

**UNIVERSIDADE FEDERAL DE SANTA CATARINA
CAMPUS ARARANGUÁ**

BRUNO DE OLIVEIRA MACHADO

**AUTOMATIZAÇÃO DE CONTEÚDOS NUM SISTEMA TUTOR INTELIGENTE
PARA O ENSINO-APRENDIZAGEM DE PROGRAMAÇÃO**

Araranguá, Novembro de 2014

BRUNO DE OLIVEIRA MACHADO

**AUTOMATIZAÇÃO DE CONTEÚDOS NUM SISTEMA TUTOR INTELIGENTE
PARA O ENSINO-APRENDIZAGEM DE PROGRAMAÇÃO**

Trabalho de Conclusão de Curso submetido à
Universidade Federal de Santa Catarina para a
obtenção do Grau de Bacharel em Tecnologias da
Informação e Comunicação.

Orientadora: Prof^a Dr^a Eliane Pozzebon

Araranguá, Novembro de 2014

BRUNO DE OLIVEIRA MACHADO

**AUTOMATIZAÇÃO DE CONTEÚDOS NUM SISTEMA TUTOR INTELIGENTE
PARA O ENSINO-APRENDIZAGEM DE PROGRAMAÇÃO**

Este Trabalho de Conclusão de Curso foi julgado aprovado para a obtenção do Título de “Bacharel em Tecnologias da Informação e Comunicação”, e aprovado em sua forma final pelo Curso de Graduação em Tecnologias da Informação e Comunicação.

Araranguá, Novembro de 2014.


Prof. Vilson Gruber
Coordenador do Curso

Banca examinadora:


Profª. Drª. Eliane Pozzebon (orientadora)


Prof. Dr. Alexandre Leopoldo Gonçalves


Profª. Drª. Luciana Bolan Frigo

Dedico este trabalho a todos aqueles que, direta ou indiretamente, me apoiaram e proporcionaram as condições necessárias para execução do mesmo.

AGRADECIMENTOS

Agradeço à minha família pelo apoio incondicional durante todo o período que estive trabalhando neste projeto.

Aos meus colegas de faculdade, não apenas por fazerem parte do processo de formação acadêmica com o compartilhamento de diversas experiências vivenciadas durante o legado acadêmico, mas também por sua amizade e companheirismo que sempre me motivaram.

À professora Eliane por mostrar-se sempre disponível, apoiando de forma indispensável à concepção e desenvolvimento deste.

À Universidade Federal de Santa Catarina, por ofertar gratuitamente o curso de Tecnologias da Informação e Comunicação, tornando possível a realização de um sonho de vida.

À Deus, por ser uma força motivadora que me encoraja a nunca desistir de meus objetivos.

RESUMO

Neste trabalho foi desenvolvido um Sistema Tutor Inteligente (STI) para o sistema operacional *Android*, onde os aprendizes com um *tablet* ou *smartphone* aceitam o desafio de responder uma sequência de perguntas para aperfeiçoar o nível de conhecimento em programação de computadores. O STI adapta automaticamente o nível de dificuldade das perguntas conforme a quantidade de acertos e erros dos aprendizes. O mesmo ocorre com o nível de conhecimento do aprendiz que é atualizado conforme seu desempenho ao responder as questões. Outro aspecto relevante deste trabalho é a inclusão do conteúdo pelo professor ou especialista. O professor pode incluir perguntas e explicações referentes à programação de computadores e o STI automaticamente seleciona a pergunta adequada ao aprendiz conforme seu nível de conhecimento. Para a validação do STI foi utilizado o método de observação direta e um questionário, onde os aprendizes responderam como foi a experiência na utilização do STI. Os resultados obtidos foram considerados positivos e corresponderam as expectativas do projeto.

Palavras-chave: Inteligência Artificial, Sistemas Multiagentes, Sistema Tutor Inteligente, Dispositivos móveis, Programação de Computadores.

ABSTRACT

In this work was developed an Intelligent Tutoring System (ITS) for the Android operating system, where learners with a tablet or smartphone accept the challenge of answering a sequence of questions to improve the level of knowledge in computer programming. The ITS automatically adapts the level of difficulty of the questions depending on the amount of hits and misses of apprentices. The same occurs with the knowledge level of the learner that is updated according to their performance in answering the questions. Another important aspect of this work is the inclusion of content by the teacher or specialist. The teacher can include questions and explanations relating to computer programming and STI automatically selects the appropriate question to the learner according to his level of knowledge. For the validation of the STI was used the direct observation method and a questionnaire where learners responded how was the experience in the use of the STI. The results were considered positive and came up with the expectations of the project.

Keywords: Artificial Intelligence, Multi-Agent Systems, Intelligent Tutor System, Mobile Devices, Computer Programming.

LISTA DE FIGURAS

Figura 1: Áreas relacionadas com a Inteligência Artificial. Extraído e adaptado de: (NORVING; RUSSEL, 2014).....	20
Figura 2: Modelo Tradicional de Sistema Tutor Inteligente. Extraído de: (RAABE, 2005).....	24
Figura 3: Aplicativo aprenda a escrever. Extraído de: (ANDROIDPIT, 2014)...	28
Figura 4: Aplicativo Duolingo. Extraído de: (ANDROIDPIT, 2014).....	28
Figura 5: Ciclo de vida do desenvolvimento do protótipo.	30
Figura 6: Arquitetura do STI.	33
Figura 7: Diagrama de Caso de Uso	34
Figura 8: Modelo físico do banco de dados.....	35
Figura 9: Comunicação entre Usuário e o Servidor.....	37
Figura 10: Tela de login do STI	39
Figura 11: Tela do Menu do Administrador.	41
Figura 12: Tela de Manutenção de Perguntas.	43
Figura 13: Tela de Inserção de Explicação.....	44
Figura 14: Tela da avaliação.	46
Figura 15: Questionamento sobre visualização da explicação.	47
Figura 16: Tela do resultado do teste.....	48
Figura 17: Comunicação do protocolo TCP. Extraído de: (SANTOS, 2014). ...	52
Figura 18: Transferência das portas do protocolo TCP. Extraído de: (SANTOS, 2014).....	52

LISTA DE GRÁFICOS

Gráfico 1: Opinião dos usuários quanto à utilização de tecnologias móveis no processo educacional.....	56
Gráfico 2: Opinião dos usuários quanto a aplicação de IA em uma avaliação. 57	
Gráfico 3: Opinião dos usuários quanto à interface do STI.	58
Gráfico 4: Opinião dos usuários quanto ao nível obtido na avaliação.	59
Gráfico 5: Opinião dos usuários quanto à clareza e objetividade das questões.	60
Gráfico 6: Opinião dos usuários quanto ao desempenho do aplicativo.	61
Gráfico 7: Opinião dos usuários quanto às explicações das questões.	62
Gráfico 8: Opinião dos usuários quanto a usabilidade do STI.....	63
Gráfico 9: Opinião dos usuários quanto à reutilização do aplicativo após o término do curso.....	64

LISTA DE ABREVIATURAS E SIGLAS

AG – Algoritmo Genético

API – Application Programming Interfaces

FIPA - Foundation for Intelligent Physical Agents

GPS - Global Positioning System

IA - Inteligência Artificial

MEC - Ministério da Educação e Ciência

PC - Personal Computer

SDK - Software Development Kit

SMA - Sistemas Multiagentes

SO – Sistema Operacional

STI – Sistemas Tutores Inteligentes

VANT - Veículo Aéreo Não Tripulado

SUMÁRIO

1-	INTRODUÇÃO/CONTEXTUALIZAÇÃO	14
1.1-	PROBLEMÁTICA	15
1.2-	OBJETIVOS.....	16
1.2.1-	OBJETIVOS GERAIS.....	16
1.2.2-	OBJETIVOS ESPECÍFICOS	16
1.3-	JUSTIFICATIVA.....	17
1.4-	METODOLOGIA	18
2-	INTELIGÊNCIA ARTIFICIAL E TÉCNICAS	19
2.1-	INTELIGENCIA ARTIFICIAL	19
2.2-	SISTEMAS MULTIAGENTES	21
2.3-	SISTEMA TUTOR INTELIGENTE.....	23
3-	DISPOSITIVOS MÓVEIS NA EDUCAÇÃO	26
3.1-	INTRODUÇÃO E CONTEXTUALIZAÇÃO	26
3.2-	TECNOLOGIAS PARA O PRESENTE E FUTURO PARA O M- LEARNING.....	27
3.3-	AMBIENTES DE APRENDIZAGEM PARA DISPOSTIVOS MÓVEIS	27
4-	PROTÓTIPO DO SISTEMA TUTOR INTELIGENTE	29
4.1-	INTRODUÇÃO	29
4.2-	REQUISITOS	30
4.2.1-	REQUISITOS FUNCIONAIS	31
4.2.2-	REQUISITOS NÃO FUNCIONAIS	31
4.3-	MODELO DO STI.....	32
4.4-	IMPLEMENTAÇÃO	34
4.4.1-	FUNCIONAMENTO DO PROTÓTIPO	38
4.4.2-	INICIO DO APLICATIVO	38
4.4.3-	TELA DE LOGIN	38
4.4.4-	MÓDULOS.....	39
4.4.5-	NIVELAMENTO DINÂMICO.....	48
4.5-	TECNOLOGIAS UTILIZADAS.....	49
4.5.1-	SOCKETS.....	50
5-	VALIDAÇÃO DO PROTÓTIPO	53
5.1-	ESTRATÉGIAS DE VALIDAÇÃO.....	53

