

**FACULDADE DE TECNOLOGIA DE SÃO ROQUE**

**FATEC SÃO ROQUE**

**ALBERTO ANDRÉ AGUIAR NETO**

**A UTILIZAÇÃO DE TECNOLOGIAS HíBRIDAS PARA O DESENVOLVIMENTO DE UM APLICATIVO PARA AUXILIAR A EDUCAÇÃO DE CRIANÇAS AUTISTAS**

São Roque

2020



**ALBERTO ANDRÉ AGUIAR NETO**

**A UTILIZAÇÃO DE TECNOLOGIAS HíBRIDAS PARA O DESENVOLVIMENTO DE UM APLICATIVO PARA AUXILIAR A EDUCAÇÃO DE CRIANÇAS AUTISTAS**

Trabalho de Graduação apresentado à Faculdade de Tecnologia São Roque, como parte dos requisitos necessários para a obtenção do título de Tecnólogo em Sistemas para Internet.

**Orientador: Prof.ª Ma Adriana Paula Borges**

São Roque

2020

**Dados Internacionais de Catalogação-na-Publicação (CIP)**

**Divisão de Informação e Documentação**

AGUIAR NETO, Alberto André

A Utilização de Tecnologias Híbridas para o Desenvolvimento de Um Aplicativo para Auxiliar a Educação de Crianças Autistas.

São Roque, 2020.

999f.

Trabalho de Graduação – Curso de Tecnologia em Sistema para Internet

FATEC de São Roque, 2020.

Orientador: Prof.ª Mª Adriana Paula Borges.

1. Sistemas para Internet. I. Faculdade de Tecnologia. FATEC de São Roque: II. A Utilização de Tecnologias Híbridas para o Desenvolvimento de Um Aplicativo para Auxiliar a Educação de Crianças Autistas.

**REFERÊNCIA BIBLIOGRÁFICA –**

AGUIAR NETO, Alberto André. **A Utilização de Tecnologias Híbridas para o Desenvolvimento de Um Aplicativo para Auxiliar a Educação de Crianças Autistas.** 2020. 999f. Trabalho de Graduação - FATEC de São Roque.

**CESSÃO DE DIREITOS**

NOME DO AUTOR: Alberto André Aguiar Neto

TÍTULO DO TRABALHO: A Utilização de Tecnologias Hibridas para o Desenvolvimento de Um Aplicativo para Auxiliar a Educação de Crianças Autistas

TIPO DO TRABALHO/ANO: Trabalho de Graduação / 2020.

É concedida à FATEC de São Roque permissão para reproduzir e emprestar cópias deste Trabalho somente para propósitos acadêmicos e científicos. O autor reserva outros direitos de publicação e nenhuma parte deste Trabalho pode ser reproduzida sem a autorização do autor.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Alberto André Aguiar Neto

RG 44.323.308-1

# dedicatória

Dedico este trabalho de graduação, especialmente, a minha noiva Rafaela, com muito amor, carinho e gratidão. Sem ela nada disso seria possível.

# AGRADECIMENTOS

Agradeço à minha família pelo suporte de todas as horas.

À minha noiva Rafaela pela compreensão e paciência demonstrada durante o período do curso que enseja este trabalho.

E um agradecimento especial aos professores Ademar, Adriana e Douglas, pelo apoio e ensinamentos que serviram de alicerce para as minhas realizações.

**Resumo**

A falta de infraestutura é uma das maiores dificuldades para a modernização do ensino e inclusão de tecnologias nas escolas. Quando tratamos sobre a educação para crianças autistas, essa escassez de ferramentas tecnológicas tende a aumentar. Para superar tal dificuldade, faz-se necessário implementar tecnologias desenvolvidas a fim de se obter metodologias e técnicas inovadoras que possibilitam a facilidade e usabilidade de forma educacional.

O presente trabalho apresentará informações detalhadas referente a criação de um aplicativo para dispositivos móveis desenvolvido utilizando uma tecnologia híbrida, qual visa inovar e auxiliar novas formas de ensino.

**Palavras-Chave**: aplicativo; autismo; educação; tecnologia híbrida; *Ionic.*

**ABSTRACT**

*The lack of infrastructure is one of the greatest difficulties for the modernization of teaching and the inclusion of technologies in schools. When it comes to education for autistic children, this shortage of technological tools tends to increase. To overcome this difficulty, it is necessary to implemente Technologies developed in order to obtain innovative and methodologies and techniques that enable ease and usabiliti in na educacional way.*

*This work will presente detailed information regarding the creation of na application for mobile devices developed using a hybrid technology, which aims to innovate and assist new forms of teaching.*

***Keywords****: application; autism; education; hybrid techonology; Ionic.*

**LISTA DE FIGURAS**

|  |  |
| --- | --- |
| Figura 1: *Wireframe* da página inicial do aplicativo ............................................ | 19 |
| Figura 2: *Wireframe* da página Sobre o aplicativo .............................................. | 20 |
| Figura 3: Tela que lista todos os *flashcards* no sistema ..................................... | 21 |
| Figura 4: Popup de exclusão de *flashcard* .......................................................... | 21 |
| Figura 5: Mensagem de sucesso de exclusão de *flashcard* ............................... | 22 |
| Figura 6: Tela do formulário de adição e edição de *flashcard* ............................ | 23 |
| Figura 7: Mensagem de sucesso ao adicionar um *flashcard* .............................. | 23 |
| Figura 8: Tela de detalhes do *flashcard* .............................................................. | 24 |
| Figura 9: Iniciando um aplicativo utilizando o *Ionic* ............................................. | 25 |
| Figura 10: Criação dos serviços, classes e páginas necessárias para o projeto desenvolvido ....................................................................................................... | 26 |
| Figura 11: Código *Ionic* da página inicial ............................................................ | 27 |
| Figura 12: Versão final da página inicial do aplicativo ........................................ | 28 |
| Figura 13: Código *Ionic* da página Sobre o aplicativo ........................................ | 29 |
| Figura 14: Versão final da página Sobre o aplicativo ......................................... | 29 |
| Figura 15: Código *Ionic* da tela de listagem de *flashcards* ................................. | 30 |
| Figura 16: *Imports* do arquivo *angular* da página de listagem de todos os *flashcards* ............................................................................................................ | 31 |
| Figura 17: Métodos de listagem de todos os *flashcards* e exclusão de algum *flashcard* específico ............................................................................................ | 32 |
| Figura 18: Versão final da página que lista todos os *flashcards* ......................... | 33 |
| Figura 19: Código *Ionic* do formulário de *flashcard* ............................................ | 34 |
| Figura 20: Código *Angular* da página de formulário, onde é realizada a identificação se é edição ou adição de *flashcards* e onde é realizado o envio do *flashcard* para o banco de dados ................................................................... | 35 |
| Figura 21: Versão final do formulário de adição e edição de *flashcard* .............. | 36 |
| Figura 22: Código *ionic* da página de detalhes do *flashcard* .............................. | 37 |
| Figura 23: Código *Angular* da página de detalhes do *flashcard* ......................... | 37 |
| Figura 24: Versão final da página de detalhes do *flashcard* ............................... | 38 |
| Figura 25: Código do serviço de banco de dados, responsável por realizar todas as atividades do banco de dados da aplicação ........................................ | 39 |
| Figura 26: Métodos de inserção, atualização e exclusão do serviço de *flashcard* ............................................................................................................. | 40 |
| Figura 27: Métodos de listagens do serviço de *flashcards* ................................. | 41 |
| Figura 28: Classe de *flashcard*, para controlar o tipo de dados do objeto .......... | 42 |

