

Jorge Habib El Khouri  
Rodrigo Kenji Aguená  
Rodrigo Miksian Magaldi

**USPolis: Desenvolvimento e expansão de um  
produto digital embasado nas necessidades dos  
usuários**

São Paulo, SP

2023

Jorge Habib El Khouri  
Rodrigo Kenji Aguená  
Rodrigo Miksian Magaldi

## **USPolis: Desenvolvimento e expansão de um produto digital embasado nas necessidades dos usuários**

Trabalho de conclusão de curso apresentado ao Departamento de Engenharia de Computação e Sistemas Digitais da Escola Politécnica da Universidade de São Paulo para obtenção do Título de Engenheiro.

Universidade de São Paulo – USP

Escola Politécnica

Departamento de Engenharia de Computação e Sistemas Digitais (PCS)

Orientador: Prof. Dr. Fábio Levy Siqueira

Coorientador: Renan de Luca Avila

São Paulo, SP

2023

# Agradecimentos

Às nossas famílias, que nos deram todo o suporte necessário para que pudéssemos chegar aonde chegamos, prestes a nos tornar engenheiros pela Escola Politécnica da USP.

Aos nossos colegas, pela colaboração ativa e pelas contribuições valiosas que desempenharam um papel crucial ao testar e validar nosso produto. Sua participação foi essencial para o aprimoramento do projeto, feito por politécnicos e para politécnicos.

Ao nosso orientador Prof. Dr. Fábio Levy Siqueira, que acompanhou o desenvolvimento deste trabalho do início ao fim, sempre com muita paciência e sabedoria, contribuindo imensamente para nosso desenvolvimento acadêmico e profissional. Seu carisma e bom humor tornaram o desenvolvimento do projeto de formatura divertido e leve, mesmo neste período conturbado do fim da graduação.

E ao Renan, nosso coorientador, que mesmo em meio às suas responsabilidades profissionais nunca deixou de lado a paixão por este projeto que iniciou há oito anos, e fez questão de acompanhar seu desenvolvimento de perto. Obrigado por nos nortear e por depositar em nós tanta confiança para tomar nossas próprias decisões.

# Resumo

No dia a dia da universidade é muito comum observar alunos - em especial novos ingressantes - sentindo-se perdidos em relação à sua grade horária. Como há muitas informações em locais diferentes, os alunos frequentemente não sabem ao exato em que salas estão alocadas suas turmas, nem onde ficam tais salas dentro dos diversos prédios da faculdade.

No contexto desse desafio, este trabalho dedicou-se à atualização e aprimoramento do aplicativo móvel USPolis, já existente mas desatualizado. Este esforço resultou em uma nova versão do aplicativo, que reúne informações sobre horários de aulas para cada turma, alocações das salas de aula e mapas dos prédios presentes na faculdade, introduzindo nessa nova versão melhorias em termos de funcionalidade e experiência do usuário.

Foram também aplicados os conceitos de Design Sprint, testes A/B e DevOps para expandir a aplicação, gerando novas funcionalidades com base nas necessidades e visões dos próprios alunos usuários.

**Palavras-chave:** EPUSP, Aplicativo Móvel, Design Sprint, Teste A/B, DevOps, Alocação de Salas de Aula

# Abstract

In the day-to-day life of the university, it is very common to observe students, especially freshmans, feeling lost regarding their class schedules. Since there is a lot of information in different places, students often don't know exactly which rooms their classes are allocated to, or where these rooms are located within the various buildings of the university. It is very common to see situations where students ask in groups or in person where a particular class will take place.

Based on this problem, this work presents a new version of the mobile application USPolis, which gathers information about class schedules for each class, classroom allocations, and maps of the buildings present in the university. With this, each student will be able to keep track of their own schedule and find their classes in the university more easily.

Concepts such as Design Sprint, A/B testing, and DevOps were also applied to expand the application, introducing new features based on the needs and perspectives of the student users themselves.

**Keywords:** EPUSP, Mobile Application, Design Sprint, A/B Testing, DevOps, Classroom Allocation

# Lista de ilustrações

Figura 1 – Agenda do Design Sprint. . . . .	15
Figura 2 – Gráfico de eficiência em relação ao número de entrevistados. . . . .	19
Figura 3 – Ciclo de experimentação de um produto . . . . .	20
Figura 4 – Exemplo de ramificação no Git. . . . .	26
Figura 5 – Diagrama esquematizando Integração Contínua. Alterações no código são integrados ao código de build do projeto . . . . .	28
Figura 6 – Integração entre CI e CD . . . . .	29
Figura 7 – Aplicativo USPólis Legado - Área de Pesquisa de Aulas . . . . .	32
Figura 8 – Aplicativo USPólis Legado - Área de Minhas Aulas . . . . .	33
Figura 9 – Aplicativo USPólis Legado - Área de Mapas . . . . .	33
Figura 10 – Gestão de salas no sistema administrativo . . . . .	34
Figura 11 – Gestão de turmas no sistema administrativo . . . . .	35
Figura 12 – Alocações de turmas em salas no sistema administrativo . . . . .	35
Figura 13 – Visualização de horários no sistema MatrUSP . . . . .	36
Figura 14 – Sistema MRBS utilizado no prédio de engenharias elétrica e da computação	37
Figura 15 – <i>Design</i> da página de Minhas Aulas do novo USPólis dentro do Figma .	39
Figura 16 – Arquitetura Simplificada do Sistema USPólis . . . . .	45
Figura 17 – Funcionamento: Github Actions + Expo . . . . .	46
Figura 18 – Arquitetura Completa do Sistema USPólis . . . . .	48
Figura 19 – Taxa de acesso dos usuários ao <i>app</i> entre o período de Julho a Setembro	49
Figura 20 – Comparação USPólis Legado e USPólis Novo - Listagem de Aulas . . . .	52
Figura 21 – Comparação USPólis Legado e USPólis Novo - Detalhe de Aula . . . .	52
Figura 22 – Comparação USPólis Legado e USPólis Novo - Mapas . . . . .	53
Figura 23 – Comparação USPólis Legado e USPólis Novo - Minhas Aulas . . . . .	53
Figura 24 – Aba de <i>feedback</i> no USPólis . . . . .	54
Figura 25 – Como poderíamos - Mapas . . . . .	55
Figura 26 – Como poderíamos - Entidades . . . . .	55
Figura 27 – Como poderíamos - Restaurantes universitários e filas . . . . .	56
Figura 28 – Como poderíamos - Disciplinas . . . . .	56
Figura 29 – Como poderíamos - Experiência do usuário . . . . .	56
Figura 30 – Como poderíamos - Resultados da votação . . . . .	57
Figura 31 – Jornada do usuário no aplicativo . . . . .	58
Figura 32 – Jornada do usuário no aplicativo com a inserção de pontos de melhoria	58
Figura 33 – Rascunhos realizados pelo grupo . . . . .	59
Figura 34 – Rascunho final da solução proposta por um dos integrantes do grupo .	60
Figura 35 – Rascunho final da solução proposta por um dos integrantes do grupo .	60

Figura 36 – Rascunho final da solução proposta por um dos integrantes do grupo . . .	61
Figura 37 – Funcionalidades mais votadas nas soluções finais rascunhadas . . . . .	61
Figura 38 – Jornada da busca facilitada por aulas . . . . .	62
Figura 39 – Jornada da avaliação de disciplinas . . . . .	62
Figura 40 – Jornada da inclusão de grupos de extensão e disciplinas optativas . . .	63
Figura 41 – Prototipação da funcionalidade escolhida . . . . .	63
Figura 42 – Versão alternativa (versão B) do buscador de disciplinas . . . . .	66
Figura 43 – Listagem de eventos no USPolis . . . . .	67
Figura 44 – Detalhes de um evento no USPolis . . . . .	68
Figura 45 – Número de usuários ativos no USPolis por dia . . . . .	69
Figura 46 – Usuários ativos em cada grupo de teste . . . . .	70
Figura 47 – Resultado do Teste A/B - Aulas adicionadas com o buscador no tempo	71
Figura 48 – Resultado do Teste A/B - Agregado de aulas adicionadas . . . . .	71
Figura 49 – Volume de eventos ao longo do tempo: aula adicionada e curso selecionado	72
Figura 50 – Aulas adicionadas antes da funcionalidade de buscador de disciplinas (abril a maio) . . . . .	72
Figura 51 – Aulas adicionadas após a funcionalidade de buscador de disciplinas (agosto a setembro) . . . . .	72
Figura 52 – Taxa de retorno de usuários - antes da funcionalidade de eventos . . . .	74
Figura 53 – Taxa de retorno de usuários - depois da funcionalidade de eventos . . .	74
Figura 54 – Uso das abas no USPolis . . . . .	75
Figura 55 – Distribuição dos tempos de sessão dos usuários . . . . .	75
Figura 56 – Porcentagem de usuários ativos que são novos usuários . . . . .	76

# Lista de tabelas

Tabela 1 – Comparativo de funcionalidades presentes no USPolis, MatrUSP e MRBS	37
Tabela 2 – Cronograma do projeto . . . . .	38
Tabela 3 – Endpoints da API Rest do USPolis . . . . .	47
Tabela 4 – Número de comentários positivos, neutros e negativos sobre os critérios C1 a C5 após o teste com usuários do Design Sprint . . . . .	64
Tabela 5 – Número de downloads por plataforma . . . . .	76



# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>11</b>
<b>1.1</b>	<b>Objetivo</b>	<b>11</b>
<b>1.2</b>	<b>Justificativa</b>	<b>12</b>
<b>1.3</b>	<b>Organização do Trabalho</b>	<b>12</b>
<b>2</b>	<b>ASPECTOS CONCEITUAIS</b>	<b>14</b>
<b>2.1</b>	<b>Design Sprint</b>	<b>14</b>
2.1.1	Dia 1	15
2.1.2	Dia 2	16
2.1.3	Dia 3	17
2.1.4	Dia 4	18
2.1.5	Dia 5	18
<b>2.2</b>	<b>Teste A/B</b>	<b>19</b>
2.2.1	Experimentação e testes	19
2.2.2	Fatores decisivos para um bom experimento	20
2.2.3	Como executar um experimento <i>online</i>	22
<b>2.3</b>	<b>DevOps</b>	<b>23</b>
2.3.1	Introdução e Princípios	23
2.3.2	Controle de Versão	24
2.3.3	Integração Contínua	25
2.3.3.1	Conflitos na Integração	26
2.3.3.2	Características da Integração Contínua	26
2.3.4	Implantação Contínua	28
2.3.5	Entrega Contínua	29
<b>2.4</b>	<b>Considerações Finais do Capítulo</b>	<b>30</b>
<b>3</b>	<b>TRABALHOS RELACIONADOS</b>	<b>31</b>
<b>3.1</b>	<b>Aplicativo Legado do USPolis</b>	<b>32</b>
<b>3.2</b>	<b>USPolis - Administrativo</b>	<b>34</b>
<b>3.3</b>	<b>Concorrentes</b>	<b>35</b>
3.3.1	MatrUSP	36
3.3.2	Meeting Room Booking System - MRBS	36
<b>3.4</b>	<b>Considerações Finais do Capítulo</b>	<b>37</b>
<b>4</b>	<b>MÉTODO DO TRABALHO</b>	<b>38</b>
<b>4.1</b>	<b>Primeiro ciclo - recriação do aplicativo legado</b>	<b>38</b>

4.2	Segundo ciclo - execução do Design Sprint . . . . .	39
4.3	Terceiro e quatro ciclos - busca automatizada de disciplinas e eventos	40
4.4	Considerações Finais do Capítulo . . . . .	40
5	<b>ESPECIFICAÇÃO DE REQUISITOS . . . . .</b>	<b>41</b>
5.1	Funcionalidades pré-existentes no aplicativo legado . . . . .	41
5.2	Funcionalidades idealizadas no Design Sprint . . . . .	42
5.3	Considerações Finais do Capítulo . . . . .	43
6	<b>ARQUITETURA E TECNOLOGIAS . . . . .</b>	<b>44</b>
6.1	Arquitetura Geral do Sistema . . . . .	44
6.2	Aplicativo Móvel . . . . .	44
6.3	API Rest . . . . .	46
6.3.1	Arquitetura Completa . . . . .	47
6.4	<b>Serviços Utilizados . . . . .</b>	<b>48</b>
6.4.1	Amplitude . . . . .	48
6.4.2	SendGrid . . . . .	49
6.5	Considerações Finais do Capítulo . . . . .	49
7	<b>DESENVOLVIMENTO DO TRABALHO . . . . .</b>	<b>51</b>
7.1	<b>Entrega Inicial . . . . .</b>	<b>51</b>
7.2	<b>Design Sprint . . . . .</b>	<b>54</b>
7.2.1	Entrevistas com Usuários . . . . .	54
7.2.2	Como Poderíamos . . . . .	55
7.2.3	Jornada do Usuário . . . . .	57
7.2.4	Problema a Ser Resolvido . . . . .	58
7.2.5	Elaboração de Propostas de <i>Design</i> . . . . .	59
7.2.5.1	Soluções Presentes no Mercado e Rascunhos . . . . .	59
7.2.5.2	Proposta de Rascunhos Finais . . . . .	59
7.2.6	<i>Storyboard</i> . . . . .	62
7.2.7	Decisão e Prototipação Final . . . . .	63
7.2.8	Testes Finais com Usuários . . . . .	64
7.3	<b>Buscador de Disciplinas . . . . .</b>	<b>65</b>
7.3.1	Descrição da Funcionalidade . . . . .	65
7.3.2	Teste A/B . . . . .	65
7.4	<b>Eventos . . . . .</b>	<b>66</b>
7.4.1	Descrição da Funcionalidade . . . . .	67
7.4.2	Lançamento da Funcionalidade . . . . .	68
7.5	Considerações Finais do Capítulo . . . . .	68
8	<b>RESULTADOS . . . . .</b>	<b>69</b>

<b>8.1</b>	<b>Versão inicial do USPolis</b> . . . . .	<b>69</b>
<b>8.2</b>	<b>Teste A/B - Buscador de Disciplinas</b> . . . . .	<b>70</b>
<b>8.3</b>	<b>Eventos</b> . . . . .	<b>73</b>
<b>8.4</b>	<b>Outros resultados</b> . . . . .	<b>74</b>
8.4.1	Uso das Abas do Aplicativo . . . . .	74
8.4.2	Distribuição dos Tempos de Sessão . . . . .	75
8.4.3	Porcentagem de Novos Usuários . . . . .	76
8.4.4	Instalações . . . . .	76
<b>8.5</b>	<b>Considerações finais do capítulo</b> . . . . .	<b>77</b>
<b>9</b>	<b>CONSIDERAÇÕES FINAIS</b> . . . . .	<b>78</b>
<b>9.1</b>	<b>Contribuições</b> . . . . .	<b>78</b>
<b>9.2</b>	<b>Perspectivas de Continuidade</b> . . . . .	<b>79</b>
	<b>REFERÊNCIAS</b> . . . . .	<b>80</b>
	<b>APÊNDICE A – RESULTADOS DAS ENTREVISTAS COM OS USUÁRIOS</b> . . . . .	<b>82</b>
<b>A.1</b>	<b>Usuário 1 - Engenharia Elétrica, 4º ano</b> . . . . .	<b>82</b>
<b>A.2</b>	<b>Usuário 2 - Engenharia da Computação, 4º ano</b> . . . . .	<b>83</b>
<b>A.3</b>	<b>Usuário 3 - Engenharia Elétrica, 1º ano</b> . . . . .	<b>84</b>
<b>A.4</b>	<b>Usuário 4 - Engenharia de Produção, 5º ano</b> . . . . .	<b>85</b>
<b>A.5</b>	<b>Usuário 5 - Engenharia Naval, 4º ano</b> . . . . .	<b>86</b>

# 1 Introdução

A Escola Politécnica da Universidade de São Paulo (EPUSP) é conhecida por sua alta qualidade em ensino de engenharia, sendo a mais bem posicionada no QS World University Rankings by Subject 2022 em Engenharia e Tecnologia na América Latina (TOP UNIVERSITIES, 2022). No entanto, os alunos, em especial os novos, enfrentam dificuldades para encontrar as salas e entender os horários de suas aulas, o que pode dificultar sua jornada acadêmica. Para ajudar a solucionar esse problema, foi criado o projeto USPolis. Este projeto é a continuação de um aplicativo anterior, desenvolvido por um ex-aluno da Escola Politécnica, que já ajudava os estudantes a localizar suas aulas, mas que precisava de atualizações e melhorias.

A motivação para este projeto surgiu da necessidade de aprimorar e modernizar o aplicativo existente, especialmente sua usabilidade, atualização de informações e integração com novas tecnologias. O USPolis<sup>1</sup> visa preencher essas lacunas, propondo uma solução digital abrangente que não apenas mantenha as funcionalidades úteis do aplicativo anterior mas também introduza inovações e melhorias. Este projeto, portanto, representa um avanço no uso da tecnologia para melhorar a experiência dos alunos na faculdade, tornando mais fácil para eles se organizarem e navegarem pela universidade.

## 1.1 Objetivo

O objetivo deste projeto consiste em revitalizar e expandir o aplicativo USPolis, uma ferramenta destinada a auxiliar os estudantes da EPUSP na navegação e obtenção de informações sobre as aulas na instituição. Sua principal funcionalidade é exibir para os usuários como as disciplinas e turmas estão alocadas nas salas de aula distribuídas pela faculdade. Esta informação não é de fácil acesso pelos alunos, o que motivou o então aluno Renan de Luca Ávila, hoje coorientador deste trabalho, a construir o *app* durante sua graduação. No entanto, o aplicativo encontrava-se sem manutenção há alguns anos, desde o fim de sua graduação em 2019.

A implementação de novas funcionalidades no aplicativo fundamentou-se em três pilares. O primeiro deles é o Design Sprint, uma metodologia desenvolvida pela Google Ventures que visa prototipar e testar produtos ou funcionalidades em um curto espaço de tempo, otimizando recursos e obtendo *feedbacks* reais antes da implementação efetiva.

O segundo pilar engloba a aplicação de testes do tipo A/B no interior da aplicação,

---

<sup>1</sup> O aplicativo pode ser baixado para dispositivos Android na loja Google Play (<https://play.google.com/store/apps/details?id=uspolis.lunadros>), e para dispositivos iOS na loja App Store (<https://apps.apple.com/br/app/uspolis/id1451455075>)

através da criação de experimentos no produto digital. Essa abordagem permite obter dados para a definição de estratégias e planos de ação no decorrer do projeto.

O terceiro pilar centra-se no estudo e implementação de conceitos de DevOps, com ênfase na integração, testagem, e distribuição contínua e automatizada do produto. Essa abordagem visa facilitar entregas e garantir a qualidade do aplicativo de maneira eficiente e consistente.

## 1.2 Justificativa

A necessidade de revitalizar o projeto do aplicativo USPolis surge de alguns pontos principais. Primeiramente, como já mencionado, o aplicativo USPolis original já não recebia manutenção há anos; ainda assim, seria possível apenas voltar a mantê-lo e ele funcionaria corretamente, dispendo as alocações de salas de aula.

No entanto, a separação das bases de código para as plataformas iOS e Android, empregando tecnologias que se tornaram obsoletas, reforça a necessidade de modernização para garantir a manutenibilidade e a eficácia a longo prazo do aplicativo. Paralelamente, a renovação e expansão do sistema administrativo da EPUSP, que permite a alocação automática de disciplinas nas salas dos prédios, conforme descrito por [Higuti e Kondo \(2022\)](#), demandava uma integração aprimorada com o aplicativo USPolis, justificando sua reconstrução.

Assim, o desenvolvimento de uma nova versão do USPolis, com um código unificado para diferentes plataformas, tornou-se importante para uma melhor integração do sistema. Essa atualização abordou as limitações técnicas do aplicativo anterior, além de permitir e facilitar o desenvolvimento de novas funcionalidades.

As melhorias incluídas na nova versão do aplicativo não só resolvem os problemas do aplicativo original, mas também oferecem vantagens em relação a outros aplicativos semelhantes, atendendo melhor às necessidades dos alunos. Muitos aplicativos semelhantes, como o MatrUSP e o MRBS (abordados no [Capítulo 3](#)) enfrentam desafios em fornecer informações atualizadas e precisas sobre as aulas, bem como em oferecer uma interface de usuário amigável e funcionalidades intuitivas. O novo USPolis aborda essas lacunas ao oferecer uma experiência de usuário aprimorada, com um design mais intuitivo, e funcionalidades mais avançadas.

## 1.3 Organização do Trabalho

Este trabalho está organizado da seguinte maneira: no [Capítulo 2](#), são apresentados aspectos conceituais fundamentais para a compreensão do estudo, introduzindo os tópicos DevOps, Teste A/B, e Design Sprint, que permeiam este projeto.

O Capítulo 3 aborda os trabalhos relacionados a este, incluindo potenciais concorrentes.

O Capítulo 4, por sua vez, discute o método utilizado no decorrer do projeto.

O Capítulo 5 abrange a especificação de requisitos, detalhando as funcionalidades implementadas.

O Capítulo 6 apresenta as tecnologias e arquitetura de sistema envolvidas no projeto.

O Capítulo 7 detalha o desenvolvimento do projeto, abordando todas as funcionalidades desenvolvidas e o processo de ideação que as gerou.

O Capítulo 8 apresenta os resultados analíticos obtidos com a implementação do aplicativo desenvolvido.

Por fim, o Capítulo 9 conclui o trabalho, apresentando considerações finais e perspectivas para a continuidade do USPolis.

## 2 Aspectos Conceituais

Este capítulo explora conceitos do desenvolvimento de produtos: Design Sprint, Testes A/B e DevOps, aspectos importantes no desenvolvimento de aplicações digitais contemporâneas, como o USPolis. O objetivo é descrever como cada uma dessas metodologias contribui para o refinamento contínuo de processos e produtos. Esse entendimento permite a execução de projetos de tecnologia que atendam às demandas e superem as expectativas dos usuários finais, destacando sua importância dentro do projeto USPolis.

### 2.1 Design Sprint

O Design Sprint é uma metodologia ágil de desenvolvimento de produtos que tem como objetivo validar ideias e propostas de forma rápida e eficiente, minimizando a relação entre custo e tempo. Criado dentro do ambiente da Google Ventures para ajudar *startups* a desenvolverem produtos com maior assertividade, a metodologia se expandiu e se tornou popular em várias indústrias (KNAPP; KOWITZ, 2016).

Inspirado no Design Thinking, o Design Sprint reúne uma equipe multidisciplinar com papéis específicos, guiados por uma agenda de cinco dias repleta de atividades (KNAPP; KOWITZ, 2016). A cada dia, a equipe se aproxima do objetivo de entender o problema a ser resolvido e desenvolver um protótipo de solução.

Segundo Knapp e Kowitz (2016), com o Design Sprint, as equipes são capazes de criar soluções inovadoras, testar hipóteses e validar ideias em um curto período. O método tem se mostrado eficaz na melhoria da experiência do usuário e na criação de produtos de sucesso.

No processo do Design Sprint, a multidisciplinaridade é importante para trazer visões diferentes sobre o mesmo problema (KNAPP; KOWITZ, 2016). Com isso, a equipe possui alguns membros que possuem papéis importantes em relação à dinâmica das atividades. Entre eles, incluem-se:

- **Facilitador:** o responsável por liderar o Design Sprint, definir a agenda e garantir que todos os participantes estejam alinhados com os objetivos da equipe.
- **Especialistas do produto:** membros da equipe que possuem conhecimento sobre o produto ou serviço que está sendo desenvolvido.
- **Designer:** um profissional que pode ajudar a equipe a criar soluções criativas e visualmente atraentes.

- **Desenvolvedor:** um membro da equipe com habilidades técnicas que pode ajudar a viabilizar as soluções propostas durante o Design Sprint.
- **Especialista em marketing:** um membro da equipe que pode fornecer *insights* sobre o mercado e o público-alvo.
- **Stakeholders:** pessoas importantes para o projeto que precisam estar presentes durante o Design Sprint para garantir que as soluções propostas atendam às necessidades da empresa e dos clientes.
- **Cliente ou usuário final:** é importante ter um representante do público-alvo presente para fornecer *feedback* sobre as soluções propostas e garantir que elas atendam às suas necessidades.

A seguir será apresentado o processo organizado por dia, seguindo o descrito por Knapp e Kowitz (2016).

Figura 1 – Agenda do Design Sprint.



Fonte: traduzido de Knapp e Kowitz (2016).

### 2.1.1 Dia 1

O primeiro dia de um Design Sprint é importante para definir a direção do projeto e garantir que todos os membros da equipe estejam alinhados com os objetivos a serem alcançados. Nesse dia, é importante escolher uma questão específica que precisa ser resolvida e garantir que todos estejam cientes do problema a ser solucionado.

Durante o primeiro dia, é fundamental que os membros da equipe participem de atividades para compreender o contexto e o objetivo do projeto. Além disso, é importante realizar uma análise de mercado para entender a concorrência e identificar oportunidades.

Ademais, o primeiro dia também é o momento para definir os critérios de sucesso do projeto. Ao estabelecer esses critérios, a equipe terá um ponto de referência claro para avaliar o sucesso do projeto no futuro.

Uma vez que o problema foi identificado, as metas definidas e os critérios de sucesso estabelecidos, é importante que a equipe crie um plano para os próximos dias do Design



Sprint. Este plano deve incluir os objetivos específicos de cada dia, as atividades que serão realizadas e a agenda geral.

Em resumo, as sequências das atividades propostas para esse dia é:

- **Como poderíamos? (CP):** consiste em capturar possíveis oportunidades, completando a frase "Como podemos..." e escrevendo em *post-its*, a fim de abordar todas as lacunas durante as entrevistas feitas com os Especialistas a nível de produto e de mercado.
- **Agrupamento e Votação:** após a escrita dos CPs, o grupo tenta agrupar os *post-its* por proximidade de contextos e cada membro deve realizar 6 votos em quais cada um acha que são os mais relevantes.
- **Construção da jornada:** o grupo mapeia a jornada do cliente ou do usuário dentro do produto. Após a definição, as oportunidades mais votadas na atividade anterior são posicionadas dentro da jornada, trazendo uma visão mais ampla de qual a etapa mais problemática e/ou possui um grande potencial.
- **Definição da meta:** por fim, com todos os membros estando cientes da situação atual em que a equipe se encontra, o próximo passo consiste em definir qual o objetivo do sprint e quais são as perguntas que precisam ser respondidas ao final.

