



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

LUCAS CHRISTOPHER DE SOUZA SILVA

BACKSTAGE:

**ORGANIZE SEU EVENTO DE FORMA LEVE E INTUITIVA, SEM
IMPEDIMENTOS E PREOCUPAÇÕES DESNECESSÁRIAS**

CAMPINA GRANDE - PB

2021

LUCAS CHRISTOPHER DE SOUZA SILVA

BACKSTAGE:

**ORGANIZE SEU EVENTO DE FORMA LEVE E INTUITIVA, SEM
IMPEDIMENTOS E PREOCUPAÇÕES DESNECESSÁRIAS**

**Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado
em Ciência da Computação do
Centro de Engenharia Elétrica e
Informática da Universidade Federal
de Campina Grande, como requisito
parcial para obtenção do título de
Bacharel em Ciência da
Computação.**

Orientadora: Professora Dra. Eliane Cristina de Araújo.

CAMPINA GRANDE - PB

2021



S586b Silva, Lucas Christopher de Souza.

Backstage: organize seu evento de forma leve e intuitiva, sem impedimentos e preocupações desnecessárias. / Lucas Christopher de Souza. - 2021.

12 f.

Orientadora: Profa. Dra. Eliane Cristina de Araújo.
Trabalho de Conclusão de Curso - Artigo (Curso de Bacharelado em Ciência da Computação) - Universidade Federal de Campina Grande; Centro de Engenharia Elétrica e Informática.

1. Backstage - aplicativo. 2. Evento. 3. Organização de eventos. 4. Aplicativo para organização e eventos. I. Araújo, Eliane Cristina de. II. Título.

CDU:004(045)

Elaboração da Ficha Catalográfica:

Johnny Rodrigues Barbosa
Bibliotecário-Documentalista
CRB-15/626

LUCAS CHRISTOPHER DE SOUZA SILVA

BACKSTAGE:

**ORGANIZE SEU EVENTO DE FORMA LEVE E INTUITIVA, SEM
IMPEDIMENTOS E PREOCUPAÇÕES DESNECESSÁRIAS**

Trabalho de Conclusão Curso
apresentado ao Curso Bacharelado em
Ciência da Computação do Centro de
Engenharia Elétrica e Informática da
Universidade Federal de Campina
Grande, como requisito parcial para
obtenção do título de Bacharel em
Ciência da Computação.

BANCA EXAMINADORA:

Professora Dra. Eliane Cristina de Araújo
Orientadora – UASC/CEEI/UFCG

Professor Dr. Fábio Jorge Almeida Moraes
Examinador – UASC/CEEI/UFCG

Professor Dr. Tiago Lima Massoni
Professor da Disciplina TCC – UASC/CEEI/UFCG

Trabalho aprovado em: 20 de outubro de 2021.

CAMPINA GRANDE - PB

ABSTRACT

Organizing an event is always a challenge. Everything behind the scenes requires caution, a lot of planning and if there is a lack of commitment or negligence in any aspect, we can see everything going downhill and feel the consequence of things not going as we expected. To organize events of relevant size, such as large conferences, concerts and even religious ceremonies that involve many people, it is common to resort to some tool that helps in the management of everything that is of interest to us. That's why Backstage was created, a mobile application that provides an easy experience in managing large events of an organization that has one or more work teams active in various aspects, such as management, ceremonial, media, music, among others, centralizing the whole information regarding times, schedules, media for consumption and availability of team members for activities organized by the organization.

Backstage: organize seu evento de forma leve e intuitiva, sem impedimentos e preocupações desnecessárias

Lucas Christopher

lucas.christopher.silva@ccc.ufcg.edu.br
Universidade Federal de Campina Grande
Campina Grande, Paraíba, Brasil

Eliane Araújo

eliane@computacao.ufcg.edu.br
Universidade Federal de Campina Grande
Campina Grande, Paraíba, Brasil

RESUMO

Organizar um evento é sempre um desafio. Tudo por trás dos bastidores exige cautela, muito planejamento e caso haja falta de compromisso ou negligência em algum aspecto, podemos ver tudo indo por água abaixo e sentir a consequência das coisas não saírem como esperávamos. Para organizar eventos de porte relevante, como grandes conferências, shows e até cerimônias religiosas que envolvem muitas pessoas, é comum recorrer-se a alguma ferramenta que traga auxílio na gestão de tudo o que é do nosso interesse. É para isto que o *Backstage* surgiu, uma aplicação mobile que auxilia a organização dos seus eventos, gerenciando pessoas, cronograma e demais aspectos.

