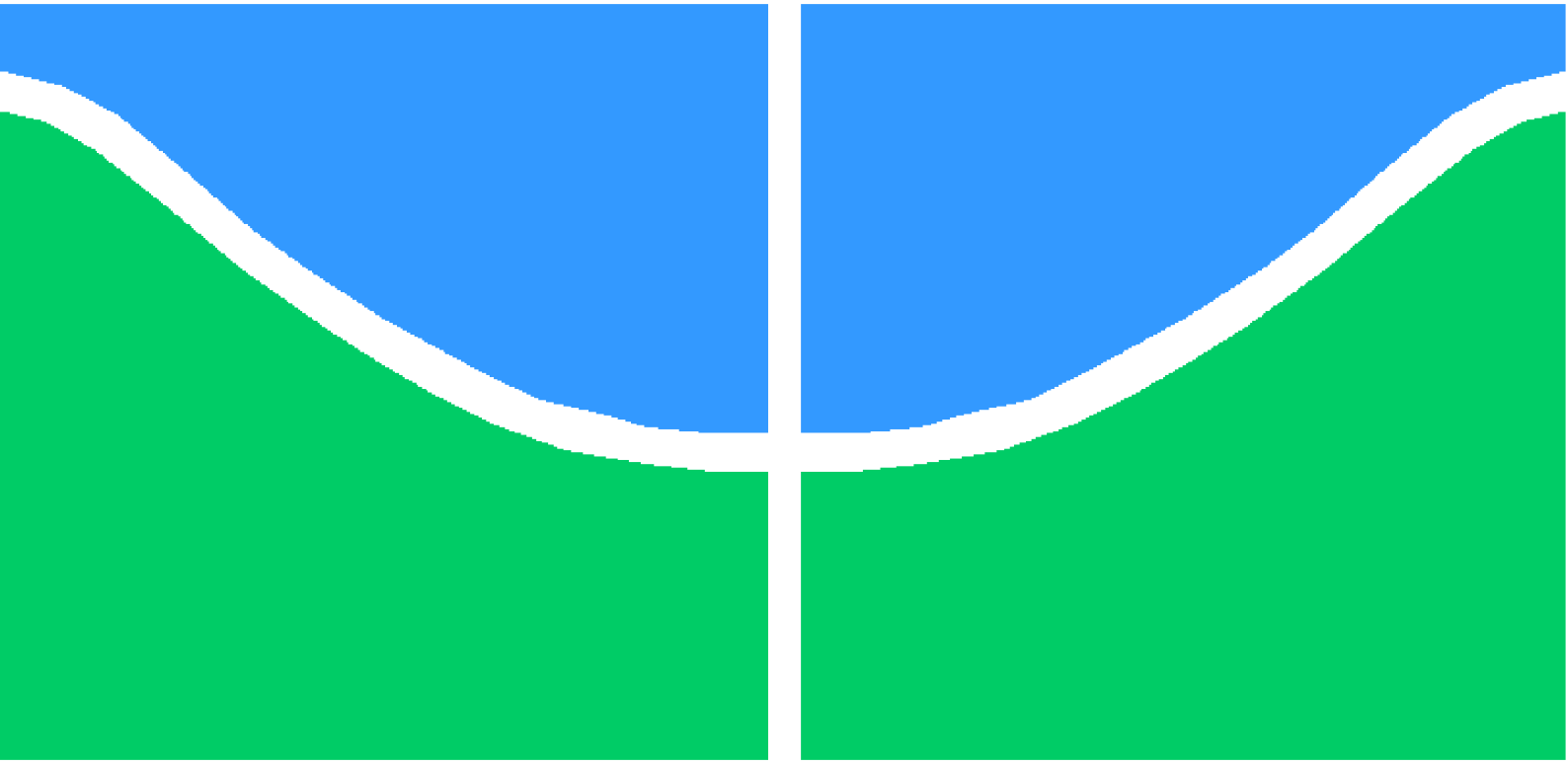

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Engenharia de Software

Aplicativo para o sistema operacional iOS integrado a um Sistema Tutor Inteligente para apoio à Educação

Autor: Luis Gustavo Avelino de Lima Jacinto

Orientador: Prof. Dr. Vandor Roberto Vilardi Rissoli

Brasília, DF

2022

Luis Gustavo Avelino de Lima Jacinto

Aplicativo para o sistema operacional iOS integrado a um Sistema Tutor Inteligente para apoio à Educação

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do título de Bacharel em Engenharia de Software.

Universidade de Brasília - UnB

Faculdade UnB Gama - FGA

Orientador: Prof. Dr. Vandor Roberto Vilardi Rissoli

Brasília, DF

2022

Luis Gustavo Avelino de Lima Jacinto

Aplicativo para o sistema operacional iOS integrado a um Sistema Tutor Inteligente para apoio à Educação/ Luis Gustavo Avelino de Lima Jacinto. – Brasília, DF, 2022-

109 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Dr. Vandor Roberto Vilardi Rissoli

Trabalho de Conclusão de Curso – Universidade de Brasília - UnB
Faculdade UnB Gama - FGA , 2022.

1. Sistema Tutor Inteligente. 2. Aplicativos Móveis. I. Prof. Dr. Vandor Roberto Vilardi Rissoli. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Aplicativo para o sistema operacional iOS integrado a um Sistema Tutor Inteligente para apoio à Educação

CDU 02:141:005.6

Luis Gustavo Avelino de Lima Jacinto

Aplicativo para o sistema operacional iOS integrado a um Sistema Tutor Inteligente para apoio à Educação

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 23 de Setembro de 2022:

Prof. Dr. Vandor Roberto Vilardi Rissoli
Orientador

Prof. Dr. André Barros de Sales
Convidado 1

Prof. Dr. Ricardo Ramos Fragelli
Convidado 2

Brasília, DF
2022

Agradecimentos

A Deus, por me conduzir e capacitar na caminhada até aqui.

A minha família, que sempre apoiou meus estudos e fizeram o possível para me dar o melhor: meu pai, Almir de Lima Jacinto, minha mãe, Joselene Vieira Avelino de Lima Jacinto, e meus irmãos, Leonardo Avelino de Lima Jacinto e Lucas Avelino de Lima Jacinto. Muito obrigado por tudo, vocês foram minha base para chegar até aqui.

A minha esposa, Laryssa Ribeiro Avelino, por ter sido colaboradora, paciente e compreensiva ao longo dessa trajetória.

Por fim, e não menos importante, aos professores que tive ao longo da graduação, em especial, meu orientador, Vandor Roberto Vilardi Rissoli, pelo auxílio e suporte durante a execução deste trabalho.

Resumo

A correta utilização da informática no contexto educacional pode promover melhorias ao processo de aprendizagem e assimilação de conteúdo. Os Sistemas Tutores Inteligentes (STIs) atuam nesse contexto, uma vez que são programas destinados ao ensino e que utilizam técnicas de Inteligência Artificial para melhorar e individualizar a experiência de aprendizagem. Nessa conjuntura, softwares como o Sistema de Apoio Educacional (SAE), que é um STI, foram desenvolvidos para cumprir a finalidade de auxiliar diferentes perfis de usuários, como discentes e docentes, que participam de forma direta ou indireta do processo de ensino e de aprendizagem. Contudo, muitos destes sistemas não foram construídos visando os dispositivos móveis, que se tornaram parte da vida moderna e são um possível aliado à Educação. Consciente dessa situação, a presente pesquisa teve como objetivo prover aos estudantes usuários do SAE a facilidade de acesso e utilização deste sistema, com o intuito de contribuir ao contexto educacional, fornecendo agilidade para continuidade do processo de aprendizagem, por intermédio dos dispositivos móveis. Assim, trata-se de uma pesquisa aplicada, que utilizou os procedimentos de pesquisa bibliográfica e produção tecnológica para compreender o contexto e desenvolver um aplicativo do SAE para o sistema operacional iOS que possa auxiliar no processo de aprendizagem. Foram utilizadas as metodologias *Scrum* e *Kanban*, na etapa de desenvolvimento, que abrangeu o levantamento e priorização de requisitos, fases de implementação e teste da aplicação. Os resultados apresentados tiveram como base dois principais aspectos: as experiências subjetivas dos usuários que testaram o *software* e a partir da análise do atendimento aos objetivos propostos às funcionalidades desenvolvidas.

Palavras-chaves: Informática na Educação. Sistema Tutor Inteligente. Dispositivos Móveis. Sistema de Apoio Educacional.

Abstract

The correct use of information technology in the educational context can promote improvements to the processes of learning and assimilation of content. Intelligent Tutoring Systems (ITS) operates in this context, since they are programs intended for teaching and use Artificial Intelligence techniques to improve and individualize the learning experience. In this conjuncture, softwares such as the Sistema de Apoio Educacional (SAE, which is an ITS, were developed to fulfill the purpose of assisting different user profiles, such as students and professors, who participate directly or indirectly in the teaching and learning processes. However, many of these systems were not built with mobile devices in mind, which have become part of modern life and are a possible ally to Education. Aware of this situation, the present research aimed to provide students using SAE with ease of access and use of this system, in order to contribute to the educational context, providing agility for the continuity of the learning process, through mobile devices. Thus, it is an applied research, which used the procedures of bibliographic research and technological production to understand the context and develop a SAE application for the iOS operating system that can assist in the learning process. Scrum and Kanban methodologies were used in the development stage that covered the requirements gathering and prioritization, implementation phases and application testing. The results presented were based on two main aspects: the subjective experiences of the users who tested the software and from the analysis of the fulfillment of the proposed objectives of the developed functionalities.

Key-words: Informatics in Education. Intelligent Tutoring System. Mobile Devices. Sistema de Apoio Educacional

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

API	*Application Programming Interface*
BDQ	Banco de Questões
BPMN	*Business Process Model and Notation*
FGA	Faculdade do Gama
IDE	*Integrated Development Environment*
ITA	*Intelligent Teaching Assistant*
LaTEd	Laboratório de Tecnologias Educacionais
LFy	Lógica Fuzzy
ML3P	Modelo Logístico de Três Parâmetros
ML4P	Modelo Logístico de Quatro Parâmetros
PMon	Projeto de Monitoria Estudantil
SAE	Sistema de Apoio Educacional
SDK	*Software Development Kit*
SIAC	Sistema Inteligente de Avaliação do Conhecimento
STI	Sistema Tutor Inteligente
TA	Testes Adaptativos
TAI	Testes Adaptativos Informatizados
TAS	Teoria da Aprendizagem Significativa
TCC	Trabalho de Conclusão de Curso
TCF	Teoria dos Conjuntos Fuzzy
TRI	Teoria de Resposta ao Item
UnB	Universidade de Brasília

Sumário

1 Introdução

O objetivo deste capítulo é apresentar o contexto no qual o trabalho está inserido. A partir do problema apontado foram apresentadas as questões de pesquisas e a proposta de solução. Os objetivos e as orientações sobre a arquitetura e organização do documento estão descritos no decorrer deste capítulo.

1.1 Contextualização

O uso da informática na educação tem crescido notavelmente nos últimos anos. O uso adequado da informática proporciona diversos benefícios à aprendizagem, como oportunizar o desenvolvimento e a organização na construção do conhecimento, despertar o interesse e a curiosidade dos alunos, além de promover a evolução de outras habilidades nos aprendizes (MATTEI, 2013).

Em seu estudo, Mattei (2013) aponta que para a educação utilizar a informática de maneira qualitativa é necessário que se articule quatros aspectos: computador, software educacional, professor e estudante (aprendiz). O uso do computador, juntamente com o software educacional, pode ser visto como uma importante ferramenta de auxílio ao processo de aprendizagem.

Para cumprir essa finalidade, existe o Sistema de Apoio Educacional (SAE), que foi desenvolvido envolvendo pesquisadores de três universidade brasileiras nos primeiros anos deste milênio, e desde 2006 são publicados trabalhos científicos sobre esse projeto e os seus resultados em diferentes áreas (FURTADO; RISSOLI, 2019), (SANTOS; RISSOLI, 2011), (NUNES; SANTOS; RISSOLI, 2018).

O SAE é um software educacional elaborado respeitando os principais fundamentos propostos pela Teoria da Aprendizagem Significativa (TAS). Este software estende a arquitetura tradicional de um Sistema Tutor Inteligente (STI), por meio da integração de novos módulos em sua estrutura, buscando atender às necessidades dos diferentes perfis de usuários desse software que participam direta ou indiretamente do processo de ensino e de aprendizagem (SOUZA; SANTOS; RISSOLI, 2019).

O SAE, como recurso tecnológico interativo de apoio educacional, tem sido utilizado em diferentes áreas de conhecimento e trabalhado com diferentes níveis educacionais, a partir dos anos finais do ensino fundamental até cursos de pós-graduação, além de diferentes modalidades de ensino, como ensino presencial, semipresencial e à distância (SOUZA; SANTOS; RISSOLI, 2019).

Este STI vem sendo empregado em diferentes áreas de conhecimento nos cursos

de Engenharia da Universidade de Brasília (UnB), na Faculdade do Gama (FGA), por exemplo na disciplina de Introdução à Engenharia, que insere os novos estudantes no contexto das engenharias, nos conteúdos explorados por mais que uma disciplina como em Lógica de Programação, Orientação a Objetos, Banco de Dados e alguns cursos de extensão universitária oferecidos pelo Laboratório de Tecnologias Educacionais (LaTEd) para docentes e discentes da UnB e da comunidade.

As pesquisas em STIs ganharam popularidade a partir dos anos 90 (FERREIRA FILHO, 2008), período que também foi fortemente impactado pela utilização dos computadores e a Internet. O SAE surge a partir desse contexto, sendo projetado para ser utilizado pelo conjunto de computador e Internet.

Contudo, com o passar do tempo, os dispositivos móveis se tornaram parte integrante da vida moderna em todo o mundo, sendo que, há alguns anos o número de celulares superou até mesmo a quantidade de computadores pessoais.

Tendo em vista que os dispositivos móveis proporcionam a capacidade de acesso a uma grande diversidade de conteúdos em qualquer hora e lugar e que tais características fazem destas ferramentas adequadas a serem utilizadas nos mais diversos contextos (MOURA, 2009), inclusive no educativo, a utilização do SAE em dispositivos móveis pode significar um avanço ao processo de aprendizagem.

Dessa forma, considerando que o uso da informática na educação proporciona diversos benefícios à aprendizagem, bem como que a evolução dos dispositivos móveis fez com que se tornassem parte integrante da vida moderna, inclusive dos estudantes, percebeu-se a existência de espaço a ser explorado neles pelo SAE.

1.2 Questão de Pesquisa

O presente trabalho tem como finalidade o desenvolvimento de um aplicativo do SAE. Por meio do uso de conceitos da Engenharia de Software, foi feito o levantamento de requisitos, implementação e validação com o usuário final. Assim, a questão de pesquisa que foi discutida neste trabalho é: *Como proporcionar ao estudante usuário do sistema SAE a possibilidade de utilização do mesmo sistema em dispositivo móvel com sistema operacional iOS como uma ferramenta que poderá auxiliá-lo no seu processo de aprendizagem?*

1.3 Justificativa

O presente trabalho parte do princípio de que as ferramentas computacionais, como os aplicativos móveis e a inteligência artificial, podem contribuir de forma positiva para a aprendizagem dos estudantes.

A utilização de dispositivos móveis em ambientes de aprendizagem foi examinada empiricamente em diversos estudos e concluiu-se que houve melhora na comunicação e aprendizagem colaborativa em sala de aula (LEITE, 2014).

Além do mais, Fonseca (2013) aponta que o *Mobile Learning*, conceito que representa a aprendizagem entregue ou suportada por meio de dispositivos de mão que carregam ou manipulam informações, acrescenta ao processo de aprendizagem por ampliar o tempo e o espaço de estudo ao quebrar as barreiras temporais e espaciais, uma vez que permite ao aprendiz acessar o material de estudo em momentos e contextos diversos.

Assim, o *Mobile Learning* pode ser visto como uma alternativa ao método tradicional de ensino, aliando os dispositivos móveis, tecnologias que os estudantes estão acostumados a utilizar diariamente, ao processo de aprendizagem.

Ademais, Marghescu, Chicioreanu e Marghescu (2007) citam aspectos positivos de estudantes que já fizeram uso do *Mobile Learning*. São eles:

- Melhora as habilidades de escrita, leitura e matemáticas;
- Encoraja o aluno no trabalho individual, bem como no trabalho em grupo;
- Ajuda o estudante a identificar áreas em que precisa de auxílio;
- Auxilia o aluno a se manter focado por mais tempo;
- Ajuda a melhorar a sua autoconfiança;
- Transforma parte do método tradicional das aulas.

Não obstante esses benefícios, dentre as pesquisas de STIs citadas neste trabalho, nenhum deles têm como plataforma final os dispositivos móveis. O SAE, no entanto, atuou nesse sentido, disponibilizando um aplicativo para o sistema operacional Android.

Este aplicativo implementa os módulos Banco de Questões (BDQ), Orientação Pedagógica e Módulo de Interface Animado (MInA), como pode ser observado na Figura 1, que busca atender às possíveis necessidades dos estudantes usuários do SAE.

De acordo com StatCounter (2020), aproximadamente 25% do mercado de dispositivos móveis no mundo é composto pelo sistema operacional iOS, que atualmente não é contemplado pelo SAE. Diante desta constatação, o presente trabalho se propôs a criar um aplicativo móvel do SAE para a plataforma iOS.

Esta aplicação, vide Figura 2, mantém o foco no estudante, em que foram implementados os módulos BDQ, Orientação Pedagógica e a MInA. Além das funcionalidades já existente na versão Android do SAE, foi implementado na versão iOS o Sistema Inteligente de Avaliação do Conhecimento (SIAC), que complementa alguns dos processos de

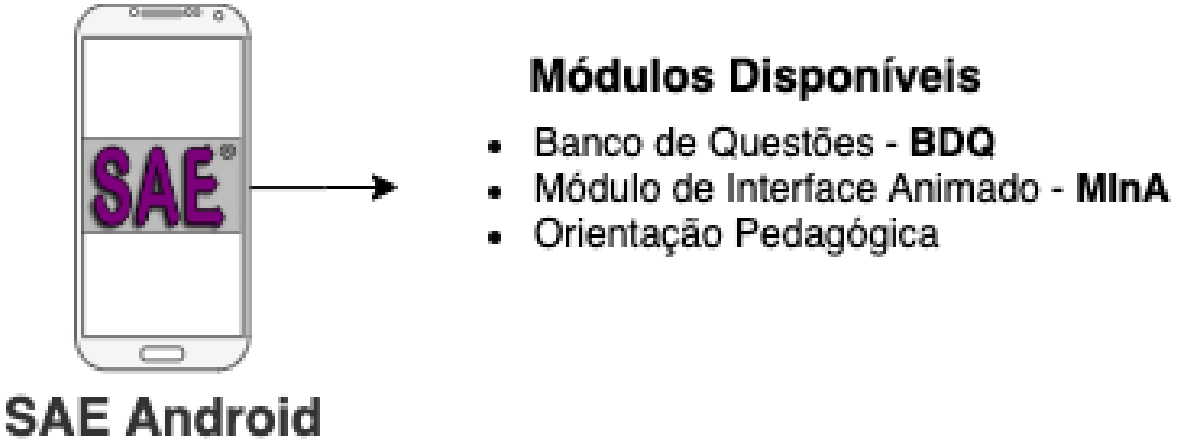

Figura 1 – Módulos do SAE, versão Android. Fonte: Autor.

inferência realizados pelo SAE, dando auxílio ainda maior ao estudante no processo de aprendizagem e aos docentes no acompanhamento de cada aprendiz.

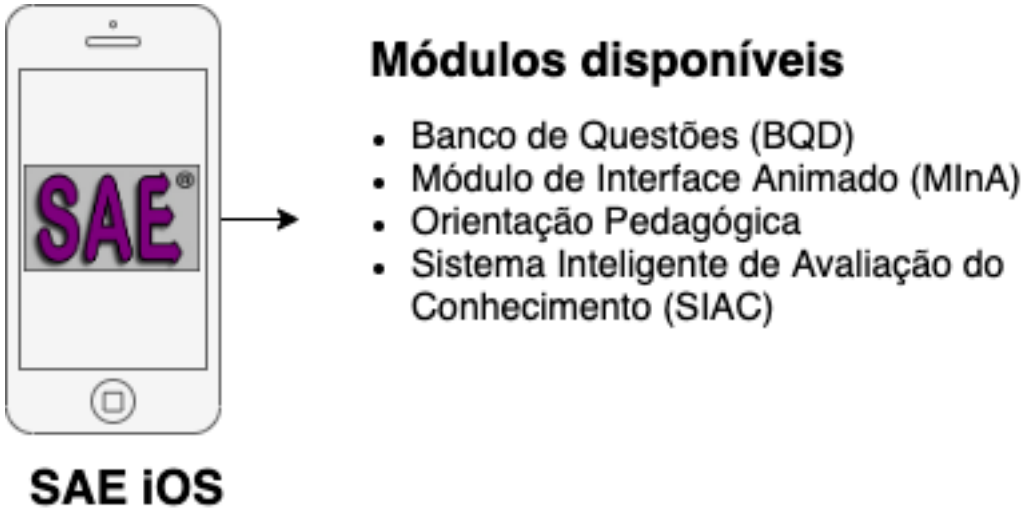

Figura 2 – Módulos do SAE, versão iOS. Fonte: Autor.

1.4 Objetivos

1.4.1 Objetivo geral

O objetivo principal deste trabalho é desenvolver uma aplicação do SAE para o sistema operacional iOS, fornecendo interação adequada aos usuários dessa plataforma nos módulos BDQ, Orientação Pedagógica, MInA e SIAC.

1.4.2 Objetivos específicos

- Proporcionar ao discente a capacidade de resolver questões e testes nas disciplinas que está cursando pelo dispositivo móvel iOS;
- Desenvolver uma aplicação compatível com o sistema operacional iOS para acesso seguro e contínuo com o SAE;
- Conceder ao discente uma ferramenta alternativa para realizar atividades e acompanhar sua situação de aprendizagem em tempo real e de maneira gráfica.

1.5 Metodologia

As metodologias utilizadas no presente trabalho podem ser divididas em duas categorias: metodologia de pesquisa e metodologia de desenvolvimento.

De acordo com GERHARDT e SILVEIRA (2009), as pesquisas podem ser classificadas quanto à sua natureza, abordagem, objetivos e procedimentos. Assim, a Tabela 1 apresenta, de forma resumida, a classificação nas diferentes categorias apresentadas acima, definidas para a realização deste trabalho.

Metodologia			
Natureza	**Abordagem**	**Objetivos**	**Procedimentos (principais)**
Aplicada (tecnológica)	Qualitativa	Descritiva	Pesquisa Bibliográfica Produção Tecnológica

Tabela 1 – Metodologia de Pesquisa. Fonte: Autor.

Sobre a metodologia de desenvolvimento utilizada neste trabalho, uma utilização do *Scrum*, em combinação com a metodologia *Kanban* foi aplicada. A metodologia de pesquisa está descrita detalhadamente no capítulo 3 e a metodologia de desenvolvimento está descrita no capítulo 4 deste trabalho, no qual foram explorados os aspectos mais relevantes desta proposta e suas características, propiciando uma análise mais significativa ao seu contexto e necessidades.

1.6 Organização do Trabalho

O presente trabalho foi dividido em seis capítulos. Segue a síntese do que foi explorado em cada um deles:

1. **Capítulo 1 - Introdução:** refere-se à contextualização, levantamento da questão de pesquisa, justificativa e definição dos objetivos principais do trabalho;

2. **Capítulo 2 - Referencial Teórico:** refere-se ao conhecimento acadêmico necessário para compreender a presente pesquisa e seus recursos. O capítulo é subdividido nas seções de Desenvolvimento de Aplicativos Móveis e de Sistema Tutor Inteligente;

3. **Capítulo 3 - Metodologia:** este capítulo explica detalhadamente as metodologia de pesquisa empregada na elaboração do trabalho;

4. **Capítulo 4 - Proposta:** aborda a metodologia de desenvolvimento utilizadas na construção do trabalho, além de apresentar as ferramentas e tecnologias usadas para o auxílio no desenvolvimento da pesquisa e na implementação do trabalho proposto;

5. **Capítulo 5 - Desenvolvimento:** retrata o desenvolvimento da proposta durante a implementação, respeitando a metodologia escolhida. Faz o detalhamento de todo o processo de desenvolvimento, junto com as dificuldades e alterações que ocorreram durante o desenvolvimento do projeto.

6. **Capítulo 6 - Considerações Finais:** este capítulo apresenta a conclusão deste trabalho, considerando os resultados obtidos ao final do projeto. Neste capítulo os trabalhos futuros são apresentados a partir de ideias e demandas encontradas no presente trabalho, podendo continuar a sua elaboração por meio de evoluções significativas.

2 Referencial Teórico

A finalidade deste capítulo é discorrer sobre os principais conceitos necessários para fundamentar o presente trabalho e elaborar a proposta de solução. Assim, o capítulo engloba conceitos relevantes ao projeto, tais como: aplicativo móvel e sistema tutor inteligente.

2.1 Aplicativos Móveis

A finalidade desta seção é apresentar desde a perspectiva histórica dos aplicativos móveis, surgimento e popularização, até detalhes específicos de desenvolvimento, com o objetivo de prover embasamento para o desenvolvimento do presente trabalho.

2.1.1 Perspectiva Histórica

Os aplicativos móveis (apps) foram desenvolvidos inicialmente para solucionar tarefas triviais, funcionando como calculadoras, calendários e conversores de moeda, por exemplo (GASIMOV et al., 2010).

Houve uma evolução nas tecnologias de redes e serviços que resultaram também em novas versões de aplicativos (NONNENMACHER; MENEZES, 2012). Versões limitadas de aplicações web foram desenvolvidas como aplicações móveis e, apesar dos esforços de melhoria das inadequações de *hardware* dos dispositivos móveis, a performance das atividades relacionadas à leitura e digitação ainda poderiam ser problemáticas quando comparadas aos computadores pessoais.

Nos primeiros anos do desenvolvimento de aplicações móveis foram caracterizados pela introdução de aplicações web em dispositivos móveis e muitos dos grandes aplicativos populares de redes sociais na web tiveram uma versão móvel.

Com as lacunas tecnológicas entre computadores pessoais e dispositivos móveis se tornando menores, mudanças mínimas em aplicações web foram necessárias para que se tornassem totalmente compatíveis com as aplicações móveis (GASIMOV et al., 2010). Apesar desse processo já ter sido considerado viável, ele não pode ser uma solução concreta, pois não utiliza os recursos exclusivos dos dispositivos móveis como comunicação em qualquer hora e lugar, por exemplo.

Assim, da inserção como apenas acessórios para dispositivos móveis, os aplicativos móveis se tornaram uma plataforma para fins sociais e comerciais.

2.1.2 Processo de distribuição de aplicativos móveis

Por muitos anos, o desenvolvimento de aplicativos e serviços para dispositivos móveis foi, em maior parte, controlado e administrado por operadoras de redes móveis, fabricantes de dispositivos móveis e provedores de conteúdo (HOLZER; ONDRUS, 2011).

Com a introdução de plataformas como Android e iOS, a estrutura do mercado e cadeia de valor da indústria de aplicativos móveis evoluiu consideravelmente, funções tradicionalmente estabelecidas foram alteradas, combinadas e trocadas. As operadoras de rede móveis perderam o controle do dispositivo, provedores de portal obtiveram novos fluxos de receita e instituições financeiras e provedores de conteúdo se integraram mais às plataformas (HOLZER; ONDRUS, 2011).

O processo de distribuição de aplicativos móveis é aquele pelo qual um aplicativo é desenvolvido, trazido ao mercado e comprado por consumidores, no caso de aplicativos pagos, ou baixado sem custos, no caso de aplicativos gratuitos. Este envolve três componentes, como pode ser observado na ilustração da Figura 3.

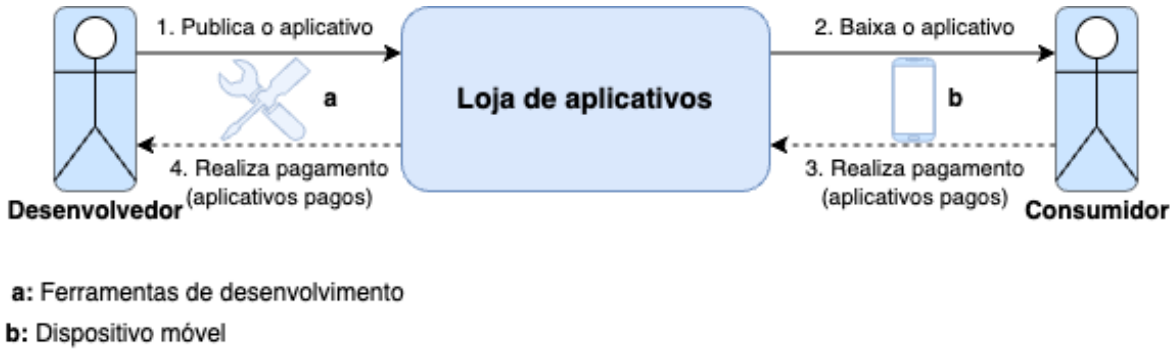

Figura 3 – Processo de distribuição de aplicativos móveis. Fonte: Adaptado de Holzer e Ondrus (2011).

