

UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE - UERN
FACULDADE DE CIÊNCIAS EXATAS E NATURAIS – FANAT
DEPARTAMENTO DE INFORMÁTICA – DI
CIÊNCIA DA COMPUTAÇÃO

WILTON SILVA DOS SANTOS JÚNIOR

**LANGUAGE ADVISER: APLICAÇÃO MULTIPLATAFORMA PARA O APOIO DO
ENSINO DE LÍNGUA ESTRANGEIRA**

MOSSORÓ - RN

2018

WILTON SILVA DOS SANTOS JÚNIOR

**LANGUAGE ADVISER: APLICAÇÃO MULTIPLATAFORMA PARA O APOIO DO
ENSINO DE LÍNGUA ESTRANGEIRA**

Monografia apresentada à Universidade do Estado do Rio Grande do Norte como um dos pré-requisitos para obtenção do grau de bacharel em Ciência da Computação, sob orientação do Prof. Me. Antônio Oliveira Filho e coorientação do Prof. Dr. Rommel Wladimir de Lima.

MOSSORÓ - RN

2018

Catálogo da Publicação na Fonte.
Universidade do Estado do Rio Grande do Norte.

S237I Santos Júnior, Wilton Silva dos
Language Adviser: Aplicação Multiplataforma para o
Apoio do Ensino de Língua Estrangeira. / Wilton Silva dos
Santos Júnior. - Mossoró, 2018.
69p.

Orientador(a): Prof. Me. Antonio Filho Oliveira.
Coorientador(a): Prof. Dr. Rommel Wladimir Lima.
Monografia (Graduação em Ciência da Computação).
Universidade do Estado do Rio Grande do Norte.

1. Ciência da Computação. I. Oliveira, Antonio Filho. II.
Universidade do Estado do Rio Grande do Norte. III.
Título.

WILTON SILVA DOS SANTOS JÚNIOR

**LANGUAGE ADVISER: APLICAÇÃO MULTIPLATAFORMA PARA
O APOIO DO ENSINO DE LÍNGUA ESTRANGEIRA**

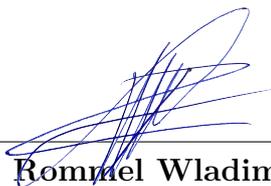
Monografia apresentada à Universidade do Estado do Rio Grande do Norte como um dos pré-requisitos para obtenção do grau de bacharel em Ciência da Computação, sob orientação do Prof. Me. Antônio Oliveira Filho e coorientação do Prof. Dr. Rommel Wladimir de Lima.

Aprovado em 23 de Maio de 2018.



Prof. Me. Antônio Oliveira Filho

Universidade do Estado do Rio Grande do Norte - UERN



Prof. Dr. Rommel Wladimir de Lima

(Coorientador)



Prof. Dr. Francisco Chagas de Lima Júnior

Universidade do Estado do Rio Grande do Norte - UERN



Profa. Ma. Ceres Germanna Braga Moraes

Universidade do Estado do Rio Grande do Norte - UERN

A minha família.

Agradecimentos

Primeiramente, agradeço a Deus, pois sem ele nada seria possível. Agradeço por ter me proporcionado oportunidades, conquistas e por me abençoado sempre com saúde.

Agradeço a minha família por sempre acreditar em mim, em especial a minha mãe, Elzamar Ferreira, meu pai Wilton Silva, meu irmão Tiago Silva e a minha vó Maria Luiza, por sempre estarem dispostos a me ajudar e por me apoiarem nos dias mais difíceis, durante essa caminhada.

Agradeço a todas as amizades que construí ao longo desses anos na faculdade, em especial, Álvaro Gabriel, Giovana Lorena e Claudivan Barreto. Aos meus amigos que adquiri ao longo deste curso, obrigado pelos agradáveis momentos que compartilhamos, lembrarei sempre dessa época com muita empolgação e alegria.

Aos professores do DI, pela contribuição para o meu desenvolvimento intelectual, pelos ensinamentos atribuídos e pelos conselhos prestados, em especial aos professores Marcelino Pereira, Ceres Germanna, Lima Júnior, André Pedro, meu orientado Antônio Oliveira por todo seu esforço e empenho na orientação a professora Cícilia Maia pela sua extrema dedicação e atenção. Por último, meu coorientador e tutor Rommel Lima, pelas valiosas contribuições neste trabalho de conclusão de curso.

*“O sucesso é a soma de pequenos
esforços repetidos dia após dia.”
Robert Collier*

Resumo

Métodos convencionais para o ensino de língua estrangeira utilizando conteúdos físicos, como livros e CDs de áudio, não tornam-se tão eficazes e atraentes para o aluno, comparados aos materiais digitais disponibilizados pelos *softwares* educativos para promover o ensino e aprendizagem. Devido a isso, faz-se necessária a utilização de tecnologias em diferentes práticas sociais e educacionais, estabelecendo parâmetros para as escolas que atuam como mediador de conteúdo, disponibilizando materiais as para plataformas digitais, evitando gastos desnecessários na disponibilidade de conteúdo, facilitando a prática educacional do aluno, não se limitando a materiais físicos para o estudo. Com a plataforma digital o aluno terá acesso ao conteúdo de qualquer lugar ou região, potencializando seu estudo em um determinado idioma. Esse trabalho tem como objetivo o desenvolvimento de uma ferramenta que implemente o conceito de interoperabilidade entre as aplicações. Permitindo a comunicação com uma API REST, para o processamento de dados da ferramenta, utilizando os *frameworks* Angular e Ionic para agilizar a criação de um *software* multiplataforma que facilite a implementação da metodologia de escolas especializadas no ensino de língua estrangeira.

Palavras-chave: API REST, Framework, Metodologia, Ensino, Conteúdo

Abstract

Conventional methods for teaching foreign languages using physical content, such as books and audio CDs, do not become as effective and attractive to the student compared to the digital materials provided by educational software to promote teaching and learning. Due to this, it is necessary to use technologies in different social and educational practices, establishing parameters for schools that act as mediator of content, making available materials for digital platforms, avoiding unnecessary expenses in the availability of content, facilitating the educational practice of the student, not limited to physical materials for the study. With the digital platform the student will have access to the content of any place or region, making possible its study in a certain language. This work aims to develop a tool that implements the concept of interoperability between applications. Allowing the communication with a REST API, for the processing of data of the tool, using the angular and Ionic frameworks to expedite the creation of a cross-platform software that facilitates the implementation of the methodology of specialized schools in foreign language teaching.

Keywords: API REST, Framework, Methodology, Teaching, Content.

Lista de ilustrações

Figura 1 – Tela inicial do Duolingo.	19
Figura 2 – Tela inicial do Babbel.	20
Figura 3 – Tela inicial do Bussu.	21
Figura 4 – Tela inicial do LinguaLeo.	22
Figura 5 – Popularidade do termo Angularjs nas pesquisas do <i>Google</i>	25
Figura 6 – Desempenho de novas versões do Angular em relação aos seus concorrentes.	26
Figura 7 – Arquitetura do Angular.	27
Figura 8 – TypeScript engloba tudo de novo do ECMAScript e adiciona as tipagens.	28
Figura 9 – Diagrama de uma aplicação híbrida.	31
Figura 10 – Comunicação com API.	38
Figura 11 – Interface do Visual Studio Code.	39
Figura 12 – Interface do GitLab.	39
Figura 13 – Tela de <i>login</i> da aplicação.	41
Figura 14 – Tela inicial do Estudante.	42
Figura 15 – Tela de diálogo com ilustrações.	43
Figura 16 – Tela de diálogo em lista.	43
Figura 17 – Tela do questionário.	44
Figura 18 – Tela inicial do administrador	45
Figura 19 – Tela de usuários.	45
Figura 20 – Tela de diálogos.	46
Figura 21 – Tela de cadastro do diálogo.	46
Figura 22 – Tela de momentos.	47
Figura 23 – Tela de empresas.	47
Figura 24 – Tela de contratos.	48

Lista de tabelas

Tabela 1 – Comparação das aplicações apresentadas com o <i>Language Adviser</i>	23
Tabela 2 – Principais Comandos do Angular CLI.	29
Tabela 3 – Métodos HTTP.	34
Tabela 4 – Códigos de estado do HTTP.	34
Tabela 5 – Exemplificação das rotas de uma API REST.	40

Lista de gráficos

Gráfico 1 – Interface gráfica.	51
Gráfico 2 – Usabilidade do <i>software</i>	51
Gráfico 3 – Eficiência da ferramenta.	52
Gráfico 4 – Frequência de erros da interface.	52
Gráfico 5 – Utilidade do software.	53
Gráfico 6 – Contribuição ao aprendizado.	53
Gráfico 7 – Práticas dos diálogos.	54

Lista de abreviaturas e siglas

API	Application Programming Interface
CRUD	Create, Read, Update, Delete
CSS	Cascading Style Sheets
DOM	Document Object Model
HTTP	Hypertext Transfer Protocol
HTML	HyperText Markup Language
REST	Representational State Transfer
TDIC	Tecnologias Digitais da Informação e Comunicação
TI	Tecnologia da Informação
SPA	Single Page Application
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
VCS	Version Control System

Sumário

1	INTRODUÇÃO	15
1.1	Contextualização	15
1.2	Problemática	16
1.3	Objetivos	17
1.4	Justificativa	18
1.5	Metodologia	18
1.6	Estrutura do Trabalho	18
2	TRABALHOS RELACIONADOS	19
2.1	Duolingo	19
2.2	Babbel	20
2.3	Busuu	21
2.4	LinguaLeo	22
3	REFERENCIAL TEÓRICO	24
3.1	Node.js	24
3.2	Aplicação web	25
3.2.1	Angular	25
3.2.1.1	Angular CLI	29
3.2.1.2	Single Page Application (SPA)	29
3.3	Aplicação híbridas	30
3.3.1	Apache Cordova	31
3.3.2	Ionic	32
3.3.2.1	Ionic CLI	32
3.4	Representação por Transferência de Estado (REST)	33
3.4.1	Protocolo de Transferência de Hipertexto (HTTP)	34
4	LANGUAGE ADVISER	35
4.1	Visão Geral	35
4.2	Aplicação	36
4.3	Ferramentas auxiliares	38
4.4	Implementação	40
4.4.1	Funcionalidades do <i>software</i>	41
4.4.1.1	Funcionalidades do Estudante	42
4.4.1.2	Funcionalidades do Administrador	44

5	PROVA DE CONCEITO	49
5.1	Ecosistema de trabalho	49
5.2	Metodologia aplicada pelas instituições de ensino	49
5.3	Aplicação do software no meio acadêmico	50
5.4	Resultados obtidos	50
6	CONSIDERAÇÕES FINAIS	55
	REFERÊNCIAS	57
	APÊNDICE A – CÓDIGO FONTE DE UM SERVIÇO	60
	APÊNDICE B – QUESTIONÁRIO DE AVALIAÇÃO DE SOFTWARE	62
	ANEXO A – RECURSOS DA APLICAÇÃO	64

1 Introdução

1.1 Contextualização

A popularização das tecnologias digitais da informação e comunicação (TDICs) permite proporcionar diferentes práticas sociais e meios de comunicação (VILAÇA; ARAUJO, 2006). Com isso a *web* tornou-se o principal veículo para prestação de serviços computacionais, atingindo um número cada vez maior e diversificado de usuários, em razão do rápido crescimento da internet (TURQUETTE, 2003). Dessa forma, segundo Pressman (2009), a demanda por aplicações *web* vem crescendo rapidamente para atender as dificuldades e as necessidades das pessoas e organizações. Essa expansão pode ser percebida não somente em relação à quantidade e diversidade, mas também no nível de complexidade das aplicações.

VILAÇA e ARAUJO (2006) afirmam que as mídias digitais, principalmente a internet, deixam de ser exclusivas do computador *desktop* e passam a ocupar outros espaços, como: escolas, ruas, praças, bancos, restaurantes, entre outros. Os dispositivos móveis vêm ganhando espaço e importância em diferentes áreas da sociedade como: educação, saúde e entretenimento. O aumento do poder computacional destes aparelhos é um dos principais fatores que contribuem para a inserção de tecnologias da informação (TI) no cotidiano das pessoas (CARLOS, 2016).

Nesse sentido, surgem novidades no mercado tecnológico, principalmente na área educacional, ferramentas tecnológicas vêm integrando-se às salas de aula, trazendo um leque de novos recursos didáticos, tais como: computadores, lousas interativas, *tablets*, entre outros e conseqüentemente, novas práticas e métodos educacionais (GONÇALVES; SILVA, 2014). Contudo, é necessário que se promova uma pedagogia interativa e inovadora, utilizando os princípios da *e-learning* para possibilitar a auto-aprendizagem através de um ensino a distância por meio de dispositivos computacionais, assim é possível ter acesso ao conteúdo através do computador conectado à internet, porém o acesso a este, em alguns lugares acaba tornando-se inviável pela ausência de mobilidade.

Assim, para proporcionar um ensino à distância interativo e dinâmico, tem-se aderido à comunicação móvel via celular, a chamada *Mobile Learning (m-learning)*, aprendizagem com mobilidade. A *m-learning* faz uso das tecnologias de redes sem fio e dos novos recursos fornecidos pela telefonia celular para dar um acesso contínuo ao processo de ensino e aprendizagem (FERNANDO; DIEGO; BOLFE, 2013). Dessa forma, é possível estender os princípios da *e-learning* para o mundo *mobile*, sendo possível aprender em qualquer situação, com os novos métodos educacionais que a *m-learning* proporciona.

Dentre as existentes formas e áreas de ensino, a necessidade de aprender uma

língua estrangeira ou uma segunda língua existe, de uma maneira bastante significativa. Em consequência, foram desenvolvidas várias formas de alcançar um melhor resultado no processo de ensino e aprendizagem de língua estrangeira. Em outras palavras, um modo que alcançasse um resultado satisfatório, que capacitasse o aprendiz a se comunicar efetivamente na língua-alvo (PEDREIRO, 2013).

Desta forma, o ensino de línguas vem sendo destaque entre os *softwares* educativos. Isso devido ao fato da liberdade e facilidade de acesso a conhecimentos de línguas, os quais despertam novas metodologias de aprendizagem e práticas gerenciáveis pelo aluno, que são adaptadas às suas necessidades de estudos. Isso torna-se o diferencial para muitos alunos, já que as ferramentas tecnológicas facilitam e agilizam suas tarefas, seja através do computador ou do celular, por exemplo. A tecnologia educacional está em plena expansão e, nesse contexto, vem para somar com as metodologias de ensino eficientes que estão em funcionamento no espaço escolar (GONÇALVES; SILVA, 2014).

1.2 Problemática

Diante do exposto, surge o seguinte questionamento: como tornar o conteúdo disponibilizado pelas escolas de língua estrangeira atraente para os alunos, em diversos dispositivos?

Segundo JALIL e PROCAILO (2009), dentre as abordagens metodológicas de ensino da língua estrangeira, destacam-se:

- Tradicional: Nesse formato o aluno aprende a tradução literal de cada palavra, por meio da gramática normativa.
- Direta: Nesta abordagem o ensino é realizado por meio do contato direto com a língua, utilizando situações, imagens e demonstrações para facilitar o ensino.
- Audiolingual: Prioriza o desenvolvimento das habilidades orais, os diálogos são aprendidos por imitação e repetição.
- Comunicativa: Apresentando situações como em um restaurante, em casa ou por telefone. Além disso, trabalha com a comunicação entre os alunos nestas situações por meio da troca de suas experiências.

As abordagens metodológicas vão sempre ser um objeto de estudo, onde serão encontradas imperfeições. Dentre os métodos apresentados, é possível tirar algo proveitoso para o ensino da língua estrangeira. Contudo, os especialistas encontraram falhas nos métodos existentes, pois estes não levavam ao aprendizado satisfatório, uma vez que alguns focaram mais na gramática e outros mais na oralidade. Sendo assim, a busca por um

método eficaz não cessou. O método ideal deveria abordar as quatro habilidades básicas: falar, ouvir, escrever e ler (*speaking, listening, writing, reading*) com a mesma importância e relevância (SILVA; SCOVILLE, 2015).

Com o surgimento de novas teorias e concepções de língua, os métodos passam por modificações. As tecnologias e mídias digitais ao mesmo tempo que servem para distribuir informações, servem para reunir conteúdos em ambientes digitais abrigando os mais diversos tipos de materiais (KENSKI, 2007). Essas tecnologias estão cada vez mais interativas e móveis, a exemplo de algumas mídias digitais (*smartphones, tablets, notebooks*). Cabe à escola integrar as mídias na educação, pois, elas já estão presentes na vida social dos sujeitos, seja no trabalho ou lazer, nas esferas públicas ou privadas. As aulas de língua estrangeira integradas às diversas mídias podem tornar os materiais (texto autênticos, vídeos, áudios, *podcasts*, imagens, etc.) bem mais atrativos ao aluno (MARIA; RAPHAELLA, 2012).

1.3 Objetivos

Com base nestas informações anteriormente citadas e com o intuito de facilitar o trabalho dos educadores, surge uma oportunidade de desenvolver uma ferramenta que colabore com a metodologia das escolas de línguas. Com base nisto, o objetivo geral deste trabalho é a implementação de um *software* multiplataforma que implemente os conceitos de interoperabilidade, para construção de uma ferramenta que utilize tecnologias híbridas, dispondo de um mecanismo de ensino para os alunos que almejam iniciar, reiniciar, ou dar continuidade ao estudo de uma determinada língua estrangeira, utilizando o próprio conteúdo das escolas, dispostas a utilizar a ferramenta, para auxiliar o ensino/aprendizado de seus alunos.

Os objetivos específicos do trabalho constituem-se em:

- Ofertar uma interface *web* e *mobile* para os usuários;
- Disponibilizar conteúdo digital para os alunos;
- Contribuir para que metodologia da escola seja executada;
- Avaliar a contribuição metodológica no ensino do *Language Adviser* aos alunos;
- Permitir a comunicação do *software* com uma API REST para o processamento de dados.

1.4 Justificativa

A ideia deste trabalho justifica-se por acreditar que as instituições de ensino devem aproveitar ao máximo as tecnologia digitais na sala de aula. Para isso, é preciso expandir as aplicações *web* para os dispositivos móveis, que fazem parte do cotidiano da população, e que vem ganhando espaço e importância em diferentes áreas da sociedade. Nesse contexto, é necessário que as aplicações sejam desenvolvidas para as mais diversas plataformas, tais como: *Android*, *iOS*, *Windows Phone*. Diante disso, uma solução para tal problema são as aplicações multiplataforma que criam aplicativos utilizando os conceitos da programação *web*. Aplicações multiplataforma, são *softwares* desenvolvidos em uma linguagem abarcada por diversas plataformas, sem a complexidade de criar uma nova aplicação para os mais diversos aparelhos.

1.5 Metodologia

Diante dos fatos abordados, para a elaboração deste trabalho, foi realizado um levantamento das tecnologias disponíveis no mercado, que oferecem didáticas atrativas de apoio ao aluno que almeja aprender uma determinada língua estrangeira. Com isso, foi realizado uma análise das funções e características que estas tecnologias oferecem para a construção e desenvolvimento de uma plataforma que agrega recursos e funções que atendam a demanda destes alunos.

Para a construção da ferramenta foi necessário o consumo de uma API REST que oferece as funções necessárias para o desenvolvimento do *software*, que se fez necessário a utilização de *frameworks* atuais para facilitar e agilizar a fabricação do produto. Todas as tecnologias necessárias foram listadas e exemplificadas neste trabalho, com o objetivo de auxiliar o seu entendimento. Dentre estas, destacam-se os *frameworks* Angular e Ionic. Para a validação foi disponibilizado um questionário para os alunos que utilizaram o *software* durante um período de 4 meses, em uma instituição de ensino de língua estrangeira na cidade de Mossoró - RN e outra em Goiânia - GO, que utilizam a ferramenta como apoio de sua metodologia, facilitando o ensino/aprendizado de seus alunos.

