

CENTRO UNIVERSITÁRIO CARIOCA - UNICARIOCA
GILVAN PRAXEDES DE ALMEIDA

**GOODPARENTS: UM APLICATIVO MÓVEL PARA ESTABELECEER
RELAÇÕES ENTRE PADRINHOS E SEUS APADRINHADOS NA ESCOLA SESC.**

RIO DE JANEIRO
2015

GILVAN PRAXEDES DE ALMEIDA

**GOODPARENTS: UM APLICATIVO MÓVEL PARA ESTABELECER RELAÇÕES
ENTRE PADRINHOS E SEUS APADRINHADOS NA ESCOLA SESC.**

Trabalho de Conclusão de Curso
apresentado ao Centro Universitário
Carioca, como requisito parcial para
conclusão do curso de ciência da
computação.

Orientador : Prof. Jorge Viana Doria Junior, M.Sc

GILVAN PRAXEDES DE ALMEIDA

**GOODPARENTS: UM APLICATIVO MÓVEL PARA ESTABELECEER RELAÇÕES
ENTRE PADRINHOS E SEUS APADRINHADOS NA ESCOLA SESC.**

Trabalho de Conclusão de Curso
apresentado ao Centro Universitário
Carioca, como requisito parcial para
conclusão do curso de ciência da
computação.

Banca Examinadora

Prof. Jorge Viana Doria Junior, M.Sc - Orientador

Centro Universitário Carioca

Prof. Luiz Avelino Sobral, M.Sc - Coordenador do curso

Centro Universitário Carioca

Prof. Alessandro Jatobá, D.Sc

Centro Universitário Carioca

A minha esposa, familiares, professores
e amigos que me apoiaram ao longo
desta e de outras caminhadas.

AGRADECIMENTOS

À Deus por esta oportunidade, bênçãos diárias, sabedoria, força e paciência até a finalização desse projeto.

À minha mãe, Luciana Praxedes, pelo seu amor infinito, por me ensinar e incentivar a acreditar nos meus sonhos, essa vitória é nossa.

Ao meu pai, Ivan Almeida (*in memorian*), por toda sua dedicação comigo, seus ensinamentos e seu esforço para que hoje eu pudesse estar concluindo este curso. Estará sempre em meu coração.

À minha irmã Juliana, pelo seu apoio e paciência.

À minha esposa, Lidia Almeida, pelo seu companheirismo, paciência nos momentos de dificuldades, por suportar minha ausência em diversos momentos e o apoio a para vencer esta etapa acadêmica.

À todos os professores que contribuiriam diariamente com seu conhecimento e dedicação e foram de suma importância na minha jornada acadêmica. Em especial ao professor Jorge Viana Doria Junior, pela orientação deste trabalho, estando sempre disponível e apto a ensinar com prazer.

RESUMO

Um grande obstáculo notado na escola SESC residente, influenciando no desenvolvimento integral dos novos alunos e até mesmo em alguns casos resulta em evasão dos mesmos, é a receptividade e o vínculo padrinho/apadrinhado. A escola possui um programa de apadrinhamento no qual os alunos mais antigos se dispõem a ajudar na mudança física e familiar dos novos alunos. Porém, nem sempre os apadrinhados ficam satisfeitos com seus padrinhos, e o principal motivo é o abandono por falta de afinidades. Portanto, este trabalho trata sobre o desenvolvimento de um aplicativo para smartphones Android e IOS, o good parents. Para sanar este problema, o aplicativo tem como objetivo auxiliar o padrinho a fazer as melhores escolhas e alertar a importância do apadrinhamento, ajudando na ambientação, integração, socialização e afetividade dos novos alunos por meio de afinidades e áreas de interesse. Utilizando regras de associação, podemos relacionar através de seus atributos e gerando pesos sobre os mesmos descobrindo gostos em geral, dificuldades e proximidades geográficas. Além disso, será detalhado o funcionamento do aplicativo, as decisões de design adotadas, juntamente com os casos de uso. A ferramenta principal a ser explorada será o ionic, um framework destinado a criação de aplicações híbridas para dispositivos móveis. Também será ponderada possibilidades de melhorias na aplicação.

Palavras-chave : Desenvolvimento de software, smartphone, tecnologia, educação

ABSTRACT

The big obstacle observed at high school resident SESC, influencing in full development of the new students and the same cases result in evasion them, is the receptivity and the bond godparent / patronized. The high school has a program of godparents in which the older students help the physical and familiar change for the younger students. However, almost always the patronized are satisfied with your godparents, the main goal it's abandon for lack of affinity. However, this work is about a software development mobile to platform Android and IOS called good parents. To solve this problem the software has as goal auxiliar the godparent to make choice better and alerting their own importance, helping the transition, integration, socialization, and affectivity for the younger students by affinity and area of interest. Using association rules, we can connect across of your attributes and generating weights on the same discovering tastes in general, difficulties and geographical proximity. Moreover, will detail the functioning of software, the decision of design and uses cases. The main tool to be explored will be the ionic, a framework made to build hybrid applications to mobile devices. In addition, will be considered possibilities for improvement in the application.

Keywords: Software Development, Smartphone, Technology, Education.

LISTA DE ILUSTRAÇÕES

Figura 1: Simon Personal Communicator	17
Figura 2: HTC Dream, o primeiro a rodar o Android comercial do mercado	18
Figura 3: Protótipo do primeiro iPhone	19
Figura 4: Tecnologias relacionadas ao Ionic	20
Figura 5: Treinando o modelo	21
Figura 6: Modelo treinado	22
Figura 7: Subáreas da Inteligência Artificial	23
Figura 8: Dados estruturados em formato json inserido no banco de dados	29
Figura 9: Casos de uso	31
Figura 10: Diagrama de classes do protótipo	37
Figura 11: Diagrama de classes do servidor	38
Figura 12: Página de login do protótipo	39
Figura 13: Página de cadastro do protótipo	40
Figura 14: Página Home do protótipo	41
Figura 15: Página de detalhes do apadrinhado	42
Figura 16: Página de perfil do padrinho logado	43
Figura 17: Tela de login do app Tinder	45
Figura 18: Tela de sugestão de amizades do app Facebook	46
Figura 19: Tela de Meetup	47
Figura 20: Tela de download do NodeJs	54
Figura 21: Tela de download do editor de texto VS Code	56

LISTA DE TABELAS

Tabela 1: Caso de Uso Efetuar Login	32
Tabela 2: Caso de Uso Cadastrar Aluno	33
Tabela 3: Caso de Uso Exibir Home	34
Tabela 4: Caso de Uso Manter Perfil	35
Tabela 5: Caso de Uso Alterar Cadastro	36

SUMÁRIO

1- Introdução	13
1.1 - Contextualização: Como é a escola residência	13
1.2 - Motivação: O programa de apadrinhamento	14
1.3 - Objetivos	14
1.4 - Organização do trabalho	15
2 - Fundamentação teórica	16
2.1 - Sobre os Dispositivos Móveis	16
2.2 - A História dos <i>Smartphones</i>	16
2.3 - Adoção dos Sistemas Google Android e iOS	18
2.4 - Aplicativos Multiplataforma	20
2.5 - Sobre os Sistemas de Recomendação	21
2.6 - Sobre as Tecnologias Escolhidas	24
2.7 - Principais Benefícios no Uso das Tecnologias Escolhidas	25
3 - O Aplicativo GoodParents	26
3.1 - Identificação do Problema	26
3.2 - Estado Atual do Mercado	26
3.3 - Requisitos do Sistema	27
3.3.1 - Requisitos Funcionais	27
3.3.2 - Requisitos não Funcionais	27
3.4 - Características do Sistema	28
3.5 - Modelagem e arquitetura do sistema	28
3.5.1 - Diagrama de Caso de Uso	29
3.5.1.1 - Efetuar Login	29
3.5.1.2 - Cadastrar Aluno	30
3.5.1.3 - Exibir Home	30
3.5.1.5 - Manter Perfil	30
3.5.1.6 - Alterar Cadastro	30
3.5.2 - Detalhamento dos Casos de Uso	32
3.5.2.1 - Efetuar Login	32
3.5.2.2 - Cadastrar Aluno	33
3.5.2.3 - Exibir Home	34
3.5.2.4 - Manter Perfil	35
3.5.2.5 - Alterar Cadastro	36
3.5.3 - Diagrama de Classes Conceitual do Sistema	37
3.5.4 - Protótipo do Sistema GoodParents	39
4 - Análise Crítica	44
4.1- Aplicativos Avaliados	44
5.1.1- Tinder	45
5.1.2 - Facebook	46
5.1.3 - Meetup	47
4.2- Quadro Comparativo	48
	10

4.3- Conclusão da Análise Comparativa	49
5 - Conclusão	50
5.1 Considerações finais	50
5.2 Possibilidade de melhorias futuras	50
5.3 Limitações encontradas no decorrer do trabalho	51
Referências	52
Apêndice - Preparando o ambiente de desenvolvimento	54
Apêndice - Instalando o Node.js	54
Apêndice - Utilizando o MongoDB	55
Apêndice - Escolhendo o editor de texto	56
Apêndice - Disponibilizando o servidor back-end	57
Apêndice - Instalando o Ionic	58

1- Introdução

Nesse capítulo falaremos sobre os objetivos deste trabalho, assim como a contextualização do problema.

1.1 - Contextualização: Como é a escola residência

A Escola SESC de Ensino Médio - ESEM, localizada no bairro de Jacarepaguá, Rio de Janeiro, é a única escola residência 100% gratuita de todo o estado. Inaugurada em 2008, com o objetivo de oferecer uma educação integral e receber adolescentes de todos os estados brasileiros.

A escola está entre as 50 melhores de todo o estado e chega a comportar até 500 alunos, com uma das mais privilegiadas estruturas de ensino médio do país possuindo um campus de 131 metros quadrados contendo espaço cultural, laboratórios, biblioteca, ateliers de arte, complexo desportivo, piscina semiolímpica, sala de ginástica e musculação, restaurante, além de vilas residenciais dos estudantes e professores.

