

**UNIVERSIDADE ESTADUAL PAULISTA "JÚLIO DE MESQUITA FILHO"**  
FACULDADE DE CIÊNCIAS - CAMPUS BAURU  
DEPARTAMENTO DE COMPUTAÇÃO  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

MARIANA ALMEIDA PEREIRA DIAS

**APLICAÇÃO PARA O ENSINO DE INGLÊS EM MOMENTO DE  
ESPERA NA TROCA DE MENSAGENS INSTANTÂNEAS -  
*MOBILE WAITCHATTER***

BAURU  
2018

MARIANA ALMEIDA PEREIRA DIAS

**APLICAÇÃO PARA O ENSINO DE INGLÊS EM MOMENTO DE  
ESPERA NA TROCA DE MENSAGENS INSTANTÂNEAS -  
*MOBILE WAITCHATTER***

Trabalho de Conclusão de Curso do Curso de Ciência da Computação da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Faculdade de Ciências, Campus Bauru.  
Orientadora: Profa. Dra. Simone das Graças Domingues Prado

BAURU  
2018

Mariana Almeida Pereira Dias APLICAÇÃO PARA O ENSINO DE INGLÊS EM MOMENTO DE ESPERA NA TROCA DE MENSAGENS INSTANTÂNEAS - *MOBILE WAITCHATTER* / Mariana Almeida Pereira Dias. – Bauru, 2018- 45 p. : il. (algumas color.) ; 30 cm.  
Orientadora: Profa. Dra. Simone das Graças Domingues Prado  
Trabalho de Conclusão de Curso – Universidade Estadual Paulista “Júlio de Mesquita Filho”  
Faculdade de Ciências  
Ciência da Computação, 2018.  
1. Wait Learning 2. Inteligência Artificial 3. Sistemas de Recomendação 4. Aplicativos Móveis 5. Micro Aprendizado

Mariana Almeida Pereira Dias

**APLICAÇÃO PARA O ENSINO DE INGLÊS EM  
MOMENTO DE ESPERA NA TROCA DE MENSAGENS  
INSTANTÂNEAS - *MOBILE WAITCHATTER***

Trabalho de Conclusão de Curso do Curso  
de Ciência da Computação da Universidade  
Estadual Paulista “Júlio de Mesquita Filho”,  
Faculdade de Ciências, Campus Bauru.

Banca Examinadora

**Profa. Dra. Simone das Graças Domingues Prado**  
Orientadora  
Departamento de Computação  
Faculdade de Ciências  
Universidade Estadual Paulista "Júlio de Mesquita Filho"

**Profa. Associada Roberta Spolon**  
Departamento de Computação  
Faculdade de Ciências  
Universidade Estadual Paulista "Júlio de Mesquita Filho"

**Prof. Dr. Wilson Massashiro Yonezawa**  
Departamento de Computação  
Faculdade de Ciências  
Universidade Estadual Paulista "Júlio de Mesquita Filho"

Bauru, 14 de novembro de 2018.

*Dedico esse trabalho a todas as mulheres da computação*

# Agradecimentos

Em primeiro lugar agradeço a Deus por estar concluindo mais uma etapa da minha vida com sucesso.

Agradeço a minha família, por todo o apoio, amor e carinho que me proporcionam desde que eu posso me lembrar.

Agradeço ao meu namorado, Nicolas Doneux, por ser meu parceiro e me incentivar em todos os momentos durante esses anos de faculdade, nada teria sido o mesmo sem a sua presença.

Agradeço aos meus amigos, por todas as risadas e por todas as madrugadas juntos, seja festejando ou trabalhando, sempre estivemos disponíveis uns para os outros.

Por fim, mas não menos importante, agradeço a todos os professores do curso por se dedicarem a formação de cada um de seus alunos.

*A flor que desabrocha na adversidade é mais rara e bela de todas*

*(Mulan, 1998)*

# Resumo

O cotidiano das pessoas é cada dia mais atarefado, e cada possibilidade de aproveitar seu tempo de maneira otimizada tende a ser aproveitada. Pensando nisso, é possível identificar diversos momentos de espera no dia a dia, incutidos em outras atividades, como esperar por um elevador, esperar o computador se conectar ao WiFi, esperar por uma página web carregar ou esperar por uma mensagem durante uma conversa de chat por exemplo. O “*Wait Learning*” ou “Aprendizado em Espera” é o conceito que sugere que tais situações sejam aproveitadas para ações de aquisição de conhecimento, por meio de atividades de micro aprendizado. Dessa forma, esse trabalho, chamado de *Mobile Wait Chatter*, se apropria de momentos de espera na troca de mensagens instantâneas para propor atividades de ensino de vocabulário do idioma Inglês. O aplicativo tem o objetivo de agregar valor ao veicular a aplicação em dispositivos móveis e utilizar um algoritmo de recomendação para personalizar a experiência do usuário no sistema.

**Palavras-chave:** Aprendizado em Espera, Sistema de Recomendação, Troca de Mensagens, Dispositivos Móveis .



# Abstract

People's life is becoming more and more busy, and every possibility of optimally enjoying their time tends to be taken advantage of. With this in mind, it is possible to identify several moments of day-to-day waiting in other activities, such as waiting for an elevator, waiting for the computer to connect to WiFi, waiting for a web page to load or waiting for a message during a chat for example. The "Wait Learning" is the concept that suggests that such situations should be used for knowledge acquisition actions, through micro-learning activities. Thus, this work, called the *Mobile Wait Chatter*, appropriates waiting times in the instant messaging exchange to propose activities of teaching English vocabulary. The app aims to add value by serving the app on mobile devices and using a recommendation algorithm to customize the user experience on the system.

**Keywords:** Wait Learning, Recommendation Systems, Chat, Mobile Devices.

# Lista de figuras

Figura 1 – Painel do <i>Wait Chatter</i> . . . . .	21
Figura 2 – Oportunidades de Espera na Troca de Mensagens Instantâneas . . . . .	22
Figura 3 – Fluxo <i>Wait Chatter</i> . . . . .	22
Figura 4 – Modelo de IA para Educação . . . . .	24
Figura 5 – Filtragem do Dados - Filtragem Híbrida . . . . .	27
Figura 6 – Árvore de Decisão para Recomendação por Classificação . . . . .	28
Figura 7 – Método <i>k-means</i> . . . . .	29
Figura 8 – Janela Principal do <i>Android Studio</i> . . . . .	31
Figura 9 – Interface do Banco de Dados do <i>Firebase</i> . . . . .	32
Figura 10 – Arquitetura do Projeto . . . . .	34
Figura 11 – Telas de Autenticação . . . . .	35
Figura 12 – Tela de Contatos . . . . .	36
Figura 13 – Tela de Perfil . . . . .	36
Figura 14 – Tela de Conversa . . . . .	37
Figura 15 – <i>Layout</i> Aplicação <i>Wait Learning</i> . . . . .	38
Figura 16 – Coleções do Banco de Dados . . . . .	40

# Lista de tabelas

Tabela 1 – Regras para Nível do Usuário . . . . .	39
Tabela 2 – Comparação entre o projeto desenvolvido e o <i>Wait Chatter</i> . . . . .	41

# Lista de abreviaturas e siglas

IA	Inteligência Artificial
IDE	Integrated Development Environment
XML	Extensible Markup Language

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>13</b>
1.1	Problema	14
1.2	Justificativa	15
1.3	Objetivos	15
1.3.0.1	Objetivo Geral	15
1.3.0.2	Objetivos Específicos	15
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>17</b>
2.1	Micro Learning	17
2.2	Wait Learning	18
2.3	Wait Chatter	20
2.4	Inteligência Artificial no Aprendizado	23
2.5	Sistemas de Recomendação	25
<b>3</b>	<b>MATERIAIS E MÉTODOS</b>	<b>30</b>
3.1	Android Studio	30
3.2	Linguagem de Programação <i>Kotlin</i>	31
3.3	<i>Google Firebase</i>	32
3.4	Metodologia	33
<b>4</b>	<b>DESENVOLVIMENTO</b>	<b>34</b>
4.0.1	Autenticação	34
4.0.2	Tela de Contatos e Perfil	35
4.0.3	Tela de Conversa e Aplicação de <i>Wait Learning</i>	37
4.1	Banco de Dados	39
4.2	Resultados e Discussões	40
<b>5</b>	<b>CONCLUSÃO</b>	<b>43</b>
5.1	Trabalhos Futuros	43
	<b>REFERÊNCIAS</b>	<b>44</b>

# 1 Introdução

Desde criança espera-se do ser humano que ele se dedique a tarefas primordiais. Inicia-se com o jardim de infância, a escola primária, secundária, o ensino médio, o curso preparatório, a universidade, o estágio, o trabalho, as especializações, além da prática de atividades físicas, a organização da casa, encontrar os amigos e ainda se atualizar e usufruir das tecnologias.

Em resumo, a rotina das pessoas, de uma maneira geral, não apresenta lacunas de tempo para se dedicar a atividades extracurriculares, como o aprendizado de um segundo ou terceiro idioma, por exemplo.

Além disso, a educação em si vem enfrentando diversos desafios em detrimento da tecnologia: como desenvolver formas de aquisição de conhecimento utilizando novas mídias culturais e tecnológicas; investigar a interligação entre os aspectos materiais da tecnologia nos processos de aprendizagem, educação e socialização; adicionar conceitos e metodologias integrando aspectos formais e informais de ensino como o *micro-learning*; construir um espaço lúdico que envolva aspectos culturais, históricos, comerciais e tecnológico; e usufruir dos dispositivos móveis e redes sociais, tão utilizados no mundo atual, para atividades educacionais. (HUG, 2006)

Tais desafios ainda devem considerar as dificuldades inerentes a aquisição de conhecimento. Focando na atividade conhecer um novo idioma, por exemplo, assim como afirma MONTREZOR e SILVA (2009), o indivíduo apenas desenvolve conhecimento por meio de estímulos seguidos de respostas, imitação e reforço contínuo. Situação que combinada com um dia a dia atribulado acaba acarretando na desistência de se dedicar a essa atividade.

Contudo, apesar da agenda comprometida, existem inúmeros momentos do dia em que são tipicamente desperdiçados devido a “espera”, isto é, momentos em que se aguarda por uma atividade se completar, como esperar o elevador, uma conexão de *WiFi*, ou a chegada de uma mensagem instantânea. (CAI; REN; MILLER, 2017)

Pensando nisso, CAI et al. (2015) introduziram o conceito de *Wait-Learning*, (“Aprendizado em Espera”), o qual pode ser classificado como uma extensão do *Micro-Learning* ou *Micro-Aprendizado*.

MALAMED (2015) apresenta a descrição de *Micro-Learning* como sendo a organização de experiências de aprendizado breves, projetadas para atingir um objetivo de aprendizado a longo prazo. Ou ainda como uma aprendizagem simples, de forma rápida, informal e contínua dentro de um ambiente pessoal.