Primeiramente, são utilizadas ferramentas de desenvolvimento para o processo de criação do aplicativo. Depois, o desenvolvedor o publica em uma loja de aplicativos, em que o consumidor poderá baixá-lo no seu dispositivo móvel.

Na Figura 3 é apresentado um modelo de dois lados, com desenvolvedores de um lado e consumidores do outro. Este modelo difere da antiga abordagem em que as operadoras de rede móveis estavam encarregadas de ser a interface entre clientes e prestadores de serviços.

No modelo apresentado, um aumento ou diminuição de um lado gera um efeito similar do outro. Assim, o aumento de consumidores em uma loja de aplicativos, por exemplo, atrairia desenvolvedores para essa plataforma (HOLZER; ONDRUS, 2011).

2.1.3 Desenvolvimento

Um aplicativo móvel corresponde a um software projetado para ser executado em dispositivos como *smartphones*, *tablets*, dentre outros (INUKOLLU et al., 2014), sendo inegável o enorme crescimento da indústria dos dispositivos móveis, que estima que até 2023 existam cerca de 6.8 bilhões de usuários de *smartphones* (STATISTA, 2022). Além disso, até 2025, 5.5 bilhões de pessoas estarão utilizando a Internet de forma móvel (STATISTA, 2022), reforçando a sua importância, pois estes conectam a computação à portabilidade.

Segundo El-Kassas et al. (2017), o crescimento de usuários de *smartphones* aumenta a demanda por aplicações que supram as necessidades dos usuários nas mais variadas áreas, como no Turismo, Saúde e Educação.

O desenvolvimento de aplicações móveis possui características particulares no desenvolvimento de software, pois devem ser considerados aspectos peculiares como os recursos do dispositivo móvel, a mobilidade e as especificações do dispositivo, como tamanho de tela, design e navegação da interface do usuário, privacidade e segurança (EL-KASSAS et al., 2017).

Segundo Amatya e Kurti (2013), os dois maiores desafios no desenvolvimento de aplicativos móveis são (i) a fragmentação de dispositivos e (ii) a fragmentação de sistemas operacionais, que resultam em uma aplicação funcionar corretamente em um smartphone e não funcionar em outro.

A fragmentação pode ser vista como *"A incapacidade de desenvolver uma aplicação em um contexto operacional de referência e alcançar o comportamento compreendido em todos os contextos"* (AMATYA; KURTI, 2013). Assim, o ecossistema de usuários de aplicativos, desenvolvedores, provedores de conteúdo e fabricantes de dispositivos são afetados pela fragmentação.

Quanto à fragmentação de dispositivos, Giron, Mendoza e Torres-Huitzil (2011) apontam que, com o intuito de atrair mais público, os fabricantes tendem a desenvolver funcionalidades diferenciadas nos smartphones, o que resulta em uma grande diversidade de funcionalidades e, consequentemente, na falta de uniformidade entre os dispositivos.

Estes fatores tornam o processo de desenvolvimento de aplicativos não homogêneo, podendo resultar no aumento dos custos, bem como na criação de versões inconsistentes para a mesma aplicação (GIRON; MENDOZA; TORRES-HUITZIL, 2011).

De modo similar, a fragmentação de sistemas operacionais também agrava as dificuldades no desenvolvimento móvel. Os fornecedores no mercado móvel têm suas plataformas rodando em diferentes sistemas operacionais e fornecem o próprio conjunto de ambiente e ferramentas de desenvolvimento na forma de SDKs (*Software Development Kits*), direcionados e otimizados para suas próprias plataformas (AMATYA; KURTI, 2013).

O desenvolvimento de aplicações móveis pode ser classificada em três grandes grupos (IBM, 2012):

1. **Aplicações Nativas:** desenvolvidos a partir da plataforma nativa do dispositivo de destino, utilizando-se diretamente as funções do sistema operacional e a distribuição por meio das lojas oficiais

2. ***Web Apps*:** desenvolvidos utilizando tecnologias *Web*, como *Hypertext Markup Language* HTML, *Cascading Style Sheets* CSS e *Javascript* e executadas por meio de *Web Views*, sem acesso a recursos avançados do sistema operacional e sem a possibilidade de distribuição por meio de lojas oficiais;

3. **Aplicações Híbridas:** desenvolvimento para múltiplos sistemas operacionais através de uma única plataforma e linguagem, tendo a possibilidade de compilar de forma nativa.

Segundo Heitkötter, Hanschke e Majchrzak (2013), os aplicativos híbridos e *web apps* surgem no panorama de crescimento exponencial da demanda por aplicativos, partindo da necessidade de desenvolvimento mais rápido, utilizando uma única base de código que pode ser executada em mais de uma plataforma. Dessa forma, desenvolver se tornava mais fácil e barato, pois removia a necessidade dos desenvolvedores terem domínio do conhecimento de construção de aplicativos para diferentes plataformas.

Apesar disso, o desenvolvimento nativo, definido como desenvolvimento de aplicações exclusivas para uma plataforma, por meio do SDK desta (AMATYA; KURTI, 2013), apresenta as seguintes vantagens em relação às abordagens web e híbridas:

- Apresentam melhor desempenho (PREZOTTO; BONIATI, 2014), (BERNARDES; MIYAKE, 2016), (SHAH; SINHA; MISHRA, 2019);

- Apresentam melhor performance (PREZOTTO; BONIATI, 2014), (EL-KASSAS et al., 2017), (SHAH; SINHA; MISHRA, 2019);

- Tem acesso total as APIs (*Application Programming Interface*) do dispositivo. Como, por exemplo, GPS (*Global Positioning System*), câmera e acelerômetro (PREZOTTO; BONIATI, 2014), (AMATYA; KURTI, 2013), (SHAH; SINHA; MISHRA, 2019);

- Garantem a melhor usabilidade (AMATYA; KURTI, 2013), (BERNARDES; MIYAKE, 2016);

- Garantem a melhor experiência móvel (AMATYA; KURTI, 2013), (SHAH; SINHA; MISHRA, 2019);

- Apresentam aparência e comportamento nativo da interface do usuário (EL-KASSAS et al., 2017);
- A publicação do aplicativo na loja de aplicativos da respectiva plataforma é geralmente mal vista no caso de abordagens multiplataforma e têm uma prioridade mais baixa na lista de recomendações. Alguns aplicativos que simplesmente injetam código HTML e JavaScript dentro de um contêiner nativo fino são rejeitados para publicação (SHAH; SINHA; MISHRA, 2019);
- Quanto maior e mais complexo se torna uma aplicativo, congelamentos e travamentos acabam sendo mais comuns. Isso ocorre pois o suporte para todos os dispositivos ainda é praticamente inviável em soluções multiplataforma (SHAH; SINHA; MISHRA, 2019);
- Alta performance, customização e conformidade com as diretrizes da plataforma de UI (*User Interface*) e UX (*User Experience*) (MEIRELLES et al., 2019);
- Baixo estresse de *hardware* (MEIRELLES et al., 2019);
- Fácil configuração do ambiente de trabalho e grande número de ferramentas de desenvolvimento (MEIRELLES et al., 2019).

Meirelles et al. (2019) recomendam a escolha de abordagens nativas para projetos com uma prioridade baixa em orçamento, que não precisam ser publicados em várias plataformas, que precisam de alto desempenho, que tenham necessidade de suporte adequado dos fornecedores da tecnologia e que precisam de uma curva de aprendizado mais rápida para a equipe de desenvolvimento.

2.1.3.1 Aplicações Nativas

Aplicações nativas são aquelas desenvolvidas especificamente para uma plataforma de destino, utilizando SDKs e *frameworks* desta (HEITKöTTER; HANSCHKE; MAJCHRZAK, 2013). Na abordagem de desenvolvimento nativa, o aplicativo é exclusivo para a plataforma de destino, podendo ser baixado e instalado através da loja de aplicativos desta plataforma.

Apesar da existência de vários sistemas operacionais para dispositivos móveis, o Android e o iOS somam, conjuntamente, mais de 99% do mercado mundial, sendo estes os sistemas operacionais de maior relevância (STATCOUNTER, 2020).

2.1.3.1.1 iOS

O sistema operacional iOS, lançado em 2007 juntamente com o primeiro iPhone, é o SO que roda em iPhone, iPad e iPod Touch. O sistema operacional gerencia o *hardware*

do dispositivo e fornece as tecnologias necessárias para implementar aplicativos nativos (APPLE, 2010).

O iOS SDK contém as ferramentas e interfaces necessárias para desenvolver, instalar, executar e testar aplicativos nativos que podem ser utilizados em dispositivos iOS (APPLE, 2010). Aplicativos nativos são criados utilizando as *frameworks* do sistema iOS e a linguagem de programação Objective-C ou Swift, e executam diretamente no iOS (APPLE, 2010) (APPLE, 2020b).

Em alto nível, o iOS atua como intermediário entre o *hardware* e os aplicativos. Os aplicativos raramente conversam diretamente com o *hardware*, em vez disso, os aplicativos se comunicam com este por meio de um conjunto de interfaces de sistema bem definidas, que protegem o aplicativo contra alterações de *hardware* (APPLE, 2010).

A implementação das tecnologias iOS também pode ser vista como que baseado em camadas, como consegue ser observado na Figura 4.

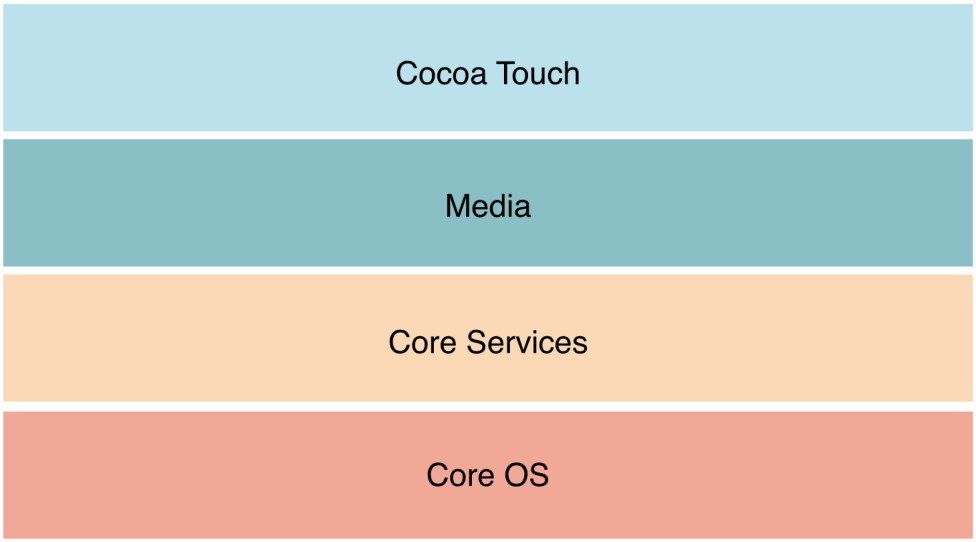

Figura 4 – Arquitetura do sistema operacional iOS. Fonte: (APPLE, 2010).

Apple (2010) define as funcionalidades das camadas da arquitetura do sistema operacional iOS, que são apresentadas a seguir:

- **Cocoa Touch:** contém as principais estruturas para a criação de aplicativos iOS. Esta camada define infraestrutura básica de aplicativos e suporte para tecnologias chave, como multitarefa, entrada baseada em toque, *push* notificações e muitos serviços de sistema de alto nível.

- **Media:** contém as tecnologias de gráficos, áudio e vídeo voltadas para a criação de experiências multimídia. As tecnologias nessa camada foram projetadas para facilitar a criação de aplicativos com ótima aparência e som.

- **Core Services:** contém os serviços fundamentais do sistema que todos os aplicativos usam, como localização, telefonia e *threads* (APPLE, 2010). Ainda que muitos aplicativos não utilizem esses serviços diretamente, muitas partes do sistema são construídas sobre eles.

- **Core OS:** contém os recursos de baixo nível em que a maioria das outras tecnologias é desenvolvida. Ainda que muitos aplicativos não utilizem essa tecnologia diretamente, elas provavelmente serão usadas por outras *frameworks*. Em situações que o aplicativo precisa lidar explicitamente com segurança ou se comunicar com um *hardware* externo acessório, isto é feito usando as *frameworks* desta camada.

A maioria das interfaces de sistema é entregue em pacotes especiais chamados *frameworks*. Uma *framework* é um diretório que contém uma biblioteca compartilhada dinâmica e os recursos, como arquivos de cabeçalho, imagens, aplicativos auxiliares, entre outros, necessários para dar suporte a essa biblioteca (APPLE, 2010).

O ambiente de desenvolvimento utilizado para criar, testar, depurar e ajustar os aplicativos é o Xcode, que é um invólucro para todas as outras ferramentas necessárias para criar os aplicativos (DEVELOPER, 2020).

Para rodar aplicativos diretamente em dispositivos iOS, ao invés de simuladores, utilizar recursos avançados e para distribuir aplicativos a App Store, loja de aplicativos iOS, é necessário a licença do *Apple Developer Program*, que tem o custo de $99 (noventa e nove dólares) anuais (APPLE, 2020a).

2.2 Sistema Tutor Inteligente

O objetivo desta seção é apresentar a perspectiva histórica dos STIs, suas características principais e arquitetura e, por fim, o detalhamento do SAE e, mais especificamente, dos módulos abordados neste projeto.

2.2.1 Perspectiva Histórica

As pesquisas em STI tiveram início em 1973 com as publicações: *The design and evaluation of an adaptive teaching system* (HARTLEY, 1973) e *Towards more intelligent teaching systems* (HARTLEY; SLEEMAN, 1973), mas o tema ganhou popularidade a partir dos anos 90 (FERREIRA FILHO, 2008).

Os STIs são programas voltados para o ensino e que utilizam técnicas de Inteligência Artificial (IA) para proporcionar ao estudante uma experiência de ensino personalizada, simulando a interação entre aluno e professor (FERREIRA FILHO, 2008).

A utilização da IA é o que diferencia os STIs de seu precursor histórico, os sistemas CAI (*Computer Aided Instruction*). Nestes sistemas mais antigos as instruções programadas eram seguidas independentemente das ações do estudante, assim, a interação não modificava a partir das ações do aprendiz. A maneira de contornar esta restrição foi com a utilização de técnicas de IA, o que deu surgimento aos sistemas ICAI(*Intelligent Computer Aided Instruction*), que atualmente são chamados de STIs e Ambientes Inteligentes de Aprendizagem (RAABE, 2005).

Os sistemas CAI guiavam o aprendiz para a resposta correta através de estímulos previamente planejados, enquanto os sistemas ICAI pretendiam simular as capacidades cognitivas do estudante e tomar decisões pedagógicas com base nos resultados obtidos (GIRAFFA, 1999).

A partir do uso de técnicas de modelagem qualitativa, característica de técnicas de IA, é possível simular o processo de solução de problemas do estudante, compreendendo como ele resolve os problemas que lhe são apresentados (GIRAFFA, 1999).

A Tabela 3 apresenta um comparativo das principais diferenças entre CAI e ICAI.

Tipo de Sistema	**CAI**	**ICAI (STI)**
Origem	Educação	Ciência da Computação
Bases Teóricas	Skinner(behaviorista)	Psicologia Cognitivista
Estruturação e Funções	Uma única estrutura algoritmicamente predefinida, em que o aluno não influi na sequenciação.	Estrutura subdividida em módulos cuja sequenciação se dá em função das respostas do aluno.
Estruturação do Conhecimento	Algorítmica	Heurística
Modelagem do Aluno	Avaliam a última resposta.	Tentam avaliar todas as respostas do aluno durante a interação.
Modalidades	Tutorial, exercício e prática, simulação e jogos educativos	Socrático, ambiente interativo, diálogo bidirecional e guia

Tabela 2 – Diferenças entre CAI e ICAI / STI. Fonte: (FERREIRA FILHO, 2008).

2.2.2 Características

Como apontado por Ferreira Filho (2008), um STI se caracteriza por separar a matéria que se ensina, as estratégias para ensiná-la e as características do aluno, com o objetivo de obter ensino individualizado. Outro aspecto importante é a necessidade da interface de comunicação ser bem projetada, de fácil utilização e que favoreça a comunicação entre o tutor e o aluno.

Jonassen e Wang (1993) consideram que um STI deve passar por três testes para ser considerado inteligente:

1. Conteúdo deve ser codificado no sistema para que o sistema possa acessar as informações, fazer inferências ou resolver problemas;
2. Sistema deve ser capaz de avaliar a aquisição do conhecimento;
3. As estratégias de ensino devem ser projetadas com o objetivo de diminuir a diferença do conhecimento do especialista e o conhecimento do aprendiz.

Urretavizcaya (2001) define que as características mais importantes dos STIs são:

1. O conhecimento do domínio é limitado e claramente articulado;
2. O sistema tem conhecimento do aluno, o que permite guiar e adaptar o ensino;
3. A sequência de ensino não é predeterminada pelo designer do sistema;
4. O sistema realiza processos diagnósticos mais adequados e detalhados ao aluno;
5. A comunicação entre tutor e aluno melhora, permitindo também que o aluno faça perguntas ao tutor.

2.2.3 Arquitetura

O princípio pedagógico que rege a concepção de um STI é que, dado que estudantes possuem estilos e formas de aprendizagem diferentes, o sistema deve conseguir fornecer instruções personalizadas para cada aprendiz, satisfazendo seu estilo e forma de aprendizagem. (BOLZAN; GIRAFFA, 2002).

As arquiteturas dos STIs variam entre as diversas implementações. Porém, estes sistemas têm como base uma arquitetura clássica baseada em quatro componentes fundamentais (FERREIRA FILHO, 2008). Essa arquitetura pode ser observada na Figura 5.

Como apontado pelo mesmo autor, ocorrem variações na nomenclatura dos módulos da arquitetura clássica. Alguns autores utilizam módulo especialista ou módulo de domínio como sinônimo de modelo de domínio, assim como módulo tutor seria equivalente a estratégia de ensino, modelo pedagógico ou modelo do tutor, enquanto que modelo de aluno, módulo aluno e modelo de estudante são equivalentes.

Segundo GIRAFFA e VICCARI (2001), esta proposta de arquitetura trouxe avanços à modelagem de ambientes educacionais, pois separou o domínio da sua forma de

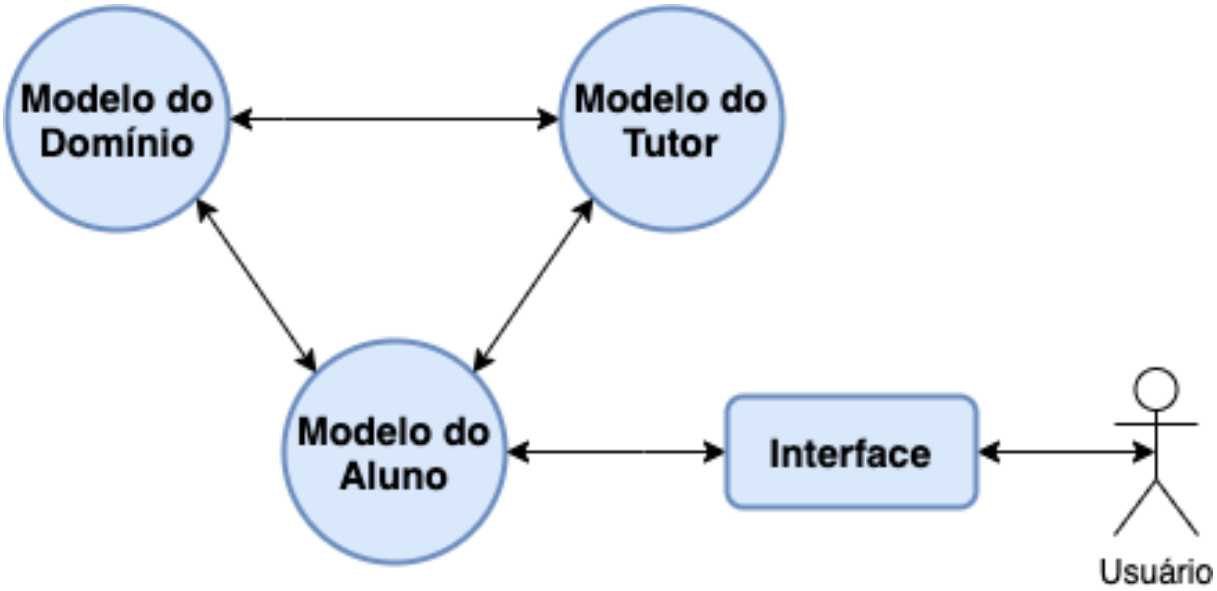

Figura 5 – Arquitetura clássica de um STI. Fonte: Adaptado de (BOLZAN; GIRAFFA, 2002).

manipulação, permitindo que estratégias de ensino fossem associadas em função das informações vindas da modelagem do aluno.

A arquitetura tradicional de um STI é composta por três modelos principais: aluno, domínio e tutor. Cada um destes se caracteriza por armazenar informações sobre o aluno, domínio e as estratégias de ensino, respectivamente. Combinando dinamicamente estes modelos, o sistema se adapta às necessidades específicas do aluno em um determinado instante do processo de aprendizagem (RAABE, 2005).

2.2.3.1 Modelo do Domínio

Segundo GIRAFFA e VICCARI (2001), o modelo do domínio é responsável por armazenar o conhecimento sobre o domínio que se deseja ensinar ao estudante e é constituído pelo material instrucional, por uma sistemática geração de exemplos, pela formulação de diagnósticos e pelos processos de simulação.

Vários modelos de representação de conhecimento podem ser utilizados: redes semânticas, regras de produção, *frames*, *scripts*, orientação a objetos, lógica, dentre outras. A escolha da forma deve ser feita de acordo com o domínio (assunto) em questão (FERREIRA FILHO, 2008).

O modelo do domínio é um componente chave do STI, ali está representado o conteúdo que o tutor irá ministrar. Esse conteúdo fica armazenado, geralmente, em uma base de conhecimento, não em uma base de dados tradicional. Este é um dos fatores principais que diferenciam os CAIs dos STIs. A base de conhecimento é o que permite ao sistema a possibilidade de "raciocinar"sobre o conteúdo representado e armazenado

(GIRAFFA; VICCARI, 2001).

2.2.3.2 Modelo do Tutor

O modelo do tutor é responsável pelas decisões de caráter pedagógico, que incorporam uma metodologia para o processo de aprendizagem (RISSOLI, 2007). Este também possui conhecimento sobre as estratégias e táticas de ensino e as apresenta individualmente para cada aluno, em função de suas características (BOLZAN; GIRAFFA, 2002). Este modelo está em constante interação com o modelo do aluno.

Como apontado por Ferreira Filho (2008) este modelo é responsável por gerar todas as reações do sistema frente às ações e comportamentos do estudante no uso do STI. Quando o aprendiz faz uma requisição ao STI este é responsável por consultar os seus dados, por meio do "modelo do aluno", selecionar uma estratégia de ensino que melhor se encaixe para este estudante e construir o material a ser exibido, utilizando as informações presentes no modelo do domínio, onde estes são selecionados segundo a estratégia de aprendizagem escolhida.

2.2.3.3 Modelo do Aluno

Este modelo armazena todo tipo de informação para cada estudante de forma individual (BOLZAN; GIRAFFA, 2002). Este representa o conhecimento e as habilidades cognitivas do aluno em um dado momento que este interage com o sistema (GIRAFFA, 1999). Este modelo permite ao STI direcionar e orientar o estudante, de acordo com suas necessidades específicas.

A tarefa de construir um modelo do conhecimento do estudante é complexa, especialmente porque a interface de comunicação entre o usuário e o computador limitam a interação mais estreita entre ambos (FERREIRA FILHO, 2008).

De acordo com Rissoli (2007), a principal característica deste modelo está em contemplar todos os aspectos de comportamento e conhecimento que os estudantes possuem e que lhe tragam consequências positivas a sua aprendizagem.

2.2.3.4 Interface

Segundo Raabe (2005), a interface é o meio pelo qual o sistema tutor se comunica com o aluno. Suas duas principais funções são propor atividades de aprendizagem ao aluno e capturar informações dele para atualizar o modelo do aluno.

De acordo com GIRAFFA e VICCARI (2001), a construção de uma boa interface é vital para o sucesso de um STI. Além disso, tem crescido sua importância, pois é por meio dela que o STI exerce suas principais funções.

2.2.4 Sistema de Apoio Educacional (SAE)

No contexto de STIs existe o Sistema de Apoio Educacional (SAE) que é um STI que partiu da arquitetura tradicional e prosseguiu integrando novos módulos em sua estrutura, a fim de melhor atender às demandas educacionais mais ativas e centradas na aprendizagem do aluno (SOUZA; SANTOS; RISSOLI, 2019).

Os diferentes módulos do SAE buscam atender às necessidades dos diferentes processos educacionais, além das diferentes perspectivas dos perfis de usuários que o SAE fornece acesso, sendo os mais tradicionais os alunos, professores, monitores, diretores ou coordenadores de cursos, passando também a fornecer acesso ao assessor do diretor do curso e aos pais ou responsáveis pelo aluno (SOUZA; SANTOS; RISSOLI, 2019).

Diante da diversidade de perfis de usuário, o SAE possui um módulo de interface variado, pois cada tipo de usuário recebe assistência adequada à sua necessidade (SOUZA; SANTOS; RISSOLI, 2019). A assistência a cada um dos diferentes perfis de usuário classifica o SAE como um Assistente Virtual de Ensino Inteligente (ITA - *Intelligent Teaching Assistant*).

Tradicionalmente, os STIs foram projetados para ensinar os estudantes sem a intervenção de professores humanos. Yacef (2002) menciona que os STIs foram dedicados para o perfil do aluno, sendo o professor, geralmente, o gerente administrativo ou o projetista do sistema.

ITAs incluem o professor como um usuário final do sistema e são dedicados tanto ao aluno quanto ao professor para o sucesso do processo de ensino e de aprendizagem (YACEF, 2002). O propósito geral do ITA é dar suporte ao processo educacional de maneira inteligente, auxiliando o professor em suas tarefas de ensino e ajudando os alunos em seus próprios processos de aprendizagem.

Os ITAs podem auxiliar os professores em tarefas como: (i) acompanhar os resultados dos alunos e reportar problemas (enquanto ajuda os alunos a aprender no seu ritmo em um ambiente adaptado à cada estudante) e; (ii) receber *feedback* e exercícios personalizados (YACEF, 2002). Com o ITA o professor continua presente no processo de aprendizagem, sendo elemento fundamental ao sucesso do processo.

O SAE foi proposto como um ITA que integra a Teoria da Aprendizagem Significativa (TAS), como uma metodologia de aprendizagem, junto com uma lógica multivalorada, conhecida como Lógica Fuzzy (LFy), para o acompanhamento da situação de aprendizagem de cada aluno, sendo esta fundamentada pela Teoria dos Conjuntos Fuzzy (TCF) (RISSOLI; SANTOS, 2011). A integração das metodologias do SAE pode ser observada na ilustração da Figura 6.

A TAS propõe uma aprendizagem facilitada por meio da associação do novo conteúdo com conhecimentos pré existentes na estrutura cognitiva do aprendiz, tornando sig-

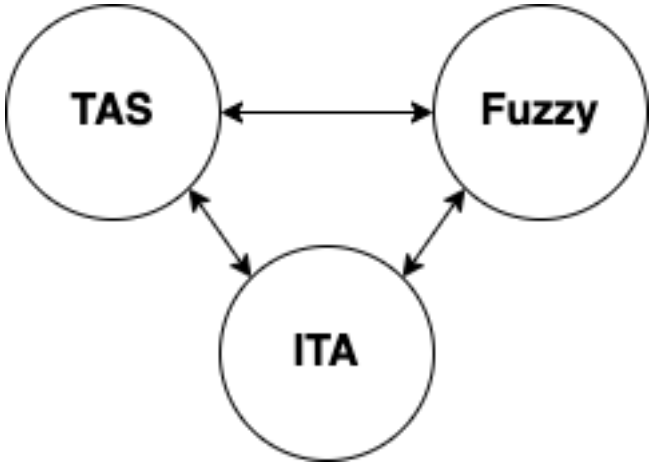

Figura 6 – Integração das metodologias do SAE. Fonte: Adaptado de (RISSOLI, 2007).

nificativo este conhecimento para ele (RISSOLI; SANTOS, 2011). O autor ainda afirma que conhecimentos significativos são aqueles que se estabelecem na estrutura mental do indivíduo, por meio da conexão com conhecimentos mais inclusivos preestabelecidos em sua estrutura mental (processo de subsunção).

Para que ocorra a aprendizagem significativa é necessário que o conteúdo que o aprendiz deve aprender se relacione com aspectos relevantes e preexistentes na estrutura de conhecimento do aprendiz, sendo esta estrutura definida como conceito subsunçor (AUSUBEL; NOVAK; HANESIAN, 1980).