Além da moradia oferecida pela escola, é oferecido também todo um suporte pedagógico aos alunos, incluindo *notebook*, uniformes, alimentação, assistência médica e odontológica. “Os alunos têm ainda a possibilidade de se tornar proficiente em inglês e podem ainda aprender Espanhol, Francês e Alemão”, segundo a coordenadora da ESEM.

1.2 - Motivação: O programa de apadrinhamento

A ESEM possui um programa de apadrinhamento que consiste em os alunos do segundo ou terceiro grau, se oferecem à ajudar aos alunos novatos, ou seja, que estão ingressando no primeiro ano/grau da escola em sua receptividade e tão importante quanto, é o vínculo de padrinho e apadrinhado, ou seja, afinidades entre eles e deficiência e a facilidades em determinadas disciplinas.

De acordo com a mestrandia Érica de Lara, aluna do Centro Universitário Carioca em seu projeto de qualificação, o desenvolvimento do aluno residente não deve se limitar somente aos aspectos cognitivos de aprendizagem tradicional. Após ter sido feita uma coleta de dados por meio de um roteiro de entrevistas semiestruturada, durante cinco meses em locais formais e não formais de aprendizagem do campus, os alunos de primeiro grau se queixaram da receptividade e abandono do padrinhos resultando em alguns casos na evasão dos mesmos.

Portando, o modo de interação e a maneira de inclusão destes novos membros em uma comunidade já existente é de suma importância, pois o padrinho não é somente mais uma possível amizade a ser feita neste ambiente, é uma referencia e auxilio na mudança física e familiar dos calouros. E ao longo deste trabalho abordaremos formas para mitigar esse problema de afinidade e afetividade.

1.3 - Objetivos

O objetivo deste trabalho é implementar um aplicativo de recomendação de apadrinhados para os padrinhos. Usando regras de associação e sobre determinados atributos como, região, gostos musicais, gênero, esportes preferidos e deficiência ou facilidade em algumas disciplinas podemos sugerir possíveis apadrinhas que se assemelham as suas características.

E também, ser o primeiro experimento de sucesso a utilizar o projeto de trabalho final de graduação para integrar-se, interagir e se aproximar com o programa de mestrado do Centro Universitário Carioca.

1.4 - Organização do trabalho

Neste primeiro capítulo fala-se sobre o estado do programa de apadrinhamento e suas queixas relatadas pelos alunos, e é demonstrado o objetivo do meu trabalho.

No segundo capítulo é apresentada a fundamentação teórica deste trabalho.

No terceiro capítulo o aplicativo em si é apresentado, com seus requisitos e opções de design. Mostramos diagramas UML, de modo a detalhar o desenvolvimento.

No quarto capítulo é feita uma análise crítica comparativa do aplicativo com relação a outros aplicativos similares de recomendação existentes, mostrando pontos fortes e pontos fracos.

No quinto capítulo fala-se sobre as conclusões alcançadas com a elaboração do trabalho, pontos a avançar, e possibilidade de novas atualizações e melhorias.

Enfim no sexto capítulo mostra-se a bibliografia pesquisada.

2 - Fundamentação teórica

2.1 - Sobre os Dispositivos Móveis

Dispositivos moveis (*mobile device or handheld computer*, do inglês) começaram a ser produzidos na década de 1990, desde então não pararam de crescer em nossa sociedade. Engana-se quem pensa que só adolescentes, adultos e pessoas de meia idade se aventuram utilizando esses aparelhos, muitas crianças já sabem manusear um *smartphone* hoje em dia, e os idosos demonstram cada dia mais interesse por essa tecnologia.

Um dispositivo móvel nada mais é que um computador de bolso relativamente pequeno capazes até de serem operados com uma das mãos em qualquer lugar, possuindo sistemas operacionais para executar aplicativos/programas para facilitar as pessoas a se conectarem seja por meio da Internet ou não em qualquer local do globo. Segundo a ABRANET (Associação Brasileira de Internet), o Brasil tem 280 milhões de dispositivos móveis conectados a Internet, dentre eles estão, *notebook*, *netbook*, *laptop*, *smartwatch*, *tablets* e *smartphone*.

2.2 - A História dos Smartphones

Em meados da década de 90, com a chegada da terceira geração (3G) os celulares ganharam novas tecnologias e novos conceitos, que visaram unir o conforto e a praticidade para satisfazer o usuário. (BATISTA, 2011).

O *smartphone* passou a ser um item indispensável na vida de muita gente, comparado até mesmo a uma peça de roupa diária, talvez ele seja o primeiro item que pegue ao acordar. Algumas pessoas conseguem resolver todas as suas tarefas diárias de trabalho apenas com um *smartphone*.

O primeiro celular considerado *smartphone* foi lançado oficialmente em 16 de agosto de 1994, pela IBM, empresa que dominava o mercado na época, seu primeiro modelo chamava-se Simon Personal Communicator. Apesar de não ser reconhecido como um

smartphone na época.

O Simon chegou a pesar 500g, chegando ao mercado por US\$ 899, possuía funcionalidades similares aos modelos atuais, como bloco de notas digital, agenda de compromissos, calendários, envio de emails, dentre outras. Mas sua bateria limitada, o que não é muito diferente dos modelos atuais o fez com que ele saísse do mercado dois anos depois. Neste ano ele completa 24 anos de idade.



Figura 1: Simon Personal Communicator

Fonte: Wikipedia¹

¹Disponível em <https://en.wikipedia.org/wiki/IBM_Simon#/media/File:IBM_Simon_Personal_Communicator.png>

¹Disponível em <https://en.wikipedia.org/wiki/IBM_Simon>

2.3 - Adoção dos Sistemas Google Android e iOS

Como consequência ao aparecimento dos *smartphones*, surgiu os sistemas operacionais móveis desenvolvidos para explorarem ao máximo seu potencial. Os dois maiores sistemas operacionais utilizado no mundo é o Android da Google e o iOS da Apple. Lançado oficialmente em 2008, o Android é um sistema *open source* baseado no kernel Linux para o controle das principais tarefas do sistema. Além de ter seu código aberto, o Android ganhou muitos fãs por ser altamente customizável e terem variedade de escolha nos dispositivos fazendo do mesmo o sistema mais utilizado no mundo.



Figura 2: HTC Dream, o primeiro a rodar o Android comercial do mercado

Fonte: Oficina da net²

- *Kernel*: software responsável por controlar as interações entre o hardware e outros programas do dispositivo. O *kernel* traduz as informações que recebe ao processador e aos demais recursos do sistema.

²Disponível em <<https://www.oficinadanet.com.br/imagens/post/13939/1.jpg>>

³Disponível em <<https://www.oficinadanet.com.br/post/13939-a-historia-do-android>>

Já iOS, conhecido como um dos melhores sistemas operacionais do mundo, é a abreviatura para *iPhone Operation System* sendo desenvolvido pela Apple. O sistema foi baseado no Sistema Operacional MAC OS X e projetado para atender as necessidades de aparelhos móveis desenvolvidos pela marca. Lançado originalmente em 2007, tendo seus concorrentes na época os aparelhos Windows Mobile, Palm OS, Symbian e BlackBerry, seus usuários se orgulham por sua estabilidade, segurança, suporte e confiança avaliando-o como o mais protegido do mundo.



Figura 3: Protótipo do primeiro iPhone

Fonte: Olhar digital ³

³Disponível em <https://sonnydickson.com/wp-content/uploads/2017/01/IMG_7004-1-1100x919.jpg>

³Disponível em <<https://olhardigital.com.br/noticia/video-mostra-prototipos-do-primeiro-iphone-rodando-o-ios/65267>>

2.4 - Aplicativos Multiplataforma

Aplicativos multiplataforma ou híbridos também chamados de *cross-platform*, são aplicativos desenvolvidos a partir de um *framework* compatível com a maioria dos sistemas operacionais disponíveis no mercado, tais como o Android, iOS, Windows Phone, BlackBerry e Symbian.

Dentre as opções de *framework* gratuitos mais utilizadas atualmente no mercado para aplicativos híbridos, temos o ionic, PhoneGap, Sencha e o Xamarim.

A adoção de desenvolver aplicativos multiplataforma hoje em dia tem se tornado bastante comum do que desenvolver um aplicativo nativo, ou seja, focado diretamente em extrair ao máximo características de cada sistema operacional por meio de suas API's (*Application Programming Interface*) diretas e nativa de cada *smartphone*. Os principais ganhos em um projeto quando se cria um aplicativo multiplataforma e a reutilização do código fonte para diversas plataformas *mobile*, diminuição no tempo de desenvolvimento, pois o mesmo está sendo desenvolvido para diversas plataformas ao mesmo tempo, ausência de equipes específica para sistemas diferentes e como consequência o custo será mais baixo.



Figura 4: Tecnologias relacionadas ao Ionic

Fonte: Devmedia⁴

Application Programming Interface: é o conjunto de padrões de programação que permite a construção de aplicativos e a sua utilização de maneira não tão clara para os usuários.

¹Disponível em <<https://arquivo.devmedia.com.br/artigos/DevMedia/ionic.png>>

⁴Disponível em <<https://www.devmedia.com.br/guia/ionic/38372>>

2.5 - Sobre os Sistemas de Recomendação

Hoje em dia podemos notar os sistemas de recomendação em qualquer lugar da Internet, seja no facebook com indicações de amizades, spotify com sugestões de tipos musicais, *ecommerce* com ofertas de produtos pesquisados recebendo muita das vezes por emails e até mesmo sites de relacionamentos como o tinder por localização.

Estes sistemas utilizam mecanismos de algoritmos triviais integrados a aplicação, inferindo dados captados com objetivo de oferecer qualidade diferenciada das demais aplicações disponíveis no mercado e satisfazer a expectativa e interesses do usuário. Como esse sistema trabalha em cima de dados obtidos, seja pelo próprio usuário ou pelo histórico de navegação, para se obter uma melhor performance, quando mais dados colhido, melhor será o resultado.

Antes de começarmos a falar dos tipos de sistemas de recomendação, precisamos entender o principal campo em que ele se enquadra na ciência da computação. A área de aprendizado de máquina (*machine learning*).

