

APLICAÇÃO MOBILE DE GERENCIAMENTO DE REPERTÓRIOS E PARTITURAS PARA ORQUESTRAS

Micael Mendes de Souza¹, Allan Farias Fávaro²

Resumo: O armazenamento de partituras em formato físico ainda é comum em orquestras, o que pode resultar em deterioração, perda e dificuldades de organização durante ensaios e apresentações. Como alternativa, recursos digitais permitem maior preservação e praticidade no uso desses materiais. Este trabalho desenvolveu uma aplicação mobile para gerenciar repertórios e partituras de orquestras, utilizando Dart e o framework Flutter, a arquitetura BLoC, o Firebase Firestore para dados estruturados e o Supabase para o armazenamento dos arquivos em PDF. A aplicação possibilita o cadastro de músicos e maestros, criação de repertórios, organização de agendamentos e compartilhamento de partituras de acordo com cada instrumento, voz e clave, além de permitir o acesso sincronizado em tempo real durante as sessões. O sistema funcionou corretamente nas sessões em tempo real e no armazenamento de músicas com o arranjo e suas partituras musicais. Como trabalhos futuros, sugere-se a integração de arquivos MIDI, funcionalidades de anotação digital e suporte a controle de páginas via pedal.

Palavras-chave: gestão de eventos; partituras digitais; orquestra; gestão de repertórios.

¹Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), micaelmendesdesouza@unesc.net

²Orientador. Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense (Unesc), allanfavar@gmail.com

ABSTRACT: The storage of sheet music in physical format is still common in orchestras, which can result in deterioration, loss, and difficulties in organization during rehearsals and performances. As an alternative, digital resources allow greater preservation and practicality in the use of these materials. This work developed a mobile application to manage orchestral repertoires and sheet music, using Dart and the Flutter framework, the BLoC architecture, Firebase Firestore for structured data, and Supabase for storing PDF files. The application enables the registration of musicians and conductors, the creation of repertoires, the organization of schedules, and the sharing of sheet music according to each instrument, voice, and clef, in addition to allowing synchronized real-time access during sessions. The system operated correctly in real-time sessions and in storing the musical arrangements and their corresponding sheet music. As future work, the integration of MIDI files, digital annotation features, and pedal-based page turning support is suggested.

Keywords: event management; digital sheet music; orchestra; repertoire management.

1 INTRODUÇÃO

Muitas orquestras ainda armazenam partituras em formato físico, os quais estão frequentemente expostos à deterioração e à perda, comprometendo o uso futuro dos acervos, das partituras, e também a preservação e a organização destes documentos (Salcedo et al., 2022). A UFSM (Universidade Federal de Santa Maria) possui uma política de indexação e catalogação onde as partituras são armazenadas em arquivos de aço para evitar o acúmulo de umidade e proliferação de micro-organismos (UFSM, 2024).

Para evitar os problemas do armazenamento físico algumas instituições digitalizam seus acervos em PDF's, disponibilizando acesso online aos músicos e Maestros (Salcedo et al., 2022). Partituras digitais otimizam os eventos para músicos que usam dispositivos móveis, extinguindo a necessidade da impressão das partituras, iluminação ou até mesmo as pilhas de folhas e trocas de páginas (Lima, 2024).

Optar por plataformas móveis para esta aplicação é vantajoso devido à sua portabilidade e acessibilidade. O crescimento significativo de dispositivos móveis impulsiona a contínua evolução do desenvolvimento mobile, resultando no surgimento de diversas tecnologias como Flutter que contribui para o desenvolvimento multiplataforma (Fava, 2023).

O Flutter é um framework do Dart que oferece suporte aos desenvolvedores na criação de aplicativos móveis, com várias ferramentas e arquiteturas limpas, como a arquitetura BLoC (Business Logic Component). O BLoC é um dos padrões mais utilizados como arquitetura, na qual fornece a separação direta de preocupações, como gerenciamento de estados, capacidade de teste e reutilização de código (Huynh, 2023).

Este trabalho tem como objetivo propor uma solução para gestão de partituras e controle de ensaios e apresentações de orquestras. A solução proposta consiste em uma aplicação mobile, desenvolvida em Flutter, utilizando o padrão de arquitetura BLoC, e o banco de dados Firebase Firestore para o armazenamento e sincronização em tempo real, junto com o Supabase para armazenar os arquivos no formato PDF. A aplicação mobile tem como foco orquestras que possuem agendas na qual precisam de repertórios que podem ter temas e eventos diferentes durante o ano.

Para alcançar esse propósito, o trabalho estabelece como objetivos específicos a análise de pesquisas sobre a funcionalidade e métodos para armazenar, preservar e gerenciar os arranjos com suas partituras; o entendimento da linguagem de programação Dart do *framework* Flutter e da biblioteca *Cloud Firestore*; e a implementação das funcionalidades do aplicativo.

O desenvolvimento de uma aplicação mobile para gerenciar as partituras e repertórios é essencial em um cenário onde se utiliza métodos que dificultam a preservação, a organização e a praticidade, podendo os documentos serem expostos a deterioração e perda, prejudicando a performance musical (Salcedo et al., 2022). O uso de dispositivos móveis agiliza o acesso às partituras, eliminando a necessidade de transportar grandes volumes de papel, imprimir constantemente, gerenciar folhas e trocar páginas, tornando ensaios e apresentações mais eficazes e melhorando a performance geral (Lima, 2024).

2 FUNDAMENTAÇÃO TEÓRICA

O uso de dispositivos móveis tem crescido nos últimos anos por causa da sua portabilidade e acessibilidade e com esse crescimento vem fazendo que seja utilizado mais tecnologias que contribuem com o desenvolvimento para multiplataformas como a plataforma Android e ferramentas como Flutter, React Native, Ionic entre outras (Fava, 2023).