HUG (2006) coloca o conceito de micro aprendizado como sendo uma das possibilidades de se superar os desafios enfrentados pelos tradicionais modelos de ensino, uma vez que oferecem uma maneira flexível e dinâmica que se adequam nas novas mídias.

A partir disso, o *Wait Suite*, conjunto de aplicações envolvendo *Wait Learning*, de CAI, REN e MILLER (2017) propõe preencher os momentos de espera com tais atividades breves para aumentar conhecimento pessoal em relação a uma língua estrangeira.

Uma das aplicações do *Wait Suite* é o *Wait Chatter*, que tem a proposta de ser utilizado durante uma conversa de *chat*. Na a troca instantânea de mensagens, a espera ocorre dentro da própria atividade. Devido a sua natureza parcialmente assíncrona, os usuários tendem a utilizar as pausas entre as conversas para realizar outras tarefas informais. (CAI et al., 2015). Além de encontrar momentos que possam ser proveitosos para o micro aprendizado, como o caso do *chat*, é interessante combinar tecnologias e metodologias para atividades de aprendizado.

Dentre os mencionados avanços da tecnologia, é possível destacar a inteligência artificial. Ela fica cada vez mais complexa e dinâmica, como por exemplo o *IBM Watson* que já incorporado em diagnósticos de câncer, análises financeiras, além de vencer humanos em programas de perguntas e respostas. (GREGO, 2012).

Conforme se introduz a tecnologia na área do aprendizado, acaba sendo natural aproveitar de conceitos e técnicas de IA para esse tipo de atividade. De acordo com PEREIRA (2012) sistemas especialistas, sistemas *fuzzy* e redes neurais, são as metodologias mais exploradas nas aplicações de ensino com IA. Elas tendem a proporcionar o acesso ao conhecimento para todos, além de se adequar ao perfil do usuário, o que possibilita que ele alcance seus objetivos de for mais rápida e eficiente.

## 1.1 Problema

As aplicações desenvolvidas por CAI, REN e MILLER (2017) para ensino de uma segunda língua mostram-se interessantes para o modo de vida atual. Entretanto, o conceito é novo e pouco disseminado.

No caso do *Wait Chatter*, a aplicação é construída dentro de uma aplicação de *chat* oferecida pela *Google* que não é mais utilizada, uma vez que o *Google Talks ou Chats* foi substituído pelo *Google Hangouts*.(ESTADAO CONTEUDO, 2017).

Outro ponto negativo é que o acesso ao *Wait Chatter* de CAI et al. (2015) é realizado apenas através do *desktop*, utilizando um navegador web específico. Sendo que a maioria das aplicações de mensagens instantâneas são utilizadas através de um *smartphone*.

Por fim, a sua apresentação de conteúdo não considera o usuário de forma particular. O que pode levar a uma perda de interesse nos casos em que o mesmo já possui

algum conhecimento sobre o idioma estudado, já que para esse tipo de usuário o processo iria demorar para agregar conhecimento.

## 1.2 Justificativa

Com a utilização de técnicas de Inteligência Artificial, seria possível identificar um comportamento do usuário em relação a seus conhecimentos prévios, e assim proporcionar para diferentes usuários uma experiência compatível com seu perfil.

No caso do projeto original de CAI, REN e MILLER (2017), a aplicação do *Wait Chatter* é voltada para pessoas cuja língua nativa é o inglês. Contextualizando, no caso de uma aplicação para uso no Brasil, como língua nativa é condizente utilizar o português e colocar o Inglês como objetivo de aprendizado.

Além disso, a construção da aplicação para ensinar em momentos de espera veiculada a dispositivos *mobile* tende a atingir uma maior quantidade de pessoas.

Tais alterações em relação ao contexto original poderiam além de diversificar a aplicação e melhorar o ensino do idioma, divulgar o conceito de *Micro Learning* e *Wait Learning* que podem ser explorados em outros projetos educacionais.

## 1.3 Objetivos

### 1.3.0.1 Objetivo Geral

Construir um aplicativo contendo atividades para o aprendizado de vocabulário e construções gramaticais de língua estrangeira, neste caso, o Inglês, para ser utilizado em um “momento de espera” durante uma conversa num aplicativo de troca de mensagens instantâneas, utilizando algoritmos de Inteligência Artificial com a intenção de melhorar a experiência do usuário de acordo com os consecutivos erros ou acertos das atividades propostas.

### 1.3.0.2 Objetivos Específicos

Dentre os objetivos específicos desse trabalho estão:

- a) Realizar uma revisão de literatura sobre *Wait Learning*
- b) Realizar uma revisão de literatura sobre o *Wait Chatter* e seu funcionamento
- c) Disseminar o conceito de aprendizado em espera
- d) Estudar sobre o funcionamento de uma aplicação de Inteligência artificial na educação
- e) Estudar sobre diferentes implementações de Sistemas de Recomendação



- f) Incorporar à aplicação de *Wait Learning* a particularização do usuário através de um sistema de recomendação e os padrões de IA na educação.

Os próximos capítulos descrevem as bases teóricas e o processo de desenvolvimento do *Mobile WaitChatter*. Tal que, o capítulo 2 apresenta a fundamentação teórica para o mesmo, definindo o *Wait Learnear* e o funcionamento do *Wait Chatter*, assim como o formato de aplicações de IA na educação e modelos de Sistemas de Recomendação.

O capítulo 3 traz as ferramentas utilizadas nesse trabalho e a metodologia implementada. Já o capítulo 4 mostra o desenvolvimento em si do *Mobile WaitChatter* e suas funcionalidades, além de uma comparação entre o *Mobile WaitChatter* e o *Wait Chatter* de CAI et al. (2015).

# 2 Fundamentação Teórica

## 2.1 Micro Learning

Tecnologias digitais estão transformando a estrutura do aprendizado, assim como os processos de distribuição e aquisição de conhecimento. Nesse cenário, é importante repensar os modelos clássicos e investigar novas possibilidades. O conceito de *microlearning* ou micro aprendizado surge como uma dessas novas possibilidades de acesso ao conhecimento. (HUG, 2006)

MALAMED (2015) define *microlearning* como a organização de experiências de aprendizado breves, projetadas para atingir um objetivo de aprendizado a longo prazo. Ou ainda como uma aprendizagem simples, de forma rápida, informal e frequentes dentro de um ambiente pessoal.

De acordo com HUG (2006), uma atividade de micro aprendizado é caracterizada por seu:

- a) Tempo: atividades rápidas, que demandam um curto período de tempo
- b) Conteúdo: pequenas e simples unidades de conteúdo
- c) Currículo: divisão do conteúdo em módulos
- d) Forma: apresentado em fragmentos, etapas
- e) Processo: atividades integradas e iterativas
- f) Tecnologia: veiculado em algum tipo de mídia
- g) Tipo de Aprendizado: atividades repetitivas e orientadas a um problema específico

Dentre as vantagens de se engajar no *Micro Learning* estão os resultados imediatos, pois permite acesso rápido ao conhecimento e conclusões a cada etapa; as formas diversas, podem ser encontrados em modelos estruturados, como textos e apostilas, ou não estruturados, como áudios e vídeos, de aprendizado, como por exemplo video aulas e jogos online; o baixo custo, já que representa uma situação autodidata sem a necessidade de professores; baixo custo de desenvolvimento, já que objetivo é ter conteúdos simplificados, sem grandes produções; e a facilidade de acesso, que pode ser uma solução para um cotidiano atribulado. (MALAMED, 2015)

## 2.2 Wait Learning

Pequenos momentos de espera ocorrem diariamente no cotidiano das pessoas por diversos motivos, como a espera em uma fila, a espera de uma caixa de *e-mail* carregar, a espera pelo elevador chegar, ou a espera de uma mensagem de texto.

Aproveitar tais momentos para desenvolver uma atividade de aprendizado, como a de um novo idioma, pode ser eficiente para pessoas que não poderiam, pela rotina atribulada, se comprometer com a tarefa como sendo prioridade.

Além disso, o ser humano tende a realizar multitarefas, de forma a aproveitar ao máximo o seu tempo, principalmente para evitar momentos negativos ou aversivos. Como por exemplo, estudantes que trocam mensagens de maneira a balancear o tédio de suas lições de casa. É de senso comum que esperar por alguma situação acontecer ou se concretizar se caracteriza como um momento irritante as pessoas. (CAI; REN; MILLER, 2017)

A partir dessas considerações, pode-se definir *Wait Learning* como sendo a realização de atividades de micro aprendizado em momentos de espera do dia a dia. (CAI; REN; MILLER, 2017).

CAI, REN e MILLER (2017) defendem que as pessoas são ainda mais receptivas com atividades em momentos de espera porque ocorrem quando uma atividade primária já está bloqueada, de forma que a pessoa já estará disponível para realizá-la sem interromper nenhum processo importante. Dessa forma, ao se utilizar gatilhos para atividades de *micro-learning* ou micro-aprendizado em momentos de espera, espera-se que o indivíduo não encontre problemas para engajar com a mesma.

Projetar tarefas de aprendizado em espera ou *Wait Learning* requer decisões sobre o tempo que a espera dura, complexidade da situação posta em espera, a frequência com que ela ocorre, e como o será a interação com a atividade de aprendizado.

Existem diversas situações de espera no dia a dia, e seus tempos variam desde segundos a horas. As atividades de micro aprendizado se encaixam melhor naquelas que levam de segundos a poucos minutos, já que tem o propósito de estarem intrínsecas, sem demandar esforço para serem executadas.

Outra questão muito importante em relação ao tempo de uma atividade de *Wait Learning* é o tempo gasto para sair da atividade central para a secundária e vice versa. A maneira que se realiza a troca entre a tarefa primária e a de aprendizado deve ser o mais transparente, isto é, imperceptível possível. O tempo tomado para a troca de uma tarefa pela outra deve ser consideravelmente menor que o tempo de espera em si, caso contrário o usuário não se interessaria em interagir com a atividade secundária, uma vez que não conseguiria concluí-la durante o período de espera.

Essa atividade de aprendizado pode ser tida como uma iteração periférica. Uma iteração periférica tem a pretensão de ser uma iteração diária que se incorpora à rotina com a realização de mínimo esforço. Ou seja, esse tipo de atividade deve ser fácil de ser inicializada e fácil de ser descartada, de maneira que não representa um empecilho para a execução da atividade central ou primária. (CAI; REN; MILLER, 2017)

Considerando o tipo da espera, isto é, o que se está esperando ocorrer, é preciso entender a capacidade de atenção do ser humano. CAI, REN e MILLER (2017) citam KAHNEMAN (1973) sobre sua “Teoria de Recurso”, a qual defende que há um único conjunto de recursos de atenção que pode ser dividido entre multitarefas, isto é, a mente aloca e libera dinamicamente recursos durante a execução de uma tarefa, o que resulta em níveis flutuantes da capacidade de atenção. Diferentes atividades recebem diferentes quantias de recursos de atenção dependendo de fatores como complexidade, esforço e empolgamento.