**LISTA DE abreviaturas e siglas**

API *Application Programming Interface*

CDC *Center for Disease Control and Prevention*

CMD *Command Prompt*

CSS *Cascading Style Sheets*

DOM *Document Object Model*

HTML *Hypertext Makeup Language*

MVC *Model, View and Controller*

NPM *Node Package Manager*

TEA Transtorno do Espectro Autista

URL *Uniform Resource Locator*

**SUMÁRIO**

|  |  |
| --- | --- |
| 1. INTRODUÇÃO ................................................................................................ | 10 |
| 1.1. Objetivo ........................................................................................................ | 11 |
| 1.2. Justificativa .................................................................................................. | 12 |
| 2. REFERÊNCIAS BIBLIOGRÁFICAS ............................................................... | 13 |
| 2.1. A utilização de linguagens híbridas no desenvolvimento de sistemas ........ | 13 |
| 2.2. *Ionic*, *Angular* e *SQLite* como ferramentas de desenvolvimento híbrido ..... | 14 |
| 2.2.1. *Ionic* .......................................................................................................... | 14 |
| 2.2.2. *Angular* ...................................................................................................... | 15 |
| 2.2.3. *SQLite* ....................................................................................................... | 16 |
| 2.3. Sistemas Operacionais ................................................................................ | 16 |
| 2.3.1. Android ...................................................................................................... | 17 |
| 2.3.2. iOS ............................................................................................................ | 17 |
| 3. MODELAGEM ................................................................................................. | 19 |
| 3.1. *Wireframes* ................................................................................................... | 19 |
| 3.1.1. Página Inicial ............................................................................................. | 19 |
| 3.1.2. Sobre o Aplicativo ..................................................................................... | 20 |
| 3.1.3. Listagem dos *flashcards* ........................................................................... | 20 |
| 3.1.4. Formulário de *flashcard* ............................................................................ | 22 |
| 3.1.5. Detalhes do *flashcard* ............................................................................... | 24 |
| 4. PROTOTIPAÇÃO ........................................................................................... | 25 |
| 4.1. Páginas do aplicativo ................................................................................... | 27 |
| 4.1.1. Página Inicial ............................................................................................. | 27 |
| 4.1.2. Sobre o Aplicativo ..................................................................................... | 28 |
| 4.1.3. Todos os *Flashcards* ................................................................................. | 30 |
| 4.1.4. Formulário de *flashcard* ............................................................................ | 33 |
| 4.1.5. Detalhes do *flashcard* ............................................................................... | 36 |
| 4.2. Serviços do aplicativo .................................................................................. | 38 |
| 4.2.1. Serviço de banco de dados ...................................................................... | 39 |
| 4.2.2. Serviço de *Flashcard* ................................................................................ | 39 |
| 4.3. Classes do aplicativo ................................................................................... | 42 |
| 5. CONSIDERAÇÕES FINAIS ............................................................................ | 43 |
| 6. REFERÊNCIAS .............................................................................................. | 44 |

# 1. INTRODUÇÃO

O autismo, conhecido cientificamente como Transtorno do Espectro Autista – TEA, é um transtorno que causa problemas no desenvolvimento da linguagem, nos processos de comunicação, na interação e comportamento social de uma pessoa. No Brasil, embora não exista nenhum estudo de prevalência de autismo, com cálculos estatísticos, estima-se que, com aproximadamente 200 milhões de habitantes, há cerca de 2 milhões de pessoas autistas, resultado baseado nos dados mundiais colhidos pelo CDC (*Centers for Disease Control and Prevention* – Centro de Controle e Prevenções de Doenças).

Com as dificuldades decorrentes do transtorno, a educação de uma criança autista tende a demandar mais atenção e técnicas de ensino, a fim de extrair da melhor maneira o que é chamado de tríade do autismo: a socialização, a comunicação e o comportamento.

Nota-se que o processo de educação da criança com espectro autista é moroso, caracterizando-se pela repetição daquilo que é dito por terceiros ou pela substituição das palavras por sons, bem como repetições de gestos realizados por terceiros ou vistos em imagens.

Embora atualmente exista a facilidade e variedade de mecanismos tecnológicos, é notória a escassez de ferramentas na utilização da metodologia de aprendizagem e desenvolvimento intelectual de uma criança portadora do TEA.

Nessa perspectiva, o sistema desenvolvido promove ao educador uma técnica inovadora em auxiliar o aluno autista a realizar as ações apresentadas em forma de desenhos claros e objetivos, de modo que ao longo do processo, o aluno desenvolva a ação sozinho.

O nome dado a ferramenta é *Flashcards*, caracterizando-se em cartões que contém informações de algo específico, além de uma imagem, o que faz operar a semiótica do autista, assimilado a imagem as informações contidas no cartão. A ação apresentada pode ser utilizada em atividades básicas, como exemplo, na higiene diária (escovar os dentes, lavar as mãos e tomar banho). O sistema desenvolvido é um recurso de apoio visual, de maneira que usado de forma habitual, o processo de educação e autonomia será desenvolvido sutilmente.

De modo geral, ao aplicar a técnica das ações ilustradas na tela do smartphone ou tablet, é oferecido aos educados que as crianças autistas desenvolvam independência e promovam o desenvolvimento nas habilidades cognitivas, além de capacidades de interagir em seu dia a dia.

Para a maior utilização do sistema, é possível que os educadores, pais ou algum outro responsável pela criança, edite as instruções das tarefas, bem como adicione novas tarefas personalizando com uso de figuras da biblioteca de imagens existentes no smartphone ou buscando imagens da internet, de modo que reforce a sequência de atividades da rotina e auxilie no processo de aprendizagem e autonomia.

O sistema é um aplicativo móvel, desenvolvido através da tecnologia hibrida*,* com um banco de dados descomplicado, acessível e de fácil utilização, disponibilizado para todas as plataformas.

Haja vista o sistema operacional para dispositivos móveis, a tecnologia escolhida para o desenvolvimento do sistema será o *Ionic* em sua versão 4,através da linguagem de *Framework JavaScript.* O banco de dados será executado através do *SQLite* em sua versão mais recente, e além disso, será utilizado o *Angular* na versão embutida ao *Ionic,* também na versão 4.

## Objetivo

Apresentar detalhadamente os procedimentos desenvolvidos para obtenção do aplicativo móvel, a fim de corroborar com a finalidade do projeto.

Desenvolver um aplicativo utilizando uma tecnologia híbrida, a promover facilidade no processo de educação das crianças autistas em forma de ilustrações, de modo que os educandos compreendam atividades diárias.

Demonstrar a eficácia do desenvolvimento hibrido nas tecnologias atuais, de modo a contribuir para o desdobramento de novos recursos tecnológicos de aprendizagem.

Inovar uma ferramenta tecnológica na área da educação a ser utilizada como forma de metodologia de ensino, permitindo um processo de aprendizagem inovadora.

Analisar a viabilidade da utilização da linguagem hibrida ao invés da linguagem nativa.

Expor o crescimento e desenvolvimento de aplicações móveis utilizando linguagens híbridas.

## Justificativa

Levando em consideração o aumento significativo de crianças portadoras do TEA nas escolas, é necessário que os educadores possuam novas metodologias de ensino, utilizando-se de técnicas inovadoras para facilitar o desempenho de aprendizagem desses alunos.

Portanto, o foco principal deste projeto encontra-se no desenvolvimento de um aplicativo utilizando tecnologia híbrida de maneira que auxilia os educadores no processo de educação da criança autista, além de inovar as opções de aplicações a serem baixadas através de smartphones ou tablets, visto que referido tema é escasso no ambiente tecnológico.

O público-alvo escolhido para base deste projeto estão relacionadas a dois grandes polos: os educadores e os alunos portadores de autismo. A ligação entre eles impulsiona o acesso ao aplicativo, visto a crescente demanda nas escolas, viabilizando assim, novas alternativas de metodologia de ensino. Por este motivo, o aplicativo apresenta a acessibilidade e a usabilidade aos usuários.