Existem diversas tecnologias utilizadas para a aplicação mobile, essas tecnologias servem para produzir aplicações multiplataformas e es-

sas possibilitam a otimização de nossos esforços e garantem entregas ágeis (Almeida, 2019).

A linguagem Dart é uma linguagem C-style, parecida com C, C++, Java, Javascript ou C#. Ela é tipada, mas a definição de tipos é opcional. Foi lançada em 2011 com o objetivo inicial de substituir a linguagem Javascript (Teixeira et al., 2024).

O Flutter é um framework reconhecido como uma estrutura multi-plataforma que oferece suporte aos desenvolvedores, possuindo varias ferramentas e arquiteturas limpas, como a arquitetura BLoC, essa arquitetura auxilia a manter a estrutura de arquivos organizada, além da arquitetura possui também bibliotecas que possibilitam o uso de vários recursos como a biblioteca Cloud Firestore (Huynh, 2023).

O Cloud Firestore é um banco de dados NoSQL na Nuvem, que pode lidar com grandes conjuntos de dados (Huynh, 2023). Esse banco de dados é flexível e escalonável para o desenvolvimento mobile, Web e servidores com o Firebase e o Google Cloud. Ele mantém os dados em sincronia usando listeners de tempo real e possui a disponibilidade off-line, oferecendo suporte para web e dispositivos móveis (Huynh, 2023).

3 TRABALHOS CORRELATOS

Em meio as pesquisas realizadas foram encontrados alguns estudos que exploram o uso de tecnologias para o armazenamento de partituras. Em meio as abordagens encontradas destacam-se o uso de aplicativos

O MobileSheets é um aplicativo para gerenciar partituras, possui diversas funcionalidades, é possível fazer anotações desenhando com uma caneta. Pode criar setlists para apresentações e atualizar os repertórios, oferece também a interação com o pedal Bluetooth e alternância no modo de exibição, além disso, possui o emparelhamento para conectar dois tablets via Wi-Fi ou Bluetooth com o modo de exibir duas páginas. Importa arquivos da nuvem e possui ferramentas para reproduzir faixas com o reproduutor de áudio e manter o tempo com o metrônomo (MobileSheets, 2025).

A plataforma digital Newzik é utilizada para gerenciamento e leitura de partituras digitais, foca em músicos profissionais, instituições musicais e orquestras (Newzik, 2025). O Newzik App é possível criar Setlists e projetos. Oferece anotações e adição de faixas de áudio ou MIDI, sendo possível que o usuário grave a si mesmo ou incorpore vídeos no YouTube. O Newzik permite a compartilhamento em tempo real com outros usuários dentro de pastas compartilhadas (Newzik, 2025).

4 MATERIAIS E MÉTODOS

Esta pesquisa resultou no desenvolvimento de uma aplicação mobile que oferece uma solução para gerenciar repertórios e partituras de orquestras. O sistema funciona offline para armazenamento local, mas requer conexão com a internet para sincronização em tempo real durante sessões com múltiplos usuários, possuindo três perfis de usuário: Administrador, Maestro e Músico, cada um com funcionalidades específicas.

4.1 DESENVOLVIMENTO DO SISTEMA

O sistema foi desenvolvido para que o armazenamento das partituras e agendamento dos eventos possa ser off-line, armazenando localmente as músicas com as partituras e o arranjo da música, que também armazena os agendamentos.

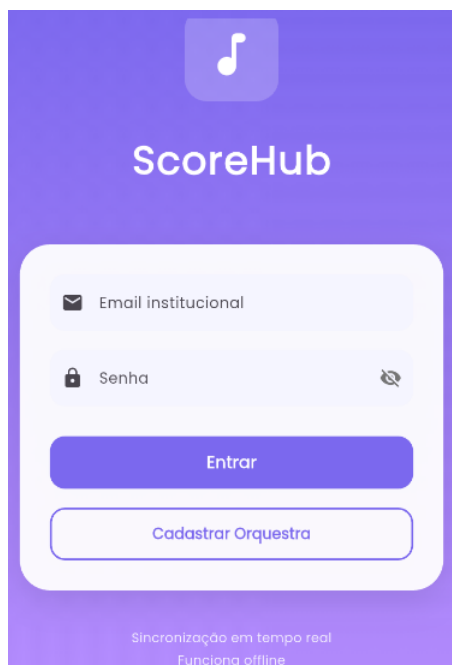
4.1.1 Ambiente do Sistema

A seguir, são apresentadas as telas da aplicação desenvolvida, ilustrando o funcionamento e as interações do usuário com o sistema.

a) Tela de Login

A Figura 1 apresenta a tela de autenticação, utilizada para cadastro da orquestra e acesso de usuários.

Figura 1 - Tela de Login e Cadastro



Fonte: Do autor.

b) Tela de Cadastro da Orquestra

A Figura 2 apresenta o cadastro institucional com nome da organização, dados do administrador (nome, e-mail) e senha.

Figura 2 - Tela de Cadastro da Orquestra

← Cadastro da Orquestra

Cadastrar Orquestra

Crie sua conta de administrador

Nome da Organização

Nome do Administrador

Email do Administrador

Senha

Confirmar Senha

CADASTRAR ORQUESTRA

Fonte: Do autor.

c) Tela de Cadastro de Usuário

A Figura 3 apresenta o cadastro de usuários. Em (a), para músicos: nome, e-mail, naípe, instrumento, voz, clave e senha. Em (b), para maestros: nome, e-mail e senha.

