

PROPOSTA DE UM APLICATIVO DE CONTRATAÇÃO DE PRESTADORES DE SERVIÇOS GERAIS PARA LARES DA REGIÃO

Kevin Gabriel da Silva – @gmail.com,
Thales José de Carvalho – ThallesJC1996@ @gmail.com,
Ana Paula Renno Costa – 00002397@fepi.edu.br

Centro Universitário de Itajubá, Av. Dr. Antônio Braga Filho, 687 - Porto Velho,
Itajubá - MG, 37501-002

RESUMO

As tecnologias desempenham um papel cada vez mais essencial no cotidiano das pessoas, e com o aumento da popularidade dos dispositivos móveis nos últimos anos, a demanda por aplicativos que facilitem o dia a dia tem crescido constantemente. É comum que residências necessitem eventualmente de serviços doméstico, e para atender essa demanda, contratar um profissional qualificado é sempre uma boa opção. No entanto, muitas vezes encontrar alguém para realizar esse serviço pode ser bastante complicado devido à falta de uma plataforma que forneça as informações necessárias.

O objetivo deste projeto é desenvolver um aplicativo que ofereça funcionalidades de busca e anúncio de prestação de serviços doméstico. Ao concluir esse projeto, pretende-se obter um produto viável e funcional, contendo recursos que auxiliem os usuários na busca e divulgação de serviços.

PALAVRAS-CHAVE: Serviço Doméstico, Tecnologia, Aplicativos Móveis.

ABSTRACT

Mobile technologies play an increasingly essential role in people's daily lives, and with the increasing popularity of devices in recent years, the demand for applications that facilitate everyday life has constantly pleased. It is common for homes to eventually need domestic services, and to meet this demand, hiring a qualified professional is always a good option. However, often finding someone to perform this service can be quite complicated due to the lack of a platform that provides the necessary information.

The objective of this project is to develop an application that features search and advertisement functionality for domestic services. Upon completion of this project, the intention is to obtain a viable and functional product, containing resources that help users in the search and dissemination of services.

KEYWORDS: Home Service, Technology, Mobile Apps.

Introdução

A tecnologia vem se expandindo cada vez mais no mundo todo, principalmente em setores ligados a sistemas de informação. Isso mostra que a utilização de uma ferramenta tecnológica facilita em diversos processos e está cada vez mais comum em nosso dia a dia.

Com o avanço do desenvolvimento de novas tecnologias, o uso de aplicativos para facilitar o dia a dia está cada vez maior. Segundo uma pesquisa feita pelo Instituto Locomotiva da cidade de São Paulo em 2021 existiu cerca de 32,4 milhões de pessoas que utilizaram algum tipo de aplicativo para trabalho, isso representa cerca de 20% da população adulta do país. Desse total, cerca de 16% utilizam dos aplicativos como fonte de renda única, isso corresponde a 5,18 milhões de brasileiros. .

O crescimento do uso dessas ferramentas cresceu ainda mais com a pandemia que assolou o mundo, segundo do laboratório Pew Research Center, o uso de aplicativos cresceu em 2020 de 30% a 400% em alguns estados do Brasil durante a pandemia, fazendo assim

que o país ocupe o segundo lugar de crescimento no mercado mobile no mundo, ficando atrás somente da Indonésia. A pandemia somente trouxe a tona um futuro que já era esperado, a facilidade e a velocidade em resolver pequenas coisas torna a tecnologia mobile ainda mais atrativa à população.

Após a pandemia não apenas o mercado mobile teve uma crescente no país, segundo o Instituto Brasileiro de Geografia e Estatística (IBGE), no primeiro trimestre de 2022 houve um crescimento de 20% na procura por empregadas domésticas no país. Utilizando a tecnologia a favor, vários problemas de procura por empregadas poderiam ser resolvidos, auxiliando assim a empregada a solucionar o problema de não encontrar um emprego.