1.6 Estrutura do Trabalho

O trabalho está estruturado da seguinte forma: o Capítulo 2 apresenta os trabalhos relacionados; o Capítulo 3 o referencial teórico; o Capítulo 4 traz uma visão geral do *Language Adviser*, a metodologia utilizada e as ferramentas para seu desenvolvimento; o Capítulo 5 descreve como o *software* foi implementando em um ambiente real de trabalho. No Capítulo 6 estão as considerações finais e perspectivas para trabalhos futuros.

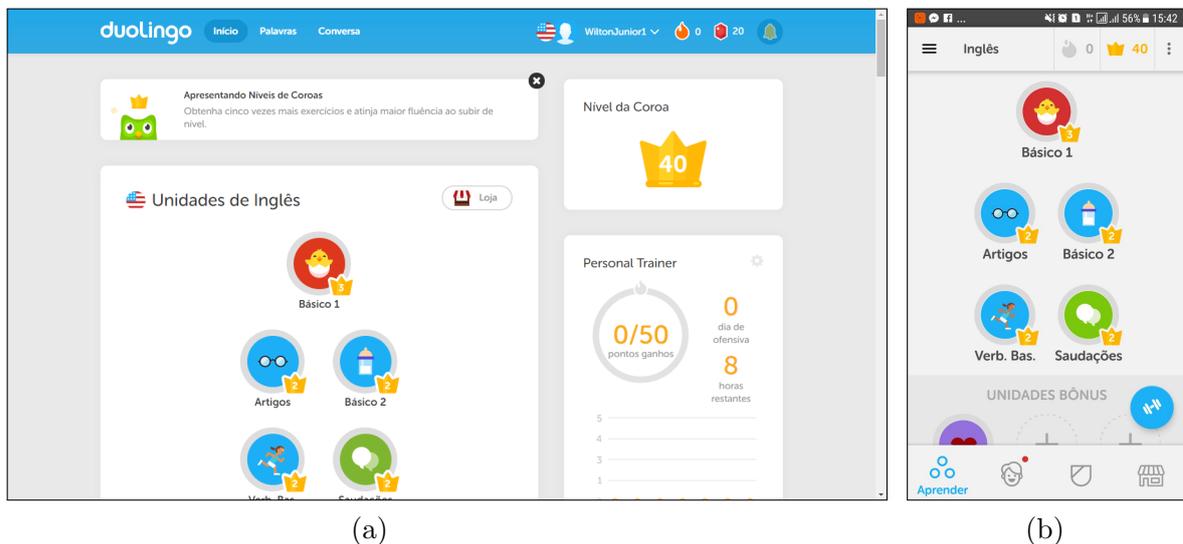
2 Trabalhos Relacionados

Este capítulo exemplifica os *softwares* que disponibilizam soluções semelhantes ao que proposto nesse trabalho. Dessa forma, foi realizado um levantamento das plataformas disponíveis no mercado que atende as necessidades dos usuários que almejam aprender uma determinada língua estrangeira.

2.1 Duolingo

O Duolingo¹ é uma plataforma que disponibiliza o ensino de idiomas de forma gratuita, através do aprendizado individual. O aplicativo possibilita treinar a escrita, fala, audição e leitura do usuário. É uma ferramenta gamificada que possui níveis que vão avançando e, conseqüentemente, aumentando o grau de dificuldade das lições, proporcionando a aprendizagem dos alunos. O *software* oferece um sistema para auxiliar os professores, chamado de Duolingo para Escolas, onde o mesmo dispõe um painel para acompanhar o progresso dos estudantes. Assim os professores o utilizam como ferramenta em sala de aula para conduzir o aprendizado dos alunos.

Figura 1 – Tela inicial do Duolingo.



Fonte: (DUOLINGO, 2018).

A plataforma fornece uma aplicação *web*, e outra *mobile*, que provê mobilidade para os usuários. A Figura 1, exemplifica os dois modelos da aplicação que o Duolingo

¹ Site oficial do Duolingo: <https://pt.duolingo.com/>

dispõe. A Figura 1a mostra a tela inicial da aplicação *web* e a Figura 1b expõe o mesmo tema em uma aplicação *mobile*.

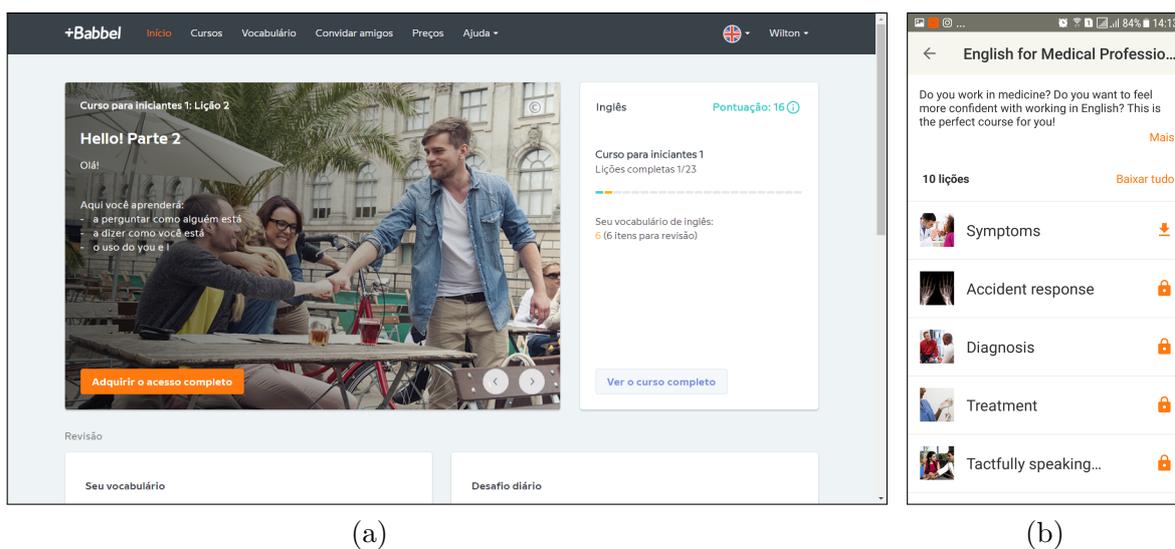
2.2 Babel

O Babel² é uma aplicação gratuita que permite aos usuários aprender um novo idioma de forma intuitiva. Todos os diálogos são gravados com falantes nativos. Isso permite compreender e aprender o idioma falado no dia a dia. A ferramenta possibilita treinar o vocabulário através de reconhecimento de voz. Mesmo tendo foco em conteúdos pagos, o *software* permite baixar os materiais de forma limitada. A ferramenta provê um aplicativo para dispositivos móveis, possibilitando uma maior versatilidade os usuários.

Segundo Sabota e Peixoto (2016), o aplicativo Babel dispõe de tecnologia e métodos de ensino que objetivam simplificar o aprendizado do idioma. Uma limitação do aplicativo é o fato de não ser permitido ao usuário revisar um item dentro do exercício, a não ser que seja feito um outro acesso.

A Figura 2a apresenta a tela inicial do Babel, disponibilizando algumas opções na barra inicial como, VOCABULÁRIO, CURSOS, entre outros, ao centro a interface disponibiliza informações sobre o curso e alguns atalhos para listar todos os cursos disponíveis da plataforma. A Figura 2b ilustra a experiência *mobile*, que a aplicação fornece, disponibilizando algumas lições gratuitas para baixar.

Figura 2 – Tela inicial do Babel.



Fonte: (BABEL, 2018).

² Site oficial do Babel: <https://pt.babel.com/>

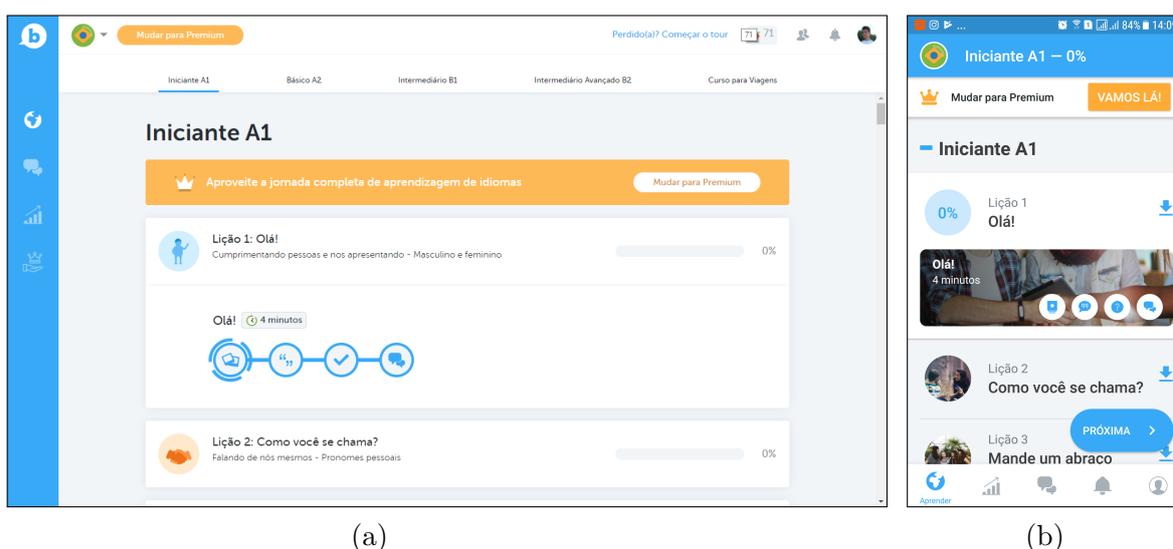
2.3 Busuu

Trata-se de uma rede social para aprendizagem de línguas criada em 2008, sendo considerado a maior rede social para aprendizagem de línguas. O Busuu³ oferece cursos gratuitos ou mediante pagamento, conforme o interesse do aprendiz. As atividades consistem em questões de múltipla escolha, exercícios de escrita e fala, leitura e compreensão. Seguindo o formato de uma rede social, os usuários enviam e recebem convites de amizade para interagir de modo síncrono ou assíncrono (SABOTA; PEIXOTO, 2016).

Este aplicativo é uma boa opção para quem está iniciando seus estudos de inglês e precisa exercitar vocabulário e gramática. O Busuu está disponível para *web* e *mobile*, permitindo maior mobilidade entre os usuários, a plataforma dispõe que as organizações obtenham acesso aos cursos *premium* do Busuu e monitorem o progresso dos alunos ao longo do tempo. Os educadores podem criar salas de aula, convidar alunos, acessar o programa de ensino completo e acompanhar seu progresso em um painel interativo.

A Figura 3a apresenta a interface *web* do Busuu, que dispõe opções do iniciante ao intermediário, divididas em categorias, para a prática do idioma, ao centro a interface dispõe de lições de acordo com a categoria selecionada para dar o andamento ao curso. A Figura 3b expõe o *layout* da versão *mobile* do Busuu. Percebe-se que a interface do aplicativo é bem distinta da versão *web*, sendo composta pelas mesmas opções da aplicação *web*, de forma compactada.

Figura 3 – Tela inicial do Bussu.



Fonte: (BUSUU, 2018).

³ Site oficial do Bussu: <https://www.busuu.com/pt/>

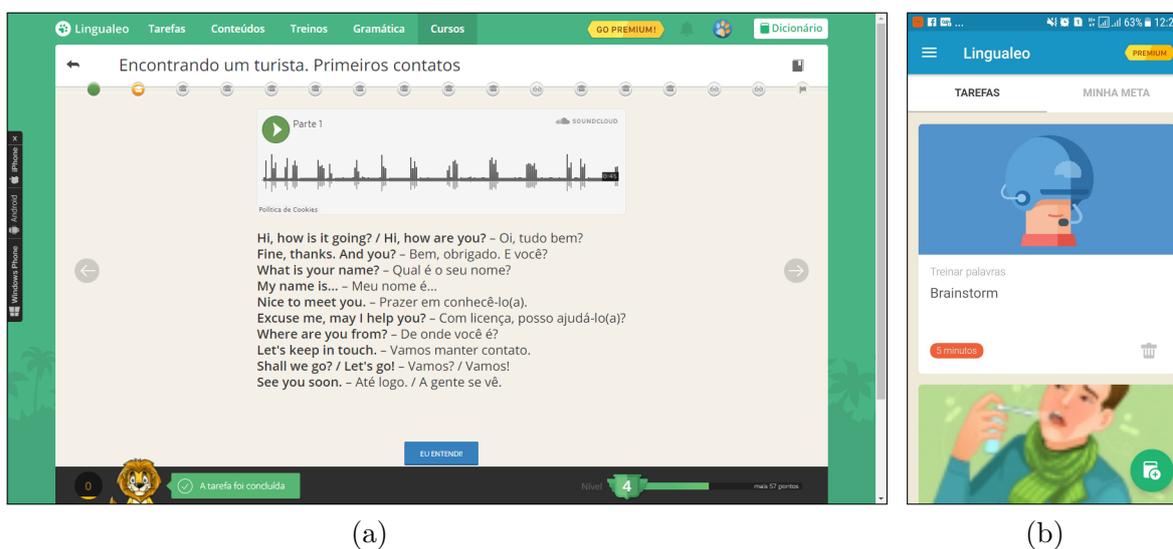
2.4 LinguaLeo

LinguaLeo⁴ é uma plataforma *freemium* que oferece serviço de aprendizagem da língua inglesa com lições interativas acompanhadas da tradução, imagens, som e a pronúncia fonética de palavras e expressões. Seguindo um instrutor representado por um simpático leão, os usuários irão explorar e aprender um enorme vocabulário de palavras em inglês (RIBEIRO, 2013).

Todos podem usar a versão básica do serviço Lingualeo gratuitamente ou adquirir o serviço *premium*, que abrange os recursos disponíveis da plataforma. O Lingualeo motiva seus usuários utilizando mecânicas de jogos combinadas com um plano de estudos personalizado, mantendo interação com outros estudantes via rede social, para adaptar as metas de estudos com o estilo de vida de cada estudante. (LINGUALEO, 2018).

A Figura 4a apresenta a interface *web* das lições do curso Lingualeo. Ela é formulada por opções como: Tarefas, Conteúdos, Gramática, Cursos, entre outras. Essas opções, disponibiliza benefícios para deixar a plataforma autônoma para o ensino de língua estrangeira. Ao centro a interface apresenta um diálogo listados por todos os seus fragmentos, seguidos pela tradução já exposta. A plataforma ainda traz questões a respeito do diálogo para facilitar o aprendizado dos alunos. A Figura 4b ilustra o *layout* da plataforma *mobile* do Lingualeo.

Figura 4 – Tela inicial do LinguaLeo.



Fonte: (LINGUALEO, 2018).

A Tabela 1 apresenta um comparativo entre os trabalhos relacionados que foram listados neste documento. A comparação se desenvolveu a partir dos principais atributos do *Language Adviser*, tais como: "Plataforma *web* e *mobile*", "Escutar trechos do diálogo

⁴ Site oficial do Lingualeo: <https://lingualeo.com/pt>

com a tradução", "Vocalizar diálogos", "Cadastrar conteúdo" e "Realizar marketing digital". Como se pode analisar, pelos atributos e funções discutidas, o *Language Adviser* se mostra uma alternativa interessante para o apoio do ensino de língua estrangeira para os mais diversos alunos.

Tabela 1 – Comparação das aplicações apresentadas com o *Language Adviser*.

Atributos	Duolingo	Babbel	Bussu	LinguaLeo	<i>Language Adviser</i>
Plataforma <i>web</i> e <i>mobile</i>	X	X	X	X	X
Escutar trechos do diálogo com a tradução				X	X
Vocalizar diálogos					X
Cadastrar conteúdo					X
Realizar marketing digital					X

Fonte – Autoria Própria.

3 Referencial Teórico

O presente capítulo tem por objetivo dar embasamento ao tema por meio da seleção de diversas tecnologias, que foram utilizadas para a criação e o desenvolvimento do projeto.

3.1 Node.js

Node.js é uma plataforma de desenvolvimento de aplicações *web*, com o objetivo de proporcionar uma forma fácil de construir aplicações rápidas e escaláveis em um modelo baseado em eventos ou assíncrono (JUNIOR, 2012). Trata-se de um ambiente de desenvolvimento em JavaScript, criado por Ryan Dahl em 2009, maioritariamente implementada em C e C++ que faz uso da *Engine V8* do *Google* para executar código JavaScript no lado do servidor.

Segundo Silva (2017) o Node.js foi criado com objetivo de resolver os problemas das arquiteturas bloqueantes. Esta tecnologia utiliza operações de I/O (*Input* e *Output*) não bloqueantes, apresentando uma boa performance com um baixo consumo de memória. Esta técnica, visa o suporte de processos de longa duração e utiliza ao máximo, e de forma eficiente, o poder de processamento dos servidores, principalmente em sistemas que produzem uma alta carga de processamento. Já uma arquitetura bloqueante, enfileira as requisições e posteriormente processa, uma por uma, não possibilitando o processamento de várias delas ao mesmo tempo. Por exemplo, enquanto requisição está executando um I/O, o processador mantém-se ocioso esperando a requisição finalizar.

De acordo com Severance (2012), a partir de projetos Node.js tornou-se possível utilizar o JavaScript como linguagem para criação de aplicações do lado do servidor, devido a linguagem ser orientada a eventos, a criação de aplicativos para *web* altamente escaláveis usando JavaScript sem gerenciar as complexidades do *multithreading* tornou-se bastante natural. Segundo Abernethy (2011) o Node.js não permite que *deadlocks* aconteçam, já que bloqueios não serão permitidos, pois o mesmo foi projetado para suportar um grande volume de tráfego e dezenas de milhares de conexões simultâneas, mas essa afirmação não afeta diretamente os bloqueios das chamadas de entrada e saída.

O Node ainda fornece o conjunto de módulos chamado de *Node Package Manager* (NPM). Com ele é possível fazer a integração de um vasto conjunto de pacotes e módulos externos desenvolvidos por uma grande comunidade de desenvolvimento e de apoio que disponibiliza-os de modo público (SOUSA, 2015). Com isso torna-se possível fazer a integração de uma aplicação com um conjunto de pequenos módulos, que são mantidos pelos seus criadores.