Palavras-chave

Backstage, evento, organizar, compromisso.

1. INTRODUÇÃO

A tecnologia sem dúvidas veio para mudar as nossas vidas. A cada dia que se passa, ela está mais presente em nosso dia a dia, em nossa rotina, nos nossos afazeres, compromissos e em praticamente tudo o que interagimos. Dessa forma, a utilização de recursos tecnológicos que facilitem nossa rotina é de toda forma bem vinda.

A organização é um fator primordial para todo caso de sucesso e, quando nos referimos a grandes eventos, é de fato o primeiro ponto a ser abordado. A divisão de tarefas, alocação de pessoas, distribuição de funções ao longo de todo o evento desde os bastidores até ao momento da ação, precisa ser excelente, interativa, rápida, dinâmica e consistente, de tal forma que as dores de cabeça sejam minimizadas e a leveza em proporcionar uma boa experiência aos envolvidos seja levantada.

Diante do que foi dito, o objetivo deste trabalho é descrever a criação de uma aplicação móvel que proporcione uma experiência fácil em gerenciar grandes eventos de uma organização que possuem uma ou mais equipes de trabalho atuantes em diversas vertentes, como gestão, cerimonial, mídia, música, dentre outros, centralizando toda a informação referente a horários, cronogramas, mídias para consumo e disponibilidade dos membros de equipe para atividades organizadas pela organização.

1.1 Soluções conhecidas

Uma das soluções apresentadas neste âmbito de organização de eventos é o Timbragem, “*aplicação que permite uma organização planejar múltiplos eventos, com diversas finalidades, através da web ou dos aplicativos disponíveis nas lojas*” [6].

Contudo, o *app* se mostra limitado: na forma de passar as informações para seus usuários (o qual a faz via e-mail, apenas); na forma com que oferece a opção de bloqueios de datas (caso algum participante de uma organização esteja remanejado em mais de uma equipe, como por exemplo, música e mídia, considere que músicos precisam ensaiar e pessoas que trabalham com mídia e edição de fotos precisam de reuniões de alinhamento, tempo para edição de fotos, vídeos e etc; se em uma data em específico ele tiver um compromisso marcado para ambos os times, pela aplicação o usuário não consegue especificar uma data de ausência para qual equipe em específico quer se ausentar); a respeito dos bloqueios de data, você só pode bloquear uma data por vez (ou seja, a marcação das indisponibilidades devem ser feitas uma a uma, não permitindo bloquear intervalos de datas, tornando o registro de indisponibilidades na aplicação uma atividade exaustiva); não exibe um histórico de eventos já passados; e, por fim, apresenta uma

interface vaga e pouco intuitiva, dificultando o uso do aplicativo.

2. SOLUÇÃO

Nesta seção, apresentaremos todos os elementos que compõem a solução em questão.

2.1 Visão geral

O mercado da tecnologia já apresenta uma gama de soluções disponíveis que oferecem o gerenciamento e a organização para compromissos e eventos, disponíveis tanto em plataforma web que apresentam responsividade, como também, soluções mobile. Contudo, as opções disponíveis no mercado apresentam funcionalidades limitadas, indo de um simples bloco de anotações a aplicações que mostram-se ter um formato muito engessado, sem oferecer personalização da parte de seus usuários.

Considerando [1], cerca da metade da população mundial tem um smartphone; [2] relata que temos mais de um smartphone por habitante no Brasil e, segundo [4], os usuários brasileiros passam em média cinco horas do seu dia em aplicações móveis.

Contextualizando, podemos ver a crescente que os aplicativos móveis têm tido no Brasil e no mundo dado ao crescimento nas aquisições de dispositivos móveis, o que nos leva a afirmar que hoje, além de sua importância no mercado, os aplicativos são essenciais para o desenvolvimento tecnológico de uma sociedade contemporânea.