Outras teorias caracterizam atenção como um filtro seletivo, que seleciona quantas mensagens recebem atenção, separando-as entre atendidas ou não. Apenas mensagens que passam por esse filtro ficam na memória a curto prazo. Portanto, apresentar uma atividade secundária durante tarefas menos minuciosas e com um grau maior de facilidade tende a ser menos perturbador, uma vez que há uma menor demanda de recursos de atenção. Por exemplo, em uma situação de espera de uma apresentação do seu trabalho para o chefe demandará concentração e grau de atenção na atividade central, a apresentação, mesmo no momento da espera. Já esperar pelo elevador chegar, reduz consideravelmente a quantidade e dificuldade de informações associadas a tarefa primária, sendo assim mais provável o engajamento com uma atividade de segundo plano para entretenimento durante a espera.

A frequência com a qual a pessoa se depara com determinado momento de espera também tem papel relevante na proposta de um exercício de *Wait Learning*. É ideal que o momento aconteça com regularidade já que o aprendizado de um conteúdo está associado com a repetição do contato com o mesmo. Instigar a pessoa a interagir com a aplicação de aprendizado também é um ponto a ser levantado. O conteúdo pode ser apresentado de três formas: *Hard Notifications* ou notificações intensas envolvem aquelas que possuem painéis e textos com animações, como *pop-ups* por exemplo; *Soft Notifications* ou notificações suaves seriam aquelas que mantêm o painel fixo, mas alteram o texto; e, por fim, a *Static Notification* ou notificações estáticas são caracterizadas por texto e painel fixo. O equilíbrio das *Soft Notifications* geralmente representam a melhor escolha, já que, no caso das *Hard Notifications* que alteram inclusive seu posicionamento, podem comprometer a atividade central, e no caso das *Static Notifications* é provável que o usuário se acostume com a visualização e não se instigue a interagir com a tarefa.

Pensando no projeto de situações para *Wait Learning*, CAI, REN e MILLER (2017)

conduziram uma pesquisa sobre tais momentos através de um conjunto de aplicações chamadas de *Wait Suite*, utilizando o ensino de língua estrangeira (Espanhol e Francês) como sendo o objeto de aprendizagem. O *Wait Suite* é composto pelas aplicações:

- a) *Elevator Learner*: usado enquanto o usuário espera pelo elevador
- b) *Pull Learner*: usado após o usuário requisitar uma atualização de um conteúdo de um aplicativo móvel
- c) *WiFi Learner*: usado enquanto o usuário espera pelo computador conectar ao *WiFi*
- d) *Email Learner*: usado na espera de um *e-mail* ser corretamente enviado
- e) *Wait Chatter*: usado enquanto o usuário espera a resposta de uma mensagem instantânea

## 2.3 Wait Chatter

Essa seção irá tratar do aplicativo *Wait Chatter* desenvolvido por CAI et al. (2015), que serviu de base e inspiração para o desenvolvimento deste trabalho.

O *Wait Chatter* é uma das aplicações desenvolvidas por CAI et al. (2015), que compõe o *Wait Suite* discutido na seção 2.2. Ele constitui em uma extensão do navegador *Google Chrome* e é veiculado através do depreciado *Google Chat*, que foi substituído pelo *Google Hangouts*. (ESTADAO CONTEUDO, 2017)

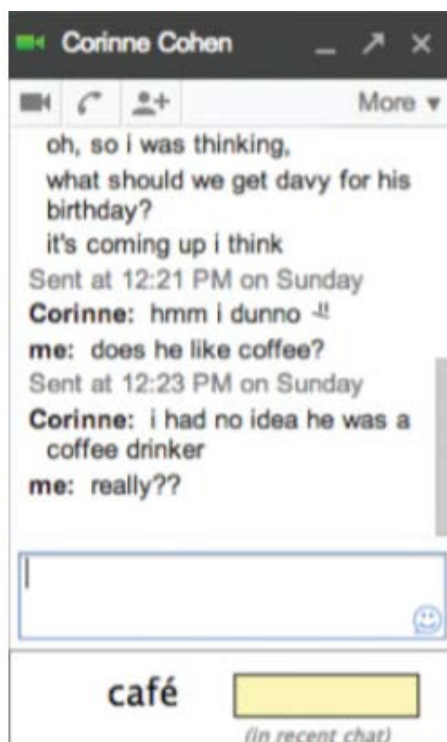
A aplicação é apresentada dentro da caixa de *chat* como pode ser visto na Figura 1 e ela tem como objetivo instigar o aprendizado de língua estrangeira, sendo propostas Espanhol e Francês durante momentos de espera na troca de mensagens instantâneas.

Segundo CAI et al. (2015), explorar a troca de mensagens instantâneas, popularmente conhecido com *chat*, para desenvolver uma atividade de *Wait Learning* é propício e interessante, uma vez que momentos de espera ocorrem dentro da atividade em si. Assim, CAI et al. (2015) realizaram estudos para detectar os principais momentos de espera dentro de um *chat*:

- a) “*i sent*”: espera que ocorre logo após o usuário enviar uma mensagem
- b) “*you typing*”: espera que ocorre enquanto o usuário espera uma resposta

A Figura 2 mostra o ciclo de ações da troca de mensagens instantâneas, e o posicionamento dos momentos de esperas detectados.

O *Wait Chatter* segue os padrões de projeto de *Soft Notification* discutidos na Seção 2.2. O vocabulário apresentado aparece no painel por 10 segundos, caso seja ignorado, a informação sobre o exercício desaparece, mas o painel se mantém fixo. O exercício

Figura 1 – Painel do *Wait Chatter*

Fonte: CAI, REN e MILLER (2017)

é apresentado ao se iniciar a conversa, e nos momentos de espera "*i sent*" e "*you typing*" mencionados.(CAI et al., 2015)

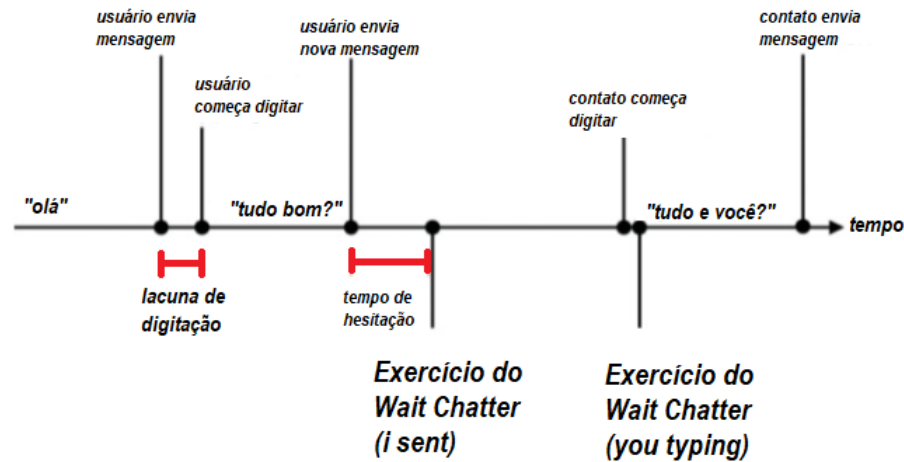
Após o usuário completar o exercício, ele pode navegar para o próximo clicando na seta a direita do painel ou apertando a seta para direita no teclado do computador. Esta funcionalidade permite ao usuário adequar a quantidade de atividades com a quantidade de tempo da espera.

Para os exercícios de ensino de língua estrangeira veiculados no *Wait Chatter*, CAI et al. (2015) consideraram apenas substantivos, porque, diferente de verbos, eles são mais simples de serem traduzidos e contextualizados.

Sempre que possível, a aplicação tenta realizar a contextualização dentro da conversa do *chat* do vocabulário. Dessa forma, quando uma mensagem é enviada ou recebida, o *Wait Chatter* procura por substantivos, e então os traduz usando *Google Translate*. Na tentativa de garantir o significado, a frase inteira também é traduzida, verificando se no contexto o substantivo também apresenta o mesmo significado. Caso não seja possível encontrar um substantivo válido na mensagem, a aplicação submete palavras descontextualizadas.(CAI et al., 2015)

CAI et al. (2015) divide as atividades de aprendizagem em duas maneiras: *Leitura* e *Quiz*. Primeiro, o vocabulário é apresentado em modo de *Leitura*, no qual o usuário pode

Figura 2 – Oportunidades de Espera na Troca de Mensagens Instantâneas



Fonte: baseada em CAI et al. (2015)

ler a palavra na segunda língua, e então revelar seu significado, através do botão *reveal*. Após ler a tradução da palavra, o usuário deverá indicar se já a conhecia ou não. No caso negativo, esse vocabulário segue para atividade em modo *Quiz*.

Já o modo *Quiz* consiste em apresentar a palavra na língua estrangeira e pedir que o usuário escreva-a na sua língua nativa ou apresentar a palavra na língua nativa e requeri-la na segunda língua. Enviar uma resposta em branco é permitido. A Figura 3 mostra o fluxo dos diferentes tipos de atividades da aplicação de CAI et al. (2015).

Figura 3 – Fluxo *Wait Chatter*

a) Modo de Estudo

b) Modo de Estudo com uma palavra contextualizada

c) Modo Quiz, tradução da língua estrangeira para língua nativa

d) Modo Quiz, tradução da língua nativa para língua estrangeira

e) Setas para avançar para próximo exercício

Fonte: CAI et al. (2015)

Para a seleção do vocabulário os autores estabeleceram um algoritmo baseado no conceito de que a repetição de um conteúdo auxilia sua fixação na memória. CAI et al. (2015) consideram um *flashcard* o par palavra na língua nativa e palavra na língua estrangeira. A aplicação mantém cinco *flashcards* rotacionando para o usuário. Para cada

*flashcard* é associado um contador de acertos, tomando as atividades em modo *Quiz* como referência para incrementá-lo ou decrementá-lo.

Dentre os cinco pares considerados, cada um é apresentado na  $n$ -ésima posição, sendo  $n$  o valor do contador de acertos. Dessa forma, aquelas palavras que possuem uma contagem de acertos menor, aparecem mais vezes para o usuário. Quando o contador chega a 4, o *flashcard* é removido e outro toma seu lugar.

Uma pesquisa foi conduzida por CAI et al. (2015) com 20 pessoas que engajaram no *Wait Chatter* pelo período de duas semanas. A intenção era medir o aprendizado e a experiência do usuário em relação aos momentos de espera.

As pessoas selecionadas eram em sua maioria estudantes de graduação, entre 17 e 20 anos, usuárias regulares do então *Google Chat* no seu navegador de internet. Além disso, elas demonstraram interesse ou estavam previamente estudando Espanhol ou Francês, idiomas disponíveis no *Wait Chatter* de (CAI et al., 2015).

Primeiramente, os participantes instalaram a aplicação em seus computadores pessoais, e receberam um tutorial sobre como usar a aplicação. Eles foram instruídos a usá-la como e nos momentos que preferissem.

Ao final das duas semanas, observou-se a troca de 47393 mensagens instantâneas pelos 20 participantes, que se comunicaram com um total de 249 amigos, sendo a média de 170 *chats* por dia.

