

**FACULDADE DE TECNOLOGIA DE CARAPICUÍBA**  
**CURSO TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**AIMÃ HARNEFER DE ASSIS SILVA**  
**JEFFERSON FELIPE MIRANDA**  
**JEOVANIR ALVES TEIXEIRA AFONSO**  
**LUCAS PEDROSA RIBEIRO**  
**RODRIGO LIMA FARIAS**

**WISER: Aplicativo *mobile* para a prática de idiomas**

**CARAPICUÍBA/SP**  
**2017**

**AIMÃ HARNEFER DE ASSIS SILVA  
JEFFERSON FELIPE MIRANDA  
JEOVANIR ALVES TEIXEIRA AFONSO  
LUCAS PEDROSA RIBEIRO  
RODRIGO LIMA FARIAS**

**WISER: Aplicativo *mobile* para a prática de idiomas**

Trabalho de Graduação apresentado ao curso de Tecnologia em Análise e Desenvolvimento de Sistemas, na Faculdade de Tecnologia de Carapicuíba, sob a orientação da Prof.<sup>a</sup> Dr.<sup>a</sup> Andreia Cristina Grisolio Machion.

**CARAPICUÍBA/SP**

**2017**

### Ficha Catalográfica

Silva, Aimã Harnefer de Assis; Miranda, Jefferson Felipe; Afonso, Jeovanir Alves Teixeira; Ribeiro, Lucas Pedrosa; Farias, Rodrigo Lima.

S578w      Wiser: Aplicativo *mobile* para a prática de idiomas– Carapicuíba – SP. 2017.  
41f.

Trabalho apresentado a FATEC - Carapicuíba como parte dos requisitos exigidos para a Conclusão do Curso de Tecnologia em Análise e Desenvolvimento de Sistemas.

Orientador: Andreia Cristina Grisolio Machion.

1. Tecnologia móvel. 2. Idiomas. 3. Análise e desenvolvimento de sistemas. I. Machion, Andréia Cristina Grisolio (orientador). II. Título.

004.61 CDD

**AIMÃ HARNEFER DE ASSIS SILVA  
JEFFERSON FELIPE MIRANDA  
JEOVANIR ALVES TEIXEIRA AFONSO  
LUCAS PEDROSA RIBEIRO  
RODRIGO LIMA FARIAS**

**WISER: Aplicativo *mobile* para a prática de idiomas**

Trabalho de Graduação apresentado ao curso de Tecnologia em Análise e Desenvolvimento de Sistemas, na Faculdade de Tecnologia de Carapicuíba, sob a orientação da Prof.<sup>a</sup> Dr.<sup>a</sup> Andreia Cristina GrisolioMachion.

**Aprovado em: 18/02/2017**

**BANCA EXAMINADORA**

---

Prof.<sup>a</sup> Dr.<sup>a</sup> Andreia Cristina GrisolioMachion

---

Prof. Rodrigo Bossini Tavares Moreira

---

Prof.<sup>a</sup>Dr<sup>a</sup> Silvia Maria Farani Costa

## RESUMO

Este trabalho apresenta uma nova opção para estudantes ou interessados em praticar suas habilidades de compreensão oral e escrita em diferentes idiomas utilizando celulares como ferramenta. A sociedade do conhecimento é marcada pela apropriação de informação e por novos métodos de aprendizagem que, certamente, contemplarão alunos que se adaptam com as novas tecnologias no mundo informatizado. Para especialistas, a melhor forma de se dominar um novo idioma é a imersão total. Isso inclui ler, ouvir e principalmente conversar com outras pessoas. A solução apresentada no decorrer deste trabalho possibilita a prática de idiomas de uma forma fácil.

**Palavras-chave:** Prática de Idiomas; Tecnologia; Celulares.

## **ABSTRACT**

This work presents a new option for students or interested in practicing their oral and written comprehension skills in different languages through the use of mobile phones. The knowledge society is marked by the appropriation of information and new methods of learning that will certainly include students who adapt to new technologies in the computerized world. For experts, the best way to master a new language is total immersion. This includes reading and especially talk with other people. The solution presented in the course of this work makes it possible to practice a language easily.

**Keywords:**Language Practice; Technology; Cellphones.

## SUMÁRIO

1 INTRODUÇÃO .....	10
1.1 Objetivo Geral .....	11
1.2 Objetivos Específicos .....	11
1.3 Justificativa .....	11
1.4 Comparativo entre aplicativos do mercado .....	12
2 PROJETO TÉCNICO .....	13
2.1 Conexões do Sistema .....	13
2.2 Requisitos.....	14
2.2.1 Requisitos Funcionais .....	14
2.2.2 Requisitos Não Funcionais.....	15
2.3 Diagramas do Sistema .....	16
2.4 Modelo de Banco de Dados .....	20
2.5 Padrão de Desenvolvimento de Projeto MVP .....	22
2.6 Diagramas de Classes .....	23
3 IMPLEMENTAÇÃO .....	26
3.1 Ambiente de Desenvolvimento.....	26
3.2 Web Service .....	26
3.2.1 Framework .....	26
3.3 Banco de Dados.....	26
3.4 Dependências .....	26
3.4.1 SDK do Facebook no Android .....	27
3.4.2 EventBus .....	28
3.4.3 Picasso.....	28
3.4.4 Retrofit.....	29
3.5 Telas.....	29

4 CONSIDERAÇÕES FINAIS .....	39
REFERÊNCIAS BIBLIOGRÁFICAS .....	40



## LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i> (Interface de Programação de Aplicativos)
DAO	<i>Data Access Object</i> (Objeto de Acesso a Dados)
DER	Diagrama de Entidade-Relacionamento
HTTP	Hypertext Transfer Protocol (Protocolo de Transferência de Hipertexto)
IDE	Integrated Development Environment (Ambiente de Desenvolvimento Integrado)
JSON	JavaScript Object Notation (Objeto de Notação JavaScript)
MER	Modelo Entidade-Relacionamento
MVC	<i>Model, View and Control</i> (Modelo, Visão e Controle)
MVP	<i>Model, View and Presenter</i> (Modelo, Visão e Apresentador)
MySQL	<i>My Structured Query Language</i> (Minha Linguagem de Consulta Estruturada)
SDK	Software Development Kit (Kit de Desenvolvimento de Software)
UML	Unified Modeling Language (Linguagem de Modelagem Unificada)

## 1 INTRODUÇÃO

O aprendizado de novos idiomas, principalmente o inglês, proporciona o aumento das oportunidades na vida pessoal e profissional dos indivíduos (Ribeiro, 2014). Vários são os fatores que justificam o interesse de grande parte da população mundial em aprender uma nova língua. Conseguir melhores vagas de emprego, conhecer novas culturas e comunicar-se com pessoas de todas as partes do mundo são alguns deles. Além disso, segundo Paixão (1998), grande parte das informações presentes na Internet está em inglês, assim como mais da metade de seus usuários se comunicam nesse idioma.

No mundo competitivo em que vivemos o conhecimento de pelo menos uma língua estrangeira é indispensável. Para Oliveira (2013, p. 50), os métodos de ensino tradicionais, entretanto, não são suficientes, é necessário oferecer ferramentas que permitam às pessoas praticar novas línguas. Devido à correria do dia a dia, é comum as pessoas não pararem para ler seus livros ou mesmo ouvir textos. Elas querem métodos que permitam praticar idiomas com outras pessoas, enquanto elas realizam outras tarefas. O momento atual requer a modernização da educação, como por exemplo, a utilização de dispositivos móveis como recurso pedagógico (Mousquer; Rolim, 2011, p. 2).

Segundo Lopes (2012, p. 10), “o acesso às redes eletrônicas tem favorecido a interatividade entre as pessoas que já podem se intercomunicar, trocar informações, executar tarefas em grupo, receber feedback on-line e até mesmo estreitar laços de conhecimento e amizade”.

As facilidades oferecidas pelas tecnologias da atualidade abriram espaço para um mercado de desenvolvimento de softwares que podem auxiliar a prática de novas línguas. Há que se considerar também que cada vez mais pessoas de todas as idades usam seus dispositivos móveis para realizar as mais diversas tarefas, incluindo aprender e exercitar assuntos de interesse.

A utilização de aplicativos para dispositivos móveis na prática de línguas, além de possibilitar uma maior comodidade aos usuários, também é capaz de contribuir para um vínculo mais dinâmico entre professores e alunos (Júnior, 2016).

Diante disso, propõe-se neste projeto a criação de um aplicativo para dispositivos móveis com o qual é possível encontrar pessoas por perto (em um raio previamente configurado pelo usuário) que estejam interessadas em praticar algum idioma.

### **1.1 Objetivo Geral**

Desenvolver e documentar um aplicativo para dispositivos móveis, compatível com o sistema operacional *Android*, que possibilite a prática de idiomas.

### **1.2 Objetivos Específicos**

- Permitir a busca por usuários próximos, utilizando APIs de geolocalização;
- Possibilitar a comunicação dos usuários por meio de mensagens de texto;
- Implementar um fórum de discussões para a troca de ideias por meio de tópicos;
- Permitir o login na aplicação por meio da API do Facebook;
- Implementar uma opção de “sugestão de assuntos” para a conversa, com base nas páginas curtidas pelos usuários no Facebook e em suas localizações, por exemplo.