Figura 3 - Tela de Cadastro de Usuário

Cadastrar Músico

MÚSICO

MAESTRO

Nome Completo

Email

Naípe

Violão

Instrumento

Trombone

Voz

1ª Trombone

Clave de Fô

Clave natural do instrumento (origem)

Senha

CADASTRAR

(a)

Cadastrar Maestro

MÚSICO

MAESTRO

Nome Completo

Email

Senha

CADASTRAR

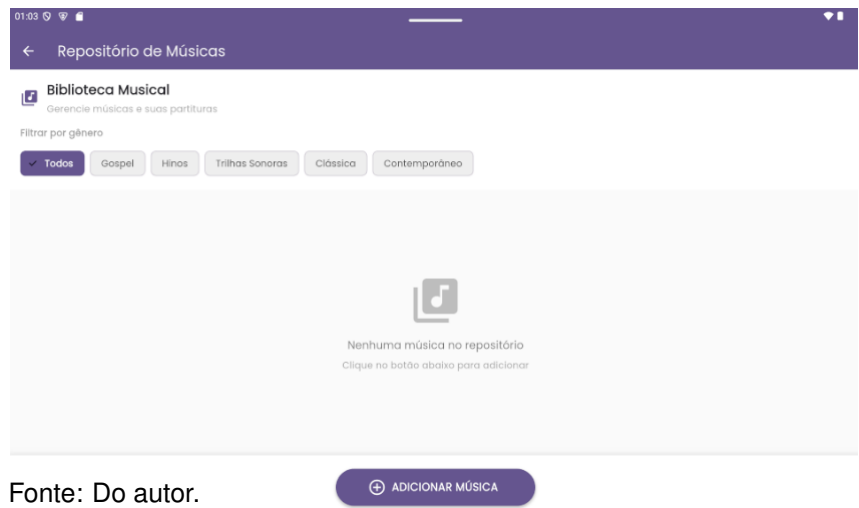
(b)

Fonte: Do autor.

d) Tela do Repositório de Músicas

A Figura 4 apresenta o repositório de músicas com filtros por gênero e acesso às músicas cadastradas.

Figura 4 - Tela do Repositório



Fonte: Do autor.

e) Tela Cadastro de Música

A Figura 5 apresenta o formulário de cadastro com título, compositor, arranjador, gênero e descrição da música.

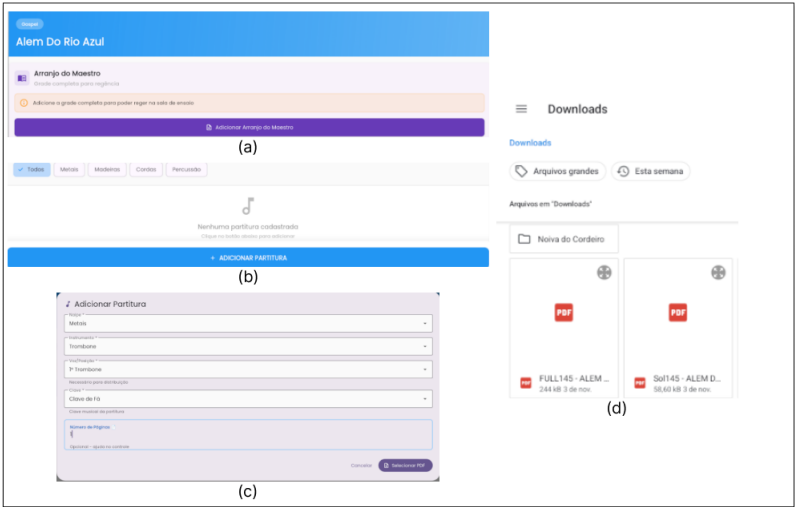
Figura 5 - Tela de Cadastro da Música

Fonte: Do autor.

f) Tela da Música

A Figura 6 demonstra o upload de partituras. Em (a) adiciona-se o arranjo, (b) abre o formulário de partituras, (c) informa-se naípe, instrumento, voz, clave e páginas, e (d) seleciona-se o arquivo PDF.

Figura 6 - Tela da Música

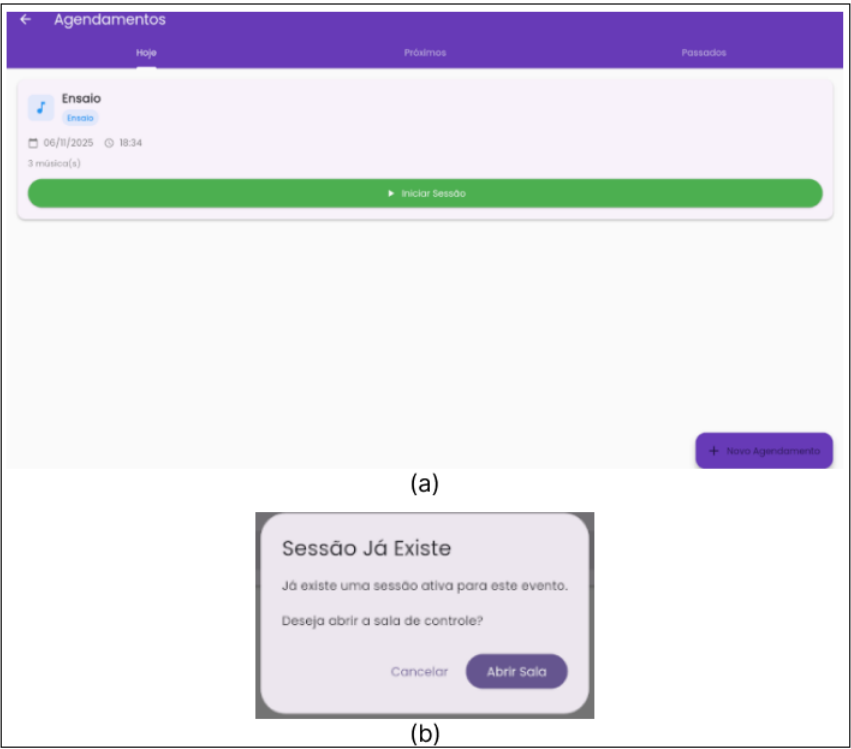


Fonte: Do autor.

g) Tela de Agendamentos

A Figura 7 apresenta a gestão de eventos. Em (a), filtros temporais (Hoje, Próximos, Passados) e botão para novos agendamentos. Em (b), janela para iniciar sessão e acessar a sala.