5.2-	RESULTADOS DA OBSERVAÇÃO DIRETA.....	54
5.3-	RESULTADOS DO QUESTIONÁRIO APLICADO	55
5.4-	CONSIDERAÇÕES DA VALIDAÇÃO	65
6-	CONSIDERAÇÕES FINAIS	66
7-	REFERÊNCIAS	68

1- INTRODUÇÃO/CONTEXTUALIZAÇÃO

A Inteligência Artificial aplicada na educação pode ser um forte aliado no processo de ensino-aprendizagem. Atualmente existem diversos estudos que tentam provar a eficiência de técnicas de IA para facilitar o processo de aprendizagem, como resultados desses estudos surgem novas ferramentas de ensino baseadas na IA.

O surgimento destas ferramentas é resultado do interesse dos pesquisadores em automatizar todo o processo de ensino e aprendizagem, tendo como objetivo facilitar o acesso ao conhecimento para todos, bem como fazer com que a aplicação se adapte ao perfil do aluno para o qual deverá ser efetuado o ensinamento (AGUIAR E HERMOSILLA, 2007).

Segundo RICKEL (1989), um STI é definido como sendo um programa computacional destinado a auxiliar o processo ensino-aprendizagem, utilizando métodos e técnicas de IA para representar o conhecimento e para conduzir a interação com o estudante. É um sistema projetado para saber o que ensinar, a quem ensinar, e como ensinar.

O desenvolvimento do STI proposto neste trabalho será por meio de um Sistema Multiagente.

Para entender um Sistema Multiagente, é necessário compreender o que é um agente, segundo a FIPA (Foundation for Intelligent Physical Agents), “um agente é uma entidade que reside em um ambiente onde interpreta dados através de sensores que refletem eventos no ambiente e executam ações que produzem efeitos no ambiente. Um agente pode ser um software ou um hardware puro...”.

Segundo Silva (2003), Sistemas Multiagentes são sistemas constituídos de múltiplos agentes que interagem ou trabalham em conjunto de forma a realizar um determinado conjunto de tarefas ou objetivos. Esses objetivos podem ser comuns a todos agentes ou não.

Neste trabalho é desenvolvido um sistema tutor inteligente com enfoque em um módulo tutor para a automatização de conteúdos. O tema proposto é continuidade do trabalho de Minatto (2013).

Para desenvolver o STI, que conta com quatro agentes: (a) um agente inteligente, que controla os níveis dos usuários e perguntas, e seleciona as perguntas para o aprendiz; (b) um agente tutor para automatização de conteúdos; (c) um agente aprendiz que realiza a avaliação; e (d) um agente administrador que insere tutores.

Num Sistema Multiagente vários agentes se comunicam e trocam informações por um objetivo em comum, para acontecer essa troca de informações é necessário um meio de comunicação entre esses agentes. Para estabelecer essa comunicação e essa troca de informação, é utilizada a comunicação via *socket*.

Socket é um mecanismo de comunicação, usado normalmente para implementar um modelo cliente/servidor, que permite a troca de mensagens entre os processos de uma máquina/aplicação servidor e de uma máquina/aplicação cliente. (SANTOS, 2014)

Com os sockets é possível implementar um Sistema Multiagente com uma base de dados remota, onde cliente (usuário) e servidor (base de dados) possuem uma comunicação online, permitindo ao usuário acesso direto as atualizações da base de dados sem necessitar atualizar o software.

A Inteligência Artificial com a utilização de Sistemas Multiagentes e as novas tecnologias como *tablets* e *smartphones* apresentam grande potencial para revolucionar as bases da educação virtual e móvel. A justificativa dada para a potencialidade destas tecnologias é que a Inteligência Artificial pode prover um conteúdo didático inteligente e objetivo para o aluno, além de que a utilização destes conteúdos em dispositivos móveis possibilita uma grande integração entre o mundo dentro e fora da escola, pois com acesso a estes materiais, o aluno pode ter sempre em mãos seu material de estudo.

1.1- PROBLEMÁTICA

Segundo uma reportagem da revista EXAME, Gabriela Ruic (2013) diz que três em cada dez brasileiros possuem *smartphones* no Brasil, sendo que 41% dos usuários destes smartphones têm entre 16 e 24 anos.

Em 2012, o MEC distribuiu cerca de 1,5 milhões de *tablets* para professores de ensino médio de escolas públicas com objetivo favorecer o

acesso a conteúdos multimídia e aperfeiçoar as aulas. Porém, o resultado não foi como o esperado. A realidade é que a grande maioria destes aparelhos estão obsoletos nas escolas devido a falta de informação e capacitação dos professores em utilizar essa ferramenta.

Segundo Liu et al. (2010), apesar da existência dos dispositivos móveis na área educacional, ainda não existe uma difusão da aplicação de técnicas de inteligência artificial nestes mesmos dispositivos, o que se apresenta como um entrave para o desenvolvimento de aplicações tendo a IA como seu paradigma. Do ponto de vista do desenvolvimento de forma geral, a Inteligência Artificial aplicada para os dispositivos móveis ainda se encontra em um estágio consideravelmente inicial.

Quanto aos problemas, relacionados ao uso de dispositivos móveis na educação, Liu et al. (2010) apresenta cinco pontos-chaves: o sistema de suporte ao aprendizado não é satisfatório nas aplicações móveis; não se podem considerar os métodos de avaliação como imparciais e totalmente confiáveis; as redes de aprendizado móveis ainda são muito limitadas; os recursos não foram totalmente adaptados para os dispositivos móveis; falta interação individual ativa do participante.

Com um aplicativo educacional de fácil utilização para professores e alunos, é possível inserir os dispositivos móveis de forma proveitosa no processo de ensino-aprendizagem de programação no ensino médio?

1.2- OBJETIVOS

1.2.1- OBJETIVOS GERAIS

O objetivo deste trabalho é a automatização de conteúdos de um Sistema Tutor Inteligente (STI) para o ensino/aprendizagem de programação de computadores.

1.2.2- OBJETIVOS ESPECÍFICOS

- Pesquisar sobre os Sistemas Tutores Inteligentes;
- Realizar um estudo sobre os Sistemas Multiagentes;

- Estudar e propor a automatização de conteúdos no STI conforme o nível de dificuldade das perguntas e no nível de conhecimento do aprendiz;
- Desenvolver um STI;
- Aplicar e validar com alunos do ensino médio.
- Escrever os resultados.

1.3- JUSTIFICATIVA

A ideia de utilizar linguagens de programação nas escolas de ensino médio tem sido amplamente discutida. Entre as principais habilidades desenvolvidas tem-se: (1) raciocínio lógico; (2) raciocínio matemático; (3) linguística; etc. (POZZEBON et al., 2013)

Diferente do Brasil, pesquisas mostram que nos Estados Unidos existem várias Instituições que estão comprometidas em levar cursos introdutórios de programação ao ensino médio. Dois exemplos incluem:

- (1) a iniciativa da Connecting Computing Educators (2014) que planeja levar 10 mil professores de Computação ao ensino médio em 10 mil escolas até 2015.
- (2) a iniciativa School Advanced Placement AP CS (2014) que pretende validar créditos do ensino médio no ensino superior.

