

**FERNANDO UCHÔA DO RÊGO BARROS FILHO
GUSTAVO BANDEIRA BARROS
VICTOR HUGO CHIMENEZ QUEIROZ**

**TEREMIN: APLICATIVO DE GESTÃO DE
GRUPOS MUSICAIS**

São Paulo
2023

**FERNANDO UCHÔA DO RÊGO BARROS FILHO
GUSTAVO BANDEIRA BARROS
VICTOR HUGO CHIMENEZ QUEIROZ**

**TEREMIN: APLICATIVO DE GESTÃO DE
GRUPOS MUSICAIS**

Trabalho apresentado à Escola Politécnica
da Universidade de São Paulo para ob-
tenção do Título de Engenheiro Eletricista.

São Paulo
2023

**FERNANDO UCHÔA DO RÊGO BARROS FILHO
GUSTAVO BANDEIRA BARROS
VICTOR HUGO CHIMENEZ QUEIROZ**

**TEREMIN: APLICATIVO DE GESTÃO DE
GRUPOS MUSICAIS**

Trabalho apresentado à Escola Politécnica
da Universidade de São Paulo para ob-
tenção do Título de Engenheiro Eletricista.

Área de Concentração:
Engenharia da Computação

Orientador:
Prof. Dr. Fábio Levy Siqueira e
Prof. Dr. Bruno Augusto Angélico

São Paulo
2023

AGRADECIMENTOS

Ao Professor Doutor Fábio Levy Siqueira, nosso orientador, expressamos nossa profunda gratidão por sua orientação e apoio ao longo de todo o projeto. Sua orientação não apenas nos guiou metodologicamente, mas também enriqueceu nosso crescimento científico e intelectual de maneira significativa.

Ao Professor Doutor Bruno Augusto Angélico, coorientador deste projeto, expressamos nossa sincera gratidão pela atenção e apoio contínuo durante o processo de definição metodológica e execução dos trabalhos relacionados. Sua contribuição foi crucial para a qualidade e eficácia do nosso trabalho.

À Professora Doutora Adriana Holtz Betiol e ao colega Dyogo Toyoshi de Oliveira Mei, manifestamos nosso apreço pela generosidade em dedicar tempo e esforço para nos ajudar a compreender o problema abordado neste projeto. Sua colaboração foi inestimável e enriqueceu consideravelmente nossa abordagem ao desafio em questão.

Expressamos também nossa profunda gratidão a nossas famílias, cujo apoio e suporte foram fundamentais em todas as questões externas à faculdade. Graças a esse apoio, pudemos focar com dedicação nos estudos e no desenvolvimento deste projeto. Agradecemos por serem fontes constantes de estímulo e suporte ao longo desta jornada acadêmica.

RESUMO

O presente trabalho tem como objetivo o desenvolvimento de um aplicativo de gestão de bandas, visando criar um produto que atenda de forma eficaz e eficiente às necessidades dos usuários. Identificou-se que a organização de bandas demanda um tempo considerável e processos manuais muito concentrados que não havia uma solução eficaz que resolvessem os desafios. Diante disso, foi proposto a criação de um aplicativo centrado no usuário, que visa otimizar a gestão de bandas, resultando em economia de tempo. Ao fim, foi possível entregar um aplicativo ao público-alvo que, através de avaliação de usabilidade, mostrou resultados positivos.

Palavras-Chave – Aplicativo, Ágil, Lean, Gestão de Bandas, Usabilidade

ABSTRACT

This work aims to develop a band management application, seeking to create a product that effectively and efficiently meets the users' needs. It was identified that band organization requires a considerable amount of time and relies on highly concentrated manual processes for which there was no effective solution to address the challenges. In response, the proposal was the creation of a user-centric application designed to optimize band management, resulting in time savings. In the end, an application was successfully delivered to the target audience, demonstrating positive results through usability evaluation.

Keywords – Application, Agile, Lean, Band Management, Usability.

LISTA DE FIGURAS

1	Estrutura de usabilidade - Norma ISO 9241-11	14
2	Avaliação de usabilidade - Norma ISO 9241 - 210	15
3	Exemplo de Agenda: Lean Inception	19
4	Visão do Produto	32
5	Persona 1: "Paulo", o líder da banda	34
6	Persona 2: "Gustavo", o músico da banda	34
7	Sequenciador de atividades	38
8	Canvas MVP	39
9	Camada de Contexto do Modelo C4	45
10	Camada de Contêineres do Modelo C4	46
11	Camada de Componentes do Modelo C4	47
12	Tela de início e entrada	49
13	Tela de cadastro de usuário	50
14	Tela de cadastro de evento	51
15	Tela de criação de escala, seleção do músico e escolha da repetição	52
16	Tela de confirmação do evento	53
17	Tela de escolha da música, cifras e vídeo	54
18	Eficiência Usuário	59
19	Satisfação Média	60
20	Tabela de Problemas	61
21	Questionário de Satisfação SUS – System Usability Scale	67

LISTA DE TABELAS

1	Comparativo entre LouveApp e Teremin	25
2	Questionário SUS - Fonte: Brookes, 1996	27
3	Organização das Sprints	30
4	Roteiro de Entrevistas	31
5	Jornada 1: Escalas Mensais e/ou Recorrentes	35
6	Jornada 2: Seleção e Disponibilização de Músicas	35
7	Tarefas do Teste de Usabilidade	56
8	Teste de Usabilidade: Eficácia	59
9	Brainstorming Jornada 1: Escalas Mensais e/ou Recorrentes	64
10	Brainstorming Jornada 2: Seleção e disponibilização de músicas	65
11	Brainstorming Funcionalidades Estruturais	66

SUMÁRIO

1	Introdução	10
1.1	Objetivo	10
1.2	Justificativa	11
1.3	Organização do Trabalho	12
2	Aspectos Conceituais	13
2.1	Design centrado no Usuário	13
2.1.1	Norma ISO 9241	14
2.2	Lean Inception	16
2.2.1	Visão do produto	16
2.2.2	É - não é - Faz - não faz	17
2.2.3	Personas	17
2.2.4	Funcionalidades	17
2.2.5	Revisão técnica	18
2.2.6	Jornadas do usuário	18
2.2.7	Sequenciador de Funcionalidades	18
2.2.8	Canvas MVP	18
2.2.9	Agenda do Lean Inception	19
2.3	Histórias de Usuário	20
2.4	Modelo C4	21
2.4.1	Contexto	21
2.4.2	Contêineres	21
2.4.3	Componentes	22
2.4.4	Código	22

3	Trabalhos Relacionados	24
3.1	Aplicativo LouveApp	24
3.2	Avaliações de Usabilidade	25
4	Metodologia	29
4.1	Sprints	29
4.2	Lean Inception	30
4.2.1	Visão do Produto, É - não é - Faz - não faz e Product Goals	32
4.2.2	Personas e Jornadas	33
4.2.3	Feature Brainstorming e Technical, Business and UX Review	36
4.2.4	Canvas MVP	38
5	Especificação de Requisitos	40
5.1	Usuários	40
5.2	Histórias do usuário	40
5.3	Considerações Finais	43
6	Arquitetura do Sistema	44
6.1	Arquitetura de Monolito Modular	44
6.2	Primeira camada da arquitetura: Contexto	45
6.3	Segunda camada da arquitetura: Contêineres	45
6.4	Terceira camada da arquitetura: Componentes	47
6.5	Considerações	48
7	Funcionalidades do Aplicativo	49
7.1	Cadastro de usuários	50
7.2	Cadastro de eventos	51
7.2.1	Escala de Músicos	51
7.3	Escolha de músicas	52

7.3.1	Cifras	52
7.3.2	Vídeo	53
8	Resultados Obtidos	55
8.1	Lançamento do Aplicativo	55
8.2	Método de Avaliação de Usabilidade	55
8.3	Resultados de Usabilidade	58
8.3.1	Eficácia	58
8.3.2	Eficiência	58
8.3.3	Satisfação	60
8.4	Conclusão	61
8.5	Alcance de Metas	62
9	Considerações Finais	63
10	Apêndice	64
10.1	Brainstorming de funcionalidades	64
11	Referências	68

1 INTRODUÇÃO

No contexto da sociedade cada vez mais digitalizada, soluções inovadoras são essenciais para atender às demandas contemporâneas. O processo de gestão de bandas envolve diversas atividades intrincadas, incluindo, mas não se limitando a, agendamento de ensaios, coordenação de eventos, gerenciamento da disponibilidade dos músicos, seleção e organização de repertório, e comunicação interna dentro da banda.

Tradicionalmente, essas tarefas são realizadas manualmente, resultando em ineficiências, possíveis erros e esforços demorados. Em resposta a esse cenário, a proposta de desenvolver um aplicativo de gestão de bandas visa preencher uma lacuna significativa, proporcionando uma solução ágil e eficiente para os desafios enfrentados por músicos e gestores de bandas.

O aplicativo, denominado "Teremin", presta homenagem aos músicos e à Engenharia Elétrica, sendo um dos primeiros instrumentos musicais totalmente eletrônicos, controlado sem qualquer contato físico pelo músico. O projeto busca inspiração em metodologias Lean e ágil para criar um produto que não apenas simplifica, mas também aprimora a gestão de bandas.

A motivação para este trabalho reside na aplicação de inovação, eficiência e na criação de valor tangível para a indústria musical. Além disso, alinhar-se às necessidades e expectativas dos usuários por meio de práticas de engenharia é um foco central. O projeto também visa contribuir para aprimorar a usabilidade e eficácia, aplicando padrões reconhecidos, como a ISO 9241.

1.1 Objetivo

Este projeto de formatura tem como objetivo o desenvolvimento de um sistema digital para otimizar processos de gestão de grupos musicais. Tais grupos possuem diversas questões e dificuldades no que diz respeito à organização dos membros do grupo, bem

como à organização dos recursos do mesmo. Dessa forma, surgiu a ideia de desenvolver um sistema intuitivo e de fácil utilização para facilitar o dia a dia dos membros do grupo.

O sistema foi desenvolvido na forma de um aplicativo para dispositivos celulares, o qual será disponibilizado nas lojas de aplicativos comuns presentes nestes aparelhos. Com isso, o acesso será facilitado aos usuários finais e também será um projeto público, ou seja, após a finalização do mesmo qualquer pessoa que tiver interesse em fazer uso do aplicativo poderá usufruir dele, mesmo não fazendo parte do grupo inicialmente utilizado como alvo deste trabalho de conclusão de curso.

1.2 Justificativa

A gestão manual de bandas apresenta uma série de desafios e dificuldades que impactam negativamente a eficiência e a organização das atividades. Muitos músicos e gestores de bandas enfrentam um processo trabalhoso e demorado ao lidar com tarefas como a escolha de músicos para apresentações, definição de repertório e organização de escalas. A natureza manual dessas atividades frequentemente resulta em processos confusos e desorganizados, levando a uma gestão ineficaz e consumindo um tempo considerável.

A falta de uma solução digital específica para a gestão de bandas cria uma lacuna, deixando os usuários dependentes de métodos manuais e desatualizados. A necessidade de simplificar esses processos, tornando-os mais ágeis e acessíveis, é evidente diante das limitações enfrentadas por aqueles envolvidos na administração de grupos musicais.

Ao analisar outros aplicativos de gestão de bandas, como o LouveApp, é possível notar a ausência de funcionalidades como visualização de cifras e o vídeo da música diretamente pelo aplicativo. Além disso, não há um aplicativo flexível para diferentes contextos de bandas que seja útil para desde bandas pequenas até maior.

A aplicação de normas reconhecidas, como a ISO 9241, na busca por melhorias na usabilidade, destaca a preocupação em criar uma ferramenta que não apenas simplifica as tarefas, mas também promove uma experiência eficaz e intuitiva. Em última análise, a justificativa para este projeto reside na necessidade de inovação, eficiência e na superação das dificuldades enfrentadas pelos músicos e gestores de bandas, proporcionando-lhes uma ferramenta digital que otimize suas atividades cotidianas e contribua para o desenvolvimento da indústria musical como um todo.

1.3 Organização do Trabalho

Este trabalho é organizado da seguinte forma: no Capítulo 2 são apresentados aspectos conceituais utilizados para desenvolvimento desse projeto; no Capítulo 3 são discutidos trabalhos relacionados; no Capítulo 4 é descrita a metodologia utilizada neste trabalho; no Capítulo 5 é apresentada a especificação de requisitos; no Capítulo 6 é detalhada a arquitetura do sistema; no Capítulo 7 são detalhadas as funcionalidades do aplicativo; e no Capítulo 8, os resultados obtidos, como lançamento e resultados de usabilidade.

2 ASPECTOS CONCEITUAIS

Neste capítulo, serão abordados os conceitos principais utilizados para a elaboração deste trabalho, desde os conceitos aplicados na metodologia para levantamento de requisitos do software escolhido até os conceitos utilizados para o desenvolvimento da arquitetura do sistema em questão.

Quatro abordagens principais foram utilizadas para este trabalho: Norma ISO 9241, Lean Inception, Histórias de Usuário e Modelo C4, cada uma responsável por cumprir um propósito diferente dentro do escopo do projeto.

2.1 Design centrado no Usuário

O conceito de design centrado no Usuário (DCU) é uma abordagem fundamental na criação e no aprimoramento de produtos, especialmente no campo do design de interfaces de usuário e desenvolvimento de aplicativos. Jakob Nielsen, Alan Shackel e Donald Norman são pesquisadores e profissionais que contribuíram significativamente para a evolução desse conceito, oferecendo perspectivas e diretrizes para sua aplicação.

Jakob Nielsen propõe a usabilidade como um aspecto crucial do design de sistemas interativos, abordando o processo de desenvolvimento centrado no usuário, a avaliação de usabilidade e a melhoria contínua da qualidade de uso.

Alan Shackel desenvolveu a ideia de "ergonomia perceptiva", que se concentra nas habilidades cognitivas e perceptivas dos usuários ao interagirem com um sistema. Ele ressalta a importância de considerar as capacidades e limitações humanas ao projetar interfaces e sistemas interativos, a fim de otimizar a experiência do usuário.

A aplicação desses conceitos no desenvolvimento de aplicativos é de extrema importância e relevância. Ao adotar uma abordagem centrada no usuário, os designers e desenvolvedores são capazes de compreender melhor as necessidades, expectativas e comportamentos dos usuários.

2.1.1 Norma ISO 9241

A Norma ISO 9241-11, intitulada "Ergonomia de interfaces homem-sistema - Parte 11: Orientações sobre usabilidade", é uma norma internacional desenvolvida pela International Organization for Standardization (ISO). Essa norma faz parte da série de normas ISO 9241, que aborda os aspectos ergonômicos da interação entre humanos e sistemas.

A ISO 9241-11 fornece diretrizes e recomendações sobre usabilidade, focando na forma como os sistemas interativos devem ser projetados para atender às necessidades e expectativas dos usuários. O objetivo principal da norma é garantir que os produtos, especialmente interfaces de usuário, sejam fáceis de usar e eficazes, resultando em uma experiência de usuário positiva.

O resultado do aplicativo desenvolvido vai se ancorar na Norma ISO 9241/11 que propõe que, para se metrificar a usabilidade, precisamos identificar os objetivos e, a partir deles, extrair eficácia, eficiência e satisfação através de atributos mensuráveis. Portanto, é através dessa análise de usabilidade que vamos avaliar se o app desenvolvido correspondeu ao objetivo determinado focado no usuário.