Deste modo, criamos uma aplicação móvel que contém as funcionalidades básicas de um simples organizador de compromissos em agendas, acrescido de demais especificações no qual detalharemos as principais abaixo:

- **Hierarquia de usuários.** Ao se cadastrar no *app*, algumas funcionalidades podem ser diferentes dependendo do nível do usuário. Um usuário *administrador* pode gerenciar e alocar pessoas, elaborar templates para cronograma, disponibilizar links de vídeos, de cifras e outras fontes de mídia, enquanto um *colaborador* pode participar de eventos, participar de equipes e consumir serviços de mídia. Todo usuário pode ser administrador - caso crie sua organização - ou colaborador - caso esteja alocado em uma organização que não seja a dele.

- **Criação de organização e equipes.** Para melhor entendimento, imagine uma grande empresa (organização) que possui várias *squads* (equipes), seja na parte de recursos humanos, desenvolvimento, gestão financeira e etc. A organização representa o todo, enquanto as equipes simbolizam times dentro de uma organização.
- **Alocação em equipes.** Cada usuário pode ser cadastrado em uma equipe com sua respectiva função dentro da organização, seja fazer parte da equipe de música, gestão, em diante.
- **Criação de cronograma e templates.** Os usuários administradores podem criar suas programações e salvá-las como templates, para possíveis usos posteriores. Serve bem para equipes que têm atuações fixas durante uma semana e que funcionam de forma análoga.
- **Registro de disponibilidade/bloqueio em agendas.** Os usuários colaboradores podem registrar sua disponibilidade para os eventos numa forma de escala, registrando a possível presença/ausência em sua respectiva equipe.
- **Consulta de cronograma e mídia.** Os usuários podem consultar a programação do seu evento e também verificar possíveis arquivos de mídia disponibilizados. Para colaboradores que fazem parte de uma equipe de música, seja de uma Igreja, ou direcionados a tocar em um casamento, por exemplo, é interessante centralizar links com as cifras e links com os vídeos das versões das músicas que precisam ser estudadas.
- **Notificações via push.** As alterações efetuadas no cronograma, na mídia ou em quaisquer aspecto relevante do projeto notificará os usuários da organização via *push-notification*, que garante que todos os membros da organização recebam as informações com o menor esforço possível.

Considerando essas funcionalidades, a aplicação torna-se muito útil para gestão de grandes eventos, aproximando as equipes e centralizando a informação para todos.

2.2 Decisões arquiteturais

A respeito das decisões arquiteturais do projeto, fizemos a separação do nosso sistema em duas partes: **back-end** e **front-end**. O primeiro, como o nome sugere, é o que está por trás de uma aplicação final. O *back-end* é a camada que tem por

incumbência gerenciar e persistir os dados no banco de dados que geralmente está alocado em algum servidor. Já o *front-end* está diretamente relacionado com aquilo que podemos interagir. É a parte que o usuário final faz uso para desfrutar da aplicação.

2.3 Ferramentas e tecnologia utilizadas

2.3.1 Back-end

Para a parte do *back-end*, nós utilizamos o **Firestore** [7], uma plataforma *BaaS* (*Backend-as-a-Service*) criada pelo Google que provê a infraestrutura e o *back-end* de uma aplicação de uma forma simples, sem a necessidade de desenvolvimento manual de solução. A ideia do Firestore é oferecer soluções para as funcionalidades mais comuns de uma aplicação, como autenticação, armazenamento, escalabilidade, serviço de notificações, suporte à *analytics*, entre outros. Além disso, oferece grande suporte aos diversos tipos de plataformas (Android, iOS, JS, etc), suíte de testes e monitoramento de desempenho.

A utilização do Firestore traz consigo muita simplicidade e agilidade para o processo de desenvolvimento de aplicações. Para este projeto, dentre a suíte de serviços que o Firestore nos oferece, utilizamos tais:

- Firestore Authentication, que nos concede ferramentas para autenticação de usuário, seja usando senhas, números de telefone ou via provedores de identidade federados, como Google, Facebook e Twitter;
- Firestore Cloud Storage, que disponibiliza um serviço seguro de armazenamento de objetos avançados, simples e econômicos;
- Firestore Cloud Messaging, uma solução para envio de mensagens entre plataformas que permite a entrega confiável de mensagens sem custos;

2.3.2 Front-end

Para o *front-end*, utilizamos a tecnologia **Flutter** [9], que é o kit de ferramentas de UI do Google para construção de aplicações mobile, web, desktop e embarcadas a partir de uma única fonte de código, concedendo alta produtividade. O Flutter traz consigo o legado forte de desenvolvimento ágil em criar UIs expressivas, flexíveis e com um desempenho nativo [10].