Existem alguns projetos da Universidade Federal de Santa Catarina com Escolas Públicas da região sul Catarinense para promover a Programação no Ensino Médio.

Uma pesquisa realizada por Pozzebon (2013) para avaliar o índice de conhecimento dos alunos das escolas em algum tipo de linguagem de programação, foi constatado que dos 582 alunos entrevistados cerca de 80% dos alunos nunca fizeram curso ou tem algum domínio em qualquer linguagem de programação.

A maioria dos alunos entrevistados já conhece ou utilizam os *smartphones* e *tablets* para algum tipo de atividade como enviar mensagens, navegar na internet, jogos eletrônicos, etc, enquanto poucos professores exploram estes recursos móveis para tornar suas aulas mais atrativas e interativas.

O tema proposto neste trabalho permite a iteração entre os professores e alunos através de um Sistema Tutor Inteligente (STI). O objetivo do STI não é substituir o professor, mas auxiliar na inclusão e distribuição de conteúdos de um determinado domínio.

1.4- METODOLOGIA

Como parte inicial do projeto, pode-se citar a caracterização do problema de pesquisa. Este remete a aplicação de uma técnica de inteligência artificial no apoio ao processo educacional, utilizando-se de uma arquitetura multiagentes. Após esta etapa inicial, é realizada uma pesquisa visando reunir referenciais bibliográficas, possibilitando um embasamento maior sobre os temas que serão abordados. Através deste material, encontram-se os conceitos relacionados aos agentes inteligentes, bem como suas propriedades e capacidades, além da adaptação da arquitetura que será proposta com a automatização de conteúdos conforme o nível de dificuldade das perguntas e no nível de conhecimento do aprendiz. Com a fundamentação necessária extraída, é dado início à implementação do STI para o ensino/aprendizagem de programação. Após a implementação do sistema, são realizados testes com os professores e turmas do ensino médio para validação do funcionamento do STI. Para a implementação do protótipo é utilizado o *Eclipse 4.4.1* (ECLIPSE, 2014) com *Android SDK* (ANDROID, 2014). Para a implementação do servidor é utilizado o *NetBeans IDE 7.1* (NETBEANS, 2014), e para armazenamento dos dados é utilizado o banco de dados *Oracle 10g* (ORACLE, 2014).

2- INTELIGÊNCIA ARTIFICIAL E TÉCNICAS

Neste capítulo é apresentando um breve histórico e conceitos sobre Inteligência Artificial, bem como as técnicas de Sistemas Multiagentes e Sistemas Tutores Inteligentes que são utilizadas no desenvolvimento deste trabalho.

2.1- INTELIGENCIA ARTIFICIAL

Inteligência artificial, ou I.A. como também é chamada, é a ciência que estuda a inteligência, como o ser humano extrai o conhecimento, ou seja, como a mente humana capta informações em um meio, processa e toma decisões.

Segundo FEIGENBAUM (1989), IA é uma área da ciência da computação responsável pelo desenvolvimento de sistemas inteligentes, ou seja, tenta representar a inteligência do ser humano na máquina, fazendo com que a máquina compreenda, simule e modele a inteligência.

NORVING e RUSSEL (2014) citam as quatro capacidades que o computador, ou plataforma onde o código com IA irá executar, necessita ter neste experimento:

- Processamento de linguagem natural: permite que o computador se comunique com sucesso no idioma natural;
- Representação de conhecimento: armazenar o que sabe e o que ouve;
- Raciocínio automatizado: permite que o computador use as informações armazenadas com a finalidade de responder as perguntas e tirar novas conclusões;
- Aprendizado de máquina: para o computador se adaptar as novas circunstâncias.

A IA está presente em diversos outros equipamentos, como, no aspirador de pó *roomba* (IROBOT, 2014), em jogos (MILLINGTON; FUNGE, 2009), na medicina (EBELL, 1993), entre outras áreas. A Figura 1 ilustra algumas áreas relacionadas com a IA.

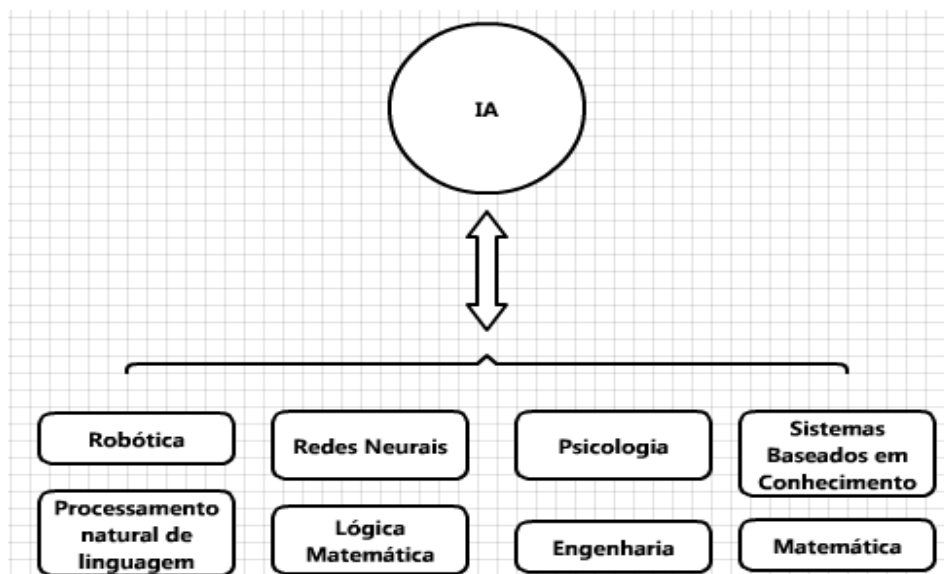


Figura 1: Áreas relacionadas com a Inteligência Artificial. Extraído e adaptado de: (NORVING; RUSSEL, 2014).

A IA trabalha em diversas áreas e é dividida em duas principais abordagens: (1) a abordagem cognitiva, também chamada de abordagem simbolista, abordagem que busca encontrar explicações de como o ser humano raciocina; (2) a abordagem conexionista, que busca dar ênfase no funcionamento do cérebro (ROSA, 2011).

Da Silva (2013) em sua pesquisa relata que os dispositivos móveis estão ganhando força no meio educacional, proporcionando ao aluno e ao discente uma maneira mais lúdica no ensino e aprendizagem. Com isso é possível ver o grande crescimento de aplicativos voltados para a educação. Alguns exemplos de aplicativos são: Rochadel (2012) com a utilização de dispositivos móveis para experimentações remotas e Paiva (2013) com aplicativo para realidade aumentada para estudo de anatomia humana.

O emprego da IA à educação visa a melhoria do aprendizado e entendimento de um conteúdo que por muitas vezes é cansativo para o estudante, para uma maneira simples e interativa. Esse emprego pode ser feito de diversas maneiras, de uma forma mais simples com um jogo com papel onde possua inteligência e faça o aluno refletir e adquirir esse conhecimento de maneira simplificada, até um jogo de computador ou celular, onde a interação é maior, fazendo com que o aluno se divirta e aprenda simultaneamente.

Giraffa (1998) aponta para o uso de técnicas de IA na educação, para

desenvolvimento de sistemas computacionais, com maior capacidade de adaptação ao contexto e mudanças de características no ambiente conforme as preferências do usuário, possibilitando assim um alto grau de interatividade entre o ambiente e o aluno.

Com o crescimento da IA, aplicativos com esta técnica também passaram a serem desenvolvidos. Santos (2012) desenvolveu o programa ITA (Assistente Virtual de Ensino Inteligente), consistindo em uma plataforma que possibilita ao estudante obter conhecimento de uma maneira agradável, garantindo qualidade e motivação, através de atividades de formação e assimilação de novos conhecimentos e Silva (2011) com um aplicativo intitulado *“Here I go New York”*, que consiste em um sistema de tutor inteligente para aprendizado da língua inglesa.

2.2- SISTEMAS MULTIAGENTES

Um SMA é um sistema computacional em que dois ou mais agentes interagem ou trabalham em conjunto de forma a desempenhar determinadas tarefas ou satisfazer um conjunto de objetivos (LESSER, 1999).

Um SMA é dividido em agentes que agem de forma autônoma, mas que interagem com outros agentes do sistema. Os agentes possuem duas características fundamentais para seu funcionamento: conseguir tomar decisões autônomas para realizar a tarefa desejada e conseguir se comunicar com os outros agentes através de um protocolo pré-definido (REIS, 2003).