Com isso, tendo em vista a utilização da tecnologia a favor de melhorias e também ao fato de que o setor mobile e o setor de empregadas domésticas estão em alta, este trabalho apresenta o desenvolvimento de um aplicativo mobile utilizando tecnologias para auxiliar as empregadas domésticas

do sul de Minas Gerais a encontrar um trabalho nos lares da região.

Material e Métodos

O presente estudo trata-se de uma revisão da literatura que utilizará de artigos dos portais eletrônicos do Google Scholar tratando da metodologia pesquisa exploratória, cuja finalidade é gerar um artigo científico e proporcionar maior familiaridade com o problema de pesquisa, objetivando construir hipóteses, procurar padrões, ideias ou hipóteses realizando descobertas acerca do assunto principal proposto.

Justificativa

O Brasil se destaca como o país com o maior número de aplicativos baixados no mundo desde a pandemia, com uma taxa de adoção de aplicativos de mais de 98% entre os usuários de smartphones (IBGE, 2022). Considerando que metade da população mundial possui telefone celular e os brasileiros têm uma taxa de posse de 84% (IBGE, 2021), o uso de dispositivos móveis e suas

funcionalidades podem ser extremamente úteis, até mesmo como um facilitador para a conexão entre empregadores e empregadas.

O uso de aplicativos sempre foi uma forma conveniente em várias áreas, inclusive no setor de trabalhos domésticos. No entanto, nesse setor em particular, é comum que o primeiro contato entre empregados e empregadores ocorra por meio de indicações ou buscas em redes sociais, o que muitas vezes dificulta a conexão entre as duas partes.

Diante disso, este trabalho se justifica pela criação de um aplicativo móvel que possa facilitar o encontro de empregadas domésticas na região, proporcionando facilidade e rapidez tanto para aqueles que procuram emprego quanto para aqueles que procuram profissionais nessa área.

O sistema será desenvolvido em ambiente móvel e web, garantindo agilidade para os usuários, pois permitirá que o primeiro contato entre eles seja feito por meio de smartphones ou dispositivos similares.

Metodologia de Pesquisa

Foi realizada uma pesquisa bibliográfica em artigos científicos indexados entre janeiro 2000 e janeiro de 2023 na base de dados do Google Acadêmico. A busca se deu por meio dos unitermos “Serviços Gerais”, “Aplicativos de Serviços Gerais” e “Empregados Domésticos”, presentes nos artigos encontrados.

A amostra do estudo foi constituída por artigos científicos em Português, completos e disponíveis, com publicação entre janeiro de 2000 e janeiro de 2023. Aplicados estes critérios a título de teste de viabilidade, foi considerado um artigo (Anexo1).

As fontes foram analisadas por meio do método de Análise de Conteúdo. Trata-se de uma metodologia de pesquisa utilizada para descrever e fazer interpretações dos conteúdos de falas, documentos, textos – é possível fazer uso de qualquer material do qual provenha comunicação, verbal ou não-verbal (BARDIN,

2010). Decorre da descrição sistemática (quantitativa ou qualitativa) na qual se reinterpreta mensagens, no sentido de encontrar significados não perceptíveis em uma leitura usual. Esse método, segundo Bardin (2010), é um conjunto de análise das comunicações que visa a, por meio de procedimentos sistemáticos e objetivos de descrição do conteúdo, inferir conhecimentos sobre a produção e recepção das mensagens. A análise de conteúdo foi dividida em três fases: pré-análise, exploração do material e tratamento dos resultados, inferências e interpretações.

Metodologia Ágil Scrum

Segundo site do Desenvolvimento Ágil (2014) o Scrum é uma metodologia ágil com objetivo de gerir e planejar projetos de software. Esses projetos são divididos em ciclos de Sprints. O Sprint configura um Time Box onde é executado um conjunto de atividades. As metodologias ágeis por onde os softwares são desenvolvidas são

interativas, sendo assim, tem seus trabalhos divididos em interações, denominados Sprints se tratando de uma Scrum. O Product Backlog é onde ficam armazenadas as funcionalidades em forma de lista, para serem implementados em um projeto.