O Flutter funciona em cima de *widgets*. Tudo em Flutter são *widgets*: botões, telas, componentes, tudo. Isso é bom, afinal, como os *widgets* fazem

parte do aplicativo e não da plataforma, os problemas de compatibilidade em diferentes versões do sistema operacional são minimizados ao extremo, o que significa menos tempo gastos em testes.

Desde seu lançamento em 2017, o Flutter vem tomando cada vez mais o mercado mobile com uma comunidade cada vez mais ativa e crescente. Por ser uma plataforma de código aberto, suas documentações são extensas e completas, sempre atualizadas. Para envolver a comunidade, o Google oferece suporte ao Pub [11], repositório que centraliza os pacotes e bibliotecas voltados para Dart e Flutter. Como o próprio nome sugere, “pub” é uma abreviação do inglês *public house*, que trata-se de um ambiente mais familiar e comunitário, onde as pessoas se reúnem para se confraternizarem.

2.3.2.1 Arquitetura do Front-end

Como já descrito, toda a responsabilidade do nosso *back-end* ficou para o Firestore. Deste modo, nos resta estabelecer uma arquitetura para o *front-end*. Assim sendo, foi utilizado o *design pattern* **MVVM** (*Model-View-Viewmodel*) para implementação da solução. O MVVM é útil para permitir a verdadeira separação entre a camada de visualização e a lógica de negócios. Em termos reais, o que isso significa é que, quando a camada de modelo precisa ser alterada, ela pode ser alterada facilmente sem a necessidade de alteração na camada de visualização e vice-versa. Descrevendo cada camada individualmente, como temos na Figura 1, temos:

- **Camada de modelo (*Model*)**. A camada de modelo no MVVM representa a única fonte de verdade que carrega os dados da aplicação, seja dados de busca em tempo real ou referentes as consultas relacionadas ao banco de dados. Esta camada pode conter lógica e validação de regras de negócios, e interage diretamente com a camada de *Viewmodel* fornecendo dados como resposta.
- **Camada de modelo-visão (*Viewmodel*)**. É o mediador entre a visualização e o modelo. Esta camada aceita eventos do usuário efetuados na *View* e solicita ao *Model* uma resposta. Após recebê-la, notifica os dados recebidos para a *View*.
- **Camada de visualização (*View*)**. Aqui é onde temos os elementos visuais no qual o usuário consegue ver e interagir com os *widgets* mostrados em tela, seja botões, campos de texto, etc.

Os três principais benefícios que fluem ao se utilizar o *MVVM* em uma aplicação mobile são:

- **Manutenibilidade:** com o passar do tempo, a base de código aumentará, mas devido o baixo acoplamento entre a camada de apresentação e a lógica de negócios, a manutenção e reutilização serão bem preservadas e de fácil implementação.
- **Testabilidade:** graças a separação que o *MVVM* traz, a elaboração de testes unitários para a camadas do *Viewmodel* fica mais fácil, já que não traz independência da camada de visualização.
- **Extensibilidade:** ao longo do tempo, esta arquitetura oferece a garantia de que o código possa ser extensível.