A principal característica de um Sistema Multiagente (SMA) é que tem como objetivo a coletividade, ao invés do indivíduo, como os paradigmas tradicionais de IA. Desta maneira deixa de ter iniciativa para compreensão de um indivíduo particular e passa a ter o foco da atenção para a forma de integração entre as entidades que formam o sistema (agentes) e para a organização. Por exemplo, se pegarmos uma formiga em particular podemos dizer que ela tem o comportamento simples, mas se pegarmos um formigueiro como um todo o sistema é complexo e inteligente (HUBNER, 2003).

Friço, Pozzebon e Bittencourt (2004), explicam que a organização de um sistema multiagente pode ser classificada como: democrática, federada ou hierárquica. Na organização democrática os agentes não possuem

organização, e atuam em graus similares de independência e autonomia, sem hierarquia alguma. Neste caso são necessários procedimentos de coordenação sofisticados para garantir convergência e evitar interações desnecessárias. Na organização federada existe algum tipo de hierarquia. Há a presença de agentes facilitadores, agentes intermediários entre o cliente e o supervisor. Grupos de agentes da federação tem graus similares de independência e autonomia, e normalmente são bastante heterogêneos. Os procedimentos de coordenação são de média complexidade dado a própria existência do facilitador. Entretanto, as arquiteturas hierárquicas são as mais utilizadas em aplicações industriais, onde o sistema multiagente está estruturado em níveis de hierarquia.

Na área computacional, dado um problema, o sistema multiagente passa a ter como objetivo a definição de modelos genéricos de agentes, interações e organização. Alvarez e Sichman (1997), descrevem as seguintes características para uma metodologia de desenvolvimento de sistemas multiagentes:

- Agentes são criados independentes para um problema em particular;
- A interação entre agentes não é pré-definida, são desenvolvidas maneiras de interação que possam ser utilizadas em situações genéricas;
- A execução das tarefas pode ser feitas pelos próprios agentes;
- Não existe um controle centralizado para resolução de problemas.

Com essas características é possível perceber que uma grande vantagem de utilizar SMA é o fato dos sistemas conseguirem se adaptar e evoluírem.

Um agente de um sistema pode ser classificado em estacionário ou móvel, persistente ou temporário, e reativo ou cognitivo (WHITBY, 2004), (ROSA, 2011) e (FRIGO; POZZEBON; BITTENCOURT, 2004):

- Estacionários ou Móveis: Estacionários são agentes que após colocados em um ambiente não se movem. Já os Móveis, possuem algum componente que permita que se mova pelo ambiente, atualizando a memória cada vez que alterar seu local;
- Persistentes ou Temporários: Agentes persistentes são agentes que não

pode ser excluídos do sistema, ou seja, o sistema dependerá da sua tarefa sempre. Já os agentes temporários, possuem um tempo de vida no sistema, que na maioria das vezes é para execução de apenas uma tarefa.

- Reativos ou Cognitivos: Os agentes reativos são baseados em modelos de organização biológica ou etológica, como por exemplo, as sociedades de formigas ou cupins. O modelo de funcionamento de um agente reativo é o de estímulo-resposta. Em geral, estes agentes não apresentam memória, não planejam suas ações futuras e não se comunicam com outros agentes, tomando conhecimento das ações dos outros agentes pelas mudanças ocorridas no ambiente. Os agentes cognitivos são baseados em organizações sociais humanas como grupos, hierarquias e mercados. Os agentes possuem uma representação explícita do ambiente e dos outros agentes, dispõem de memória, e por isto são capazes de planejar ações futuras.

Neste trabalho é utilizado um sistema multiagente para o desenvolvimento de um sistema tutor inteligente.

2.3- SISTEMA TUTOR INTELIGENTE

Sistema tutor inteligente (STI), pode ser entendido com um sistema para computador que tem como objetivo o *feedback* ao aluno ou que fornece instruções diretas a ele, ou seja, o STI funciona sem a intervenção humana. A ideia principal do STI é o aluno aprender fazendo, com isso, auxiliando não somente o aluno, mas também facilitando o conteúdo repassado pelo professor. Como existem diferentes perfis de alunos é importante que um STI se adapte as escolhas de cada um, para isso o STI deve realizar questionários, diretamente ou indiretamente, para se modelar conforme as escolhas de cada um.

A arquitetura tradicional de um STI é dividida em três partes: o módulo aluno; a base do domínio e o módulo tutor, como mostra a Figura 2 (RAABE, 2005).

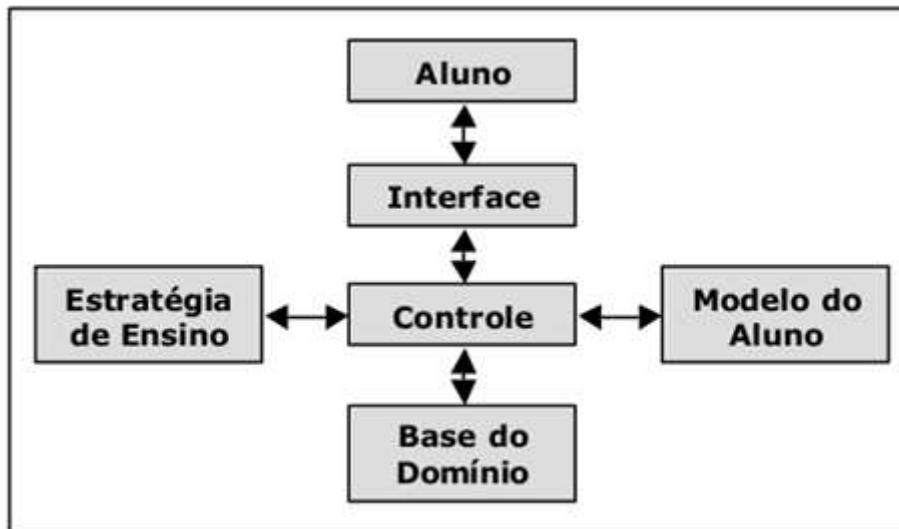


Figura 2: Modelo Tradicional de Sistema Tutor Inteligente. Extraído de:
(RAABE, 2005).

Neste modelo tradicional, na base do domínio fica armazenado o conteúdo instrucional, que está organizado conforme o propósito da aprendizagem. O módulo aluno é construído a partir de informações do aluno, que na maioria das vezes são classificados em estatísticas, que são informações como nome, idade, etc, ou dinâmicas, que são informações sobre o desempenho do aluno e o seu estado cognitivo. As estratégias pedagógicas, ou seja, quais ações que o sistema deve tomar em determinadas situações conforme as definições do aluno estão no módulo tutor (RAABE, 2005).

Além do modelo tradicional, MELO (2013), explica que um STI na maioria das vezes é dividido em 4 (quatro) partes, ou subsistemas como também são chamados:

- Módulo de Interface: responsável por disparar interações com todos os módulos, apresentando na tela o conteúdo de acordo com o perfil do usuário, este módulo é responsável pela interação do usuário e programa;
- Módulo especialista: módulo responsável por armazenar e disponibilizar o conteúdo que será apresentado para o usuário;
- Módulo aluno: responsável por identificar o perfil do aluno, na maioria das vezes disponibilizando um questionário, onde o aluno possa expor suas características de aprendizado, bem como, sua idade, grau de

instruções entre outras;

- Módulo tutor: neste módulo é onde as informações fornecidas pelo usuário serão avaliadas. Conforme as informações recebidas, este módulo irá configurar e customizar o programa conforme as preferências do usuário. Por exemplo, caso o usuário se recuse a visualizar determinada mídia sugerida pelo sistema, o módulo tutor irá modificar por uma de seu agrado. Porém, mesmo alterando a mídia, é importante salientar que o conteúdo será mantido.

O modelo utilizado neste trabalho é um modelo adaptado ao explicado anteriormente por MELO (2013).

Exemplos de sistemas tutores inteligentes: DANTAS (2013), desenvolveu uma plataforma onde é possível que uma variedade de agentes inteligentes possam se comunicar, para gerenciar diversos tipos de tarefas de coordenação de cursos e acadêmica, desta forma alunos e professores podem compartilhar materiais didáticos e tarefas com mais facilidade; SIMÕES (2013), utiliza agentes em um aplicativo voltado para a área médica, onde o usuário pode adquirir conhecimento a respeito de lombalgia.