A codificação com a tecnologia híbrida apresenta facilidade e possibilidade de ser desenvolvida para diversas plataformas utilizando-se de apenas uma linguagem, ou seja, com *framework* *Ionic,* é possível desenvolver para a plataforma *Android, iOS* e *Windows.*

# 2. REFERÊNCIAS BIBLIOGRÁFICAS

Neste capítulo conceitos teóricos serão esclarecidos a fim de fornecer o entendimento básico das linguagens de programação utilizadas no projeto, além de demonstrar em forma de números e estatísticas o crescimento da utilização da tecnologia híbrida no desenvolvimento de sistemas.

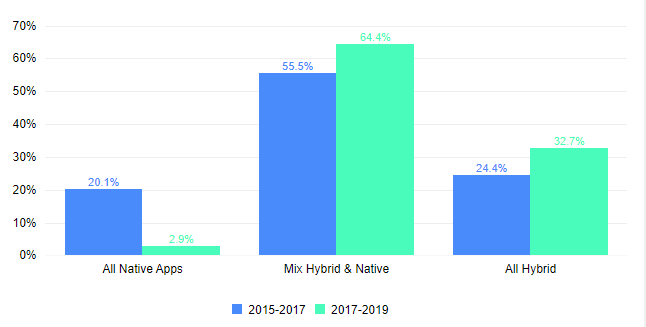
## 2.1. A utilização de linguagens híbridas no desenvolvimento de sistemas

Embora as tecnologias híbridas ainda não tenham tomado as mesmas proporções das tecnologias nativas, elas têm evoluído significativamente, visto que a utilização dos dispositivos móveis está cada vez maior, tornando os aplicativos uma ferramenta de avanço tecnológico.

As ferramentas híbridas visam diminuir a dificuldade de desenvolvimento de aplicações, para assim otimizar o tempo, diminuir a curva de aprendizado de uma tecnologia, dada a consideração de que, de maneira nativa, para desenvolver para mais de um sistema operacional, é necessário aprender as tecnologias referente ao sistema operacional desejado, tal como o Java é utilizado para o desenvolvimento de aplicações nativas para o *Android* e o *Swift/Objective-C* para aplicações nativas do *iOS*.

Com o passar do tempo, a utilização de tecnologias híbridas vem aumentando de maneira significativa, pois torna o processo de desenvolvimento de uma aplicação menos custoso, já que não demanda profissional especialista em um único sistema operacional.

Hoje em dia, há cerca de 32.7% de aplicações desenvolvidas de maneira totalmente híbrida, e apenas 2.9% de aplicações desenvolvidas de maneira totalmente nativa.



**Gráfico 1 – Comparação de aplicações móveis nativas, mistura de híbridas e nativas e totalmente híbridas**

**Fonte: (*Ionic Survey*, 2019)**

## 2.2. *Ionic*, *Angular* e *SQLite* como ferramentas de desenvolvimento híbrido

Para que ocorra o desenvolvimento híbrido, é necessário ferramentas e serviços que disponibilizem um *framework* simples, operante e ativo. Em meio a tantos, destacam-se o *Ionic*, por se tratar de um *framework* completo para o desenvolvimento de aplicativos híbridos, capaz de criar aplicações usando tecnologia *Web;* o *Angular* por ser uma poderosa ferramenta de manipulação do *DOM*, com uma vasta documentação e o *SQLite*, por ser um sistema de banco de dados de fácil implementação e utilização.

## 2.2.1. *Ionic*

O *Ionic* é um *framework* de código aberto, ou seja, de utilização gratuita, que visa o desenvolvimento de sistemas em uma só linguagem, porém que seja executado em diversos sistemas operacionais.

O *framework Ionic* foi criado no ano de 2013 pela empresa Drifty Co., através de seus fundadores programadores Max Lynch e Adam Braley e pelo designer Ben Sperry. Em novembro de 2013 o *Ionic* teve a sua primeira versão alpha lançada. Em março de 2014 fora lançada a versão beta 1.0, que teve a sua versão final em maio de 2014.

Para obter um desempenho satisfatório nas plataformas quais o *Ionic* se dispõe a funcionar, o *Ionic* utiliza o *Apache Cordova*, que é uma estrutura de desenvolvimento de aplicativos móveis que permite que seja possível criar aplicações híbridas utilizando *HTML*, *CSS* e *JavaScript*, não dependendo das *APIs* de plataformas específicas, como *iOS* ou *Android*.

Os aplicativos que são desenvolvidos utilizando o *framework Ionic* não são nativos, entretanto, devido ao esforço dos desenvolvedores do *framework*, eles se assemelham e atuam como tal, podendo utilizar os recursos nativos dos dispositivos, tal como câmera, entrada e saída de áudio e notificação *push*.

* Há diversas vantagens em utilizar o *Ionic Framework* para o desenvolvimento de aplicações híbridas, tais como:
* Manutenção centralizada de código. Será necessário alterar somente uma vez, mesmo que para vários sistemas operacionais;
* Comunidade de desenvolvimento ativa, onde é possível encontrar diversos tutoriais e exemplos de desenvolvimento;
* Documentação completa;
* Desenvolvimento de aplicações móveis com baixa complexidade e economia de tempo;
* Você escreve apenas uma vez e roda em qualquer lugar, que é o slogan do *Ionic*, inclusive.

## 2.2.2. *Angular*

O *Angular*, que a princípio era conhecido e denominado como *AngularJS*, é um *framework front-end*, utilizado majoritariamente para a manipulação do DOM (*Document Object Model*) do *HTML*, desenvolvido inicialmente em *JavaScript*, que tinha como principal objetivo padronizar aplicações web com *MVC* (*model, view e controller*), facilitando assim o desenvolvimento de sistemas.

O *AngularJS* foi criado pelos programadores Misko Hevery e Adams Abrons, no ano de 2009 e hoje é mantido pelo Google.

Em setembro de 2014 foi anunciado o *Angular* 2.0, que passou a ser considerado um novo *framework*, que não é compatível com a versão anterior, *AngularJS*. Em abril de 2015 a versão alfa do *Angular* 2.0 foi lançada e em dezembro de 2015 a versão beta. A versão final do *framework* foi lançada em setembro de 2016. A principal diferença do *AngularJS* para o *Angular* 2.0 é a linguagem base. Enquanto no *AngularJS* a linguagem base é o *JavaScript*, no *Angular* 2.0 a linguagem base é o *TypeScript*.

Atualmente o *Angular* está em sua versão 8, que tem retro compatibilidade com a versão 2.0.

## 2.2.3. *SQLite*

O *SQLite* foi criado pelo programador Dwayne Richard Hipp, no ano 2000, enquanto ele era funcionário da *General Dynamics*. O objetivo do *SQLite* é permitir que uma aplicação seja operada sem a instalação de um sistema de gerenciamento de banco de dados. Essa ferramenta nada mais é do que um arquivo de texto que armazena todas as informações solicitadas. Em agosto de 2000 a versão 1.0 do *SQLite* foi lançada.