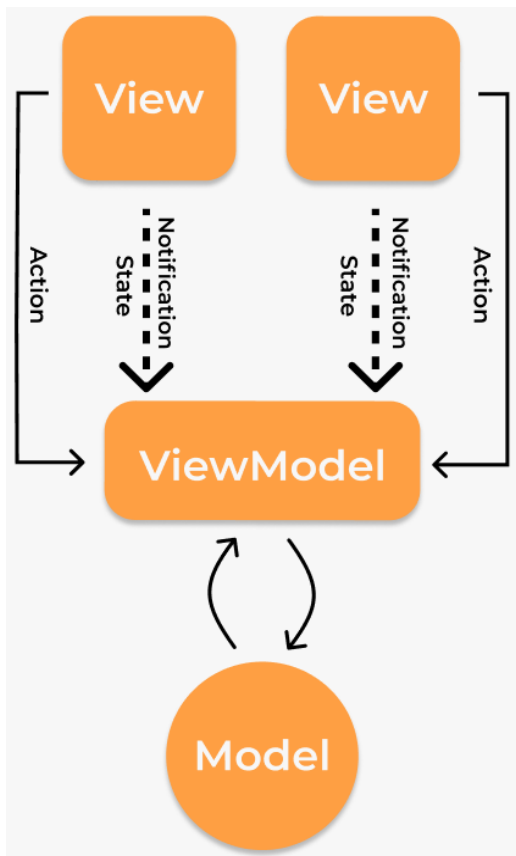


Figura 1: Arquitetura MVVM

Por fim, para somar na arquitetura do projeto, foi adotado alguns *design's patterns*, destacando-se entre eles, o *Repository Pattern*, que nos oferece uma camada que lida diretamente com os dados consumidos pela aplicação, proporcionando uma interface que facilita uma possível de banco de dados utilizado pelo *app* sem afetar o sistema como um todo.

Por fim, ainda agregando no nosso desenvolvimento, utilizamos o *MobX* [12] para a gerência de estados e o *Slidy* [13] junto com o *Flutter Modular* [14] para estruturação do projeto de forma padronizada, separando a aplicação em módulos isolados e bem arranjados. Vale destacar que, em nosso projeto, denominamos a camada de *Viewmodel* de *Stores*, como o padrão *MobX* para o Flutter sugere. A Figura 2 mostra com exatidão a distribuição de camadas segundo a estrutura do projeto.

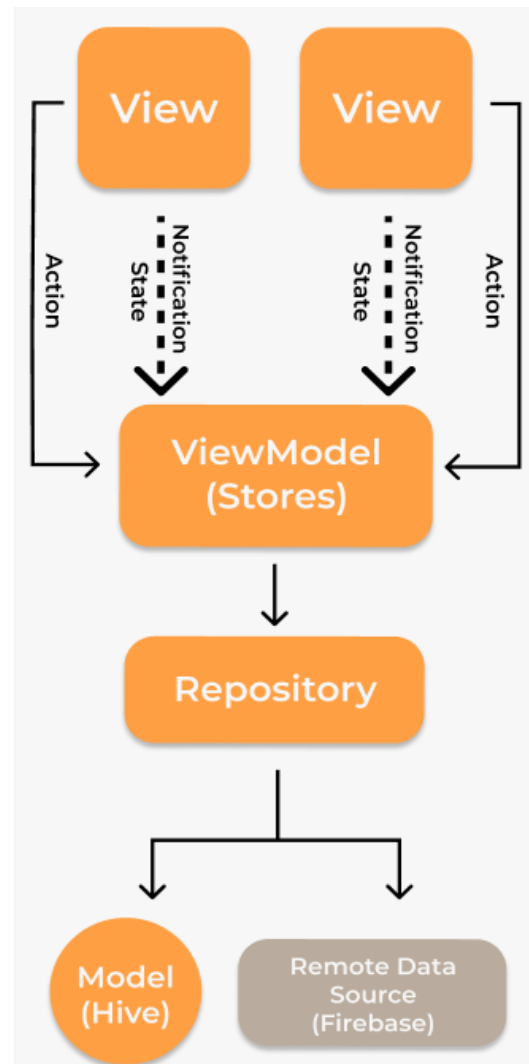


Figura 2: Arquitetura do app Backstage

3. EXPERIÊNCIA

3.1 Processo de desenvolvimento

Em um primeiro momento, foi realizada uma pesquisa que tinha como objetivo avaliar o grau de satisfação do uso de aplicações de organização de eventos entre os líderes e membros de equipes da Igreja Verbo da Vida Sede em Campina Grande. Em especial, a Igreja tem adotado o uso do Timbragem

durante suas programações semanais para a organização de seus cultos, registrando o cronograma do culto (momento de música, avisos, momento da Palavra, etc), as músicas que serão tocadas e as letras que serão exibidas no telão do templo para melhor acessibilidade dos membros.