Segundo Rissoli e Santos (2011), as interações que o SAE proporciona aos seus usuários contribuem com o processo de subsunção, ancorando um novo conhecimento em um conceito subsunçor já existente na estrutura mental do aprendiz, promovendo assim a aprendizagem significativa.

A transformação de um novo conteúdo em conhecimento assimilado à estrutura mental do aprendiz, por meio da subsunção, é acompanhado pelo SAE mediante a aplicação da TCF sobre a aprendizagem constatada na interação com este sistema (RISSOLI; SANTOS, 2011).

Baseado na TAS, o SAE acompanha o esforço, desempenho e participação de cada estudante, por meio de análises quantitativas e qualitativas que empregam a Lógica Fuzzy no tratamento das incertezas e imprecisões inerentes às informações (RISSOLI; SANTOS, 2013).

Segundo Rissoli e Santos (2013), a Teoria dos Conjuntos Fuzzy tem como principal objetivo efetuar tratamento matemático sobre certos termos linguísticos subjetivos. Desta forma, torna-se possível realizar análises e cálculos sobre informações vagas e imprecisas.

Estes autores ainda apontam que *"O uso da Lógica Fuzzy se sobressai à Lógica Convencional na realização desta inferência, principalmente na representação do racio-*

cínio humano, onde sua capacidade de expressão se mostra mais coerente ao tratamento necessário destas imprecisões, possibilitando a sua conversão em informações objetivas e melhores definidas por meio do rigor matemático". A inferência, efetuada em linguagem mais natural, pode ser observada na Figura 7.

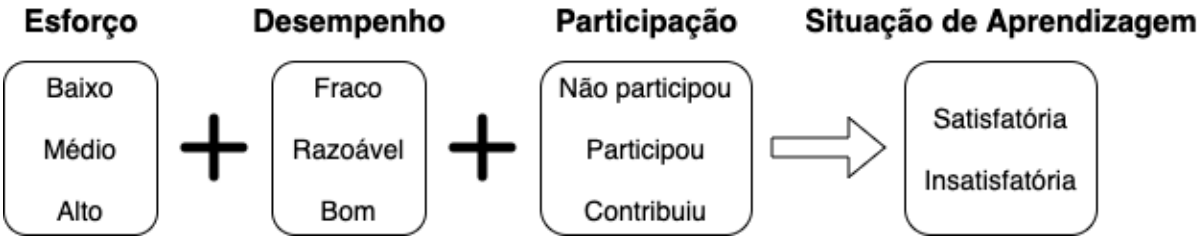

Figura 7 – Representação linguística do SAE pela Lógica Fuzzy. Fonte: Adaptado de (RISSOLI; SANTOS, 2013).

Com isso, o SAE obtêm dados que o permite fornecer orientação pedagógica adequada ao estado cognitivo do aluno em determinado conteúdo e a qualquer momento que o aprendiz possa estar estudando.

2.2.4.1 Arquitetura

A arquitetura proposta para o SAE consiste em uma extensão da arquitetura de elaboração de um ITA proposta por Yacef (2002), que deixa de ser uma arquitetura didática "triangular" e passa a ser "quadrangular", com a inclusão do monitor humano, além de assistir mais perfis participantes do processo educacional, assim como obtém suas colaborações para o sucesso do processo de ensino (RISSOLI, 2007). A arquitetura original do SAE pode ser observada na Figura 8.

Como definido por Rissoli (2007) *"a interação entre todos estes modelos proporciona um eficiente acompanhamento e orientação do aprendiz, através de suas comunicações entre recursos e interfaces de assistência e coleta de dados fornecidos por seus agentes reais, além das realizações do próprio aprendiz junto ao sistema"*.

Rissoli (2007) define cada um dos módulos da arquitetura do SAE nos seguintes termos:

- **Módulo do Domínio**: responsável em prover conteúdo potencialmente significativo que possibilite a organização lógica e direcione a associação dos novos conceitos para estrutura cognitiva do aluno, de maneira substantiva, não-arbitrária e não-literal. Este modelo pode ser formado por textos, imagens, sons, vídeos e exercícios interativos, que geralmente precisam de certas habilidades cognitivas como pré-requisito para serem apresentados.

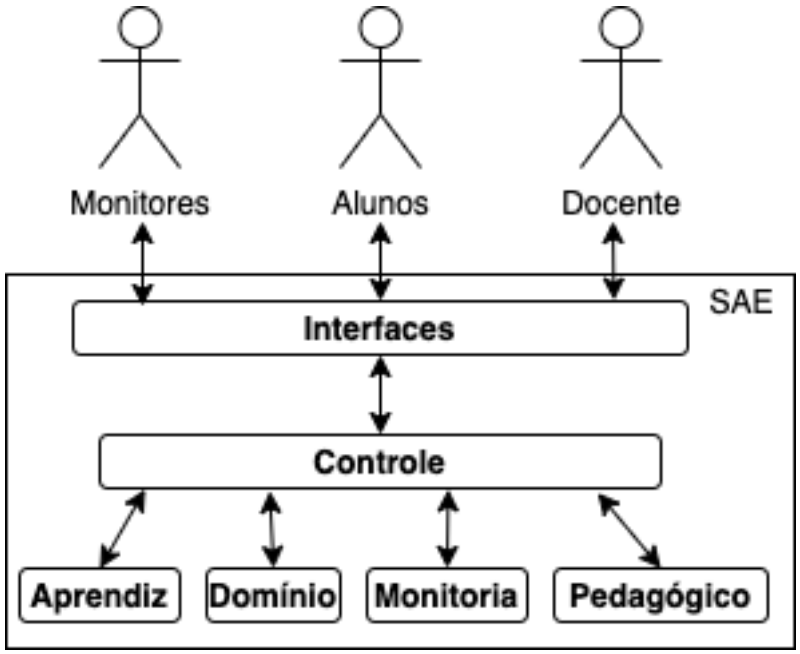

Figura 8 – Representação da Arquitetura do SAE. Fonte: Adaptado de (RISSOLI; SANTOS, 2011).

- **Módulo do Aprendiz**: possui a incumbência de colecionar as informações referentes à interação do aluno com o sistema, possibilitando a este último efetuar a modelagem cognitiva de cada aluno dinamicamente. Por meio do compartilhamento realizado com os outros módulos, e controlado pelo módulo de controle, torna-se possível um diagnóstico mais realista sobre a situação cognitiva do aluno e sua aprendizagem significativa.

- **Módulo Pedagógico**: responsável pela identificação e aplicação da estratégia pedagógica condizente com a situação cognitiva do aluno, sendo esta identificação baseada no diagnóstico efetuado pelo sistema através do compartilhamento estabelecido com seus outros modelos. Por meio destas estratégias pedagógicas o sistema consegue auxiliar o aluno sobre qual conteúdo este deve se dedicar mais e quais atividades realizar para uma aprendizagem significativa eficiente, além da própria promoção do aprendiz para um novo tópico no mesmo conteúdo ou de seu conteúdo sub-sequente.

- **Modelo da Monitoria**: acrescido à arquitetura do ITA com a finalidade de permitir a interação do monitor humano com o sistema, possibilitando a este importante agente real (monitor humano), sua colaboração no desafiador processo de modelar cada aluno em sua situação de aprendizagem significativa.

- **Módulo de Controle**: com o objetivo de organizar e controlar as interações e compartilhamentos necessários ao funcionamento de cada módulo deste ITA e, consequentemente, sua eficiência no apoio ao processo de ensino e de aprendizagem, com

diagnóstico coerente da situação cognitiva de cada aluno, acompanhamento mais realista sobre o esforço e os resultados obtidos, além do fornecimento de orientação pedagógica digna de um ensino significativo para uma aprendizagem associativa aos saberes do aprendiz, este módulo de controle desempenha suas atribuições nesta arquitetura.

- **Interfaces**: representa os canais de comunicação existentes entre o sistema e seus diferentes usuários, sendo adaptáveis de acordo com o perfil e responsabilidades de cada um destes no apoio ao ensino-aprendizagem almejado. As assistências oferecidas pelo ITA, para o acompanhamento da aprendizagem dos alunos, assim como nas percepções da inteligência do mesmo, no diagnóstico e orientação fornecidos pelo sistema, são acessados através deste módulo, sendo destacado o Módulo de Interface Animado que emprega o agente MinA na interação direta com o estudante, a fim de tornar o processo mais lúdico e de fácil compreensão sobre qual é a real situação de aprendizagem do próprio estudante que está interagindo com o SAE naquele momento.

2.2.4.2 Módulos do SAE envolvidos neste projeto

O SAE fornece informações pertinentes às diferentes necessidades dos perfis de usuário atendidos pelo sistema. Essas informações são fornecidas através dos recursos implementados no SAE, por meio dos seus diferentes módulos acionados pelo mecanismo de controle (SOUZA; SANTOS; RISSOLI, 2019).

Nas seções seguintes estão descritos alguns importantes módulos do SAE que visam atender às necessidades do aprendiz.

2.2.4.2.1 Banco de Questões (BDQ)

O módulo BDQ é um repositório de questões dividido por disciplinas e tópicos específicos. A função desse módulo é apoiar a relação aluno-professor, no que diz respeito ao acompanhamento do processo de aprendizagem (PEREIRA; AGUIAR; SARAIVA, 2007).

Esse módulo provê aos professores um ambiente para criar e manter questões que estejam coerentes com o plano de ensino e, a partir disso, consigam gerar listas de exercícios, sendo estas de fixação, revisão ou avaliativa, de acordo com a necessidade da turma.

Os alunos resolvem os exercícios propostos, obtendo *feedback* automático em relação ao seu progresso, a fim de aprender o conteúdo e construir uma base de conhecimento com os resultados obtidos. Com isso, o SAE consegue processar esses dados a fim de personalizar a aprendizagem para cada aluno.

De acordo com os autores, o módulo BDQ permite a criação de cinco tipos de questões, sendo estas:

1. **Preencher lacuna:** possibilita avaliação de uma amostra significativa do conteúdo em pouco tempo e avalia a capacidade de memorização e compreensão do conteúdo;

2. **Múltipla escolha:** permite a verificação de objetivos nos níveis de compreensão, interpretação, raciocínio dedutivo e indutivo, julgamento e aplicação, além de reduzir a possibilidade de erro casual;

3. **Verdadeiro ou falso:** tem o objetivo de abranger uma amostra significativa de conteúdo, visto que são questões rápidas e objetivas. Deve-se atentar a ambiguidade na criação das questões;

4. **Escolha múltipla:** avalia a interpretação e a capacidade do aluno associar mais de uma resposta correta a um determinado conteúdo;

5. **Questão aberta:** permite verificar a capacidade reflexiva do estudante, verificando se o aluno é capaz de organizar suas ideias e expressá-las claramente por escrito. Nesse tipo de questão não é possível ter correção automática, cabendo ao professor corrigi-las e disponibilizar a correção, com possíveis orientações, aos alunos.

2.2.4.2.2 Módulo de Interface Animado (MInA)

O sucesso no processo de ensino-aprendizagem está também relacionado aos estados afetivos dos alunos, que podem interferir na aptidão em aprender (RISSOLI; SANTOS, 2013). Assim, no projeto de um agente pedagógico animado, aspectos visuais e afetivos são relevantes no aprendizado do estudante.

MInA é o nome do agente, apresentado ao usuário como um personagem animado, que passou a atuar no módulo de interfaces do SAE. Este agente é representado por Mangá, uma técnica de quadrinhos japoneses, que é um poderoso recurso para transmissão de ideias e sentimentos.

Segundo Rissoli e Santos (2013), as principais diretrizes para o projeto da MInA foram:

1. Projetar uma entidade feminina, que confere maior receptividade aos usuários, principalmente pelos aspectos afetivos relacionados a esse gênero;

2. Atribuir ao agente expressões faciais e corporais variadas, compartilhando emoções como alegria, tristeza, susto e outras;

3. Vestir o agente com roupas e adereços típicos de um ambiente de trabalho significativo ao contexto de estudo e trabalho de seus usuários para colaborar na percepção da necessidade de ação, esforço e proatividade;

4. Demonstrar uma fisionomia limpa, jovem e alegre;

5. Utilizar cores quentes em sua composição, que priorizam motivação e disposição;

6. Interagir com usuário por meio de gestos, balões e sons, para tentar garantir que a experiência do usuário não se torne monótona;

7. Utilizar frases e metáforas que informem ao aluno sua situação de aprendizagem, de forma realista e motivadora, posicionando o agente como assistente e companheiro nos desafios educacionais.

Algumas ilustrações desse agente (MinA) podem ser observadas na Figura 9.

Figura 9 – Modelo gráfico do agente MInA e a indicação de algumas animações. Fonte: (RISSOLI; SANTOS, 2013).

A MInA se integra ao SAE como mais um módulo que compõe sua arquitetura, sendo este capaz de trabalhar aspectos emotivos no reforço da compreensão do próprio

aprendiz sobre a sua própria situação de aprendizagem inferida pelo SAE (SOARES; RISSOLI, 2011).

Com o acompanhamento de cada estudante, por meio da Lógica Fuzzy, o SAE fornece orientações pedagógicas a cada estudante. Estas orientações são apresentadas pelo agente MInA, que almeja realizar uma interação mais emotiva e significativa à compreensão de seus aprendizes sobre a sua atual situação de aprendizagem. Esse agente ainda mostra a seus aprendizes um caminho de estudo que possa ser mais motivador e coerente para superar possíveis dificuldades de aprendizagem (RISSOLI; SANTOS, 2011).

2.2.4.2.3 Módulo do Sistema Inteligente de Avaliação do Conhecimento (SIAC)

O Sistema Inteligente de Avaliação do Conhecimento (SIAC) tem como objetivo informatizar exames ou testes adaptativos à realidade cognitiva de cada estudante, empregando Teoria de Resposta ao Item (TRI) como recurso para mensurar a habilidade de cada aprendiz e lhe fornecer questões coerentes com seu nível de conhecimento (FERREIRA et al., 2011).

Segundo Gonçalves (2004), na década de 50 deu-se início a uma nova forma de avaliar alunos mediante testes, caracterizados por selecionar questões para o aluno, segundo os níveis estimados de habilidade de cada aprendiz.

Os Testes Adaptativos (TA) têm como característica individualizar os testes. Assim, as questões são selecionadas para cada aluno de acordo com sua habilidade, interpretada como grau de conhecimento do aluno sobre o domínio. Os Testes Adaptativos Informatizados (TAI) automatizam o processo do TA, que antes era feito por lápis e papel.

O SIAC permite aos estudantes realizarem testes nos quais os itens (questões) são selecionados de acordo com sua habilidade, sendo essa estimativa calculada por meio de fórmulas matemáticas da TRI que emprega a lógica probabilística no SAE com o Modelo Logístico de Quatro Parâmetros (ML4P) (TEIXEIRA; JúNIOR; RHODEN, 2015).

Segundo Andrade (2001), a TRI baseia-se em modelos que representam a probabilidade de um indivíduo acertar um item em função dos parâmetros deste e da habilidade do respondente.

A TRI surgiu na década de 50, por meio de discussões sobre a possibilidade de comparação das habilidades dos indivíduos examinados, que foram submetidos a testes diferentes (TEIXEIRA; JúNIOR; RHODEN, 2015).

A habilidade pode ser pensada como o quão bom um estudante é em determinado conteúdo, sendo esta habilidade um traço latente, variável psicológica que não pode ser medida diretamente, por exemplo, como acontece com o peso e a altura de uma pessoa

(TEIXEIRA; JúNIOR; RHODEN, 2015).

De acordo com Andrade (2001), os modelos de TRI dependem, fundamentalmente, de três fatores:

1. **Natureza do item**: dicotômicos e não dicotômicos;
2. **Número de populações envolvidas**: apenas uma ou mais do que uma;
3. **Quantidade de traços latentes que está sendo medida**: apenas um ou mais que um.

Como apontado por Gonçalves (2004), existem quatro principais modelos logístico de resposta, sendo eles:

- Modelo Logístico de Um Parâmetro;
- Modelo Logístico de Dois Parâmetros;
- Modelo Logístico de Três Parâmetros (ML3P);
- Modelo Logístico de Quatro Parâmetros (ML4P).

O ML3P é o modelo mais difundido e usado no Brasil, sendo implementado no SAE o ML4P que propicia análises envolvendo o quatro parâmetro relativo ao tempo, que possui potencial relevante na assistência fornecida aos docentes, principalmente, além do próprio módulo de controle do SAE.

Como apresentado por Gonçalves (2004) e Teixeira, Júnior e Rhoden (2015), o ML4P incorpora um parâmetro relacionado ao tempo no ML3P, visto que em situações de testes sempre existe um limite de tempo. O tempo está relacionado com outros três valores necessários ao modelo:

- **d**: lentidão do item;
- **p**: lentidão do indivíduo;
- **t**: tempo total de duração do teste.

Os demais parâmetros seguem o ML3P, sendo eles:

- **P(θ):** é a probabilidade de um indivíduo com habilidade θ de acertar o item;
- θ é o nível de habilidade do aluno;
- **a:** é o parâmetro de discriminação do item;

- **b:** é o parâmetro de dificuldade do item;
- **c:** representa a probabilidade de um aluno com baixa habilidade responder corretamente o item, ou seja, a probabilidade de acerto casual (chute);
- **e**: é número neperiano, constante matemática cujo valor é 2,718.

[1]

Esse modelo é representado pela equação 2.1.

$$P(\theta) = c + \frac{1-c}{1+e^{-1,7a(\theta-(\frac{pd}{t})-b))}} \tag{2.1}$$

Por todo o exposto, concluiu-se pela viabilidade de atuação deste projeto no contexto de STIs, agregando ao SAE a plataforma nativa do sistema operacional iOS, a fim de fornecer aos estudantes a assistência oferecida por este ITA, por meio dos módulos BQD, Orientação Pedagógica, MInA e SIAC.

[1] θ: Letra grega theta

3 Metodologia

O presente capítulo visa abordar detalhadamente a metodologia de pesquisa utilizado no trabalho.

3.1 Metodologia de Pesquisa

3.1.1 Classificação da Pesquisa

Como descrito na seção 1.5, a presente pesquisa é classificada quanto à sua natureza, abordagem, objetivos e procedimentos. Cada uma destas categorias estão descritas nas seções a seguir.

3.1.2 Natureza da Pesquisa

A pesquisa aplicada (tecnológica) consiste na utilização do conhecimento da pesquisa básica e da tecnologia para se obter aplicações práticas, como produtos ou processos (SOUSA, 2018).

Dado o contexto prático deste trabalho, cujo objetivo é o desenvolvimento de uma interface de comunicação do SAE para o sistema operacional iOS, esta pesquisa é de natureza aplicada de desenvolvimento de produção tecnológica.

3.1.3 Abordagem da Pesquisa

A abordagem deste trabalho pode ser definida como qualitativa. Segundo MORESI (2003), na pesquisa qualitativa considera-se que existe um vínculo indissociável entre o mundo objetivo e a subjetividade do sujeito que não pode ser traduzido em números. Ademais, a interpretação de fenômenos é um processo básico na pesquisa qualitativa e nela os pesquisadores tendem a analisar seus dados indutivamente.

Neste trabalho buscou-se entender o uso de tecnologias dos STIs na área da Educação, por meio da plataforma SAE, viabilizando a adaptação dessa tecnologia para o contexto dos recursos móveis, como os *smartphones*. Além disso, os dados coletados foram analisados de maneira não quantificável, buscando apontar os dados como satisfatórios ou não, por exemplo.

3.1.4 Objetivos da Pesquisa

A pesquisa descritiva objetiva descrever os fatos e fenômenos de determinada realidade (GERHARDT; SILVEIRA, 2009).

Quanto aos objetivos, o presente trabalho é classificado como descritivo, visto que descreve o desenvolvimento do aplicativo proposto em um outro ambiente operacional (iOS).

3.1.5 Procedimentos da Pesquisa

Como apontado por MORESI (2003), os tipos de pesquisa não são mutuamente exclusivos. Assim, quanto aos procedimentos, essa pesquisa é definida como pesquisa bibliográfica e produção tecnológica.

3.1.5.1 Pesquisa Bibliográfica

Segundo MORESI (2003), *"Pesquisa bibliográfica é o estudo sistematizado desenvolvido com base em material publicado em livros, revistas, jornais, redes eletrônicas, isto é, material acessível ao público em geral. Fornece instrumental analítico para qualquer outro tipo de pesquisa, mas também pode esgotar-se em si mesma."*. Tendo isso em vista, a pesquisa bibliográfica foi utilizada para conhecer o que já foi estudado sobre o assunto da pesquisa.

As informações no Capítulo 2 sobre os temas Aplicativo Móveis e Sistema Tutor Inteligente são frutos da pesquisa bibliográfica. Neste trabalho, foram realizadas pesquisas nas bases de dados científicas da CAPES, Scopus, SciELO e IEEE Xplore.

3.1.5.2 Produção Tecnológica

De acordo com Serzedello e Tomaél (2011), a produção tecnológica é definida por uma comunidade científica e pela geração de produtos e processos tecnológicos com o objetivo de contribuir na solução de problemas práticos. Geralmente, ela tem como finalidade atender demandas da sociedade, por meio da criação e desenvolvimento de invenções, que, por consequência, impactam o desenvolvimento tecnológico, econômico e social.

Inicialmente, o conceito de produção tecnológica foi utilizado para produção de produtos e processos tecnológicos gerados a partir de inspirações, sem embasamento ou acúmulo de conhecimento (SERZEDELLO; TOMAéL, 2011). Entretanto, em seu trabalho, os autores demonstram que foi confirmada a necessidade de conhecimento prévio e acúmulo de conhecimento para desenvolver os produtos e processos tecnológicos.

Existe grande variedade das formas de produção tecnológica e suas classificações, como as exemplificadas na Tabela 3.

Tipos de Produção Tecnológica	
Modalidade	**Especificação**
Softwares	Abrange os sistemas computacionais
Produtos	Projetos, protótipos, pilotos entre outros.
Cartas mapas ou similares	Produtos cartográficos desenvolvidos para tecnologias aplicáveis.
Relatório de pesquisa	Relatórios periódicos ou finais que descrevem as pesquisas que contenham fins tecnológicos.
Patente	Fonte de informação tecnológica. Criações, invenções no âmbito da tecnologia.
Manuais	Manuais que documentem e descrevam o produto.

Tabela 3 – Tipos de produção tecnológica. Fonte: Adaptado de (SERZEDELLO; TOMAéL, 2011).

Esta pesquisa é classificada como produção tecnológica, visto que objetiva a construção de um aplicativo (software) para resolução de um problema prático.

Primeiramente, foi elaborada uma pesquisa bibliográfica, para obter conhecimento prévio da solução que será desenvolvida. Em seguida, foi iniciada a etapa de produção tecnológica.

3.2 Processo Metodológico

Segundo as normas atuais da Universidade de Brasília (FAC, 2020), o TCC é realizado em duas fases, denominadas: TCC1 e TCC2. Durante o TCC1, o foco deste trabalho foi desenvolver os pilares teóricos para embasamento do projeto, bem como, elaborar uma proposta coerente e viável. Durante o TCC2 o foco do trabalho foi atingir o objetivo geral proposto na seção 1.4.1, conforme planejado no TCC1, com a realização e implementação da proposta.

Com o objetivo de definir e acompanhar o processo de desenvolvimento do TCC todo (TCC1 e TCC2), foi modelado um processo metodológico, o qual se encontra na Figura 10. Além disso, na Tabela 4 e Tabela 5 estão os cronogramas das atividades definidas no processo metodológico para o TCC1 e TCC2, respectivamente.

A seguir será descrito o que cada atividade do processo representa:

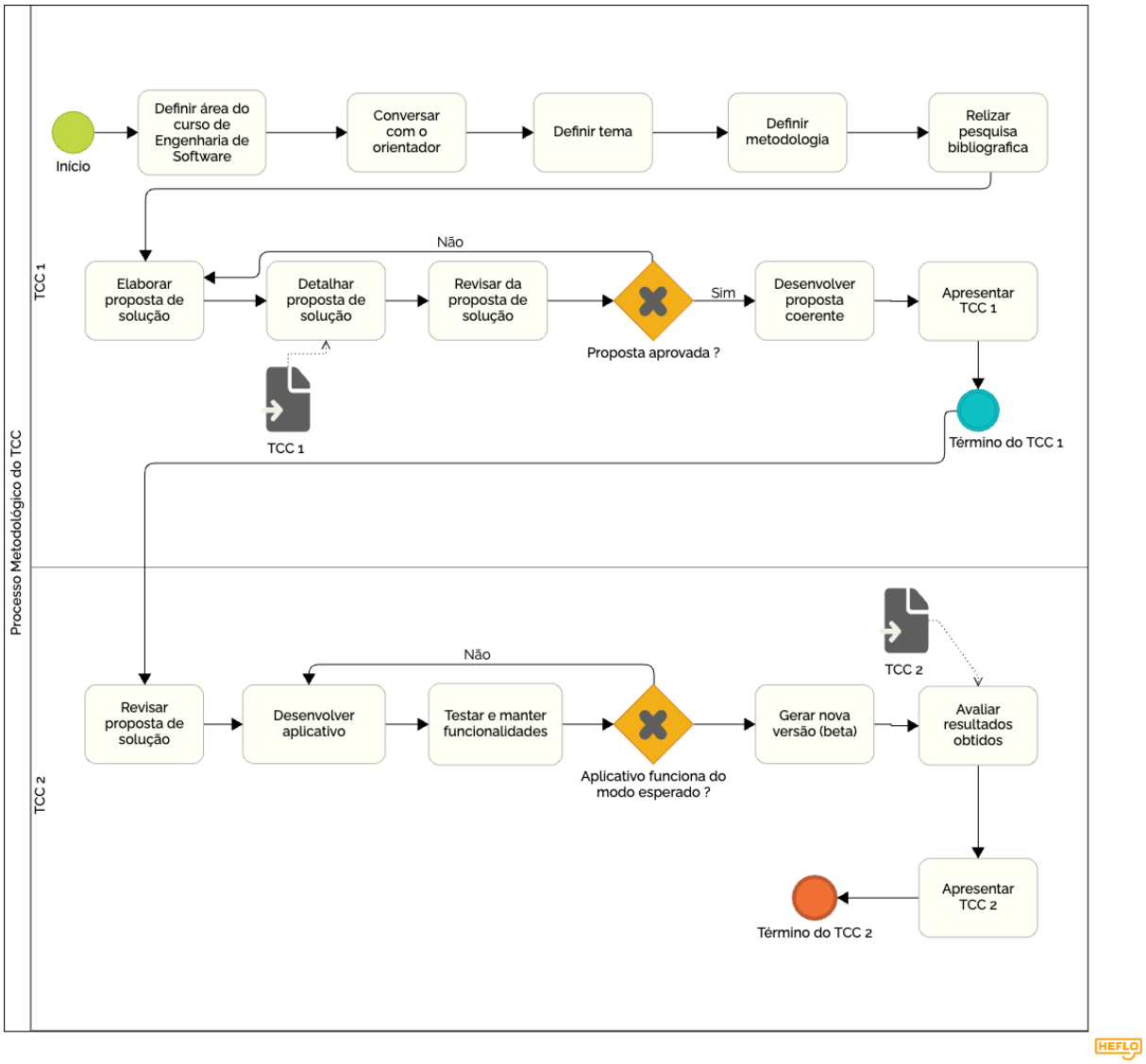

Figura 10 – Processo Metodológico do TCC completo. Fonte: Autor.

3.2.1 TCC1

- **Definir área do curso de Engenharia de Software:** escolher uma área do curso de Engenharia de Software que o estudante tenha interesse para pensar em possíveis temas de projeto;
- **Conversar com o orientador:** contactar o orientador para verificar a disponibilidade e interesse deste em orientar o TCC;
- **Definir tema:** definir um tema, juntamente com o orientador, relacionado à área do curso definida na primeira atividade do processo;
- **Definir metodologia:** definição da metodologia utilizada para guiar a pesquisa e o desenvolvimento. A partir da metodologia foi possível definir o processo do desenvolvimento do TCC.

- **Realizar pesquisa bibliográfica:** a pesquisa de autores, artigos e publicações a respeito dos temas relevantes para a problemática, com o objetivo de ampliar os conhecimentos em relação aos temas, conhecendo pesquisas em diferentes contextos e de diversas bases científicas, como CAPES, IEEE Xplore, SciELO e Scopus. O resultado parcial dessa atividade se encontra descrita no Capítulo 2;

- **Elaborar proposta de solução:** a partir de conversas com o orientador e com o conhecimento adquirido sobre os temas, elaborou-se a proposta de desenvolvimento de um aplicativo do SAE para o sistema operacional iOS;

- **Detalhar proposta de solução:** essa atividade consistiu em detalhar a proposta de solução descrevendo a arquitetura, os requisitos e outras informações pertinentes ao aplicativo que será construído, bem como as tecnologias selecionadas para apoio ao desenvolvimento. Ao fim desta atividade, obteve-se o TCC 1 como artefato.