Machine learning é uma ramo de inteligência artificial que utiliza método de análise de dados que automatiza a construção de modelos analíticos baseado na ideia de que sistemas podem aprender de dados obtidos, identificar padrões e tomar decisão sem o mínimo de intervenção humana.

Observe o exemplo a seguir:

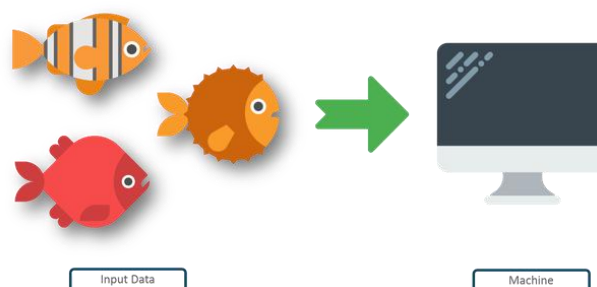


Figura 5: Treinando o modelo

Fonte: Quora⁵

⁵Disponível em <<https://www.quora.com/What-is-machine-learning-4>>

Uma vez que a máquina aprendeu todas as características associadas a um peixe, será testado o seu resultado com a entrada de um novo dado.

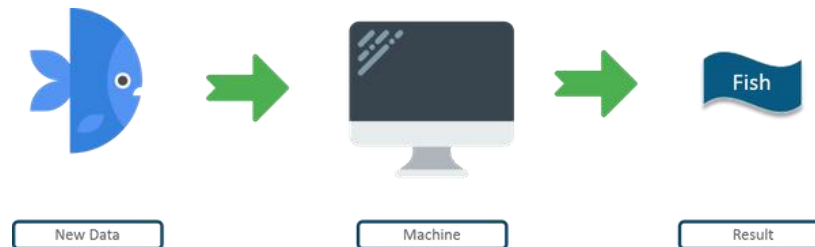


Figura 6: Modelo treinado

Fonte: Quora⁶

Portanto, sistemas de recomendação tem o objetivo de recomendar algo ao usuário, fazendo previsões de dados obtidos de *cookies* do *browser*; histórico de navegação, compras passadas, localidade geográfica ou de forma mais direta solicitando ao usuário *feedbacks*. Essa técnica é abusivamente utilizada pela área de *marketing* hoje e dia.

Dependendo de como a recomendação é feita, os sistemas de recomendação são classificados em três tipos.

Filtragem colaborativa, é quando a recomendação dos itens vem de gostos parecidos com outras pessoas mediante ao um histórico. Parte-se do princípio de que se um usuário gostou do item A e B o outro usuário também gostará. A desvantagem deste método é que demanda bastante informação sobre os usuários.

Baseada em conteúdo, é quando o usuário demonstrou interesse por um tipo de produto, gosto musical ou gênero filmico como a netflix. A vantagem desse método é que não demanda um grande número de informações. Por outro lado, as recomendações podem ser repetitivas.

E finalmente, o sistema híbrido que compõe-se em unir os dois métodos mencionados acima, com o objetivo de extrair o melhor de ambos.

⁶Disponível em <<https://www.quora.com/What-is-machine-learning-4>>

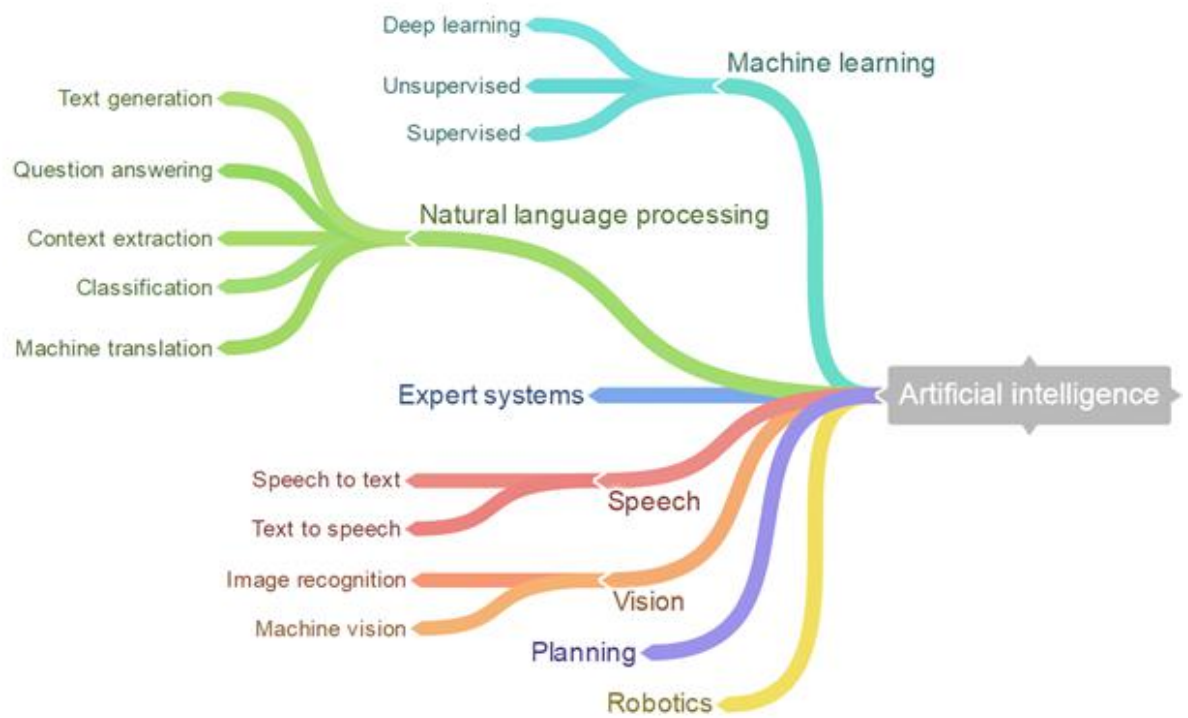


Figura 7: Subáreas da Inteligência Artificial

Fonte: Heckernoon⁷

⁷Disponível em <https://cdn-images-1.medium.com/max/1600/1*SQC439dWhYeJ31ph9Oesqg.png>

2.6 - Sobre as Tecnologias Escolhidas

Dentre as tecnologias adotadas para o desenvolvimento deste projeto, uma delas é o Ionic. O Ionic é um *framework* de código aberto que visa o desenvolvimento de aplicativos móveis híbridos.

Mas o que é *framework*? *Framework* é um conjunto de ferramentas contidas de código genéricos com o objetivo de resolver problemas comuns dos programadores, contendo os mais variados tipos de arquiteturas proporcionando mais produtividade as equipes envolvidas no projeto.

Sendo assim, o Ionic utiliza tecnologias da Web como html5, css, javascript e adota como base outros *frameworks* que é o Apache Cordova e o AngularJS que trás recursos simples ao desenvolvimento é um design mais profissional ao app.

Como todo projeto de desenvolvimento de software a sua modelagem e arquitetura é a parte principal do projeto. Portanto, utilizaremos a ferramenta Astah* UML, que é um editor para modelagem UML (Unified Modeling Language - Linguagem de Modelagem Unificada) de software e ajuda as equipes a estabelecer uma linguagem visual comum no desenvolvimento facilitando a compreensão de todos.

O git, uma ferramenta tão importante quanto as demais, ele é um sistema que permite versionar arquivos de texto e códigos fontes de projeto de software, possibilitando todas as pessoas envolvidas no projeto a contribuir para o mesmo e acompanhar cada fase.

2.7 - Principais Benefícios no Uso das Tecnologias Escolhidas

No caso do Ionic, além de ser gratuito e código aberto, estaremos criando um app disponível para iOS, android e windows phone simultaneamente já implicando na produtividade, levando ao desenvolvedor focar no objetivo do app, ele também possui componentes gráficos prontos e customizáveis beneficiado sua vasta comunidade. Além de tudo, possui padrão MVC, para manter uma boa organização, leitura e entendimento do código fonte.

A UML permite criar de uma forma padrão e fácil uma linguagem visual compreensível para todos os usuários do projeto, documentando quais características e funcionalidades o software projeto terá.

Com o git nós da mais controle e segurança do código fonte, nós permite ver qual alteração, em qual arquivo, quem fez e quando fez a tal modificação. Podemos administrar este projeto para as versão destes arquivos e data que quisermos.

3 - O Aplicativo GoodParents

3.1 - Identificação do Problema

Conforme foi visto no item anterior, a Escola SESC de Ensino Médio - ESEM, investe em um programa de apadrinhamento. Tal programa compõe-se em ajudar os alunos que estão se achegando na escola no primeiro ano, a ter uma transição mais suave possível.

Então, os alunos do segundo e terceiro ano, ajudam os novatos em sua socialização no novo ambiente. No entanto, um fator determinante para o sucesso de cada caso no programa, é a afinidade, sintonia, similaridade de gostos entre o padrinho e o apadrinhado.

E é nesse ponto que o aplicativo *GoodParents* visa minimizar os casos de insucessos do programa, recomendando os novos alunos denominados de apadrinhados que tem um certo tipo de ligação com os padrinhos disponíveis.

Concluído o levantamento dos requisitos, o principal usuário do aplicativo será o padrinho, pois estando sempre nesta condição poderá visualizar os recomendados alunos disponíveis a serem apadrinhados, e que por sua vez os apadrinhados, em sua interação com o aplicativo se limitará fazendo somente o cadastro em seu primeiro ano.

3.2 - Estado Atual do Mercado

Constatou-se que após a identificação do problema levantado, foi feita uma análise sobre os apps disponíveis nas lojas oficiais de aplicativos *Apple Store* e *Google Play*, respectivamente destinadas aos dispositivos iOS e Android que atendessem a questão apresentada de forma específica para a instituição de ensino em um ambiente educativo sem que o objetivo fosse desviado, por exemplo, o tinder, ou que os aplicativos já disponíveis no mercado não atendessem aos requisitos necessários. Por isso, foi-se tomada a decisão da criação do aplicativo em foco.