Os sistemas tutores inteligentes abrangem uma grande área e pode ser empregado de diferentes maneiras, a mais comum é utilizando inteligência artificial, exemplo disso temos (SILVEIRA, 1992) e (DORÇA, 2002).

3- DISPOSITIVOS MÓVEIS NA EDUCAÇÃO

Neste capítulo é apresentado um estudo sobre o uso dos dispositivos móveis na educação e suas tecnologias, apresentando os potenciais e problemas.

3.1- INTRODUÇÃO E CONTEXTUALIZAÇÃO

A ideia de utilizar dispositivos móveis na educação, tanto *smartphones* como *tablets*, foi inspirada no *m-learning* ou *e-learning* que é uma modalidade apresentada no ensino a distância, que possibilita ao aluno aprender de maneira lúdica o conteúdo, além de poder aprender no seu horário de preferência.

Segundo Marghescuet al. (2007), este método é uma alternativa para o método tradicional de ensino, aliando tecnologias, que os estudantes já estão acostumados a utilizar no dia a dia, ao estudo.

Com o foco de dispositivos móveis na educação, os conceitos de *eletronic-learnig* e *mobile-learnig* acabam se misturando, devido a capacidade do dispositivo se conectar facilmente a banco de dados através da internet.

Com o grande crescimento do uso de dispositivos móveis, várias empresas e órgãos de pesquisa começaram direcionar seus investimentos para áreas onde a tecnologia possa favorecer o estudo (TAROUCO et al., 2004). HERPICH e FRANCISCATTO (2007), opinam que esta área deve ser vista como grandes oportunidades para pesquisas e desenvolvimento para áreas de aprendizado coletivo.

Kukulska-Hulme e Traxler (2013), explicam que os recursos didáticos fornecidos pelos dispositivos móveis, como tela *touch*, acelerômetro, GPS, acesso a internet móvel, entre outros, podem potencializar o contexto nos processos de ensino e aprendizagem, além de fornecer a portabilidade do dispositivo, podendo estudar em diversos lugares e horários.

Em países mais desenvolvidos como nos Estados Unidos, Japão e alguns países europeus, é comum os alunos utilizarem diferentes tipos de dispositivos conectados na internet como auxílio no processo de ensino. Por exemplo, o professor disponibilizar materiais relacionados aos assuntos

estudados, exercícios e soluções para os alunos, assim o estudante tem fácil acesso ao material, além de interagir com o professor ou até mesmo com outros alunos (MINATTO, 2013).

3.2- TECNOLOGIAS PARA O PRESENTE E FUTURO PARA O M-LEARNING

Ferramentas como *smartphones* e *tablets* possuem tecnologias com potencial para revolucionar e impulsionar as bases da educação básica e móvel. A grande vantagem e justificativa destas tecnologias serem utilizadas nas salas de aulas é a possibilidade da grande interatividade entre o mundo dentro e fora da escola, fornecendo ao aluno material em qualquer horário.

Porém, com todas as facilidades e comodidade, o uso dessas tecnologias também apresentam barreiras, como o alto custo, segurança, além de fatores na integração, e construção do material.

É possível ainda perceber certa rejeição dos professores, por esse motivo é importante que seja feito um treinamento ou adaptação do conteúdo apresentado em aula para que o professor consiga interagir de forma adequada com seus alunos. Para isso o governo federal através do Ministério da Educação (MEC), disponibilizou um investimento de R\$150 milhões de reais para a compra de 600 mil *tablets* para os professores utilizarem no ensino médio nas escolas públicas de todo o Brasil. Com isso, o governo também pretende oferecer cursos de capacitação e mais instrumentos para os professores (PORTAL BRASIL, 2012).

3.3- AMBIENTES DE APRENDIZAGEM PARA DISPOSITIVOS MÓVEIS

Com o aumento do uso de dispositivos móveis na educação, a diversidade de aplicativos voltados para esta área também cresceu, desde aplicativos voltados para crianças não alfabetizadas, auxiliando neste processo, como o ABC do Bitá, “Crianças Aprenda a Escrever” (Figura 3), Kid 123 Primeiras Palavras, entre outros, até aplicativos voltados para adultos, como Duolingo (Figura 4) ensinando diferentes idiomas, *Wolfram alpha*, auxiliando para resoluções matemáticas, entre outros.

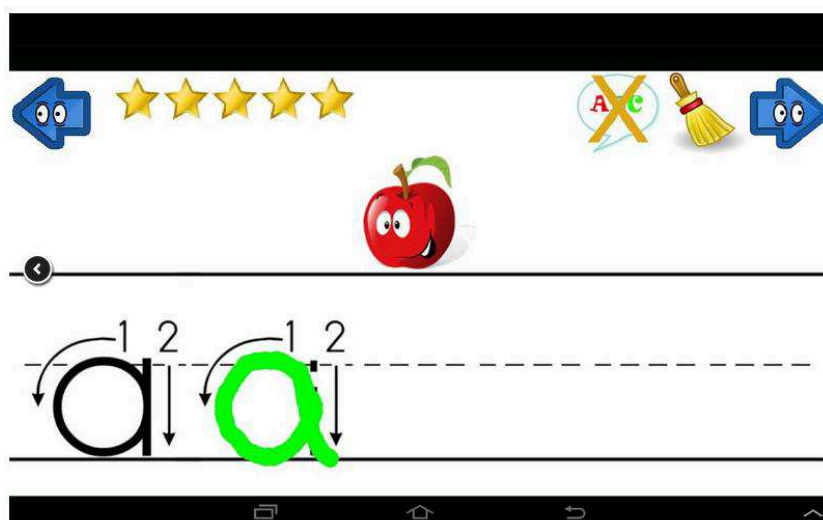


Figura 3: Aplicativo aprenda a escrever. Extraído de: (ANDROIDPIT, 2014).

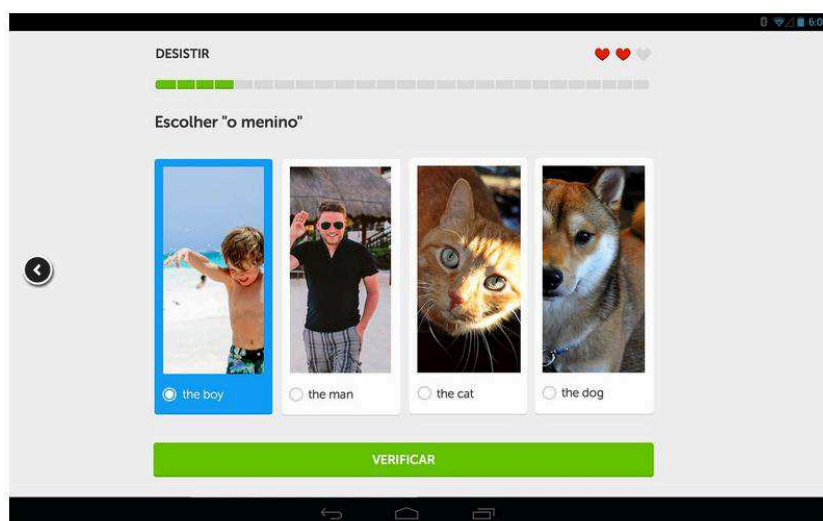


Figura 4: Aplicativo Duolingo. Extraído de: (ANDROIDPIT, 2014).

É importante perceber a diferença entre os dois aplicativos apresentados, o primeiro onde o aplicativo é voltado para crianças, possui mais cores e menos informações, o que prende a atenção do usuário, já o segundo aplicativo, voltado para o público mais familiarizado com o dispositivo, possui menos cores e mais informações em uma tela. Ambos os aplicativos estão disponíveis para as principais plataformas de celulares, *Android* da *Google* e *IOS* da *Apple*.

4- PROTÓTIPO DO SISTEMA TUTOR INTELIGENTE

Neste capítulo é descrito o processo de desenvolvimento do protótipo do STI, as ferramentas utilizadas, e as etapas necessárias até chegar a versão final do protótipo.

4.1- INTRODUÇÃO

O objetivo do protótipo STI proposto é auxiliar os professores e alunos a ensinar e aprender o conteúdo de programação para ensino médio através de um aplicativo para dispositivo móvel com sistema operacional *Android*. Com este aplicativo é pode-se acompanhar o rendimento do aprendiz no decorrer da disciplina, e o próprio aluno irá colaborar com o nivelamento das perguntas inseridas pelos tutores.