### 2.1.2 Dia 2

No segundo dia do Design Sprint a equipe começa a gerar ideias para solucionar o problema definido no dia anterior. Cada membro deve apresentar suas ideias, e a equipe deve discuti-las para avaliar sua viabilidade e eficácia.

Nesta fase é importante que cada membro da equipe apresente suas ideias de forma clara e justifique sua proposta. Isso permitirá que a equipe entenda as razões por trás de cada ideia e avalie sua relevância para o problema a ser resolvido.

Uma técnica comum utilizada no segundo dia do Design Sprint é a *brainstorming*, na qual a equipe gera uma grande quantidade de ideias em um curto período, sem julgamentos ou críticas. Em seguida, a equipe seleciona as melhores ideias e as aprimora.

Ao final do segundo dia, a equipe deve ter um conjunto de ideias para solucionar o problema escolhido. Essas ideias devem ser avaliadas para identificar as mais promissoras e que melhor atendam aos critérios de sucesso estabelecidos anteriormente.

A agenda para esse dia consiste em:

- **Benchmark:** cada membro do grupo tenta encontrar soluções que o mercado utiliza e que possuem potencial de inspiração ou de embasamento. Todas essas ideias são compartilhadas entre os membros.
- **Crazy 8:** de forma desprezenciosa, os membros constroem variações de soluções na forma de 8 esboços para atender as oportunidades do dia anterior, inspirados nas ideias dos *benchmarks*, e apresentam para o resto da equipe juntamente com a linha de raciocínio de cada esboço apontando prós e contras de cada um.
- **Construção de esboço das soluções:** após o compartilhamento de todo esse material voltado em construir uma solução para o problema escolhido, cada membro deve construir uma solução consolidada individualmente, a qual será compartilhada somente no dia seguinte.

### 2.1.3 Dia 3

Durante o terceiro dia, a equipe trabalha em conjunto para revisar e avaliar as ideias geradas no segundo dia, levando em consideração a sua viabilidade, originalidade e potencial para resolver o problema em questão. A equipe pode usar critérios de seleção, como uma matriz de decisão ou votação, para selecionar as melhores ideias para avançar para a próxima fase do processo.

Após a seleção das melhores ideias, a equipe pode criar um *storyboard* para visualizar como as ideias podem ser implementadas e como elas podem levar a solução do problema em questão. Além disso, a equipe pode fazer protótipos rápidos das ideias selecionadas para testar a funcionalidade e viabilidade de cada uma.

A agenda para esse dia consiste em:

- **Apresentação das soluções:** cada integrante deve apresentar ao resto da equipe a sua proposta de solução, apontando quais foram os pontos de oportunidades que a solução irá atacar.
- **Votação:** posteriormente às apresentações, o grupo deve votar quais foram os "conceitos" mais promissores, consistindo em qualquer elemento presente dentro de cada solução ou jornada apresentada.
- **Nova jornada:** construir como será a jornada do usuário dentro dessa possível solução, sendo importante todos entrarem num consenso de como será essa experiência.
- **Estruturação da jornada:** consiste em consolidar numa única versão os elementos mais interessantes das soluções, juntamente com a jornada definida. Esse material guia a equipe para a construção do protótipo no quarto dia e auxilia na construção das entrevistas para o quinto dia.

### 2.1.4 Dia 4

O quarto dia do Design Sprint é dedicado à criação de um protótipo da solução escolhida no dia anterior. A equipe trabalha em conjunto para criar um modelo físico ou digital que simula a experiência final do usuário com a solução proposta.

Durante este dia é importante que a equipe trabalhe com eficiência para garantir que o protótipo seja concluído até o final do dia. A equipe deve se concentrar nos principais recursos e funcionalidades da solução, para que o protótipo represente fielmente a experiência final do usuário.

### 2.1.5 Dia 5

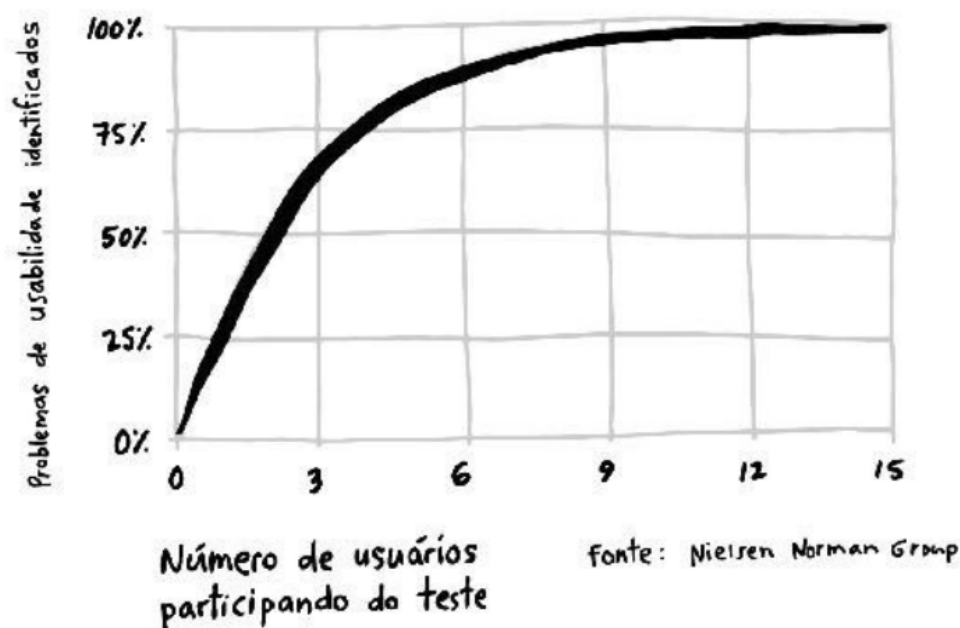
No quinto e último dia do Design Sprint, a equipe se dedica a testar o protótipo criado no dia anterior. A equipe seleciona um grupo de usuários reais para testar o protótipo, e observar sua interação com a solução proposta.

A quantidade de usuários a serem entrevistados é uma grande dúvida, por ser indesejado ter um falso positivo sobre a percepção da solução, invalidando toda a proposta do método de trazer uma resposta convincente em pouco tempo e recurso. Porém, o excesso de candidatos também é ruim por gastar o tempo e energia da equipe em realizar as entrevistas, compila-las e avaliar o resultado final. Essa influência entre o número de entrevistados e a eficiência do processo de entrevistas é possível compreender melhor através da Figura 2. Dessa forma, segundo Knapp e Kowitz (2016), a quantidade ideal de entrevistados é de 5 pessoas, sendo uma quantidade em que seja possível tirar a maior parte das conclusões da forma mais eficiente possível em relação aos recursos necessários.

Durante os testes, é importante que a equipe esteja atenta às reações dos usuários e às dificuldades que eles encontram. A equipe deve registrar todas as observações e *feedbacks* relevantes, para que possa avaliar a eficácia da solução e identificar pontos de melhoria.

Após os testes, a equipe se reúne para discutir os resultados e avaliar o sucesso da solução proposta. Caso a solução tenha sido eficaz, a equipe pode seguir em frente para a implementação. Caso contrário, a equipe pode realizar ajustes e repetir o processo do Design Sprint, até que uma solução satisfatória seja encontrada.

Figura 2 – Gráfico de eficiência em relação ao número de entrevistados.



Fonte: traduzido de [Knapp e Kowitz \(2016\)](#).

## 2.2 Teste A/B

O Design Sprint, por ser um processo inerentemente criativo e exploratório, pode gerar diferentes soluções para um mesmo problema ou funcionalidade. Uma maneira de avaliar empiricamente e objetivamente as alternativas produzidas é através da realização de testes do tipo A/B, que confrontam variações de uma mesma solução para determinar aquela que obteve melhor desempenho.

A seguir, será discutida a necessidade da experimentação como complemento natural da inovação, culminando nos testes A/B *online* e na descrição de como executá-los.

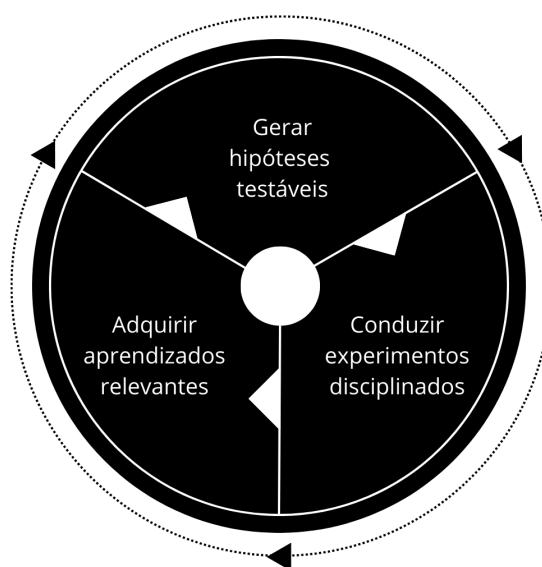
### 2.2.1 Experimentação e testes

Dentro de qualquer organização ou empresa criando e gerenciando um produto digital, a experimentação com produtos vem se tornando cada vez mais frequente. Seja para adicionar uma nova funcionalidade, validar alguma hipótese ou testar variações de algum cenário de uso já existente, a etapa de testagem tornou-se essencial para a obtenção dos resultados desejados de maneira mais eficaz e eficiente - isto é, gastando menos recursos e sendo mais certo ([THOMKE, 2021](#)). Não apenas os testes com o produto devem ser executados, mas também integrados como parte central da estratégia para o crescimento, sendo incorporados desde o planejamento da organização.

A descoberta de novas ideias e *insights* sobre o negócio ocorre dessa maneira, como

ilustrado na Figura 3: com base em algum tipo de evidência, geram-se hipóteses para conduzir experimentos e testá-las. A partir destes experimentos, são aprendidas novas lições sobre o negócio - sejam formas de aumentar o tempo de uso do produto pelo cliente, taxas de conversão para acesso de alguma página específica ou mesmo o faturamento gerado por ele. Segundo Thomke (2021), esse processo deve ser realizado de maneira iterativa para que a agilidade na experimentação permita que cada vez mais dados sejam gerados, os quais serão úteis para a concretização e crescimento do produto.

Figura 3 – Ciclo de experimentação de um produto



Fonte: traduzido de Thomke (2021).

É essencial repetir o procedimento continuamente, pois as hipóteses testadas podem ser rejeitadas, não produzindo os resultados desejados. Por exemplo, ao se testar a hipótese de que aumentar o tamanho do letreiro de um produto em desconto em uma de suas lojas aumentará suas vendas, pode-se descobrir na realidade que variar tamanho do letreiro não afetou o número de vendas. Embora as hipóteses rejeitadas possam fornecer *insights* valiosos para a aprendizagem, se as métricas estabelecidas não apresentarem melhorias, é necessário realizar novos testes até encontrar fatores decisivos para obter os resultados almejados - neste exemplo, poderiam ser testados também o posicionamento do produto ou as cores do letreiro.

### 2.2.2 Fatores decisivos para um bom experimento

Para que um experimento possa ser bem sucedido, isto é, possa gerar dados efetivamente úteis para o negócio, Thomke (2021) enumera uma série de fatores que devem ser considerados no seu planejamento:

- **Hipóteses fortes e testáveis.** A hipótese criada para o experimento deve ser embasada em observações, ideias ou dados obtidos da utilização do produto, e deve ter viabilidade de implementação, ou seja, deve ser testável de forma simples e direta. Além disso, o experimento deve ter variáveis mensuráveis, de forma que seja possível obter um resultado claro e objetivo.
- **Adesão aos resultados obtidos.** É importante que, uma vez obtidos os resultados de um teste realizado, estes não sejam ignorados. Evidentemente, podem surgir *insights* não previstos que geram novas ideias, mas sugere-se fortemente que os planos de ação já estejam criados desde o início para cada resultado possível de ser obtido.
- **Viabilidade de execução.** Para que o experimento seja bem sucedido, o teste criado deve ser efetivamente executável. Dentro do contexto de produtos digitais existem uma série de ferramentas que podem ser utilizadas para a condução de testes de *design* e experiência do usuário, temas centrais neste tipo de negócio.
- **Confiabilidade.** Para garantir a confiabilidade dos resultados obtidos, são empregadas simultaneamente duas estratégias: (i) amostragem aleatória - os indivíduos a serem colocados em cada grupo para teste são escolhidos aleatoriamente, de forma a evitar vieses de seleção - e (ii) teste cego - os indivíduos testados não sabem que estão participando de um experimento, evitando assim quaisquer vieses subjetivos ou inconscientes dos participantes.
- **Causalidade.** Entender as relações entre as variáveis envolvidas no experimento é fundamental para que não haja confusão entre causalidade e correlação. Causalidade refere-se à relação de causa e efeito, ou seja, quando uma variável tem influência sobre outra. Por outro lado, correlação indica uma associação entre duas variáveis, mas não implica necessariamente uma relação de causa e efeito. Assim, para evitar interpretações equivocadas, é recomendável executar apenas um teste por vez, permitindo a análise separada dos efeitos de cada variável.
- **Valor obtido.** Cada teste realizado pode produzir um volume grande de dados relativos à forma com a qual o usuário se relaciona com o produto em questão. Para obter o máximo valor possível de cada experimento, é importante realizar análises completas com estes dados, possibilitando a geração de mais conhecimento sobre o negócio.
- **Decisões direcionadas.** De acordo com [Thomke \(2021, p.80\)](#), "nem todas as decisões de gestão podem ou devem ser tomadas com base em experimentos", mas ainda assim é importante testar tudo o que pode ser testado. Assim, evita-se que grandes investimentos, sejam de tempo, esforço ou dinheiro, sejam realizados sem que haja uma prévia do resultados que podem ser obtidos. Além disso, o processo

decisório ganha em transparência, uma vez que as decisões podem ser apoiadas pelos dados obtidos de experimentos.

### 2.2.3 Como executar um experimento *online*

Nesta seção serão conceituados e discutidos apenas os experimentos *online* do tipo teste A/B. Testes deste tipo baseiam-se na criação de diferentes variações de um produto com relação a alguma funcionalidade ou elemento de *design* que serão expostas a grupos distintos de usuários, para determinar qual das variações obteve melhor desempenho com base em alguma métrica estabelecida. Dessa forma, decisões acerca do produto podem ser tomadas com base não em suposições, mas em dados reais obtidos pelo seu uso.

O primeiro passo para conseguir realizar um teste deste tipo é a geração de uma hipótese. Ela está dividida em duas partes específicas: a **variável independente**, ou seja, aquilo que estamos de fato alterando para tentar alavancar os resultados do produto, e **variáveis dependentes**, isto é, os objetivos metrificadas que querem ser atingidos. Por exemplo, se for tomada como hipótese a frase “trocar a cor do botão de ação de verde para azul deve aumentar o seu *click-through rate*”, a variável independente é a cor do botão, e a dependente é a taxa de clique, métrica de sucesso que está sendo presumida dependente da cor apresentada - evidentemente, os resultados finais do teste devem concluir a real relação entre as variáveis escolhidas.

É importante notar que mesmo alterações pequenas e que geram pouco esforço - como a cor de um botão ou a frase de *call to action* nele apresentada - podem ter resultados muito significativos, positiva ou negativamente, na métrica observada. Assim, sempre que for possível realizar um teste com uma nova ideia, ele deve ser realizado antes da real implementação em larga escala da ideia no produto.

Testes do tipo A/B são realizados com um grupo **controle** e um grupo **desafiante**. Ao grupo controle de usuários é apresentado à variação A, isto é, o produto como ele já existe e é usado atualmente. O controle é essencial para servir como base de comparação em relação às mudanças efetuadas. Ao desafiante, por sua vez, é apresentado o produto com a variação B da variável independente escolhida.

Na realização do experimento, [Thomke \(2021\)](#) explicita que é essencial também se atentar a alguns outros fatores primordiais para que seus resultados sejam confiáveis:

- **Divisão dos grupos.** Os grupos de usuários para testagem devem ser escolhidos de forma aleatória, sem qualquer viés de seleção, e com o mesmo número de participantes. Assim, a ferramenta a ser utilizada para a realização do teste deve ser capaz de fazer tal divisão.
- **Testes independentes entre si.** Para que a relação correta seja estabelecida entre

variável independente e variável dependente, apenas um teste deve ser realizado por vez no mesmo domínio do produto. Dessa forma, garante-se que de fato foi apenas aquela variável que teve influência na métrica observada.

- **Simultaneidade.** O fator temporal também pode ser relevante para o produto especificado, ou seja, pode haver sazonalidade na métrica observada estabelecida. Por exemplo, um carrinho de sorvetes na praia efetua um número maior de vendas em dias não chuvosos de verão. Assim, ambas as variações devem ser apresentadas aos seus respectivos grupos simultaneamente, para evitar viés temporal na métrica.
- **Volume de dados.** Para que o teste tenha alguma validade estatística, é importante que a amostragem seja suficientemente grande. Para tal, o experimento deve ser realizado durante tempo suficiente para que os dados necessários sejam colhidos.
- **Feedback qualitativo.** Os usuários podem ter algo a dizer que justifique ou esclareça os resultados obtidos para cada variação. Por isso, nunca é uma má ideia obter, quando possível, *feedback* qualitativo de cada grupo de teste.

Uma vez realizado o teste e obtidos os resultados, é importante que eles sejam efetivamente utilizados para a construção gradual do crescimento do produto - afinal, foram coletados uma série de dados acerca de seu padrão de uso. Se os grupos não tiverem dados estatisticamente diferentes, aprende-se que esse fator é irrelevante para o uso do produto. Se tiverem, passa-se a usar o grupo vencedor como padrão no produto, pois ele ajuda melhor a alcançar o objetivo.

## 2.3 DevOps

Após explorar a profundidade do Teste A/B, entendendo como essa abordagem pode melhorar o desempenho de produtos digitais por meio da experimentação e análise de dados, surge o questionamento sobre como pode-se integrar esse conceito em um ciclo de desenvolvimento e operações contínuas e eficientes. Dessa motivação surge o conceito de DevOps uma tentativa de estreitar a colaboração entre as equipes de desenvolvimento (Dev) e operações (Ops). Esta seção introduzirá seus princípios fundamentais, explorando como essa abordagem pode melhorar o ciclo de vida do desenvolvimento de software.

### 2.3.1 Introdução e Princípios

DevOps é um conjunto de práticas, princípios e ferramentas que procuram solucionar os principais problemas que envolvem a passagem do produto do desenvolvimento de software para a operação de tecnologia da informação, como descreve [Valente \(2022\)](#). Esta abordagem tem o intuito de integrar as metodologias ágeis durante o processo de transição



de uma aplicação para a produção, enfocando na otimização da comunicação e colaboração entre duas equipes essenciais: a equipe de desenvolvimento e a equipe de operações. No contexto do projeto USPolis, o DevOps é relevante devido à necessidade de atualizações frequentes da aplicação móvel, visando melhorias, expansões e correções.

De acordo com [Kim et al. \(2016\)](#), o conceito de DevOps requer a construção e entrega do software de ponta a ponta, visando integrar e harmonizar as culturas de desenvolvimento e operação, possibilitando implementações mais ágeis, rápidas e eficientes.

A fim de atingir tal tração na implantação, delinea-se os seguintes princípios para alcançar a entrega de software com confiabilidade, eficiência e frequência nas entregas ([HUMBLE; FARLEY, 2010](#)):

1. **Construir com qualidade:** o processo de entrega de software deve ser confiável, para que possa ser repetível e escalável.
2. **Automatize tarefas repetitivas:** tarefas como construção, teste e implantação devem ser automatizadas para reduzir o risco de erros humanos.
3. **Manter tudo em controle de versão:** não apenas o código-fonte, mas também código de infraestrutura e esquemas de banco de dados, arquivos de configuração devem ser colocados no controle de versão.
4. **Concluído significa lançado:** uma alteração só está concluída quando está funcionando com sucesso no ambiente de produção.
5. **Construir uma vez, implantar em qualquer lugar:** compile e empacote sua aplicação uma vez e use esse mesmo pacote em todos os ambientes.
6. **Priorize manter o software implantável:** em vez de focar em novos recursos, garanta que o software possa ser lançado a qualquer momento

### 2.3.2 Controle de Versão

Após compreender a importância do DevOps no ciclo de vida do desenvolvimento de software, aborda-se um dos seus principais componentes: o controle de versão. Este elemento é uma das bases do DevOps, mas também se torna um pilar fundamental para a eficiência no processo de desenvolvimento. Enquanto o DevOps facilita a integração e entrega contínua, o controle de versão oferece a base necessária para essa integração, ao gerenciar e rastrear as mudanças no código-fonte.

Assim, dentro do contexto de desenvolvimento de software, o controle de versão surge como um sistema e conjunto de práticas para gerenciar mudanças em documentos, programas, e outras informações armazenadas, sendo a peça chave para garantir a integridade do código e colaboração dos desenvolvedores.

Um sistema de controle de versão é projetado com dois objetivos principais: proporcionar um armazenamento centralizado para o código fonte atual, um **repositório**, e, paralelamente, registrar o histórico completo de modificações do código. Esses registros não só contêm as alterações efetuadas no código, mas também carregam informações contextuais como a data da alteração e a identidade do autor responsável.

A capacidade de rastrear e reverter para qualquer ponto específico do histórico de desenvolvimento é uma característica intrínseca de tais sistemas e desempenha um papel importante em ambientes de desenvolvimento. Em situações em que o software apresenta falhas ou comportamento inesperado, essa funcionalidade assegura que os desenvolvedores possam retornar a uma versão anteriormente estável, minimizando assim possíveis interrupções e prejuízos.

No âmbito dos sistemas de controle de versão, o Git se destaca. Como destacado por [Chacon e Straub \(2014\)](#), o Git, um sistema de controle de versão distribuído, é amplamente reconhecido e adotado na indústria de tecnologia atual. O Git não apenas facilita a gestão colaborativa do código, mas também pode ser operado em diversas plataformas sem incorrer em custos significativos.

O **Github**, por sua vez, é uma plataforma *online* que se apoia na tecnologia Git para fornecer serviços de hospedagem para repositórios de código. Ele possibilita que desenvolvedores e organizações centralizem seus códigos em um ambiente digital acessível, com opções de repositórios públicos ou privados. Conforme discutido por [Dabbish et al. \(2012\)](#), este serviço simplifica a infraestrutura de controle de versão para empresas, permitindo-lhes focar mais no desenvolvimento e menos na manutenção de sistemas.

### 2.3.3 Integração Contínua

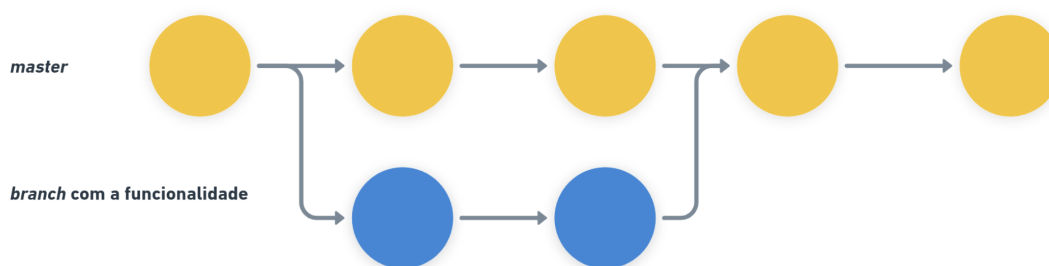
Após a exploração do controle de versão e sua importância no gerenciamento e rastreabilidade do código-fonte, é natural abordar outro componente importante no desenvolvimento de software e DevOps: a Integração Contínua (CI). A transição do controle de versão para a CI é uma progressão natural, pois, enquanto o controle de versão oferece a fundação para uma colaboração eficaz os desenvolvedores, a CI leva este processo um passo adiante, permitindo a integração frequente e automática do código-fonte ([HUMBLE; FARLEY, 2010](#)). Este fluxo de trabalho contínuo e integrado, facilitado pela CI, não apenas promove um desenvolvimento mais eficiente, mas também possibilita práticas avançadas como *Deployment* Contínuo, unindo-se assim aos princípios do DevOps para um ciclo de desenvolvimento mais ágil.

### 2.3.3.1 Conflitos na Integração

Quando se trata de Integração Contínua, é importante considerar a gestão dos conflitos que podem surgir durante a integração de códigos. Enquanto a CI possibilita a atualização frequente do código principal com constantes mudanças, ela também aumenta a probabilidade de ocorrerem conflitos de integração - um fenômeno em que diferentes modificações no código entram em desacordo umas com as outras.

A Figura 4 ilustra o sistema de *branches* no GitHub, que é fundamental para entender a origem dos conflitos de integração. Em um projeto Git, a ramificação principal recebe o nome de *master* ou *main*, podendo existir inúmeras ramificações secundárias. Quando uma *branch* secundária, que foi desenvolvida separadamente, é reintegrada à *branch* principal, surge o potencial para conflitos. Esses conflitos ocorrem quando alterações distintas foram feitas no mesmo segmento de código em diferentes *branches*.

Figura 4 – Exemplo de ramificação no Git.



Fonte: de autoria própria.

Para mitigar tais conflitos de integração, algumas práticas são recomendadas, como a integração frequente de pequenas mudanças, a comunicação efetiva entre os membros da equipe e o uso de ferramentas automatizadas para detecção e resolução de conflitos. Além disso, a revisão de código e o teste contínuo são essenciais para identificar e resolver rapidamente problemas de integração (FOWLER; FOEMMEL, 2006).

### 2.3.3.2 Características da Integração Contínua

Ao adotar a Integração Contínua e seus principais componentes delineados anteriormente, é comum que a ramificação principal da aplicação seja constantemente atualizada com novos códigos. Para assegurar essa fluidez, é importante adotar um conjunto de diretrizes a fim de prevenir obstáculos como falhas de compilação e *bugs* inesperados (DUVALL; MATYAS; GLOVER, 2007):

1. **Automação de *build*:** o termo *build* é usado como a compilação de todos os arquivos do sistema para a geração de uma versão executável. É importante para toda a esteira de automação que todas as alterações da ramificação principal do repositório

façam com que automaticamente o código seja compilado. Outra importante boa prática para essa etapa é a eficiência na compilação, sendo indicado por alguns autores como Beck (1999) a ter um limite de tempo de 10 minutos, isso é, curto o suficiente para que a execução da *build* mantenha a integração contínua saudável.

