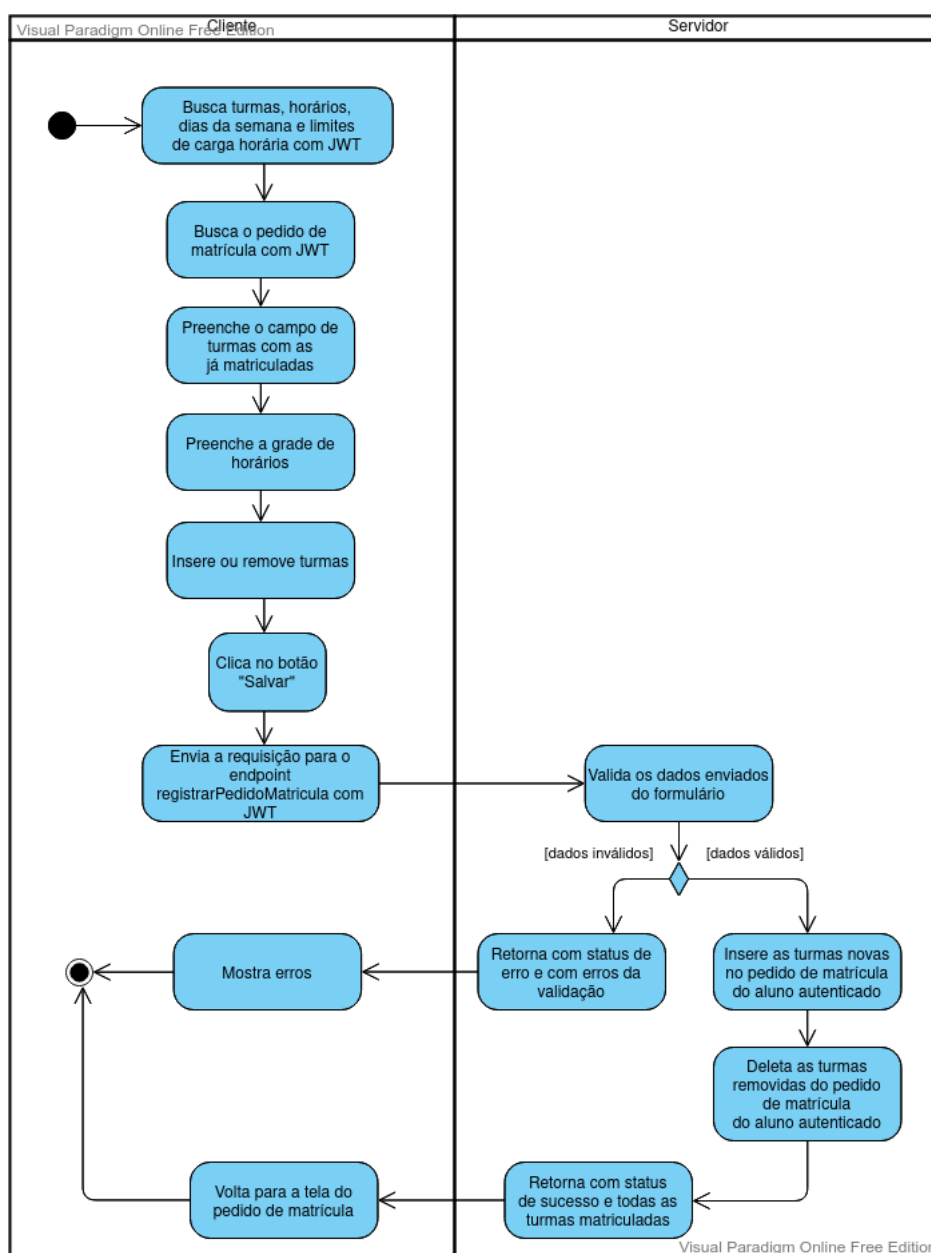
Como o Spring Security não possui um filtro pronto para a autorização com **JWT**, foi necessário desenvolver um filtro, *JWTFilter*, e adicioná-lo na *Filter Chain*, cuja configuração pode ser encontrada na classe *WebSecurityConfiguration*. Além disso, essa classe possui toda a configuração para restringir e liberar os endpoints da aplicação.

O *JWTFilter*, com o auxílio da classe *JWTService* que é a camada de serviço para manipular e validar os **JWTs**, extraí o **JWT** do cabeçalho da requisição e valida o token. Se a validação for um sucesso, o *JWTFilter* busca o aluno pela matrícula através do campo *subject* do **JWT** e autentica para que o próximo filtro, *UsernamePasswordAuthenticationFilter*, libere o acesso aos endpoints protegidos. Caso contrário, o *UsernamePasswordAuthenticationFilter* impede o acesso por não passar o usuário e senha pela requisição e retorna um erro com código 403 (*Forbidden*).

4.2.4 Criando ou editando um pedido de matrícula

Um pedido de matrícula pode ser definido como o conjunto de turmas que um determinado aluno deseja se matricular e, como um aluno pode mudar de ideia sobre quais turmas deseja, é importante permitir que ele as altere conforme à demanda. Nessa funcionalidade, foi necessário implementar tanto no servidor quanto no cliente. O fluxo dessa funcionalidade pode ser visto na figura 34 através de um diagrama de atividades.

Figura 34 – Fluxo ao criar e editar um pedido de matrícula



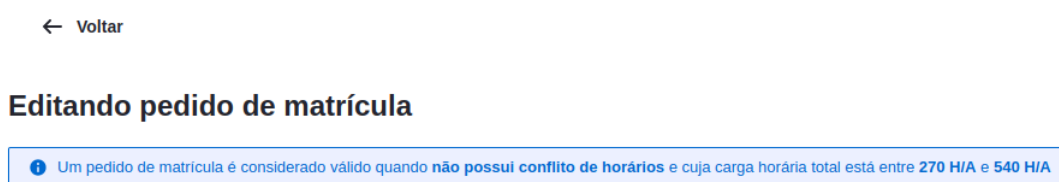
Fonte: Própria

Os detalhes de implementação no cliente e servidor serão explicados nas seções seguintes.

4.2.4.1 Cliente

No cliente, foi necessário desenvolver uma tela que contém um texto informando sobre os conflitos de horário e os limites de carga horária, que pode ser visto na figura 35, um formulário para selecionar as turmas no pedido de matrícula e a grade de horários. Optou-se por manter a mesma estrutura da grade de horários do CAGR para facilitar a leitura já que os alunos estão acostumados com ela. A figura 36 mostra a grade de horários.

Figura 35 – Texto sobre conflitos e carga horária



Fonte: Própria

Figura 36 – Grade de horários

Grade de horários

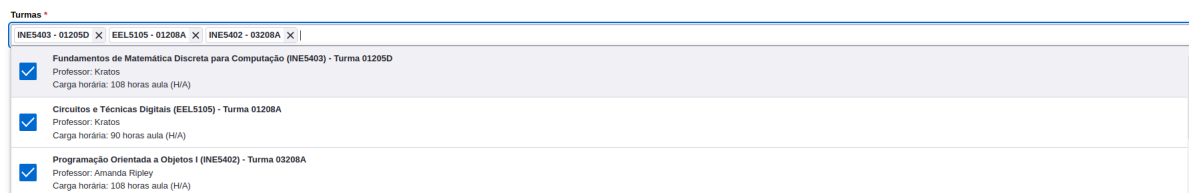
	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
07:30				INE5402 - 03208A LICT3		
08:20	INE5403 - 01205D EF1204	INE5403 - 01205D EF1204		INE5402 - 03208A LICT3		
09:10	INE5403 - 01205D EF1204	INE5403 - 01205D EF1204		INE5402 - 03208A LICT3		
10:10						
11:00 - 11:50						
13:30						
14:20						
15:10	EEL5105 - 01208A EEL004			INE5402 - 03208A CTC301		
16:20	EEL5105 - 01208A EEL004			INE5402 - 03208A CTC301		
17:10 - 18:00	EEL5105 - 01208A LABSDG			INE5402 - 03208A CTC301		
18:30	EEL5105 - 01208A LABSDG		INE5403 - 01205D EF1204			
19:20	EEL5105 - 01208A LABSDG		INE5403 - 01205D EF1204			
20:20						
21:10 - 22:00						

Fonte: Própria

Em relação ao formulário, o campo de seleção de turmas permite o usuário filtrar pelo código e nome da disciplina, código da turma e nome do professor, como pode ser visto na figura 38. A figura 37 mostra o campo de turmas com turmas selecionadas pelo usuário.

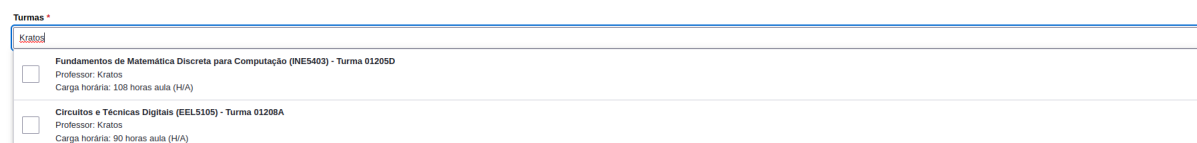
Sobre a carga horária total do pedido e a grade de horários, essas informações são atualizadas conforme o aluno seleciona as turmas no formulário. No caso da carga horária total, a cor do texto é alterada para mostrar ao usuário se está dentro ou fora dos limites definidos, como pode ser visto nas figuras 39 e 40.

Figura 37 – Campo de turmas com turmas selecionadas



Fonte: Própria

Figura 38 – Buscando pelo nome do professor no campo de turmas



Fonte: Própria

Figura 39 – Texto sobre a carga horária total quando está fora dos limites definidos

Editando pedido de matrícula

Um pedido de matrícula é considerado válido quando **não possui conflito de horários** e cuja carga horária total está entre **270 H/A** e **540 H/A**

Carga horária total: **198 H/A**

Fonte: Própria

Figura 40 – Texto sobre a carga horária total quando está dentro dos limites definidos

Editando pedido de matrícula

Um pedido de matrícula é considerado válido quando **não possui conflito de horários** e cuja carga horária total está entre **270 H/A** e **540 H/A**

Carga horária total: **306 H/A**

Fonte: Própria

Existem dois botões nessa tela, um para voltar para a visualização do pedido de matrícula com o título 'Voltar', e o outro para submeter os dados ao servidor com o título 'Salvar'. Ao clicar no botão de salvar, uma requisição POST para o endpoint `/registrarPedidoMatricula` é enviada ao servidor com o corpo contendo os códigos das turmas solicitadas no formato **JSON** e o **JWT** para autorização. Se não houver nenhum erro de validação por parte do servidor, o usuário será redirecionado para a tela do pedido

de matrícula com os dados atualizados. Caso contrário, os erros serão mostrados nos campos assim como nas funcionalidades anteriores.

Para buscar os dados para os componentes explicados anteriormente, requisições GET são enviadas para os seguintes endpoints:

- */turmas*: Busca todas as turmas registradas no sistema;
- */limitesCargaHoraria*: Busca os limites de carga horária estabelecidos pelo sistema;
- */horarios*: Busca os horários registrados no sistema para a grade de horários;
- */diasSemana*: Busca os dias da semana registrados no sistema para a grade de horários;

Além disso, caso o aluno selecione turmas com conflito de horários, a grade de horários mostrará esse conflito com um texto em vermelho para deixar o mais evidente possível sobre o erro, como pode ser visto na figura 41.

Figura 41 – Conflito na grade de horários

	Segunda	Terça	Quarta
07:30			
08:20	INE5403 - 01205D EF1204	INE5403 - 01205D EF1204	
09:10	INE5403 - 01205D EF1204	INE5403 - 01205D EF1204	
10:10			
11:00 - 11:50			
13:30			
14:20			
15:10	EEL5105 - 01208A EEL004		
16:20	EEL5105 - 01208A EEL004		
17:10 - 18:00	EEL5105 - 01208A LABSDG		INE5401 - 03202A CTC102
18:30	EEL5105 - 01208A LABSDG		INE5403 - 01205D EF1204 INE5401 - 03202A CTC102
19:20	EEL5105 - 01208A LABSDG		INE5403 - 01205D EF1204
20:20			
21:10 - 22:00			

Fonte: Própria

No caso do aluno já ter criado o pedido de matrícula, assim que o aluno acessa essa tela, os dados desse pedido são buscados no servidor através de uma requisição GET para o endpoint */pedidoMatricula*, e, em seguida, as turmas são colocadas no campo de turmas e a grade de horários preenchida com os horários das turmas solicitadas no pedido. A implementação associada a esse endpoint será explicada com mais detalhes na seção 4.2.5.

4.2.4.2 Servidor

No servidor, foi criado um novo controller chamado *PedidoMatriculaController* que ficou responsável por tratar as requisições dos endpoints citados na seção anterior. Para os

endpoints */turmas*, */horarios* e */diasSemana*, foi desenvolvido métodos na camada de serviço, *PedidoMatriculaService*, que buscam todos os registros presentes nas tabelas Turma, Horario e DiaSemana respectivamente. Nesses casos, não há necessidade de validação porque nenhum dado é enviado na requisição. Já no caso do endpoint */limitesCargaHoraria*, foi necessário apenas pegar as variáveis de ambiente que definem esses valores e retornar ao cliente.

Em relação ao endpoint */registrarPedidoMatricula*, há uma etapa de validação para garantir as seguintes regras:

- No mínimo, uma turma deve ser solicitada;
- Todos os códigos enviados devem estar associados a uma turma registrada no banco de dados;
- Inexistência de conflito de horários entre as turmas solicitadas;
- A carga horária total do pedido de matrícula deve respeitar os limites das cargas horárias definidas pelo sistema.

Se não houverem erros de validação, a camada de serviço busca as turmas que o aluno já estava matriculado e define quais são as turmas novas, que é a diferença entre turmas solicitadas e as matriculadas, as removidas, definida como a diferença entre as turmas matriculadas e as solicitadas, e mantidas, que é a interseção entre as solicitadas e as matriculadas. No caso das turmas novas, o pedido de matrícula, que consiste na associação entre um aluno e uma turma, é persistido na tabela, enquanto que os registros das turmas removidas são deletados da tabela. Por fim, são retornadas as turmas matriculadas ao cliente, que é a união entre as turmas novas e as mantidas.

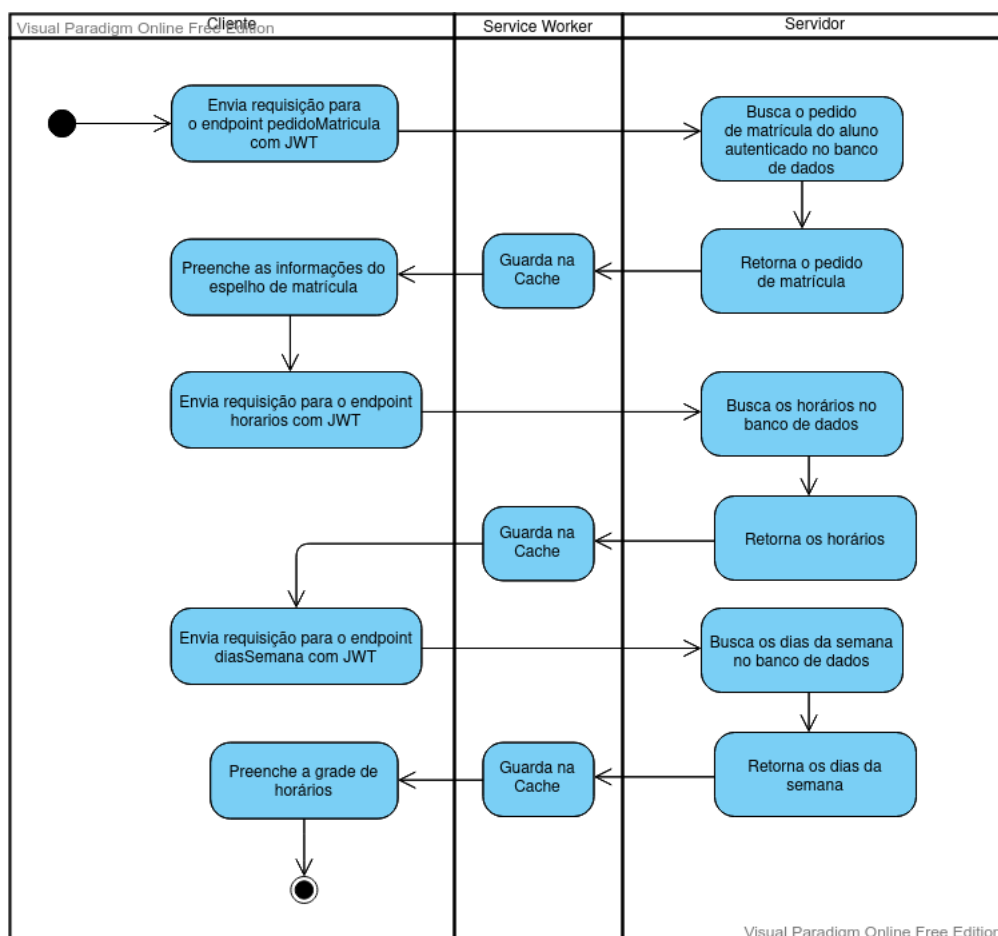
4.2.5 Visualizando grade de horários e status da vaga através do espelho de matrícula

A grade de horários permite que o aluno saiba em quais salas, dias e horários da semana serão as aulas de cada turma matriculada ou solicitadas através do pedido de matrícula. O espelho de matrícula permite visualizar o pedido de matrícula do aluno e quais turmas foram efetivamente matriculadas. No caso dessa aplicação, como é apenas abordado a etapa de solicitação e ajuste do pedido de matrícula, não estará contemplado o cenário de mostrar as turmas que foram efetivamente matriculadas.

Nessa funcionalidade, foi necessário implementar tanto no servidor quanto no cliente. Os fluxos dessa funcionalidade pode ser vistos nas figuras 42 e 43 para os dois estados possíveis de conexão com a rede através de diagramas de atividades.

Os detalhes de implementação no cliente e servidor serão explicados nas seções seguintes.

Figura 42 – Fluxo ao visualizar grade de horários e espelho de matrícula com conexão com a rede



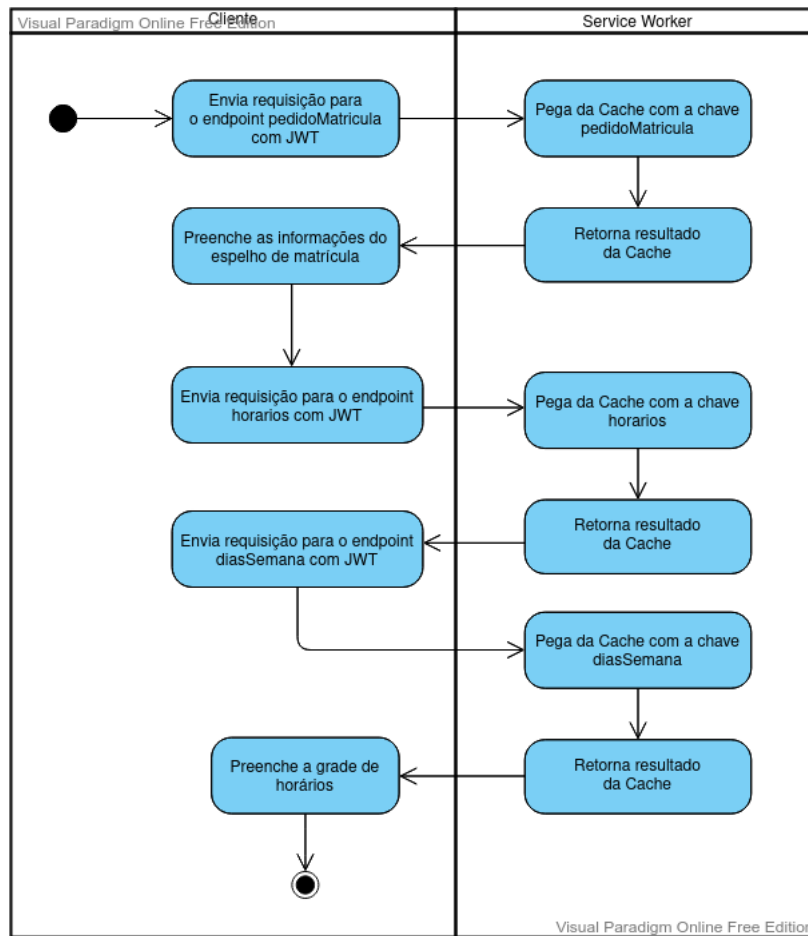
Fonte: Própria

4.2.5.1 Cliente

No cliente, foi necessário desenvolver uma tela composta pela grade de horários, o mesmo que é utilizado na criação e edição do pedido de matrícula, e o espelho de matrícula, que mostra cada turma que foi solicitada no pedido e seu professor, o status e posição da vaga do aluno. A figura 44 mostra o espelho de matrícula com os dois status definidos pela aplicação, o vaga possível, evidenciando uma possível matrícula na turma já que não é possível dar uma garantia porque vários alunos podem entrar e tomar a vaga, e na fila de espera.

Para buscar os dados para o espelho de matrícula, o cliente envia uma requisição GET para o endpoint `/pedidoMatricula` com o JWT para autorização e recebe todas as turmas matriculadas e a posição do aluno em cada uma. Para manter os dados atualizados, optou-se por utilizar a técnica de polling, que consiste no reenvio da requisição após um intervalo de tempo, devido à simplicidade da implementação do polling apesar do custo

Figura 43 – Fluxo ao visualizar grade de horários e espelho de matrícula sem conexão com a rede



Fonte: Própria

Figura 44 – Espelho de matrícula

Pedido de matrícula

A posição em uma turma é determinada pelo Índice de Aproveitamento Semestral Acumulado (IAA) e o seu é 1. Esse valor foi gerado aleatoriamente pelo servidor

Espelho de matrícula

Disciplina - Turma	Professor	Status da vaga	Sua posição
Fundamentos de Matemática Discreta para Computação (INE5403) - 01205C	Aloy	⚠ Fila de espera	1º lugar na fila de espera
Fundamentos de Matemática Discreta para Computação (INE5403) - 01205D	Kratos	✅ Vaga possível	1º lugar de 50
Circuitos e Técnicas Digitais (EEL5105) - 01208A	Kratos	✅ Vaga possível	1º lugar de 5
Programação Orientada a Objetos I (INE5402) - 03208A	Amanda Ripley	✅ Vaga possível	1º lugar de 5

Editar pedido de matrícula

Fonte: Própria

ao servidor. O cálculo do status é feito a partir da quantidade de vagas ofertadas pela disciplina e da posição do aluno, sendo que a vaga possível é definida quando o aluno possui uma posição igual ou inferior à quantidade de vagas ofertadas pela disciplina e a fila de espera caso a posição seja superior a número de vagas ofertadas. Já para a grade de horários, são realizadas requisições GET para os endpoints `/horarios` e `/diasSemana` para pegar os horários e dias da semana respectivamente para preencher a grade de horários corretamente. Além disso, os dados trazidos pela requisição ao endpoint `/pedidoMatricula` também são utilizados na grade para preencher os horários das turmas matriculadas.

4.2.5.2 Servidor

No servidor, foi necessário adicionar stored procedures na tabela de pedido de matrícula para atualizar as posições dos alunos para cada inserção, um aluno pedindo a matrícula em uma turma nova, ou remoção, um aluno desistindo de uma turma ao registrar ou editar o pedido de matrícula. O caso mais simples é o da remoção, já que basta decrementar a posição dos alunos da turma que possuem uma posição maior que a do aluno que desistiu. No caso da inserção, é necessário recalculer todas as posições dos alunos solicitantes porque não há garantia sobre o IAA do aluno novo, podendo ser maior ou menor que qualquer aluno solicitante. As figuras 45 e 46 mostram um exemplo da execução das stored procedures quando um aluno entra e desiste de uma turma respectivamente. O arquivo `triggers.sql` contém o script para criar e adicionar as stored procedures na tabela.

Depois da adição das stored procedures, foi necessário adicionar um novo método no `PedidoMatriculaController` para tratar as requisições para o endpoint `/pedidoMatricula`, que basicamente busca todas as turmas matriculadas do aluno autenticado, que é encontrado através do campo `subject` do JWT, as posições e quantidade de vagas ofertadas para cada uma das turmas.

4.2.5.3 Sem conexão com a rede

Nessa seção, serão descritos os passos para implementação no cliente para a visualização da grade de horários e do espelho de matrícula quando o usuário não tiver conexão com a rede. Além disso, serão descritas outros componentes que foram desenvolvidos para essa funcionalidade.

4.2.5.3.1 Service Worker

Para permitir o acesso à grade de horários e espelho de matrícula quando o usuário estiver sem conexão com a rede, foi necessário registrar e configurar o service worker.

Para registrá-lo, o código do `App.tsx` foi alterado para chamar a função `serviceWorkerRegistration.register()` ao invés da `serviceWorkerRegistration.unregister()`, assim o service worker seguirá o ciclo de vida que foi especificado na seção 2.8.1. Em seguida,

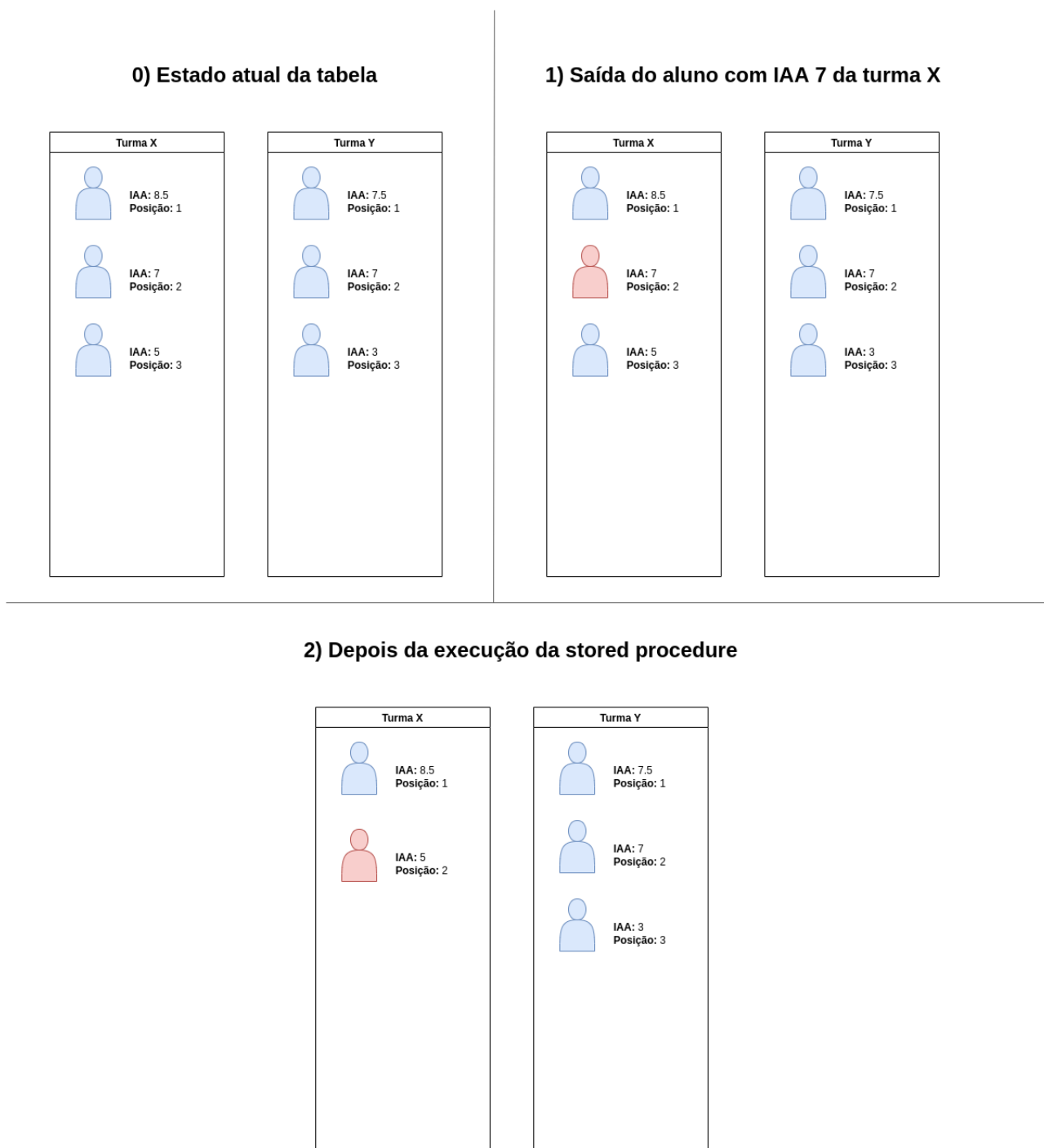
Figura 45 – Exemplo de execução da stored procedure quando um aluno entra em uma turma



Fonte: Própria

o arquivo *serviceWorker.js* foi atualizado para inicializar a cache com alguns arquivos

Figura 46 – Exemplo de execução da stored procedure quando um aluno sai de uma turma



Fonte: Própria

fundamentais para a funcionalidade offline após a instalação do service worker, como os ícones, manifesto, *index.html* e outros arquivos estáticos, e caso um dos arquivos não seja encontrado, toda a operação de inicializar é cancelada. O código abaixo mostra a implementação para essa parte. Para mais detalhes sobre a API da Cache, acesse [este](#)

[link](#).

```
// public/serviceWorker.js
const CACHE_NAME = 'MATRICULA_UFSC_PWA'

const IMAGES = [
  'images/book-stack-16.png',
  'images/book-stack-24.png',
  'images/book-stack-32.png',
  'images/book-stack-64.png',
  'images/book-stack-128.png',
  'images/book-stack-256.png',
  'images/book-stack-512.png',
]

const FILES_TO_CACHE = [
  '/',
  'index.html',
  'manifest.json',
  'favicon.ico',
  '/static/js/bundle.js',
  ...IMAGES
]

// Espera o service worker ser instalado
this.addEventListener('install', (event) => {
  event.waitUntil(
    // Abre uma cache para aplicação
    caches.open(CACHE_NAME).then((cache) => {
      // Adiciona todos os arquivos que precisam ser guardados.
      cache.addAll(FILES_TO_CACHE)
    })
  )
})
```

Em seguida, um Event Listener para o evento de fetch, que é acionado para cada requisição HTTP feita pelo cliente. Para essa aplicação, optou-se pela estratégia 'Network first, falling back to cache', ou seja, o cliente sempre busca as informações no servidor primeiro, e se a requisição for um sucesso, guarda o resultado na cache no caso de não ter conexão com a rede no futuro, caso contrário, pega o resultado guardado na cache, porque as informações da grade de horários e do espelho de matrícula precisam ser a mais atuais sempre que for possível. Para evitar de guardar dados na cache que não seriam utilizados, os endpoints guardados na cache foram */pedidoMatricula*, */horarios* e */diasSemana* já que não faria sentido editar o pedido de matrícula sem ter conexão com a rede. O código abaixo mostra implementação dessa parte e a figura 47 mostra o resultado da requisição do endpoint */pedidoMatricula* na cache.

```

// public/serviceWorker.js
const OFFLINE_URLS = ['/pedidoMatricula', '/horarios', '/diasSemana']

// Estrategia "Network first, falling back to cache"
// Ref: https://developer.chrome.com/docs/workbox/caching-strategies-overview/
this.addEventListener('fetch', (event) => {
  event.respondWith(
    caches.open(CACHE_NAME).then((cache) => {
      var fetchRequest = event.request.clone()
      return fetch(fetchRequest)
        .then((response) => {
          // Se der certo e for um endpoint que deve ser guardado, guarde na cache
          if (OFFLINE_URLS.some((url) => event.request.url.includes(url))) {
            console.debug(`[serviceWorker] Guardando na cache para fetchUrl=${event.request.url}`)
            cache.put(event.request, response.clone())
          }
          return response
        })
      // Se der algum problema na requisição, busca na cache
      .catch(() => {
        var fetchRequest = event.request.clone()
        return cache.match(fetchRequest.url)
      })
    })
  )
})

```

Figura 47 – Resultado da requisição para o endpoint `/pedidoMatricula` guardado na cache

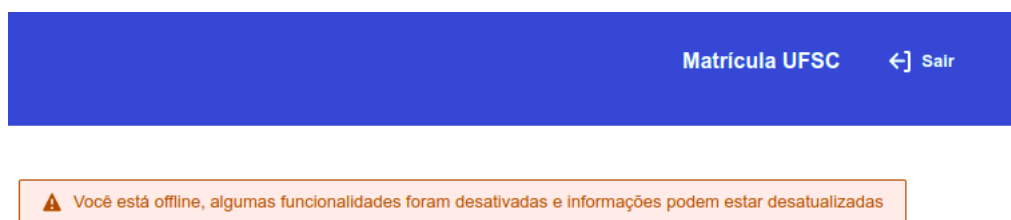
14	/pedidoMatricula	cors	application...	0	11/14/202...	Origin, Acc...
<div style="display: flex; border-bottom: 1px solid #ccc;"> ▼ [[{turma: {codigo: "01205C", vagasOfertadas: 2, professor: {nome: "Aloy"},...}, posicao: 3},...] </div> <ul style="list-style-type: none"> ▶ 0: {turma: {codigo: "01205C", vagasOfertadas: 2, professor: {nome: "Aloy"},...}, posicao: 3} ▶ 1: {turma: {codigo: "01205D", vagasOfertadas: 50, professor: {nome: "Kratos"},...}, posicao: 1} ▶ 2: {turma: {codigo: "01208A", vagasOfertadas: 5, professor: {nome: "Kratos"},...}, posicao: 1} ▶ 3: {turma: {codigo: "03208A", vagasOfertadas: 5, professor: {nome: "Amanda Ripley"},...}, posicao: 1} 						

Fonte: Própria

4.2.5.3.2 Cliente

Caso o usuário fique sem conexão, a funcionalidade de edição e cadastro do pedido de matrícula e notificações são desabilitadas removendo os botões associados, 'Editar pedido matrícula' e 'Ativar notificações' respectivamente, e uma mensagem de alerta avisando que as informações podem estar desatualizadas e que algumas funções foram desabilitadas devido à falta de conexão. As figuras 48 e 49 mostram a mensagem de alerta e o espelho de matrícula sem o botão de edição quando há perda de conexão.