Para representar o ciclo de vida do desenvolvimento do protótipo é utilizado o modelo cascata.

O modelo cascata também é conhecido como abordagem “*top-down*”, tendo como sua principal característica a sequência de atividades onde cada fase transcorre completamente e seus produtos são vistos como entrada para uma nova fase, este modelo foi idealizado por Royce em 1970 (ROYCE, 1970 apud MACORATTI, 2014).

O ciclo de vida do protótipo do STI desenvolvido consiste em seis etapas conforme ilustra a Figura 5, onde primeiramente é feito uma coleta de requisitos, para saber com maior precisão o que precisa ser implementado no protótipo, após a coleta dos requisitos os mesmos são analisados para decidir o que de fato é viável e deve ser implementado, e então é dado início a implementação.

Após a implementação é necessário efetuar testes de sistema, para testar a estabilidade da conexão entre o cliente e o servidor, e também analisar o desempenho do protótipo.

Após os testes, o protótipo é validado com os professores e alunos de ensino médio que participam do projeto de Programação no Ensino Médio.

No fim do ciclo, caso necessário, são feitas as devidas manutenções e ajustes, com base nos resultados obtidos através da validação.



Figura 5: Ciclo de vida do desenvolvimento do protótipo.

A seguir será explicado com mais detalhes cada processo do ciclo de vida do desenvolvimento do STI.

4.2- REQUISITOS

Antes de desenvolver um software, é necessário conhecer as necessidades do usuário, e saber para qual fim ele será utilizado, através dessas informações é feito a análise de requisitos.

Segundo Sommerville (2007), requisitos de sistema são descrições dos serviços fornecidos pelo sistema e as suas restrições operacionais. Esses requisitos refletem as necessidades dos clientes de um sistema que ajuda a resolver algum problema.

Os requisitos de sistema são frequentemente classificados em requisitos funcionais e não funcionais. Requisitos funcionais são declarações dos serviços que o sistema deve oferecer, como o sistema deve reagir a entradas específicas e como o sistema deve se comportar em determinadas situações. Requisitos não funcionais são restrições sobre os serviços ou as funções oferecidas pelo sistema (SOMMERVILLE, 2007).

4.2.1- REQUISITOS FUNCIONAIS

Os requisitos funcionais representam as necessidades do sistema, nesta etapa são acordadas quais funcionalidades serão desenvolvidas para satisfazer as necessidades do usuário, são eles:

- Obrigatoriedade de usuário e senha para acessar o STI.
- Permitir o cadastro de usuários dos tipos Administradores, Tutores e Alunos.
- Permitir o cadastro/edição/exclusão de perguntas.
- Permitir o cadastro de explanação para uma pergunta.
- Permitir que um usuário responda uma avaliação com uma série de perguntas.
- Registrar um histórico com o resultado das perguntas respondidas (Quem respondeu, qual pergunta, quando, se acertou/errou).
- Não permitir responder a mesma pergunta em uma mesma avaliação.
- Atualizar dinamicamente o nível dos usuários conforme a quantidade de acertos e erros.
- Atualizar dinamicamente o nível das perguntas conforme a quantidade de vezes que respondam corretamente ou incorretamente a mesma.
- Permitir visualizar a explanação de uma pergunta (quando houver) ao errar a resposta da mesma,
- Mostrar o desempenho do aluno ao final de cada bateria de perguntas.

4.2.2- REQUISITOS NÃO FUNCIONAIS

Os requisitos não funcionais são aqueles que não dizem respeito diretamente às funcionalidades fornecidas pelo sistema. Podem estar relacionados a propriedades de sistemas emergentes, como confiabilidade, tempo de resposta, espaço em disco, desempenho e outros atributos de qualidade do produto (PAULA FILHO, 2000), são eles:

- Bom desempenho na utilização da aplicação.

- Interface simples, objetiva e usual.
- Compatibilidade com o sistema operacional *Android* 2.3 ou superior.
- Disponibilidade do servidor em 90% do tempo.

4.3- MODELO DO STI

Para desenvolver o protótipo do aplicativo, será implementado um Sistema Tutor Inteligente em um Sistema Multiagente.

O STI conta com quatro agentes que são classificados conforme a tabela a seguir:

Nome do Agente	Tipo	Função
Agente inteligente	Software, estacionário, persistente e cognitivo.	É responsável por selecionar a pergunta para o agente aprendiz, e também fazer o nivelamento dinâmico das perguntas e dos usuários.
Agente aprendiz	Software, estacionário, temporário, reativo.	É o agente que fornece o <i>layout</i> para o usuário realizar a avaliação quando logado como aprendiz.
Agente administrador	Software, estacionário, temporário, reativo.	É o agente que fornece o <i>layout</i> para o usuário cadastrar ou editar questões e cadastrar um novo tutor quando logado como administrador.
Agente tutor	Software, estacionário, temporário, reativo.	É o agente que fornece o <i>layout</i> para o usuário cadastrar ou editar questões e inserir explicações quando logado como tutor.

A Figura 6 ilustra a arquitetura do STI e o comportamento dos agentes. As setas representam a direção da informação, ou seja, as setas

bidimensionais significam que a entidade envia e recebe informações da sua conexão.

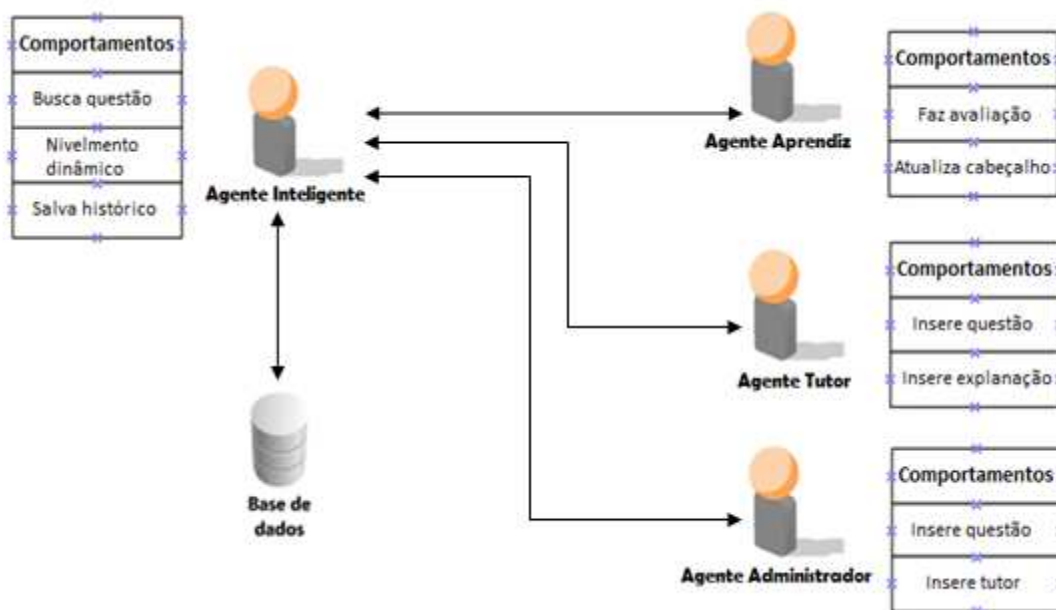


Figura 6: Arquitetura do STI.

Como podemos observar na Figura 6, apenas o Agente Inteligente tem acesso direto a base de dados, e é a única entidade que interage diretamente com todas as outras. Os agentes administrador, aprendiz e tutor, dependem do agente inteligente para completar suas ações. Nas próximas sessões será explicado com mais detalhes sobre as funções de cada agente.

Estes agentes são responsáveis pelas tomadas de decisões do sistema. A Figura 7 ilustra o diagrama de caso de uso do STI.