2. **Servidores de Integração Contínua:** esses servidores facilitam a automação de *builds*, possuindo a capacidade de clonar o repositório de código e acionar o *build* após um novo *commit* (FOWLER; FOEMMEL, 2006).

- Por meio de um sistema de controle de versão, este *commit* notifica o servidor de integração contínua, que, por sua vez, inicia o processo automatizado
- Além de gerar a *build*, o servidor de integração contínua é incumbido de executar testes no código-fonte.
- Posteriormente, ao concluir a compilação e os testes, ele informa o desenvolvedor acerca do resultado da *pipeline* de integração, seja ela bem-sucedida ou não.

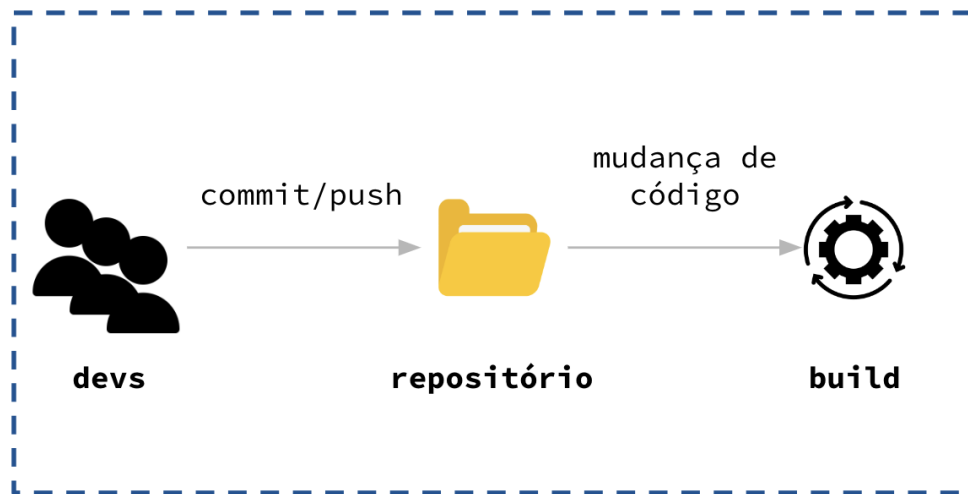
3. **Testes Automatizados:** é importante assegurar que todas as funcionalidades operem conforme o esperado. Nesse sentido, a base de código necessita ter ampla cobertura de testes, proporcionando que a IC valide consistentemente as assertivas desses testes nas novas *builds* (CRISPIN; GREGORY, 2009).

Por conseguinte, os servidores de integração contínua imprimem no sistema e no fluxo de desenvolvimento as seguintes características (DUVALL; MATYAS; GLOVER, 2007):

- **Prevenção de códigos problemáticos:** códigos com problema de *build* ou comportamento não são integrados.
- **Remediação imediata de *builds* falhas:** Caso o servidor de integração contínua indique incongruências nos testes ou na compilação, a prioridade do desenvolvedor deve ser a retificação dessas falhas, prevenindo potenciais transtornos para outros colegas de equipe
- **Progressão condicionada ao resultado do CI:** Os desenvolvedores só devem avançar para outras tarefas após receberem o resultado do servidor de CI, garantindo que o último *commit* foi bem-sucedido.

Na Figura 5, temos a representação da CI. O gráfico ilustra o processo que começa com desenvolvedores realizando *commits* ou *push* de suas mudanças de código para um repositório central. Após a submissão dessas mudanças, é iniciado um processo de *build* automatizado, na qual o código é compilado e testado para garantir que as alterações se

Figura 5 – Diagrama esquematizando Integração Contínua. Alterações no código são integrados ao código de build do projeto



Fonte: de autoria própria.

integrem sem problemas ao projeto existente. A figura também aponta que as alterações de código são integradas de forma contínua, enfatizando a iteratividade constante deste processo.

### 2.3.4 Implantação Contínua

Caso exista IC, novos códigos são frequentemente integrados à *branch* principal do projeto, mesmo que não esteja pronto para produção.

Um passo adiante no sentido dos princípios antes elucidados do DevOps, introduz-se o conceito de implantação contínua, uma extensão da integração contínua, na qual novos *commits* na *master*, além de gerarem novas *builds* do projeto, entram também rapidamente em produção. As regras da integração contínua continuam valendo nos quesitos confiabilidade e segurança: o comportamento e a compilação da base de código continuam sendo aferidos no servidor de integração contínua, e bloqueados caso seu resultado seja uma falha (VALENTE, 2022).

De acordo com Humble e Farley (2010), as vantagens que a implantação contínua traz consigo são:

- O tempo de entregas de novas funcionalidade é reduzido, diminuindo o intervalo entre *releases*.
- Novas distribuições ocorrem frequentemente, eliminando *deadlines* vinculadas à *deployment*, reduzindo o estresse entre desenvolvedores e operadores de sistemas.
- Favorece o crescimento orientado a dados e *feedback* dos usuários.

### 2.3.5 Entrega Contínua

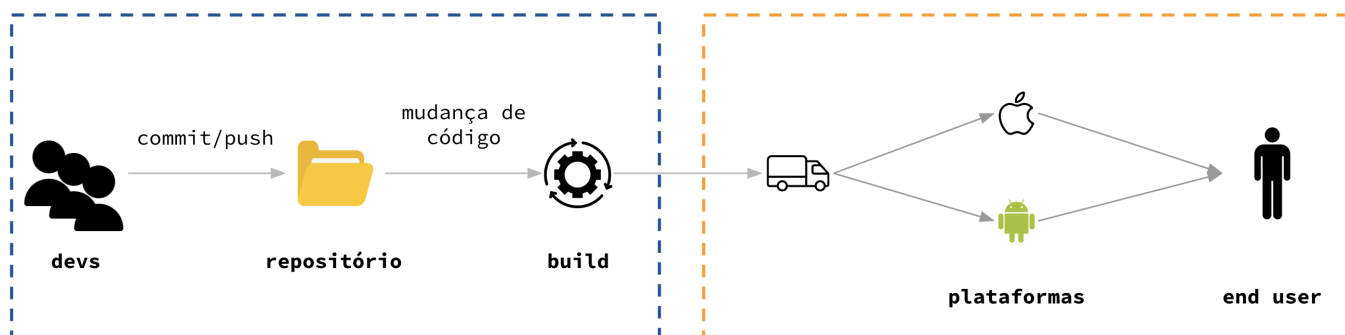
A combinação sistemática das técnicas de integração contínua (CI) e implantação contínua (CD) pode ser conceptualizada como uma esteira de entrega contínua (EC) (FOWLER; FOEMMEL, 2006). Ao adotar este paradigma de entrega contínua, cada *commit* realizado é potencialmente viável para ser introduzido no ambiente de produção. Dependendo da estratégia adotada, esta decisão de implantação pode ser mediada por uma entidade ou autoridade externa, responsável por estabelecer critérios específicos que determinam se um *commit* particular é adequado para ser promovido ao ambiente de produção (HUMBLE; FARLEY, 2010).

Nesse contexto, emergem dois conceitos fundamentais que são cruciais para a compreensão e implementação efetiva da entrega contínua:

- **Delivery:** Define-se como o processo metódico de preparação e liberação de uma versão específica do software para a etapa subsequente, que é a implantação (DUVALL; MATYAS; GLOVER, 2007).
- **Implantação:** Refere-se ao mecanismo estruturado pelo qual uma versão pré-selecionada do software é finalmente disponibilizada para os usuários finais, garantindo sua efetiva operacionalização no ambiente desejado (CRISPIN; GREGORY, 2009).

A Figura 6 ilustra os conceitos chave das práticas de integração contínua (CI) e implantação contínua (CD), que inicia-se no fluxo inicial de desenvolvimento estendendo-se até chegar ao usuário final, auxiliando na promoção de uma cultura orientada à entrega contínua.

Figura 6 – Integração entre CI e CD



Fonte: de autoria própria.

## 2.4 Considerações Finais do Capítulo

Este capítulo ofereceu um panorama dos aspectos conceituais que fundamentam práticas modernas de desenvolvimento de software, evidenciando como ferramentas como Design Sprint, Testes A/B e DevOps se interligam para impulsionar a inovação e a eficiência. Com a exploração dessas técnicas, estabeleceu-se uma compreensão de como podem ser aplicadas para melhorar continuamente a qualidade e a relevância do USPolis.

No próximo capítulo, que explorará trabalhos relacionados, o USPolis poderá ser visto dentro de um contexto mais amplo de inovações tecnológicas no âmbito educacional na EPUSP.

## 3 Trabalhos Relacionados

Este projeto tem como base um aplicativo já existente e que está em operação desde 2015 chamado USPolis. O USPolis foi originalmente idealizado por Renan de Luca Avila, ex-aluno de Engenharia da Computação, em 2015, em um de seus anos iniciais da faculdade, para resolver um problema específico da EPUSP. Assim como muitos outros alunos da faculdade, ele tinha dificuldade para encontrar informações sobre onde seriam as aulas de cada turma para cada uma de suas disciplinas. Como forma de mitigar esta dificuldade, Renan criou um aplicativo para celular (doravante denominado neste documento de "aplicativo legado"), que continha tais informações e as exibia para os usuários.

O aplicativo legado, apesar de ter sido utilizado por alunos da faculdade entre 2015 e 2019, não era mantido após o término da graduação de Renan. Dessa forma, ele estava defasado e não alimentava mais o banco de dados com informações das aulas atuais. Também era necessário melhorar a experiência do usuário do aplicativo, tanto na interface quanto na usabilidade.

Além disso, o grupo viu neste trabalho oportunidades de atualizar as ferramentas e linguagens de programação utilizadas no projeto. O aplicativo legado havia sido desenvolvido utilizando duas bases de código distintas, para plataformas Android e iOS. Em vez de precisar manter dois códigos diferentes, optou-se pela utilização de algum *framework* de desenvolvimento híbrido. Uma pesquisa realizada anualmente entre desenvolvedores de software indica que, desde 2014, a linguagem Javascript é a mais utilizada no mercado, com cerca de 60% de uso entre respondentes. Também nota-se que em 2022 a tecnologia React Native possui cerca de 12% de uso, sendo a segunda mais utilizada para desenvolvimento de aplicativos móveis multiplataforma ([STACK OVERFLOW, 2022](#)). Assim, optou-se pelo uso de React Native para o desenvolvimento do novo USPolis.

Este trabalho atua também como uma continuação de um projeto de formatura realizado por [Higuti e Kondo \(2022\)](#) também no Departamento de Engenharia de Computação e Sistemas Digitais (PCS) da EPUSP. Tal trabalho visou desenvolver um sistema administrativo para cadastro e gerenciamento de disciplinas por parte dos departamentos da EPUSP. As disciplinas ali cadastradas são as exibidas no aplicativo.

A seguir, se apresentará mais detalhes de ambas as aplicações mencionadas, além de outros sistemas utilizados na USP que, por terem funcionalidades similares às do USPolis, podem ser considerados possíveis concorrentes deste projeto.



### 3.1 Aplicativo Legado do USPolis

O projeto original do aplicativo USPolis tem o objetivo de solucionar o problema da busca pelas salas de aula das disciplinas. A principal funcionalidade do aplicativo é a listagem de disciplinas, sendo possível também pesquisá-las por nome ou código.

Também é possível adicionar tais turmas e disciplinas a uma área separada chamada "Minhas Aulas", em que são listadas em ordem cronológica na semana todas as aulas daquele usuário, para que ele possa acompanhar seus horários semanais da faculdade.

Além disso, há uma outra área no aplicativo em que podem ser consultados os mapas de cada prédio da EPUSP, para ajudar os alunos a localizarem as salas em que vão acontecer suas aulas.

Nas Figuras 7, 8 e 9 estão ilustradas respectivamente as funcionalidades mencionadas do aplicativo USPolis legado.

Figura 7 – Aplicativo USPolis Legado - Área de Pesquisa de Aulas

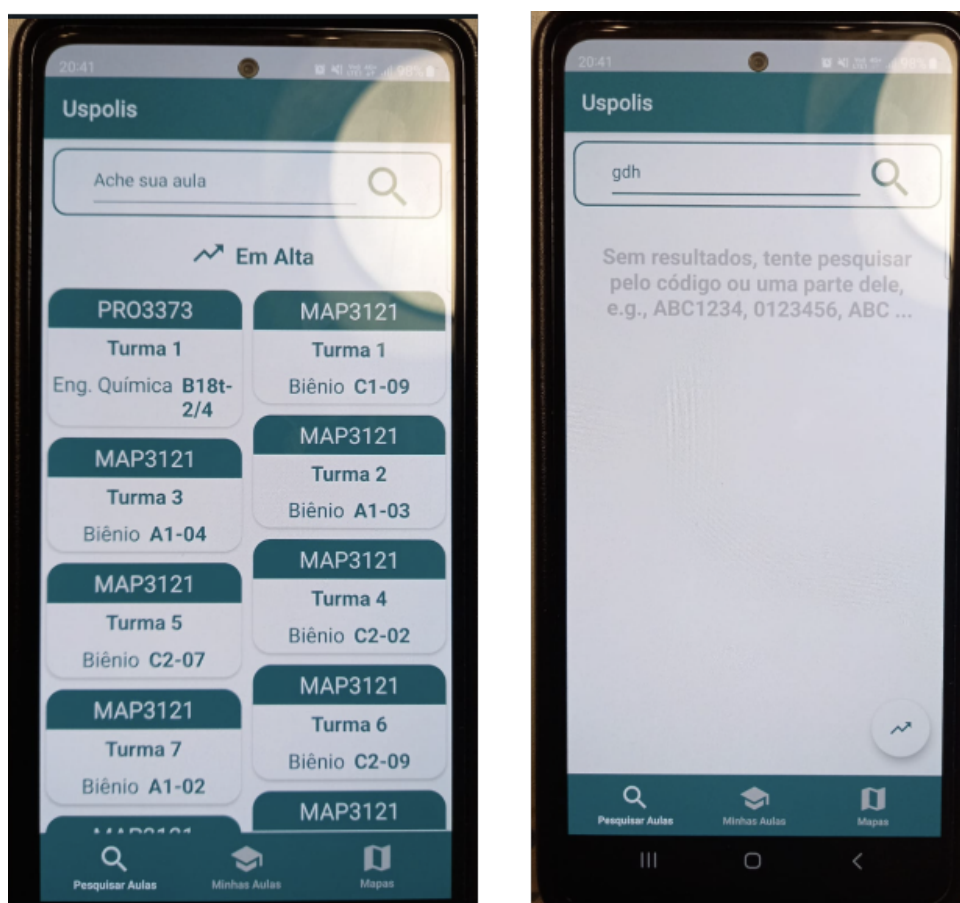
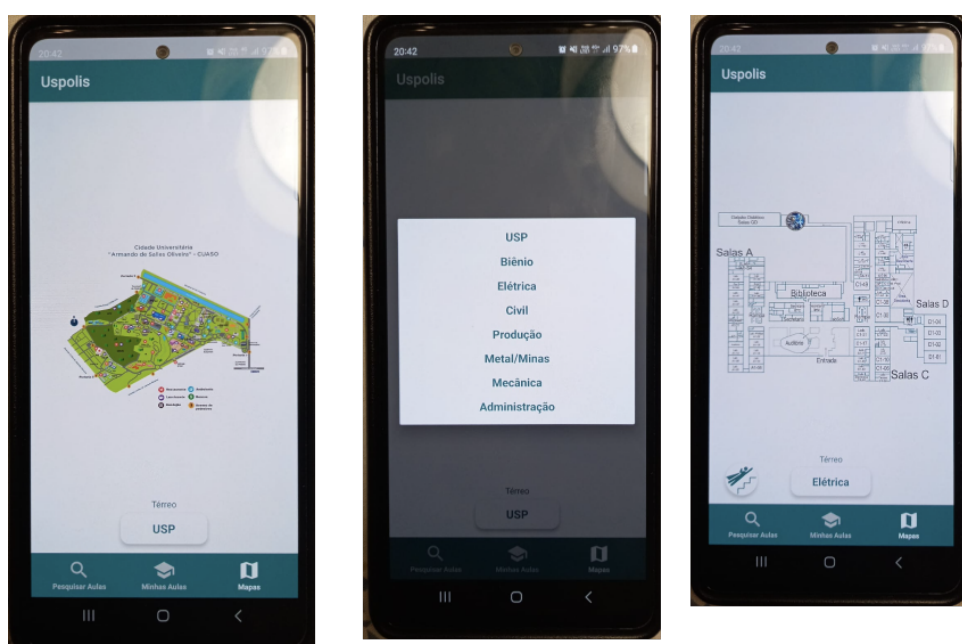


Figura 8 – Aplicativo USPolis Legado - Área de Minhas Aulas



Figura 9 – Aplicativo USPolis Legado - Área de Mapas



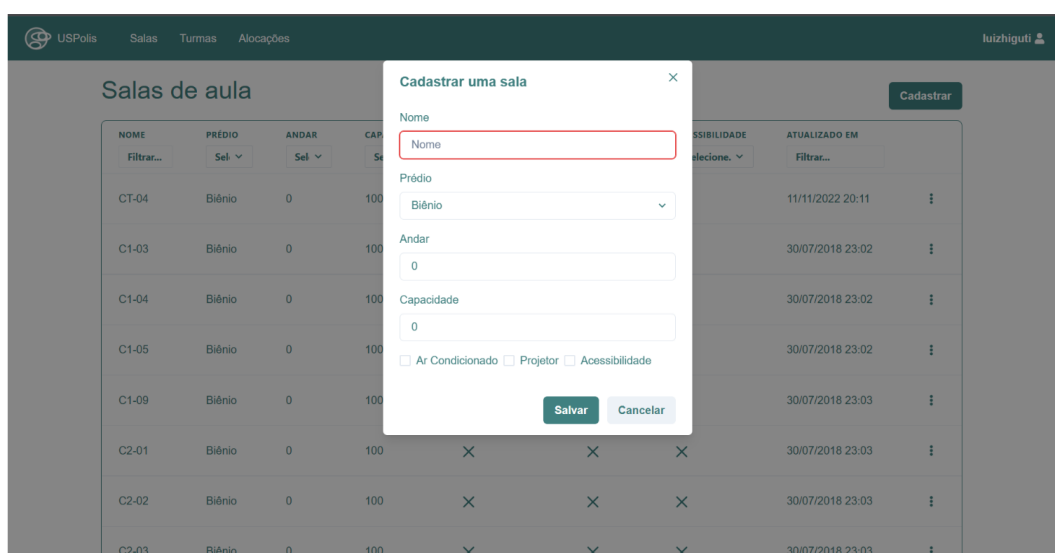
## 3.2 USPolis - Administrativo

O USPolis conta também com uma plataforma administrativa disponível na *web*. Sua responsabilidade é permitir o cadastro e gerenciamento de salas de aulas de disciplinas (HIGUTI; KONDO, 2022). Além disso, o serviço por trás dessa aplicação é o responsável e dono pelos dados dos edifícios, disciplinas e salas de aula, que alimentam o aplicativo desenvolvido neste trabalho.

Suas principais funcionalidades são:

1. **Gestão de salas:** gerenciamento de cada uma das salas presente em cada prédio da faculdade, como ilustrado na Figura 10. As salas podem ser adicionadas, removidas e editadas, e contemplam características como capacidade máxima de alunos e indicações se há projetor e ar-condicionado.

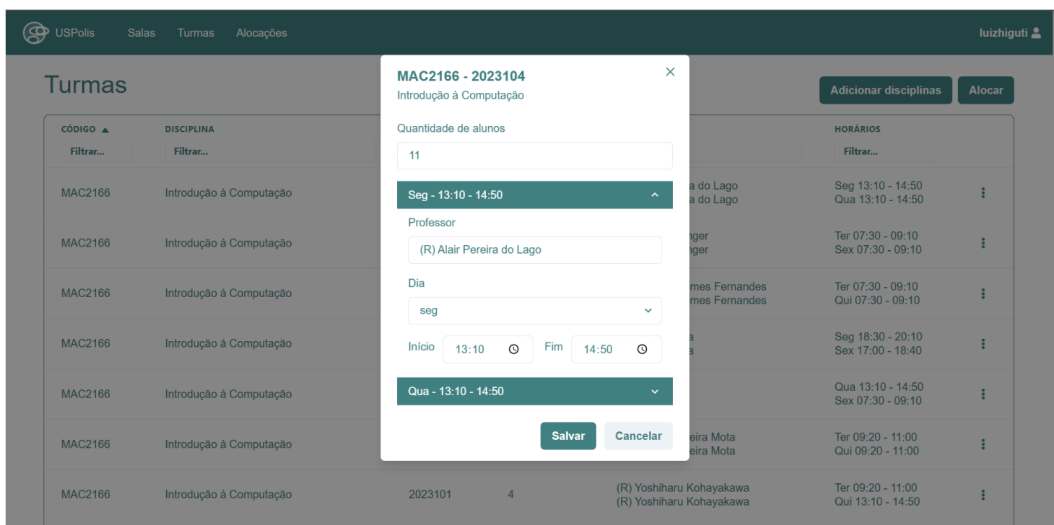
Figura 10 – Gestão de salas no sistema administrativo



Fonte: Higuti e Kondo (2022).

2. **Gestão de disciplinas, turmas e aulas:** as disciplinas e suas respectivas turmas e horários de aulas podem ser adicionadas ao sistema de forma automatizada, com a utilização de um *web scrapper* que busca as informações publicadas no JupiterWeb. A Figura 11 ilustra o painel de gestão de turmas.
3. **Alocação de salas:** com as salas e turmas adicionadas, o sistema administrativo permite que seja realizada uma alocação automatizada de turmas entre as salas de aula disponíveis (Figura 12). Além disso, a alocação obtida pode ser gerenciada e alterada manualmente.

Figura 11 – Gestão de turmas no sistema administrativo



Fonte: Higuti e Kondo (2022).

Figura 12 – Alocações de turmas em salas no sistema administrativo



Fonte: Higuti e Kondo (2022).

O sistema administrativo tem o papel não apenas de gerenciar as turmas que são exibidas no aplicativo, mas servir de apoio para os departamentos da faculdade realizarem suas alocações de forma simplificada e unificada.

### 3.3 Concorrentes

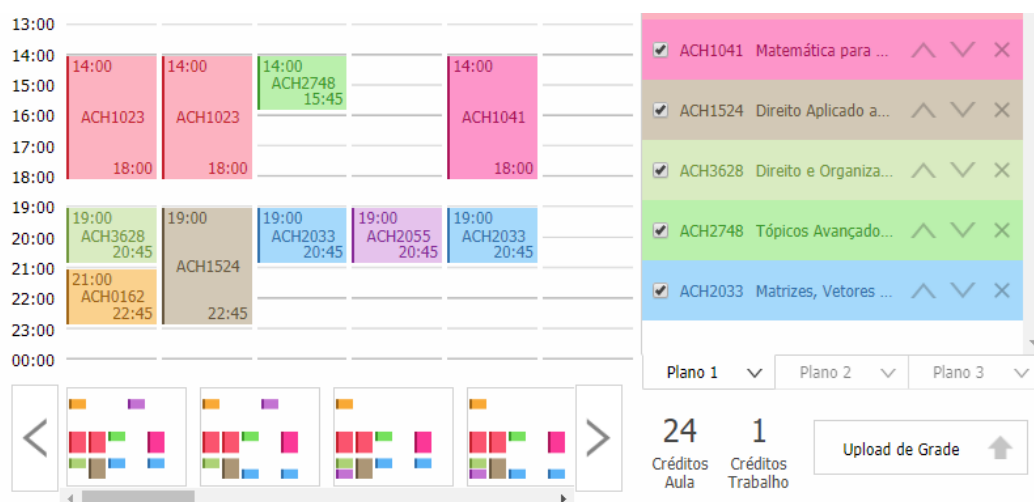
Dada a importância do problema de acompanhamento de turmas, horários e alocações, existem no mercado alguns concorrentes diretos do USPôlis. Serão discutidos nesta seção duas soluções *open source*, o MatrUSP e o *Meeting Room Booking System* (MRBS).

### 3.3.1 MatrUSP

**MatrUSP** (2011) é uma ferramenta de planejamento de horários acadêmicos desenvolvido por alunos e professores do Instituto de Matemática e Estatística da USP, e não tem vínculo oficial com a universidade. O sistema proporciona um ambiente interativo para a composição de horários semanais, permitindo aos usuários alocar disciplinas de acordo com os créditos acadêmicos correspondentes, como apresentado na Figura 13. O sistema, além de contar com disciplinas de cursos de toda a USP, também permite que o usuário monte múltiplas grades, as salve local e remotamente, e as exporte para ferramentas de calendário pessoal.

Entretanto, é importante notar que, diferentemente do USPolis, que também auxilia no acompanhamento de horários, o MatrUSP não conta com as alocações de salas de aula. Além disso, ele não tem sido atualizado ou mantido desde 2019 e portanto não reflete o atual cronograma acadêmico da universidade.