Figura 48 – Mensagem de alerta e botão de instalação e notificações removidos quando falta conexão



Fonte: Própria

Figura 49 – Espelho de matrícula sem o botão de editar quando falta conexão



Fonte: Própria

Para atualizar os componentes sobre o estado da conexão, foi utilizada Context API do React (REACT, 2022) e dois Event Listeners para os eventos *offline*, quando o navegador perde a conexão, e *online*, quando o navegador retoma a conexão, para atualizar o valor do contexto. O arquivo *OnlineStatusProvider.tsx* apresenta essa implementação.

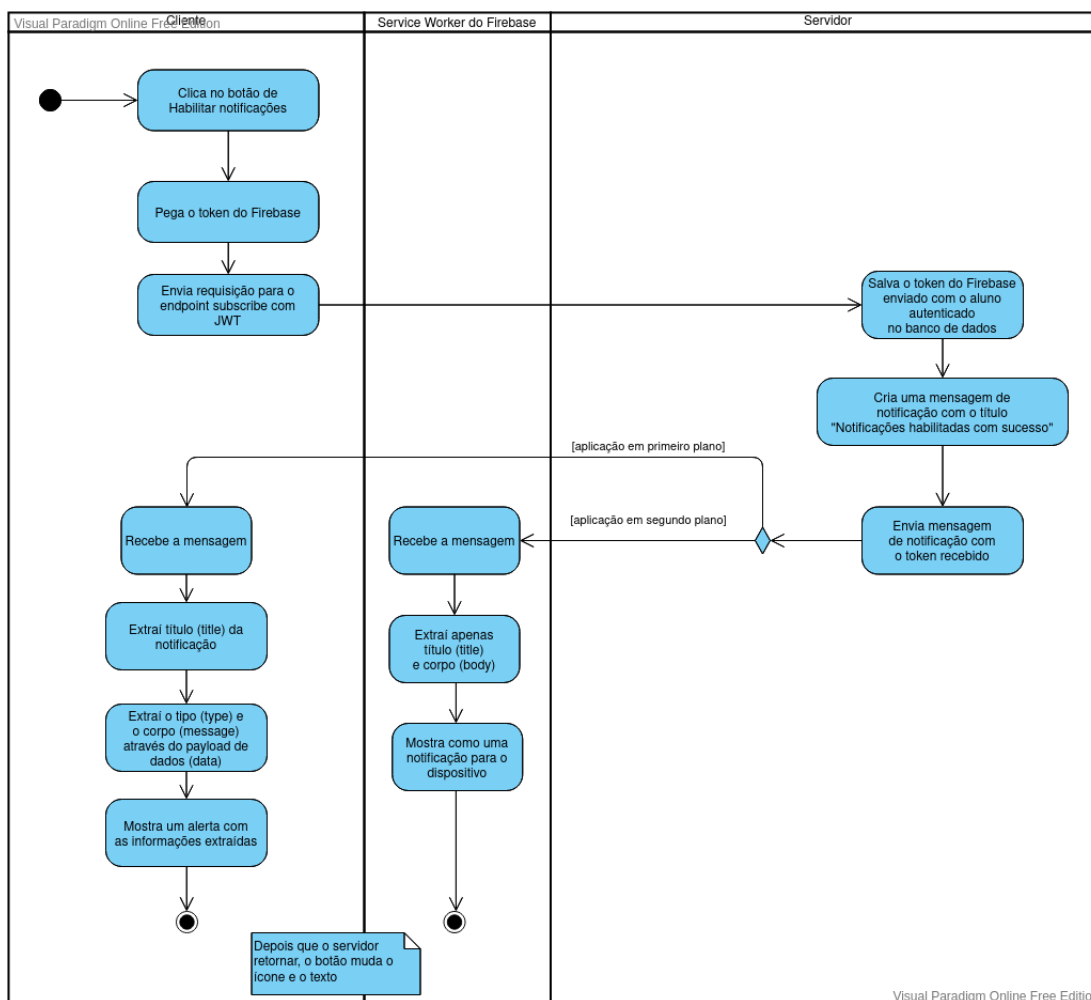
Não foi necessária nenhuma alteração nos componentes do espelho de matrícula e grade de horários porque as requisições serão respondidas com os valores guardados na cache através do service worker. A única modificação ocorreu no polling, que fica desativado quando não há conexão, já que os dados na cache serão os mais atuais até a retomada da conexão com a rede.

4.2.6 Notificações

As notificações permitem que os alunos fiquem mais informados sobre os status de cada turma solicitada no pedido de matrícula, ou seja, se entraram ao não pertencer mais ao grupo de alunos classificados ou saíram de uma fila de espera de uma turma ao entrar no grupo de alunos classificados após a saída de outro.

Nessa funcionalidade, foi necessário implementar tanto no servidor quanto no cliente e a ferramenta **FCM** foi utilizada devido à facilidade na configuração e uso sem precisar se preocupar com o Web Push Protocol. Os fluxos dessa funcionalidade pode ser vistos nas figuras 50 e 51 ao habilitar e desabilitar as notificações com diagramas de atividades.

Figura 50 – Fluxo ao habilitar as notificações



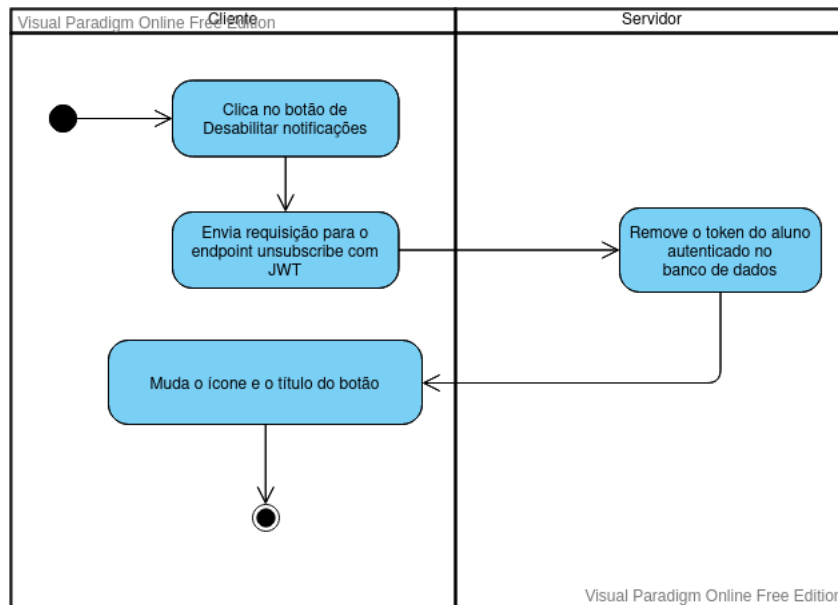
Fonte: Própria

Os detalhes de implementação no cliente e servidor serão explicados nas seções seguintes.

4.2.6.1 Passos iniciais

Antes do início da implementação, foi necessário criar um projeto na plataforma do Firebase a partir [deste link](#) para configurar as **SDKs** no cliente e servidor corretamente. As etapas da configuração serão explicadas nas seções seguintes para cada componente da aplicação.

Figura 51 – Fluxo ao desabilitar as notificações



Fonte: Própria

4.2.6.2 Cliente

Nessa seção, serão descritas as etapas para implementar a funcionalidade de notificações no cliente.

4.2.6.2.1 Configurando a SDK

Primeiramente, foi criada uma aplicação web no projeto do Firebase seguindo [este tutorial](#) para gerar o **JSON** de configuração da **SDK**. Em seguida, foi adicionada a biblioteca `firebase` e criado o arquivo `firebase.ts` para realizar a configuração da **SDK** e inicializar a aplicação web e o serviço do **FCM**, como é mostrado no código abaixo. O arquivo `firebase.ts` é importado no `index.tsx`.

```
// src/firebase.ts
import { initializeApp } from 'firebase/app';

// Colocando o JSON da aplicação web gerado aqui
const firebaseConfig = {
  // ...
};

// Inicializando a aplicação para poder usar os serviços do Firebase
const app = initializeApp(firebaseConfig);

// Inicializando o serviço do FCM e exportando para outros arquivos terem acesso
export const messaging = getMessaging(app);
```

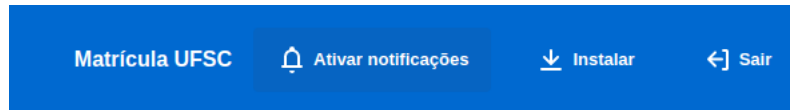

4.2.6.2.2 Habilitando as notificações

Antes de habilitar as notificações, é necessário ter a chave pública **VAPID** para que o Firebase realize a subscrição no push service corretamente. Com isso, o par de chaves **VAPID** foi gerado na plataforma do Firebase seguindo [este tutorial](#).

Em seguida, foi criado um botão com o título 'Habilitar notificações', que pode ser visto na figura 52, na barra da aplicação para permitir que o usuário as habilite. Ao clicar nele, o cliente verifica se o navegador possui suporte para as notificações, e se tiver, avalia a permissão dada pelo usuário para as notificações através da Notification **API**. Se a permissão estiver no estado padrão, ou seja, se o usuário nunca foi questionado sobre ela, uma janela de pop-up é aberta para solicitar a permissão e, caso o usuário permita, é utilizado o método `getToken` da biblioteca `firebase` para pegar o token que identificará unicamente o dispositivo. Esse método tem como parâmetros: o serviço messaging que foi exportado no `firebase.ts` e a chave pública **VAPID**. Por fim, é feita uma requisição **POST** para o endpoint `/subscribe` junto com o **JWT** para enviar o token do aluno ao servidor, permitindo enviar notificações futuramente. O código abaixo mostra a implementação para pegar o token e enviá-lo ao servidor. A figura 53 mostra todos os passos para essa etapa através de um diagrama de atividades.

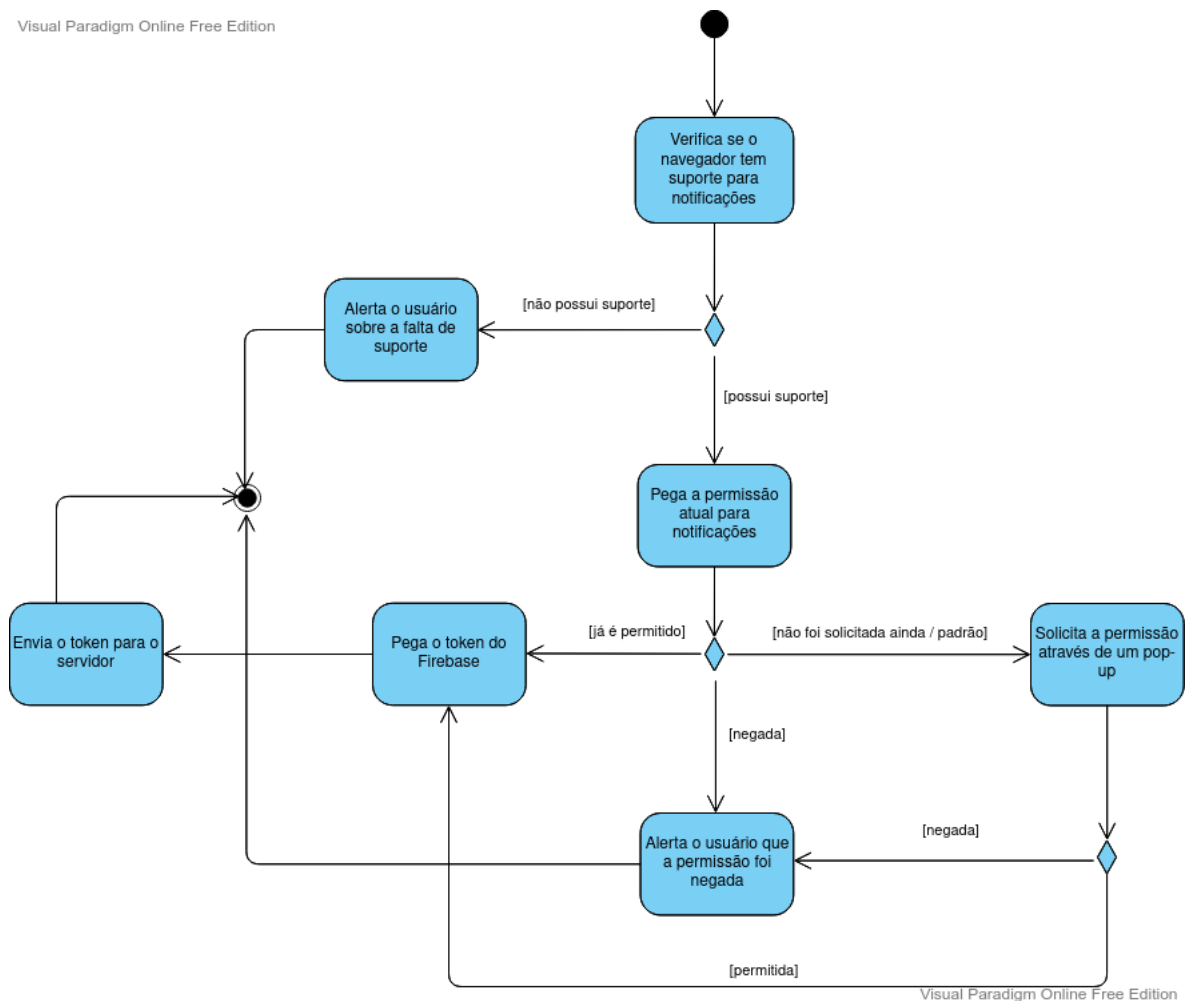
```
// src/notifications/subscribe.ts
function subscribeUser(updateStates: (status: boolean, loading: boolean) => void) {
  navigator.serviceWorker.ready.then(() => {
    getToken(messaging, {
      /**
       * A chave pública VAPID é definida como variável de ambiente nessa aplicação,
       * mas poderia ser recebida como parâmetro ou constante
       */
      vapidKey: process.env.REACT_APP_PUBLIC_VAPID_KEY,
    }).then((token) => {
      fetchWithAuthorization('subscribe', {
        method: 'POST',
        body: JSON.stringify({ token }),
        headers: {
          'Content-Type': 'application/json',
        },
      })
      // (...)
    })
  })
}
```

Figura 52 – Botão para habilitar notificações



Fonte: Própria

Figura 53 – Diagrama de atividades mostrando os passos para habilitar as notificações



Fonte: Própria

4.2.6.2.3 Recebendo as notificações

Para as mensagens recebidas em segundo plano, foi adicionado um novo arquivo na pasta *public* com o nome *firebase-messaging-sw.js*, que é mandatório para funcionar corretamente. Nesse arquivo, os scripts do firebase são importados e a aplicação e o serviço do FCM são inicializados. O código abaixo mostra como ficou esse arquivo.

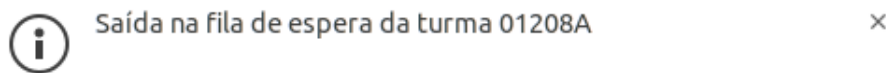
```
// public/firebase-messaging-sw.js
// Importando os scripts para esse service worker funcionar
importScripts('https://www.gstatic.com/firebasejs/9.0.0/firebase-app-compat.js')
importScripts('https://www.gstatic.com/firebasejs/9.0.0/firebase-messaging-compat.js')

// Configurações do Firebase para inicializar a aplicação
const firebaseConfig = {
  // ...
}

// Inicializa a aplicação e o FCM
firebase.initializeApp(firebaseConfig)
const messaging = firebase.messaging()
```

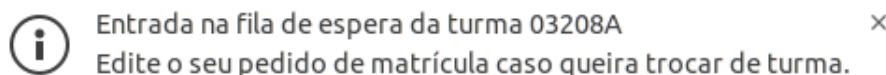
Não foi necessário utilizar a função *onBackgroundMessage* porque enviaremos apenas as mensagens de notificação com payload de dados, que são tratadas automaticamente pela SDK do FCM (FIREBASE, 2022b). Caso seja necessário enviar mensagens de dados e tratá-las em segundo plano, basta seguir [este tutorial](#). As figuras 54 e 55 mostram as notificações quando um aluno sai e entra na fila de espera respectivamente que é gerada automaticamente pelo *firebase-messaging-sw.js*.

Figura 54 – Notificação ao sair da fila de espera de uma turma



Fonte: Própria

Figura 55 – Notificação ao entrar na fila de espera de uma turma



Fonte: Própria

Já no caso das mensagens em primeiro plano, foi criado um componente que utiliza a função *onMessage* e extrai o título, mensagem e tipo de alerta, que pode ser sucesso ou aviso, da mensagem de notificação para mostrar um alerta ao usuário. O código abaixo mostra a implementação desse componente e as figuras 56 e 57 mostram as notificações quando um aluno sai e entra na fila de espera respectivamente. Além disso, esse componente criará um evento chamado *atualizar* para o componente responsável por

renderizar o espelho de matrícula e a grade de horários buscar dados mais atuais através de uma requisição GET para o endpoint `/pedidoMatricula`, já que as notificações serão enviadas caso o aluno entre ou saia de uma fila de espera.

```
// src/components/PushNotificationsAlert.tsx
export function PushNotificationsAlert() {
  const [showAlert, setShowAlert] = useState<boolean>(false)
  const [message, setMessage] = useState<AlertMessage>({})

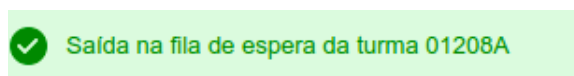
  onMessage(messaging, (payload) => {
    // pega o título da notificação
    const title = payload.notification.title
    // pega os dados que vai conter a mensagem e o tipo de alerta
    const data: NotificationData = payload.data

    if (data) {
      // atualiza o valor da mensagem a ser mostrada
      setMessage({ title, body: data.message, type: data.type })
      // mostra o alerta
      setShowAlert(true)

      // Avisar o componente responsável pela visualização do pedido de matrícula
      // para buscar os dados mais recentes
      const event = new Event('atualizar')
      window.dispatchEvent(event)
    }
  })

  return (
    showAlert &&
    message && (
      <Alert type={message.type} onCloseClick={() => setShowAlert(false)}>
        {message.title} {message.body && `: \${message.body}`}
      </Alert>
    )
  )
}
```

Figura 56 – Notificação ao sair da fila de espera de uma turma



Fonte: Própria

Figura 57 – Notificação ao entrar na fila de espera de uma turma

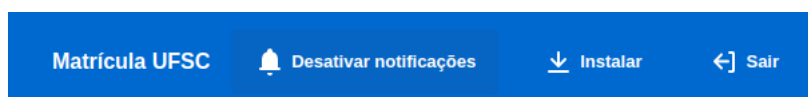


Fonte: Própria

4.2.6.2.4 Desabilitando as notificações

Tendo as notificações habilitadas, o botão das notificações tem o título alterado para 'Desabilitar notificações' para permitir que o aluno as cancele quando sentir a necessidade. Ao clicar nesse botão, o cliente envia uma requisição PUT para o endpoint `/unsubscribe` com o **JWT** para remover o token da base de dados. Caso a requisição seja um sucesso, o botão é alterado novamente para 'Habilitar notificações'. A figura ?? mostra esse botão.

Figura 58 – Botão de desativar notificações



Fonte: Própria

4.2.6.3 Servidor

No servidor, foi criado um novo controller chamado *PushNotificationController* para receber as requisições nos endpoints `/subscribe` e `/unsubscribe` e uma camada de serviço chamada *PushNotificationService* para solicitar o envio das notificações através da biblioteca do Firebase.

4.2.6.3.1 Inicializando a Firebase SDK Admin

Para utilizar a biblioteca, foi necessário inicializar a **SDK Admin** com a chave privada da conta de serviço, possibilitando que o servidor se autentique ao acessar os serviços da Firebase. Essa **SDK** é um conjunto de bibliotecas para permitir que os servidores interajam com o Firebase e seus serviços com privilégios de administrador, concedendo a permissão para enviar notificações no **FCM** (FIREBASE, 2022a). Para gerar essa chave, seguiu-se [este tutorial](#). Nesse projeto, a chave foi armazenada em uma variável de ambiente para evitar que outras pessoas tenham acesso.

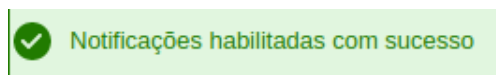
4.2.6.3.2 *Habilitando e desabilitando notificações*

Como foi explicado anteriormente, o *PushNotificationController* foi desenvolvido para tratar as duas requisições relacionadas às notificações.

Para o endpoint */unsubscribe*, requisitado quando o aluno não quer mais receber notificações, o token do aluno autenticado é removido da tabela no banco de dados.

No caso do endpoint */subscribe*, o token enviado é associado ao aluno autenticado na tabela do banco de dados e uma notificação com o título 'Notificações habilitadas com sucesso' é enviada para o dispositivo para confirmar que as notificações foram habilitadas, como pode ser visto na figura 40.

Figura 59 – Notificação depois de habilitar as notificações



Fonte: Própria

4.2.6.3.3 *Determinando quais alunos devem receber as notificações*

Para determinar quais alunos devem receber as notificações que ganharam ou perderam uma vaga em uma turma ao saírem ou entrarem na fila de espera respectivamente, foram criadas duas colunas, que podem ter o valor como verdadeiro ou falso, na tabela do pedido de matrícula. Essas colunas são preenchidas pelas stored procedures que foram descritas anteriormente. Além disso, apenas são selecionados os alunos que habilitaram as notificações, ou seja, que possuem um token do Firebase registrado no banco de dados.

Quando um aluno decide sair de uma turma, a coluna associada ao ganho da vaga é recalculada para todos os outros alunos que estavam nessa turma. O valor dessa coluna é definido como verdadeiro para todos os alunos que saíram da fila de espera, ou seja, os que tinham uma posição superior à quantidade de vagas ofertadas antes da remoção e ficaram com a posição inferior ou igual à quantidade de vagas ofertadas depois dessa operação. Caso contrário, o valor será definido como falso.

Quando um aluno entra em uma turma, a coluna relacionada à perda da vaga é redefinida para todos os outros alunos que estavam nessa turma. O valor dessa coluna é definido como verdadeiro para todos os alunos que não estavam na fila de espera e acabaram entrando nela, ou seja, os que tinha uma posição inferior ou igual à quantidade de vagas ofertadas antes do novo aluno entrar e ficaram com a posição superior à quantidade de vagas ofertadas depois dessa operação. Caso contrário, o valor será definido como falso.

As figuras 60 e 61 mostram um exemplo da execução das stored procedures quando um aluno entra e desiste de uma turma respectivamente em relação às colunas de notificação.

Figura 60 – Exemplo de execução da stored procedure quando um aluno entra em uma turma para determinar se deve ser notificado ou não



Fonte: Própria

4.2.6.3.4 Enviando notificações

O envio das notificações ocorre após a criação ou alteração de um pedido de matrícula, pegando as turmas novas e removidas para avaliar quais alunos dessas turmas devem ser notificados sobre a entrada e saída na fila de espera respectivamente.

Figura 61 – Exemplo de execução da stored procedure quando um aluno sai de uma turma para determinar se deve ser notificado ou não



Fonte: Própria

Como descrito anteriormente, o *PushNotificationService* ficou responsável pelo envio das notificações aos dispositivos dos alunos que as habilitaram. A implementação foi simples, apenas precisando criar a notificação e mensagem através dos builders e chamando a função *send* da biblioteca do *FCM*. Os pares chave-valor personalizados para o payload da mensagem foram *title*, o título da notificação, *message*, a mensagem que deve ser mostrada no alerta, e *type*, se o alerta deve ser de sucesso (saída da fila de espera) ou aviso (entrada na fila de espera), e eles foram inseridos através do método *putAllData*. A

implementação desse envio pode ser vista no código abaixo.

```
// PushNotificationService.kt
private fun sendNotification(notificacao: NotificacaoAluno, subscriptionToken: String) {
    logger.debug("Sending notification ({notificacao.type.name}) to {subscriptionToken}")

    // Criação da notificação
    val notification = Notification.builder()
        .setTitle(notificacao.title)
        .setBody(notificacao.message)
        .build()

    // Criação da mensagem de notificação que será enviada pelo FCM
    val message = Message.builder()
        // Payload dos dados da mensagem
        .putAllData(notificacao.convertToNotificationData())
        // Token que identifica unicamente o dispositivo do aluno autenticado
        .setToken(subscriptionToken)
        .setNotification(notification)
        .build()

    FirebaseMessaging.getInstance().send(message)
}
```

Após esse envio, as duas colunas associadas às notificações de todos os alunos que foram notificados são redefinidas para falso.

A imagem 62 mostra o fluxo completo para o envio das notificações no servidor através de um diagrama de atividades.

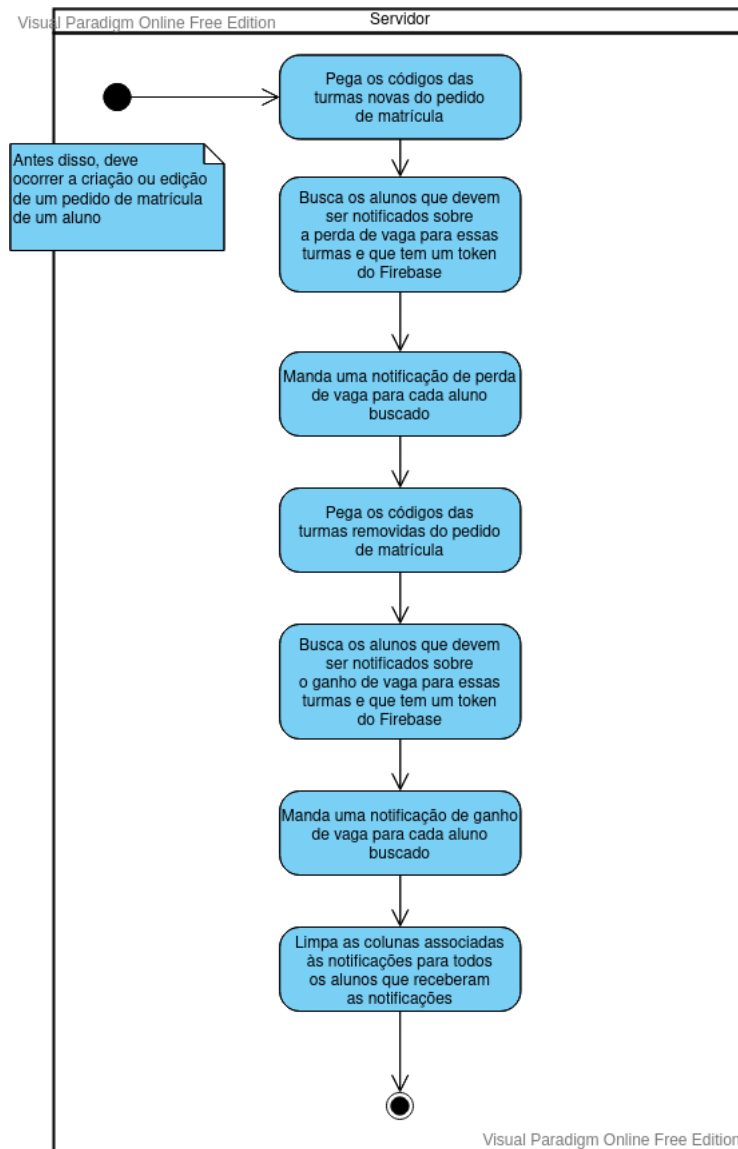
4.2.7 Instalação

A instalação permite que o aluno utilize a aplicação como um aplicativo de celular ou desktop sem precisar acessar uma página da loja de aplicativos, podendo acessá-la com apenas um clique no ícone. No caso dos PWAs, essa instalação é opcional, garantindo mais liberdade ao usuário para apenas instalar a aplicação quando achar importante ou necessário.

Para tornar a PWA instalável, foi necessário alterar o manifesto gerado automaticamente para conter as informações corretas sobre a aplicação, adicionar um link para o manifesto no *index.html* com o atributo *rel* tendo o valor *manifest* e garantir que o manifesto cumpria os requisitos mínimos para ser instalável, que são ter os seguintes campos definidos (LEPAGE, 2022b):

- *short_name* ou *name*;
- Ícones de 192px e 512px;

Figura 62 – Fluxo do envio de notificações após criar ou editar um pedido de matrícula



Fonte: Própria

- start_url;
- display.

O código abaixo mostra o manifesto implementado.

```

// public/manifest.json
{
  "short_name": "Matrícula UFSC",
  "name": "Matrícula UFSC - PWA",
  "start_url": ".",
  "display": "standalone",
  "theme_color": "#000000",

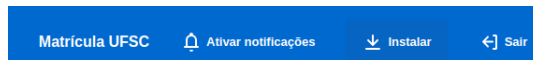
```

```
"background_color": "#ffffff",
"orientation": "landscape",
"icons": [
  {
    "src": "favicon.ico",
    "sizes": "16x16",
    "type": "image/x-icon"
  },
  {
    "src": "images/book-stack-16.png",
    "type": "image/png",
    "sizes": "16x16"
  },
  {
    "src": "images/book-stack-24.png",
    "type": "image/png",
    "sizes": "24x24"
  },
  {
    "src": "images/book-stack-32.png",
    "type": "image/png",
    "sizes": "32x32"
  },
  {
    "src": "images/book-stack-64.png",
    "type": "image/png",
    "sizes": "64x64"
  },
  {
    "src": "images/book-stack-128.png",
    "type": "image/png",
    "sizes": "128x128"
  },
  {
    "src": "images/book-stack-256.png",
    "type": "image/png",
    "sizes": "256x256"
  },
  {
    "src": "images/book-stack-512.png",
    "type": "image/png",
    "sizes": "512x512"
  }
]
}
```

Além disso, foi implementado um botão para permitir que o aluno instalasse a aplicação. Para desenvolvê-lo, foi necessário adicionar um Event Listener para o evento *beforeinstallprompt*, que é acionado quando o cliente atende aos requisitos de instalação

(LEPAGE, 2021). Com esse Event Listener, é possível guardar esse evento para acionar a instalação quando o usuário clicar no botão através do método *prompt*. A figura 63 mostra esse botão. Ressalta-se que esse evento ainda não é disparado pelo navegador Mozilla quando acessado pelo desktop, fazendo com o que essa funcionalidade não se comporte como é o esperado.

Figura 63 – Botão de instalação



Fonte: Própria

O código abaixo mostra a implementação feita para o botão de instalação.

```
// src/components/header-bar/InstallButton.tsx
export function InstallButton(props: InstallButtonProps) {
  const { deferredPrompt, reset, theme } = props

  // Baseado em: https://www.amitmerchant.com/adding-custom-install-button-in-progressive-web-app
  const handleDownloadClick = async () => {
    // mostra o pop-up de instalação
    deferredPrompt.prompt()
    // pega a resposta do usuário
    const { outcome } = await deferredPrompt.userChoice
    console.debug(`[HeaderBar] userChoice: ${outcome}`)
    if (outcome === 'accepted') reset()
  }

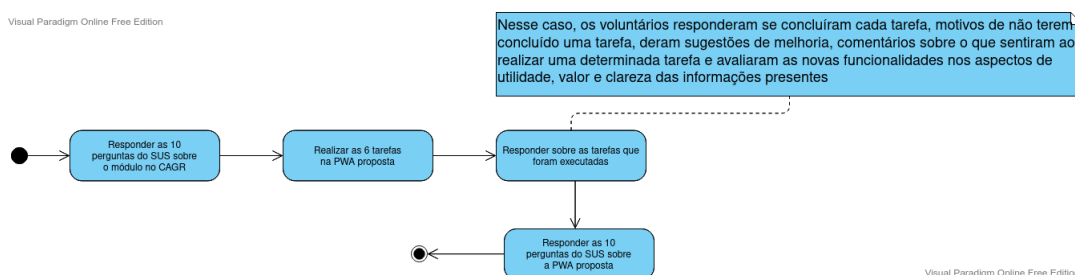
  return (
    <Button onClick={handleDownloadClick} skin='ghost' size='large'>
      <HFlow hSpacing={0.25} alignItems='center'>
        <Icon icon='download' style={{ color: theme.pallete.gray.c100 }} />
        <Text style={{ color: theme.pallete.gray.c100 }}>Instalar</Text>
      </HFlow>
    </Button>
  )
}
```

4.3 TESTE DE USABILIDADE

Como descrito na metodologia, o teste de usabilidade tem como objetivo mensurar a satisfação dos usuários em relação à aplicação oficial disponibilizada no CAGR e a que está sendo proposta com a participação de veteranos e recém-formados da UFSC, representando o público-alvo dessas aplicações, para permitir que a autora realize uma comparação quantitativa entre elas. Esse teste pode ser dividido nas seguintes etapas: avaliação da

satisfação dos alunos sobre o módulo oficial através de um questionário, execução de 6 tarefas no novo módulo para simular tarefas reais e avaliação da satisfação em relação à PWA desenvolvida e das novas funcionalidades propostas através de outro questionário. A figura 64 mostra essas etapas que foram feitas pelos voluntários através de um diagrama de atividades. Todas as etapas desse teste foram feitas de forma remota e assíncrona e as respostas foram anônimas para assegurar a privacidade, sigilo e confidencialidade de todos os participantes. Para medir a satisfação, foram criados dois questionários através do Google Forms e um conjunto de 6 tarefas para os participantes do teste realizarem na interface nova para adquirir experiência com ela. Ressalta-se que o foco desse projeto estava na avaliação da usabilidade apenas sob o aspecto da satisfação dos usuários, sendo que os outros aspectos como eficácia e eficiência não foram considerados relevantes para essa avaliação visto que ela estava voltada para verificar o impacto da nova abordagem na realização das matrículas assim como a reação aos feedbacks que o sistema atual não fornece.