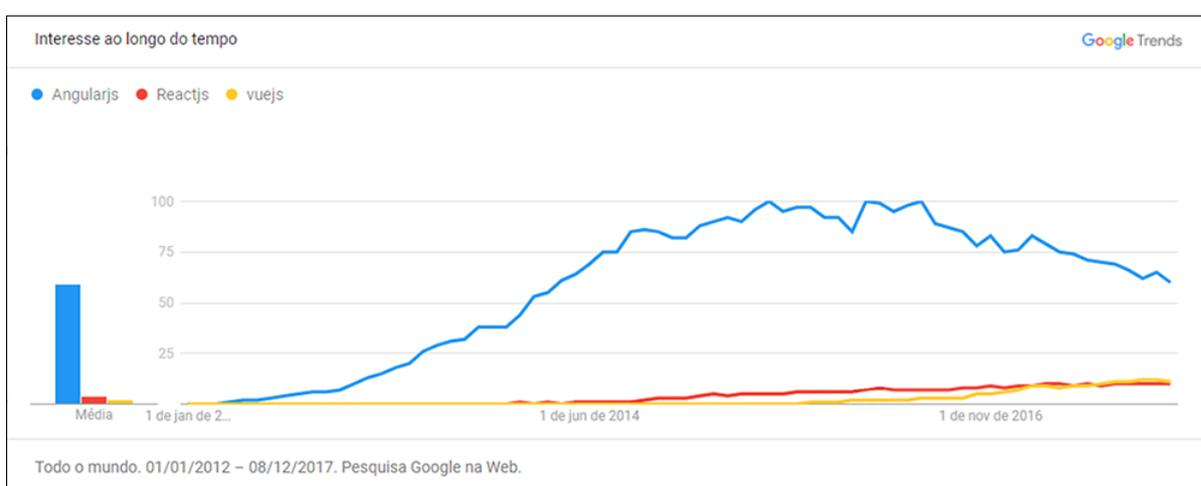
3.2 Aplicação web

A internet passou por várias mudanças significativas, dentre elas desta-se as aplicações *web* que se tornam maiores e melhores, em contra partida, a complexidade a ser administrada por um desenvolvedor foi ampliada consideravelmente. Uma solução *JavaScript* pura, nem sempre tinha a estrutura adequada para garantir uma alta velocidade no desenvolvimento, ou a possibilidade de manutenção em longo prazo (GREEN, 2014). O Angular surgiu para resolver esses problemas e atender as necessidades do desenvolvedor, acelerando o processo de desenvolvimento, implementando conceitos da engenharia de *software* e deixando as aplicações interativas.

3.2.1 Angular

O Angular é um *framework* desenvolvido e mantido pelos programadores da *Google* que utiliza do Nodejs para desenvolvimento *front-end*, simplificando a criação de aplicações *web* robustas, atualmente se encontra disponível em sua versão 6. Por este motivo, tem-se mostrado em crescente ascensão, sendo considerada a tecnologia do momento para desenvolvimento de aplicações *web*. Uma pesquisa realizada no *Google Trends*, demonstra a popularidade do termo Angularjs comparado a Vuejs e Reactjs, que também são ferramentas que seguem o mesmo princípio do Angular e estão alta no mercado de trabalho.

Figura 5 – Popularidade do termo Angularjs nas pesquisas do *Google*.



Fonte: Autoria própria.

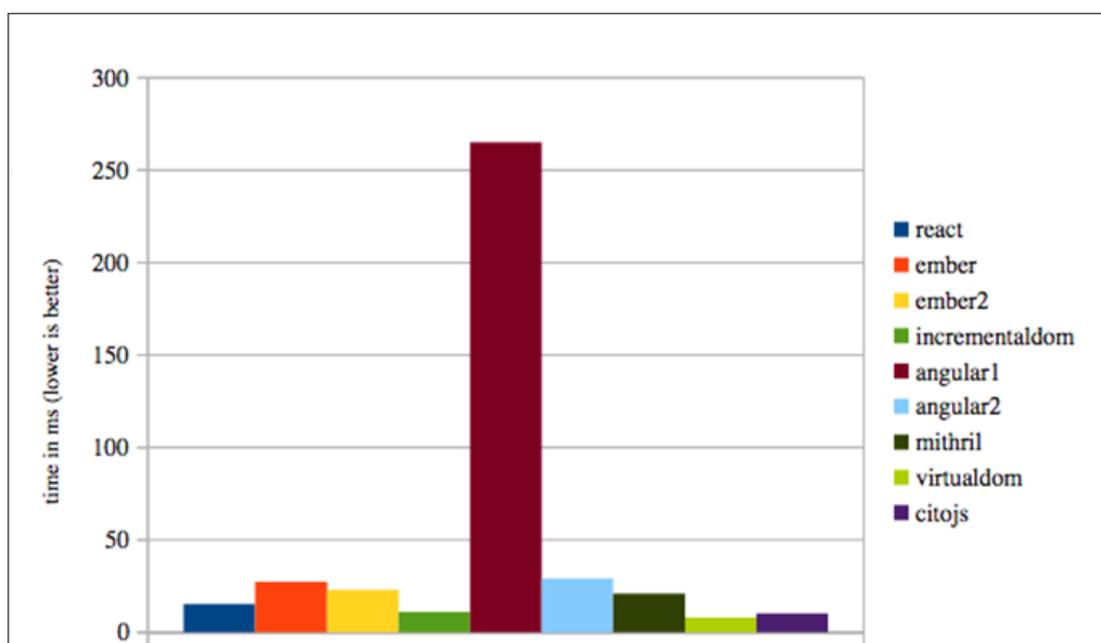
Como pode-se observar na Figura 5, o termo 'Angularjs' demonstrou ter um maior índice de popularidade nas pesquisas da *google* realizadas em todo o mundo no período de 2012 a 2017 em relação aos seus concorrentes: Vuejs e Reactjs que são *frameworks* que

obedecem o mesmo princípio do Angular, isso atrai a comunidade e as empresas para adotar essa tecnologia.

Angular combina uma série de ferramentas integradas para resolver desafios de desenvolvimento. O *framework* reutiliza código e técnicas para criar aplicativos para qualquer destino de implantação, seja *web*, *mobile* ou *desktop*. Esse *framework* oferece a produtividade e a infra-estrutura escalável que suporta as maiores aplicações da Google (ANGULAR, 2018).

A Figura 6 ilustra a evolução dos *frameworks* em performance, o ponto de destaque foi a evolução do Angular em relação às suas versões anteriores, de acordo com Guedes (2017), o Angular 1 ou AngularJs, de fato, oferece uma experiência de baixa performance devido às excessivas interações com o *Document Object Model* (DOM). O Novo Angular (Angular 2), com sua nova configuração, deixa a resposta muito mais rápida e a usabilidade dinâmica.

Figura 6 – Desempenho de novas versões do Angular em relação aos seus concorrentes.



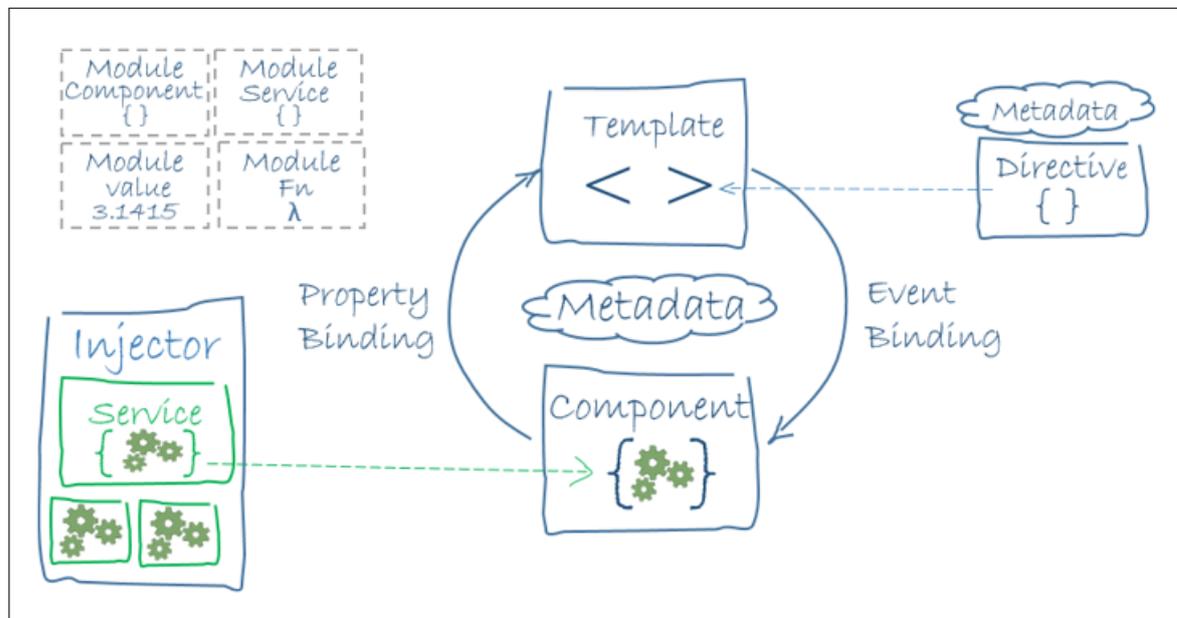
Fonte: (PEYROTT, 2016).

O Angular é encarregado pelo *front-end*, sendo a camada responsável pela comunicação com o usuário através do *browser*, por onde há entrada e saída de informação, dessa forma a arquitetura do novo Angular, nas suas versões atuais é formulada através de alguns conceitos essenciais, tais formulações ajudam o *framework* a chegar no patamar que se encontra hoje, sendo um dos mais populares para desenvolvimento *front-end*.

A figura 7 mostra a arquitetura do Angular. A estrutura é composta de várias

bibliotecas que cooperam entre si, algumas básicas e outras opcionais.

Figura 7 – Arquitetura do Angular.



Fonte: (ANGULAR, 2018).

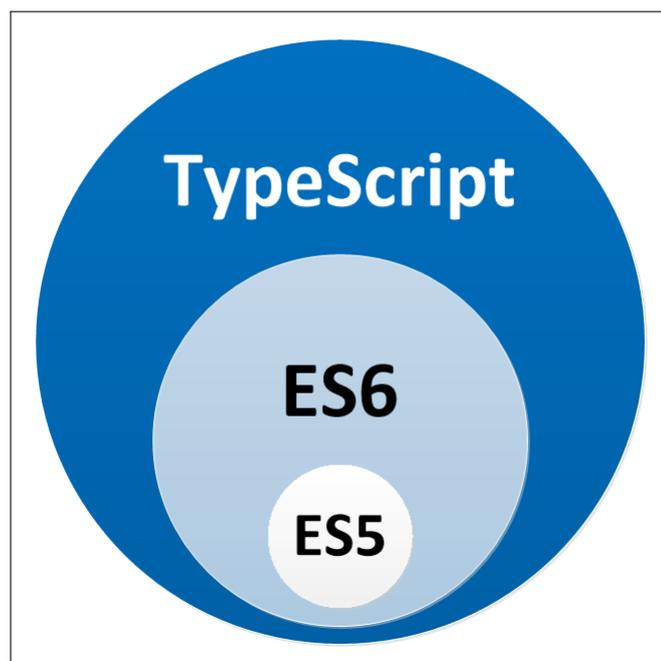
Para entender a sua arquitetura, é necessário compreender e analisar alguns conceitos básicos, tais conceitos são listados e formulados abaixo:

- *Module*:
 - O módulo é um mecanismo para agrupar todos os recursos relacionados a aplicação, como: componentes, diretivas, *pipes* e serviços.
- *Component*:
 - Componentes são uma parte crucial na construção da aplicação, tendo em vista que o Angular é “orientado a componentes”. Esses componentes criam uma comunicação entre os serviços e *templates* de determinada funcionalidade.
- *Template*:
 - São definidos para gerar uma *view* que possibilita o usuário visualizar dados dos componentes. O *template* é composto por HTML e diretivas, que dizem ao Angular como renderizar os dados
- *Directive*:

- As diretivas são usadas para alterar o DOM, estendendo as funcionalidades de um elemento.
- *Services*:
 - Serviços são classes que encapsulam algum tipo de funcionalidade e as fornece para um ou mais componentes por meio de injeção de dependência. Essa maneira de armazenar as funcionalidades promove o reuso de código na aplicação.

Sabendo disso é possível compreender os fundamentos teóricos e práticos da ferramenta, possibilitando o desenvolvimento de futuras aplicações utilizando o *framework* Angular para desenvolvimento *web* e *mobile* na construção de aplicações rápidas e escaláveis. Com este *framework* nas suas versões mais atuais, a codificação é realizada utilizando TypeScript que é um *superset*, um superconjunto JavaScript, como ilustra a Figura 8. De acordo com (GUEDES, 2017) o TypeScript ainda acrescenta sintaxe simplificada e uma variedade de ferramentas sobre a linguagem JavaScript. Ele traz o poder e a produtividade da tipagem estática e técnicas de desenvolvimento orientado a objetos. Isso traz uma série de benefícios para o desenvolvedor, utilizando os conceitos de programação orientada a objetos, que só o ECMAScript não oferece. O código TypeScript é compilado e convertido para ECMAScript 5. Por isso, pode ser utilizado normalmente em qualquer projeto JavaScript.

Figura 8 – TypeScript engloba tudo de novo do ECMAScript e adiciona as tipagens.



Fonte: (GUEDES, 2017).

3.2.1.1 Angular CLI

O Angular CLI (*Command Line Interface*) é uma interface por linha de comando desenvolvida pelos próprios criadores do Angular, surgindo a partir da versão 2 do *framework*. Essa ferramenta é fundamental na construção de um projeto em Angular, facilitando o processo de criação das aplicações, evitando erros e aumentando a produtividade. Esta interface é baseada no *webpack*, uma ferramenta que ajuda a processar e agrupar os diversos arquivos TypeScript, JavaScript, CSS, HTML e imagens. O *webpack* é responsável por estruturar o ponto de entrada da aplicação, ou seja, a raiz contextual (arquivo de inicialização) (ALURA, 2018).

O Angular CLI cria um novo projeto pronto para ser executado, a ferramenta elabora arquivos necessários para que tudo seja pronto em questão de minutos, gerando uma estrutura de aplicativos baseada nas melhores práticas para um projeto em Angular. Novos elementos podem ser inseridos no projeto, os quais são estruturas pre-definidas existentes no angular, como exemplo: *component*, *services* e *class*. Ao adicionar elementos criar-se uma estrutura de diretórios apropriada, em alguns casos, a ferramenta adicionará código a outros componentes dentro do projeto para que os novos elementos possam ser utilizados corretamente. A Tabela 2 exemplifica os principais comandos do Angular CLI.

Tabela 2 – Principais Comandos do Angular CLI.

Comando	DESCRIÇÃO
<i>ng new</i>	Cria um novo projeto.
<i>ng serve</i>	Inicializa o servidor.
<i>ng test</i>	Realiza o teste unitário de componentes.
<i>ng build</i>	Gerar uma aplicação para produção ou desenvolvimento.
<i>ng generate</i> ou <i>ng g</i>	Utilizado para criar componentes, serviços, classes e tudo que for utilizado no projeto.

Fonte – Autoria Própria.

3.2.1.2 Single Page Application (SPA)

O Angular é um *meta-framework* para *Single Page Application* (Aplicação de Página Única), que são aplicações desenvolvidas em JavaScript que rodam quase inteiramente no lado do cliente (*browser*), como qualquer outra aplicação para *desktop*. O Google foi o primeiro nesta tecnologia com o Gmail, e atualmente temos diversas aplicações que utilizam esse padrão (GUEDES, 2017).

Com a evolução do JavaScript, a *web* tem mudado nos últimos anos, aplicações de apenas uma página com grande parte de seu código no cliente em JavaScript, tem se tornado cada vez mais comum. Segundo Jadhav, Sawant e Deshmukh (2015) na era moderna da tecnologia *web* a maioria dos sites estão utilizando SPA, nesta arquitetura todos

os arquivos como CSS, imagens, *scripts* e outros recursos são carregados ao mesmo tempo na página inicial e os componentes da página são substituídos por outros componentes dinamicamente, dependendo da interação do usuário.

Segundo Ngoc (2014) uma SPA é composta de componentes individuais que podem ser substituídos de forma independente, sem a necessidade de recarregar a página a cada ação do usuário, proporcionando-lhe uma melhor experiência e performance. As aplicações que utilizam SPA, fazem uma transição dos componentes para o *template*, sem *reload* de página e sem que o usuário viaje de uma página para outra. De acordo com Mikowski e Powell (2013) páginas escritas em SPA minimizam o tempo de resposta para o usuário. Já que o *browser* apenas renderizar trechos de códigos ao cliente diminuindo assim o tráfego na rede e ganhando em performance para o usuário.

3.3 Aplicação híbridas

Atualmente, o mercado de dispositivos móveis tem uma grande variedade de sistemas operacionais (SO), os mais populares são Android e iOS, que compõem cerca de 96% do mercado de *software*, porém eles tem grandes diferenças quanto a forma de desenvolvimento das aplicações. (ALENCAR, 2015). Segundo dados da (INTELLIGENCE, 2018), já existem mais dispositivos *mobile* do que pessoas no planeta. A partir disso as empresas se preocupam mais em desenvolver aplicativos funcionais e úteis para seus clientes com o objetivo de expandir esse mercado cada vez mais.

Para desenvolver aplicações em dispositivos móveis é necessário uma série de conhecimentos específicos a respeito de cada SO. Neste contexto, empresas de *software* precisam de funcionários com competências em cada um dos diferentes sistemas operacionais que conheçam as particularidades das plataformas. Isto leva o desenvolvimento de aplicações móveis a se tornar caro e lento, com manutenções dispendiosas (WAHLBRINCK; BONIATI, 2017). As aplicações multiplataformas utilizam uma única linguagem de programação para se adaptada a diferentes plataformas, para se adequarem aos diversos aparelhos e sistemas operacionais. Dessa maneira, o desenvolvedor preocupa-se apenas com a aplicação e suas funções, trabalhando principalmente na usabilidade e em melhorias do aplicativo.

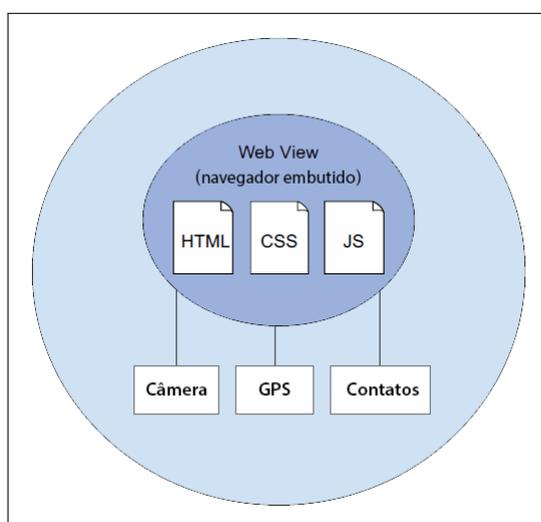
A solução mais comum atualmente para construir aplicativos multiplataforma é o Apache Cordova. Ele beneficia-se do ponto forte da *web* em ter linguagens padronizadas, utilizando o navegador como ambiente de execução para construir aplicativos, dispondo de recursos nativos se necessário, tudo escrito em HTML, CSS e JavaScript (LOPES, 2016). Dessa forma a programação multiplataforma se torna uma solução viável para o programador, reduzindo a complexidade da criação de aplicações para varias plataformas.

3.3.1 Apache Cordova

O Apache Cordova é um *framework* de código aberto desenvolvido pela *Apache Software Foundation* para a construção de aplicações móveis, utilizando os conceitos da programação *web*, para o desenvolvimento de aplicações multiplataformas ou híbridas. (WARGO, 2015). Ele permite utilizar as tecnologias *web* moderna como: HTML5, CSS3 e JavaScript, combinado com a tecnologia nativa do dispositivo para a criação de aplicações multiplataformas.

O Apache Cordova utiliza JavaScript para acessar as funcionalidades nativas do dispositivo. Essas funções são compreendidas através de uma API em JavaScript unificada, permitindo desenvolver facilmente um conjunto de códigos para atingir vários dispositivos com distintos SO (CORDOVA, 2017). O *framework* acessa os recursos do dispositivo para que a aplicação funcione no *browser* do aparelho, possibilitando a utilização de tecnologias no *front-end* como a diversidade de bibliotecas HTML, CSS e JavaScript que existem hoje, isso foi possível pois, assim que a aplicação é aberta pelo usuário, todo o código é compilado dentro da *webview*.

Figura 9 – Diagrama de uma aplicação híbrida.



Fonte: Adaptado de (CAMDEN, 2015).

