Revista do *Fórum Gerencial*, v. 1, n. 2, 2021
© Centro Universitário de Patos de Minas
https://revistas.unipam.edu.br/index.php/forumgerencial

FRILA: APLICATIVO PARA PRESTADORES DE SERVIÇO[1]

Adryel Klever Vieira de Almeida[2]
Mislene Dalila da Silva[3]

RESUMO: Esse estudo teve como objetivo o desenvolvimento de uma ferramenta de divulgação de serviços para auxiliar os trabalhadores autônomos, como eletricistas, pedreiros etc. Na aplicação da ferramenta, ela se demonstrou eficiente por simplificar os processos de divulgação dos serviços e de comunicação entre trabalhadores e clientes. A comunicação entre os envolvidos foi facilitada pela utilização de uma plataforma mobile com serviços do Parse Server.
PALAVRAS-CHAVE: *Dart(Flutter), Parse Server,* Aplicativo *Mobile.*

ABSTRACT: This study aimed to develop a service dissemination tool to assist self-employed workers, such as electricians, bricklayers, etc. In the application of the tool, it proved to be efficient by simplifying the processes of disclosing services and communicating between workers and customers. Communication between those involved was facilitated by the use of a mobile platform with Parse Server services.
KEYWORDS: *Dart(Flutter), Parse Server, Mobile App.*

1 INTRODUÇÃO

Com o crescimento do mercado de *smartphones* e consequentemente dos aplicativos, as pessoas passaram, cada vez mais, a usar o celular no dia a dia, seja para diversão ou seja para trabalho. O número de trabalhadores que prestam serviços de maneira autônoma aumentou, com isso, o mercado se tornou mais competitivo. Diante dessa competitividade, surge, então, a necessidade de se ter um diferencial, algo que possa tornar o trabalho mais atraente, prático e eficiente, para atingir as expectativas dos clientes.

Assim, o desenvolvimento dessa ferramenta tem o intuito de agregar valores aos serviços prestados pelos trabalhadores autônomos, bem como de facilitar o acesso dos clientes aos serviços por eles prestados. Em uma parte considerável dos casos analisados, os tipos de serviços citados anteriormente, são divulgados no *Marketplace*, grupos e páginas do *Facebook*, ou no site de compras e vendas da OLX. Tais plataformas têm o foco na venda e na compra de produtos, não em prestação de serviços.

O objetivo deste estudo foi criar uma ferramenta para os trabalhadores autônomos com o propósito de auxiliar o trabalho e a divulgação dos serviços prestados. Para isso foi considerado os dados básicos de divulgação dos serviços como a descrição das atividades desempenhadas, os valores dos serviços a serem prestados e a experiência dos trabalhadores. Com as informações tratadas pela aplicação, será

[1] Artigo apresentado por Adryel Klever Vieira de Almeida.

[2] Estudante de Graduação do Curso de Sistemas de Informação do UNIPAM. E-mail: adryelklever@unipam.edu.br.

[3] Professor orientador. E-mail: mislene@unipmam.edu.br.

possível listar todos os serviços cadastrados, filtrando por categoria e região. Com o serviço escolhido, o cliente poderá ver todas as informações como, valores, descrição dos serviços, nome do trabalhador e avaliação que outros clientes deram para o serviço. Para se atingir o objetivo geral, apresentam-se os seguintes objetivos deste estudo:

- Auxiliar a cadastrar, modificar e gerenciar as informações dos serviços;
- Aumentar a quantidade de informação disponível de cada trabalhador e seus serviços;
- Aumentar a comunicação entre o trabalhador autônomo e seus clientes;
- Melhorar a forma de pagamento para seus clientes.