Figura 64 – Etapas do teste de usabilidade que foram executadas pelos voluntários



Fonte: Própria

Em relação aos participantes que foram escolhidos para realizar todas as etapas do teste, optou-se pelos veteranos e recém-formados por já terem experiência com a aplicação oficial, possibilitando que respondessem a primeira avaliação e realizassem uma comparação entre as duas interfaces. A participação deles foi voluntária e os convites foram feitos pelo e-mail do curso e canais de comunicação do Laboratório Bridge, já que a autora trabalha nessa instituição e o laboratório possui diversos veteranos e recém-formados da UFSC. Antes de receberem as tarefas, os participantes precisaram assinar digitalmente o [Termo de Consentimento Livre e Esclarecido \(TCLE\)](#), cujo conteúdo pode ser encontrado no apêndice A. Esse termo foi fortemente baseado [nesse modelo](#). O Parecer da aprovação pode ser encontrado no anexo B e o [Certificado de Apresentação de Apreciação Ética \(CAAE\)](#) da pesquisa é **64772322.3.0000.0121**.

Nas seções seguintes, as tarefas e os questionários serão descritos com mais detalhes.

4.3.1 Tarefas

Sobre as tarefas, elas foram enviadas por e-mail ou Slack para cada participante do teste por meio de um documento junto com instruções para auxiliar e orientar os participantes. O objetivo dessas tarefas era permitir que o participante explorasse todas as funcionalidades propostas para avaliá-las no segundo questionário. Nessas tarefas, há uma descrição breve do contexto imaginário que o participante estava inserido e um conjunto de atividades que deveriam ser realizadas para concluir a tarefa. As atividades não continham um passo a passo claro do que deve ser feito, visto que isso afetaria a avaliação dos usuários sobre o sistema já que ele deve ser fácil de entender e usar. Esse modelo de teste usabilidade foi baseado no que é feito pelo Laboratório Bridge para a realização dos testes das versões. A figura 65 mostra um exemplo de tarefa que foi proposta para os usuários realizarem. Todas as tarefas realizadas pelos participantes podem ser encontradas no apêndice C.

Figura 65 – Primeira tarefa que foi proposta aos participantes do teste

Tarefa 1

Imagine que a Universidade Federal de Santa Catarina (UFSC) desenvolveu um novo sistema de matrícula e que está no período da solicitação de matrícula. Além disso, você e seu amigo combinaram de ingressar nas mesmas turmas das disciplinas **INES403**, **INES402** e **EEL5105** para fazer os trabalhos em dupla e que você deveria se matricular em Cálculo 1 (**MTM3101**) porque passou na prova de proficiência, mas o seu amigo não, então ele precisaria fazer Pré-Cálculo primeiro. Além disso, **você não poderia ter aulas na quinta-feira no período matutino (7h até 12h)** porque precisa cuidar dos seus irmãos mais novos até o horário da aula deles e um dos veteranos comentou que a **Aloy** era uma excelente professora e que você deveria pegar todas as turmas que ela ministrasse.

1. Acesse o novo sistema de matrícula que está disponível neste [link](#)
2. Realize o cadastro e a autenticação
3. Solicite as turmas das disciplinas

Anotações sobre a Tarefa 1

Nessa parte, anote se você conseguiu concluir a tarefa e quais foram as suas impressões sobre o sistema e essa tarefa. Sugestão de perguntas para serem respondidas: Foi fácil de achar as informações? Foi rápido de realizar a tarefa? As informações são fáceis de ler e objetivas? Existem melhorias que poderiam ser feitas?

Se você não conseguiu concluir, deixe o motivo (ex: "Não achei o botão", "Não havia toda a informação necessária", etc...)

Fonte: Própria

Após a conclusão das tarefas, os participantes tiveram liberdade para mexer no módulo proposto.

4.3.2 Questionários

Os questionários foram desenvolvidos para medir a satisfação dos voluntários sobre as duas aplicações, **CAGR** e **PWA**, através de perguntas do questionário **SUS** que foram respondidas utilizando a Escala Likert para calcular o *score* de cada módulo. Com isso, a

autora pôde realizar uma comparação quantitativa entre os *scores* obtidos para avaliar se houve uma melhora na satisfação dos usuários para a **PWA** desenvolvida.

Acerca da primeira avaliação, foram realizadas apenas as 10 perguntas do questionário **SUS** para mensurar a satisfação dos usuários em relação ao módulo do **CAGR**. As perguntas e opções de resposta para essa avaliação podem ser encontradas no apêndice **B**.

Em relação à segunda avaliação, ela foi feita no dia posterior ao da execução das tarefas que foram descritas anteriormente e a motivação para isso foi para dar tempo das notificações serem exibidas aos participantes para poderem avaliar essa nova funcionalidade. Nesse questionário, foram realizadas as mesmas dez perguntas do questionário **SUS** para mensurar a satisfação dos usuários acerca da aplicação proposta e mais os seguintes questionamentos:

- O navegador e o dispositivo (celular, computador, tablet, etc...) que foram utilizados;
- Se o participante conseguiu concluir a tarefa, e, caso não tenha conseguido, o motivo para não ter conseguido;
- Se o participante possui uma sugestão de melhoria para uma determinada funcionalidade ou para o sistema como um todo;
- Se as funcionalidades novas são úteis (1 - Sem utilidade, 5 - Útil);
- Se as informações das notificações, status e posição da vaga são fáceis de entender e objetivas através de afirmações que os voluntários respondiam seguindo a Escala Likert (1 - Discordo fortemente, 5 - Concordo fortemente);
- Se o participante vê valor na liberdade de escolha para instalar ou não.

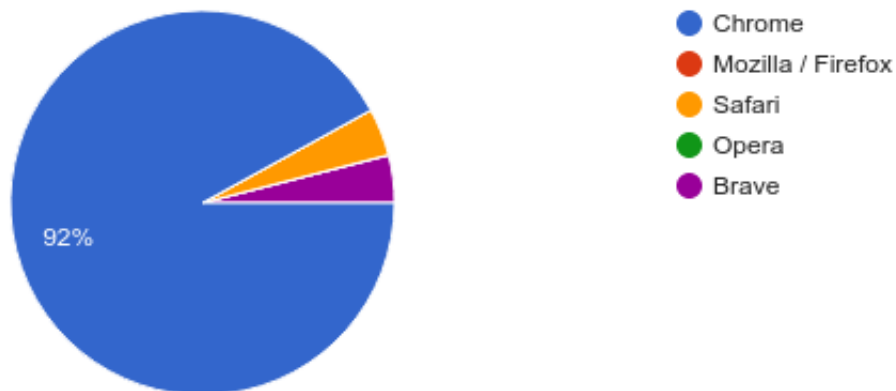
As perguntas e opções de resposta para essa avaliação podem ser encontradas no apêndice **D**. Todas as perguntas associadas às novas funcionalidades propostas para avaliar a utilidade delas só foram respondidas pelos voluntários que concluíram as tarefas relacionadas.

4.3.3 Resultados

Para o primeiro questionário, foram obtidas 30 respostas e um *score* do **SUS** de 50,25, ficando abaixo do recomendado (**SAURO, 2011**), para o módulo de matrícula do **CAGR** enquanto que, para o segundo questionário, 25 dos 30 voluntários realizaram as tarefas e o responderam, alcançando um *score* de 90,01 para o módulo proposto (**PWA**), atingindo um aumento de 79,1%. O *score* de cada módulo foi obtido como a média dos *scores* de cada usuário, cujo cálculo pode ser encontrado na tabela **2**. Ademais, 92% dos voluntários utilizaram o navegador Chrome, que foi uma sugestão da autora para o uso da aplicação já que possui mais suporte para o **PWA**, e apenas 8% realizaram as tarefas do

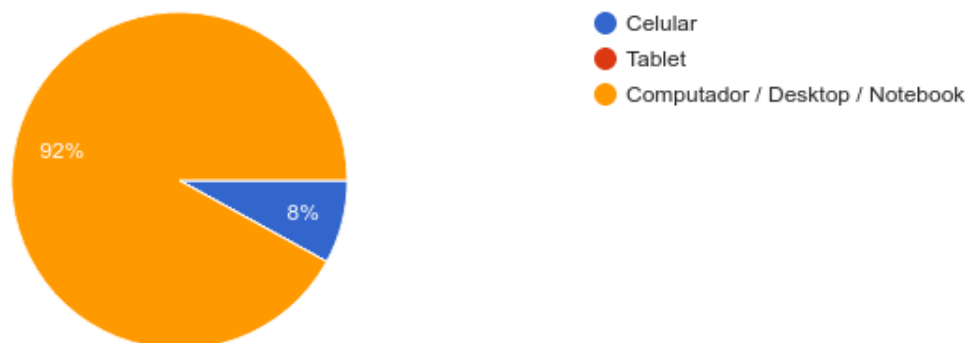
módulo proposto pelo celular enquanto que o público restante utilizou um computador, desktop ou notebook. As figuras 66 e 67 mostram os navegadores e dispositivos utilizados pelos voluntários respectivamente.

Figura 66 – Navegadores usados pelos voluntários do teste



Fonte: Própria

Figura 67 – Dispositivos usados pelos voluntários do teste



Fonte: Própria

Além disso, houve uma melhora em diversas respostas do questionário do SUS para o módulo proposto, como pode ser visto na tabela 3 que mostra a moda das respostas para cada pergunta do questionário para ambos os módulos. A moda foi escolhida em detrimento da média devido ao fato das respostas serem dados nominais. Em relação às perguntas 2 ('Eu acho o módulo desnecessariamente complexo'), 6 ('Eu acho que o módulo apresenta muita inconsistência') e 8 ('Eu acho o módulo confuso de usar') houve uma

redução de 75% na moda, indicando que houve uma redução na complexidade do módulo, ou seja, o sistema, aparentemente, ficou mais simples e intuitivo. Ademais, as perguntas 3 ('Eu acho o módulo fácil de usar'), 5 ('Eu acho que várias funções do módulo estão muito bem integradas'), 7 ('Eu imagino que as pessoas aprendem como usar esse módulo rapidamente') e 9 ('Eu me sinto confiante ao usar esse módulo') também apresentaram um aumento, sendo que a 5 e 7 tiveram uma melhora de 150%. Logo, o módulo proposto deixou os voluntários mais satisfeitos e confortáveis e aparenta possuir uma curva de aprendizado mais curta que a do **CAGR**.

Pergunta	M_{CAGR}	M_{PWA}	Variação percentual (%)
Pergunta 1	5	5	0,0
Pergunta 2	4	1	-75,0
Pergunta 3	3	5	66,7
Pergunta 4	1	1	0,0
Pergunta 5	2	5	150,0
Pergunta 6	4	1	-75,0
Pergunta 7	2	5	150,0
Pergunta 8	4	1	-75,0
Pergunta 9	3	5	66,7
Pergunta 10	1	1	0,0

Tabela 3 – Moda das respostas obtidas para cada pergunta do questionário do **SUS** separadas pelo módulo de matrícula

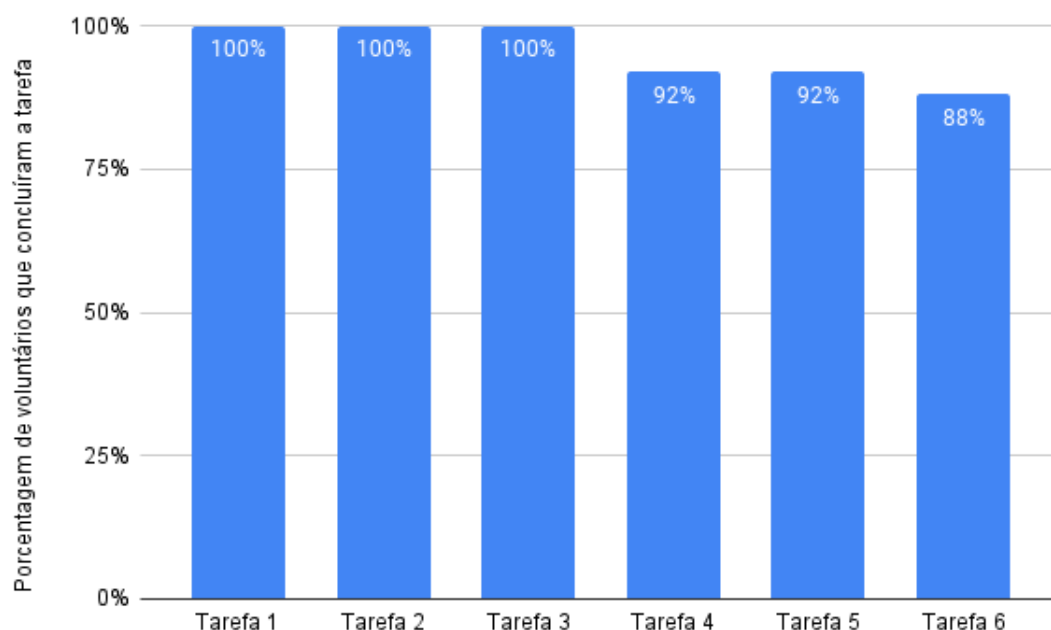
Fonte: Própria

Acerca da realização das tarefas pelos voluntários, a sexta tarefa, relacionada a acessar os dados quando não há conexão com a rede, apresentou a menor taxa de conclusão entre os voluntários que foi de 88%, todavia não houveram comentários para identificar os motivos. A figura 68 mostra as taxas de conclusão para cada tarefa. Em relação às outras tarefas, na quarta tarefa os voluntários expressaram que o motivo para não finalizá-la estava no fato da aplicação comentar que as notificações tinham sido revogadas e os usuários não conseguiam habilitá-las no dispositivo. Enquanto que na quinta, um dos participantes comentou que não tinha o celular em mãos para instalar e o outro relatou que não recebeu um feedback da aplicação e apenas conseguiu acessá-la através do link disponibilizado no documento das tarefas.

4.3.3.1 Utilidade das funcionalidades propostas

Nas próximas seções, serão mostrados a opinião dos voluntários sobre as novas funcionalidades que foram propostas.

Figura 68 – Porcentagem de usuários que conseguiram concluir as atividades por tarefa



Fonte: Própria

4.3.3.1.1 Espelho de matrícula com status e posição da vaga

Relembrando sobre essa funcionalidade, ela permite que o aluno tenha mais informações sobre a sua vaga em uma turma, ou seja, se está na fila de espera e qual a posição em relação aos outros estudantes que solicitaram a mesma turma, para ajudar na tomada de decisão de qual turma solicitar ao editar o pedido de matrícula. A figura 69 mostra as duas perguntas que foram feitas e as respostas disponibilizadas. No geral, os voluntários consideraram ambas as informações úteis, visto que a moda foi de 5 (Útil) para as duas perguntas. As figuras 70 e 71 mostram todas as respostas que foram obtidas.

4.3.3.1.2 Notificações

As notificações foram desenvolvidas para deixar o aluno mais atualizado sobre o status da vaga, se entrou ou saiu da fila de espera, para auxiliar na tomada de decisão também. Apenas 7 voluntários receberam notificações e conseguiram avaliar essa funcionalidade. A figura 72 mostra as perguntas que foram feitas. A maioria considerou as notificações como úteis, visto que a moda foi 5 (Útil), e as mensagens claras e objetivas, já que a moda foi 5 também. As figuras 73 e 74 mostram todas as respostas.

Figura 69 – Perguntas sobre o espelho de matrícula com status e posição da vaga

O quão útil é a informação sobre o status (fila de espera / vaga possível) da vaga ?

1 2 3 4 5

Sem utilidade Útil

O quão útil é a informação sobre a sua posição na vaga ?

1 2 3 4 5

Sem utilidade Útil

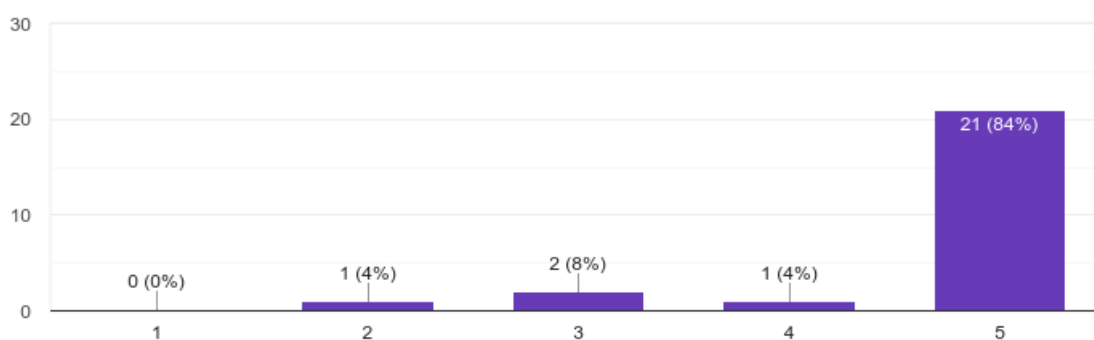
Fonte: Própria

Figura 70 – Respostas sobre a posição da vaga no espelho de matrícula

O quão útil é a informação sobre a sua posição na vaga ?

[Copiar](#)

25 respostas



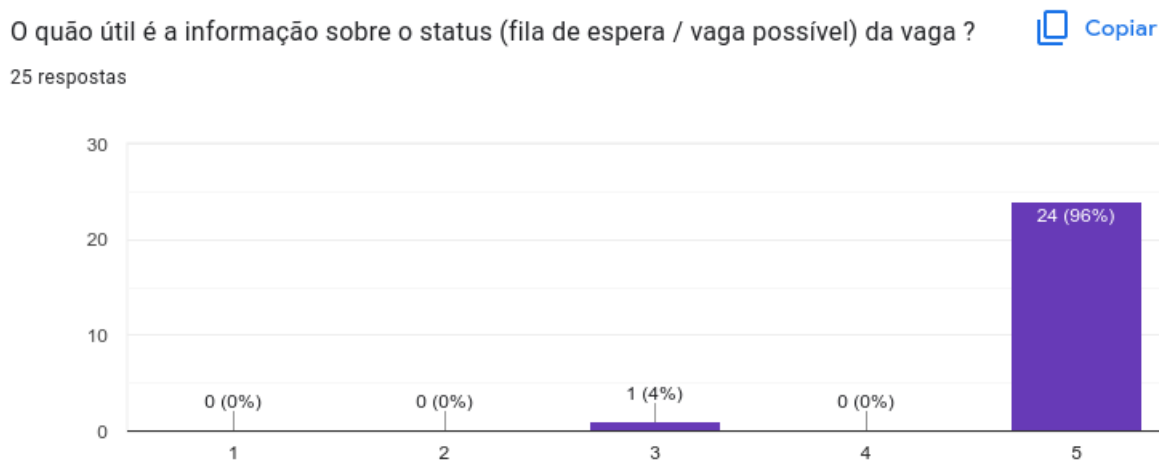
Fonte: Própria

4.3.3.1.3 Instalação

A possibilidade de instalação foi proposta para facilitar o acesso à aplicação e dar uma experiência similar aos aplicativos móveis. A figura 75 mostra as perguntas que foram feitas. Os participantes consideraram útil e gostaram da liberdade de escolha sobre instalar ou não, visto que as modas foram 5 para ambas as perguntas. As figuras 76 e 77 mostram todas as respostas recebidas.

Apesar desses resultados, um ponto que foi levantado por alguns voluntário foi que, devido a sazonalidade do processo de matrícula, a instalação parece não fazer tanto

Figura 71 – Respostas sobre o status da vaga no espelho de matrícula



Fonte: Própria

Figura 72 – Perguntas sobre as notificações

O quão útil é a funcionalidade de notificações?

1 2 3 4 5

Sem utilidade Útil

As mensagens das notificações foram fáceis de entender e objetivas

1 2 3 4 5

Discordo fortemente Concordo fortemente

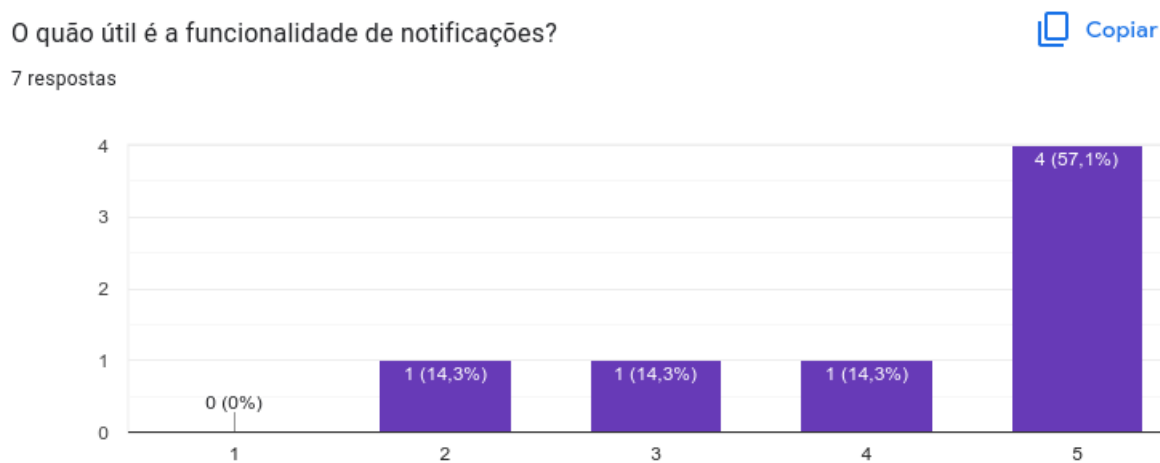
Fonte: Própria

sentido, visto que seria necessário usar a aplicação por um breve período de tempo. A figura 78 mostram alguns comentários que foram feitos sobre essa funcionalidade.

4.3.3.1.4 Visualização do espelho de matrícula e grade de horários independente da conexão com a rede

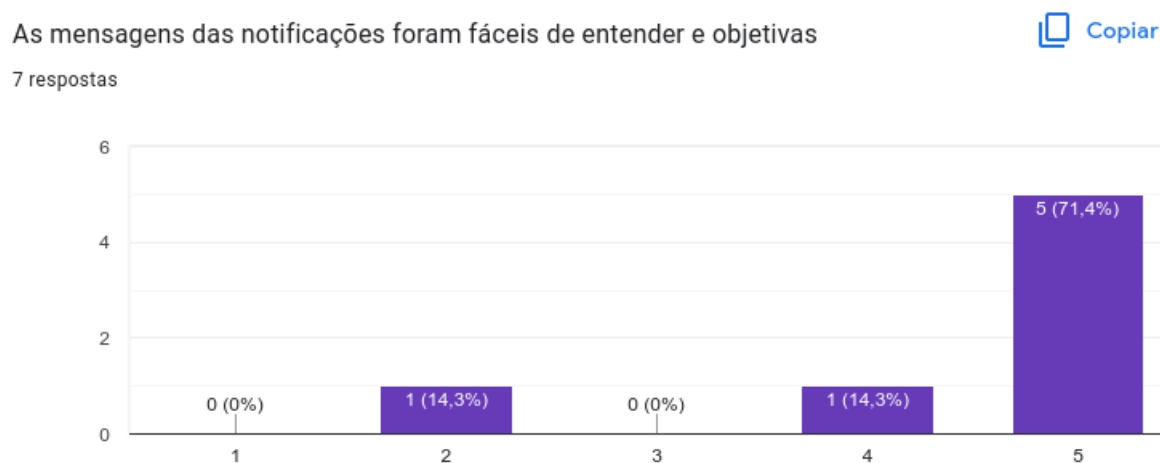
Essa funcionalidade foi criada para permitir que o aluno teve acesso às informações do espelho de matrícula e grade de horários mesmo sem conexão com a rede. As figuras 79 e 80 mostram as perguntas que foram feitas. Os voluntários, em sua maioria, acharam

Figura 73 – Respostas sobre a utilidade das notificações



Fonte: Própria

Figura 74 – Respostas sobre a clareza das mensagens nas notificações



Fonte: Própria

essa funcionalidade útil, as informações no espelho de matrícula e grade de horários claras e objetivas e que a aplicação conseguiu avisar com facilidade sobre a queda da conexão, já que as modas para todas as perguntas foram 5. As figuras 81, 82, 83 e 84 mostram todas as respostas que foram obtidas.

Figura 75 – Perguntas sobre a instalação

O quão útil é a funcionalidade de instalar?

1 2 3 4 5

Sem utilidade Útil

⋮

Eu vejo valor em ter a escolha de instalar o módulo no dispositivo ou não

1 2 3 4 5

Discordo fortemente Concordo fortemente

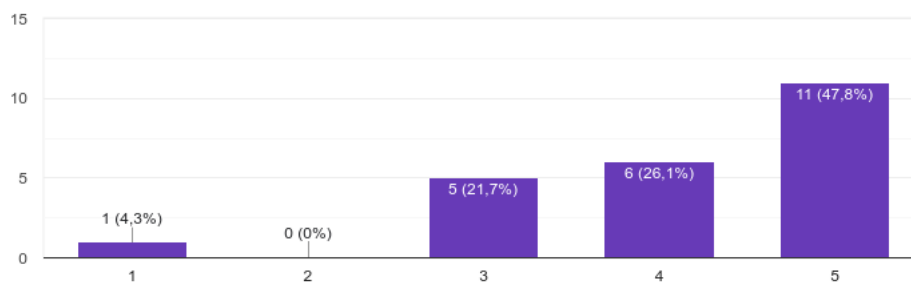
Fonte: Própria

Figura 76 – Respostas sobre a utilidade da instalação

O quão útil é a funcionalidade de instalar?

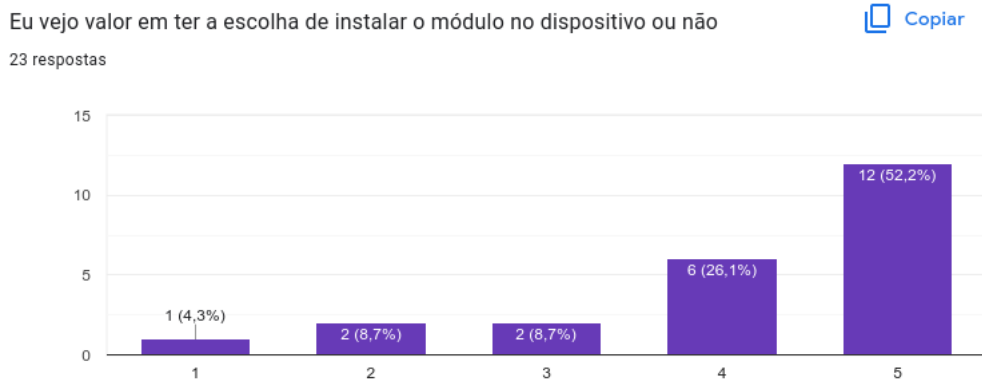


23 respostas



Fonte: Própria

Figura 77 – Respostas sobre o valor da possibilidade de instalação



Fonte: Própria

Figura 78 – Alguns dos comentários sobre a instalação

Tem alguma sugestão de melhoria ou comentário sobre essa funcionalidade ?

6 respostas

- No desktop, a presença da barra superior deixou o uso confortável, além de ter ficado muito bonita hehe
- Nessa tarefa foi instalado a aplicação no desktop, o que não vejo tanta utilidade. Mas também foi tentado instalá-la no celular, mas não foi possível encontrar o botão. Creio que a utilidade da aplicação no celular seja maior, porém sazonal.
- Foi fácil de encontrar o botão de instalar, a aplicação instalou e abriu rapidamente. Porém acho que um aplicativo apenas de matrícula não é muito interessante, ainda mais que existe uma versão de site web que possui as mesmas funcionalidades, não instalaria ele em outras circunstâncias.
- A funcionalidade está bem implementada, e funcionando perfeitamente. Particularmente, acredito que não instalaria no dispositivo, usaria apenas o website, por preferência pessoal mesmo.
- Instalação no celular é interessante, mas instalação no desktop não senti grande impacto, visto que o acesso ao navegador é simples (inclusive, normalmente ele está sempre aberto no desktop).
- nao acho tao necessario essa funcionalidade, mas contando que da pra ver as coisas offline é interessante

Fonte: Própria

Figura 79 – Perguntas sobre o espelho de matrícula

O quão útil é a funcionalidade de visualizar o pedido de matrícula estando sem internet?

	1	2	3	4	5	
Sem utilidade	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Útil

As informações no espelho de matrícula são suficientes e objetivas

	1	2	3	4	5	
Discordo fortemente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo fortemente

Fonte: Própria

Figura 80 – Perguntas sobre a grade de horários e a interface da aplicação desenvolvida quando há perda de conexão com a rede

As informações na grade de horários são suficientes e objetivas

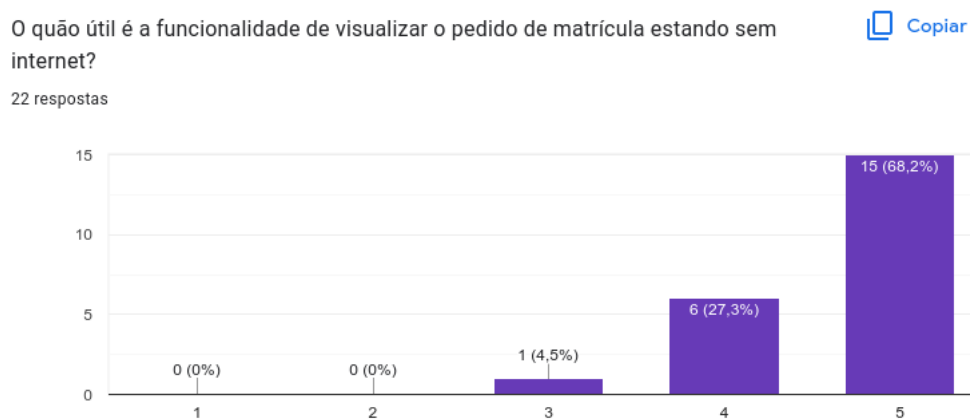
	1	2	3	4	5	
Discordo fortemente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo fortemente

Eu consegui identificar facilmente quando estava sem conexão com a internet pela aplicação

	1	2	3	4	5	
Discordo fortemente	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Concordo fortemente

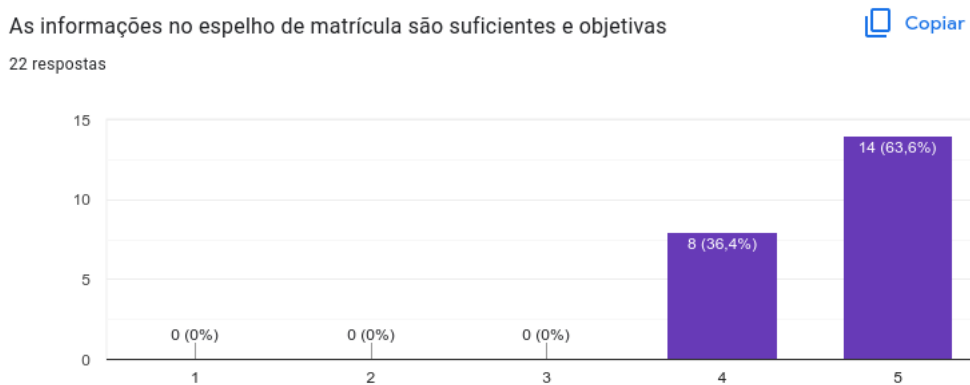
Fonte: Própria

Figura 81 – Respostas sobre a utilidade de ver as informações estando sem conexão com a rede



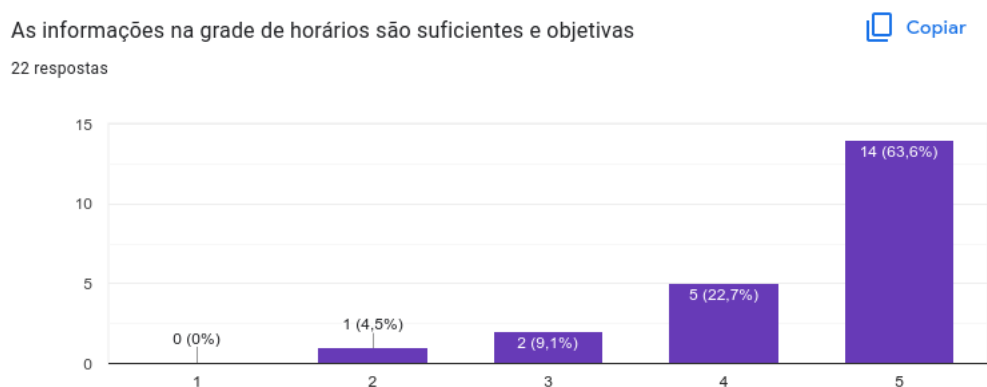
Fonte: Própria

Figura 82 – Respostas sobre a clareza das informações no espelho de matrícula



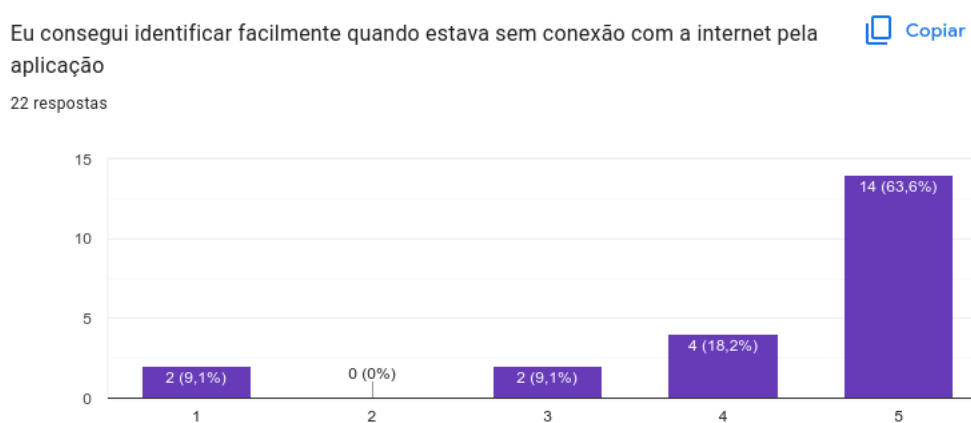
Fonte: Própria

Figura 83 – Respostas sobre a clareza das informações na grade de horários



Fonte: Própria

Figura 84 – Respostas sobre a facilidade de identificar a falta de conexão pela aplicação



Fonte: Própria

5 CONCLUSÃO E TRABALHOS FUTUROS

Em relação ao objetivo de desenvolver uma **PWA** com React e o **Bold Design System**, observa-se que ele foi atingido e permitiu que os alunos realizassem as operações básicas em um processo de matrícula como buscar as turmas e solicitar e ajustar o pedido de matrícula. Apesar disso, a aplicação não contemplou todos os requisitos oficiais que foram propostos no anexo **A**, como, por exemplo, a opção de filtrar as turmas por semestre corrente, remover turmas aprovadas e cujos pré-requisitos ainda não foram atendidos e exibir ou ocultar disciplinas de outros cursos, extracurriculares e sem pré-requisitos. Esse impedimento ocorreu devido ao fato da autora precisar se dedicar no desenvolvimento de um servidor já que houveram atrasos na implementação da **API** oficial por parte da **SeTIC** para conseguir realizar o teste de usabilidade dentro do prazo.