Ainda segundo o mesmo autor, o Sprint Planning Meeting é feito a partir do início de cada Sprint, onde se reúnem para definir o planejamento no qual o Product Owner elege itens do Product Backlog e são definidas as atividades para cada equipe implementar no início de cada Sprint. As atividades definidas para realizar uma sprint são enviadas do Product Backlog para o Sprint Backlog. São feitas reuniões em cada dia de uma Sprint, geralmente de manhã denominada Daily Scrum. É feita para saber o que já foi feito no dia anterior, verificar se há algum impedimento e eleger o trabalho no início do dia. Ao término de cada Sprint são apresentadas as funcionalidades implementadas através de uma Sprint Review Meeting. Após a conclusão é feita uma Sprint Retrospective e começa um novo planejamento para a próxima Sprint.

Para o desenvolvimento do nosso trabalho, usamos a metodologia ágil Scrum, onde se realizaram quatro Sprints para definir as atividades que seriam feitas. Essas Sprints duravam duas semanas cada uma, sendo possível desenvolver as atividades elencadas durante o início. Na primeira Sprint foram feitos o back-end e o banco de dados, após a realização dessas atividades iniciou-se a segunda Sprint, onde foi desenvolvido o front-end e usado Expo para visualização do funcionamento. Então partimos para a terceira Sprint onde foi feita a integração da Web Api do back-end com o front-end. Na quarta Sprint realizamos os testes e validações para concluir o desenvolvimento do software.

Back-end e Banco de dados

O desenvolvimento da Web API começou com a criação do back-end utilizando a plataforma NodeJS, juntamente com a linguagem de programação Javascript, compilando no ambiente do Visual Studio Code. Para iniciar o projeto, foi criada uma pasta para todos os arquivos necessários para

o back-end. Por meio do terminal, acessando essa pasta, o yarn foi instalado utilizando o comando "yarn init -y" para configurar o projeto inicialmente. Para instalar os pacotes e módulos necessários para manter o projeto funcionando, o express foi instalado usando o comando "yarn add express".

Para evitar ter que reiniciar o servidor toda vez que desejar iniciar o projeto, o nodemon foi instalado com o comando "yarn add nodemon". Essa ferramenta permite que o projeto seja compilado enquanto fazemos alterações nos arquivos. Para isso, um script de inicialização foi criado no "package.json". Quando o projeto é iniciado usando o comando "yarn dev", o aplicativo continuará funcionando até que seja solicitado para ser interrompido.

Foi utilizado um ORM (Object-Relational Mapper) chamado Sequelize com suporte para NodeJS para definir o banco de dados. Segundo a documentação oficial do Sequelize, "Sequelize é um ORM baseado em promessas para Node.js e io.js. Suporta os bancos de dados PostgreSQL, MySQL, MariaDB,

SQLite e MSSQL e possui um poderoso sistema de consulta, migração de dados e gerenciamento de transações" (Sequelize, 2022). Para preparar o Sequelize, foi criado um arquivo na raiz do projeto chamado "sequelizerc". Esse arquivo indica o caminho de configuração do Sequelize dentro do projeto e para as migrações.

Antes de criar o banco de dados, é necessário criar um arquivo de configuração chamado "database.js". Esse arquivo contém código que define o acesso ao localhost em nossa máquina, o dialeto utilizado (SQL) e o nome do banco de dados.

Através do terminal do projeto, foram instalados o Sequelize e seu suporte ao MySQL para NodeJS com os comandos "yarn add sequelize sequelize-cli" e "yarn add mysql2", respectivamente. Com o banco de dados "bancoAgro" já criado com o comando "yarn sequelize db:create", foram definidos modelos para cada tabela antes da criação das migrações. Esses modelos gerenciam as regras dos dados nas funções, incluindo seus tipos e tabelas associadas. As migrações foram criadas com o

comando "yarn sequelize migration:create --name=create-usuarios", contendo as definições das colunas e seus tipos desejados para as tabelas de usuários, animais, dados de produção e sanitários.