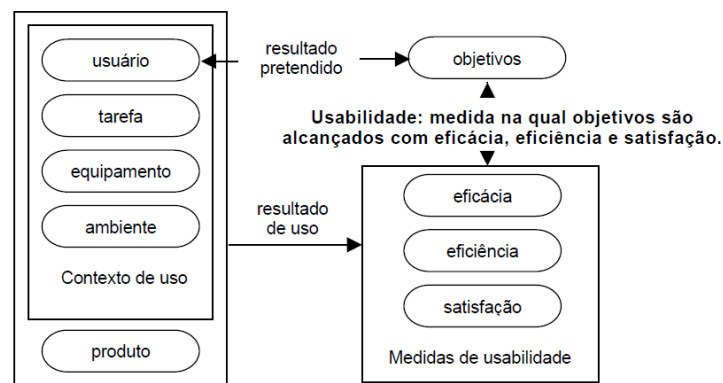


Figura 1: Estrutura de usabilidade - Norma ISO 9241-11

A Norma ISO 9241-210 tem como objetivo fornecer diretrizes e recomendações para o processo de projeto e desenvolvimento de sistemas interativos. Ela enfatiza a importância de adotar uma abordagem sistemática e centrada no usuário para garantir a qualidade da interação entre os usuários e os sistemas.

Sendo assim, para que alcancemos o objetivo de entregar um aplicativo centrado no usuário, através da avaliação que a parte 11 da norma propõe, utilizaremos como base teórica a lógica proposta pela parte 210:

De acordo com a norma, o processo para avaliação de usabilidade é feito de acordo com o processo abaixo:

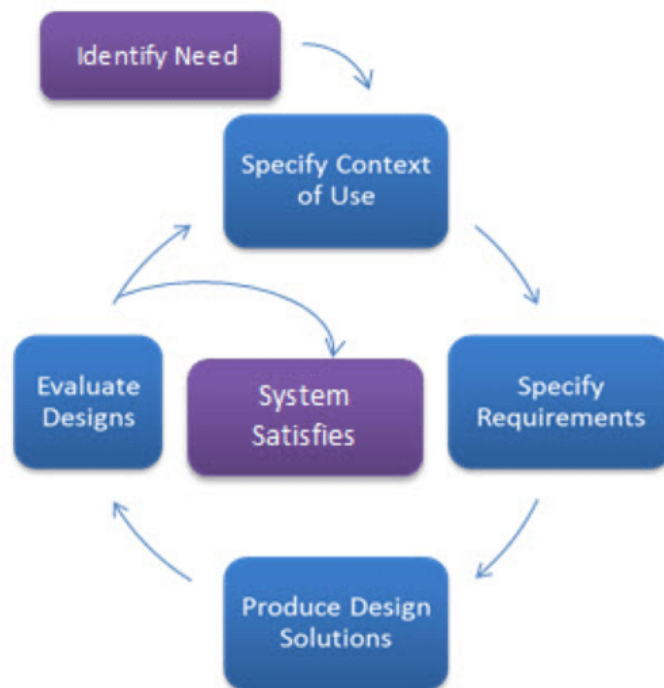


Figura 2: Avaliação de usabilidade - Norma ISO 9241 - 210

1. Compreensão do contexto de uso: A norma destaca a importância de compreender o contexto em que o sistema será utilizado. Isso envolve a identificação dos usuários-alvo, suas necessidades, tarefas e ambiente de trabalho. Para tal, são utilizadas pesquisas e entrevistas com usuários para entendimento. Essa compreensão é crucial para projetar um sistema que seja adequado ao seu contexto de uso específico.
2. Definição de requisitos de usabilidade: A norma enfatiza a necessidade de estabelecer requisitos claros de usabilidade para o sistema. Isso envolve identificar os critérios de desempenho desejados, como eficiência, eficácia, facilidade de aprendizado e satisfação do usuário. A definição adequada desses requisitos é essencial para orientar o processo de design e avaliação do sistema.
3. Iteração e prototipação: A norma recomenda a utilização de iterações e prototipação durante o processo de design. Isso permite que os designers obtenham *feedback* dos usuários em estágios iniciais do desenvolvimento e façam ajustes iterativos com base nesse *feedback*. A prototipação também ajuda a validar as soluções propostas antes da implementação final do sistema.
4. Avaliação da usabilidade: A norma destaca a importância da avaliação contínua da usabilidade durante o processo de desenvolvimento. Isso envolve a realização de testes de usabilidade, avaliações especializadas e outras técnicas de avaliação para identificar problemas de usabilidade e oportunidades de melhoria. A avaliação da usabilidade ao longo

do processo permite que os problemas sejam corrigidos de forma mais eficaz e econômica.

2.2 Lean Inception

A metodologia conhecida como Lean Inception, desenvolvida por Paulo Carioli, baseia-se nos princípios do Lean Startup e Design Thinking. Seu objetivo é auxiliar equipes de desenvolvimento na criação de produtos inovadores e com valor agregado para os clientes.

A Lean Inception é uma abordagem eficaz para iniciar o desenvolvimento de um aplicativo de gestão de bandas, focando na definição enxuta e alinhamento de objetivos estratégicos. Comparada a metodologias ágeis como Scrum e Kanban, a Lean Inception se destaca por sua ênfase na fase inicial do projeto, promovendo colaboração, economia de recursos e flexibilidade. Isso ajuda a evitar retrabalho, manter o foco no que é essencial e garantir a participação ativa do time como um todo, tornando-a uma escolha sólida para alcançar o sucesso no desenvolvimento de aplicativos por estudantes em trabalho de conclusão de curso.

A Lean Inception é realizada em um período curto, geralmente entre uma e duas semanas, e envolve atividades colaborativas, como entrevistas com *stakeholders*, definição de personas, elaboração de histórias de usuário e criação de um produto mínimo viável. O resultado é uma tabela que compila o escopo do projeto, para ser alinhada com o time o, possibilitando reduzir o risco de retrabalho e garantindo que o produto final atenda às necessidades dos clientes.

A metodologia busca, por meio de dinâmicas de aplicação, levantar as características do produto, identificando problemas a serem resolvidos, fluxos necessários e funcionalidades. Além disso, é possível definir um MVP a ser implementado para validar a ideia proposta.

A seguir, serão detalhadas as etapas propostas pela metodologia Lean Inception que são definidas pelo autor.

2.2.1 Visão do produto

Na definição da visão do produto, o objetivo é estabelecer uma visão clara e compartilhada entre a equipe em relação ao produto a ser desenvolvido. Nessa etapa, cada membro da equipe descreve sua visão sobre diversos aspectos do projeto, como a finalidade do produto, o público-alvo, as principais características e como ele se diferencia de

soluções existentes no mercado.

Ao final dessa etapa, espera-se ter uma visão consolidada e concisa do produto, com a consolidação das ideias apresentadas pelos membros da equipe.

2.2.2 É - não é - Faz - não faz

Nesta etapa, os membros do grupo devem listar, de acordo com sua visão, o que o produto é, as características e funcionalidades que devem estar presentes. Além disso, é importante definir o que o produto não é, ou seja, o que ele não resolverá e quais características e funcionalidades não devem ser incluídas. Também é necessário estabelecer o que o produto deve fazer, ou seja, quais ações e funcionalidades devem ser contempladas. Por fim, é preciso definir o que o produto não está previsto para fazer.

Ao final dessa etapa, espera-se ter uma visão organizada e clara do foco do produto a ser desenvolvido.

2.2.3 Personas

Nesta etapa, a equipe define as personas, representações fictícias de usuários que irão interagir com o produto. Essa definição é fundamental, pois influencia decisões de design e interfaces, levando em consideração as características e necessidades do usuário. A definição das personas também ajuda a equipe a compreender as dores e expectativas dos usuários, oferecendo uma visão mais precisa das expectativas em relação ao produto.

2.2.4 Funcionalidades

Nesta etapa, cada membro da equipe elabora um conjunto de funcionalidades que considera relevante para o produto, com base nas discussões anteriores sobre personas e visão do produto.

Ao final dessa etapa, espera-se ter um conjunto significativo de funcionalidades que podem ser implementadas no MVP. Isso é importante para gerar discussões posteriores sobre a priorização de cada funcionalidade.

2.2.5 Revisão técnica

Nesta etapa, é realizada uma revisão das funcionalidades propostas anteriormente, considerando a complexidade técnica percebida por cada membro da equipe. Em seguida, as funcionalidades são classificadas de acordo com a complexidade e o valor para o cliente. Isso facilita a identificação das funcionalidades prioritárias e aquelas que requerem uma análise mais detalhada em termos técnicos, evitando atrasos no desenvolvimento devido a dificuldades técnicas não previstas.

2.2.6 Jornadas do usuário

Nessa fase, são representadas as etapas que um usuário potencial pode percorrer ao interagir com o sistema, incluindo todas as interações relacionadas às funcionalidades descritas anteriormente.

A jornada do usuário é uma ferramenta para compreender as necessidades e expectativas dos usuários, além de identificar pontos de melhoria nos processos existentes no produto. Isso possibilita melhorar gradualmente a experiência do usuário no aplicativo.

2.2.7 Sequenciador de Funcionalidades

O sequenciador de funcionalidades é utilizado para definir a ordem de desenvolvimento das funcionalidades. Ele consiste em uma página com várias linhas, em que cada linha representa um conjunto de três funcionalidades. Dessa forma, é possível visualizar quais funcionalidades estarão presentes no MVP, além de estabelecer fases para o desenvolvimento e incremento posterior do produto após a implementação do conjunto inicial de funcionalidades.

2.2.8 Canvas MVP

Por fim, após a conclusão de todas as etapas anteriores, é gerado um canvas MVP, que guiará a equipe no desenvolvimento do produto final. Esse canvas resume todas as discussões anteriores, fornecendo uma visão das personas, funcionalidades, jornadas, custo do projeto, cronograma, métricas de avaliação das hipóteses de negócio e resultados esperados do projeto.

Dessa forma, a metodologia Lean Inception é aplicada integralmente, e o desenvolvimento do produto segue um caminho mais facilitado a partir desse ponto.

2.2.9 Agenda do Lean Inception

O livro traz como exemplo a agenda mostrada na Figura 1 como sugestão, mas detalha que algumas atividades da metodologia podem ser executadas em ordem diferente a depender do quanto se está já preparado para realizar a etapa seguinte.

Como ponto importante o autor ressalta a necessidade de começar a aplicação do Lean Inception com a definição da visão do produto, para que todos estejam na mesma página ao longo das sessões. Além disso, na sequência, o autor ressalta a necessidade de deixarem claro o que o produto faz e não faz, é e não é para que não haja ideias distoantes do escopo do produto ao longo das sessões.

Em seguida, é sugerida a descrição das personas para que garantir que todas as soluções pensadas sejam centradas no usuário. A partir disso, o autor sugere a ideação de funcionalidades a partir da visão do produto, público-alvo e do escopo do produto.

Como próximo passo, é então sugerido priorizar as funcionalidades levantadas com base nos critérios que a metodologia sugere: técnico, negócio e usuário. Após essa etapa, é importante a construção das jornadas dos usuários para que se realize o sequenciador de funcionalidades.

Por fim, como produto final, recomenda-se que seja desenvolvido o Canvas MVP e apresentado, no último dia de dinâmica, aos envolvidos e interessados com o produto.



Figura 3: Exemplo de Agenda: Lean Inception

2.3 Histórias de Usuário

As histórias do usuário representam um formato para documentar elementos de uma lista de tarefas, sendo de fácil compreensão tanto para indivíduos ligados ao negócio quanto para aqueles com conhecimento técnico. Este conceito é comumente resumido como as "3C's- cartão, conversa e confirmação (Rubin, 2012).

Normalmente, segue-se um modelo para a criação de um "cartão", aderindo à estrutura "Como [perfil do usuário], eu gostaria de [objetivo] para que [benefício]". Com isso, a finalidade primordial do cartão consiste em servir como um marcador para debates mais profundos entre as partes envolvidas. É importante ressaltar que não é esperado que se haja um detalhamento completo de todas as informações no "cartão", é necessário que tenha os pontos cruciais a serem entendidos pela equipe de desenvolvimento.

São definidos três níveis de história do usuário (Rubin, 2012). O nível mais abrangente e menos minucioso conhecida como "épico". O nível com um nível um pouco maior de detalhamento é denominado "feature". Por fim, as histórias e *features* podem ser alocadas em *sprints*, que correspondem a ciclos de trabalho com planejamento detalhado das atividades.

As histórias do usuário variam em termos de profundidade, e podem ir sendo complementadas. As narrativas mais abrangentes e menos minuciosas, conhecidas como "épicos," fornecem uma perspectiva global dos requisitos e têm uma duração de vários meses. Já as narrativas um pouco mais detalhadas, chamadas "funcionalidades," têm uma duração de semanas. Por último, as narrativas com um grau mais elevado de detalhe, com uma duração de alguns dias, podem ser atribuídas a iterações e, durante o planejamento, são subdivididas em tarefas, com o foco mudando de "o que construir" para "como construir". (Rubin, 2012)

A fim de garantir que as narrativas estejam bem concebidas, Mike Cohn propôs o critério INVEST (Independentes, Negociáveis, Valiosas, Estimáveis, Pequenas/Tamanho Adequado, Testáveis):

- **Independentes:** As histórias devem ser autônomas umas em relação às outras ou, no mínimo, ter poucas dependências, facilitando a estimativa, a priorização e o planejamento.

- **Negociáveis:** Os detalhes das narrativas devem ser sujeitos a negociação, permitindo que o Proprietário do Produto defina o porquê da necessidade de uma narrativa, em vez de determinar o seu modo de desenvolvimento, criando espaço para inovação.

- **Valiosas:** As narrativas devem acrescentar valor ao cliente, que adquire o produto, ou aos utilizadores do produto. Narrativas de cariz técnico podem ser explicadas ao Proprietário do Produto, que tomará a decisão sobre a sua inclusão na lista de tarefas.

- **Estimáveis:** A equipa de desenvolvimento deve ser capaz de estimar as narrativas, sendo esta estimativa fundamental para a priorização e o refinamento.

- **Pequenas/Tamanho Adequado:** As histórias devem ter um tamanho apropriado, adequado ao período de desenvolvimento planeado. Para as iterações, devem ser pequenas, embora narrativas maiores (épicas) também possam ser contempladas.

- **Testáveis:** As histórias devem ser testáveis de forma binária, ou seja, podem ser consideradas aprovadas ou reprovadas, o que compõe as condições de satisfação (Wake, 2003).

2.4 Modelo C4

A arquitetura de software desempenha um papel fundamental no desenvolvimento de sistemas complexos, fornecendo uma estrutura para organizar e relacionar os componentes do sistema. O Modelo C4 (Brown, 2011), criado por Simon Brown, é uma abordagem para a criação de diagramas de arquitetura de software que visa fornecer uma visão clara e concisa da arquitetura de um sistema. Ele fornece uma estrutura simples e eficaz para descrever a arquitetura em vários níveis de abstração, desde uma visão de alto nível até detalhes mais técnicos. O Modelo C4 é composto por quatro camadas principais:

2.4.1 Contexto

A camada de contexto é a visão de mais alto nível do sistema, identificando os atores externos e outros sistemas envolvidos. Nessa etapa, é possível entender os objetivos do sistema e definir os limites do seu escopo e identificar as dependências externas. A camada de contexto é uma ferramenta útil para representar essa visão geral e comunicar a arquitetura de uma maneira clara e concisa para todas as partes interessadas, mesmo que não sejam parte do time de engenharia.