- **Revisar proposta de solução:** apresentação do artefato concebido na atividade anterior para o orientador, que o analisa e indica ajustes evolutivos e corretivos a serem realizados. Neste caso, volta-se à etapa de elaboração de proposta de solução. No caso de aprovação do orientador, o processo continua;

- **Desenvolver proposta coerente:** implementação de uma proposta coerente que demonstre a viabilidade da arquitetura e tecnologias envolvidas para solucionar a questão de pesquisa apresentada no TCC 1, validando as metodologias indicadas em relação aos objetivos e o tempo planejado;

- **Apresentar TCC1:** apresentar os resultados obtidos até o momento para a banca examinadora.

Mês / **Atividade**	**Ago**	**Set**	**Out**	**Nov**	**Dez**
Definir área do curso de Engenharia de Software	X				
Conversar com o orientador	X				
Definir Tema	X				
Definir metodologia	X				
Realizar pesquisa bibliográfica	X	X	X		
Elaborar proposta de solução			X		
Revisar proposta de solução			X	X	
Desenvolver proposta coerente				X	X
Apresentar TCC1 para banca					X

Tabela 4 – Cronograma do TCC1. Fonte: Autor.

3.2.2 TCC2

- **Revisar proposta de solução:** correções provenientes dos apontamentos levantados após a atividade de apresentação do TCC1 para banca, gerando uma versão coerente com a proposta aprovada pela banca como solução;
- **Desenvolver aplicativo:** realização das atividades definidas no processo de desenvolvimento do TCC, conforme a Figura 15. Ao fim desta atividade se obterá como resultado um software funcional (aplicativo);
- **Testar e manter funcionalidades:** Teste e manutenção das funcionalidades implementadas na etapa anterior, para garantir que estas funcionem corretamente e do modo esperado;
- **Gerar nova versão:** Gerar uma versão *beta* da aplicação, em ambiente de homologação, para permitir que essa possa ser utilizada;
- **Avaliar resultados obtidos:** avaliação do software desenvolvido e descrição dos resultados alcançados. Ao final desta etapa é obtido o TCC2 como artefato;
- **Apresentar TCC2:** apresentar os resultados finais para a banca examinadora.

Mês / **Atividade**	**Fev**	**Mar**	**Abr**	**Mai**	**Jun**
Revisar proposta de solução	X				
Desenvolver aplicativo	X	X	X		
Testar e manter funcionalidades			X	X	
Gerar nova versão				X	
Avaliar resultados obtidos				X	
Apresentar TCC2					X

Tabela 5 – Cronograma do TCC2. Fonte: Autor.

4 Proposta

O presente capítulo visa apresentar a proposta inicial do aplicativo desenvolvido, abordando detalhadamente a metodologia de desenvolvimento do software, seus requisitos, arquitetura e ferramentas de desenvolvimento utilizadas.

4.1 Metodologia de Desenvolvimento

4.1.1 Metodologia Ágil

O desenvolvimento ágil de software é uma abordagem emergente de engenharia de software que consiste em um conjunto de princípios, que orientam a natureza genérica de todas as diferentes metodologias ágeis que são propostas, inicialmente defendido por poucos grupos de profissionais da área, mas agora praticado por muitos profissionais de software (MISRA et al., 2012).

Um grupo de 17 profissionais de software se reuniram em Snowbird, Utah, nos Estados Unidos em 2001 para racionalizar sua filosofia comum e a de desenvolvimento de software como "ágil". Além disso, desenvolveram o Manifesto para Desenvolvimento Ágil de Software, que afirma os valores e princípios desta filosofia (MISRA et al., 2012). O manifesto ágil tem os seguintes valores:

- Indivíduos e interações, ao invés de processos e ferramentas;
- Software funcionando, ao invés de documentação extensiva;
- Colaboração com cliente, ao invés de negociação de contratos;
- Responder às mudanças, ao invés de seguir um plano.

Para a fase de desenvolvimento do aplicativo proposto neste trabalho optou-se pela utilização de práticas e ferramentas baseadas em duas metodologias de desenvolvimento, *Scrum* e *Kanban*, aliadas aos aspectos principais da metodologia ágil.

4.1.1.1 Scrum

O *Scrum* é uma metodologia ágil que se concentra no gerenciamento de projetos de software que tem atraído significativa atenção dos profissionais de software nos últimos anos (MAHNIC; DRNOVSCEK, 2005).

Esta metodologia parte da premissa que o desenvolvimento de software é muito complexo e imprevisível para ser completamente planejado com antecedência. Assim, um

processo de controle mais empírico deve ser aplicado, objetivando garantir visibilidade, inspeção e adaptação.

De acordo com Mahnic e Drnovscek (2005), as diferentes variáveis envolvidas no projeto de software como, por exemplo, prazo, qualidade, requisitos, recursos, tecnologias e ferramentas de implementação e até métodos de desenvolvimento, devem ser constantemente controladas para poder se adaptar às mudanças com flexibilidade. Isso é alcançado por meio de um processo de desenvolvimento iterativo e incremental. Uma visão geral do *Scrum* pode ser vista na Figura 11.

Figura 11 – Visão geral do *Scrum*. Fonte: Adaptado de (SCHWABER, 2004).

Segundo Sommerville (2011), no *Scrum* existem três fases bem definidas, sendo elas:

1. **Planejamento Geral:** objetivos gerais e arquitetura do software são estabelecidos. O ponto de partida para o planejamento é o *backlog* do produto, que é a lista do trabalho a ser feito no projeto;

2. ***Sprints***: ocorre uma série de ciclos de *sprint*, em que cada ciclo desenvolve um novo incremento do sistema. *Sprints* ocorrem com um comprimento fixo, normalmente de duas a quatro semanas, e são divididos nas quatro etapas a seguir:

    a) **Avaliar:** a *sprint* é revisada e as prioridades e os riscos são identificados. O cliente está diretamente ligado a esta fase, podendo introduzir novos requisitos e tarefas;

    b) **Selecionar:** Envolve todos os membros da equipe do projeto, que trabalham juntamente com o cliente para selecionar os recursos e as funcionalidades que serão desenvolvidas durante a *sprint*;

    c) **Desenvolver:** a equipe se organiza para desenvolver o software. Ocorrem reuniões diárias rápidas para o acompanhamento do progresso;

d) **Revisar:** o trabalho é revisto e apresentado aos *stakeholders*. O próximo ciclo *sprint* começa em seguida.

3. **Encerramento do Projeto**: finaliza-se a documentação exigida e as lições aprendidas com o projeto e estas são avaliadas.

O processo do *Scrum* pode ser observado na Figura 12.

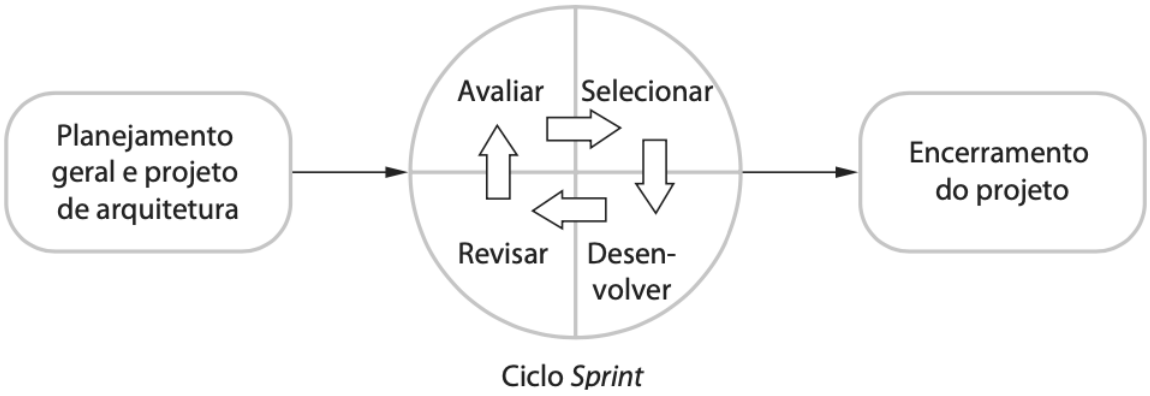

Figura 12 – O Processo do *Scrum*. Fonte: (SOMMERVILLE, 2011).

No *Scrum* existem três papéis principais (SCHWABER, 2004), sendo estes:

1. ***Product Owner***: representa os interesses dos *stakeholders* no projeto. Também é responsável por manter e priorizar o *Product Backlog*, que é a lista de requisitos e funcionalidades que devem ser desenvolvidas;

2. ***Scrum Master***: responsável pelo processo do *Scrum*, ensiná-lo para os envolvidos no projeto, implementá-lo de modo que se encaixe na cultura da organização, garantindo que entregue os benefícios esperados e que todos sigam as práticas do *Scrum*;

3. **Time**: as equipes devem ser autogerenciadas, auto-organizadas e multifuncionais, e são os responsáveis por transformar o *Product Backlog* em um incremento de funcionalidade em uma iteração. São coletivamente responsáveis pelo sucesso de cada iteração e do projeto como um todo.

Para atender às necessidades e restrições do projeto, as atividades e papéis do *Scrum* foram adaptados. Dentre as práticas utilizadas estão *sprints* de quinze dias, o *backlog* do produto, o *backlog* da *sprint*, revisão e retrospectiva da *sprint*.

O principal motivo para as alterações nos papéis ocorreu devido à quantidade de pessoas envolvidas no projeto. O papel de Time foi desempenhado exclusivamente pelo autor deste trabalho, pois é ele quem tem o conhecimento técnico para o desenvolvimento

do projeto. O papel de *Scrum Master* foi desempenhado pelo professor orientador, por ser quem está orientando o desenvolvimento do trabalho, além de ser um dos idealizadores do SAE. O papel de *Product Owner* foi feito de maneira conjunta pelo autor e o professor orientador. O professor atuou priorizando o *Product Backlog*, sendo ele, um dos principais interessados no projeto. O autor atuou mantendo o *Product Backlog*.

4.1.1.2 *Kanban*

Kanban é uma palavra da língua japonesa que significa "cartão de sinal"e uma de suas características é evidenciar problemas existentes no processo (ARRUDA, 2012). Este método surgiu no Japão com o sistema de produção da Toyota que procurava controlar a fabricação de automóveis de forma coerente com a demanda que definia o ritmo de produção. Assim, a indústria adaptava sua velocidade de produção de acordo com o nível de consumo dos clientes (SILVA; SANTOS; NETO, 2012).

Na área de desenvolvimento de software, o *Kanban* registrou aumento significativo desde 2007 e, desde então, o uso desta metodologia vem sendo estudada e experimentada por equipes de software (SILVA; SANTOS; NETO, 2012).

De acordo com Genari e Ferrari (2015), as principais características do *Kanban* são:

- Visualização do fluxo de trabalho;
- Limitação do fluxo trabalho;
- Gerenciamento e acompanhamento do fluxo de trabalho.

Dentre as metodologias para desenvolvimento de software, o *Kanban* é a menos prescritiva, portanto altamente adaptativa. Por isso que, ao ser adotada, deve-se estar atento ao processo a fim de que se consiga visualizar locais de melhoria e adaptações, de modo que o processo possa fluir de forma satisfatória (SILVA; SANTOS; NETO, 2012).

Para fornecer a percepção visual de continuidade do fluxo de trabalho, o método *Kanban* utiliza o quadro que leva seu nome. Trata-se de uma ferramenta que as equipes usam para visualizar seu fluxo de trabalho (STELLMAN; GREENE, 2014).

Normalmente, o quadro *kanban* é elaborado com três colunas que indicam a situação das ações planejadas: **por fazer** (*To Do*), **em processo** (*Doing*) e **feito** (*Done*). No entanto, estas colunas nos painéis geralmente variam de equipe para equipe (STELLMAN; GREENE, 2014). A Figura 13 mostra um esquema do quadro mencionado.

No presente trabalho, a metodologia *Kanban* será aplicada com apoio da ferramenta *Trello*, descrita na seção 4.2.3, e será mais utilizada durante o processo de desenvolvimento ao longo do TCC2.

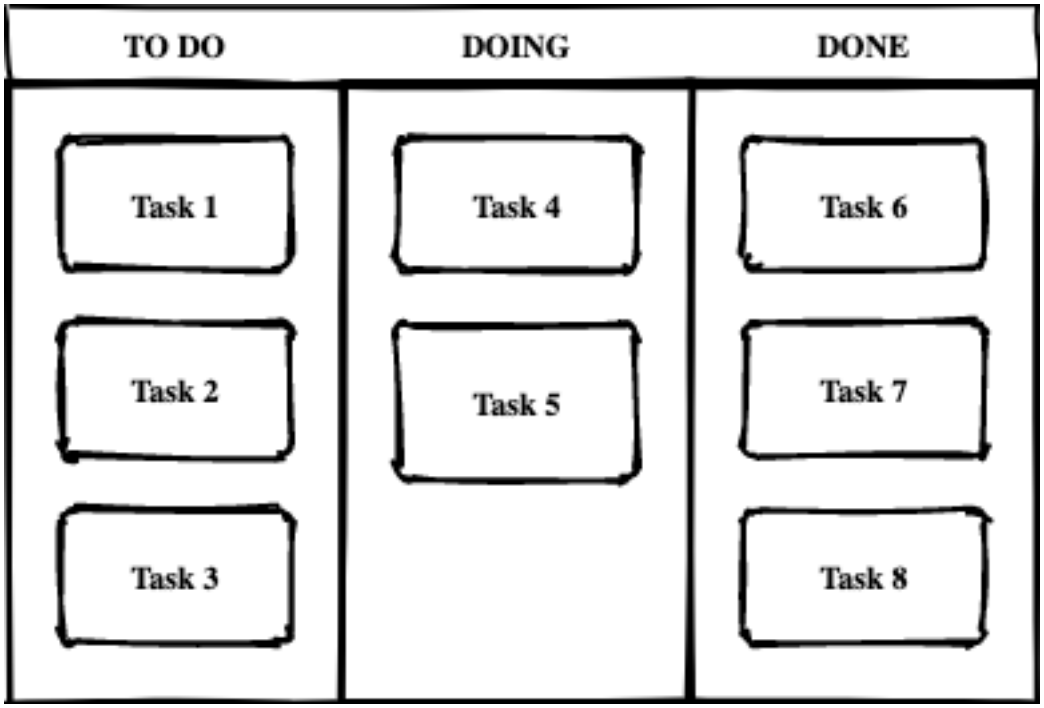

Figura 13 – Esquema de um quadro *Kanban*. Fonte: Autor.

O quadro foi construído separando as etapas do desenvolvimento que sejam mais relevantes para o projeto, sendo elas:

- ***Product Backlog:*** todas as funcionalidades que deverão ser realizadas ao longo do projeto;
- ***Sprint Backlog:*** referente às histórias à serem feitas ao longo da *sprint* em andamento;
- ***Doing:*** histórias que estão sendo desenvolvidas;
- ***In Review:*** refere-se às histórias que foram desenvolvidas, mas que estão em fase de revisão;
- ***Done:*** histórias concluídas.

A Figura 14 apresenta o modelo do quadro *Kanban* que também foi utilizado pelo projeto, com algumas poucas adaptações.

4.1.1.3 Fluxo de Desenvolvimento

O termo *feature* é uma adaptação dentro do *Scrum* e define uma funcionalidade que deve ser implementada no software e é composta por histórias de usuários que narram o que o usuário deseja realizar e a motivação da ação.

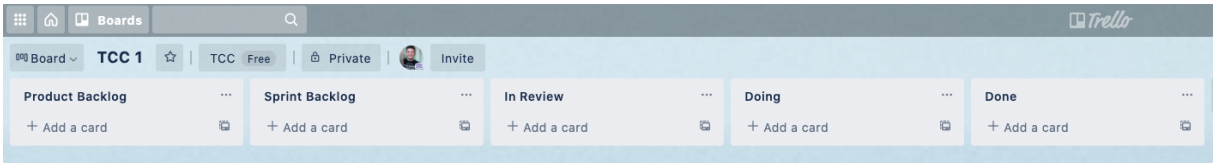

Figura 14 – Modelo do quadro *Kanban* utilizado para o desenvolvimento do software. Fonte: Trello.

A Figura 15 representa como as práticas das metodologias foram adaptadas para o contexto do projeto.

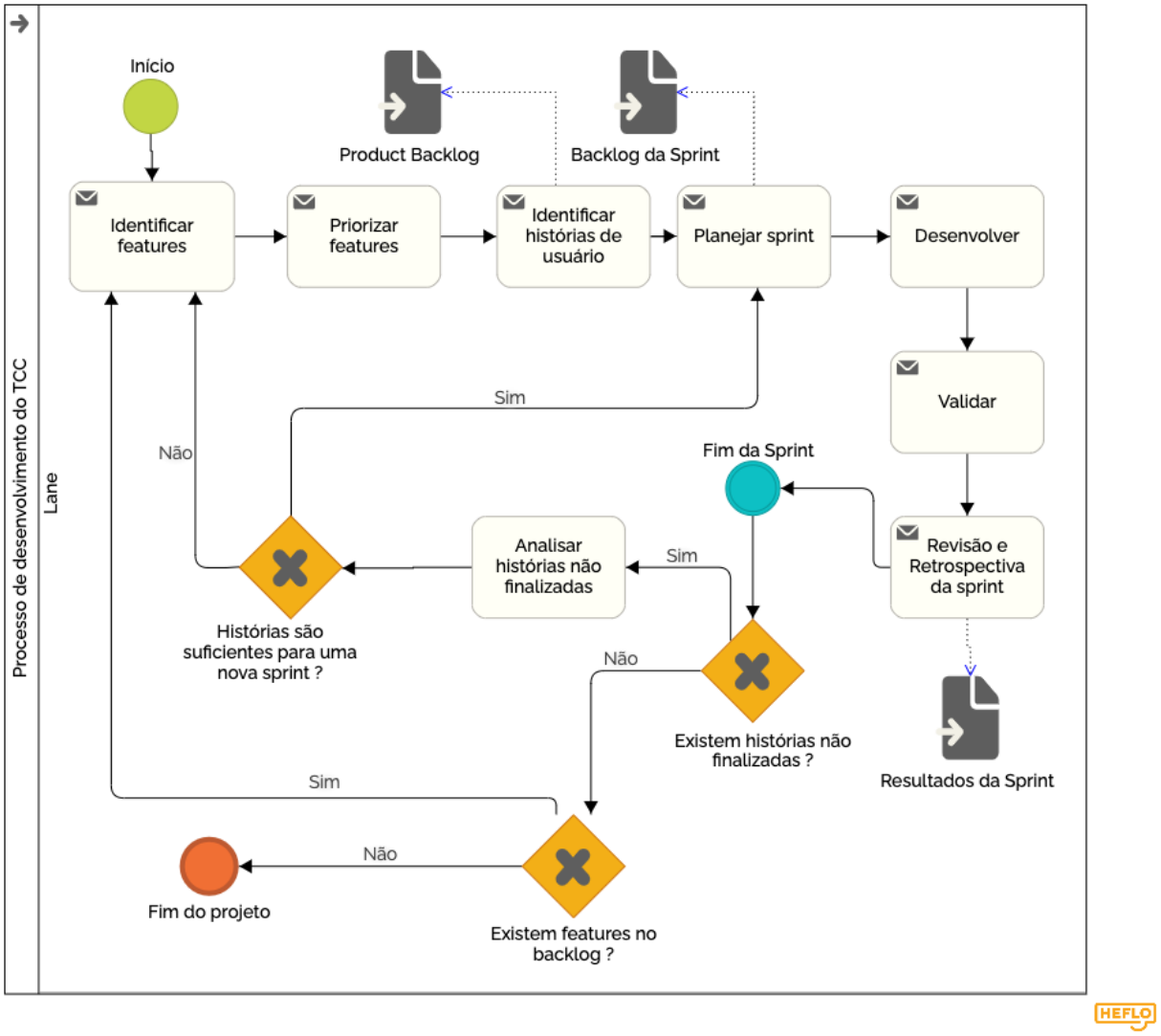

Figura 15 – Processo de Desenvolvimento do TCC. Fonte: Autor.

A seguir será descrito o que cada atividade do processo representa:

- **Identificar *features*:** realizar o levantamento dos requisitos das funcionalidades que foram implementadas;

- **Priorizar *feature*:** esta atividade consiste em definir a ordem que as *features* identificadas foram implementadas;
- **Identificar histórias de usuário:** descrever as funcionalidades de forma detalhada, de modo a facilitar o entendimento. Além disso, foi utilizado a voz ativa de usuário como recurso para descrever a funcionalidade, como pode ser observado na seção 4.1.1.4;
- **Planejar *sprint:*** escolher quais serão as histórias de usuário que foram desenvolvidas ao longo da *sprint*;
- **Desenvolver:** esta atividade acontece quando o código fonte da história de usuário é de fato desenvolvido;
- **Validar:** realização de testes para assegurar que o comportamento da funcionalidade está conforme o esperado;
- **Revisão e Retrospectiva da *sprint:*** revisar quais histórias de usuário foram desenvolvidas no decorrer da *sprint* e analisar os seus pontos positivos e negativos;
- **Analisar histórias pendentes:** consiste em verificar se existem histórias já levantadas ou se existe a necessidade de recomeçar o ciclo de requisitos para a elicitação de mais histórias.

4.1.1.4 Requisitos

Os requisitos do aplicativo foram elicitados juntamente com o orientador, que tem o conhecimento necessário sobre o contexto e sobre as regras de negócios do SAE, que serão desenvolvidas no aplicativo. As técnicas de elicitação de requisitos utilizadas para compreender melhor o problema e definir a solução foram *brainstorm* e entrevistas entre o autor e o seu orientador.

Para um levantamento inicial do projeto foram identificadas as seguintes *features* e histórias de usuário.

Features:

- **Autenticação - F01:** essa *feature* permitiu ao usuário fazer *login* e *logout* no aplicativo;
- **Questões - F02:** permitir aos alunos realizar questões das disciplinas que estão cursando. Referente ao módulo BDQ;
- **Listas - F03:** semelhante a anterior, essa *feature* é refente ao aluno realizar as listas de questões que foram criadas pelo professor da disciplina. Referente ao módulo BDQ;

- **Testes - F04:** *feature* responsável pela realização de testes. Referente ao módulo SIAC;

- **Situação - F05:** permitir ao aluno solicitar a sua atual situação na disciplina que está cursando. Além disso, o estudante conseguirá visualizar gráficos que mostram seu desempenho nas questões realizadas;

- **Orientação - F06:** responsável por mostrar uma orientação ao aluno que o permite saber quais são as ações recomendadas para tornar sua aprendizagem mais eficiente e melhorar seu desempenho.

Histórias de usuário:

- **F01 - US01:** Eu, como aluno, gostaria de realizar *login* no aplicativo para conseguir utilizar suas funcionalidades;

- **F01 - US02:** Eu, como aluno, gostaria de realizar *logout* no aplicativo para que minhas informações não fiquem salvas;

- **F02 - US03:** Eu, como aluno, gostaria de resolver questões sobre as disciplinas que estou cursando para verificar o meu aprendizado;

- **F02 - US04:** Eu, como aluno, gostaria de filtrar questões de acordo com conteúdo, dificuldade e tipo de questão para realizar questões que melhor se adequa ao meu nível de aprendizado;

- **F03 - US05:** Eu, como aluno, gostaria de realizar listas de questões sobre as disciplinas que estou cursando para verificar o meu aprendizado;

- **F04 - US06:** Eu, como aluno, gostaria de realizar teste sobre os conteúdos das disciplinas que estou cursando para verificar o meu aprendizado;

- **F04 - US07:** Eu, como aluno, gostaria de visualizar os resultados dos testes que realizei para verificar meu desempenho;

- **F05 - US08:** Eu, como aluno, gostaria de verificar minha situação na disciplina, de maneira gráfica, para verificar meu desempenho nos conteúdos e progressão na aprendizagem;

- **F06 - US09:** Eu, como aluno, gostaria de solicitar orientação para encontrar o caminho mais adequado para melhorar meu desempenho na disciplina cursada.

4.1.1.5 Arquitetura

O sucesso do desenvolvimento de um aplicativo não depende apenas das habilidades do programador, mas também dos padrões arquiteturais selecionados. Assim, a seleção do padrão de arquitetura é crucial em uma fase de projeto de software (SHOLICHIN et al., 2005).

O padrão arquitetural *Model-View-ViewModel* (MVVM) foi anunciado pela *Microsoft* em 2005 e tem como proposta a separação entre a lógica da camada *view* da sua aparência (ANDRADE, 2001) e a lógica de negócio agora contemplada pela camada *ViewModel*.

Andrade (2001) define a cada camada as responsabilidades assim:

- **View**: corresponde à camada de apresentação;
- **Model:** é a camada de dados;
- **ViewModel:** corresponde à camada de lógica de negócios.

Como pode ser observado na Figura 16, as camadas *view* e *controller* se tornam um único domínio.

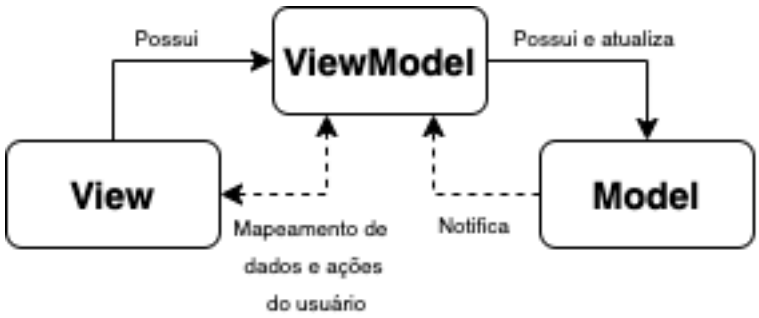

Figura 16 – Padrão MVVM. Fonte: Autor.

A *controller* não mantém mais referência à camada *model*, em vez disso, se comunicam por meio da camada ViewModel. Ter a ViewModel como intermediária traz os seguintes benefícios em termos de testabilidade e escalabilidade (NGUYEN, 2017):

- A *View* contém referência à *ViewModel*, mas esta desconhece a *View*. Desta forma, a *ViewModel* pode ser amplamente reutilizada e adotada por diferentes *Views*;
- A *ViewModel* é capaz de observar mudanças na *Model* e sincronizar-se com esses eventos e uma vez que existe uma ligação entre a *View* e a *ViewModel*, o primeiro é atualizado de acordo. Assim, obtém-se uma experiência responsiva para os usuários;

- No MVVM a *ViewModel* é uma camada independente da interface de usuário. Assim, a camada *Model* permanece isolada de efeitos colaterais da interface de usuário. A execução de testes unitários na *ViewModel* é, portanto, mais fácil.

Por esses motivos, foi adotado o padrão arquitetural MVVM para o desenvolvimento do aplicativo proposto neste trabalho.

4.2 Suporte Tecnológico

4.2.1 Diagrams.net

O Diagrams.net é o software de diagramação baseado em navegador mais usado do mundo (DIAGRAMS.NET, 2020) e funciona em conjunto com ferramentas como Google Drive e Github.

Por contemplar diversos tipos de diagramas e ser utilizado no navegador, esta ferramenta foi empregada para ilustrar as figuras da presente pesquisa.

4.2.2 Git e Github

O Git é um sistema de versionamento *open-source* distribuído, que possui objetivo de lidar com projetos pequenos até projetos grandes com velocidade e eficiência (GIT, 2020).

A escolha desta ferramenta se deu em razão da utilização do Git como ferramenta de controle de versão do código da aplicação desenvolvida e foi utilizado o ambiente do Github para hospedagem do código. Além disso, o Github ainda permite gerenciar de modo remoto e descentralizado repositórios em que o código está hospedado e possibilita a automação de testes e integração e implantação contínua.

4.2.3 Trello

O Trello é uma ferramenta utilizada para gerenciar projetos e atividades por meio de cartões, baseado na metodologia *Kanban* (TRELLO, 2020). Nele é possível a criação de atividades e cartões personalizados, conforme a demanda do projeto e estratégia de gerenciamento adotada.

Além disso, também está disponível como aplicativo móvel, para Android e iOS, no qual é possível acompanhar o gerenciamento do projeto.

O Trello é utilizado no projeto para dar apoio à metodologia *Kanban*, na qual foram criadas as etapas *Product Backlog*, *Sprint Backlog*, *Doing*, *In Review* e *Done*, e

cada história de usuário é constantemente atualizada quanto ao estado no decorrer do projeto.

4.2.4 Xcode

Xcode é uma IDE (*Integrated Development Enviroment*) e software livre da Apple e consiste em um conjunto de ferramentas que os desenvolvedores usam para construir aplicativos para plataformas Apple (DEVELOPER, 2020). O Xcode é utilizado para gerenciar todo o fluxo de trabalho de desenvolvimento, desde a criação do seu aplicativo até o teste, a otimização e o envio para a *App Store*.

O Xcode é a ferramenta de desenvolvimento oficial e recomendada pela Apple e foi escolhida por possuir todas funcionalidades necessárias para o desenvolvimento do aplicativo iOS, além de o pesquisador ter experiência prévia com a ferramenta.

4.2.5 Swift

O Swift é uma linguagem de programação fácil de usar e em código aberto, desenvolvida pela Apple para a criação de apps para iOS, Mac, Apple TV e Apple Watch (APPLE, 2022).