Em relação ao objetivo de desenvolver o servidor com escopo reduzido para que a autora conseguisse elaborar o teste de usabilidade, constata-se que ele foi concluído com sucesso. No entanto, ressalta-se que esse servidor não atende aos requisitos oficiais presentes no anexo **A** porque não era o objetivo desse projeto, visto que, inicialmente, a **PWA** utilizaria a **API** desenvolvida pela **SeTIC**, e não deve ser colocado em um ambiente de produção devido aos problemas de segurança que foram relatados na seção **4**. Esse mesmo comentário vale para a **PWA** desenvolvida.

Acerca do objetivo de elaborar um roadmap para que outros alunos consigam desenvolver uma **PWA**, é possível constatar que foi atingido com ressalvas. Esse roadmap só poderia ser utilizado para desenvolvedores que queiram implementar uma **PWA** com React que recebe notificações tanto em primeiro como segundo plano através da infraestrutura oferecida pela **FCM** e que é instalável no Chrome e acessa dados pela cache do service worker. Além disso, permite que desenvolvam um servidor com Spring que envie as notificações através da **FCM**. Acerca do Web Push Protocol, não foi explicado na prática como utilizá-lo, no entanto, foi descrito em um nível mais abstrato para permitir a compreensão das entidades envolvidas e suas comunicações.

No que se refere à meta de melhorar a satisfação dos usuários, ela foi alcançada e pode ser evidenciada através do aumento de 79,1% no score do **SUS** para a **PWA** proposta em relação ao módulo do **CAGR**. Além disso, as perguntas 2 ('Eu acho o módulo desnecessariamente complexo'), 6 ('Eu acho que o módulo apresenta muita inconsistência') e 8 ('Eu acho o módulo confuso de usar') apresentaram uma redução de 75% nas modas das respostas da **PWA** em comparação com as do **CAGR**. No caso das perguntas 5 ('Eu acho que várias funções do módulo estão muito bem integradas') e 7 ('Eu imagino que as pessoas aprendem como usar esse módulo rapidamente') apresentaram uma melhora de 150% nas modas das respostas da **PWA** em comparação com as do **CAGR**. Contudo, devido a uma amostragem muito pequena, não é possível inferir com um alto grau de certeza que todos os alunos da UFSC ficariam mais satisfeitos com essa aplicação e que

essa [PWA](#) possui um bom suporte para outros navegadores, como Mozilla e Safari, e que traria uma boa experiência para usuários de celular devido ao fato de que apenas 8% dos voluntários realizaram as tarefas com um celular.

A respeito das [PWAs](#), constatou-se que elas possuem diversos recursos, como as notificações, instalação e controle das requisições através do service worker, que poderiam ser aproveitados no desenvolvimento de sistemas para melhorar a experiência do usuário, como foi observado nos os resultados da pesquisa desse projeto. Todavia, essa tecnologia possui algumas limitações dependendo do navegador ou dispositivo devido ao suporte incremental que é dado pelas empresas e instituições, podendo reduzir a qualidade da experiência do usuário ao impedir que realize uma determinada tarefa, conforme foi constatado durante o teste de usabilidade do [PWA](#) do módulo de matrícula e no desenvolvimento da funcionalidade de instalação com o navegador Mozilla.

Por fim, as notificações, status e posição da vaga no espelho de matrícula e visualização do espelho e da grade de horários mesmo sem conexão com a rede foram as funcionalidades que foram vistas com mais valor pelos participantes do teste de usabilidade e que poderiam ser implementadas no módulo oficial. Por outro lado, a instalação, devido a sazonalidade do processo de matrícula, foi considerada menos útil pelos voluntários, como foi observado nos comentários enviados no segundo questionário. Em vista disso, o desenvolvimento de aplicações nativas ou investimento na experiência de instalação para um [PWA](#) parece não ser tão interessante para esse tipo de sistema, já que os usuários não pretendem instalá-lo.

Como trabalhos futuros, são propostas as seguintes tarefas:

- Integrar com o sistema de matrícula que está sendo desenvolvido pela [SeTIC](#);
- Avaliar e implementar as melhorias que foram propostas pelos voluntários da pesquisa;
- Contemplar todos os requisitos propostos no anexo [A](#);
- Elaborar uma análise estatística mais aprofundada sobre os dados que foram obtidas no teste de usabilidade;
- Verificar a questão do suporte em outros dispositivos e navegadores.

REFERÊNCIAS

ABNT. **NBR 9241-11**. Rio de Janeiro, 2002.

ANDRADE, Ana Paula de. **O que é Firebase?** [S.l.], 2020. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-firebase>. Acesso em: 7 nov. 2022.

ARCHIBALD, Jake. **Introducing Background Sync**. [S.l.], 2019. Disponível em: <https://developers.google.com/web/updates/2015/12/background-sync>. Acesso em: 1 mai. 2022.

BARBOZA, Anderson. **Medindo a usabilidade do seu produto com System Usability Scale (SUS)**. [S.l.], 2019. Disponível em: <https://medium.com/design-contaazul/medindo-a-usabilidade-do-seu-produto-com-system-usability-scale-sus-3956612d9229>. Acesso em: 4 nov. 2022.

BRIDGE, Laboratório. **Bold Design System**. [S.l.]. Disponível em: <https://bold.bridge.ufsc.br/pt/>. Acesso em: 1 jan. 2022.

BROOKE, John. SUS: A quick and dirty usability scale. **Usability Eval. Ind.**, v. 189, nov. 1995.

CHARLAND, Andre; LEROUX, Brian. Mobile Application Development: Web vs. Native: Web Apps Are Cheaper to Develop and Deploy than Native Apps, but Can They Match the Native User Experience? **Queue**, Association for Computing Machinery, New York, NY, USA, v. 9, n. 4, p. 20–28, abr. 2011. ISSN 1542-7730. DOI: [10.1145/1966989.1968203](https://doi.org/10.1145/1966989.1968203). Disponível em: <https://doi.org/10.1145/1966989.1968203>.

CORRAL, Luis *et al.* Evolution of Mobile Software Development from Platform-Specific to Web-Based Multiplatform Paradigm. *In: PROCEEDINGS of the 10th SIGPLAN Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*. Portland, Oregon, USA: Association for Computing Machinery, 2012. (Onward! 2011), p. 181–183. DOI: [10.1145/2048237.2157457](https://doi.org/10.1145/2048237.2157457). Disponível em: <https://doi.org/10.1145/2048237.2157457>.

CURSO DE CIÊNCIAS DA COMPUTAÇÃO DA UFSC, Secretaria do. **Processo de Matrícula para Veteranos**. [S.l.]. Disponível em: <https://cco.ufsc.br/sobre-o-curso/processo-de-matricula-para-veteranos/>. Acesso em: 1 jan. 2022.

CURSO DE GRADUAÇÃO EM LETRAS – LÍNGUA PORTUGUESA E LITERATURAS DE LÍNGUA PORTUGUESA DA UFSC, Coordenadoria do. **Manual de Matrículas**. [S.l.]. Disponível em: <https://letrasportugues.grad.ufsc.br/manual-de-matriculas/>. Acesso em: 1 jan. 2022.

CURSOS DE GRADUAÇÃO EM EDUCAÇÃO FÍSICA DA UFSC, Coordenadoria dos. **Matrículas – Renovação via CAGR**. [S.l.]. Disponível em: <https://edfisica.grad.ufsc.br/renovacao-de-matriculas/>. Acesso em: 1 jan. 2022.

FIREBASE. **Adicione o SDK Admin do Firebase ao seu servidor**. [S.l.], 2022. Disponível em: <https://firebase.google.com/docs/admin/setup?hl=pt-br>. Acesso em: 19 nov. 2022.

_____. **Firestore Cloud Messaging**. [S.l.], 2020. Disponível em: <https://firebase.google.com/docs/cloud-messaging?hl=pt-br>. Acesso em: 7 nov. 2022.

_____. **Sobre as mensagens do FCM**. [S.l.], 2022. Disponível em: <https://firebase.google.com/docs/cloud-messaging/concept-options>. Acesso em: 16 nov. 2022.

GAUNT, Matt. **Service Workers: an Introduction**. [S.l.], 2021. Disponível em: <https://developers.google.com/web/fundamentals/primers/service-workers>. Acesso em: 1 mai. 2022.

JONES, M.; BRADLEY, J.; SAKIMURA, N. **JSON Web Token (JWT)**. [S.l.], mai. 2015. <http://www.rfc-editor.org/rfc/rfc7519.txt>. Disponível em: <http://www.rfc-editor.org/rfc/rfc7519.txt>.

KEMP, Simon. **Digital 2022: Global Overview Report**. [S.l.], 2022. Disponível em: <https://datareportal.com/reports/digital-2022-global-overview-report>. Acesso em: 1 mar. 2022.

LEPAGE, Pete. **Como fornecer sua própria experiência de instalação in-app**. [S.l.], 2021. Disponível em: <https://web.dev/i18n/pt/customize-install/>. Acesso em: 19 nov. 2022.

_____. **Storage for the web**. [S.l.], 2022. Disponível em: <https://web.dev/storage-for-the-web/>. Acesso em: 1 jul. 2022.

_____. **What does it take to be installable?** [S.l.], 2022. Disponível em: <https://web.dev/install-criteria/>. Acesso em: 19 nov. 2022.

LIMA, Cleyson. **O que é JWT?** [S.l.], 2021. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-jwt>. Acesso em: 5 nov. 2022.

MALAVOLTA, Ivano. Beyond Native Apps: Web Technologies to the Rescue! (Keynote). *In: PROCEEDINGS of the 1st International Workshop on Mobile Development*. Amsterdam, Netherlands: Association for Computing Machinery, 2016. (Mobile! 2016),

p. 1–2. DOI: [10.1145/3001854.3001863](https://doi.org/10.1145/3001854.3001863). Disponível em:
<https://doi.org/10.1145/3001854.3001863>.

MATRUFSC2. **Rematrícula na UFSC: Etapas e Procedimentos**. [S.l.]. Disponível em: https://matrufsc2.appspot.com/ajuda/artigo/rematricula-na-ufsc-etapas-e-procedimentos/VO_TNSkAALQmIh1C/. Acesso em: 1 jan. 2022.

_____. **Sobre**. [S.l.]. Disponível em:
<https://matrufsc2.appspot.com/sobre/VPCrvCkAAPE-I8ch>. Acesso em: 1 jan. 2022.

MOZILLA. **API de Armazenamento na Web**. [S.l.], 2022. Disponível em:
https://developer.mozilla.org/pt-BR/docs/Web/API/Web_Storage_API. Acesso em: 19 nov. 2022.

_____. **API Push**. [S.l.], 2022. Disponível em:
https://developer.mozilla.org/pt-BR/docs/Web/API/Push_API. Acesso em: 1 nov. 2022.

_____. **Cache**. [S.l.], 2022. Disponível em:
<https://developer.mozilla.org/en-US/docs/Web/API/Cache>. Acesso em: 1 jul. 2022.

_____. **Começando com React**. [S.l.], 2022. Disponível em:
https://developer.mozilla.org/pt-BR/docs/Learn/Tools_and_testing/Client-side_JavaScript_frameworks/React_getting_started. Acesso em: 3 nov. 2022.

_____. **IndexedDB key characteristics and basic terminology**. [S.l.], 2022. Disponível em: https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API/Basic_Terminology. Acesso em: 1 jul. 2022.

_____. **Introdução Express/Node**. [S.l.], 2022. Disponível em:
https://developer.mozilla.org/pt-BR/docs/Learn/Server-side/Express_Nodejs/Introduction. Acesso em: 5 nov. 2022.

_____. **Introduction to progressive web apps**. [S.l.], 2022. Disponível em: https://developer.mozilla.org/en-US/docs/Web/Progressive_web_apps/Introduction. Acesso em: 1 mar. 2022.

_____. **Notifications API**. [S.l.], 2022. Disponível em:
https://developer.mozilla.org/en-US/docs/Web/API/Notifications_API. Acesso em: 1 nov. 2022.

_____. **ServiceWorkerRegistration**. [S.l.], 2022. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/API/ServiceWorkerRegistration>. Acesso em: 1 nov. 2022.

- MOZILLA. **Web app manifests**. [S.l.], 2022. Disponível em: <https://developer.mozilla.org/en-US/docs/Web/Manifest>. Acesso em: 14 ago. 2022.
- OKTA. **Introduction to JSON Web Tokens**. [S.l.], 2022. Disponível em: <https://jwt.io/introduction>. Acesso em: 5 nov. 2022.
- OLIVEIRA, Robisson. **Como o JavaScript funciona: Service Workers, seu ciclo de vida e casos de uso**. [S.l.], 2019. Disponível em: <https://medium.com/reactbrasil/como-o-javascript-funciona-service-workers-seu-ciclo-de-vida-e-casos-de-uso-8404870ee728>. Acesso em: 1 jun. 2022.
- OSMANI, Addy. **Getting started with Progressive Web Apps**. [S.l.], 2015. Disponível em: <https://developer.chrome.com/blog/getting-started-pwa/>. Acesso em: 1 mar. 2022.
- PETE LEPAGE THOMAS STEINER, François Beaufort. **Add a web app manifest**. [S.l.], 2022. Disponível em: <https://web.dev/add-manifest/>. Acesso em: 14 ago. 2022.
- REACT. **Context - React**. [S.l.], 2022. Disponível em: <https://reactjs.org/docs/context.html>. Acesso em: 15 nov. 2022.
- REENSKAUG, Trygve. **Thing-model-view-editor-an example from a planning system. technical note, Xerox Parc**, 1979.
- RODRIGUES, Fabio de Faria Tolentino. **Redesign do SIGAA acadêmico módulo aluno: uma abordagem centrada no usuário**, 2020.
- SAM RICHARD, Pete LePage. **What are Progressive Web Apps?** [S.l.], 2020. Disponível em: <https://web.dev/what-are-pwas/>. Acesso em: 1 mar. 2022.
- SAURO, Jeff. **Measuring Usability with the System Usability Scale (SUS)**. [S.l.], 2011. Disponível em: <https://measuringu.com/sus/>. Acesso em: 4 nov. 2022.
- SECRETARIA INTEGRADA DE GRADUAÇÃO (SIG) DO CENTRO DE CIÊNCIAS, Tecnologias e Saúde da UFSC. **Matrícula de Veteranos**. [S.l.]. Disponível em: <https://sig.ararangua.ufsc.br/manual-de-matricula/>. Acesso em: 1 jan. 2022.
- SPRING. **Spring - Why Spring ?** [S.l.], 2022. Disponível em: <https://spring.io/why-spring>. Acesso em: 3 nov. 2022.
- STATCOUNTER. **Desktop vs Mobile vs Tablet Market Share Worldwide**. [S.l.], 2022. Disponível em: <https://gs.statcounter.com/platform-market-share/desktop-mobile-tablet/worldwide/#monthly-201201-202201>. Acesso em: 1 mar. 2022.

STATCOUNTER. **Mobile Operating System Market Share Worldwide**. [S.l.], 2022. Disponível em: <https://gs.statcounter.com/os-market-share/mobile/worldwide/#monthly-200901-202206-bar>. Acesso em: 5 jun. 2022.

TEIXEIRA, Fabricio. **O que é o SUS (System Usability Scale) e como usá-lo em seu site**. [S.l.], 2015. Disponível em: <https://brasil.uxdesign.cc/o-que-%C3%A9-o-sus-system-usability-scale-e-como-us%C3%A1-lo-em-seu-site-6d63224481c8>. Acesso em: 4 nov. 2022.

THOMSON, M. **Message Encryption for Web Push**. [S.l.], nov. 2017.

THOMSON, M.; BEVERLOO, P. **Voluntary Application Server Identification (VAPID) for Web Push**. [S.l.], nov. 2017.

THOMSON, M.; DAMAGGIO, E.; RAYMOR, B. **Generic Event Delivery Using HTTP Push**. [S.l.], dez. 2016.

THOMSON, Martin; CACERES, Marcos; BEVERLOO, Peter. **Push API**. [S.l.], jun. 2022. <https://www.w3.org/TR/2022/WD-push-api-20220630/>.

TREINAWEB. **Spring Boot: Tudo que você precisa saber!** [S.l.], 2022. Disponível em: <https://blog.geekhunter.com.br/tudo-o-que-voce-precisa-saber-sobre-o-spring-boot/>. Acesso em: 3 nov. 2022.

WIKIPÉDIA. **Model–view–controller**. [S.l.], 2011. Disponível em: <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller>. Acesso em: 4 nov. 2022.

Apêndices

APÊNDICE A – TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

O (A) Senhor (a) está sendo convidado (a) a participar de uma pesquisa. Por favor, leia este documento com bastante atenção antes de assiná-lo. Caso haja alguma palavra ou frase que o (a) senhor (a) não consiga entender, converse com o pesquisador responsável pelo estudo ou com um membro da equipe desta pesquisa para esclarecê-los.

A proposta deste termo de consentimento livre e esclarecido (TCLE) é explicar tudo sobre o estudo e solicitar a sua permissão para participar do mesmo. O objetivo desta pesquisa é avaliar o novo módulo que foi desenvolvido para a etapa de matrícula da Universidade Federal de Santa Catarina (UFSC) e tem como justificativa permitir uma análise comparativa entre o módulo proposto e o oficial, que está presente no Sistema de Controle Acadêmico da Graduação (CAGR).

Se o(a) Sr.(a) aceitar participar da pesquisa, os procedimentos envolvidos em sua participação são os seguintes: Avaliação do módulo oficial através de um questionário com dez (10) perguntas que deve durar cerca de dez (10) minutos, execução de seis (6) tarefas no novo módulo para simular tarefas reais, que serão enviadas por e-mail, e avaliação das tarefas que foram realizadas e do novo módulo através de outro questionário que deve durar de 20 a 30 minutos. Essa última avaliação será feita no dia posterior à realização das tarefas. Todas as etapas da pesquisa serão feitas de forma remota e assíncrona e um e-mail será enviado em cada etapa com as instruções necessárias. Além disso, todos os questionários serão respondidos de forma anônima para assegurar a sua privacidade, sigilo e confidencialidade.

Toda pesquisa com seres humanos envolve algum tipo de risco. No nosso estudo, os possíveis riscos ou desconfortos decorrentes da participação na pesquisa são desconforto ou estresse por não conseguir concluir uma tarefa e, caso isso aconteça, anote no documento de tarefas que não foi possível concluir junto com o motivo para não ter concluído (ex: não achar o botão ou ocorrer um erro) e passe para a próxima tarefa. Outro risco seria a impossibilidade de acessar a aplicação para as cinco primeiras tarefas devido à falta de conexão com a internet e outros possíveis problemas, e, nesse caso, entre em contato com a pesquisadora assistente, Maria Eduarda de Melo Hang (maira.eduarda.melo.hang@gmail.com), para avisar sobre a impossibilidade de realizar as tarefas. Por fim, há um risco de quebra de sigilo das informações, mesmo que involuntária e não intencional, podendo causar constrangimento aos participantes.

Contudo, esta pesquisa também pode trazer benefícios. Os possíveis benefícios resultantes da participação na pesquisa são possuir uma avaliação feita pelo público-alvo sobre a facilidade de uso e aprendizado dos dois módulos, permitir que os pesquisadores comparem os dois módulos para verificar se houve uma melhora na usabilidade no módulo proposto e conferir se as novas funcionalidades que foram propostas são realmente úteis para os alunos da UFSC. Todas as informações poderão ser repassadas para a instituição

responsável pelo módulo oficial para melhorá-lo, tornando-o mais fácil e simples de utilizar.

Sua participação na pesquisa é totalmente voluntária, ou seja, não é obrigatória. Caso o(a) Sr.(a) decida não participar, ou ainda, desistir de participar e retirar seu consentimento durante a pesquisa, não haverá nenhum prejuízo à avaliação curricular que você recebe ou possa vir a receber na instituição. Não haverá um retorno sobre os resultados obtidos para os participantes e não será possível realizar a exclusão dos dados da pesquisa durante o processo de registro / consentimento devido ao fato das respostas aos formulários serem anônimas.

Não está previsto nenhum tipo de pagamento pela sua participação na pesquisa e o(a) Sr.(a) não terá nenhum custo com respeito aos procedimentos envolvidos, porém, poderá receber por despesas decorrentes de sua participação. Essas despesas serão pagas pelo orçamento da pesquisa. Caso ocorra algum problema ou dano com o(a) Sr.(a), resultante de sua participação na pesquisa, o(a) Sr.(a) receberá todo o atendimento necessário, sem nenhum custo pessoal e garantimos indenização diante de eventuais fatos comprovados, com nexos causal com a pesquisa.

É assegurada a assistência durante toda pesquisa, bem como é garantido ao Sr.(a), o livre acesso a todas as informações e esclarecimentos adicionais sobre o estudo e suas consequências, enfim, tudo o que o(a) Sr.(a) queira saber antes, durante e depois da sua participação. Além disso, é assegurada sua privacidade, sigilo e confidencialidade durante todas as fases da pesquisa.

Caso o(a) Sr.(a) tenha dúvidas, poderá entrar em contato com o pesquisador responsável Ricardo Pereira e Silva, pelo telefone (48) 3721-7561, endereço UFSC, Campus David Ferreira Lima, Centro Tecnológico, Depto de Informática e Estatística, sala 520 e/ou pelo e-mail ricardo.silva@ufsc.br, com o pesquisador Maria Eduarda de Melo Hang pelo telefone (48) 9-8443-4140 e pelo e-mail maria.eduarda.melo.hang@gmail.com ou com o Comitê de Ética em Pesquisa com Seres Humanos (CEPSH-UFSC - COMITÊ DE ÉTICA EM PESQUISA COM SERES HUMANOS; HOSPITAL UNIVERSITÁRIO PROF. EDGARD SANTOS- UFBA. Endereço: PRÉDIO REITORIA II, R: DESEMBARGADOR VITOR LIMA, Nº 222, SALA 701, TRINDADE, FLORIANÓPOLIS/SC - CEP: 88.040-400; FONE: (48) 3721-6094 / E-MAIL: cep.propesq@contato.ufsc.br. O CEPSH é um órgão colegiado interdisciplinar, deliberativo, consultivo e educativo, vinculado à Universidade Federal de Santa Catarina, mas independente na tomada de decisões, criado para defender os interesses dos participantes da pesquisa em sua integridade e dignidade e para contribuir no desenvolvimento da pesquisa dentro de padrões éticos.

Esse Termo é assinado em duas vias digitais, sendo uma do(a) Sr.(a) e a outra para os pesquisadores.

**APÊNDICE B – QUESTIONÁRIO SOBRE O MÓDULO DO SISTEMA DE
CONTROLE ACADÊMICO DA GRADUAÇÃO**

Avaliação do módulo de matrícula do CAGR

Obrigado por colaborar com a pesquisa do projeto !

Nesse projeto, foi desenvolvido um modo de matrícula alternativo com novas funcionalidades para melhorar a experiência dos alunos ao realizar o pedido de matrícula, sendo elas: visualização do status do pedido de uma turma, se está em uma fila de espera ou não, notificações caso saia ou entre na fila de espera e possibilidade de instalar a aplicação. Com isso, é importante mensurar a usabilidade de ambos os módulos de matrícula, do Sistema de Controle Acadêmico da Graduação (CAGR) e do projeto, para realizar uma análise comparativa e verificar se realmente houve uma melhora na usabilidade.

Sabendo disso, essa primeira avaliação terá 10 perguntas para medir a usabilidade do módulo de matrícula do Sistema de Controle Acadêmico da Graduação (CAGR) e, posteriormente, será apresentado outro formulário para avaliar o módulo que está sendo proposto pelo projeto. A imagem abaixo mostra a tela do módulo do CAGR. A avaliação é bem objetiva e deve durar de 5 a 10 minutos.

Semestre: **Etapa: 1**

Matricula:

Curso:

Local:

Plano 1: Plano 2: Plano 3:

	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
07:30						
08:20		INES410		INES410		
09:10		INES410		INES410		
10:10		INES411		INES411	INES411	
11:00		INES411		INES411	INES411	
13:30						
14:20						
15:10		INES408		INES408		
16:20		INES408		INES408		
17:10		INES408		INES408		
18:30						
19:20						
20:20						
21:10						

Ano e Semestre da provável Formatura:

Situação:

Curriculo:

Curriculo	Curso	Geral
INES411 03208A 40	Organização de Computado	
INES410 03208B 30	Programação Concorrente	
INES411 03208B 25	Organização de Computado	
INES411 03208C 25	Organização de Computado	
INES413 04208 40	Gráficos	
INES414 04208 70	Redes de Computadores I	

Disciplinas Solicitadas

INES408 03208A 6
INES410 03208B 4
INES411 03208B 6

Aulas (Min/Max 12/28)

Sobre a sua experiência ao utilizar o módulo de etapa de matrícula do CAGR

1. 1 - Eu pretendo usar esse módulo com frequência

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo fortemente

2. 2 - Eu acho o módulo desnecessariamente complexo

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo fortemente

3. 3 - Eu acho o módulo fácil de usar

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo fortemente

4. 4 - Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o módulo

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo fortemente

5. 5 - Eu acho que várias funções do módulo estão muito bem integradas

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo fortemente

6. 6 - Eu acho que o módulo apresenta muita inconsistência

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo fortemente

7. 7 - Eu imagino que as pessoas aprendem como usar esse módulo rapidamente

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo fortemente

8. 8 - Eu acho o módulo confuso de usar

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo fortemente

9. 9 - Eu me sinto confiante ao usar esse módulo

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo fortemente

10. 10 - Eu precisei aprender várias coisas novas antes de conseguir usar o módulo

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo fortemente

11. Tem alguma sugestão de melhoria ou comentário sobre o sistema como um todo ?

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários

APÊNDICE C – TAREFAS REALIZADAS PELOS VOLUNTÁRIOS

C.1 TAREFA 1

Imagine que a Universidade Federal de Santa Catarina (UFSC) desenvolveu um novo sistema de matrícula e que está no período da solicitação de matrícula. Além disso, você e seu amigo combinaram de ingressar nas mesmas turmas das disciplinas INE5403, INE5402 e EEL5105 para fazer os trabalhos em dupla e que você deveria se matricular em Cálculo 1 (MTM3101) porque passou na prova de proficiência, mas o seu amigo não, então ele precisaria fazer Pré-Cálculo primeiro. Além disso, você não poderia ter aulas na quinta-feira no período matutino (7h até 12h) porque precisa cuidar dos seus irmãos mais novos até o horário da aula deles e um dos veteranos comentou que a Aloy era uma excelente professora e que você deveria pegar todas as turmas que ela ministrasse.

1. Acesse o novo sistema de matrícula que está disponível neste [link](#);
2. Realize o cadastro e a autenticação;
3. Solicite as turmas das disciplinas

C.2 TAREFA 2

Imagine que o seu amigo mandou uma mensagem perguntando quais as turmas que você solicitou no pedido para escolher as mesmas. Além disso, ele perguntou quais eram os professores das turmas para perguntar aos veteranos e quais eram as salas das aulas que terminavam às 11:50 para ver se ficaria muito longe do Restaurante Universitário (RU).

1. Procure pelas informações que foram solicitadas.

C.3 TAREFA 3

Imagine que seu amigo mandou outra mensagem comentando que vocês tinham esquecido da disciplina INE5401 e também perguntou se você também ficou na fila de espera da turma da Aloy porque estava pensando em ir para outra turma.

1. Verifique se você ficou na fila de espera também;
2. Solicite a disciplina que você esqueceu e, se você ficou na fila de espera, solicite a outra turma que não é ministrada pela professora Aloy.

C.4 TAREFA 4

Imagine que você queira receber notificações caso saia ou entre na fila de espera para alterar o pedido de matrícula conforme a necessidade.

1. Habilite as notificações.

Observações:

- Alguns navegadores podem não ter suporte para as notificações. Se não conseguir concluir essa tarefa, anote o motivo e siga para a próxima tarefa;
- Se nenhuma notificação sobre a entrada ou saída de uma turma aparecerem após habilitar, é possível que surjam ao longo do dia conforme outros participantes realizem as tarefas;
- No final deste documento serão passadas instruções para remover a permissão de notificações caso você tenha habilitado;
- No final deste documento serão passadas instruções para remover a permissão de notificações caso você tenha habilitado;
- Para concluir essa tarefa, não é necessário visualizar uma notificação, apenas habilitá-las.

C.5 TAREFA 5

Imagine que você queira ter a aplicação da matrícula no celular ou no desktop como um aplicativo, então você decide instalá-la.

1. Instale a aplicação;
2. Acesse a aplicação que foi instalada.

Observações:

- Alguns navegadores podem não ter suporte para instalar uma aplicação;
- No final deste documento, serão passadas instruções para desinstalar a aplicação.

C.6 TAREFA 6

Imagine que você ficou sem acesso à internet durante uma aula e uma amiga sua perguntou sobre o seu pedido de matrícula porque queria saber em quais matérias vocês fariam juntos e se poderiam ir na feirinha, que acontece todas às quartas, da UFSC às 11h. Ela também perguntou se você pegou alguma matéria com o professor Kratos.

1. Se você estiver com o celular, ative o Modo Avião. Caso esteja pelo computador, desligue o Wi-Fi ou desconecte o cabo de rede;
2. Acesse a aplicação da matrícula;

3. Busque as informações sobre as turmas;
4. Verifique se você tem alguma aula com o professor Kratos;
5. Verifique se você poderá ir na feirinha com ela.

APÊNDICE D – QUESTIONÁRIO SOBRE A INTERFACE PROPOSTA

Formulário - Módulo proposto

Obrigado mais uma vez por colaborar com a pesquisa do projeto !

Nessa segunda avaliação, faremos algumas perguntas para saber o que você achou de cada uma das tarefas que foram realizadas no dia anterior. A avaliação deve durar de 20 a 30 minutos

Primeiro, algumas perguntinhas sobre o que você estava usando na hora de realizar o teste

1. Qual navegador você estava utilizando ?

Marcar apenas uma oval.

- Chrome
- Mozilla / Firefox
- Safari
- Opera
- Outro: _____

2. Qual tipo de dispositivo você estava utilizando ?

Marcar apenas uma oval.

- Celular
- Tablet
- Computador / Desktop / Notebook
- Outro: _____

Pular para a pergunta 3

Questionário sobre as tarefas

Sobre as tarefas que foram realizadas no dia anterior, responda as próximas perguntas

Tarefa
1

Imagine que a Universidade Federal de Santa Catarina (UFSC) desenvolveu um novo sistema de matrícula e que está no período da solicitação de matrícula. Além disso, você e seu amigo combinaram de ingressar nas mesmas turmas das disciplinas INE5403, INE5402 e EEL5105 para fazer os trabalhos em dupla e que você deveria se matricular em Cálculo 1 (MTM3101) porque passou na prova de proficiência, mas o seu amigo não, então ele precisaria fazer Pré-Cálculo primeiro. Além disso, você não poderia ter aulas na quinta-feira no período matutino (7h até 12h) porque precisa cuidar dos seus irmãos mais novos até o horário da aula deles e um dos veteranos comentou que a Aloy era uma excelente professora e que você deveria pegar todas as turmas que ela ministrasse.