2.4.2 Contêineres

A camada de contêineres descreve os principais blocos de construção do sistema, chamados de contêineres. Um contêiner é uma unidade executável, como um servidor

de aplicativos, um banco de dados, um aplicativo de desktop ou até mesmo um website. Esses contêineres são agrupados com base em suas responsabilidades funcionais e podem ser físicos (como um servidor) ou lógicos (como um serviço). O diagrama de contêineres é usado para ilustrar como esses contêineres se relacionam entre si e com o ambiente externo, facilitando a análise de escalabilidade, distribuição e balanceamento de carga.

Esta camada já fornece um nível de especificação mais detalhado e aprofundado, sendo assim a camada de contêineres ajuda na comunicação eficaz com a equipe técnica, permitindo que os desenvolvedores compreendam melhor a estrutura do sistema e colaborem de forma mais eficiente.

2.4.3 Componentes

Na camada de componentes, o interior dos contêineres é explorado para garantir a identificação dos componentes principais que desempenham um papel importante na arquitetura do sistema. Um componente pode ser uma biblioteca, um serviço, um módulo ou uma classe, dependendo do nível de granularidade necessário. Nesta camada, é possível mostrar como os componentes se encaixam nos contêineres e como se relacionam. O diagrama de componentes é utilizado para visualizar essas dependências e interfaces, proporcionando uma compreensão mais detalhada da estrutura interna do contêiner.

A camada de componentes é particularmente valiosa para os desenvolvedores, pois fornece uma visão mais detalhada dos elementos de código que compõem o sistema. Isso ajuda a analisar a modularidade, o acoplamento e a coesão dos componentes, facilitando a manutenção, extensão e reutilização do código.

2.4.4 Código

A camada de código é a mais detalhada do modelo, onde exemplos concretos de implementação são fornecidos para ilustrar como os componentes são realmente implementados. Isso pode incluir diagramas de código, trechos de código ou referências a repositórios de código-fonte. Essa camada ajuda a entender a estrutura interna dos componentes e como o código é organizado dentro deles. Também pode fornecer uma visão mais detalhada das dependências entre as classes ou dos módulos que compõem o componente.

Nesta camada é possível utilizar vários modelos e técnicas para representar a estrutura e a implementação do código em um nível mais detalhado: o Modelo UML (Unified Modeling Language) (Booch, 2005) é comumente usado para representar a camada de código

em diagramas de componentes mais detalhados. O UML é uma linguagem padrão amplamente adotada para modelagem visual de sistemas de software e oferece uma variedade de diagramas para representar diferentes aspectos da arquitetura.

O UML oferece uma variedade de diagramas, cada um deles enfocando um aspecto específico do sistema, como estrutura, comportamento, interações e processos. Alguns dos diagramas UML mais comuns incluem:

1. Diagrama de Casos de Uso: Representa os atores externos e os casos de uso do sistema, ajudando a identificar os requisitos do sistema e as interações entre os atores e o sistema.

2. Diagrama de Classes: Descreve a estrutura de classes no sistema, incluindo atributos, métodos e relacionamentos entre classes.

3. Diagrama de Sequência: Mostra a interação entre objetos ou componentes ao longo do tempo, destacando a ordem das mensagens trocadas entre eles.

4. Diagrama de Atividades: Modela o fluxo de controle do sistema, mostrando como atividades individuais estão relacionadas e como o controle é transferido entre elas.

5. Diagrama de Componentes: Representa os componentes físicos ou lógicos do sistema e suas dependências.

6. Diagrama de Estados: Modela o comportamento de um objeto ou entidade ao longo de diferentes estados e as transições entre eles.

7. Diagrama de Implantação: Descreve como os componentes de software são implantados em hardware ou ambientes físicos.

3 TRABALHOS RELACIONADOS

Neste capítulo serão abordados sistemas semelhantes ao proposto neste trabalho que já estão em uso no mercado, bem como trabalhos relacionados que também utilizaram conceitos que foram aplicados neste projeto. Serão explorados o sistema LouveApp, aplicativo para gestão de bandas e também alguns trabalhos acadêmicos acerca de avaliações de usabilidade.

3.1 Aplicativo LouveApp

O aplicativo LouveApp tem a finalidade de facilitar a organização de grupos musicais religiosos. Suas funcionalidades vão desde criação de eventos, separação de repertório para um determinado evento e controle de escala, isto é, os músicos e as funções que estes irão desempenhar em uma apresentação.

As funcionalidades presentes neste sistema são muito semelhantes às que foram pensadas para o sistema deste trabalho, visto que ambos aplicativos possuem o mesmo objetivo, mas visando públicos diferentes. O público do LouveApp é voltado a equipes musicais religiosas, como de igrejas. Assim, toda a comunicação do sistema com o usuário utiliza expressões que não são comuns ao público em geral, o que pode dificultar o uso de quem não está inserido no contexto que o aplicativo foi idealizado. A proposta deste trabalho é servir a bandas de modo geral e não apenas de um nicho específico como no caso do LouveApp.

Dentre outras, o LouveApp apresenta as seguintes funcionalidades principais:

1. Cadastro de usuário em mais de uma banda
2. Controle de eventos
3. Controle de escalas
4. Controle de repertório

5. Busca de referências das músicas a partir do nome e autor

A Tabela 1 abaixo apresenta um resumo das funcionalidades presentes no LouveApp e no sistema desenvolvido para este trabalho, chamado Teremin. A Tabela 1 serve também como um breve comparativo entre esses dois sistemas.

	LouveApp	Teremin
Autenticação de usuário	Sim	Sim
Controle de eventos	Sim	Sim
Controle de escalas	Sim	Sim
Controle de repertório	Sim	Sim
Busca de referências das músicas	Sim	Sim
Cifras e vídeos direto no aplicativo	Não	Sim

Tabela 1: Comparativo entre LouveApp e Teremin

Nota-se que ambos os sistemas apresentam quase as mesmas funcionalidades. Contudo, durante as entrevistas realizadas com possíveis usuários, notamos que poder visualizar cifras e o vídeo da música diretamente pelo aplicativo seria um diferencial relevante para o sistema que está sendo desenvolvido. Além disso, como citado anteriormente, o fato de o público alvo do Teremin ser mais abrangente também justifica o desenvolvimento de um novo sistema, apesar de o conjunto de funcionalidades serem semelhantes.

3.2 Avaliações de Usabilidade

A avaliação de usabilidade de sistemas iterativos é explorada na área de Desenvolvimento de Produtos de Engenharia, através de estudos de caso, assim como esse trabalho.

A partir da literatura, um método baseado nos estudos de Jakob Nielsen para avaliar usabilidade levando em conta 5 métricas foi proposto: eficácia, eficiência, satisfação, eficiência relativa e período produtivo (Betiol, 2004).

- **Eficácia:** Refere-se à capacidade de alcançar os resultados desejados ou atingir os objetivos propostos. É a medida de quão bem uma ação, processo ou solução atende às necessidades e metas estabelecidas.
- **Eficiência:** Diz respeito à utilização adequada e otimizada dos recursos disponíveis para realizar uma tarefa ou alcançar um objetivo. Está relacionada à minimização de desperdícios, maximização da produtividade e obtenção do melhor resultado com o menor esforço ou custo.

- Satisfação: Refere-se ao grau de contentamento, felicidade ou satisfação experimentado pelos usuários ou clientes em relação a um produto, serviço ou experiência. Está relacionada ao atendimento das expectativas, necessidades e desejos do usuário, gerando uma sensação de satisfação e bem-estar.
- Eficiência relativa: É uma medida comparativa que avalia a eficiência de diferentes processos, soluções ou alternativas em relação umas às outras. Ela compara o desempenho ou a utilização de recursos entre diferentes opções para determinar qual delas é mais eficiente em termos de resultado obtido versus recursos investidos.
- Período produtivo: Refere-se ao intervalo de tempo em que uma pessoa ou uma equipe é capaz de realizar atividades de forma produtiva e eficaz. É o período em que há um alto nível de concentração, energia e capacidade de executar tarefas com eficiência, gerando resultados significativos. O período produtivo pode variar de pessoa para pessoa e estar relacionado a fatores como ritmo circadiano, motivação, ambiente de trabalho e gestão do tempo.

Além disso, também é proposto, naquele mesmo trabalho, o Questionário de Satisfação SUS – System Usability Scale como referência para medir satisfação. O SUS fornece uma abordagem quantitativa para medir a percepção do usuário em relação à usabilidade de um sistema. Este questionário é composto por 10 itens, cada um avaliado em uma escala de Likert de cinco pontos, variando de "Discordo Totalmente" a "Concordo Totalmente". A seguir, será detalhado cada um dos itens presentes no questionário (Brooke, 1996):

Além disso, foi sugerida uma categorização dos problemas mapeados de usabilidade (Jeffries, 1991).

- Consistência: este critério avalia se o problema identificado revela um aspecto da interface que está em conflito com outras partes do sistema. A consistência é fundamental para garantir que a interface ofereça uma experiência coesa e previsível ao usuário.
- Recorrência: este critério explora se o problema interfere na interação apenas na primeira vez que ocorre ou se é uma questão recorrente. Isso ajuda a determinar se o problema é uma ocorrência isolada ou se pode ser um obstáculo persistente para os usuários.
- Generalidade: aqui, o foco está em entender se o problema identificado aponta para uma falha geral que afeta várias partes da interface ou se é específico de uma única

Número	Afirmação	Descrição
1	Eu acho que gostaria de usar esse sistema com frequência.	Este item avalia a predisposição do usuário em utilizar o sistema de forma regular. Busca compreender a atratividade percebida do sistema para o usuário.
2	Eu acho o sistema desnecessariamente complexo.	Aqui, busca-se entender a percepção do usuário sobre a complexidade do sistema. Indica se o usuário considera que o sistema é mais complexo do que o necessário.
3	Eu achei o sistema fácil de usar.	Este item avalia a facilidade percebida pelo usuário ao interagir com o sistema. Procura compreender se o sistema é intuitivo e simples de ser utilizado.
4	Eu acho que precisaria de ajuda de uma pessoa com conhecimentos técnicos para usar o sistema.	Busca-se entender se o usuário percebe a necessidade de assistência técnica para utilizar o sistema. Indica a dependência de conhecimentos específicos.
5	Eu acho que as várias funções do sistema estão muito bem integradas.	Este item avalia a percepção do usuário sobre a integração das diversas funcionalidades do sistema. Indica se o usuário percebe uma boa coesão entre as diferentes partes do sistema.
6	Eu acho que o sistema apresenta muita inconsistência.	Busca-se compreender se o usuário percebe inconsistências na interface ou no funcionamento do sistema. Indica a uniformidade na experiência do usuário.
7	Eu imagino que as pessoas aprenderão como usar esse sistema rapidamente.	Este item visa avaliar a percepção do usuário sobre a curva de aprendizado do sistema. Indica se o usuário acredita que outros aprenderiam a utilizar o sistema rapidamente.
8	Eu achei o sistema atrapalhado de usar.	Busca-se entender se o usuário percebeu algum desconforto ou dificuldade durante o uso do sistema. Indica a experiência geral de utilização.
9	Eu me senti confiante ao usar o sistema.	Avalia o nível de confiança do usuário durante a interação com o sistema. Indica se o usuário se sente seguro em suas ações.
10	Eu precisei aprender várias coisas novas antes de conseguir usar o sistema.	Este item busca compreender a quantidade de novos conhecimentos que o usuário teve que adquirir para operar o sistema. Indica a necessidade de aprendizado prévio.

Tabela 2: Questionário SUS - Fonte: Brookes, 1996

área. Isso é crucial para determinar a extensão do impacto do problema no sistema como um todo.

Frente a abordagem realizada no trabalho citado, este projeto adotou o Questionário SUS como método de avaliação de satisfação. Além disso, foram adotadas as definições de eficácia e eficiência como medidas de usabilidade.

Outro estudo importante para este trabalho foi a revisão sistemática dos métodos de avaliação de usabilidade de aplicações *mobile* feita por Nugroho (Nugroho, 2002).

De acordo com a revisão sistemática de 22 artigos, o estudo revelou que cerca de 3/4 dos métodos empregados são testes de usabilidade e cerca de 1/4 avaliações de heurísticas. Além disso, os autores constataram que os testes de usabilidade são conduzidos em etapas, envolvem participantes e se concentram em efetividade, eficiência e satisfação do usuário. Somado a isso, o estudo traz o uso da Escala *Likert* nos métodos que envolvem questionários.

Portanto, conforme ocorre com frequência na literatura, este trabalho adotará como método de avaliação de usabilidade o teste de usabilidade. Ademais, como citado anteriormente, o questionário SUS será utilizado para extração de satisfação, através do uso de Escala *Likert* como proposto pela revisão sistemática.

4 METODOLOGIA

Este capítulo explica e detalha as metodologias utilizadas nesse projeto, tanto na estratégia, quanto na execução.

4.1 Sprints

A metodologia de trabalho foi baseada em *sprints* (Sutherland, 2014), isto é, ciclos de trabalho com escopo e planejamento definido. Cada *sprint* teve2 duração de 30 dias.

Cada *sprint*, era iniciada com uma *Sprint Planning* em que eram definidas as histórias de usuários que seriam desenvolvidas na *sprint*. A partir disso, uma *Sprint Goal*, ou seja, um objetivo palpável era escolhido pelo time para demonstrar o sucesso do ciclo de trabalho.

Nessa reunião, também eram definidas as tarefas necessárias para entregar as *features* da história com os envolvidos. Ao longo das *sprints*, o time se reunia semanalmente para discutir os avanços da *sprint* em relação ao planejado. Por fim, cada *sprint* finalizava com uma *Sprint Review*, em que eram compartilhadas as entregas desenvolvidas ao longo do ciclo trabalho.

Na "Tabela 2", é mostrada uma tabela com as *sprints* que ocorreram e as *sprints goals* correspondentes. No total, foram 8 ciclos de trabalho de 30 dias para produção do MVP.

As *sprints* 2, 3, 4 e 7 foram dedidacas a desenvolvimento das funcionalidades do aplicativo - descritas no capítulo 7 "Funcionalidades do Aplicativo". Enquanto as *sprints* 1, 5 e 8 foram dedicadas a dinâmicas e organização de lançamentos.

A Dinâmica *Lean Inception* será melhor descrita na seção 4.2 deste capítulo e os lançamentos do aplicativo na seção 8.1. Ambas as atividades demandaram uma série de ações necessárias para se chegar no produto final e no aplicativo disponível, por isso demandaram *sprints* do projeto para tal

Sprint	Sprint Goal	Detalhamento do trabalho
1	Dinâmica Lean Inception	Realizar a dinâmica Lean Inception para definir a visão do produto e priorizar as funcionalidades.
2	Desenho da Arquitetura e Protótipo do MVP	Criar o desenho da arquitetura do aplicativo no modelo C4 e desenvolver um protótipo do Minimum Viable Product (MVP).
3	Implementar Arquitetura Backend	Iniciar a implementação da arquitetura do backend, alocando os principais componentes e estrutura do servidor.
4	Épico: Escala de Músicos	Desenvolver a funcionalidade de criação de escalas de músicos, permitindo a criação de eventos.
5	Lançamento 1	Realizar o primeiro lançamento do aplicativo com as funcionalidades iniciais.
6	Avaliação de Usabilidade	Coletar feedback dos usuários e realizar testes de usabilidade para identificar melhorias.
7	Épico: Escolha de Repertório	Implementar a funcionalidade de seleção e disponibilização de músicas no repertório da banda com cifras e vídeos das músicas.
8	Lançamento 2	Realizar o segundo lançamento do aplicativo com as funcionalidades adicionais.

Tabela 3: Organização das Sprints

4.2 Lean Inception

Como descrito na seção 2.2 *Lean Inception* dos aspectos conceituais, a metodologia ágil para construção do produto utilizada foi a *Lean Inception*.