Figura 7 - Tela de Agendamentos

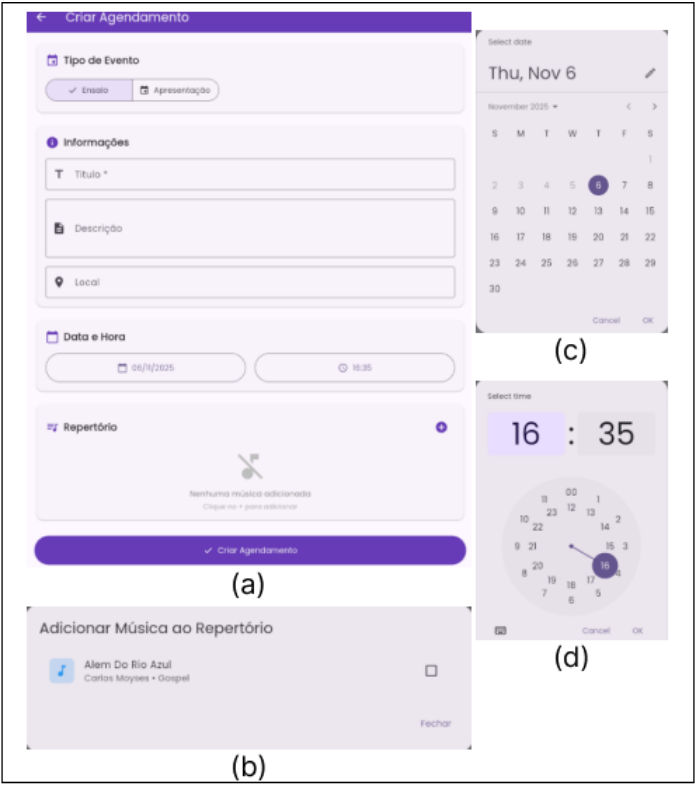


Fonte: Do autor.

h) Tela de Cadastro de Agendamento

A Figura 8 apresenta o formulário de eventos. Em (a), tipo (Ensaio/Apresentação) (c) seleção de data, (d) horário, e (b) adição de músicas ao repertório.

Figura 8 - Tela de Cadastro de Agendamento

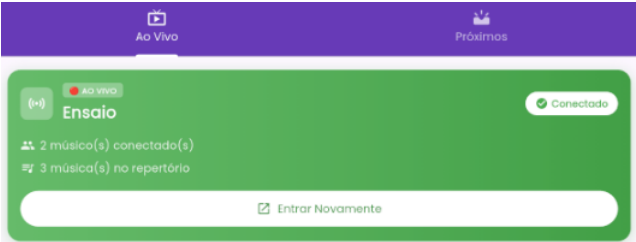


Fonte: Do autor.

i) Tela da Agenda do músico

Esta tela é usada para verificar se possui algum evento agendado, através do Ao Vivo, e Próximos como aparece na Figura 9.

Figura 9 - Tela da Agenda Música

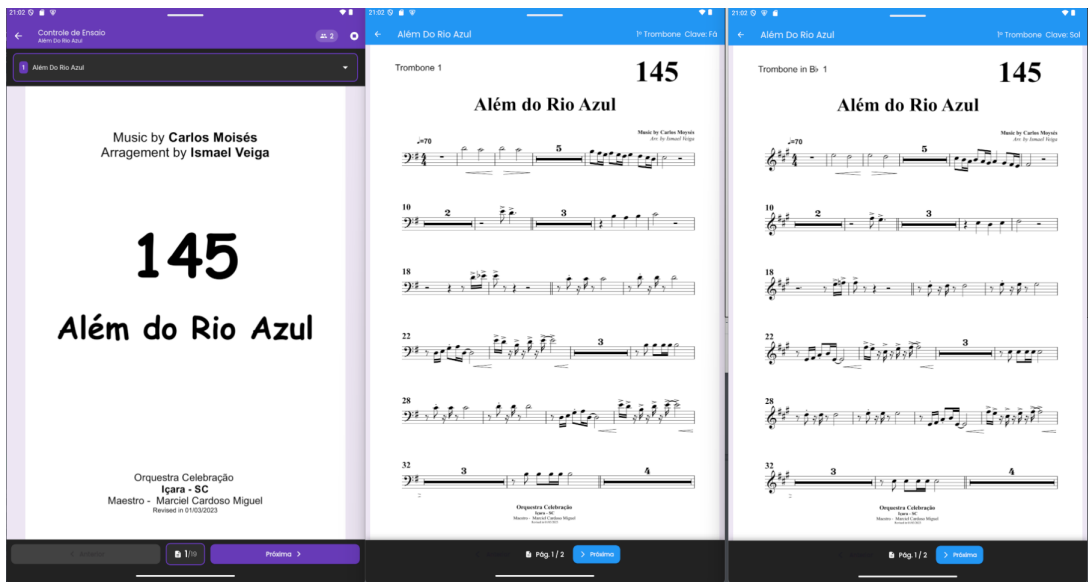


Fonte: Do autor.

j) Tela sala da sessão com os usuários músico

A Figura 10 demonstra a sincronização em tempo real com três usuários conectados (um maestro e dois músicos).

Figura 10 - Tela da Sala da Sessão



Fonte: Do autor.