Para aplicar as migrações, foi executado o comando "yarn sequelize db:migrate" para definir as tabelas no banco com suas colunas de acordo com as configurações dos arquivos de migração. As controllers, que intermediam as requisições das views e fornecem as respostas da model, foram construídas para processar os dados fornecidos pelo usuário e controlá-los para que possam ser respondidos de forma adequada. Foram criados métodos HTTP nas controllers, como GET para solicitação de informações, POST para recebimento de dados do usuário, PUT para atualização de informações e DELETE para exclusão de dados.

No arquivo principal do back-end, o "index.js", foram realizadas a configuração e inicialização das rotas criadas na aplicação, bem como a chamada da base de dados das models definidas, contendo

também a definição da porta de funcionamento.

Front-End

Durante o processo de decisão sobre a escolha da biblioteca para o desenvolvimento da interface do usuário, foram levados em consideração diversos fatores, tais como recursos, facilidade de desenvolvimento e popularidade da biblioteca. A React Native, uma das mais populares bibliotecas de desenvolvimento mobile em Javascript, foi escolhida devido à sua capacidade de fornecer recursos úteis para a criação de aplicativos móveis. Segundo Arjun, em seu artigo sobre as vantagens do uso da React Native, "Os desenvolvedores podem criar aplicativos móveis incríveis usando apenas Javascript, o que não só economiza tempo e dinheiro, mas também torna o processo de desenvolvimento mais simples e eficiente".

Para o desenvolvimento do código, foi escolhido o ambiente do Visual Studio Code, uma das IDEs mais populares no mercado, que oferece diversos recursos para

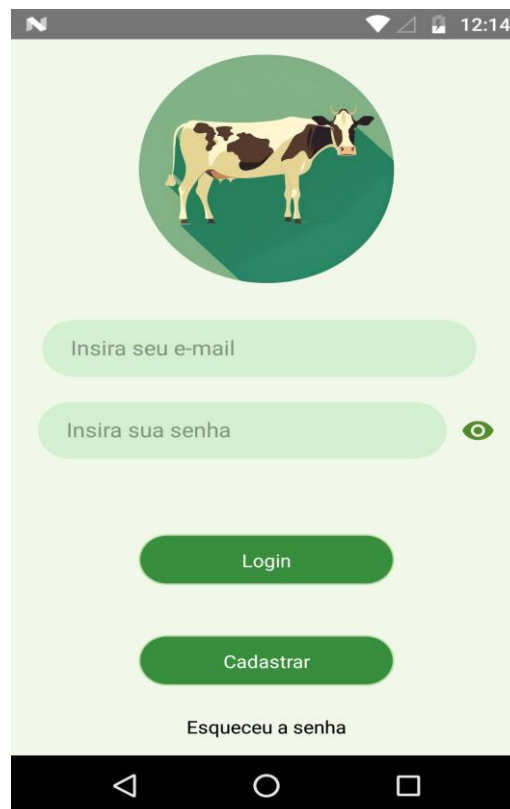
tornar o processo de desenvolvimento mais fácil e produtivo. De acordo com o relatório State of JS de 2020, o Visual Studio Code é a IDE mais utilizada pelos desenvolvedores Javascript, o que confirma a escolha da equipe de desenvolvimento em relação à sua popularidade (State of JS, 2020).

A escolha da React Native e do Visual Studio Code permitirá que a equipe de desenvolvimento crie uma interface de usuário eficiente e fácil de usar, além de um código organizado e de fácil manutenção. Essa combinação também fornecerá ao usuário uma experiência mais agradável, contribuindo para o sucesso do sistema desenvolvido.

A figura 01, abaixo mostra a tela de login, onde o usuário irá inserir o seu email e senha caso já possua cadastro, caso contrário, ele deverá clicar no botão cadastrar, e irá redirecionar para a tela de cadastro (figura 02).