Para a aplicação da metodologia proposta, foi essencial compreender o contexto de uso do usuário-alvo, suas dores e necessidades. Para isso, foram realizadas entrevistas com membros e líderes de bandas, que compõem o público-alvo do aplicativo.

Durante as entrevistas, foi elaborado um roteiro de perguntas estruturado, com o objetivo de identificar as necessidades principais do público-alvo e compreender o contexto em que eles atuam. O roteiro de entrevistas incluiu as seguintes seções detalhadas abaixo na tabela 4 :

As entrevistas foram realizadas com um total de 8 pessoas em grupos de três pessoas,

Seção	Perguntas e Tópicos
1	Perfil <ul style="list-style-type: none"> - Nome - Idade - Grau de escolaridade
2	Aquecimento <ul style="list-style-type: none"> - Envolvimento com música - Importância da música - Como a música se encaixa em suas vidas atualmente - Participação em shows - Funcionamento de suas bandas
3	Jornada <ul style="list-style-type: none"> - Processo de ingresso na banda - Aprendizado necessário ao entrar na banda - Organização diária da banda - Rotina em dias de eventos - Como são informados sobre eventos - Processo de seleção de músicas para eventos - Desafios na comunicação entre membros da banda e na coordenação de eventos - Organização de ensaios - Preocupações diárias relacionadas à banda - Lidar com a falta de disponibilidade de músicos para ensaios e apresentações - Dificuldades enfrentadas por novos membros da banda - Expectativas em relação a um aplicativo de gestão de bandas - Sugestões específicas para o desenvolvimento do aplicativo

Tabela 4: Roteiro de Entrevistas

compostos pelo entrevistado, o entrevistador e um anotador para registrar os relatos.

Após a coleta de dados das entrevistas, foram identificadas as dores dos líderes e músicos de bandas. Abaixo, é mostrada a lista de necessidades mapeadas nas entrevistas. Ao lado de cada demanda, é adicionada em quantas entrevistas ela apareceu:

1. Escolha de músicas automatizadas e ranqueadas (7/8): os líderes e músicos necessitam de um sistema que simplifique a seleção de músicas, oferecendo sugestões e classificações baseadas em temas ou preferências, visto que essa tarefa consome muito tempo.

2. Elaboração de escalas mensais (8/8): existe a necessidade de criar escalas mensais para definir as responsabilidades de cada músico em ensaios e apresentações, devido à complexidade e ao número de pessoas envolvidas.

3. Upload de músicas/cifras para o aplicativo (6/8): Os músicos desejam a capacidade de fazer upload de suas próprias músicas ou cifras no aplicativo, facilitando o acesso e o compartilhamento entre os membros da banda, já que o armazenamento atual é manual e centralizado em um único líder.

4. Facilitar o compartilhamento de cifras (6/8): Os músicos necessitam de uma maneira mais eficiente para compartilhar cifras, incluindo a capacidade de compilar todas as músicas em um único arquivo, evitando a abordagem manual atual, que consome tempo e é suscetível a erros.

Com base nas entrevistas, as informações coletadas foram organizadas para iniciar a aplicação da Lean Inception, a fim de estabelecer a visão do produto de acordo com a metodologia descrita na seção 2.2 *Lean Inception*. Na próxima seção, será apresentada a implementação da metodologia no projeto.

4.2.1 Visão do Produto, **É - não é - Faz - não faz** e **Product Goals**

O primeiro passo foi definir a visão do produto, alinhada com as necessidades do usuário-alvo, uma vez que se trata de um produto centrado no usuário.

Para músicos e gestores de bandas, cuja organização dos membros e recursos do time toma muito tempo e é trabalhosa, o Teremin é um app que facilita o dia a dia de organização das bandas, diferentemente da forma descentralizada e manual, nosso produto torna a organização mais eficiente gerando economia de tempo com a gestão da banda.

Figura 4: Visão do Produto

Após a definição da visão do produto, a metodologia propõe que seja definido o que o produto é e o que ele faz:

1. Produto é: app para grupo de bandas
2. Produto não é: website, app para anunciar trabalhos de músicos, app de assessoria de bandas, rede social de músicos, streaming de músicas, organizador de tarefas
3. Produto faz: organiza calendários de eventos, facilita a montagem de escalas de músicos, simplifica a organização de bandas, apresentações e repertórios
4. Produto não faz: contrata eventos, vende eventos, comunica-se com pessoas além da banda.

Em seguida, as atividades da Lean Inception continuaram com a definição das metas do produto em conjunto:

1. 1. Publicar aplicativo na Play Store até fim de Outubro/2023
2. 2. Atingir pelo menos 3 bandas diferentes usando o App semanalmente

4.2.2 Personas e Jornadas

A próxima etapa da metodologia Lean Inception foi a definição das personas. Foram criadas duas personas fictícias: "Paulo", o líder da banda, e "Gustavo", o músico da banda.

Com base nas necessidades das personas e na visão do produto, em tornar mais simples, rápida e fácil a organização de bandas, foram priorizadas duas jornadas que mais demandam tempo dos líderes da banda: a definição de escalas e a seleção e disponibilização de músicas.

A primeira jornada é apresentada na tabela 5 'Escalas mensais e/ou recorrentes' e é uma jornada complexa porque envolve a iteração de múltiplos usuários para a tarefa ser executada com eficiência. O líder da banda precisa definir quem vai participar como músico em cada apresentação. Esta atividade é complexa porque envolve diferentes disponibilidades dos músicos, além da necessidade de rodízio e revezamento dos membros. Outros desafios existem com o fato de um mesmo músico tocar mais de um instrumento.

Já a segunda jornada "Seleção e disponibilização de músicas", apresentada na 6, está voltada à seleção de músicas das apresentações. Cada apresentação pode ter alterações

Persona 1: "Paulo, o líder da banda"
 Descrição: Paulo possui um bom conhecimento musical e toca desde jovem. Ele é uma pessoa organizada, dedicada à banda e apaixonada pela música. Paulo gosta de se envolver nas atividades administrativas, como a seleção de músicas e a disponibilização de cifras corretas para os músicos. A banda é sua prioridade, e ele se esforça para garantir que tudo ocorra sem problemas, desde ensaios até apresentações. Paulo valoriza a comunicação e deseja disponibilizar a playlist do evento para os membros da banda. Ele aprecia sugestões dos outros músicos e a confirmação de presença para um planejamento eficaz das atividades.

Figura 5: Persona 1: "Paulo", o líder da banda

Persona 2: "Gustavo", o músico da banda.
 Descrição: Gustavo é um jovem apaixonado por tocar instrumentos musicais e, por isso, se juntou a uma banda. Ele é comprometido com os ensaios e apresentações, mas acha desafiador lidar com as comunicações e tarefas de organização. Gustavo adora acessar as cifras das músicas e gostaria de contribuir mais com a seleção de músicas para os eventos. Ele deseja uma maneira mais rápida de confirmar sua presença em ensaios e shows.

Figura 6: Persona 2: "Gustavo", o músico da banda

no repertório, de acordo com a ocasião. Além disso, após a definição das músicas a serem tocadas no evento, os músicos precisam receber as cifras do repertório, que é um processo atualmente bem manual, buscando as cifras armazenadas fisicamente, tirando fotos e enviando aos membros. Além disso, o líder precisa levar em conta os músicos que vão tocar naquele para alocar um repertório que se adeque aos profissionais.

Passo	Atividade
1	Criar lista de músicos.
2	Atualizar lista de músicos, se necessário.
3	Acessar lista de músicos.
4	Saber que instrumento cada um toca.
5	Agrupar essa lista por instrumentos.
6	Pegar calendário.
7	Verificar se há alguma indisponibilidade pelos membros.
8	Verificar escala e confirmar presença.
9	Enviar escala para os membros.
10	Revisar se existe algum problema.
11	Alocar grupos de músicos por dia de evento.
12	Marcar próximos dias de eventos.
13	Ajustar escala com base no retorno dos membros.
14	Enviar escala final.
15	Escala concluída.

Tabela 5: Jornada 1: Escalas Mensais e/ou Recorrentes

Passo	Atividade
1	Acessar base de cifras.
2	Atualizar base de cifras.
3	Pensar se haverá um tema específico com os responsáveis.
4	Escolher músicas relacionadas ao tema, se aplicável.
5	Verificar se há sugestões dos membros.
6	Preencher o restante das músicas.
7	Realizar ajustes nas cifras, se necessário.
8	Compilar as cifras do repertório.
9	Enviar o repertório para os membros.

Tabela 6: Jornada 2: Seleção e Disponibilização de Músicas

4.2.3 Feature Brainstorming e Technical, Business and UX Review

Com base nas jornadas dos usuários priorizadas, as personas e ancorados na visão do produto, bem como a definição de "É - não é - Faz - não Faz", a próxima atividade foi realizar um brainstorming de funcionalidades, conforme instrui a metodologia, que pode atender às necessidades do usuário, bem como se aproximar da visão do produto.

Para atender à Jornada 1, focada em Escalas mensais e/ou recorrentes, diversas funcionalidades foram identificadas para facilitar o gerenciamento das escalas da banda. Desde o cadastro de músicos com informações detalhadas sobre habilidades e disponibilidade até a geração de relatórios e integração com calendários externos, o aplicativo proporcionará uma gestão eficiente. O líder terá a capacidade de atribuir papéis, criar eventos, comunicar-se diretamente com os músicos, revisar e aprovar escalas, e até mesmo escolher a metodologia de alocação. Detalhes completos sobre essas funcionalidades podem ser consultados no apêndice do trabalho.

Já para a Jornada 2, que se concentra na Seleção e disponibilização de músicas, o aplicativo trará facilidades como uma biblioteca de cifras, upload e busca de músicas, sugestões dos membros, seleção de temas e organização eficiente do repertório. A marcação de preferências, compartilhamento do repertório e ajustes de cifras proporcionam uma experiência completa. O líder terá a autonomia de editar as músicas mesmo após a sua seleção. Para um entendimento mais aprofundado, recomenda-se a consulta ao apêndice do trabalho.

Adicionalmente, funcionalidades estruturais foram identificadas para fortalecer o desenvolvimento contínuo do aplicativo. O cadastro detalhado de músicos, perfis de usuário, comunicação interna, agenda de eventos, confirmação de presença, controle de presença, gerenciamento financeiro e análise de performance contribuirão para uma gestão abrangente e eficaz da banda. Detalhes completos sobre essas funcionalidades estruturais estão disponíveis no apêndice do trabalho.

Após o levantamento das oportunidades de funcionalidades, o Lean Inception propõe a revisão baseada em 3 pilares: técnico, experiência do usuário e negócios. Equilibrando os 3 pontos, será possível garantir uma priorização de funcionalidades mais assertiva para atender à visão do produto.

Para cada funcionalidade levantada, foi classificada numa nota de 1-3 o quão relevante ela era para o usuário - com base nas entrevistas inicialmente realizadas. Além disso,

também foi dada uma nota 1-3 de quanto a funcionalidade pode ter potencial de impacto no negócio, pensando na escala desse aplicativo. E por fim, foi dada uma nota de 1-3 para o esforço necessário para produzir cada feature.

Sendo assim, foram priorizadas quais funcionalidades seriam implementadas neste projeto, buscando as features com maiores notas em usuário e negócio e menor nota em esforço.

Lista das features priorizadas após a revisão técnica, UX e negócio:

1. Biblioteca de cifras: o app deve ter uma biblioteca de cifras para acesso rápido e fácil aos acordes das músicas.
2. Upload de cifras: o app deve permitir o upload de cifras em diversos formatos, como texto ou PDF, para facilitar a organização e atualização do repertório.
3. Busca de músicas: o app deve permitir a busca de músicas por título, artista ou gênero musical, facilitando a seleção de novas músicas para o repertório.
4. Seleção de temas: o app deve permitir a seleção de temas para o repertório, como festas temáticas, datas comemorativas ou eventos específicos, facilitando a seleção de músicas relacionadas ao tema.
5. Organização do repertório: o app deve permitir a organização do repertório por ordem alfabética, artista ou gênero musical, facilitando a seleção e execução das músicas.
6. Cadastrar músicos próprios e pelo líder: o app deve permitir o cadastro de músicos com informações como nome, contato, habilidades, instrumentos tocados e disponibilidade.
7. Compartilhamento de repertório: o app deve permitir o compartilhamento do repertório com os membros da banda, para que possam acessar as cifras e informações das músicas a qualquer momento.
8. Compilação de cifras: o app deve permitir a compilação de todas as cifras do repertório em um único arquivo, facilitando a organização e acesso às músicas.
9. Atribuir papéis no evento (alocação de escala): o app deve permitir a atribuição de papéis a cada músico (por exemplo, vocalista, guitarrista, baterista, etc.).

10. Criação e edição de eventos: o app deve permitir a criação de eventos futuros, facilitando a programação e organização das escalas.

Após as features serem priorizadas, foi montado o sequenciador de atividades que apenas ordena as ações do usuário de forma macro ao iteragir com o produto.



Figura 7: Sequenciador de atividades

4.2.4 Canvas MVP

Após a execução de todas as atividades propostas do Lean Inception, foi produzido o Canvas MVP, apresentado na figura Canvas MVP, o qual compila as principais informações de todas as atividades realizadas.

Foram definidas as personas: Líder/gestor de bandas. Além disso, foram colocadas as jornadas priorizadas: i) Escalas mensais/eou recorrentes; e ii) Seleção e disponibilização de músicas para o repertório.

Além disso, o objetivo do MVP foi definido: realizar uma validação se um app para facilitar a organização de eventos resolve a dor dos gestores de banda de gastarem muito tempo na preparação de apresentações.

Também foram listadas as *features*, ou seja, as funcionalidades propostas ao MVP. Para a organização do trabalho, foi definido o ciclo de trabalho de 30 dias em que haverá planejamento detalhado das histórias de usuário e *features* que serão desenvolvidas no ciclo, como apresentado no início do capítulo.

Por fim, definiu-se o resultado esperado na avaliação de usabilidade associado à Eficácia, Eficiência e Satisfação do usuário, que será discutido na seção 8.3 "Resultados de usabilidade". E, também, uma métrica de validação, que corresponderá ao volume de bandas que usam o app semanalmente.



Figura 8: Canvas MVP

5 ESPECIFICAÇÃO DE REQUISITOS

A partir das entrevistas realizadas com os potenciais usuários do sistema, foram elaboradas histórias do usuário para definir os requisitos e composição dos itens de desenvolvimento para o progresso deste trabalho. Neste capítulo serão abordados os usuários identificados e as histórias do usuário.

5.1 Usuários

A partir da aplicação da metodologia Lean Inception já apresentada anteriormente, foram identificados dois usuários principais para o sistema de gestão de grupos musicais que está sendo desenvolvido:

Músico: Membro comum da banda e que deve acessar o sistema para visualizar as informações sobre os eventos da equipe, bem como em quais dias ele está escalado e também para acessar as músicas que serão tocadas em determinada apresentação.

Líder: Este, semelhantemente ao usuário anterior também tem interesse em ter as mesmas informações, porém sua demanda maior é realizar a gestão da equipe. Para este usuário é necessário que o sistema forneça uma maneira de ele elaborar escalas de músicos para os eventos, além de escolher as músicas que serão apresentadas diretamente pelo aplicativo. Ou seja, este é o principal usuário do sistema.

Com a visão dos usuários do sistema em mãos, foram elaboradas as histórias de usuário que atendem os requisitos solicitados durante as entrevistas.

