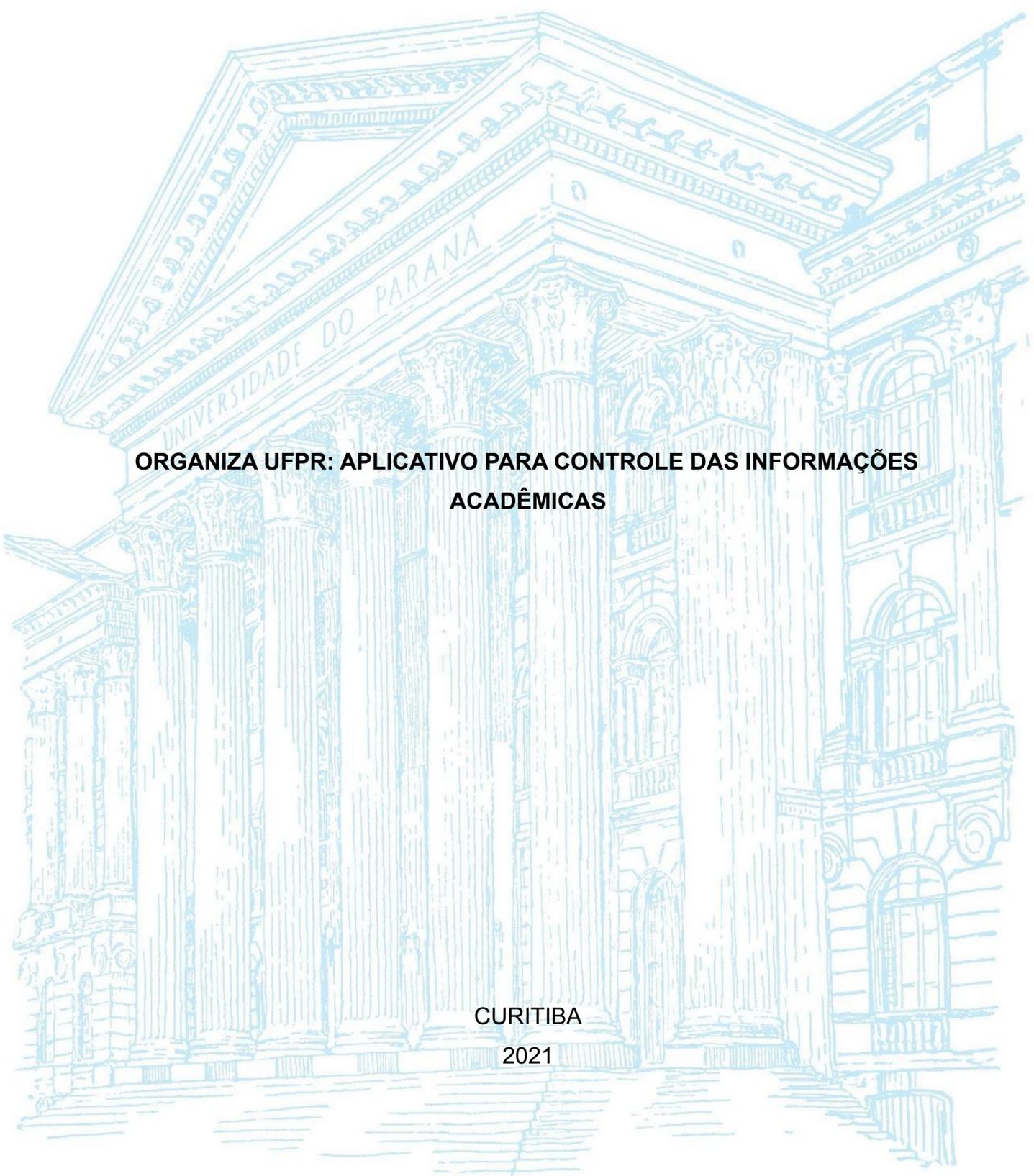


UNIVERSIDADE FEDERAL DO PARANÁ

NICOLLAS SAQUE



**ORGANIZA UFPR: APLICATIVO PARA CONTROLE DAS INFORMAÇÕES
ACADÊMICAS**

CURITIBA
2021

NICOLLAS SAQUE

**ORGANIZA UFPR: APLICATIVO PARA CONTROLE DAS INFORMAÇÕES
ACADÊMICAS**

TCC apresentado ao curso de graduação em Bacharelado em Ciência da Computação, Setor de Exatas, Universidade Federal do Paraná, como requisito parcial à obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Andrey Ricardo Pimentel

CURITIBA

2021

AGRADECIMENTOS

Aos meus pais, por fornecer todos os recursos necessários para eu chegar até aqui e por sempre acreditarem na minha vitória.

Aos meus colegas, pela troca de conhecimento e ajuda ao longo de todo o curso.

À todo departamento de Ciência da Computação e a Universidade Federal do Paraná, por fornecer toda a infraestrutura para eu me tornar um profissional melhor.

Especificamente ao professor Dr. Eduardo Todt, por toda a confiança, oportunidades dadas e crença no meu trabalho.

E, principalmente, a minha família que esteve sempre presente e me ajudou a atingir essa conquista.

RESUMO

Os alunos da UFPR necessitam acessar informações acadêmicas ao longo de toda a graduação, porém, não há disponível um sistema unificado de acesso a todas elas. Sendo assim, o objetivo do trabalho foi mapear o potencial de uma aplicação com tal finalidade. Foi produzido, então, um aplicativo chamado Organiza UFPR, onde é possível importar informações do comprovante de matrícula da graduação da UFPR relacionadas às disciplinas e dados acadêmicos do(a) estudante. É possível modificar as informações, criar novas disciplinas, visualizar a grade horária e salvar anotações individuais para cada disciplina. O aplicativo passou por duas avaliações, a primeira focada em analisar itens de acessibilidade e usabilidade por alunos da pós-graduação do laboratório de Design de Interação para a Inclusão Social e a Educação Universal (IHC) que serviram para melhorá-lo para a segunda avaliação, que foi realizada com alunos e ex-alunos para medir o nível de aceitação e o potencial da aplicação.

Palavras-Chave: Aplicativo. Informações acadêmicas. TAM. UFPR

ABSTRACT

UFPR students need to access academic information throughout their undergraduate course, however, there is no unified system for accessing all of them. Therefore, the objective of the work was to map the potential of an application with such purpose. An application called Organiza UFPR was then produced, where it is possible to import information from the UFPR graduation enrollment certificate related to the student's disciplines and academic data. It is possible to modify the information, create new subjects, view the timetable and save individual notes for each subject. The application underwent two assessments, the first focused on analyzing accessibility and usability items by graduate students of the Interaction Design laboratory for Social Inclusion and Universal Education (IHC), which served to improve it for the second assessment , which was carried out with students to measure the level of acceptance and the potential of the application.

Keyword: Application. Academic information. TAM, UFPR

LISTA DE FIGURAS

FIGURA 1 - <i>FRAMEWORKS MOBILE</i> MULTIPLATAFORMA USADOS POR DESENVOLVEDORES DE 2019 A 2021	11
FIGURA 2 - CAMADAS DO <i>FRAMEWORK</i> FLUTTER	13
FIGURA 3 - PÁGINAS DO APLICATIVO HORÁRIO DE AULA	18
FIGURA 4 - PÁGINAS DO APLICATIVO TIMETABLE	19
FIGURA 5 - PÁGINAS DO APLICATIVO CLASS TIMETABLE	21
FIGURA 6 - PÁGINAS DO APLICATIVO AGENDA ESCOLAR	22
FIGURA 7 - PÁGINAS DO APLICATIVO AGENDA DO ESTUDANTE PRO - ORGANIZE-SE!	23
FIGURA 8 - PÁGINAS DO APLICATIVO SIGGA MOBILE	25
FIGURA 9 - PÁGINAS DO APLICATIVO SIGAUFMG	26
FIGURA 10 - PÁGINAS DO APLICATIVO UFU MOBILE	27
FIGURA 11 - FUNÇÃO MAIN DO ORGANIZA UFPR	29
FIGURA 12 - FUNÇÃO MYAPP DO ORGANIZA UFPR	30
FIGURA 13 - PÁGINA INICIAL DO ORGANIZA UFPR	31
FIGURA 14 - PÁGINA DE DISCIPLINAS DO ORGANIZA UFPR	32
FIGURA 15 - RECORTE DA FUNÇÃO DE IMPORTAR PDF	33
FIGURA 16 - DADOS ACADÊMICOS	34
FIGURA 17 - DADOS DAS DISCIPLINAS	35
FIGURA 18 - FUNÇÃO DE SALVAR OS DADOS ACADÊMICOS	36
FIGURA 19 - RECURSOS DO ORGANIZA UFPR	37
FIGURA 20 - FUNÇÃO DE SALVAR ANOTAÇÕES	38
FIGURA 21 - LIMPAR DADOS DISCIPLINAS	39
FIGURA 22 - CRIAR UM DISCIPLINA	40
FIGURA 23 - GRADE HORÁRIA, DIA E HORÁRIOS	41
FIGURA 24 - CONFIGURAÇÕES DE TELA	42
FIGURA 25 - TEMAS DO ORGANIZA UFPR	43
FIGURA 26 - FONTES DO ORGANIZA UFPR	44

LISTA DE TABELAS

TABELA 1 - APLICATIVO HORÁRIO DE AULA	17
TABELA 2 - APLICATIVO TIMETABLE	24
TABELA 3 - APLICATIVO CLASS TIMETABLE.....	20
TABELA 4 - APLICATIVO AGENDA ESCOLAR.....	21
TABELA 5 - APLICATIVO AGENDA DO ESTUDANTE PRO - ORGANIZE-SE!	23
TABELA 6 - APLICATIVO SIGGA MOBILE	24
TABELA 7 - APLICATIVO SIGAUFGM	25
TABELA 8 - APLICATIVO UFU MOBILE	27
TABELA 9 - COMPARAÇÃO DE FUNCIONALIDADE DOS APLICATIVOS ANALISADOS	28
TABELA 10 - FUNCIONALIDADES DO APLICATIVO ORGANIZA UFPR ...	45
TABELA 11 - PACOTES E PLUGINS UTILIZADOS	45
TABELA 12 - QUESTIONÁRIO DE AVALIAÇÃO E NOTAS	51

LISTA DE ABREVIATURAS OU SIGLAS

2D	-	Duas dimensões
AOT	-	Ahead of time
APK	-	Android Package
ARM	-	Arquitetura ARM (advanced RISC machine)
GRR	-	Graduação (GR) regular (R)
IDE	-	Integrated development environment
JIT	-	Just in time
NDK	-	Native development kit
PEU	-	Perceived Ease of Use
PU	-	Perceived usefulness
SIGAUFR	-	Sistema Integrado de Gestão Acadêmica da UFPR
TAM	-	Technology Acceptance Model
UFPR	-	Universidade Federal do Paraná
UI	-	User interface
VM	-	Virtual machine
X64	-	Arquitetura de processadores de 64 bit
X86	-	Arquitetura X86 (processadores baseados no Intel 8086)

SUMÁRIO

1 INTRODUÇÃO	9
2 CONCEITOS BÁSICOS	11
2.1 FLUTTER	11
2.2 DART	13
2.3 ANDROID STUDIO	15
3 TRABALHOS CORRELATOS	17
3.1 HORÁRIO DE AULA	17
3.2 APLICATIVO TIMETABLE	18
3.3 APLICATIVO CLASS TIMETABLE	20
3.4 APLICATIVO AGENDA ESCOLAR	21
3.5 APLICATIVO AGENDA DO ESTUDANTE PRO - ORGANIZE-SE!	22
3.6 APLICATIVO SIGGA MOBILE	24
3.7 APLICATIVO SIGAUFGM	25
4 ORGANIZA UFPR	29
5 VALIDAÇÃO	46
5.1 Resultados	46
5.2 TRABALHOS FUTUROS	54
6 CONCLUSÃO	56
REFERÊNCIAS	57

1 INTRODUÇÃO

É certo que a todo momento da graduação, nós alunos devemos ter acesso a informações acadêmicas. Mais especificamente na UFPR¹³, temos que saber o número da matrícula ou registro acadêmico - conhecido como GRR⁵ para a UFPR que significa graduação (GR) regular (R) - para todas as atividades dentro do *campus* que exijam uma identificação. O acesso ao restaurante universitário e visualização de notas das disciplinas são realizados majoritariamente a partir dessa identificação. Além do GRR, saber a sala, dia e horário onde as disciplinas serão ministradas também é um fator imprescindível no dia a dia do estudante, pois sem essas informações não seria possível estar presente nas aulas e conseqüentemente concluir a graduação.

Com isso, uma das inúmeras dificuldades que os estudantes da UFPR têm de enfrentar ao longo da graduação é a responsabilidade de juntar, lembrar ou decorar informações acadêmicas e das disciplinas provenientes de diferentes documentos e sistemas. Informações disponíveis a partir do Sistema Integrado de Gestão Acadêmica da UFPR, conhecido como SigaUFPR¹¹, são frequentemente requeridos pelos alunos, mas nem sempre está à disposição uma conexão com a internet para ter acessos a essas informações, além do fato de necessitar autenticação, essa tarefa nem sempre é rápida quanto o necessário. Uma alternativa adotada pelos alunos para o acesso facilitado às principais informações acadêmicas foi a partir do armazenamento local dos principais documentos - como comprovante de matrícula - e imagens capturadas das telas de seus celulares. Essa solução, contudo, está limitada à antecipação das informações que serão necessárias e requer a etapa de seleção e organização dos documentos.

Sendo o celular, hoje, o principal meio de acesso a internet no Brasil (TOKARNIA, 2020), é por ele que os sistemas acadêmicos são acessados e onde os indivíduos guardam, modificam e consultam as informações, inclusive relacionadas ao ambiente acadêmico. Os aplicativos tornam essas tarefas mais fáceis, por proporcionarem um ambiente único e interativo para o usuário, tendo a opção de oferecer recursos sem necessitar uma conexão de internet. Mas a importância dos aplicativos vai além de tudo isso, eles proporcionam uma proximidade maior com o usuário, aumentando o engajamento no uso e facilitando o

uso de funcionalidades - alarmes, notificações, acesso a arquivos - dos dispositivos móveis (ALPHACODE, 2021).

Então, o objetivo do trabalho é mapear o potencial que uma alternativa que reúne as informações acadêmicas, por meio dos dispositivos móveis, demandadas pelo aluno e de suas disciplinas cursadas em um só lugar a partir de um aplicativo exclusivo. O aplicativo tem como uma de suas principais vantagens a não necessidade de conexão à internet e a importação das informações encontradas no documento de identificação do aluno da UFPR de forma individualizada. Um aplicativo com tal finalidade potencialmente seria de grande utilidade para a maioria dos estudantes universitários e, atualmente, não possui nenhuma solução em desenvolvimento pela UFPR.

A próxima seção, conceitos básicos, irá tratar sobre as três principais tecnologias usadas para construir o aplicativo - Flutter, Dart e Android Studio - e suas principais características. Após, serão listados os trabalhos correlatos selecionados que possuem similaridades com o aplicativo desenvolvido, tanto de recursos quanto de objetivos. A quarta seção irá apresentar as páginas e recursos do Organiza UFPR, além de alguns detalhes de implementação e pacotes/*plugins* utilizados. A validação é a próxima seção, separada em resultados, onde serão explicados os resultados e como foram realizados os dois experimentos de validação a que o aplicativo foi submetido e a parte de trabalhos futuros, que listará algumas melhorias futuras interessantes para o aplicativo. Por fim temos as conclusões, com a opinião final do autor sobre o aplicativo e as validações.