### **1.3 Justificativa**

Cada vez mais pessoas procuram alternativas às técnicas comuns quando se propõem a praticar uma nova língua. Aplicativos para dispositivos móveis podem ser um meio mais rápido, fácil e prático para se atingir esse objetivo (Souza, 2015, p. 192).

De acordo com os dados de uma pesquisa realizada em 2012, sobre o uso de celulares no Brasil, é possível observar que 40% dos brasileiros utilizam o smartphone todo dia, 27% prefere ficar sem televisão a ficar sem o smartphone e 73% dos usuários não saem de casa sem seu dispositivo móvel (Google, 2012).

Além disso, as aplicações para celulares podem ser consideradas de fácil manuseio para a maioria dos usuários.

### 1.4 Comparativo entre aplicativos do mercado

Atualmente, existem muitos aplicativos no mercado voltados ao aprendizado de idiomas, de acordo com Novaes (2014), tem-se o Duolingo, Babbel e Rosetta Stone. Alguns deles possibilitam a interação com outros usuários e utilizam recursos multimídia para facilitar o aprendizado. O Quadro 1 ilustra um comparativo entre os principais aplicativos disponíveis e a solução proposta neste trabalho.

Comparativo entre aplicativos de Idiomas						
Recursos disponíveis	Aplicativos					
	Babbel	Busuu	Duolingo	Speaky	uTandem	Wiser
Exercícios e vocabulário de línguas	Sim	Sim	Sim	Não	Não	Não
Corretor ortográfico	Sim	Sim	Sim	Não	Não	Não
Bate-papo	Não	Sim	Não	Sim	Sim	Sim
Mensagens com áudios	Não	Não	Não	Não	Não	Não
Mensagens com imagens	Não	Não	Não	Não	Não	Não
Chamadas de vídeo	Não	Não	Não	Não	Não	Não
Fórum de discussões	Não	Não	Sim	Não	Não	Sim
Login com redes sociais	Sim	Sim	Sim	Sim	Sim	Sim
Escolha de contatos para estudo	Não	Não	Sim	Sim	Sim	Sim
Localizador de estudantes próximos	Não	Não	Não	Não	Sim	Sim
Sugestões de assuntos de interesse	Não	Não	Não	Não	Não	Sim

Quadro 1: Comparativo entre aplicativos do mercado x Wiser

Conforme descrito anteriormente, o conteúdo didático dos aplicativos disponíveis no mercado é produzido pelos próprios desenvolvedores, sejam eles especialistas em ensino ou não.

A proposta do Wiser é proporcionar a prática dos idiomas por meio da interação entre os usuários, os quais poderão trocar conhecimentos através do Bate-papo, Fórum de discussões ou pessoalmente, caso escolham se encontrarem usando o recurso de localização de estudantes próximos.

Como recurso inovador, o Wiser possui a funcionalidade de sugerir assuntos de interesse, a qual exibe temas (baseadas no perfil do usuário no Facebook) durante as conversas do Bate-papo.

## 2 PROJETO TÉCNICO

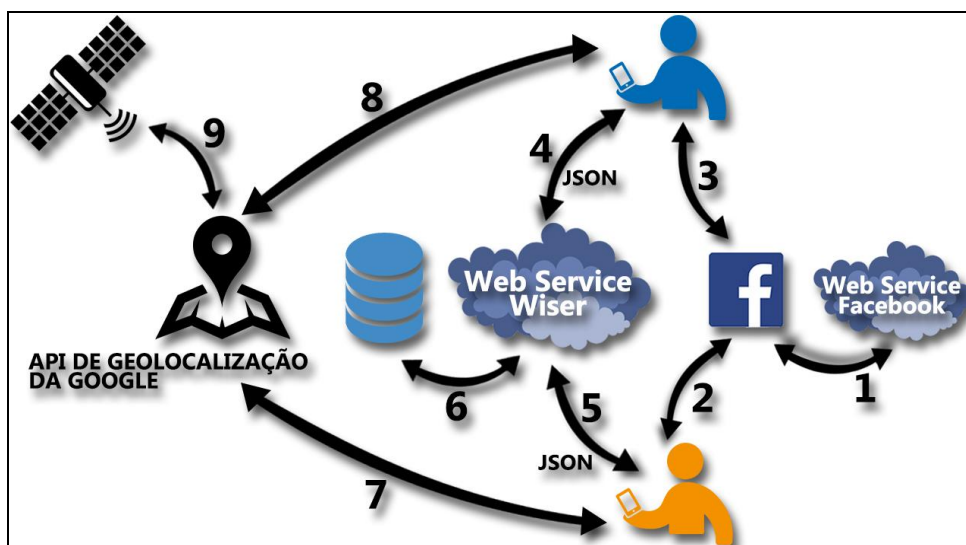
Neste capítulo aborda-se o funcionamento do sistema, bem como a modelagem dos dados, utilizando elementos da linguagem UML (*Unified Modeling Language*) e do padrão de projetos MVP.

### 2.1 Conexões do Sistema

O aplicativo Wiser permite encontrar pessoas por perto que estejam interessadas em praticar algum idioma, desde que essas possuam o aplicativo instalado.

A Figura 1 representa as conexões que o sistema realiza com a API do Facebook, o Web Service do Wiser e a API de geolocalização do Google. Por meio dela é possível perceber como a comunicação entre os usuários acontece.

Figura 1: Conexões do Sistema



Fonte: Elaborada pelos autores

Na conexão 1, acontece a comunicação entre a Graph API do Facebook e o Servidor da rede social, é neste momento que a API autentica o login e que retorna os dados requisitados, como nome do usuário e idade. Nas conexões 2 e 3, os usuários solicitam a Graph API que seus dados sejam autenticados para que eles possam acessar o Wiser. Nas conexões 4 e 5, o Wiser faz requisições diretamente ao Web Service, para enviar e receber informações como configurações, conversas dos usuários e discussões do fórum. Na conexão 6 acontece a conexão do Web

Service com o banco de dados, para que os dados sejam salvos no banco e possam ser acessados. Nas conexões 7 e 8, o Wiser se conecta com a API de geolocalização do Google para solicitar a localização atual do usuário. E por fim, na conexão 9, a API do Google se conecta com o GPS do dispositivo móvel e retorna a sua latitude e longitude.

## **2.2 Requisitos**

Requisitos de um sistema descrevem as funções e restrições do sistema que devem ser satisfeitas no seu processo de desenvolvimento (Sommerville, 2007). A seguir são apresentados os requisitos que foram identificados durante o desenvolvimento do aplicativo.

### **2.2.1 Requisitos Funcionais**

Requisitos Funcionais (RF) são declarações de serviços que o sistema deve prover. Descrevem detalhadamente as funções e comportamentos do sistema, suas entradas, saídas e exceções (Sommerville, 2007). Os RF's identificados para o aplicativo são:

[RF01] Permitir ao usuário fazer login utilizando a sua conta no Facebook;

[RF02] Permitir ao usuário escolher o seu idioma, seu nível de fluência, definir seus status e desativar a conta;

[RF03] Possibilitar a busca por usuários fisicamente próximos, oferecendo uma opção para escolher o raio máximo em Km;

[RF04] Permitir ao usuário adicionar novos contatos à sua lista de amigos;

[RF05] Possibilitar ao usuário visualizar todos os contatos que ele adicionou, em ordem alfabética;

[RF06] Permitir a comunicação dos usuários por meio de mensagens de texto (chat);

[RF07] Possibilitar ao usuário visualizar todas as conversas que ele tem, tanto com contatos quanto com desconhecidos;

[RF08] Permitir copiar e até apagar as conversas, selecionando uma ou mais mensagens;

[RF09] Possibilitar ao usuário bloquear outros usuários, impedindo assim que estes enviem novas mensagens para ele;

[RF10] Permitir ao usuário pesquisar por uma palavra ou frase dentro da conversa;

[RF11] Fornecer uma opção de “sugestão de assuntos” para a conversa, com base nas páginas curtidas pelo usuário no Facebook ou em sua localização, por exemplo;

[RF12] Possibilitar a criação de tópicos para a troca de ideias entre os usuários (Fórum de Discussões);

[RF13] Permitir ao usuário visualizar todas as discussões criadas por ele, tanto ativas quanto desativas, além dos tópicos dos demais usuários;

[RF14] Permitir ao usuário compartilhar uma discussão;

[RF15] Notificar ao usuário quando uma nova mensagem chegar;

[RF16] Possibilitar ao usuário se desconectar da sua conta do Facebook, permitindo assim fazer login em outra conta, caso queira.