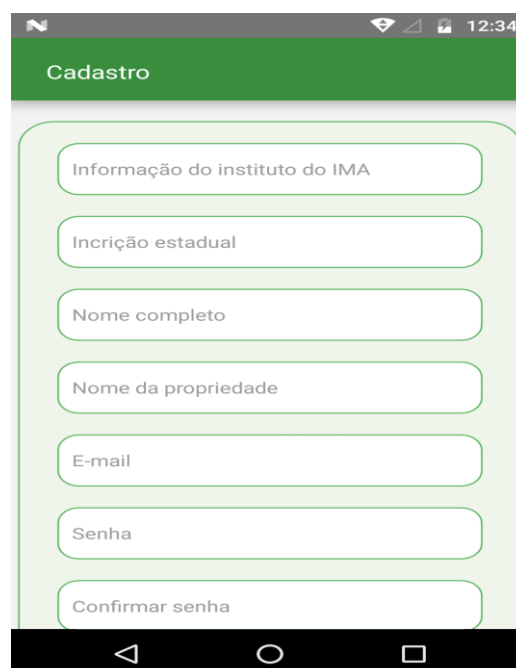
Se o usuário por um lapso, esqueceu sua senha, na opção “esqueceu a senha”, é possível fazer a redefinição dela.

Figura 01 - Telas Login do Aplicativo



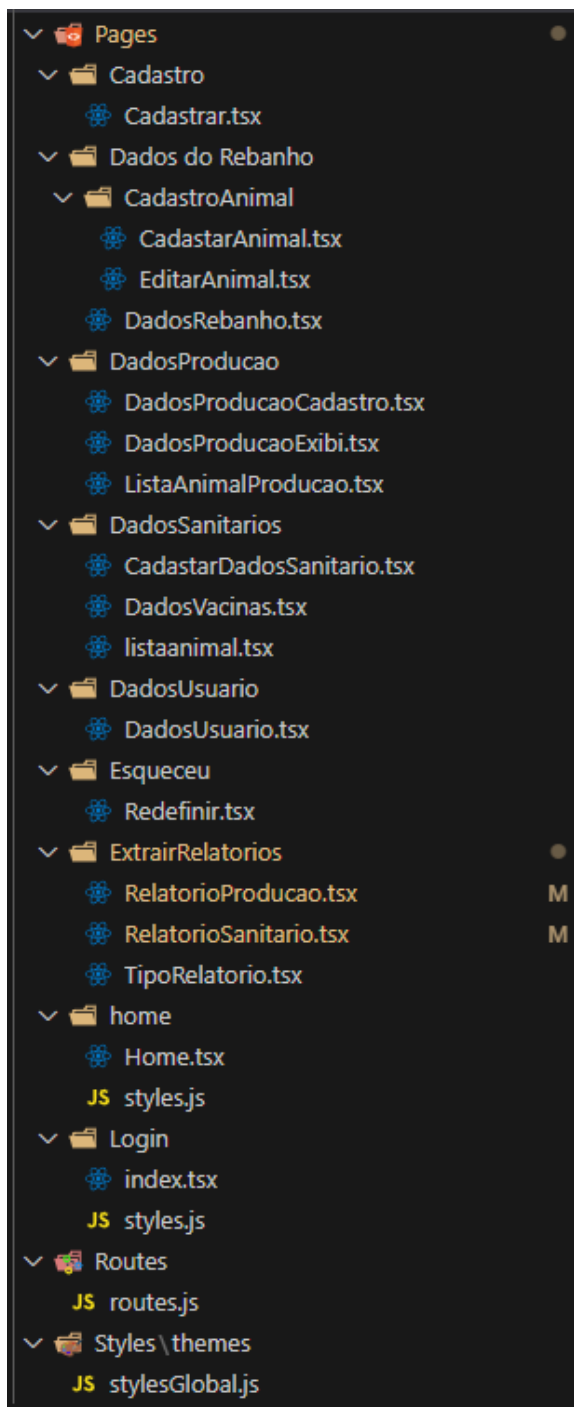
Fonte: Elaborado pelo Autor

Figura 02 - Telas Cadastro do Aplicativo



Fonte: Elaborado pelo Autor

Figura 03 - Arquivos de tela



Fonte: Elaborado pelo Autor

No desenvolvimento do front-end da aplicação, foi adotada uma abordagem organizada para criar as telas que seriam visualizadas e interagidas pelo usuário. Para isso,

foram criadas pastas específicas para cada tela desejada, com nomes que identificam claramente o propósito de cada tela (Figura 03).