Para desenvolver o aplicativo foi utilizado a documentação oficial do *Flutter* e *Dart* na versão *mobile*. Para o banco de dados, envio de *e-mail* e notificações, monitoramento de usuários, *dashboard* para controle de tarefas e *upload* de imagens, foi utilizado a documentação oficial do *Back4App*. Desse modo, segue a listagem e ferramentas utilizadas para o desenvolvimento do aplicativo e suas respectivas descrições:

- Dart (Flutter): Como linguagem de desenvolvimento;
- Android Studio: desenvolvimento do aplicativo;
- Visual Studio Code: gerar apk;
- Parser Server: Banco de dados e notificações;
- Back4App: Banco de dados;

Portanto o aplicativo foi desenvolvido contemplando todos os objetivos propostos para solucionar as necessidades dos trabalhadores autônomos, começando pela gerência dos serviços, que será realizada pelos seus *smartphones* em qualquer lugar, aumentar a rede de divulgação de seus serviços para que mais pessoas sejam alcançadas e assim, aumentando seus clientes, melhorar e facilitar a comunicação entre o trabalhador e cliente, e propor novas condições de pagamentos.

2 REFERENCIAL TEÓRICO

2.1 APLICATIVOS

Aplicativos são programas de software presentes em celulares *Android*, *iPhone (iOS)* e outros diversos dispositivos inteligentes, como *smart TVs*. Os *apps* podem ser gratuitos ou pagos. Eles desempenham diversas funções: mensageiro *online*, *streaming*, gerenciadores, editores de fotos e vídeos. Alguns já vêm instalados de fábrica, enquanto outros podem ser obtidos na Apple Store ou na *Play Store*. Os aplicativos facilitam a vida do usuário e se tornaram indispensáveis no dia a dia.

Nem todos os aplicativos buscam facilitar tarefas ou proporcionar entretenimento. O app nocivo, chamado pelo *Google* de "aplicativo potencialmente nocivo" coleta dados do usuário, como contatos e localização em tempo real, exibe anúncios intrusivos e ainda deixa o dispositivo vulnerável para outros programas maliciosos. Esse tipo de app ataca principalmente os dispositivos *Android*. Eles podem se infiltrar na loja de aplicativos do *Google* e ser instalados por vários usuários antes que a fiscalização da loja detecte e ele seja removido da plataforma. Aplicativos

intrusivos costumam levantar suspeitas com permissões excessivas e desnecessárias. Já o *iOS* é um sistema operacional mais seguro. Segundo a *Apple*, ele foi projetado "com segurança em sua essência".

2.2 MERCADO DE APLICATIVOS

A indústria de aplicativos movimenta, de maneira virtual, o mundo todo. O Brasil aparece entre os países emergentes, trazendo grande potencial para desenvolvedores e investidores. O Brasil é o segundo mercado de aplicativo que mais cresce no mundo estando atrás apenas da Indonésia e à frente da Coreia do Sul. O uso de apps no Brasil é alto; os brasileiros passam cerca de três horas por dia utilizando aplicativos. Na média, de 70 a 80 softwares são instalados no smartphones, sendo que 30 a 40 são efetivamente utilizados. Enquanto o mercado global de smartphones apresentou uma leve queda durante 2019, o setor brasileiro foi pelo caminho oposto, registrando a venda de *smartphone* maior em relação ao ano de 2018. Os *smartphones* intermediários ajudaram nessa entrada.

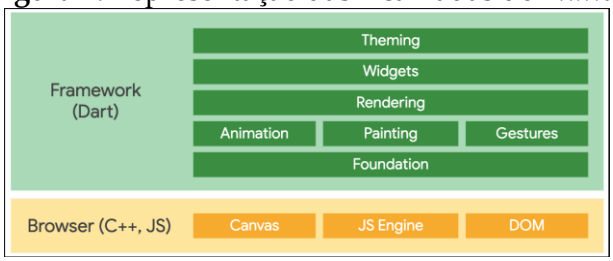
Aplicativos de transporte, apps de relacionamento, *delivery* e robôs são os aplicativos mais utilizados no dia-a-dia. Enquanto há alguns anos era necessário ligar o computador ou até mesmo fazer ligações para realizar uma compra, hoje basta abrir o aplicativo no celular e ter todas as funções desejadas em poucos cliques. Essa é uma das principais razões pelas quais o mercado de aplicativos tem ganhado tanto espaço e está se expandindo rapidamente.