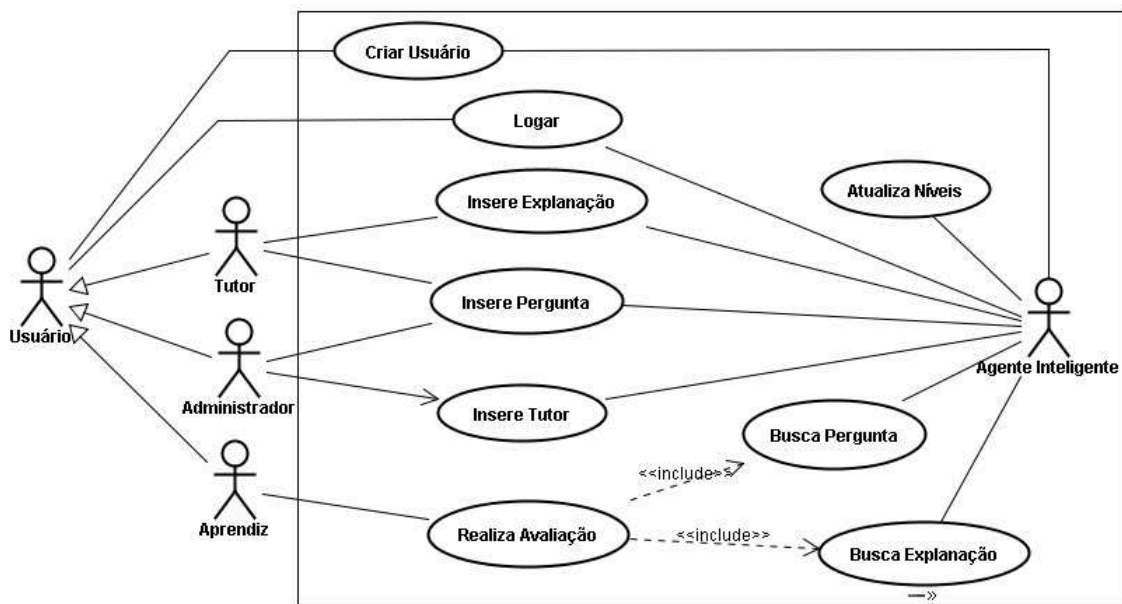


Figura 7: Diagrama de Caso de Uso

No diagrama podemos analisar a função de cada tipo de usuário do STI. O tutor é responsável por inserir explicações e perguntas na base de dados, o administrador pode inserir perguntas na base de dados além de cadastrar um novo tutor, o aprendiz pode realizar a avaliação que para tal, precisa que o agente inteligente busque uma pergunta conforme seu nível de conhecimento. O agente inteligente além de buscar a pergunta para o aprendiz, é responsável por buscar uma explicação sobre uma pergunta e mostrar ao aprendiz, quando o mesmo errar uma questão na avaliação. O agente inteligente também é responsável por atualizar os níveis dos alunos e das perguntas dinamicamente.

4.4- IMPLEMENTAÇÃO

A primeira etapa do desenvolvimento é a criação de um banco de dados para armazenar as perguntas, as explicações, os usuários e o histórico de perguntas respondidas pelos aprendizes do STI. O banco de dados é composto basicamente por quatro tabelas: “tb_per_usu”, “tb_pergunta”, “tb_usuario”, e “tb_explanacao”. Na Figura 8 podemos ver o modelo físico do banco de dados.

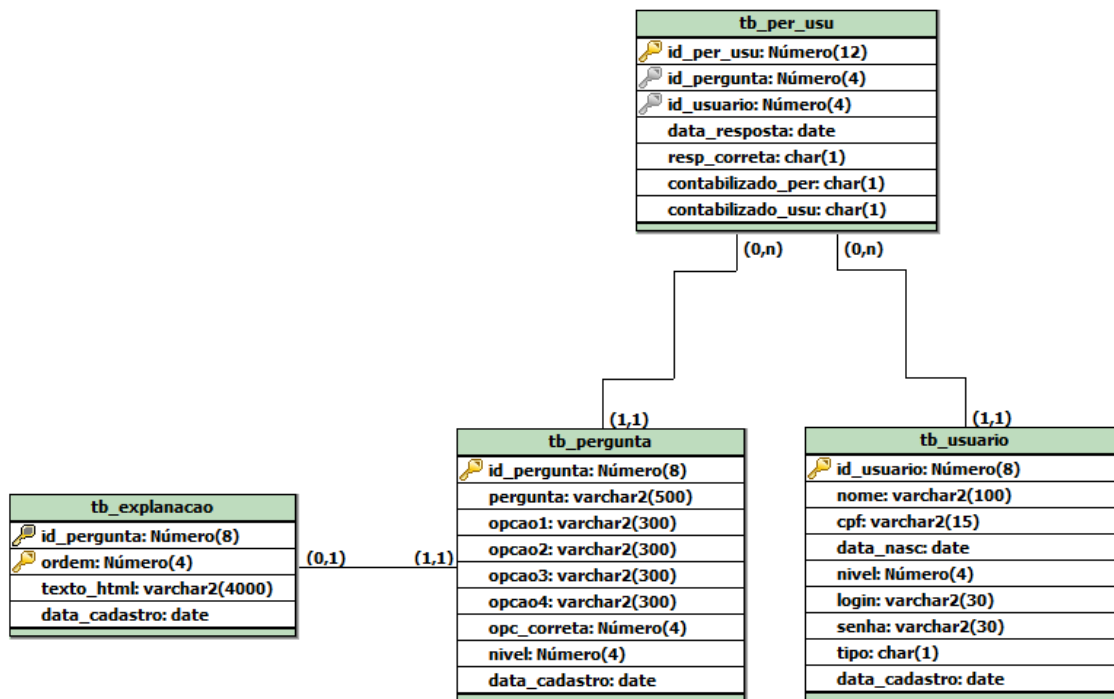


Figura 8: Modelo físico do banco de dados.

A **tabela de perguntas** (“tb_pergunta”) armazena todas as perguntas do banco. A seguir a definição dos campos:

- “id_pergunta”: É o identificador único da pergunta, a chave primária da tabela.
- “pergunta”: É a descrição da pergunta.
- “opcao 1/2/3/4”: São as opções de resposta para cada pergunta.
- “opc_correta”: É a opção verdadeira e única das quatro possíveis para cada pergunta.
- “nivel”: É o nível atual da pergunta, que será alterado dinamicamente através de uma procedure que será explicado nos próximos capítulos.
- “data_cadastro”: É a data do cadastro da pergunta.

A **tabela de explicação** (“tb_explanacao”) armazena uma explicação para determinada pergunta, fazendo uma ligação das tabelas com chave estrangeira através do campo “id_pergunta”, a seguir a definição dos campos:

- “id_pergunta”: Identifica a qual pergunta se refere tal explicação, é uma chave estrangeira para a tabela “tb_pergunta”.

- “texto_html”: É o código ou parte do código *html* que representa a explicação.
- “ordem”: É a ordem para montar o código *html*. Quando um determinado código não couber em um campo `varchar2(4000)` que é o tamanho máximo para uma string no Oracle, o sistema irá dividir esse código em partes, e organizar essas partes em ordem através do campo `ordem`.
- “data_cadastro”: É a data do cadastro da explicação.

A **tabela de usuários** (“`tb_usuario`”) armazena todos os usuários do sistema. A seguir a definição dos campos:

- “`id_usuario`”: É o identificador único do usuário, a chave primária da tabela.
- “`nome`”: Representa o nome do usuário.
- “`cpf`”: É o número do cadastro de pessoa física do usuário.
- “`data_nasc`”: Data de nascimento do usuário.
- “`nivel`”: É o nível do usuário, que será alterado dinamicamente bem como o nível das perguntas, também será explicado nos próximos capítulos.
- “`login`”: Login do usuário para acessar o sistema.
- “`senha`”: Senha do usuário para acessar o sistema.
- “`tipo`”: Identifica o tipo do usuário, onde “A” representa um Administrador, “T” representa um Tutor, e “N” representa um aprendiz.
- “`data_cadastro`”: É a data do cadastro do usuário.

A **tabela de Perguntas x Usuário** (“`tb_per_usu`”) armazena o histórico de todas as perguntas respondidas no STI, gravando o usuário, a pergunta, se acertou ou errou a questão, entre outros dados, a seguir a definição dos campos:

- “`id_per_usu`”: É o identificador único da tabela, a chave primária.
- “`id_pergunta`”: Representa qual pergunta foi respondida. É uma chave estrangeira para tabela “`tb_pergunta`”.
- “`id_usuario`”: Representa qual usuário respondeu a pergunta. É uma chave estrangeira para a tabela “`tb_usuario`”.
- “`data_resposta`”: É a data em que a pergunta foi respondida.