Para verificar o aprendizado, os participantes realizaram um teste sobre o vocabulário ao qual foram expostos. Em média, eles foram capazes de lembrar 57 novas palavras, ou seja, que o usuário alegou não conhecer interagindo com a aplicação, o que equivale a aproximadamente 4 palavras por dia. (CAI et al., 2015)

## 2.4 Inteligência Artificial no Aprendizado

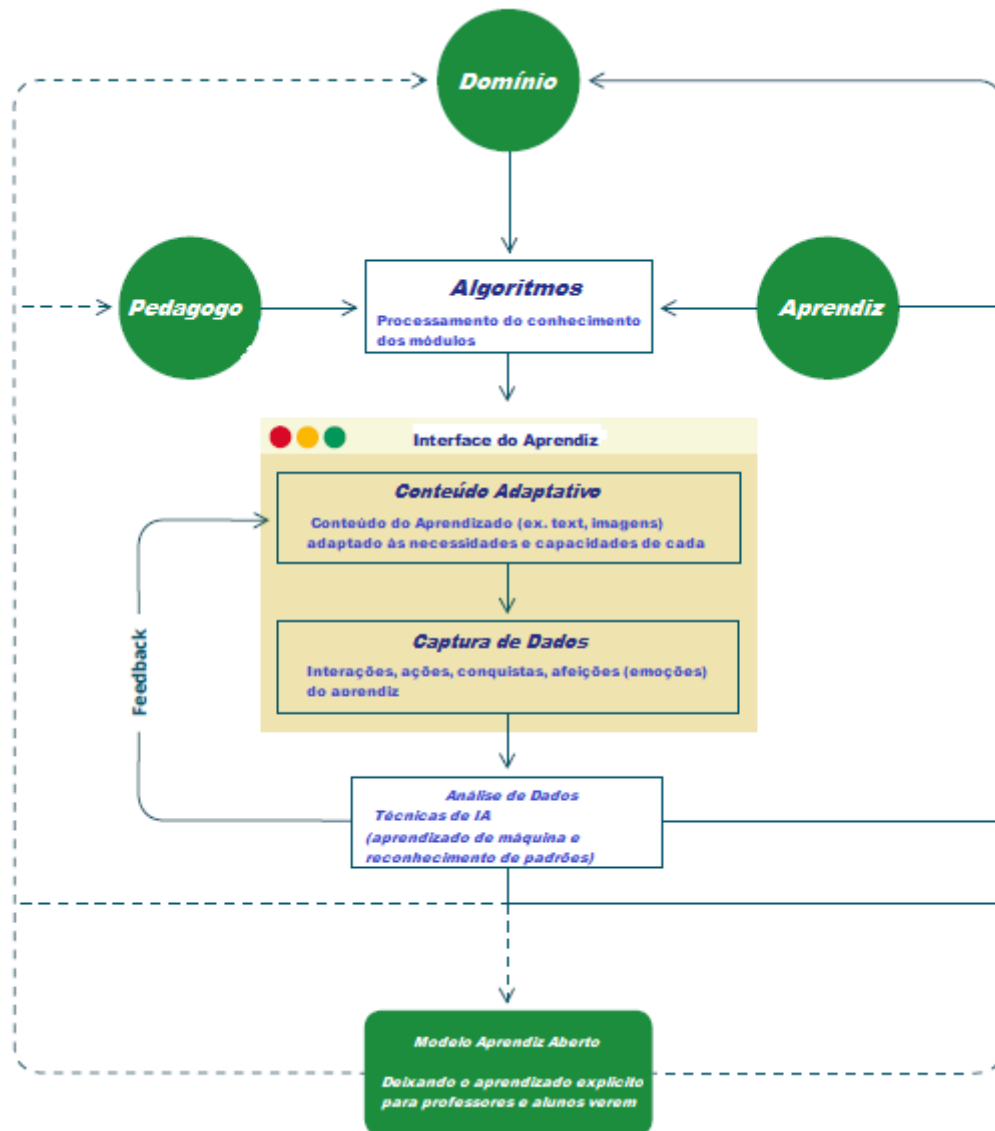
Como discutido na Seção 2.1, os mecanismos de aprendizado vem se diversificando e se modernizando conforme a tecnologia avança. A introdução de metodologias e conceitos de Inteligência Artificial, IA, no campo da educação é um exemplo de novas vertentes incorporadas a área.

IA envolve um *software* programado para interagir com o mundo de maneiras que normalmente exigem inteligência humana. Isso significa que a IA depende do conhecimento sobre o mundo e de algoritmos para processar de forma inteligente esse conhecimento. (LUCKIN et al., 2016) Dessa forma, uma atividade educacional fomentada por aspectos da IA pode ser chamada de um "modelo baseado em tutor adaptativo". A Figura 4 mostra os componentes desse modelo.

O *Learner Model* ou Modelo do Aprendiz, representa o conhecimento individual



Figura 4 – Modelo de IA para Educação



Fonte: baseado em LUCKIN et al. (2016)

de quem está aprendendo, já o *Pedagogy Model* ou Modelo do Pedagogo é o conhecimento do professor e o *Domain Model* refere-se ao conteúdo a ser aprendido.

Realizando a comunicação entre essas três partes estão os algoritmos, que irão processar o conteúdo e selecionar o mais apropriado para o aprendiz de acordo com suas capacidades e necessidades individuais.

Enquanto o conteúdo é ministrado para o aprendiz, através de uma interface, (que contém textos, imagens, sons, entre outras coisas), suas interações, como as respostas de exercícios por exemplo, são captadas e então analisadas. São essas análises que geram *feedbacks* tanto para o sistema, na sua geração de conteúdo, quanto para o *Pedagogy Model* e o *Learner Model*, ajudando-os a terem percepções do processo de aprendizagem

transparentes em modelos tradicionais de educação(LUCKIN et al., 2016).

Essas percepções representam uma das grandes vantagens da utilização de Inteligência Artificial no processo de aprendizagem, não somente para quem está aprendendo, mas também para o educador. Este, pode utilizar as análises do sistema para melhor entender como o aluno chegou em determinada resposta, ou ainda sobre processos cognitivos como lembrar e esquecer de um conteúdo e seus impactos nas performances dos alunos. Assim como se o estudante está cansado, entediado ou frustrado, permitindo aos professores identificar oportunidades de melhorias nas suas abordagens (LUCKIN et al., 2016).

Adaptar o *Domain Model* de acordo com o perfil de cada individuo também representa uma vantagem, já que permite filtrar, dentro de uma grande massa de dados, o conteúdo com maior potencial de ser efetivamente aprendido.

Em resumo, quando usada dentro do campo da educação, o objetivo da IA é fazer agir não somente na agilidade e intensidade do aprendizado, mas também trazer revelações sobre a forma de ensino e as dificuldades dos alunos. Ou seja, além de intensificar o aprendizado, ela tende a revelar aspectos do processo de aprendizagem em si (LUCKIN et al., 2016).

## 2.5 Sistemas de Recomendação

A quantidade de dados e informações que cercam o ser humano atualmente é enorme, porém o individuo não possui a capacidade de assimilar todos esses conteúdos, necessitando de alguma maneira reduzi-los. Uma das mais recorrentes ações para esse propósito é pedir recomendações a outras pessoas no dia a dia, a sites de busca ou redes sociais. Quando o sistema já realiza automaticamente esse filtro de informações, e traz exatamente conteúdo relacionado aos interesses da pessoa, a tendencia é melhorar a aceitação e o engajamento da mesma.

Segundo Reategui e Cazella (2005) para esse tipo de sistema, um dos grandes desafios é descobrir a relação entre o que está sendo recomendado e para quem está sendo recomendado. Pensando nisso, a primeira etapa para um sistema de recomendação é a identificação do usuário no momento em que ele acessa o sistema. Essa identificação pode ser realizada de duas maneiras:

- a) Identificação no Servidor: normalmente realizada através de um cadastro explícito ao usuário, que insere informações pessoais como nome, *e-mail*, telefone. Além disso, necessariamente tem a segurança de *login*/senha. Tais dados ficam armazenados em um banco de dados no servidor. Sempre que acessa o sistema, é necessário realizar a autenticação desse usuário. Essa abordagem permite o

sistema saber com maior precisão quem receberá as recomendações

- b) Identificação no Cliente: normalmente realizado por meio de *cookies*, mecanismo utilizado por *websites* para identificar que determinado computador se conectou a ele mais de uma vez. Essa versão assume que a máquina é utilizada sempre pela mesma pessoa. É um modelo de implementação mais simples, mas menos preciso.

Após a identificação do usuário, inicia-se a coleta de dados, que pode ser de maneira explícita, na qual o usuário indica de forma espontânea o que é importante, como a seleção de cantores prediletos em um aplicativo de música; ou implícita, isto é, a partir ações tomadas pelo usuário dentro do sistema é possível inferir informações sobre ele, como por exemplo, o gênero de música dos artistas escolhidos de forma explícita. (REATEGUI; CAZELLA, 2005)

Em seguida deve-se definir o modelo de filtragem dos dados e a estratégia da recomendação.

A filtragem dos dados refere-se a identificação e escolha de quais dados são relevantes para determinar a recomendação. Essa filtragem por ser feita de três maneiras: Filtragem por Conteúdo, Filtragem Colaborativa ou Filtragem Híbrida.

Reategui e Cazella (2005) define Filtragem por Conteúdo como sendo aquela que realiza uma seleção baseada na análise de conteúdo dos itens e perfil do usuário, isto é, os interesses do indivíduo são obtidos através de informações fornecidas ou ações realizadas por ele mesmo. Entretanto a Filtragem por Conteúdo apresenta a necessidade de se ter conhecimento sobre o conteúdo, o que se torna mais complexo com algumas mídias, como imagens e sons. Tomando como exemplo uma plataforma de filmes que implementa recomendações, quando ela sugere ao usuário um filme de romance porque ele assistiu um filme do mesmo gênero, ocorreu uma filtragem por conteúdo e saber a qual gênero o filme pertence é fundamental para essa abordagem.

Já a Filtragem Colaborativa não apresenta tal necessidade, ela se baseia na troca de experiências entre os usuários do sistema que apresentam interesses comuns. Esse tipo de filtragem toma como base as avaliações das pessoas sobre o conteúdo. Nessa filtragem, o mais comum é que cada usuário atribua pontuações para qualidade do conteúdo. Tais pontuações são coletadas recomendação se faz através desses pontos. Dessa forma, a pessoa tem conhecimento de itens considerados "bons" pelo grupo.