Webview é o nome dado ao *browser* que executa a aplicação híbrida, contendo apenas o necessário para que o HTML, CSS e JavaScript funcionem, sendo uma parte desenvolvida em código nativo, e a outra feita de código não nativo (HTML, CSS e JavaScript) que implementa todo o visual e o comportamento da aplicação. Ela também consegue acessar recursos nativos do dispositivo como a câmera, GPS e contatos. Isso é possível graças a uma interface javascript, que torna a *webview* apta a executar código nativo nos dispositivos, como ilustra a Figura 9.

Aplicações híbridas são um tipo de aplicação *web* que estendem suas funcionalidades por meio de APIs de plataformas nativas de determinado dispositivo (GOK; KHANNA, 2013). Um dos grandes problemas das soluções híbridas é a perda de performance em relação às aplicações nativas, já que o código JavaScript não será compilado tão rapidamente como código nativo escrito no dispositivo.

3.3.2 Ionic

Ionic é um *framework front-end* SDK para o desenvolvimento de aplicações *mobile* híbridas, que permite aos desenvolvedores criar aplicativos móveis de alta performance e qualidade utilizando tecnologias *web* já conhecidas (HTML, CSS e JavaScript). O Ionic utiliza o Cordova para a construção da aplicação de um *webview*, capaz de renderizar as tecnologias previamente citadas. Ainda é focado principalmente na aparência, ou seja, na experiência do usuário com a interface de uma aplicação, através da construção de *layouts* poderosos e de simples desenvolvimento (WAHLBRINCK; BONIATI, 2017).

Ionic *Framework* é totalmente *open source* de modo que os desenvolvedores podem criar e publicar seus aplicativos para o mercado, sem qualquer custo, utilizando os princípios do Angular com o Apache Cordova, dessa forma o *framework* inclui um conjunto de ferramentas para facilitar a construção e o desenvolvimento de aplicações em grande escala e de nível empresarial, tornando-a compreensível para os colaboradores, para futuras alterações. O Ionic tem evoluído bastante ao longo dos anos, desde de sua criação com o desenvolvimento e melhoria contínua.

Ionic é modelado em SDKs para o desenvolvimento *mobile* nativo, facilitando o entendimento para qualquer pessoa que já tenha desenvolvido algo para Android ou iOS, utilizando o TypeScript para *plugins* Cordova que torna a adição de qualquer funcionalidade nativa que precisa-se para o aplicativo móvel, funcionando perfeitamente em todos os tipos de dispositivos.

3.3.2.1 Ionic CLI

A CLI do Ionic *framework* simplifica o processo de criação de aplicativos iônicos. As aplicações são desenvolvidas principalmente através do utilitário de linha de comando Ionic CLI, utilizando o Apache Cordova para criação de aplicativos nativos. A maioria das ferramentas na CLI é baseada no Node e é gerenciada pelo NPM.

O Ionic CLI dispõe de recursos para simplificar o desenvolvimento das aplicações *mobile*. A ferramenta oferece funcionalidades como compilar, executar e emular o *software*, passando pelos testes em *browser*, sem a necessidade de instalar a aplicação em um dispositivo. O *framework* contém alguns *templates* já prontos, em que todo código pode ser gerado com apenas um comando (GOIS, 2017).

3.4 Representação por Transferência de Estado (REST)

Com o surgimento de vários padrões para comunicação entre aplicações, nasce uma solução simples e inovadora chamada *Representational State Transfer* REST (Representação por Transferência de Estado). Esse padrão foi criado nos anos 2000 pela tese de doutorado de Roy Thomas Fielding, um dos principais autores da especificação *Hypertext Transfer Protocol* (HTTP). Em sua tese Fielding apresentou o REST como uma arquitetura de *software*, com o foco na criação de serviços *web* e foi desenvolvida para sistemas de hipermídia distribuídos, descrevendo os princípios de engenharia de *software* (FIELDING, 2000).

Seguindo os conceitos de praticidade do protocolo HTTP, REST surge como uma nova opção para problemas de integração nos projetos de *software*. O REST foi concebido baseando-se em HTTP, até pela formação de Fielding que foi um dos principais autores da especificação do HTTP, a arquitetura de um sistema REST geralmente é cliente-servidor e os serviços não possuem estado (*stateless*) (ROBERTO, 2015).

O REST trata qualquer tipo de serviço ou informação (páginas, vídeos e documentos) como um recurso. Os recursos são identificados por um Identificador Uniforme de Recurso do inglês *Uniform Resource Identifier* (URI) que são acessados por meio do protocolo HTTP (SILVA, 2017).

URI é o nome e o endereço de um recurso. Se uma informação não tiver uma URI, não será vinculado a nenhum recurso e conseqüentemente não estará disponível na *web* (RICHARDSON; RUBY, 2008). A URI tem a atribuição de identificar e localizar um recurso globalmente. Ao ser identificado por uma URI, um recurso pode ser acessado remotamente em qualquer local do mundo.

A URI é constituída por uma *Uniform Resource Locator* (URL) ou *Uniform Resource Name* (URN), URL é a forma mais conhecida de localizar um recurso na internet, já a URN identifica um recurso por *namespace* particular de forma persistente e independente da sua localização. Os programas clientes utilizam uma Interface de Programação de Aplicativos do inglês *Application Programming Interface* (API) para comunicar-se com os serviços *web*.

Uma API trata-se de um conjunto de rotinas e padrões estabelecidos e documentados por uma aplicação, para que outras aplicações consigam utilizar as suas funcionalidades, sem precisar conhecer detalhes da implementação do *software*. Desta forma, entendemos que as APIs permitem uma interoperabilidade dos sistemas. De um modo geral, uma API expõe um conjunto de dados e funções para facilitar as interações entre os programas de computador, permitindo-lhes a trocar informações (MASSE, 2011).

3.4.1 Protocolo de Transferência de Hipertexto (HTTP)

O HTTP é uma aplicação para informações distribuídas, colaborativas e de sistemas de hipermídia. Desta forma não é surpreendente que o estilo REST é idealmente concebido para uso do HTTP, sendo um dos protocolos mais utilizados do mundo (SAUDATE, 2013). De acordo com Masse (2011) uma API REST abranger todos os aspectos do protocolo de transferência de hipertexto, incluindo os seus métodos de solicitação, códigos de resposta, e os cabeçalhos de mensagem.

O HTTP possui diversos métodos, são eles: GET, POST, PUT, DELETE. Esses métodos permitem que se obtenha uma interface uniforme. Associado a estes métodos tem-se a necessidade de criar de aplicativo de negócios padrão, mais conhecido por CRUD (*Create, Read, Update, Delete*), ou seja, armazenamento, busca, atualização e deleção (DEVMEDIA, 2017). Uma API REST mapeia as operações do CRUD para métodos HTTP. A tabela 3 especifica qual método HTTP mapeia a qual operação.

Tabela 3 – Métodos HTTP.

OPERAÇÃO CRUD	MÉTODO HTTP	DESCRIÇÃO
<i>Read</i>	GET	Obtém um recurso existente.
<i>Create</i>	POST	Cria um novo recurso.
<i>Update</i>	PUT	Atualiza um recurso.
<i>Delete</i>	DELETE	Deleta um recurso.

Fonte – Autoria Própria.

O REST também utiliza outro recurso do protocolo HTTP que são os *status codes*. Eles são padrões de retorno quando alguma requisição, independente do método é solicitada. De acordo com a RFC2616 (1999) o *status code* é um código inteiro de 3 dígitos que entender e satisfaz o pedido. Os códigos de estado estão divididos em cinco categorias baseado em seus significados, como ilustra a tabela 4.

Tabela 4 – Códigos de estado do HTTP.

CATEGORIA	DESCRIÇÃO
1xx:Informativa	Comunica informações do protocolo de transferência.
2xx:Sucesso	Indica que a requisição do cliente foi aceita com êxito.
3xx:Redirecionamento	Indica que o cliente deve tomar alguma ação adicional para completar seu pedido.
4xx:Erro de Cliente	Indica os casos em que o cliente pode ter cometido algum erro.
5xx:Erro de Servidor	Indica um erro do servidor ao processar a requisição.

Fonte – (MASSE, 2011).

4 Language Adviser

Este capítulo é destinado à apresentação da aplicação *Language Adviser*. Estão apresentadas a visão geral do sistema, a aplicação, as ferramentas que foram utilizadas para desenvolvê-lo e sua implementação.

4.1 Visão Geral

O *Language Adviser* é uma alternativa para estabelecer o *marketing* digital, através do ensino de idiomas, promovendo publicidade para seus associados. O *software* foi desenvolvido de forma que atenda todos os requisitos para facilitar o ensino e aprendizado de um novo idioma, sendo também uma grande ferramenta no processo de comunicação e persuasão dos consumidores.

O *software* é uma plataforma onde instituições especializadas no ensino de idiomas irão dispor de conteúdo próprio para alimentá-la, com a presença de recursos que tornam o ensino e aprendizado atrativo e dinâmico aos alunos. A inserção de conteúdo, que é realizada pelo *software*, permite que no futuro escolas possam alterar ou adicionar novos conteúdos para os seus estudantes, sem ter que gastar esforços e recursos com a atualização do material disponibilizados aos alunos.

Uma das funcionalidades da plataforma é realizar *marketing* explícito como forma de captar recursos. Empresas associadas às escolas podem contratar termos presentes em seus materiais, para que estas se associam a eles. Quando uma organização adquire um termo, o nome desta estará relacionado com o mesmo, toda vez que o usuário for praticar uma lição e esse termo estiver presente, o nome da organização estará logo após.

Se uma escola possuir filiais espalhadas pelo país, elas tem a opção de comercializar um único termo para várias empresas distintas, diferenciando a região de atuação, ou seja, em que parte do mundo a empresa quer que o seu nome esteja relacionado com o termo desejado. A empresa pode definir um país inteiro, um estado ou uma única cidade como região de atuação. O *Language Adviser* usa o conceito de geolocalização para identificar em que localização o usuário se encontra no momento do acesso à plataforma, para definir os contratos ativos nesta região e fazer todo o processo de busca de termos, concatenando os nomes das empresas aos termos selecionados.

Um contrato para o *Language Adviser* é a confirmação da aquisição de um ou mais termos realizado por uma empresa com uma instituição detentora dos direitos do material vigente na plataforma. No contrato estará presente além dos termos adquiridos, a região em que esses termos estão vinculados e uma data de início e de término do contrato. Os

termos vão estar associados a uma empresa até a data de vencimento que foi acordado. Após o vencimento da data de término esses termos estarão livres para um novo contrato.

O *Language Adviser* é uma ferramenta com fundamentos de gerenciamento complexos, mas que também possui funcionalidades de interação com os usuários. Eles podem avaliar as lições propostas na plataforma, com o intuito de obter um *feedback* do material disponibilizado para a escola e propor a criação de novos materiais que agradam mais os estudantes. A plataforma adapta-se às necessidades das instituições de ensino e o *software* é alimentado através do material disponibilizado pela própria escola, possibilitando auxiliar o ensino e aprendizado dos seus alunos.

A inserção de conteúdo é realizada pela interface administrativa que a ferramenta dispõe, permitindo que no futuro essas escolas possam alterar ou adicionar novos conteúdos para os seus alunos.

4.2 Aplicação

Dentre a abordagem geral do *Language Adviser*, este trabalho tem como objetivo avaliar a ferramenta como mecanismo de apoio às instituições de ensino, facilitando a sua prática metodológica e promovendo conteúdo para a plataforma, auxiliando o ensino dos alunos, tendo em vista que, as novas tecnologias estão mudando cada vez mais o cotidiano e a maneira de pensar da sociedade.

Essa constatação provoca mudanças no ensino de maneira geral, em especial em línguas. Sabendo disso, o *Language Adviser* surge como uma solução para facilitar a aprendizagem do aluno em um determinado idioma. A plataforma permite que a própria instituição de ensino promova seu conteúdo, por meio de uma interface administrativa que o *Language Adviser* oferece.

Tendo em vista que, umas das principais dificuldade das escolas de línguas é a atualização constante de conteúdo, gerando custos de impressão do material, que em sua grande maioria são livros acompanhados de CDs de áudio, contendo todo o conteúdo da instituição. Dessa forma o *Language Adviser* permite a redução de custos associados a atualização de material.

A plataforma oferece uma área exclusiva para o estudante revisar as lições propostas pela escola, facilitando o estudo dos diálogos. O aluno pode estudá-los com suas respectivas traduções, podendo responder às questões que estão disponíveis na plataforma referente a lição. O *Language Adviser* é um *software* que visa apoiar a escola e os estudantes que pretendem estudar uma determinada língua estrangeira. Os principais requisitos da plataforma foram divididos por categoria, da seguinte forma:

- Administrador
 - Interface *web* intuitiva para as instituições de ensino;
 - Preenchimento de formulários através de uma interface dinâmica;
 - Tratamento individual dos dados armazenados e processados na API;
 - Atualização constante de conteúdo, providos de outros meios;
 - Promover *marketing* digital.

- Estudante
 - Interface *web* e *mobile* para os alunos regularmente matriculados;
 - Acompanhamento dos diálogos propostos em sala de aula;
 - Resolução dos questionários a respeito dos diálogos;
 - Vocalização dos diálogos contidos no material.

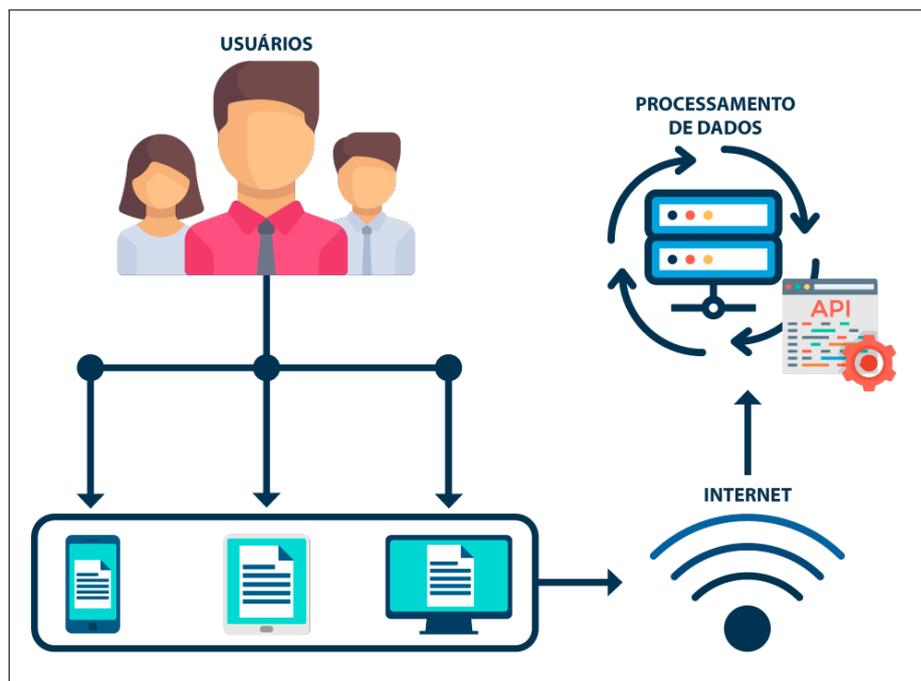
De acordo com os requisitos listados acima, pode-se afirmar que o *Language Adviser* é ferramenta que agrega recursos úteis para as instituições de ensino, oferecendo um suporte especializado que compõe a própria metodologia da instituição, beneficiando a coerência do aluno no decorrer de sua vida acadêmica. A plataforma é um conjunto de três aplicações que trabalham de forma sincronizada. Duas aplicações (administrador e estudante) estão na plataforma *web*, e a outra é uma interface *mobile* formulada através das funções do estudante desenvolvida em Ionic.

O *Language Adviser* é um plataforma distribuída que permite a comunicação entre os diversos dispositivos de forma sincronizada, através da internet. A aplicação se comunica com uma API REST que efetua o processamento das informações para serem disponibilizadas aos aparelhos em diversos sistemas operacionais, disponibilizando conteúdo para os alunos devidamente cadastrados na plataforma, como ilustra a Figura 10.

O conteúdo que o *Language Adviser* oferece é provido através da própria instituição de ensino que transforma todo o material físico em conteúdo digital, de forma o aluno terá acesso ao conteúdo em qualquer lugar, utilizando o recurso da internet.

O *Language Adviser* foi pensado para atender todos os tipos de usuários, disponibilizando um sistema *web* e outro *mobile*. O usuário pode acessar a plataforma sem se preocupar com o sistema operacional do aparelho, ou do dispositivo a ser acessado, já que a plataforma oferece um *layout* totalmente responsivo trabalhado com tecnologias híbridas.

Figura 10 – Comunicação com API.



Fonte: Autoria própria.

4.3 Ferramentas auxiliares

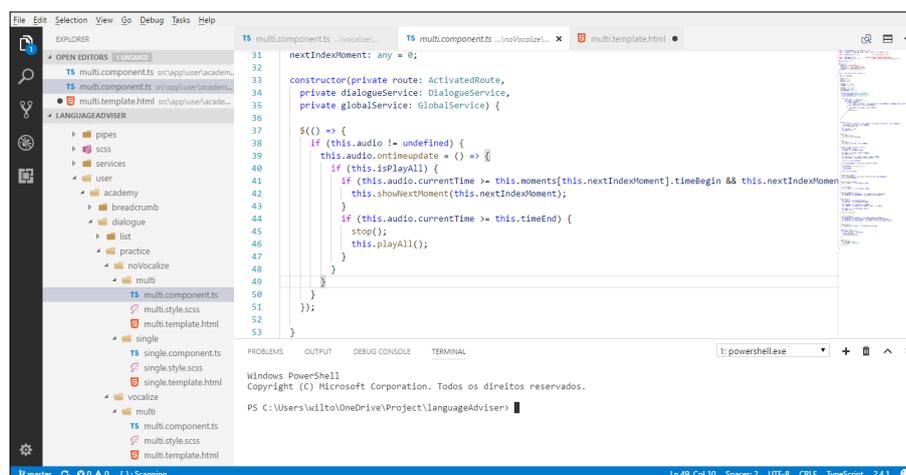
Para a codificação do *Language Adviser*, foi utilizado o Visual Studio Code (VS Code), que é um editor de texto multiplataforma disponibilizado pela Microsoft para o desenvolvimento de aplicações *web*.

O VS Code é um ambiente de desenvolvimento integrado leve, mas poderosa que possibilita criar aplicações em diversas linguagens de programação como por exemplo, C++, Java, Python, PHP. O mesmo oferece um suporte incorporado para JavaScript, TypeScript e Node.js (CODE, 2018). Ele possui uma série de ferramentas e recursos que agilizam a produtividade do programador, como *extensions* que podem ser utilizadas como auxílio na programação de *frameworks*, como: Angular e Ionic.

A plataforma ainda possui *prompt* de comando integrado, que facilita os comandos do NPM, já que não há necessidade de ficar trocar de tela para um terminal, ao dar comandos na plataforma. Ele possui uma maior interação com classes, funções e interfaces, podendo serem levadas automaticamente ao documento raiz ao serem clicadas.