5.2 Histórias do usuário

Como dito anteriormente, com base nas entrevistas com os usuários, foram elaboradas as histórias de usuário que descrevem as ações dos agentes, bem como os motivos dessas

ações. As histórias foram organizadas conforme explanado na teoria apresentada em capítulos anteriores, seguindo o modelo sugerido por Cohn (2004): "Como [usuário], eu gostaria de fazer [ação] para [finalidade]".

As histórias foram divididas em três temas principais:

- **Escala de músicos:** Diz respeito às tarefas referentes à montagem de escalas de músicos pelos líderes de banda para um evento específico.
- **Escolha de repertório:** Diz respeito às tarefas de escolha de músicas a serem tocadas em determinado evento.
- **Funcionalidades gerais:** Diz respeito às funcionalidades que são úteis para uma melhor experiência do usuário dentro do sistema.

A seguir são apresentadas as histórias em si dentro de cada um dos temas acima. Algumas estão com um nível de detalhamento maior porque foram as julgadas de maior prioridade no momento. Para essas, foram definidos os critérios de aceitação, que fazem parte da etapa de confirmação de uma história de usuário.

O MVP (Mínimo Produto Viável) do sistema foi definido com base na priorização das histórias e corresponde àquelas que tem os critérios de aceitação definidos, os quais serão mostrados abaixo. Foram priorizadas as quatro primeiras histórias, que são referentes à montagem de escalas de músicos.

1. Escala de músicos:

- (a) Como membro de banda, quero me cadastrar com minhas informações pessoais e instrumentos para que eu possa usar o aplicativo de bandas (MVP)
 - Verificar se banda já existe;
 - Verificar se já existe usuário com o mesmo email;
 - Verificar se a banda já tem um líder;
 - Se não tiver líder e a pessoa que está criando não for um líder, deve ter um indicativo de que a banda não tem líder.
- (b) Como líder de banda, quero planejar datas de eventos para que eu possa ter uma melhor organização dos eventos, de forma eficiente e centralizada (MVP)
 - Verificar se já tem evento agendado para mesma data e hora;

- Verificar se o evento está sendo criado em uma data no passado.
- (c) Como líder de banda, quero alocar músicos e papéis para o evento para que eu gaste menos tempo fazendo a escala de maneira manual (MVP)
- Verificar se músico está alocado em uma habilidade a qual este domina.
- (d) Como líder de banda, quero salvar minha escolha de eventos e músicos para eventos futuros para que eu tenha menos trabalho repetitivo (MVP)
- Não salvar repertório e escala juntamente com o evento recorrente;
 - Verificar se alteração em um evento recorrente altera todos os outros;
 - Verificar se é possível salvar a escala montada para uso posterior.
- (e) Como líder de banda, quero conseguir alocar músicos com base em critérios de seleção para que a montagem de escalas seja rápida e me tome pouco tempo

2. Escolha de repertório:

- (a) Como líder de banda, eu quero selecionar músicas para um evento para que eu possa visualizar durante a apresentação
- (b) Como líder de banda, eu quero compartilhar o repertório de um evento com os membros da minha banda para que todos fiquem alinhados acerca do que será apresentado
- (c) Como líder de banda, eu quero sugestões de músicas com base em um tema específico para me poupar tempo no momento de escolha
- (d) Como líder de banda, eu quero adicionar cifras e vídeo de referência para as músicas, para que os músicos tenham mais facilidade ao ensaiar para as apresentações

3. Funcionalidades gerais:

- (a) Como líder de banda, eu quero cadastrar pessoas à minha banda para poder fazer o controle das pessoas que fazem parte da mesma
- (b) Como músico ou líder, eu quero acessar os eventos da banda para poder me organizar
- (c) Como músico ou líder, eu quero alterar minhas informações de perfil
- (d) Como músico ou líder, eu quero visualizar a lista em vídeo das músicas que serão tocadas na próxima apresentação diretamente pelo aplicativo, para que eu não precise perder tempo procurando as referências em outro lugar

5.3 Considerações Finais

Neste capítulo foram apresentados os requisitos do sistema e a priorização do MVP com base nas entrevistas realizadas.

No capítulo a seguir será mostrada a arquitetura do sistema, que foi construída levando em consideração estes requisitos.

6 ARQUITETURA DO SISTEMA

O sistema desenvolvido seguiu uma arquitetura monolito modular. Para descrevê-la, foi utilizado o modelo C4 (apresentado na Seção 2.4). A arquitetura escolhida foi baseada em princípios de design e escalabilidade, buscando atender aos requisitos funcionais e não funcionais do sistema.

6.1 Arquitetura de Monolito Modular

A abordagem de monólito modular, como discutida neste parágrafo, foi inspirada nas ideias apresentadas por Simon Brown na GOTO Conference de 2018. Brown destacou a relevância da construção de sistemas como um único componente com uma estrutura modular interna, permitindo a separação lógica das funcionalidades em módulos distintos. Essa perspectiva visa facilitar o desenvolvimento, manutenção e evolução do sistema.

No contexto deste projeto, o monolito modular foi escolhido para atender às necessidades específicas do sistema. Ele oferece um equilíbrio entre a simplicidade de um único componente e a capacidade de dividir as responsabilidades em módulos isolados. Isso possibilita uma melhor organização do código, facilita a reutilização e a escalabilidade, e simplifica as operações de implantação e monitoramento.

A abordagem arquitetural do monolito modular tem uma sinergia forte com Domain-Driven Design (DDD) (EVAN, 2003). O DDD é uma abordagem de design de software que se concentra na modelagem e no design do domínio do problema, esta perspectiva propõe a criação de modelos semânticos que refletem as entidades (domínios) do negócio. Ao utilizar o DDD em conjunto com um Monolito Modular, os módulos podem ser estruturados em torno das diferentes partes do domínio, alinhando-se com as práticas e conceitos definidos pelo DDD. Em uma arquitetura deste tipo, cada módulo pode conter suas próprias regras de negócio, agregados, serviços e contextos.

6.2 Primeira camada da arquitetura: Contexto

Na primeira camada da arquitetura proposta são definidos os atores externos e suas dependências com o sistema em questão. A aplicação (que pode ser vista na Figura 5) receberá ações de um Músico, que é um usuário do sistema, seja ele um Líder ou que apenas participe de uma banda. O músico poderá gerenciar seus eventos, escalas e bandas, assim como descrito em capítulos anteriores. Além disso, existem outros dois agentes exteriores que são dependências da aplicação principal:

- Google Cloud Plataform: será usada a plataforma de Cloud da Google para persistir as cifras que já são conhecidas pelo sistema.
- Cifras Club: este é um website que atua como um repositório de cifras, sendo uma referência utilizada pelos músicos entrevistados, será usado como fonte para cifras que o sistema desconhece.



Figura 9: Camada de Contexto do Modelo C4

6.3 Segunda camada da arquitetura: Contêineres

A camada de Contêineres tem como objetivo detalhar a composição interna da aplicação principal, descrita no diagrama de Contexto.

Na arquitetura representada na Figura 9, observa-se que o Músico interage diretamente com um aplicativo mobile. Nesta camada são indicadas as tecnologias que serão utilizadas em cada componente. Em especial, o aplicativo será desenvolvido em React Native.

O aplicativo atuará como o lado do cliente da aplicação, fazendo as requisições de dados necessários para um servidor (descrito como API Service na imagem), o servidor será desenvolvido em Typescript. TypeScript é uma linguagem de programação que estende o JavaScript, adicionando tipagem estática opcional. Ela traz mais segurança, detecta erros em tempo de compilação e oferece recursos avançados como classes, interfaces e módulos para desenvolvimento de código mais robusto e escalável (MICROSOFT, 2012).

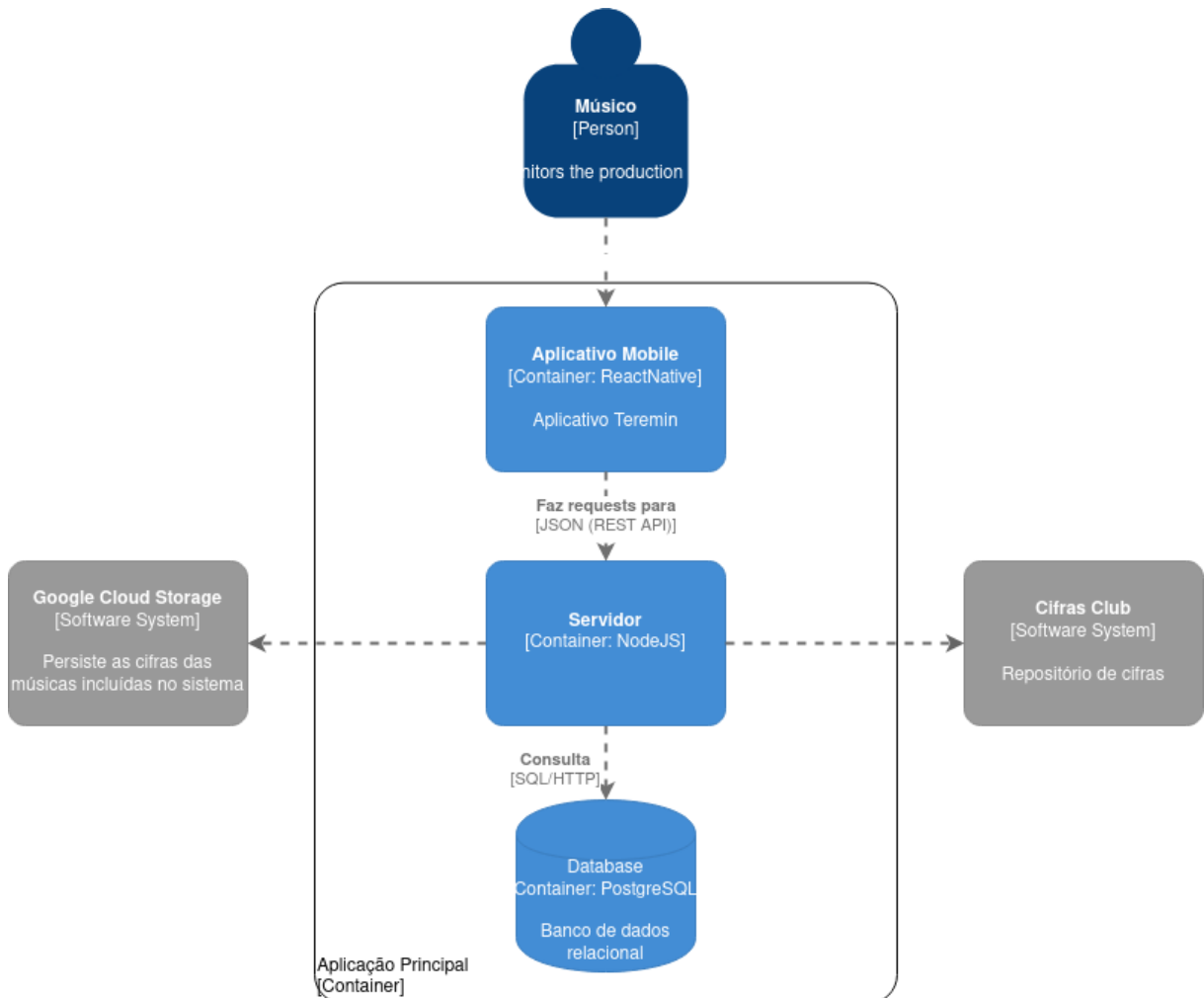


Figura 10: Camada de Contêineres do Modelo C4

O servidor será responsável pelas regras de negócio da aplicação, sendo também responsável pela comunicação com o Google Cloud Storage, para armazenar as cifras conhecidas, e com o Cifras Club, para busca de cifras desconhecidas. Por fim, o servidor persistirá todos os dados da aplicação, sobre os usuários, músicas, bandas, eventos, escalas e instrumentos. Para isso é usado o Postgres o sistema de gerenciamento de banco de dados relacional.

6.4 Terceira camada da arquitetura: Componentes

A camada de componentes irá detalhar o Contêiner nomeado como Servidor na camada anterior. Esta é a camada mais próxima da estrutura do código do ponto de vista arquitetural.

O servidor foi projetado para seguir a arquitetura de monolito modular. Com isso, cada entidade do sistema corresponderá a um módulo (domínio). Cada módulo tem uma separação lógica e organizacional bem definida. Conforme apresentado na Figura 10, em que as linhas pontilhadas representam a separação lógica entre os módulos.

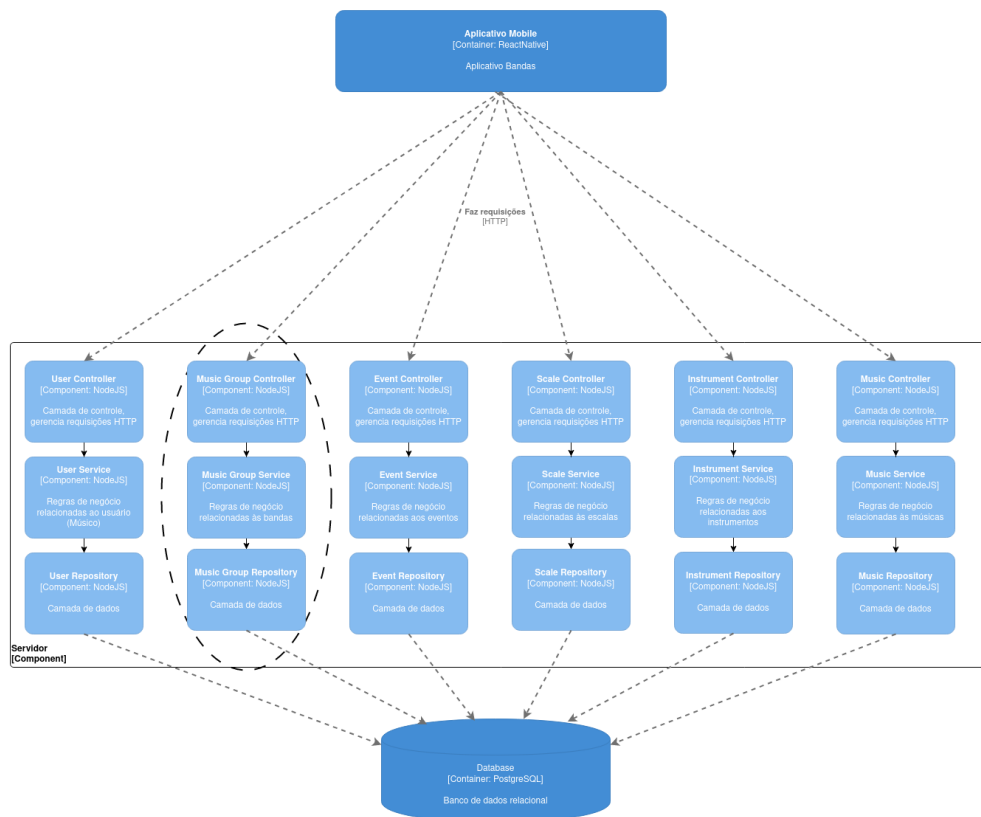


Figura 11: Camada de Componentes do Modelo C4

Cada módulo segue um padrão arquitetural chamado MVCS (Model-View-Controller-Service) é uma variação do padrão arquitetural MVC (Model-View-Controller) (FOWLER, 2002) que adiciona a camada de serviço. O MVCS separa as responsabilidades de uma aplicação em quatro partes: o Modelo (Model), responsável pela lógica de negócio e persistência dos dados, que foi chamado de Repositório no diagrama acima; a Visão (View), responsável pela apresentação da interface com o usuário, que é representada como o aplicativo mobile; o Controlador (Controller), responsável pela coordenação das requisições HTTP (interações entre a Visão e o Modelo); e o Serviço (Service), responsável

por abstrair a lógica de negócio complexa e realizar operações entre os demais componentes. Essa separação ajuda a promover a modularidade, a reutilização de código e a manutenibilidade da aplicação.