3.3 - Requisitos do Sistema

Os requisitos do sistema foram definidos juntamente com a mestrandia Érica de Lara, do Centro Universitário Carioca, no mesmo momento em que desenvolve sua dissertação. Após o levantamento dos requisitos definimos uma estratégia afim de que o problema fosse sanado e o objetivo alcançado. Logo, as funcionalidades cruciais foram detalhadas para que consigamos desenvolver o produto para que foi proposto.

3.3.1 - Requisitos Funcionais

Os requisitos funcionais são aqueles em que o sistema deve realizar, ou seja, cumprir o que foi proposto e atingir as expectativas do usuário. Em nosso caso, deve-se:

- Sugerir apadrinhados aos padrinhos
- Os coordenadores podem fazer os cadastros dos apadrinhados
- Os padrinhos podem fazer seu próprio cadastro
- Informar o período para a revelação ao apadrinhado
- Informar a importância de ser padrinho ao fazer o cadastro

3.3.2 - Requisitos não Funcionais

Os requisitos não funcionais não são aqueles que o sistema irá cumprir e sim como ele fará, estes requisitos estão atrelados ao desempenho, interface e testes, em outras palavras, esta relação à área de qualidade de software. Em nosso caso, o software deve-se:

- Ser simples e objetivo em seu uso
- Responder suas solicitações de forma ágil ao usuário.
- Requerer o mínimo possível de *hardware*, de forma que o maior número de estudantes tenham acesso.

3.4 - Características do Sistema

As características selecionadas para a aplicação destinam-se ao design intuitivo, ou seja, de fácil navegação, onde o usuário comece a interagir e chegue onde quer, visamos também que seja rápido e estável, pois um bom aplicativo não leva ao travamento do dispositivo nem a fazer com que os usuários percam tempo excessivos em suas iterações.

Outro fator importante, é que o aplicativo conduza o usuário ao seu objetivo em poucos cliques. Com o aplicativo *GoodParents*, após o cadastro e *login* do padrinho (principal usuário) será feita uma consulta na base de dados retornando todos os alunos disponíveis a serem apadrinhados, e logo em seguida o algoritmo de recomendação baseado em conteúdo é executado sugerindo os alunos com um perfil semelhante ao do padrinho logado. O conteúdo utilizado para fazer comparação de similaridade é informado no ato do cadastro, e todos os campos do mesmo são obrigatório.

3.5 - Modelagem e arquitetura do sistema

De um modo geral, o sistema implementara a arquitetura cliente-servidor. O cliente o *smartphone*, responsável pela interface com o usuário, coleta e envio das informações de cadastro. A responsabilidade do servidor é manter o banco de dados integro e nós retornar os possíveis apadrinhados recomendados.

A comunicação entre o cliente e o servidor será por meio arquivos de texto no formato json (*JavaScript Object Notation*), visando a fácil leitura dos dados e escalabilidade do produto. No primeiro contato para realizar login no sistema, será realizado o cadastro, o usuário irá preencher um formulário fornecendo todas as informações necessárias para alimentar o banco de dados local *NoSQL* (Not only Structured Query Language) MongoDB. Ao fazer login, será listado todos os possíveis apadrinhados cadastrados.

O banco de dados será alimentado pelos próprios usuário.

```

1 {
2   .... "id" :: ObjectId("5be77e1386be5b0021d3d882"),
3   .... "email" :: "i.almeida@gmail.com",
4   .... "senha" :: "e80b5017098950fc58aad83c8c14978e",
5   .... "nome" :: "Ivan",
6   .... "sobrenome" :: "Almeida",
7   .... "nascimento" :: ISODate("2018-11-11T00:00:00.000+0000"),
8   .... "estado" :: "RJ",
9   .... "padrinho" :: false,
10  .... "facilidade" :: "português, matemática, geografia",
11  .... "dificuldade" :: "inglês, espanhol, filosofia, sociologia",
12  .... "esportes" :: "futebol, vôlei, tênis de Mesa, natação",
13  .... "musicas" :: "samba-pagode, mpb",
14  .... "filmes" :: "ação, animação, aventura, comédia",
15  .... "descricao" :: "Sou uma pessoa com muitos sonhos e projetos, e pretendo realizá-los",
16  .... "foto" :: "data:image/jpeg;base64,/9j/4AAQSKZJRgABAQAASABIAAD/4QBYRXhpZgAATU0AKgAAAagAASAA",
17  .... "createdAt" :: ISODate("2018-11-11T00:55:47.672+0000"),
18  .... "updatedAt" :: ISODate("2018-11-11T00:55:47.672+0000"),
19  .... "__v" :: NumberInt(0)
20 }

```

Figura 8: Dados estruturados em formato json inserido no banco de dados

Fonte: Elaborado pelo autor

3.5.1 - Diagrama de Caso de Uso

O diagrama de caso de uso é uma breve descrição do que o sistema faz no ponto de vista do usuário. Ou seja, nós demonstra as principais funcionalidades e iterações do sistema sem entrar em detalhes técnicos.

Os seguintes casos de uso foram identificados para serem implementados no sistema : Efetuar Login, Cadastrar Aluno, Exibir Home, Manter Perfil, e Alterar Cadastro.

3.5.1.1 - Efetuar Login

Esta fase ocorre quando o aplicativo é inicializado, ela dará a opção de fazer login para os padrinhos já cadastrados e em seguida os levando para a tela *home* do sistema, e de cadastro para os novos padrinhos e os alunos que serão apadrinhados, conduzindo-o os mesmos para a tela de cadastro.

3.5.1.2 - Cadastrar Aluno

Todos os dados coletados nesta fase serão persistidos no banco de dados. Após a finalização desta etapa o usuário será conduzido a tela de login.

3.5.1.3 - Exibir Home

Esta fase é a principal do sistema, aqui será sugerido os apadrinhados com perfis semelhantes ao do padrinho logado e dará a possibilidade de ver o perfil completo do alunos e retirar da lista de apadrinhados disponíveis.

3.5.1.5 - Manter Perfil

Esta fase possibilita ao usuário logado ver informações detalhadas sobre seu perfil ou dos alunos listados e caso deseje, ao ver seu dados cadastrados tem a opção de editar seu perfil ou deletar.

3.5.1.6 - Alterar Cadastro

Esta fase possibilitará ao usuário alterar o seu próprio cadastro ou dos usuários cadastrados.

3.5.2 - Detalhamento dos Casos de Uso

3.5.2.1 - Efetuar Login

Caso de uso: Efetuar Login

Atores: Usuário(Padrinho, Apadrinhado)

Finalidade: Efetuar login no aplicativo para ver os apadrinhados disponíveis.

Tabela 1: Caso de Uso Efetuar Login

Sequência de Eventos	
Ação do Ator	Resposta do Sistema
1. Inicia o aplicativo. 2. Informa email e senha.	
	3. Verifica se o usuário já possui cadastro. Caso contrário, irá disparar a exceção E1 ou E2. 4. Exibirá a tela de apadrinhado disponíveis.
5. Visualiza a lista retornada pelo sistema.	
Exceções	
E1: Informa que o usuário não possui cadastro.	
E2: Informa que o usuário não tem permissão de acesso.	

3.5.2.2 - Cadastrar Aluno

Nome: Cadastrar Aluno

Atores: Usuário(Padrinho, Apadrinhado)

Finalidade: Coletar dados fornecidos pelo usuário.

Observação: Assume-se que o usuário não possui cadastro.

Tabela 2: Caso de Uso Cadastrar Aluno

Sequência de Eventos	
Ação do Ator	Resposta do Sistema
1. Acessa a opção de cadastrar-se.	
	2. Coleta todas as informações fornecidas pelo usuário. 3. Irá se conectar ao banco de dados. Caso contrário, lança a exceção E1. 4. Persistir os dados coletados no banco de dados. 5. Exibirá que o usuário foi cadastrado com sucesso. 6. O aplicativo exibirá a tela de Login.
7. Visualiza a tela de login.	
Exceções	
E1: Exibe a mensagem que não foi possível se conectar ao banco de dados e pede para o usuário tentar novamente.	

3.5.2.3 - Exibir Home

Nome: Exibir Home

Atores: Usuário (Padrinho)

Finalidade: Exibir os alunos com perfil semelhante ao do usuário logado.

Observação: Assume-se que o usuário possui cadastro.

Tabela 3: Caso de Uso Exibir Home

Sequência de Eventos	
Ação do Ator	Resposta do Sistema
1. Acessa a opção de login. 2. Informa o email e senha.	
	3. O aplicativo retorna os apadrinhados com perfil semelhante ao do usuário logado.
4. O usuário visualiza a lista retornada pelo sistema.	

3.5.2.4 - Manter Perfil

Nome: Manter Perfil

Atores: Usuário(Padrinho)

Finalidade: Exibir informações detalhadas do aluno selecionado.

Tabela 4: Caso de Uso Manter Perfil

Sequência de Eventos	
Ação do Ator	Resposta do Sistema
1. Acessa a opção de ver perfil	
	2. Retorna informações detalhadas sobre o perfil selecionado.
3. Visualiza as informações do apadrinhado.	

3.5.2.5 - Alterar Cadastro

Nome: Alterar Cadastro

Atores: Usuário (Padrinho)

Finalidade: Alterada o cadastro dos usuário cadastrado no sistema.

Tabela 5: Caso de Uso Alterar Cadastro

Sequência de Eventos	
Ação do Ator	Resposta do Sistema
1. Acessa a opção de alterar cadastro.	
	2. Retorna o formulário de cadastro no sistema. 3. Efetua conexão no banco de dado, caso contrario irá disparar a exceção E1. 4. Irá persistir as informações no banco de dados. 5. Solicitará ao usuário para fazer login novamente.
4. Visualiza a tela de login.	
Exceções	
E1: Exibe a mensagem que não foi possível se conectar ao banco de dados e pede para o usuário tentar novamente.	