2 CONCEITOS BÁSICOS

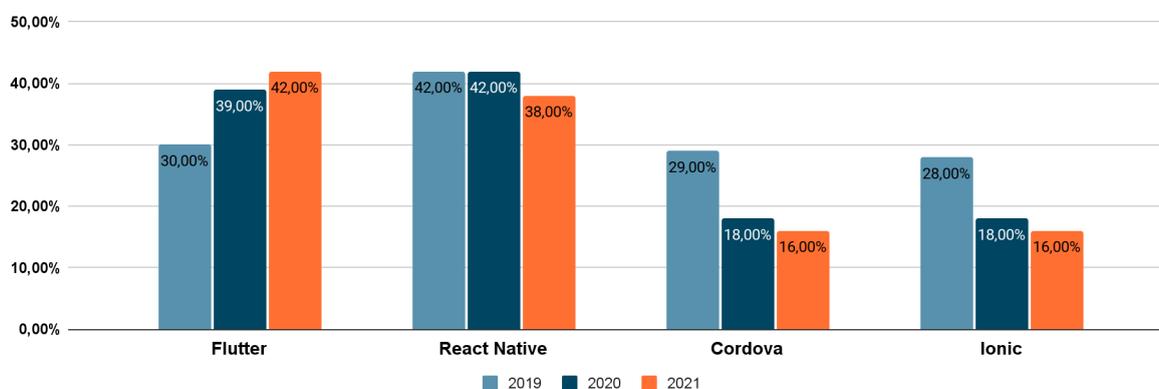
Para mapear o potencial de desenvolver uma solução a partir de um aplicativo para celular para os alunos da UFPR, foi utilizado o kit de desenvolvimento de interface de usuário (UI¹⁴ *toolkit*) Flutter, conseqüentemente a linguagem Dart e o ambiente de desenvolvimento integrado (IDE⁶) Android Studio. Dentre as opções disponíveis, as alternativas selecionadas foram por causa da afinidade dessas tecnologia com o Android por se tratar de *softwares* do Google.

2.1 FLUTTER

O Flutter é um UI *toolkit* de software livre criado pelo Google em maio de 2017. No presente trabalho, ele foi usado para compilar para o Android, porém ele também compila para todas as outras principais plataformas, como o IOS, Windows, Mac, Linux e *web* (SHARMA, 2021), sendo essa característica uma de suas grandes vantagens.

Em 2021, o uso do Flutter está em constante crescimento, ultrapassando o React Native como o *framework mobile* multiplataforma mais usado por desenvolvedores do mundo inteiro, com mais de 100.000 aplicativos postados na principal loja de aplicativos do Android, a Google Play (FLUTTER, 2021a). A Figura 1 apresenta a porcentagem desses desenvolvedores que usam os quatro softwares de maior relevância, ressaltando que 42% deles estão utilizando o Flutter.

FIGURA 1 - FRAMEWORKS MOBILE MULTIPLATAFORMA USADOS POR DESENVOLVEDORES DE 2019 A 2021



FONTE: Adaptado de STATISTA, 2021.

Levando em conta que o Flutter foi lançado em 2017 (SHARMA, 2021), percebemos que ele alcançou uma posição de destaque muito rápido dentro do cenário de desenvolvimento de aplicativos, em apenas dois anos (um ano da versão 1.0) ele já era o segundo *framework* (*mobile* multiplataforma) mais utilizado, ultrapassando outros *softwares* de destaque, como o Cordova (lançado em 2009) e o Ionic (lançado em 2013).

Essa grande ascensão se dá pelas vantagens do *software*, além de ter uma alta performance por utilizar linguagem de alto desempenho em sua *engine* - C e C++ - ele cria aplicativos compilados nativamente a partir de uma mesma base de código para as diferentes plataformas, eliminando a necessidade de ter várias bases de código para ter *apps* nativos para cada uma delas, permitindo o uso de *hardware* e *software* específico com um melhor desempenho e usufruindo das tecnologias mais recentes fornecidas pelos dispositivos (GILLIS, 2020). Em um contexto empresarial, essa característica permite que um único time de desenvolvedores seja responsável pelos aplicativos *web*, *desktop* e *mobile* (FLUTTER, 2021a).

A ideia que os próprios desenvolvedores relacionam ao Flutter é de que ele constrói aplicativos bonitos - de grande aceitação visual -, isso por ele ter integrado mecanismos de renderização 2D¹ otimizados para aplicativos móveis, mais especificamente a biblioteca gráfica Skia - biblioteca gráfica de renderização 2D - e uma grande variedade de *widgets* em conformidade com as linguagens de design Material Design, para o caso dos componentes do Google, e Cupertino, esse para IOS. Isso faz as aplicações ficarem com um visual atualizado e condizentes com os padrões de design atuais (FLUTTER, 2021a).

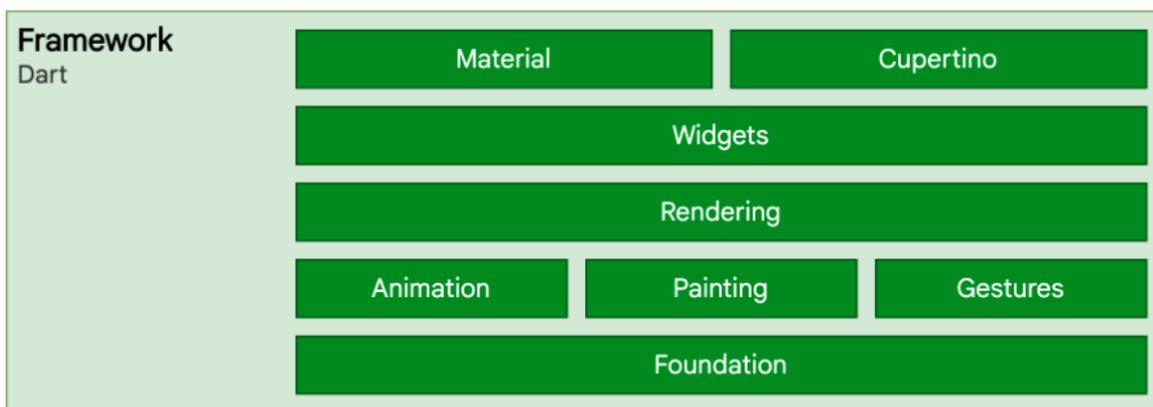
Outra vantagem do *framework* é o *hot reload*, onde o Flutter atualiza instantaneamente o *software* em execução, injetando código fonte na VM¹⁵ Dart que é usada durante a depuração, mantendo o estado do *software* e atualizando as classes e métodos sem a necessidade de recompilação, agilizando bastante o desenvolvimento, pois modificando o código fonte e salvando-o é possível visualizar as modificações na aplicação de maneira rápida e previsível (FLUTTER, 2021a).

Como já citado anteriormente, o Flutter é constituído pelas linguagens C, C++, Dart e pela biblioteca Skia (implementada em C++). No Android, os códigos C e C++ do mecanismo são compilados com o NDK⁸ - kit de desenvolvimento nativo - do Android, que é um conjunto de ferramentas que permite implementar partes do app em código nativo. Já o Dart é compilado de maneira AOT² (Ahead-Of-Time), no

caso todo código é transformado para linguagem de máquina antes da execução (FLUTTER, 2021b). Isso é feito por bibliotecas nativas ARM⁴ e X86¹⁷ e durante a execução, qualquer renderização, entrada e manipulação de evento são passados para o Flutter compilado (já incluído o código do aplicativo) tratar (FLUTTER, 2021b).

Para uma visão mais geral e integrada do *framework*, a Figura 2 apresenta a camada mais alta da arquitetura que representa o Flutter. Na base, temos representada a biblioteca Foundation, escrita em Dart que fornece as classes e funções básicas utilizadas pelas bibliotecas de animação, pintura e gestos que estão em um nível superior a ela. Essas três bibliotecas fornecem abstrações bastante utilizadas pela camada de renderização, responsável pelo *layout* da aplicação. Mais acima temos a camada de *widgets*, que define em classes os objetos da camada de renderização, permitindo a reutilização das classes. Por último temos as já citadas linguagens de *design*, Material (Android) e Cupertino (IOS), modificando os componentes da classe de *widgets* (FLUTTER, 2021b).

FIGURA 2 - CAMADAS DO *FRAMEWORK* FLUTTER



FONTE: Adaptado de FLUTTER, 2021c.

A interação do desenvolvedor com o flutter se dá pela linguagem Dart, que falaremos no tópico a seguir.

2.2 DART

A linguagem de programação Dart foi criada no final de 2011 pela Google com o objetivo de substituir o Javascript como a principal linguagem utilizada em

navegadores, por sinal, o objetivo não foi alcançado, mas seu recente sucesso se dá por outros motivos. Ela passou a focar no desenvolvimento de aplicativos do lado do cliente, com a promessa de entregar aplicativos rápidos em qualquer plataforma, com alta produtividade para os desenvolvedores e grande potencial de ser ligada à *frameworks* de aplicativos, como visto no caso do Flutter, onde o Dart foi usado tanto para construir a base do software quanto para ser a linguagem que os desenvolvedores utilizam para construir as aplicações (DART, 2021a).

Em relação a compilação, o Dart possui duas tecnologias que permitem executar códigos de diferentes maneiras. A primeira delas para dispositivos móveis e *desktops* inclui uma VM Dart com compilação JIT⁷ (just-in-time) - em tempo de execução - e compilação AOT (ahead-of-time) - anterior a execução -, ambas responsáveis por produzir código de máquina (DART, 2021a).

O compilador JIT possui um sistema de recompilação incremental, que em conjunto com um suporte avançado de depuração que a linguagem fornece - Dart DevTools - torna o desenvolvimento muito mais rápido. Enquanto esse primeiro compilador é usado para agilizar o processo de desenvolvimento, o compilador AOT é utilizado em produção, transformando os códigos Dart para ARM ou x64¹⁶ nativo (DART, 2021a).

Para plataformas *web*, seguindo a mesma ideia dos dispositivos *mobile/desktop*, a linguagem inclui um compilador de desenvolvimento (*dartdevc*) e um compilador de produção (*dart2js*) que transformam o código dart em JavaScript. O *dartdevc*, assim como o compilador JIT do *mobile/desktop*, suporta compilação incremental, com ele é possível executar e depurar um aplicativo *web* no Chrome, navegador do Google. O segredo do compilador está no fato dele apenas compilar os módulos que foram atualizados/modificados e não todo o programa (DART, 2021). O *dart2js* é usado em produção, mas ambos têm a capacidade de criar código Javascript rápido, compacto e usando algumas técnicas, como a eliminação de código inútil (DART, 2021a).

Um ponto importante das linguagens de programação é o tempo de execução dos programas escritos. Em relação a isso, o tempo que o Dart leva para execução, basicamente, é o tempo de três tarefas principais (DART, 2021b). O gerenciamento de memória é a primeira delas, onde a linguagem usa um modelo de memória gerenciada controlada por um “coletor de lixo”. Outra tarefa que gasta tempo é as verificações de tipo, a maioria delas é realizada em tempo de compilação, porém

algumas são feitas em tempo de execução, por exemplo, para os operadores “as”, “is” e “is!”. A última seria o gerenciamento dos “isolados”, que é a alternativa utilizada no lugar do uso de Threads, eles são áreas seguras na memória criadas com heap próprio e não acessíveis por outros isolados (DART, 2021c).

A seguir estão listados alguns conceitos importantes da linguagem Dart:

- Tudo que pode ser colocado em uma variável é um objeto, herdado da classe Object, sendo que cada objeto é uma instância dessa classe. Isso permite o uso de tipos genéricos na linguagem, como List<Object>;
- É uma linguagem fortemente tipada, porém não é necessário declarar o tipo de uma variável, pois o Dart pode inferir qual é ele;
- O Dart oferece suporte ao chamado *sound null safety*, onde as variáveis de um programa não podem assumir um valor nulo, exceto quando especificado, a quebra dessa regra ocasiona erro de compilação;
- A linguagem não possui os atributos *public*, *protected* e *private*, como no Java e C++, nela se um elemento começa com o carácter “_” ele é privado;
- Erros podem ocorrer em tempo de compilação, impedindo que o programa seja executado e em tempo de execução, resultando em uma exceção.

2.3 ANDROID STUDIO

Também foi utilizado no presente trabalho o IDE Android Studio, um dos mais comuns na cena de desenvolvimento *mobile* e criado, assim como o Flutter e Dart, pelo Google. Ele possui vários recursos interessantes para o desenvolvimento de aplicações para o Android, como editor de *layout* visual, analisador de APK³, visando reduzir o tamanho dos aplicativos e emulador de celular de fácil interação (GOOGLE, 2021).

O principal motivo do uso do Android Studio é pelo fato dele possuir *plugin* Flutter e Dart, oferecendo recursos como: autocompletar código, destaque de sintaxe, assistência de edição de *widget* e suporte para execução e depuração. Além de tudo isso, ele cria um projeto com uma estrutura já iniciada de um

aplicativo, fazendo com que o desenvolvedor apenas se preocupe em criar a aplicação em si e não em criar arquivos necessários para se gerar um arquivo de extensão APK (GOOGLE, 2021).

Um dos recursos mais utilizados no desenvolvimento deste trabalho foi a possibilidade de executar a aplicação diretamente em um dispositivo móvel real. A partir disso era possível modificar o aplicativo e observar diretamente as mudanças já no dispositivo, sem a necessidade de realizar qualquer ação a não ser salvar o arquivo modificado.

3 TRABALHOS CORRELATOS

Esta seção visa apresentar os aplicativos disponíveis que possuem funcionalidades e objetivos próximos ao do presente trabalho. Foi utilizada a loja para dispositivos móveis Google Play Store para realizar a pesquisa dos aplicativos. Foram selecionados os aplicativos com as melhores avaliações na loja (maior que 4) em conjunto com um número relevante de *downloads* (mais de 10 mil), além de outros que não se encaixam nesses critérios, porém foram escolhidos por terem sido desenvolvidos por instituições públicas de ensino, entendendo que há uma grande proximidade com o objetivo deste trabalho.

A seguir serão listados os principais *softwares* selecionados e a Tabela 9 para facilitar a comparação entre eles.

3.1 HORÁRIO DE AULA

O primeiro aplicativo (TABELA 1) permite gerenciar mais de um horário e dentro de cada um deles é possível cadastrar as disciplinas, com informações referentes ao nome, professor, notas, sala e horário. Ele também permite adicionar eventos para cada disciplina, com o objetivo de cadastrar provas e trabalhos, os eventos possuem informações como nome, prioridade, dia e horário (GOOGLE PLAY, 2021a).