Com o isolamento causado pelo coronavírus, os serviços do apps tiveram um aumento significativo durante a quarentena. Aplicativos de delivery apresentaram um aumento de 15% em suas instalações entre os meses de fevereiro e março. Empresas apontam cada vez mais o aumento no volume de vendas online originadas de dispositivos móveis, principalmente em datas comemorativas. O mercado *mobile* está cada vez mais aquecido e não há previsões de quedas expressivas no setor. Investir no desenvolvimento de apps, na construção de plataformas digitais sólidas e robustas, é apostar em um mundo com possibilidades quase infinitas onde pode-se conectar mais rapidamente empresas, fornecedores e consumidores finais.

2.3 FLUTTER

O *Flutter* é dividido em duas camadas em sua *Engine*. A primeira é responsável por todo o Framework *Flutter* escrito em *Dart*. A segunda é algo que dificilmente terá que se preocupar, pois se trata do core do *Flutter*, onde o cérebro dele está onde estará tratamentos específicos de cada sistema e a *engine* gráfica (*OpenGLES*), que é o diferencial do *Flutter* com *React Native*.

Figura 1: Representação das 2 camadas do *Flutter*



Fonte: flutter.dev/imagens/Dart-framework-v-browser-framework.png, 2020.

Como o *Flutter* tem sua própria *engine* de renderização, ela não necessita de um Bridge específica para cada sistema executar o código. No *React Native*, por exemplo, sempre que você usa um componente de *View*, basicamente debaixo dos panos. Ele está chamando uma *View* em cada sistema específico, que de alguma forma ou outra, trará um gargalo para as *views* mais complexas.

2.4 PARSE SERVER

Em 2011, a empresa Parse Inc. foi fundada e disponibiliza o serviço com o mesmo nome: Parse. O Parse permite que os desenvolvedores mobile armazenem dados na nuvem, utilizando métodos de autenticação (e-mail e redes sociais) e enviem notificações por *push*. Com o serviço, desenvolvedores contam com um *backend* escalável, sendo capaz de lançar aplicações mobile em tempo recorde, sem a necessidade de se preocupar com a infraestrutura e gerenciamento de servidores.

Em 28/01/2016 o *Facebook* anuncia o fim do serviço *Parse*, com desligamento no prazo de 1 ano. Milhares de desenvolvedores ficariam preocupados e teriam que migrar todos os seus aplicativos para outros serviços. Com isso, o *Facebook* decide abrir o código do Parse, assim nenhum desenvolvedor teria que migrar seus aplicativos para outros serviços. Assim nasce o *Open Source Parse Serve*. Com isso, várias empresas ofereceram *hosting* para o Parse Server e suas características são:

- Projeto: *Open Source*;
- Arquitetura: *Node.JS, Express, MongoDB / PostgreSQL, API Rest / GraphQL*;
- Pode ser utilizado com outros bancos de dados através de Adaptadores;
- Possibilidade de desenvolver e testar localmente;
- Várias opções de hospedagem;
- Melhorou várias deficiências do Parse (limites de conexões, live queries, etc);
- Mantido com recursos financeiros da própria comunidade.

Disponibiliza *SKD's* e bibliotecas para várias tecnologias:

- *iOS + MacOS + tvOS;*
- *Android;*
- *JavaScript*
- *.NET + Xamarin;*
- *Unity;*
- *PHP;*
- *Arduino;*
- *Embedded C.*

- *REST API;*
- *GraphQL;*
- *Go;*
- *Python;*
- *Ruby;*
- *Flutter.*

É possível integrar o Parse Server com:

- *Parse Server Hosting;*
- *Back4app.com;*
- *Sashido.io;*
- Servidor Próprio *(VPS);*
- *Hostgator;*
- *AWS;*
- *Azure;*
- *Google.*

3 DESENVOLVIMENTO E RESULTADOS

Entre os fatores importantes para a elaboração deste trabalho está a utilização do Parse Server. Para utilizar o *Parse Server* é necessário informar o nome do *package* no arquivo *pubspec.yaml* criado pelo próprio *Flutter.* O package deve ser informado em *dependencies* > *flutter* > *sdk*, como mostra a figura 1. Após informar o nome do package, é preciso clicar em Pub get para que o package seja instalado na aplicação e seja possível importá-lo em qualquer arquivo dentro do projeto. Todo *package* que será usado no Flutter, deve ser nesse caminho descrito anteriormente.