3.5.3 - Diagrama de Classes Conceitual do Sistema

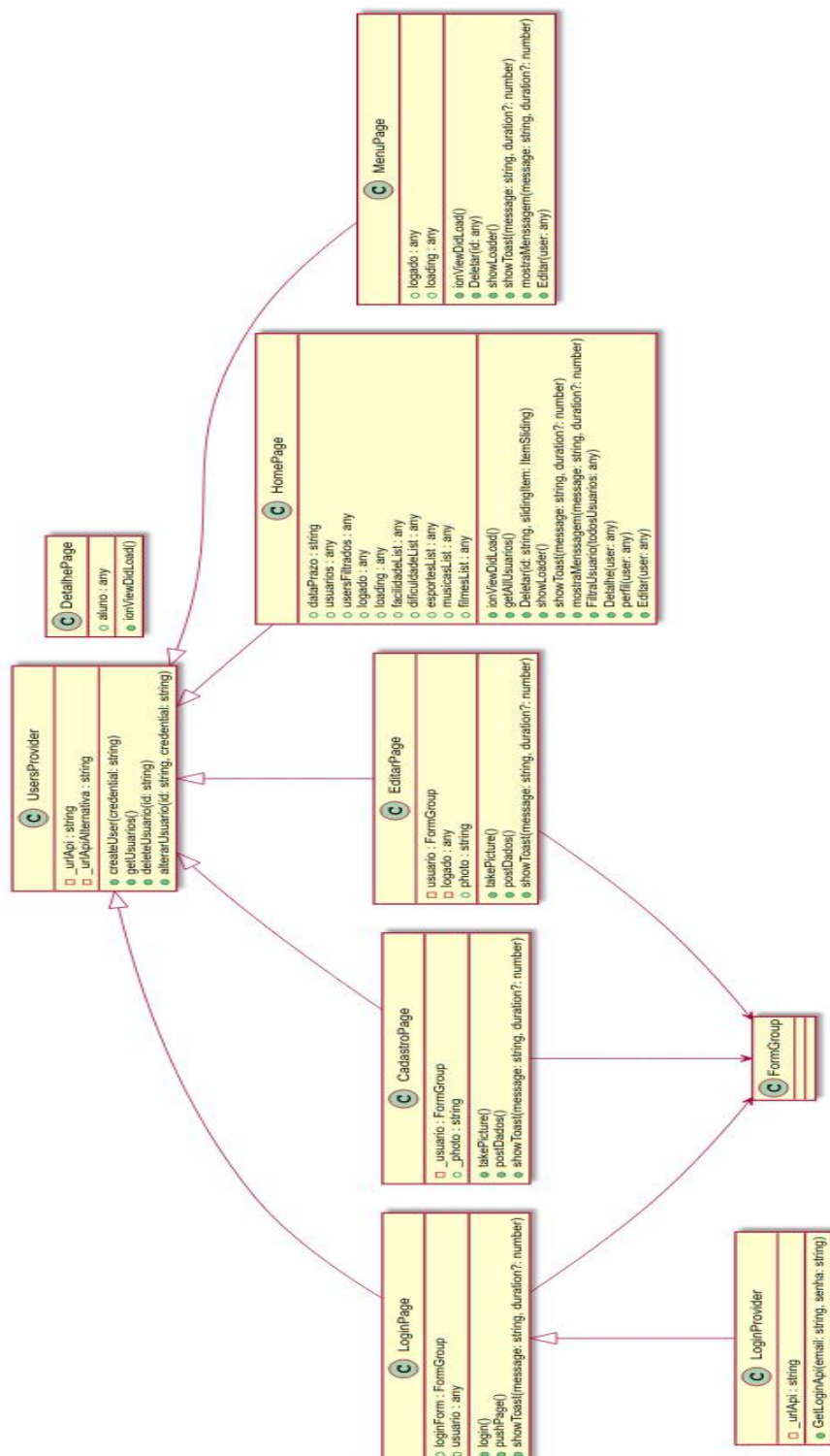
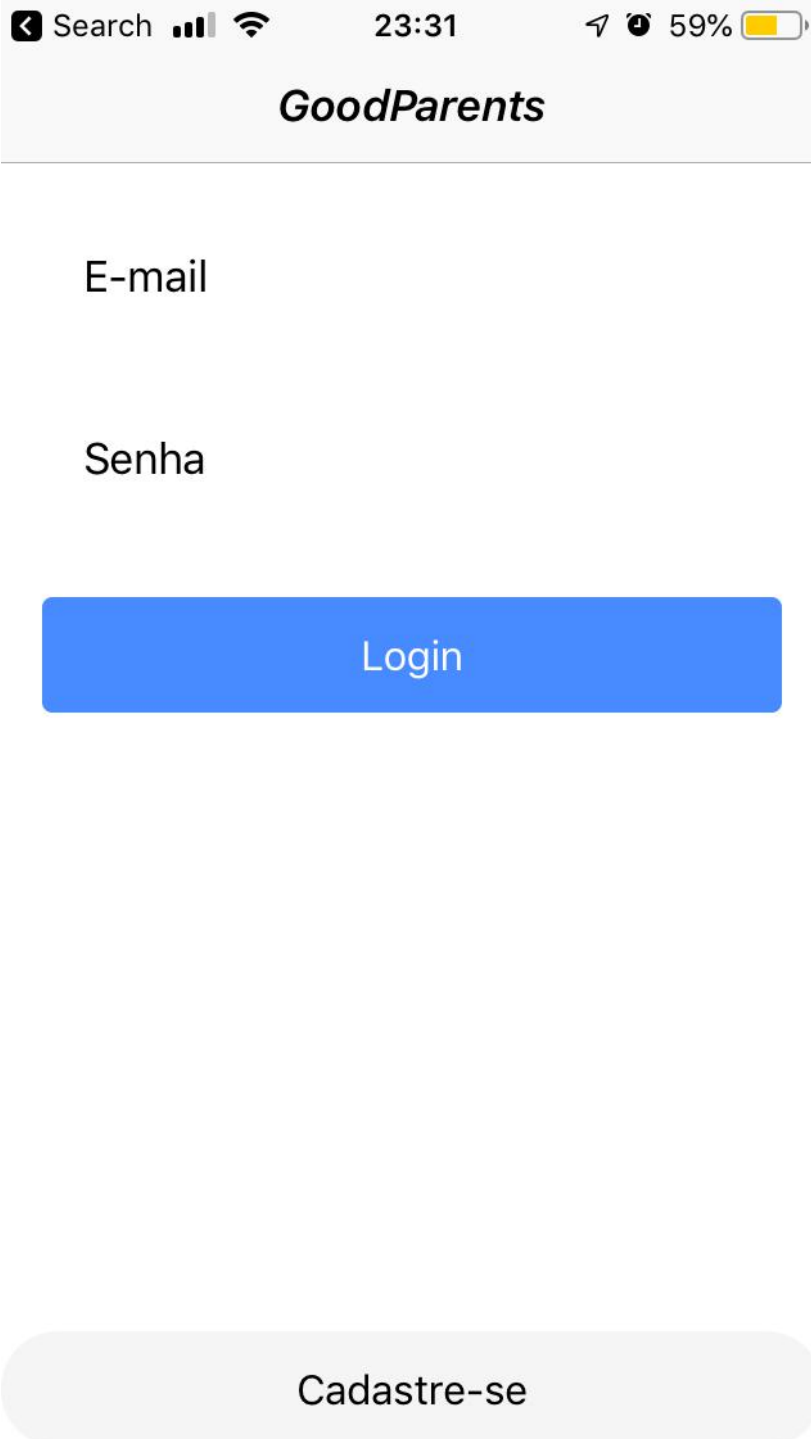


Figura 10: Diagrama de classes do protótipo

Fonte: Elaborado pelo autor

3.5.4 - Protótipo do Sistema GoodParents



The image shows a mobile application interface for 'GoodParents'. At the top, there is a status bar with a back arrow, the word 'Search', signal strength bars, a Wi-Fi icon, the time '23:31', a location pin icon, a clock icon, '59%', and a battery icon. Below the status bar is a header with the text 'GoodParents' in a bold, italicized font. The main content area contains two text input fields labeled 'E-mail' and 'Senha'. Below these fields is a blue button with the text 'Login'. At the bottom of the screen is a light gray button with the text 'Cadastre-se'.