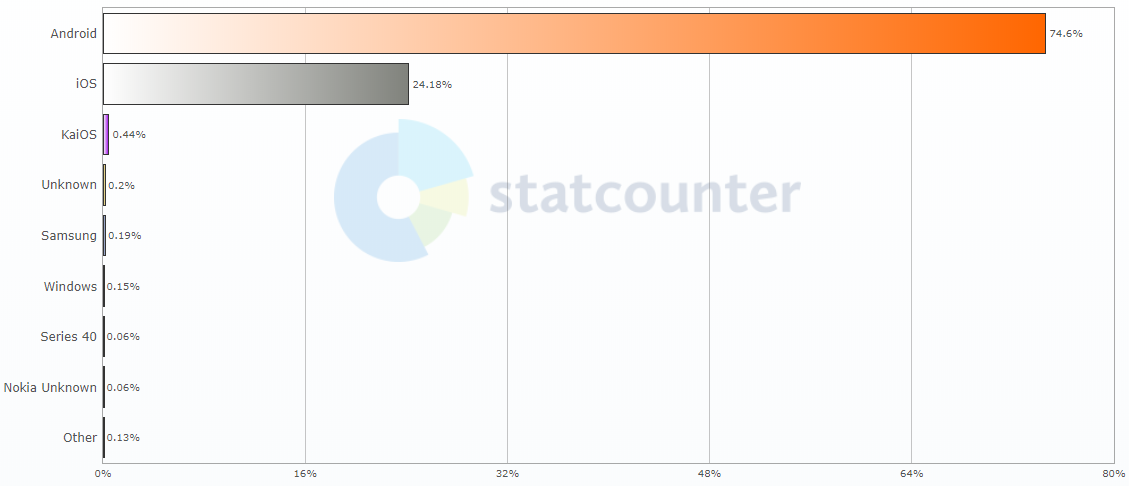
O *SQLite* é considerado um sistema de gestão de base de dados relacional, uma biblioteca *open-source* que tem como linguagem de desenvolvimento a linguagem C. A tecnologia *SQLite* permite o armazenamento de dados na própria aplicação, o que recebe a denominação de banco de dados embutido. Esta tecnologia é recomendada para aplicações de pequeno porte, que armazenam poucas informações.

## 2.3. Sistemas Operacionais

Os sistemas operacionais funcionam como estrutura que sustenta e administra todos os recursos do computador pelo *software* e *hardware*. São diversas as opções de sistemas operacionais disponíveis no mercado, cuja complexidade varia de acordo com a funcionalidade desejada, bem como o tipo de modelo do computador.

No que tange aos sistemas operacionais *mobile*, poucas são as opções, dentre as quais se destacam *Android* e *iOS*. Esses sistemas são desenvolvidos especificamente para executar operações em smartphones, tablets ou outros dispositivos móveis.

Os sistemas operacionais *mobile* deste projeto foram escolhidos diante da análise de vantagens ao desenvolver um aplicativo que possibilita a execução em *Android* e *IOS,* visto a facilidade de acesso a essas operações, além de serem considerados sistemas fáceis, práticos e mais utilizados no mundo, como podemos observar no gráfico abaixo:



**Gráfico 2 – Sistemas operacionais móveis utilizado no mundo todo entre maio de 2019 a maio de 2020**

**Fonte: (STATCOUNTER, 2020)**

## 2.3.1. *Android*

O sistema operacional *Android* foi desenvolvido em 2003 por Andy Rubin, Rich Miner, Nick Sears e Chris White e tinha como objetivo ser um sistema operacional para câmeras digitais. Vendo as possibilidades do mercado, a Android Inc. decidiu que o mercado de dispositivos móveis era mais interessante e migraram para este mercado.

Em 2005 a Google comprou o *Android* e dois anos depois o *Android* foi lançado por uma parceria entre a Google e algumas fabricantes de dispositivos móveis, tais como Samsung, Motorola e LG, com o objetivo de ser uma plataforma única para todos os celulares.

A primeira versão funcional do *Android* foi lançada em setembro de 2008 e desde então o sistema operacional não parou de crescer, transformando-se hoje no sistema operacional mais utilizado no mundo.

## 2.3.2. *iOS*

O *iOS*, sistema operacional exclusivo dos produtos Apple, foi lançado em 2007, como o sistema operacional do primeiro *iPhone*.

O sistema operacional móvel da Apple, o *iOS*, é utilizado em todos os dispositivos móveis da empresa, sendo o *iPhone*, *iPods* e *iPads* e é uma variante do *MacOS*.

O *iOS* foi criado para dispositivos *touchscreen*, e foi o primeiro sistema operacional a permitir a utilização dos dedos para gerir a interface.

# 3. modelagem

A modelagem deste projeto é o procedimento antecedente a prototipagem e desenvolvimento do aplicativo. Faz parte deste processo de modelagem a criação do *wireframe*.

## 3.1. *Wireframes*

O *wireframe* é a idealização visual do projeto, o esqueleto que visa exemplificar a usabilidade das páginas presente no aplicativo.

Para o aplicativo de *Flashcards*, foram desenvolvidas através da ferramenta *Pencil* todas as telas do aplicativo.

## 3.1.1. Página Inicial

A página inicial do aplicativo contra com três botões centralizados na vertical, com ícones e textos.

O primeiro botão direciona para a tela de adição de um novo *flashcard*, o segundo botão direciona para a tela onde é efetuada a listagem de todos os *flashcard* e o terceiro, e último, botão, direciona o usuário para uma tela contendo informações sobre o aplicativo.



**Figura 1: *Wireframe* da página inicial do aplicativo**

**Fonte: Própria (2020)**

## 3.1.2. Sobre o Aplicativo

A página sobre o aplicativo contém apenas texto com informações sobre o aplicativo. Informações como o projeto qual o aplicativo faz parte e o repositório online para quem puder contribuir com melhorias no aplicativo.



**Figura 2: *Wireframe* da página Sobre o aplicativo**

**Fonte: Própria (2020)**

## 3.1.3. Listagem dos flashcards

A página de listagem dos *flashcards* conta com uma lista de todos os *flashcards* existentes no sistema, e com a possibilidade de editar e excluir um *flashcard* específico. Nesta tela, também consta o *popup* de confirmação e de sucesso de exclusão, quando a exclusão de um *flashcard* é realizada.



**Figura 3: Tela que lista todos os *flashcards* no sistema**

**Fonte: Própria (2020)**

****

**Figura 4: *Popup* de exclusão de *flashcard***

**Fonte: Própria (2020)**

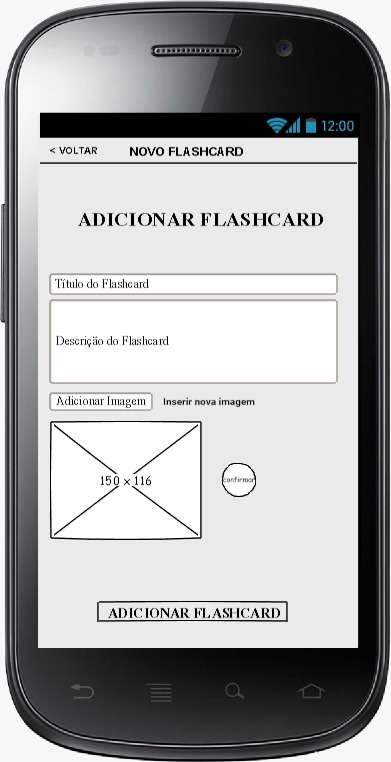
****

**Figura 5: Mensagem de sucesso de exclusão de *flashcard***

**Fonte: Própria (2020)**

## 3.1.4. Formulário de flashcard

O formulário de adição e edição de *flashcards* é a mesma tela, onde mudará somente as mensagens do sistema.



**Figura 6: Tela do formulário de adição e edição de *flashcard***

**Fonte: Própria (2020)**

****

**Figura 7: Mensagem de sucesso ao adicionar um *flashcard***

**Fonte: Própria (2020)**

## 3.1.5. Detalhes do flashcard

A página de detalhes do *flashcard* conta com todas as informações do *flashcard* selecionado.