A pesquisa se deu por um formulário de avaliação de desempenho com algumas perguntas básicas que avaliavam o Timbragem em usabilidade e interface, além de outras questões que pediam a opinião dos usuários a respeito das qualidades e defeitos do Timbragem. No final da pesquisa, uma última pergunta feita tinha como intenção colher a opinião dos usuários sobre qual mudança na aplicação poderia ser feita para que ela vivesse atender as suas necessidades por completo.

Tomando como base a pesquisa realizada, em seguida foram levantadas as principais qualidades e os principais defeitos e/ou pontos de melhoria que os usuários encontraram no Timbragem. Entre suas qualidades, destacaram-se: organização e praticidade; entre os pontos de melhoria, destacaram-se: falta de clareza e objetividade na interface, dificuldades em bloqueios de datas (caso o usuário não possa participar do compromisso), notificações apenas via e-mail, falta de histórico de utilização/participação nos eventos e ausência de personalização do *app*.

Com todas essas informações em mente, foi realizado o levantamento de requisitos da aplicação, assim como o cronograma de desenvolvimento e criação do design da aplicação, de uma forma que sanasse as insatisfações dos usuários e melhorasse os pontos de qualidade que o Timbragem possui.

Posto isto, foi criada uma POC (*Proof of Concept*), uma prova de conceito que reunia as principais demandas que os líderes e membros da Igreja necessitavam para a organização de suas escalas e eventos, trazendo consigo uma interface limpa mas complexa, com poucos botões que realizam muitas tarefas e que proporcionasse uma boa experiência de uso. Para fins iniciais e avaliação, foi disponibilizado um *apk* [15] com dados de testes criados diretamente no console do Firebase.

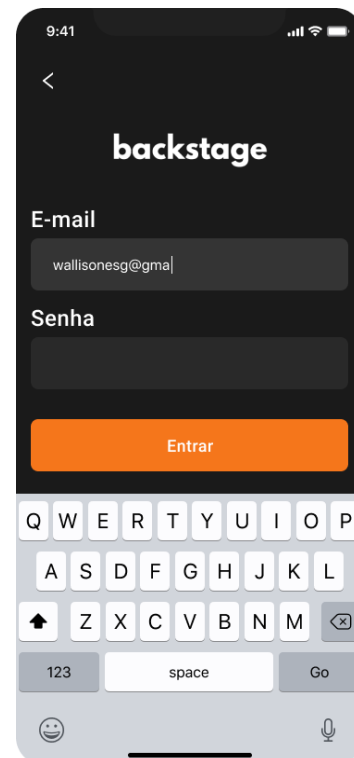


Figura 3: Wireframe da tela de login

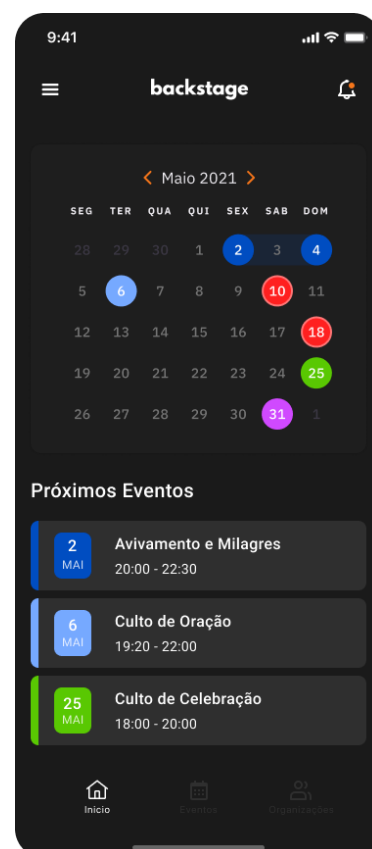


Figura 4: Wireframe da tela inicial, com dias bloqueados (em vermelho) e eventos marcados (outras cores).



Figura 5: Wireframe da tela de mídia, com listagem de vídeo e músicas fixados.



Figura 7: Wireframe da tela de horários frisados no cronograma.