A escolha desta linguagem de programação se deu por ser uma das linguagens oficiais para desenvolvimento na plataforma iOS adotada pela Apple. Além disso, por possibilitar interoperabilidade com Objective-C, ela permite utilizar *frameworks* e bibliotecas legado úteis para desenvolvimento.

5 Desenvolvimento

Este capítulo refere-se ao desenvolvimento da proposta apresentada no Capítulo 4, conforme a metodologia e processo de desenvolvimento do TCC 2. Será abordado como foi o processo de desenvolvimento do aplicativo iOS, assim como as dificuldades e adaptações realizadas na proposta inicial para conclusão do trabalho.

5.1 Alterações de Requisitos

Iniciado o desenvolvimento e, portanto, com uma compreensão maior sobre o projeto, foram necessárias alterações nos requisitos elicitados, conforme pode ser visto na Figura 17. Percebe-se a adição da *feature* - 07: Configuração de Ambiente que possibilita a configuração do ambiente que o professor indicar, sendo esta funcionalidade já presente no aplicativo Android. Além disso, algumas *features* foram subdivididas em mais histórias de usuário.

Feature	História de Usuário	Plataforma	Sprint	Priorização
[F01] Autenticação	[US01] Eu, como aluno, gostaria de realizar login no aplicativo para conseguir utilizar suas funcionalidades	iOS	Sprint 1	Must
	[US02] Eu, como aluno, gostaria de realizar logout no aplicativo para minhas informações não ficarem salvas	iOS	Sprint 1	Must
[F02] Questões	[US03] Eu, como aluno, gostaria de resolver questões sobre as disciplinas que estou cursando para verificar o meu aprendizado	iOS	Sprint 2	Must
	[US04] Eu, como aluno, gostaria de filtrar as questões que desejo realizar pelo tipo de questão, para realizar questões apenas do tipo que desejar	iOS	Sprint 2	Could
	[US05] Eu, como aluno, gostaria de filtrar as questões que desejo realizar pela dificuldade, para realizar questões apenas da dificuldade que desejar	iOS	Sprint 2	Could
[F03] Listas	[US06] Eu, como aluno, gostaria de realizar listas de questões sobre as disciplinas que estou cursando para verificar o meu aprendizado	iOS	Sprint 2	Must
	[US07] Eu, como aluno, gostaria de ver meus resultados nas listas que já fiz para verificar o meu resultado	iOS	Sprint 2	Should
[F04] Testes	[US08] Eu, como aluno, gostaria de realizar teste sobre os conteúdos das disciplinas que estou cursando para verificar o meu aprendizado	iOS	Sprint 5	Must
	[US09] Eu, como aluno, gostaria de verificar meus resultados em testes já realizados para conseguir ver meus resultados e receber orientações	iOS	Sprint 5	Should
	[US10] Eu, como aluno, gostaria de desistir de realizar um teste que já comecei para poder refazê-lo do início	iOS	Sprint 5	Could
	[US11] Eu, como aluno, gostaria de continuar um teste que já comecei para poder finalizá-lo	iOS	Sprint 5	Could
	[US12] Eu, como desenvolvedor, gostaria de implementar o endpoint de buscar testes realizados para fornecer ao aluno os testes que ele já realizou	Web	Sprint 4	Should
	[US13] Eu, como desenvolvedor, gostaria de implementar o endpoint de buscar questões de um teste para que o aluno consiga visulizá-la	Web	Sprint 4	Must
	[US14] Eu, como desenvolvedor, gostaria de implementar o endpoint de responder questão de um teste para que o aluno consiga respondê-la	Web	Sprint 4	Must
	[US15] Eu, como desenvolvedor, gostaria de implementar o endpoint de desistir de um teste para que o aluno consiga recomeçar o teste	Web	Sprint 4	Could
	[US16] Eu, como desenvolvedor, gostaria de implementar o endpoint de buscar resultados de um teste para que o aluno consiga visualizar seu desempenho	Web	Sprint 4	Should
[F05] Situação	[US17] Eu, como aluno, gostaria de verificar minha situação na disciplina, de maneira gráfica, para verificar meu desempenho nos conteúdos e progressão na aprendizagem;	iOS	Sprint 3	Must
[F06] Orientação	[US18] Eu, como aluno, gostaria de solicitar orientação para encontrar o caminho mais adequado para melhorar meu desempenho na disciplina cursada	iOS	Sprint 3	Must
[F07] Configuração de Ambiente	[US19] Eu, como aluno, gostaria de configurar o ambiente do aplicativo para escolher a configuração que o meu professor indicar	iOS	Sprint 1	Should

Figura 17 – Requisitos do produto atualizados. Fonte: Autor.

Ainda pode ser observado que o desenvolvimento das histórias de usuário foi divido

em *sprints*, sendo estas de duas semanas, conforme a metodologia detalhada na seção 4.1, realizadas na seguinte forma:

- ***Sprint* 1**: 06 de Junho até 19 de Junho;
- ***Sprint* 2**: 20 de Junho até 03 de Julho;
- ***Sprint* 3**: 04 de Julho até 17 de Julho;
- ***Sprint* 4**: 18 de Julho até 31 de Julho;
- ***Sprint* 5**: 01 de Agosto até 14 de Agosto.

Também foi adicionada priorização às histórias de usuário, que foi feita pelo autor deste trabalho, seguindo a metodologia MoSCoW, que é um método para priorizar requisitos com base no custo, risco e valor de negócio (MARTHASARI; SUHARSO; ARDIANSYAH, 2018). Segundo Ahmad et al. (2017), o MoSCoW é o método de priorização de requisitos de software mais citado e utilizado e esta metodologia agrupa os requisitos nas quatro categorias a seguir:

- ***Must* (deve ter)**: define os requisitos que devem ser incluídos no produto de software final;
- ***Should* (deveria ter)**: requisitos de alta prioridade que devem ser incluídos, se possível, dentro do prazo de entrega;
- ***Could* (poderia ter)**: requisitos desejáveis de se ter e podem ser incluídos, sem incorrer em muito esforço ou custo;
- ***Won't* (não será feito, por enquanto)**: requisitos que a parte interessada deseja ter, mas todas as partes interessadas têm o consenso de que os requisitos não serão implementados na versão atual.

Por fim, foi adicionado também às histórias de usuário a plataforma de desenvolvimento correspondente a ela. Isto deve-se ao fato de que para a construção do aplicativo Android, foi criado uma API no servidor do SAE para prover as funcionalidades dos módulos BDQ, MInA e Orientação Pedagógica para a aplicação Android. Assim, como o presente trabalho adiciona a implementação do módulo SIAC, foram criadas histórias de usuário referentes à criação de novas funcionalidades à API.

5.2 Alterações de Tecnologias

O desenvolvimento da aplicação Android foi feito por meio do modelo cliente-servidor, em que o aplicativo Android atua como um cliente e o aplicação web do SAE atua como servidor, provendo uma API para comunicação e troca de dados.

O modelo cliente-servidor pode ser definido como uma arquitetura de software composto tanto pelo cliente quanto pelo servidor, em que os clientes sempre enviam solicitações, enquanto o servidor responde as solicitações enviadas (KEDAH; OLUWATOSIN, 2014). Uma representação deste modelo pode ser observado na Figura 18.

O modelo cliente-servidor fornece uma comunicação entre processos porque envolve a troca de dados do cliente e do servidor em que cada um deles desempenha funções diferentes. Alguns dos protocolos padrões que clientes e servidores usam para se comunicarem entre si incluem: *File Transfer Protocol* (FTP), *Simple Mail Transfer Protocol* (SMTP) e *Hypertext Transfer Protocol* (HTTP).

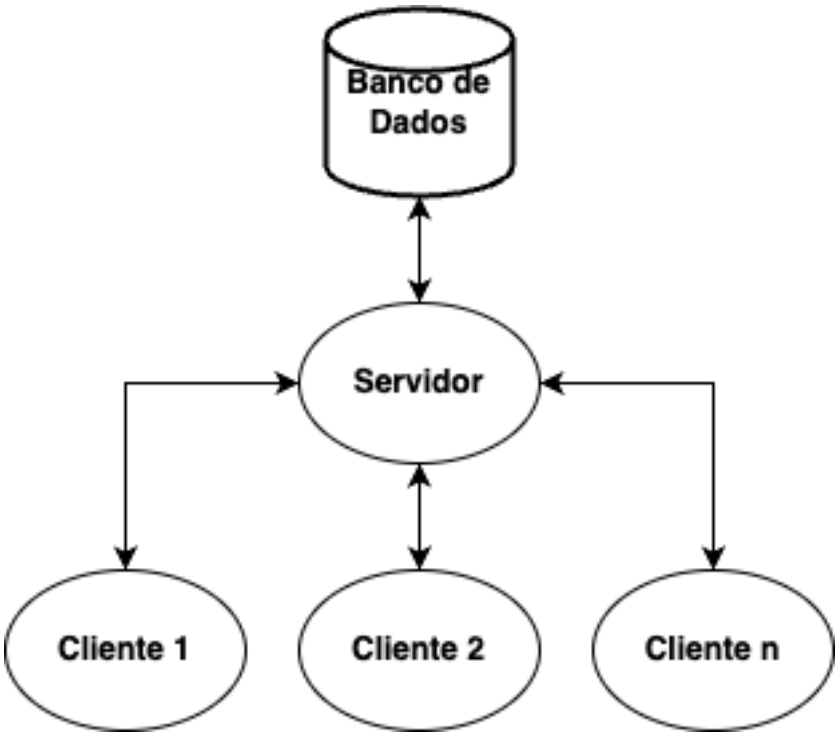

Figura 18 – Representação do modelo cliente-servidor. Fonte: Adaptado de (KEDAH; OLUWATOSIN, 2014).

Assim, para aplicação Android foi desenvolvido tanto o aplicativo quanto a API para os módulos BDQ, MInA e Orientação Pedagógica, conforme pode ser observado na Figura 19. Neste trabalho, foi desenvolvido o aplicativo iOS, com as tecnologias descritas na seção 2.1.3.1, e também a API para o módulo SIAC, que podem ser visto na Figura 20, utilizando a linguagem de programação Java e o *framework* Spring.

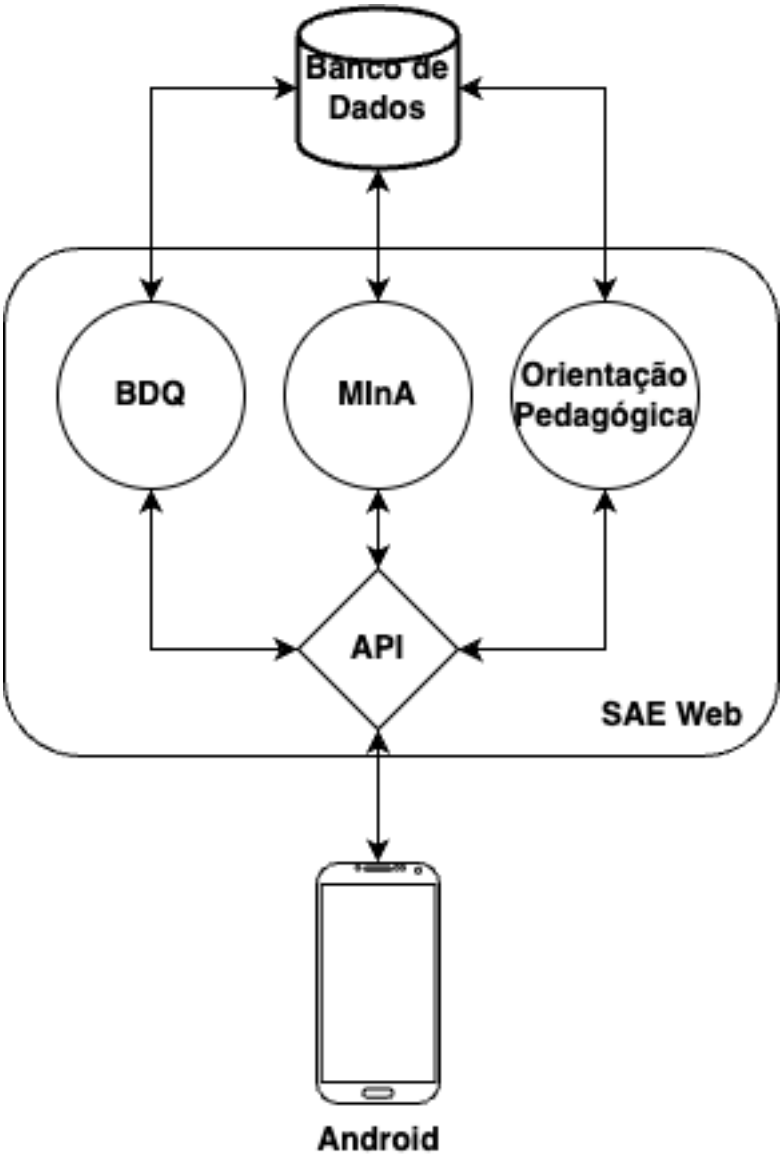

Figura 19 – Arquitetura cliente-servidor no contexto da aplicação Android. Fonte: Autor.

A linguagem de programação Java possibilita a utilização do paradigma de orientação a objetos, sendo compilada para gerar *bytecode* que para ser interpretada necessita da *Java Virtual Machine* (JVM) (ORACLE, 2022).

O Spring é um *framework* popular e de código aberto utilizado para criar aplicativos independentes de nível de produção que são executados na JVM. Além disso, torna o desenvolvimento de aplicativos web e microsserviços mais rápido e fácil por meio de três recursos principais: configuração automática, abordagem opinativa para configuração e capacidade de criar aplicativos independentes (IBM, 2021).

A escolha dessas tecnologias se deu por motivos arquiteturais, visto que a API já as utilizavam.

Ainda para o desenvolvimento da API, foi utilizado o Eclipse, que é uma IDE

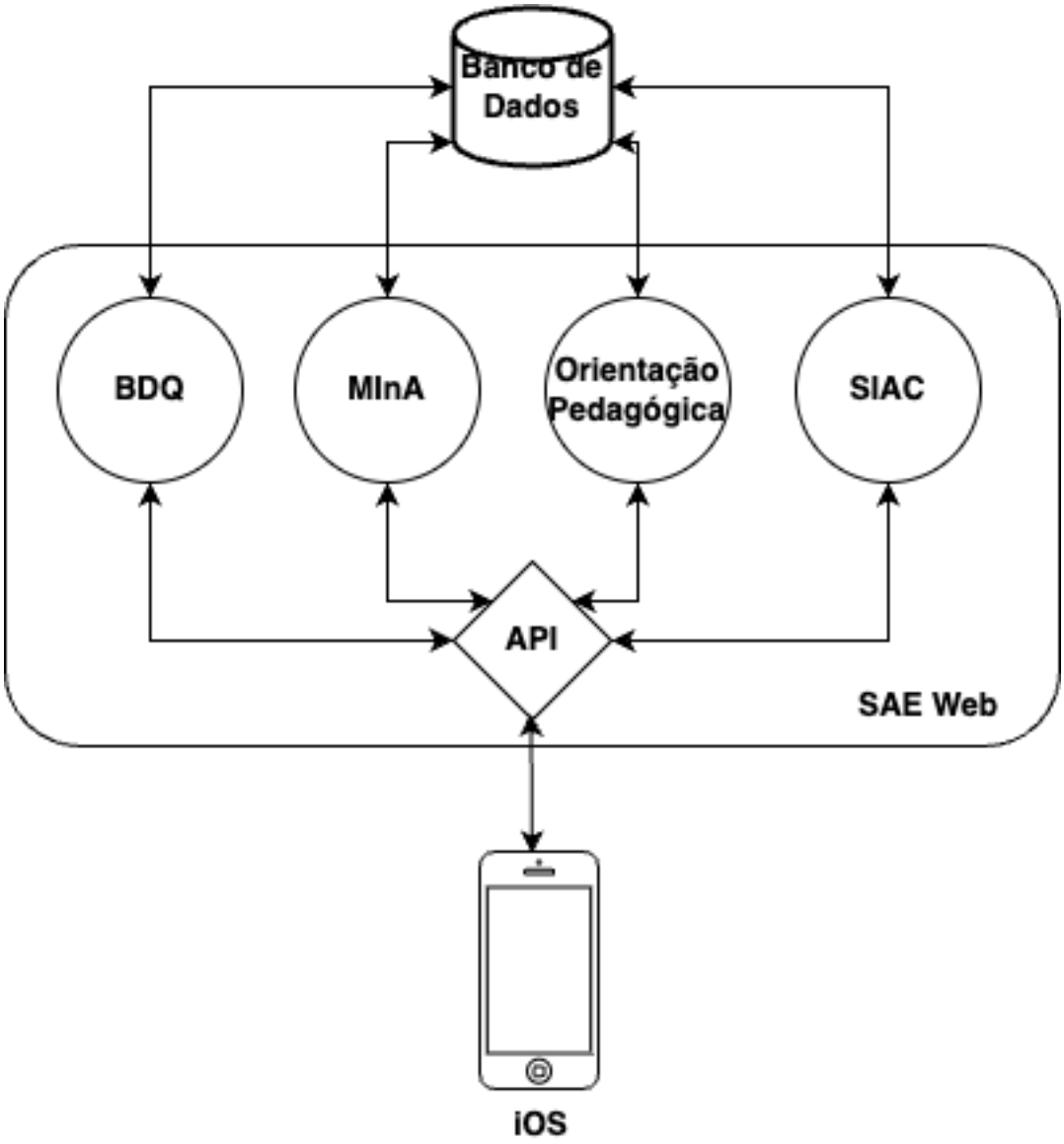

Figura 20 – Arquitetura cliente-servidor no contexto da aplicação iOS. Fonte: Autor.

criado para desenvolvedores de várias linguagem de programação, entre elas a Java, que reúne várias ferramentas que auxiliam o desenvolvedor, sendo algumas delas para depurar o código, realizar testes, entre outras (FOUNDATION, 2022). A escolha desta ferramenta também se deu levando em consideração que já era utilizada para o desenvolvimento da API.

Por fim, e como apontado por Brajesh (2017), documentação de API é um fator chave para que desenvolvedores a compreendam, além de ser importante para que sua adoção (integração) seja bem-sucedida. Assim, foi também utilizado o Swagger, que é um *framework* para documentação de API. Esta escolha se deu, pois este fornece uma maneira padrão e independente de linguagem para definir uma interface de API (BRAJESH, 2017).

5.3 Alterações de Cronograma

Ao longo do projeto diversos fatores levaram a alterações no cronograma. A pandemia de Covid-19, que levou ao encerramento temporário das atividades presenciais na Universidade de Brasília (UnB) foi um dos maiores causadores de atrasos. Assim, o cronograma teve de ser alterado algumas vezes, sendo a versão final ao desenvolvimento da segunda etapa do projeto (TCC 2) apresentado na Tabela 6.

Mês / **Atividade**	**Jun**	**Jul**	**Ago**	**Set**
Revisar proposta de solução	X			
Desenvolver aplicativo	X	X	X	
Testar e manter funcionalidades			X	
Gerar nova versão			X	
Realizar teste de usabilidade				X
Avaliar resultados obtidos				X
Apresentar TCC 2				X

Tabela 6 – Cronograma do TCC 2 ajustado. Fonte: Autor.

Quanto às atividades, a única alteração foi a adição da "Realizar teste de usabilidade", que se refere ao planejamento, execução e análise dos resultados de um teste de usabilidade do aplicativo, com o intuito de obter informações que auxiliem na detecção de problemas de usabilidade e que ajudem a melhorar a aplicação.

5.4 Implementação

Ao longo do desenvolvimento do projeto inteiro foi utilizado as metodologias *Scrum* e *Kanban*, conforme proposto na seção 4.1, assim, as histórias de usuários foram alocadas em *sprints*, de acordo com sua priorização.

A ferramenta "Trello"foi utilizada para apoiar o uso da metodologia, propiciando uma ampla visão do desenvolvimento ao longo do cronograma. Todas as histórias de usuário foram inseridas na ferramenta, conforme pode ser observado na Figura 21, que representa o quadro *Kanban* em um momento ao longo da *sprint* 3.

A fim de aprimorar a organização interna das tarefas, foram utilizadas *labels* (representadas por cores conforme apresentado na Figura 21) que simbolizavam uma característica da tarefa, sendo estas: "iOS", "Web", "Autenticação", "Questões", "Listas", "Testes", "Situação", "Orientação"e "Configuração de Ambiente".

O início da implementação deu-se a partir da primeira *sprint*, que objetivava implementar as funcionalidades de *login*, *logout* e configuração de ambiente, conforme a

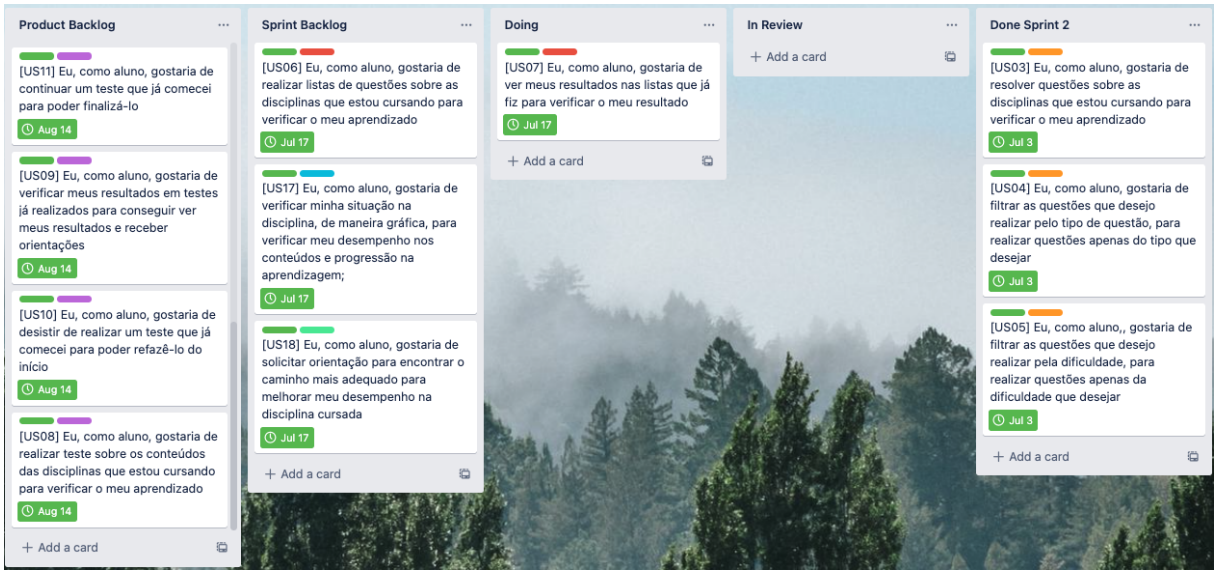

Figura 21 – Quadro *Kanban* ao longo da *sprint* 3. Fonte: Autor.

priorização, por serem funcionalidades essenciais ao projeto e também necessárias para utilização das demais funcionalidades.

O desenvolvimento evoluiu da simulação da proposta, que pode ser observado no Apêndice A, e foi implementado as telas que podem ser observadas na Figura 22. Na primeira imagem o usuário insere um ambiente válido do servidor SAE, que será passado pelo docente responsável, após isso, na imagem posterior, o aluno insere corretamente os dados de *login*, email e senha, da sua conta no SAE. Por fim, tem acesso a tela inicial do aplicativo, que contém as demais funcionalidades. No Apêndice B.1 pode ser visto a funcionalidade de *logout*, que também foi desenvolvido ao longo dessa *sprint*.

Na *sprint* seguinte, deu-se prioridade para a implementação dos módulo de questões, que fornece opções de resolver questões avulsas ou de listas, pois abrange, dentre outras funcionalidades, a possibilidade de visualizar e responder questões, funcionalidade esta que foi utilizada tanto para este módulo quanto para o módulo do SIAC.

Assim, foram criadas as telas que podem ser vistas na Figura 23. A partir da tela inicial de questões, observado na primeira imagem, podem-ser adicionados, opcionalmente, filtro de tipo de questão, que pode ser observado na segunda imagem, filtro de dificuldade e escolha de curso, disciplina e conteúdo. Após isso, o aluno pode visualizar e responder a questão avulsa, conforme a terceira imagem, e, por fim, receber *feedback* do seu acerto ou erro, conforme a quarta imagem da Figura 23 ainda.

Esta foi a única *sprint* em que nem todas as histórias de usuário alocadas foram concluídas, devido a quantidade dos tipos diferentes de questões, que podem ser vistas no Apêndice B.2. Assim, foi desenvolvido apenas o módulo de questões nessa *sprint*, e, na *sprint* 3 foi desenvolvido o módulo que corresponde ao acesso das listas de exercícios. Na

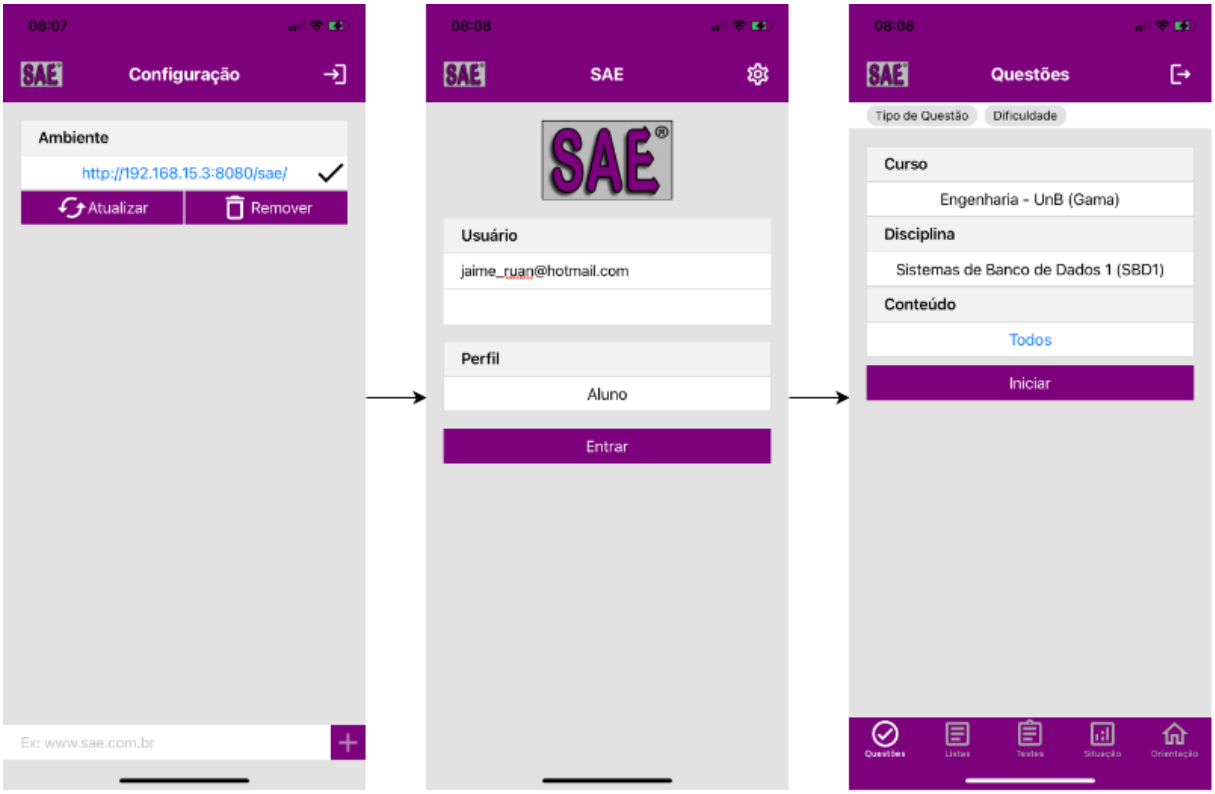

Figura 22 – Fluxo de *login* do usuário. Fonte: Autor.

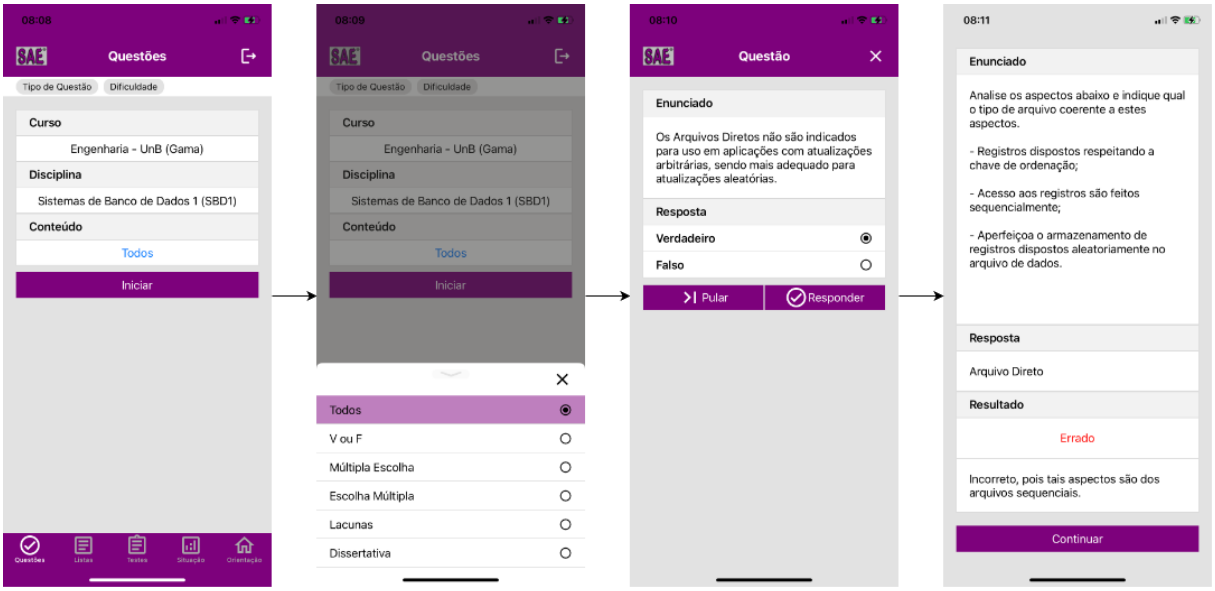

Figura 23 – Fluxo de questões avulsas. Fonte: Autor.