Figura 13 – Visualização de horários no sistema MatrUSP



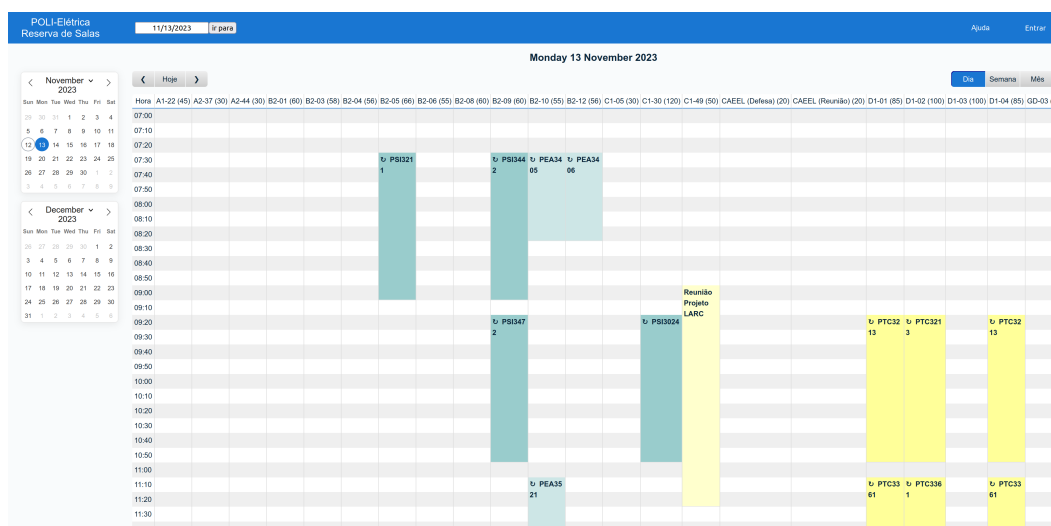
### 3.3.2 Meeting Room Booking System - MRBS

O **MRBS** (2000), ilustrado na Figura 14, é uma aplicação *web* para reserva de salas, adotado no prédio de Engenharia Elétrica da EPUSP. Por este prédio, estão espalhados códigos QR que levam à página *online* do sistema em uso, permitindo que os alunos vejam a alocação em tempo real das salas que serão utilizadas por cada turma.

Contudo, apesar de sua utilidade e funcionalidades avançadas, o MRBS possui limitações quando comparado ao USPolis. Sua utilização restrita ao prédio de Engenharia Elétrica da EPUSP não atende às necessidades de toda a comunidade acadêmica, especialmente considerando que muitos alunos têm disciplinas em diversos prédios. Além disso, ao contrário do USPolis, o MRBS não oferece a opção de personalização de grade horária para os usuários, um recurso que permite aos alunos organizarem seus horários de aula de

maneira personalizada e eficaz na seção de 'Minhas Aulas' do USPólis. O aplicativo também inclui mapas detalhados dos prédios da universidade, auxiliando os alunos a localizarem suas salas de aula com mais facilidade. Esses recursos, combinados com a capacidade de atualização constante das informações de aulas e eventos, e a compatibilidade com múltiplas plataformas móveis, tornam o USPólis uma ferramenta adaptável às necessidades dos alunos.

Figura 14 – Sistema MRBS utilizado no prédio de engenharias elétrica e da computação



### 3.4 Considerações Finais do Capítulo

Neste capítulo foram introduzidos o sistema administrativo de USPólis e o aplicativo legado, que estabelecem a base para a criação do projeto do novo aplicativo USPólis.

Além disso, foram apresentados possíveis concorrentes do USPólis e suas respectivas funcionalidades. A Tabela 1 apresenta um comparativo entre funções presentes em cada um deles, resumindo a relevância do USPólis dentro de seu contexto: o aplicativo reúne funcionalidades presentes em ambos os seus principais concorrentes, tornando-o adequado para o uso na comunidade politécnica.

Tabela 1 – Comparativo de funcionalidades presentes no USPólis, MatrUSP e MRBS

Funcionalidade	USPólis	MatrUSP	MRBS
Listagem de disciplinas e turmas	Sim	Sim	Sim
Mapas	Sim	Não	Não
Grade horária personalizada	Sim	Sim	Não
Alocações de salas de aula	Sim	Não	Sim

No capítulo seguinte, será discutido o método de trabalho que levou à construção do novo aplicativo USPólis, com as funcionalidades presentes na Tabela 1 e outras que serão descritas nos próximos capítulos.

## 4 Método do trabalho

No desenvolvimento deste projeto, houve quatro diferentes etapas, definidos como ciclos de entregas:

- No primeiro, o aplicativo legado foi recriado com o uso das novas tecnologias escolhidas e novo *design*.
- No segundo, foi executado um Design Sprint com o objetivo de descobrir novas potenciais funcionalidades para o aplicativo.
- No terceiro, foi implementada uma ferramenta de busca automatizada de disciplinas, além de feito um teste A/B com essa funcionalidade.
- No quarto, foi criada a esteira de CI/CD do projeto e desenvolvida a funcionalidade de eventos.

Tais ciclos foram desenvolvidos seguindo o cronograma apresentado na Tabela 2.

Tabela 2 – Cronograma do projeto

Ciclo	Descrição	Período
1	Recriação do <i>app</i> legado	dezembro/2022 a março/2023
2	<i>Design Sprint</i>	abril/2023 a maio/2023
3	Teste A/B e Busca automatizada de disciplinas	junho/2023 a agosto/2023
4	Eventos e esteiras de CI/CD	setembro/2023 a novembro/2023

Neste capítulo, será descrito o método de trabalho utilizado na execução de cada um destes ciclos de entregas.

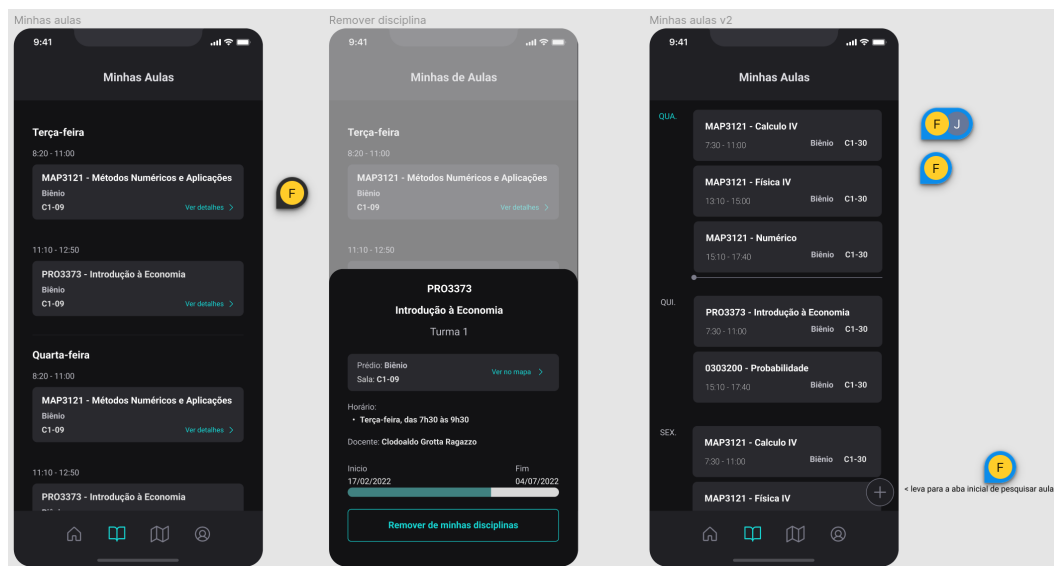
### 4.1 Primeiro ciclo - recriação do aplicativo legado

O passo inicial do desenvolvimento do projeto foi a reconstrução e repaginação do aplicativo legado. Como o legado já não tinha manutenção e o novo *app* foi construído usando outras tecnologias, este passo foi essencial para que novas funcionalidades pudessem depois ser adicionadas, expandindo o escopo inicial do aplicativo.

Assim, foram levantadas as funcionalidades do USPolis legado, para que fossem desenvolvidas sem nenhuma alteração, que englobam listagem de disciplinas, visualização da alocação de turmas nas salas de aula, acompanhamento de grade horária e visualização de mapas, e estão descritas com detalhes no Capítulo 5.

Identificados os requisitos do sistema, o grupo preparou, fazendo uso da ferramenta Figma (<https://www.figma.com/>), o novo *design* para a repaginação do aplicativo, que pode ser observado na Figura 15. Esta interface criada serviu de base para o desenvolvimento subsequente do *software* relativo às funcionalidades previstas.

Figura 15 – *Design* da página de Minhas Aulas do novo USPólis dentro do Figma



Esta versão inicial do USPólis foi desenvolvida antes do início do ano letivo de 2023, para que os alunos da faculdade já tivessem acesso ao aplicativo desde a primeira semana de aula e pudessem utilizá-lo para montar suas grades e observar as alocações de suas turmas. Assim, além de fornecer estas facilidades aos alunos, o grupo já começou a obter dados de uso do aplicativo, que orientaram futuras tomadas de decisão.

## 4.2 Segundo ciclo - execução do Design Sprint

Uma vez desenvolvida a versão inicial do aplicativo replicando as funcionalidades do USPólis legado, o grupo pôde começar a pensar em sua expansão. Para isso, foi utilizada a metodologia do Design Sprint (Seção 2.1), com algumas adaptações para o contexto deste projeto.

A primeira adaptação a ser pontuada é o fato de que todos os encontros do Design Sprint foram realizados de maneira remota, de forma a facilitar a participação de todos os integrantes necessários. Para tal, além de softwares de videoconferência, foi utilizada a ferramenta Miro como uma lousa virtual para as anotações de cada atividade proposta.

Além disso, uma característica do Design Sprint original é sua realização durante o período de uma semana, todos os dias do início da manhã ao fim da tarde. Neste projeto, as atividades foram distribuídas em encontros de duas a três horas ao longo de algumas semanas.



Também destaca-se a composição do time que executou o Design Sprint. Em vez da composição original indicada por Knapp e Kowitz (2016), o *sprint* foi executado pelos integrantes deste grupo de trabalho, que assumiram os papéis de desenvolvedor, especialista em produto e facilitador, além do orientador e co-orientador, que participaram dos encontros enquanto *stakeholders*.

O desenvolvimento do Design Sprint e seus resultados serão descrito em detalhes no Capítulo 7.

### 4.3 Terceiro e quatro ciclos - busca automatizada de disciplinas e eventos

No terceiro e quarto ciclos de entregas, o método de desenvolvimento seguido foi similar ao do primeiro ciclo. A partir dos insumos gerados no Design Sprint, foram montadas as especificações de funcionalidades (descritas no Capítulo 5). Em seguida, foi planejado o *design* de cada uma delas no Figma, e com isso partiu-se para o desenvolvimento do software.

Destacam-se além destas atividades a realização de um teste A/B no terceiro ciclo e a implementação da esteira de CI/CD no quarto. Todo o desenvolvimento do projeto será descrito no Capítulo 7.

### 4.4 Considerações Finais do Capítulo

Neste capítulo, foi exposto o método utilizado para o desenvolvimento de cada ciclo de entregas deste projeto. No capítulo seguinte, será descrita toda a especificação de funcionalidades dos mesmos ciclos de entrega do novo USPólis em formato de histórias de usuário.

# 5 Especificação de Requisitos

Como já apresentado no capítulo anterior, o andamento do projeto USPolis foi dividido em quatro etapas, sendo três delas de desenvolvimento *software* para suas funcionalidades.

O *backlog* de funcionalidades foi composto por épicos, ou seja, requisitos de alto nível do produto. Cada épico, por sua vez, foi refinado em uma série de histórias de usuário, que descrevem uma ação de um tipo de usuário do sistema. Por fim, as histórias de usuário foram divididas em tarefas para a equipe de desenvolvimento, facilitando a implementação.

Neste capítulo serão apresentados os épicos e histórias de usuário advindos do aplicativo legado e do Design Sprint, além de descritos os épicos efetivamente desenvolvidos como resultado final do aplicativo USPolis.

## 5.1 Funcionalidades pré-existentes no aplicativo legado

Como funcionalidades pré-existentes no aplicativo, foram destacados os seguintes épicos (numerados E1, E2, etc.) e as seguintes histórias de usuário (numeradas H1, H2, etc.):

- **E1** - Visualização de mapas
  - **H1** - Como aluno, gostaria de visualizar o mapa de um prédio.
  - **H2** - Como aluno, gostaria de navegar entre os andares de cada prédio da EPUSP.
  - **H3** - Como aluno, gostaria de aproximar e afastar os mapas de cada um dos prédios da EPUSP.
- **E2** - Pesquisa por aulas
  - **H4** - Como aluno, gostaria de pesquisar por cada uma das matérias da EPUSP por código ou nome da disciplina.
  - **H5** - Como aluno, gostaria de obter uma lista de matérias que apresente sua sigla, turma e prédio em que a aula será ministrada.
  - **H6** - Como aluno, gostaria de obter informações detalhadas de cada uma das matérias da lista
- **E3** - Acompanhamento da grade horária

- **H7** - Como aluno, gostaria de acompanhar as principais informações sobre minhas disciplinas que estou cursando.
- **H8** - Como aluno, gostaria de adicionar e remover disciplinas da minha grade.

Além disso, foi incluída uma história de usuário relativa a *feedback*, pois a equipe desenvolvedora gostaria de **receber as opiniões dos usuários** acerca do aplicativo desde seu lançamento:

- **E4** - *Feedback* dos usuários
  - **H9** - Como aluno, gostaria de enviar à equipe USPolis minhas opiniões sobre minha experiência com o aplicativo.

## 5.2 Funcionalidades idealizadas no Design Sprint

Como será abordado no Capítulo 7, o Design Sprint realizado deu origem a diferentes possibilidades de funcionalidades novas para o aplicativo a ser expandido. Cada uma das funcionalidades desenvolvidas foi descrita como um épico e detalhado em histórias.

- **E5** - Busca automatizada de aulas com base no curso e período
  - **H10** - Como aluno, gostaria de selecionar meu curso e período no aplicativo.
  - **H11** - Como aluno, gostaria de obter automaticamente as disciplinas do meu curso e período para não precisar selecionar cada uma delas manualmente.
  - **H12** - Como aluno, gostaria de selecionar a minha turma nas disciplinas obtidas.
  - **H13** - Como aluno, gostaria de adicionar as turmas de minha escolha no meu horário semanal.
- **E6** - Eventos e grupos de extensão
  - **H14** - Como aluno, gostaria de visualizar as informações de eventos que vão ocorrer na EPUSP.
  - **H15** - Como entidade, gostaria de divulgar um evento que realize no USPolis.
  - **H16** - Como grupo de extensão, gostaria de divulgar meu processo seletivo para os alunos no USPolis.
  - **H17** - Como aluno, gostaria de "curtir" um evento no USPolis, para dar a ele mais visibilidade.

## 5.3 Considerações Finais do Capítulo

Neste capítulo, foi descrita a especificação das funcionalidades desenvolvidas neste projeto. O Capítulo 6 descreverá as tecnologias utilizadas no seu desenvolvimento, e o Capítulo 7 discutirá o processo criativo que levou à decisão de criar cada uma delas.

## 6 Arquitetura e Tecnologias

Este capítulo explora a arquitetura e as tecnologias do USPolis, abordando o aplicativo móvel, a API Rest e os serviços utilizados, detalhando as escolhas tecnológicas e suas implementações.

### 6.1 Arquitetura Geral do Sistema

O USPolis é um sistema que integra um aplicativo móvel e um módulo web administrativo por meio de um *back-end* monolítico. Este *back-end* é o principal componente do sistema, fornecendo uma interface unificada para seus consumidores. No projeto de formatura realizado por [Higuti e Kondo \(2022\)](#), tal monolito foi inicialmente desenvolvido contando com as principais funcionalidades referentes ao USPolis administrativo: criação de disciplinas e alocação de tais disciplinas nas salas de aula. Outro aspecto herdado foi a escolha inicial de um banco de dados não relacional, o que atribui uma alta eficiência no gerenciamento de dados.

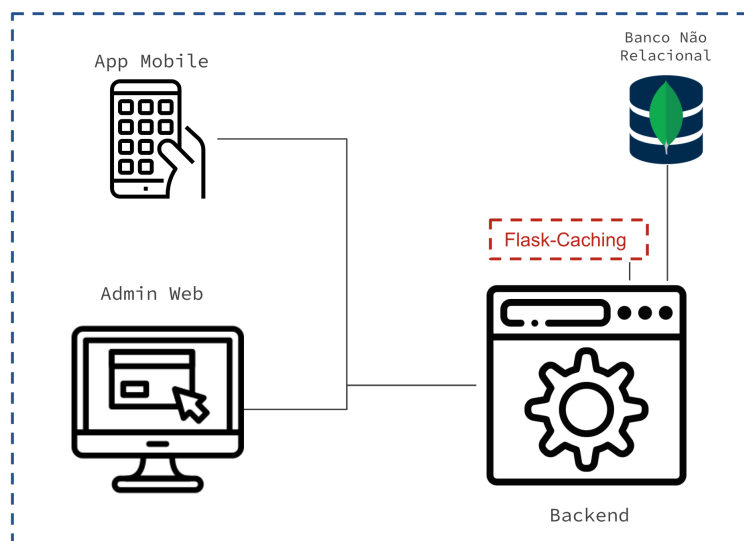
Um aspecto da arquitetura do USPolis é a sua capacidade de escalar. Apesar de ser monolítico, o *back-end* foi projetado de forma a suportar um crescimento significativo, tanto em termos de tráfego de usuários quanto de volume de dados. O sistema foi preparado para suportar um grande volume de consultas de aulas e permitir uma grande quantidade de aulas cadastradas. Além disso, a implementação de uma camada de cache é uma escolha a fim de melhorar o desempenho do aplicativo móvel, reduzindo a latência e proporcionando uma experiência de usuário mais fluida e responsiva. É possível observar na [Figura 16](#) a arquitetura descrita de maneira simplificada.

### 6.2 Aplicativo Móvel

Desenvolvido principalmente em Javascript, utilizando-se do framework *React Native*, o aplicativo móvel USPolis está disponível tanto para Android quanto para iOS. O uso do React Native não só permite uma rápida iteração no desenvolvimento de aplicativos para diferentes sistemas operacionais simultaneamente, mas também garante uma experiência de usuário consistente.

O Expo (<https://expo.dev/>) foi usado como ferramenta auxiliar, a fim de facilitar o processo de desenvolvimento e oferecer um conjunto de bibliotecas e ferramentas que aceleram a construção de funcionalidades complexas. Tais funcionalidades, utilizadas e integradas no aplicativo USPolis, incluem edição personalizada de fonte, integração com

Figura 16 – Arquitetura Simplificada do Sistema USPolis



Fonte: de autoria própria.

o aplicativo de e-mail do celular do usuário e armazenamento *offline* de informações do usuário.

Além disso, a ferramenta Expo desempenha um papel importante na promoção da integração e entrega contínua, elementos centrais do DevOps. Por meio de suas funcionalidades e sua plataforma Web, o Expo permite configurações específicas a fim de integrar-se com uma base de código. Ele automatiza o processo de compilação do aplicativo, detectando mudanças no código e integrando-as continuamente. Isto não só agiliza o desenvolvimento, mas também assegura consistência e eficiência nas compilações para as plataformas iOS e Android, que devem ser realizadas de maneira independente. Além disso, o Expo simplifica o processo de implantação do aplicativo nas principais lojas, otimizando a distribuição e o gerenciamento de versões.

No contexto do DevOps, o GitHub Actions é a ponte entre a base de código e a esteira de distribuição automática. Integrado ao repositório do projeto, o GitHub Actions permite a criação de *workflows* automatizados para compilação, teste e implantação do aplicativo. Com ele, cada atualização da base de código pode desencadear uma série de ações predefinidas, como a execução de testes automatizados e a compilação de builds, garantindo que cada alteração esteja em conformidade com os padrões de qualidade antes de ser integrada. No caso do aplicativo USPolis, a utilização do GitHub Actions é o que permite a integração do repositório com as funcionalidades de entrega contínua do Expo.

A Figura 17 ilustra essa integração, em que cada novo commit no repositório aciona o Github Actions, iniciando os eventos automáticos de testes, e subsequentemente, aciona a interface do Expo a fim de realizar o build do aplicativo. O Expo, então, operando como um serviço de nuvem, recebe o código-fonte atualizado e inicia o processo de compilação

para as plataformas Android e iOS independentemente. Após a compilação, o Expo também gerencia a distribuição das compilação para as lojas de aplicativos.

Figura 17 – Funcionamento: Github Actions + Expo



Fonte: de autoria própria.

### 6.3 API Rest

A API Rest do USPolis é construída em Python, utilizando a biblioteca *open source* Flask. Essa escolha, mantida do projeto de formatura anterior, fornece uma base flexível e robusta para a construção de serviços web, ideal para as necessidades do USPolis. O MongoDB, adotado como banco de dados não relacional, é outro elemento herdado do projeto do sistema administrativo (HIGUTI; KONDO, 2022).

A API existente foi expandida e adaptada para atender às novas demandas do projeto USPolis. Entre as funcionalidades desenvolvidas estão a listagem de salas de aula e rotas de importação de alocações, a fim de facilitar a alocação de salas, aspectos importantes para a interação entre o componente móvel e administrativo do sistema. Na Tabela 3, é apresentada a listagem de rotas desenvolvidas no *back-end* a fim de alimentar o aplicativo.

Além disso, o *back-end* também conta um sistema completo de CI/CD utilizando exclusivamente o Github Actions, um recurso não presente no projeto original. Este sistema compila e distribui a aplicação por meio de um *workflow* automatizado, permitindo que cada commit à *branch* principal do projeto seja integrado e implantado em produção de maneira eficiente.

Tabela 3 – Endpoints da API Rest do USPolis

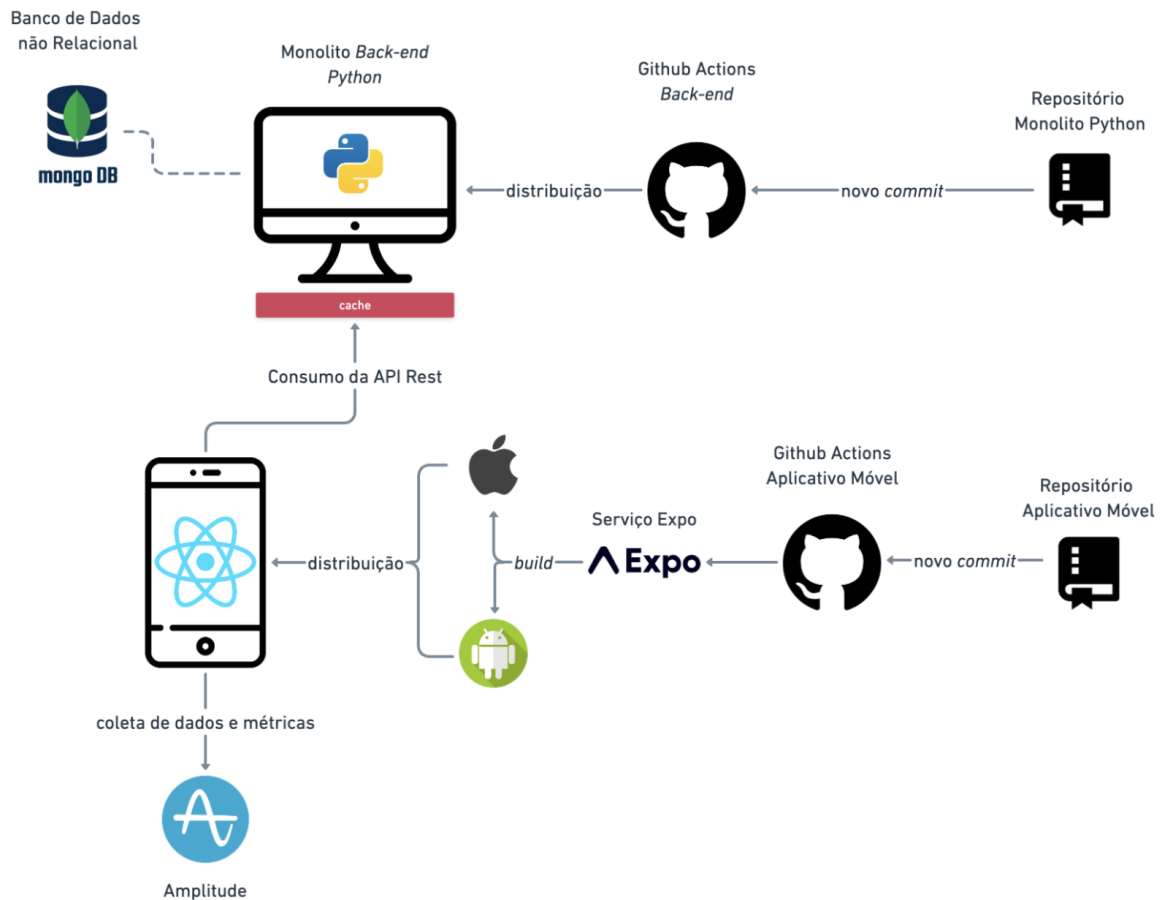
Endpoint	Método	Cliente	Descrição
/mobile/classes	GET	App	Retorna uma lista de todas as turmas disponíveis.
/mobile/comments	POST	App	Permite a criação de um comentário ou <i>feedback</i> sobre o aplicativo.
/mobile/programs	GET	App	Retorna a lista de cursos disponíveis na EPUSP.
/mobile/programs/classes	GET	App	Lista todas as disciplinas pertencentes a um curso.
/mobile/institutional-events	GET	App	Lista todos os eventos acontecendo ou por vir.
/mobile/institutional-events/<eventId>/like	PATCH	App	Adiciona um <i>like</i> no evento.
/mobile/institutional-events/<eventId>/remove-like	PATCH	App	Remove um <i>like</i> do evento.
/institutional-events	POST	Admin	Cria um novo evento.
/institutional-events/<eventId>	PATCH	Admin	Atualiza um evento.
/institutional-events/<eventId>	DELETE	Admin	Deleta um evento.
/institutional-events	GET	Admin	Lista todos os eventos.

### 6.3.1 Arquitetura Completa

A Figura 18 apresenta um diagrama que descreve a arquitetura completa do USPolis. Inicialmente, aliado ao repositório de código, tem-se o uso de GitHub Actions para o back-end e para o aplicativo móvel, possibilitando a adoção das práticas do DevOps no projeto. Adicionalmente, a arquitetura indica a utilização do serviço Expo, como descrito anteriormente, que ajuda a facilitar o desenvolvimento e a distribuição do aplicativo móvel, bem como a ferramenta Amplitude para a coleta de métricas associadas ao uso do aplicativo. Esses componentes representados na figura, destaca como eles interagem entre si, e como contribuem para um fluxo de trabalho automatizado e eficiente, do desenvolvimento à entrega do produto final.



Figura 18 – Arquitetura Completa do Sistema USPolis



Fonte: de autoria própria.