## 2.2.2 Requisitos Não Funcionais

Os Requisitos Não Funcionais (RNF) não estão diretamente relacionados às funções específicas fornecidas pelo sistema, eles permitem especificar desempenho, proteção, disponibilidade e outras propriedades emergentes do sistema (Somerville, 2007). Os RNF's identificados para o aplicativo são:

[RNF01] O aplicativo deve ser implementado utilizando a linguagem Java;

[RNF02] O aplicativo deve ser desenvolvido sob a arquitetura MVP;

[RNF03] O aplicativo deve ser compatível com o sistema operacional Android, com versão igual ou superior a 5.0 (Lollipop);

[RNF04] O aplicativo deve solicitar ao usuário permissão para controle do GPS do aparelho, a fim de utilizar os recursos de geolocalização;

[RNF05] O aplicativo deve fornecer uma interface amigável e de fácil navegação;

[RNF06] O aplicativo deve armazenar os dados dos usuários em nuvem;

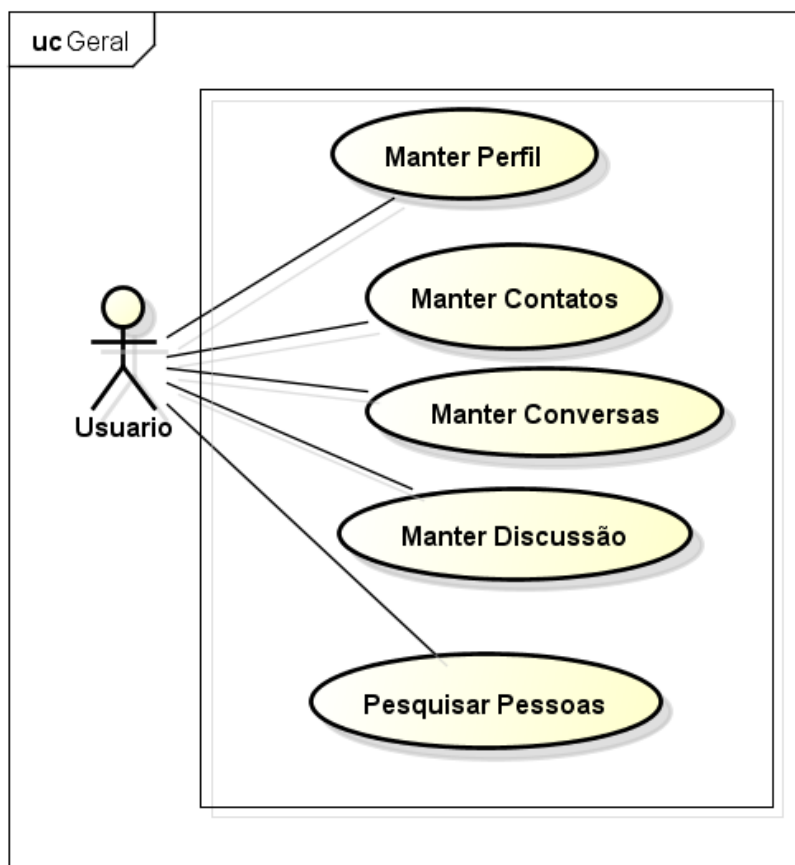
[RNF07] O aplicativo deve ter um tempo de resposta inferior a 3 segundos durante as conversas;

[RNF08] O aplicativo poderá ser instalado a partir da loja oficial da Google (*Play Store*) ou por meio de arquivo compilado (formato “.apk”).

## 2.3 Diagramas do Sistema

Nesta seção, descreve-se como o usuário interage com o sistema por meio de Diagramas de Casos de Uso e a ordem temporal em que as mensagens são trocadas entre os objetos envolvidos, utilizando Diagramas de Sequência. A Figura 2 ilustra o Diagrama de Caso de Uso Geral do aplicativo. Para detalhamento, foram escolhidos os casos de uso Manter Perfil e Manter Conversa, por representarem funcionalidades especiais do app. Os demais casos de uso seguem o mesmo padrão.

Figura 2: Diagrama de Caso de Uso Geral

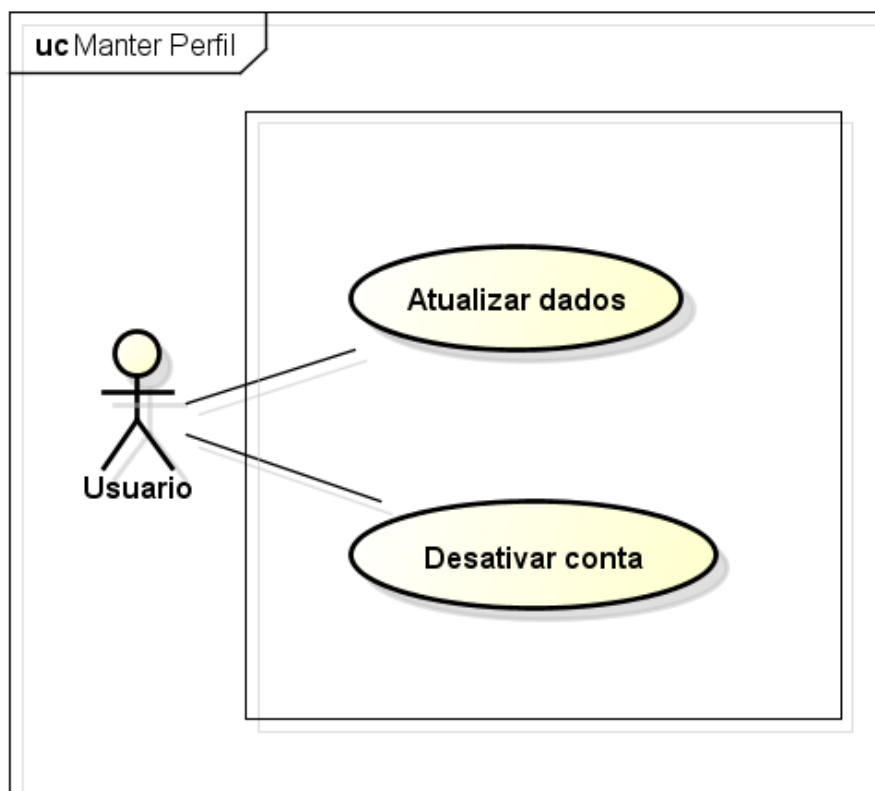


Fonte: Elaborada pelos autores

A Figura 3 ilustra o Diagrama de Caso de Uso Manter Perfil, que é descrito no Quadro 1, seguido do seu Diagrama de Sequência (Figura 4).



Figura 3: Diagrama de Caso de Uso Manter Perfil



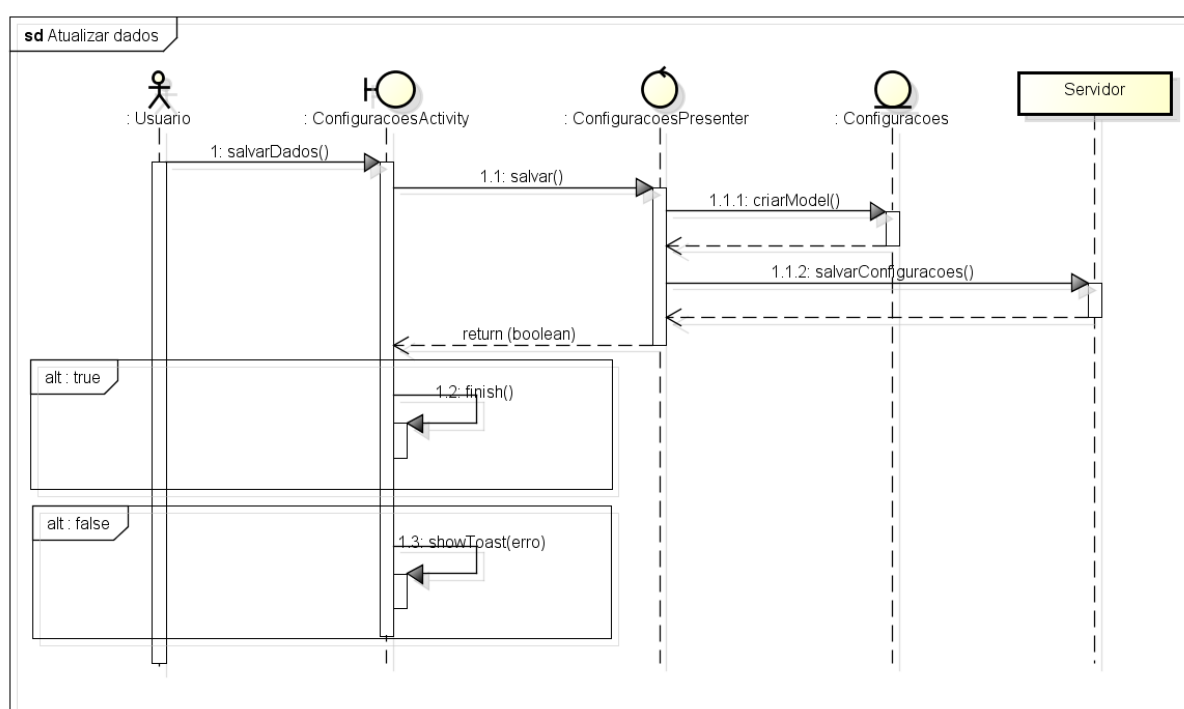
Fonte: Elaborada pelos autores

Quadro 1: Descrição do Caso de Uso Manter Perfil

Caso de Uso	Manter Perfil
Resumo	O usuário pode atualizar os dados de seu perfil ou mesmo desativar a sua conta
Ator	Usuário
Pré-Condições	Estar com o aplicativo aberto na tela “Configurações”
Pós-Condições	Atualização de perfil / desativação da conta
Fluxo Principal (FP)	<ol style="list-style-type: none"> <li>1. O usuário seleciona o menu da aplicação;</li> <li>2. O aplicativo exibe uma lista de opções;</li> <li>3. O usuário seleciona a opção “Configurações”;</li> <li>4. O aplicativo abre a tela de configurações;</li> <li>5. O usuário altera os dados de perfil;</li> <li>6. O aplicativo salva as alterações automaticamente.</li> </ol>
Fluxo Alternativo (FA)	<p>FA01 -O usuário escolhe a opção para desativar a conta;</p> <p>FA02 - O usuário cancela a operação, selecionando o botão voltar do aparelho móvel.</p>
Fluxo de Exceção (FE)	A operação é finalizada.