Por fim, o repositório é responsável pela abstração e gerenciamento das operações de persistência de dados no banco de dados. Ele fornece uma interface para realizar operações CRUD (Create, Read, Update, Delete) no banco de dados, ocultando a complexidade da interação direta com o PostgreSQL. O repositório também permite a separação clara entre a lógica de negócio da aplicação e as operações de banco de dados. Dessa maneira é garantido o princípio da responsabilidade única (MARTIN, 2000), que facilita a implementação de testes de unidade, trazendo mais manutenibilidade e testabilidade ao sistema.

6.5 Considerações

O sistema desenvolvido adota uma arquitetura monolito modular, seguindo o modelo C4 para descrição da arquitetura. Essa escolha busca atender aos requisitos funcionais e não funcionais do sistema, proporcionando uma estrutura organizada e modular que facilita o desenvolvimento, manutenção e evolução. A abordagem de monolito modular, inspirada nas ideias de Simon Brown, enfatiza a construção de sistemas como um único componente com uma estrutura interna modular, promovendo a separação lógica de funcionalidades em módulos distintos.

A arquitetura monolito modular adotada no desenvolvimento do sistema se alinha diretamente com o objetivo central do projeto. Ao buscar otimizar os processos de gestão de grupos musicais, a escolha dessa arquitetura visa oferecer uma solução coesa e eficiente para a organização dos membros do grupo e dos recursos disponíveis. A abordagem modular permite uma fácil adaptação às necessidades específicas dos grupos musicais, garantindo simplicidade de uso e facilidade de manutenção.

7 FUNCIONALIDADES DO APLICATIVO

Conforme mostrado na seção de Metodologia, o aplicativo desenvolvido conta com funcionalidades que atendem às necessidades dos músicos e gestores de bandas e que ajudem na economia de tempo para organização da produção de apresentações.

A partir do nome Teremin, foi desenvolvida uma identidade visual para o aplicativo e um padrão de fontes, componentes e elementos para serem utilizados na interface das funcionalidades.

Abaixo apresentamos as Telas de entrada e Tela de início do aplicativo e neste capítulo serão descritas e apresentadas as funcionalidades existentes no produto.

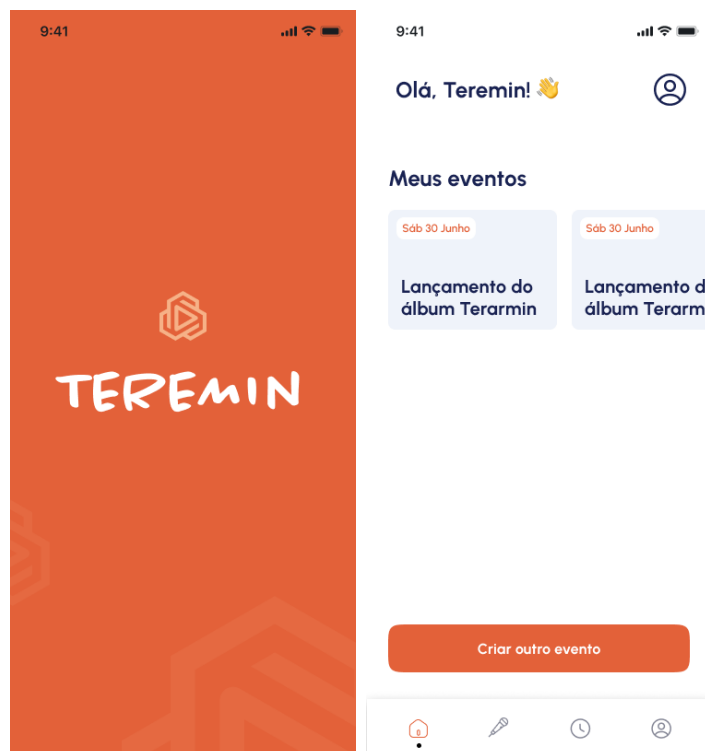


Figura 12: Tela de início e entrada

7.1 Cadastro de usuários

A seção de cadastro de usuário coleta informações importantes sobre os usuários, incluindo seus nomes, informações sobre a banda à qual pertencem, endereços de e-mail, além de detalhes específicos sobre as funções e instrumentos que desempenham.

A inclusão do nome da banda é crucial para estabelecer conexões entre membros de uma mesma banda e facilitar a comunicação e colaboração. A funcionalidade de seleção múltipla para função na banda oferece flexibilidade ao usuário, permitindo que indiquem se desempenham papéis de líder, músico ou ambas as funções. Essa abordagem reconhece a diversidade de papéis desempenhados pelos usuários dentro de uma banda.

Similar à função, a escolha múltipla para "Instrumentos" amplia a abrangência das habilidades do músico, possibilitando a indicação de todos os instrumentos que tocam. Isso é especialmente útil para conectar músicos com habilidades diversas e facilitar a formação de flexibilidade de escalas de músicos.

9:41

← Cadastro de conta

Nome

Tera

Nome da banda

Teramim

E-mail

Teramim@gmail.com

Função da banda

Líder Músico

Instrumento

Vocal, Guitarra, Violão

Salvar

Figura 13: Tela de cadastro de usuário

7.2 Cadastro de eventos

A funcionalidade de cadastro de eventos facilita a organização de bandas, oferecendo praticidade e eficiência ao reunir informações sobre o evento.

O campo "Nome do Evento" é projetado para identificar de maneira única cada ocasião, contribuindo para a clareza e reconhecimento. Um título distintivo destaca o evento na plataforma, facilitando a comunicação com os participantes.

Incluir a data do evento, horário e local é fundamental para informar os usuários sobre o momento específico do acontecimento. Essa informação permite que os participantes organizem suas agendas e estejam preparados para participar.

A opção de repetição do evento adiciona flexibilidade, especialmente para bandas que se apresentam regularmente em um mesmo local.

9:41

← Cadastro de evento

Nome do evento

Lançamento do álbum Terarmin

Data

08/08/2023

Horário de início Horário de fim

21:00 00:00

Local

Rua Teramim da Silva, São Paulo, N°0

Deseja repetir o evento?

Não repetir

Proximo

Figura 14: Tela de cadastro de evento

7.2.1 Escala de Músicos

A funcionalidade de escala de Músicos é uma ferramenta importante em plataformas dedicadas ao universo musical, proporcionando uma gestão eficiente na escolha de músicos vinculados a uma banda. Essa funcionalidade permite que os usuários armaze-

nem configurações frequentemente utilizadas, agilizando o processo de criação de escalas para eventos recorrentes.

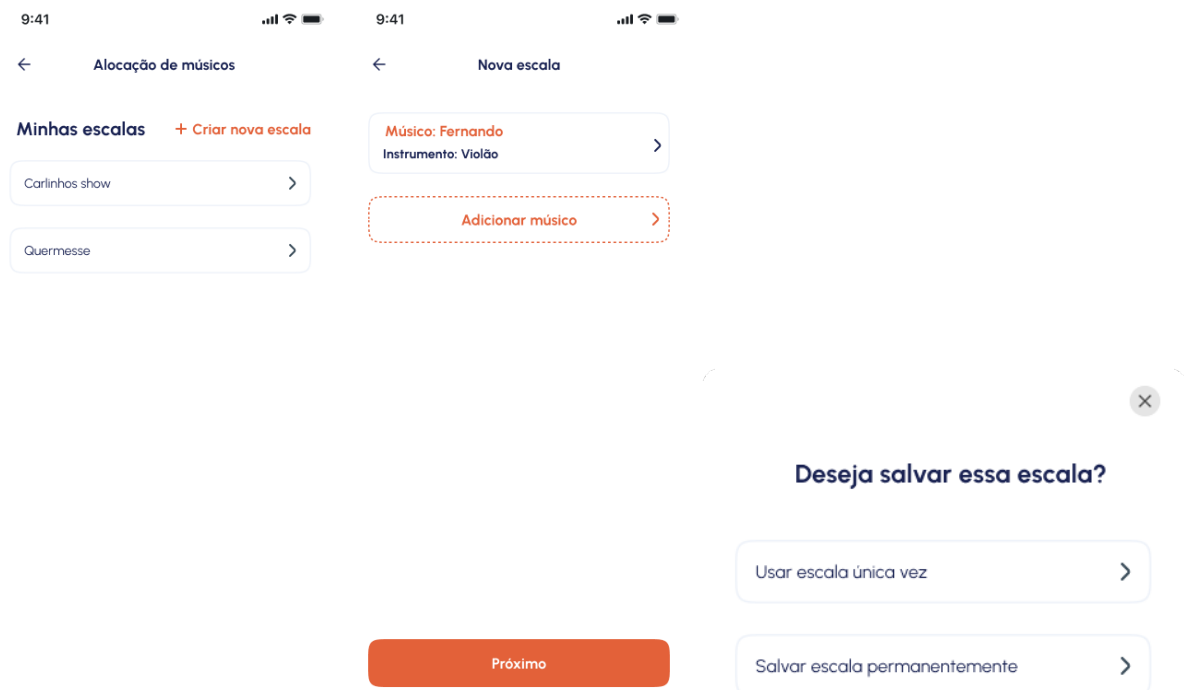


Figura 15: Tela de criação de escala, seleção do músico e escolha da repetição

Por fim, é exibida ao usuário uma tela de confirmação do evento para que seja possível verificar todas as informações definidas para o evento como data, horário, local, músicos e músicas e finalizar a criação do evento.

7.3 Escolha de músicas

A escolha das músicas possibilita que os membros da banda selecionem de maneira estratégica as canções que melhor se alinham ao estilo e ao público-alvo do evento.

Os usuários podem adicionar sempre novas músicas através do nome da música e nome do cantor/banda. Além disso, as músicas já adicionadas pelo usuário ficam salvas para serem selecionadas futuramente de forma mais rápida a novos eventos.

7.3.1 Cifras

Um dos componentes mais relevantes para o público-alvo do produto identificado na pesquisa qualitativa foi a cifra da música, pois permite ao músico ensaiar para a

9:41

← Confirmação de cadastro

Evento [✎ Editar](#)

Nome do evento
Lançamento do álbum Terarmin

Data
08/08/2023

Horário de início Horário de fim
21:00 00:00

Local
Rua Terarim da Silva, São Paulo, N°0

Escala [✎ Editar](#)

Nome do músico	Instrumento
Fernando	Violão
Carlos	Bateria
Luiz	Vocal
Paula	Guitarra

Música [✎ Editar](#)

Nome do música
Doce mel
Beija-flor

Criar evento

Figura 16: Tela de confirmação do evento

apresentação. Diante disso, o Teremin possui a funcionalidade de localização das cifras da música a partir de seu nome e o nome do autor, que esteja publicada na plataforma *Cifra Club*¹.

7.3.2 Vídeo

Além das cifras, como parte da preparação dos músicos para realizarem uma boa performance nos eventos, o vídeo relativo a determinada música do repertório é também importante. Os músicos relataram que podem, com isso, se familiarizarem com a melodia e tempo de uma nova música adicionada.

No aplicativo, os usuários conseguem acessar o vídeo oficial no *YouTube*² da música, a partir de seu nome e o nome do autor.

¹ Link: <https://www.cifraclub.com.br/>

² Link: <https://www.youtube.com/>

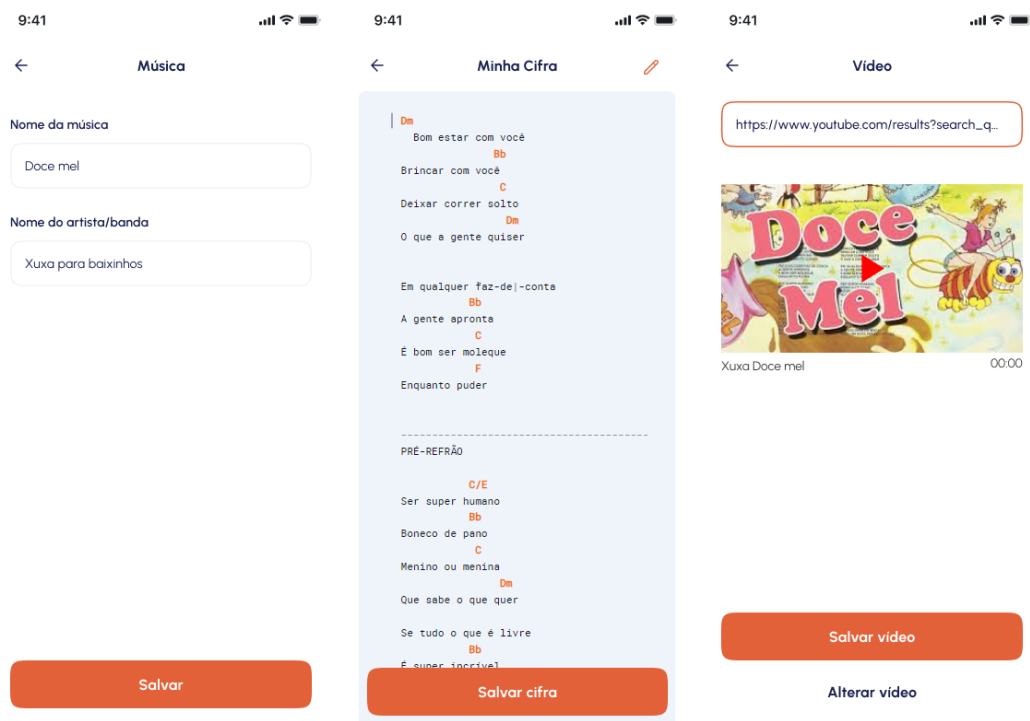


Figura 17: Tela de escolha da música, cifras e vídeo

8 RESULTADOS OBTIDOS

Neste capítulo, é apresentado o processo de lançamento, avaliação e resultados deste trabalho.

8.1 Lançamento do Aplicativo

O lançamento do Teremin foi conduzido em várias etapas. Como descrito na seção 4.1, a *sprint* 5 foi dedicada ao primeiro lançamento do aplicativo. Inicialmente, a plataforma selecionada para o lançamento prioritário foi a Google Play Store. Foi necessário criar uma conta de desenvolvedor na plataforma para a disponibilização do aplicativo para os usuários.

Ao submeter o aplicativo, ele foi revisado pelo Google Play para verificar que o aplicativo atendesse a todas as diretrizes e padrões de qualidade estabelecidos pela plataforma.

Durante o desenvolvimento do Teremin, desafios relacionados à hospedagem do banco de dados na nuvem foram enfrentados. Para superar esse obstáculo, o banco foi lançado no BigQuery¹, uma solução de banco de dados fornecida pelo Google. Essa plataforma provou ser bem-sucedida e permitiu que o aplicativo funcionasse sem problemas.

Após a resolução dessas questões, o MVP foi finalmente disponibilizado com sucesso na Google Play Store, tornando-o acessível aos usuários em geral.

8.2 Método de Avaliação de Usabilidade

Após o lançamento do aplicativo para o público, foi conduzida uma pesquisa com o objetivo de avaliar a usabilidade do sistema de acordo com a Norma ISO 9241, detalhado na Seção 2.1.1.

O teste de usabilidade foi conduzido para avaliar o aplicativo desenvolvido para ges-

¹Link: <https://cloud.google.com/bigquery>

tores de bandas. Cinco usuários, representando o público-alvo, foram envolvidos no processo, que ocorreu remotamente. A metodologia adotada foi o teste de usabilidade com moderação, visando a coleta de dados estruturados por meio de um questionário.