A Figura 11 mostra o ambiente de desenvolvimento em execução, com um projeto aberto. Ao lado esquerdo da imagem tem-se a estrutura de diretórios que a interface gráfica oferece, podendo editar códigos do mesmo e arquivos de configuração, ao lado direito tem-se um trecho de código de um dos componentes da aplicação

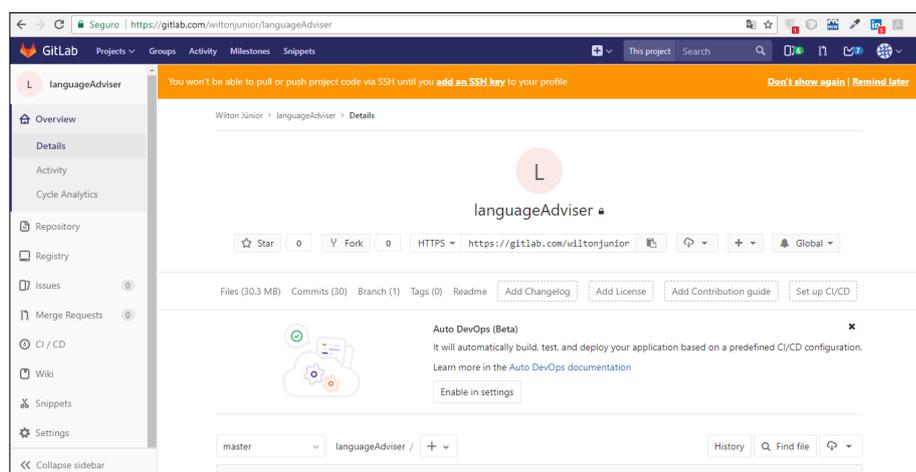
Figura 11 – Interface do Visual Studio Code.



Fonte: (CODE, 2018).

Outro recurso muito importante que foi utilizado para o desenvolvimento do projeto foi o GitLab, que é um serviço de armazenamento *web* que permite administrar projetos públicos e privados de forma gratuita através do git, que gerencia o controle de versões do projeto, em inglês *Version Control System (VCS)*, sendo um sistema que registra as mudanças feitas em um arquivo ou um conjunto de arquivos ao longo do tempo, de forma que se possa recuperar versões específicas da aplicação.

Figura 12 – Interface do GitLab.



Fonte: (GITLAB, 2018).

O VCS permite reverter arquivos para um estado anterior comparando as mudanças feitas ao decorrer do tempo ou até mesmo, verificar atualizações que podem causar possíveis problemas. Além disso, não é necessário a conectividade com um servidor ou a uma rede,

podendo executar localmente, controlando todo o projeto sem maiores esforços (GIT, 2017). A Figura 12 ilustra a interface do GitLab, com o repositório *languageAdviser* aberto, com o *software* é possível acessar uma serie de recursos do git, como a quantidade de *commits* da aplicação, o histórico de versões associados ao projeto, e até mesmo a gerencia de usuários que tem acesso a aplicação, todos esses fatores são facilmente encontrados na ferramenta.

4.4 Implementação

A comunicação do *front-end* com o *back-end* foi realizada através de uma API REST que forneceu uma documentação gerada atravez do Swagger. Ele é um serviço que auxilia a descrever recursos de forma padronizada, facilitando o entendimento de cada rota existente na API e sua estrutura, sem que haja a necessidade de visualizar o código-fonte. A interface da API REST é constituída por um conjunto de rotas que foram consumidas pela aplicação, de maneira que os mais variados tipos de programas acessem os serviços disponibilizados pela API via HTTP, mantendo a interoperabilidade dos sistemas. Uma API REST é responsável por armazenamento, consulta e alteração de dados da aplicação, que disponibiliza uma conjunto de serviços REST para servir qualquer aplicação de cunho social.

Para facilitar a compreensão supõe-se um exemplo simples de uma API REST, que esta no endereço *http://languagead.com* compondo uma URN "usuario". Todas as ações a respeito do usuário vão ser estabelecidas através desta URI (*http://languagead.com/usuario*). A Tabela 5 exemplifica os métodos HTTP associados com a URI de uma API REST.

Tabela 5 – Exemplificação das rotas de uma API REST.

MÉTODO	URI
POST	<i>http://languagead.com/usuario</i>
GET	<i>http://languagead.com/usuario</i>
PUT	<i>http://languagead.com/usuario/id</i>
DELETE	<i>http://languagead.com/usuario/id</i>

Fonte – Autoria Própria.

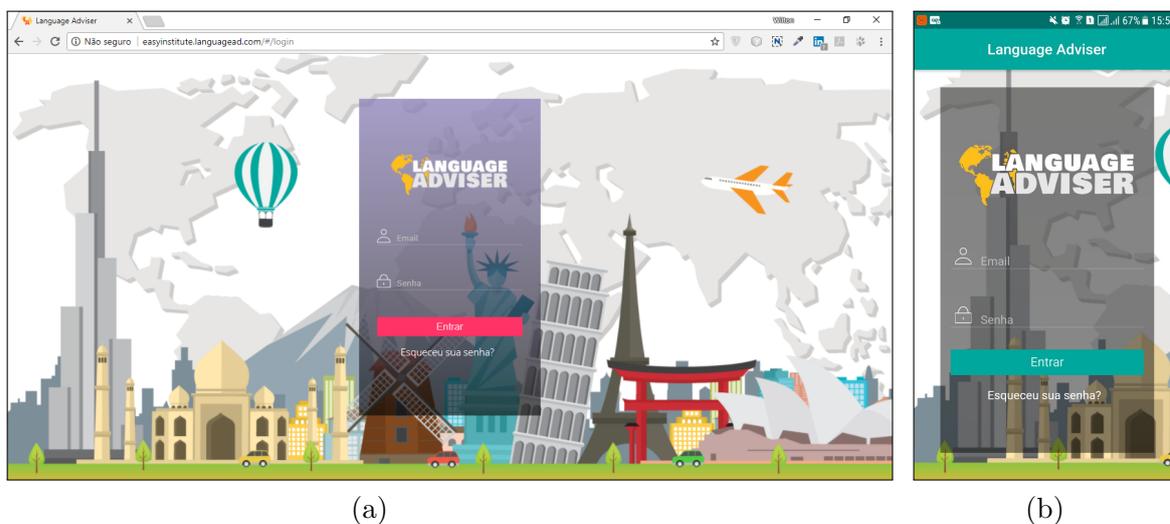
A integração da API com os *framework* Angular e Ionic foi realizada através de *services* que implementam os recursos do HTTP, ou seja, são serviços que foram usados na aplicação para servir de ponte entre os componentes e a API, para realizar a troca de informações da aplicação. O trecho de código descrito no Apêndice A, ilustra um serviço que implementa todas as recursos da Tabela 5, mostrando como o serviço se conecta com a API e como essa informação é enviada para o *back-end*.

Todos os serviços da aplicação utilizam o mesmo modelo para estabelecer uma comunicação com a API. A programação foi realizada por meio da linguagem TypeScript empregando as boas práticas de programação para os respectivos *frameworks*, Angular e Ionic na criação dos serviços. O Anexo A apresenta todas os recursos que foram utilizados nesta aplicação, descritos por meio de tabelas que identificam a URI e o método HTTP para estabelecer a comunicação.

4.4.1 Funcionalidades do *software*

O *Language Adviser*, foi construído utilizando linguagens de marcação de texto (*HyperText Markup Language* (HTML), Javascript, *Cascading Style Sheets* (CSS)) voltadas ao desenvolvimento *web*. Sua estrutura de *layout* foi realizada utilizando os *frameworks* Angular e Ionic, descritos na seção 3. Ele possui uma interface sólida, dinâmica e intuitiva, facilitando a navegação entre as páginas da aplicação. Durante toda a fase de desenvolvimento foram utilizadas as boas práticas de programação, com Typescript, e os *frameworks* descritos anteriormente.

Figura 13 – Tela de *login* da aplicação.



Fonte: Autoria própria.

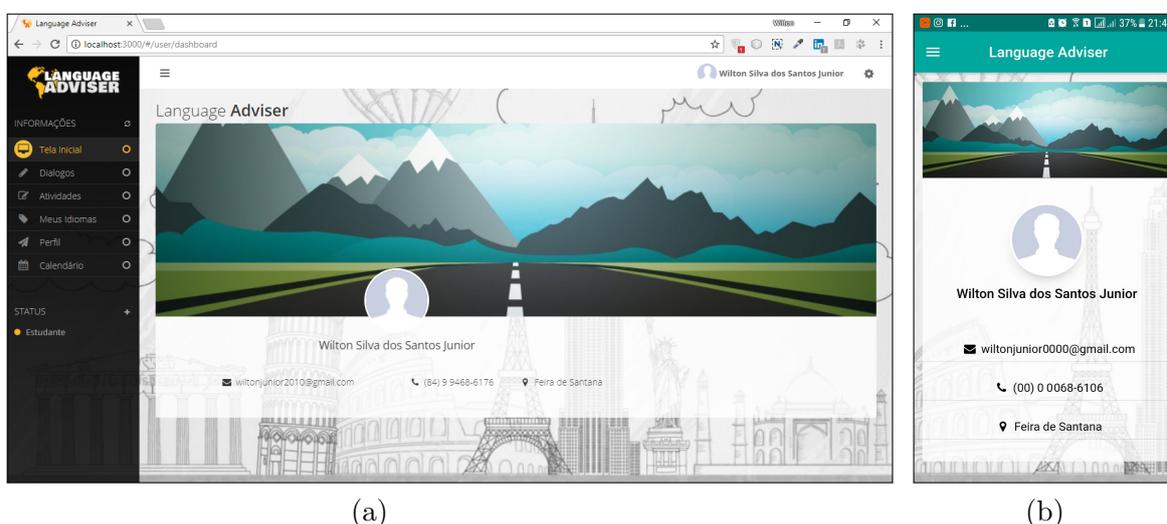
A Figura 13, ilustra a tela de *login* do *Language Adviser*. Para efetua-lo é necessário fornecer *e-mail* e senha devidamente cadastrados na plataforma. Nesta etapa a aplicação enviará as coordenadas geográficas do usuário para a API, com sua devida autorização. Com isso, o *software* encontrará a localização geográfica do usuário, para o processamento das informações. Nesta mesma tela, o usuário tem a opção para recuperar a sua senha, caso o aluno por algum motivo a tenha esquecido.

4.4.1.1 Funcionalidades do Estudante

Esta Seção tem como objetivo exemplificar as funções do estudante por meio de sua interface. As funcionalidades foram divididas em duas aplicações, uma *web* e outra *mobile* para facilitar a mobilidade e o acesso das informações ao aluno.

A Figura 14a mostra a Tela Inicial da aplicação. Ao centro, a tela fornece algumas informações, como: *e-mail*, telefone e endereço; ao lado esquerdo, a interface dispõe de opções de navegação, como: Tela Inicial, Diálogos, Atividades, Meus Idiomas, Perfil e Calendário, elas são páginas que oferecem recursos e facilidades para o usuário. A Figura 14b mostra essas mesmas informações em uma interface *mobile*.

Figura 14 – Tela inicial do Estudante.



(a)

(b)

Fonte: Autoria própria.

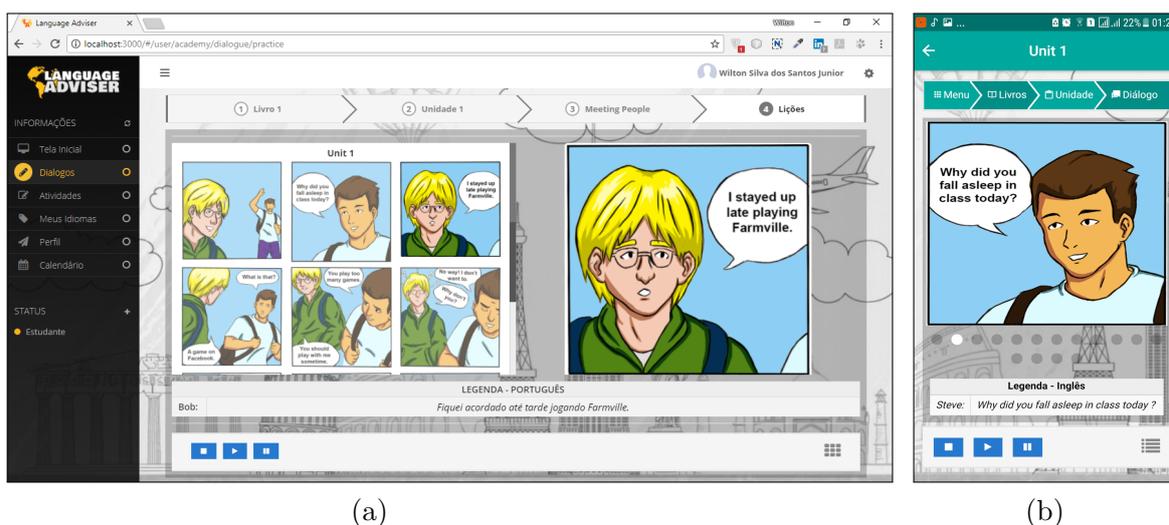
A opção “Diálogos”, leva o estudante aos diálogos cadastrados na plataforma, o *software* necessita de algumas informações para alcançar os diálogos, tais informações são escolhidas pelo estudante nas transições de tela.

Para facilitar o entendimento e o contexto do diálogo, a plataforma dispõe de uma interface responsiva, isso contribui para expandir a mobilidade do aluno no estudo dos diálogos contidos no material. A Figura 15a ilustra como os diálogos são adaptados na plataforma. Para ouvir um determinado trecho é preciso clicar na imagem a qual o trecho do diálogo se encontra. Os áudios podem ser pronunciados por falantes nativos ou vocalizados digitalmente por meio da API da Amazon Polly¹. Este é um serviço que transforma texto em falas realistas, incluindo um conjunto de vozes em vários idiomas, com recursos de fala que funcionam em vários países diferentes. O usuário ainda tem a opção de visualizar a tradução do trecho clicando em “Legenda”.

¹ Site oficial da Amazon Polly: <https://aws.amazon.com/pt/polly/>

Ainda na Figura 15a, pode-se observar acima o *breadcrumb* que informa as páginas que o usuário percorreu para alcançar o diálogo. Abaixo tem-se algumas opções para controlar o diálogo, são elas *Pause*, *Stop* e *Play*, no canto inferior direito tem-se a opção de mudar o *layout* para a tela ilustrada na Figura 16a. A Figura 15b mostra como essa informações são adaptadas na interface *mobile*.

Figura 15 – Tela de diálogo com ilustrações.

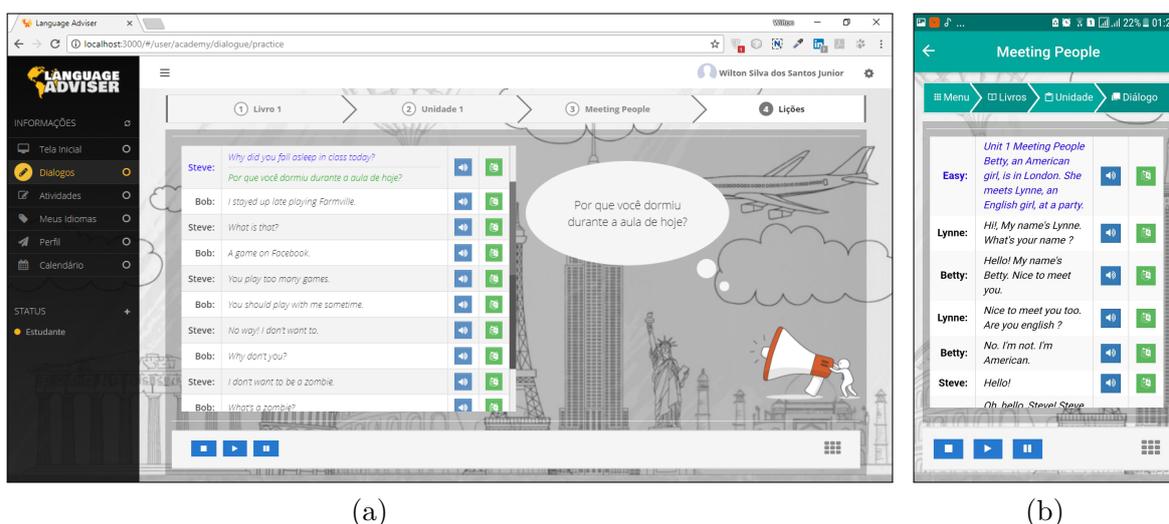


(a)

(b)

Fonte: Autoria própria.

Figura 16 – Tela de diálogo em lista.



(a)

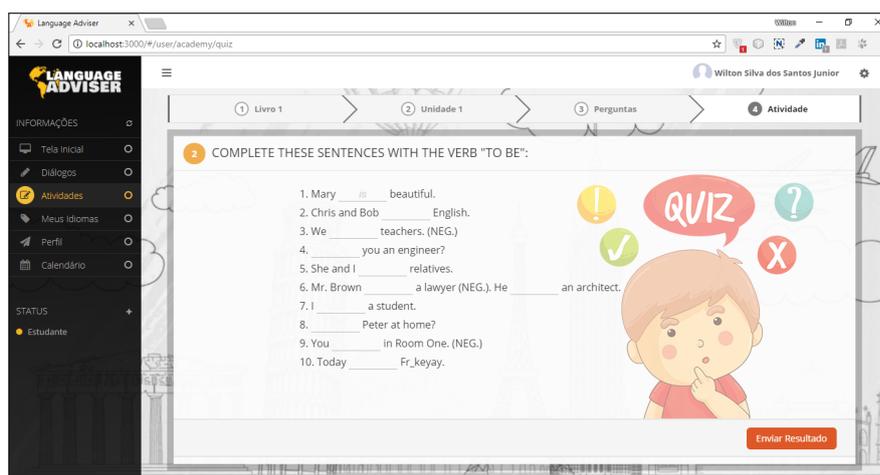
(b)

Fonte: Autoria própria.

A Figura 16a, apresenta o diálogo sobre outro panorama, nesta interface o usuário visualizar todos os trechos do diálogo em lista, para ampliar o campo de visão e agilizar o seu estudo. Nessa lista o aluno tem a opção de escutar e ver a tradução dos respectivos trechos do diálogo, ao lado tem um balão interativo, que está sincronizado com o trecho do diálogo

atual. A figura 16b mostra essas opções na interface *mobile*. A opção “Atividades”, leva o estudante para as questões contidas no material, que foram cadastradas na plataforma, como em “Diálogos” o *software* necessita de alguma informações para encontrar a atividade correspondente, por isso o estudante deve selecionar algumas opções nas transições de tela.

Figura 17 – Tela do questionário.



Fonte: Autoria própria.

A Figura 17 ilustra um modelo de atividade que pode ser encontrada na plataforma. Outros modelos podem surgir, como questões subjetivas ou objetivas. A opção “Meus Idiomas” lista os idiomas que o aluno está estudando no momento, podendo adicionar outros ou removê-los. A tela “Perfil” traz todas as informações cadastrais do estudante, que podem ser alteradas, caso necessário. A opção “Calendário” apresenta uma agenda intuitiva e dinâmica para o aluno planejar as aulas de estudos.

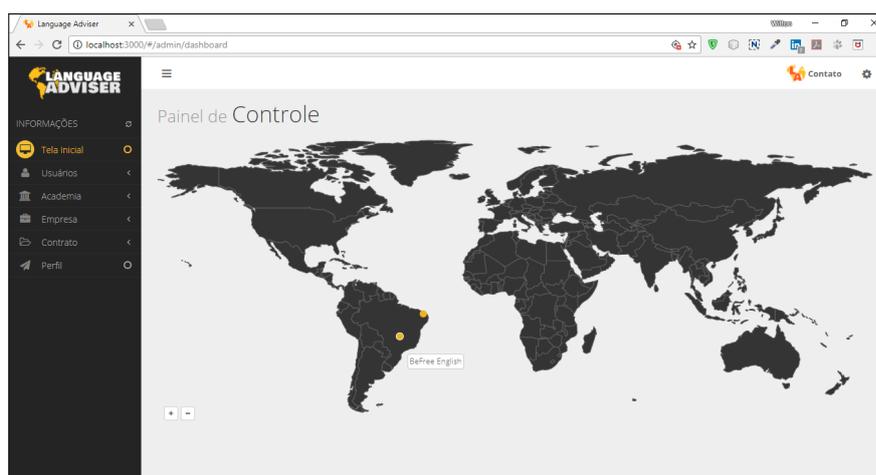
4.4.1.2 Funcionalidades do Administrador

Nesta Seção, são apresentadas e exemplificadas as funcionalidades do Administrador por meio de telas que a interface responsiva do *Language Adviser* dispõe para seus usuários. O Administrador é responsável por imputar dados na plataforma através de formulários, como: lições, diálogos, questões e os dados dos usuários. Os administradores ainda podem realizar *marketing* digital através da plataforma, utilizando termos presentes em seus conteúdos.