4.1.2 Ambiente de Desenvolvimento

A aplicação foi desenvolvida utilizando o framework Flutter com a linguagem Dart, organizada na IDE IntelliJ. Utilizou-se o Firebase Firestore para manipulação e gerenciamento local dos dados e o Supabase para armazenar os arquivos PDF das partituras. Para os testes, foram utilizados o simulador do Android Studio (tablet Android versão 15.0) e um tablet Galaxy Tab A9. O ambiente de desenvolvimento utilizou Windows 11 Pro, 16GB RAM, AMD Ryzen 5 5600X e NVIDIA GeForce RTX 3060. Foram utilizados arranjos e partituras da orquestra Celebração da igreja Assembleia de Deus de Içara, com permissão do maestro Marciel C. Miguel.

4.1.3 Banco de Dados

O armazenamento foi estruturado com Firebase Firestore, aproveitando sua natureza NoSQL e capacidade de sincronização em tempo real. O Firestore armazena dados estruturados da orquestra, administrador, usuários músicos (nome, e-mail, naípe, instrumento, voz, clave), repositório de músicas com links de acesso, e dados dos agendamentos com repertórios. O Supabase foi integrado para armazenamento dos arquivos PDF, com cada registro no Firestore contendo o link correspondente, otimizando o desempenho ao separar dados textuais das mídias.

4.2 METODOLOGIA DE AVALIAÇÃO

A validação da aplicação móvel ScoreHub foi realizada através de uma Avaliação Heurística de Usabilidade baseada nas 10 heurísticas propostas por (Nielsen, 1994), combinada com Testes Funcionais de Caixa Preta para verificação da corretude do sistema. O método adotado foi o Walkthrough Cognitivo, onde o avaliador simula o percurso de diferentes perfis de usuários (administrador, maestro e músico) para identificar problemas de usabilidade e falhas funcionais.

4.2.1 Avaliação Heurística de Nielsen

A Avaliação Heurística é um método de inspeção de usabilidade onde um avaliador examina a interface do sistema utilizando princípios de design reconhecidos, chamados de heurísticas (Nielsen, 1994). As 10 heurísticas de usabilidade utilizadas são:

H1 - Visibilidade do Status do Sistema: Manter os usuários informados sobre o que está acontecendo;

H2 - Correspondência entre Sistema e Mundo Real: Usar linguagem familiar ao usuário;

H3 - Controle e Liberdade do Usuário: Fornecer opções de desfazer e refazer;

H4 - Consistência e Padrões: Manter padrões em toda a interface;

H5 - Prevenção de Erros: Design que previne problemas;

H6 - Reconhecimento ao invés de Memorização: Tornar objetos e ações visíveis;

H7 - Flexibilidade e Eficiência de Uso: Atalhos para usuários experientes;

H8 - Estética e Design Minimalista: Evitar informações irrelevantes;

H9 - Ajudar Usuários a Reconhecer e Recuperar Erros: Mensagens claras e soluções;

H10 - Ajuda e Documentação: Fornecer informações quando necessário.

4.2.2 Classificação de Severidade

Os problemas identificados foram classificados segundo a escala de severidade proposta por Nielsen (Nielsen, 1995), que considera frequência, impacto e persistência:

- 1) **0** (Não é problema): Não afeta a operação do sistema
- 2) **1** (Cosmético): Problema estético que não impacta tarefas
- 3) **2** (Menor): Baixa prioridade, usuário consegue contornar facilmente
- 4) **3** (Maior): Alta prioridade, dificulta significativamente a tarefa
- 5) **4** (Catastrófico): Impede conclusão da tarefa ou causa perda de dados

4.3 Procedimento de Avaliação

4.3.1 Simulação de Usuários

O avaliador simulou três perfis:

- 1) Administrador (cadastro e gerenciamento de usuários);
- 2) Maestro (gerenciamento de músicas, eventos e controle de sessões);
- 3) Músico (visualização de partituras e acompanhamento em tempo real).

4.3.2 Funcionalidades Avaliadas

Foram avaliadas 4 funcionalidades:

- 1) Cadastro de Usuários;
- 2) Armazenamento de Músicas;
- 3) Criação de Eventos;
- 4) Sincronização em Tempo Real.

4.3.3 Execução dos Testes

Para cada funcionalidade foram executados cenários cobrindo casos de sucesso, validação e erro, classificados como Passou ou Falhou, identificando heurística violada e severidade.

5 RESULTADOS E DISCUSSÃO

Este capítulo apresenta os resultados obtidos durante a avaliação funcional e de usabilidade da aplicação ScoreHub. A validação foi conduzida através de testes sistemáticos organizados por funcionalidade, utilizando o método de Avaliação Heurística de Nielsen combinado com Testes Funcionais de Caixa Preta. Cada funcionalidade crítica do sistema foi submetida a cenários de teste específicos, permitindo identificar problemas de usabilidade e verificar a corretude do comportamento esperado.

5.1 CADASTRO DE USUÁRIOS

A funcionalidade de Cadastro de Usuários foi avaliada através de 5 cenários de teste, abrangendo a criação, edição e inativação de usuários músicos no sistema. A Tabela 1 apresenta os resultados obtidos.

Tabela 1 - Resultados dos Testes de Cadastro de Usuários

ID	Cenário	Resultado	Severidade
A1	Cadastrar músico completo	Passou	0
A2	Cadastrar sem email	Passou	0
A3	Cadastrar com email inválido	Falhou	3
A4	Editar músico existente	Passou	0
A5	Inativar músico	Passou	0
Taxa de Sucesso		80,0%	

Fonte: Elaborado pelo autor.

Dos 5 cenários testados, 4 apresentaram comportamento conforme esperado, resultando em uma taxa de sucesso de 80,0%. O único problema identificado (cenário A3) refere-se à validação incompleta do campo email, que aceita endereços com domínio incompleto, como `nome@g.com`.