A figura a seguir ilustra o Diagrama de Sequência de uma atualização dos dados de um usuário na aba de configuração do aplicativo. Primeiramente, o usuário altera os dados e então clica em salvar. Isso acionará um método na Activity que indicará a Presenter que o usuário está tentando salvar os dados, assim o Presenter busca os dados na Activity e os salva em uma nova instancia da classe Configurações. Após isso, essa instancia é enviada para o servidor e retorna uma mensagem de sucesso ou falha. Caso seja um sucesso, a Activity é encerrada, e no caso de uma falha, é apresentado ao usuário uma mensagem de erro.

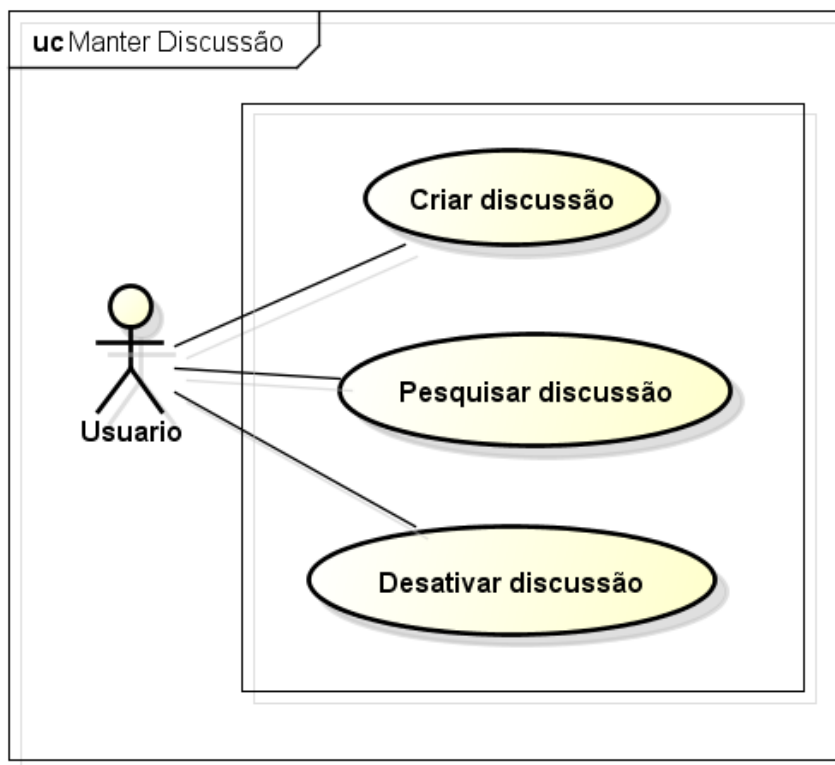
Figura 4: Diagrama de Sequência Atualizar Dados



Fonte: Elaborada pelos autores

A Figura 5 mostra o Diagrama de Caso de Uso Manter Discussão, que é detalhado no Quadro 2, seguido do seu Diagrama de Sequência (Figura 6).

Figura 5: Diagrama de Caso de Uso Manter Discussão



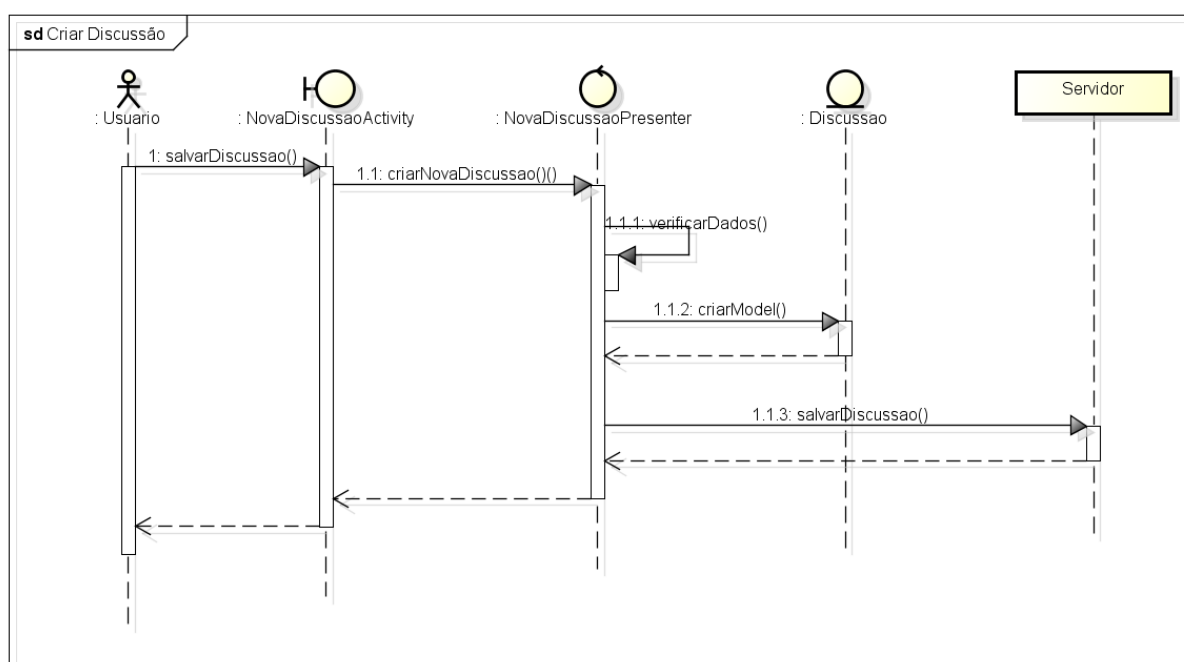
Fonte: Elaborada pelos autores

Quadro 2: Descrição do Caso de Uso Criar Discussão

Caso de Uso	CriarDiscussão
Resumo	O usuário cria uma nova discussão no aplicativo
Ator	Usuário
Pré-Condições	Estar com o aplicativo aberto na aba "Fórum"
Pós-Condições	Nova discussão criada
Fluxo Principal (FP)	<ol style="list-style-type: none"> <li>1. O usuário seleciona a aba "Fórum" do app e pressiona o botão "Nova";</li> <li>2. O sistema exibe a tela "Nova Discussão";</li> <li>3. O usuário preenche os campos "Título" e "Descrição" e pressiona o botão "Criar";</li> <li>4. O sistema solicita uma confirmação para salvamento da discussão;</li> <li>5. O usuário seleciona a opção "Sim";</li> <li>6. O sistema salva a discussão e a exibe no Fórum.</li> </ol>
Fluxo Alternativo (FA)	<p>FA01 - O usuário cancela a operação, selecionando o botão voltar do aparelho móvel;</p> <p>FA02 - O usuário cancela a operação, selecionando a opção "Não" para salvamento da discussão no Fórum.</p>
Fluxo de Exceção (FE)	A operação é finalizada.

A imagem a seguir demonstra o Diagrama de Sequência do processo de criação de uma nova discussão. Primeiramente, o usuário preenche os campos e clica em Salvar. Isso acionará a Activity que por sua vez indicará ao Presenter que o usuário está tentando criar uma nova discussão. O Presenter, após validar os dados, cria uma instância da classe Discussão onde salva estes dados e então envia essa nova instancia para o Servidor. Por fim, o aplicativo retorna para a lista de discussões no Forum.