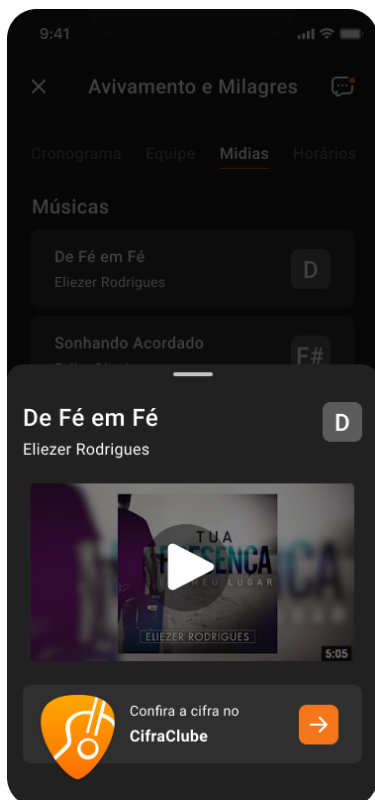


Figura 6: Wireframe do modal de exibição do conteúdo de mídia disponibilizado.

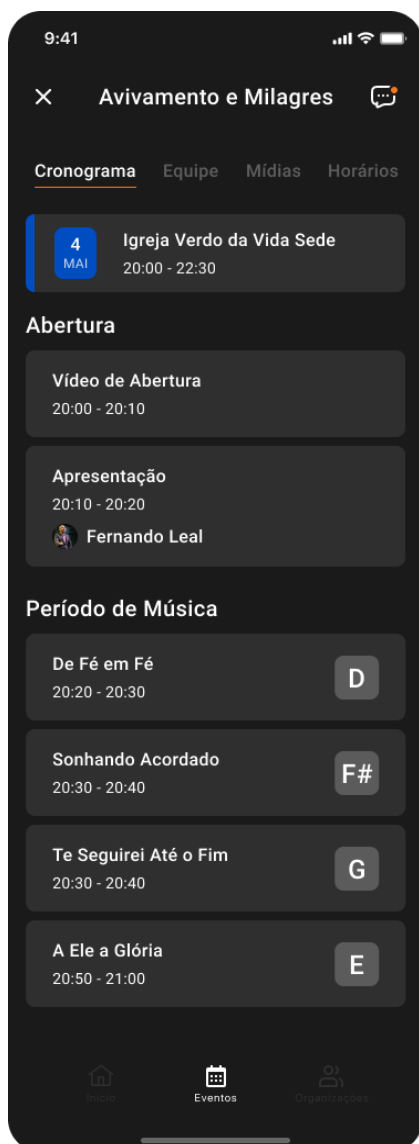


Figura 8: Wireframe do cronograma direcionado a uma programação de culto semanal da Igreja.

3.2 Desafios e soluções

O maior desafio encontrado em todo o processo de desenvolvimento deste trabalho foi o de criação e personalização de templates para cronograma de eventos. Sem dúvidas, foi uma tarefa árdua, visto que o Flutter não oferece bibliotecas para este tipo de problemática por se tratar de um domínio muito específico. Para solucionar tal problema, foi feito um *mini-framework* dentro da estrutura da aplicação que acomodasse os layouts de uma forma uniforme, limitando um pouco a personalização do usuário, oferecendo-lhe apenas campos de título e sessões de inclusões de conteúdo, com *cards* e textos.

4. TRABALHOS FUTUROS

O trabalho desenvolvido trouxe uma abertura de novas funcionalidades para o aplicativo, deixando-o mais completo e com perfil único no mercado, diria. Se destacam:

- **Autenticação/ verificação em duas etapas.** Segurança é algo indispensável, principalmente quando as informações necessitam de certo sigilo. Foi pensado uma futura implementação de autenticação em duas etapas, utilizando-se de envios de código via SMS e/ou e-mail.
- **Chat interno.** A possibilidade de tornar a organização bem centralizada no app, trouxe em mente a criação de um sistema de bate-papo interno, centralizando informações e comunicações entre todos os membros da organização.
- **Alternância entre temas.** Apesar da POC construída ter sido feita em tema *dark* que oferece economia de recursos de energia no dispositivo e melhor visibilidade para usuários com problemas de visão e sensibilidade, é sempre bom dar opções para usuário escolher o que ele bem entende como melhor. Então, foi pensado na disponibilidade futura de um tema *light*.
- **Templates prontos.** Por fim, uma opção seria oferecer ao usuário templates de cronogramas criados previamente para sua personalização.