Este problema foi classificado como severidade 3 (maior), pois pode resultar em emails inválidos cadastrados no sistema, impedindo a comunicação com usuários e exigindo correção manual posterior. A falha viola a heurística H5 (Prevenção de Erros) de Nielsen (Nielsen, 1994), indicando ausência de validação adequada antes do salvamento dos dados. Esta deficiência na validação contrasta com o comportamento correto observado em outros campos obrigatórios do formulário de cadastro.

As demais funcionalidades (criação completa, edição e inativação de usuários) operaram corretamente, incluindo validações obrigatórias (cenário A2) e feedback visual apropriado durante a inativação de usuários (cenário A5). O cenário A1 demonstrou que o cadastro completo de músicos, incluindo todos os campos necessários como naipe, instrumento, voz e clave, funciona adequadamente. Já o cenário A4 confirmou que a edição de informações de músicos existentes opera sem problemas, permitindo atualizações quando necessário.

A taxa de sucesso de 80,0% indica boa qualidade funcional desta funcionalidade. O problema identificado deve ser corrigido prioritariamente antes do lançamento em produção. A solução recomendada é implementar validação mais robusta do formato de email, verificando não apenas a presença do caractere @, mas também a existência de um domínio completo

(ex: .com, .com.br, .org). Isto pode ser implementado utilizando expressões regulares (regex) ou bibliotecas de validação que já são disponíveis no Flutter, como o pacote `email_validator`.

5.2 ARMAZENAMENTO DE MÚSICAS

A funcionalidade de Armazenamento de Músicas foi avaliada através de 5 cenários de teste, abrangendo upload, visualização e exclusão de partituras em formato PDF. A Tabela 2 apresenta os resultados obtidos.

Tabela 2 - Resultados dos Testes de Armazenamento de Músicas

ID	Cenário	Resultado	Severidade
M1	Upload de PDF válido	Passou	0
M2	Upload de arquivo não-PDF	Passou	0
M3	Upload de PDF grande	Passou	0
M4	Visualizar partitura	Passou	0
M5	Excluir partitura	Passou	0
Taxa de Sucesso		100%	

Fonte: Elaborado pelo autor.

Todos os 5 cenários testados apresentaram comportamento conforme esperado, resultando em uma taxa de sucesso de 100%. A funcionalidade demonstrou excelente implementação de prevenção de erros e feedback ao usuário.

O cenário M2 merece destaque pela implementação de validação no nível do explorador de arquivos, que restringe a seleção apenas a arquivos PDF, impedindo que o usuário cometa erro ao tentar enviar arquivos em formatos não suportados. Esta abordagem aplica a heurística H5 (Prevenção de Erros) no nível mais eficaz possível.

O sistema também demonstrou robustez ao processar arquivos grandes (M3), mantendo feedback visual durante o upload (M1) através de barra de progresso e mensagens claras de sucesso. A funcionalidade de exclusão (M5) implementa corretamente confirmação antes de remover arquivos, prevenindo exclusões acidentais.

5.3 CRIAÇÃO DE EVENTOS

A funcionalidade de Criação de Eventos foi avaliada através de 6 cenários de teste, abrangendo a criação, edição e gerenciamento de eventos musicais (ensaios e apresentações) com seus respectivos repertórios. A Tabela 3 apresenta os resultados obtidos.

Tabela 3 - Resultados dos Testes de Criação de Eventos

ID	Cenário	Resultado	Severidade
E1	Criar evento completo	Passou	0
E2	Criar evento sem data	Passou	0
E3	Adicionar 15 músicas ao repertório	Passou	0
E4	Reordenar músicas no repertório	Passou	0
E5	Remover música do repertório	Passou	0
E6	Editar evento existente	Falhou	3
Taxa de Sucesso		83,3%	

Fonte: Elaborado pelo autor.

Dos 6 cenários testados, 5 apresentaram comportamento conforme esperado, resultando em uma taxa de sucesso de 83,3%. O problema identificado (cenário E6) refere-se à ausência de funcionalidade para editar ou excluir eventos após sua criação.

A funcionalidade demonstrou excelente implementação de prevenção de erros. O cenário E2 revelou que o sistema utiliza valores padrão inteligentes para campos obrigatórios: a data é preenchida automaticamente com o dia atual e o horário com uma hora futura próxima, tornando impossível criar eventos sem esses dados essenciais. Adicionalmente, o calendário de seleção de datas bloqueia datas passadas, prevenindo a criação de eventos com datas retroativas. O sistema também exige título e pelo menos uma música no repertório, garantindo que eventos sejam criados com informações mínimas necessárias.

O gerenciamento de repertório (cenários E3, E4 e E5) apresentou funcionalidade completa e intuitiva. O sistema permite adicionar múltiplas músicas facilmente (testado com 15 músicas sem problemas de performance), reordenar através de interface *drag-and-drop* (arrastar e soltar), e remover músicas do repertório sem necessidade de confirmação — decisão apropriada considerando que a remoção é facilmente reversível e não exclui dados permanentemente do sistema.

No entanto, o cenário E6 revelou uma limitação significativa: após a criação do evento, não é possível editá-lo ou excluí-lo. Este problema foi classificado como severidade 3 (maior), pois impede que o usuário corrija erros em informações essenciais como data, horário, local ou repertório. A impossibilidade de correção viola a heurística H3 (Controle e Liberdade do Usuário) de Nielsen (Nielsen, 1994), que estabelece que usuários devem ter a capacidade de desfazer ações e corrigir erros.

A ausência desta funcionalidade tem impacto alto no uso prático do sistema: se um maestro cria um evento com horário incorreto (por exem-

plo, 19:00 ao invés de 20:00), não há forma de corrigir o erro, sendo necessário manter o evento incorreto ou, potencialmente, criar um novo evento e comunicar manualmente aos músicos sobre a mudança que compromete a centralização da informação no sistema.