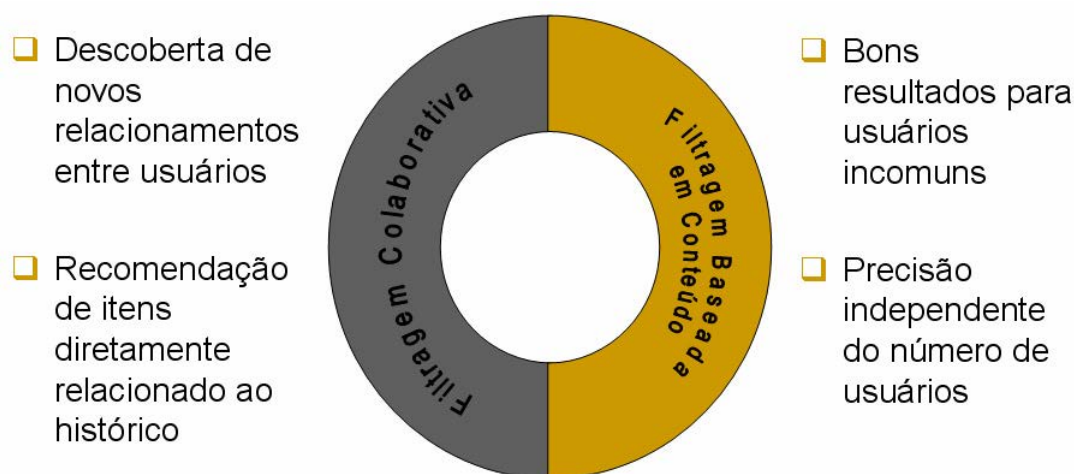
Porém, a filtragem colaborativa também apresenta alguns problemas. Eles são apresentados por Reategui e Cazella (2005) como:

- a) Problema do primeiro avaliador: quando se tem um novo conteúdo, ele ainda não foi avaliado e então, por essa abordagem não é possível recomendá-lo;

- b) Problema de pontuações esparsas: se o número de usuários for pequeno em relação ao volume de informações, existe o risco das pontuações se tornarem esparsas;
- c) Problema de similaridade: caso o usuário tenha gostos que diferenciem do senso comum do seu nicho, ele provavelmente não se interessará pelos conteúdos recomendados.

Desse modo, a Filtragem Híbrida é aquela que combina a filtragem por conteúdo e a filtragem colaborativa, na tentativa de unir as vantagens das duas maneiras, uma vez que os pontos fracos de uma são supridos pela outra, assim como mostra a Figura 5.

Figura 5 – Filtragem do Dados - Filtragem Híbrida



Fonte: Reategui e Cazella (2005)

A estratégia de recomendação refere-se a forma estrutural e visual que os dados serão recomendados. Reategui e Cazella (2005) mostram alguns exemplos como as Listas de Recomendação e a abordagem de que "Usuários que se interessaram por X também se interessaram por Y".

As Listas de Recomendação constituem em manter listas de conteúdos organizadas por tipos de interesses, como "Itens mais vendidos", "Lançamentos", entre outras coisas. A abordagem "Usuários que se interessaram por X também se interessaram por Y" tem como exemplo prático o "websites" de comércio eletrônico que mostram "Pessoas que compraram o item X também compraram item Y". Representa uma maneira de associar conteúdo através de ações dos usuários.

Por fim, existe a lógica a ser utilizada para recomendação, ou seja, as funções algorítmicas e estruturas de dados que serão construídas para o sistema. Reategui e Cazella (2005) explicam a regra de associação, classificação e agrupamento.

Uma regra de associação é definida como um relacionamento  $X \Rightarrow Y$ , tal que X e Y

são conjuntos de itens, atributos dos dados, e a intersecção entre os mesmos constitui-se em um conjunto vazio. Cada regra de associação apresenta um fator de suporte ( $FSup$ ), que é quantidade de tuplas que satisfazem  $X$  e  $Y$  dividida pelo total de tuplas, como mostra a Equação 2.1, e um fator de confiança ( $FConf$ ) que representa a razão entre o número de tuplas que satisfazem  $X$  e  $Y$  e apenas  $X$ , como está na Equação 2.2.

$$FSup = \frac{|X \cup Y|}{N} \quad (2.1)$$

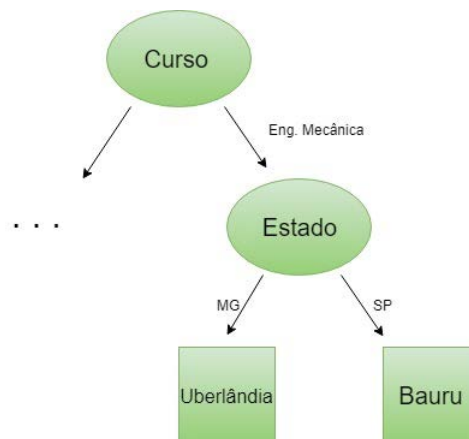
$$FConf = \frac{|X \cup Y|}{|X|} \quad (2.2)$$

A partir de então, a extração de regras busca encontrar os conjuntos que respeitem um suporte e confiança mínimos para o problema em questão. Tais regras podem ser usadas como a recomendação em si, de forma que se o usuário se interessa pelo conjunto  $X$ , isso implica, com confiança  $FConf$  e suporte  $FSup$ , que ele também se interessaria pelo conjunto  $Y$ , que deve então ser recomendado. O desafio dessa lógica é encontrar os valores mínimos de suporte e confiança que melhor se adéquam a situação.

A classificação parte da premissa que cada item pertence a uma classe dentro de um conjunto predefinido de classes. Seu princípio está em descobrir algum relacionamento entre os atributos para revelar a classificação de algum atributo desconhecido.

Suponha um sistema que guarde informações sobre alunos do ensino médio, como seu curso de interesse e o estado no qual reside, e deseja-se prever ou recomendar a cidade da sua possível universidade a partir desses dados. Uma classificação do tipo se o curso de interesse for "Engenharia Mecânica" e o estado for "São Paulo", a recomendação é "Bauru". Esse tipo de metodologia é comumente implementado como uma árvore de decisão. Como está explícito na Figura 6.

Figura 6 – Árvore de Decisão para Recomendação por Classificação

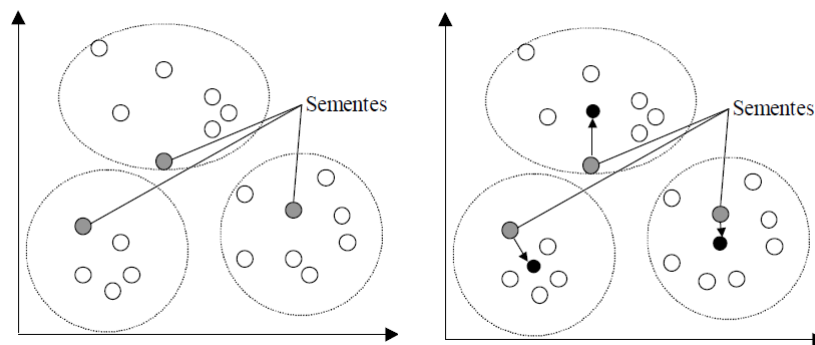


Fonte: elaborado pela autora.

O agrupamento refere-se a criação de grupos ou partições nos dados a partir de uma similaridade entre as tuplas. Um exemplo é o agrupamento pela avaliação de qualidade, que separa conteúdos cuja classificação é mais elevada dos itens pior avaliados. Reategui e Cazella (2005) defendem que o método é um tipo de aprendizado não supervisionado, pois é dever do algoritmo definir quais atributos e seus valores serão usados para montar os grupos.

Um algoritmo amplamente utilizado é o *k-means*. Primeiramente, deve-se indicar quanto grupos se deseja, esse é o valor  $k$ . Então são aleatoriamente selecionadas  $k$  tuplas, que representam os  $k$  grupos. Em seguida, cada uma das demais tuplas é associada a um dos grupos, através de uma distância de similaridade que é calculada de forma particular ao problema. Para cada grupo é estabelecido a média dessa distância como sendo a nova "semente", então o processo é repetido até que não haja alterações de uma iteração para outra. A Figura 7 ilustra o processo.

Figura 7 – Método *k-means*



Fonte: Reategui e Cazella (2005).

## 3 Materiais e Métodos

Este capítulo irá apresentar as ferramentas e tecnologias utilizadas para o desenvolvimento do *Mobile Wait Chatter*, assim como a metodologia seguida para a sua implementação.

### 3.1 Android Studio

Para o aplicativo em si, o ambiente de desenvolvimento utilizado foi o *Android Studio*, que é a IDE oficial para construção de código nativo para a plataforma de dispositivos móveis *Android*. A linguagem de programação utilizada foi *Kotlin* e os dispositivos alvo são os *smartphones*.

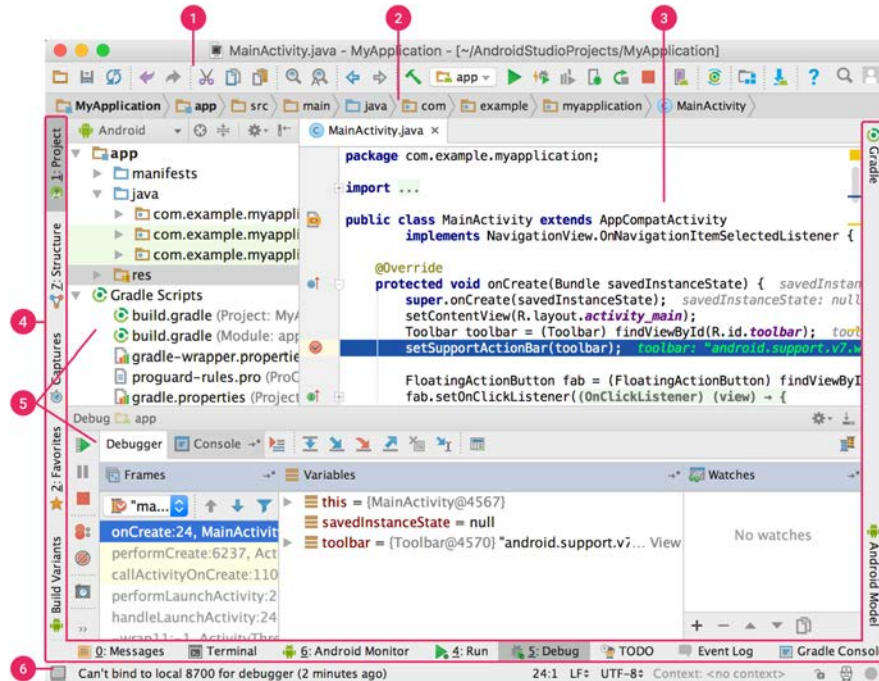
Além do editor de código, o *Android Studio* oferece recursos para aumentar a produtividade no desenvolvimento de aplicativos móveis, como:

- a) Sistema de compilação flexível
- b) Emulador com diferentes recursos de dispositivo
- c) Ambiente unificado para desenvolvimento para todos os dispositivos *Android*
- d) *Instant Run*: aplicar alterações aos aplicativos em execução
- e) Modelos de códigos e integração com sistemas de versionamento
- f) Ferramentas e estruturas de testes

Cada módulo do aplicativo contém os diferentes arquivos: manifestos, *java* e recursos. Os manifestos são arquivos que contém informações sobre a forma como aplicação será executada, como plataforma alvo e classe principal; enquanto isso, os arquivos *java* correspondem aos códigos fonte do projeto; e por fim, os recursos são as propriedades do aplicativo que não são código, como *layouts XML*, imagens e cores.