A Figura 18 mostra a tela inicial do administrador. Ao centro a interface apresenta um painel de controle, contendo as posições geográficas das empresas associadas ao *Language Adviser*. Ao lado esquerdo, a interface apresenta um menu composto pelas opções: “Usuários”, “Academia”, “Empresa”, “Contrato” e “Perfil”, algumas destas opções são divididas em subopções, que serão apresentadas em seguida.

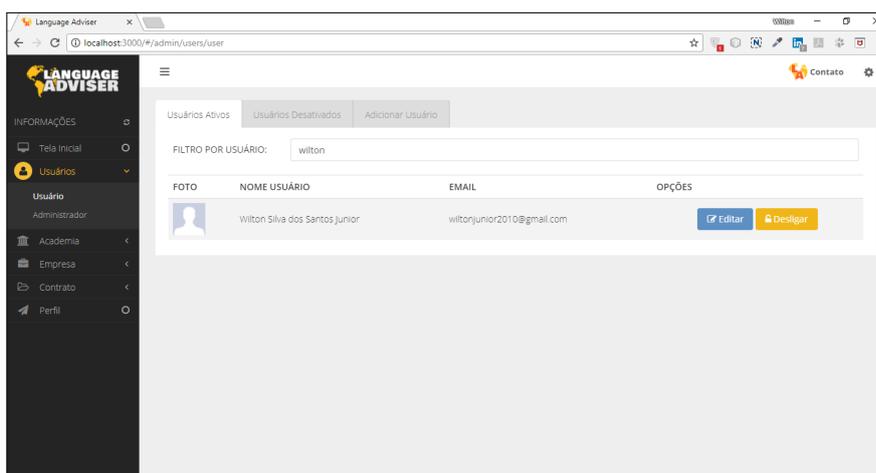
A opção “Usuário” têm subopções “Usuário” e “Administrador”. A Figura 19, ilustra a tela da subopção “Usuário”. Ela é formada por uma lista de usuários ativos que podem ser desativados ou editados, lista de usuários desativados e a opção de cadastrar usuário. A subopção “Administrador” segue a mesma estrutura da subopção “Usuário”, contendo as mesma opções e subopções.

Figura 18 – Tela inicial do administrador



Fonte: Autoria própria.

Figura 19 – Tela de usuários.



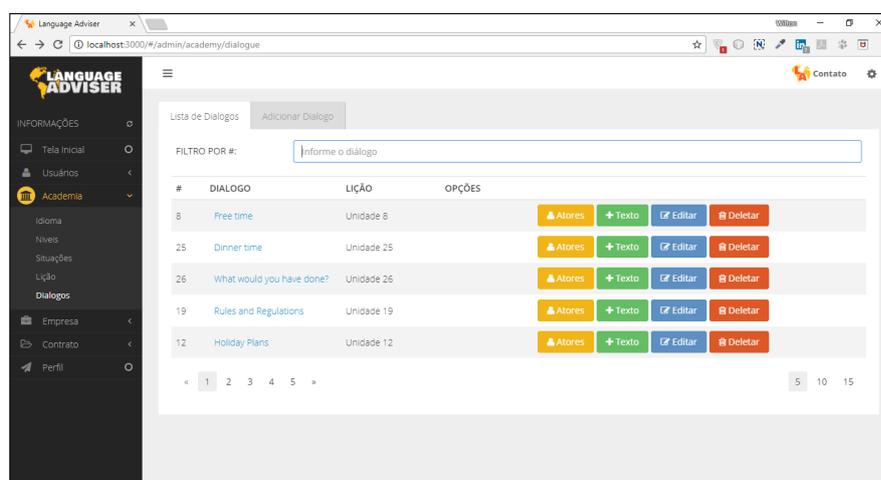
Fonte: Autoria própria.

A opção “Academia” tem algumas subopções, tais como: “Idioma”, “Níveis”, “Situações”, “Lição” e “Diálogos”. Em todas subopções, foram realizadas as operações de adição, leitura, atualização e deleção (CRUD). A Figura 20 apresenta a tela dos diálogos,

ele é composto por vários fragmentos de diálogo, para facilitar a compreensão neste trabalho vamos denominar esse trechos de momentos.

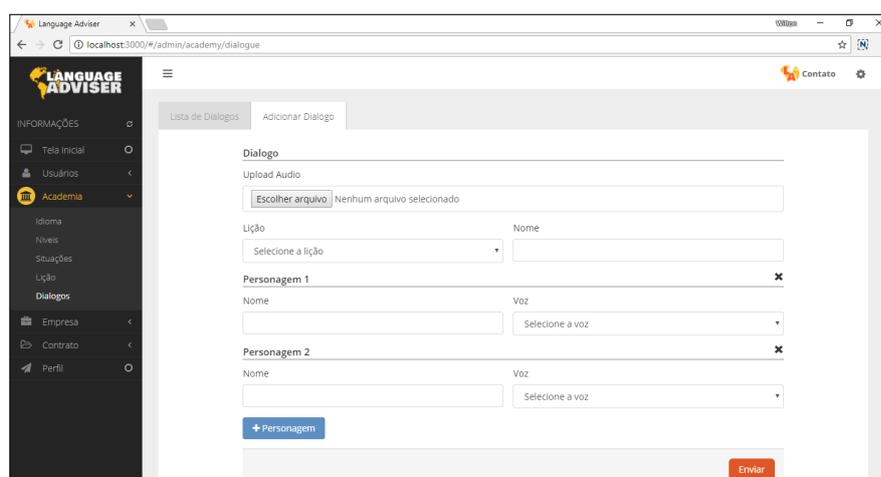
Um diálogo é formado por personagens (Atores) que são imputados junto com o diálogo dinamicamente. A opção “Atores” é responsável por listar todos os personagens do diálogo. A opção “Texto” serve para adicionar momentos ao diálogo. A Figura 22 exemplifica esse processo.

Figura 20 – Tela de diálogos.



Fonte: Autoria própria.

Figura 21 – Tela de cadastro do diálogo.

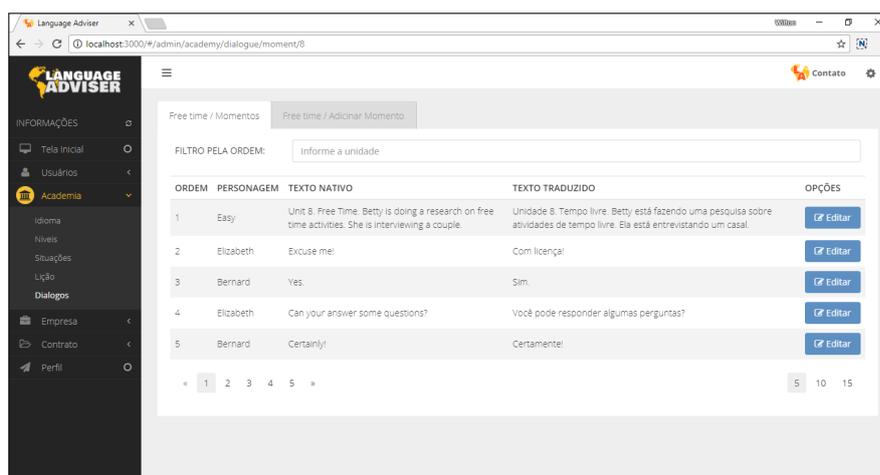


Fonte: Autoria própria.

A Figura 21 expõe a opção “Adicionar Diálogo” presente na interface do Administrador. Ela tem como objetivo cadastrar o diálogo, e para isso é necessário informar o áudio,

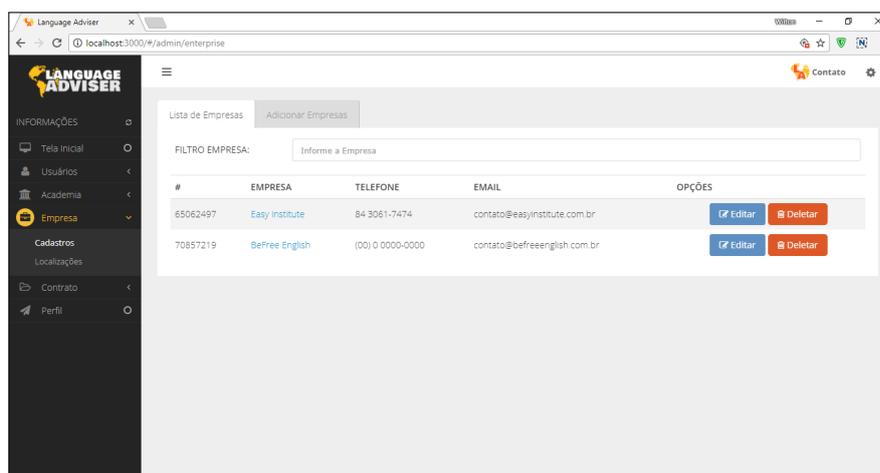
lição, nome do diálogo e os personagens que o compõem. A opção “Personagem” tem como objetivo adicionar mais personagens ao diálogo dinamicamente. Após o preenchimento de todos os campos, a opção “Enviar” encaminha as informações para a API, efetuando o processamento dos dados.

Figura 22 – Tela de momentos.



Fonte: Autoria própria.

Figura 23 – Tela de empresas.

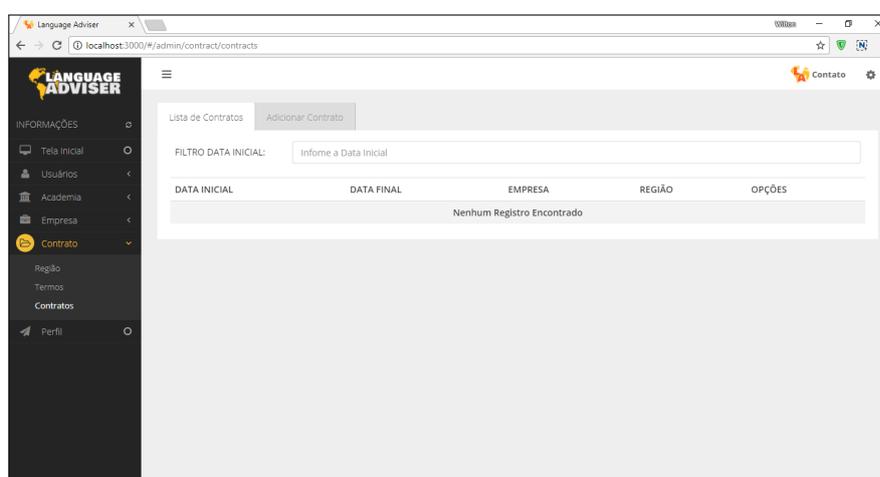


Fonte: Autoria própria.

A opção “Empresa”, contém duas subopções, “Cadastros” e “Localizações”. A Figura 23, exemplifica a subopção “Cadastros”, que lista as empresas cadastradas na plataforma, elas podem ser editadas ou excluídas. Para adicionar uma nova empresa, basta selecionar a opção “Adicionar Empresa”, e preencher o formulário. A subopção “Localização”, exibe um mapa contendo as regiões geográficas das empresas.

A opção “Contrato” tem três subopções “Região”, “Termos” e “Contratos”. A Figura 24 exemplifica a subopção “Contrato”. Nela tem-se a lista de contratos que foram estabelecidos. Para adicionar um contrato, é necessário selecionar a opção “Adicionar Contrato” e, logo após, preencher um formulário com os seguintes dados: datas inicial e final do contrato, a empresa firmada e a região geográfica. As subopções “Região” e “Termos” realizam as operações de leitura, escrita, atualização e remoção, em seus respectivos formulários. A opção “Perfil”, apresenta os dados cadastrais do administrador, podendo ser alterados, se necessário.

Figura 24 – Tela de contratos.



Fonte: Autoria própria.

5 Prova de Conceito

Este capítulo tem como objetivo explicar como a aplicação se desenvolveu em um ambiente real de trabalho, bem como apresentar as empresas que utilizaram a plataforma para auxiliar o ensino aprendido de seus alunos e como o conteúdo das escolas foi disponibilizado aos estudantes.

5.1 Ecossistema de trabalho

Como já mencionado, o *Language Adviser* é um *software* que leva conteúdo de forma atrativa e atraente aos alunos, que almejam aprender ou praticar uma determinada língua estrangeira. Sabendo disso, a plataforma foi adotada por duas instituições, especializadas no ensino do idioma Inglês, uma localizada na cidade de Mossoró - RN e outra em Goiânia - GO, que viabilizaram o conteúdo físico (Livros e CD's), para torná-lo digital e atraente para seus alunos, sendo disponível nas mais diversas plataformas aos estudantes. Para não expor as empresas, neste trabalho elas vão ser identificadas como: empresa A e B, respectivamente.

A empresa A, com sede na cidade de Mossoró, possui uma estimativa de 700 alunos dispersos pelo país, distribuído em unidades franqueadas da instituição, nos quais, apenas alguns alunos foram selecionados para utilizar a ferramenta para análise de seus resultados. Dessa forma, o *Language Adviser* foi utilizado pelos alunos da unidade de Mossoró, que compõe aproximadamente 100 alunos de seu total. Para a empresa B, centralizada na cidade de Goiânia, a plataforma foi disponibilizada para todos os seus 100 alunos matriculados. Os resultados do *Language Adviser* foram coletados a partir das experiências dos estudantes dessas duas instituições de ensino.

5.2 Metodologia aplicada pelas instituições de ensino

A empresa A e B utilizam metodologias similares para efetuar suas práticas metodológicas, focando na conversação e no estudo dos diálogos contidos no material. Como as instituições focam na conversação oral, antes das aulas práticas, os alunos deveriam realizar uma sequência de instruções para aprimorar e fixar o conteúdo contido no material, utilizando os livros e um CD de áudio disponibilizados aos alunos. Dentre os passos que os alunos realizam, destaca-se a ato de escutar o diálogo acompanhando pelo livro uma determinada quantidade de vezes durante o dia, com pausa e sem pausa entre as falas (momentos).

Para realizar as instruções, os alunos, em sua grande maioria, deviam gastar tempo e recurso selecionando trechos específicos do áudio correspondentes no diálogo contido no livro. Com a adoção da ferramenta *Language Adviser* à metodologia das escolas, a realização dessas instruções se tornaram rápidas e dinâmicas, permitindo flexibilidade e mobilidade aos alunos, não dependendo de materiais impressos para a prática dos diálogos, como livro ou um CD de áudio, que em sua grande maioria acabam se tornando insuficientes pela ausência de praticidade e não adaptáveis aos novos recursos digitais.

5.3 Aplicação do software no meio acadêmico

Para possibilitar o uso da ferramenta pelos estudantes, foi necessário preparar a plataforma com o conteúdo próprio das instituições de ensino. Dessa forma, as organizações viabilizaram seus materiais didáticos, para alimentar e preparar o *Language Adviser* com o conteúdo necessário. Ele é composto por diálogos, ilustrações e atividades relacionadas aos mesmos, que estavam contidos em três livros distintos. Os áudios dos diálogos estavam em um CD que disponibiliza todos os diálogos do material em formato MP3.

Todos os diálogos foram cadastrado por meio da ferramenta, os textos em inglês foram inseridos no *software* seguidos de sua tradução e suas respectivas ilustrações, que foram recortadas do material e preparadas para serem enviadas à plataforma, justamente com o trecho de diálogo do áudio original em milisegundos que estavam sincronizados com os momentos do diálogo.

5.4 Resultados obtidos

No Apêndice B se encontra o questionário que foi enviado aos 200 estudantes que utilizaram a plataforma, para à análise de seus resultados. O questionário foi construído com 5 sessões que abordam pontos fundamentais da ferramenta, tais como: Interface gráfica, Utilização do *software*, Utilidade do *software*, Funções da ferramenta e a Metodologia de estudo dos diálogos. Todos esses pontos serão discutidos separadamente no decorrer deste capítulo. No total 64 alunos responderam o questionário, no período de duas semanas em que o formulário esteve disponível.

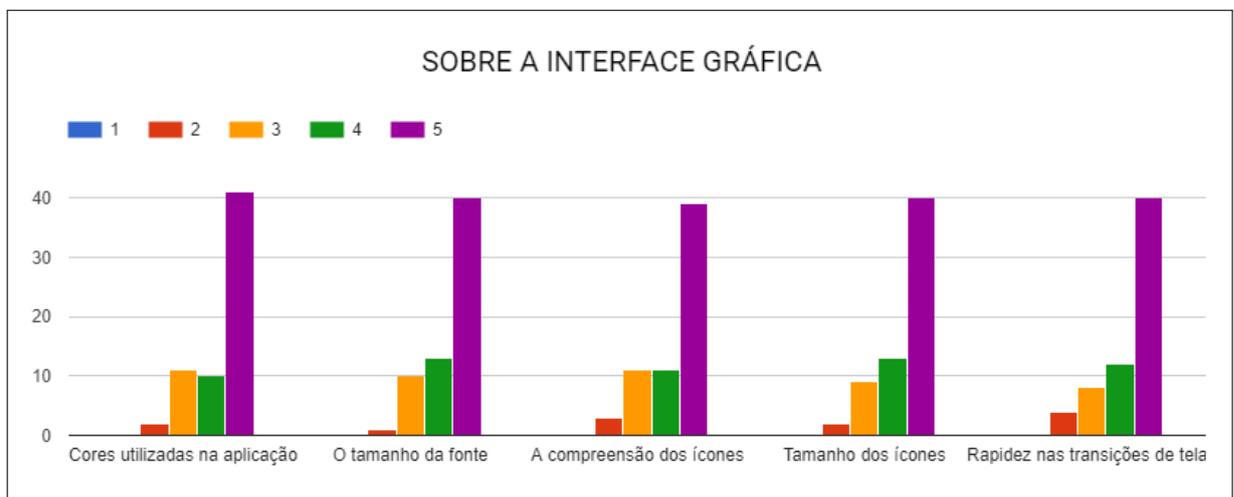
Para avaliar a interface gráfica os usuários atribuíram notas de 1 a 5, considerando um 1 como nota mínima e 5 como nota máxima aos quesitos.

- Cores utilizadas na aplicação;
- Tamanho da fonte;
- Tamanho dos ícones;

- Compreensão dos ícones;
- Rapidez nas transições de telas.