Search 23:31 59%

GoodParents

E-mail

Senha


Login


Cadastre-se

Figura 12: Página de login do protótipo
Fonte: Elaborado pelo autor

Search 23:33 58%

[Back](#) Cadastro





Name

Ivan

Último Nome

Paes

E-mail

i.almeida@gmail.com

Senha

●●●●●●

Data de Nascimento

28/4/1990

Figura 13: Página de cadastro do protótipo

Fonte: Elaborado pelo autor

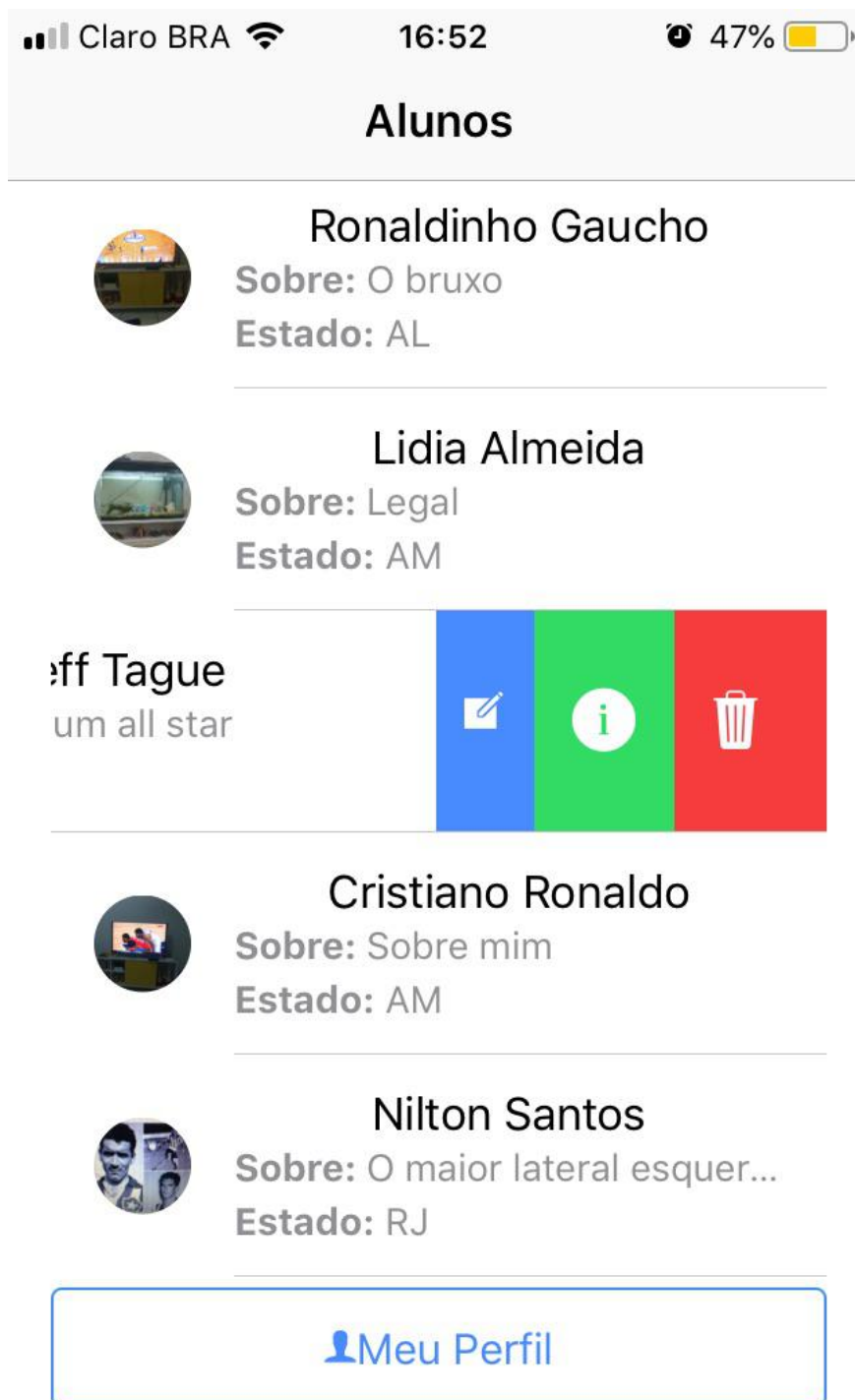


Figura 14: Página Home do protótipo

Fonte: Elaborado pelo autor



Figura 15: Página de detalhes do apadrinhado

Fonte: Elaborado pelo autor



Figura 16: Página de perfil do padrinho logado
Fonte: Elaborado pelo autor

4 - Análise Crítica

Pretendemos nesta etapa do trabalho apresentar-lhes outros aplicativos do mesmo, e que disponibilizam funcionalidades similares. A partir daí, avaliá-los com relação ao GoodParents, comparando as suas características e diferenças, para ponderarmos possíveis melhorias no futuro.

4.1- Aplicativos Avaliados

Até o momento do desenvolvimento deste estudo, todos os aplicativos avaliados foram encontrados em pesquisas nas lojas oficiais de aplicativos de suas respectivas plataformas, *Google Play* para dispositivo Android e *App Store* para dispositivo IOS, como dito antes. Os apps avaliados foram:

- Tinder

- Facebook

- Meetup

As avaliações seguem nas próximas páginas, sendo esse *layout* escolhido para facilitar a organização do trabalho. As imagens das telas dos aplicativos foram todas feitas pelo autor do trabalho, testando os mesmos.

5.1.1- Tinder

Tinder é um aplicativo de socialização baseado em localização, popularizado como um aplicativo de encontro, paquera e relacionamento. O aplicativo permite conhecer novas pessoas que possuem interesses em comum e que estejam relativamente próximas.

Para obter uma melhor experiência aplicativo requer uma conta em outra plataforma, o Facebook, a partir daí o app acessa informações importantes para seu funcionamento, como fotos, endereço, idade e profissão, além de cruzar baseadas em suas amizades e do seu GPS.

Conforme as sugestões de perfis é sugerida para o usuário, temos a opção de deslizar o dedo em cima da foto para a direita ou clicar no coração verde dando um “like”, sinalizando que você tem interesse na pessoa. Ao clicar no “X”, ou deslizando o dedo para esquerda, você estaria descartando o perfil visualizado. É fuge totalmente da proposta de apoio, integração e educação do GoodParents.

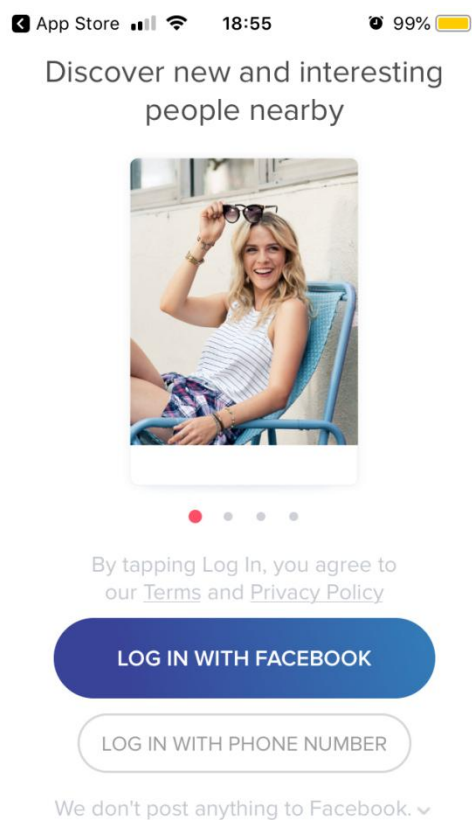


Figura 17: Tela de login do app Tinder

Fonte: *Print screen* da aplicação no sistema operacional IOS

5.1.2 - Facebook

Na rede social mais utilizada do mundo originada na faculdade, é possível criar um perfil pessoal, *fanpage*, eventos, perfis comerciais, até mesmo uma “*marketplace*”, interagir com outras pessoas através de mensagens instantâneas, postagens, participar de grupos de interesse, dentre outras funcionalidades.

Analisando mais especificamente a funcionalidade de recomendação, podemos observar que suas sugestões são baseadas em diversos serviços e cruzamento de informações, como lista de contato telefônico, amigos em comum, email, marcação em fotos, endereços, escola, universidade, trabalho e participação nos mesmos grupos, ou seja, não atende a real proposta de apadrinhamento do aplicativo GoodParents.

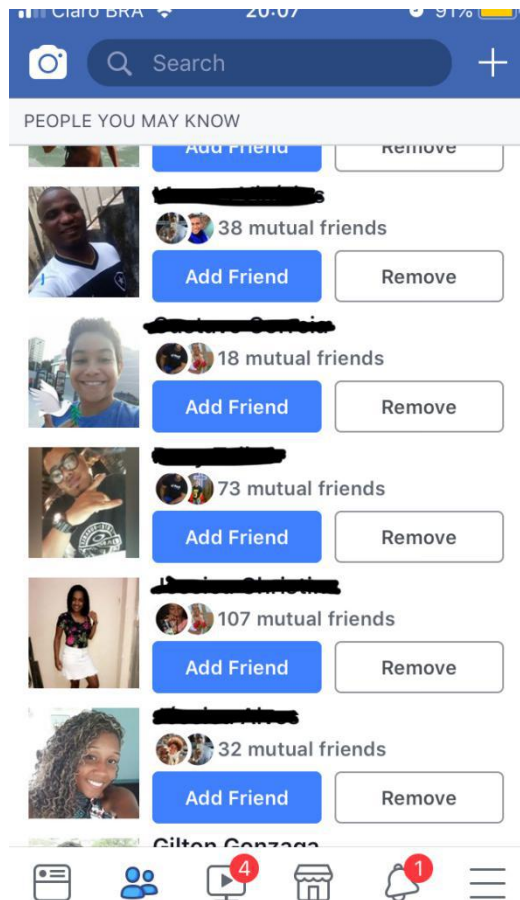


Figura 18: Tela de sugestão de amigos do app Facebook
Fonte: *Print screen* da aplicação no sistema operacional IOS

5.1.3 - Meetup

O meetup é um app de clube, em seu primeiro contato o app colhe informações dos quais assuntos você se interessa e a sua localização, a partir daí, o app lhe sugere encontros coletivos, eventos que ocorreram em determinados lugares com pessoas que gostam das mesmas coisas que você e que provavelmente você nunca viu na vida.

Existe dos mais variados grupos, grupos de literatura ao *crochet*, e se você não estiver em nenhum grupo, o app lhe oferece o que está acontecendo na cidade, e ir, se for de sua preferência.

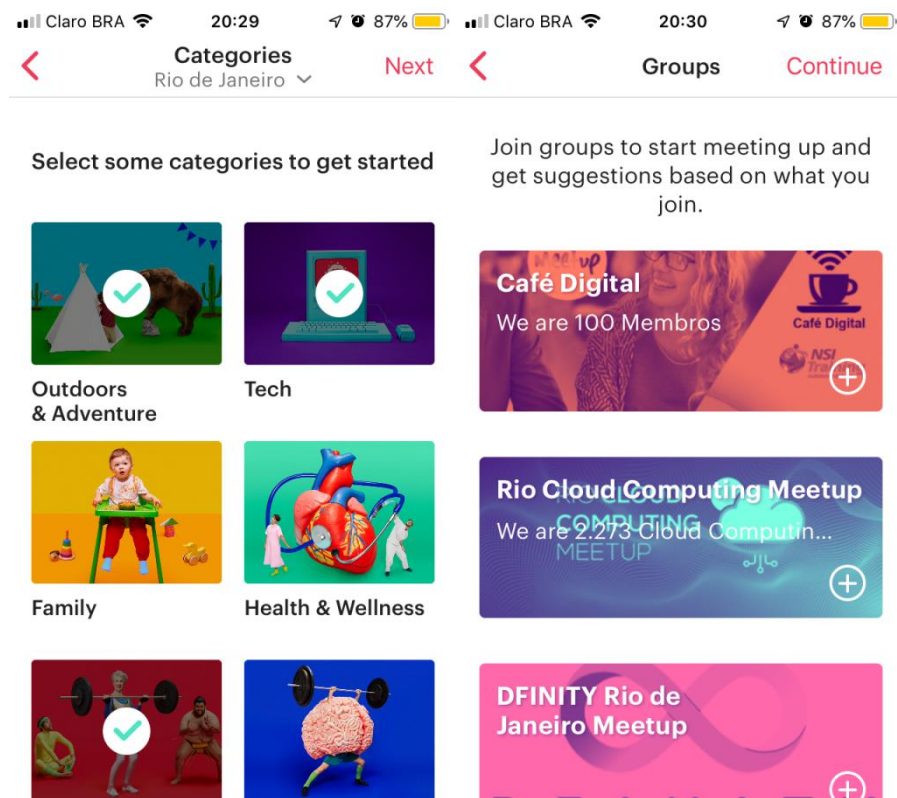


Figura 19: Tela de Meetup

Fonte: *Print screen* da aplicação no sistema operacional IOS

4.2- Quadro Comparativo

Na tabela abaixo são demonstrados as características do *GoodParents*, com relação os aplicativos avaliados. Objetivo é solidificar que a principal funcionalidade que o *GoodParents* possui, não existe em nenhum outro aplicativo, até a data de desenvolvimento deste trabalho, que poderia ser aproveitado pela escola SESC.