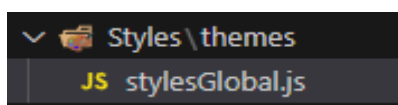
Conforme afirmação de Dan Abramov, em seu artigo sobre boas práticas em React: "Organizar os componentes em pastas e arquivos separados de acordo com suas funções torna o código mais fácil de entender e manter". Dessa forma, a organização das pastas em nosso projeto permite uma melhor compreensão e manutenção do código.

Dentro de cada pasta, encontra-se o arquivo "index.tsx", que contém toda a configuração de funcionamento da tela, como a chamada de componentes, textos, design, imagens e chamadas de funções. Além disso, para garantir um design atrativo e bem estruturado, foi criado o arquivo de CSS chamado "styles.js" dentro da mesma pasta (Figura 04).

De acordo com a documentação oficial do React Native, a criação de arquivos separados para estilos é uma prática recomendada, uma vez que permite o reuso de estilos em diversos componentes, e ainda

mantém o código organizado e fácil de manter. Com a adoção dessa abordagem em nosso projeto, buscamos garantir uma interface de usuário bem estruturada e de fácil manutenção.

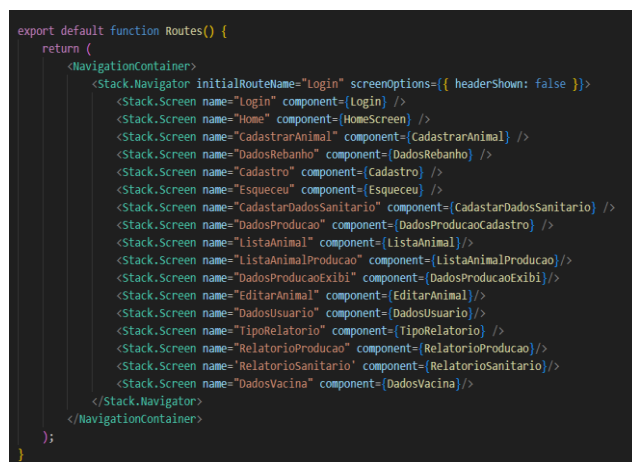
Figura 04 - Arquivo global de cores trocar



Fonte: Elaborado pelo Autor

Foram criados arquivos específicos para a navegação entre telas (conforme ilustrado na Figura 05), cada um com sua respectiva configuração de rotas. O arquivo "routes.js" é fundamental para interligar todas as telas do aplicativo após o login, permitindo a transição da tela Home para a tela de informações dos animais, por exemplo. Sendo o arquivo principal de rotas, ele determina qual rota será acessada no aplicativo.

Figura 05 - Arquivos de rotas de navegação trocar



Fonte: Elaborado pelo Autor

Integração da Web Api

Em resumo, para integrar o front-end e o back-end em uma aplicação, é comum utilizar bibliotecas e funções em JavaScript. As bibliotecas mais utilizadas para desenvolvimento front-end são React, Vue.js e Angular, enquanto as mais populares para back-end são Express, Koa e NestJS. O JavaScript oferece uma ampla variedade de bibliotecas e frameworks para criar uma arquitetura de aplicação robusta e escalável. Para a integração entre front-end e back-end, é necessário configurar o arquivo de rotas, que define as rotas pelas quais o front-end pode acessar e utilizar os

métodos HTTP definidos nas controllers do back-end. A escolha adequada de bibliotecas e funções pode facilitar o processo de integração e torná-lo mais eficiente (Zampieri, 2019).

Resultado e discussões.

O objetivo deste trabalho foi desenvolver um aplicativo para dispositivos móveis que facilite a conexão entre prestadores de serviços e contratantes. O objetivo principal é permitir que as pessoas encontrassem profissionais que ofereçam uma ampla variedade de serviços. O aplicativo foi projetado para agilizar a comunicação entre ambas às partes.