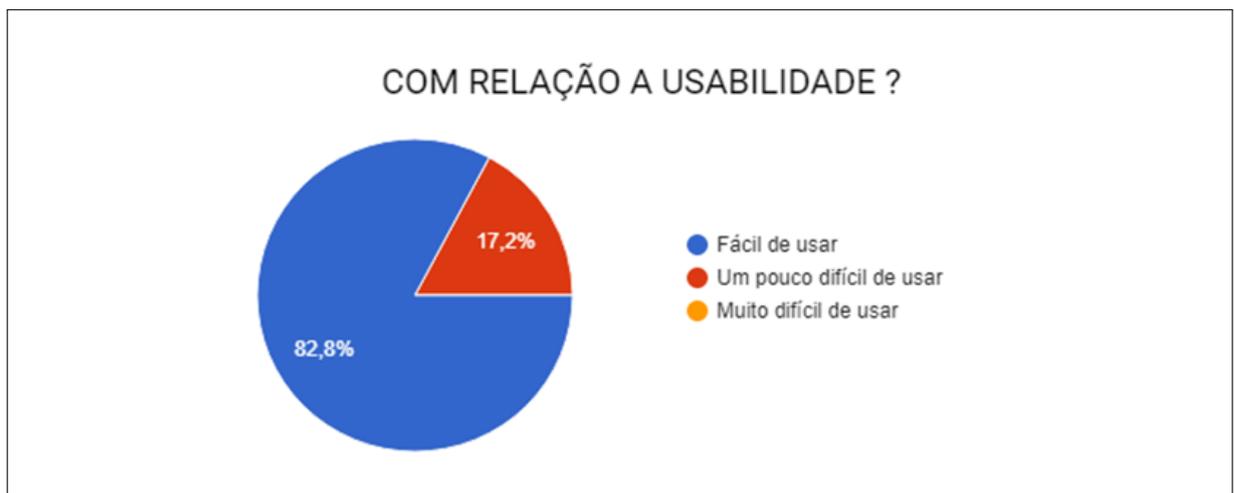
O Gráfico 1 apresenta um análise estatística das respostas obtidas através da Questão 1. Essa parte do questionário permitiu identificar, que a interface gráfica mostrose satisfatória em todos os seus elementos, por uma grande porção dos usuários, já que todos os quesitos abordados obtiveram o maior percentual no questionário.

Gráfico 1 – Interface gráfica.



Fonte – Autoria própria.

Gráfico 2 – Usabilidade do *software*.



Fonte – Autoria própria.

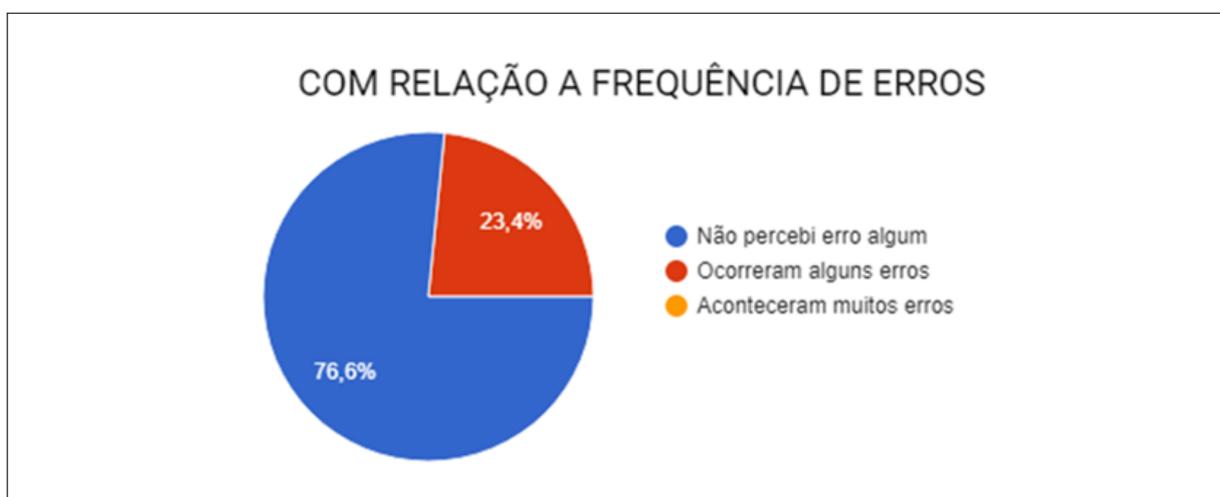
A segunda sessão do questionário, oferece questões a respeito da utilização do *software*, em termos de usabilidade, experiência e eficiência da plataforma. O Gráfico 2 apresenta dados a respeito da usabilidade, é possível analisar que 82,8% os usuários, não tiveram problemas em utilizar a plataforma, isso mostra que a ferramenta é um *software* de fácil manuseio. O Gráfico 3 apresenta uma análise da eficiência de resposta da aplicação, esta pesquisa mostra que 78,7% dos usuários acharam a aplicação com um tempo de resposta satisfatório. Já 20,3% dos estudantes perceberam que a ferramenta, em alguns casos não obteve um tempo de resposta satisfatório, deixando a aplicação um pouco lenta.

Gráfico 3 – Eficiência da ferramenta.



Fonte – Autoria própria.

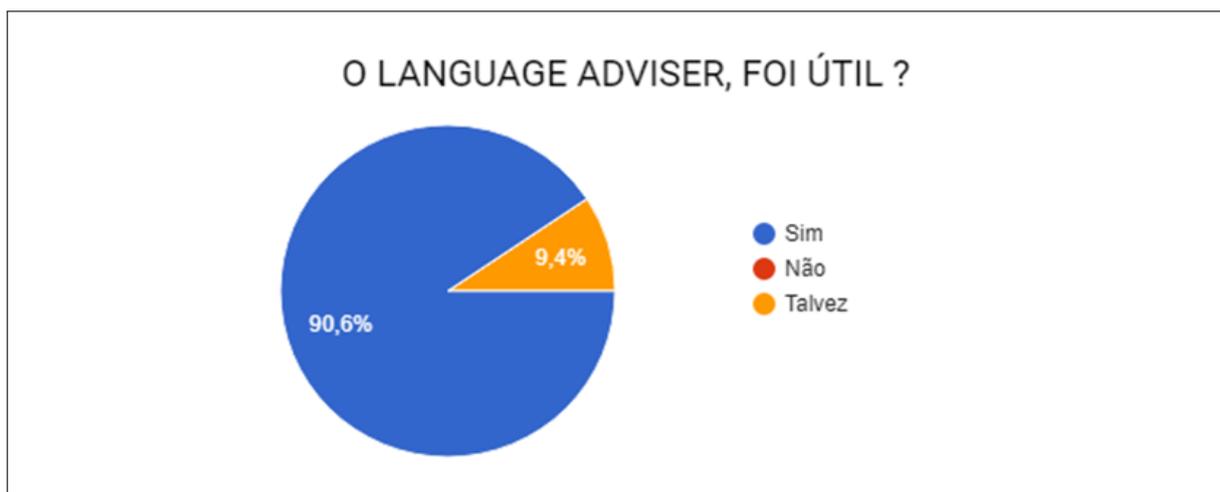
Gráfico 4 – Frequência de erros da interface.



Fonte – Autoria própria.

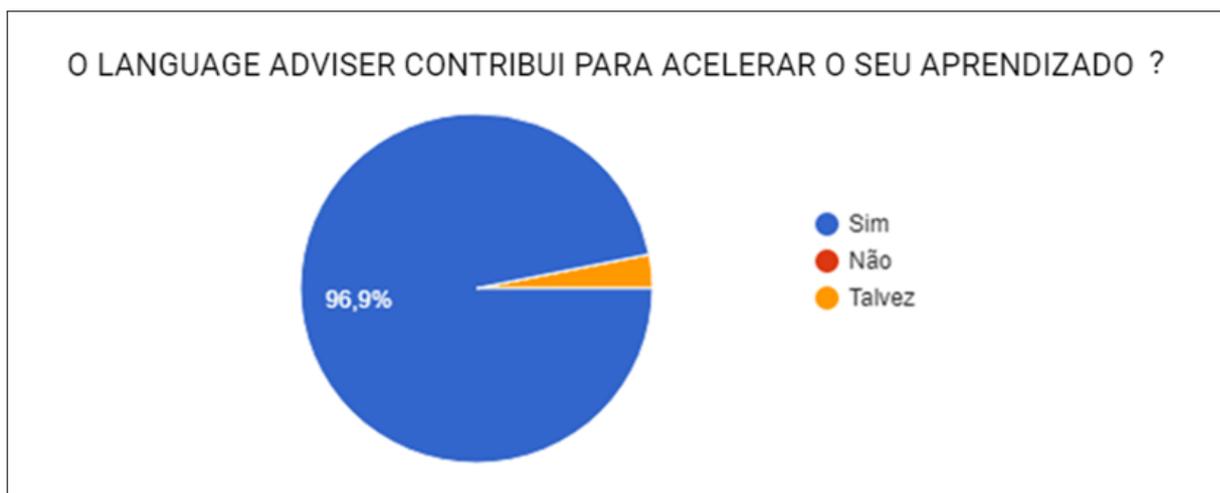
Para verificar a frequência de erros da aplicação, o questionário disponibilizou um quesito para esta análise. O Gráfico 4 mostra o resultado da pesquisa abordada, percebe-se que 78,6% dos alunos não se depararam com erros no *software*, mas 23,4% dos usuários encontram algum tipo de anomalia da ferramenta, isso mostra que a plataforma ainda necessita de atualizações, para minimizar a frequência de erros da aplicação.

Gráfico 5 – Utilidade do software.



Fonte – Autoria própria.

Gráfico 6 – Contribuição ao aprendizado.



Fonte – Autoria própria.

A sessão 3 do questionário, disponibilizou um quesito para a utilidade da ferramenta. Nesta questão os usuários teriam que responder entre sim, não ou talvez para avaliar a utilidade do *Language Adviser*. O Gráfico 5 apresenta o resultado dessa pesquisa, nesta

análise 90,6% dos alunos avaliaram o *software* como útil e apenas 9,4% do total por responder "Talvez".

A sessão 4 do questionário aborda as funções do *Language Adviser*. Esta questão explora a contribuição da ferramenta para o ensino e aprendizado dos estudantes. O Gráfico 6 apresenta esta análise, e percebe-se que 96,9% dos usuários aprovam que o ferramenta acelera o aprendizado de uma determinada língua estrangeira.

O Gráfico 7 explora os resultados obtidos através da sessão 5 do questionário, este quesito traz uma comparação da aplicação da metodologia da escola com e sem a utilização da plataforma. Esta análise constatou que a prática dos diálogos se torna proveitosa e simplificada com a utilização da plataforma, como se pode observar 92,2% dos alunos apontam que o *Language Adviser* facilita o estudo dos diálogos de acordo com a metodologia da escola.

Gráfico 7 – Práticas dos diálogos.



Fonte – Autoria própria.

Dessa forma, dentre os resultados analisados, percebe-se que a aplicação mostrou-se promissora em todos os quesitos abordados no questionário. Tendo em vista que, a ferramenta contribui para acelerar o aprendizado, atuando em um papel fundamental na pratica dos diálogos, tornando o seu estudo dinâmico e atrativo, facilitando a execução metodologias das instituições de ensino de língua estrangeira.

6 Considerações Finais

No mundo globalizado, a internet permeia a comunicação de diferentes áreas de conhecimento. A utilização de Tecnologias Digitais da Informação e Comunicação, aprimora o acesso ao conteúdo disponibilizado, aumentando a popularização das aplicações *web* e *mobile*, pela praticidade, disponibilidade e facilidade de utilização no ambiente escolar. Isso traz reflexões a respeito das tecnologias como mediadora de conteúdo, com objetivo de expandir a qualidade e a eficiência, dos institutos especializados no ensino de língua estrangeira. Já que métodos convencionais de ensino não são suficiente para potencializar as habilidades do aluno, é necessário a utilização de ferramentas para flexibilizar e maximizar o ensino de língua estrangeira.

Diante disso, este trabalho apresentou a aplicação *Language Adviser*, como um *software* multiplataforma construído em JavaScript, utilizando os *framework* Angular e Ionic, para a criação de uma aplicação *web* e *mobile* que se adeque a todos os dispositivos, permitindo uma ampla mobilidade aos usuários. A ferramenta possui funcionalidades que se adapta às necessidades das instituições, promovendo interatividade digital, por meio de conteúdo próprio da escola concedido a plataforma, para ser viabilizado aos alunos, aumentando a produtividade dos estudantes e professores em sala de aula. O conteúdo torna-se inteiramente disponível no *software*, podendo ser acessado de qualquer lugar, através de internet.

O *Language Adviser* oferece várias vantagens e permitir novas abordagens didáticas, quando utilizado de maneira consciente para expandir a construção do conhecimento dos alunos, que pretendem desenvolver uma nova habilidade de comunicação, no estudo de língua estrangeira. A ferramenta permite a manipulação de materiais autênticos e atualizados para serem trabalhados em sala de aula, apresentado diferentes características didáticas, que propiciam e fomentam aprendizagem autônoma para os seus usuários.

Este trabalho teve como objetivo facilitar e auxiliar a execução da metodologia das instituições das ensino de língua estrangeira, viabilizando uma ferramenta que se adeque aos varios dispositivos e sistemas operacionais, presentes nos aparelhos, no consumo de uma API REST para efetivar o processamento dos dados da aplicação. A ferramenta foi viabilizada para que as escolas disponibilizem o seu conteúdo, tornando todo o material físico em conteúdo digital, para ser entregues aos alunos que almejam iniciar, reiniciar, ou dar continuidade ao estudo de línguas, a plataforma realiza o planejamento e o controle do material cadastrado, podendo ser atualizado a qualquer momento a partir da área administrativa que o *Language Adviser* oferece.

De acordo com as informações levantadas do questionário, discutidas na seção

5, pode-se observar que a aplicação, mostrou-se satisfatória em vários quesitos dentro da interface gráfica e de suas funcionalidades. Sobre a utilidade e a contribuição para o aprendizado, percebe-se que ferramenta atua em um papel decisivo para potencializar e melhorar as habilidades de comunicação da língua estrangeira. A partir do estudo do questionário é possível verificar a eficiência da plataforma sobre as práticas dos diálogos. Com o *Language Adviser* é possível otimizar, maximar a metodologia das instituições de ensino de língua estrangeira, tornando o estudo proveitoso e interativo.

Com a realização deste trabalho é possível executar algumas possibilidades de trabalhos futuros, como a implementação de novas funcionalidade para *Language Adviser*, como comando de voz, para facilitar a comodidade do aluno, informando comandos específicos para serem interpretados pela plataforma. Por exemplo, muitas pessoas gostam de ler utilizando o livro físico, dessa forma o usuário pode acompanhar o dialogo do livro, relacionado com o da plataforma, através de comandos de voz. Assim, ao dizer: *Language Adviser, position, dialog, seven* (*Language Adviser, posicione, diálogo sete*), seguido de *play*, o *Language Adviser* tocará o sétimo diálogo do livro 1.

Essas funcionalidades permitiram que o *Language Adviser* possa ser comercializado como apoio para livros físicos, em substituição ao CD de áudios. O reconhecimento de voz reúne um conjunto de funcionalidades, que na prática ampliarão a utilidade do *Language Adviser* pelos usuários. No *Language Adviser*, o reconhecimento de voz servirá a dois propósitos bem distintos:

- Facilitar a interação entre usuário e o *Language Adviser*:
 - Comandos simples como *play, stop, pause*, dentre outros, se realizados a partir de comandos de voz, permitirão que o *Language Adviser* seja utilizado em diversas situações, evitando a manipulação direta do aplicativo.
- Avaliar a qualidade da pronúncia do estudante:
 - A avaliação da qualidade da pronúncia do aluno, torna-se um aspecto determinante para o *Language Adviser*, devido metodologia das escolas que endossam este projeto. Essa funcionalidade permitirá que o *Language Adviser* possa ser distribuído via *App Store* (loja de aplicativos), ampliando assim as possibilidades comerciais do software.

Ambos os propósitos necessitam de uma vasta pesquisa científica por causa das específicas características do reconhecimento de voz. Por fim, a tecnologia resultante tem um forte apelo inovador, pois de forma inédita será utilizada para o auxílio na interação entre aluno e aplicativo, tornando o *Language Adviser* cada vez mais autônomo para o ensino de língua estrangeira.

Referências

- ABERNETHY, M. *Just what is Node.js*. IBM Corporation. 2011. Acessado em 2018. Disponível em: <<https://pt.scribd.com/document/54504131/What-is-node-JS-developerWorks#scribd>>.
- ALENCAR, F. *Android e iOS abocanham mais de 96IDC*. 2015. Acessado em 2018. Disponível em: <<http://www.techtudo.com.br/noticias/noticia/2015/02/android-e-ios-abocanham-mais-de-96-do-mercado-segundo-idc.html>>.
- ALURA. *Criando aplicações Angular com Angular CLI*. 2018. Acessado em 2018. Disponível em: <<http://blog.alura.com.br/criando-aplicacoes-angular-com-angular-cli/>>.
- ANGULAR. *Angular, One framework. Mobile and desktop*. 2018. Documentação do ANGULAR, Acessado em 2018. Disponível em: <<https://angular.io>>.
- BABEL. *Fale um idioma como sempre sonhou*. 2018. Acessado em 2018. Disponível em: <<https://pt.babel.com/>>.
- BUSUU. *Fale um novo idioma com 10 minutos por dia*. 2018. Acessado em 2018. Disponível em: <<https://www.busuu.com/pt/>>.
- CAMDEN, R. K. *Apache Cordova in action*. [S.l.]: Manning Publications Co., 2015.
- CARLOS, W. G. D. O. *Sace: Um sistema em plataforma móvel e web de auxílio ao combate de endemias*. 2016.
- CODE, V. *Visual Studio Code*. 2018. Documentação do VS Code, Acessado em 2018. Disponível em: <<https://code.visualstudio.com/>>.
- CORDOVA. *Apache Cordova*. 2017. Documentação do CORDOVA, Acessado em 2018. Disponível em: <<https://cordova.apache.org/docs/en/latest/guide/overview/index.html>>.
- DEVMEDIA. *Introdução ao REST WS*. 2017. Acessado em 2018. Disponível em: <<https://www.devmedia.com.br/introducao-ao-rest-ws/2696>>.
- DUOLINGO. *Aprenda idiomas de graça. Para sempre*. 2018. Acessado em 2018. Disponível em: <<https://www.duolingo.com/>>.
- FERNANDO, L. d. S.; DIEGO, E. d. O.; BOLFE, M. *Mobile learning: Aprendizado com mobilidade*. 2013.
- FIELDING. *Architectural styles and the design of network-based software architectures*. [S.l.]: University of California, Irvine Doctoral dissertation, 2000.
- GIT. *Git – fast-version-control*. 2017. Documentação do GIT, Acessado em 2018. Disponível em: <<https://git-scm.com/book/pt-br/v1/Primeiros-passos-Sobre-Controle-de-Vers%C3%A3o>>.
- GITLAB. *DevOps simultâneos, Um único aplicativo para o ciclo de vida completo do DevOps*. 2018. Acessado em 2018. Disponível em: <<https://about.gitlab.com/>>.