1. Acesse o novo sistema de matrícula que está disponível neste [link](#)
2. Realize o cadastro e a autenticação
3. Solicite as turmas das disciplinas

3. *Marcar apenas uma oval.*

- Consegui concluir a tarefa *Pular para a pergunta 5*
- Não consegui concluir a tarefa *Pular para a pergunta 4*

Sobre a Tarefa 1

4. Por que não foi possível concluir a tarefa? Tem alguma sugestão de melhoria?

Tarefa
2

Imagine que o seu amigo mandou uma mensagem perguntando quais as turmas que você solicitou no pedido para escolher as mesmas. Além disso, ele perguntou quais eram os professores das turmas para perguntar aos veteranos e quais eram as salas das aulas que terminavam às 11:50 para ver se ficaria muito longe do Restaurante Universitário (RU).

1. *Procure pelas informações que foram solicitadas*

5. *Marcar apenas uma oval.*

- Consegui concluir a tarefa *Pular para a pergunta 7*
- Não consegui concluir a tarefa *Pular para a pergunta 6*

Sobre a Tarefa 2

6. Por que não foi possível concluir a tarefa? Tem alguma sugestão de melhoria?

Imagine que seu amigo mandou outra mensagem comentando que vocês tinham esquecido da disciplina INE5401 e também perguntou se você também ficou na fila de espera da turma da Aloy porque estava pensando em ir para outra turma.

Tarefa
3

1. *Verifique se você ficou na fila de espera também*
2. *Solicite a disciplina que você esqueceu e, se você ficou na fila de espera, solicite a outra turma que não é ministrada pela professora Aloy*

7. *Marcar apenas uma oval.*

- Consegui concluir a tarefa *Pular para a pergunta 8*
 Não consegui concluir a tarefa *Pular para a pergunta 10*

Sobre o espelho de matrícula

8. O quão útil é a informação sobre o status (fila de espera / vaga possível) da vaga ?

Marcar apenas uma oval.

1 2 3 4 5
Serr Útil

9. O quão útil é a informação sobre a sua posição na vaga ?

Marcar apenas uma oval.

1 2 3 4 5
Serr Útil

Pular para a pergunta 11

Sobre a Tarefa 3

10. Por que não foi possível concluir a tarefa? Tem alguma sugestão de melhoria?

Tarefa
4

Imagine que você queira receber notificações caso saia ou entre na fila de espera para alterar o pedido de matrícula conforme a necessidade.

1. *Habilite as notificações*

11. *Marcar apenas uma oval.*

- Consegui concluir a tarefa *Pular para a pergunta 13*
- Não consegui concluir a tarefa *Pular para a pergunta 12*

Sobre a Tarefa 4

12. Por que não foi possível concluir a tarefa? Tem alguma sugestão de melhoria?

Pular para a pergunta 17

Sobre as notificações da Tarefa 4

13. *Marcar apenas uma oval.*

- Recebi pelo menos uma notificação *Pular para a pergunta 14*
- Não recebi notificações *Pular para a pergunta 17*

Sobre a experiência com as notificações

14. O quão útil é a funcionalidade de notificações?

Marcar apenas uma oval.

1 2 3 4 5

Sem Útil

15. As mensagens das notificações foram fáceis de entender e objetivas

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo fortemente

16. Tem alguma sugestão de melhoria ou comentário sobre essa funcionalidade ?

Tarefa
5

Imagine que você queira ter a aplicação da matrícula no celular ou no desktop como um aplicativo, então você decide instalá-la.

1. *Instale a aplicação*
2. *Acesse a aplicação que foi instalada*

17. *Marcar apenas uma oval.*

- Consegui concluir a tarefa *Pular para a pergunta 19*
- Não consegui concluir a tarefa *Pular para a pergunta 18*

Sobre a Tarefa 5

18. Por que não foi possível concluir a tarefa? Tem alguma sugestão de melhoria?

Pular para a pergunta 22

Sobre a experiência com a instalação

19. O quão útil é a funcionalidade de instalar?

Marcar apenas uma oval.

1 2 3 4 5

Sem Útil

20. Eu vejo valor em ter a escolha de instalar o módulo no dispositivo ou não

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo fortemente

21. Tem alguma sugestão de melhoria ou comentário sobre essa funcionalidade ?

Tarefa
6

Imagine que você ficou sem acesso à internet durante uma aula e uma amiga sua perguntou sobre o seu pedido de matrícula porque queria saber em quais matérias vocês fariam juntos e se poderiam ir na feirinha, que acontece todas às quartas, da UFSC às 11h. Ela também perguntou se você pegou alguma matéria com o professor Kratos.

1. Se você estiver com o celular, ative o Modo Avião. Caso esteja pelo computador, desligue o Wi-Fi ou desconecte o cabo de rede
2. Acesse a aplicação da matrícula
3. Busque as informações sobre as turmas
4. Verifique se você tem alguma aula com o professor Kratos
5. Verifique se você poderá ir na feirinha com ela

22.

Marcar apenas uma oval.

- Consegui concluir a tarefa *Pular para a pergunta 24*
- Não consegui concluir a tarefa *Pular para a pergunta 28*

Sobre a Tarefa 6

23. Por que não foi possível concluir a tarefa? Tem alguma sugestão de melhoria?

Pular para a pergunta 28

Sobre a experiência de visualizar o pedido de matrícula sem internet

24. O quão útil é a funcionalidade de visualizar o pedido de matrícula estando sem internet?

Marcar apenas uma oval.

1 2 3 4 5

Sem Útil

25. As informações no espelho de matrícula são suficientes e objetivas

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo fortemente

26. As informações na grade de horários são suficientes e objetivas

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo fortemente

27. Eu consegui identificar facilmente quando estava sem conexão com a internet pela aplicação

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo fortemente

Sobre a sua experiência ao utilizar o módulo proposto

28. 1 - Eu pretendo usar esse módulo com frequência

Marcar apenas uma oval.

1 2 3 4 5
Disc Concordo fortemente

29. 2 - Eu acho o módulo desnecessariamente complexo

Marcar apenas uma oval.

1 2 3 4 5
Disc Concordo fortemente

30. 3 - Eu acho o módulo fácil de usar

Marcar apenas uma oval.

1 2 3 4 5
Disc Concordo fortemente

31. 4 - Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o módulo

Marcar apenas uma oval.

1 2 3 4 5
Disc Concordo fortemente

32. 5 - Eu acho que várias funções do módulo estão muito bem integradas

Marcar apenas uma oval.

1 2 3 4 5
Disc Concordo fortemente

33. 6 - Eu acho que o módulo apresenta muita inconsistência

Marcar apenas uma oval.

1 2 3 4 5
Disc Concordo fortemente

34. 7 - Eu imagino que as pessoas aprendem como usar esse módulo rapidamente

Marcar apenas uma oval.

1 2 3 4 5
Disc Concordo fortemente

35. 8 - Eu acho o módulo confuso de usar

Marcar apenas uma oval.

1 2 3 4 5
Disc Concordo fortemente

36. 9 - Eu me sinto confiante ao usar esse módulo

Marcar apenas uma oval.

1 2 3 4 5
Disc Concordo fortemente

37. 10 - Eu precisei aprender várias coisas novas antes de conseguir usar o módulo

Marcar apenas uma oval.

1 2 3 4 5

Disc Concordo fortemente

38. Tem alguma sugestão de melhoria ou comentário sobre o sistema como um todo ?

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários

APÊNDICE E – CÓDIGOS FONTES DAS APLICAÇÕES DESENVOLVIDAS NO PROJETO

O código da prova de conceito por ser encontrado [nesse repositório](#).

O código do servidor que foi desenvolvido pode ser encontrado [nesse repositório](#) e o da [PWA](#) [nesse repositório](#).

APÊNDICE F – ARTIGO NO FORMATO SBC

Desenvolvimento de uma Progressive Web Application para o módulo de matrícula da UFSC

Maria Eduarda de Melo Hang¹, Ricardo Pereira e Silva¹

¹Departamento de Informática e Estatística (INE) – Universidade Federal de Santa Catarina (UFSC)
Caixa Postal: 476 - CEP: 88040-370 - Campus Universitário Trindade, Sala 102
Florianópolis, SC - Brasil

maira.eduarda.melo.hang@gmail.com, ricardo.silva@ufsc.br

Abstract. *With the growth of the use of mobile devices and internet access in the world, institutions and companies try to reach this new audience through the development of native applications or web with the Responsive Web Development (RWD) approach. In this scenario, the Progressive Web Application (PWA) emerged, uniting the two approaches. Given this context, the objective of this project was to develop a PWA for the new enrollment module at UFSC to avoid creating two interfaces (desktop and mobile) and to assess whether there was an improvement in user satisfaction in relation to the official. After carrying out a usability test on the proposed interface with 30 volunteers, an improvement was observed in the score of the System Usability Scale (SUS) and an increase in two odd responses from users in the SUS questionnaire. In addition, it was found that most of the proposed new features were well evaluated by most of the volunteers in these tests.*

Resumo. *Com o crescimento do uso de dispositivos móveis e acesso à internet no mundo, as instituições e empresas tentam atingir esse novo público através do desenvolvimento de aplicativos nativos ou web com a abordagem Responsive Web Development (RWD). Nesse cenário, a Progressive Web Application (PWA) surgiu, unindo as duas abordagens. Diante desse contexto, o objetivo desse projeto foi desenvolver uma PWA para o novo módulo de matrícula da UFSC para evitar a criação de duas interfaces (desktop e mobile) e avaliar se houve uma melhoria na satisfação do usuário em relação à oficial. Após a realização de um teste de usabilidade na interface proposta com 30 voluntários, observou-se uma melhora no score do System Usability Scale (SUS) um aumento em duas respostas ímpares dos usuários no questionário SUS. Além disso, constatou-se que a maioria das novas funcionalidades propostas foram bem avaliadas pela maioria dos voluntários desses testes.*

1. Introdução

Desde 2021, a quantidade de usuários na internet teve um aumento de 192 milhões e atingiu um valor de 4,95 bilhões de usuários, representando cerca de 62,5% da população mundial [Kemp 2022]. Simultaneamente, a quantidade de usuários únicos com celular aumentou em 95 milhões e atingiu um valor de 5,31 bilhões, representando cerca de 67,1% da população mundial [Kemp 2022].

Apesar do número de usuários de celulares ter aumentado significativamente, os computadores ainda possuem cerca de 42,54% das visitas em sites, representando

uma parcela significativa nos acessos, enquanto que os celulares possuem 54,98% [Statcounter 2022]. Com isso, as empresas que desenvolvem softwares possuem novos usuários a serem alcançados com diferentes formas de acessar a aplicação e o seguinte questionamento surge: como alcançar todos esses usuários com um único código e manter uma boa experiência do usuário ao utilizar a aplicação.

Como as aplicações web com a abordagem RWD acabam não trazendo a melhor experiência aos usuários de celulares, já que não possuem um desempenho superior aos aplicativos nativos e acesso a alguns recursos disponíveis nos dispositivos móveis, como notificações, localização, fotos e câmera [Malavolta 2016], essa abordagem pode se tornar inadequada para as empresas, apesar de conseguir atingir todos os usuários com um único código. Da mesma forma, as aplicações nativas, que possuem acesso aos recursos dos dispositivos e uma performance superior às aplicações web, podem custar caro caso a empresa queira desenvolver a aplicação para todas as plataformas disponíveis no mercado, visando atingir todos os usuários possíveis. Esse alto custo é causado pelos seguintes motivos [Malavolta 2016, Corral et al. 2012]:

- intransponibilidade entre as plataformas, impedindo a reutilização de código e, por consequência, criando código redundante nos projetos;
- treinamentos sobre as plataformas e linguagens, podendo ser necessário aprender até sete linguagens diferentes [Charland and LeRoux 2011] caso haja necessidade por parte da empresa;
- manutenção e testes de projetos com código redundante.

Para contornar esses problemas, o conceito de PWA foi criado pela Google, unindo a experiência das aplicações nativas ao usuário com a independência de plataforma das aplicações web, permitindo que usuário utilize a aplicação sem conexão, receba notificações e instale a aplicação no celular [Sam Richard 2020].

2. Metodologia

O projeto pode ser dividido em 5 fases que serão descritas nas subseções seguintes.

2.1. Fase 1 - Análise do contexto atual

Nessa fase, foi feita uma busca por aplicações que são utilizadas pelos alunos, sejam elas oficiais ou não para observar as principais funcionalidades propostas e como cada aplicação aborda e soluciona os problemas associados ao pedido de matrícula. Além disso, foi lido o documento de requisitos do Sistema Integrado de Gestão de Atividades Acadêmicas (SIGAA), que está sendo desenvolvido pela Secretaria de Tecnologia da Informação e Comunicação (SeTIC), para identificar as funcionalidades que foram propostas para o novo módulo de matrícula, focando apenas nas que estão associadas aos alunos e no registro do pedido de matrícula.

2.2. Fase 2 - Aprendizado sobre PWAs

Nessa fase, foi feita uma Prova de Conceito (PoC) para que a autora aprendesse sobre as tecnologias associadas às PWAs e suas limitações, ganhando experiência antes de desenvolver a PWA proposta.

2.3. Fase 3 - Desenvolvimento da PWA

Nessa fase, foi desenvolvido um servidor com escopo extremamente reduzido para que fosse possível elaborar um teste de usabilidade sobre a PWA proposta porque o projeto da SeTIC atrasou e não foi possível utilizar a Application Programming Interface (API) oficial. Em seguida, foi desenvolvida a PWA, seguindo os requisitos que foram descritos no documento de requisitos, relacionados aos alunos e seu pedido de matrícula.

2.4. Fase 4 - Elaboração do teste de usabilidade

Nessa fase, o servidor e o cliente foram hospedados na plataforma Heroku e a pesquisa do teste de usabilidade foi enviada para aprovação do Comitê de Ética em Pesquisa com Seres Humanos da Universidade Federal de Santa Catarina (CEPSH-UFSC). Nesse teste, foram elaboradas 6 tarefas para os usuários realizarem e conhecerem a PWA proposta e dois questionários para mensurar a satisfação dos voluntários sobre as duas aplicações, a oficial e a PWA, e as funcionalidades novas que foram propostas, como as notificações quando um aluno entra ou sai da fila de espera de uma turma, visualização dos dados do pedido de matrícula, incluindo a posição e status (se está na fila de espera ou não) na turma matriculada, independente da conexão com a rede e instalação como um aplicativo no celular ou desktop.

2.5. Fase 5 - Execução, coleta e análise do dados do teste

Nessa fase, o teste foi executado e seus dados coletados. Além disso, foi feita uma comparação quantitativa entre as duas aplicações para avaliar se houve uma melhora da satisfação dos voluntários na solução proposta através do cálculo do *score* do SUS e se o público-alvo considerou as novas funcionalidades como úteis.

3. Trabalhos relacionados

Os sites MatrUFSC e MatrUFSC2, que é uma evolução do MatrUFSC, são aplicações web que foram desenvolvidas por alunos do Curso de Ciência da Computação da UFSC para permitir que os alunos planejem diversas grades de horário para o próximo semestre [MatrUFSC2], uma vez que o módulo de matrícula oficial apenas fica disponível durante a etapa de matrícula. Essas aplicações permitem que o aluno veja todos os horários disponíveis, quantidade de vagas livres e ocupadas, e os professores para uma determinada matéria no semestre especificado. Apesar disso, elas não possuem uma integração com o módulo oficial e não possuem funcionalidades no caso de perda de conexão, notificações e instabilidade.

Em relação ao módulo presente no Sistema de Controle Acadêmico da Graduação (CAGR), é uma aplicação web contendo diversas funcionalidades importantes para os alunos da UFSC, como a realização da matrícula. Todavia, ela não contém funcionalidades no caso de perda de conexão, notificações e instabilidade assim como os MatrUFSC e MatrUFSC 2, possui uma interface tecnologicamente e esteticamente defasada e não mostra informações como o professor responsável por ministrar. Esse problema na interface podem ser evidenciados pela criação de tutoriais e manuais para explicar aos alunos como realizar a matrícula [Secretaria Integrada de Graduação (SIG) do Centro de Ciências, do Curso de Ciências da Computação, dos Cursos de Graduação em Educação Física, do Curso de Graduação em Letras].

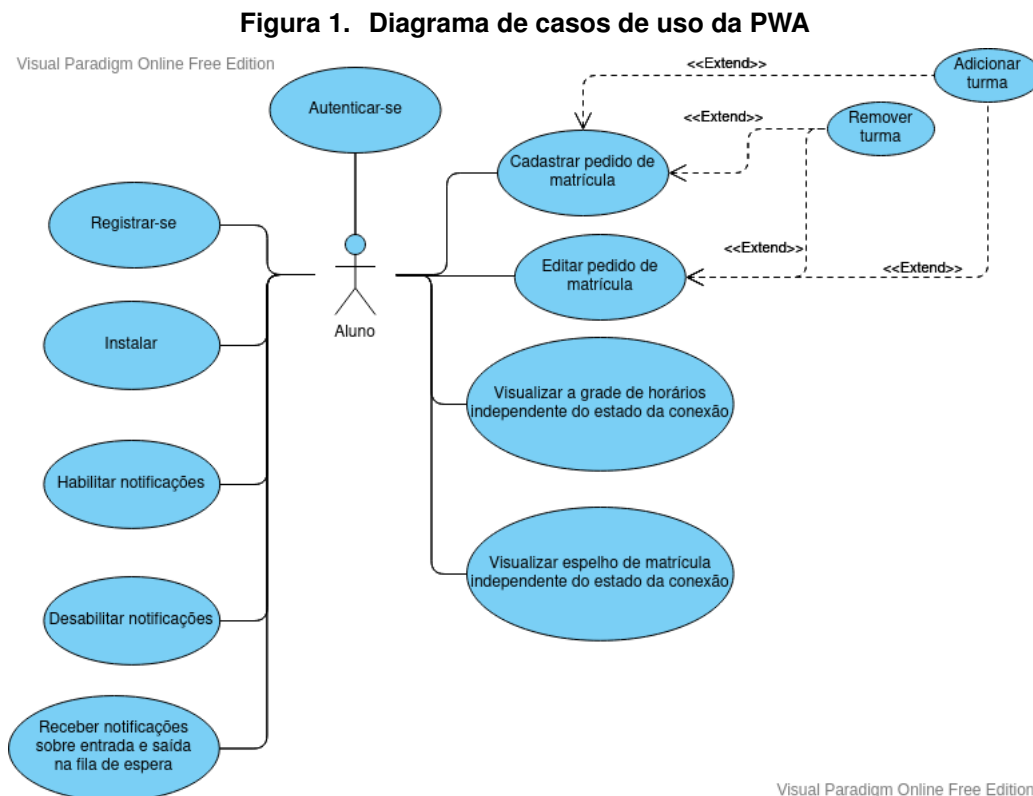
Em [Rodrigues 2020], foi proposta uma nova interface responsiva para o SIGAA da Universidade de Brasília (UnB), uma aplicação usada por instituições de ensino superior para gerenciar as atividades referentes à vida acadêmica dos discentes, devido aos problemas de usabilidade encontrados após a realização de uma pesquisa virtual com os alunos. Nessa interface, foram desenvolvidas mais funcionalidades do que as foram propostas por esse projeto como uma agenda para anotar as atividades, aba de links úteis para a biblioteca central e o restaurante universitário, acesso ao cronograma e ementa de cada disciplina do semestre cursado, entre outras. Além disso, foram desenvolvidas funcionalidades semelhantes ao da PWA, como a realização do pedido de matrícula e visualização da grade de horários.

4. PWA proposta

A aplicação foi desenvolvida para disponibilizar as seguintes funcionalidades aos alunos:

- Submissão e edição do pedido de matrícula;
- Visualização da grade de horários e do status da vaga em cada turma solicitada, se está fora (vaga possível) ou dentro de uma fila de espera e a posição do aluno, independente do estado de conexão do dispositivo;
- Possibilidade de instalação no computador, celular ou tablet e outros dispositivos;
- Envio de notificações caso o aluno entre ou saía de uma fila de espera, se for concedida a permissão.

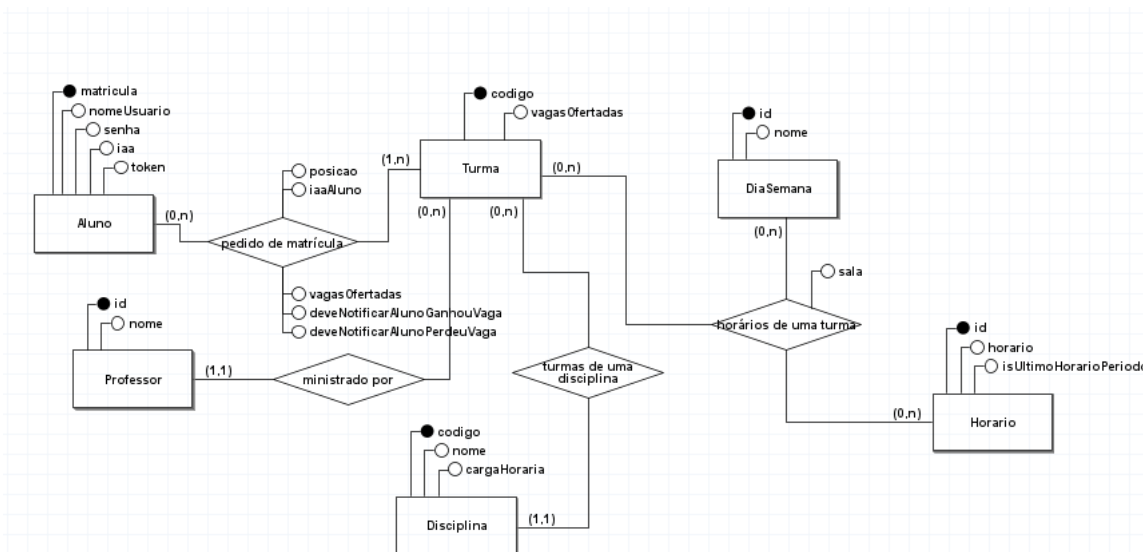
A figura 1 mostra as funcionalidades descritas através de uma diagrama de casos de uso.



Como ocorreram atrasos no desenvolvimento do servidor do módulo de matrícula pela SeTIC, um servidor com escopo extremamente reduzido foi implementado para permitir a realização do teste de usabilidade. Nesse servidor, não há preocupação com questões de segurança para evitar mais aumento de escopo no projeto e atrasos no desenvolvimento da aplicação proposta. Além disso, a posição de um aluno em uma turma é unicamente determinado pelo Índice de Aproveitamento Semestral Acumulado (IAA) para manter o servidor o mais simples possível, mas funcional para a realização do teste de usabilidade. Com esse atraso, as seguintes funcionalidades complementares foram necessárias: registro, autenticação e autorização de um aluno no servidor.

Para o desenvolvimento do servidor, foi utilizado o framework Spring Boot, Java 8 e Kotlin. Já para o cliente, foi utilizado o React e o Bold Design System [Bridge], desenvolvido pelo Laboratório Bridge. Para persistir as informações, foi utilizado o PostgreSQL com um banco de dados relacional. A figura 2 mostra a modelagem conceitual desse banco, sendo que algumas colunas foram replicadas nos relacionamentos para evitar a realização de joins nas stored procedures do banco. Todas essas escolhas foram feitas devido à familiaridade da acadêmica com essas tecnologias e ferramentas.

Figura 2. Modelagem conceitual do banco de dados



Todo o código desenvolvido para o cliente e servidor pode ser encontrado nos seguintes repositórios do GitHub: cliente e servidor. Essa separação foi necessária para distribuir as duas aplicações na plataforma do Heroku.

Nas próximas subseções, cada funcionalidade será descrita com um diagrama de atividades para mostrar o fluxo geral e apresentar telas da aplicação.

4.1. Registrando um aluno

Nessa funcionalidade, é feito o registro do aluno para permitir que se autentique posteriormente para acessar todas as outras funcionalidades disponíveis e a tela, presente na figura 3, foi desenvolvida, que contém um formulário para preencher as informações necessárias que são o nome de usuário e a senha. Depois de preencher os campos e clicar em

'Registrar', o servidor valida os dados e, se estiver tudo certo, adiciona um novo registro na tabela de alunos, sendo que a senha passará por uma função hash para não guardá-la como texto simples e o IAA será um número inteiro aleatório entre 0 e 9 gerado pelo servidor.

Figura 3. Tela para registrar o aluno

← Voltar

Registrando um aluno

Usuário *
Digite o seu usuário

Senha *
Digite sua senha

Registrar

4.2. Autenticando um aluno

Nessa funcionalidade, é feita a autenticação para acessar todas as outras funcionalidades disponíveis e a tela, presente na figura 4, foi desenvolvida, que contém um formulário para preencher as informações necessárias que são o nome de usuário e a senha. Depois de preencher os campos e clicar em 'Entrar', o servidor valida e autentica os dados e, se estiver tudo certo, o usuário será redirecionado para a tela do pedido de matrícula e guardará o JSON Web Tokens (JWT) no Local Storage do navegador para autorizar as próximas requisições.

Figura 4. Tela para autenticar o aluno

Matrícula UFSC

Login

Usuário *
Digite o seu usuário

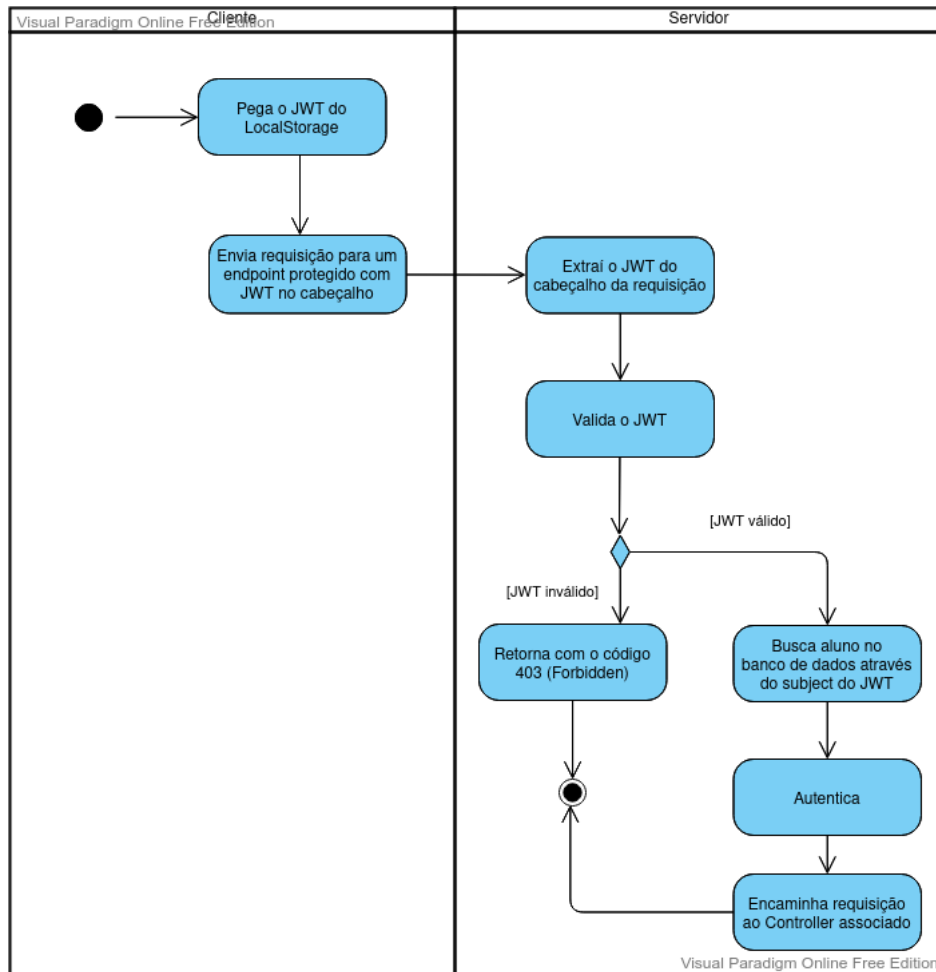
Senha *
Digite sua senha

Registrar-se Entrar

4.3. Autorizando um aluno

A autorização é responsável por garantir que apenas um aluno autenticado tenha acesso às funcionalidades de criar, editar e visualizar o pedido de matrícula através do envio do JWT presente no Local Storage para cada requisição para os endpoints do servidor. O fluxo dessa funcionalidade pode ser visto na figura 5 através de um diagrama de atividades.

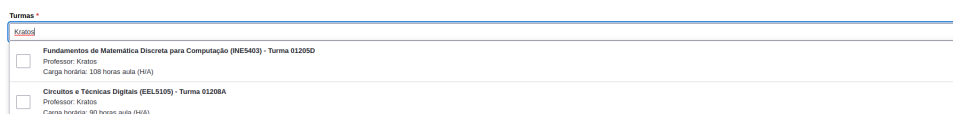
Figura 5. Fluxo ao autorizar um aluno



4.4. Criando ou editando um pedido de matrícula

Um pedido de matrícula pode ser definido como o conjunto de turmas que um determinado aluno deseja se matricular e, como um aluno pode mudar de ideia sobre quais turmas deseja, é importante permitir que ele as altere conforme à demanda. Para essa funcionalidade, foi desenvolvida uma tela contendo um formulário para selecionar as turmas no pedido de matrícula, presente na figura 6, a grade de horários, como pode ser visto na figura 7, e um campo contendo a carga horária total do pedido de matrícula. O campo de turmas permite o usuário filtrar pelo código e nome da disciplina, código da turma e nome do professor.

Figura 6. Buscando pelo nome do professor no campo de turmas



Sobre a carga horária total do pedido e a grade de horários, essas informações são atualizadas conforme o aluno seleciona as turmas no formulário. No caso da carga horária

Figura 7. Grade de horários

Grade de horários

	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
07:30						
08:20	INES403 - 01205D EF204	INES403 - 01205D EF204		INES402 - 03208A LIC13		
09:10	INES403 - 01205D EF204	INES403 - 01205D EF204		INES402 - 03208A LIC13		
10:10						
11:00 - 11:50						
13:30						
14:20						
15:10	EEL5105 - 01208A EEL004			INES402 - 03208A CTC301		
16:20	EEL5105 - 01208A EEL004			INES402 - 03208A CTC301		
17:10 - 18:00	EEL5105 - 01208A LABSDG			INES402 - 03208A CTC301		
18:30	EEL5105 - 01208A LABSDG		INES403 - 01205D EF204			
19:20	EEL5105 - 01208A LABSDG		INES403 - 01205D EF204			
20:20						
21:10 - 22:00						

total, a cor do texto é alterada para mostrar ao usuário se está dentro ou fora dos limites definidos, como pode ser visto nas figuras 8 e 9.

Figura 8. Texto sobre a carga horária total quando está fora dos limites definidos

Editando pedido de matrícula

Um pedido de matrícula é considerado válido quando não possui conflito de horários e cuja carga horária total está entre 270 H/A e 540 H/A

Carga horária total: 198 H/A

Figura 9. Texto sobre a carga horária total quando está dentro dos limites definidos

Editando pedido de matrícula

Um pedido de matrícula é considerado válido quando não possui conflito de horários e cuja carga horária total está entre 270 H/A e 540 H/A

Carga horária total: 306 H/A

Além disso, caso o aluno selecione turmas com conflito de horários, a grade de horários mostrará esse conflito com um texto em vermelho para deixar o mais evidente possível sobre o erro, como pode ser visto na figura 10.

No caso do aluno já ter criado o pedido de matrícula, assim que o aluno acessa essa tela, os dados desse pedido são buscados no servidor e, em seguida, as turmas são colocadas no campo de turmas e a grade de horários preenchida com os horários das turmas solicitadas no pedido.

4.5. Visualizando grade de horários e status da vaga através do espelho de matrícula

A grade de horários permite que o aluno saiba em quais salas, dias e horários da semana serão as aulas de cada turma matriculada ou solicitadas através do pedido de matrícula. O espelho de matrícula permite visualizar o pedido de matrícula do aluno e quais turmas foram efetivamente matriculadas. No caso dessa aplicação, como é apenas abordado a etapa de solicitação e ajuste do pedido de matrícula, não estará contemplado o cenário

Figura 10. Conflito na grade de horários

	Segunda	Terça	Quarta
07:30			
08:20	INE5403 - 01205D EF1204	INE5403 - 01205D EF1204	
09:10	INE5403 - 01205D EF1204	INE5403 - 01205D EF1204	
10:10			
11:00 - 11:50			
13:30			
14:20			
15:10	EEL5105 - 01208A EEL004		
16:20	EEL5105 - 01208A EEL004		
17:10 - 18:00	EEL5105 - 01208A LABSDG		INE5401 - 03202A CTC102
18:30	EEL5105 - 01208A LABSDG		INE5403 - 01205D EF1204 INE5401 - 03202A CTC102
19:20	EEL5105 - 01208A LABSDG		INE5403 - 01205D EF1204
20:20			
21:10 - 22:00			

de mostrar as turmas que foram efetivamente matriculadas. Foi desenvolvida uma tela composta pela grade de horários, o mesmo que é utilizado na criação e edição do pedido de matrícula, e o espelho de matrícula, que mostra cada turma que foi solicitada no pedido e seu professor, o status e posição da vaga do aluno. A figura 11 mostra o espelho de matrícula com os dois status definidos pela aplicação, o vaga possível, evidenciando uma possível matrícula na turma já que não é possível dar uma garantia porque vários alunos podem entrar e tomar a vaga, e na fila de espera.