**Figura 8: Tela de detalhes do *flashcard***

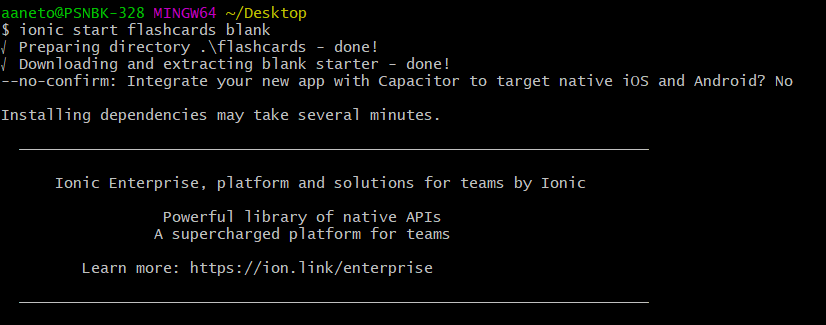
**Fonte: Própria (2020)**

# 4. prototipação

Neste capitulo serão abordados a codificação detalhada de cada página, tal como o layout final delas.

Como já descrito anteriormente, o aplicativo foi desenvolvido utilizando as tecnologias *Ionic, Angular* e *SQLite.*

Para iniciar o desenvolvimento, é necessário ter o *NodeJS* e o gerenciador de pacotes *NPM* instalados na máquina a ser utilizado no desenvolvimento da aplicação. Com as configurações iniciais já realizadas, é possível iniciar um aplicativo *Ionic* através de um comando a ser executado no *CMD* do sistema operacional.

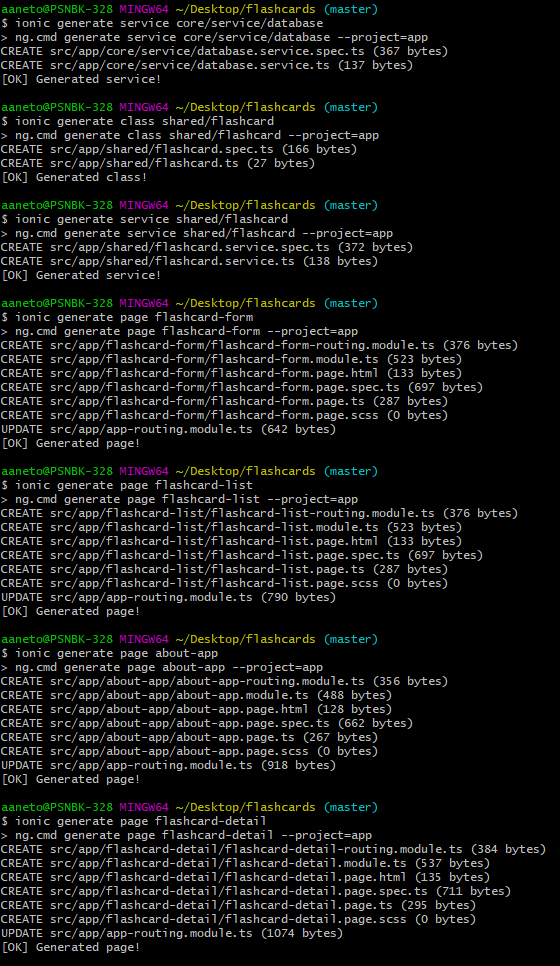


**Figura 9: Iniciando um aplicativo utilizando o Ionic**

**Fonte: Própria (2020)**

Quando iniciado o projeto desenvolvido, através de linhas de comandos também é possível criar todas as páginas, serviços e classes a serem utilizadas no decorrer do desenvolvimento da aplicação.

O projeto desenvolvido tem a classe denominada *Flashcard*, que contem as informações do *flashcard* a ser salvo, editado ou listado pela aplicação, o serviço de conexão com o banco de dados, que no caso é o *SQLite*, e as páginas inicial, listagem de *flashcards*, denominada *flashcards-list*, formulário de *flashcards*, denominada *flashcard-form*, detalhes do *flashcard*, denominada *flashcard-detail* e uma página sobre o aplicativo, denominada *about-app*.



**Figura 10: Criação dos serviços, classes e páginas necessárias para o projeto desenvolvido**

**Fonte: Própria (2020)**

## 4.1. Páginas do aplicativo

Abaixo serão detalhados o desenvolvimento de cada página do aplicativo.

## 4.1.1. Página Inicial

A página inicial é a página qual o usuário acessará antes de realizar qualquer ação no aplicativo. Nesta página está declarada os *links* para as demais páginas do aplicativo.



**Figura 11: Código *Ionic* da página inicial**

**Fonte: Própria (2020)**

Há dois atributos do Angular que se repetem nesta página, sendo eles o atributo *routerLink* e o atributo *routerDirection*.

O atributo *routerLink* indica para qual página o usuário será direcionado se clicar no elemento que o contém. Ou seja, se o usuário clicar na *div* onde o valor de *routerLink* é “/*flashcards*”, o usuário será direcionado para a página que lista todos os *flashcards*.

O atributo *routerDirection* indica a direção qual a animação de mudança de página terá. Ao atribuir o valor *forward* em *routerDirection* é indicado que a animação deverá ir para frente.

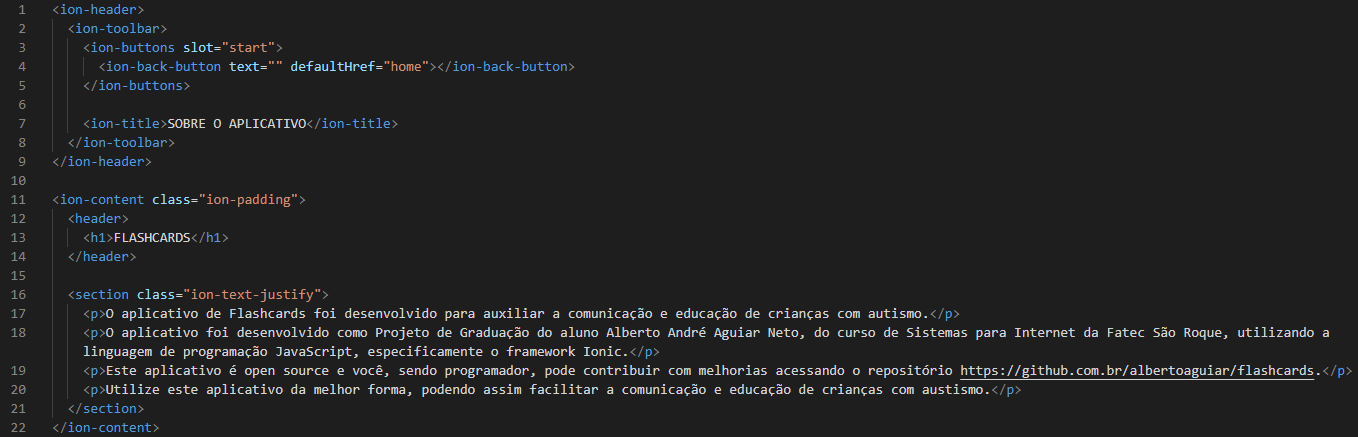


**Figura 12: Versão final da página inicial do aplicativo**

**Fonte: Própria (2020)**

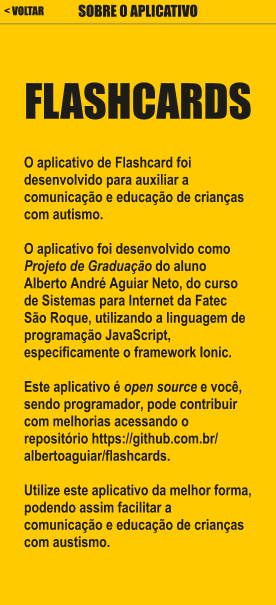
## 4.1.2. Sobre o Aplicativo

A página Sobre o Aplicativo conta apenas com uma diagramação estática realizada no *Ionic*, para somente promover informações sobre o aplicativo desenvolvido.



**Figura 13: Código *Ionic* da página Sobre o aplicativo**

**Fonte: Própria (2020)**



**Figura 14: Versão final da página Sobre o aplicativo**

**Fonte: Própria (2020)**

## 4.1.3. Todos os *Flashcards*

A página “todos os *flashcards*” conta com, além da codificação em *Ionic*, uma codificação lógica, que consiste em receber as informações de todos os *flashcards* que estão salvos no *SQLite* e mostrar a listagem na tela.