Duas soluções podem resolver este problema. A primeira e mais completa seria implementar funcionalidade de edição de eventos, permitindo alteração de data, horário, local e repertório, com validação apropriada (por exemplo, impedir mudança de data para o passado ou mudança de horário após o início do evento). A segunda, mais simples mas menos ideal, seria implementar apenas a funcionalidade de exclusão de eventos, permitindo que o usuário delete eventos incorretos e os recrie com informações corretas. Ambas as soluções devem incluir confirmação antes da ação destrutiva, aplicando a heurística H5 (Prevenção de Erros).

Apesar da limitação identificada, a taxa de sucesso de 83,3% indica boa qualidade funcional. As funcionalidades de criação e gerenciamento de repertório estão bem implementadas, com excelente aplicação de prevenção de erros e interface intuitiva. A correção do problema E6 é recomendada antes do lançamento em produção para garantir experiência de usuário completa e flexível.

6 CONCLUSÃO

Este trabalho apresentou o desenvolvimento e avaliação do ScoreHub, uma aplicação móvel para digitalização e gerenciamento de partituras musicais em orquestras. O sistema foi concebido para substituir partituras físicas, centralizar a gestão de repertório e facilitar a coordenação entre maestros e músicos durante ensaios e apresentações. A aplicação desenvolvida utiliza Flutter como framework principal, Firebase Firestore para sincronização em tempo real dos dados e Supabase para armazenamento dos arquivos PDF das partituras.

A validação do sistema foi realizada através de Avaliação Heurística baseada nos princípios de usabilidade de Nielsen (Nielsen, 1994), combinada com Testes Funcionais de Caixa Preta. O método adotado foi o Walkthrough Cognitivo, onde o avaliador simulou diferentes perfis de usuários do sistema para identificar problemas de usabilidade e falhas funcionais. Foram executados 16 cenários de teste distribuídos entre três funcionalidades principais: Cadastro de Usuários, Armazenamento de Músicas e Criação de Eventos. A Tabela 4 apresenta a consolidação dos resultados obtidos.

Tabela 4 - Resultados Gerais dos Testes Funcionais

Funcionalidade	Total	Passou	Falhou	Taxa (%)
Cadastro de Usuários	5	4	1	80,0
Armazenamento de Músicas	5	5	0	100,0
Criação de Eventos	6	5	1	83,3
TOTAL	16	14	2	87,5

Fonte: Elaborado pelo autor.

A avaliação resultou em uma taxa de sucesso geral de 87,5%, com 14 cenários apresentando comportamento conforme esperado e apenas 2 cenários apresentando falhas de severidade maior e nenhum problema catastrófico. A distribuição de severidade dos problemas, apresentada na Tabela 5, revela que 87,5% dos cenários testados não apresentaram problemas de usabilidade.

Tabela 5 - Distribuição de Severidade dos Problemas

Severidade	Quantidade	Percentual (%)
0 - Não é problema	14	87,5
1 - Cosmético	0	0,0
2 - Menor	0	0,0
3 - Maior	2	12,5
4 - Catastrófico	0	0,0
TOTAL	16	100,0

Fonte: Elaborado pelo autor.

Os dois problemas identificados foram classificados como severidade 3 (maior), representando 12,5% dos casos, com média de severidade de 0,375. Não foram identificados problemas catastróficos (severidade 4), indicando que nenhuma funcionalidade impede completamente o uso do sistema ou causa perda de dados, demonstrando que as funcionalidades essenciais do sistema operam de forma adequada para atender às necessidades básicas de maestros e músicos durante ensaios e apresentações. A análise das heurísticas violadas, apresentada na Tabela 6, revela que apenas duas das dez heurísticas de Nielsen foram violadas: H5 (Prevenção de Erros) e H3 (Controle e Liberdade do Usuário), cada uma com 50% das violações. Este resultado indica que as demais oito heurísticas foram atendidas satisfatoriamente pelo sistema, incluindo aspectos como visibilidade do status, consistência de padrões e estética minimalista.

Tabela 6 - Heurísticas de Usabilidade Violadas

Heurística	Violações	Percentual (%)
H5 - Prevenção de Erros	1	50,0
H3 - Controle e Liberdade do Usuário	1	50,0
TOTAL	2	100,0

Fonte: Elaborado pelo autor.

A Tabela 7 detalha os dois problemas de severidade maior identificados durante a avaliação.

Tabela 7 - Problemas Críticos Identificados

ID	Cenário	Sev.	Descrição
A3	Cadastrar com email inválido	3	Sistema aceita emails com domínio incompleto (ex: nome@g.com)
E6	Editar evento existente	3	Sistema não permite editar ou excluir eventos após criação

Fonte: Elaborado pelo autor.

O primeiro problema (A3) refere-se à validação incompleta de email, que aceita endereços com domínio incompleto como `nome@g.com`. A solução recomendada é implementar validação robusta utilizando expressões regulares ou bibliotecas especializadas disponíveis no Flutter, como o pacote `email_validator`, verificando a existência de domínio completo e caracteres válidos. O segundo problema (E6) trata da impossibilidade de editar ou excluir eventos após sua criação, impedindo correção de erros em informações como data, horário ou repertório. A solução recomendada é implementar funcionalidade de edição de eventos com validações apropriadas, ou minimamente funcionalidade de exclusão com confirmação. Ambos os problemas são tecnicamente viáveis de corrigir e não requerem mudanças arquiteturais significativas. Sua correção elevaria a taxa de sucesso para 100% e a média de severidade para 0,0.