Figura 24 pode ser observado a busca dessas listas e o resultado da lista, quantidade de acertos e erros, de forma gráfica, no caso de listas resolvidas. Para listas ainda não realizadas, o usuário é direcionado para a tela de responder questões, conforme demonstrado na Figura 23.

Ainda na *sprint* 3, foram desenvolvidos também os módulos de orientação e situ-

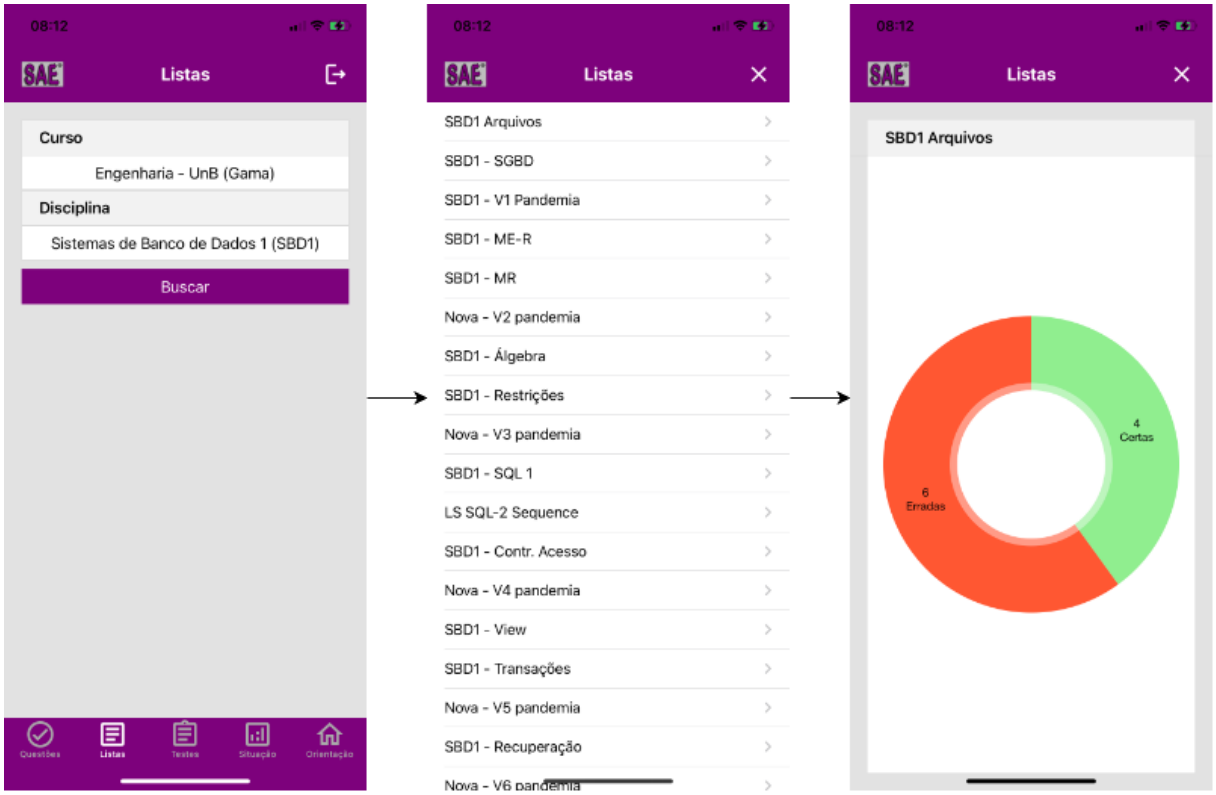

Figura 24 – Fluxo para listas de exercícios no SAE. Fonte: Autor.

ação, que podem ser observados na Figura 25 e Figura 26, respectivamente.

No módulo de Orientação, o estudante pode escolher o curso e disciplina que deseja receber orientação pedagógica do SAE, conforme a primeira imagem da Figura 25 apresenta, e, após isso, é apresentada a orientação pedagógica para a disciplina, incluindo a representação da MInA realizando uma interação emotiva e significativa à situação atual de aprendizagem do aluno. Tal representação acontece por meio de uma imagem estática e não a apresentação da animação da interação do agente pedagógico mostrado no SAE que utiliza arquivos padrões swp.

De maneira semelhante, no módulo de situação, observado na Figura 26, o estudante escolhe o curso e a disciplina que deseja ver sua situação acompanhada pelo SAE. Após isso, é apresentado a ele, de forma gráfica, seu resultado com base nas questões realizadas.

A quarta *sprint* teve como foco a criação das funcionalidades do módulo SIAC na API do SAE, uma vez que se trata da única maneira do aplicativo ser capaz de acessar as funcionalidades deste módulo. Assim, a API foi estudada e incrementada para tal processamento que estará acontecendo somente para o mobile em iOS, sendo, necessárias as seguintes funcionalidades:

1. **Verificar habilidade do aluno:** como apresentado na seção 2.2.4.2.3, a habili-

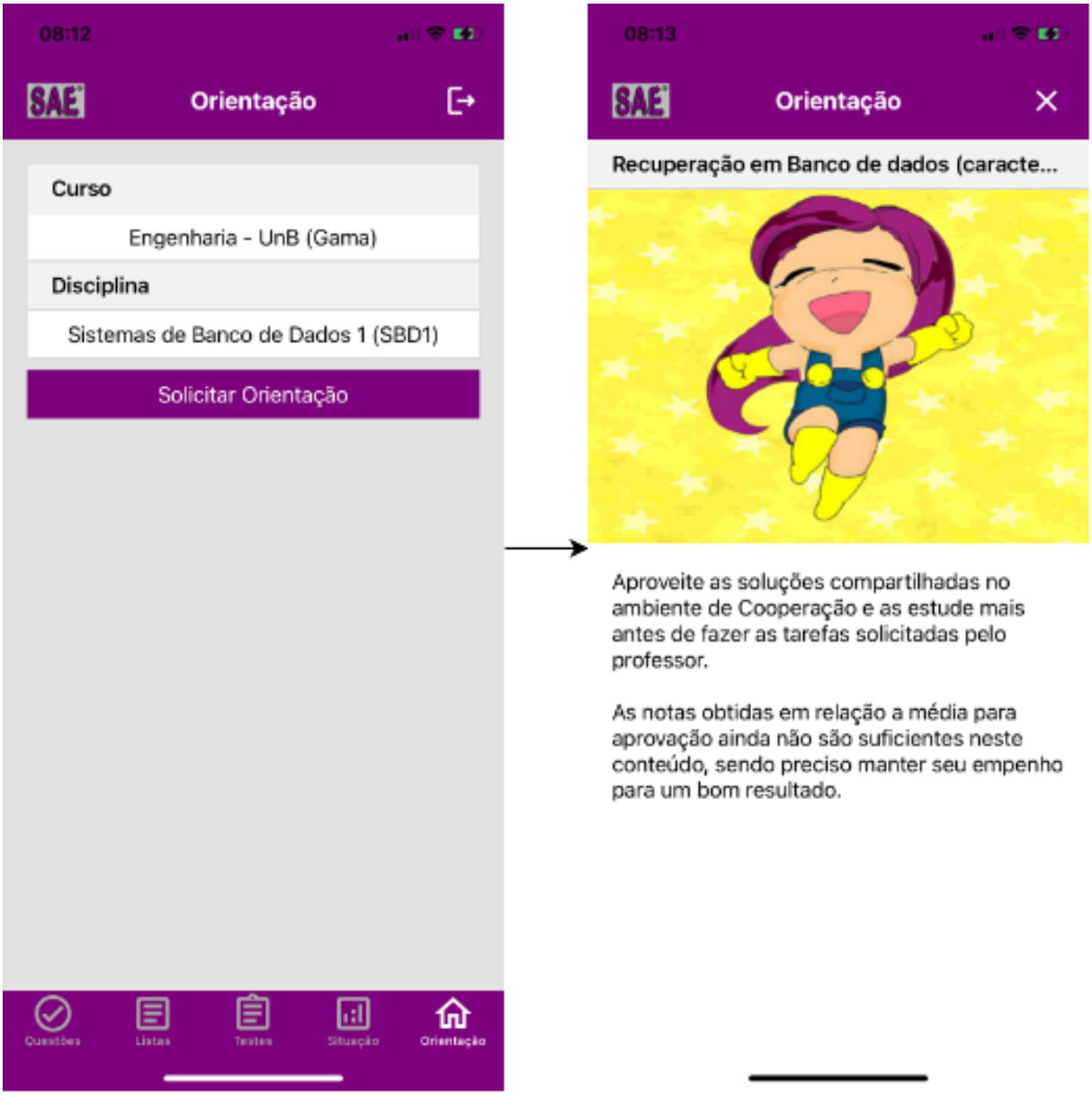

Figura 25 – Fluxo para o módulo de Orientação. Fonte: Autor.

dade do aluno é considerada para que ele receba uma questão apropriada ao seu conhecimento momentâneo;

2. **Verificar soma da informação:** outro parâmetro que é levado em consideração para a que o estudante receba uma questão apropriada a sua realidade cognitiva;

3. **Buscar questão inicial do teste:** apresentar a questão inicial do teste para o aluno;

4. **Responder questão do teste:** responder a questão apresentada;

5. **Verificar se existem testes incompletos:** buscar testes incompletos, a fim de que o estudante decida se irá continuá-lo ou desistir de realizá-lo;

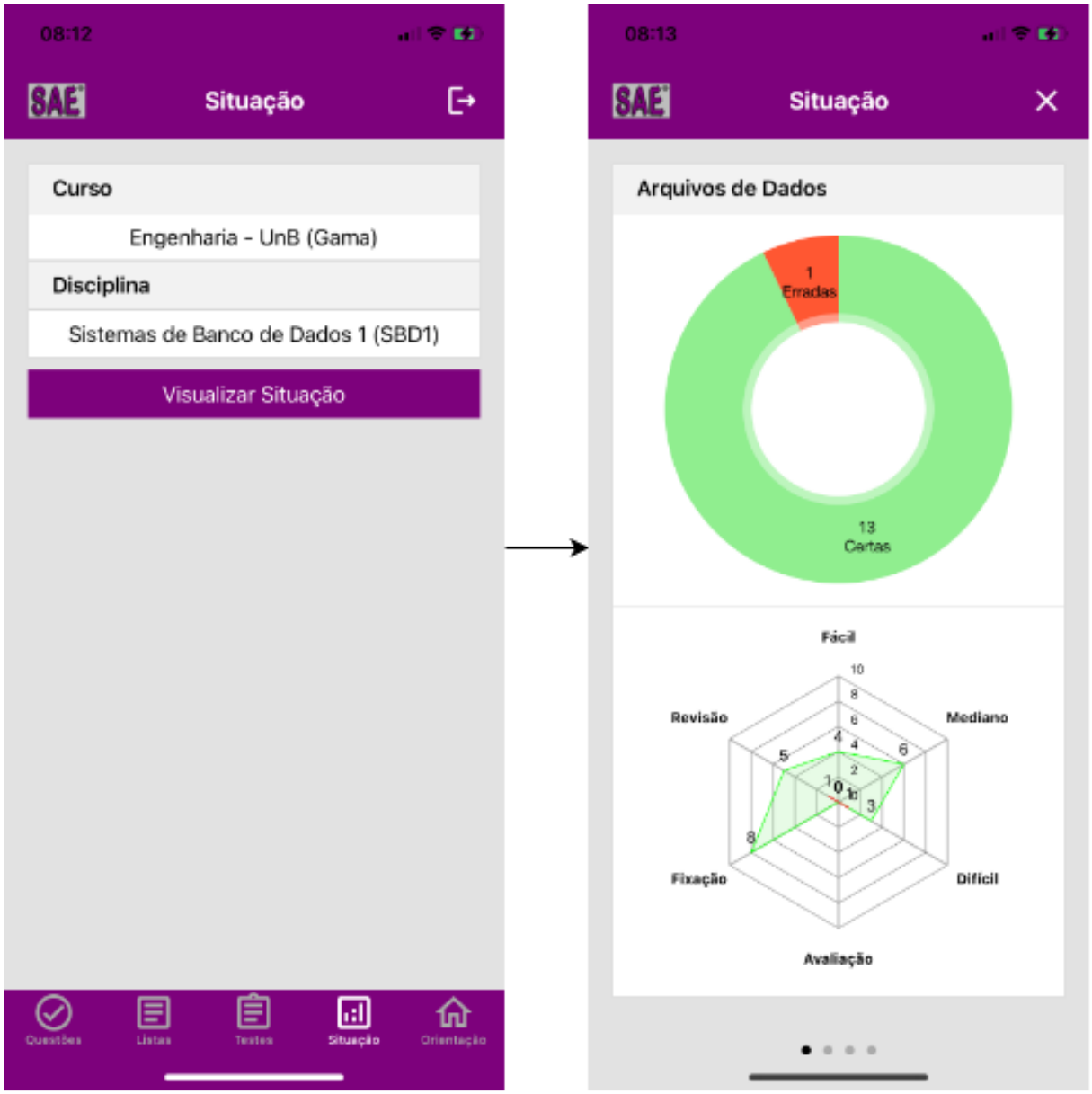

Figura 26 – Fluxo para visualizar a sua própria situação acompanhada pelo SAE. Fonte: Autor.

6. **Desistir de um teste:** desistir de realizar um teste iniciado previamente;
7. **Buscar mais questões do teste, se existir:** buscar mais questões de um teste, tanto para testes em andamento, como para testes incompletos;
8. **Buscar testes feitos:** buscar todos os testes já realizados pelo estudante;
9. **Verificar resultados de um teste feito:** apresentar os resultados e sugestões do teste para o aluno.

Após a identificação das funcionalidades da API, estas foram desenvolvidas e, como apresentada na seção 5.2, a API inteira foi documentada com o *framework* Swagger, com o

intuito de registrar e facilitar o entendimento de seu uso. Na Figura 27 pode ser observada a documentação da API do módulo SIAC.

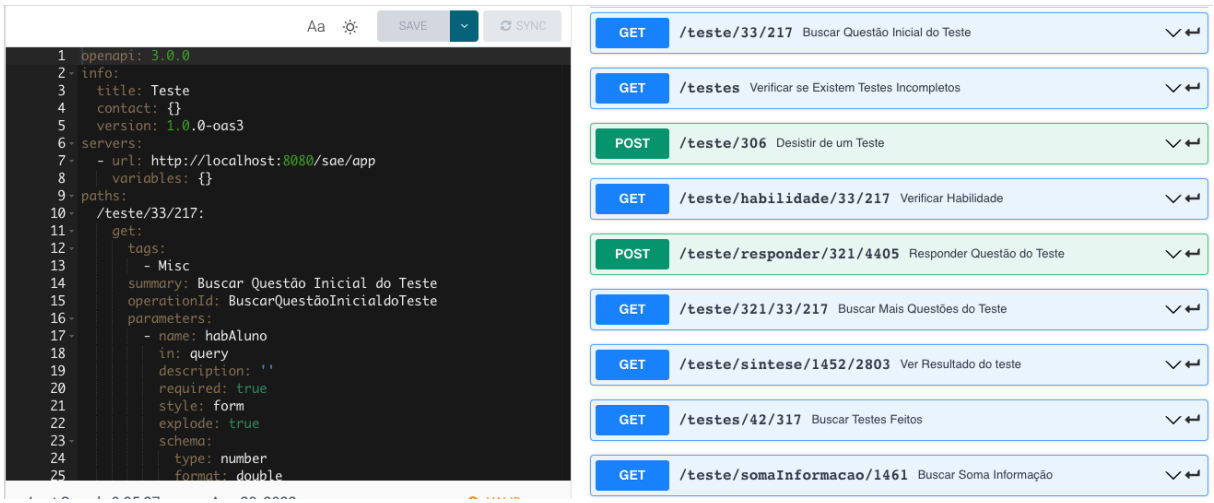

Figura 27 – Documentação das API do módulo SIAC. Fonte: Autor.

Na *sprint* final foram desenvolvidas todas as funcionalidades referentes ao módulo SIAC no aplicativo. Na Figura 28 observa-se o fluxo de visualização de resultado de um teste, em que a primeira imagem representa a tela em que o estudante escolhe o curso e a disciplina, na segunda tela da Figura 28 o usuário escolhe o teste realizado e nas demais lhe é apresentado o resultado do teste, que contém os dados gerais do aluno e do teste realizado, seus erros e acertos, além de sugestões pedagógicas baseadas na habilidade do estudante no resultado obtido no referido teste sobre um conteúdo que compõe uma disciplina.

Para a realização de um teste o aluno é direcionado para a tela de responder questões, conforme já apresentado no módulo de questões. Outras funcionalidades referentes a este módulo podem ser vistas no Apêndice B.3.

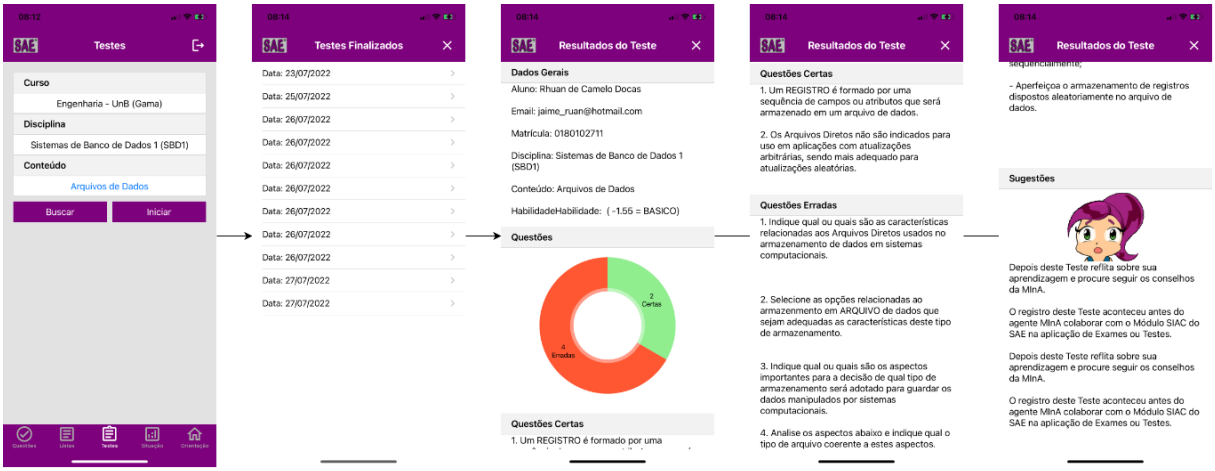

Figura 28 – Fluxo de realização de teste. Fonte: Autor.

A Figura 29 apresenta o quadro *Kanban* ao final do desenvolvimento do projeto, evidenciando a *sprint* em que cada história de usuário foi finalizada.

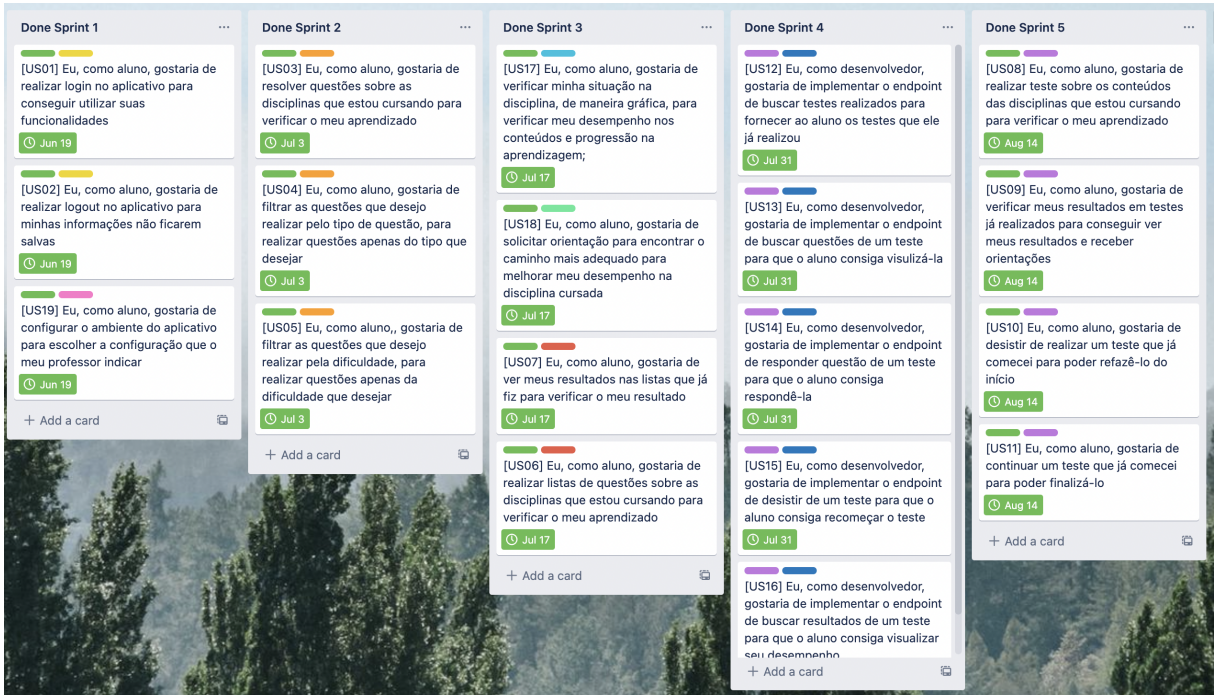

Figura 29 – Quadro *Kanban* ao final da última *sprint*. Fonte: Autor.

5.5 Teste de Usabilidade

A usabilidade é considerada um dos atributos mais importantes de qualidade das aplicações para dispositivos móveis, uma vez que afeta a satisfação dos usuários (KETOLA; RöYKKEE, 2001). Segundo a ISO/IEC-25010 (2011), a usabilidade pode ser definida como: *"capacidade do produto de software de ser compreendido, aprendido, operado e atraente ao usuário, quando usado sob condições específicas"*. Ela é importante para a aceitação das aplicações e, por esta razão, já foram propostos métodos de avaliação com o objetivo de melhorar esse atributo.

Um dos seus principais métodos de avaliação, sob a perspectiva do usuário, é o teste de usabilidade, que captura e analisa os dados dos usuários finais, enquanto utilizam a aplicação seguindo um conjunto pré-determinado de atividades (FERNANDEZ; INSFRAN; ABRAHãO, 2011). A partir da análise dos resultados é possível obter informações que auxiliam na detecção de problemas de usabilidade e ajudam a melhorar a aplicação final.

Como apontado por Cooper (1999), ao agrupar os tipos de usuários semelhantes, é mais fácil analisar e tirar conclusões dos testes, e, ao utilizar representantes de usuários reais, as interferências no processo diminuem. Assim, foram realizados testes com dois

grupos de usuários, em que cada grupo é representado por uma persona, que descreve as especificações comportamentais que incorporam os objetivos e necessidades dos usuários arquetípicos (FAILY; LYLE, 2013). O teste foi aplicado com três usuários correspondente a cada persona e, a seguir, se encontra a descrição de cada persona:

- **Persona 1**: estudante que está entre o quinto e o oitavo semestre de Engenharia de Software, conhece e já utilizou a plataforma web do SAE devido as disciplinas que cursou e está familiarizado com o uso de dispositivos móveis, apesar de preferir utilizar o computador;
- **Persona 2**: estudante do ensino fundamental ou médio, não conhece e nunca utilizou o SAE, mas já utilizou outros aplicativos educacionais, e está bastante familiarizado com o uso de dispositivos móveis, inclusive prefere utilizá-los para auxiliar seu processo de aprendizagem;

Para a realização de um teste de usabilidade deve-se selecionar atividades estratégicas para o funcionamento adequado do aplicativo (CRUZ; LIMA, 2014). Desse modo, as seguintes atividades foram selecionadas:

- **AT01**: Realizar *login* no aplicativo, incluindo a configuração de ambiente;
- **AT02**: Realizar ao menos três questões, adicionando ou não filtros;
- **AT03**: Realizar um teste, incluindo a visualização dos resultados;
- **AT04**: Verificar situação em uma disciplina/conteúdo;
- **AT05**: Visualizar orientação em uma disciplina.

A coleta dos dados consistiu na sessão de observação do teste e na entrevista pós teste. Esta etapa consistiu no teste de usabilidade em si, e aconteceu do seguinte modo: o usuário foi brevemente apresentado ao teste e às tarefas que deveria realizar. Após isso, ele utilizou o aplicativo, gastando por volta de cinco a dez minutos. E, por fim, foram feitas perguntas para saber a opinião dos usuários sobre o sistema testado.

O objetivo dos testes foi avaliar as seguintes metas de usabilidade: quantidade de erros encontrados e satisfação com o aplicativo.

Não foi encontrado nenhum erro durante a execução dos testes, considerando erro como o comportamento em que um usuário tentou realizar uma função não permitida pelo software ou quando ele conseguiu finalizar a interação mas esta não produzia o efeito desejado.

Em relação à satisfação dos usuários, foi verificada a percepção destes no que se refere à utilidade, facilidade de uso e interface do aplicativo, como pode ser observado na Tabela 7, Tabela 8 e Tabela 9. Foram utilizadas como base as perguntas do questionário aplicado por (VALENTIM et al., 2014). Para simbolizar respostas foi considerado: "S"como satisfeito; "M"como nem satisfeito e nem insatisfeito; e "I"como insatisfeito.

Tema	**Pergunta**	**Persona 1**			**Persona 2**		
		S	M	**I**	S	M	**I**
Interface	Considero as cores e botões do aplicativo agradáveis	3	0	0	3	0	0
	Consigo visualizar bem todos os botões e informações dentro do aplicativo	2	1	0	3	0	0
	Entendo com facilidade as palavras, nomenclaturas e informações dentro do aplicativo	3	0	0	3	0	0
	As imagens e ícones no aplicativo são de fácil reconhecimento	3	0	0	3	0	0
	Consigo visualizar todas as funcionalidades dentro do aplicativo	3	0	0	3	0	0
	Consigo navegar bem por todas as telas do aplicativo	3	0	0	3	0	0
Total		17	1	0	18	0	0

Tabela 7 – Percepção dos usuários sobre a interface do aplicativo. Fonte: Autor.

Analisando os dados da Tabela 7 em relação à interface do aplicativo, pode ser observado que, de maneira geral, ambas personas gostaram da interface e conseguiram navegar e visualizar suas funcionalidades. O comentário de um usuário foi que: *"Gostei bastante da organização da interface, bem familiar, com a organização em abas. A visualização do rendimento também foi bem facilitada, podendo deslizar entre os gráficos para cada conteúdo cursado(...)"*.

No entanto, houve uma discordância quanto à visualização das informações apresentadas pelo aplicativo, sobre a qual um usuário comenta: *"Quando os textos dos itens de uma questão são longos, seria interessante visualizá-los por completo."*. Assim, observa-se que o aplicativo pode precisar de melhorias na sua interface, pois se os usuários tiverem dificuldades para visualizar as informações apresentadas, a utilização da aplicação poderá ser prejudicada.

Tema	**Pergunta**	**Persona 1**			**Persona 2**		
		S	**M**	**I**	**S**	**M**	**I**
Facilidade de Uso	Foi fácil aprender a utilizar o aplicativo	3	0	0	2	1	0
	Eu conseguia entender o que acontecia durante o uso do aplicativo	3	0	0	2	1	0
	É fácil de lembrar como utilizar o aplicativo	3	0	0	2	1	0
	Considero o aplicativo fácil de utilizar	3	0	0	2	1	0
Total		12	0	0	8	4	0

Tabela 8 – Percepção dos usuários sobre a facilidade de uso do aplicativo. Fonte: Autor.

Na Tabela 8 pode ser observado que houve uma discordância considerável em relação a facilidade de utilização do aplicativo entre as personas. Uma possível causa para isso pode ser que os usuários da persona 1 já tem experiência prévia com o SAE web, incluindo o comentário de um testador representante dessa persona: *"Em comparação com o SAE 'original', é mais fácil de utilizar pois se parece com outros aplicativos."*. Deste

modo, percebe-se que o aplicativo pode não ser tão fácil para usuários que desconhecem o SAE, podendo ser necessário a realização de testes de usabilidade mais exigentes para averiguar esse critério.