Figura 2: Importação do *package Parse Server SDK*

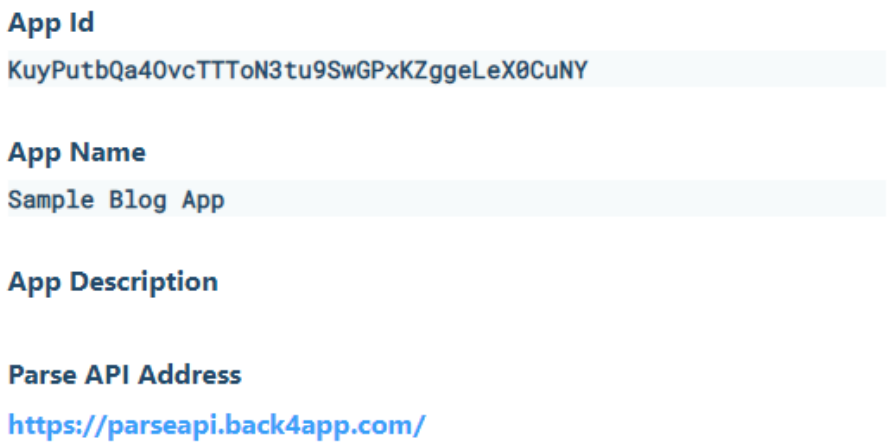
```
dependencies:
  flutter:
    sdk: flutter


  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.0
  parse_server_sdk: ^1.0.26
```

Fonte: Dados da Pesquisa, 2020.

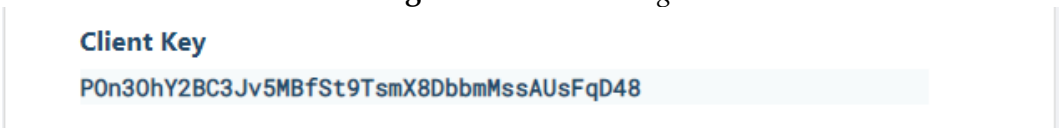
Após criar um projeto no Back4App, é possível encontrar a opção *Server Setting* onde há informações importantes para realização da conexão do *Flutter* com o Parse. Essas informações são: *App Id, Parser API Andress e Client Key*, como mostram as figuras, 2, 3 e 4. Essas informações devem ser mantidas em segredo; todas as informações mostradas nas imagens são de um projeto teste.

Figura 3: *Server Settings*



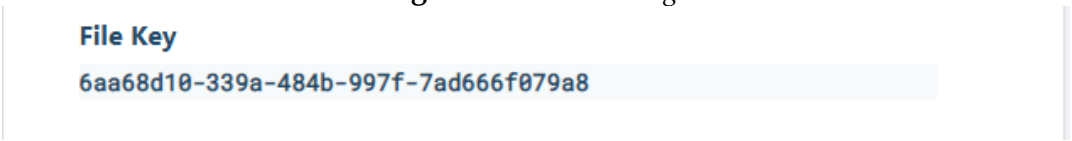
Fonte: Dados da Pesquisa, 2020.

Figura 4: *Server Settings*



Fonte: Dados da Pesquisa, 2020.

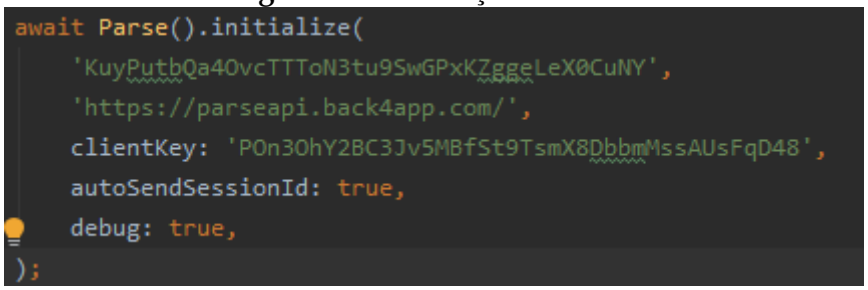
Figura 5. *Server Settings*



Fonte: Dados da Pesquisa, 2020.