## 6.4 Serviços Utilizados

Na construção do USPolis, alguns serviços externos foram selecionados a fim de facilitar a observabilidade do aplicativo bem como o desenvolvimento de algumas *features*. Esta seção detalha tais serviços utilizados, explicando como cada um contribui para o desempenho geral e a experiência do usuário.

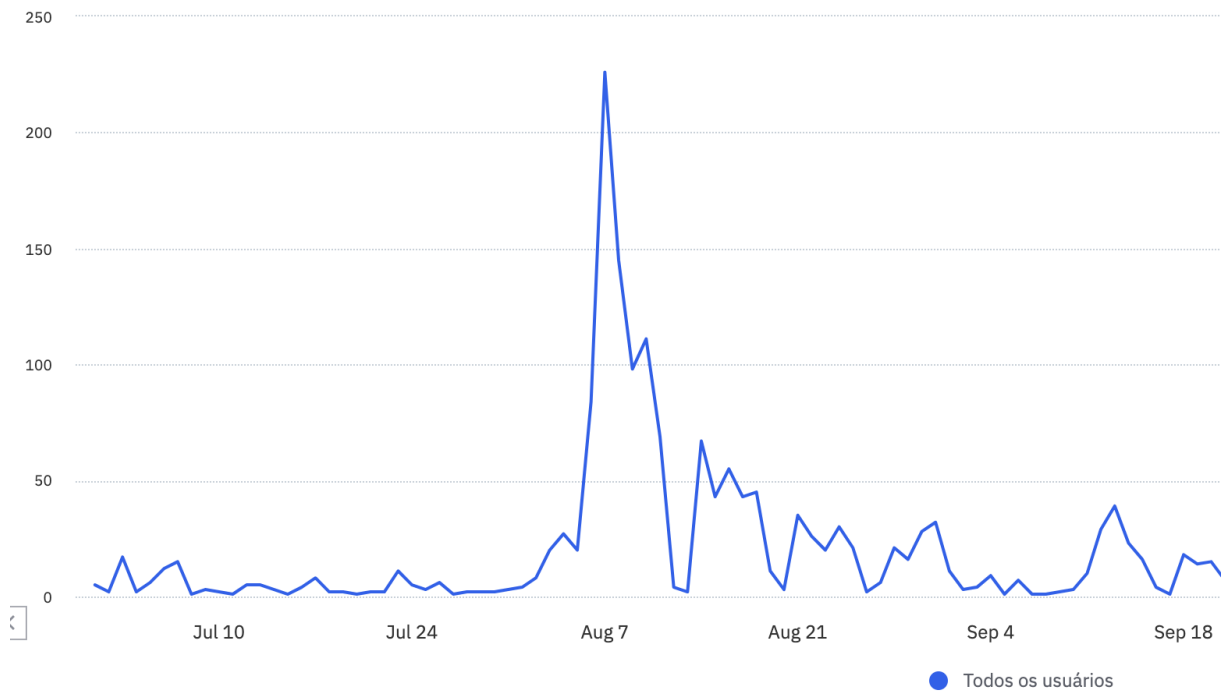
### 6.4.1 Amplitude

O uso do Amplitude (<https://amplitude.com/>) no USPolis permite monitorar métricas importantes como padrões de uso, frequência de acesso e comportamento do usuário, permitindo ajustes estratégicos para melhorar a experiência do usuário e otimizar o desempenho do aplicativo.

Com essa ferramenta, é possível analisar o comportamento do usuário ao longo de sua jornada de uso no aplicativo, e quais são as áreas de principal interação, permitindo identificar os pontos chave de melhoria e engajamento do aplicativo.

A Figura 19 ilustra um exemplo da métrica principal do aplicativo, que é o número de usuários diários. Outras métricas e análises serão aprofundadas no Capítulo 8.

Figura 19 – Taxa de acesso dos usuários ao *app* entre o período de Julho a Setembro



Fonte: de autoria própria.

### 6.4.2 SendGrid

Integrado ao USPolis, o SendGrid (<https://sendgrid.com/en-us>) desempenha um papel importante na comunicação com os usuários. Sua utilidade no sistema é na aba de *feedback* no aplicativo, onde os usuários podem enviar suas opiniões e sugestões. Cada *feedback*, juntamente com o e-mail do usuário, é automaticamente encaminhado para o e-mail oficial do USPolis, facilitando a gestão de *feedbacks* e o engajamento com a comunidade de usuários.

## 6.5 Considerações Finais do Capítulo

Sobre as tecnologias empregadas no USPolis, é importante destacar a sinergia entre a herança do projeto anterior e as inovações implementadas. A escolha de manter o Python e Flask, juntamente com o MongoDB, do projeto de formatura precedente, revelou-se uma estratégia eficiente, garantindo uma transição suave e o desenvolvimento contínuo. As inovações introduzidas, incluindo novas funcionalidades na API, o desenvolvimento do aplicativo móvel com React Native, a esteira completa de CI/CD com o Github Actions

e ferramentas externas utilizadas refletem um esforço de modernização e melhoria na usabilidade e funcionalidades do sistema.

Com esta base tecnológica estabelecida, o próximo capítulo irá explorar a jornada do USPolis desde sua concepção inicial até a implementação final, abordando como as metodologias, estratégias e tecnologias adotadas impactaram no desenvolvimento do projeto, a fim de enriquecer a experiência educacional da EPUSP.

## 7 Desenvolvimento do Trabalho

Este capítulo apresenta a trajetória do aplicativo USPolis, detalhando a implementação de suas funcionalidades desde sua concepção. Por meio da execução da metodologia do Design Sprint, entrevistas com usuários e teste A/B, este capítulo mostra o processo evolutivo que resultou na entrega do aplicativo visando enriquecer a experiência do aluno na EPUSP.

### 7.1 Entrega Inicial

Partindo das histórias de usuário para o primeiro ciclo de entregas descritas na seção 5.1 (épicos E1 a E4), as tarefas foram divididas entre os membros da equipe e executadas, partindo do zero até um *app* novo, obedecendo as funcionalidades anteriormente levantadas. O novo aplicativo foi então distribuído para os usuários que já utilizavam a versão anterior e disponibilizado nas lojas de aplicativo para novos usuários, e dessa forma, seria possível o início da realização de análises em cima dos resultados e coleta de dados de tais usuários.

Nas figuras a seguir, é possível encontrar um comparativo lado a lado das experiências em cada uma das jornadas dos usuários no aplicativo legado e na versão nova.

A Figura 20 faz a comparação na listagem de aulas. O aplicativo novo, além de ter um aspecto mais moderno e simplista, mantém todas as informações de aulas numa mesma coluna, evitando o desalinhamento visto no aplicativo legado, e também conta com uma barra seletora para visualizar apenas as aulas alocadas em cada prédio.

Na Figura 21, está a comparação no detalhe de uma aula. O aplicativo novo conta com um componente do tipo *drawer* - um painel deslizante que sobe da parte de baixo da tela - no lugar do modal do legado, com informações mais compactadas na tela. Além disso, o botão "ver no mapa" leva o usuário ao mapa do prédio em que aquela aula foi alocada, para que ele possa encontrar a sala de aula.

A Figura 22 compara a aba de Mapas nos aplicativos. No novo sistema, existe também a barra seletora de prédio, como na aba de listagem de aulas, e o texto indicando o andar visualizado ganhou mais destaque. Além disso, os botões de subir e descer de andar tiveram seus lados invertidos - no aplicativo legado, o botão de subir estava na esquerda, o que o grupo havia considerado contraintuitivo - e a imagem foi substituída por uma seta.

Por fim, a Figura 23 ilustra a aba de Minhas Aulas, que indica a grade horária semanal do usuário e também ganhou um aspecto mais moderno e simples, além de indicar claramente para o usuário o nome de cada disciplina, não apenas o seu código.

Figura 20 – Comparação USPolis Legado e USPolis Novo - Listagem de Aulas

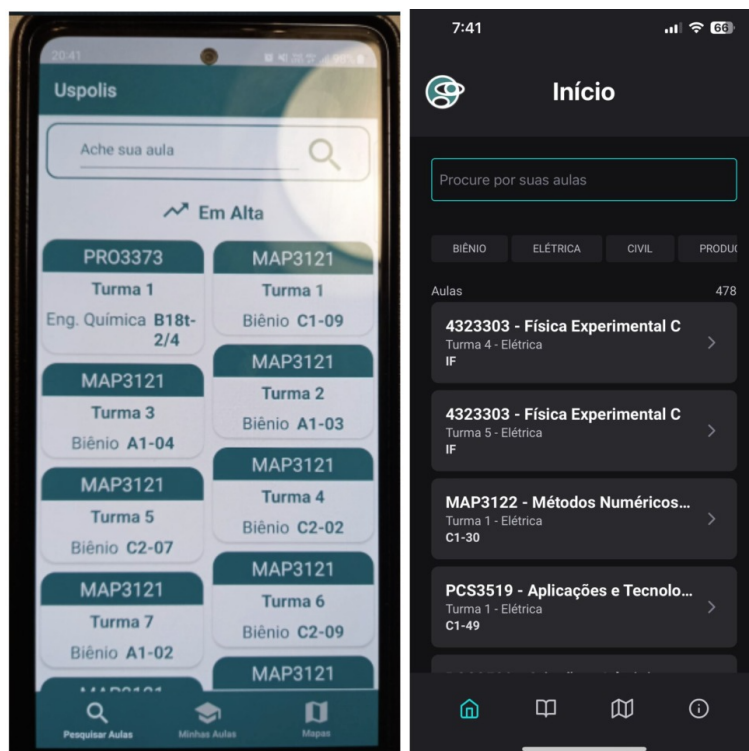


Figura 21 – Comparação USPolis Legado e USPolis Novo - Detalhe de Aula

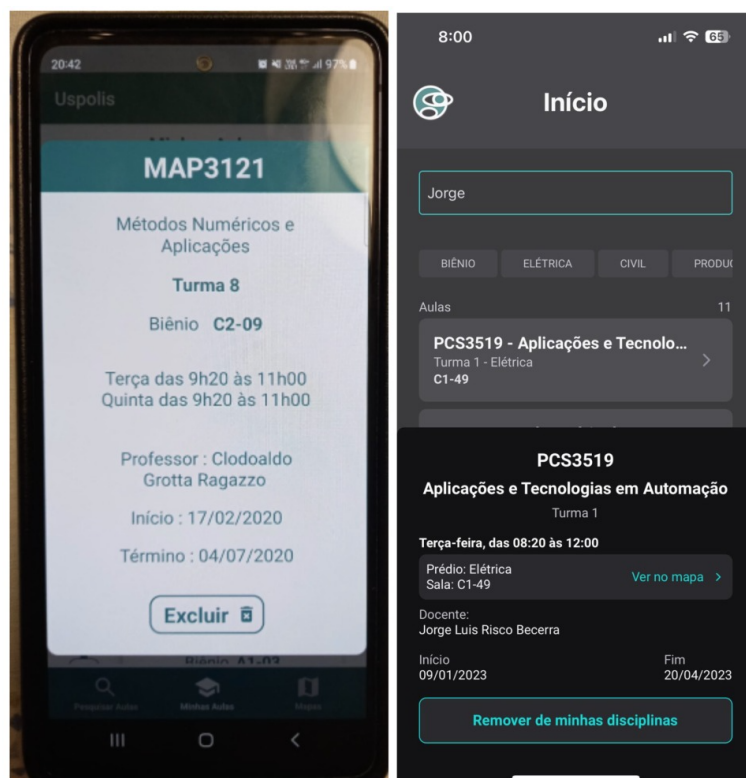


Figura 22 – Comparação USPolis Legado e USPolis Novo - Mapas

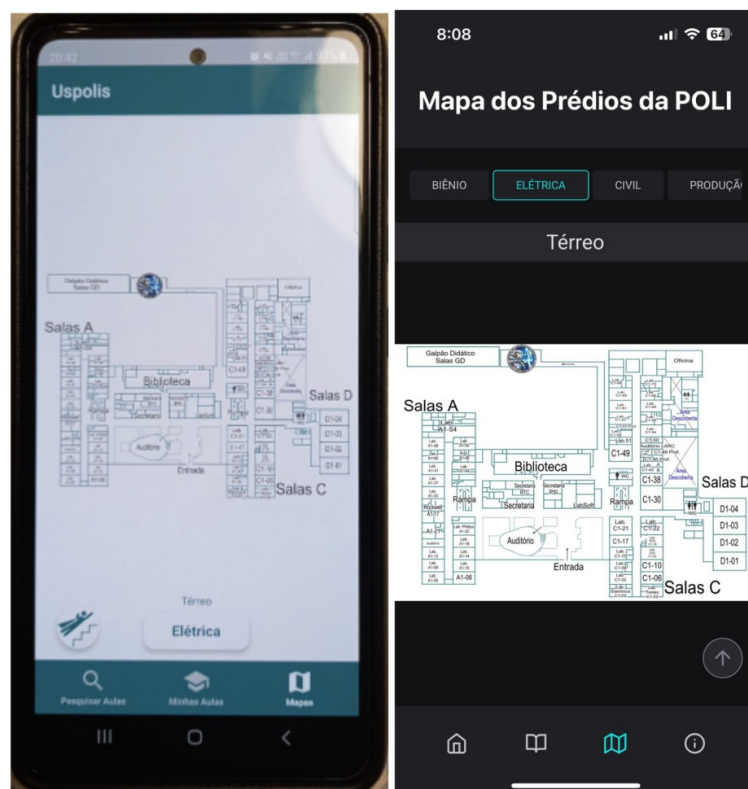
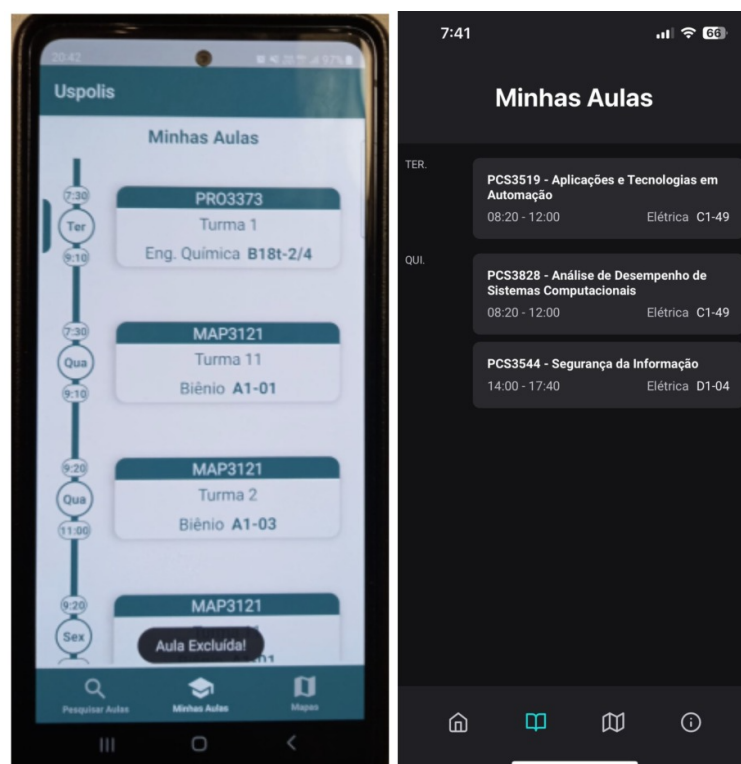
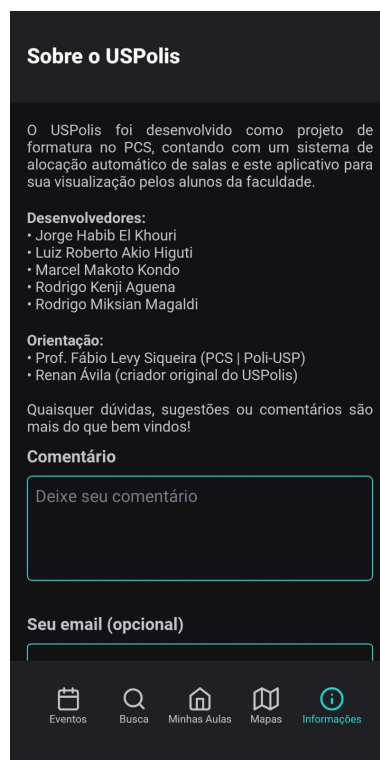


Figura 23 – Comparação USPolis Legado e USPolis Novo - Minhas Aulas



Além das funcionalidades já existentes no aplicativo legado, foi criada a funcionalidade prevista de *feedback*, que ficou em uma aba separada, como visto na Figura 24. Na mesma aba, ainda há um pequeno texto explicando o objetivo do aplicativo e indicando seus criadores e desenvolvedores.

Figura 24 – Aba de *feedback* no USPolis



## 7.2 Design Sprint

Feita a entrega inicial, o grupo tinha em mãos a oportunidade de expandir o aplicativo para abarcar mais e diferentes funcionalidades. Como metodologia para ideação, visando gerar diferentes futuras possibilidades para o USPolis, o grupo optou por utilizar o Design Sprint.

### 7.2.1 Entrevistas com Usuários

Como preparação para o início do Design Sprint, o grupo se propôs a entrevistar usuários reais do sistema, buscando possíveis novas percepções de funcionalidade e experiência dos usuários. Tais entrevistas foram conduzidas de maneira individual, totalizando cinco entrevistados. Além de perguntas básicas como informações pessoais e para contato, foram feitas perguntas mais direcionadas sobre o uso do aplicativo e como os usuários interagiram com ele. O modelo das perguntas feitas nas entrevistas e seus resultados estão documentados no Apêndice A, e foram utilizados como insumo e ponto de partida para o início da execução da metodologia do Design Sprint.

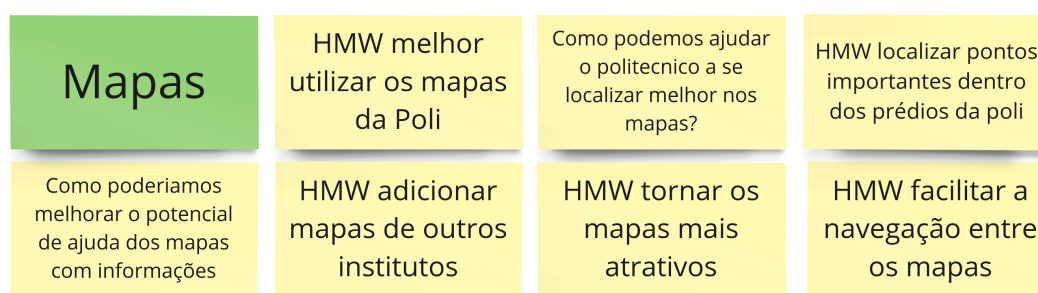
## 7.2.2 Como Poderíamos

A primeira atividade proposta no Dia 1 do Design Sprint envolveu cada um dos integrantes do grupo realizarem um relato de suas entrevistas, resumindo e evidenciando seus principais pontos. Ao mesmo tempo, os outros integrantes, enquanto escutam atentamente, anotam num quadro branco os principais pontos observados. Os integrantes devem sempre anotar utilizando o formato de frase "Como poderíamos".

Nas figuras 25 a 29 é possível observar a conclusão de tal atividade, com todos os pontos anotados no quadro branco por cada integrante, agrupados por similaridade no assunto do comentário.

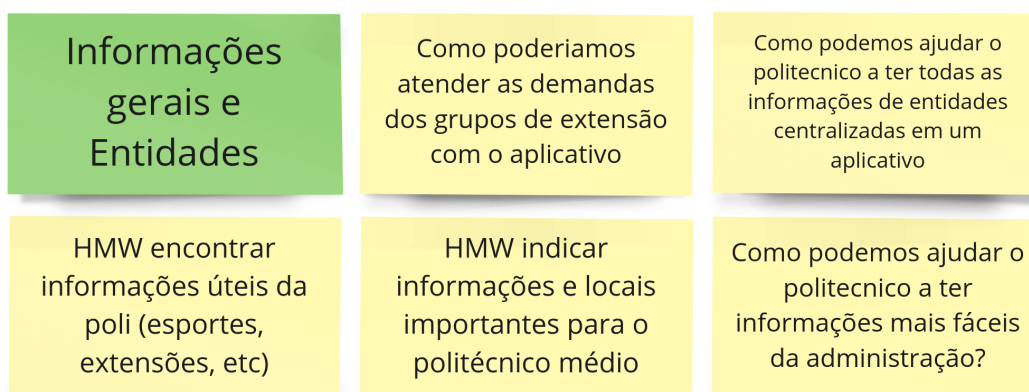
A Figura 25 apresenta as anotações referentes a mapas, que destacam preocupações relativas a melhor utilizá-los no aplicativo, como adicionar mapas de outros institutos da USP, facilitar sua navegação e localizar pontos importantes dentro da EPUSP.

Figura 25 – Como poderíamos - Mapas



Na Figura 26 se encontram os "Como Poderíamos" relativos a informações gerais da faculdade e de entidades estudantis, destacando a necessidade de fornecer informações relevantes para o politécnico.

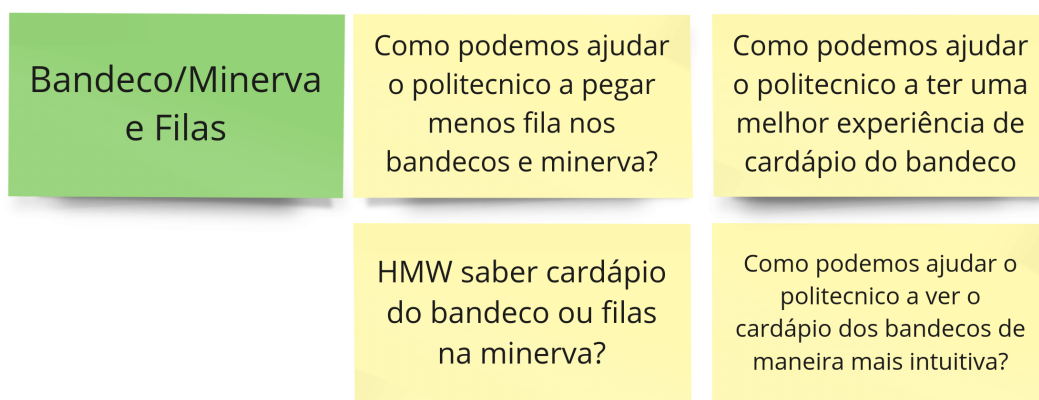
Figura 26 – Como poderíamos - Entidades



A Figura 27 ilustra as anotações sobre pontos que surgiram com frequência nas entrevistas, que são os restaurantes universitários - os entrevistados gostariam de ter informações sobre cardápios e filas dentro do USPolis.

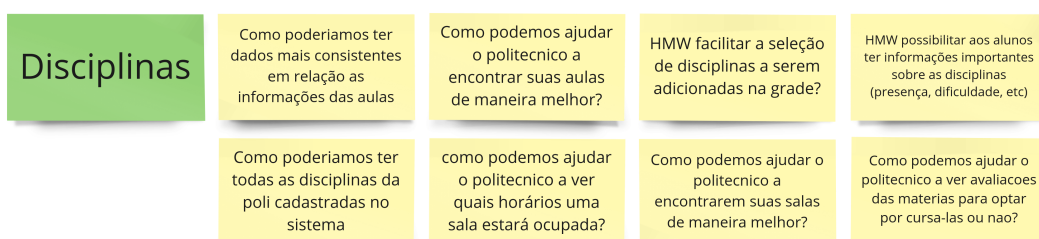


Figura 27 – Como poderíamos - Restaurantes universitários e filas



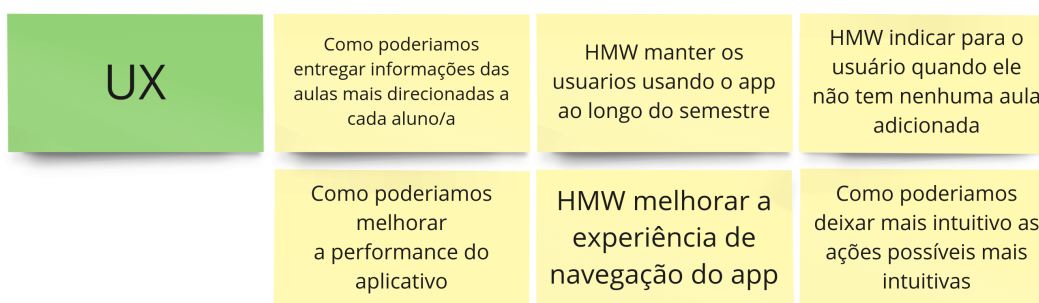
Na Figura 28, estão indicadas as anotações sobre disciplinas, assunto que mais gerou engajamento nas entrevistas. Entre as possibilidades levantadas pelo grupo, estão garantir a consistência dos dados fornecidos no aplicativo, permitir a avaliação de disciplinas dentro do USPolis e facilitar a seleção de disciplinas para adicionar à grade horária pessoal.

Figura 28 – Como poderíamos - Disciplinas



Por fim, a Figura 29 aponta outras preocupações relativas à experiência do usuário (UX), como melhorar a performance e experiência de navegação entre as telas do aplicativo, entregar informações mais direcionadas a cada usuário e mantê-lo usando o aplicativo ao longo do semestre, e não apenas em seu início para descobrir suas alocações de salas.