- GOIS, A. *Ionic Framework Construa aplicativos para todas as plataformas mobile*. [S.l.]: Editora Casa do Código, 2017.
- GOK, N.; KHANNA, N. *Building Hybrid Android Apps with Java and JavaScript: Applying Native Device APIs*. [S.l.]: "O'Reilly Media, Inc.", 2013.
- GONÇALVES, J. A.; SILVA, V. Inglês na palma da mão: Possibilidades de aprendizagem através dos dispositivos móveis conectados à internet. *Revista de Estudos Acadêmicos de Letras*, v. 7, n. 01, p. 49–57, 2014.
- GREEN, S. B. *Desenvolvendo Com Angularjs: Aumento de Produtividade Com Aplicações Web Estruturadas*. [S.l.]: Novatec Editora, 2014.
- GUEDES, T. *Crie Aplicações com Angular: O novo framework do Google*. [S.l.]: Casa do Código, 2017.
- INTELLIGENCE, G. *Definitive data and analysis for the mobile industry*. 2018. Acessado em 2018. Disponível em: <<https://www.gsmainelligence.com/>>.
- JADHAV, M. A.; SAWANT, B. R.; DESHMUKH, A. Single page application using angularjs. *International Journal of Computer Science and Information Technologies*, Citeseer, v. 6, n. 3, p. 2876–2879, 2015.
- JALIL, S. A.; PROCAILO, L. Metodologia de ensino de línguas estrangeiras: perspectivas e reflexões sobre os métodos, abordagens e o pós-método. In: *Congresso Nacional de Educação, IX. Encontro Sul Brasileiro de Psicopedagogia, III. PUCRS*. [S.l.: s.n.], 2009. v. 26.
- JUNIOR, F. d. A. R. *Programação Orientada a Eventos no lado do servidor utilizando Node. js*. 2012.
- KENSKI, V. M. *Educação e tecnologias*. [S.l.]: Papirus editora, 2007.
- LINGUALEO. *Aprenda Inglês Jogando*. 2018. Acessado em 2018. Disponível em: <<https://lingualeo.com/pt>>.
- LOPES, S. *Aplicações mobile híbridas com Cordova e PhoneGap*. [S.l.]: Editora Casa do Código, 2016.
- MARIA, C. R.; RAPHAELLA, K. S. Ensino de Língua estrangeira, mídias digitais e o pensamento complexo. 2012.
- MASSE, M. *REST API Design Rulebook: Designing Consistent RESTful Web Service Interfaces*. [S.l.]: "O'Reilly Media, Inc.", 2011.
- MIKOWSKI, M. S.; POWELL, J. C. Single page web applications. *B and W*, 2013.
- NGOC, H. H. Single page web application with restful api and angularjs: Best practices with verto monitor. *Metropolia Ammattikorkeakoulu*, 2014.
- OLIVEIRA, I. G. G. *Construção de aplicações distribuídas utilizando-se de apis rest*. 2018.
- PEDREIRO, S. Ensino de línguas estrangeiras–métodos e seus princípios. *Especialize IPOG Revista on line, jan*, 2013.

- PEYROTT, S. *More Benchmarks: Virtual DOM vs Angular 1 e 2 vs Others*. 2016. Acessado em 2018. Disponível em: <<https://auth0.com/blog/more-benchmarks-virtual-dom-vs-angular-12-vs-mithril-js-vs-the-rest/>>.
- PRESSMAN, R. *Engenharia web*. [S.l.: s.n.], 2009.
- RFC2616. *Hypertext Transfer Protocol – HTTP/1.1*. 1999. Acessado em 2018. Disponível em: <<https://www.ietf.org/rfc/rfc2616>>.
- RIBEIRO, D. *Com LinguaLeo aprenda inglês brincando em seu Android*. 2013. Acessado em 2018. Disponível em: <<http://www.techtudo.com.br/tudo-sobre/lingualeo.html>>.
- RICHARDSON, L.; RUBY, S. *RESTful web services*. [S.l.]: "O'Reilly Media, Inc.", 2008.
- ROBERTO, d. S. W. Restful web services e a api jax-rs, conheça o poder dos serviços rest e a implementação de referência da sun, o jersey. 2015.
- SABOTA, B.; PEIXOTO, S. M. Busuu e babbel: reflexões acerca do potencial de contribuição de aplicativos para o processo de ensino e aprendizagem de inglês como língua estrangeira. *Revista Horizontes de Linguística Aplicada*, v. 14, n. 2, 2016.
- SAUDATE, A. *REST Construa API's inteligentes de maneira simples*. [S.l.]: Casa do Código, 2013.
- SEVERANCE, C. Javascript: Designing a language in 10 days. *Computer*, IEEE, v. 45, n. 2, p. 7–8, 2012.
- SILVA, A. N. C.; SCOVILLE, A. L. M. L. de. O ensino da língua estrangeira: Processos metodológicos na aprendizagem. *REVISTA INTERSABERES*, v. 10, n. 21, p. 627–642, 2015.
- SILVA, C. B. Um modelo computacional para integração de problemas de otimização utilizando banco de dados orientados a grafos. 2017.
- SOUSA, F. P. de. Criação de framework rest/hateoas open source para desenvolvimento de apis em nodejs. 2015.
- TURQUETTE, D. de F. *Engenharia de software para web*. 2003.
- VILAÇA, M. L. C.; ARAUJO, E. V. F. D. *Tecnologia, sociedade e educação na era digital*. 2006.
- WAHLBRINCK, K. A.; BONIATI, B. B. Aplicações mobile híbridas: Um estudo de caso do framework ionic para construção de um diário de classe. 2017.
- WARGO, J. M. *Apache Cordova 4 Programming*. [S.l.]: Pearson Education, 2015.

APÊNDICE A – Código fonte de um serviço

```
1 import { Observable } from 'rxjs/Observable';
2 import { Http, RequestOptions, Headers } from '@angular/http';
3
4 @Injectable()
5 export class UsuarioService {
6
7     private URI: String = "http://languagead.com/usuario";
8     headers: Headers; options: RequestOptions;
9
10    constructor(private http: Http) {
11        this.headers = new Headers({'Content-Type': 'application/json',
12            'Authorization': '#####'});
13        this.options = new RequestOptions({ headers: this.headers });
14    }
15
16    private handleError(error: any): Promise<any> {
17        console.error('Ocorreu um erro:', error);
18        return Promise.reject(error.message || error);
19    }
20
21    private addUsuario(usuario: Usuario) {
22        return this.http.post(this.URI,
23            JSON.stringify(usuario), this.options)
24            .map(response => response.json());
25        .catch(this.handleError);
26    }
27
28    private getUsuario(): Observable<Usuario[]> {
29        return this.http.get(this.URI, this.options)
30            .map(request => request.json());
31        .catch(this.handleError);
32    }
33
34    private updateUsuario(usuario: Usuario, key: String) {
35        const uri = `${this.URI}/${key}`;
36        return this.http.put(uri,
37            JSON.stringify(usuario), this.options)
38            .map(request => request.json());
```

```
38     .catch(this.handleError);
39   }
40
41   private deleteUsuario(key: String) {
42     const uri = `${this.URI}/${key}`;
43     return this.http.delete(uri, this.options)
44       .map(request => request.json());
45     .catch(this.handleError);
46   }
47
48 }
```

APÊNDICE B – Questionário de avaliação de software

UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE – UERN
 FACULDADE DE CIÊNCIAS EXATAS E NATURAIS – FANAT
 DEPARTAMENTO DE INFORMÁTICA – DI
 BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

QUESTIONÁRIO DE AVALIAÇÃO DE SOFTWARE

1. SOBRE A INTERFACE GRÁFICA:

Defina uma nota entre 1 a 5 para cada quesito da tabela abaixo considerando um 1 como nota mínima e 5 como nota máxima.

QUESITOS	NOTAS
1. Cores utilizadas na aplicação	
2. O tamanho da fonte	
3. A compreensão dos ícones	
4. Tamanho dos ícones	
5. Rapidez nas transições de telas	

2. SOBRE A UTILIZAÇÃO DO SOFTWARE:

- a) Com relação a usabilidade ?
- i. Fácil de usar
 - ii. Um pouco difícil de usar
 - iii. Muito difícil de usar
- b) Com relação a eficiência
- i. Tempo de resposta foi satisfatório
 - ii. Aplicação um pouco lenta às vezes
 - iii. Sistema muito lento
- c) Com relação a frequência de erros
- i. Não percebi erro algum
 - ii. Ocorreram alguns erros

iii. Aconteceram muitos erros

3. SOBRE A UTILIDADE DO *LANGUAGE ADVISER*:

a) O *Language Adviser*, foi útil? ?

i. Sim

ii. Não

iii. Talvez

4. SOBRE AS FUNÇÕES DO *LANGUAGE ADVISER*:

a) O *Language Adviser* contribui para acelerar o seu aprendizado ?

i. Sim

ii. Não

iii. Talvez

5. SOBRE A APLICAÇÃO DA METODOLOGIA DA ESCOLA, ESPECIFICAMENTE NO QUE DIZ RESPEITO AS PRÁTICAS REFERENTES AO ESTUDO DOS DIÁLOGOS.

a) Considere as instruções a seguir, sendo executada por você, com e sem o apoio do *Language Adviser*: "Escute a faixa do dialogo correspondente a unidade e acompanhe no livro por cinco vezes consecutivas. Uma vez sem pausa e outra com pausa entre as falas", você percebeu alguma diferença ?

i. Não percebo diferença

ii. O *Language Adviser* facilita a execução das instruções

iii. O *Language Adviser* dificulta a execução das instruções.

ANEXO A – Recursos da Aplicação

Neste anexo são descritos todos os recursos que foram consumidas na aplicação, para estabelecer a comunicação do *font-end* com o *back-end* na criação do *Language Adviser*.

Tabela 1 – Recursos do Login

URI	MÉTODO	DESCRIÇÃO
<i>/login</i>	POST	Faz login na API.
<i>/login/redefinir</i>	PUT	Redefine a senha de usuário.

Fonte – (OLIVEIRA, 2018).

Tabela 2 – Recursos do Administrador

URI	MÉTODO	DESCRIÇÃO
<i>/administradores</i>	POST	Cadastra um administrador.
<i>/administradores</i>	GET	Lista todos os administradores.
<i>/administradores/:id</i>	GET	Busca um administrador pelo ID.
<i>/administradores/:id</i>	PUT	Edita os dados de um administrador.
<i>/administradores/:id/imagem</i>	PUT	<i>Upload</i> da imagem de perfil.
<i>/administradores/:id</i>	DELETE	Deleta um administrador.

Fonte – (OLIVEIRA, 2018).

Tabela 3 – Recursos do Contrato

URI	MÉTODO	DESCRIÇÃO
<i>/contratos</i>	POST	Cadastra um contrato.
<i>/contratos</i>	GET	Lista todos os contratos .
<i>/contratos/:id</i>	GET	Busca um contrato pelo ID.
<i>/contratos/:id</i>	PUT	Edita as informações de um contrato.
<i>/contratos/:id</i>	DELETE	Deleta um contrato.
<i>/contratos/ativo</i>	GET	Lista todos os contratos ativos.
<i>/contratos/expirado</i>	GET	Lista todos os contratos expirados.
<i>/contratos/:idEmpresa</i>	GET	Lista de contratos de uma empresa.
<i>/contratos/:idTermos</i>	GET	Lista todos os contratos de um termo.
<i>/contratos/:idRegiao</i>	GET	Lista todos os contratos de uma região.
<i>/contratos/adicionarTermo</i>	PUT	Adiciona um termo em um contrato.
<i>/contratos/removeTermo</i>	PUT	Remove um termo em um contrato.

Fonte – (OLIVEIRA, 2018).

Tabela 4 – Recursos do Diálogo

URI	MÉTODO	DESCRIÇÃO
<i>/dialogos</i>	POST	Cadastra um diálogo.
<i>/dialogos</i>	GET	Lista todos os diálogos.
<i>/dialogos/:id</i>	GET	Busca um diálogo pelo ID.
<i>/dialogos/:id</i>	PUT	Edita as informações de um diálogo.
<i>/dialogos/:id</i>	DELETE	Deleta um diálogo.
<i>/dialogos/:id/estudar</i>	PUT	Diálogo com todos os momentos.
<i>/dialogos/licao/:idLicao</i>	GET	Lista todos os diálogos de uma lição.
<i>/dialogos/:id/audio</i>	PUT	<i>Upload</i> do áudio para o diálogo.

Fonte – (OLIVEIRA, 2018).

Tabela 5 – Recursos da Empresa

URI	MÉTODO	DESCRIÇÃO
<i>/empresas</i>	POST	Cadastra uma empresa.
<i>/empresas</i>	GET	Lista todas as empresas.
<i>/empresas/:id</i>	GET	Busca um empresa pelo ID.
<i>/empresas/:id</i>	PUT	Edita as informações de uma empresa.
<i>/empresas/:id</i>	DELETE	Deleta uma empresa.

Fonte – (OLIVEIRA, 2018).

Tabela 6 – Recursos de Estudo

URI	MÉTODO	DESCRIÇÃO
<i>/estudos</i>	POST	Cadastra um idioma para um usuário.
<i>/estudos</i>	GET	Lista todos os idiomas estudados.
<i>/estudos/:id</i>	GET	Busca o idioma de um usuário.
<i>/estudos</i>	PUT	Deleta um idioma de estudo do usuário.
<i>/estudos/:id</i>	DELETE	Exclui todos os idiomas do usuário.

Fonte – (OLIVEIRA, 2018).

Tabela 7 – Recursos do Idioma

URI	MÉTODO	DESCRIÇÃO
<i>/idiomas</i>	POST	Cadastra um idioma.
<i>/idiomas</i>	GET	Lista todos os idiomas.
<i>/idiomas/:id</i>	GET	Busca um idioma pelo ID.
<i>/idiomas/:id</i>	PUT	Edita as informações de um idioma.
<i>/idiomas/:id/imagem</i>	PUT	<i>Upload</i> de imagem para o idioma.
<i>/idiomas/:id</i>	DELETE	Deleta um idioma.

Fonte – (OLIVEIRA, 2018).

Tabela 8 – Recursos das Lições

URI	MÉTODO	DESCRIÇÃO
<i>/licoes</i>	POST	Cadastra uma lição.
<i>/licoes</i>	GET	Lista todas as lições
<i>/licoes/:id</i>	GET	Busca uma lição pelo ID.
<i>/licoes/:id</i>	PUT	Edita as informações de uma lição.
<i>/licoes/:id</i>	DELETE	Deleta um lição.
<i>/licoes/usuarios/:idUsuario</i>	GET	Lista todas as lições de um usuário.
<i>/licoes/situacao/:idSituacao</i>	GET	Lista todas as lições de um situação.
<i>/licoes/avaliacao</i>	PUT	Avalia uma lição.

Fonte – (OLIVEIRA, 2018).

Tabela 9 – Recursos de Momentos

URI	MÉTODO	DESCRIÇÃO
<i>/momentos</i>	POST	Cadastra um momento.
<i>/momentos/:id</i>	PUT	Edita as informações de um momento.
<i>/momentos/:id/imagem</i>	PUT	<i>Upload</i> de imagem para o momento.
<i>/momentos/:id</i>	DELETE	Deleta um momento.

Fonte – (OLIVEIRA, 2018).

Tabela 10 – Recursos dos Níveis

URI	MÉTODO	DESCRIÇÃO
<i>/niveis</i>	POST	Cadastra um nível.
<i>/niveis</i>	GET	Lista todas os níveis.
<i>/niveis/:id</i>	GET	Busca uma nível pelo ID.
<i>/niveis/:id</i>	PUT	Edita as informações de um nível.
<i>/niveis/:id/imagem</i>	PUT	<i>Upload</i> de imagem para o nível.
<i>/niveis/:id</i>	DELETE	Deleta um nível.

Fonte – (OLIVEIRA, 2018).

Tabela 11 – Recursos das Regiões

URI	MÉTODO	DESCRIÇÃO
<i>/regioes</i>	POST	Cadastra uma região.
<i>/regioes</i>	GET	Lista todas as regiões.
<i>/regioes/:id</i>	GET	Busca uma região pelo ID.
<i>/regioes/:id</i>	PUT	Edita as informações de um região.
<i>/regioes/:id</i>	DELETE	Deleta uma região.

Fonte – (OLIVEIRA, 2018).

Tabela 12 – Recursos das Questões

URI	MÉTODO	DESCRIÇÃO
<i>/questoes</i>	POST	Cadastra uma questão.
<i>/questoes/:id</i>	GET	Busca uma questão pelo ID.
<i>/questoes/:id</i>	PUT	Edita as informações de uma questão.
<i>/questoes/:id</i>	DELETE	Deleta uma questão.

Fonte – (OLIVEIRA, 2018).

Tabela 13 – Recursos dos Questionários

URI	MÉTODO	DESCRIÇÃO
<i>/questionarios</i>	POST	Cadastra uma questionário.
<i>/questionarios/:id</i>	GET	Busca um questionário pelo ID.
<i>/questionarios/licao/:idLicao</i>	GET	Lista todos os questionários da lição.
<i>/questionarios/:id</i>	PUT	Edita as informações do questionário.
<i>/questionarios/:id</i>	DELETE	Deleta um questionário.

Fonte – (OLIVEIRA, 2018).

Tabela 14 – Recursos das Respostas

URI	MÉTODO	DESCRIÇÃO
<i>/respostas</i>	POST	Cadastra uma resposta.
<i>/respostas/:id</i>	GET	Busca uma resposta pelo ID.
<i>/respostas/:id</i>	PUT	Edita as informações de uma resposta.

Fonte – (OLIVEIRA, 2018).

Tabela 15 – Recursos de *Study*

URI	MÉTODO	DESCRIÇÃO
<i>/studies</i>	POST	Cadastrar <i>study</i> .
<i>/studies/usuarios/:idUsuario</i>	GET	Busca todos os <i>study</i> de um usuário.
<i>/studies/:id/respostas</i>	GET	Listar todos as respostas de um <i>study</i> .
<i>/studies/:id</i>	GET	Busca um <i>study</i> pelo ID.
<i>/studies/:id</i>	PUT	Edita as informações de um <i>study</i> .

Fonte – (OLIVEIRA, 2018).

Tabela 16 – Recursos das Situações

URI	MÉTODO	DESCRIÇÃO
<i>/situacoes</i>	POST	Cadastra uma situação.
<i>/situacoes</i>	GET	Lista todas as situações.
<i>/situacoes/:id</i>	GET	Busca uma situação pelo ID.
<i>/situacoes/idioma/:idIdioma</i>	GET	Lista todas as situações de um idioma.
<i>/situacoes/nivel/:idNivel</i>	GET	Lista todas as situações de um nível.
<i>/situacoes/:id</i>	PUT	Edita as informações de uma situação.
<i>/situacoes/:id/imagem</i>	PUT	<i>Upload</i> de imagem da situação.
<i>/situacoes/:id</i>	DELETE	Excluir uma situação.

Fonte – (OLIVEIRA, 2018).

Tabela 17 – Recursos dos Termos

URI	MÉTODO	DESCRIÇÃO
<i>/termos</i>	POST	Cadastra uma termo.
<i>/termos</i>	GET	Lista todos os termos.
<i>/termos/:id</i>	GET	Busca um termo pelo ID.
<i>/termos/idioma/:idIdioma</i>	GET	Lista todas os termos de um idioma.
<i>/termos/:id</i>	PUT	Edita as informações de um termo.
<i>/termos/:id</i>	DELETE	Excluir um termo.

Fonte – (OLIVEIRA, 2018).

Tabela 18 – Recursos do Usuário

URI	MÉTODO	DESCRIÇÃO
<i>/usuarios</i>	POST	Cadastra um usuário.
<i>/usuarios</i>	GET	Lista todos os usuários.
<i>/usuarios/:id</i>	GET	Busca um usuário pelo ID.
<i>/usuarios/:id/avaliacao</i>	GET	Avalia um usuário.
<i>/usuarios/ranking</i>	GET	Ranking dos usuários.
<i>/usuarios/:id/questionarios</i>	GET	Lista os questionários do supervisor.
<i>/usuarios/nivel/:idNivel</i>	GET	Lista os usuários de um nível.
<i>/usuarios/ativos</i>	GET	Lista os usuários ativos.
<i>/usuarios/desativos</i>	GET	Lista os usuários desativados.
<i>/usuarios/:idU/questionarios/:idQ</i>	PUT	Atribui um questionário ao usuário.
<i>/usuarios/:id/supervisor</i>	PUT	Atribui a função de supervisor.
<i>/usuarios/:id</i>	PUT	Edita as informações de um usuário.
<i>/usuarios/:id/imagem</i>	PUT	<i>Upload</i> da imagem de perfil.
<i>/usuarios/:idU/questionarios/idQ</i>	DELETE	Deleta o questionário do supervisor.
<i>/usuarios/:id</i>	DELETE	Deleta um usuário.

Fonte – (OLIVEIRA, 2018).