TABELA 1 - APLICATIVO HORÁRIO DE AULA

Aplicativo	Horário de aula
Nota	4,3
Downloads	+100 mil
Loja	Play Store
Desenvolvedor	Leonardo Palominos Barraza

FONTE: O autor (2021)

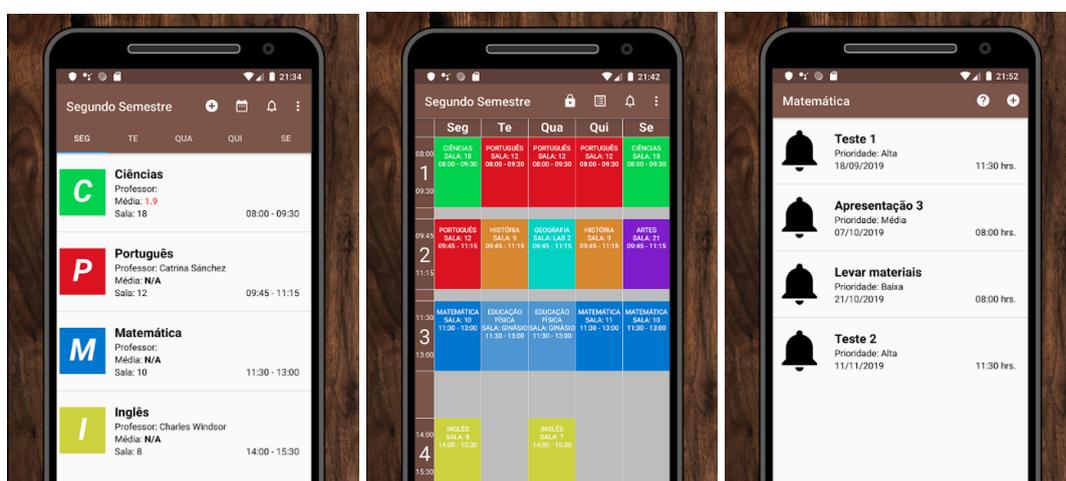
As propagandas são frequentes ao navegar pelo aplicativo, sendo que ele possui uma versão paga (que requer *login* com uma conta do *Google*) para não ser importunado por elas e acessar certas funcionalidades, como *download*, *backup*,

compartilhamento dos horários e a possibilidade de cadastrar infinitos horários e aulas. Outra funcionalidade do aplicativo é o *widget*, onde as informações, notas e eventos são apresentados na tela inicial do dispositivo.

Em relação a acessibilidade, basicamente, o aplicativo não possui nenhum suporte, é possível mudar a cor do aplicativo, mas essa mudança não auxilia pessoas com dificuldade ou problema de visão, e mudar o tamanho da fonte apenas na visualização gráfica dos horários, porém não para uma fonte grande, sendo que o tamanho máximo é 13, bem abaixo do *default* para várias linguagem focadas em aplicativos móveis que é 17.

A Figura 3 apresenta algumas páginas relevantes do aplicativo.

FIGURA 3 - PÁGINAS DO APLICATIVO HORÁRIO DE AULA



FONTES: GOOGLE PLAY, 2021

LEGENDA: a) Visão diária das disciplinas
b) Página da grade horária
c) Página dos eventos

3.2 APLICATIVO TIMETABLE

Um dos aplicativos com mais funcionalidades dentre os analisados, o aplicativo "Timetable" (TABELA 2) possui a visualização gráfica por semana e a visão por dia das matérias cadastradas nele. As informações referentes às disciplinas são: nome, horário, dia, professor e sala. É também possível cadastrar tarefas e provas para cada uma das matérias, sendo que o aplicativo mostra já na página inicial quais não foram concluídas e quais são as que estão próximas. É possível cadastrar feriados no aplicativo, onde ele atualiza os eventos para não disparar nesses dias (GOOGLE PLAY, 2021b).

TABELA 2 - APLICATIVO TIMETABLE

Aplicativo	Timetable
Nota	4,0
Downloads	+1 mi
Loja	Play Store
Desenvolvedor	Gabriel Ittner

FONTE: O autor (2021)

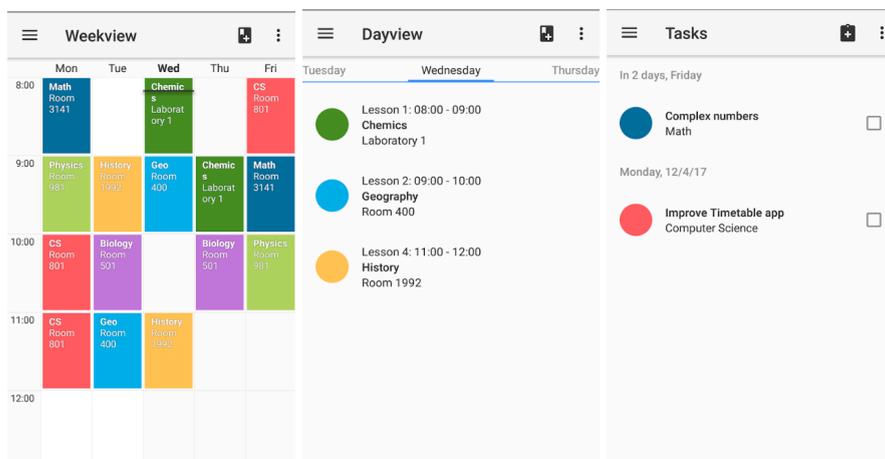
Não é necessário entrar em cada uma das disciplinas para ter acesso às provas e tarefas, o aplicativo possui páginas específicas para esses eventos onde aparece todos os eventos cadastrados em todas as matérias, facilitando bastante o gerenciamento deles.

Assim como o primeiro aplicativo, ele também possui um *widget* que mostra as matérias do dia e as próximas tarefas a vencer. Ele possui ligação com notificações do dispositivo, sendo possível disparar um alarme para avisar sobre matérias, provas e tarefas.

Este aplicativo possui um maior leque de opções de customização, tanto do tema (escuro, claro), cor de destaque de cada matéria, notificações, customização do *widget* e da visualização gráfica. Não é possível modificar o tamanho da fonte em nenhuma parte do aplicativo.

A Figura 4 apresenta algumas páginas relevantes do aplicativo.

FIGURA 4 - PÁGINAS DO APLICATIVO TIMETABLE



FONTE: GOOGLE PLAY, 2021

LEGENDA: a) Página da grade horária
b) Visão diária das disciplinas
c) Página dos eventos

3.3 APLICATIVO CLASS TIMETABLE

Um aplicativo simples que permite cadastrar disciplinas por dia, não tendo uma visualização gráfica semanal, nele é possível colocar apenas o nome e horário, porém ele fornece uma caixa de texto para escrever livremente outras informações (TABELA 3). A visualização das matérias é apenas diária e não é possível copiar uma matéria e as informações dela para mais de um dia, será necessário escrever mais de uma vez, isso para cada dia da matéria (GOOGLE PLAY, 2021c).

TABELA 3 - APLICATIVO CLASS TIMETABLE

Aplicativo	Class Timetable
Nota	4,0
Downloads	+1 mi
Loja	Play Store
Desenvolvedor	Class Timetable LLC

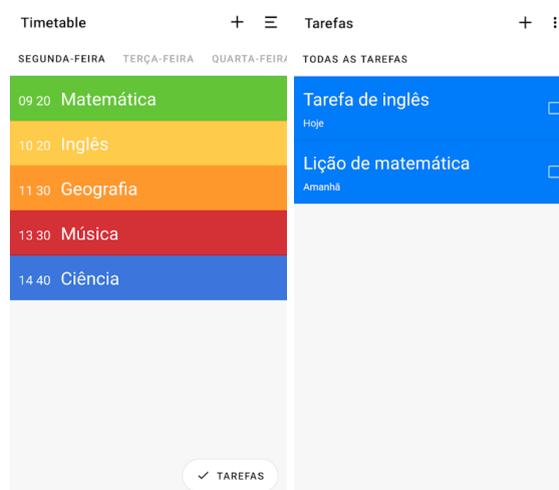
FONTE: O autor (2021)

O cadastro de tarefas também está presente no aplicativo, sendo que eles podem ou não estar relacionados a uma das disciplinas e só são notificados no aparelho com a versão paga do aplicativo, que também remove os anúncios, aprimora o widget e possibilita a exportação dos horários.

Além de escolher a cor de destaque para cada disciplina, o aplicativo não tem mais nenhuma opção de customização, principalmente opções de acessibilidade. Ele permite cadastrar até quatro semanas diferentes que podem ter suas próprias disciplinas e tarefas.

A Figura 5 apresenta algumas páginas relevantes do aplicativo.

FIGURA 5 - PÁGINAS DO APLICATIVO CLASS TIMETABLE



FONTE: GOOGLE PLAY, 2021

LEGENDA: a) Visão diária das disciplinas

b) Página das tarefas

3.4 APLICATIVO AGENDA ESCOLAR

Aplicativo (TABELA 4) um pouco mais complexo por ter um foco diferente do que os outros. Seu objetivo é fornecer uma agenda, onde os elementos principais são as tarefas e eventos e não a visualização dos dados referentes a uma disciplina em si, mesmo que seja possível ter este tipo de informação nele, porém há um esforço maior do usuário para chegar até essas informações (GOOGLE PLAY, 2021d).

TABELA 4 - APLICATIVO AGENDA ESCOLAR

Aplicativo	Agenda Escolar
Nota	4,6
Downloads	+10 mil
Loja	Play Store
Desenvolvedor	Andrea Dal Cin

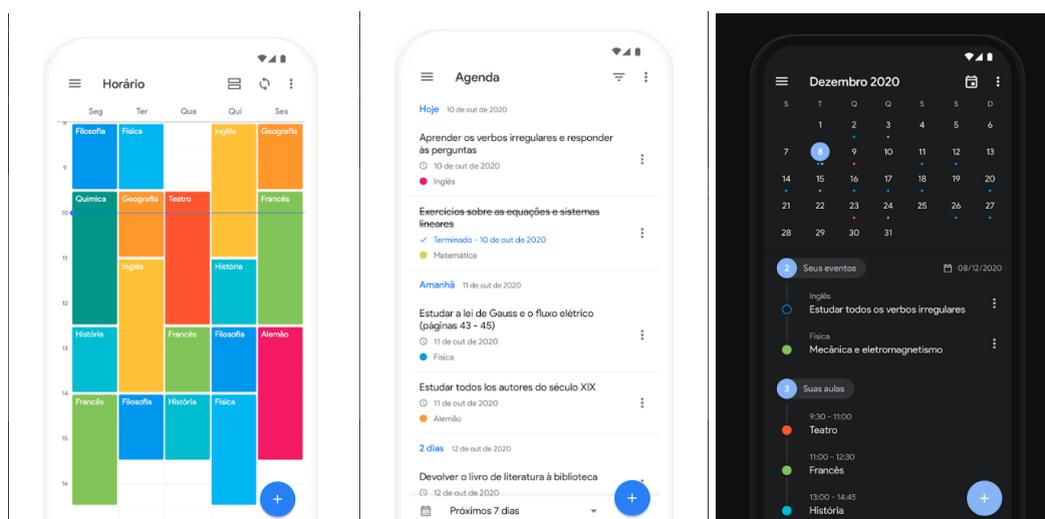
FONTE: O autor (2021)

Assim como em uma agenda, as tarefas e eventos que estão próximos ganham destaque no aplicativo e também ele mostra de maneira gráfica a quantidade de eventos por dia, sendo possível observar rapidamente a carga de trabalho da semana. Ele também tem um calendário para observar os eventos cadastrados e uma opção interessante de gravar e armazenar áudios dentro do aplicativo. As opções de visualização das informações do aplicativo se dá através de páginas diferentes para cada uma delas, por exemplo, ele tem uma página específica para os horários, agendas, calendário e disciplinas.

As opções de customização do aplicativo não incluem modificação do tamanho da fonte. O modo escuro está presente e na versão paga do aplicativo é possível construir mais de uma agenda no mesmo dispositivo e ter acesso a tudo o que o aplicativo fornece.

A Figura 6 apresenta algumas páginas relevantes do aplicativo.

FIGURA 6 - PÁGINAS DO APLICATIVO AGENDA ESCOLAR



FONTE: GOOGLE PLAY, 2021

LEGENDA: a) Página da grade horária
b) Página da agenda
c) Página do calendário

3.5 APLICATIVO AGENDA DO ESTUDANTE PRO - ORGANIZE-SE!

Com um calendário e uma visão gráfica dos eventos semanais, que inclui os horários e eventos cadastrados, o aplicativo permite relacionar um nome, cor e professor a uma disciplina (TABELA 5). O ponto chave do aplicativo é que as disciplinas não são vinculadas diretamente a algum horário, mas os horários são

relacionados a alguma disciplina e também nele é possível colocar a sala e um lembrete de livre escrita. Então, para cadastrar o horário e outras informações já citadas editamos a própria grade horária (GOOGLE PLAY, 2021e).

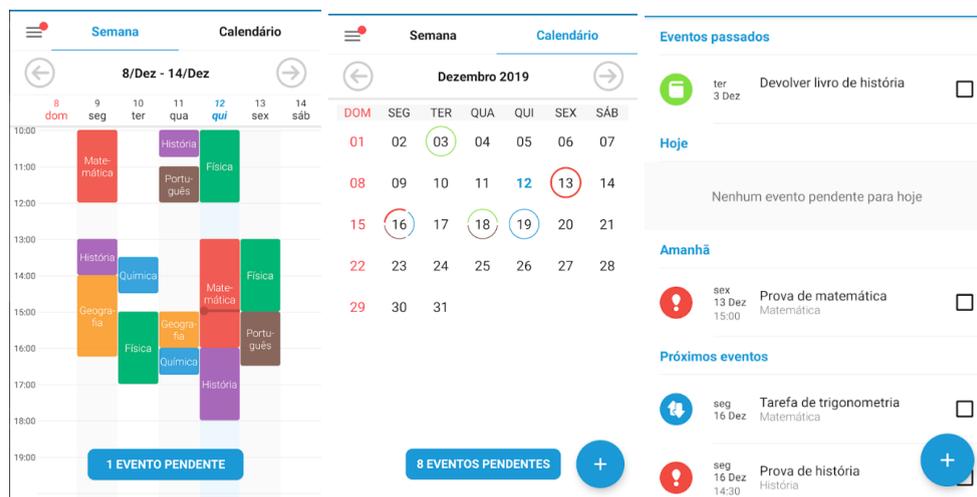
TABELA 5 - APLICATIVO AGENDA DO ESTUDANTE PRO - ORGANIZE-SE!

Aplicativo	Agenda do estudante pro - Organize-se!
Nota	4,7
Downloads	+1 mi
Loja	Play Store
Desenvolvedor	Apps CC

FONTE: O autor (2021)

Uma aba específica para a anotação de notas das disciplinas também está presente no aplicativo, nela é possível listar cada nota relacionada às disciplinas cadastradas anteriormente, além de selecionar um peso e informações adicionais para cada nota.

FIGURA 7 - PÁGINAS DO APLICATIVO AGENDA DO ESTUDANTE PRO - ORGANIZE-SE!



FONTE: GOOGLE PLAY, 2021

LEGENDA: a) Página da grade horária
b) Página do calendário
c) Página de eventos

Em relação às opções de customização, o aplicativo apresenta a possibilidade de relacionar cores diferentes para cada um dos eventos (prova,

tarefa/trabalho, devolução de livro e outros), disciplinas e do widget, porém não têm a possibilidade de alterar a cor da aplicação em si para uma de contraste ou também modificar o tamanho da fonte.

A Figura 7 apresenta algumas páginas relevantes do aplicativo.

3.6 APLICATIVO SIGGA MOBILE

Criado pela Universidade Federal do Rio Grande do Norte - UFRN para uso interno de docentes e discentes, o aplicativo (TABELA 6) permite o acesso a informações diretamente do sistema de gestão da universidade automaticamente. O aluno tem acesso às suas informações de vínculo com a instituição, além de suas frequências nas aulas e disciplinas matriculadas, tudo isso sem a necessidade de conexão com a internet, exceto para a atualização das informações (GOOGLE PLAY, 2021f).

TABELA 6 - APLICATIVO SIGGA MOBILE

Aplicativo	Sigga Mobile
Nota	3,7
Downloads	+100 mil na Play Store
Loja	Play Store e App Store
Desenvolvedor	Superintendência de Informática da UFRN

FONTE: O autor (2021)

O sistema permite que o aluno receba notificações das disciplinas enviadas pelo docente, além de acessar as notícias e outras informações das disciplinas (FIGURA 8). Os eventos cadastrados, como os horários de aula e provas são registrados no calendário presente no aplicativo, sendo esse integrado ao Google Agenda. É possível também ter acesso ao sistema da biblioteca da universidade, sendo possível ver o acervo disponível e gerenciar os empréstimos realizados dos materiais.