A Figura 8 contém a janela principal do ambiente de desenvolvimento em questão. É possível personalizá-la de acordo com os interesses do usuário. Essa janela principal, é dividida em:

1. Barra de ferramentas. Por ela, é possível realizar diversas ações, incluindo executar aplicativos.
2. Barra de navegação. Meio de navegação pelos arquivos do projeto.
3. Janela do editor. Na qual o desenvolvedor cria e modifica código.

Figura 8 – Janela Principal do *Android Studio*

Fonte: Android Developers (2018)

4. Barra de janela ferramentas. Contém os botões que permitem expandir ou recolher a janela de cada ferramenta.
5. Janela de ferramentas. Acesso a tarefas como gerenciamento de projetos, controle de versão, entre outros.
6. Barra de status. Mostra o status do projeto e da IDE e mensagens

## 3.2 Linguagem de Programação *Kotlin*

A linguagem de programação *Kotlin* é uma linguagem de desenvolvimento oficial para desenvolvimento *Android*. De acordo com Zanella e Freitas (2017), a *Google* realizou o anúncio do suporte de *Kotlin* no evento *Google I/O 2017*.

Segundo Zanella e Freitas (2017), o *Kotlin* apresenta uma curva de aprendizado menor e compartilha conceitos com o *Java*, que também é uma linguagem de programação oficial para *Android* nativo e a mais utilizada.

Em comparação com a linguagem *Java*, Zanella e Freitas (2017) defendem que as principais vantagens do *Kotlin* é ser conciso, uma vez que, elimina a necessidade de conversões de tipos, que são inferidos pelo contexto; seguro, pois contém dois tipos de dados diferentes: objetos que podem ser nulos e objetos que não podem, assim não há necessidade de lidar o exceções de objetos nulos; e as ferramentas e ecossistema, já que a



linguagem é mantida pela *JetBrains* que é uma consolidada empresa de desenvolvimento de *software*.

### 3.3 Google Firebase

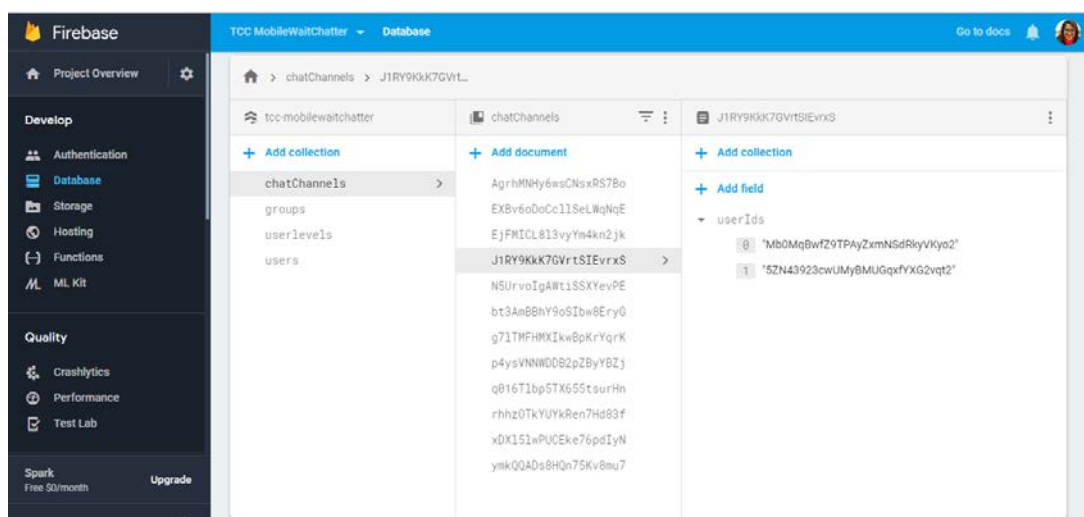
O *Google Firebase* é um serviço em nuvem para desenvolvimento de aplicativos móveis e aplicações web. Ele oferece diversos recursos como armazenamento em nuvem, autenticação, banco de dados em tempo real, *analytics*, laboratório de testes, notificações, entre outras coisas.

Para esse projeto estão sendo utilizados os serviços de autenticação, banco de dados e armazenamento em nuvem.

Para autenticação foi habilitado o modelo de *e-mail*/senha para inscrever o usuário no sistema. A seção 4.0.1 contém mais detalhes sobre o mecanismo.

O banco de dados oferecido pelo *Firebase* é não relacional e é organizado em coleções. Cada coleção possui documentos, identificados por um código único, que por sua vez representa um objeto com diferentes campos. Cada documento ainda pode conter coleções, com novos documentos, e assim sucessivamente. A Figura 9 tem a interface do banco de dados em questão, é possível realizar alterações no banco através dela, ou através da aplicação que o acessa.

Figura 9 – Interface do Banco de Dados do *Firebase*



Fonte: elaborado pela autora.

Para guardar as imagens inseridas pelo usuário no aplicativo como a sua foto de perfil e as imagens trocadas durante uma conversa foi utilizado armazenamento em nuvem disponibilizado pelo *Firebase*. Dessa maneira, no banco de dados é salvo apenas o endereço

da imagem, mas o arquivo em si fica armazenado de forma separada na seção de *Storage* não sobrecarregando o banco de dados em tempo real.

### 3.4 Metodologia

Para incorporar uma atividade de *Wait Learning* ao aplicativo desenvolvido, foi necessário entender realizar um estudo sobre os requisitos e funcionamentos das aplicações semelhantes, principalmente o *Wait Chatter* cuja dinâmica serviu de inspiração para o *Mobile Wait Chatter*

Em seguida, foi desenvolvido um aplicativo de troca de mensagens instantâneas, que apresenta as funcionalidades comumente encontradas nesse tipo de aplicação. Sendo elas, a criação de um perfil de usuário e a troca de mensagens em formato de texto e imagens entre os usuários cadastrados.

A próxima etapa consistiu em reunir um conjunto de palavras para compor o banco de dados da aplicação. O vocabulário ainda foi categorizado de acordo com seu significado, e dentro dos grupos semânticos dividido em níveis de dificuldade. Tais palavras foram salvas no banco de dados previamente ao acesso do primeiro usuário a aplicação de aprendizado em espera do *Mobile Wait Chatter*.

O terceiro passo foi construir o *layout* da aplicação de *Wait Learning* adicioná-lo ao aplicativo de *chat* previamente desenvolvido.

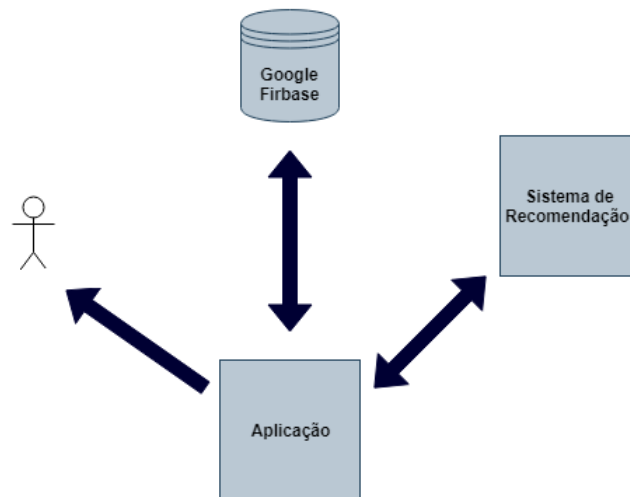
Após o *layout* pronto, foi definido e implementado fluxo das telas da aplicação de *Wait Learning* assim como a integração do *frontend* com as coleções de palavras do banco de dados.

Por fim, após estudar sobre o funcionamento e requisitos de um Sistemas de Recomendações, descritos na seção 2.5, ele foi implementado para o aplicativo seguindo uma filtragem por conteúdo e uma lógica de classificação.

## 4 Desenvolvimento

O *Mobile Wait Chatter* consiste em um aplicativo de troca de mensagens instantâneas, conhecido como *chat* que contém integrado ao sistema uma aplicação de *Wait Learning* para o ensino do idioma Inglês, partindo do contexto que o usuário tem como língua materna o Português. Ele ainda agrega a aplicação de aprendizado em espera um algoritmo de recomendação por conteúdo que tenta aprimorar a experiência do usuário. A arquitetura do projeto segue o descrito na Figura 10.

Figura 10 – Arquitetura do Projeto



Fonte: elaborado pela autora.

As próximas seções irão apresentar os métodos e telas desenvolvidos para o aplicativo *Mobile Wait Chatter*

### 4.0.1 Autenticação

Como visto na seção 2.5 a identificação do usuário em um sistema de recomendação é fundamental. A abordagem utilizada para reconhecer o usuário na aplicação consiste em um cadastro de forma explícita.

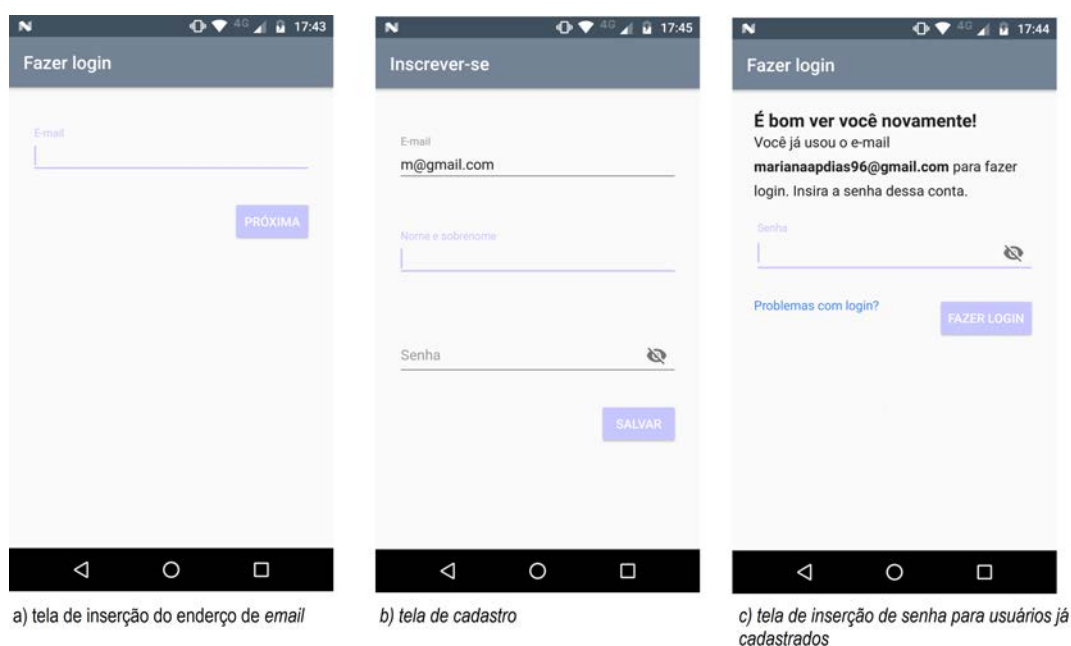
Essa autenticação do usuário foi construída com a ferramenta *Google Firebase* utilizando o formato de "e-mail e senha". Essa ferramenta, disponibiliza um sistema de autenticação transparente que implementa práticas recomendadas para autenticação em dispositivos móveis e sites.