Figura 6: Diagrama de Sequência Criar Discussão



Fonte: Elaborada pelos autores

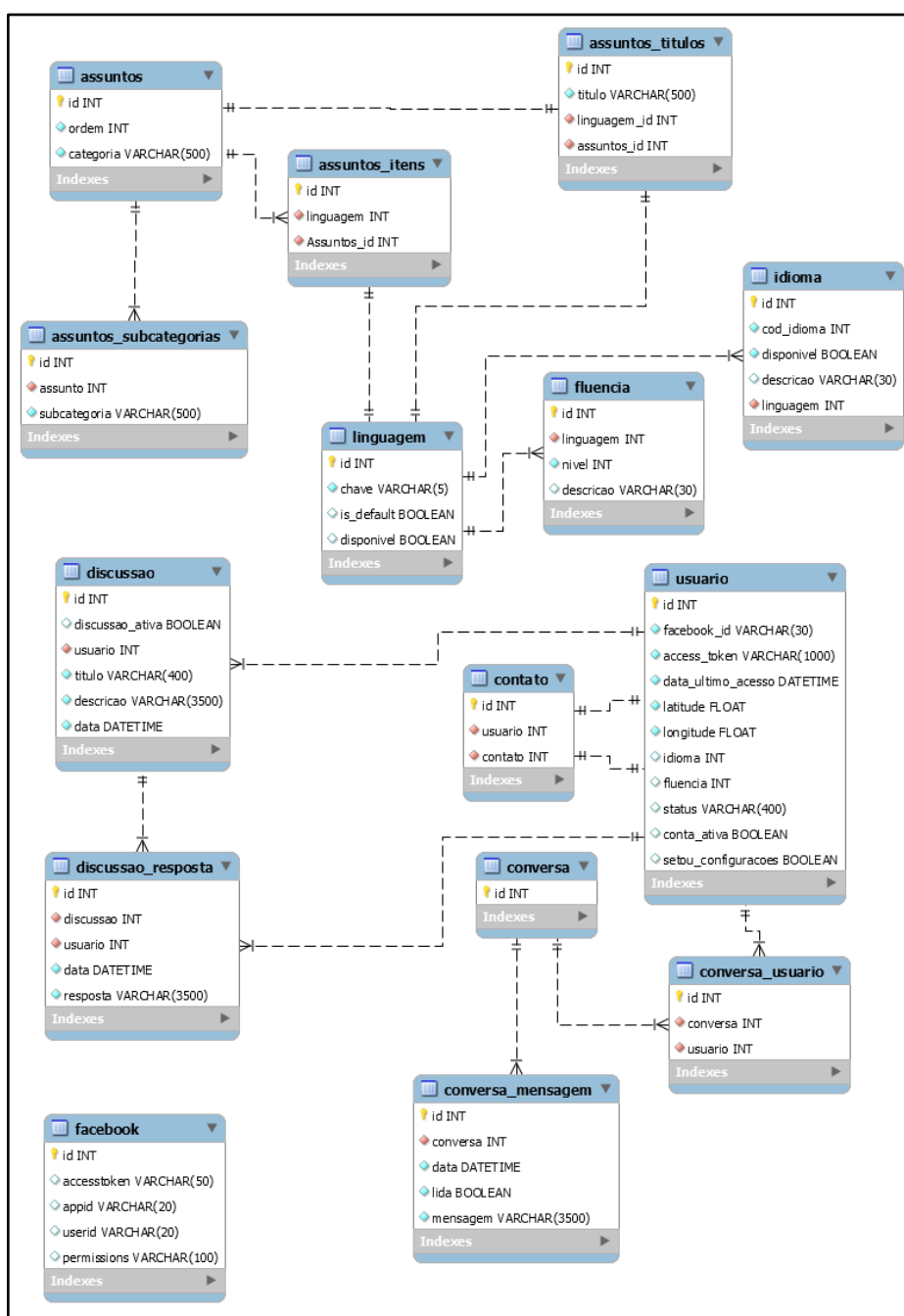
## 2.4 Modelo de Banco de Dados

Modelo de banco de dados é uma descrição dos tipos de informações que estão armazenadas em um banco de dados. Por exemplo, pode informar que o banco armazena informações sobre produtos e que, para cada produto, são armazenados seu código, preço e descrição. O modelo não informa quais produtos estão armazenados, apenas que tipo de informações contém. Existem modelos em diferentes níveis de abstração e com diferentes objetivos como: Conceitual, Físico e Lógico.

O modelo lógico compreende uma descrição das estruturas que serão armazenadas no banco e que resulta numa representação gráfica dos dados de uma maneira lógica, inclusive nomeando os componentes e ações que exercem uns

sobre os outros.(Heuser, 1998). A Figura 7 representa o diagrama de modelo lógico do Wiser.

Figura 7: Diagrama do Banco de Dados – modelo lógico



Fonte: Elaborada pelos autores

As tabelas do Diagrama da Figura 7 já estão na 3FN (Terceira Forma Normal), pois estão na 2FN e nenhuma coluna não chave é dependente de qualquer outra coluna não chave, definições importantes para este nível de normalização.

A tabela chamada “facebook” da Figura 7 não possui relacionamentos com as demais entidades, pois é utilizada apenas para armazenar os dados de configurações da API do Facebook.

## 2.5 Padrão de Desenvolvimento de Projeto MVP

O MVP é um padrão de projetos derivado do MVC e tem como conceito que toda a regra de negócio que normalmente iria ligar a interface do usuário com os dados seja movida para uma classe separada, assim, a interface do usuário apenas exibe os dados que a classe de controle determina.

Apesar de existir uma relação entre o conceito de *Controller* em MVC e *Presenter* em MVP, o Presenter tem algumas responsabilidades adicionais. Em MVP, a View deixa de ter uma ligação direta ao modelo, uma vez que o Presenter passa a ser responsável pelo fornecimento de dados à View. Essa é a principal diferença entre os dois padrões, tornando a View menos dependente do modelo (Ponte, 2009).

A comunicação entre o Presenter e a View passa também a ser feita através de uma interface, facilitando a substituição da View e permitindo testar a lógica de negócio independentemente da interface com o utilizador (Ponte, 2009).

No Wiser, este padrão de projeto foi aplicado devido a sua fácil expansão, manutenção e troca de dados entre *Activities* (Atividades), e foi estruturado da seguinte forma:

- Modelo (*Model*) - Formado pelas classes de domínio que contém informações como Usuários, Discussões, Conversas e seus possíveis comportamentos.
- Visão (*View*) - Composto por *Activities* e *Fragments* que recebem uma interação do usuário e então chama os métodos da camada de Apresentação.
- Apresentação (*Presenter*) - Formado pelas classes que contém as regras de negócio e intermediam a interface do usuário com os dados.

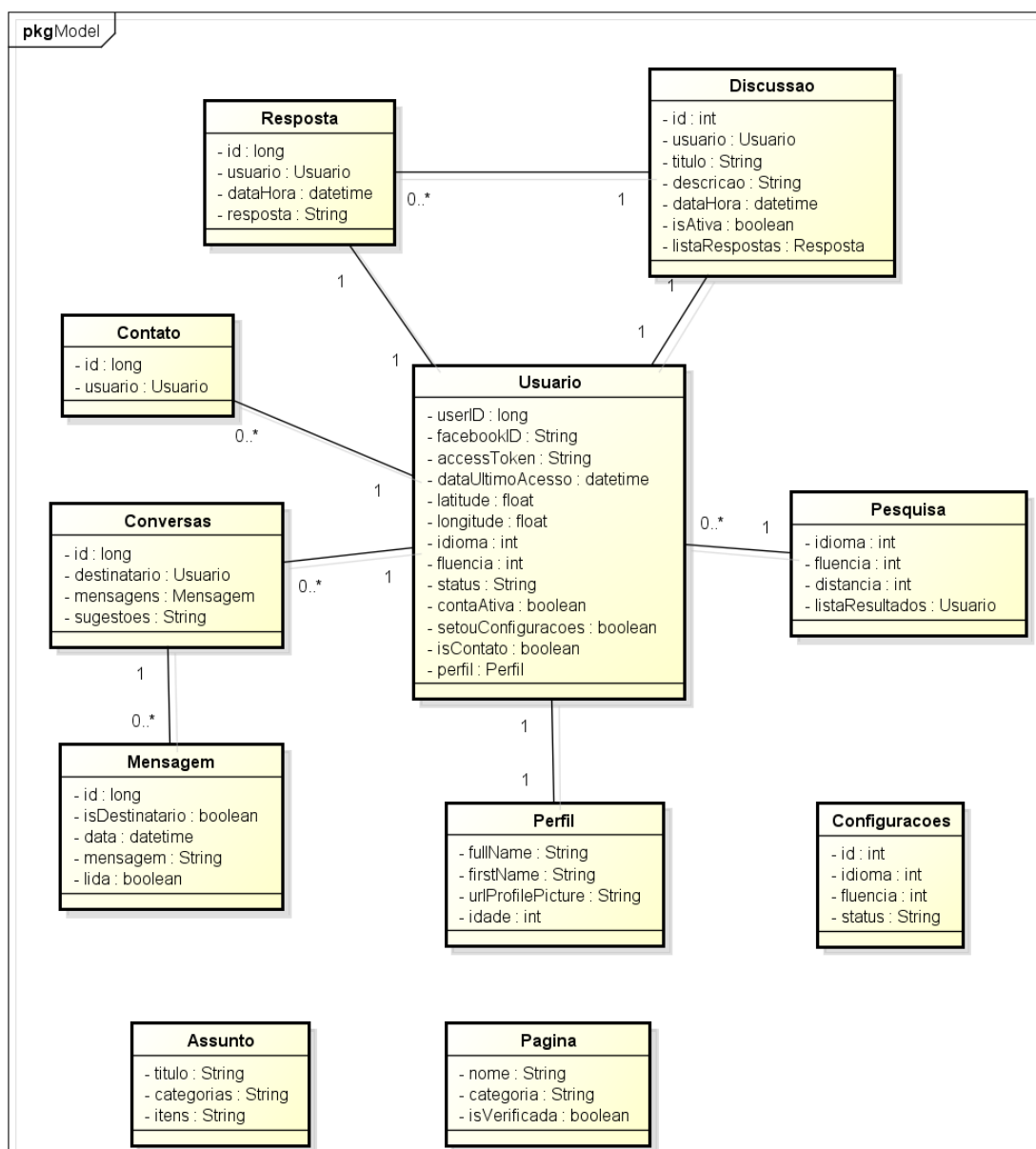
## 2.6 Diagramas de Classes

Os Diagramas de Classes são utilizados para a modelagem estática de objetos do sistema, com seus atributos e métodos, além de ilustrar as classes, as interfaces e os relacionamentos e associações entre elas (Larman, 2007).