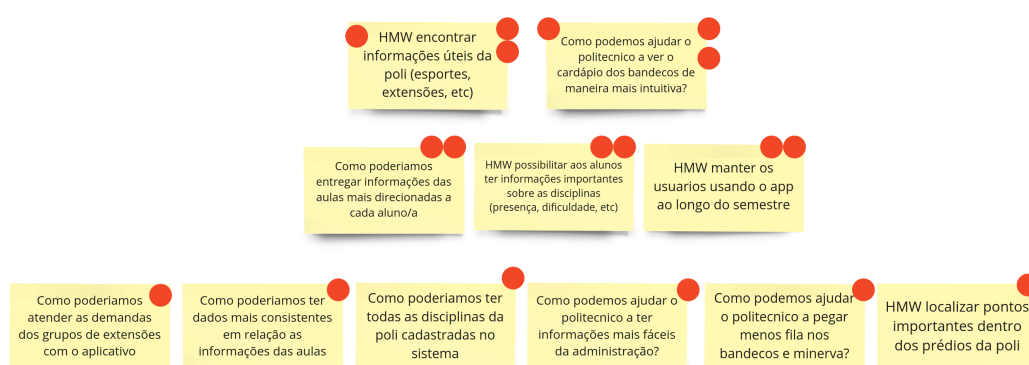
Figura 29 – Como poderíamos - Experiência do usuário



Depois de elencados os principais pontos extraídos das entrevistas, realizou-se uma votação entre os integrantes do grupo a fim de destacar os pontos mais críticos da aplicação numa visão geral. Cada um dos participantes contou com 6 votos, e o resultado da votação

pode ser conferido na Figura 30. Com 3 votos, a necessidade de facilitar a localização de informações úteis da universidade, tanto sobre esportes e extensões quanto sobre restaurantes universitários, destaca-se como uma preocupação central. Em segundo lugar, com 2 votos, surgem as questões de aprimorar a entrega de informações personalizadas sobre aulas, proporcionar informações sobre disciplinas como dificuldade e método de avaliação, e manter a constância de uso do aplicativo ao longo do semestre e não apenas nos inícios de semestre, em que os alunos ainda não conhecem as alocações de salas. Por fim, com apenas 1 voto, estão preocupações tidas pelo grupo como menos relevantes para o momento, e que portanto não foram mais exploradas neste processo.

Figura 30 – Como poderíamos - Resultados da votação



Tal levantamento serviu de insumo para a próxima etapa do Design Sprint, em que foram apontados onde tais questionamentos interferem na jornada do usuário.

### 7.2.3 Jornada do Usuário

Após a definição dos pontos da atividade anterior, foi realizado o mapeamento da jornada atual do usuário no aplicativo. O usuário tem um total de 3 ações possíveis:

- Procurar por turmas de disciplinas
- Adicionar uma turma em suas aulas
- Verificar na seção de mapas onde estão as salas em que suas aulas foram alocadas

Assim, realizou-se o mapa que indica a jornada do usuário, que pode ser observado na Figura 31. Sobre o mapa, foi possível adicionar os "Como Poderíamos" anteriormente mais votados pela equipe, identificando onde tais pontos afetam a jornada do usuário e quais deles são as mais problemáticas, como pode ser visto na Figura 32.

Pode-se notar que houve uma série de pontos de melhoria alocados de forma alheia às funcionalidades já existentes no aplicativo, justamente por serem sugestões de funcionalidades novas. Os principais incluem: (i) visualizar cardápios dos restaurantes

Figura 31 – Jornada do usuário no aplicativo

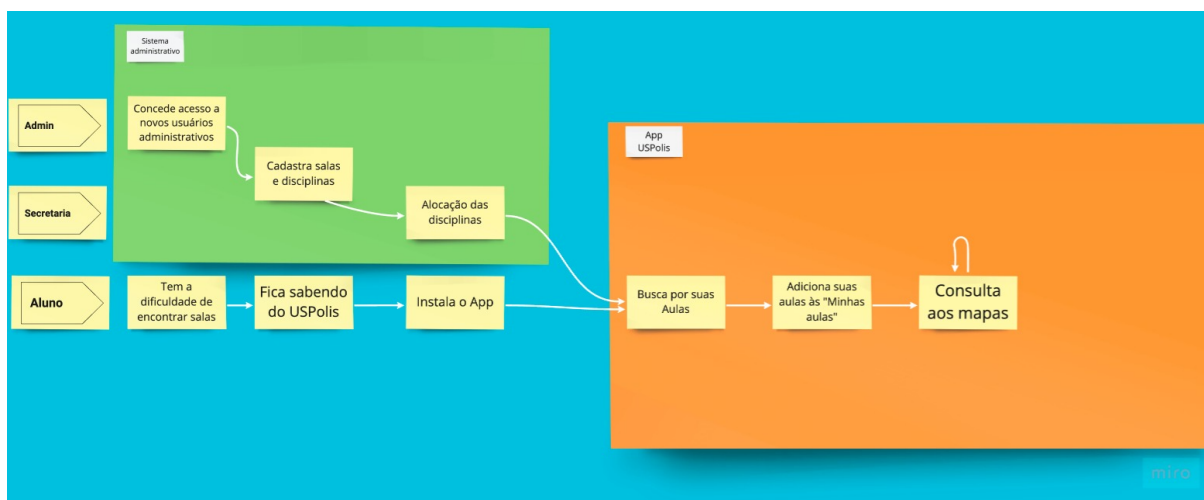
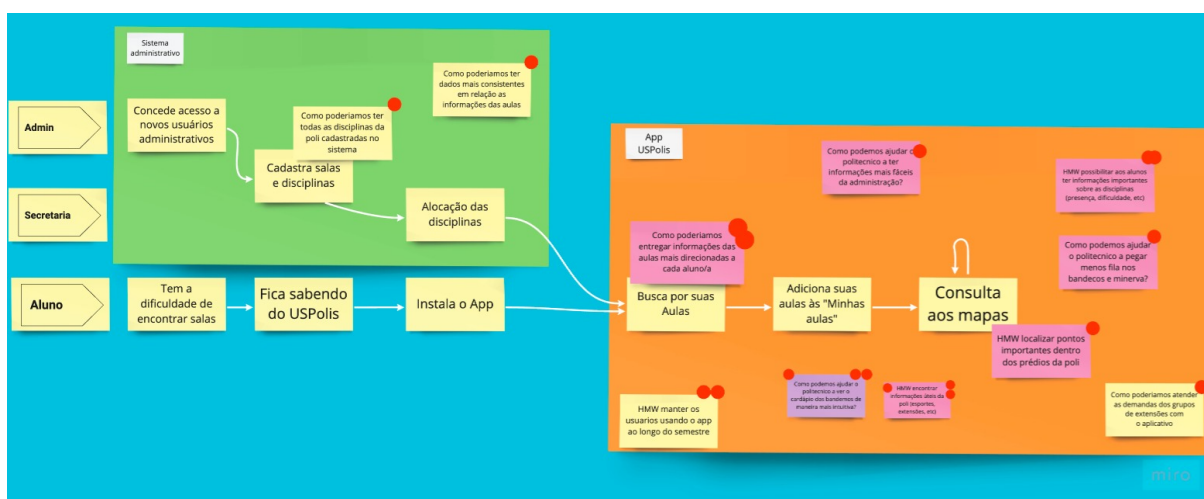


Figura 32 – Jornada do usuário no aplicativo com a inserção de pontos de melhoria



universitários; (ii) apresentar informações sobre outras atividades na faculdade, como esportes e grupos de extensão; (iii) incluir outras informações relevantes sobre as disciplinas, como dificuldade e método de avaliação; e (iv) manter o usuário engajado com o aplicativo mesmo fora do início de semestre.

### 7.2.4 Problema a Ser Resolvido

Tendo em vista o resultado final da atividade anterior, em que identificou-se na jornada do usuário quais são as principais oportunidades, o grupo optou por resolver o seguinte problema relacionado ao ecossistema do aplicativo USPolis:

**"Melhorar a experiência do usuário na busca de informações de aulas"**

## 7.2.5 Elaboração de Propostas de *Design*

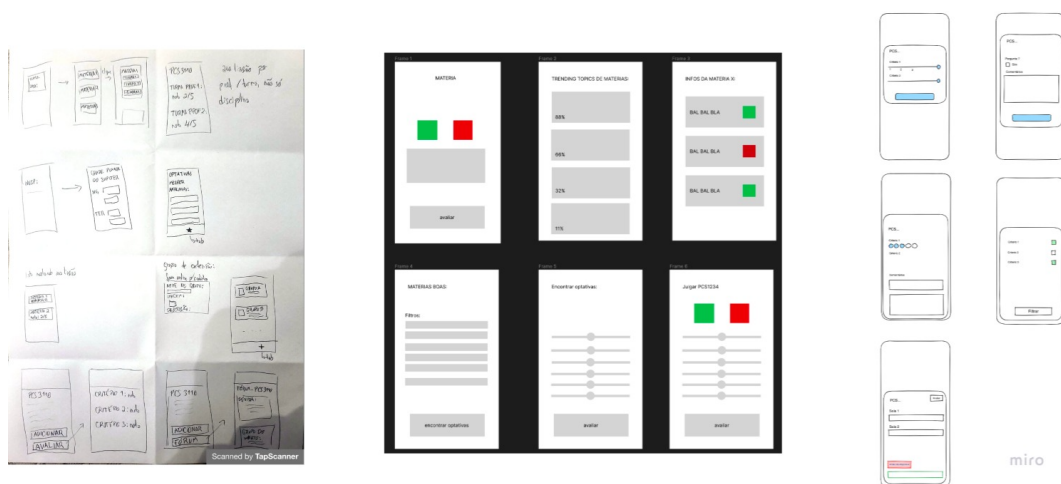
A partir do problema proposto a ser resolvido, o grupo se preparou para a elaboração de uma solução envolvendo mudar partes dos fluxos presentes no aplicativo. Para atingir tal objetivo, o grupo passou por algumas etapas envolvendo *design*.

### 7.2.5.1 Soluções Presentes no Mercado e Rascunhos

Primeiramente, os integrantes do grupo procuraram por referências já existentes no mercado que poderiam ser adaptadas e utilizadas como inspiração no escopo deste projeto, auxiliando na melhoria da experiência do aluno na busca de informações de suas aulas.

A partir de tais referências, os integrantes do grupo esboçaram rascunhos a fim de representar de maneira visual as alterações na jornada do usuário, cada um ainda feito de maneira individual, a fim de manter a pluralidade de ideias. O resultado de tais rascunhos pode ser visto na Figura 33.

Figura 33 – Rascunhos realizados pelo grupo



### 7.2.5.2 Proposta de Rascunhos Finais

Cada integrante do grupo, com base nos rascunhos anteriores, criou uma versão completa da nova jornada de usuário, incluindo a funcionalidade proposta que resolveria o problema central proposto no início do Design Sprint.

A partir desses fluxos finais propostos por cada um dos integrantes, seria realizada uma votação pelas principais alterações de cada um dos esboços. O objetivo da votação é reunir o melhor aspecto de cada um dos protótipos, a fim de criar uma versão final que englobe todas as possíveis melhorias. Nas Figuras 34 a 36, são apresentadas as propostas de rascunho final de cada um dos integrantes do grupo.

Figura 34 – Rascunho final da solução proposta por um dos integrantes do grupo



Figura 35 – Rascunho final da solução proposta por um dos integrantes do grupo

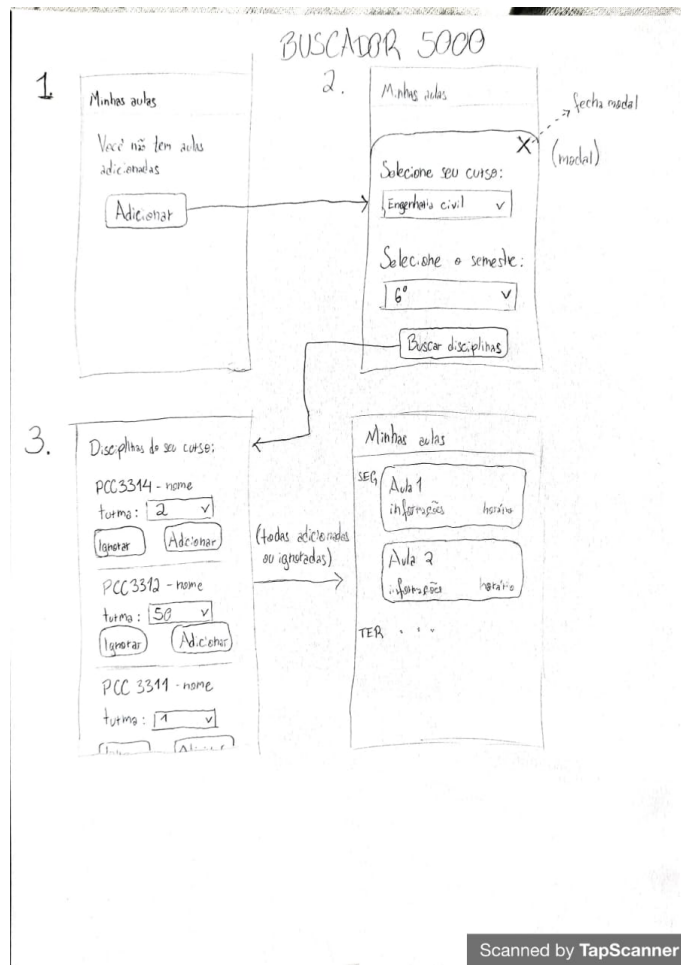
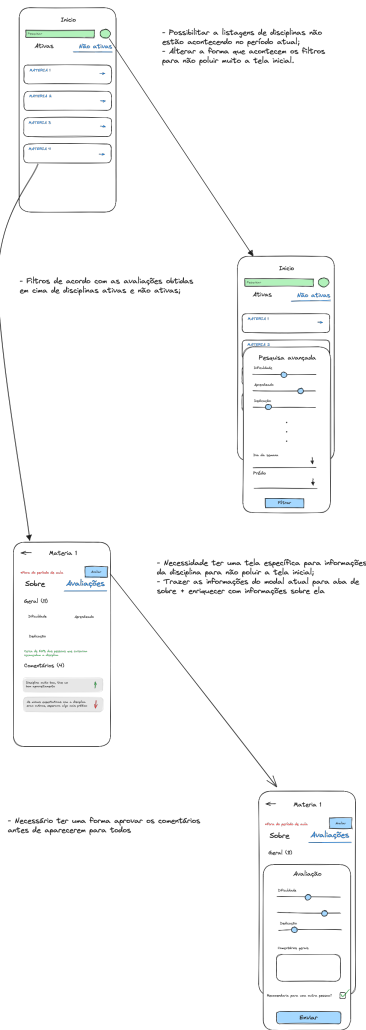


Figura 36 – Rascunho final da solução proposta por um dos integrantes do grupo



Depois de apresentados os protótipos individualmente, realizou-se a votação das principais funcionalidades, fazendo-se uma marcação em verde na funcionalidade alvo do voto. Cada um dos integrantes contou com 6 votos. Na Figura 37, é possível encontrar o resultado da votação, que evidencia três pontos com mais votos a favor: a busca facilitada por disciplinas com base em curso e período, a avaliação de disciplinas no aplicativo e a inclusão de informações sobre grupos de extensão e disciplinas optativas no USP. Polís.

Figura 37 – Funcionalidades mais votadas nas soluções finais rascunhadas

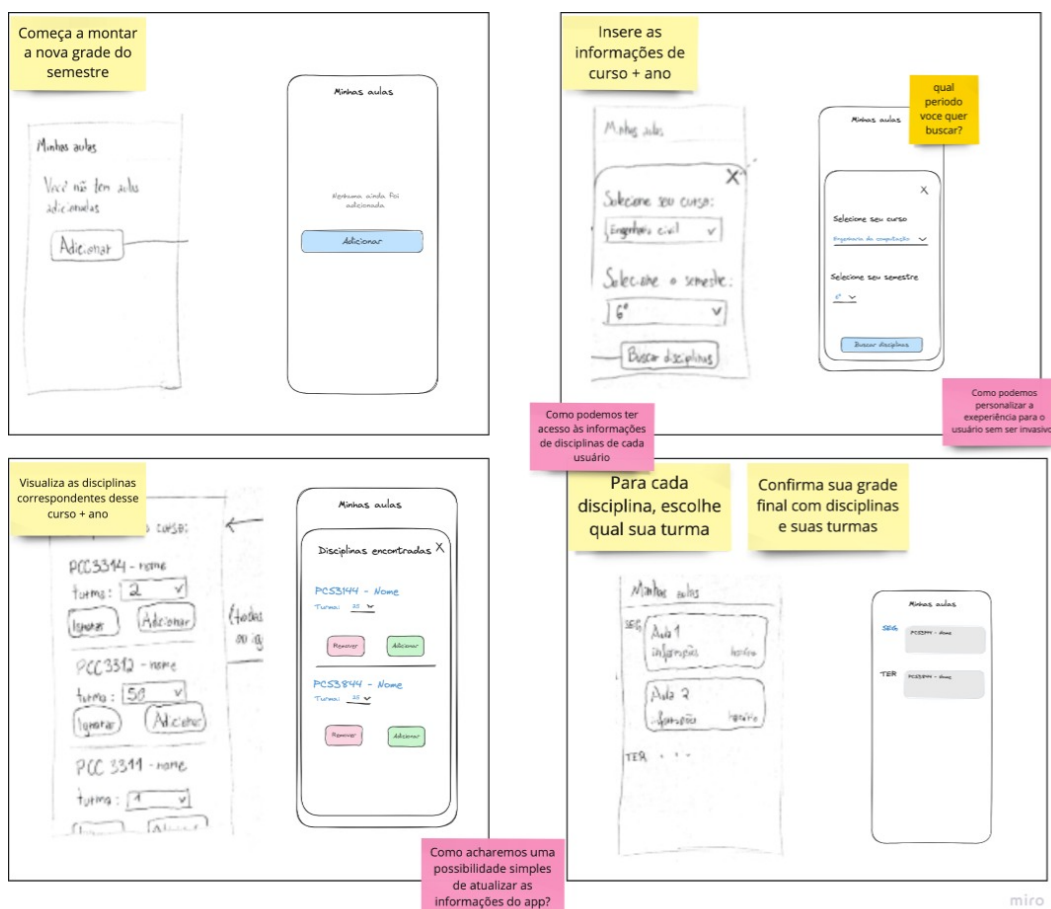


### 7.2.6 Storyboard

Dados os principais aspectos da votação realizada anteriormente, o grupo partiu para a elaboração da nova jornada de usuário, contendo cada uma das alterações mais votadas. Com isso, o grupo obteve três jornadas de usuários novas e distintas:

- Jornada envolvendo procura facilitada por aulas (Figura 38):

Figura 38 – Jornada da busca facilitada por aulas



- Jornada de classificação e ranking de disciplinas (Figura 39):

Figura 39 – Jornada da avaliação de disciplinas



- Jornada envolvendo grupos de extensão e procura por matérias optativas (Figura 40):

Figura 40 – Jornada da inclusão de grupos de extensão e disciplinas optativas

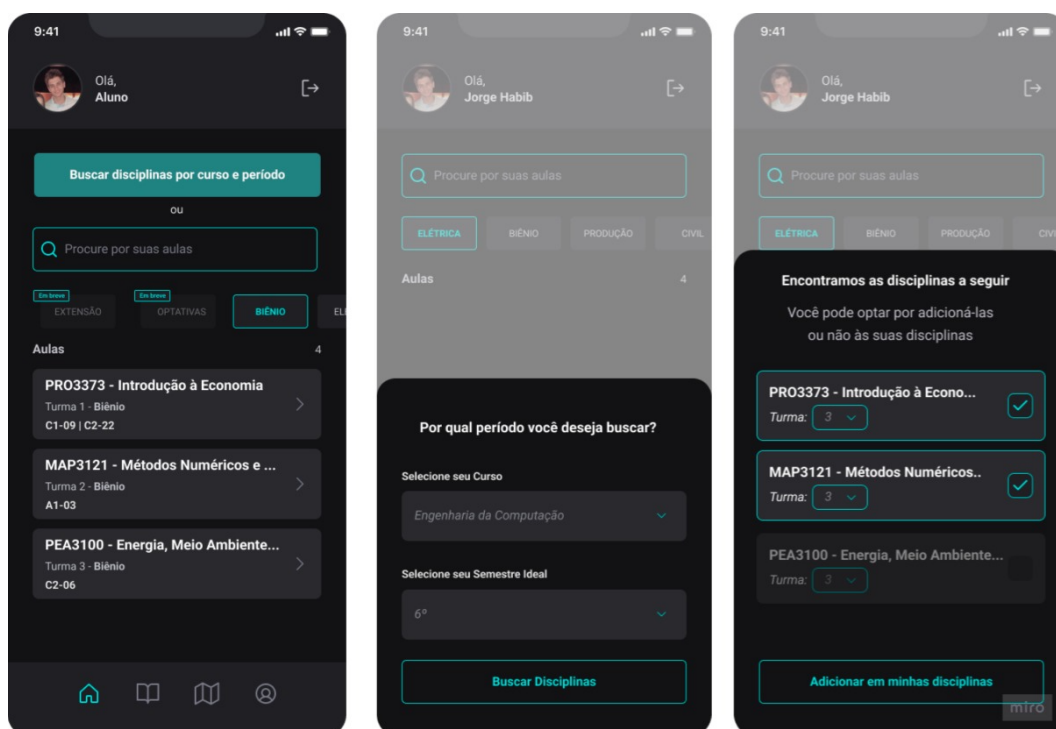


### 7.2.7 Decisão e Prototipação Final

Com base nas prototipações e novas jornadas de usuário, o grupo então optou por seguir inicialmente com a solução de uma **jornada envolvendo a procura facilitada por aulas**, deixando as outras soluções propostas como épicos no backlog do produto. Em seguida, com uma solução definida, foi criada uma prototipação final. Tal protótipo foi utilizado em testes com usuários reais, validando se a solução proposta na nova jornada seria bem recebida.

Na Figura 41 é possível conferir o resultado da prototipação final, que passaria por validação com os usuários.

Figura 41 – Prototipação da funcionalidade escolhida





### 7.2.8 Testes Finais com Usuários

Para finalizar o Design Sprint, o protótipo criado passou por validações com usuários. Na realização destes testes, foi demandado dos usuários que tomassem a ação de adicionar à sua grade todas as disciplinas que fizessem parte de seu curso e período ideal. Conforme explicitado por [Knapp e Kowitz \(2016\)](#), foi deixado claro aos testadores que o que estava sendo avaliado era apenas o produto, e pediu-se para que eles fossem vocais em expressar tudo o que pensaram para realizar a ação demandada.

O teste foi feito com cinco usuários distintos, e no decorrer de cada um deles foram tomadas notas de forma estruturada com base nas seguintes ações do usuário, definidos como critérios C1 a C5:

- **C1** - Usuário começa a montar a nova grade do semestre
- **C2** - Usuário insere as informações de curso + ano
- **C3** - Para cada disciplina, usuário escolhe qual é sua turma
- **C4** - Usuário visualiza as disciplinas correspondentes desse curso e período
- **C5** - Usuário confirma sua grade final com suas disciplinas e turmas

As anotações feitas foram compiladas na Tabela 4, na qual estão indicados o número de comentários tomados como positivos, neutros e negativos acerca de cada critério observado.

Tabela 4 – Número de comentários positivos, neutros e negativos sobre os critérios C1 a C5 após o teste com usuários do Design Sprint

<b>Critério</b>	<b>Comentários positivos</b>	<b>Comentários neutros</b>	<b>Comentários negativos</b>
<b>C1</b>	2	0	3
<b>C2</b>	5	1	0
<b>C3</b>	5	1	1
<b>C4</b>	5	4	0
<b>C5</b>	4	2	0

Os resultados obtidos evidenciam que o principal ponto de desaprovação por parte dos usuários é o critério C1, que marca o início da busca por disciplinas. Inspeccionando os comentários de cada usuário, o grupo detectou que todos os comentários negativos acerca deste critério levantaram a mesma preocupação: o usuário não tinha notado ou achado importante o botão "Buscar disciplinas por curso e período", que inicia o fluxo da nova funcionalidade. Por conta disso, o grupo realizou um teste A/B com duas versões distintas desta funcionalidade, que será descrito a seguir.

## 7.3 Buscador de Disciplinas

Após a realização dos testes finais com usuários no Design Sprint, ficou evidente a necessidade de melhorar ainda mais a experiência dos alunos na busca por informações acadêmicas. Os *insights* obtidos desses testes mostraram que, apesar de as funcionalidades existentes no aplicativo serem valiosas, havia espaço para otimização e inovação, especialmente no processo de busca e organização das aulas.

A funcionalidade de buscador de disciplinas (épico E5 do Capítulo 5) desenvolvida no USPolis foi projetada e testada no intuito de simplificar a jornada do aluno na busca e organização de suas disciplinas. Esta funcionalidade surgiu como uma solução para os alunos conseguirem filtrar e adicionar aulas de maneira mais rápida do que com o filtro já presente no aplicativo, representando um importante papel na usabilidade do aplicativo.

### 7.3.1 Descrição da Funcionalidade

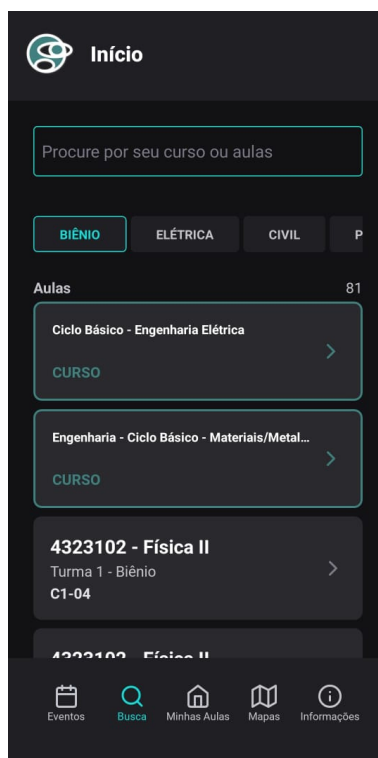
No desenvolvimento do buscador, foram implementadas diferentes interfaces de interação com o usuário. Na Figura 41 é possível observar, na primeira tela, um botão que leva o usuário a "Buscar disciplinas por curso e período". A tela seguinte oferece filtros por curso e período, permitindo uma busca mais direcionada e personalizada. A última tela permite observar uma lista de disciplinas correspondentes aos critérios de busca, com uma opção clara e direta para adicionar a disciplina e turma desejadas ao plano de estudos do aluno.

### 7.3.2 Teste A/B

Para otimizar a experiência do usuário, e com base nos testes finais do Design Sprint descritos na seção 7.2.8, foi realizado um teste A/B com duas versões distintas do buscador. Na primeira versão (versão A), já apresentada na Figura 41, havia um botão com ação clara, incentivando a busca por cursos e períodos específicos. A segunda versão (versão B) integrava à barra de pesquisa os diferentes cursos oferecidos na faculdade, além das disciplinas, com um design diferenciado para distinguir claramente os tipos de resultados (Figura 42). A partir do clique no curso, o fluxo segue da mesma forma que na versão A. Dessa forma, seria possível entender qual das opções atrai mais o usuário para o fluxo do buscador de disciplinas: a versão com o botão principal ou a versão que inclui os cursos na listagem.