Durante o teste, os usuários interagiram com o aplicativo, buscando atingir seus objetivos relacionados à programação de eventos, como definir local e data, escala de músicos e músicas. A eficácia da tarefa foi verificada através da conclusão bem-sucedida de tarefas específicas.

Foram identificadas algumas dificuldades nas tarefas, destacando aspectos da interface do usuário, navegação e outros elementos que impactaram a eficiência da execução de determinadas atividades no aplicativo, associadas às ações principais para utilizar as funcionalidades, detalhadas na tabela 7.

A satisfação dos usuários foi avaliada, levando em consideração a percepção geral dos gestores de bandas em relação ao sistema. Os dados coletados foram analisados para fornecer *insights* sobre a experiência do usuário.

As métricas de usabilidade propostas pela norma ISO 9241 foram aplicadas. A eficiência foi mensurada pelo tempo gasto na execução de cada tarefa, enquanto a satisfação do usuário foi capturada por meio da percepção subjetiva dos usuários em relação ao sistema como um todo.

Número	Tarefa
1	Realize o cadastro no aplicativo
2	Cadastre um novo evento
3	Aloque os músicos no seu evento criando uma nova escala e a salvando
4	Defina o repertório do seu evento
5	Conclua a criação do seu novo evento
6	Visualize seu evento
7	Edite o Local do seu evento
8	Edite um músico do seu evento
9	Edite uma música do seu evento
10	Crie um novo evento
11	Aloque os músicos com a escala já salva
12	Acesse a aba de perfil do usuário

Tabela 7: Tarefas do Teste de Usabilidade

Cada tarefa recebeu uma atribuição de valores para as medidas de eficácia e eficiência. A eficácia foi avaliada com base na conclusão ou não da tarefa, atribuindo 0 se o usuário não a concluisse e 1 se a concluisse. Quanto à eficiência, foi registrado o tempo gasto para

a execução de cada tarefa.

Durante o cálculo da eficiência, o tempo total dedicado pelo usuário à manipulação do sistema durante a execução da tarefa foi considerado. Nos momentos de pausa na interação, seja para comunicação com o moderador ou por outros motivos, o tempo não foi contabilizado. O registro do tempo foi retomado quando o usuário reiniciava a interação com o sistema.

Essa abordagem permitiu uma avaliação retrospectiva das tarefas, fornecendo dados sobre a eficácia do usuário na conclusão das tarefas atribuídas e o tempo efetivamente empregado para realizar cada uma delas. Essas informações foram cruciais para a análise da usabilidade do sistema e a identificação de áreas potenciais para melhorias.

Além da atribuição dos valores da eficácia e eficiência, o moderador também tomou nota de todas as observações notadas na iteração do usuário com o sistema.

Após a execução das tarefas, foi coletada a métrica de satisfação do usuário com o aplicativo. Para tal, foi utilizado o Questionário de Satisfação SUS proposto pelo System Usability Scale descrito na seção 3.2. O questionário utiliza da escala Likert e traz uma série de afirmações que auxiliam na mensuração de quão satisfeito o usuário está com o sistema.

Após a realização de todos os testes de usabilidade, a próxima etapa envolveu a compilação dos problemas identificados durante os testes. Esses problemas foram categorizados com base em sua natureza, utilizando critérios específicos para avaliar diferentes aspectos da interface do usuário, como mencionado na seção 3.2.

Após essa categorização, os problemas foram então avaliados de acordo com seu Grau de Severidade, seguindo um algoritmo específico:

- Alto Grau de Severidade: quando o problema interfere significativamente na eficácia de uma tarefa.
- Médio Grau de Severidade: quando o problema impacta a eficiência de uma tarefa para mais de um usuário.
- Baixo Grau de Severidade: quando o problema afeta a eficiência de uma tarefa, mas apenas para pelo menos um usuário.

Essa metodologia de avaliação de problemas proporciona uma abordagem estruturada para identificar, classificar e priorizar os problemas de usabilidade, permitindo que as

equipes de design e desenvolvimento concentrem seus esforços nas áreas mais críticas para melhorar a experiência do usuário.

8.3 Resultados de Usabilidade

Como descrito na seção anterior, os testes foram conduzidos para análise do desempenho do aplicativo em resolver as necessidades dos usuários.

8.3.1 Eficácia

Analisando o desempenho dos usuários testados em relação à conclusão das doze tarefas, onze atividades foram concluídas por todos os usuários, demonstrando um bom desempenho na maioria das tarefas. No entanto, há uma atividade que não foi concluída por nenhum dos usuários, a qual consiste na edição de um músico para um evento específico.

Devido a essa não conformidade, o problema foi classificado como tendo um alto grau de severidade, indicando que é uma questão crítica que requer atenção imediata.

Ao analisar o diagnóstico de usabilidade, foram identificadas duas deficiências no fluxo do sistema. Primeiro, o fluxo não prevê a visualização dos músicos associados a uma escala, o que pode ser um obstáculo para os usuários que desejam verificar as informações dos músicos antes de fazer alterações. Além disso, a funcionalidade de edição dos músicos também não está contemplada no fluxo atual, o que impede que os usuários façam ajustes necessários nas informações dos músicos.

8.3.2 Eficiência

No contexto da eficiência do aplicativo, pode-se observar na Figura 18 que o tempo despendido em cada tarefa não teve alto desvio entre cada usuário, com exceção do Cadastro no aplicativo que será comentado mais abaixo.

A partir da análise dos testes, é possível compreender que o fluxo de Cadastro no aplicativo é uma atividade que, em termos gerais, apresentou uma eficiência moderada. Alguns usuários enfrentaram dificuldades específicas durante o processo de cadastro, notadamente na seleção dos instrumentos desejados. Portanto, melhorias nessa etapa podem ser implementadas para aprimorar a satisfação e eficácia dos usuários.

Número	Tarefa	Eficácia (%)
1	Realize o cadastro no aplicativo	100%
2	Cadastre um novo evento	100%
3	Aloque os músicos no seu evento criando uma nova escala e a salvando	100%
4	Defina o repertório do seu evento	100%
5	Conclua a criação do seu novo evento	100%
6	Visualize seu evento	100%
7	Edite o Local do seu evento	100%
8	Edite um músico do seu evento	0%
9	Edite uma música do seu evento	100%
10	Crie um novo evento	100%
11	Aloque os músicos com a escala já salva	100%
12	Acesse a aba de perfil do usuário	100%

Tabela 8: Teste de Usabilidade: Eficácia

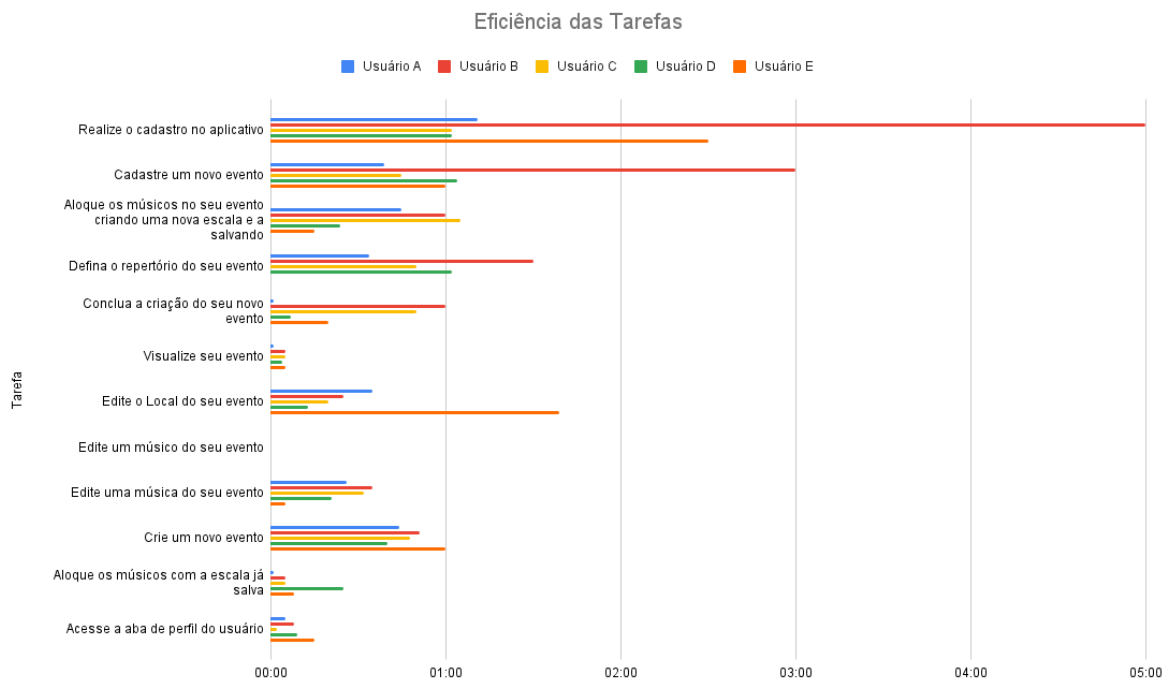


Figura 18: Eficiência Usuário

A Alocação de músicos nas escalas, por sua vez foi identificada como uma área de alto impacto na usabilidade. As atividades relacionadas a essa função geraram problemas significativos, afetando a eficácia, eficiência e satisfação dos usuários. A complexidade da alocação de músicos se destacou como um desafio e requer atenção imediata para melhorar a experiência global.

Por outro lado, a Criação de eventos, Visualização e Edição tiveram uma usabilidade

satisfatória. Os usuários conseguiram realizar essas tarefas de maneira eficaz, eficiente e satisfatória. Isso indica que essas áreas do aplicativo já atendem às expectativas dos usuários em termos de usabilidade, tornando-as aspectos positivos da experiência geral.

8.3.3 Satisfação

Por fim, como mencionado no método de avaliação, foi aplicado o questionário SUS com os usuários. A partir dele, foi possível entender que os elementos do aplicativo são considerados intuitivos e fáceis de usar, o que proporciona uma experiência satisfatória para os usuários.

Além disso, o sistema não é considerado desnecessariamente complexo, exceto em alguns aspectos específicos, como a criação de escalas, que foi avaliada como desafiadora (Alto nível de complexidade).

A integração das várias funções do aplicativo é avaliada positivamente, o que contribui para a conveniência e eficiência na realização das tarefas. No entanto, a percepção de segurança ao utilizar o sistema é considerada moderada, sugerindo que ainda existem áreas a serem melhoradas para aumentar a confiança dos usuários.

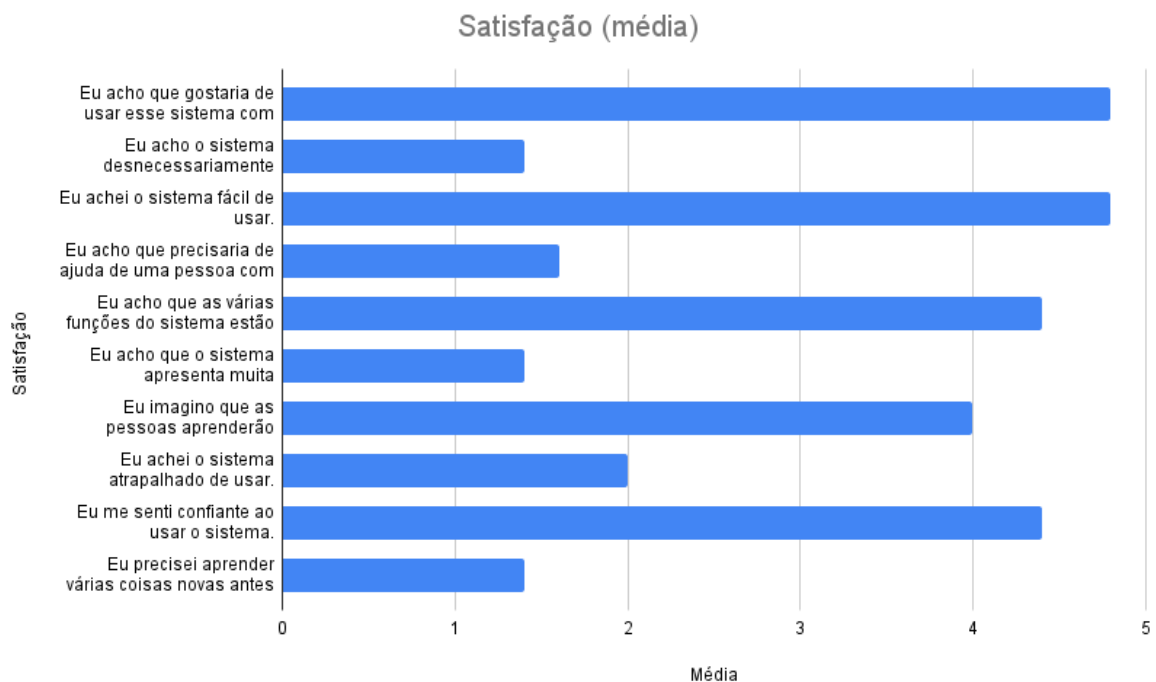


Figura 19: Satisfação Média

8.4 Conclusão

Os problemas levantados foram listados classificados de acordo com dois atributos: tipo do problema e grau de severidade.

Problema	Tipo	Severidade
Dificuldade em entender a necessidade de criar uma escala para alocar músicas e músicos.	Consistência	Alto
Falta de clareza na edição das escalas e na capacidade de visualizar e editar as escalas existentes.	Recorrência	Alto
Travamento da interface ao criar uma nova escala e não entender como adicionar novos nomes.	Recorrência	Alto
Problemas na seleção de instrumentos e falta de botão de confirmação na seleção.	Consistência	Alto
Problemas com senhas curtas ao se cadastrar.	Generalidade	Alto
Dúvidas sobre a necessidade de selecionar instrumentos ou músicos durante o cadastro.	Generalidade	Médio
A confusão sobre o código da banda e a necessidade de esclarecer como funciona.	Consistência	Médio
A dificuldade em voltar todos os passos para fazer uma alteração durante a criação de um evento.	Recorrência	Médio
Falta de clareza no significado de "local" no contexto do aplicativo.	Generalidade	Médio
Dificuldade em entender o propósito de "Minhas escalas" no aplicativo.	Generalidade	Médio
Pequenos problemas de usabilidade, como a localização dos botões de seleção.	Generalidade	Baixo
Dúvidas sobre a possibilidade de auto completar com base no CEP e a integração com o Google Maps.	Consistência	Baixo
Questionamentos sobre a existência de grupos pré-existentes para adicionar músicos.	Generalidade	Baixo
Tentativa de fazer login diretamente sem se cadastrar primeiro.	Generalidade	Baixo
Erro de email já em uso antes de concluir o cadastro.	Generalidade	Baixo
Dúvida sobre se apenas um instrumento poderia ser selecionado durante o cadastro de músicos.	Recorrência	Baixo

Figura 20: Tabela de Problemas

Foram identificados diversos problemas no uso do aplicativo, abrangendo áreas como consistência, generalidade e recorrência. Nenhum deles foi corrigido ao longo deste trabalho, devido ao prazo, mas foram documentados para correções futuras.

Um desafio significativo é a dificuldade em compreender a necessidade de criar escalas para alocar músicas e músicos, o que compromete a consistência do aplicativo. Para solucionar, sugere-se melhorias na comunicação da funcionalidade, como a inclusão de tutoriais ou mensagens explicativas.

Outro problema recorrente é a falta de clareza na edição das escalas e na visualização e edição das escalas existentes, impactando a consistência. Propõe-se aprimoramentos na interface do usuário, como rótulos mais claros, botões intuitivos e feedback visual.