Figura 11. Espelho de matrícula

Pedido de matrícula

ⓘ A posição em uma turma é determinada pelo Índice de Aproveitamento Semestral Acumulado (IAA) e o seu é 1. Esse valor foi gerado aleatoriamente pelo servidor

Espelho de matrícula

Disciplina - Turma	Professor	Status da vaga	Sua posição
Fundamentos de Matemática Discreta para Computação (INE5403) - 01205C	Aloy	⚠ Fila de espera	1º lugar na fila de espera
Fundamentos de Matemática Discreta para Computação (INE5403) - 01205D	Kratos	✅ Vaga possível	1º lugar de 50
Circuitos e Técnicas Digitais (EEL5105) - 01208A	Kratos	✅ Vaga possível	1º lugar de 5
Programação Orientada a Objetos I (INE5402) - 03208A	Amanda Ripley	✅ Vaga possível	1º lugar de 5

Editar pedido de matrícula

Para permitir o acesso à grade de horários e espelho de matrícula quando o usuário estiver sem conexão com a rede, foi necessário registrar e configurar o service worker. Em seguida, um Event Listener para o evento de fetch, que é acionado para cada requisição HTTP feita pelo cliente. Para essa aplicação, optou-se pela estratégia 'Network first, falling back to cache', ou seja, o cliente sempre busca as informações no servidor primeiro, e se a requisição for um sucesso, guarda o resultado na cache no caso de não ter conexão com a rede no futuro, caso contrário, pega o resultado guardado na cache, porque

as informações da grade de horários e do espelho de matrícula precisam ser a mais atuais sempre que for possível. Para evitar de guardar dados na cache que não seriam utilizados, os endpoints guardados na cache foram os associados à grade de horários e ao espelho de matrícula já que não faria sentido editar o pedido de matrícula sem ter conexão com a rede.

Caso o usuário fique sem conexão, a funcionalidade de edição e cadastro do pedido de matrícula e notificações são desabilitadas removendo os botões associados, 'Editar pedido matrícula' e 'Ativar notificações' respectivamente, e uma mensagem de alerta avisando que as informações podem estar desatualizadas e que algumas funções foram desabilitadas devido à falta de conexão. As figuras 12 e 13 mostram a mensagem de alerta e o espelho de matrícula sem o botão de edição quando há perda de conexão.

Figura 12. Mensagem de alerta e botão de instalação e notificações removidos quando falta conexão

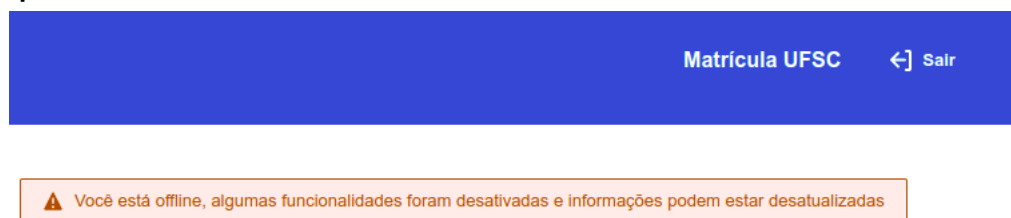


Figura 13. Espelho de matrícula sem o botão de editar quando falta conexão



Grade de horários

4.6. Notificações

As notificações permitem que os alunos fiquem mais informados sobre os status de cada turma solicitada no pedido de matrícula, ou seja, se entraram ou não pertencer mais ao grupo de alunos classificados ou saíram de uma fila de espera de uma turma ao entrar no grupo de alunos classificados após a saída de outro. Para essa funcionalidade, foi utilizada a ferramenta FCM devido à facilidade na configuração e uso sem precisar se preocupar com o Web Push Protocol. Para permitir que o usuário habilite e desabilite as notificações, foi criado um botão para as notificações que altera o ícone e o título caso o usuário as tenha habilitado ou não. A figura 14 mostra o botão quando o usuário ainda não habilitou as notificações e a figura 15 quando as habilitou.

Figura 14. Botão para habilitar notificações

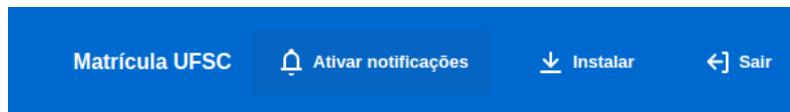
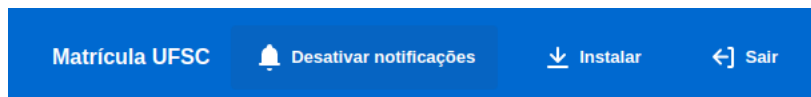


Figura 15. Botão de desativar notificações



As notificações que vieram em primeiro plano, quando a aplicação está com o foco, são mostradas pelas figuras 16 e 17 e as em segundo plano, quando está oculta atrás de outras janelas ou fechada, são mostradas pelas figuras 18 e 19.

Figura 16. Notificação ao sair da fila de espera de uma turma em primeiro plano

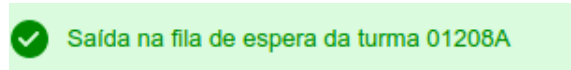


Figura 17. Notificação ao entrar na fila de espera de uma turma em primeiro plano



Figura 18. Notificação ao sair da fila de espera de uma turma em segundo plano

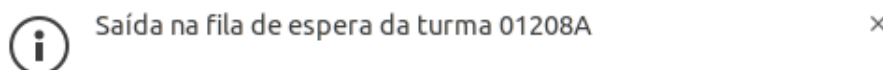
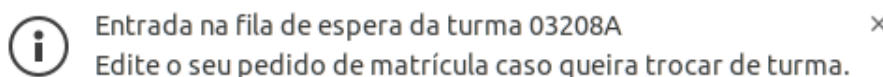


Figura 19. Notificação ao entrar na fila de espera de uma turma em segundo plano



4.7. Instalação

A instalação permite que o aluno utilize a aplicação como um aplicativo de celular ou desktop sem precisar acessar uma página da loja de aplicativos, podendo acessá-la com apenas um clique no ícone. No caso dos PWAs, essa instalação é opcional, garantindo mais liberdade ao usuário para apenas instalar a aplicação quando achar importante ou necessário.

Para tornar a PWA instalável, foi necessário alterar o manifesto gerado automaticamente para conter as informações corretas sobre a aplicação, adicionar um link para o manifesto no *index.html* com o atributo *rel* tendo o valor *manifest* e garantir que o manifesto cumpriria os requisitos mínimos para ser instalável, que são ter os seguintes campos definidos [LePage 2022]:

- *short_name* ou *name*;
- Ícones de 192px e 512px;
- *start_url*;
- *display*.

O código abaixo mostra o manifesto implementado.

```
// public/manifest.json
{
  "short_name": "Matrícula UFSC",
  "name": "Matrícula UFSC - PWA",
  "start_url": ".",
  "display": "standalone",
  "theme_color": "#000000",
  "background_color": "#ffffff",
  "orientation": "landscape",
  "icons": [
    {
      "src": "favicon.ico",
      "sizes": "16x16",
      "type": "image/x-icon"
    },
    {
      "src": "images/book-stack-16.png",
      "type": "image/png",
      "sizes": "16x16"
    },
    {
      "src": "images/book-stack-24.png",
      "type": "image/png",
      "sizes": "24x24"
    },
    {
      "src": "images/book-stack-32.png",
      "type": "image/png",
      "sizes": "32x32"
    },
    {
      "src": "images/book-stack-64.png",
      "type": "image/png",
      "sizes": "64x64"
    },
  ],
}
```

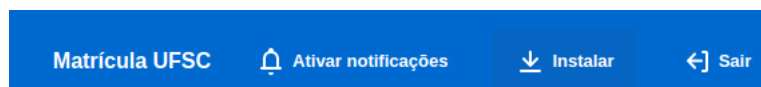
```

    {
      "src": "images/book-stack-128.png",
      "type": "image/png",
      "sizes": "128x128"
    },
    {
      "src": "images/book-stack-256.png",
      "type": "image/png",
      "sizes": "256x256"
    },
    {
      "src": "images/book-stack-512.png",
      "type": "image/png",
      "sizes": "512x512"
    }
  ]
}

```

Além disso, foi implementado um botão para permitir que o aluno instalasse a aplicação, que pode ser visto na figura 20.

Figura 20. Botão de instalação



5. Teste de usabilidade

O teste de usabilidade tem como objetivo mensurar a satisfação dos usuários em relação à aplicação oficial disponibilizada no CAGR e a que está sendo proposta com a participação de veteranos e recém-formados da UFSC, representando o público-alvo dessas aplicações, para permitir que a autora realize uma comparação quantitativa entre elas. Esse teste pode ser dividido nas seguintes etapas: avaliação da satisfação dos alunos sobre o módulo oficial através de um questionário, execução de 6 tarefas no novo módulo para simular tarefas reais e avaliação da satisfação em relação à PWA desenvolvida e das novas funcionalidades propostas através de outro questionário. Todas as etapas desse teste foram feitas de forma remota e assíncrona e as respostas foram anônimas para assegurar a privacidade, sigilo e confidencialidade de todos os participantes. Para medir a satisfação, foram criados dois questionários através do Google Forms e um conjunto de 6 tarefas para os participantes do teste realizarem na interface nova para adquirir experiência com ela. Ressalta-se que o foco desse projeto estava na avaliação da usabilidade apenas sob o aspecto da satisfação dos usuários, sendo que os outros aspectos como eficácia e eficiência não foram considerados relevantes para essa avaliação visto que ela estava voltada para verificar o impacto da nova abordagem na realização das matrículas assim como a reação aos feedbacks que o sistema atual não fornece.

Em relação aos participantes que foram escolhidos para realizar todas as etapas do teste, optou-se pelos veteranos e recém-formados por já terem experiência com

a aplicação oficial, possibilitando que respondessem a primeira avaliação e realizassem uma comparação entre as duas interfaces. A participação deles foi voluntária e os convites foram feitos pelo e-mail do curso e canais de comunicação do Laboratório Bridge, já que a autora trabalha nessa instituição e o laboratório possui diversos veteranos e recém-formados da UFSC.

5.1. Tarefas

O objetivo dessas tarefas era permitir que o participante explorasse todas as funcionalidades propostas para avaliá-las no segundo questionário. Nessas tarefas, há uma descrição breve do contexto imaginário que o participante estava inserido e um conjunto de atividades que deveriam ser realizadas para concluir a tarefa. As atividades não continham um passo a passo claro do que deve ser feito, visto que isso afetaria a avaliação dos usuários sobre o sistema já que ele deve ser fácil de entender e usar. Esse modelo de teste usabilidade foi baseado no que é feito pelo Laboratório Bridge para a realização dos testes das versões. A figura 21 mostra um exemplo de tarefa que foi proposta para os usuários realizarem.

Figura 21. Primeira tarefa que foi proposta aos participantes do teste

Tarefa 1

Imagine que a Universidade Federal de Santa Catarina (UFSC) desenvolveu um novo sistema de matrícula e que está no período da solicitação de matrícula. Além disso, você e seu amigo combinaram de ingressar nas mesmas turmas das disciplinas **INE5403**, **INE5402** e **EEL5105** para fazer os trabalhos em dupla e que você deveria se matricular em Cálculo 1 (**MTM3101**) porque passou na prova de proficiência, mas o seu amigo não, então ele precisaria fazer Pré-Cálculo primeiro. Além disso, **você não poderia ter aulas na quinta-feira no período matutino (7h até 12h)** porque precisa cuidar dos seus irmãos mais novos até o horário da aula deles e um dos veteranos comentou que a **Aloy** era uma excelente professora e que você deveria pegar todas as turmas que ela ministrasse.

1. Acesse o novo sistema de matrícula que está disponível neste [link](#)
2. Realize o cadastro e a autenticação
3. Solicite as turmas das disciplinas

Anotações sobre a Tarefa 1

Nessa parte, anote se você conseguiu concluir a tarefa e quais foram as suas impressões sobre o sistema e essa tarefa. Sugestão de perguntas para serem respondidas: Foi fácil de achar as informações? Foi rápido de realizar a tarefa? As informações são fáceis de ler e objetivas? Existem melhorias que poderiam ser feitas?

Se você não conseguiu concluir, deixe o motivo (ex: "Não achei o botão", "Não havia toda a informação necessária", etc...)

5.2. Questionários

Os questionários foram desenvolvidos para medir a satisfação dos voluntários sobre as duas aplicações, CAGR e PWA, através de perguntas do questionário SUS que foram respondidas utilizando a Escala Likert para calcular o *score* de cada módulo. Com isso, a autora pôde realizar uma comparação quantitativa entre os *scores* obtidos para avaliar se houve uma melhora na satisfação dos usuários para a PWA desenvolvida.

Acerca da primeira avaliação, foram realizadas apenas as 10 perguntas do questionário SUS para mensurar a satisfação dos usuários em relação ao módulo do CAGR. Em

relação à segunda avaliação, ela foi feita no dia posterior ao da execução das tarefas que foram descritas anteriormente e a motivação para isso foi para dar tempo das notificações serem exibidas aos participantes para poderem avaliar essa nova funcionalidade. Nesse questionário, foram realizadas as mesmas dez perguntas do questionário SUS para mensurar a satisfação dos usuários acerca da aplicação proposta e mais os seguintes questionamentos:

- O navegador e o dispositivo (celular, computador, tablet, etc...) que foram utilizados;
- Se o participante conseguiu concluir a tarefa, e, caso não tenha conseguido, o motivo para não ter conseguido;
- Se o participante possui uma sugestão de melhoria para uma determinada funcionalidade ou para o sistema como um todo;
- Se as funcionalidades novas são úteis (1 - Sem utilidade, 5 - Útil);
- Se as informações das notificações, status e posição da vaga são fáceis de entender e objetivas através de afirmações que os voluntários respondiam seguindo a Escala Likert (1 - Discordo fortemente, 5 - Concordo fortemente);
- Se o participante vê valor na liberdade de escolha para instalar ou não.

5.3. Resultados

Para o primeiro questionário, foram obtidas 30 respostas e um *score* do SUS de 50,25, ficando abaixo do recomendado [Sauro 2011], para o módulo de matrícula do CAGR enquanto que, para o segundo questionário, 25 dos 30 voluntários realizaram as tarefas e o responderam, alcançando um *score* de 90,01 para o módulo proposto (PWA), atingindo um aumento de 79,1%. O *score* de cada módulo foi obtido como a média dos *scores* de cada usuário, cujo cálculo pode ser encontrado na tabela ???. Ademais, 92% dos voluntários utilizaram o navegador Chrome, que foi uma sugestão da autora para o uso da aplicação já que possui mais suporte para o PWA, e apenas 8% realizaram as tarefas do módulo proposto pelo celular enquanto que o público restante utilizou um computador, desktop ou notebook.

Pergunta	M_{CAGR}	M_{PWA}	Variação percentual (%)
Pergunta 1	5	5	0,0
Pergunta 2	4	1	-75,0
Pergunta 3	3	5	66,7
Pergunta 4	1	1	0,0
Pergunta 5	2	5	150,0
Pergunta 6	4	1	-75,0
Pergunta 7	2	5	150,0
Pergunta 8	4	1	-75,0
Pergunta 9	3	5	66,7
Pergunta 10	1	1	0,0

Tabela 1. Moda das respostas obtidas para cada pergunta do questionário do SUS separadas pelo módulo de matrícula

Acerca da realização das tarefas pelos voluntários, a sexta tarefa, relacionada a acessar os dados quando não há conexão com a rede, apresentou a menor taxa de con-

clusão entre os voluntários que foi de 88%, todavia não houveram comentários para identificar os motivos. Em relação às outras tarefas, na quarta tarefa os voluntários expressaram que o motivo para não finalizá-la estava no fato da aplicação comentar que as notificações tinham sido revogadas e os usuários não conseguiam habilitá-las no dispositivo. Enquanto que na quinta, um dos participantes comentou que não tinha o celular em mãos para instalar e o outro relatou que não recebeu um feedback da aplicação e apenas conseguiu acessá-la através do link disponibilizado no documento das tarefas.

Em relação às funcionalidades propostas, as notificações, status e posição da vaga no espelho de matrícula e visualização do espelho e da grade de horários mesmo sem conexão com a rede foram as funcionalidades que foram vistas como úteis pelos participantes do teste de usabilidade e que poderiam ser implementadas no módulo oficial. Por outro lado, a instalação, devido a sazonalidade do processo de matrícula, foi considerada menos útil pelos voluntários, como foi observado nos comentários enviados no segundo questionário, e, no geral, viram valor em ter a liberdade em decidir instalar a aplicação.

6. Conclusões

Em relação ao objetivo de desenvolver uma PWA com React e o Bold Design System [Bridge], observa-se que ele foi atingido e permitiu que os alunos realizassem as operações básicas em um processo de matrícula como buscar as turmas e solicitar e ajustar o pedido de matrícula. Apesar disso, a aplicação não contemplou todos os requisitos oficiais que foram propostos, como, por exemplo, a opção de filtrar as turmas por semestre corrente e remover turmas aprovadas. Esse impedimento ocorreu devido ao fato da autora precisar se dedicar no desenvolvimento de um servidor já que houveram atrasos na implementação da API oficial por parte da SeTIC para conseguir realizar o teste de usabilidade dentro do prazo.

Em relação ao objetivo de desenvolver o servidor com escopo reduzido para que a autora conseguisse elaborar o teste de usabilidade, constata-se que ele foi concluído com sucesso.

Acerca do objetivo de elaborar um roadmap para que outros alunos consigam desenvolver uma PWA, é possível constatar que foi atingido com ressalvas. Esse roadmap só poderia ser utilizado para desenvolvedores que queiram implementar uma PWA com React que recebe notificações tanto em primeiro como segundo plano através da infraestrutura oferecida pela Firebase Cloud Messaging (FCM) e que é instalável no Chrome e acessa dados pela cache do service worker. Além disso, permite que desenvolvam um servidor com Spring que envie as notificações através da FCM. Acerca do Web Push Protocol, não foi explicado na prática como utilizá-lo, no entanto, foi descrito em um nível mais abstrato para permitir a compreensão das entidades envolvidas e suas comunicações.

No que se refere à meta de melhorar a satisfação dos usuários, ela foi alcançada e pode ser evidenciada através do aumento de 79,1% no score do SUS para a PWA proposta em relação ao módulo do CAGR. Além disso, as perguntas 2 ('Eu acho o módulo desnecessariamente complexo'), 6 ('Eu acho que o módulo apresenta muita inconsistência') e 8 ('Eu acho o módulo confuso de usar') apresentaram uma redução de 75% nas modas das respostas da PWA em comparação com as do CAGR. No caso das perguntas 5 ('Eu acho que várias funções do módulo estão muito bem integradas') e 7 ('Eu imagino que as pessoas aprendem como usar esse módulo rapidamente') apresentaram uma melhora

de 150% nas modas das respostas da PWA em comparação com as do CAGR. Contudo, devido a uma amostragem muito pequena, não é possível inferir com um alto grau de certeza que todos os alunos da UFSC ficariam mais satisfeitos com essa aplicação e que essa PWA possui um bom suporte para outros navegadores, como Mozilla e Safari, e que traria uma boa experiência para usuários de celular devido ao fato de que apenas 8% dos voluntários realizaram as tarefas com um celular.

A respeito das PWAs, constatou-se que elas possuem diversos recursos, como as notificações, instalação e controle das requisições através do service worker, que poderiam ser aproveitados no desenvolvimento de sistemas para melhorar a experiência do usuário, como foi observado nos os resultados da pesquisa desse projeto. Todavia, essa tecnologia possui algumas limitações dependendo do navegador ou dispositivo devido ao suporte incremental que é dado pelas empresas e instituições, podendo reduzir a qualidade da experiência do usuário ao impedir que realize uma determinada tarefa, conforme foi constatado durante o teste de usabilidade do PWA do módulo de matrícula e no desenvolvimento da funcionalidade de instalação com o navegador Mozilla.

Por fim, as notificações, status e posição da vaga no espelho de matrícula e visualização do espelho e da grade de horários mesmo sem conexão com a rede foram as funcionalidades que foram vistas com mais valor pelos participantes do teste de usabilidade e que poderiam ser implementadas no módulo oficial. Por outro lado, a instalação, devido a sazonalidade do processo de matrícula, foi considerada menos útil pelos voluntários, como foi observado nos comentários enviados no segundo questionário. Em vista disso, o desenvolvimento de aplicações nativas ou investimento na experiência de instalação para um PWA parece não ser tão interessante para esse tipo de sistema, já que os usuários não pretendem instalá-lo.

Como trabalhos futuros, são propostas as seguintes tarefas:

- Integrar com o sistema de matrícula que está sendo desenvolvido pela SeTIC;
- Avaliar e implementar as melhorias que foram propostas pelos voluntários da pesquisa;
- Contemplar todos os requisitos oficiais propostos;
- Elaborar uma análise estatística mais aprofundada sobre os dados que foram obtidas no teste de usabilidade;
- Verificar a questão do suporte em outros dispositivos e navegadores.

Referências

- [Bridge] Bridge, L. *Bold Design System*. Laboratório Bridge.
- [Charland and LeRoux 2011] Charland, A. and LeRoux, B. (2011). Mobile application development: Web vs. native: Web apps are cheaper to develop and deploy than native apps, but can they match the native user experience? *Queue*, 9(4):20–28.
- [Corral et al. 2012] Corral, L., Sillitti, A., Succi, G., Garibbo, A., and Ramella, P. (2012). Evolution of mobile software development from platform-specific to web-based multiplatform paradigm. In *Proceedings of the 10th SIGPLAN Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software*, Onward! 2011, page 181–183, New York, NY, USA. Association for Computing Machinery.

- [do Curso de Ciências da Computação] do Curso de Ciências da Computação, S. *Processo de Matrícula para Veteranos*. Secretaria do Curso de Ciências da Computação.
- [do Curso de Graduação em Letras] do Curso de Graduação em Letras, C. *Manual de Matrículas*. Coordenadoria do Curso de Graduação em Letras.
- [dos Cursos de Graduação em Educação Física] dos Cursos de Graduação em Educação Física, C. *Matrículas – Renovação via CAGR*. Coordenadoria dos Cursos de Graduação em Educação Física.
- [Kemp 2022] Kemp, S. (2022). *Digital 2022: Global Overview Report*. DataReportal.
- [LePage 2022] LePage, P. (2022). *What does it take to be installable?* Google.
- [Malavolta 2016] Malavolta, I. (2016). Beyond native apps: Web technologies to the rescue! (keynote). In *Proceedings of the 1st International Workshop on Mobile Development, Mobile! 2016*, page 1–2, New York, NY, USA. Association for Computing Machinery.
- [MatrUFSC2] MatrUFSC2. *Sobre*. MatrUFSC2.
- [Rodrigues 2020] Rodrigues, F. d. F. T. (2020). Redesign do sigaa acadêmico módulo aluno: uma abordagem centrada no usuário.
- [Sam Richard 2020] Sam Richard, P. L. (2020). *What are Progressive Web Apps?* Google.
- [Sauro 2011] Sauro, J. (2011). *Measuring Usability with the System Usability Scale (SUS)*. MeasuringU.
- [Secretaria Integrada de Graduação (SIG) do Centro de Ciências] Secretaria Integrada de Graduação (SIG) do Centro de Ciências, T. e. S. d. U. *Matrícula de Veteranos*. Secretaria Integrada de Graduação (SIG) do Centro de Ciências, Tecnologias e Saúde da UFSC.
- [Statcounter 2022] Statcounter (2022). *Desktop vs Mobile vs Tablet Market Share Worldwide*. Statcounter.

Anexos

ANEXO A – REQUISITOS OFICIAIS RELACIONADOS AOS VETERANOS

1 - Regras de negócio da matrícula de veteranos

RN01 - Estrutura do quadro de horário: a alocação de turmas da graduação é feita em quadro de horários que prevê dias úteis como de segunda a sábado e horários de alocação como múltiplos de 50 minutos, sendo o período matutino entre 07:30h e 11:50h (com intervalo entre 10:00h e 10:10h), o período vespertino entre 13:30h e 18:00h (com intervalo entre 16:00h e 16:20h) e o período noturno entre 18:30h e 22:00h (com intervalo entre 20:10 e 20:20h). Tal definição de horários é passível de modificação, a partir de alteração da legislação vigente (assim, o seu tratamento por sistema computadorizado demanda que tanto os três períodos como a duração das aulas sejam dados parametrizáveis). O registro de matrícula, além das disciplinas com horários definidos, também deve prever o registro de atividades acadêmicas com carga horária a registrar, mas sem um horário encaixável no quadro de horários, como estágio curricular, atividades acadêmicas complementares e outras, previstas em currículo de curso.

RN02 - Disciplina equivalente: uma disciplina X é equivalente a uma disciplina Y se, para efeito de integralização curricular, cursar a disciplina X gerar o mesmo resultado de cursar a disciplina Y. A relação de equivalência é unidirecional. Assim, o fato de uma disciplina apontar outra disciplina como sua equivalente não pressupõe a relação recíproca. Para efeito de verificação de cumprimento de restrições curriculares, assim como para definição de prioridade no preenchimento das vagas nas disciplinas, uma disciplina equivalente deve ser tratada como se fosse a própria disciplina presente no currículo.

RN03 - Disciplina extracurricular: uma disciplina extracurricular é uma disciplina ausente do currículo do curso e que não é equivalente a uma disciplina do currículo. Considerando que a matrícula em disciplina extracurricular implica na matrícula em turma (da disciplina) de outro curso, para poder matricular-se, o estudante deve cumprir as restrições curriculares previstas no currículo do curso associado à turma, ou as restrições curriculares previstas em qualquer outro curso que tenha a disciplina em seu currículo.

RN04 - Choque de horário: o choque de horário é caracterizado quando os horários de duas ou mais turmas coincidem total ou parcialmente. A ocorrência de choque de horário entre turmas em uma solicitação de matrícula é proibida, exceto em situações de exceção previstas em legislação específica, nos termos previstos em tal legislação, para períodos letivos específicos, como ocorrido no período de ensino não presencial, durante a pandemia de Covid-19. Com isso, a proibição de choque de horário entre disciplinas deve estar atendida em uma solicitação de matrícula, excetuando:

1. O estudante autorizado formalmente - por deferimento de requisição prévia - a descumprimento de choque de horário, nos termos das normas vigentes, com autorização registrada e acessível ao sistema de matrícula, especificando o(s) choque(s) entre disciplinas autorizado(s).

RN05 - Carga horária: a carga horária do conjunto de disciplinas/turmas inserido no quadro de horários mais a carga horária das atividades curriculares sem horário específico (mas com carga horária), deve resultar em uma carga horária total que obedeça aos limites de cargas horárias máxima e mínima previstos no currículo do curso do estudante. As restrições de carga horária podem ser suspensas em situações de exceção previstas em legislação específica, nos termos previstos em tal legislação, para períodos letivos específicos, como ocorrido no período de ensino não presencial, durante a pandemia de Covid-19. A exigência de observação dos limites de carga horária deve estar atendida em uma solicitação de matrícula, excetuando:

1. A possibilidade de estudante formando cursar carga inferior ao limite mínimo, incondicionalmente;
2. O estudante autorizado formalmente - por deferimento de requisição prévia - a matricular-se acima da carga máxima ou abaixo da carga mínima, nos termos das normas vigentes, com autorização registrada e acessível ao sistema de matrícula, especificando a carga permitida.

RN06 - Pré-requisito: uma disciplina é pré-requisito de outra caso a matrícula nesta só seja permitida se aquela já fizer parte do histórico escolar do estudante, com aprovação ou, ao invés dela, uma que lhe seja equivalente, com aprovação. Uma determinada carga horária também pode ser estabelecida como pré-requisito para matrícula em disciplina. A previsão de pré-requisito estabelecida em currículo para uma disciplina, seja considerando outra(s) disciplina(s) e/ou carga horária, deve estar atendida, considerando o histórico escolar, excetuando:

O estudante autorizado formalmente - por deferimento de requisição prévia - a descumprimento de pré-requisito, nos termos das normas vigentes, com autorização registrada e acessível ao sistema de matrícula, especificando que pré-requisito(s) pode(m) ser descumprido(s).

RN07 - Correquisito: uma disciplina é correquisito de outra caso a matrícula nesta só seja permitida juntamente com a matrícula naquela ou numa que lhe seja equivalente ou ainda se aquela já fizer parte do histórico escolar do estudante, com aprovação ou, ao invés dela, uma que lhe seja equivalente, com aprovação. O uso do correquisito se destina a garantir que uma disciplina ou seja cursada simultaneamente a outra, ou após ela. Trata-se de uma relação unidirecional. Assim, o fato de uma disciplina ser correquisito de outra não supõe a relação recíproca. Com isso, a previsão de correquisito estabelecida em currículo para uma disciplina deve estar atendida, considerando o histórico escolar e o restante da solicitação de matrícula, excetuando:

1. O estudante autorizado formalmente - por deferimento de requisição prévia - a descumprimento de correquisito, nos termos das normas vigentes, com autorização registrada e acessível ao sistema de matrícula, especificando que correquisito(s) pode(m) ser descumprido(s).

RN08 - Restrição curricular específica: a solicitação de matrícula deve estar de acordo com quaisquer restrições curriculares específicas, previstas no currículo do curso, excetuando:

1. O estudante autorizado formalmente - por deferimento de requisição prévia - a descumprimento da restrição, nos termos das normas vigentes, com autorização registrada e acessível ao sistema de matrícula, para ser tratada de forma automatizada, especificando que restrição(ões) pode(m) ser descumprida(s).

RN09 - Estudante formando: estudante apto à integralização curricular no período letivo subsequente, considerando que carga horária que resta para a integralização pode ser totalmente inserida em sua solicitação de matrícula para o período letivo subsequente, consideradas as restrições curriculares previstas nas regras de negócio RN04 a RN08, com destaque para a possibilidade de estudante formando cursar carga inferior ao limite mínimo, prevista na '*RN05 - Carga horária*'.

RN10 - Prioridade de estudante em turma: a prioridade de um estudante para obtenção de vaga em uma turma, que deve ser informada a ele pelo sistema de matrícula, é definida com base na legislação da UFSC, sendo atualmente estabelecida pelo Art. 42 da Resolução 17/CUn/97, de 30 de setembro de 1997. Considerando a necessidade de alteração dessa legislação define-se que o *preenchimento das vagas nas turmas de disciplinas, na renovação de matrícula e no ajuste da mesma, será realizado na seguinte ordem de prioridade, respeitado em cada caso o índice de matrícula (IM):*

I - estudante regular, do curso e do turno em que a turma está alocada, tendo por base o período letivo de ingresso via Processo Seletivo;

II - Estudante formando, do curso e do turno em que a turma está alocada;

III - estudante do curso e do turno em que a turma está alocada e que não sofreu reprovação anterior na disciplina requerida;

IV - estudante do curso e do turno em que a turma está alocada e que foi anteriormente reprovado, com frequência suficiente (FS), ou que cancelou a matrícula anteriormente;

V - estudante do mesmo curso, mas de outro turno e que foi anteriormente reprovado, com frequência suficiente(FS);

VI - estudante do curso e do turno em que a turma está alocada e que foi reprovado anteriormente, com frequência insuficiente (FI) na disciplina;

VI - estudante do mesmo curso, mas de outro turno e que foi anteriormente reprovado, com frequência insuficiente(FI);

VIII - estudante de outro curso que possui a disciplina em seu currículo ou que a disciplina é equivalente a alguma disciplina de seu currículo e que não sofreu reprovação anterior na disciplina ~~ou bloco de disciplinas~~ requerido;

IX - estudante de outro curso que possui a disciplina em seu currículo ou que a disciplina é equivalente a alguma disciplina de seu currículo, que foi anteriormente reprovado com frequência suficiente (FS) na respectiva disciplina ~~ou bloco de disciplinas~~, ou que cancelou a matrícula anteriormente;

X - estudante de outro curso que possui a disciplina em seu currículo ou que a disciplina é equivalente a alguma disciplina de seu currículo e foi anteriormente reprovado com frequência insuficiente (FI), na respectiva disciplina ~~ou bloco de disciplinas~~;

XI - estudante solicitando matrícula em disciplina extracurricular e que não sofreu reprovação anterior na disciplina requerida;

XII - estudante solicitando matrícula em disciplina extracurricular e que foi anteriormente reprovado, com frequência suficiente(FS);

XIII - estudante solicitando matrícula em disciplina extracurricular e que foi reprovado anteriormente, com frequência insuficiente (FI) na disciplina;

XIV - estudante solicitando matrícula em disciplina em que foi aprovado anteriormente.