A Figura 8 apresenta algumas páginas relevantes do aplicativo.

FIGURA 8 - PÁGINAS DO APLICATIVO SIGGA MOBILE



FONTE: GOOGLE PLAY, 2021

LEGENDA: a) Página do fórum
 b) Página da disciplina
 c) Página do calendário

3.7 APLICATIVO SIGAUFGM

Mais um aplicativo desenvolvido por uma Instituição pública, agora pela Universidade Federal de Minas Gerais, com o objetivo de fornecer um sistema para acesso das informações acadêmicas aos alunos (TABELA 7). As disciplinas matriculadas e todas as informações relacionadas a elas estão acessíveis, além de ocorrências, planos de estudo, notas, frequências e documentos, como o comprovante de matrícula (GOOGLE PLAY, 2021g).

TABELA 7 - APLICATIVO SIGAUFGM

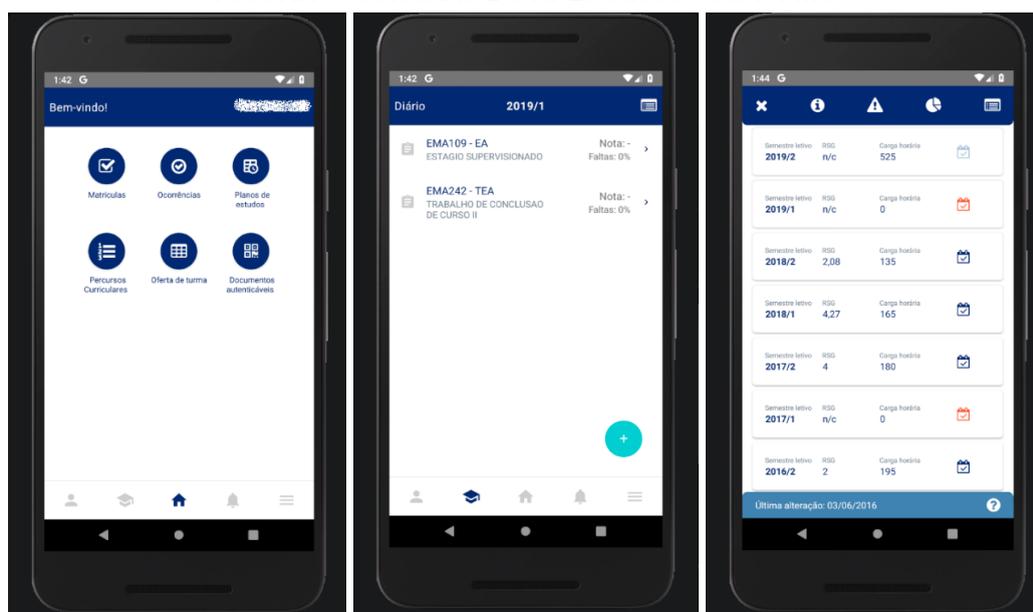
Aplicativo	SigaUFMG
Nota	3,5 na Play Store
Downloads	+10 mil na Play Store
Loja	Play Store e App Store
Desenvolvedor	DTI UFGM

FONTE: O autor (2021)

Como tudo é importado automaticamente do sistema após autenticação, não é possível realizar qualquer tipo de modificação ou anotação dentro do aplicativo, sendo ele especificamente para a visualização dos dados cadastrados no sistema de gerência da universidade.

A Figura 9 apresenta algumas páginas relevantes do aplicativo.

FIGURA 9 - PÁGINAS DO APLICATIVO SIGAUFMG



FONTE: GOOGLE PLAY, 2021

LEGENDA: a) Página inicial

b) Página das disciplinas

c) Página do histórico

3.8 APLICATIVO UFU MOBILE

Assim como os dois sistemas anteriores apresentados, o *UFU Mobile* (TABELA 8) visa expor as informações acadêmicas aos discentes, nesse caso da Universidade Federal de Uberlândia. Diferente de todos os outros aplicativos, ele traz informações do restaurante universitário da instituição e do transporte interno entre os *campus* (intercampi), notícias e comunicados de toda comunidade acadêmica, mapas das instalações da faculdade e calendário acadêmico (GOOGLE PLAY, 2021h).

TABELA 8 - APLICATIVO UFU MOBILE

Aplicativo	UFU Mobile
Nota	3,5 (Play Store)
Downloads	+10 mil na Play Store
Loja	Play Store e App Store
Desenvolvedor	Centro de Tecnologia da Informação CTI/UFU

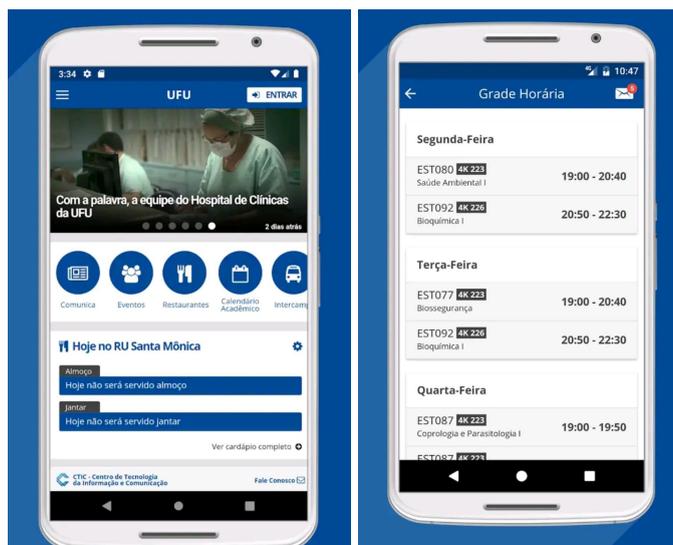
FONTE: O autor (2021)

Em relação às disciplinas, foco do presente trabalho, o aplicativo lista, por dia, as disciplinas do aluno matriculado (acesso após realizar um *login*) com nome, código e horário delas, além do histórico das disciplinas dos semestres anteriores com as notas e frequências obtidas.

Como não foi possível testar o aplicativo por causa da autenticação necessária, ele requer um cadastro na instituição, as informações sobre as opções de customização e acessibilidade do sistema não puderam ser obtidas.

A Figura 10 apresenta algumas páginas relevantes do aplicativo.

FIGURA 10 - PÁGINAS DO APLICATIVO UFU MOBILE



FONTE: GOOGLE PLAY, 2021

LEGENDA: a) Página inicial do aplicativo
b) Página das disciplinas

Para facilitar a comparação dos aplicativos, foi construída a Tabela 9 a seguir, ressaltando alguns pontos considerados relevantes.

TABELA 9 - COMPARAÇÃO DE FUNCIONALIDADE DOS APLICATIVOS ANALISADOS (continua)

Nome do aplicativo	Possui anúncios	Alerta / Notificações	Acessibilidade	Widget	Grade horária
Horário de aula	sim	sim	não	sim	sim
Timetable	não	sim	modo escuro	sim	sim
Class Timetable	sim	sim (pago)	não	sim	não
Agenda escolar	sim	sim	modo escuro	sim	sim
Agenda do estudante pro	sim	sim	não	sim	sim
SIGA Mobile	não	-	-	não	sim
SigaUFMG	não	-	-	não	não
UFU Mobile	não	-	-	não	não

TABELA 9 - COMPARAÇÃO DE FUNCIONALIDADE DOS APLICATIVOS ANALISADOS (conclusão)

Nome do aplicativo	Eventos	Anotações	Notas	Importar pdf	Informações
Horário de aula	sim	não	sim	não	nome, dia, horário, professor, sala
Timetable	sim	não	não	não	nome, dia, horário, professor, sala
Class Timetable	sim	sim	não	não	nome, dia, horário
Agenda escolar	sim	sim	sim	não	nome, dia, horário, sala, professor
Agenda do estudante pro	sim	sim	sim	não	nome, professor
SIGAA Mobile	sim	não	sim (automaticamente)	preenchimento automático	nome, dia, horário, professor, código, carga, frequência
SigaUFMG	não	não	sim (automaticamente)	preenchimento automático	nome, dia, horário, professor código, carga, frequência
UFU Mobile	não	não	sim (automaticamente)	preenchimento automático	nome, dia, horário, professor código, carga, frequência

FONTE: O autor (2021).

4 ORGANIZA UFPR

O aplicativo foi construído com a versão 2.9 da linguagem Dart, com o Flutter 2.2.2 e Android Studio versão 4.2.1. O nome escolhido para ele foi Organiza UFPR com o intuito de indicar sua principal funcionalidade.

Sobre a implementação, o Dart, assim como as linguagens C e C++, suporta funções de nível superior, no caso a função `main()`, o exemplo mais comum desse tipo de função, é obrigatória em qualquer programa, servindo de ponto de entrada para o aplicativo. A seguir (FIGURA 11) temos um recorte da função `main()` do *software* produzido.

FIGURA 11 - FUNÇÃO MAIN DO ORGANIZA UFPR

```
void main() => runApp(  
  ChangeNotifierProvider(  
    create: (BuildContext context) => ThemeModel(),  
    child:  
      ChangeNotifierProvider(  
        create: (BuildContext context) => FontModel(),  
        child: MyApp()  
      ) // ChangeNotifierProvider  
    ) // ChangeNotifierProvider  
);
```

FONTE: O autor (2021).

A partir dela podemos ver algumas características do Flutter/Dart. Primeiramente acho interessante falar sobre o operador “=>”, ele é uma abreviação da tecnologia que pode ser utilizado no lugar de “{ return expr; }” para funções que contém apenas uma expressão. Nesse caso a função `main()` possui apenas a expressão `runApp(...)`, que é uma função especial do Flutter que coloca o *widget* passado por parâmetro na raiz da árvore de *widgets* - conhecida por *Widget Tree* -, ele será o primeiro elementos de interação e todos os outros serão filhos dele, além de que o aplicativo será construído a partir dele.

A classe `ChangeNotifierProvider` é responsável por ouvir uma classe estendida de `ChangeNotifier` - classe notificadora de mudança -, expor seus dados aos seus filhos e reconstruí-los quando a função específica for chamada. No presente trabalho, dois objetos dessa classe foram usados para controlar o tema (cor) e o tamanho da fonte dos textos da aplicação, dois dos itens possíveis de se customizar na aplicação, os quais serão abordados posteriormente. Podemos ver

também a forma como passamos os parâmetros para os construtores das classes, nesse caso o “create” e “child” são parâmetros do construtor.

Por último temos a classe `BuildContext`, que serve para localizar *widgets* na *Widget Tree*. Ele é usado com o `context` para guardar o contexto em que um *widget* específico foi criado, sendo que todo *widget* tem um `BuildContext` vinculado, porém, às vezes ele não necessariamente está exposto no código. É importante ressaltar também que os *widgets* filhos têm acesso ao contexto do seu pai, a partir daí conseguimos “carregar” informações dentre os nodos da árvore de *widgets*. No caso do aplicativo utilizamos o contexto para propagar as informações da fonte e tema do aplicativo.

O último filho exposto no main é a classe `MyApp()`, a Figura 12 mostra essa classe. A primeira questão dessa classe é que ela é uma extensão da `StatelessWidget`, classe de *widgets* que não possibilita alterações dinâmicas em tempo de execução, no caso, ela não tem controle de estado. O Estado pode ser entendido como elementos que podem ser alterados por alguma ação do usuário, logo, um `StatelessWidget` é caracterizado por elementos fixos na aplicação, que não incluem nenhum tipo de entrada de dados.^[10]

FIGURA 12 - FUNÇÃO MYAPP DO ORGANIZA UFPR

```
class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Organiza UFPR',
      theme: Provider.of<ThemeModel>(context).currentTheme,
      home: StartPage(),
    ); // MaterialApp
  }
}
```

FONTE: O autor (2021).

A `MaterialApp` é uma classe base para um aplicativo que usa a material design - linguagem de design do Google -, ela é responsável por fornecer funções de interação com o sistema operacional, por exemplo, vincular o botão de voltar do sistema para encerrar o aplicativo. Nessa parte do código também passamos o nome do *app* no sistema - Organiza UFPR -, além de setar o tema inicial do

aplicativo, resgatado utilizando o método *Provider.of<T>(BuildContext context)*, o qual vai olhar a *Widget Tree* partindo do *widget* associado ao contexto passado por parâmetro e retornará a variável mais próxima - de nível superior - do tipo T, com isso conseguimos pegar a classe *ThemeModel()* passada anteriormente no main.

A classe *StartPage* é a responsável por gerar a página inicial do aplicativo, é possível vê-la na Figura 13. Outra estrutura interessante é o *Scaffold*, classe que implementa a estrutura básica de *layout* do material design. ele seria a base para começar a tratar o layout da aplicação. Ele possui como característica preencher todo espaço disponível, logo, ele ocupa toda a tela do dispositivo, independente do tamanho, auxiliando para que o aplicativo seja responsivo em diferentes dispositivos.

FIGURA 13 - PÁGINA INICIAL DO ORGANIZA UFPR



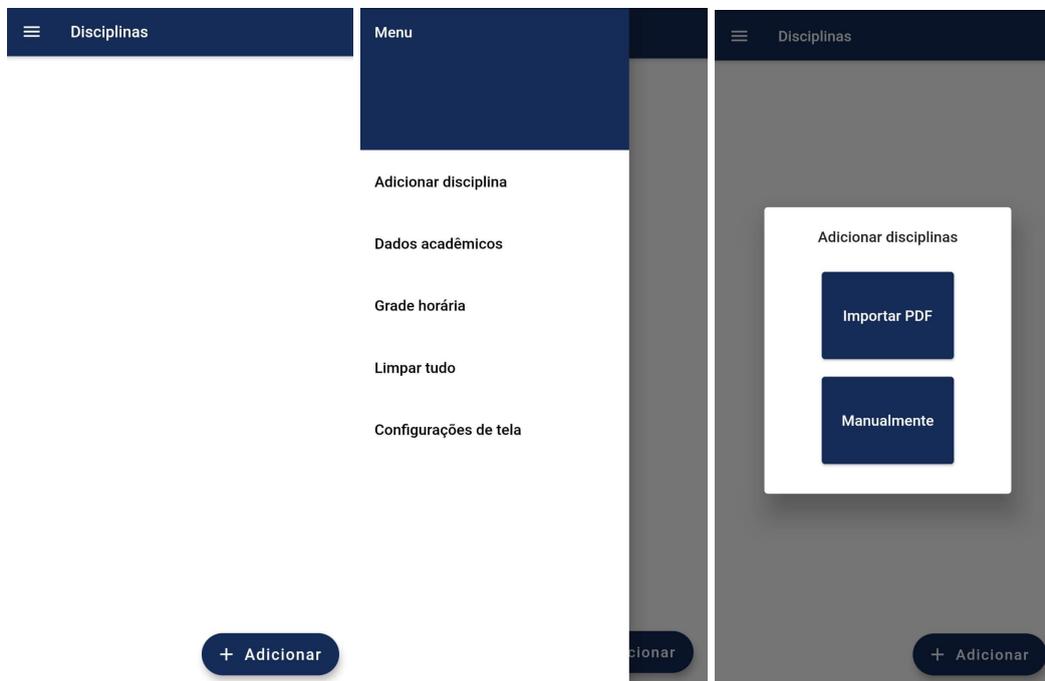
FONTE: O autor (2021).