Para clareza de apresentação, foram criados os Diagramas de Classe das camadas Modelo (*Model*), Visão (*View*) e Apresentação (*Presenter*) separadamente.

A Figura 8 representa o Diagrama de Classes da camada *Model*.

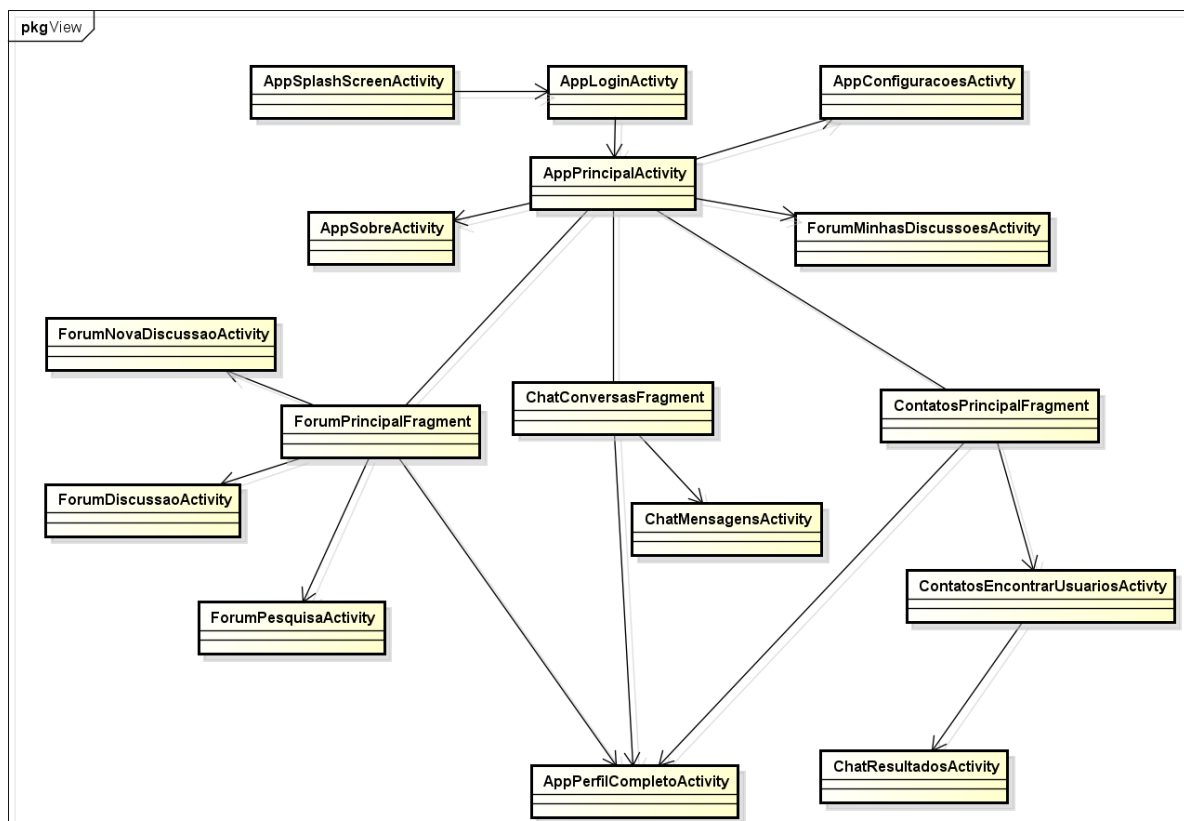
Figura 8: Diagrama de Classes—camada *Model*



Fonte: Elaborada pelos autores

A Figura 9 representa o Diagrama de Classes da camada *View*.

Figura 9: Diagrama de Classes –camada *View*

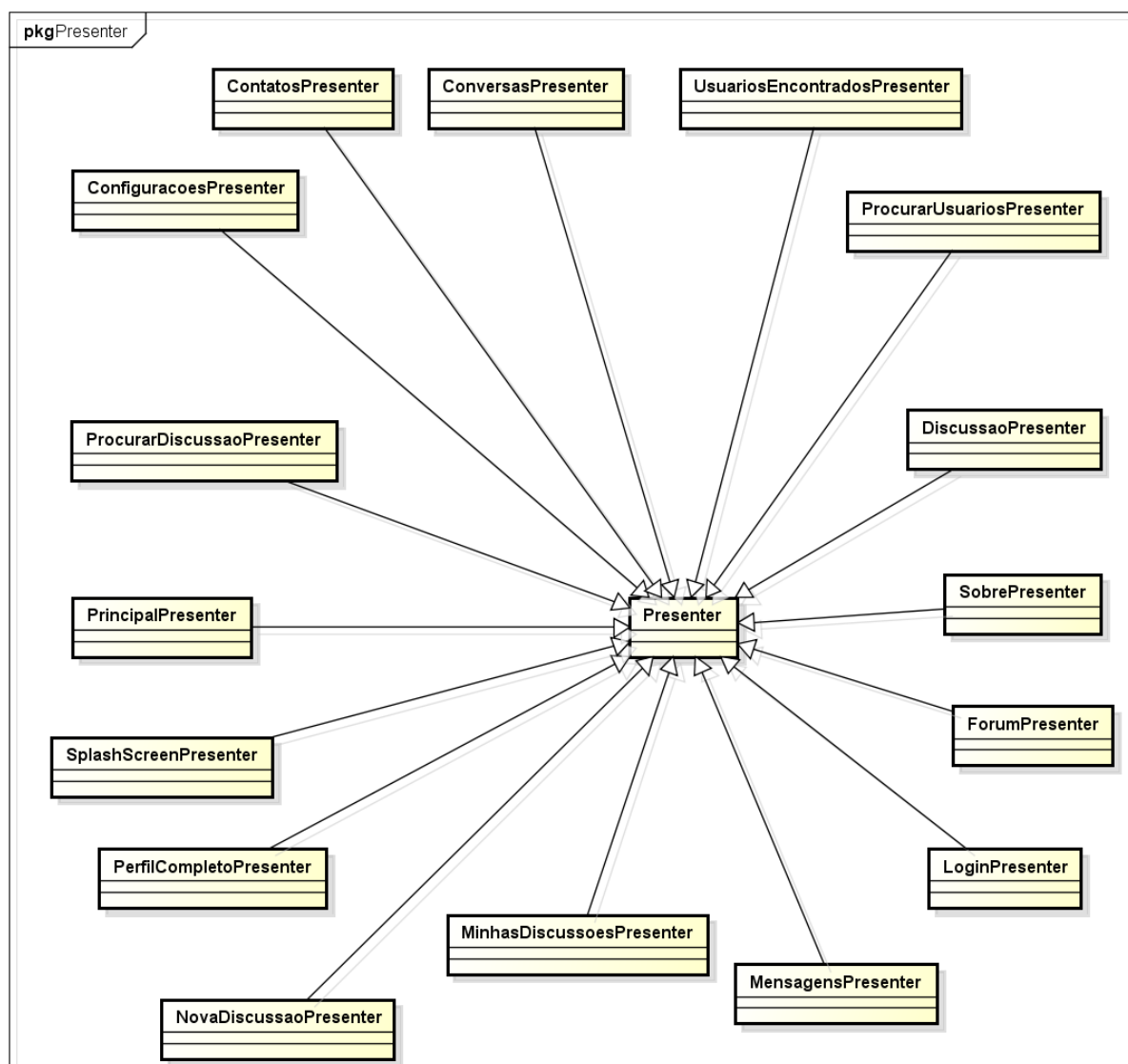


Fonte: Elaborada pelos autores



A Figura 10 representa o Diagrama de Classes da camada *Presenter*.

Figura 10: Diagrama de Classes –camada *Presenter*



Fonte: Elaborada pelos autores

## 3IMPLEMENTAÇÃO

Neste capítulo, mostram-se os detalhes de implementação do sistema. São apresentadas também algumas das telas e trechos de códigos utilizados na programação.

### 3.1 Ambiente de Desenvolvimento

O aplicativo Wiser foi desenvolvido na linguagem de programação Java, utilizando a IDE (*Integrated Development Environment*) *Android Studio*, da Google.

### 3.2Web Service

O Web Service foi criado para integrar a aplicação Wiser ao banco de dados. Para a transmissão de pacotes entre o Web Service e a aplicação, foi utilizado o padrão de formatação JSON.