RN11 - Período de renovação de matrícula: Para efeito do estabelecimento das restrições temporais para a ocorrência de renovação de matrícula, consideram-se as definições a seguir:

- **Período letivo:** intervalo de tempo em que as atividades previstas nos planos de ensino são desenvolvidas.
- **Período de recesso escolar:** intervalo de tempo sem o desenvolvimento de atividades previstas em planos de ensino. O período de recesso escolar é disjuncto em relação ao período letivo. Sempre há um período de recesso escolar final, que ocorre entre períodos letivos distintos (tratado a partir deste ponto, ao longo deste documento - como recurso de abreviação - apenas como período de recesso escolar). Pode haver um ou mais períodos de recesso escolar que interrompam um período letivo (tal tipo de período é tratado a partir deste ponto, ao longo deste documento, como período intermediário de recesso escolar). É possível, durante um período de recesso escolar o desenvolvimento de atividades curriculares que não estejam vinculadas ao calendário acadêmico, bem como, em caráter extraordinário, por decisão de colegiado de curso e com a anuência do departamento envolvido, o desenvolvimento de atividades letivas, como cursos de verão, por exemplo
- **Período de lançamento de notas:** intervalo de tempo em que são feitos os lançamentos de nota de um período letivo. Este período pode iniciar antes ou após o término do período letivo e deve encerrar antes do período letivo subsequente, para não comprometer o processo de renovação de matrícula.

Como regra, o período de renovação de matrícula para um período letivo, isto é, o intervalo de tempo em que a renovação de matrícula ocorre, desde o início das solicitações de matrícula por parte dos estudantes até o preenchimento dessas turmas, pode iniciar durante o período letivo precedente ou após o encerramento deste e deve encerrar imediatamente antes do início do período letivo, concomitantemente com o encerramento do período de recesso escolar. A Figura SMatr01 ilustra o sequenciamento dos períodos supracitados.

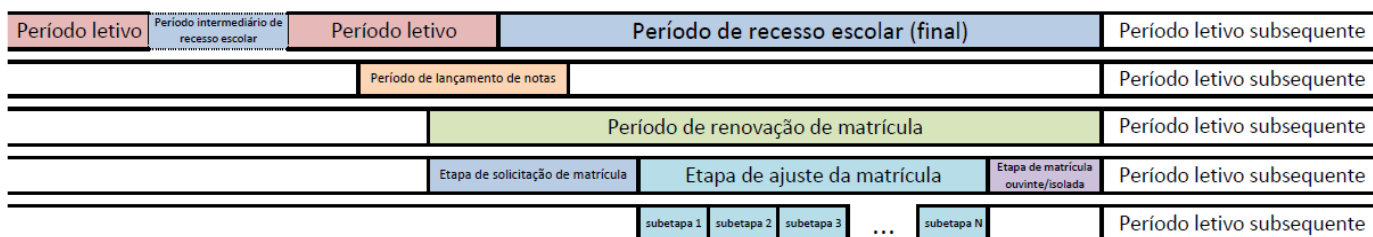


Figura SMatr01 - Organização do período letivo

RN12 - Etapas do período de renovação de matrícula: O período de renovação de matrícula é subdividido em três períodos necessariamente disjuntos e sucessivos, na seguinte ordem:

1. **Etapa de solicitação de matrícula:** primeira etapa do período de renovação de matrícula, que inicia com a composição das solicitações de renovação da matrícula por parte dos estudantes e encerra com o resultado das solicitações, no sentido da composição das turmas com os estudantes que solicitaram e obtiveram matrícula nelas, sendo que a definição e publicação do resultado ocorre exatamente ao final da etapa. Esta etapa pode iniciar antes do período de lançamento de notas, especificado em 'RN11 - Período de renovação de matrícula', mas deve encerrar apenas após o final desse período. Esta etapa é caracterizada por iniciar com as turmas não preenchidas, com exceção das turmas reservadas para calouros e por vagas reservadas em turmas, de acordo com a legislação;
2. **Etapa de ajuste de matrícula:** etapa voltada à realização de ajuste no resultado da matrícula obtido na etapa anterior. Esta etapa é caracterizada por iniciar com as turmas parcial ou totalmente preenchidas. Diferente da etapa anterior, esta etapa pode ser fracionada em subetapas disjuntas e sucessivas, sendo que cada subetapa gera um novo resultado de matrícula, que constitui a situação inicial da subetapa subsequente (quando houver uma).
3. **Etapa de matrícula ouvinte/isolada:** etapa voltada a solicitações de matrícula em disciplina isolada e matrícula para aluno-ouvinte, nos termos previstos na legislação acadêmica vigente. A etapa inicia com as turmas parcial ou totalmente preenchidas, a partir do resultado da etapa anterior.

A renovação de matrícula é obrigatória a todos os estudantes que estiverem cursando período letivo, sendo a não renovação considerada abandono (Art. 47 da Resolução 17/CUn/97, de 30 de setembro de 1997) - exceto para estudantes formandos, que não permanecerão sendo estudantes no período letivo subsequente. Assim, se um estudante não participar da etapa de solicitação de matrícula, necessariamente deve participar da etapa de ajuste de matrícula. Cada uma das etapas e subetapas acima descritas terá duração mínima de 24 horas, iniciando à zero hora da data de início e encerrando às 24 horas da data de término. A Figura SMatr01 ilustra a organização das etapas do período de renovação da matrícula.

RN13 - Estudantes calouro: Considera-se calouro aquele estudante que esteja matriculado para o seu primeiro período letivo, cuja matrícula é feita de forma compulsória. Aos calouros é permitida a alteração de sua matrícula compulsória na etapa de ajuste da matrícula.

RN14 - Fila de espera: No caso de não atendimento de pedido de matrícula por falta de vaga em turma, deve ser facultada ao estudante a possibilidade de optar por colocar-se em fila de espera para aquela turma, caso haja etapa ou subetapa posterior à da publicação do resultado. Também deve ser facultada ao estudante a possibilidade de solicitar matrícula em turma sem vaga disponível, sendo tal ato interpretado como o de colocar-se em fila de espera para aquela turma. O deferimento de matrícula em turma para estudante em fila de espera, durante a etapa de ajuste de matrícula, depende de disponibilização de vaga na turma, o que pode ocorrer por desistência de permanecer matriculado, por parte de estudante(s) que já havia(m) obtido matrícula na turma, e/ou por aumento de vaga(s) na turma.

RN15 - Ocupação de vaga em turma durante o período de renovação de matrícula: Considerando que a situação de um estudante o coloque dentre os aptos à ocupação de uma das vagas de uma turma que tenha sido solicitada por ele, a efetiva ocupação de vaga numa turma ocorrerá, em cada etapa, conforme descrito abaixo:

- A. Na primeira etapa, de solicitação de matrícula, as matrículas serão deferidas ao final da etapa;
- B. Nas subetapas que compõem a etapa de ajuste de matrícula, o deferimento das matrículas pode ocorrer:
 - a. Ao final da subetapa quando ela for configurada para processamento posterior, ou seja, o deferimento de vagas é precedido de um período inicial de pedidos de novas matrículas e de cancelamentos;
 - b. Imediatamente após a solicitação de pedidos de novas matrículas e de cancelamentos quando a subetapa for configurada para processamento imediato.
- C. Na etapa de matrícula ouvinte/isolada, o deferimento das matrículas pode ocorrer:
 - a. Ao final da subetapa quando ela for configurada para processamento posterior, ou seja, o deferimento de vagas é precedido de um período inicial de pedidos de novas matrículas e de cancelamentos;
 - b. Imediatamente após a solicitação de pedidos de novas matrículas e de cancelamentos quando a subetapa for configurada para processamento imediato.

RN16 - Tratamento excepcional de matrícula: A concessão ou a exclusão de matrícula em turma de disciplina em situação que caracterize excepcionalidade (motivo de força maior plenamente justificado), fora do período de renovação de matrícula, dar-se-á por decisão do colegiado do curso a que o estudante estiver vinculado ou por decisão do coordenador do curso *ad referendum* do correspondente colegiado, devendo ser procedida pela coordenação do curso, havendo a necessidade de registrar a justificativa que motivou a ação. A matrícula em turma é condicionada à disponibilidade de vaga, assim como às restrições curriculares previstas nas regras de negócio RN04 a RN08, observadas as exceções nelas previstas. Além disso, a matrícula em turma só pode ocorrer durante o primeiro quarto (25%) do período letivo.

RN17 - Responsabilidades no processo de renovação de matrícula: É de responsabilidade da PROGRAD a configuração de parâmetros do sistema de matrícula previstas nas regras de negócio RN01, RN04 e RN12. É de responsabilidade da coordenação o registro das permissões à não observância de restrições curriculares, conforme previsto nas regras de negócio RN04 a RN08, como também proceder ou alterar matrícula de estudante em caráter excepcional - em função de decisão do colegiado ou do coordenador, *ad referendum* do colegiado - sempre com a necessidade de registrar a justificativa para o tratamento excepcional. Considera-se aqui excepcionalidade alguma motivação não prevista explicitamente na legislação, como a interpretação da '*RN10 - Prioridade de estudante em turma*' à luz de informações tácitas ou o cumprimento de determinação judicial.

2 - Funcionalidades acessíveis na interface do estudante

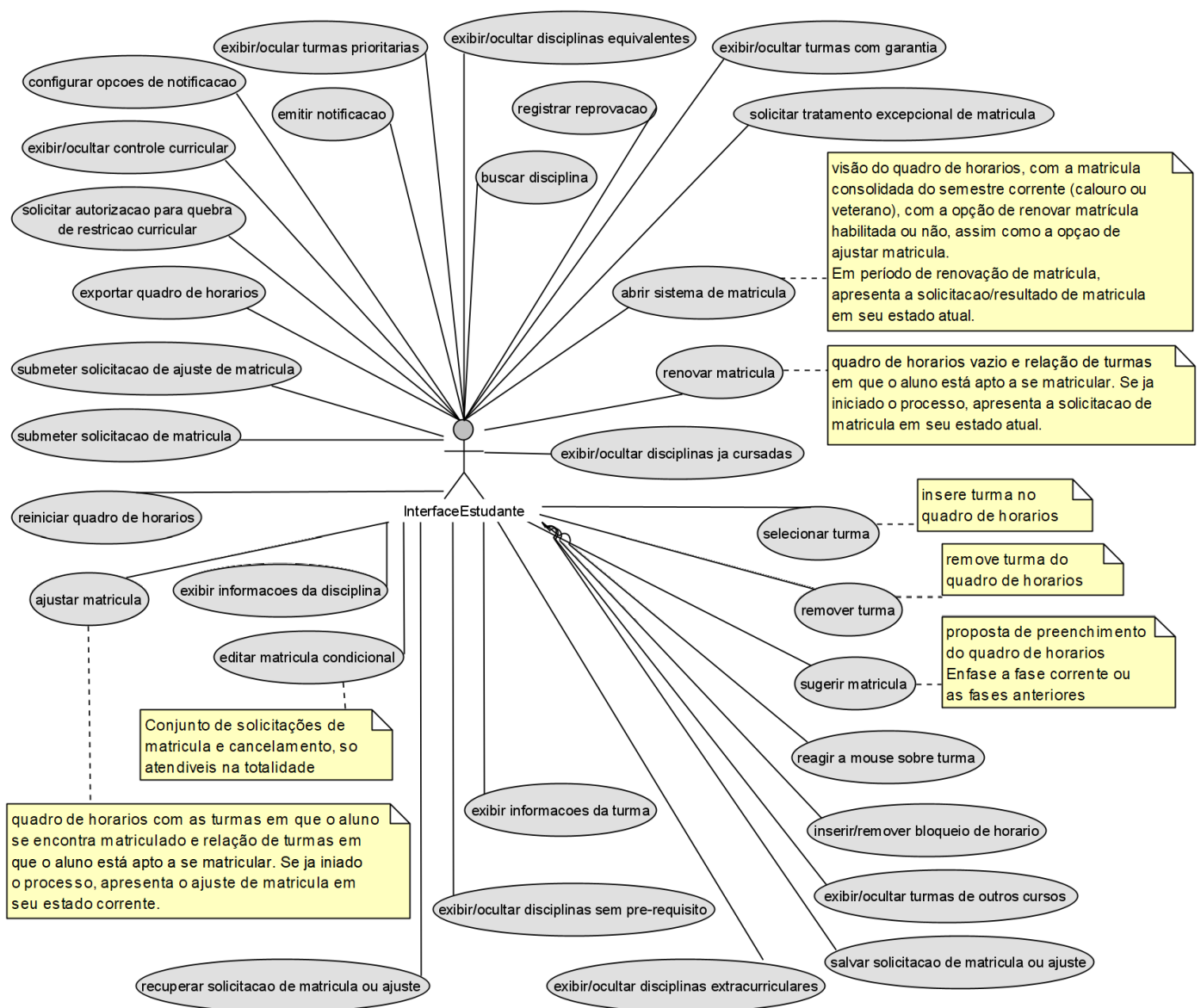


Figura SMatr02 - Funcionalidades do sistema de matrícula acessíveis ao estudante

Requisitos Funcionais (casos de uso)

Requisito Funcional SMatr01 - Abrir sistema de matrícula

Funcionalidade acessível na página inicial de acesso dos estudantes (página para que é direcionado após autenticar-se). Leva à página inicial do sistema de matrícula, que apresenta a visão do quadro de horários, com a matrícula consolidada do período letivo corrente (calouro ou veterano), com as opções renovar matrícula e ajustar matrícula, habilitadas ou não - dado que estas funcionalidades devem estar habilitadas apenas durante períodos específicos -, e a opção exportar quadro de horários, permanentemente habilitada. Durante o período de renovação de matrícula, a página inicial do sistema de matrícula apresenta a solicitação de matrícula (renovação ou ajuste) em seu estado corrente (quadro de horários vazio, no início do período, com a solicitação de matrícula ou com resultado de etapa).

A Figura SMatr03 apresenta um esboço de página inicial do estudante, que permite acesso ao sistema de matrícula. A Figura SMatr04 apresenta um esboço de página inicial do sistema de matrícula, com acesso às suas funcionalidades. Apenas funcionalidades habilitadas devem estar presentes nas opções apresentadas na página inicial do sistema de matrícula.



Figura SMatr03 - Esboço de página inicial de acesso dos estudantes (acesso ao sistema de matrícula)

Este esboço mostra a interface principal do sistema de matrícula. No topo, há uma barra de navegação com o Brasil, o texto 'BRASIL', 'Acesso à informação' e 'Partic'. Abaixo, um banner azul contém o brasão da Universidade Federal de Santa Catarina, o nome da instituição e o título 'Sistema de Matrícula'.

Abaixo do banner, há uma seção de informações do usuário:

Universidade Federal de Santa Catarina - UF SC
 Curso: Xxxxxx
 Aluno: Zezezeze

Horário	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
07:30 - 08:20	3		3	2		
08:20 - 09:10						
09:10 - 10:00						
10:10 - 11:00		1	2	1		
11:00 - 11:50						
13:30 - 14:20					4	
14:20 - 15:10	5					
15:10 - 16:00						
16:20 - 17:10				4	6	
17:10 - 18:00						
18:30 - 19:20						
19:20 - 20:10						
20:20 - 21:10						
21:10 - 22:00						

Abaixo da tabela, há um botão grande com o texto 'Controle Curricular'.

Figura SMatr04 - Esboço de página inicial do sistema de matrícula

Requisito Funcional SMatr02 - Renovar matrícula

Esta funcionalidade fica habilitada na página do sistema de matrícula apenas durante a etapa de solicitação de matrícula, do período de renovação de matrícula. Na primeira tentativa de renovação de matrícula do estudante, a funcionalidade carrega na página do sistema de matrícula um quadro de horários vazio e a relação de turmas em que o estudante está apto a se matricular (excluídas as turmas das disciplinas em que o estudante não tenha pré-requisito cumprido e as turmas das disciplinas já cursadas). Se já iniciado o processo de renovação de matrícula do estudante, apresenta a solicitação de matrícula em seu estado corrente. A Figura SMatr05 apresenta a um esboço da página do sistema de matrícula, após a primeira execução da funcionalidade.

Na página do sistema de matrícula (Figura SMatr05), no “**quadro controle curricular**” (que deve ter a possibilidade de exibir/ocultar), devem ser exibidas as seguintes informações:

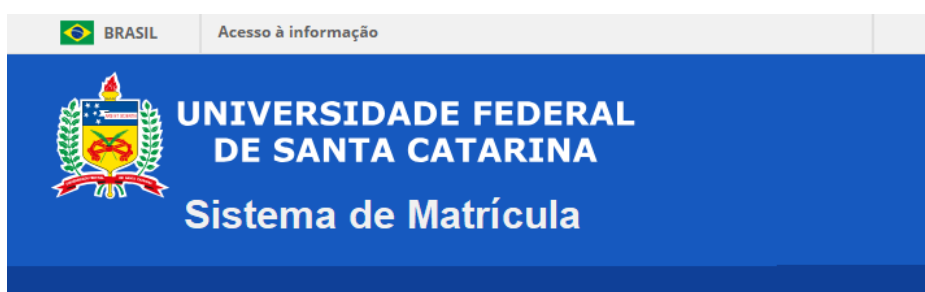
- Fase atual, tendo por base o período letivo de ingresso no curso;
- Situação em relação ao cumprimento das condições para a conclusão de seu curso (integralização curricular), de acordo com o currículo do curso, o que deve incluir:
 - **Carga horária total obrigatória** (total previsto, total integralizado, total matriculado, total restante a integralizar e os correspondentes percentuais para esses 4 valores);
 - **Carga horária de disciplinas obrigatórias** (total previsto, total integralizado, total matriculado, total restante a integralizar e os correspondentes percentuais para esses 4 valores);
 - **Carga horária de disciplinas optativas** (total previsto, total integralizado, total matriculado, total restante a integralizar e os correspondentes percentuais para esses 4 valores);
 - **Carga horária referente a estágio curricular** (total previsto, total integralizado, total matriculado, total restante a integralizar e os correspondentes percentuais para esses 4 valores);
 - **Carga horária referente a trabalho de conclusão de curso ou similar** (total previsto, total integralizado, total matriculado, total restante a integralizar e os correspondentes percentuais para esses 4 valores);
 - **Carga horária referente a atividades acadêmicas complementares** (total previsto, total integralizado, total matriculado, total restante a integralizar e os correspondentes percentuais para esses 4 valores);
 - **Carga horária referente a atividades acadêmicas de extensão** (total previsto, total integralizado, total matriculado, total restante a integralizar e os correspondentes percentuais para esses 4 valores);
 - **Carga horária prevista em legislação** (total previsto, total integralizado, total matriculado, total restante a integralizar e os correspondentes percentuais para esses 4 valores).
- Situação em relação às disciplinas do curso, às disciplinas já cursadas pelo estudante (registradas em seu histórico escolar) e as disciplinas em que ele esteja matriculado (e solicitando matrícula), explicitando:
 - Se disciplina já cursada, ainda não cursada, atualmente matriculada ou integralizada por disciplina(s) equivalente(s) cursada(s) ou por validação;
 - Cumprimento ou não dos pré-requisitos para disciplina não cursadas e nem integralizadas por equivalência ou validação;

Características da **apresentação de turmas**, destacadas na Figura SMatr05:

- As turmas das disciplinas que o estudante pode cursar são organizadas por fase-sugestão, o que inclui disciplinas obrigatórias e optativas.
- A descrição de cada turma de disciplina, além da identificação (nome e código da correspondente disciplina) e da sua quantidade de vagas, inclui a posição do estudante na lista de classificação daquela turma, de acordo com o previsto na *'RN10 - Prioridade de estudante em turma'*. Assim, se

o número referente à sua posição for menor ou igual à quantidade de vagas na turma, ele teria vaga garantida (ver detalhe na Figura SMatr06).

- Cada turma é identificada por uma cor. No caso da imagem, sugeridas as cores verde ou azul (ver Figura SMatr06):
 - Verde significa que o estudante tem vaga garantida na turma (situação descrita anteriormente);
 - Azul, que a posição do estudante na lista de classificação (o número correspondente) é superior à quantidade de vagas da turma. Ao solicitar matrícula em uma turma nesta situação, o estudante correria o risco de não obter vaga - o que aconteceria se aqueles que estão à sua frente na lista de classificação solicitassem matrícula na turma e a preenchessem.
- Ao escolher uma das opções de turma disponíveis, o estudante teria acesso a três funcionalidades, que são requisitos funcionais descritos a seguir (ver Figura SMatr07):
 - selecionar turma ('Requisito Funcional SMatr03 - Selecionar turma');
 - informações da turma ('Requisito Funcional SMatr06 - Exibir Informações da turma');
 - informações da disciplina ('Requisito Funcional SMatr07 - Exibir Informações da disciplina').



Universidade Federal de Santa Catarina - UFSC Curso: XXXXXX Aluno: Zezezeze						
Horário	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
07:30 - 08:20						
08:20 - 09:10						
09:10 - 10:00						
10:10 - 11:00						
11:00 - 11:50						
13:30 - 14:20						
14:20 - 15:10						
15:10 - 16:00						
16:20 - 17:10						
17:10 - 18:00						
18:30 - 19:20						
19:20 - 20:10						
20:20 - 21:10						
21:10 - 22:00						

Quadro de horários

botões das funcionalidades específicas da renovação de matrícula

Controle curricular

Quadro de controle curricular

	vagas	prioridade	
Fase 01	30	5	
	30	45	
Fase 02			
Fase 03			

Quadro de opções disponíveis

Figura SMatr05 - Esboço da página do sistema de matrícula apta à renovação de matrícula

	vagas	prioridade
Fase 01		
XYX9876 - Xxxz Zzzz	30	5
XYX9878 - Yyy Zzzz	30	45
Fase 02		

Figura SMatr06 - destaques na apresentação das turmas, na página do sistema de matrícula

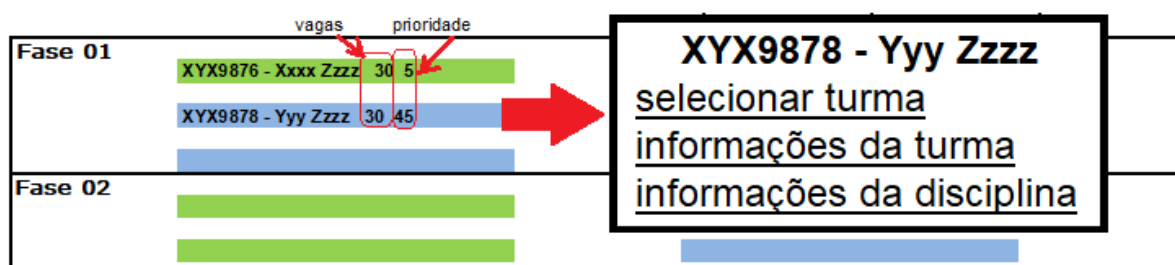


Figura SMatr07 - funcionalidades acessíveis na escolha de uma turma, na página do sistema de matrícula

Requisito Funcional SMatr03 - Selecionar turma

Funcionalidade acessível na página do sistema de matrícula, a ser usada para compor proposta de solicitação e de ajuste de matrícula. A funcionalidade selecionar turma consiste em uma das duas possibilidades a seguir:

- Inserir turma, escolhida do “quadro de opções disponíveis”, no “quadro de horários”, como ilustrado na Figura SMatr08;
- Inserir componente curricular que não se classifique como turma de disciplina, por não ser encaixável no quadro de horários, mas que tem carga horária associada e não tem restrição de quantidade de vagas (como estágio curricular, atividade acadêmica complementar e outros, de acordo com o currículo), escolhido do “quadro de opções disponíveis”, no espaço logo abaixo do quadro de horários, com a correspondente carga horária.

A execução da funcionalidade selecionar turma implicará na atualização automática do quadro de controle curricular (ver Requisito Funcional SMatr02 - Renovar matrícula).

Características da seleção de turma:

- A turma passa a ser representada no quadro de horários, nas posições (horários) correspondentes a seus dias e horários de aula;
- A turma passa a aparecer em destaque no quadro de opções disponíveis (indicando que foi selecionada);
- A seleção de turmas não é impedida por choque de horário, o que possibilita a realização de simulações de seleção, até o estudante chegar a uma situação de matrícula que satisfaça suas expectativas. Choques de horário não autorizados formalmente devem ser removidos antes da submissão de solicitação de renovação ou de ajuste de matrícula (ver requisitos funcionais correspondentes).



UNIVERSIDADE FEDERAL DE SANTA CATARINA

Sistema de Matrícula

Universidade Federal de Santa Catarina - UFSC

Curso: Xxxxxx

Aluno: Zezezeze

Horário	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
07:30 - 08:20	3		3			
08:20 - 09:10				2		
09:10 - 10:00			2			
10:10 - 11:00		1		1		
11:00 - 11:50						
13:30 - 14:20	5				4	
14:20 - 15:10						
15:10 - 16:00					6	
16:20 - 17:10				4		
17:10 - 18:00						
18:30 - 19:20						
19:20 - 20:10						
20:20 - 21:10						
21:10 - 22:00						

botões das funcionalidades específicas da renovação de matrícula

Controle curricular

	vagas	prioridade
Fase 01	1	2
	3 - XYX9878 - 30 45	4 - XYX9899 - 30 44
Fase 02	5	6
Fase 03		

Figura SMatr08 - Esboço da página do sistema de matrícula com seleção de turma

Requisito Funcional SMatr04 - Reagir a mouse sobre turma

Funcionalidade acessível quando o quadro de opções disponíveis estiver sendo exibido na interface (etapas de solicitação e ajuste de matrícula) e disponível em interfaces de equipamentos que usem mouse. Ao posicionar o cursor do mouse sobre uma das turmas do quadro de horários, a turma passa a ser representada no quadro de horários, nas posições (horários) correspondentes a seus dias e horários de aula, de forma similar ao definido no *'Requisito Funcional SMatr03 - Selecionar turma'*, porém, sem que a seleção de turma ocorra e apenas enquanto o cursor do mouse estiver posicionado sobre a turma.

Requisito Funcional SMatr05 - Inserir/remover bloqueio de horário

Funcionalidade acessível na página do sistema de matrícula, a ser usada para compor a proposta de solicitação e de ajuste de matrícula. Consiste em marcar ou desmarcar um ou mais horários do quadro de horários como indisponíveis para a alocação de turmas, com a possibilidade de incluir um texto de identificação desses horários (causa do bloqueio). Horários bloqueados são considerados indisponíveis para efeito de alocação de turma, seja por meio de *'Requisito Funcional SMatr03 - Selecionar turma'* ou de *'Requisito Funcional SMatr08 - Sugerir matrícula'*.

Requisito Funcional SMatr06 - Exibir Informações da turma

Funcionalidade acessível na página do sistema de matrícula, aplicável à turma escolhida no quadro de opções disponíveis. Consiste em exibir na interface do estudante informações referentes à turma escolhida:

- Nome e código da correspondente disciplina;
- Curso para que a turma é oferecida;
- Quantidade de vagas;
- Espaço(s) físico(s) alocado(s) para a turma (localização);
- Professor(es) da turma;
- Acesso ao plano de ensino da disciplina/turma (link).

Requisito Funcional SMatr07 - Exibir Informações da disciplina

Funcionalidade acessível na página do sistema de matrícula, aplicável à turma escolhida no quadro de opções disponíveis. Consiste em exibir na interface do estudante informações referentes à disciplina da turma escolhida:

- Informações básicas da disciplina (nome, cargas horárias teórica e prática, ementa)
- Objetivos da disciplina (fração do programa);
- Pré-requisito(s) da disciplina, se possuir;
- Correquisito(s) da disciplina, se possuir;
- Acesso ao programa de ensino da disciplina (link).

Requisito Funcional SMatr08 - Sugerir matrícula

Funcionalidade acessível na página do sistema de matrícula, durante as etapas de solicitação e de ajuste de matrícula, aplicável em duas situações

1. **Com o quadro de horários vazio:** (ver *'Requisito Funcional SMatr15 - Reiniciar quadro de horários'*) ou apenas com horário(s) bloqueado(s) (ver *'Requisito Funcional SMatr05 - Inserir/remover bloqueio de horário'*). Consiste em sugerir ao estudante um conjunto de turmas/disciplinas em que ele se matricularia, levando em conta o conjunto de disciplinas das fases-sugestão do currículo do curso do estudante e procurando gerar uma opção que fique (parâmetro a ser definido pelo estudante): o mais próxima possível do limite superior de carga

horária semestral, previsto no currículo; o mais próxima possível do limite inferior de carga horária semestral, previsto no currículo ou o mais próxima possível da carga horária semestral média, prevista no currículo. A sugestão de matrícula não poderá alocar turmas em horários previamente bloqueados. A alternativa preferencial desta funcionalidade é tentar preencher o quadro de horários exatamente com o conjunto de disciplinas da fase-sugestão do currículo, correspondente à fase atual do estudante (*tendo por base o período letivo de ingresso*), o que só será possível se ele já tiver sido aprovado em todas as disciplinas das fases anteriores à atual. Caso isso não seja possível, a construção da sugestão de matrícula será feita adotando uma das duas possibilidades de ênfase a seguir, a escolher (parâmetro a ser definido pelo estudante):

- a. **Ênfase à fase atual:** considerando o período letivo atual do estudante (“fase” atual), definido em função do seu período letivo de ingresso no curso, incluir o máximo possível de turmas da respectivo fase-sugestão e inserir disciplinas adicionais, de períodos letivos anteriores, caso haja essa possibilidade (considerando o limite de carga horária e a impossibilidade de choque de horário entre turmas). A prioridade desta opção é manter o estudante na maior quantidade possível de turmas do seu período letivo atual.
 - b. **Ênfase às fases anteriores:** considerando o período letivo atual do estudante (“fase” atual), definido em função do seu período letivo de ingresso no curso, incluir as turmas das fases-sugestão anteriores (com prioridade aos mais distantes), de modo a procurar eliminar atrasos (disciplinas de fases-sugestão anteriores ao que o estudante cursa). Após a alocação das disciplinas anteriores possíveis (considerando o limite de carga horária e a impossibilidade de choque de horário entre turmas), se possível, completar o quadro sugestão com disciplinas da fase-sugestão atual do estudante. A prioridade desta opção é matricular o estudante na maior quantidade possível de turmas de períodos letivos anteriores ao seu período letivo atual.
2. **Com o quadro de horários previamente preenchido:** Similar à situação anterior, o estudante precisa definir sua opção em relação à carga horária (máxima, mínima ou média) e à ênfase (fase atual ou fases anteriores). A funcionalidade deve buscar turmas de disciplinas que completem o quadro de horários previamente preenchido, levando em conta os parâmetros definidos.

Requisito Funcional SMatr09 - Exibir/ocultar controle curricular

Exibe ou oculta as informações do quadro controle curricular, descritas em '*Requisito Funcional SMatr02 - Renovar matrícula*'.

Requisito Funcional SMatr10 - Exibir/ocultar disciplinas já cursadas

Atualiza a relação de turmas disponíveis para composição de solicitação de matrícula ou ajuste, incluindo ou removendo as turmas das disciplinas já cursadas do quadro de opções disponíveis. A princípio, um estudante não cursa novamente uma disciplina em que já tenha sido aprovado, mas essa possibilidade existe. Assim, na composição inicial do quadro de opções disponíveis só são inseridas turmas de disciplinas ainda não cursadas, com aprovação. O presente requisito funcional visa possibilitar a inserção de turmas das disciplinas já cursadas no quadro de opções disponíveis, se estas não estiverem nele, ou removê-las, caso já estejam.

Requisito Funcional SMatr11 - Exibir/ocultar turmas de outros cursos

Atualiza a relação de turmas disponíveis para composição de solicitação de matrícula ou ajuste, incluindo ou removendo as turmas das disciplinas do currículo do curso do estudante, mas oferecidas a outros cursos. Há disciplinas para que são oferecidas muitas turmas, o que pode prejudicar a legibilidade do quadro de opções disponíveis. Assim, na composição inicial do quadro de opções disponíveis só são inseridas turmas pertencentes ao curso do estudante. O presente requisito funcional visa possibilitar a inserção de turmas das disciplinas de outros cursos no quadro de opções disponíveis, se estas não estiverem nele, ou removê-las, caso já estejam.

Requisito Funcional SMatr12 - Exibir/ocultar disciplinas sem pré-requisito

Atualiza a relação de turmas disponíveis para composição de solicitação de matrícula ou ajuste, incluindo ou removendo as turmas das disciplinas em que o estudante não tenha cumprido o pré-requisito. Por regra, o estudante não pode se matricular na turma de uma disciplina para que ele não tenha o pré-requisito cumprido, porém, como é possível obter autorização para cursar disciplina sem pré-requisito cumprido, é necessário dar ao estudante acesso às tais disciplinas. Assim, na composição inicial do quadro de opções disponíveis só são inseridas turmas de disciplinas com pré-requisito cumprido. O presente requisito funcional visa possibilitar a inserção de turmas das disciplinas sem pré-requisito cumprido no quadro de opções disponíveis, se estas não estiverem nele, ou removê-las, caso já estejam.

Requisito Funcional SMatr13 - Exibir/ocultar disciplinas extracurriculares

Atualiza a relação de turmas disponíveis para composição de matrícula ou ajuste, incluindo ou removendo as turmas das disciplinas que não estejam presentes no currículo do curso do estudante e em que ele já tenha cumprido o pré-requisito (neste caso, considera-se a existência de pré-requisito intrínseco de disciplina, definido pelo correspondente departamento de ensino, uma vez que se uma disciplina não está presente em um currículo de curso, este não pode ter pré-requisito previsto para ela, o que não significa que a disciplina possa ser cursada incondicionalmente, em relação a conhecimentos anteriores requeridos). Como um estudante pode se matricular em disciplinas que não estejam no currículo de seu curso, é necessário que ele tenha acesso às turmas/disciplinas correspondentes. Na composição inicial do quadro de opções disponíveis, essas turmas - que ocorrem em grande quantidade - não são exibidas. O presente requisito funcional visa possibilitar a exibição delas no quadro de opções disponíveis, se estas não estiverem nele, ou removê-las, caso já estejam.