Além de facilitar e gerenciar o *backend* do processo de identificação, o *Firebase* possui o *FirebaseUI* que lida com os fluxos de interface de usuário para fazer login dos usuá-

rios, oferecendo recursos como "Esqueci Minha Senha" por exemplo. Portanto, os *layouts* das telas de autenticação não precisaram ser codificados, sendo incorporados das bibliotecas do *Firebase*.

Ao abrir o aplicativo pela primeira vez, o usuário encontra uma tela com um botão para "Entrar" na aplicação. Em seguida é direcionado para uma tela onde insere seu *e-mail*, que pode ser vista na Figura 11a. Nesse momento o *FirebaseUI* verifica se endereço de *e-mail* já é cadastrado. Em caso negativo, o usuário é levado para tela de cadastro, disponível na Figura 11b, na qual insere seu "Nome" e uma "Senha".

Figura 11 – Telas de Autenticação



Fonte: elaborado pela autora.

No caso do usuário já ter entrado no sistema previamente, ele é direcionado para a tela da Figura 11c para inserir sua senha, que por sua vez é autenticada, permitindo ou não o acesso ao sistema. Dessa forma, a identificação do usuário é formalizada com seu endereço de *e-mail*. No caso de utilizar diferentes dispositivos, suas informações no sistema são recuperadas, assim como, num segundo acesso ao aplicativo pelo mesmo dispositivo, efetuar o processo de autenticação não é necessário.

#### 4.0.2 Tela de Contatos e Perfil

Após efetivar o *login* o usuário entra na tela de contatos, na qual visualiza uma lista de seus contatos. Cada usuário tem como contato todos os demais usuários do aplicativo.

A lista de contatos, mostrada na Figura 12, apresenta as informações de "Nome", "Foto" e "Mensagem de Boas Vindas", fornecidas por cada indivíduo, dentro de um cartão.

Ao clicar sobre esse cartão ocorre um redirecionamento para uma tela de conversa, que será explorada na subseção 4.0.3, na qual pode iniciar a troca de mensagens instantâneas. Para retornar à tela de contato, o usuário deve clicar na seta de voltar da aplicação ou dispositivo.

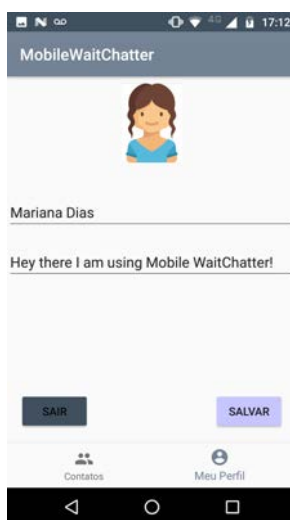
Figura 12 – Tela de Contatos



Fonte: elaborado pela autora.

Na Figura 12 também é possível notar, no inferior da tela, a possibilidade de navegação para a tela de perfil, ao clicar na aba "Meu Perfil". É nessa tela, ilustrada na Figura 13, que o usuário insere as informações mostradas no cartão da página de contatos. Ele consegue interagir com cada campo, alterar seus dados e então clicar no botão "Salvar" para enviar as informações para o banco de dados.

Figura 13 – Tela de Perfil



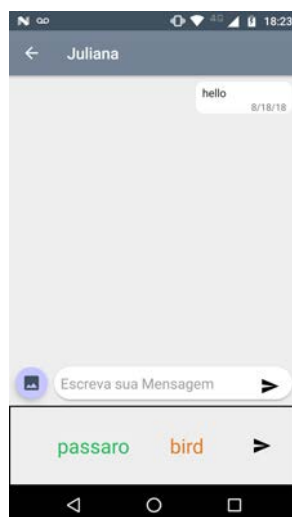
Fonte: elaborado pela autora.

### 4.0.3 Tela de Conversa e Aplicação de *Wait Learning*

A tela de conversa do aplicativo, é aquela que na qual o usuário realiza a troca de mensagens. Ela é composta por um espaço onde são mantidas as mensagens trocadas pelos usuários, um campo de texto para escrita do novo texto, um botão para acessar a galeria para envio de imagens e um botão para "enviar" a mensagem, Figura 12.

Também é nessa tela que se encontra o painel destinado a aplicação de aprendizado em espera. Ele é inspirado no *Wait Chatter* de CAI et al. (2015) e mantido fixo utilizando uma abordagem de *Soft Notification*. Na Figura 14 pode-se identificar os elementos descritos.

Figura 14 – Tela de Conversa



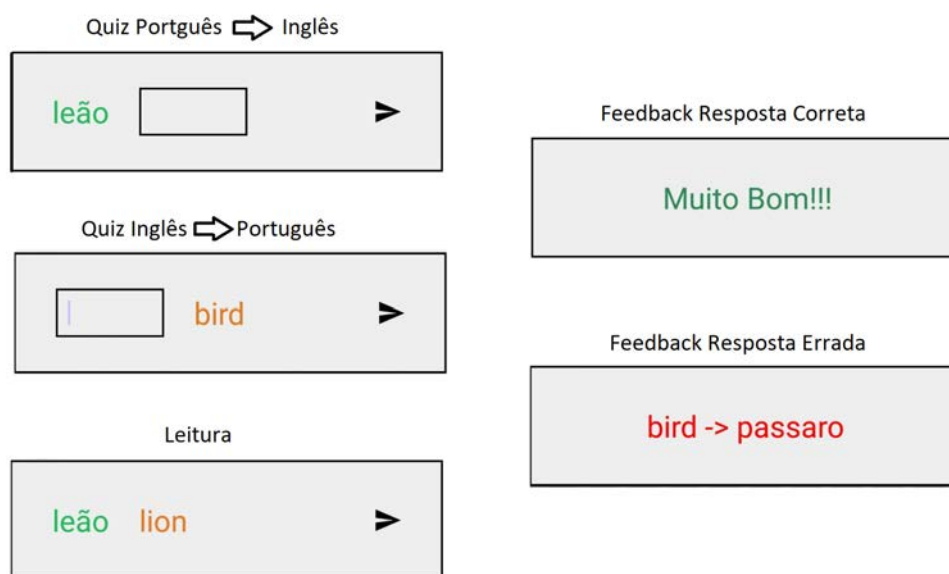
Fonte: elaborado pela autora.

Assim como na aplicação de CAI et al. (2015) as atividades propostas para ensino do idioma são divididas em dois tipos: "Leitura" e "Quiz". Em "Leitura" apresenta o vocabulário escrito em português e inglês para que o usuário tenha o primeiro contato com a palavra.

Já o modo *Quiz* é dividido em duas situações: o sistema apresenta a palavra em português e requisita sua escrita em inglês ou ao contrário. A resposta é avaliada como correta ou incorreta. É possível enviar respostas em branco, porém essa ação é considerada como um erro.

Para finalizar uma atividade o usuário deve clicar na seta a direita. Uma mensagem indicando se a resposta do usuário no modo *Quiz* está certa ou não, aparece no painel. A Figura 15 contém exemplos das três atividades e das mensagens de *feedback* para usuário.

A intenção de personalizar a experiência do usuário na aplicação de *Wait Learning* parte do princípio de tentar engajar ao sistema, além das pessoas que não possuem conhe-

Figura 15 – *Layout* Aplicação *Wait Learning*

Fonte: elaborado pela autora.

cimento na língua estrangeira, mas também aqueles que já possuem alguma habilidade na área.

Assim, a intenção é avançar o nível de dificuldade do vocabulário conforme o usuário demonstra já ter conhecimento. E, ao mesmo tempo, para os iniciantes garantir a construção de conhecimento de maneira gradativa e sólida.

Para atingir esse objetivo foi implementado um sistema de recomendação, com filtragem baseada em conteúdo, uma vez que a experiência do usuário em adquirir conhecimento em Inglês independe da experiências dos demais usuário.

Como foi explicado na seção 2.5, quando se tem um sistema baseado em conteúdo é necessário ter conhecimento sobre o conteúdo. Tal conteúdo é representado pelos conjuntos de palavras português/inglês.

Dessa forma, foram aplicadas classificações para o vocabulário em grupos e níveis. Cada palavra pertence a um "grupo" que se correlaciona com a sua semântica, por exemplo: animais, profissões, lugares, família, entre outros. Dentro de cada "grupo" as palavras é atribuído um valor numérico crescente, associado com o "nível" de dificuldade da mesma, sendo o valor 1 para o nível mais baixo e 3 para o nível mais alto.

Portanto, para cada grupo de vocabulário o usuário possui um nível no qual se encontra na aplicação. Sendo que quando ele interage pela primeira vez, a todos seus grupos é atribuído nível 1.

Para realizar a classificação do vocabulário da aplicação foi utilizado o site *Speak*

*Languages* que já apresenta vocabulário de inglês dividido em categorias através de seu significado, assim como o proposto. Para o nivelamento das palavras entre os grupos, a experiência própria da autora serviu como base.

Para a geração de vocabulário para as atividades é selecionado um "grupo" de maneira aleatória, então desse grupo são selecionadas cinco palavras do nível do usuário, também de maneira aleatória. Então primeiramente são exibidas as cinco palavras em modo de "Leitura", em seguida os dois tipos de atividade de "Quiz" são intercalados para abranger as cinco palavras, iniciando pela situação Português para Inglês.

A esse conjunto de cinco palavras é associado um contador de acertos, que se inicia com valor zero e incrementa a cada resposta correta do usuário nas atividades de *Quiz*.

Ao terminar esse conjunto de palavras, o sistema de recomendação é acionado e, após seu processamento, o algoritmo se repete.

O sistema de recomendação tem como entrada o grupo de palavras atual, o nível do usuário nesse grupo e o contador de acertos. Então, seguindo a Tabela 1, ele incrementa ou decrementa o nível do usuário no grupo.

Dessa maneira, usuários com desempenho ótimo ou muito bom, avançam de maneira mais rápida no processo de aprendizagem. Porém se começar a cometer erros, o sistema regride seu nível no grupo na tentativa de evitar perda no processo de aprendizagem.

Tabela 1 – Regras para Nível do Usuário

Acertos	Nível do Usuário no Grupo
5	Sobe de nível no grupo
4	Sobe de nível no grupo
3	Mantém o nível no grupo
2	Regride de nível no grupo
1	Regride de nível no grupo
0	Regride de nível no grupo

Fonte: elaborado pela autora.