Tema	**Pergunta**	**Persona 1**			**Persona 2**		
		S	**M**	**I**	**S**	**M**	**I**
Utilidade	Considero o aplicativo útil para melhorar meu aprendizado	3	0	0	3	0	0
	Considero que o aplicativo melhoraria minha produtividade para realização das atividades e aprendizado	3	0	0	3	0	0
	Considero que o aplicativo facilitaria a realização das minhas atividades	3	0	0	3	0	0
Total		9	0	0	9	0	0

Tabela 9 – Percepção dos usuários sobre a utilidade do aplicativo. Fonte: Autor.

Pode-se notar na Tabela 9 que os usuários consideraram o aplicativo como uma ferramenta possivelmente útil para auxiliar no seu processo de aprendizagem, em que um testador comenta: *"(...)Ter um acesso 100% adaptado para o mobile também foi muito bom, pois houveram vários momentos, principalmente durante a pandemia, nos quais tive que usar o SAE pelo celular."*

6 Considerações Finais

Neste capítulo estão descritas as análises e resultados finais obtidos após o desenvolvimento do projeto, além da avaliação de se os objetivos propostos foram alcançados. Além disso, foram sugeridos possíveis trabalhos futuros que podem ser desenvolvidos a partir deste.

6.1 Conclusões

No decorrer deste trabalho, foi possível aplicar diversos aspectos abordados ao longo do curso de Engenharia de Software, desde o planejamento até a execução do processo de desenvolvimento do aplicativo SAE para o sistema operacional iOS. Este trabalho possibilitou o contato com áreas de software que mais interessavam e motivavam os envolvidos nesta pesquisa, além de outras áreas do conhecimento que forem necessárias.

Com o intuito de alcançar os objetivos planejados, inicialmente foi realizado um levantamento bibliográfico acerca do tema, além da realização da documentação referente à metodologia e arquitetura do projeto. Assim, foi descrito todo funcionamento do sistema proposto e seu processo de desenvolvimento a partir da análise e planejamento, utilizando uma metodologia adequada aos objetivos almejados.

Em relação ao projeto, conforme descrito no Capítulo 5, alguns fatores provocaram alterações e adições significativas, além da pandemia de Covid-19, que levou ao encerramento temporário das atividades presenciais na Universidade de Brasília. Estes causaram alterações no cronograma, porém nada que tenha impedido o trabalho de ser realizado em um tempo maior que o estimado inicialmente.

O objetivo principal deste trabalho foi desenvolver uma aplicação do SAE para o sistema operacional iOS, integrando os módulos já abordados neste trabalho, fornecendo interação adequada aos usuários dessa plataforma, possibilitando-os utilizar seu dispositivo móvel com iOS como uma ferramenta para auxiliar no seu processo de aprendizagem por meio do uso do SAE.

Após a etapa de desenvolvimento, em que todas as funcionalidades propostas foram implementadas, a API do SAE foi incrementada, possibilitando que outras aplicações, de dispositivos móveis ou não, utilizem os recursos do módulo SIAC.

Além disso, foi adicionado ao ecossistema do SAE uma aplicação para iOS, que possibilita o estudante responder questões, realizar e visualizar resultados de listas e testes, verificar, de maneira gráfica, a situação de aprendizagem nas disciplinas que está cursando com acompanhamento do SAE, receber orientação pedagógica, entre outros recursos, por

meio de seu dispositivo.

Dessa forma, se proporcionou ao usuário a possibilidade de desfrutar dos aspectos positivos do *mobile learning*, abordados na seção 1.3 deste trabalho, e todas as funcionalidades citadas, que ele tem na aplicação web, no seu dispositivo móvel.

Ainda com a finalidade de cumprir os objetivos, foi realizado um teste de usabilidade do aplicativo com duas personas de usuários do SAE. Assim, foram coletados e analisados tanto dados relativos à aplicação quanto à percepção do usuário ao utilizá-la. Percebeu-se que poderá ser necessário alterações que visem melhorar a experiência e entendimento do usuário ao utilizar o aplicativo, mas, para ambas personas, o aplicativo atendeu ao proposto no que se refere aos aspectos de usabilidade abordados.

A partir da análise dos resultados obtidos é possível observar que os objetivos propostos na seção 1.4.1 foram atendidos. Visto que a aplicação proposta foi inteiramente desenvolvida, poderá ser fornecida aos estudantes uma alternativa para continuar seu aprendizado por meio da ferramenta do SAE para iOS e, ao menos de modo inicial, pôde-se observar que os usuários que testaram a aplicação tiveram uma percepção de que ela poderia contribuir em seu aprendizado.

Após a execução deste trabalho, e como já bem abordado na literatura (DUMAS; REDISH, 1999), (NIRANJANAMURTHY et al., 2014), (FERRE et al., 2001), (HOLZINGER, 2005), foi possível perceber que é fundamental a realização de testes com o usuário final a fim de validar se, na perspectiva dele, o software efetivamente cumpre sua finalidade.

A aplicação ainda não foi devidamente distribuída por meio da loja de aplicativos e todos os testes realizados utilizaram dados reais de questões, listas de exercícios e testes de disciplinas em períodos anteriores, além disso, obedeceram todas as regras de privacidade e manteve os dados dos participantes (testadores) em sigilo. Porém, mesmo em um ambiente controlado, foi possível validar o funcionamento da aplicação. Posteriormente, o aplicativo poderá ser publicado na loja de aplicativos correspondente, ou ainda passar por outros testes mais exigentes antes de ser disponibilizado para o uso em ambiente de produção.

6.2 Trabalhos Futuros

O trabalho apresentado implementou um aplicativo do SAE para o sistema operacional iOS que integra as funcionalidades dos módulos BDQ, Orientação Pedagógica, MInA e SIAC, fornecendo a possibilidade do usuário do SAE, com perfil de estudante, utilizar seu dispositivo móvel como mais uma ferramenta para seu auxílio no processo de aprendizagem. No entanto, existem diversas possibilidades de continuidade evolutiva, e possivelmente corretiva, para o presente trabalho, diante dos desafios que a área da Infor-

mática na Educação vem trilhando com o fim de tornar o processo ensino-aprendizagem mais eficiente e de melhor qualidade.

Primeiramente, como já abordado, o SAE implementa recursos de modo a fornecer informações para as diferentes necessidades dos perfis de usuários que atende, sendo estes: aluno, professor, monitor, diretor de curso, coordenador pedagógico ou assessor do diretor e responsável pelo aluno. Uma oportunidade de trabalho futuro seria adicionar no aplicativo iOS suporte aos diferentes perfis de usuário aos módulos BDQ e SIAC, em que:

- O professor será responsável pelo cadastramento de questões nos módulos BDQ e SIAC, indicando seu grau de dificuldade, categoria e discriminação. Além disso, o docente ainda fornecerá à questão o tópico do conteúdo ao qual estaria vinculada, o tipo da questão e qual categoria. Cada professor ainda poderá acompanhar o resultado dos testes avaliativos resolvidos por seus estudantes, envolvendo os detalhes de cada questão que possa ter participado em cada teste;

- Para o monitor será possível consultar os resultados alcançados no BDQ e no SIAC de todos os alunos que estão matriculados na disciplina vinculados a sua monitoria durante o respectivo semestre letivo vigente.

- O diretor poderá visualizar os resultados de qualquer estudante que esteja matriculado nas disciplinas vinculadas aos cursos que estão sob sua coordenação, além de observar a atuação dos docentes responsáveis por cada turma destas disciplinas acompanhadas pelo SAE.

- Ao responsável por alunos o acompanhamento dos resultados obtidos para colaborar na participação e colaboração ao sucesso do estudante em formação.

Ainda com o intuito de adicionar suporte a diferentes perfis de usuários do SAE no aplicativo iOS, outra oportunidade de trabalho futuro seria o desenvolvimento do módulo conhecido como PMon (Projeto de Monitoria Estudantil). Como observado por Assunção, Lopes e Rissoli (2008), este módulo almeja o acompanhamento personalizado de cada aprendiz, oferecendo ao docente informações detalhadas sobre a dedicação de cada aluno nestas atividades complementares, além do próprio parecer do monitor, que realizou o atendimento do aprendiz, sobre a atual situação do estudante nos conteúdos abordados durante este atendimento. Esses autores ainda apontam que a integração deste módulo ao SAE na web já levou ao aumento de até 14% no índice de aprovação. Assim, esta integração poderia trazer mais benefícios ao processo educacional.

Este trabalho ainda implementou o módulo SIAC na aplicação iOS, recurso este não disponível até então em nenhuma aplicação para dispositivos móveis. Este módulo emprega a TRI como recurso para mensurar a habilidade de cada aprendiz e lhe fornecer

questões coerentes com seu nível de conhecimento em testes a serem realizados em conteúdo específico que compõe uma disciplina. Portanto, outra oportunidade de trabalho futuro seria integrar este módulo na aplicação para Android, a fim de que os usuários desse outro sistema operacional possam aproveitar dos possíveis benefícios já disponíveis para os usuários do sistema operacional iOS, em que a API do SAE para o SIAC foi elaborada por este trabalho e facilitará a integração deste importante módulo do SAE também para os usuários dos dispositivos em Android.

Referências

AHMAD, K. S. et al. Fuzzy moscow: A fuzzy based moscow method for the prioritization of software requirements. p. 5, 2017. Citado na página 68.

AMATYA, S.; KURTI, D. A. Cross-platform mobile development: An alternative to native mobile development. p. 67, Oct 2013. Citado 2 vezes nas páginas 29 e 30.

ANDRADE, D. F. Comparando desempenhos de grupos de alunos por intermédio da teoria da resposta ao item. 2001. Citado 3 vezes nas páginas 45, 46 e 63.

APPLE. 2022. Disponível em: <http://www.apple.com/br/swift/>. Citado na página 65.

APPLE, I. ios technology overview. Apple, 2010. Citado 3 vezes nas páginas 11, 32 e 33.

APPLE, I. *Apple Developer Program*. 2020. Disponível em: <https://developer.apple.com/programs/>. Citado na página 33.

APPLE, I. *Swift. Uma linguagem aberta e poderosa para todo mundo criar apps incríveis.* 2020. Disponível em: <http://www.apple.com/br/swift/>. Citado na página 32.

ARRUDA, L. Desenvolvimento Ágil de software: Uma análise sintética a partir da metodologia kanban. In: . Congresso Norte Nordeste de Pesquisa e Inovação, 2012. Disponível em: <:<http://propi.ifto.edu.br/ocs/index.php/connepi/vii/paper/viewFile/3644/961>>. Citado na página 58.

ASSUNçãO, B. S. B.; LOPES, E. S.; RISSOLI, V. R. V. Sistema tutor inteligente integrado a monitoria estudantil para elaboração de um assistente virtual de ensino inteligente. p. 10, 2008. Citado na página 85.

AUSUBEL, D. P.; NOVAK, J. D.; HANESIAN, H. *Psicología Educacional*. Rio de Janeiro: Editora Interamericana, 1980. Citado na página 39.

BERNARDES, T. F.; MIYAKE, M. Y. Cross-platform mobile development approaches: A systematic review. *IEEE Latin America Transactions*, v. 14, n. 4, p. 1892–1898, Apr 2016. ISSN 1548-0992. Citado na página 30.

BOLZAN, W.; GIRAFFA, L. M. M. Estudo comparativo sobre sistemas tutores inteligentes multiagentes web. p. 54, Jul 2002. Citado 4 vezes nas páginas 11, 35, 36 e 37.

BRAJESH, D. *API Documentation*. Berkeley, CA: Apress, 2017. 59–80 p. Disponível em: <http://link.springer.com/10.1007/978-1-4842-1305-6_4>. Citado na página 71.

COOPER, A. *The Inmates Are Running the Asylum: why High-Tech Products Drive Us Crazy and How to Restore Sanity*. [S.l.]: Sams Publishing, 1999. Citado na página 79.

CRUZ, A. K. B. S. d.; LIMA, L. C. M. Estudo e testes de usabilidade em sistemas de autoria de software: Scratch e alice. In: *Anais do 11º Congresso Brasileiro de Pesquisa e Desenvolvimento em Design*. Gramado, RS: Editora Edgard Blücher, 2014. p. 3673–3685.

Disponível em: <http://www.proceedings.blucher.com.br/article-details/12941>. Citado na página 80.

DEVELOPER, A. *Xcode*. 2020. Disponível em: <https://developer.apple.com/xcode/>. Citado 2 vezes nas páginas 33 e 65.

DIAGRAMS.NET. *Diagram Software and Flowchart Maker*. 2020. Disponível em: <https://www.diagrams.net/>. Citado na página 64.

DUMAS, J. S.; REDISH, J. *A practical guide to usability testing*. Rev. ed. Exeter, England; Portland, Or: Intellect Books, 1999. ISBN 978-1-84150-020-1. Citado na página 84.

EL-KASSAS, W. S. et al. Taxonomy of cross-platform mobile applications development approaches. *Ain Shams Engineering Journal*, v. 8, n. 2, p. 163–190, Jun 2017. ISSN 20904479. Citado 3 vezes nas páginas 29, 30 e 31.

FAC, U. *Manual do TCC*. 2020. Disponível em: <http://fac.unb.br/tcc/index.html#tcc>. Citado na página 51.

FAILY, S.; LYLE, J. Guidelines for integrating personas into software engineering tools. In: *Proceedings of the 5th ACM SIGCHI symposium on Engineering interactive computing systems - EICS '13*. London, United Kingdom: ACM Press, 2013. p. 69. ISBN 978-1-4503-2138-9. Disponível em: <http://dl.acm.org/citation.cfm?doid=2494603.2480318>. Citado na página 80.

FERNANDEZ, A.; INSFRAN, E.; ABRAHãO, S. Usability evaluation methods for the web: A systematic mapping study. *Information and Software Technology*, v. 53, n. 8, p. 789–817, Aug 2011. ISSN 09505849. Citado na página 79.

FERRE, X. et al. Usability basics for software developers. *IEEE Software*, v. 18, n. 1, p. 22–29, Jan 2001. ISSN 0740-7459. Citado na página 84.

FERREIRA, C. P. et al. Siac – sistema inteligente de avaliaÇÃo do conhecimento – sae mÓdulo testes. p. 94, 2011. Citado na página 45.

FERREIRA FILHO, R. C. M. Estratégia de elaboração de projetos de engenharia em sistema tutor inteligente. *Informática na educação: teoria prática*, v. 11, n. 2, Dec 2008. ISSN 1982-1654, 1516-084X. Disponível em: <https://seer.ufrgs.br/InfEducTeoriaPratica/article/view/11584>. Citado 7 vezes nas páginas 13, 22, 33, 34, 35, 36 e 37.

FONSECA, A. G. Mendes Fernandes da. Aprendizagem, mobilidade e convergÊncia: Mobile learning com celulares e smartphones. $Revista_{M}\acute{i}dia_{eC}otidiano, v.\ 2, n.\ 2, p.\ 265, Jun 2013. ISSN 2178-602X. Citado na página 23.$

FOUNDATION. *The Community for Open Innovation and Collaboration | The Eclipse Foundation*. 2022. Disponível em: <https://www.eclipse.org/home/>. Citado na página 71.

FURTADO, A.; RISSOLI, V. Tecnologia "inteligente" associada a aprendizagem significativa em bioquímica. In: *Anais do XXV Workshop de Informática na Escola (WIE 2019)*. Brazilian Computer Society (Sociedade Brasileira de Computação - SBC),

2019. p. 560. Disponível em: <https://br-ie.org/pub/index.php/wie/article/view/8547>. Citado na página 21.

GASIMOV, A. et al. Visiting mobile application development: What, how and where. In: *2010 Ninth International Conference on Mobile Business and 2010 Ninth Global Mobility Roundtable (ICMB-GMR)*. IEEE, 2010. p. 74–81. ISBN 978-1-4244-7423-3. Disponível em: <http://ieeexplore.ieee.org/document/5494784/>. Citado na página 27.

GENARI, J. O. S.; FERRARI, F. C. Times de alto desempenho no contexto das metodologias scrum e kanban. p. 9, 2015. Citado na página 58.

GERHARDT, T. E.; SILVEIRA, D. T. Métodos de pesquisa. 2009. Citado 2 vezes nas páginas 25 e 50.

GIRAFFA, L. M. M. Uma arquitetura de tutor utilizando estados mentais. p. 177, May 1999. Citado 2 vezes nas páginas 34 e 37.

GIRAFFA, L. M. M.; VICCARI, R. M. Fundamentos dos sistemas tutores inteligentes. 2001. Citado 3 vezes nas páginas 35, 36 e 37.

GIRON, J. E.; MENDOZA, S.; TORRES-HUITZIL, C. Mechanism for dynamic deployment of plastic mobile cross-platform user interfaces. In: *2011 8th International Conference on Electrical Engineering, Computing Science and Automatic Control*. IEEE, 2011. p. 1–5. ISBN 978-1-4577-1013-1. Disponível em: <http://ieeexplore.ieee.org/document/6106613/>. Citado na página 29.

GIT. *Git*. 2020. Disponível em: <https://git-scm.com/>. Citado na página 64.

GONçALVES, J. P. *A integração de Testes Adaptativos Informatizados e Ambientes Computacionais de Tarefas para o aprendizado do inglês instrumental*. Tese (Doutorado) — Universidade de São Paulo, Mar 2004. Disponível em: <http://www.teses.usp.br/teses/disponiveis/55/55134/tde-03052004-160334/>. Citado 2 vezes nas páginas 45 e 46.

HARTLEY, J. The design and evaluation of an adaptive teaching system. *International Journal of Man-Machine Studies*, v. 5, n. 3, p. 421–436, Jul 1973. ISSN 00207373. Citado na página 33.

HARTLEY, J.; SLEEMAN, D. Towards more intelligent teaching systems. *International Journal of Man-Machine Studies*, v. 5, n. 2, p. 215–236, Apr 1973. ISSN 00207373. Citado na página 33.

HEITKöTTER, H.; HANSCHKE, S.; MAJCHRZAK, T. A. Evaluating cross-platform development approaches for mobile applications. In: _____. *Web Information Systems and Technologies*. Springer Berlin Heidelberg, 2013. (Lecture Notes in Business Information Processing, v. 140), p. 120–138. ISBN 978-3-642-36607-9. Disponível em: <http://link.springer.com/10.1007/978-3-642-36608-6_8>. Citado 2 vezes nas páginas 30 e 31.

HOLZER, A.; ONDRUS, J. Mobile application market: A developer's perspective. *Telematics and Informatics*, v. 28, n. 1, p. 22–31, Feb 2011. ISSN 07365853. Citado 2 vezes nas páginas 11 e 28.

HOLZINGER, A. Usability engineering methods for software developers. *Communications of the ACM*, v. 48, n. 1, p. 71–74, Jan 2005. ISSN 0001-0782, 1557-7317. Citado na página 84.

IBM. Native, web or hybrid mobile-app development. p. 12, 2012. Citado na página 30.

IBM. *What is Java Spring Boot?* 2021. Disponível em: <https://www.ibm.com/cloud/learn/java-spring-boot>. Citado na página 70.

INUKOLLU, V. et al. Factors influncing quality of mobile apps: Role of mobile app development life cycle. *International Journal of Software Engineering Applications*, v. 5, 10 2014. Citado na página 29.

ISO/IEC-25010. Systems and software engineering – square - software product quality requirements and evaluation – system and software quality models. *International Organization for Standardization*, 2011. Citado na página 79.

JONASSEN, D. H.; WANG, S. The physics tutor: Integrating hypertext and expert systems. *Journal of Educational Technology Systems*, v. 22, n. 1, p. 19–28, Sep 1993. ISSN 0047-2395, 1541-3810. Citado na página 35.

KEDAH, M. School of C. U. U. M.; OLUWATOSIN, H. S. Client-server model. *IOSR Journal of Computer Engineering*, v. 16, n. 1, p. 57–71, 2014. ISSN 22788727, 22780661. Citado 2 vezes nas páginas 11 e 69.

KETOLA, P.; RöYKKEE, M. The three facets of usability in mobile handsets. p. 8, 2001. Citado na página 79.

LEITE, B. S. M-learning: o uso de dispositivos móveis como ferramenta didática no ensino de química. *Revista Brasileira de Informática na Educação*, v. 22, n. 03, p. 55, Dec 2014. ISSN 1414-5685, 1414-5685. Citado na página 23.

MAHNIC, V.; DRNOVSCEK, S. Agile software project management with scrum. p. 7, 2005. Citado 2 vezes nas páginas 55 e 56.

MARGHESCU, G.; CHICIOREANU, T. D.; MARGHESCU, I. An alternative to the traditional methods in education - m-learning: a glance into the future. In: *EUROCON 2007 - The International Conference on "Computer as a Tool"*. IEEE, 2007. p. 2410–2414. ISBN 978-1-4244-0812-2. Disponível em: <http://ieeexplore.ieee.org/document/4400623/>. Citado na página 23.

MARTHASARI, G.; SUHARSO, W.; ARDIANSYAH, F. A. Personal extreme programming with moscow prioritization for developing library information system. *Proceeding of the Electrical Engineering Computer Science and Informatics*, v. 5, n. 5, p. 537–541, Nov 2018. ISSN 2407-439X, 2407-439X. Citado na página 68.

MATTEI, C. O prazer de aprender com a informÁtica na educaÇÃo infantil. p. 15, 2013. Citado na página 21.

MEIRELLES, P. et al. A students' perspective of native and cross-platform approaches for mobile application development. In: _____. *Computational Science and Its Applications – ICCSA 2019*. Springer International Publishing, 2019. (Lecture Notes in Computer Science, v. 11623), p. 586–601. ISBN 978-3-030-24307-4. Disponível em: <http://link.springer.com/10.1007/978-3-030-24308-1_47>. Citado na página 31.

MISRA, S. et al. Agile software development practices: evolution, principles, and criticisms. *International Journal of Quality Reliability Management*, v. 29, n. 9, p. 972–980, Oct 2012. ISSN 0265-671X. Citado na página 55.

MORESI, E. Metodologia de pesquisa. p. 108, 2003. Citado 2 vezes nas páginas 49 e 50.

MOURA. GeraÇÃo mÓvel: Um ambiente de aprendizagem suportado por tecnologias mÓveis para a "geraÇÃo polegar". p. 29, 2009. Citado na página 22.

NGUYEN, L. N. K. Application of protocol-oriented mvvm architecture in ios development. p. 52, 2017. Citado na página 63.

NIRANJANAMURTHY, M. et al. Research study on importance of usability testing / user experience (ux) testing. In: . [S.l.: s.n.], 2014. Citado na página 84.

NONNENMACHER, R. F. P.; MENEZES, D. C. Estudo do comportamento do consumidor de aplicativos mÓveis. p. 70, 2012. Citado na página 27.

NUNES, C. F. D.; SANTOS, G. A.; RISSOLI, V. R. V. Tecnologia 'inteligente' no apoio ao processo educacional em engenharia. In: *Anais do XXV Workshop de Informática na Escola (WIE 2019)*. [S.l.]: International Symposium on Project Approaches in Engineering), 2018. p. 839–847. Citado na página 21.

ORACLE. *O que é Java e por que preciso dele?* 2022. Disponível em: <https://www.java.com/pt-BR/download/help/whatis_java.html>. Citado na página 70.

PEREIRA, D. C. d. F.; AGUIAR, D. R. d. C.; SARAIVA, R. d. N. G. Bdq - banco de dados de questões. 2007. Citado na página 42.

PREZOTTO, E. D.; BONIATI, B. B. Estudo de frameworks multiplataforma para desenvolvimento de aplicações mobile híbridas. n. 1, p. 8, 2014. Citado na página 30.

RAABE, A. L. A. Uma proposta de arquitetura de sistema tutor inteligente baseada na teoria das experiências de aprendizagem mediadas. *Informática na educação: teoria prática*, v. 8, n. 2, Nov 2005. ISSN 1982-1654, 1516-084X. Disponível em: <https://seer.ufrgs.br/InfEducTeoriaPratica/article/view/9718>. Citado 3 vezes nas páginas 34, 36 e 37.

RISSOLI, V. R. V. Universidade federal do rio grande do sul centro interdisciplinar de novas tecnologias na educaÇÃo programa de pÓs-graduaÇÃo em informÁtica na educaÇÃo. p. 237, 2007. Citado 4 vezes nas páginas 11, 37, 39 e 40.

RISSOLI, V. R. V.; SANTOS, G. A. Um assistente inteligente fuzzy no acompanhamento da aprendizagem significativa. p. 12, 2011. Citado 5 vezes nas páginas 11, 38, 39, 41 e 45.

RISSOLI, V. R. V.; SANTOS, G. A. O agente pedagógico animado mina. In: . [s.n.], 2013. Disponível em: <http://www.br-ie.org/pub/index.php/sbie/article/view/2561>. Citado 5 vezes nas páginas 11, 39, 40, 43 e 44.

SANTOS, G. A.; RISSOLI, V. R. V. Benefícios no uso de um assistente inteligente no ensino-aprendizagem de programação computacional. *Anais do XXII Simpósio Brasileiro de Informática na Educação. Aracaju*, p. 2244–2253, 2011. Citado na página 21.

SCHWABER, K. *Agile project management with Scrum*. [S.l.]: Microsoft Press, 2004. ISBN 978-0-7356-1993-7. Citado 3 vezes nas páginas 11, 56 e 57.

SERZEDELLO, N. T. B.; TOMAéL, M. I. Produção tecnológica da universidade estadual de londrina (uel): Mapeamento da área de ciências agrárias pela plataforma lattes. *AtoZ: novas práticas em informação e conhecimento*, v. 1, n. 1, p. 23, Jun 2011. ISSN 2237-826X. Citado 3 vezes nas páginas 13, 50 e 51.

SHAH, K.; SINHA, H.; MISHRA, P. Analysis of cross-platform mobile app development tools. In: *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*. IEEE, 2019. p. 1–7. ISBN 978-1-5386-8075-9. Disponível em: <https://ieeexplore.ieee.org/document/9033872/>. Citado 2 vezes nas páginas 30 e 31.

SHOLICHIN, F. et al. Review of ios architectural pattern for testability, modifiability, and performance quality. . *Vol.*, n. 15, p. 16, 2005. Citado na página 63.

SILVA, D. V. d. S.; SANTOS, F. A. d. O.; NETO, P. S. Os benefícios do uso de kanban na gerência de projetos de manutenção de software. In: . [S.l.]: VIII Simpósio Brasileiro de Sistemas de Informação, 2012. Citado na página 58.

SOARES, V. C.; RISSOLI, V. R. V. Agente inteligente no apoio ao ensino-aprendizagem. p. 4, 2011. Citado na página 45.

SOMMERVILLE, I. *Engenharia de software*. [S.l.]: Pearson Prentice Hall, 2011. ISBN 978-85-7936-108-1. Citado 3 vezes nas páginas 11, 56 e 57.

SOUSA, F. V. F. d. S. Software solatium®: Processo de enfermagem para o cuidado de conforto de pessoas com adoecimento cardiovascular. 2018. Disponível em: <http://siduece.uece.br/siduece/trabalhoAcademicoPublico.jsf?id=82418>. Citado na página 49.

SOUZA, L.; SANTOS, G. A.; RISSOLI, V. Sae - sistema de apoio educacional. In: *Anais dos Workshops do VIII Congresso Brasileiro de Informática na Educação (CBIE 2019)*. Brazilian Computer Society (Sociedade Brasileira de Computação - SBC), 2019. p. 1413. Disponível em: <https://br-ie.org/pub/index.php/wcbie/article/view/9109>. Citado 3 vezes nas páginas 21, 38 e 42.

STATCOUNTER. *Mobile Operating System Market Share Worldwide*. 2020. Disponível em: <https://gs.statcounter.com/os-market-share/mobile/worldwide>. Citado 2 vezes nas páginas 23 e 31.

STATISTA. *Number of smartphone subscriptions worldwide from 2016 to 2021, with forecasts from 2022 to 2027*. 2022. Disponível em: <https://www.statista.com/statistics/330695/number-of-smartphone-users-worldwide/>. Citado na página 29.

STELLMAN, A.; GREENE, J. *Learning Agile*. First edition. [S.l.]: O'Reilly, 2014. ISBN 978-1-4493-3192-4. Citado na página 58.

TEIXEIRA, J. M. R.; JúNIOR, N. N. P.; RHODEN, V. R. Pró-reitoria de graduação curso de ciência da computação trabalho de conclusão de curso. p. 117, 2015. Citado 2 vezes nas páginas 45 e 46.

TRELLO. *About | What is Trello?* 2020. Disponível em: <https://trello.com/about>. Citado na página 64.

URRETAVIZCAYA, M. Sistemas inteligentes en el ambito de la educacion. *INTELIGENCIA ARTIFICIAL*, v. 5, n. 12, p. 292, Apr 2001. ISSN 1988-3064, 1137-3601. Citado na página 35.