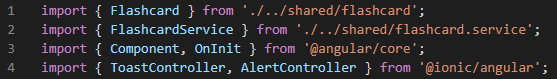


**Figura 15: Código *Ionic* da tela de listagem de *flashcards***

**Fonte: Própria (2020)**

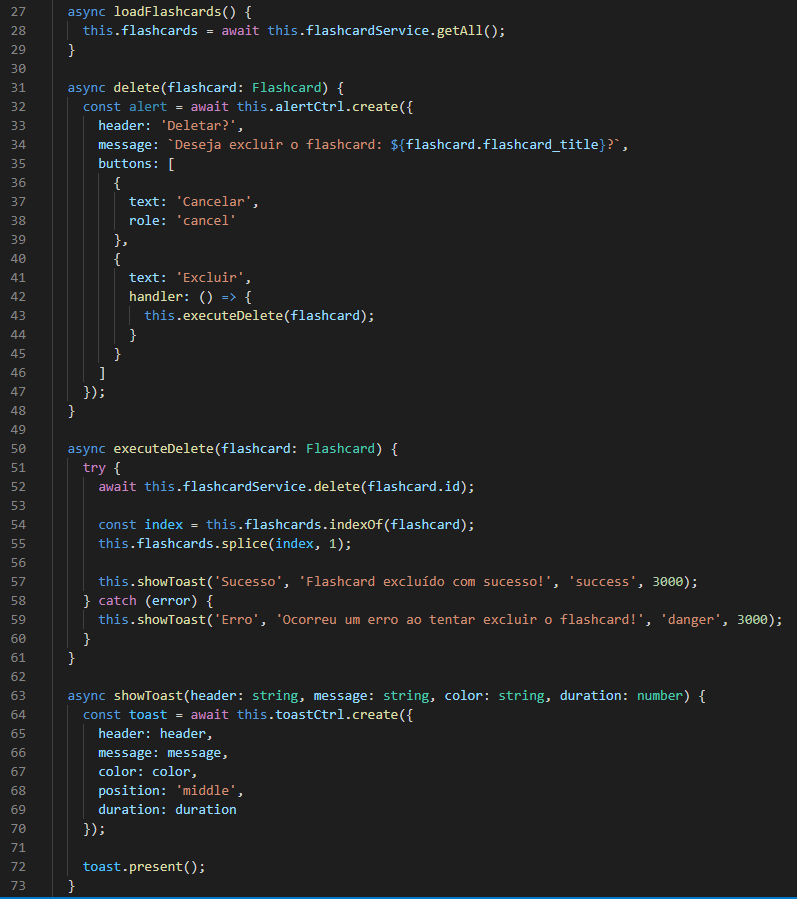
No código *Ionic* da tela de listagem de *flashcards* é utilizado novos atributos, sendo eles o *\*ngFor*, que realiza um *loop* no objeto *flashcards*, para encontrar e listar todos os *flashcards* disponíveis no banco. Além deste atributo, há o atributo *(click)*, que tem em seu valor o método delete, para deletar o *flashcard* selecionado e o atributo *[routerLink]* que tem como valor um objeto que contém primeiramente a rota de direcionamento, podendo ser a rota para editar o *flashcard* ou para visualizar os detalhes do mesmo, e o id do *flashcard* a ser editado.

No arquivo que conta com a parte lógica da página de todos os *flashcards*, constam os *imports* do *service* e da *class* de *flashcards*, para que seja possível efetuar as ações necessárias, como excluir, por exemplo.



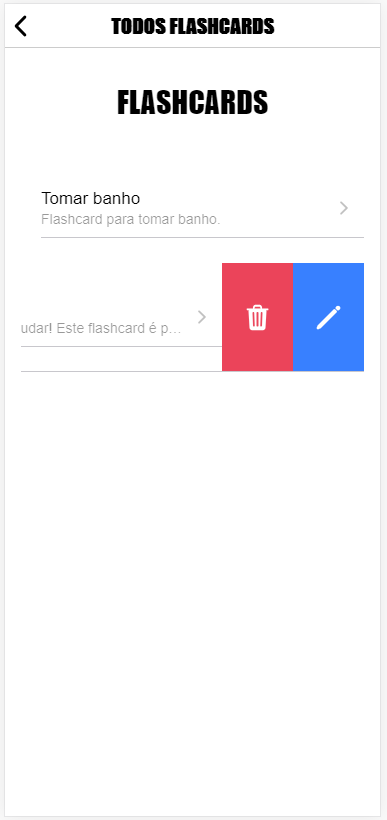
**Figura 16: *Imports* do arquivo *angular* da página de listagem de todos os *flashcards***

**Fonte: Própria (2020)**



**Figura 17: Métodos de listagem de todos os *flashcards* e exclusão de algum *flashcard* específico**

**Fonte: Própria (2020).**



**Figura 18: Versão final da página que lista todos os *flashcards***

**Fonte: Própria (2020)**

## 4.1.4. Formulário de flashcard

O formulário do *flashcard* será o mesmo para edição e para inclusão. O aplicativo identifica se há o parâmetro id passado na *URL*, caso haja, é edição de um *flashcard* já existente. Caso este parâmetro não seja encontrado, é adição de um novo *flashcard*.

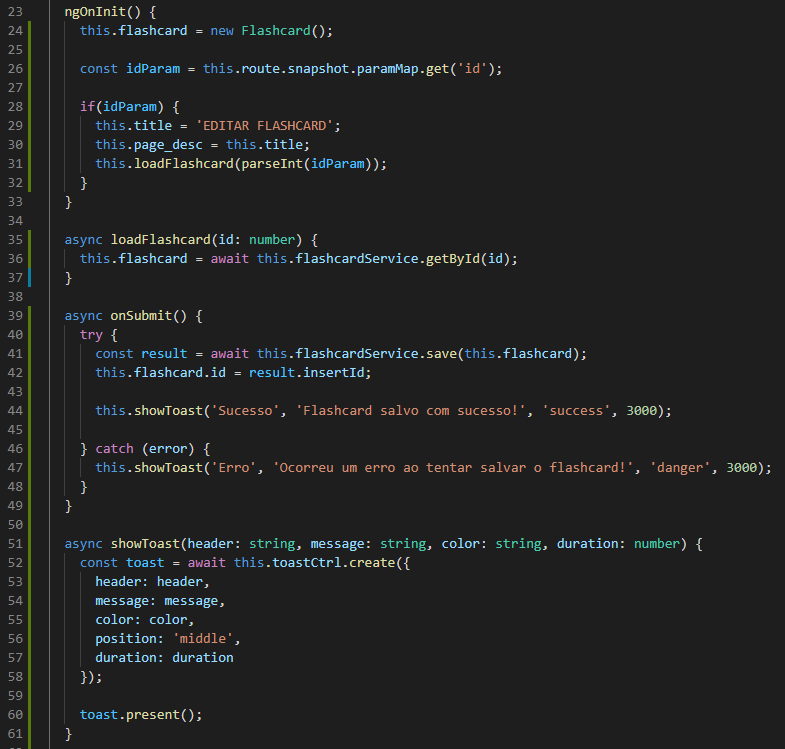


**Figura 19: Código *Ionic* do formulário de *flashcard***

**Fonte: Próprio (2020)**

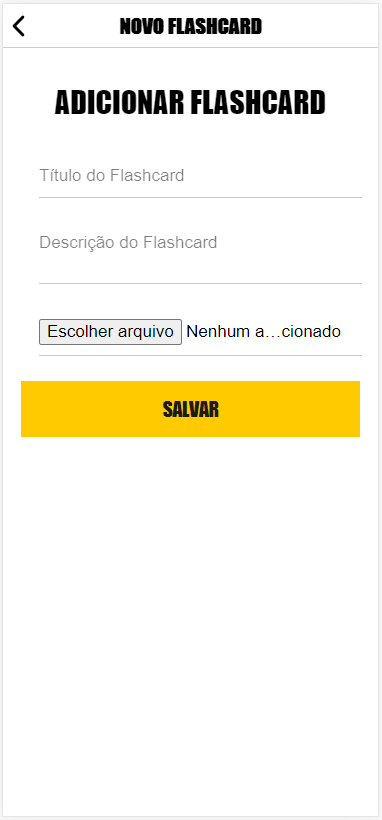
Há uma singularidade no código *ionic* da página do formulário. Nesta página, fez-se necessário indicar valores ou atributo *[(ngModel)]*, isto para que na inserção e na edição, o objeto de *flashcard* receba os dados corretamente.