Requisito Funcional SMatr14 - Exibir/ocultar disciplinas equivalentes

Atualiza a relação de turmas disponíveis para composição de matrícula ou ajuste, incluindo ou removendo as turmas das disciplinas ausentes do currículo, mas que correspondam a disciplinas equivalentes a disciplinas do currículo. Neste caso, no quadro de opções disponíveis, além das informações da turma, definidas em '*Requisito Funcional SMatr02 - Renovar matrícula*', deve ser informada a disciplina curricular de qua a disciplina da turma apresentada é equivalente. Na composição inicial do quadro de opções disponíveis, essas turmas - que ocorrem em grande quantidade - não são exibidas. O presente requisito funcional visa possibilitar a exibição delas no quadro de opções disponíveis, se estas não estiverem nele, ou removê-las, caso já estejam.

Requisito Funcional SMatr15 - Exibir/ocultar turmas com garantia

Atualiza a relação de turmas disponíveis para composição de matrícula ou ajuste, exibindo apenas as turmas em que o estudante tenha garantia de matrícula, em função de sua classificação, ou retorna à situação inicial da exibição de disciplinas (turmas do curso de disciplinas obrigatórias e optativas, organizadas por fase-sugestão).

Requisito Funcional SMatr16 - Exibir/ocultar turmas prioritárias

Atualiza a relação de turmas disponíveis para composição de matrícula ou ajuste, exibindo apenas as turmas com a prioridade estabelecida (parâmetros a definir), em função de sua classificação, ou retorna à situação inicial da exibição de disciplinas (turmas do curso de disciplinas obrigatórias e optativas, organizadas por fase-sugestão). A execução dessa funcionalidade exige a definição de 2 filtros:

- Posição de classificação limite (número inteiro para definir a posição máxima na fila de classificação);
- Fila de espera: definição se devem ser exibidas ou não turmas em que já haja fila de espera (turmas sem vaga disponível).

Requisito Funcional SMatr17 - Reiniciar quadro de horários

Funcionalidade acessível na página do sistema de matrícula, durante as etapas de solicitação e de ajuste de matrícula. Na etapa de solicitação de matrícula, remove do quadro de horários todas as disciplinas anteriormente selecionadas. Na etapa de ajuste de matrícula, carrega o quadro de horários resultante do processamento da solicitação de matrícula (de etapa ou subetapa anterior), sem as alterações procedidas durante a subetapa corrente da etapa de ajuste (seleções e remoções de turma). Esta funcionalidade, antes de ser executada, deve questionar se o estudante tem certeza de que deseja remover todas as disciplinas do quadro de horários. Também deve lembrar ao estudante a possibilidade de salvar o quadro de horários atual, para posterior recuperação. Os horários bloqueados não são alterados por esta funcionalidade.

Requisito Funcional SMatr18 - Remover turma

Funcionalidade acessível na página do sistema de matrícula durante as etapas de solicitação e de ajuste de matrícula. Em cada etapa tem finalidade distinta:

- **Remover turma / solicitação de matrícula:** remove do quadro de horários uma turma previamente selecionada;
- **Remover turma / ajuste de matrícula:** remove do quadro de horários uma turma previamente selecionada, para disciplinas selecionadas durante a subetapa corrente de ajuste. Para disciplinas em que o estudante já tenha sido matriculado durante a etapa de solicitação de matrícula ou subetapa anterior de ajuste, além da remoção da turma do quadro de horários, remove o estudante da relação de matriculados naquela turma, liberando sua vaga. Esta funcionalidade, antes de ser executada, especificamente para as turmas em que o estudante já esteja matriculado, deve questionar se o estudante tem certeza de que deseja remover a turma - pois, em caso de arrependimento, é possível que não possa dispor mais daquela vaga obtida anteriormente - e também deve sugerir a edição de matrícula condicional - e dar acesso a essa funcionalidade (ver *Requisito Funcional SMatr21 - Editar matrícula condicional*, que permite condicionar a remoção de uma ou mais turmas à obtenção de vaga em uma ou mais turmas).

Requisito Funcional SMatr19 - Salvar solicitação de matrícula ou ajuste

Funcionalidade acessível na página do sistema de matrícula, durante as etapas de solicitação e de ajuste de matrícula. Consiste em armazenar um quadro de horários produzido, possibilitando a sua recuperação posterior. Deve ser possível armazenar pelo menos dez quadros de horários, com a possibilidade de associação de texto de identificação para cada quadro.

Requisito Funcional SMatr20 - Recuperar solicitação de matrícula ou ajuste

Funcionalidade acessível na página do sistema de matrícula, durante as etapas de solicitação e de ajuste de matrícula. Consiste em carregar no quadro de horários sob edição um quadro de horários previamente armazenado.

Requisito Funcional SMatr21 - Submeter solicitação de matrícula

Funcionalidade acessível na página do sistema de matrícula durante a etapa de solicitação de matrícula. Consiste em submeter a solicitação de matrícula ao processamento da Universidade. Durante a etapa de solicitação de matrícula, o estudante pode executar essa funcionalidade mais de uma vez. Pode utilizá-la, por exemplo, para refazer sua solicitação de matrícula em função da perda de vaga (ou da iminência de perda) em uma turma em que tenha baixa prioridade.

Critérios de consistência a serem avaliados e que, se não cumpridos, impedem a submissão da solicitação de matrícula:

- Em relação a choque de horário entre disciplinas, deve ser observado o estabelecido na 'RN04 - Choque de horário';
- Em relação a carga horária, deve ser observado o estabelecido na 'RN05 - Carga horária';
- Em relação a disciplinas com pré-requisito, deve ser observado o estabelecido na 'RN06 - Pré-requisito';
- Em relação a disciplinas com correquisito, deve ser observado o estabelecido na 'RN07 - Correquisito';
- Em relação a outras restrições curriculares, deve ser observado o estabelecido na 'RN08 - Restrição curricular específica';

Como resultado da execução da funcionalidade, além das informações de quantidade de vagas e posição do estudante na lista de classificação, para turmas selecionadas, aparecerá também, **em tempo real (salvo se restrições técnicas acarretarem algum atraso)**, a posição atual do estudante na lista dos solicitantes daquela turma, para aquelas turmas em que ele não tenha vaga garantida (ver detalhe na Figura SMatr09), para permitir ao estudante monitorar sua situação. Isso possibilitará ao estudante saber se deixou de ter vaga numa turma antes do final da etapa de solicitação de matrícula (porque estudantes com maior prioridade preencheram as vagas disponíveis) ou saber se está próximo dessa situação. As turmas selecionadas no quadro de horários poderão estar representadas, após a execução da funcionalidade, de três formas distintas (representadas por cores, sendo sugeridas as cores verde, azul e magenta, no presente documento):

- Verde significa que o estudante tem vaga garantida na turma (situação descrita anteriormente);
- Azul, que o estudante está entre os aptos a ocupar uma vaga na turma, em função de sua classificação. Tal situação pode mudar na medida em que outros estudantes submetam suas solicitações de matrícula e coloquem o estudante em uma posição de classificação superior à quantidade de vagas da turma;
- Magenta, que o estudante não está entre os aptos a ocupar uma vaga na turma, em função de sua classificação, que o coloca (posição atual) além da quantidade de vagas da turma. A

princípio, esta situação indicaria a necessidade do estudante alterar sua solicitação de matrícula, em função da falta de perspectiva de obter vaga na(s) turma(s) nesta condição. A solicitação, porém, pode ser mantida nesta condição, com as seguintes consequências:

- O estudante se coloca em fila de espera para obtenção de vaga em turma (deve ser solicitada confirmação da intenção de colocar-se em fila de espera);
- Caso a não obtenção de vaga na(s) turma(s) pretendida(s) implique em carga horária abaixo do mínimo, deve ocorrer a advertência dessa situação (deve ser solicitada confirmação de que deseja manter a solicitação inalterada mesmo assim).

Esta funcionalidade deve atuar em duas situações distintas, durante a etapa de solicitação de matrícula:

- Antes do encerramento do período de lançamento de notas: neste íterim, as notas do período letivo corrente ainda não estão definidas e as disciplinas ora sendo cursadas não são consideradas para o cálculo do índice de matrícula (IM) e considera-se que o estudante tenha obtido aprovação nelas, para efeito de classificação e verificação de restrições curriculares (exceto se o estudante registrar a reprovação - ver funcionalidade correspondente);
- Após o encerramento do período de lançamento de notas: com a atualização do histórico escolar, decorrente do lançamento das notas do período letivo, e consequente atualização do índice de matrícula, assim como da situação das disciplinas do período letivo quanto à aprovação ou reprovação, a classificação de todos os estudantes deve ser redefinida (prioridade e posição atual).

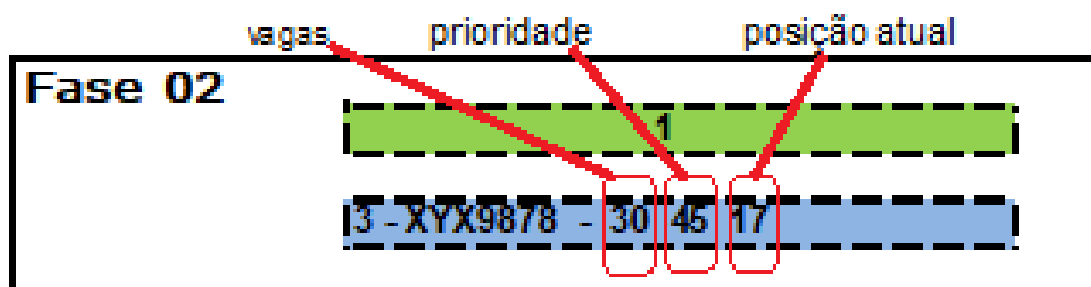


Figura SMatr09 - destaques na apresentação das turmas, na página do sistema de matrícula

Requisito Funcional SMatr22 - Ajustar matrícula

Esta funcionalidade fica habilitada na página do sistema de matrícula apenas durante a etapa de ajuste de matrícula. Na primeira execução, a funcionalidade carrega na página do sistema de matrícula o quadro de horários com as turmas em que o estudante esteja matriculado, a partir do processamento de sua solicitação, produzida na etapa de solicitação de matrícula, e a relação de turmas em que o estudante está apto a se matricular (excluídas as turmas em que o estudante já esteja matriculado, as turmas das disciplinas em que o estudante não tenha pré-requisito cumprido, as turmas das disciplinas já cursadas, as turmas de outros cursos e as turmas de disciplinas que não estejam no currículo do curso do estudante). Se já submetida uma solicitação de ajuste de matrícula, apresenta a solicitação de ajuste de matrícula em seu estado corrente. A Figura SMatr10 apresenta um esboço da página do sistema de matrícula, após a primeira execução da funcionalidade.

Características da apresentação de turmas, destacadas na figura SMatr10:

- Assim como na página de solicitação de matrícula, as turmas das disciplinas que o estudante pode cursar são organizadas por fase-sugestão, o que inclui disciplinas obrigatórias e optativas.
- A descrição de cada turma, além da identificação e da quantidade remanescente de vagas após a conclusão da etapa de solicitação de matrícula e da quantidade de vagas inicial da turma, inclui a posição do estudante na lista de classificação daquela turma. Assim, se o número referente à sua

posição for menor ou igual à quantidade de vagas ainda disponíveis na turma, ele teria vaga garantida.

- Cada turma é identificada por uma cor. No caso da imagem, sugeridas as cores verde, azul e magenta:
 - Verde significa que o estudante tem vaga garantida na turma (situação descrita anteriormente);
 - Azul, que a posição do estudante na lista de classificação (o número correspondente) é superior à quantidade de vagas ainda disponíveis na turma. Ao solicitar matrícula em uma turma nesta situação, o estudante correria o risco de não obter vaga - o que aconteceria se aqueles que estão à sua frente na lista de classificação solicitassem matrícula na turma e a preenchessem;
 - Magenta, que a turma não tem vagas disponíveis, isto é, todas as vagas foram preenchidas no processo de renovação de matrícula. Solicitar matrícula em uma turma nesta condição significa colocar-se em uma fila de espera, cujo atendimento da solicitação só pode ocorrer mediante aumento de vagas da turma, pelo respectivo departamento de ensino, ou em função de remoção da turma da matrícula de outros estudantes previamente matriculados nela.
- Ao escolher uma das opções de turma disponíveis, o estudante teria acesso a três funcionalidades, assim como na página de solicitação de matrícula (ver correspondentes requisitos funcionais, anteriormente descritos):
 - selecionar turma (*'Requisito Funcional SMatr03 - Selecionar turma'*);
 - informações da turma (*'Requisito Funcional SMatr06 - [Exibir] Informações da turma'*);
 - informações da disciplina (*'Requisito Funcional SMatr07 - [Exibir] Informações da disciplina'*).

Requisito Funcional SMatr23 - Editar matrícula condicional

Matrícula condicional é uma fração da solicitação de ajuste composta por uma ou mais seleções de turma (solicitações de matrícula) e uma ou mais solicitações de remoção de turma em que o estudante já esteja matriculado, em que ou há possibilidade de atendimento de todas as solicitações ou nenhuma alteração é procedida, isto é, o estudante só deixa de estar matriculado na(s) disciplina(s) prevista(s) para remoção se todo o conjunto de solicitações de matrícula puder ser atendido. Caso contrário, nenhuma seleção ou remoção de turma ocorrerá.

A presente funcionalidade consiste em criar uma matrícula condicional e compô-la ou alterá-la, com a inclusão de uma ou mais seleções de turma e/ou de uma ou mais remoções de turma.

A matrícula condicional passa a ser parte da etapa de ajuste de matrícula, a ser processada ao final da subetapa de ajuste em que foi feita (consideradas as duas possibilidades de ocupação de vaga previstas para as subetapas da etapa de ajuste da matrícula, previstas em *'RN15 - Ocupação de vaga em turma durante o período de renovação de matrícula'*).


Requisito Funcional SMatr24 - Submeter solicitação de ajuste de matrícula

Funcionalidade acessível na página do sistema de matrícula, durante a etapa de ajuste de matrícula. Consiste em submeter a solicitação de ajuste de matrícula ao processamento da Universidade. Durante a etapa de ajuste de matrícula, o estudante pode executar essa funcionalidade mais de uma vez. Os critérios de consistência de uma solicitação de ajuste são exatamente os mesmos previstos para uma solicitação de matrícula, considerando o total de turmas/disciplinas - as com matrícula previamente concedidas e as da solicitação de ajuste. Sobre critérios de consistência, ver aqueles definidos em *'Requisito Funcional SMatr21 - Submeter solicitação de matrícula'*. Critérios para atendimento da solicitação de ajuste observam o previsto em *RN15 - Ocupação de vaga em turma durante o período de renovação de matrícula*.



UNIVERSIDADE FEDERAL DE SANTA CATARINA

Sistema de Matrícula

 Universidade Federal de Santa Catarina - UF SC Curso: Xxxxxx Aluno: Zezezeze						
Horário	Segunda	Terça	Quarta	Quinta	Sexta	Sábado
07:30 - 08:20	3		3	2		
08:20 - 09:10						
09:10 - 10:00						
10:10 - 11:00		1	2	1		
11:00 - 11:50						
13:30 - 14:20					4	
14:20 - 15:10	5					
15:10 - 16:00						
16:20 - 17:10				4	6	
17:10 - 18:00						
18:30 - 19:20						
19:20 - 20:10						
20:20 - 21:10						
21:10 - 22:00						

botões das funcionalidades específicas do ajuste de matrícula

Controle curricular

	vaças	prioridade
Fase 01	XYX9888 - Xxxx Yttt 0/30	8
Fase 02	XYX9889 - Zccc Yttt 8/30	3
	XYX9855 - Baaa Yttt 5/30	8
Fase 03		



Figura SMatr10 - Esboço da página do sistema de matrícula apta ao ajuste de matrícula

Requisito Funcional SMatr25 - Exportar quadro de horários

Possibilita ao estudante exportar o quadro de horários em seu estado atual para impressão ou para salvar localmente em formatos diversos (pdf, xlsx e outros a definir).

Requisito Funcional SMatr26 - Solicitar autorização para quebra de restrição curricular

Funcionalidade permanentemente acessível na página do sistema de matrícula, voltada tanto a produzir uma solicitação para o colegiado do curso como a consultar a situação - quanto ao deferimento ou indeferimento - de autorização(ões) anteriormente solicitada(s). Uma solicitação de autorização demanda a apresentação de justificativa e pode referir-se a uma das seguintes possibilidades:

- ocorrência de choque de horário entre turmas em uma solicitação de matrícula, observando o estabelecido na RN04 - Choque de horário;
- matricular-se acima da carga máxima ou abaixo da carga mínima, observando o estabelecido na RN05 - Carga horária;
- descumprimento de pré-requisito, observando o estabelecido na RN06 - Pré-requisito;
- descumprimento de correquisito, observando o estabelecido na RN07 - Correquisito;
- descumprimento de restrição curricular específica do curso, observando o estabelecido na RN08 - Restrição curricular específica.

Requisito Funcional SMatr27 - Solicitar tratamento excepcional de matrícula

Funcionalidade acessível na página do sistema de matrícula, fora do período de renovação de matrícula, voltada tanto a produzir uma solicitação de tratamento excepcional de matrícula para o colegiado do curso como a consultar a situação de solicitação anteriormente submetida - quanto ao deferimento ou indeferimento. Além de especificar o ajuste solicitado, deve conter justificativa que caracterize motivo de força maior, isto é, uma excepcionalidade, na situação prevista na RN16 - Tratamento excepcional de matrícula.

Requisito Funcional SMatr28 - Configurar opções de notificação

Funcionalidade permanentemente acessível na página do sistema de matrícula, voltada a configurar as opções de notificação emitidas pelo sistema de matrícula para o estudante, que podem incluir e-mail, SMS e notificação na interface do sistema de matrícula, sendo que pelo menos uma das possibilidades deve estar selecionada. Deve ser possível também configurar a periodicidade de emissão de notificações (como repetir diariamente ou a cada dois dias enquanto a situação perdurar, por exemplo).

Requisito Funcional SMatr29 - Buscar disciplina

Funcionalidade acessível na página do sistema de matrícula, durante do período de renovação de matrícula, voltada à busca de disciplina, usando nome ou código da disciplina como critério de busca (completos ou incompletos). O resultado da busca pode retornar zero, um ou mais resultados de turma(s) associada(s) à disciplina buscada. A prioridade de classificação do estudante em cada uma das turmas retornadas deve ser informada, de modo a permitir-lhe escolher a opção mais adequada, segundo seu julgamento, conseguindo ponderar entre sua prioridade e adequação de horários. Selecionando uma das turmas retornadas, é possível executar uma das seguintes funcionalidades (anteriormente descritas):

- selecionar turma (*'Requisito Funcional SMatr03 - Selecionar turma'*);
- informações da turma (*'Requisito Funcional SMatr06 - [Exibir] Informações da turma'*);
- informações da disciplina (*'Requisito Funcional SMatr07 - [Exibir] Informações da disciplina'*).

Requisito Funcional SMatr30 - Emitir notificação

Funcionalidade com participação passiva do estudante e consiste na emissão de notificação ao estudante, nos termos configurados por ele (ver '*Requisito Funcional SMatr26 - Configurar opções de notificação*'), nas seguintes situações:

- Registro de decisão sobre solicitação de autorização para quebra de restrição curricular;
- Registro de decisão sobre solicitação de matrícula em caráter excepcional;
- Perda de vaga em turma em que solicitou matrícula causada por solicitações de outros estudantes;
- Encontrar-se abaixo da carga mínima em solicitação de matrícula submetida, por perda de vaga em turma;
- Encontrar-se abaixo da carga mínima no resultado da matrícula;
- Início ou final de etapa ou subetapa do período de renovação de matrícula;
- Emissão de resultado de etapa ou subetapa do período de renovação de matrícula.

Requisito Funcional SMatr31 - Registrar reprovação

Funcionalidade acessível na página do sistema de matrícula, durante a interseção da etapa de solicitação de matrícula com o período de lançamento de notas, voltada a permitir que o estudante marque uma disciplina do período letivo com nota não ainda registrada com o resultado reprovação com FS ou reprovação com FI (e, com isso, possa alterar a situação default, que é considerar disciplinas nessa condição como tendo obtido aprovação). Isso serve para corrigir a classificação do estudante e a sua posição nas turmas de sua solicitação de matrícula, especificamente durante o intervalo supracitado.

ANEXO B – PARECER DE APROVAÇÃO DO CEPESH-UFSC

PARECER CONSUBSTANCIADO DO CEP

DADOS DO PROJETO DE PESQUISA

Título da Pesquisa: Desenvolvimento de um front-end PWA para o módulo de matrícula da UFSC

Pesquisador: Ricardo Pereira e Silva

Área Temática:

Versão: 2

CAAE: 64772322.3.0000.0121

Instituição Proponente: Universidade Federal de Santa Catarina

Patrocinador Principal: Financiamento Próprio

DADOS DO PARECER

Número do Parecer: 5.795.847

Apresentação do Projeto:

Segundo o pesquisador:

Resumo:

Com o crescimento do uso de dispositivos móveis e acesso à internet no mundo (KEMP, 2022), as instituições e empresas tentam atingir esse novo público através do desenvolvimento de aplicativos nativos ou web com a abordagem Responsive Web Development (RWD). As aplicações nativas possuem a vantagem de explorar ao máximo os recursos do dispositivo e trazer uma melhor experiência para o usuário, todavia, o custo de desenvolvimento e manutenção são altos, uma vez que é necessário desenvolver um sistema para cada plataforma e dar suporte para cada um deles (MALAVOLTA, 2016; CORRAL et al., 2012). Para evitar esse problema, a RWD permite o desenvolvimento de um sistema independente da plataforma com o custo de perda na qualidade da experiência do usuário, uma vez que não pode aproveitar todos os recursos do dispositivo. Nesse cenário, a Progressive Web Application (PWA) surgiu para solucionar os problemas de ambas as abordagens, se aproveitando da independência de plataforma da RWD e dos recursos extras providos por aplicações nativas como as notificações push e possibilidade de instalação (SAM RICHARD, 2020). Diante desse contexto, o objetivo desse projeto será desenvolver uma interface alternativa com PWA para o novo módulo de matrícula da Universidade Federal de Santa Catarina (UFSC), que está sendo desenvolvido pela Secretaria de Tecnologia da Informação e Comunicação (SeTIC) no modelo tradicional de aplicações web, para

Endereço: Universidade Federal de Santa Catarina, Prédio Reitoria II, R: Desembargador Vitor Lima, nº 222, sala 701

Bairro: Trindade

CEP: 88.040-400

UF: SC

Município: FLORIANOPOLIS

Telefone: (48)3721-6094

E-mail: cep.propesq@contato.ufsc.br

Continuação do Parecer: 5.795.847

evitar a criação de duas interfaces (desktop e mobile), reduzir a quantidade de projetos que precisarão de manutenção e atingir os públicos-alvos do mobile e do desktop com uma única interface. Além disso, é esperado que essa interface apresente uma melhoria de usabilidade em relação à antiga, tornando-a mais intuitiva e fácil de usar.

Metodologia

Para atingir os objetivos descritos anteriormente, o método desta pesquisa será composto pelas seguintes etapas:

Conhecer as aplicações que são utilizadas pelos alunos, sejam elas oficiais ou não para observar as principais funcionalidades propostas e como cada aplicação aborda e soluciona os problemas associados ao pedido de matrícula;

Ler o documento de requisitos para identificar as funcionalidades que estão sendo propostas para o novo módulo de matrícula, focando apenas nas que estão associadas aos alunos e no registro do pedido de matrícula;

Desenvolver uma Proof of Concept (PoC) para estudar sobre as tecnologias associadas às PWAs, ganhando experiência e entendendo as limitações delas;

Desenvolver um servidor com escopo extremamente reduzido para o cliente PWA consumir a API, visto que não foi possível utilizar o servidor da SeTIC devido aos atrasos no desenvolvimento. Esse servidor torna-se necessário porque será realizado um teste de usabilidade no cliente proposto;

Desenvolver o cliente PWA, seguindo os requisitos que foram descritos no documento de requisitos, relacionados aos alunos e seu pedido de matrícula;

Implantar o servidor e o cliente em uma plataforma gratuita para a realização de um teste de usabilidade;

Elaborar o teste de usabilidade que será composto por um conjunto de tarefas que os usuários deverão realizar e dois questionários para mensurar a usabilidade da aplicação oficial do CAGR e da que está sendo proposta com o System Usability Scale (SUS); Submeter a pesquisa para o Comitê de Ética em Pesquisa com Seres Humanos da Universidade Federal de Santa Catarina (CEPSH-UFSC) para aprovação;

Executar o teste de usabilidade e coletar os dados oriundos dos questionários após a aprovação do CEPSH-UFSC;

Realizar uma comparação quantitativa entre as duas aplicações para avaliar se houve uma melhora de usabilidade na solução proposta através do cálculo do score do SUS; Verificar se o público-alvo considerou as novas funcionalidades (notificações, ver espelho de matrícula e grade de horários

Endereço: Universidade Federal de Santa Catarina, Prédio Reitoria II, R: Desembargador Vitor Lima, nº 222, sala 701

Bairro: Trindade

CEP: 88.040-400

UF: SC

Município: FLORIANOPOLIS

Telefone: (48)3721-6094

E-mail: cep.propesq@contato.ufsc.br

Continuação do Parecer: 5.795.847

estando sem conexão e visualizar o status da vaga) como úteis. Em relação ao teste de usabilidade, ele pode ser dividido nas seguintes etapas: avaliação do módulo oficial através de um questionário, execução de 6 tarefas no novo módulo para simular tarefas reais, e avaliação das tarefas que foram realizadas e do novo módulo através de outro questionário. Todas as etapas desse teste serão feitas de forma remota e assíncrona e as respostas serão anônimas para assegurar a privacidade, sigilo e confidencialidade de todos os participantes. Além disso, os questionários serão feitos através do Google Forms e os dados armazenados no Google Drive. Acerca da primeira avaliação, serão realizadas 10 perguntas do questionário SUS para mensurar a usabilidade do sistema. O questionário deve durar cerca de 10 minutos e seus resultados serão utilizados para realizar uma análise comparativa com a aplicação proposta. Sobre as tarefas, elas serão enviados por e-mail para cada participante do teste por meio de um documento com as instruções necessárias e o objetivo delas é permitir que o participante explore todas as funcionalidades propostas para poder avaliá-las no segundo questionário. Nessas tarefas, há uma descrição breve do contexto imaginário que o participante estará inserido e um conjunto de atividades que devem ser realizadas para concluir a tarefa. As atividades não devem conter um passo a passo claro do que deve ser feito, visto que isso afetaria a avaliação da usabilidade do sistema já que ele deve ser fácil de entender e usar. Os participantes terão 1 dia para executar todas as tarefas. Esse modelo de teste usabilidade foi baseado no que é feito pelo Laboratório Bridge para a realização dos testes das versões. Após a conclusão das tarefas, o participante terá liberdade para mexer no módulo proposto.

Objetivo da Pesquisa:

Segundo o pesquisador:

Objetivo:

Esse trabalho tem como objetivo o desenvolvimento de um cliente PWA alternativo para o módulo de matrícula da UFSC e torná-lo mais intuitivo e fácil de utilizar pelos alunos.

Avaliação dos Riscos e Benefícios:

Segundo o pesquisador:

Riscos:

No nosso estudo, os possíveis riscos ou desconfortos decorrentes da participação na pesquisa são desconforto ou estresse por não conseguir concluir uma tarefa e, caso isso aconteça, o participante deve anotar no documento de tarefas que não foi possível concluir junto com o

Endereço: Universidade Federal de Santa Catarina, Prédio Reitoria II, R: Desembargador Vitor Lima, nº 222, sala 701

Bairro: Trindade

CEP: 88.040-400

UF: SC

Município: FLORIANOPOLIS

Telefone: (48)3721-6094

E-mail: cep.propesq@contato.ufsc.br

Continuação do Parecer: 5.795.847

motivo para não ter concluído (ex:não achar o botão ou ocorrer um erro) e passar para a próxima tarefa. Outro risco seria a impossibilidade de acessar a aplicação para as cinco primeiras tarefas devido à falta de conexão com a internet e outros possíveis problemas, e, nesse caso, o participante deve entrar em contato com a pesquisadora assistente, Maria Eduarda de Melo Hang (maira.eduarda.melo.hang@gmail.com), para avisar sobre a impossibilidade de realizar as tarefas.

Benefícios:

Os benefícios seriam possuir uma avaliação feita pelo público-alvo sobre a facilidade de uso e aprendizado dos dois módulos, permitir que os pesquisadores comparem os dois módulos para verificar se houve uma melhora na usabilidade no módulo proposto e conferir se as novas funcionalidades que foram propostas são realmente úteis para os alunos da UFSC. Todas as informações poderão ser repassadas para a instituição responsável pelo módulo oficial para melhorá-lo, tornando-o mais fácil e simples de utilizar.

Comentários e Considerações sobre a Pesquisa:

A pesquisa faz parte do Trabalho de Conclusão de Curso da acadêmica do curso de Ciência da Computação, Maria Eduarda de Melo Hang, orientada por Ricardo Pereira e Silva, professor do Departamento de Informática e Estatística, INE, no CTC da UFSC. A pesquisa pedirá aos participantes, alunos da Universidade Federal de Santa Catarina, que já tiveram contato com o sistema de matrícula do CAGR, para avaliarem a usabilidade dos dois módulos de matrícula - o oficial que está no CAGR, e o proposto pelo projeto-, utilizando o modelo System Usability Scale, que permitirá a realização de uma comparação quantitativa entre os dois módulos. A participação dos alunos será voluntária e os convites serão feitos pelo e-mail do curso e canais de comunicação do Laboratório Bridge. As etapas serão virtuais e assíncronas. É esperado que a aplicação proposta apresente uma melhora de usabilidade, obtendo um escore superior ao módulo oficial, e que os alunos sintam-se satisfeitos com as novas funcionalidades desenvolvidas. É um estudo nacional, realizado com financiamento próprio e serão pesquisados 20 alunos veteranos. Previsão de início do estudo em 13/12/22 e término em 16/12/22 (no formulário PB).

Considerações sobre os Termos de apresentação obrigatória:

Vide campo "Conclusões ou Pendências e Lista de Inadequações.

Recomendações:

Vide campo "Conclusões ou Pendências e Lista de Inadequações.

Endereço: Universidade Federal de Santa Catarina, Prédio Reitoria II, R: Desembargador Vitor Lima, nº 222, sala 701
Bairro: Trindade **CEP:** 88.040-400
UF: SC **Município:** FLORIANOPOLIS
Telefone: (48)3721-6094 **E-mail:** cep.propesq@contato.ufsc.br

Continuação do Parecer: 5.795.847

Conclusões ou Pendências e Lista de Inadequações:

Este comitê aprova o projeto de pesquisa, considerando que as pendências e inadequações foram resolvidas nesta versão.

Considerações Finais a critério do CEP:

Este parecer foi elaborado baseado nos documentos abaixo relacionados:

Tipo Documento	Arquivo	Postagem	Autor	Situação
Informações Básicas do Projeto	PB_INFORMAÇÕES_BÁSICAS_DO_PROJETO_2042743.pdf	25/11/2022 11:50:31		Aceito
Outros	CartaResposta.pdf	25/11/2022 11:49:52	MARIA EDUARDA DE MELO HANG	Aceito
Projeto Detalhado / Brochura Investigador	projeto.pdf	24/11/2022 09:32:56	MARIA EDUARDA DE MELO HANG	Aceito
TCLE / Termos de Assentimento / Justificativa de Ausência	TCLE.pdf	24/11/2022 09:31:04	MARIA EDUARDA DE MELO HANG	Aceito
Projeto Detalhado / Brochura Investigador	projetoFinal.pdf	02/11/2022 11:10:37	Ricardo Pereira e Silva	Aceito
Folha de Rosto	folhaDeRostoAssinada.pdf	02/11/2022 11:09:44	Ricardo Pereira e Silva	Aceito
Outros	FormularioPWA.pdf	31/10/2022 16:21:42	MARIA EDUARDA DE MELO HANG	Aceito
Outros	FormularioCAGR.pdf	31/10/2022 16:21:27	MARIA EDUARDA DE MELO HANG	Aceito

Situação do Parecer:

Aprovado

Necessita Apreciação da CONEP:

Não

FLORIANOPOLIS, 06 de Dezembro de 2022

Assinado por:
Nelson Canzian da Silva
(Coordenador(a))

Endereço: Universidade Federal de Santa Catarina, Prédio Reitoria II, R: Desembargador Vitor Lima, nº 222, sala 701
Bairro: Trindade **CEP:** 88.040-400
UF: SC **Município:** FLORIANOPOLIS
Telefone: (48)3721-6094 **E-mail:** cep.propesq@contato.ufsc.br

Continuação do Parecer: 5.795.847

Endereço: Universidade Federal de Santa Catarina, Prédio Reitoria II, R: Desembargador Vitor Lima, nº 222, sala 701
Bairro: Trindade **CEP:** 88.040-400
UF: SC **Município:** FLORIANOPOLIS
Telefone: (48)3721-6094 **E-mail:** cep.propesq@contato.ufsc.br