A ocorrência frequente de travamentos durante a criação de uma nova escala e a confusão sobre como adicionar novos nomes representam desafios graves e recorrentes. A solução envolve ajustes na lógica do aplicativo para proporcionar uma experiência mais suave e orientações claras.

Problemas na seleção de instrumentos e a ausência de um botão de confirmação são questões de consistência com alta severidade. Para resolver, sugere-se melhorias na interface, incluindo confirmações visuais e feedback imediato.

A dificuldade em retroceder durante a criação de um evento e a falta de clareza no significado de "local" são desafios recorrentes e gerais. Recomenda-se a implementação de

um sistema de navegação mais intuitivo e esclarecimentos na terminologia.

Problemas menores, como a localização dos botões de seleção, são questões gerais com baixa severidade, solucionáveis com ajustes na interface para aprimorar a intuição do usuário.

Dúvidas sobre a capacidade de auto preenchimento com base no CEP e a integração com o Google Maps são problemas de consistência com baixa severidade. A solução inclui melhorias no preenchimento automático e uma comunicação mais clara sobre as características do aplicativo.

Questionamentos sobre a existência de grupos pré-existentes para adicionar músicos e tentativas de login direto sem cadastro são desafios gerais com baixa severidade. Esclarecer esses pontos por meio de mensagens informativas pode aprimorar a compreensão do usuário.

O erro de email já em uso antes de concluir o cadastro representa um desafio geral com baixa severidade. Melhorias na validação de dados durante o processo de cadastro podem resolver esse problema.

Dúvidas sobre a possibilidade de selecionar apenas um instrumento durante o cadastro de músicos são questões recorrentes com baixa severidade. Esclarecer essa opção durante o processo de cadastro pode ser suficiente para resolver essa dúvida.

8.5 Alcance de Metas

Com base nos indicadores estabelecidos na seção 4.2, as metas propostas foram alcançadas durante o desenvolvimento do projeto. A publicação do aplicativo na Play Store² foi concluída até o final de outubro de 2023, seguindo as etapas delineadas pela metodologia *Lean Inception*, desde o desenvolvimento até a implementação.

Quanto à segunda meta, que visava à utilização ativa do aplicativo por, pelo menos, três bandas diferentes semanalmente, os dados de satisfação apresentados na seção 8.3.3 de resultados indicam uma adesão consistente e superiores às expectativas iniciais. A boa avaliação dos usuários foi capaz de indicar e incentivar o uso do Teremin para amigos e colegas.

Atualmente, o Teremin teve 8 downloads, sendo seu uso realizado principalmente pelos gestores de bandas em uma frequência de uso semanal.

²Link para download: <https://play.google.com/store/apps/details?id=com.gustavobarros.teremin>

9 CONSIDERAÇÕES FINAIS

Este trabalho propôs a elaboração do aplicativo TERE MIN, voltado para a gestão de grupos musicais.

Ao longo do texto, foi apresentada metodologias, como a especificação de requisitos, a metodologia ágil utilizada no desenvolvimento do projeto, a arquitetura do sistema e a aplicação de conceitos teóricos, como o design centrado no usuário e a Lean Inception. Além disso, o trabalho destacou a importância das entrevistas realizadas com potenciais usuários para a elaboração das histórias do usuário, que serviram como base para o levantamento de requisitos e a composição dos itens de desenvolvimento do aplicativo.

O app, que já está disponível na Play Store¹, conta com funcionalidades como agendamento de eventos, escalas de músicos, seleção de músicas e visualização de cifras. O app já foi testado por mais de 5 músicos e gestores de bandas.

A validação do aplicativo aconteceu através de entrevistas de usabilidade com os usuários finais, os quais foram requisitados a desempenharem algumas tarefas no aplicativo. A partir das respostas e da avaliação do grupo acerca do andamento das tarefas realizadas pelos usuários, foram feitas pontuações em relação à eficiência e eficácia da tarefa, o que trouxe clareza acerca da qualidade geral do sistema construído.

¹Link para download: <https://play.google.com/store/apps/details?id=com.gustavobarros.teremin>

10 APÊNDICE

10.1 Brainstorming de funcionalidades

Funcionalidade	Descrição
Cadastrar músicos	O app deve permitir o cadastro de músicos com informações como nome, contato, habilidades, instrumentos tocados e disponibilidade.
Gerar lista de músicos	A partir dos músicos cadastrados, o app deve permitir a criação de uma lista de músicos disponíveis para escala, facilitando a organização.
Verificar disponibilidade	O app deve permitir a visualização da disponibilidade dos músicos para cada data, facilitando a escolha da escala mais adequada.
Atribuir papéis	O app deve permitir a atribuição de papéis a cada músico (por exemplo, vocalista, guitarrista, baterista, etc.).
Criação de eventos	O app deve permitir a criação de eventos futuros, facilitando a programação e organização das escalas.
Atribuição de músicos a eventos	O app deve permitir a alocação dos músicos aos eventos, considerando a disponibilidade e habilidades de cada um.
Confirmação de presença	O app deve permitir que os músicos confirmem ou recusem sua presença em um evento, agilizando o processo de organização.
Comunicação com os músicos	O app deve permitir a comunicação direta com os músicos, para confirmar presença, resolver conflitos, enviar informações e lembrar sobre eventos.
Revisão e aprovação da escala	O app deve permitir a revisão e aprovação da escala pelos músicos, antes da confirmação final.
Geração de relatórios	O app deve permitir a geração de relatórios com informações como escalas passadas, frequência de participação de cada músico, habilidades mais utilizadas, entre outras.
Integração com calendário	O app deve permitir a integração com calendários externos (como Google Calendar ou Apple Calendar), facilitando a visualização de eventos futuros e disponibilidade dos músicos.
Forma de escolher a metodologia de alocação	O app deve dar ao músico a possibilidade de selecionar se quer alocar os músicos com base em algum critério, como aleatoriedade, por exemplo.
Edição dos músicos da escala	Líder consegue mudar músicos selecionados, mesmo após salvar.

Tabela 9: Brainstorming Jornada 1: Escalas Mensais e/ou Recorrentes

Funcionalidade	Descrição
Biblioteca de cifras	O app deve ter uma biblioteca de cifras para acesso rápido e fácil aos acordes das músicas.
Upload de cifras	O app deve permitir o upload de cifras em diversos formatos, como texto ou PDF, para facilitar a organização e atualização do repertório.
Busca de músicas	O app deve permitir a busca de músicas por título, artista ou gênero musical, facilitando a seleção de novas músicas para o repertório.
Sugestões de músicas	O app deve permitir que os membros da banda sugiram músicas para o repertório, facilitando a colaboração e aumentando o engajamento dos membros.
Seleção de temas	O app deve permitir a seleção de temas para o repertório, como festas temáticas, datas comemorativas ou eventos específicos, facilitando a seleção de músicas relacionadas ao tema.
Organização do repertório	O app deve permitir a organização do repertório por ordem alfabética, artista ou gênero musical, facilitando a seleção e execução das músicas.
Marcação de preferências	O app deve permitir a marcação de preferências por cada membro da banda, como músicas favoritas ou instrumentos tocados com mais facilidade, facilitando a seleção de músicas para apresentações.
Compartilhamento de repertório	O app deve permitir o compartilhamento do repertório com os membros da banda, para que possam acessar as cifras e informações das músicas a qualquer momento.
Ajustes de cifras	O app deve permitir ajustes nas cifras, para que se adaptem às preferências de afinação e arranjo da banda.
Compilação de cifras	O app deve permitir a compilação de todas as cifras do repertório em um único arquivo, facilitando a organização e acesso às músicas.
Edição das músicas do repertório	Líder consegue mudar as músicas selecionadas, mesmo após salvar.

Tabela 10: Brainstorming Jornada 2: Seleção e disponibilização de músicas

Funcionalidade	Descrição
Cadastro de músicos	O app deve permitir o cadastro de todos os membros da banda, com informações como nome, telefone, email, instrumento principal e outros instrumentos, habilidades e experiência musical, para que o líder possa gerenciar e acessar facilmente todas as informações da banda em um só lugar.
Perfis de usuário	O app deve permitir que cada músico tenha um perfil próprio, onde possa acessar informações sobre a banda, ver a agenda de shows, confirmar presença em eventos, visualizar as músicas do repertório e se comunicar com outros membros da banda.
Comunicação interna	O app deve permitir que os membros da banda se comuniquem entre si, por meio de um chat interno ou fórum, para que possam discutir detalhes das músicas, acertar horários de ensaios e eventos, e compartilhar sugestões e ideias para o repertório.
Agenda de eventos	O app deve permitir a criação de uma agenda de eventos, onde o líder possa cadastrar os próximos shows, ensaios e eventos da banda, e os músicos possam visualizar e confirmar presença em cada um deles.
Confirmação de presença	O app deve permitir que os músicos confirmem sua presença em cada evento, para que o líder possa ter uma visão clara da disponibilidade e compromisso de cada um deles.
Controle de presença	O app deve permitir que o líder faça o controle de presença de cada músico nos ensaios e eventos, para que possa avaliar o desempenho e compromisso de cada um deles.
Gerenciamento financeiro	O app deve permitir que o líder faça o gerenciamento financeiro da banda, registrando as receitas e despesas de cada evento, e calculando a remuneração de cada músico de acordo com o combinado previamente.
Análise de performance	O app deve permitir que o líder faça a análise de performance da banda, acompanhando o desempenho em cada evento, registrando feedbacks e avaliações de público e contratantes, e identificando oportunidades de melhoria para a banda.

Tabela 11: Brainstorming Funcionalidades Estruturais

	Discordo completamente	Discordo	Neutro	Concordo	Concordo completamente
1. Eu penso que gostaria de usar este sistema freqüentemente.					
2. Achei este sistema desnecessariamente complexo.					
3. Achei que foi fácil usar este sistema.					
4. Eu penso que precisaria de ajuda para poder usar este sistema.					
5. Achei que as várias funções deste sistema estavam bem integradas.					
6. Achei que havia muita inconsistência neste sistema.					
7. Eu imagino que a maioria das pessoas aprenderia a usar este sistema rapidamente.					
8. Achei este sistema muito incômodo de usar.					
9. Eu me senti muito seguro(a) utilizando este sistema.					
10. Eu precisei aprender muitas coisas antes de utilizar este sistema.					

Figura 21: Questionário de Satisfação SUS – System Usability Scale

11 REFERÊNCIAS

CAROLI, P. **Lean Inception: como alinhar pessoas e construir o produto certo**. São Paulo: Casa do Código, 2018.

ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11 Guidance on usability (1998)

CONSTANTINE, Larry; LOCKWOOD, Lucy. **Usage-Centered Design and Agile Methods: A High-Speed Approach to User-Centered Design**. Addison-Wesley Professional, 2005.

SARKER, Suprateek e MAHMOOD, Anisul. **The impact of lean UX practices on software quality: An empirical study**. Journal of Systems and Software, v. 117, p. 412-427, dezembro de 2016.

CONSTANTINE, L; LOCKWOOD, L. **Software for Use: A Practical Guide to the Models and Methods of Usage-Centered Design**. Reading, MA: Addison-Wesley Professional, 1999.

NORMAN, D. A. **The psychology of everyday things**. New York: Basic Books, 1988.

NIELSEN, J. **Usability engineering**. Academic Press, 1993.

COOPER, Alan. **The inmates are running the asylum: why high-tech products drive us crazy and how to restore the sanity**. 1a ed. Indianapolis, IN: Sams Publishing, 1999.

CAROLI, P. **Lean Inception: como alinhar pessoas e construir o produto certo**. São Paulo: Casa do Código, 2018.

CAROLI, Paulo. **Direto ao ponto: criando produtos de forma enxuta**. 1. ed. São Paulo: Casa do Código, 2014.

RIES, Eric. **The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses**. Nova Iorque: Crown Business, 2011.

RIES, Eric. **The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses**. Nova Iorque: Crown Business, 2011.

TORRES, Teresa. **Continuous Discovery Habits: Discover Products that Create Customer Value and Business Value**. 1ª edição. San Francisco, CA: Product Talk, 2020.

BETIOL, A. H.; **Avaliação de usabilidade para os computadores de mão: um estudo comparativo entre três abordagens para ensaios de iteração**. Universidade Federal de Santa Catarina: Programa de Pós-Graduação em Engenharia de Produção. Florianópolis, 2004.

SAURO, Jeff; LEWIS, James R. **Quantifying the User Experience: Practical Statistics for User Research**. 2. ed. San Francisco: Morgan Kaufmann, 2016.

LINCOLN, M; **Customer Success: How Innovative Companies Are Reducing Churn and Growing Recurring Revenue**. Estados Unidos: Customer Success LLC, 2014.

DINIZ, T. A., CERQUEIRA, R. C., BARCELLOS, M. P. and CONTE, T. U. **The impact of lean UX practices on software quality: An empirical study**. Journal of Systems and Software, v. 150, p. 92-103, Feb. 2019.

BROWN. S. **Modular Monoliths**. Apresentado na GOTO Conference, 2018. Disponível em: <https://www.youtube.com/watch?v=5OjqD-ow8GE>

BROWN. S. **The C4 model for visualising software architecture**. Disponível em:

<https://c4model.com/>. Acesso em: 12 jun. 2023.

EVANS, E. **Domain-Driven Design: Tackling Complexity in the Heart of Software**. 1. ed. Boston: Addison-Wesley, 2003.

Microsoft. **Por que criar TypeScript**. 2012.

Disponível em: <https://www.typescriptlang.org/why-create-typescript>. Acesso em: 09 dez. 2023.

BOOCH, G.; RUMBAUGH, J.; JACOBSON, I. **The Unified Modeling Language User Guide**. 1. ed. Boston: Addison-Wesley, 2005

FOWLER, M. **Patterns of Enterprise Application Architecture**. 1. ed. Boston: Addison-Wesley, 2002.

MARTIN, R. C. **Design Principles and Design Patterns**. Academic Press, 2000.

NUGROHO, Adi; SANTOSA, Paulus Insap; HARTANTO, Rudy. **Usability Evaluation Methods of Mobile Applications: A Systematic Literature Review**. In: 2022 International Symposium on Information Technology and Digital Innovation (ISITDI). IEEE, 2022. p. 92-95.

MOUMANE, Karima; IDRI, Ali; ABRAN, Alain. **Usability evaluation of mobile applications using ISO 9241 and ISO 25062 standards**. SpringerPlus, v. 5, p. 1-15, 2016.

PARK, Junho; ZAHABI, Maryam. **A novel approach for usability evaluation of mobile applications**. In: Proceedings of the Human Factors and Ergonomics Society Annual Meeting. Sage CA: Los Angeles, CA: SAGE Publications, 2021. p. 437-441.

SUTHERLAND, Jeff. **SCRUM: A arte de fazer o dobro de trabalho na metade do tempo**. Leya, 2014.

BROOKE, John. **Sus: a “quick and dirty” usability**. Usability evaluation in industry, v. 189, n. 3, p. 189-194, 1996.

NUGROHO, Adi; SANTOSA, Paulus Insap; HARTANTO, Rudy. **Usability Evaluation Methods of Mobile Applications: A Systematic Literature Review.** In: 2022 International Symposium on Information Technology and Digital Innovation (ISITDI). IEEE, 2022. p. 92-95.

Jeffries, R., Miller, J., Wharton, C., Uyeda, K. **User interface evaluation in the real world: a comparison of four techniques.** Proceedings of CHI'91, New Orleans, April 28-May 3, 1991.