Na figura 5, está sendo inicializado o *Parse*, informando o *App Id*, *Parse API Address* e o *Client Key*. O *autoSendSessionId* identifica quem está fazendo uma requisição no *Parse*, ou seja, não há necessidade de passar quem está fazendo uma requisição, todas as vezes que for fazê-la. Passando *true*, o *Parse* se encarrega de fazer todo esse trabalho. O *debug* é uma espécie de *print*, onde tudo que o *Parse* fizer, será mostrado no console do *Android Studuio*.

Figura 6: Inicialização do Parse



Fonte: Dados da Pesquisa, 2020.

Na figura 6, mostra um exemplo de como criar um objeto dentro do Back4App. No *response* tem *await* pois essa escrita não é síncrona e sim assíncrona, podendo gastar um tempo para fazer a inscrição no Back4App.

Figura 7: Criando objeto no Back4App

```
final category = ParseObject('Categoria')
  ..set('Title', 'Camisetas')
  ..set('Position', 2);

final response = await category.save();
```

Fonte: Dados da Pesquisa, 2020.

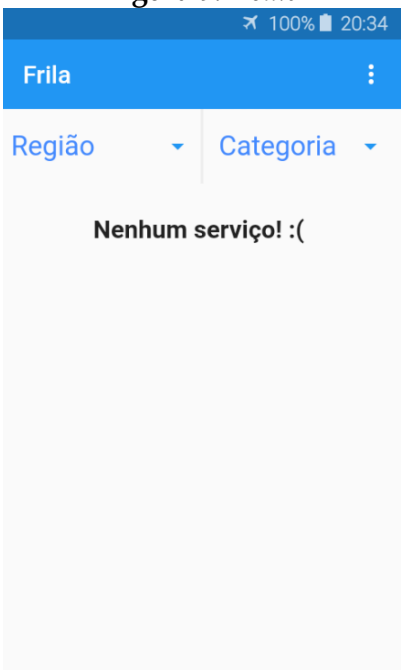
Na tela de login, o usuário tem a opção de *login* ou se cadastrar. A tela possui um slide onde o botão de ação muda o texto de acordo com a escolha do usuário. A opção "Ir para anúncios" dá liberdade para um cliente visualizar os serviços sem a necessidade de ter um cadastrado no aplicativo.

Figura 8: Tela de *Login* / Cadastrar

Fonte: Dados da Pesquisa, 2020.

Após acesso ao aplicativo, na tela *Home* o usuário pode visualizar todos os serviços cadastrados no *app.* Os serviços podem ser filtrados informando região e categoria.

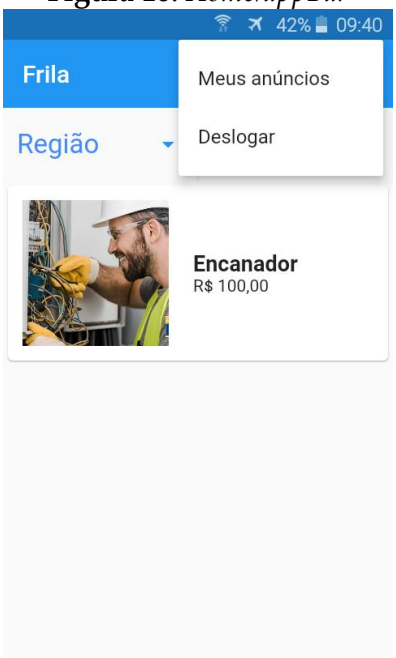
Figura 9: *Home*



Fonte: Dados da Pesquisa, 2020.

A *appBar* que está localizada na tela Home, possui a funcionalidade "Entrar / Cadastrar" ou caso o usuário esteja logado, deslogar ou consultar seus serviços cadastrados.