## 4.1 Banco de Dados

A estrutura do banco de dados do *Mobile Wait Chatter* segue a Figura 16. As coleções de *users* e *chatChannels* são dado utilizados pelo aplicativo de troca de mensagens em si, sendo que coleção *users* guarda as informações referentes ao perfil do usuário e os *chats* no quais estão inseridos; a coleção *chatChannels* mantém os dados sobre as mensagens trocadas pelos usuários. Já as demais, servem de fontes para a aplicação de aprendizado em espera, tal que, a coleção *groups* contém as palavras separadas pelas



diferentes categorias e a coleção *userLevels* salva os níveis de cada usuário em cada grupo de palavras.

Figura 16 – Coleções do Banco de Dados



Fonte: elaborado pela autora.

## 4.2 Resultados e Discussões

O trabalho pode ser comparado com a aplicação desenvolvida por CAI et al. (2015), descrita na seção 2.3, uma vez que ambos exploram as mesmas oportunidades de espera e o *Wait Chatter* serviu de base e inspiração para o projeto.

As principais diferenças do *Mobile Wait Chatter* e do *Wait Chatter* de CAI et al. (2015) estão no meio de distribuição e na individualização da experiência do usuário.

O *Mobile Wait Chatter* foi construído como um aplicativo para *smartphones Android*, que representam o principal meio no qual as pessoas utilizam aplicações de *chat*. Enquanto o *Wait Chatter* é uma extensão do navegador *Google Chrome* para ser utilizado dentro de um mecanismo de troca de mensagem depreciado, através do navegador de internet. Essa alteração promove uma maior disponibilidade de acesso ao sistema, engajando um público maior.

Através da implementação de um Sistema de Recomendação por conteúdo, este projeto busca melhorar experiência do usuário com o aprendizado do idioma, pois incorpora a recomendação sobre o conteúdo das atividades de *Wait Learning*, fazendo com que pessoas que se interessam por estudar inglês, mas possuem diferentes experiências com o idioma, possam usufruir de tarefas compatíveis a seus interesses.

A categorização das palavras em grupos semânticos e níveis de dificuldade dentro desse grupo também é novidade do aplicativo desenvolvido. Essa organização está diretamente associada com a utilização do sistema de recomendação.

O algoritmo do fluxo de vocabulário usado neste trabalho apresenta semelhanças e diferenças em relação ao utilizado no projeto de CAI et al. (2015). Assim como no *Wait Chatter*, este trabalho considera cinco palavras diferentes por vez, mas assim que percorre todas as palavras, ele é inteiramente trocado, podendo as palavras reaparecerem no caso do usuário não ter subido de nível no grupo. Além disso, o contador de acertos está relacionado com o conjunto de cinco palavras. Já a aplicação de CAI et al. (2015) apenas troca as palavras consideradas aprendidas dentro do conjunto, o substituindo então palavra a palavra, uma vez que o contador de acertos está associado a cada palavra individualmente.

Além dos pontos discutidos, os quais representam os aspectos mais significativos das diferenças do aplicativo desenvolvido e a aplicação de CAI et al. (2015), existem outros pontos que podem ser comparados. A Tabela 2 apresenta um resumo comparativo de ambos projetos.

Tabela 2 – Comparação entre o projeto desenvolvido e o *Wait Chatter*

<b>Característica</b>	<b>Mobile WaitChatter</b>	<b>Wait Chatter</b>
Distribuição	Aplicativo Android	Extensão para Google Chrome
Categorização do Vocabulário	Apresenta	Não Apresenta
Perfil do usuário	Apresenta	Não Apresenta
Sistema de Recomendação	Apresenta	Não Apresenta
Contextualização do Vocabulário	Não Apresenta	Apresenta
Idioma Nativo Considerado	Português	Inglês
Idioma Estrangeiro Considerado	Inglês	Espanhol e Francês
Apresentação de Atividade	<i>Soft Notification</i>	<i>Soft Notification</i>
Modos de Atividade	Leitura e <i>Quiz</i>	Leitura e <i>Quiz</i>
Pesos para as diferentes atividades de <i>Quiz</i>	Não apresenta	Não Apresenta

Fonte: elaborado pela autora.

Na Tabela 2 também é possível identificar semelhanças entre o (CAI et al., 2015) e o *Mobile WaitChatter* como a apresentação da atividade de aprendizado em espera seguir o padrão de *Soft Notification* em ambas.

O mesmo ocorre para os modos de atividade, que são divididos em "Leitura" e "*Quiz*" e não diferenciação na contagem de acertos para as duas propostas de atividades de "*Quiz*", que usuário responde na sua língua materna ou no idioma estrangeiro proposto.

Os idiomas nativos e estrangeiros considerados, por uma questão de contextualização e regionalidade, mudam entre as duas aplicações. O *Wait Chatter* tem o Inglês como idioma nativo, e como objetivo de aprendizado apresenta o Espanhol e o Francês. Já o *Mobile WaitChatter* entende como idioma nativo ao usuário o Português e como

estrangeiro o Inglês.

O último ponto que é interessante ser comparado é a contextualização do vocabulário, que ocorre apenas no *Wait Chatter*. O aplicativo *Mobile WaitChatter* não contém essa funcionalidade, uma vez que ele considera o vocabulário categorizado para o funcionamento do sistema de recomendação por classificação.

## 5 Conclusão

O mundo moderno inspira as pessoas para cada vez mais se capacitarem intelectual e profissionalmente, porém as varias atividades do cotidiano limitam as disponibilidades de tempo para realizar tais aprimoramentos pessoais. O aplicativo desenvolvido nesse projeto traz uma possível solução para essa incoerência.

Dessa forma, o projeto consiste num aplicativo para troca de mensagens instantâneas, que integra uma aplicação de *Wait Learning* para aprendizado de Inglês. Ele explora o *Wait Learning*, que, como mostra a seção 2.2, é o conceito de que momentos de espera incorporados em atividades diárias podem ser utilizados para aprendizado.

Por meio da distribuição em aplicativos móveis e utilização de um sistema de recomendação, o aplicativo desenvolvido é capaz de disseminar o conceito de *Wait Learning*, combinando diferentes tecnologias para aprimorar a proposta original de CAI, REN e MILLER (2017) e CAI et al. (2015).

Além disso, o aplicativo, assim como outras aplicações de *Wait Learning*, proporciona uma alternativa para modelos educacionais tradicionais, que estão perdendo espaço atualmente, ao incluir o aprendizado na rotina das pessoas.

### 5.1 Trabalhos Futuros

Como trabalho futuro pode ser sugerida, a incorporação de novos idiomas. Para essa implementação, se a mesma classificação de vocabulário usada, não seriam necessárias alterações na aplicação, apenas a adição dos dados e alterações nas referencias de acesso a ele.

Também é possível considerar uma diferenciação para os acertos nos diferentes tipos de atividade de *Quiz*, atribuindo pesos por exemplo, uma vez que a ação de escrever uma palavra numa segunda língua é relativamente mais complexo, já que além de saber seu significado, é necessário se recordar da sua grafia.

# Referências

- ANDROID DEVELOPERS. Conheça o android studio. 2018. Disponível em: <<http://developer.android.com/studio/intro/?hl=pt-br>>. Acesso em: 29 out. 2018.
- CAI, C. J.; GUO, P. J.; GLASS, J. R.; MILLER, R. C. Wait-learning: Leveraging wait time for second language education. *ACM Trans. Comput.-Hum. Interact.*, v. 24, 2015. Disponível em: <<http://dspace.mit.edu/handle/1721.1/90407>>. Acesso em: 29 out. 2018.
- CAI, C. J.; REN, A.; MILLER, R. C. Waitsuite: Productive use of diverse waiting moments. *ACM Trans. Comput.Hum. Interact.*, v. 24, 2017. Disponível em: <[http://people.csail.mit.edu/ccai/publications/Cai\\_WaitSuite](http://people.csail.mit.edu/ccai/publications/Cai_WaitSuite)>. Acesso em: 29 out. 2018.
- ESTADAO CONTEUDO. Google talk será substituído pelo hangouts em 26 de junho. *Exame Abril*, 2017. Disponível em: <<http://exame.abril.com.br/tecnologia/google-talk-sera-substituido-pelo-hangouts-em-26-de-junho/>>. Acesso em: 29 out. 2018.
- GREGO, M. Watson, o fascinante computador da ibm que venceu os humanos. *Exame Abril*, 2012. Disponível em: <<http://exame.abril.com.br/tecnologia/watson-o-fascinante-computador-da-ibm-que-venceu-os-humanos/>>. Acesso em: 29 out. 2018.
- HUG, T. Microlearning: A new pedagogical challenge. *Nova Spivack*, 2006. Disponível em: <[http://www.researchgate.net/publication/272178142\\_Microlearning\\_A\\_New\\_Pedagogical\\_Challenge](http://www.researchgate.net/publication/272178142_Microlearning_A_New_Pedagogical_Challenge)>. Acesso em: 29 out. 2018.
- KAHNEMAN, D. *Attention and effort*. Citeseer, 1973. Disponível em: <[https://scholar.princeton.edu/sites/default/files/kahneman/files/attention\\_hi\\_quality.pdf](https://scholar.princeton.edu/sites/default/files/kahneman/files/attention_hi_quality.pdf)>. Acesso em: 29 out. 2018.
- LUCKIN, R.; HOLMES, W.; GRIFFITHS, M.; FORCIER, L. B. *Intelligence Unleashed: An argument for ai in education*. Pearson, 2016. Disponível em: <<http://www.gettingsmart.com/wp-content/uploads/2016/02/Pearson-Intelligence-Unleashed-v11-RW-WEB.pdf>>. Acesso em: 29 out. 2018.
- MALAMED, C. Is microlearning the solution you need?: A closer look at bite-sized learning. *The eLearning Coach*, 2015. Disponível em: <<http://thelearningcoach.com/elearning2-0/what-is-microlearning/>>. Acesso em: 29 out. 2018.
- MONTREZOR, B.; SILVA, A. B. A dificuldade no aprendizado da língua inglesa. *UniFOA*, 2009. Disponível em: <<http://web.unifoa.edu.br/cadernos/edicao/10/27.pdf>>. Acesso em: 29 out. 2018.
- PEREIRA, G. V. A inteligência artificial na educação. *Universidade do Estado de Santa Catarina – UDESC.*, 2012. Disponível em: <[http://www.ceavi.udesc.br/arquivos/id\\_submenu/387/guilherme\\_vota\\_pereira.pdf](http://www.ceavi.udesc.br/arquivos/id_submenu/387/guilherme_vota_pereira.pdf)>. Acesso em: 29 out. 2018.

REATEGUI, E. B.; CAZELLA, S. C. Sistemas de recomendação. *XXV Congresso da Sociedade Brasileira de Computação*, 2005. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.92.2811rep=rep1type=pdf>>. Acesso em: 29 out. 2018.

ZANELLA, J. R.; FREITAS, G. Kotlin: explorando uma nova linguagem. *Medium*, 2017. Disponível em: <<http://medium.com/android-dev-br/kotlin-explorando-uma-nova-linguagem-%EF%B8%8F-799ffc99a1a2>>. Acesso em: 29 out. 2018.