#### 3.2.1 Framework

O *framework* escolhido para ser utilizado no Web Service foi o *Sails.js*. O mesmo torna fácil a criação de aplicações MVC para servidores em *Node.js*, fazendo com que o programador não se preocupe em escrever as *queries* no banco de dados, o que reduz código e facilita o desenvolvimento.

### 3.3Banco de Dados

O sistema gerenciador de banco de dados escolhido foi o MySQL, devido a sua alta confiabilidade e rapidez na execução. Além de ser compatível com diversos sistemas operacionais.

### 3.4Dependências

O Wiser faz uso dos recursos de diversas APIs do SDK do Facebook, que fornecem integrações com outros serviços, executam tarefas em segundo plano e diminuem as linhas de código.

### 3.4.1 SDK do Facebook no Android

O SDK do Facebook é um conjunto de ferramentas que ajudam a integrar uma aplicação Android ao Facebook. O Wiser utiliza-se de duas ferramentas do SDK, a primeira é o Login com Facebook, com essa ferramenta é possível realizar a autenticação dos usuários usando as credenciais do Facebook e receber as permissões desses usuários para recuperar informações em nome deles.

A segunda ferramenta que o Wiser utiliza é a Graph API, essa ferramenta permite que aplicações acessem dados dos usuários que concederem as permissões necessárias. O Wiser utiliza essa ferramenta para obter informações como nome do usuário, idade e paginas que ele curte para prover uma melhor experiência para o usuário.

A seguir, é demonstrado um trecho de código onde ocorre a implementação de uma requisição utilizando a Graph API:

```
private void request(final Usuario usuario, final String node, final
    Bundle parametros, final ICallback callback) {

    new Thread() {

        public void run() {

            new GraphRequest(createAccessToken(usuario),

                "/" + usuario.getFacebookID() + "/" + node,
                parametros, HttpMethod.GET,

                new GraphRequest.Callback() {

                    public void onCompleted(GraphResponse graphResponse)
                    {

                        if (graphResponse != null) {

                            callback.setResponse(graphResponse);

                        }

                    }

                })
                .executeAndWait();

        }

    }.start();

}
```

### 3.4.2 EventBus

O EventBus é uma API para Android que utiliza o padrão Publisher/Subscriber permitindo uma central de comunicação para classes desacopladas com poucas linhas de código – simplificando o código, removendo dependências e acelerando o desenvolvimento de aplicações. A comunicação com a EventBus é usada no serviço do Wiser para o carregamento de mensagens das conversas dos usuários, ao ser carregadas as mensagens do Web Service do Wiser, essas mensagens são distribuídas entre as Activities que exibem essas mensagens.

A seguir um exemplo de um registro de uma Activity, através de um Presenter (como o Wiser utiliza o padrão de projetos MVP, o registro da EventBus é realizado na classe Presenter ao invés da Activity), ao EventBus para receber os dados assim que eles forem carregados:

```
public void onStart() {  
    EventBus.getDefault().register(view);  
}  
  
public void onStop() {  
    EventBus.getDefault().unregister(view);  
}  
  
public void onEvent(List<Conversas> listaConversas) {  
    if (!listaConversas.isEmpty()) {  
        this.listaConversas = listaConversas;  
    }  
}
```

### 3.4.3 Picasso

A API Picasso é uma biblioteca para o download e armazenamento em cache de imagens para o Android. Essa biblioteca permite que o carregamento das

imagens seja feito em um segundo plano e com poucas linhas de código, além de possuir diversos recursos para melhorar a experiência do usuário com a aplicação.

#### 3.4.4 Retrofit

O Retrofit é uma API que realiza requisições HTTP no Android em segundo plano, possui alta escalabilidade, rapidez, segurança e possui suporte ao JSON através do Google GSON (biblioteca para serializar e deserializar objetos Java para JSON) para as respostas das requisições. O Wiser utiliza esta biblioteca para que seja realizada requisições ao Web Service sem que a aplicação fique travada até que seja executada a requisição.

### 3.5 Telas

Nesta seção, são apresentadas as interfaces das telas do Wiser, além de uma breve descrição de suas funcionalidades.

A Figura 11 exibe a tela “*Login*”, onde é possível se logar na aplicação utilizando a conta do Facebook. No exemplo da imagem, o usuário já havia se autenticado anteriormente, bastando confirmar com “Ok”.

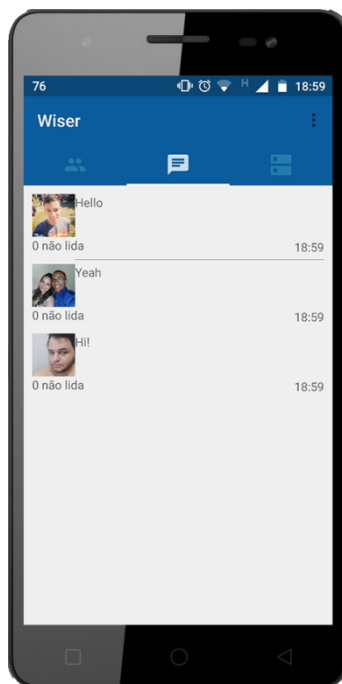
Figura 11: Tela *Login*



Fonte: Elaborada pelos autores

Após o Login, é apresentada a tela inicial do app Wiser (Figura 12), já selecionada por padrão na aba *Mensagens*, onde são listadas todas as conversas trocadas com os outros usuários.

Figura 12: Tela Inicial – aba *Mensagens*



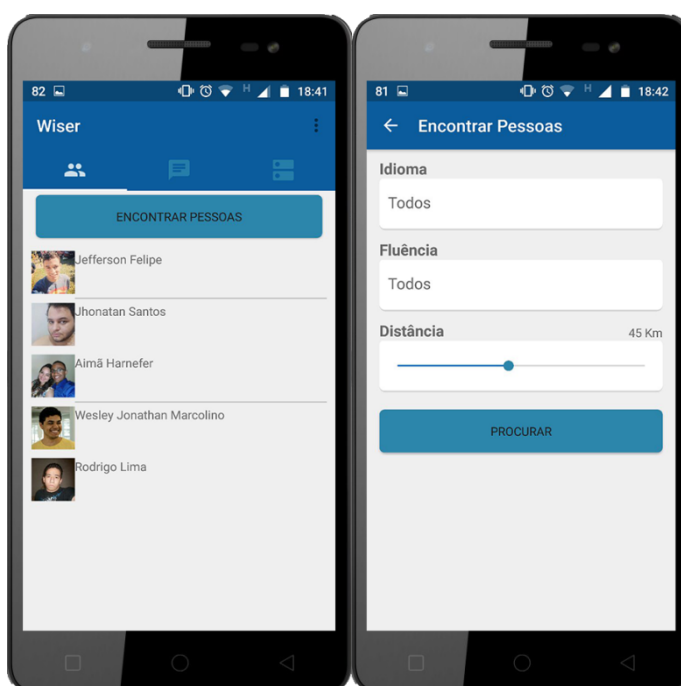
Fonte: Elaborada pelos autores

Ao escolher alguma conversa na aba “*Mensagens*” (Figura 12), é exibida a interface de “*Conversa*” (Figura 13), onde são listadas todas as mensagens trocadas com o usuário escolhido.

Figura 13: Tela *Conversa*

Fonte: Elaborada pelos autores

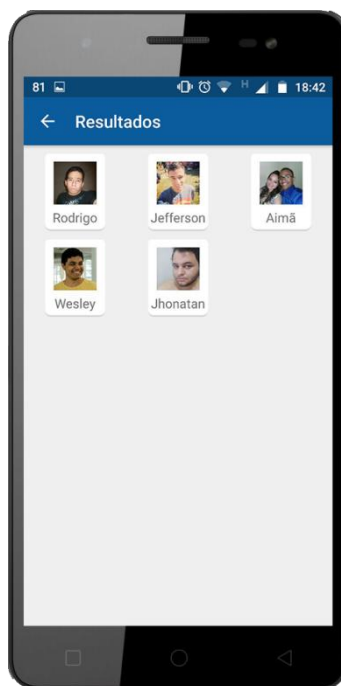
Na aba do lado esquerdo, temos a tela “*Encontrar Pessoas*”(Figura 14), onde é possível pesquisar por usuários fisicamente próximos, com base nos filtros de busca *Idioma*, *Fluência* e *Distância*.

Figura 14: Tela *Encontrar Pessoas*

Fonte: Elaborada pelos autores

Os usuários localizados são exibidos na tela “*Resultados*”(Figura 15).

Figura 15: Tela *Resultados*

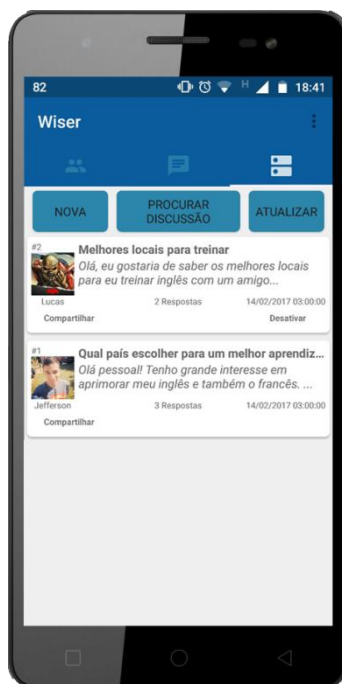


Fonte: Elaborada pelos autores

A Figura 16 mostra a aba “*Fórum*”, onde são listados todos os tópicos publicados por todos os usuários cadastrados. É possível criar uma Nova Discussão, Procurar por uma discussão específica ou responder as que forem do interesse.