Tabela 6: Quadro comparativo de aplicativo móveis

	Tinder	Facebook	Meetup	GoodParents
Gratuito	Sim	Sim	Sim	Sim
Facilidade de uso	Fácil	Médio	Fácil	Fácil
Requer cadastro	Sim	Sim	Não	Sim
Recomenda usuários	Sim	Sim	Não	Sim
Recomenda usuários com perfis semelhantes	Não	Não	Não	Sim
Os usuários recomendados tem objetivo de ajudar na socialização e no reforço escolar	Não	Não	Não	Sim

4.3- Conclusão da Análise Comparativa

Sendo assim, analisando o quadro comparativo, constatou-se que apesar de todos os sistemas serem gratuitos, alguns se aproximam com sugestões de usuários, mas nenhum atende aos requisitos apresentados pelos casos de insucesso deparados no programa de apadrinhamento da escola SESC.

5 - Conclusão

Este capítulo trata-se dos objetivos do trabalho, até que ponto foram alcançados. Além do mais, dedica-se também de possíveis melhorias no aplicativo.

5.1 Considerações finais

Recordando o que foi descrito no primeiro capítulo de introdução, o propósito deste trabalho é implementar um aplicativo voltado para Escola SESC de Ensino Médio - ESEM, localizada no bairro de Jacarepaguá, Rio de Janeiro, de recomendação de alunos fundamentado nas preferências, facilidades e dificuldades em aprendizado sobre determinadas matérias aos padrinhos.

Conforme foi retratado, pesquisas foram feitas sobre os aplicativos no atual mercado, acerca de sistemas que atendessem os requisitos demonstrados no parágrafo anterior. Em seguida, foi feita a análise e modelagem do aplicativo em si, é o desenvolvimento do mesmo. Além do mais, geramos uma análise comparativa com relação a outros aplicativos similares.

Após a conclusão do trabalho, somos capazes de dizer que o propósito foi alcançado. Além disso, novos conhecimentos foram adquiridos com o desenvolvimento do aplicativo e com a pesquisa sobre a área, além da análise comparativa. Desta maneira, algumas ideias para a melhoria do aplicativos foram surgindo ao longo do desenvolvimento do mesmo. Tais ideias não foram implementadas no sistema por motivos de celeridade, entretanto, algumas delas serão mencionadas mais adiante.

5.2 Possibilidade de melhorias futuras

Conforme foi feito o estudo de outros aplicativos na etapa de análise comparativa e ao longo do desenvolvimento do aplicativo, foram identificados possibilidades de melhorias futuras ao app, para o que mesmo possa se tornar mais atrativo e consistente.

Uma destas possibilidades é a funcionalidade de permitir no ato do cadastro dos usuários, a integração com as API's do Facebook e do Gmail (Google), com isso teríamos acesso a informações importantes, como nome e sobrenome, foto de perfil e estado, diminuindo o esforço do usuário. Além do mais, tais API's se encarregariam do processo de validação dos email. Tal implementação, também iria requerer uma atenção especial quanto ao armazenamento das fotografias de perfil, como já frisado no paragrafo anterior.

Outra melhoria possível, seria um serviço de alerta aos padrinhos com o perfil semelhante a cada novo apadrinhado cadastrado no aplicativo, levando ao mesmo a analisar o perfil do detalhado com novo aluno recém chegado a instituição.

Nada impede o que as alternativas sugeridas acima possam ser executadas ao mesmo tempo, Pois as mesmas não desviarão o propósito do qual o aplicativo foi criado, que é recomendação, facilidade de uso e agilidade do app.

5.3 Limitações encontradas no decorrer do trabalho

Em síntese, o principal desafio e limitação encontrado no que concerne ao processo de construção do trabalho de conclusão de curso está relacionado a escassez de tempo. De acordo com as informações levantadas na pesquisa, constata-se como esta uma das dificuldades para administrar o tempo, efetivamente, com autonomia e responsabilidade.

E conseqüentemente com a dificuldade em termos de falta de tempo e dedicar-se nas outras disciplinas necessárias para conclusão do curso, juntamente com o empenho aplicado ao desenvolvimento do trabalho em questão, impacta no escopo do projeto, levando-se a dedicação somente na principal funcionalidade.

Outro desafio encontrado foi ter acesso ao campo a ser pesquisado e a materiais bibliográficos, recursos tecnológicos, além de redigir tal texto científico. Porém, tais obstáculos não podem ser justificativa para um insucesso na fase de encerramento de curso.

Referências

SESC, “**Sesc abre inscrições para escola-residência no Rio de Janeiro**”. Disponível em <<http://novosite.sesc-se.com.br/destaque/escolasescdeensinomedio2018>>. Acesso em setembro de 2018.

Carta Educação, “**Escola e residência**”. Disponível em <<http://www.cartaeducacao.com.br/reportagens/escola-e-residencia/>>. Acesso em setembro de 2018.

Folha de S. Paulo, “**Veja o desempenho da sua escola no Enem de 2017**”. Disponível em <<https://www1.folha.uol.com.br/educacao/2018/06/veja-o-desempenho-da-sua-escola-no-enem-2017.shtml>> Acesso em setembro de 2018.

Abranet, “**Brasil possui 280 milhões de dispositivos conectados a internet**”. Disponível em <<http://www.abranet.org.br/Noticias/Brasil-possui-280-milhoes-de-dispositivos-moveis-conectaveis-a-Internet-1419.html?UserActiveTemplate=site#.W6Q0Ut-YXQq>>. Acesso em setembro de 2018.

Globo, “**Primeiro smartphone do mundo completa 20 anos**”. Disponível em <<https://oglobo.globo.com/sociedade/tecnologia/primeiro-smartphone-do-mundo-completa-20-anos-13630167>> Acesso em setembro de 2018.

Oficina da Net, “**A história do Android**”. Disponível em <<https://www.oficinadanet.com.br/post/13939-a-historia-do-android>> Acesso em setembro de 2018.

Oficina da Net, “**A história do iOS**”. Disponível em <<https://www.oficinadanet.com.br/post/17950-a-historia-do-ios>> Acesso em setembro de 2018.

RICH, Elaine. **Inteligência Artificial**. São Paulo: MacGraw-Hill, 1988.

Carvalho, Luís Alfredo Vidal de. **Datamining - A Mineração de Dados no Marketing , Medicina, Economia, Engenharia e Administração**. Rio de Janeiro: Editora Ciência Moderna Ltda.,2015.

Larman, Craig. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objeto**. Porto Alegre: Bookman, 2000.

DtDigital, “**Aplicativos Mobile - Multiplataforma vs Nativo: Qual utilizar?**”. Disponível em <<http://dtidigital.com.br/blog/multiplataforma-vs-nativo-qual-utilizar-em-aplicativos/>> Acesso em setembro de 2018.

Coursera, “**What is Machine Learning**”. Disponível em <<https://www.coursera.org/lecture/machine-learning/what-is-machine-learning-Ujm7v>> Acesso em setembro de 2018.

Quora, “**What is Machine Learning**”. Disponível em <<https://www.quora.com/What-is-machine-learning-4>> Acesso em setembro de 2018.

Ibm, “**Sistemas de Recomendação**”. Disponível em <https://www.ibm.com/developerworks/br/local/data/sistemas_recomendacao/> Acesso em setembro de 2018.

IGTI, “**Entenda como funcionam os Sistemas de Recomendação**”. Disponível em <<http://igti.com.br/blog/como-funcionam-os-sistemas-de-recomendacao/>> Acesso em setembro de 2018.

TechTudo, “**Astah Community**”. Disponível em <<https://www.techtudo.com.br/tudo-sobre/astah-commmunity.html>> Acesso em setembro de 2018.

AQUILES, Alexandre; FERREIRA, Rodrigo. **Controlando versões com Git e Github**. São Paulo: Casa do Código.

RockApps, “**7 motivos pelos quais a sua empresa precisa utilizar a Ionic framework para desenvolvimento de aplicativos híbridos**”. Disponível em <<https://www.rockapps.com.br/motivos-pelos-quais-a-sua-empresa-precisa-utilizar-a-ionic-framework-para-desenvolvimento-de-aplicativos-hibridos-android-e-ios/>> Acesso em setembro de 2018.

Apêndice - Preparando o ambiente de desenvolvimento

Abaixo vamos descrever alguns serviços e orientações para que você possa dar os primeiros passos para desenvolver seu aplicativo e utilizar as tais ferramentas.

Apêndice - Instalando o Node.js

O Node.js é uma das principais ferramentas do projeto, considerado o canivete suíço dos programadores, o Node.js é uma plataforma Javascript para programação de propósito geral, que permite aos usuários construírem aplicações de rede rapidamente. Podendo o Javascript ser usado tanto ao front-end quanto ao back-end.