Uma das principais funcionalidades do aplicativo é facilitar a comunicação entre contratantes e prestadores de serviços, permitindo que o contratante entre em contato com o prestador através do WhatsApp, com base nas informações fornecidas pelo prestador sobre seus serviços. Para o desenvolvimento do aplicativo, foi escolhida a plataforma Flutter, que oferece diversas funcionalidades que aumentam a

produtividade durante o desenvolvimento. O armazenamento de dados e autenticação foram implementados utilizando o Firebase, que fornece recursos em tempo real, segurança, API pronta e outras funcionalidades.

Foram desenvolvidas telas para o cadastro de serviços gerais, profissionais e clientes, permitindo também a edição dos cadastros de serviços e perfis, bem como a exclusão de serviços. As telas foram projetadas com interações e fluxo de navegação entre elas.

Com base nos resultados obtidos, constatou-se que o aplicativo oferece recursos básicos satisfatórios para atender às necessidades de busca e divulgação de serviços. O desenvolvimento do aplicativo empregou tecnologias modernas, incluindo React, JavaScript, Laravel e Firebase.

Do ponto de vista acadêmico, a realização deste trabalho contribuiu significativamente para a formação em desenvolvimento de aplicações mobile. Além disso, o processo de desenvolvimento dessa aplicação permitiu ao autor adquirir novos conhecimentos e competências, ao mesmo tempo em que consolidou

habilidades previamente adquiridas durante a graduação.

Referencias

NORMANN, Richard. Administração de serviços: estratégia e liderança na empresa de serviços. São Paulo: Atlas, 1993. Acesso em: 29 maio 2023.

TÉBOUL, James. A era dos serviços: uma nova abordagem ao gerenciamento. Rio de Janeiro: Qualitymark Ed., 1999 Acesso em: 29 maio 2023.

LEUCOTRON. Tecnologia da informação e comunicação. Disponível em: <https://blog.leucotron.com.br/tecnologia-da-informacao-e-comunicacao/>. Acesso em: 29 maio 2023.

Sertms. Benefícios da tecnologia. Blog Sertms, [s.d.]. Disponível em: <https://www.sertms.com/blog/beneficios-da-tecnologia/>. Acesso em: 29 maio 2023.

G1. (2022, 25 de setembro). MEIs representam quase 70% das empresas em atividade no Brasil: só

em 2021, mais de 3 milhões de cadastros foram criados. PEGN.

Disponível em:

<https://g1.globo.com/empreendedoris/mo/pegn/noticia/2022/09/25/meis-representam-quase-70percent-das-empresas-em-atividade-no-brasil-so-em-2021-mais-de-3-milhoes-de-cadastros-foram-criados.ghtml>.

Acesso em: 29 maio 2023.

Attri. (s.d.). Aplicativo de prestação de serviços. Disponível em: <https://www.attri.com.br/blog/aplicativo-o-prestacao-de-servicos/>. Acesso em: 29 maio 2023.

REFERA. Aplicativo Prestadores de Serviços: Tudo o que você precisa saber. Disponível em: <https://refera.com.br/blog/aplicativo-prestadores/>. Acesso em: 29 maio 2023.

TECHTUDO. Aplicativo de diarista e serviços gerais: veja lista com melhores apps. Disponível em: <https://www.techtudo.com.br/listas/2019/07/aplicativo-de-diarista-e-servicos-gerais-veja-lista-com-melhores-apps.ghtml>. Acesso em: 29 maio 2023.

GETNINJAS. Plataforma GetNinjas.

Disponível em:

<https://www.getninjas.com.br/#what-is-getninjas>. Acesso em: 29 maio 2023.

INTELIGÊNCIA & INOVAÇÃO.

Triider. Disponível em:

<https://inteligenciaeinovacao.com/triider-uber-dos-servicos-domesticos-chega-a-sao-paulo/>.

Acesso em: 29 maio 2023.

Information Science, v. 893, p. 168-175, 2018.

BARDIN, L. Análise de Conteúdo. 4. ed. Lisboa: Edições 70, 2010.