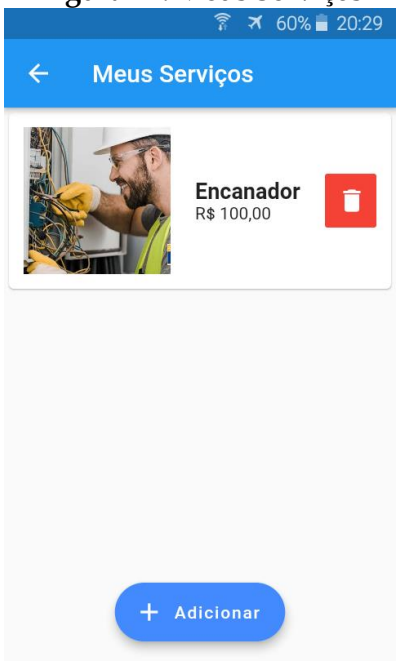
Figura 10: *Home/appBar*



Fonte: Dados da Pesquisa, 2020.

A tela “Meus Serviços” lista todos os serviços cadastrados pelo trabalhador autônomo, com a possibilidade de excluir quando quiser. Caso ele queira cadastrar um novo serviço, basta clicar no botão “Adicionar”.

Figura 11: Meus Serviços



Fonte: Dados da Pesquisa, 2020.

A tela “Novo Serviço”, deve ser informado até 5 imagens, a região, a categoria, título, preço, telefone e uma descrição com no máximo 200 caracteres.

Figura 12: Tela Novo Serviço.

Fonte: Dados da Pesquisa, 2020.

4 CONSIDERAÇÕES FINAIS

Esta pesquisa tem como intenção trazer praticidade e eficiência no trabalho dos prestadores de serviços (autônomos) e busca por serviços do dia-a-dia. O aplicativo foi desenvolvido para solucionar as necessidades do prestadores de serviço, a começar pela comodidade de cadastrar seus serviços prestados e poder gerenciá-los pelos seus smartphones em qualquer lugar com conexão à internet e também daqueles que precisam de um determinado serviço, podem procurar pelos prestadores em qualquer lugar, caso haja um prestador cadastrado na sua região.

Cada serviço cadastrado no aplicativo poderá ser avaliado por quem o contratou, ajudando outros usuários a encontrar os melhores prestadores de serviço dentro do aplicativo. Cada serviço contará com uma descrição, imagens, valor, número celular para entrar em contato, facilitando a comunicação e fechamento do negócio.

Dessa maneira, este estudo conseguiu atingir seus objetivos, alinhando a popularidade dos *smarthphones* com a necessidade de uma plataforma para unir prestadores de serviços autônomos e aqueles que necessitam de pessoas que realizam serviços do dia-a-dia. O aplicativo além de agregar valor ao trabalho dos prestadores de serviço, traz praticidade e uma maior objetividade para pessoas que precisam desses serviços. Melhorias na usabilidade, layout e novas funcionalidades como pagamento, chat, avaliação do serviço serão realizadas em trabalhos futuros com base nos *feedbacks* dos usuários, além de ajustes e funções disponíveis em outros aplicativos.

REFERÊNCIAS

APPLE. Apple, c2020. **App store**. Disponível em: apple.com/br/ios/app-store. Acesso em: 20 de fev. de 2020.

CIOLFI, Daniel. Udemy, c2020. **Criação de Apps Android e iOS com Flutter**. Acesso em: 02 de set. de 2020.

DAMASCENO, Jamilton. Udemy, c2020. **Desenvolvimento Android e iOS com Flutter**. Disponível em: udemy.com/course. Acesso em: 24 de ago. de 2020.

DÂMASO, Lívia. **Techtudo,** c2000. Notícias. Disponível em: techtudo.com.br. Acesso em: 14 de fev. de 2020.

DEMARTINI, Felipe. **CanalTech,** c2020. Apps. Disponível em: canaltech.com.br/apps. Acesso em: 28 de fev. de 2020.

FLUTTER. **Flutter,** c2020. Flutter Dev. Disponível em: flutter.dev. Acesso em: 02 de mar. de 2020.

LUCCA, Allysson. **O caminho dos Apps**, c2020. ed. Luccaco *be Digital, 2013.

SERVER, Parse, c2020. **Parse Server Documentation**. Disponível em: docs.parseplatform.org. Acesso em: 03 de mar. de 2020.