LEGENDA: a) Código da página inicial
b) Visual da página inicial

Foi estabelecido que a transição da página inicial para a página intitulada como disciplinas - principal página do aplicativo - seja feita após dois segundos, isso para permitir que as informações sejam carregadas sem a percepção do usuário. Além das duas páginas já citadas temos também a página de “grade horária”, então, o aplicativo é composto por essas três páginas. É na página de disciplinas (FIGURA 14) que o menu principal pode ser acessado (FIGURA 14), as informações das disciplinas podem ser vistas e modificadas, anotações podem ser realizadas, os

dados acadêmicos podem ser acessados, entre outras funções do aplicativo que serão apresentados no decorrer do trabalho.

FIGURA 14 - PÁGINA DE DISCIPLINAS DO ORGANIZA UFPR



FONTE: O autor (2021).

LEGENDA: a) Página de disciplinas inicial
b) Menu principal
c) Recurso adicionar

Um dos recursos mais interessantes disponíveis no aplicativo é a importação dos dados das disciplinas a partir do comprovante de matrícula dos cursos de graduação da UFPR disponibilizado aos estudantes pelo SIGA - Sistema Integrado de Gestão Acadêmica. Ao clicar no botão “Adicionar” - canto inferior direito - uma caixa de diálogo com duas opções (FIGURA 14) nos dá a possibilidade de selecionar e importar o pdf, que já deve estar baixado no dispositivo.

Por se tratar de tecnologias do Google e ser direcionado ao desenvolvimento *mobile*, o Flutter tem pacotes que facilitam demais a interação com o hardware de dispositivos Android. Para o caso da importação do PDF, foi usado o pacote “file_picker”, que permite o uso do explorador de arquivos nativo do dispositivo para selecionar e importar um arquivo dentro da aplicação. O recorte da função de importar pdf (FIGURA 15) nos dá uma ideia de como esse pacote é usado, sendo possível constatar que a ação é realizada em apenas três linhas de código.

FIGURA 15 - RECORTE DA FUNÇÃO DE IMPORTAR PDF

```
Future _importPDF() async {  
  var filePickerResult = await FilePicker.platform.pickFiles();  
  
  if (filePickerResult != null)  
    _pdfDoc = await PDFDoc.fromPath(filePickerResult.files.single.path);  
  
  String text = await _pdfDoc.text;
```

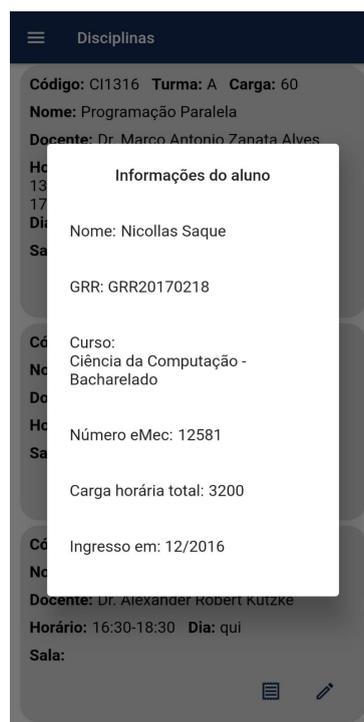
FONTE: O autor (2021).

Outras peculiaridades da linguagem podem ser observadas com o trecho de código anterior, no caso, o retorno `Future`, que representa um valor potencial que estará disponível em algum momento no futuro e a declaração de função `async` e o `await`, itens relacionados a programação assíncrona, que tem por objetivo permitir que a aplicação continue sua execução sem ficar travada por itens de acesso externo ou que levam algum tempo para executar, no código, caracterizado pela seleção do PDF a ser importado.

Em relação ao tratamento do PDF após a importação, podemos ver na Figura 15 - última linha de código - que ele é transformado em uma única `String`, para isso foi usado o plugin `pdf_text`. Essa string, então, é “recortada” para extrairmos as informações que nos interessam. Primeiramente, são extraídos os “Dados acadêmicos” (FIGURA 16), dados que são preenchidos automaticamente e são acessados a partir do menu na página de disciplinas (FIGURA 16).

FIGURA 16 - DADOS ACADÊMICOS

IDENTIFICAÇÃO					
NOME			MATRÍCULA		
NICOLLAS SAQUE			GRR20170218		
CPF	RG	ÓRGÃO EMISSOR	DATA NASCIMENTO	UF NASCIMENTO	NACIONALIDADE
10825136954	91840910	Não informado	05/12/1996	Paraná	BR
CURSO					
CURSO			NÚMERO eMEC	CARGA HORÁRIA TOTAL	
Ciência da Computação - Bacharelado			12581	3200	
HABILITAÇÃO					
Ciência da Computação					
FORMA INGRESSO	MÊS/ANO INGRESSO	DATA CONCLUSÃO	DATA DA COLAÇÃO	DATA DE EXPEDIÇÃO DO DIPLOMA	
Reopção	12 / 2016	-	-	-	
SITUAÇÃO ALUNO		CURRÍCULO	PERÍODO ATUAL		
Matriculado		2019	7		



FONTE: O autor (2021).

LEGENDA: a) Recorte do comprovante de matrícula da UFPR
b) Tela do Organiza UFPR

A extração das informações é feita de tal forma: primeiro indicamos o limite superior e inferior - que são *substrings* dentro da *string* que representa todo o pdf -, assim temos uma *string* reduzida com todas as informações que queremos, com isso, essa *string* é transformada em uma lista onde cada palavra é um item da lista. Temos conhecimento que essa lista seguirá o padrão cabeçalho seguido da informação, com isso, é só percorrer essa lista item a item e ir armazenando o dado na variável correspondente, por exemplo, sabemos que sempre na primeira posição da lista estará a string “NOME” e da segunda posição até a posição de conteúdo “MATRÍCULA” - que é o próximo campo - estará o nome do(a) estudante, independente do tamanho do sobrenome.

Para os dados das disciplinas (FIGURA 17) é usada a mesma ideia de extração dos dados, porém há uma dificuldade a mais pois o padrão que a string fica não é cabeçalho seguido de informação e sim informação seguido de informação, isso pelos dados estarem representados por meio de uma tabela. Para resolver isso é adicionado o carácter “#” antes do início de cada linha - antes do código da disciplina e dos dias da semana - para termos noção onde é o início de

cada conjunto de dados. Esse procedimento é realizado separadamente para os dados das disciplinas e da grade horária, então temos duas listas diferentes.

FIGURA 17 - DADOS DAS DISCIPLINAS

DISCIPLINAS					
PERÍODO					
2° Semestre/2020					
CÓDIGO	NOME DA DISCIPLINA	TURMA	CH	SITUAÇÃO	DOCENTE E TITULAÇÃO
DS122	DESENVOLVIMENTO WEB I	V2	60	Matriculado	Dr. ALEXANDER ROBERT KUTZKE
CI1238	OTIMIZAÇÃO	B	60	Matriculado	Dr. ANDRÉ LUIZ PIRES GUEDES
CI1316	PROGRAMAÇÃO PARALELA	A	60	Matriculado	Dr. MARCO ANTONIO ZANATA ALVES
CI1098	TRABALHO DE CONCLUSÃO DE CURSO 1 EM INFORMÁTICA NA EDUCAÇÃO	ARP	150	Matriculado	Dr. ANDREY RICARDO PIMENTEL

GRADE HORÁRIA							
DIA SEMANA	HORA INÍCIO	HORA FIM	DATA INÍCIO	DATA FIM	CÓDIGO	TURMA	NOME DISCIPLINA
Segunda-Feira	13:30	14:30	02/03/2020	27/06/2020	CI1238	A	OTIMIZAÇÃO
Segunda-Feira	14:30	15:30	02/03/2020	27/06/2020	CI1238	A	OTIMIZAÇÃO
Terça-Feira	13:30	14:30	04/05/2021	30/07/2021	CI1316	A	PROGRAMAÇÃO PARALELA
Terça-Feira	14:30	15:30	04/05/2021	30/07/2021	CI1316	A	PROGRAMAÇÃO PARALELA
Terça-Feira	17:30	18:30	02/03/2020	27/06/2020	CI1316	A	PROGRAMAÇÃO PARALELA
Terça-Feira	18:30	19:30	02/03/2020	27/06/2020	CI1316	A	PROGRAMAÇÃO PARALELA
Quarta-Feira	13:30	14:30	02/03/2020	27/06/2020	CI1238	A	OTIMIZAÇÃO

Disciplinas		
Código: CI1316	Turma: A	Carga: 60
Nome: Programação Paralela		
Docente: Dr. Marco Antonio Zanata Alves		
Horário: 13:30-15:30, 17:30-19:30, 13:30-15:30, 17:30-19:30		
Dia: ter, ter, qui, qui		
Sala:		
 		
Código: CI1238	Turma: B	Carga: 60
Nome: Otimização		
Docente: Dr. André Luiz Pires Guedes		
Horário: 19:30-21:30		
Dia: qui		
Sala:		
 		
Código: DS122	Turma: V2	Carga: 60
Nome: Desenvolvimento Web I		
Docente: Dr. Alexander Robert Kutzke		
Horário: 16:30-18:30		
Dia: qui		
Sala:		
 		

FONTE: O autor (2021).

LEGENDA: a) Recorte do comprovante de matrícula da UFPR
b) Tela do Organiza UFPR

Para juntar as duas listas são usadas as informações do código e da turma das disciplinas para mapear as informações de uma tabela na outra. Os dados de código, turma, carga, nome e docente são retirados da tabela disciplinas do pdf e o horário e dia são retirados da tabela grade horária. A partir disso, cada uma das informações são transformadas separadamente em um *widget* do tipo texto para facilitar a modificação dos dados e aparição na tela para o usuário.

Em paralelo a tudo isso, as informações das disciplinas e os dados acadêmicos são salvos no armazenamento interno do celular, mantendo a integridade dos dados quando o aplicativo é encerrado. Para isso usamos o *plugin* `shared_preferences`, responsável por guardar dados simples utilizando a opção nativa do Android chamada de Shared Preferences, onde os dados ficam guardados em um XML dentro do aplicativo. Eles são guardados como pares de chave/valor, onde a chave é usada para recuperar o valor. Na Figura 18 podemos observar como

é salvo os dados acadêmicos, primeiro pegamos as instâncias das Shared Preferences e após construir uma lista de strings onde cada posição é uma informação, salvamos utilizando o método `setStringList`, passando o valor da chave e a lista em si com os dados.

FIGURA 18 - FUNÇÃO DE SALVAR OS DADOS ACADÊMICOS

```
void addInfoToList(aluno, grr, curso, emec, cargaTotal, ingresso) async {
  final prefs = await SharedPreferences.getInstance();
  infos[0] = aluno;
  infos[1] = grr;
  infos[2] = curso;
  infos[3] = emec;
  infos[4] = cargaTotal;
  infos[5] = ingresso;
  prefs.setStringList('infos', infos);
}
```

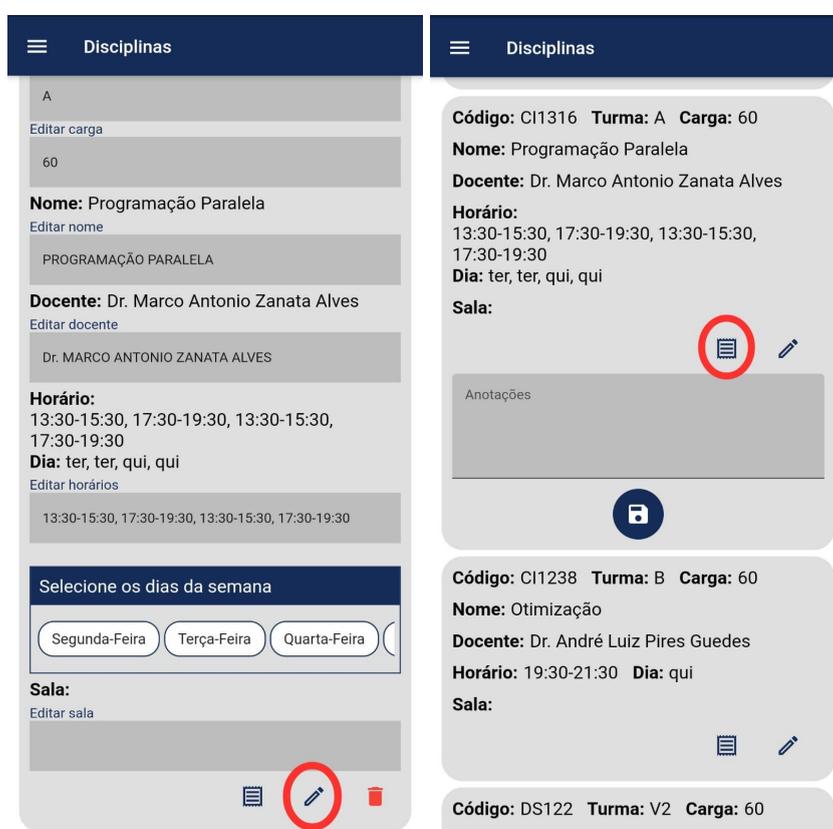
FONTE: O autor (2021).

Também salvamos da mesma maneira as anotações das disciplinas - recurso disponível para cada disciplina -, já para os dados das disciplinas a única diferença no processo de salvar é que transformamos os dados para um Json, optou-se por esse formato pela facilidade do Dart em codificar e decodificar os dados. Como optou-se por salvar os dados de todas as disciplinas juntas, para não ter um número muito grande de chaves, precisávamos salvar uma lista de listas, onde cada posição é uma lista com os dados das disciplinas, porém o Shared Preferences só aceita salvar dados dos tipos `Int`, `Bool`, `Double`, `String` e `StringList`, logo, houve a necessidade de transformar os dados de uma disciplina para Json - formato de texto - e assim foi possível salvar os dados como uma `StringList`, onde cada posição da lista são os dados de uma única disciplina no formato Json.

Além dos dados acadêmicos, das anotações e informações das disciplinas, é guardado o tamanho da fonte de texto e tema do aplicativo, como `Double` e `String`, respectivamente. Toda vez que o aplicativo é iniciado, então, temos que recuperar todos esses dados. A partir dos métodos `get` (`getStringList`, `getDouble` e `getString`) e passando a respectiva chave de cada dado, recuperamos as informações e assim é possível colocá-las dentro das variáveis que em seguida serão usadas pelos *widgets* para expor os dados na tela para o usuário.

Ainda relacionado às disciplinas, temos uma outra funcionalidade do aplicativo, que é a possibilidade de editar as suas informações. A Figura 19 mostra as caixas de entrada de texto que são abertas após clicar no ícone do “lápis” dentro do espaço demarcado para cada disciplina. A partir daí é possível editar as informações que estão vazias e que foram importadas automaticamente, podendo cada indivíduo deixar da maneira que lhe agrade mais as informações das disciplinas, exceto os campos de dia e horário, que pelo motivo de serem usados para construir a grade horária foi necessário que seguissem um padrão.

FIGURA 19 - RECURSOS DO ORGANIZA UFPR



FONTE: O autor (2021).

LEGENDA: a) editar informação das disciplinas
b) anotações de uma disciplina

Outro recurso já citado anteriormente é a caixa de anotações (FIGURA 19), onde cada disciplina pode ter suas anotações próprias. Qualquer informação como nota, data de provas e trabalhos podem ser salvas e rapidamente serem acessadas com um mínimo de ações do usuário. Aliás, no desenvolvimento da aplicação, houve uma grande preocupação com a interação do usuário, e como resultado

temos que qualquer recurso que o aplicativo forneça pode ser realizado com no máximo três toques na tela, tanto acessar as informações, modificá-las, importar dados, mudar as configurações da tela, entre outros.

A atualização dos dados, tanto das anotações ou de alguma modificação nas informações das disciplinas são feitas pela classe `setState`. Essa classe notifica a estrutura que ele está dentro que o estado do objeto que teve o valor modificado dentro do `setState` foi alterado, com isso, força a interface atualizar com os novos valores. Um exemplo de uso está sendo apresentado na Figura 20, onde primeiro é usado o `setState` para indicar a mudança da anotação de uma disciplina e após, essa nova anotação é salva no armazenamento interno do telefone.