VALENTIM, N. M. C. et al. Avaliando a qualidade de um aplicativo web móvel através de um teste de usabilidade: um relato de experiência. In: *Anais do XIII Simpósio Brasileiro de Qualidade de Software (SBQS 2014)*. Brasil: Sociedade Brasileira de Computação - SBC, 2014. p. 256–263. Disponível em: <https://sol.sbc.org.br/index.php/sbqs/article/view/15258>. Citado na página 81.

YACEF, K. Intelligent teaching assistant systems. In: *International Conference on Computers in Education, 2002. Proceedings.* IEEE Comput. Soc, 2002. v. 1, p. 136–140. ISBN 978-0-7695-1509-0. Disponível em: <http://ieeexplore.ieee.org/document/1185885/>. Citado 2 vezes nas páginas 38 e 40.

Apêndices

APÊNDICE A – Simulação da Proposta

Com o propósito de validar, ainda que de modo inicial, a viabilidade da proposta apresentada no TCC1, foi desenvolvido uma simulação com a implementação de parte da aplicação, seguindo a metodologia e a arquitetura propostas nas seções 4.1 e 4.1.1.5, respectivamente, a fim de averiguar a coerência da proposta e sua viabilidade.

Na Figura 30 pode ser observado um panorama geral do aplicativo desenvolvido, em que o estudante do SAE interage com a interface iOS e recebe na tela do seu dispositivo informações e instruções para realizar as operações que desejar. Essa interface realiza solicitações ao servidor SAE, que a responde.

Figura 30 – Panorama geral da aplicação. Fonte: Autor.

Assim, visando a viabilidade, foi planejado a execução da história de usuário "US01 - Eu, como aluno, gostaria de realizar *login* no aplicativo para conseguir utilizar suas funcionalidades", pertencente a *feature* "F01 - Autenticação". O foco se deu no desenvolvimento da funcionalidade, sem grandes preocupações com a interface, visto que o propósito era de validar a viabilidade do projeto usando a metodologia proposta.

No desenvolvimento da história US01, foram criados os seguintes elementos:

- **LoginView:** pertence à camada *View* e nesta classe foram implementados os elementos da interface;
- **LoginCredentials:** pertence à camada *Model* e contém os dados de *e-mail* e senha, usados para realizar *login* no SAE;
- **LoginViewModel:** pertence à camada *ViewModel* e contém a lógica de negócio utilizada no *login*;

- **NetworkingService**: serviço que contém classes de comunicação do aplicativo com o servidor.

Na Figura 31 pode ser observado o diagrama de classes que contempla a história de usuário:

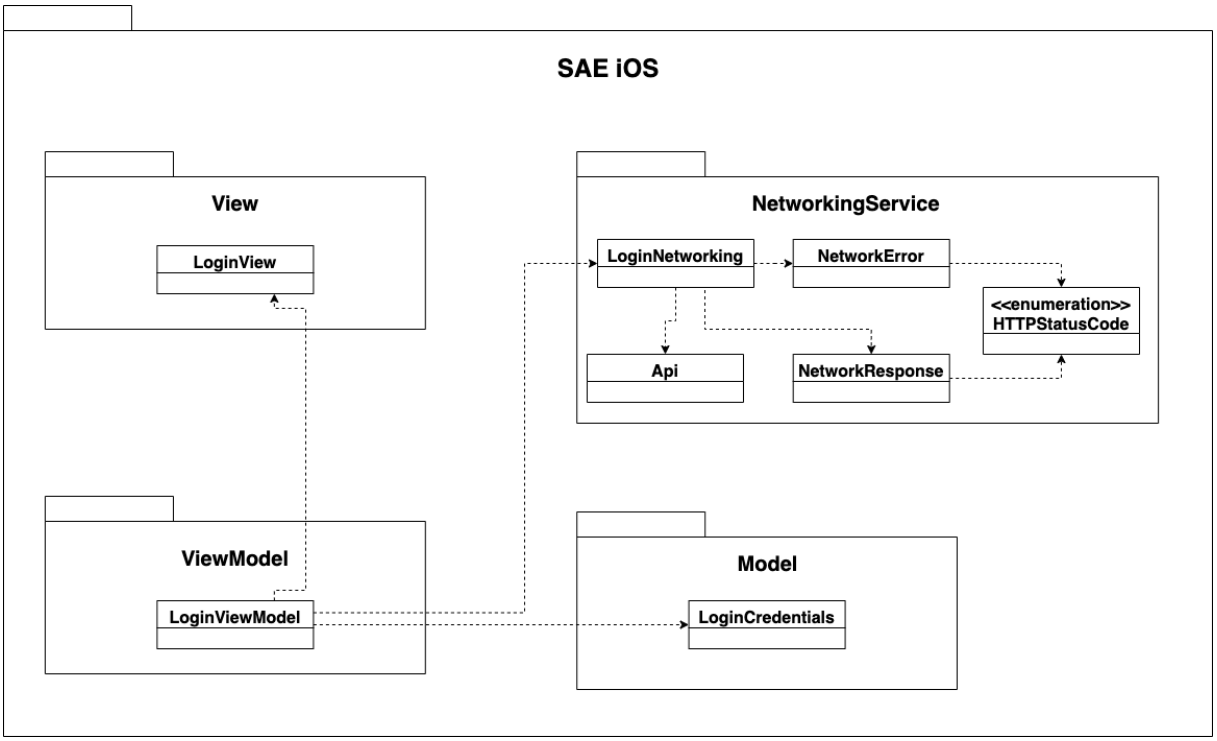

Figura 31 – Diagrama de classes do *login*. Fonte: Autor.

Nas imagens da Figura 32 observa-se a implementação da história de usuário US01:

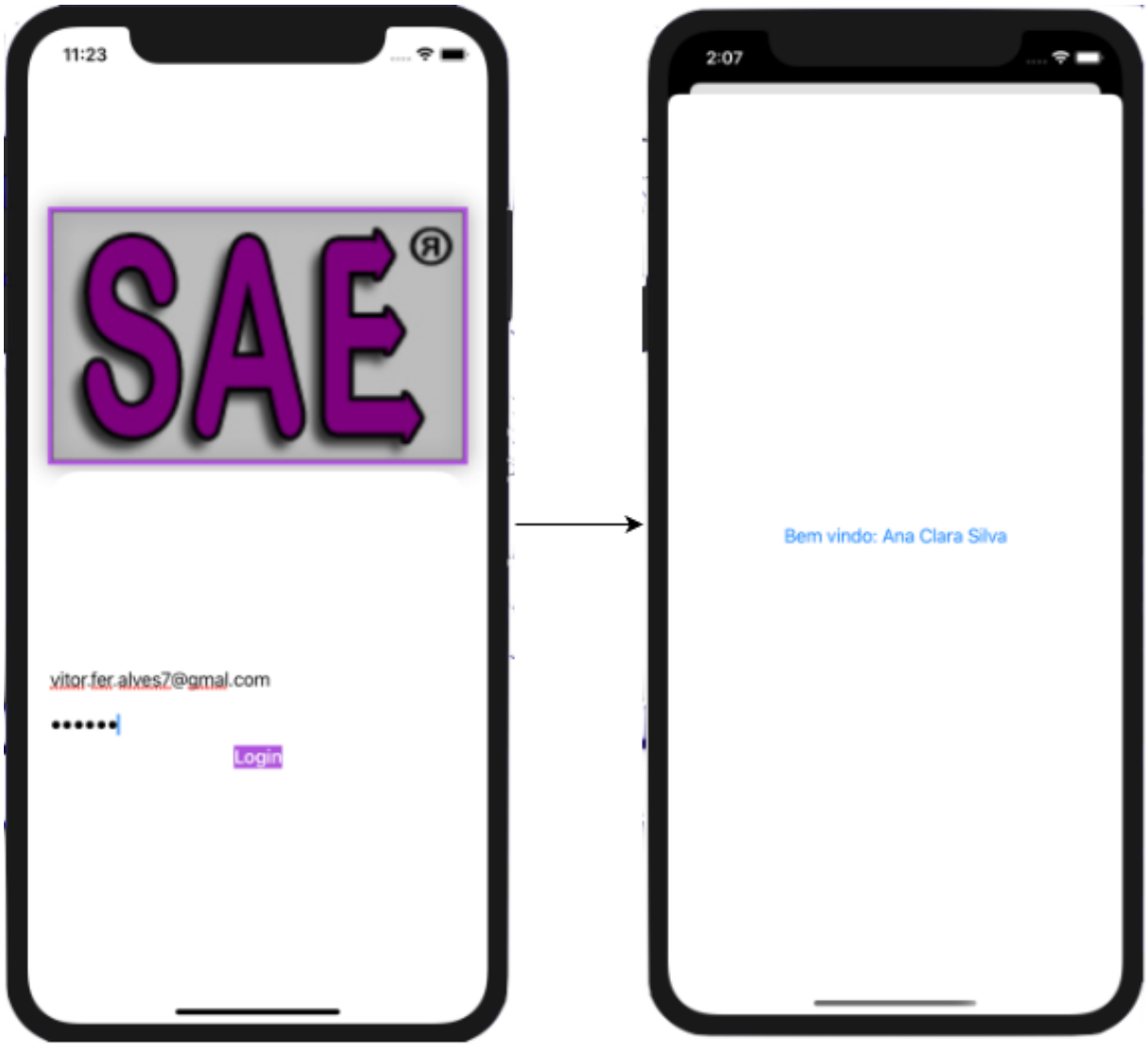

Figura 32 – Fluxo da funcionalidade de *login*. Fonte: Autor.

APÊNDICE B – Fluxos Alternativos

B.1 *Logout*

Na Figura 33, pode ser observado o fluxo de *logout*, implementado ao longo da primeira *sprint*. Na primeira imagem, assim que o usuário clica no ícone no canto superior direito da tela, aparece uma janela com duas opções: cancelar, que, consequentemente, fecha a janela, e sair, que leva o usuário para a tela de *login* novamente, como pode ser observado na segunda imagem.

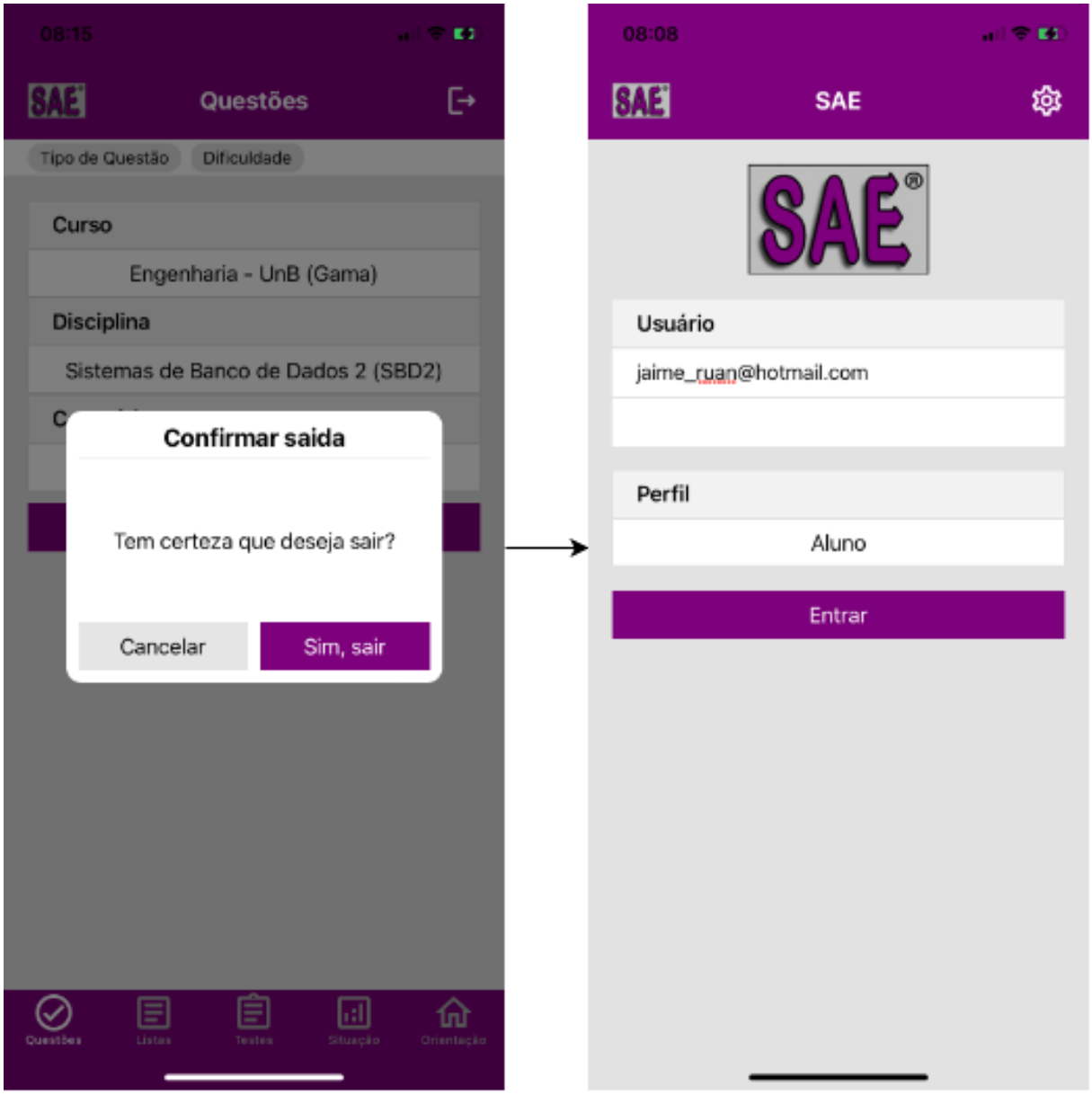

Figura 33 – Fluxo de *logout*. Fonte: Autor.

B.2 Tipos de Questões

Como apresentado na seção 2.2.4.2.1, o módulo de questões contém cinco tipos de questões diferentes, sendo a questão de verdadeiro ou falso apresentada na seção 5.4. Na Figura 34, pode ser observado as questões múltipla escolha, escolha múltipla, preencher lacuna e questão aberta, respectivamente.

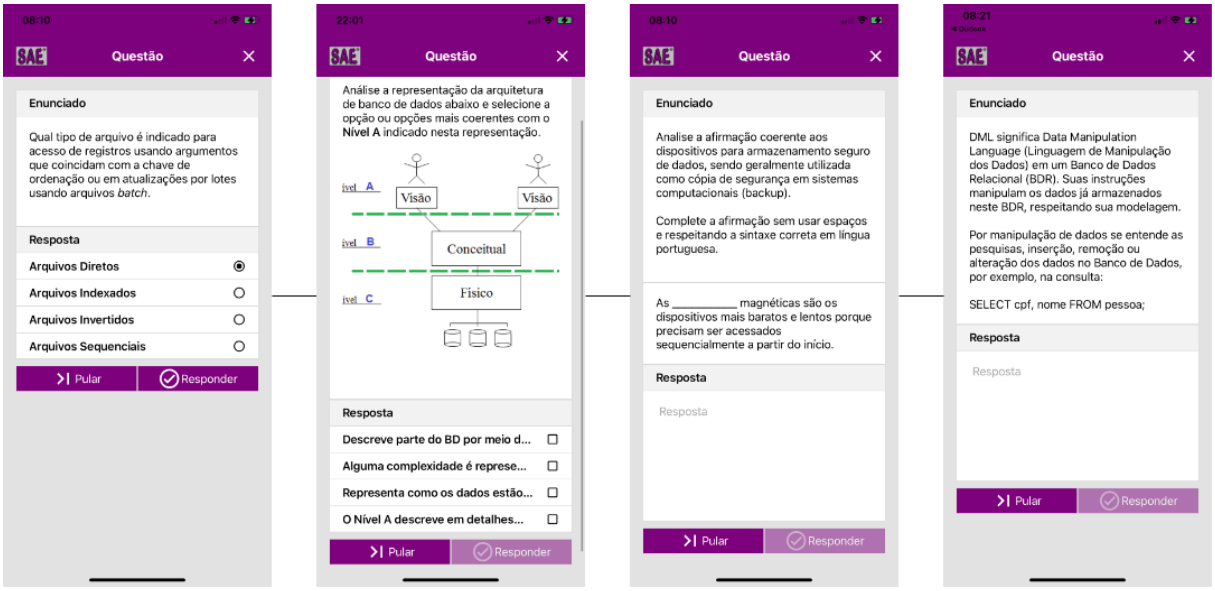

Figura 34 – Diferentes tipos de questões. Fonte: Autor.

Ainda na Figura 34, observa-se, na segunda imagem, que o aplicativo desenvolvido também é capaz de apresentar imagens no enunciado de qualquer questão.

B.3 Continuar ou Desistir de um Teste

Na imagem central da Figura 35, observa-se que o usuário tem duas opções em relação a um teste previamente iniciado: desistir, nesse caso o usuário é redirecionado para a tela central do módulo de testes, conforme pode ser visto na imagem à direita, ou continuar, conforme pode ser visto na imagem à esquerda, o usuário é redirecionado para responder novas questões do teste.

B.4 Modelo do formulário usado no teste de usabilidade

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)

Você está sendo convidado(a) para participar, como voluntário(a), em uma pesquisa científica. Caso você não queira participar, não há problema algum. Você não precisa me explicar porque, e não haverá nenhum tipo de punição por isso. Você tem todo o direito de não querer participar do estudo, basta selecionar a opção correspondente no final desta página.

Para confirmar sua participação você precisará ler todo este documento, preencher as perguntas e assinar ao final dele. Este documento se chama TCLE (Termo de Consentimento livre e esclarecido). Nele estão contidas as principais informações sobre o estudo, objetivos, metodologias, riscos e benefícios, dentre outras informações.

Este TCLE se refere ao projeto de pesquisa Aplicativo para o sistema operacional iOS integrado a um Sistema Tutor Inteligente para apoio à Educação. Para ter uma cópia deste TCLE você deverá imprimi-lo, ou deverá gerar uma cópia em pdf para guardá-lo em seu computador. Você também poderá solicitar aos pesquisadores do estudo uma versão deste documento a qualquer momento por um dos e-mails registrados no final deste termo.

A pesquisa será realizada por meio de 13 perguntas. A precisão de suas respostas é determinante para a qualidade da pesquisa.

Você não será remunerado, visto que sua participação nesta pesquisa é de caráter voluntário. Caso decida desistir da pesquisa você poderá interromper o questionário e sair do estudo a qualquer momento, sem nenhuma restrição ou punição.

Os pesquisadores garantem e se comprometem com o sigilo e a confidencialidade de todas as informações fornecidas por você para este estudo. Da mesma forma, o tratamento dos dados coletados seguirá as determinações da Lei Geral de Proteção de Dados (LGPD – Lei 13.709/18).

Para contatar um dos pesquisadores da pesquisa, você poderá encaminhar um e-mail, ligar ou mandar mensagem pelo WhatsApp para eles a qualquer momento:

Nome, celular e e-mail do Pesquisador Responsável: Luis Gustavo Avelino de Lima Jacinto; (61) 98273-1122; luis.gustavo3013@hotmail.com

INTERFACE

Q1 - Considero as cores e botões do aplicativo agradáveis

- ☐ Satisfeito
- ☐ Nem satisfeito nem insatisfeito
- ☐ Insatisfeito

Q2 - Consigo visualizar bem todos os botões e informações dentro do aplicativo

- ☐ Satisfeito
- ☐ Nem satisfeito nem insatisfeito
- ☐ Insatisfeito

Q3 - Entendo com facilidade as palavras, nomenclaturas e informações dentro do aplicativo

- ☐ Satisfeito
- ☐ Nem satisfeito nem insatisfeito
- ☐ Insatisfeito

Q4 - As imagens e ícones no aplicativo são de fácil reconhecimento

- ☐ Satisfeito
- ☐ Nem satisfeito nem insatisfeito
- ☐ Insatisfeito

Q5 - Consigo visualizar todas as funcionalidades dentro do aplicativo

- ☐ Satisfeito
- ☐ Nem satisfeito nem insatisfeito
- ☐ Insatisfeito

Q6 - Consigo navegar bem por todas as telas do aplicativo

- ☐ Satisfeito
- ☐ Nem satisfeito nem insatisfeito
- ☐ Insatisfeito

UTILIDADE

Q7 - Considero o aplicativo útil para melhorar meu aprendizado

- ☐ Satisfeito
- ☐ Nem satisfeito nem insatisfeito
- ☐ Insatisfeito

Q8 - Considero que o aplicativo melhoraria minha produtividade para realização das atividades e aprendizado

- ☐ Satisfeito

- ☐ Nem satisfeito nem insatisfeito
- ☐ Insatisfeito

Q9 - Considero que o aplicativo facilitaria a realização das minhas atividades

- ☐ Satisfeito
- ☐ Nem satisfeito nem insatisfeito
- ☐ Insatisfeito

FACILIDADE DE USO

Q10 - Foi fácil aprender a utilizar o aplicativo

- ☐ Satisfeito
- ☐ Nem satisfeito nem insatisfeito
- ☐ Insatisfeito

Q11 - Eu conseguia entender o que acontecia durante o uso do aplicativo

- ☐ Satisfeito
- ☐ Nem satisfeito nem insatisfeito
- ☐ Insatisfeito

Q12 - É fácil de lembrar como utilizar o aplicativo

- ☐ Satisfeito
- ☐ Nem satisfeito nem insatisfeito
- ☐ Insatisfeito

Q13 - Considero o aplicativo fácil de utilizar

- ☐ Satisfeito
- ☐ Nem satisfeito nem insatisfeito
- ☐ Insatisfeito

COMENTÁRIOS

-

CONSENTIMENTO DE PARTICIPAÇÃO

Eu, concordo em participar voluntariamente do presente estudo como participante. O pesquisador me informou sobre tudo o que vai acontecer na pesquisa, o que terei que fazer, inclusive sobre os possíveis riscos e benefícios envolvidos na minha participação. O pesquisador me garantiu que eu poderei sair da pesquisa a qualquer momento, sem dar nenhuma explicação, e que esta decisão não me trará nenhum tipo de penalidade ou interrupção de meu tratamento.

Fui informado também que devo imprimir ou gerar um pdf do TCLE para ter a minha cópia do TCLE e que posso solicitar uma versão dele via e-mail para os pesquisadores.

Nome Completo:
Data de Nascimento:
Profissão:
Instituição:
Sexo:
Já usou o SAE ou não:

_____

Assinatura do participante

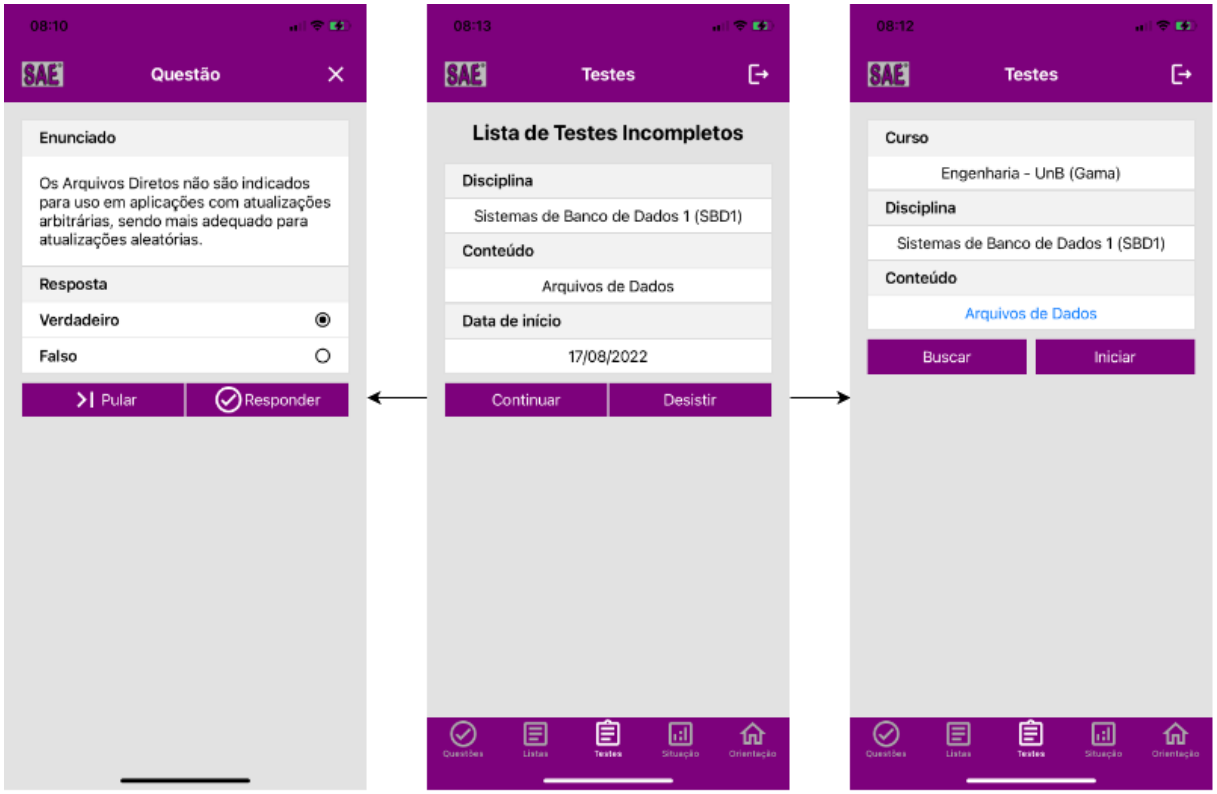

Figura 35 – Fluxo de continuar ou desistir de um teste. Fonte: Autor.

B.5 Resultados do teste de usabilidade

Como apresentado na seção 5.5, o teste de usabilidade foi realizado com dois grupos de usuários (personas): persona 1 que é um estudante de Engenharia de Software que está familiarizado com dispositivo móveis e já conhece e usou o SAE web e a persona 2 que é um estudante do ensino fundamental ou médio que é bastante familiarizado com dispositivos móveis mas não conhece o SAE.

A partir das perguntas definidas no modelo do teste de usabilidade, apresentado no Apêndice B.4, pode-se observar a seguir as respostas do teste, em que a Tabela 10 apresenta as respostas referentes a persona 1, a Tabela 11 apresenta as respostas referentes a persona 2 e, por fim, a Tabela 12 apresenta as respostas de ambas personas em conjunto.

Tema	Pergunta	S	M	I
Interface	Q1	100% (3)	0% (0)	0% (0)
	Q2	66% (2)	34% (1)	0% (0)
	Q3	100% (3)	0% (0)	0% (0)
	Q4	100% (3)	0% (0)	0% (0)
	Q5	100% (3)	0% (0)	0% (0)
	Q6	100% (3)	0% (0)	0% (0)
Utilidade	Q7	100% (3)	0% (0)	0% (0)
	Q8	100% (3)	0% (0)	0% (0)
	Q9	100% (3)	0% (0)	0% (0)
Facilidade de Uso	Q10	100% (3)	0% (0)	0% (0)
	Q11	100% (3)	0% (0)	0% (0)
	Q12	100% (3)	0% (0)	0% (0)
	Q13	100% (3)	0% (0)	0% (0)

Tabela 10 – Respostas das perguntas teste de usabilidade dos usuários da persona 1. Fonte: Autor.

Tema	Pergunta	S	M	I
Interface	Q1	100% (3)	0% (0)	0% (0)
	Q2	100% (3)	0% (0)	0% (0)
	Q3	100% (3)	0% (0)	0% (0)
	Q4	100% (3)	0% (0)	0% (0)
	Q5	100% (3)	0% (0)	0% (0)
	Q6	100% (3)	0% (0)	0% (0)
Utilidade	Q7	100% (3)	0% (0)	0% (0)
	Q8	100% (3)	0% (0)	0% (0)
	Q9	100% (3)	0% (0)	0% (0)
Facilidade de Uso	Q10	66% (2)	34% (1)	0% (0)
	Q11	66% (2)	34% (1)	0% (0)
	Q12	66% (2)	34% (1)	0% (0)
	Q13	66% (2)	34% (1)	0% (0)

Tabela 11 – Respostas das perguntas teste de usabilidade dos usuários da persona 2. Fonte: Autor.

Tema	Pergunta	S	M	I
Interface	**Q1**	**100% (6)**	**0% (0)**	**0% (0)**
	Q2	**83% (5)**	**17% (1)**	**0% (0)**
	Q3	**100% (6)**	**0% (0)**	**0% (0)**
	Q4	**100% (6)**	**0% (0)**	**0% (0)**
	Q5	**100% (6)**	**0% (0)**	**0% (0)**
	Q6	**100% (6)**	**0% (0)**	**0% (0)**
Utilidade	**Q7**	**100% (6)**	**0% (0)**	**0% (0)**
	Q8	**100% (6)**	**0% (0)**	**0% (0)**
	Q9	**100% (6)**	**0% (0)**	**0% (0)**
Facilidade de Uso	**Q10**	**83% (5)**	**17% (1)**	**0% (0)**
	Q11	**83% (5)**	**17% (1)**	**0% (0)**
	Q12	**83% (5)**	**17% (1)**	**0% (0)**
	Q13	**83% (5)**	**17% (1)**	**0% (0)**

Tabela 12 – Respostas das perguntas teste de usabilidade dos usuários de ambas personas. Fonte: Autor.