Assim, o aplicativo foi distribuído para o público em suas duas versões, obtido por cada usuário de forma aleatória. Este teste A/B foi monitorado com o auxílio da ferramenta Amplitude, coletando dados sobre o comportamento do usuário e sua interação com a nova funcionalidade, fornecendo assim *insights* valiosos sobre a eficácia do design de cada versão.

Figura 42 – Versão alternativa (versão B) do buscador de disciplinas



Como será apresentado no Capítulo 8, a versão B foi a vencedora do teste, sendo atualmente a única versão disponível no USPolis.

## 7.4 Eventos

Após aprimorar a jornada acadêmica dos alunos com a funcionalidade de busca automatizada por aulas, o USPolis se deparou com o desafio de manter o engajamento dos alunos ao longo do semestre. A funcionalidade de eventos surge como uma estratégia para aumentar o uso contínuo do aplicativo, estendendo seu valor além do período inicial de procura por salas de aula, que tende a ter uma demanda decrescente com o passar do tempo, como será discutido no Capítulo 8. Essa nova funcionalidade também se alinha com a missão do aplicativo de ser uma plataforma centralizadora e informativa para a comunidade da EPUSP.

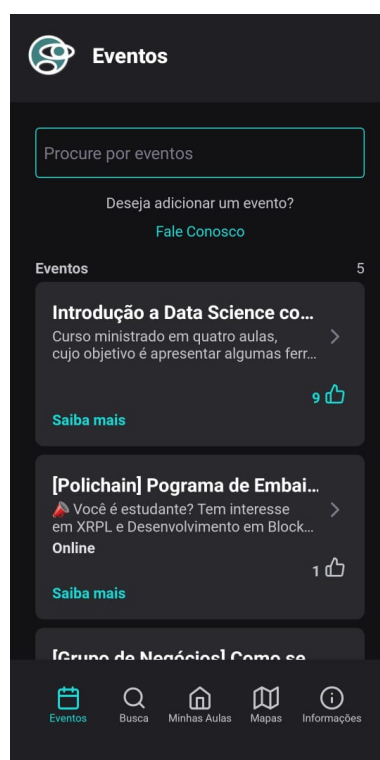
Este épico (E6 do Capítulo 5) surgiu como uma adaptação de uma funcionalidade ideada no Design Sprint, removendo as funcionalidades previstas para disciplinas optativas, e se tornando mais generalista com eventos na faculdade - afinal, as entidades e grupos de extensão podem divulgar seus processos seletivos e demais eventos que organizam, mas outros tipos de eventos organizados na faculdade, como palestras e workshops, também podem ser incluídos. Esta funcionalidade foi escolhida levando também em conta a Resolução CoG, CoCEX e CoPq N<sup>o</sup> 7788, de 26 de agosto de 2019 da USP, que indica que atividades acadêmicas complementares se tornaram parte obrigatória dos cursos de

graduação (UNIVERSIDADE DE SÃO PAULO, 2019). Assim, a aba de eventos serve como um recurso que facilita o acesso dos alunos a eventos e oportunidades de extensão dentro do ambiente universitário.

### 7.4.1 Descrição da Funcionalidade

A nova aba de eventos apresenta-se como um centro de interação e informação. Na interface inicial, presente na Figura 43, é possível encontrar uma lista de eventos com seus respectivos títulos, descrições resumidas e a contagem de *likes*, permitindo uma rápida visualização e avaliação da popularidade de cada evento. Ao selecionar um evento, ilustrado na Figura 44, detalha-se uma nova tela com informações completas, como descrição extensa, categoria e detalhes logísticos como local e horário. Além disso, um link externo está disponível para aqueles que buscam informações adicionais sobre o evento em específico, cenário comum em casos de processos seletivos envolvendo formulários.

Figura 43 – Listagem de eventos no USPolis



Ainda na Figura 43, é possível observar um convite explícito para a colaboração na adição de eventos aos interessados, apresentado por meio de um aviso que direciona o usuário imediatamente para o e-mail do USPolis. Esta funcionalidade proativa não só simplifica o processo de inclusão de novos eventos pelos organizadores e entidades estudantis, mas também fomenta uma cultura de participação ativa na comunidade acadêmica, assegurando que o aplicativo esteja sempre atualizado.

Figura 44 – Detalhes de um evento no USPolis



## 7.4.2 Lançamento da Funcionalidade

O preenchimento inicial da funcionalidade de eventos é importante para estabelecer o aplicativo como um ponto de referência confiável para informações de eventos na universidade. Ao garantir que essa funcionalidade seja lançada com uma variedade de eventos interessantes já cadastrados, tenta-se criar um impulso inicial de engajamento que incentiva os alunos a retornarem ao aplicativo com mais frequência, de forma a conferir quais eventos na EPUSP possam interessá-los.

## 7.5 Considerações Finais do Capítulo

No decorrer deste capítulo, foram descritas em detalhes as entregas do novo USPolis, desde a entrega da versão inicial do aplicativo até as funcionalidades mais recentemente desenvolvidas, passando pelo processo de ideação do Design Sprint que as gerou.

A seguir, serão apresentados os resultados do projeto e análises dos dados colhidos ao longo do ano, culminando nas perspectivas futuras para o USPolis.

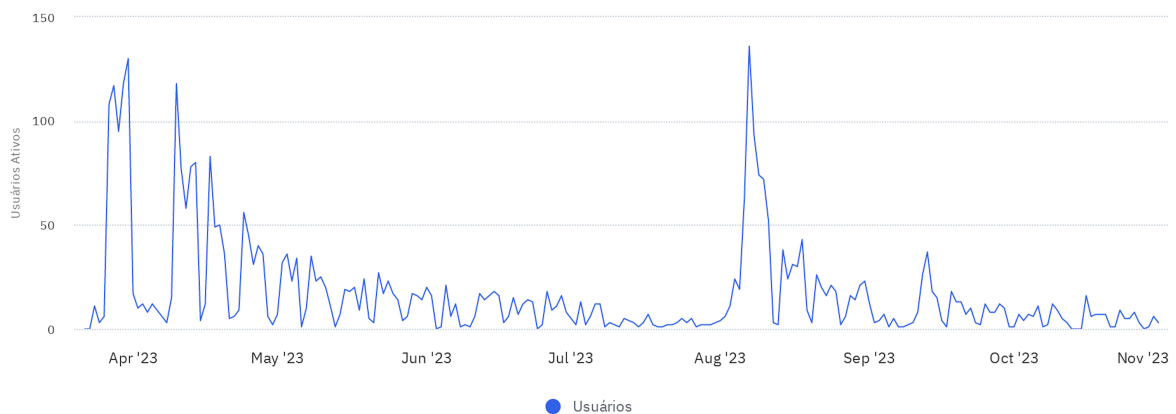
## 8 Resultados

Este capítulo apresenta uma análise dos resultados obtidos com a implementação do USPolis, focando nos dados analíticos coletados pelo aplicativo via Amplitude e nas experiências dos usuários da EPUSP. Serão abordadas as funcionalidades e a aceitação do USPolis em sua versão inicial, a eficácia do teste A/B no buscador de disciplinas, o impacto dos eventos, e então considerações finais sobre a contribuição do USPolis para a experiência dos alunos na faculdade.

### 8.1 Versão inicial do USPolis

A versão inicial do USPolis (contemplando os épicos E1 a E4 do Capítulo 5) foi lançada para a comunidade politécnica ainda antes do início do ano letivo de 2023, reintroduzindo ferramentas para que os alunos pudessem acompanhar suas grades horárias e alocações de salas desde a primeira semana de aula, como faziam com o aplicativo legado. A estratégia de *marketing* adotada, baseada na divulgação em redes sociais, contribuiu significativamente para a popularização inicial do aplicativo, atraindo um número considerável de usuários sem a necessidade de investimentos financeiros em publicidade. Os dados iniciais do USPolis revelaram um padrão interessante de uso, com picos de cerca de 125 usuários ativos por dia no lançamento e início de semestres letivos, estabilizando-se em torno de 20 usuários diários ao longo do semestre. Esta tendência sugere uma aderência inicial significativa, seguida por um uso moderado e consistente, salvo em período de recesso acadêmico. A figura 45 ilustra essa dinâmica de utilização ao longo do tempo.

Figura 45 – Número de usuários ativos no USPolis por dia



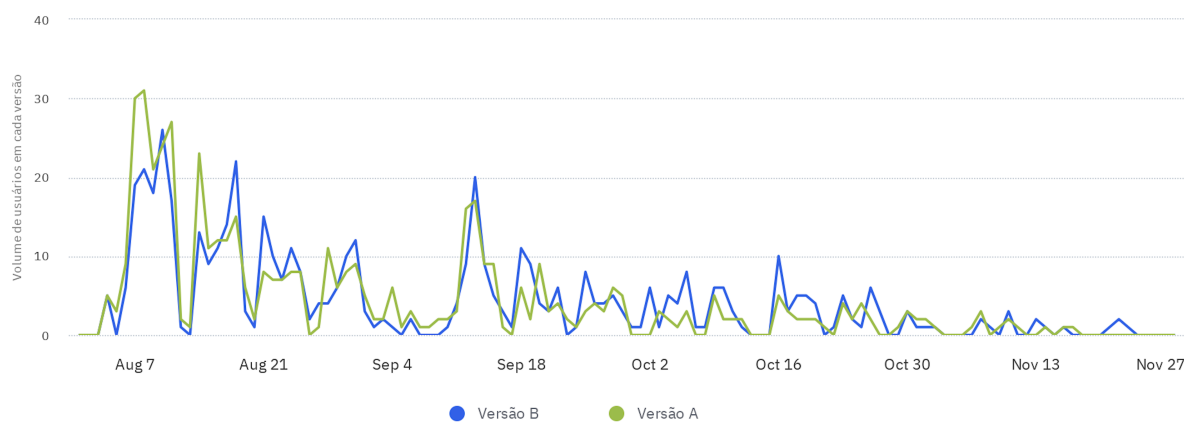
## 8.2 Teste A/B - Buscador de Disciplinas

Como discutido no capítulo 7, a próxima funcionalidade desenvolvida, relativa ao buscador de disciplinas (épico E5 do Capítulo 5), foi submetida a um teste A/B, de forma a investigar qual seria a melhor opção para a sua implementação. A versão A, apresentada na Figura 41, conta com um botão que inicia o fluxo de busca automatizada de disciplinas, enquanto a versão B, que pode ser vista na Figura 42, lista os cursos juntamente às turmas, e cada um dos cursos, se selecionado, inicia o fluxo de busca automatizada de disciplinas.

Inicialmente, na Figura 46, apresenta-se a quantidade de usuários distribuídos entre os grupos de teste A e B. É possível perceber um acompanhamento do volume de ambas as amostras, diminuindo ao longo do tempo à medida que a versão do aplicativo em que o teste A/B está inserido é substituído por uma nova versão ou desinstalado.

Para comparar as duas versões de maneira objetiva, estabeleceu-se como métrica do teste A/B o número de aulas adicionadas na grade de um usuário usando o buscador automatizado. Assim, a versão com mais aulas adicionadas, contabilizando todos os usuários, seria considerada vencedora.

Figura 46 – Usuários ativos em cada grupo de teste



A partir da análise dos dados coletados, identificou-se uma diferença notável entre os dois grupos testados, na Figura 47 e 48, com o Grupo B apresentando resultados significativamente melhores. Este resultado levou a decisão de adotar permanentemente as alterações do Grupo B.

Figura 47 – Resultado do Teste A/B - Aulas adicionadas com o buscador no tempo

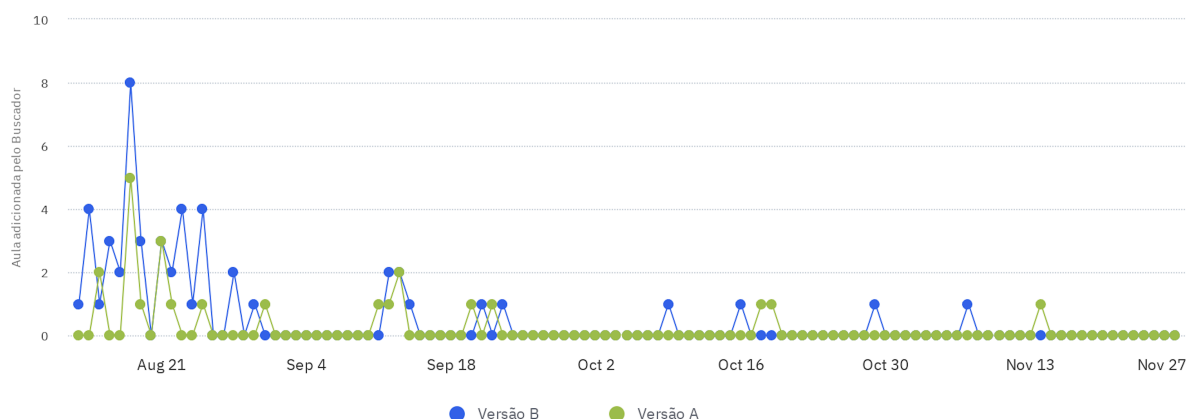
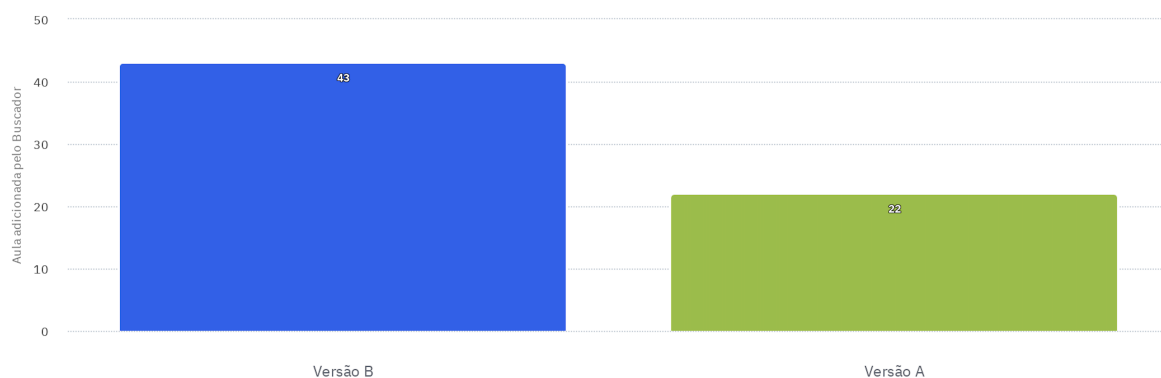


Figura 48 – Resultado do Teste A/B - Agregado de aulas adicionadas



Com isso, a adoção dessas alterações melhorou a experiência do usuário, representado pela Figura 49, em que o número de aulas adicionadas ao cronograma teve um aumento depois da adição da funcionalidade de buscador de disciplinas, o que reforçar a eficácia do processo de desenvolvimento iterativo e baseado em dados. Importante ressaltar que, apesar das análises terem sido realizadas em diferentes épocas do ano, a comparação é feita com base na proporção relativa à base de usuários, o que torna a análise percentual válida para ambos os períodos de utilização da feature. O aumento no número de aulas adicionadas em relação à base de usuários fica mais evidente ao comparar as Figuras 50 e 51. Na primeira figura, é possível observar o volume de 12 aulas adicionadas, com uma base de 199 usuários ativos, totalizando aproximadamente 6% de utilização da feature. Já na figura depois da adição da feature, observa-se um volume de 30 adições em uma base de 254 usuários ativos, totalizando aproximadamente 12% de utilização.



Figura 49 – Volume de eventos ao longo do tempo: aula adicionada e curso selecionado

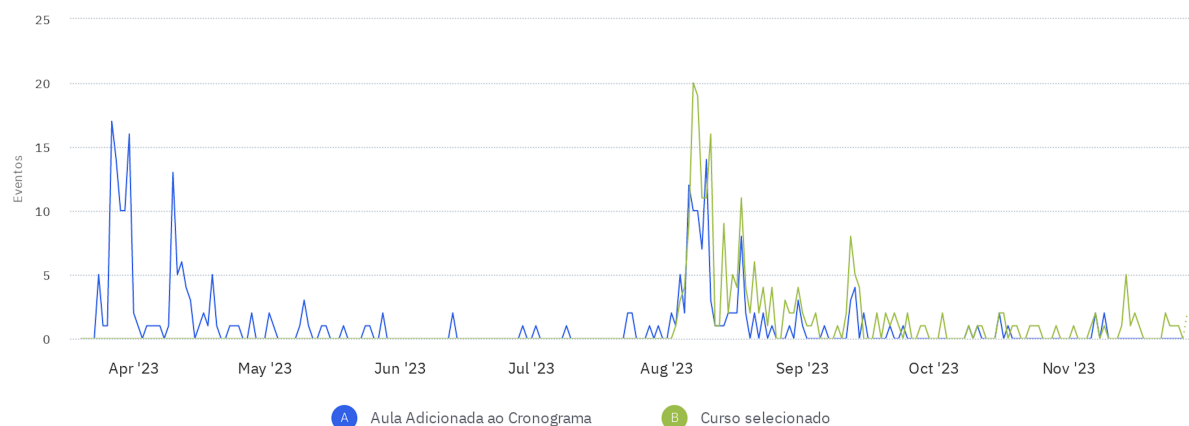


Figura 50 – Aulas adicionadas antes da funcionalidade de buscador de disciplinas (abril a maio)

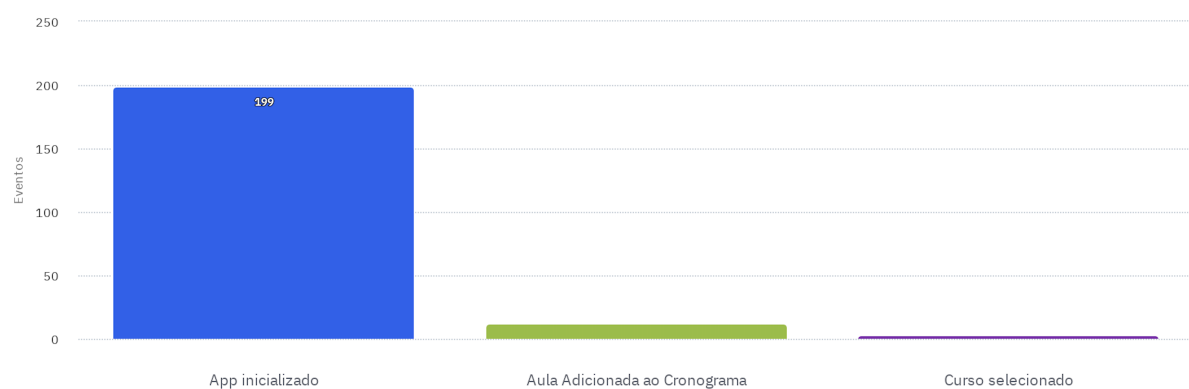
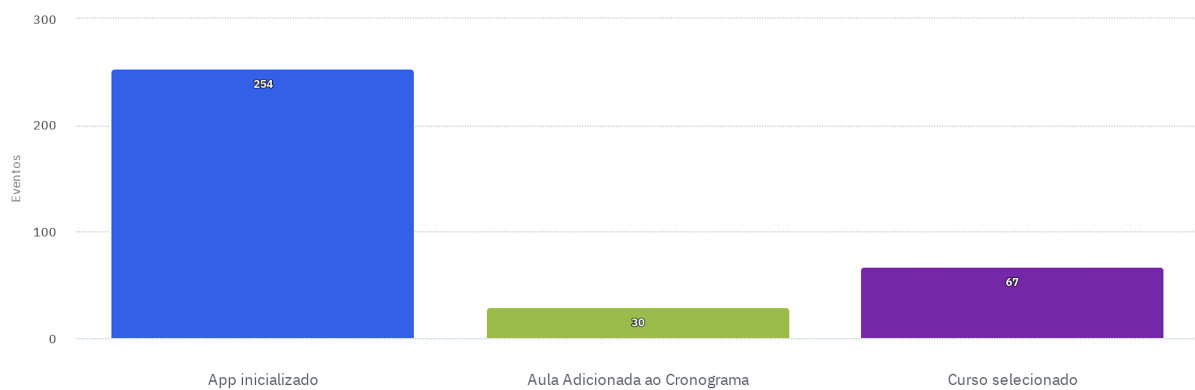


Figura 51 – Aulas adicionadas após a funcionalidade de buscador de disciplinas (agosto a setembro)



## 8.3 Eventos

Após a finalização do Design Sprint e a implementação da funcionalidade de buscador de disciplinas, a equipe do USPolis concentrou-se em consolidar os resultados obtidos e em desenvolver uma nova funcionalidade de eventos, descrita como épico E6 no Capítulo 5.

A concepção da funcionalidade de eventos no aplicativo USPolis originou-se como uma resposta ao desafio de manter o engajamento dos alunos com o aplicativo ao longo do semestre. Os resultados aqui discutidos, como o da Figura 45, evidenciaram que o uso do aplicativo para localizar salas de aula apresentava uma demanda decrescente com o tempo. Esta funcionalidade foi projetada, então, para manter os usuários engajados com o aplicativo mesmo em épocas do semestre em que já não precisam mais consultar seus locais de aulas.

Assim, o lançamento inicial da funcionalidade de eventos focou divulgações orgânicas do aplicativo, com alguns eventos previamente cadastrados pelo grupo, assegurando que a funcionalidade fosse lançada com uma variedade de eventos já cadastrados. O objetivo era criar um engajamento inicial, incentivando os alunos a retornarem ao aplicativo com maior frequência para conferir os eventos disponíveis na EPUSP.

Para analisar o impacto da funcionalidade de eventos, focou-se na taxa diária de retorno dos usuários ao aplicativo, comparando os períodos antes e após o lançamento da funcionalidade. É importante mencionar que, devido às diferenças sazonais entre os períodos analisados, a avaliação concentrou-se em variações relativas, e não em números absolutos, para determinar se houve um aumento no engajamento do aplicativo.

A Figura 52 ilustra a taxa de retorno dos usuários do USPolis antes do lançamento da funcionalidade de eventos, destacando uma tendência inicial de cerca de 20%, declinando progressivamente, com pequenos picos de retorno nos dias 14, 21 e 28. Já a Figura 53 mostra a taxa de retorno depois do lançamento da feature. O decréscimo da taxa de retorno é evidente, indicando que a funcionalidade não alcançou o engajamento esperado. Notou-se também uma redução geral no uso do aplicativo, atribuível à sazonalidade do segundo semestre, um período de menor atividade em comparação ao primeiro semestre.

O aumento esperado na taxa de retorno nos intervalos entre picos de uso não foi observado após a introdução da funcionalidade de eventos.

Figura 52 – Taxa de retorno de usuários - antes da funcionalidade de eventos

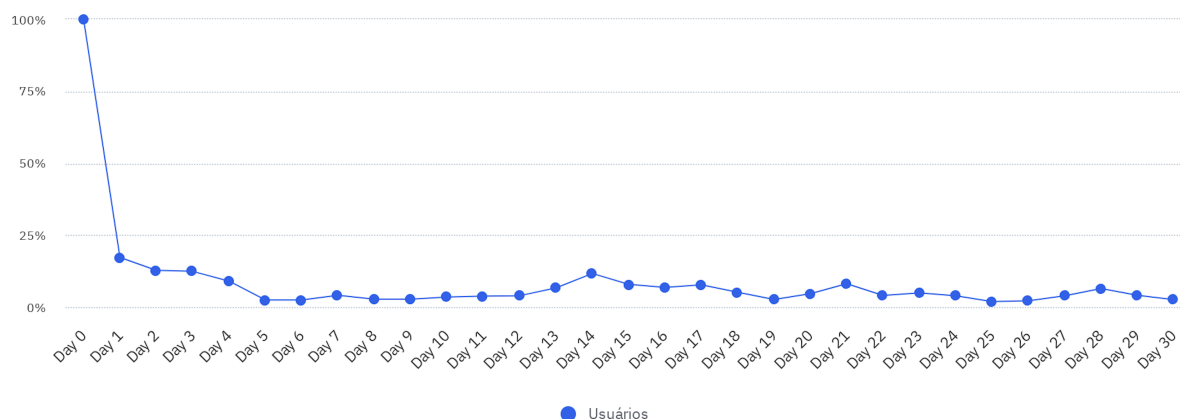
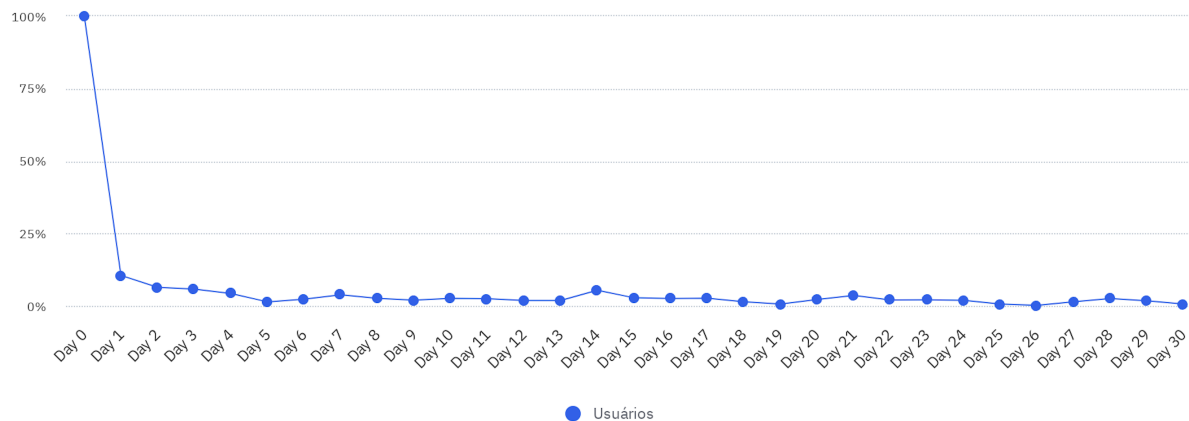


Figura 53 – Taxa de retorno de usuários - depois da funcionalidade de eventos



## 8.4 Outros resultados

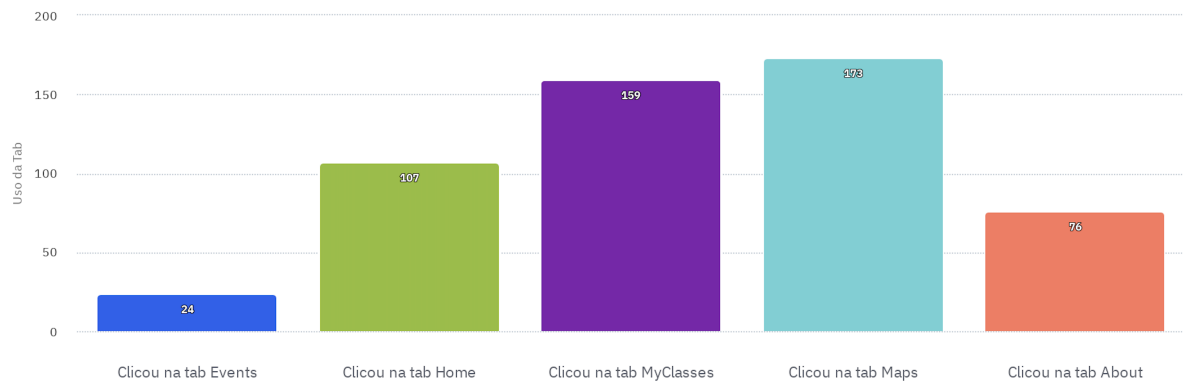
Nesta seção será abordada uma análise mais abrangente dos resultados do USPolis, explorando aspectos variados do aplicativo e do seu uso pela EPUSP. Dados analíticos detalhados sobre a interação dos usuários com diferentes funcionalidades do aplicativo são importantes para compreender o impacto do aplicativo USPolis na universidade e suas áreas de potencial melhoria.