Portanto, se você não tem o Node instalado, poderá baixá-lo em <https://nodejs.org/en/download/>.

The screenshot shows the Node.js download page. At the top, there's a navigation bar with links: HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, NEWS, and FOUNDATION. Below this, the 'Downloads' section highlights the 'Latest LTS Version: 10.13.0 (includes npm 6.4.1)'. It instructs users to download the source code or a pre-built installer. The page is divided into two main sections: 'LTS Recommended For Most Users' and 'Current Latest Features'. Under 'LTS', there are links for 'Windows Installer' (node-v10.13.0-x86.msi), 'macOS Installer' (node-v10.13.0.pkg), and 'Source Code' (node-v10.13.0.tar.gz). Under 'Current', there are links for 'Windows Installer' (node-v10.13.0-x86.msi), 'macOS Installer' (node-v10.13.0.pkg), and 'Source Code' (node-v10.13.0.tar.gz). A table below lists the available binaries for both LTS and Current versions, categorized by architecture (32-bit, 64-bit) and platform (ARMv6, ARMv7, ARMv8).

LTS		Current	
Recommended For Most Users		Latest Features	
32-bit	64-bit	32-bit	64-bit
32-bit	64-bit	32-bit	64-bit
64-bit		64-bit	
64-bit		64-bit	
64-bit		64-bit	
ARMv6	ARMv7	ARMv8	
node-v10.13.0.tar.gz			

Figura 20: Tela de download do NodeJs

Na fase de instalação, se certifique que o NPM também será instalado. O NPM é a o pacote de gerenciamento para a linguagem de programação Javascript.

Caso esteja usando um ambiente Linux, a ferramenta pode ser instalada com os seguintes comandos.

```
$ sudo apt update
```

```
$ sudo apt install nodejs
```

```
$ sudo apt install npm
```

Para se certificar que tudo foi instalado corretamente, digite em seu shell ou prompt, os seguintes comandos:

```
$ node -v
```

```
$ npm -v
```

Tais comandos mostrará a versão instalada das ferramentas.

Apêndice - Utilizando o MongoDB

Como foi-se mencionado antes, o MongoDB é um banco não relacional e um *document database* (banco de dados de documentos). A representatividade dos documentos é semelhante ao JSON com esquemas.

Visando a escalabilidade e a produtividade foi-se ser utilizado os serviços do mLab.

O mLab é um serviço de hospedagem de banco de dados Mongo na nuvem, totalmente gerenciado, e é executado em provedores como a Amazon Web Services (AWS), Google Cloud e Microsoft Azure. E a melhor parte é que o serviço tem um plano gratuito que oferece 0,5gb para armazenamento de dados. Em poucos segundos pode-se configurar e deixa-lo rodando o seu banco de dados na nuvem.

O serviço pode ser encontrado aqui <https://mlab.com/>.

Apêndice - Escolhendo o editor de texto

Ferramenta fundamental para qualquer desenvolvimento de projeto com tais finalidades, precisa-se ao mínimo de um editor de texto. Com muitas opções de editores e IDE's no mercado, cada programador tem sua preferência, sendo assim será listado duas sugestões para serem utilizadas.

- Sublime Text : Um sofisticado editor de texto multiplataforma, não uma IDE como muitos dizem, é muito leve, simples e com interface agradabilíssima. Link para baixa-lo <https://www.sublimetext.com/3>.
- Visual Studio Code : É um editor de código-fonte / IDE de mínimos recursos multiplataforma, com suporte para depuração e compilação para tais projetos. Link para baixa-lo <https://code.visualstudio.com/download>.

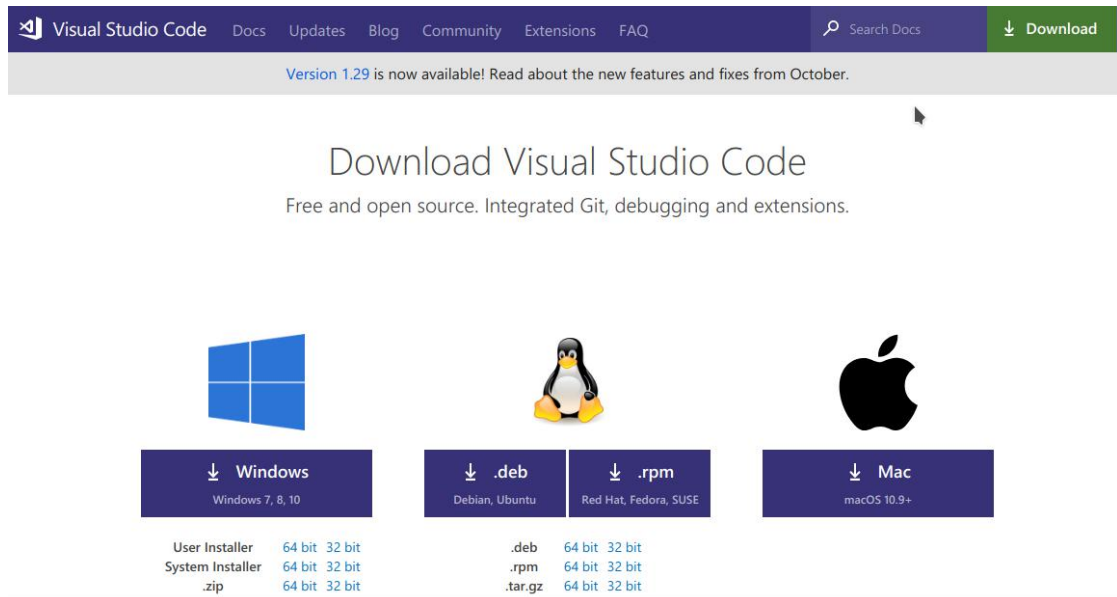


Figura 21: Tela de download do editor de texto VS Code

Apêndice - Disponibilizando o servidor back-end

Como foi-se mencionado anteriormente, utilizaríamos a ferramenta git, e atrelado a ela o github. Portanto, pode-se ter acesso a todo o código do back-end no seguinte link <https://github.com/iamgilvan/ApiGoodParents.git>. Ele é responsável por fazer toda a integração com o banco de dados e o CRUD (Create, Read, Update, and Delete).

Pensando em acessibilidade e performance, foi-se utilizado o Heroku para hospedar nossa API. Heroku é uma plataforma como um serviço (PaaS), permite operar aos criar, executar e operar aplicativos inteiramente na nuvem. Além de suportar várias linguagem de programação e possuir planos gratuitos. Para mais informações <https://www.heroku.com/>.

Parte do banco de dados no mLab e com a API hospedada no Heroku pode ser vista aqui <https://good-parents-server.herokuapp.com/>.

Outra ótima ferramenta e primordial no desenvolvimento de API's é o Postman. O Postman é um API Client que facilita aos desenvolvedores criar, compartilhar, testar e documentar API's. Isso é feito, permitindo aos usuários criar e salvar solicitações **HTTP** e **HTTPs** simples e complexas, bem como ler suas respostas.

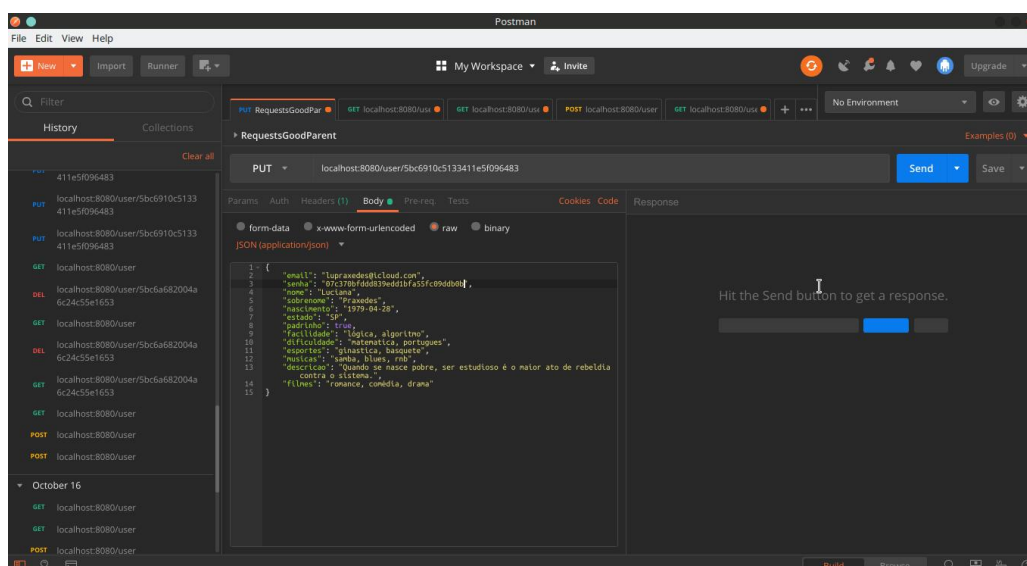


Figura 22: Tela do software Postman durante o desenvolvimento do aplicativo

Fonte: Elaborado pelo Autor

O Postman pode ser baixado aqui <https://www.getpostman.com/>.

Apêndice - Instalando o Ionic

Com o Node e o NPM devidamente configurado, vamos instalar o Ionic e o Cordova CLI. Em um prompt de comando (Windows) ou shell bash (Linux) digite os seguintes comandos.

```
$ npm install -g ionic cordova
```

O parâmetro `-g` significa que é uma instalação global, talvez para Linux será necessário rodar o comando com `sudo`.

Para rodar o aplicativos nas plataformas Android e IOS, execute os comandos abaixo.

```
$ ionic platform add android
```

```
$ ionic platform add ios
```

Uma vez feito isso, crie seu primeiro app ionic

```
$ ionic start helloWorld black
```

Para rodar o app, entre no diretório que foi criado e rode o comando para testar sua aplicação no browser.

```
$ cd helloWorld
```

```
$ ionic serve
```

Lembrando que o código desenvolvido para este app se encontra no github com o link <https://github.com/iamgilvan/AppGoodParents.git>.