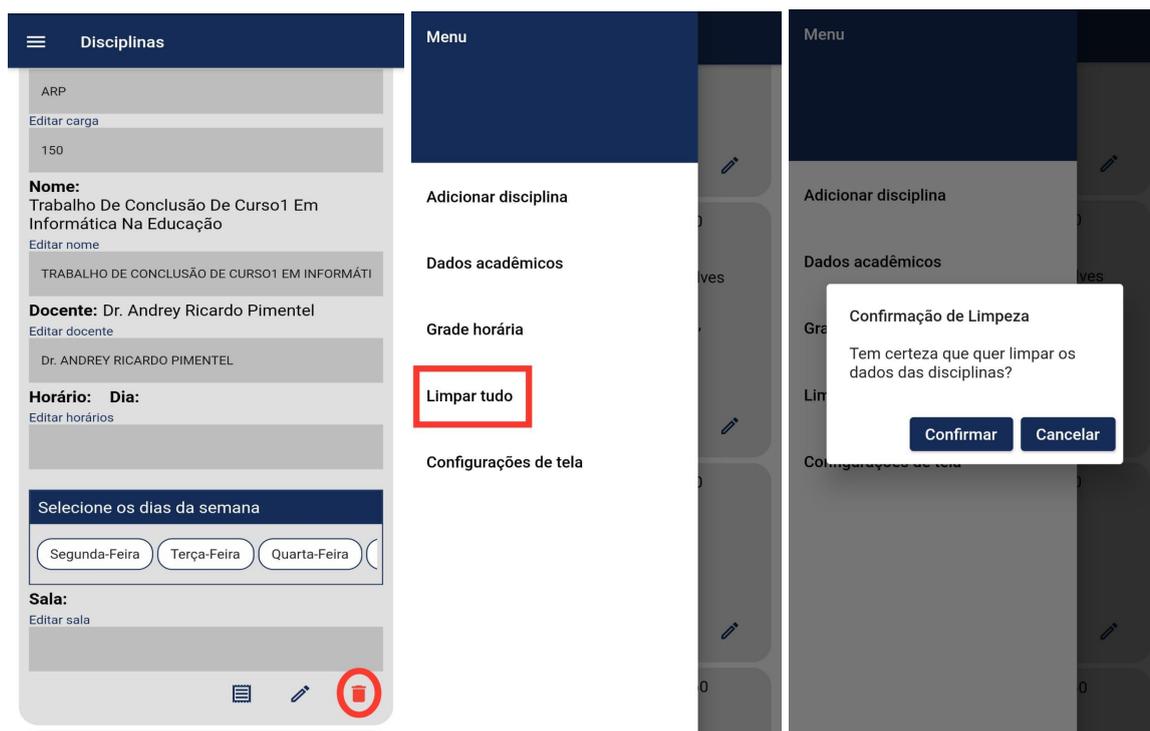
FIGURA 20 - FUNÇÃO DE SALVAR ANOTAÇÕES

```
void saveNotes(String note, int index) async {  
  final prefs = await SharedPreferences.getInstance();  
  setState(() {  
    widget.savents[index] = note;  
  });  
  prefs.setStringList('notes', widget.savents);  
}
```

FONTE: O autor (2021).

É possível também excluir uma disciplina específica - após clicar no botão de editar (lápis) aparece um botão vermelho com um ícone de lixeira (FIGURA 21) - ou limpar todas as disciplinas - a partir do menu (FIGURA 21). As ações críticas dentro do aplicativo, que incluem a exclusão de uma disciplina e de todas as disciplinas possuem uma segunda confirmação (FIGURA 21) para evitar que acidentalmente os dados sejam apagados, pois se confirmada a ação não é possível reverter, porque além da interface ser atualizada sem as informações, os dados salvos no dispositivos também são atualizados, apagando os dados necessários.

FIGURA 21 - LIMPAR DADOS DISCIPLINAS

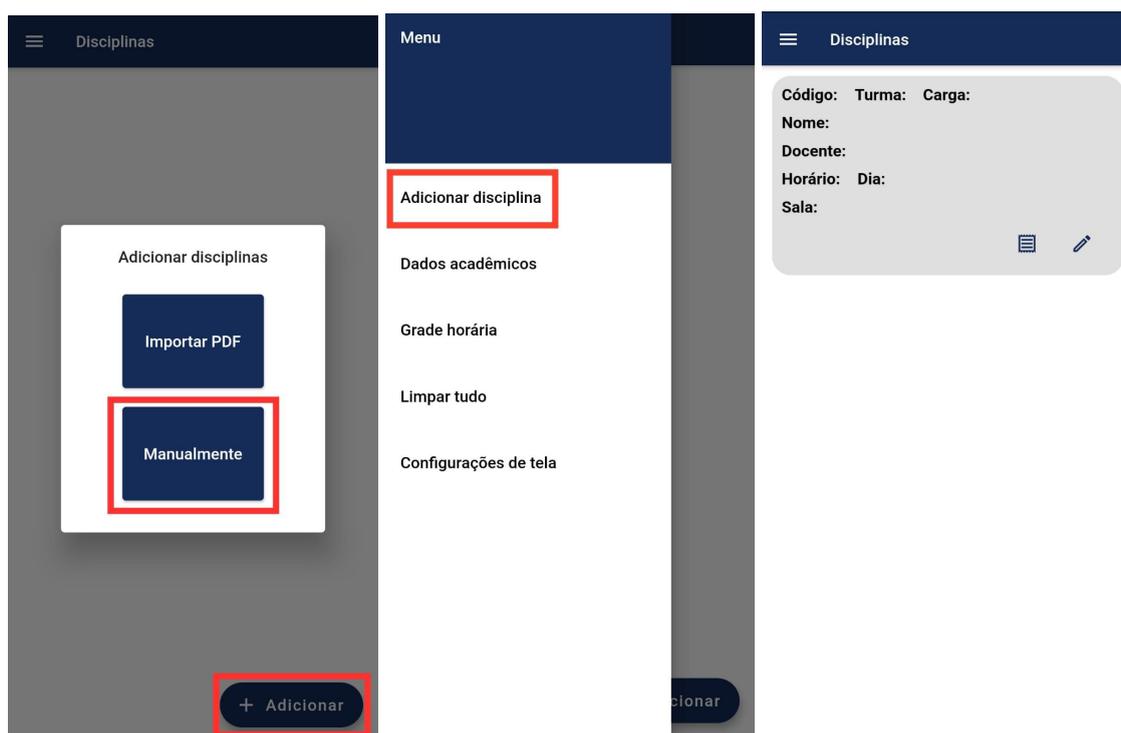


FONTE: O autor (2021).

LEGENDA: a) opção apagar uma disciplina
 b) opção apagar todas as disciplinas
 c) confirmação de exclusão

O aplicativo não funciona somente com a importação dos dados do comprovante de matrícula, é possível criar uma disciplina “manualmente” - opção ao clicar no botão adicionar do canto inferior direito (quando a lista de disciplinas está vazia), o resultado dessa ação pode ser vista na Figura 22, ou no menu por “Adicionar disciplina” (FIGURA 22) -, onde uma disciplina com todas informações vazias é criada, com isso, o usuário pode editar as informações. Essa ação é interessante para adicionar disciplinas fora do âmbito acadêmico, como aulas de línguas, esportes e instrumentos, por exemplo. Em relação a codificação dessa parte, a criação da turma vazia é exatamente igual a criação com os dados do PDF, obviamente, ao criar a turma vazia são passados todos os parâmetros vazios para a função responsável por criar a disciplina.

FIGURA 22 - CRIAR UM DISCIPLINA



FONTE: O autor (2021).

LEGENDA: a) opção adicionar disciplina (quando vazia)
 b) opção adicionar disciplina pelo menu
 c) disciplina vazia criada

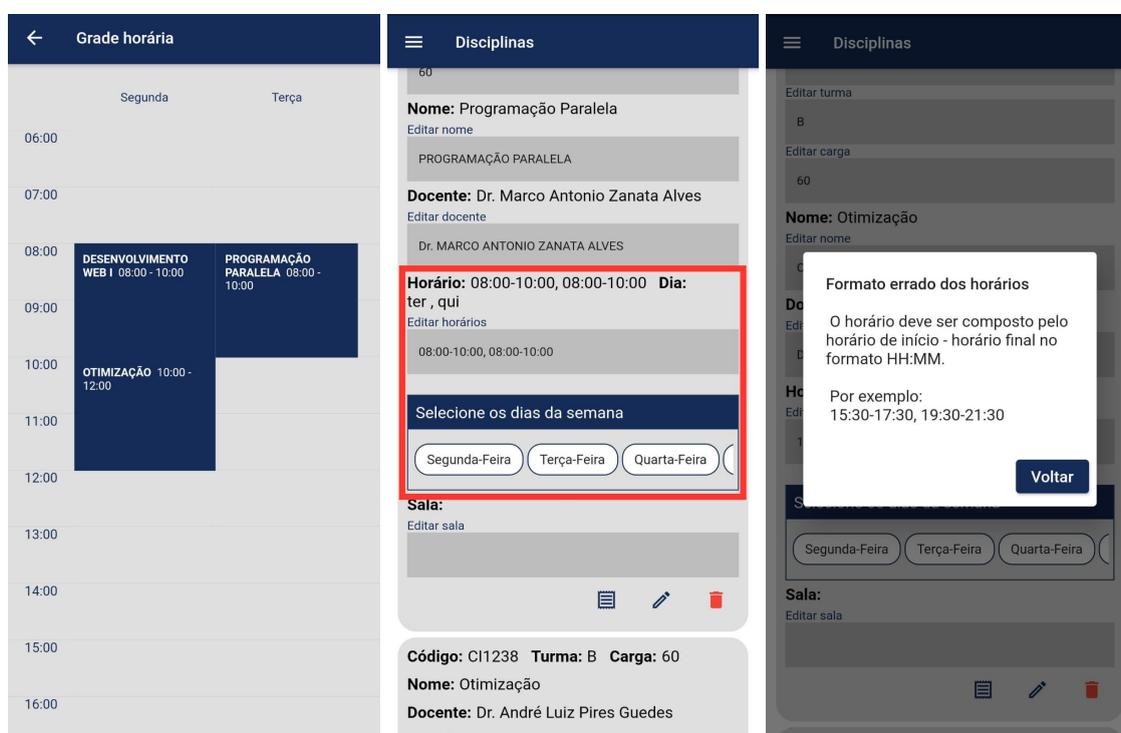
A visualização dos horários no formato de grade horária é mais um dos recursos implementados no aplicativo (FIGURA 23), o pacote `flutter_timetable_view` foi utilizado para implementá-la, as principais mudanças realizadas na estrutura fornecida foram relacionadas a parte estética. A partir dos horários inseridos na página de disciplinas a grade é gerada em uma outra página que pode ser acessada a partir do menu. Como já citado anteriormente, a construção da grade horária foi o grande motivo para os campos “horário” e “dia” requererem um padrão de preenchimento. Em relação ao campo dia, optou-se pela estrutura de seletor múltiplo (FIGURA 23) oferecido pelo pacote `multi_select_flutter`, pois temos um conjunto finito e pequeno de possibilidade - somente 7 dias da semana. O seletor, além de obrigar que os dados estejam em um padrão favorece também o usuário, pois a seleção de uma caixa é muito mais rápida do que a escrita de alguma informação.

Para o horário, foi decidido manter o formato de escrita (FIGURA 23), pois outras estruturas testadas, como a caixa de rolagem e o relógio de ponteiro -

implementações comuns em aplicativos - tiveram alguns problemas como: ocupar muito espaço dentro da estrutura delimitadora de cada disciplina, ser mais lento para o usuário selecionar, pois cada horário deve ser composto por horário de início e horário final, logo para cada dia que a disciplina tem horário, dois horários deveriam ser selecionados, nos casos mais comuns - disciplinas com aulas dois dias na semana -, quatro horários deveriam ser selecionados.

Algumas restrições foram feitas para evitar erros no *input* dos dados e por consequência ajudar o usuário, dentre elas, forçar que o teclado numérico apareça ao editar os horários, evitando a inserção de letras, permitir somente a inserção de números, não aceitando caracteres que aparece no teclado numérico, como vírgula, ponto, espaço e traço - eles são preenchidos automaticamente pois o padrão usa vírgula e traço -, permitir apenas os números 0,1 e 2 no início dos horários, evitando vários horários inválidos. Também ao inserir um horário fora do padrão, o usuário recebe um alerta de erro, indicando como deve ser o *input* correto dos dados (FIGURA 23).

FIGURA 23 - GRADE HORÁRIA, DIA E HORÁRIOS

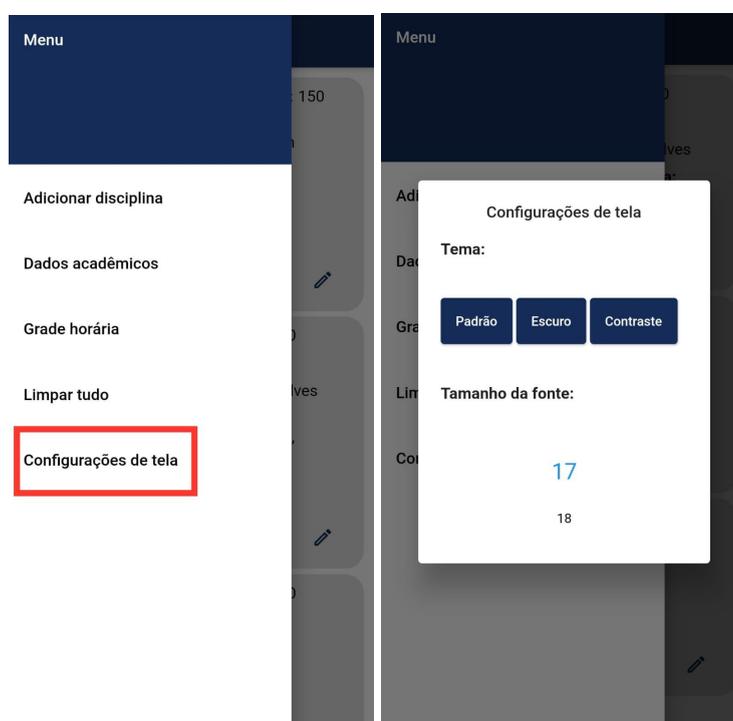


FONTE: O autor (2021).

LEGENDA: a) página da grade horária
b) editor de dia e horários
c) mensagem de horário inválido

O último recurso que o aplicativo fornece ao usuário são as configurações de tela, que incluem dois itens de acessibilidade do aplicativo. Elas podem ser acessadas a partir do menu na página de disciplinas (FIGURA 24) - como todos recursos do aplicativo - e oferecem as opções de modificar o tema (cor) e tamanho de todos os textos do aplicativo (FIGURA 24). Ao modificar a fonte ou o tema, praticamente todos elementos sofrem alterações, como os textos, avisos e botões incluídos nas páginas - inicial, grade horária e disciplina.

FIGURA 24 - CONFIGURAÇÕES DE TELA



FONTE: O autor (2021).