Figura 16: Tela *Fórum*





Fonte: Elaborada pelos autores

A Figura 17exibe a tela “*Nova Discussão*”, onde é possível publicar uma discussão nova no Fórum.O usuário preenche os campos *Título* e *Descrição* e, em seguida, seleciona a opção “Criar”.

Figura 17: Tela *Nova Discussão*



Fonte: Elaborada pelos autores

Ao selecionar a opção “Criar”, é exibida a tela de confirmação de salvamento (Figura 18). Ao escolher “Sim”, a discussão é publicada no Fórum.

Figura 18: Tela de confirmação



Fonte: Elaborada pelos autores

Para visualizar todas as discussões criadas pelo usuário atualmente logado, basta selecionar a opção “*Minhas Discussões*” no menu suspenso (Figura 16). A tela “*Minhas Discussões*” é apresentada (Figura 19).

Figura 19: Tela *Minhas Discussões*



Fonte: Elaborada pelos autores

Ao selecionar uma das discussões, é exibida a tela Figura 20, onde são listadas todas as respostas dos usuários e o conteúdo da discussão.

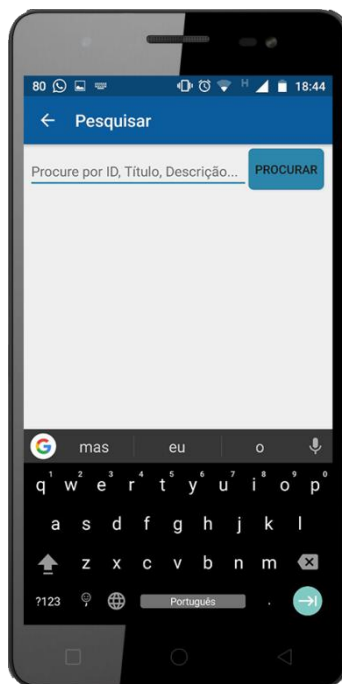
Figura 20: Tela *Discussões*



Fonte: Elaborada pelos autores

Para pesquisar por uma discussão específica, basta selecionar a opção “Pesquisar” no menu suspenso (Figura 20). A tela “Pesquisar”(Figura 21)é apresentada.

Figura 21: Tela *Pesquisar*



Fonte: Elaborada pelos autores

A Figura 22expõe os resultados encontrados para a busca da tela “Pesquisar”(Figura 21).

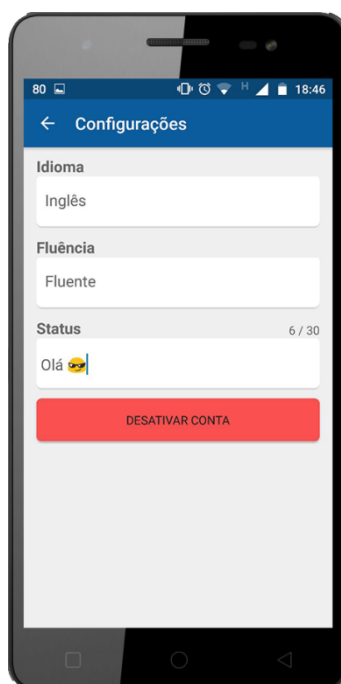
Figura 22: Tela *Pesquisar* – Resultados encontrados



Fonte: Elaborada pelos autores

Para alterar as configurações de perfil do usuário, basta selecionar a opção “*Configurações*” no menu suspenso (Figura 12). A tela “*Configurações*” é apresentada (Figura 23). É possível alterar o *Idioma*, nível de *Fluência* e *Status*, além de poder desativar a conta.

Figura 23: Tela *Configurações*



Fonte: Elaborada pelos autores

A Figura 24 exibe a tela “Sobre”, onde consta o nome do aplicativo, sua versão, seu ícone, ano e local de desenvolvimento e um link para compartilhar o app. Tal janela é acessada via menu suspenso, escolhendo a opção “Sobre”.

Figura 24: Tela Sobre



Fonte: Elaborada pelos autores

#### 4 CONSIDERAÇÕES FINAIS

Sabe-se que o conhecimento de novos idiomas acarreta num acréscimo considerável na formação intelectual e cultural dos indivíduos. Além disso, melhores são as oportunidades profissionais oferecidas aos que dominam outras línguas.

Este trabalho cumpriu com os objetivos inicialmente propostos, possibilitando a prática de línguas estrangeiras por meio de mensagens de texto, utilizando-se de recursos de triangulação para a localização dos usuários fisicamente próximos. O aplicativo desenvolvido permite que se faça *login* usando a conta do Facebook, automatizando assim o processo de cadastro. Além disso, é possível criar tópicos de discussões para a troca de ideias.

Os conhecimentos teóricos adquiridos durante as aulas foram aplicados neste projeto, formando assim uma base de referência para os próprios alunos. Como trabalhos futuros, pretende-se adicionar novas funcionalidades ao app Wiser, como conversas por vídeo e compartilhamento de arquivos (ex.: PDF, DOC), além de publicá-lo na loja *Play Store*, da Google.

## REFERÊNCIAS BIBLIOGRÁFICAS

GOOGLE; IPSOS OTX MediaCt. **OUR Mobile Planet: Brazil. Understanding the Mobile Consumer.**2012. Disponível em: <[http://www.mmaglobal.com/files/Brazil\\_English.pdf](http://www.mmaglobal.com/files/Brazil_English.pdf)>. Acesso em: 25 set. 2016.

HEUSER, C. **Projeto de Banco de Dados.** 4. ed. Rio Grande do Sul: Sagra Luzzatto, 1998.

JÚNIOR, José; **Tlearning - Aplicativo Para O Ensino Da Língua Inglesa.** Disponível em: <<http://www.sied-enped2016.ead.ufscar.br/ojs/index.php/2016/article/view/1925/798>>. Acesso em: 29 set. 2016.

LARMAN, C. **Utilizando UML e Padrões.** 3. ed. Porto Alegre. Bookman, 2007.

LOPES, D. V. **As novas tecnologias e o ensino de línguas estrangeiras.** 2012. Disponível em: <[http://www.unibratec.edu.br/tecnologus/wp-content/uploads/2012/08/tecnologus\\_edicao\\_06\\_artigo\\_01.pdf](http://www.unibratec.edu.br/tecnologus/wp-content/uploads/2012/08/tecnologus_edicao_06_artigo_01.pdf)>. Acesso em: 29 set. 2016.

MOUSQUER, T.; ROLIM, C. O. **A utilização de dispositivos móveis como ferramenta pedagógica colaborativa na educação infantil,** 2011. Santo Ângelo. Universidade Regional Integrada. Disponível em: <<http://www.santoangelo.uri.br/stin/Stin/trabalhos/11.pdf>>. Acesso em: 29 set. 2016.

NOVAES, R. **Os melhores app para te ajudar na hora de aprender um novo idioma,** 2014. Disponível em: <<http://www.psafec.com/blog/melhores-apps-idiommas>>. Acesso em: 15 out. 2016.

OLIVEIRA, N.; CAMPOS, F. **Tecnologia na Educação: a aprendizagem da Língua Inglesa por meio da rede social LiveMocha,** 2013. Disponível em: <<http://publicacoes.fatea.br/index.php/eccom/article/viewFile/592/422>>. Acesso em: 01 out. 2016.

PAIXÃO, R. **O idioma mundial: o que os adultos precisam fazer para aprender.** Revista Veja On-line. n. 1564. Editora Abril S.A, 1998. Disponível em: <[http://veja.abril.com.br/idade/educacao/160998/p\\_144.html](http://veja.abril.com.br/idade/educacao/160998/p_144.html)>. Acesso em: 01 out. 2016.

PONTE, L. Reengenharia de aplicações baseadas em componentes: Aplicação ao Processo Clínico, 2009, pág. 26-27. Disponível em: < <https://repositorio-aberto.up.pt/bitstream/10216/58958/1/000136959.pdf> >. Acesso em: 22 fev. 2017.

RIBEIRO, A.; SILVA, F.; ANDRADE, J.; SILVA, W. **O Uso da tecnologia Android para ensino preliminar da língua inglesa para crianças,** 2014. Disponível em: <<http://www.revistarefas.com.br/index.php/RevFATECZS/article/view/3/25>>. Acesso em: 01 out. 2016.

SOMMERVILLE, I., **Engenharia de Software,** 8. ed. São Paulo: Pearson, Editora: Addison Wesley, 2007.

SOUZA, I; ARRUDA, B. **Gamificação: o aprendizado de idiomas com aplicativos para dispositivos móveis,** 2015. Disponível em:



<[sinte.btv.ifsp.edu.br/index.php/SInTE/article/download/0001-0025/25](http://sinte.btv.ifsp.edu.br/index.php/SInTE/article/download/0001-0025/25)>. Acesso em: 01 out. 2016.