O desenvolvimento da aplicação apresentou resultados satisfatórios em relação aos objetivos propostos, demonstrando de forma clara o potencial de utilização de uma ferramenta digital voltada ao gerenciamento de repertórios e partituras para orquestras. Este trabalho contribui para a área de tecnologia musical e gestão orquestral ao desenvolver solução completa e funcional para digitalização de partituras musicais, incluindo gestão de usuários, armazenamento de arquivos PDF e organiza-

ção de eventos com repertório; validar quantitativamente a qualidade funcional através de avaliação heurística estruturada, demonstrando taxa de sucesso de 87,5% e média de severidade de 0,375; implementar boas práticas de usabilidade como prevenção de erros (valores padrão inteligentes, validação no explorador de arquivos, bloqueio de datas passadas) e feedback visual consistente; demonstrar viabilidade técnica de sistema multiplataforma utilizando Flutter, Firebase Firestore e Supabase para gestão orquestral; contribuir para sustentabilidade ambiental através da redução no uso de papel em orquestras; e comprovar a praticidade e aplicabilidade no armazenamento de arquivos, gerenciamento de eventos agendados e organização de partituras segundo informações do usuário.

A avaliação realizada apresenta algumas limitações que devem ser consideradas: a avaliação heurística foi conduzida por um único avaliador, o próprio desenvolvedor do sistema, enquanto (Nielsen, 1994) recomenda 3-5 avaliadores independentes para identificar 75-90% dos problemas de usabilidade, enquanto um único avaliador identifica aproximadamente 35%; a metodologia utilizou simulação de usuários (walkthrough cognitivo) ao invés de testes com usuários reais (maestros e músicos de orquestras), sendo que usuários reais podem identificar problemas não antecipados durante a simulação; os testes foram realizados em ambiente de desenvolvimento controlado, não em condições reais de uso durante ensaios ou apresentações musicais; e o sistema foi testado apenas com dispositivos Android, sem validação em dispositivos iOS.

Como proposta para trabalhos futuros, sugere-se as adições de um suporte a arquivos Musical Instrument Digital Interface para demonstração de execução musical, incluindo tom, tempo e outras dinâmicas musicais, integrando também o pedal *Bluetooth* para a troca de páginas, permitindo navegação sem o uso das mãos durante a execução musical e o uso de caneta digital para anotações nas partituras durante ensaios ou apresentações.

REFERÊNCIAS

ALMEIDA, R. R. d. **Tecnologias para o desenvolvimento de aplicações multiplataforma: Um estudo sobre os frameworks React Native e Flutter**. Porto Alegre, RS: [s.n.], 2019. Acesso em: 17 set. 2025. Disponível em: <<https://raam.alcidesmaya.com.br/index.php/projetos/article/view/54/54>>.

FAVA, F. B. **Ferramenta para Visualização do Consumo de Recursos de Hardware para Aplicações Flutter**. Alegrete: [s.n.], 2023. Acesso em: 19

jun. 2025. Disponível em: <<https://dspace.unipampa.edu.br/bitstream/riu/8818/1/TCC%20-%20Felipe%20Bedinotto%20Fava%20-%202023.pdf>>.

HUYNH, H. **VAASA Student Guide: Mobile Application in Flutter**. 2023. Acesso em: 19 jun. 2025. Disponível em: <<https://www.theseus.fi/bitstream/handle/10024/800119/Hong%20Huynh%20-%201900318%20thesis.pdf?sequence=2>>.

LIMA, R. D. **Música e Tecnologia: Recursos Digitais como Ferramentas Auxiliares ao Rendimento Pessoal e à Dinâmica de Ensaios Camerísticos - Relato Autoetnográfico**. 2024. Acesso em: 24 set. 2024. Disponível em: <<https://repositorio.ufba.br/bitstream/ri/39433/1/TC%20Rodolfo%20Lima%20-%20Mestrado%20Profissional%20em%20M%c3%basica%20-%20UFBA%20%e2%80%a2%2028.04.2024.pdf>>.

MOBILESHEETS. **MobileSheets – Sheet Music Reader App**. 2025. Acesso em: 19 jun. 2025. Disponível em: <<https://www.zubersoft.com/mobilesheets/>>.

NEWZIK. **Newzik – Beyond Sheet Music**. 2025. Acesso em: 19 jun. 2025. Disponível em: <<https://newzik.com/en/>>.

NIELSEN, J. **10 Usability Heuristics for User Interface Design**. 1994. Acesso em: 2 dez. 2025. Disponível em: <<https://www.nngroup.com/articles/ten-usability-heuristics/>>.

NIELSEN, J. **Severity Ratings for Usability Problems**. 1995. Acesso em: 2 dez. 2025. Disponível em: <<https://www.nngroup.com/articles/how-to-rate-the-severity-of-usability-problems/>>.

SALCEDO, D. A. et al. **Preservação documental das partituras manuscritas de frevo da Banda Capitão Zuzinha de Pernambuco**. 2022. 1-16 p. Acesso em: 19 jun. 2025. Disponível em: <<https://dialnet.unirioja.es/servlet/articulo?codigo=8403433>>.

TEIXEIRA, G. F. et al. **Fundamentos de Flutter e Dart para Desenvolvimento de Apps Móveis**. Espírito Santo, Brasil: Edifes, 2024. 143 p. Acesso em: 9 jun. 2025. Disponível em: <https://play.google.com/store/books/details/GIOVANY_FROSSARD_TEIXEIRA_Fundamentos_de_Flutter_e?id=vhEMEQAAQBAJ>.

UFSM. **Manual de Catalogação de Partituras no Padrão CCAA2R2002 e Formato Bibliográfico MARC 21**. 2024. Acesso em: 21 abr. 2025. Disponível em: <<https://repositorio.ufsm.br/bitstream/handle/1/33257/MANUALdePARTITURAS.pdf?sequence=1&isAllowed=y>>.