Tal como na página de listagem, é necessário efetuar a importação do serviço e da classe de *flashcard* no arquivo lógico desta página, para que assim seja possível a realização da criação ou edição dos dados do *flashcard*.



**Figura 20: Código *Angular* da página de formulário, onde é realizada a identificação se é edição ou adição de *flashcards* e onde é realizado o envio do *flashcard* para o banco de dados**

**Fonte: Própria (2020)**



**Figura 21: Versão final do formulário de adição e edição de *flashcard***

**Fonte: Própria (2020)**

## 4.1.5. Detalhes do flashcard

A página de detalhes do *flashcard* conta com todas as informações do *flashcard* selecionado.

O arquivo *ionic* da página de detalhes é extremamente simples, já que essa página tem única e exclusivamente como objetivo mostrar dados recuperados anteriormente.



**Figura 22: Código *ionic* da página de detalhes do *flashcard***

**Fonte: Própria (2020)**



**Figura 23: Código *Angular* da página de detalhes do *flashcard***

**Fonte: Própria (2020)**



**Figura 24: Versão final da página de detalhes do *flashcard***

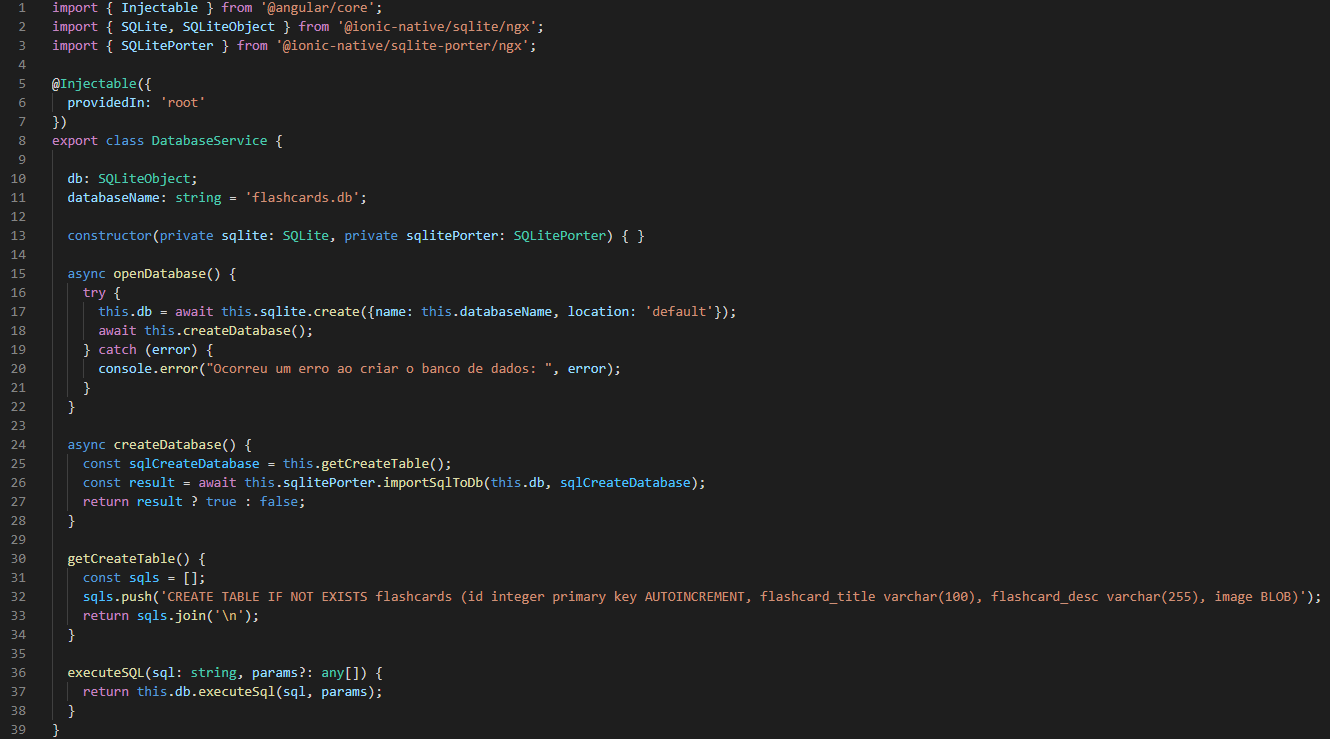
**Fonte: Própria (2020)**

## 4.2. Serviços do aplicativo

Abaixo serão detalhados o desenvolvimento dos *services* desenvolvidos para este aplicativo.

## 4.2.1. Serviço de banco de dados

O serviço de banco de dados controla todas as ações a serem realizadas no *SQLite*. Este serviço é responsável por criar o banco de dados, quando o mesmo é inexistente, criar a única tabela presente no banco de dados deste projeto, tabela esta denominada *flashcards*, quando a mesma também é inexistente, e executar todas as queries, através do método criado com o nome de *executeSQL.*



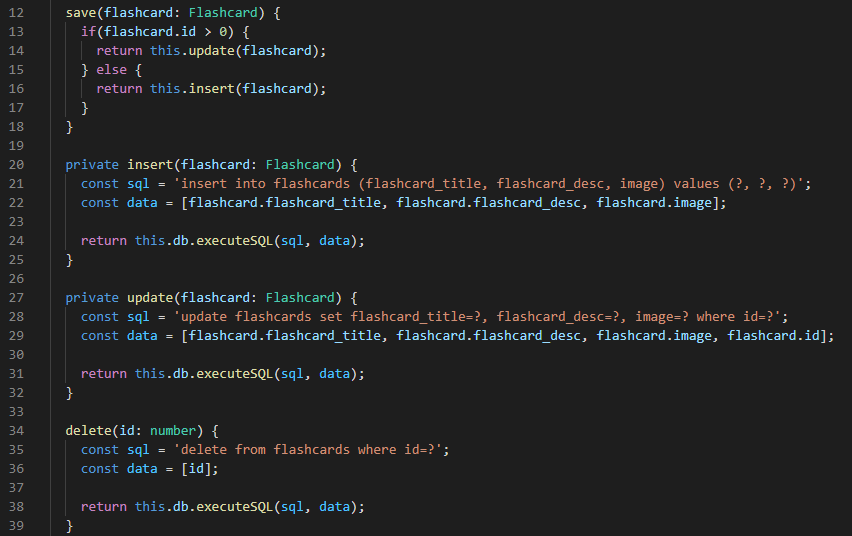
**Figura 25: Código do serviço de banco de dados, responsável por realizar todas as atividades do banco de dados da aplicação**

**Fonte: Própria (2020)**

## 4.2.2. Serviço de Flashcard

O serviço de *Flashcard* é responsável por gerenciar as ações envolvendo *flashcards*. Este serviço, por exemplo, envia e recebe informações do serviço de banco de dados.