5. AGRADECIMENTOS

Finalmente, deixo meus agradecimentos: primeiramente a Deus, pela oportunidade de poder desenvolver todo este trabalho; a minha noiva e futura esposa por estar comigo em todo o tempo; a meus familiares e amigos, que me desejam o bem; a todos os professores que me acompanharam durante todo o curso de graduação com extrema dedicação e maestria. Desde já, meus agradecimentos sinceros.

6. REFERÊNCIAS

- [1] Pesquisa estima que metade da população mundial tem smartphones. Disponível em: <<https://www.tecmundo.com.br/mercado/22000-9-pesquisa-estima-metade-populacao-mundial-t-em-smartphones.htm>>. Acesso em: 06 de outubro de 2021.
- [2] Brasil tem 424 milhões de dispositivos digitais em uso. Disponível em: <<https://portal.fgv.br/noticias/brasil-tem-424-milhoes-dispositivos-digitais-uso-revela-31a-pesquis>>

- a-anual-fgv<|. Acesso em: 06 de outubro de 2021.
- [3] Brasil tem dois dispositivos digitais por habitante. Disponível em: <https://portal.fgv.br/noticias/brasil-tem-dois-dispositivos-digitais-habitante-revela-pesquisa-fgv?utm_source=portal-fgv&utm_medium=fgvnoticias&utm_campaign=fgvnoticias-2021-05-25>. Acesso em: 06 de outubro de 2021.
- [4] Brasileiro passa mais de cinco horas por dia em apps no celular. Disponível em: <<https://www.terra.com.br/noticias/tecnologia/brasileiro-passa-mais-de-5-horas-por-dia-em-apps-de-celular,ae59609393373dfb3f03726aceae1626945axshg.html>>. Acesso em: 06 de outubro de 2021.
- [5] App Annie, 2021. O tempo diário gasto em aplicativos aumenta cerca de 45% em dois anos. Disponível em: <<https://www.appannie.com/en/insights/market-data/q2-2021-market-index-regional-rankings/>>. Acesso em: 06 de outubro de 2021.
- [6] Timbragem Plan. Disponível em: <<https://www.timbragemplan.com.br/>>. Acesso em 04 de outubro de 2021.
- [7] Firebase. Disponível em: <<https://firebase.google.com/>>. Acesso em: 05 de outubro de 2021.
- [8] Documentação dos serviços do Firebase por plataforma. Disponível em: <<https://firebase.google.com/docs/>>. Acesso em: 07 de outubro de 2021.
- [9] Flutter. Disponível em: <<https://flutter.dev/>>. Acesso em: 05 de outubro de 2021.
- [10] Examinando o desempenho entre Flutter, React Native e nativo. Disponível em: <<https://medium.com/swlh/flutter-vs-native-vs-react-native-examining-performance-31338f081980>>. Acesso em: 07 de outubro de 2021.
- [11] O repositório oficial de pacotes para aplicativos Dart e Flutter: Pub. Disponível em: <<https://pub.dev/>>. Acesso em: 07 de outubro de 2021.
- [12] MobX, uma biblioteca de gerenciamento de estados. Disponível em: <<https://pub.dev/packages/mobx>>. Acesso em: 07 de outubro de 2021.
- [13] Slidy, gerenciador de pacotes e modelos em Flutter. Disponível em: <<https://pub.dev/packages/slidy>>. Acesso em: 07 de outubro de 2021.
- [14] Flutter Modular, responsável pelo sistema de injeção de dependências e gerenciamento de rotas. Disponível em: <https://modular.flutterando.com.br/docs/flutter_modular/start/>. Acesso em: 07 de Outubro de 2021.
- [15] Android Developers. Publish your app. Disponível em: <<https://developer.android.com/studio/publish>>. Acesso em: 07 de outubro de 2021.
- [16] Backstage wireframes, criados por Wallyngson Guedes - UI/UX Design. Disponível em: <<https://bit.ly/backstage-ui>>. Acesso em: 01 de outubro de 2021.