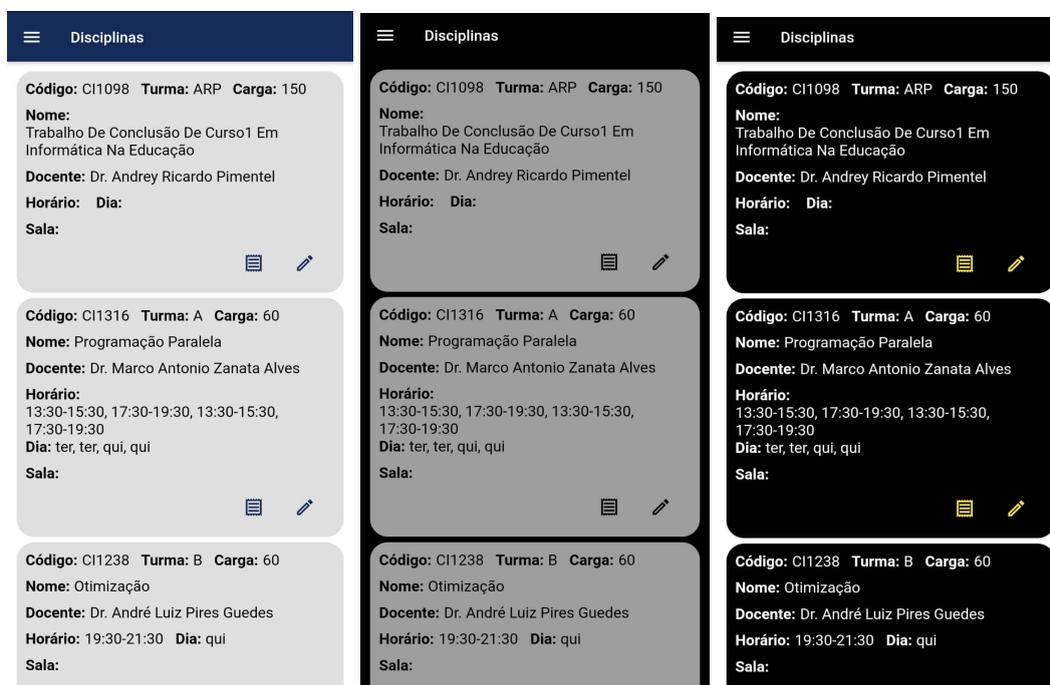
LEGENDA: a) opção de configurações de tela no menu
b) configurações de tela disponíveis

Segundo as diretrizes de acessibilidade de conteúdo da web (W3C, 2008), a cor é um item de extrema importância para acessibilidade em sistemas, sendo que o contraste entre elas é um dos pontos que estão relacionados a isso. Esse documento também fornece padrões relacionados às taxas de contraste para que um sistema esteja dentro de limites aceitáveis de acessibilidade. O nível mais exigente denominado como WCAG AAA requer uma taxa de contraste de pelo menos 7:1 para texto de tamanho normal e 4.5:1 para texto grande - tamanho da

fonte maior que 24 ou 18 em negrito. Em relação a isso, os três temas fornecidos no aplicativo: padrão, escuro e contraste (FIGURA 25) estão dentro desses padrões, mas, obviamente, eles possuem taxas de contraste diferentes.

O tema denominado como escuro é o que tem a menor taxa de contraste, 7.83:1 (entre preto e o tom de cinza utilizado), significando que ele é, possivelmente, o mais propenso a trazer complicações na visualização dos dados por pessoas que tenham algum problema de visão. Por segundo vem o tema padrão, com uma taxa no pior dos casos - entre o preto e o tom de cinza utilizado - de 11.17:1, valor agora bem acima do limite para textos de tamanho normal. Por último temos o tema contraste, que a pior taxa, que é entre o preto e o amarelo, tem um valor de 17.19:1. A título de curiosidade, a melhor taxa é entre branco e o preto absoluto que possuem uma taxa de 21:1, ressaltando que a maioria do texto no tema contraste possui essas duas cores contrastando. A partir disso, então, concluímos que o tema contraste, dentre as opções, seria mais adequado para indivíduos com algum tipo de deficiência visual.

FIGURA 25 - TEMAS DO ORGANIZA UFPR



FONTE: O autor (2021).

LEGENDA: a) Tema padrão
b) Tema escuro
c) tema contraste

O benefício de aumentar o tamanho da fonte para indivíduos com dificuldade de visão é indiscutível. O aplicativo por padrão possui um tamanho de fonte de 17, com exceção da parte de editar as informações das disciplinas que sempre está 4 números menor. A opção de modificar a fonte permite um tamanho de fonte de 17 até 34 sem comprometer a visualização dos dados em nenhuma parte do aplicativo, na grade horária foi fixado um valor de no máximo 20 para não ter esse tipo de problema. A Figura 26 apresenta três tamanhos diferentes de texto, a primeira imagem é o tamanho padrão - 17 -, a segunda um tamanho de 26 e a terceira o valor máximo de 34.

FIGURA 26 - FONTES DO ORGANIZA UFPR



FONTE: O autor (2021).

LEGENDA: a) Fonte tamanho 17
b) Fonte tamanho 26
c) Fonte tamanho 34

A Tabela 10 foi construída com o objetivo de servir de comparação com a Tabela 9, que mostra os recursos oferecidos pelos trabalhos - aplicativos - correlatos.

TABELA 10 - FUNCIONALIDADES DO APLICATIVO ORGANIZA UFPR (continua)

Nome do aplicativo	Possui anúncios	Alerta / Notificações	Acessibilidade	Widget	Grade horária
Organiza UFPR	não	não	tamanho da fonte e cor de contraste	não	sim

TABELA 10 - FUNCIONALIDADES DO APLICATIVO ORGANIZA UFPR (conclusão)

Nome do aplicativo	Eventos	Anotações	Notas	Importar pdf	Informações
Organiza UFPR	não	sim	não	sim	nome (disciplina), dia, horário, professor, código, turma, carga, GRR, Curso, carga(curso), nome(estudante), número eMec, ano ingresso

FONTE: O autor (2021).

Agora que todos os recursos disponíveis no aplicativo foram apresentados, a Tabela 11 apresenta todos os *plugins* e pacotes utilizados e suas respectivas funções.

TABELA 11 - PACOTES E PLUGINS UTILIZADOS

Nome do pacote/ <i>plugin</i>	Versão	Função
pdf_text	0.0.5	transformar PDF em String
shared_preferences	2.0.6	salvar os dados no dispositivo
flutter_timetable_view	0.2.0	<i>widget</i> da grade horária
multi_select_flutter	3.1.8	<i>widget</i> de seleção dos dias da semana
mask_text_input_formatter	2.0.0	máscara para padronizar os horários
provider	6.0.1	resgatar facilmente em todo código o valor da fonte e tema
numberpicker	2.1.1	<i>widget</i> de seleção do tamanho da fonte
file_picker	3.0.2	importar arquivo do armazenamento

FONTE: O autor (2021).

5 VALIDAÇÃO

5.1 Resultados

O primeiro experimento de validação a que o aplicativo foi submetido foi uma avaliação de usabilidade e acessibilidade realizada por integrantes do Laboratório de Design de Interação para a Inclusão Social e a Educação Universal (IHC) da UFPR. Após contato com o laboratório, seis alunos de mestrado e doutorado aceitaram participar voluntariamente da avaliação.

Foi, então, marcada uma reunião com cinco alunos - não conseguimos achar um horário onde todos os seis poderiam participar - para apresentação do aplicativo e explicação do seu objetivo, porém um dos alunos não apareceu na reunião. Todos os recursos que o aplicativo oferece foram apresentados, além de como interagir com eles em tempo real.

Com isso, foi acordado com os avaliadores que o aplicativo seria disponibilizado para ser instalado no celular de cada um e teriam um prazo de duas semanas para realizar a avaliação. Foi produzido também, a pedido, um documento para guiá-los na avaliação contendo um resumo do objetivo do aplicativo: “desenvolver um aplicativo capaz de auxiliar os alunos da UFPR na organização das informações referentes às disciplinas de cada curso. Englobando em um só lugar os nomes, horários, salas, notas, horários de cada uma delas. Tendo um acesso rápido e prático na palma das mãos”, um resumo do objetivo da avaliação: “elencar pontos falhos que podem ser melhorados ou que não estão presentes no aplicativo (tanto estético, de usabilidade e acessibilidade) para facilitar/auxiliar o uso dele pelos usuários finais” e alguns pontos que poderiam ser avaliados no aplicativo:

- “Analisar quão fácil é navegar e acessar os recursos do aplicativo e se essa navegação tem sentido. (menu, grade horário, informação do aluno, anotações, opções...)”
- A parte textual do aplicativo faz sentido e é adequada a todo usuário? (títulos das informações, botões do menu, alerta de confirmações...)
- Analisar quão fácil é chegar as informações que o aplicativo fornece, o objetivo do trabalho é que seja bastante simples chegar as informações específicas de cada disciplina, por exemplo, se viu alguma dificuldade, onde está o problema?
- Os botões estão bem dispostos e fazem sentido?

- É acessível a todos? Se não, onde estão os problemas?
- É necessário algum tipo de ajuda e informação além para o uso do aplicativo? Se sim, qual a melhor maneira de passar para o usuário?
- Houve problemas na execução do aplicativo no seu celular? (card para fora da tela, texto fora dos limites, algum bug ou problema ao iniciar, fechar, reiniciar, dados perdidos...)
- Você teve alguma dificuldade em utilizar o aplicativo? Por exemplo, para aparecer os horários na grade horária, editar algumas informações, apagar a disciplina, criar disciplina.
- Pontos que você pessoalmente acha que ficaria melhor (cor ou disposição dos cards, botões, menu)".

Algumas ações que seriam interessantes de realizar também foram listadas no documento:

- "importar pdf (pode ser mais de uma vez)
- adicionar uma disciplina vazia (manualmente)
- editar informações das disciplinas (código, nome, turma, carga, nome, docente, horário, dia, sala)
- excluir uma disciplina
- fazer uma anotação exclusiva para cada disciplina
- acessar as informações do aluno
- acessar a grade horária (ver se os horários estão sendo apresentados)
- limpar todas as disciplinas
- nas opções, mudar tema e tamanho da fonte
- fechar o app e abrir novamente, verificando se as informações foram salvas".

O trabalho recebeu como resposta duas avaliações, uma dentro do período de duas semanas e outra três dias depois do prazo estabelecido. Então, dos seis alunos que foram voluntários, três deles não enviaram a avaliação e uma quarta pessoa não pode realizá-la por não possuir um dispositivo com o sistema Android. A respeito disso, foi tentado gerar o aplicativo para dispositivos IOS, porém sem sucesso, só seria possível realizar tal ação por meio de um *desktop* com o sistema macOS do qual não foi possível ter acesso. Tentativas de rodar o aplicativo por meio de emulação na internet também não resolveram o problema por motivos de desempenho, que tornava difícil a interação e principalmente a importação de pdf dentro do aplicativo.

Foi ressaltado em ambas as avaliações, realizadas pelos alunos Leander de Oliveira e João Ricardo, o “grande potencial” de uso do aplicativo por parte dos estudantes, favorecendo uma “melhor e mais prática organização” das informações referentes às disciplinas, segundo João, “certamente será utilizado pelos alunos da UFPR”.

Outra questão abordada nas avaliações foi a preocupação com a acessibilidade, citada como “interessante” e que pode facilitar “a visualização das informações do *app* por pessoas que têm algum problema de visão”. Uma sugestão acerca disso foi o incremento de features futuras, como leitura de tela e lupa que podem auxiliar os usuários ainda mais.

A confidencialidade e privacidade dos dados do aplicativo foram itens citados em uma das avaliações, para João, “todo e qualquer aplicativo que tenha informações com dados precisa ter uma tela de autenticidade (*login*)”, porém, como optou-se por uma construção simples, tanto no modo e número de interações para realizar alguma ação dentro do aplicativo, foi decidido não considerar essa sugestão. Segundo Leander, essa simplicidade “é um ponto positivo para um *app* desta natureza que busca facilitar uma atividade cotidiana”, considerando, então, que uma tela de autenticidade iria adicionar uma complexidade a mais na interação com o aplicativo, além de nenhuma informação ser confidencial dentro do aplicativo.

Algumas melhorias foram realizadas após o *feedback* dos avaliadores. A primeira delas foi a mudança no tom do cinza do tema escuro, pois estava muito escuro, o que causou um desconforto visual em um dos avaliadores. Outro ponto foi a página inicial, antes, era necessário clicar sobre o símbolo da UFPR para se ter acesso a página de disciplinas, porém, para um dos avaliadores esse botão não era tão claro e não tinha motivo para ficar “preso” na página inicial, assim foi estipulado um tempo para a passagem de página ocorrer automaticamente. As outras mudanças foram apenas de texto no menu, uma delas foi a mudança de “opções” para “configurações de tela”, pois, segundo o avaliador, é o que está contido ali, além da denominação “opções” se remeter “a uma amplitude maior de possibilidades”. A última mudança foi a troca de “informações do aluno” para “dados acadêmicos”, isso para evitar utilizar uma palavra com identificação de gênero, evitando qualquer desconforto ou constrangimento ao usuário.

Algumas considerações feitas nas avaliações foram consideradas como possíveis implementações futuras, como: possibilidade de importar informações do

pdf da pós graduação da UFPR ao invés de apenas da graduação, salvamento na nuvem para abrir as informações em mais de um dispositivo e outras chegaram a ser cogitadas no desenvolvimento porém não implementadas, como: verificar se o aplicativo é funcional para dispositivos IOS - não foi possível ter acesso a um dispositivo com macOS e a um dispositivo com IOS -, criação de uma logo para o aplicativo, explorar mais a funcionalidade das anotações - a sugestão foi de abrir “uma nova tela com as anotações em separado que poderiam ser marcadas com níveis de importância, cores, como post-its digitais” -, salvar um conjunto de horários, por exemplo, para quando mudar o semestre (quando há mudança nas disciplinas) não precisar apagar os dados atuais ou ficar com tudo exposto na tela (atuais e antigos), seria algo como um histórico, e por último ter um espaço além das anotações para conter as notas de provas e trabalhos das disciplinas.

O segundo experimento de validação foi realizado com alunos e ex-alunos (formados recentemente) da UFPR com o intuito de ter uma noção melhor da interação do usuário final com a aplicação. Assim, como no outro experimento, foi liberado o aplicativo para instalarem nos seus dispositivos móveis, além de um comprovante de matrícula para interagirem com o *software*. Um documento foi redigido para listar algumas informações pertinentes: como instalar a aplicação no dispositivo, aviso sobre não funcionar em dispositivos IOS, indicação do comprovante para teste, alerta para que não esquecer de salvar os dados modificados dentro do *app* - confirmação no teclado -, alerta sobre o padrão para aparecer os horários na grade horária - mesma quantidade de dias e horários -, o mesmo guia de ações possíveis de se realizar no aplicativo enviado no experimento anterior, aviso de que os “dados acadêmicos” são preenchidos ao importar o pdf e o *link* para o formulário de avaliação. Foram disponibilizados cinco dias para essa etapa da avaliação.

O formulário de avaliação foi baseado no Modelo de Aceitação de Tecnologia (TAM¹²), importante recurso já consolidado para avaliação da aceitação e usabilidade de um *software* (DAVIS, 1989). Esse modelo considera a utilidade percebida (PU¹⁰) - quanto um indivíduo acredita que o *software* vai melhorar o seu desempenho no trabalho, no nosso caso, pode ser entendido como quanto o *software* vai facilitar o acesso às informações acadêmicas - e facilidade de uso percebida (PEU⁹) - medida de quanto um indivíduo acredita que será fácil utilizar tal

tecnologia - como os principais fatores que influenciam na intenção de usar um *software*.

O TAM padrão conta com 12 perguntas, sendo seis perguntas relacionadas a PU e seis relacionadas à PEU. O modelo visa fornecer uma noção probabilística do futuro uso do *software* e não a experiência de uso real dele, questão que foi crucial para utilização dele no presente trabalho, pois o grande objetivo era prever o potencial que uma aplicação tem, com as características e utilidades presentes no aplicativo produzido. E com o *feedback* recebido do formulário, o potencial do aplicativo foi mais que confirmado.