### 8.4.1 Uso das Abas do Aplicativo

A análise do uso das abas do aplicativo USPolis revelou possíveis tendências sobre as preferências e comportamentos dos usuários. A aba de mapas mostrou-se a mais popular, com 173 cliques, indicando que a funcionalidade de mapas é de grande valor para os usuários, provavelmente para a localização de salas de aula e instalações na EPUSP. Segue-se a aba de aulas favoritas, com 160 cliques, refletindo a importância da organização

e planejamento acadêmico para os alunos. A aba inicial, apesar de ser a primeira tela vista ao abrir o aplicativo, registrou 107 cliques, o que pode ser atribuído ao seu acesso automático ao iniciar o aplicativo. A Figura 54 ilustra esse resultado.

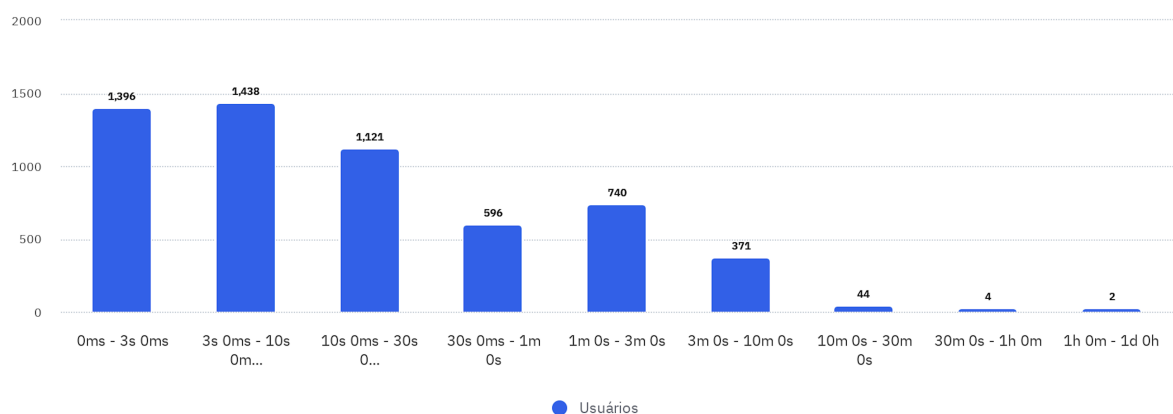
Figura 54 – Uso das abas no USPolis



#### 8.4.2 Distribuição dos Tempos de Sessão

A distribuição dos tempos de sessão dos usuários do USPolis oferece possíveis visões sobre como o aplicativo é utilizado, e está ilustrado na Figura 55. A maioria das sessões, cerca de 1400, durou entre 0 e 3 segundos, sugerindo um uso rápido para consultas pontuais, como verificar mapas ou horários de aulas. As sessões mais longas, entre 3 minutos e 10 minutos, foram menos frequentes, indicando que o aplicativo é usado principalmente para consultas rápidas e específicas.

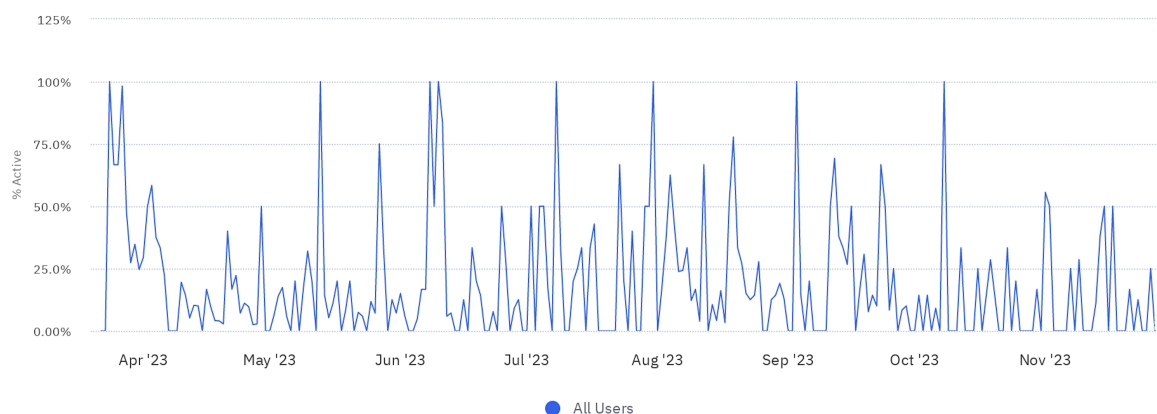
Figura 55 – Distribuição dos tempos de sessão dos usuários



### 8.4.3 Porcentagem de Novos Usuários

A Figura 56 mostra a porcentagens de usuários diários que nunca haviam entrado no app antes. É possível perceber um grande número de dias em que esse número é maior do que 50%. A análise da porcentagem de novos usuários diários do USPolis revelou uma constante aquisição de novos usuários ao longo do semestre. Esta observação sugere uma reciclagem da base de usuários, com novos alunos descobrindo e adotando o aplicativo regularmente.

Figura 56 – Porcentagem de usuários ativos que são novos usuários



### 8.4.4 Instalações

As estatísticas apresentadas na Tabela 5 indicam um total de 918 instalações, para Android e iOS, constituindo as novas instalações dessa versão aprimorada do aplicativo. Este número representa uma presença significativa dentro da comunidade da EPUSP, indicando que uma considerável parcela dos alunos optou por adotar a nova versão do USPolis.

Essa adesão reflete a eficácia das estratégias de divulgação e o valor percebido do aplicativo pelos alunos, e também constrói uma base sólida para futuras atualizações, uma vez que uma parte da comunidade estudantil já esteja familiarizada com o USPolis.

Tabela 5 – Número de downloads por plataforma

Plataforma	Número de Downloads
iOS	271
Android	647

## 8.5 Considerações finais do capítulo

Este capítulo analisou os resultados obtidos com a implementação do USPolis, destacando o impacto que o aplicativo teve na comunidade de alunos da faculdade. A versão inicial, com funcionalidades focadas na localização de salas e gestão de horários, mostrou uma aderência inicial, como evidenciado pelos picos de usuários ativos durante os lançamentos e início dos semestres. Apesar de um declínio natural na utilização do USPolis à medida que o semestre avança, seu padrão de uso revela a relevância do aplicativo USPolis como uma ferramenta disponível aos alunos no início dos períodos letivos.

A execução e análise do teste A/B para o buscador de disciplinas forneceram tendências sobre a preferência dos usuários, resultando na adoção de uma versão que melhorou significativamente a experiência do aplicativo. Entretanto, a funcionalidade de eventos do USPolis, embora inovadora e alinhada à visão de centralização das informações da EPUSP, não atingiu o nível de engajamento esperado entre os alunos. As análises comparativas revelaram uma redução na taxa de retorno dos usuários, mesmo após a introdução dessa nova funcionalidade.

Por fim, a discussão sobre as instalações do aplicativo, a porcentagem de novos usuários diários, distribuição dos tempos de sessão e seu uso geral destacou uma dinâmica interessante em como os alunos da EPUSP utilizam o app USPolis. O número expressivo de downloads e a constante aquisição de novos usuários indicam que o USPolis continuou atraindo novos usuários ao longo do tempo, mesmo sem uma divulgação constante. Este padrão sugere um ciclo saudável de crescimento e renovação, importante para a evolução contínua do aplicativo.

Em resumo, os resultados analisados neste capítulo refletem os desafios enfrentados pelo USPolis na EPUSP. Eles evidenciam a importância de um design centrado no usuário e de uma abordagem iterativa para o desenvolvimento de aplicativos, assegurando que as soluções tecnológicas se alinhem com as necessidades dos usuários, isso é, dos alunos da faculdade.

## 9 Considerações Finais

O projeto USPolis foi desenvolvido com o objetivo de otimizar a experiência educacional na EPUSP, fornecendo aos alunos um aplicativo móvel para localizar salas de aula, gerenciar horários e participar de eventos acadêmicos. O projeto envolveu a modernização de um aplicativo existente e a integração com uma plataforma administrativa de gestão de salas, utilizando metodologias como Design Sprint e testes A/B, além da análise de *feedbacks* dos usuários da plataforma.

A relevância do projeto USPolis está em seu âmbito acadêmico e ligação direta com o dia a dia do estudante. No contexto universitário, ele representa um facilitador para a gestão educacional e organização de salas, e este trabalho incorporou e estendeu esse significado, melhorando a eficiência e a organização dos alunos.

Com isso, os resultados obtidos indicam o sucesso do USPolis. A versão inicial, focada na localização de salas e gestão de horários, demonstrou uma aderência inicial robusta, com picos de usuários ativos nos lançamentos e início dos semestres. O teste A/B da funcionalidade de buscador de disciplinas evidenciou a preferência dos usuários, resultando na adoção de uma versão que aprimorou a experiência do aplicativo. Contudo, a funcionalidade de eventos, embora interessante, não atingiu o engajamento esperado.

É importante destacar que o objetivo principal do Design Sprint de **"Melhorar a experiência do usuário na busca de informações de aulas"** foi efetivamente atingido. A abordagem adotada no Design Sprint, embora tenha desviado da composição original proposta por [Knapp e Kowitz \(2016\)](#), provou ser bem-sucedida, com a participação ativa de membros do grupo de trabalho, usuários ativos do sistema, orientador e co-orientador. Essas implementações estratégicas resultaram em um aplicativo funcional e intuitivo, alinhado com as expectativas da comunidade da EPUSP.

### 9.1 Contribuições

O projeto USPolis procurou contribuir para a comunidade acadêmica da Escola Politécnica ao proporcionar uma ferramenta digital que facilita a organização e o acesso à informação. Sua implementação resultou em um aplicativo funcional e intuitivo, que busca melhorar a eficiência e a experiência educacional dos alunos.

As 918 instalações do aplicativo refletem o alcance dentro da comunidade da Escola Politécnica. Essa importância se reflete no número de 32.710 sessões iniciadas, destacando como o aplicativo se tornou uma ferramenta utilizada pelos alunos. Uma das funcionalidades mais valorizadas, a de mapas, registrou uma alta taxa de uso, sublinhando

sua utilidade na orientação dos alunos pelo campus e na localização das salas de aula e prédios. Além disso, o interesse pelo USPolis se manteve consistente ao longo do ano, como demonstra a porcentagem significativa de novos usuários. Este engajamento contínuo ganhou um novo impulso com a introdução do buscador automático de disciplinas, uma funcionalidade que levou os alunos a dobrarem o número de adições na aba 'Minhas Aulas'.

Com esses resultados, este projeto, portanto, atendeu ao objetivo de melhorar e contribuir para experiência do usuário na busca por informações de aulas, e também estabeleceu um precedente para o uso de metodologias ágeis e *feedback* do usuário no desenvolvimento de aplicativos educacionais.

## 9.2 Perspectivas de Continuidade

Para o futuro, recomenda-se a continuação do aplicativo com foco no aprimoramento da funcionalidade de eventos do USPolis, explorando novas estratégias para aumentar o engajamento e a retenção de usuários. Isso pode incluir a implementação de técnicas interativas mais envolventes ou a integração de elementos de gamificação para atrair e manter a atenção dos usuários.

Além disso, visa-se a expansão do USPolis para além da gestão de horários e localizações físicas, dando continuidade às funcionalidades pensadas durante a realização do Design Sprint, presentes no tópico 7.2.3. A avaliação de disciplinas é uma dessas funcionalidades, em que alunos poderiam avaliar e votar suas disciplinas, permitindo que alunos se antecipem quanto a sua dificuldade e outros atributos. Outra importante funcionalidade seria a adição de disciplinas optativas da Escola Politécnica, possibilitando aos alunos encontrarem disciplinas variadas além daquelas obrigatórias. Por fim, uma comunicação personalizada e assertiva, informando alunos sobre eventos, disciplinas e notícias de acordo com seus interesses permitiria criar um ambiente mais interativo e dinâmico. Essa abordagem promove uma experiência de usuário alinhada às necessidades de cada aluno, contribuindo para um maior engajamento deles com a vida universitária. Ao integrar essas funcionalidades ao USPolis, o aplicativo torna-se uma ferramenta cada vez mais versátil e adaptada aos desafios contemporâneos da educação, aproximando-se de um canal eficaz para a troca de informações e experiências enriquecedoras entre os membros da Escola Politécnica.



## Referências

- BECK, K. *Extreme Programming Explained: Embrace Change*. USA: Addison-Wesley Longman Publishing Co., Inc., 1999. ISBN 0201616416. Citado na página 27.
- CHACON, S.; STRAUB, B. *Pro Git*. [S.l.]: Apress, 2014. Citado na página 25.
- CRISPIN, L.; GREGORY, J. *Agile Testing: A Practical Guide for Testers and Agile Teams*. [S.l.]: Addison-Wesley Professional, 2009. Citado 2 vezes nas páginas 27 e 29.
- DABBISH, L. et al. Social coding in github: transparency and collaboration in an open software repository. In: *ACM Conference on Computer Supported Cooperative Work*. [S.l.: s.n.], 2012. Citado na página 25.
- DUVALL, P. M.; MATYAS, S. M.; GLOVER, A. *Continuous Integration: Improving Software Quality and Reducing Risk*. [S.l.]: Addison-Wesley Professional, 2007. Citado 3 vezes nas páginas 26, 27 e 29.
- FOWLER, M.; FOEMMEL, M. Continuous integration. *ThoughtWorks*, 2006. Citado 3 vezes nas páginas 26, 27 e 29.
- HIGUTI, L. R. A.; KONDO, M. M. *USPOLIS: SISTEMA OPEN SOURCE PARA ALOCAÇÃO E VISUALIZAÇÃO DE SALAS DE AULA*. [S.l.], 2022. Disponível em: <[https://pcs.usp.br/pcspf/wp-content/uploads/sites/8/2022/12/Monografia\\_PCS3560\\_SEM\\_2022\\_Grupo\\_S17.pdf](https://pcs.usp.br/pcspf/wp-content/uploads/sites/8/2022/12/Monografia_PCS3560_SEM_2022_Grupo_S17.pdf)>. Acesso em: 22 out 2022. Citado 6 vezes nas páginas 12, 31, 34, 35, 44 e 46.
- HUMBLE, J.; FARLEY, D. *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*. Addison-Wesley Professional: Addison-Wesley Professional, 2010. Citado 4 vezes nas páginas 24, 25, 28 e 29.
- KIM, G. et al. *The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. Portland, OR: IT Revolution Press, 2016. Citado na página 24.
- KNAPP, J. Z. J.; KOWITZ, B. *Sprint: o método usado no Google para testar e aplicar novas ideias em apenas cinco dias*. 3. ed. San Francisco: Simon Schuster, 2016. Acesso em: 25 jun 2023. Citado 7 vezes nas páginas 14, 15, 18, 19, 40, 64 e 78.
- MATRUSP. *MatrUSP*. 2011. Disponível em: <<https://bccdev.ime.usp.br/matrusp/>>. Acesso em: 12 nov 2023. Citado na página 36.
- MRBS. *Meeting Room Booking System*. 2000. Disponível em: <<https://mrbs.sourceforge.io/>>. Acesso em: 12 nov 2023. Citado na página 36.
- STACK OVERFLOW. *Stack Overflow Developer Survey*. 2022. Disponível em: <<https://insights.stackoverflow.com/survey>>. Acesso em: 12 nov 2023. Citado na página 31.

THOMKE, S. H. *A cultura da experimentação: como os experimentos nos negócios podem melhorar sua capacidade de inovação*. São Paulo: Benvirá, 2021. Citado 4 vezes nas páginas 19, 20, 21 e 22.

TOP UNIVERSITIES. *QS World University Rankings by Subject 2022: Engineering & Technology*. 2022. Disponível em: <<https://www.topuniversities.com/university-subject-rankings/engineering-technology/2022?region=Latin%20America>>. Acesso em: 2 dez 2023. Citado na página 11.

UNIVERSIDADE DE SÃO PAULO. *Resolução CoG, CoCEX e CoPq N° 7788, de 26 de agosto de 2019*. [S.l.], 2019. Disponível em: <<https://leginf.usp.br/?resolucao=resolucao-cog-cocex-e-copq-no-7788-de-26-de-agosto-de-2019>>. Acesso em: 18 nov 2023. Citado na página 67.

VALENTE, M. T. *Engenharia de Software Moderna*. 1. ed. Brazil: Independente, 2022. Acesso em: 25 jun 2023. Citado 2 vezes nas páginas 23 e 28.

# APÊNDICE A – Resultados das entrevistas com os usuários

Para a condução das entrevistas antes do início do Design Sprint, o grupo preparou um pequeno roteiro composto por algumas perguntas e tópicos para guiar a discussão com os entrevistados:

- Você conseguiu usar o aplicativo? Conseguiu adicionar suas aulas, encontrar sua sala?
- Você teve alguma dificuldade geral no uso dessas funcionalidades ? Não conseguiu encontrar sua turma, não encontrou facilmente algum botão, etc?
- Quão frequentemente você entra no aplicativo?
- Além de encontrar suas salas de aula, existem outras dificuldades encontradas ao longos dos semestres na Poli que poderiam ser facilitadas e centralizadas no USPolis?

Cada uma das entrevistas realizadas está sumarizada em seus pontos mais importantes a seguir.

## A.1 Usuário 1 - Engenharia Elétrica, 4º ano

**Você conseguiu usar o aplicativo? Conseguiu adicionar suas aulas, encontrar sua sala?**

Consegui usar o aplicativo e encontrar minhas aulas, mas tive algumas dificuldades no caminho. O fluxo de adicionar disciplinas foi muito fácil, gostei bastante.

**Você teve alguma dificuldade geral no uso dessas funcionalidades ? Não conseguiu encontrar sua turma, não encontrou facilmente algum botão, etc?**

Sim, algumas:

- Os botões de navegação dentro do aplicativo estão bem com tempo de respostas bem devagar, parecem travados.
- Subir e descer os andares nos mapas não foi intuitivo.
- A seleção de prédio para ver os mapas podia ser um seletor, não a barra com rolagem.

- Procurar as disciplinas não foi fácil, parece não estar correspondendo corretamente ou não tratando corretamente os textos para busca.

### **Quão frequentemente você entra no aplicativo?**

Só entrei no começo do semestre, quando eu não sabia onde eram as aulas. Depois, eu acabei decorando com o tempo e não entrei mais.

**Além de encontrar suas salas de aula, existem outras dificuldades encontradas ao longos dos semestres na Poli que poderiam ser facilitadas e centralizadas no USPolis?**

Não.

### **Outros comentários**

- Seria legal ver assinalado nos mapas o local onde é cada sala de aula das minhas turmas.
- A experiência com o MatrUSP é boa para montar a grade, e poderia servir de inspiração! Você seleciona seu curso e ano, e ele já adiciona essas disciplinas no seu horário.

## **A.2 Usuário 2 - Engenharia da Computação, 4º ano**

**Você conseguiu usar o aplicativo? Conseguiu adicionar suas aulas, encontrar sua sala?**

Sim, consegui. Achei o aplicativo bonito, com um visual muito melhor do que eu esperava.

**Você teve alguma dificuldade geral no uso dessas funcionalidades ? Não conseguiu encontrar sua turma, não encontrou facilmente algum botão, etc?**

Alguns pontos:

- A navegação está um pouco devagar.
- Quando entrei no aplicativo, naveguei entre as abas, e não ficou claro se o USPolis ia puxar minhas disciplinas automaticamente.
- A barra de pesquisa também está demorada.

### **Quão frequentemente você entra no aplicativo?**

Entre duas ou três vezes no começo do ano, para checar pontualmente onde seria a aula de alguma disciplina da minha grade.

**Além de encontrar suas salas de aula, existem outras dificuldades encontradas ao longo dos semestres na Poli que poderiam ser facilitadas e centralizadas no USPolis?**

- Vocês poderiam expandir a funcionalidade dos mapas, criando localizações relevantes dentro da Poli além das salas de aula. Assim, seria possível explicitar onde ficam coisas importantes, como grupos de extensão, onde é o local de pegar o cartão USP no começo da graduação, esse tipo de coisa.
- Num ecossistema de informações importantes para os politécnicos, seria importante ter os cardápios dos restaurantes universitários.
- Seria legal saber em tempo real onde estão os ônibus circulares dentro da USP.
- Seria útil ter uma página ou seção com telefones e contatos de secretarias ou departamentos na Poli.

#### **Outros comentários**

- Como faço parte de um grupo de extensão, sempre temos muita dificuldade para conseguir reservar uma sala dentro da Poli para alguma reunião. Se vocês quiserem atacar as dificuldades dos grupos de extensão, seria um bom ponto de partida.
- Seria ótimo se o USPolis tivesse também uma função de exportar minha grade para ferramentas de calendário pessoal. Eu, por exemplo, organizo meus horários com o Google Calendar.

### **A.3 Usuário 3 - Engenharia Elétrica, 1º ano**

**Você conseguiu usar o aplicativo? Conseguiu adicionar suas aulas, encontrar sua sala?**

A minha experiência com o USPolis foi bem tranquila. O aplicativo roda bem no meu celular e tem todas as informações que precisei.

**Você teve alguma dificuldade geral no uso dessas funcionalidades ? Não conseguiu encontrar sua turma, não encontrou facilmente algum botão, etc?**

Meus professores de computação e cálculo trocaram a sala de aula no meio do semestre, então o aplicativo ficou desatualizado. Seria legal poder trocar isso manualmente, só pra mim mesmo no meu celular.

**Quão frequentemente você entra no aplicativo?**

Usei o uspolis mais nas primeiras semanas mesmo, pra saber onde eram minhas aulas. Mas a verdade é que eu ainda uso sempre que vou em algum prédio que não conheço muito bem, porque geralmente não sei onde ficam as salas. Os mapas são muito bons, para mim fez uma diferença enorme ter eles no aplicativo.

**Além de encontrar suas salas de aula, existem outras dificuldades encontradas ao longos dos semestres na Poli que poderiam ser facilitadas e centralizadas no USPolis?**

Nada que me vem à cabeça agora.

#### **Outros comentários**

Seria legal ter um mapa da faculdade, ou mesmo da USP, que indica onde são os prédios de cada faculdade e cada engenharia. A Poli é muito grande, e eu acho que não sei onde ficam todos os prédios.

### **A.4 Usuário 4 - Engenharia de Produção, 5º ano**

**Você conseguiu usar o aplicativo direitinho? Achou fácil adicionar suas aulas e achar as salas?**

Sim, consegui usar, mas senti que poderia ser mais rápido. Adicionar as aulas e achar as salas funcionou, mas às vezes era meio lento.

**Teve alguma coisa no app que te deu trabalho para usar?**

A navegação nos mapas podia ser mais fácil. Também notei que algumas vezes as informações das aulas não batiam, o que me deixou confuso algumas vezes.

**Você usa o app com que frequência?**

Mais no começo do semestre. Depois que eu decorei onde são as aulas, usei menos.

**Que melhorias você sugeriria para o app?**

Seria legal se o app mostrasse informações mais personalizadas sobre as aulas e ajudasse os grupos de extensão de alguma forma. Também acho que daria para incluir informações úteis do campus, tipo o cardápio do bandeirão.

#### **Outros comentários**

Melhorar a navegação no app, especialmente nos mapas, e mostrar locais importantes no campus, como bibliotecas e áreas de lazer, seria muito útil.

## A.5 Usuário 5 - Engenharia Naval, 4º ano

**Conseguiu usar o app sem problemas? Foi fácil adicionar as aulas e encontrar as salas?**

Usei sim, mas a parte de adicionar as disciplinas podia ser mais simples. E a troca entre os andares nos mapas às vezes é meio confusa.

**Você encontrou dificuldades no uso de alguma funcionalidade?**

A navegação entre os mapas podia ser mais clara. E seria bom se o app avisasse quando eu não adicionasse nenhuma aula.

**Com que frequência você entra no app?**

No começo do semestre eu uso mais. Se tivesse informações atualizadas das disciplinas, eu usaria mais.

**O que mais você acha que poderia ser adicionado ou melhorado no app?**

Os mapas podiam ter uma cara mais moderna, talvez até com realidade aumentada. Informações sobre as filas dos bandejões ou o cardápio do dia seriam super úteis. Expandir para outros institutos da USP também seria legal, mostrando matérias optativas também.

**Outros comentários**

A experiência de uso geral do app pode melhorar. Um design mais intuitivo e respostas rápidas aos comandos fariam uma grande diferença. E ter acesso mais fácil a informações e locais importantes do campus ajudaria bastante.