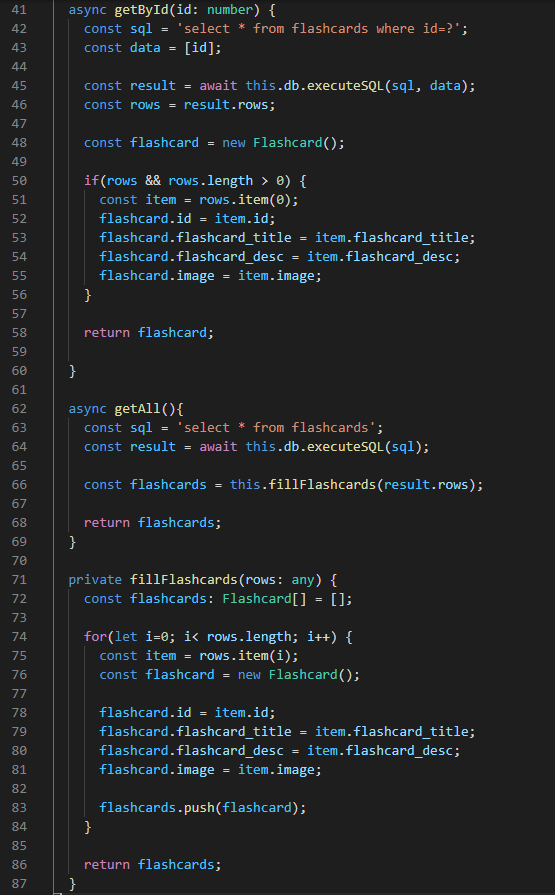
Este serviço conta com um método de inserção e atualização de *flashcards*, que são métodos privados e podem ser utilizados através do método público *save,* e também o método delete, que envia a execução para excluir algum *flashcard* do banco de dados.



**Figura 26: Métodos de inserção, atualização e exclusão do serviço de *flashcard***

**Fonte: Própria (2020)**

Além dos métodos acima citado, o serviço de *flashcard* também conta com métodos de listagem de *flashcard*, sendo uma listagem individual, onde a busca é realizada por id, ou buscando todos os *flashcards* existentes no banco de dados.



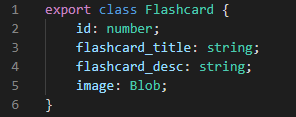
**Figura 27: Métodos de listagens do serviço de *flashcards***

**Fonte: Própria (2020)**

## 4.3. Classes do aplicativo

Como o aplicativo conta com apenas uma tabela no banco de dados, ele também conta com apenas uma classe para o gerenciamento desta tabela.

Esta classe é denominada *Flashcard* e existe apenas para que seja possível controlar os tipos das informações a serem salvas no banco de dados.



**Figura 28: Classe de *flashcard*, para controlar o tipo de dados do objeto**

**Fonte: Própria (2020)**

# 5. CONSIDERAÇÕES FINAIS

O aplicativo desenvolvido tem capacidade de alcançar o objetivo proposto, que é auxiliar na educação de crianças autistas, auxiliando os educandos na comunicação com os mesmos.

O desenvolvimento utilizando uma linguagem híbrida demandou menos tempo e conhecimento técnico, levando em consideração que para desenvolver para dois sistemas operacionais, como feito no decorrer deste projeto, de forma nativa, o programador há de ser especialista em, no mínimo, duas linguagens de programação, enquanto utilizando o *Ionic,* o conhecimento em *JavaScript* e nas singularidades do *framework* já se faz mais do que necessário.

Ainda assim, houveram diversas dificuldades no desenvolvimento da aplicação, principalmente no *upload* da imagem do celular do usuário para o banco de dados utilizado.

A perspectiva deste projeto é melhorá-lo, assim corrigindo potenciais erros de desenvolvimento, e posterior a isto, adicionar novas funcionalidades, tais como criação de rotinas, onde é possível gerenciar *flashcards* de mais de um educando.

# 6. REFERÊNCIAS

LOPES, Sérgio. A Web Mobile: Programe para um mundo de muitos dispositivos. São Paulo: Casa do Código, 2013.

CIRIACO, Douglas. Android cresce, iOS diminui e Windows Phone quase some, aponta relatório. 2017. Disponível em: <https://www.tecmundo.com.br/dispositivos-moveis/119411-android-ios-windows-phone-mercado.htm>. Acesso em: 16 de mai. 2020.

DIGITAL, Convergência. Celular é o principal meio de acesso à Internet nas classes D e E. 2017. Disponível em: <http://www.convergenciadigital.com.br/cgi/cgilua.exe/sys/start.htm?UserActiveTemplate=site&infoid=46427&sid=14>. Acesso em: 16 de mai. 2020.

WIKIDOT. Ionic – Pesquisas de compiladores. 2018. Disponível em: <http://pesquompile.wikidot.com/ionic>. Acesso em: 16 de mai. 2020.

IONICONF. About Ionic Cross - Plataform Mobile Development Technologies. Disponível em: <https://ionicframework.com/about>. Acesso em: 20 de mai. 2020.

IONICONF. Ionic for Enterprise - Why Ionic. Disponível em: <https://ionicframework.com/enterprise/why-ionic>. Acesso em: 20 de mai. 2020.

TREINAWEB, blog. O que é Ionic?. 2019. Disponível em: <https://www.treinaweb.com.br/blog/o-que-e-ionic/>. Acesso em: 20 de mai. 2020.

OLIVEIRA, Darcius. A Evolução do Angular. 2018. Disponível em: <http://darcius.com.br/?p=21>. Acesso em: 22 de mai. 2020.

ANGULAR. ANGULAR. Disponível em: <https://angular.io/>. Acesso em: 22 de mai. 2020.

BLOG, Geek. Angular 8: Quais são as novidades da versão?. 2019. Disponível em: <https://blog.geekhunter.com.br/angular-8-novidades-da-versao/>. Acesso em: 22 de mai. 2020.

GSTI, Portal. O que é SQLite?. Disponível em: <https://www.portalgsti.com.br/sqlite/sobre/>. Acesso em: 27 de mai. 2020.

DOCS, Microsoft. Use a SQLite database in UWP app – UWP applications. Disponível em: <https://docs.microsoft.com/en-us/windows/uwp/data-access/sqlite-databases?redirectedfrom=MSDN>. Acesso em: 27 de mai. 2020.

NET, Oficina da. A história do iOS. 2018. Disponível em: <https://www.oficinadanet.com.br/post/17950-a-historia-do-ios>. Acesso em: 27 de mai. 2020.

TECMUNDO. A história do Android, o robô que domina o mercado mobile. 2017. Disponível em: <https://www.tecmundo.com.br/ciencia/120933-historia-android-robo-domina-o-mercado-mobile-video.htm>. Acesso em: 28 de mai. 2020.

INTEGRAR, Autismo Projeto. O que são desenhos roteirizados. 2019. Disponível em: <http://autismoprojetointegrar.com.br/o-que-sao-desenhos-roteirizados/>. Acesso em: 28 de mai. 2020.

AUTISMO, Inspirados pelo. Aplicativos para pessoas com autismo. Disponível em: <https://www.inspiradospeloautismo.com.br/aplicativos-para-pessoas-com-autismo/>. Acesso em: 01 de jun. 2020.

INTEGRAL, Centro de Referências em Educação. Autismo e escola: os desafios e a necessidade de inclusão. 2014. Disponível em: <https://educacaointegral.org.br/reportagens/autismo-escola-os-desafios-necessidade-da-inclusao/>. Acesso em: 01 de jun. 2020.

GLOBAL, GFC. Informática Básica: o que é um sistema operacional?. Disponível em: <https://edu.gcfglobal.org/pt/informatica-basica/o-que-e-um-sistema-operacional/1/>. Acesso em: 06 de jun. 2020.