Segundo Lewis (2019), o sistema de medição utilizado em um questionário baseado na experiência do usuário, como o TAM, pode ajudar o usuário a não cometer erros. Ele propõe o uso de um sistema no formato SUS / UMUX-Lite que foi utilizado no questionário de avaliação do Organiza UFPR, onde as opções são numéricas, com os níveis de concordância mais baixos à esquerda, isso vai em contrapartida ao proposto no TAM original, onde a escala é verbal (Extremamente, bastante, um pouco, nenhum, um pouco, bastante, extremamente) e com os níveis de concordância mais baixos a direita. Também foi adaptado desse mesmo artigo as perguntas para a avaliação.

O sistema utilizado no presente trabalho conta com sete opções de concordância. A Tabela 12 apresenta as perguntas do questionário de avaliação realizadas, a quantidade de votos de cada uma das sete opções nas colunas seguintes e na última coluna a média das respostas. Foram obtidas no total 16 avaliações, como todas as perguntas eram obrigatórias, tivemos então, o mesmo número de respostas para cada uma das perguntas.

TABELA 12 - QUESTIONÁRIO DE AVALIAÇÃO E NOTAS

(continua)

Questão	1 (Discordo totalmente)	2	3	4	5	6	7 (Concordo totalmente)	Média
Usar este aplicativo na minha rotina me permite realizar consultas mais rapidamente do que pelos navegadores (Chrome, Safari...)	-	-	-	-	3	3	10	6,4
Usar este aplicativo na minha rotina me permite realizar consultas mais rapidamente do que por imagens (fotos salvas, prints...) e arquivos salvos no celular	-	-	-	-	6	2	8	6,1
Usar este aplicativo na minha rotina me permite realizar consultas mais rapidamente do que os recursos disponíveis (SIGA, site do ensalamento...)	-	-	-	-	-	-	16	7
Por ser um aplicativo para o celular, consigo consultar informações mais rapidamente	-	-	-	-	2	6	8	6,4
Usar este aplicativo me ajuda a me organizar melhor	-	-	-	1	5	3	7	6,0
Eu achei este aplicativo útil	-	-	-	-	2	4	10	6,5
Aprender a interagir com este aplicativo foi fácil para mim	-	-	-	1	1	3	11	6,5
Considerando que o aplicativo fosse distribuído nas principais lojas de aplicativos (Google play e App Store), seria fácil obtê-lo	-	-	-	-	-	3	13	6,8

TABELA 12 - QUESTIONÁRIO DE AVALIAÇÃO E NOTAS

(conclusão)

Questão	1 (Discordo totalmente)	2	3	4	5	6	7 (Concordo totalmente)	Média
Considerando que o aplicativo fosse distribuído fora das lojas de app (Google play e App Store) por meio de um arquivo executável do celular (como foi disponibilizado para realizar esta avaliação), seria fácil obtê-lo	-	1	2	3	3	0	7	5,3
Minha interação com este aplicativo foi clara e compreensível	-	-	-	-	3	7	6	6,2
Achei este aplicativo fácil de modificar de acordo com as minhas informações	-	-	-	2	1	7	6	6,1
Acredito que será fácil para mim tornar-me hábil no uso deste aplicativo	-	-	-	2	-	5	9	6,3
Achei este aplicativo fácil de usar	-	-	-	-	1	5	10	6,6

FONTE: O autor (2021)

No geral, o aplicativo teve notas muito boas, refletindo o grande potencial dele. O interessante é que 100% dos participantes concordaram totalmente que o sistema permite realizar consultas mais rapidamente do que os disponíveis para os alunos hoje, como o SIGA UFPR e o site do ensalamento - principal meio de divulgação das salas de aula -, demonstrando o descontentamento com esses recursos para o acesso de informações relacionados às disciplinas e acadêmicas no geral. No questionário também foi disponibilizado a possibilidade dos participantes deixarem suas sugestões, críticas e elogios, e foi em uma das respostas que um dos participantes citou que as plataformas da UFPR não são bem adaptadas à dispositivos móveis, possivelmente sendo esse um dos principais motivos para a insatisfação de todos.

Os participantes também concordaram que o Organiza UFPR facilita o acesso aos dados quando relacionado aos navegadores - no caso por páginas web - e arquivos salvos no celular - fotos, imagens, texto armazenado de alguma forma. Pelos navegadores a concordância foi maior, possivelmente pelo fato de serem mais restritas as opções, tanto do número de navegadores, quanto das opções que eles oferecem. Já em relação aos arquivos salvos no celular, existem infinitas possibilidades de aplicativos, opções e recursos nativos dos dispositivos que os usuários têm acesso para organizar-se da forma que cada um achar melhor e ter acesso rápido às informações. Porém, mesmo com tantas possibilidades, um aplicativo tal como o desenvolvido possui um potencial de uso relativamente bom.

Uma outra alternativa ao aplicativo para dispositivos móveis seria, por exemplo, um aplicativo para desktop, por esse motivo a afirmação “por ser um aplicativo para o celular, consigo consultar informações mais rapidamente” foi feita. Visto que a média foi alta, representando uma concordância alta com a afirmação, acredito que o melhor tipo de dispositivo para distribuir o aplicativo seria pelos dispositivos móveis.

Já que entramos na questão da distribuição, temos duas opções que foram consideradas no trabalho, distribuir pelas lojas de aplicativos convencionais - Google play para Android e App Store para IOS - ou por fora delas. Como esperado, a distribuição pelas lojas foi preferida em relação a pôr fora delas, acredito eu, principalmente pela segurança que esses sistemas fornecem. Um dos participantes até citou a dificuldade que teve em instalar o aplicativo - que não foi fornecido a partir de alguma loja e sim por um arquivo para baixar no celular a partir de uma pasta no Google Drive - por conta das verificações de segurança do próprio sistema operacional. A facilidade de procurar, ver avaliações e informações sobre os aplicativos nas lojas também são muito mais confiáveis e fáceis por todos estarem familiarizados com elas - ou espera-se que estejam. A distribuição do Organiza UFPR na Google Play foi uma questão levada em consideração na parte de desenvolvimento, porém, por questões burocráticas e financeiras optou-se por não colocar o aplicativo na loja, mesmo acreditando que seria mais fácil a obtenção dele. O nível de discordância não foi alto para a maneira que foi distribuído, logo, acredita-se que ela não teve influência no resultado da avaliação e se, porventura, o aplicativo fosse distribuído fora das lojas, não seria um empecilho tão grande para os usuários finais.

Outra questão que o TAM expos é sobre a utilidade do aplicativo, a maioria dos participantes concordaram que o aplicativo é útil, tanto que alguns comentários como: “é sem dúvidas uma opção muito boa para facilitar a vida dos alunos” e “fazia muita falta ter algo assim ao longo do curso” surgiram, afirmando ainda mais essa questão. É certo que o aplicativo poderia oferecer vários outros recursos relevantes - que vão ser tratados mais à frente -, tanto para ajudar os usuários a se organizarem melhor tanto para facilitar a sua utilização em si, porém nenhuma das questões relacionadas a essas questões tiveram alguma nota baixa na avaliação. Considerando essa questão, um dos fatores primordiais no desenvolvimento, que foi tentar tornar a interação do usuário a mais simples possível, com um mínimo de esforço para realizar qualquer ação foi alcançada com êxito.

Mesmo com esse bom desempenho no TAM, não quer dizer que o aplicativo pode melhorar ainda mais. Acredita-se que, por exemplo, para a interação do usuário ser mais clara e compreensível, dicas ou alguma ajuda de como funciona a aplicação poderiam ser fornecidas. Isso também ajudaria na facilidade de interação com o aplicativo, pois certos participantes perguntaram algumas questões de como mexer no aplicativo que poderiam ser facilmente resolvidas com essas ações.

5.2 TRABALHOS FUTUROS

Acredita-se que vários *upgrades* são pertinentes serem implementados, algumas sugestões surgiram nas duas avaliações e outras foram levantadas ao longo do desenvolvimento, sendo que algumas ideias surgiram com a análise dos trabalhos relacionados. Serão listadas as modificações a seguir com o intuito de servirem como base a trabalhos futuros:

- *Zoom out* na grade horária para facilitar a visibilidade dela inteira;
- Dicas e botões de ajuda para facilitar a interação do usuário;
- Possibilidade de adicionar outras atividades além de somente disciplinas, como ic e monitoria;
- Ter um espaço próprio para controle de notas, datas de entrega de trabalhos e controle de faltas;
- Usar cores para diferenciar visualmente as disciplinas;
- Integração com o Google Agenda e com o Maps (para localizar as salas de aula);

- Importação do comprovante de matrícula da pós-graduação;
- Mais itens de acessibilidade, lupa e leitor de tela, por exemplo;
- Importar as salas de aula das disciplinas automaticamente;
- Possibilidade de guardar um conjunto de disciplinas (semestre anterior) e criar outro;
- Possibilidade de editar os dados acadêmicos;
- Disparar notificações e alarmes para avisar sobre eventos (provas, trabalhos e reuniões);
- Backup dos dados na nuvem;
- Poder escolher a ordem das disciplinas na página de disciplinas;
- Funcionamento *online* e *offline*;
- Lógica para o caso de importar o comprovante de matrícula mais de uma vez;
- Integração com itens de acessibilidade do sistema nativo;
- Visualização do cardápio do restaurante universitário;
- Editar as informações das disciplinas sem confirmação do teclado do dispositivo.

O código fonte da aplicação pode ser acessado em:
<https://github.com/nicollasaque/organiza-UFPR>

6 CONCLUSÃO

A partir deste trabalho foi possível iniciar um estudo do potencial que uma aplicação com a finalidade de centralizar as informações acadêmicas possui. Ele foi comparado com outros sete aplicativos, com base em suas características e relevância na loja do Google Play, com a intenção de situar-se no mercado amplo das aplicações *mobile*.

No desenvolvimento foram usadas as tecnologias Flutter, Dart e Android Studio, que foram escolhidas por serem de autoria do Google, pois acreditava-se que por esse motivo teriam vantagens na construção de uma aplicação para o Android, que também é do Google, e no final isso foi confirmado. A facilidade em gerar o arquivo apk e em utilizar pacotes que usam funções nativas, o apoio a depuração, teste e desenvolvimento, além da utilização de recursos de *design* da biblioteca material design foram um dos pontos que justificaram a escolha das tecnologias.

A partir dos dois experimentos de validação, o primeiro com estudantes da pós-graduação para avaliar a acessibilidade e usabilidade e o segundo com alunos e ex-alunos da UFPR para mensurar o nível de aceitação, o aplicativo Organiza UFPR se mostrou ser muito promissor, principalmente pelo fato da UFPR não fornecer um sistema como esse, com os mesmos objetivos e recursos. Os participantes mostraram grande interesse, levando em conta as notas que a aplicação obteve no questionário baseado no TAM - boa aceitação - e nas sugestões, críticas e elogios recebidos, isso mesmo somente com os recursos disponibilizados, pois, principalmente por motivo de tempo, não foi possível oferecer um aplicativo mais completo, com mais recursos do que a versão disponibilizada.

Por fim, as contribuições gerais do trabalho foram a aplicação em si produzida, que fornece uma boa base para trabalhos futuros com as mesmas características, objetivos e recursos, também o uso prático do Flutter e da linguagem Dart que pode servir como modelo para outros trabalhos. O questionário TAM adaptado também foi uma contribuição, pois ele pode ser usado para medir a aceitação de outros *apps* que o questionário TAM original não se encaixa e seus resultados servem de base para justificar o desenvolvimento de uma aplicação com o objetivo de juntar as informações acadêmicas em um só lugar.

REFERÊNCIAS

ALEXANDER S. GILLIS. Techtarget, 2020. Disponível em: <https://searchsoftwarequality.techtarget.com/definition/native-application-native-app>.

Acesso em: 4 dez. 2021.

ALPHACODE. Disponível em: <https://site.alphacode.com.br/desenvolver-aplicativo/>.

Acesso em: 4 dez. 2021.

DART. Disponível em: <https://dart.dev/overview>. Acesso em: 5 dez. 2021a.

DART. Disponível em: <https://dart.dev/guides>. Acesso em: 5 dez. 2021b.

DART. Disponível em: <https://dart.dev/guides/language/language-tour>. Acesso em: 5 dez. 2021c.

DAVIS, Fred D. Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. **MIS Quarterly**, v. 13, n. 3, p. 319-339, 1989.

FLUTTER. Disponível em: <https://docs.flutter.dev/resources/faq>. Acesso em: 4 dez. 2021a.

FLUTTER. Disponível em: <https://flutter.dev/docs>. Acesso em: 5 dez. 2021b.

FLUTTER. Disponível em: https://docs.google.com/presentation/d/1cw7A4HbvM_Abv320rVgPVGiUP2msVs7fGbkgrTy0l/edit#slide=id.gbb3c3233b_0_162. Acesso em: 5 dez. 2021c.

GOOGLE. Google developers, 2021. Disponível em: <https://developer.android.com/studio/preview>. Acesso em: 5 dez. 2021.

GOOGLE PLAY. Disponível em: https://play.google.com/store/apps/details?id=com.lpalominos.horariodeclases&hl=pt_BR&gl=US. Acesso em: 5 dez. 2021a.

GOOGLE PLAY. Disponível em:
https://play.google.com/store/apps/details?id=com.gabrielittner.timetable&hl=pt_BR&gl=US. Acesso em: 5 dez. 2021b.

GOOGLE PLAY. Disponível em:
https://play.google.com/store/apps/details?id=com.icemediacreative.timetable&hl=pt_BR&gl=US. Acesso em: 5 dez. 2021c.

GOOGLE PLAY. Disponível em:
https://play.google.com/store/apps/details?id=daldev.android.gradehelper&hl=pt_BR&gl=US. Acesso em: 5 dez. 2021d.

GOOGLE PLAY. Disponível em:
https://play.google.com/store/apps/details?id=com.claudivan.agendadoestudanteplus&hl=pt_BR&gl=US. Acesso em: 5 dez. 2021e.

GOOGLE PLAY. Disponível em:
https://play.google.com/store/apps/details?id=br.ufrn.sinfo.sigaambeta&hl=pt_BR&gl=US. Acesso em: 5 dez. 2021f.

GOOGLE PLAY. Disponível em:
https://play.google.com/store/apps/details?id=br.ufmg.ddp.cecom.academico.sigaufmg&hl=pt_BR&gl=US. Acesso em: 5 dez. 2021g.

GOOGLE PLAY. Disponível em:
https://play.google.com/store/apps/details?id=br.ufu.cti.apps.mobile&hl=pt_BR&gl=US. Acesso em: 5 dez. 2021h.

LEWIS, James R. Comparison of Four TAM Item Formats: Effect of Response Option Labels and Order. **Journal of Usability Studies**, v. 14, n. 4, 2019.

MARIANA TOKARNIA. AgênciaBrasil, 2020. Disponível em: <https://agenciabrasil.ebc.com.br/economia/noticia/2020-04/celular-e-o-principal-meio-de-acesso-internet-no-pais>. Acesso em: 4 dez. 2021.

SIDDHARTH SHARMA. Flutterdevs, 2020. Disponível em: <https://medium.flutterdevs.com/the-growth-of-flutter-development-3years-after-the-birth-of-alpha-78baee809dff>. Acesso em 5 dez. 2021.

STATISTA. Disponível em: <https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/>. Acesso em: 4 dez. 2021.

W3C. Disponível em: <https://www.w3.org/TR/2008/REC-WCAG20-20081211/>. Acesso em: 5 dez. 2021.