- “resp_correta”: Identifica se o usuário acertou ou errou a questão. Caso acerte, grava “S”, caso erre grava “N”.
- “contabilizado_per”: Identifica se o registro já foi utilizado para o nivelamento das perguntas, se já foi utilizado uma vez, é gravado “S” para não ser contabilizado novamente.
- “contabilizado_usu”: Identifica se o registro já foi utilizado para o nivelamento dos usuários, se já foi utilizado uma vez, é gravado “S” para não ser contabilizado novamente.

Após a criação do banco de dados é dado início a implementação do servidor. O STI funciona com uma comunicação entre Cliente/Servidor através de *sockets*, conforme ilustra a Figura 9. O cliente é o usuário do sistema, que poderá ser um administrador, um aprendiz ou um tutor, enquanto no servidor é criado o agente inteligente.

Para facilitar a comunicação e a troca de dados entre os agentes, foi criado um protocolo de comunicação. Cada envio de mensagem possui um identificador como prefixo, por exemplo, quando o usuário quer conectar no servidor, o usuário envia uma mensagem para o servidor: “CONNECTAR#usuario#senha”, então o servidor reconhece o prefixo da mensagem e abre uma conexão com o usuário, após tentar abrir a conexão o servidor envia uma resposta para o usuário com a resposta se conseguiu ou não estabelecer uma conexão. Apenas o servidor terá acesso direto ao banco de dados e será responsável pela intermediação entre o usuário e o banco.



Figura 9: Comunicação entre Usuário e o Servidor.

Após a implementação do servidor é iniciado a implementação do protótipo.

4.4.1- FUNCIONAMENTO DO PROTÓTIPO

Para a utilização do protótipo é necessário um dispositivo móvel com *Android 2.3* ou superior conectado a internet, devido ao fato das informações do STI estarem gravadas em um banco de dados remoto.

4.4.2- INICIO DO APLICATIVO

Ao abrir o aplicativo, ele tenta estabelecer uma conexão com o servidor. Caso obtenha sucesso, o usuário é encaminhado para a tela de login, caso contrário é emitido uma mensagem com o problema ocorrido.

4.4.3- TELA DE LOGIN

Na tela de login o usuário pode informar um login e senha para acessar o aplicativo, ou clicar no botão “Criar Usuário” caso o mesmo ainda não possua um cadastro. A Figura 10 ilustra a tela de login do STI.



Figura 10: Tela de login do STI

Após informar um login e senha, o aplicativo envia uma mensagem ao servidor solicitando um login, ao receber a mensagem, o servidor procura o cadastro do usuário no banco de dados e envia uma resposta ao aplicativo, caso encontre um usuário válido para o login e senha informados, o sistema retorna uma mensagem de sucesso com o nome, nível e o tipo do usuário encontrado, caso contrário, retorna uma mensagem de aviso.

Ao receber a resposta do servidor, através do protocolo de comunicação o aplicativo identifica se o servidor encontrou ou não um usuário. Se encontrar o aplicativo cria um agente correspondente ao tipo de usuário.

4.4.4- MÓDULOS

O protótipo do aplicativo representa o usuário do STI, conforme o tipo do usuário, seu respectivo agente é criado. Quando o usuário for do tipo “A”, o STI identifica o tipo do usuário e cria um agente administrador temporário. Quando o usuário for do tipo “N”, o STI identifica o tipo do usuário e cria um agente aprendiz temporário. Quando o usuário for do tipo “T”, o STI identifica o tipo do usuário e cria um agente tutor temporário. Após a criação do agente, seu respectivo módulo é carregado da tela.

4.4.4.1- MÓDULO ADMINISTRADOR

O módulo do administrador é o módulo que tem controle total sobre o sistema, o usuário com acesso a este módulo pode fazer manutenção das questões do STI, e é o único com permissão para inserir ou remover tutores.

Ao entrar neste módulo o aplicativo abre uma tela que representa o menu do administrador, essa tela contém uma lista com os tutores já cadastrados no STI, listando ID – NOME – CPF, um botão para cadastrar tutores, um botão para realizar manutenção de perguntas e um botão para fechar o aplicativo, conforme ilustra a Figura 11.

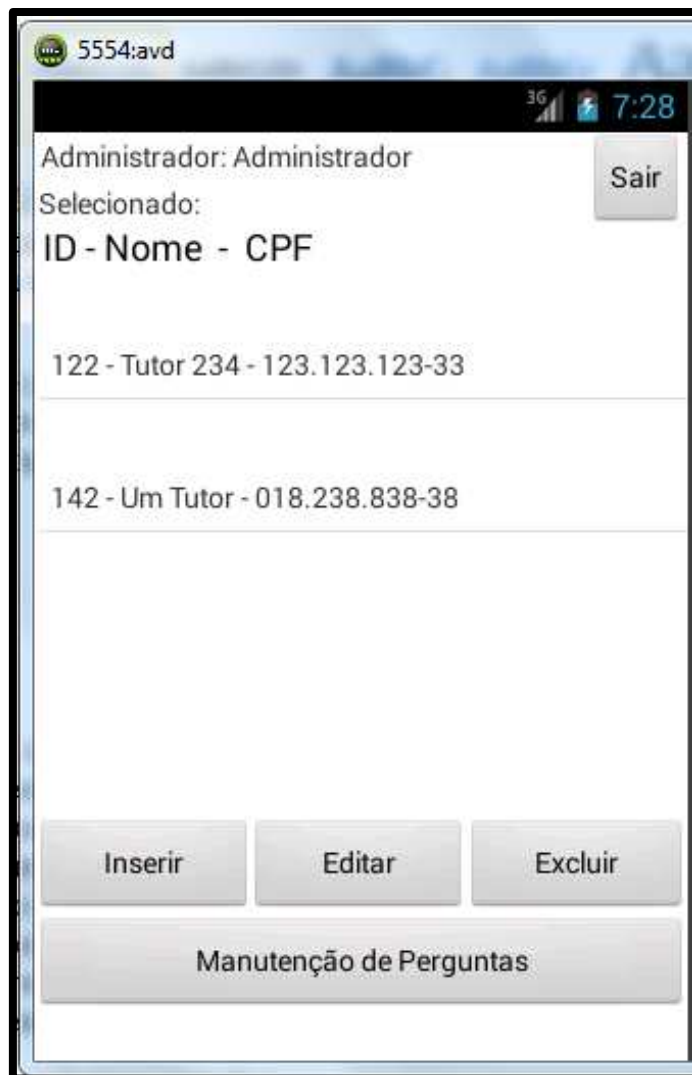


Figura 11: Tela do Menu do Administrador.

O botão “Inserir”, abre uma tela onde é possível cadastrar um usuário que será do tipo “T”, ou seja, quando este novo usuário efetuar o login no aplicativo, ele será identificado como o tipo “T” e então o STI irá criar um agente tutor e carregar o módulo tutor.

O botão “Manutenção de Perguntas” encaminha o usuário para a tela de manutenção de perguntas do sistema. Essa tela será explicada na próxima subseção.

O botão “Sair” encerra a conexão com o servidor e fecha o aplicativo, deixando a porta que o agente usuário estava ocupando livre para outro usuário poder se conectar ao servidor.

Futuramente serão desenvolvidas novas funcionalidades para o este módulo, como a parametrização do tempo de resposta para cada pergunta e a quantidade de perguntas necessárias para completar uma avaliação.

4.4.4.2- MÓDULO TUTOR

Neste módulo o usuário tem acesso às questões cadastradas no sistema, bem como o módulo administrador. O usuário com acesso a este módulo pode cadastrar, editar ou remover perguntas e explicações.

Ao entrar nesse módulo o aplicativo abre uma tela que representa o menu do tutor, esta tela tem um botão para realizar manutenção nas perguntas do sistema, um botão para realizar manutenção nas explicações e um botão para sair do aplicativo.

O botão “Manutenção de Perguntas” encaminha o usuário para a tela de manutenção de perguntas, a mesma encontrada no módulo do administrador. Ao abrir esta tela o aplicativo carrega a lista de todas as perguntas que estão com status “A” (ativo) no sistema, e então o usuário pode inserir, editar ou remover uma pergunta. Remover uma pergunta não significa remover ela do banco de dados, apenas é mudado seus status para “I” (Inativo), pois é importante manter esse registro para manter os históricos de perguntas respondidas, já que a tabela de perguntas respondidas tem chave estrangeira com a tabela de perguntas. A Figura 12 ilustra a tela de “Manutenção de Perguntas”.

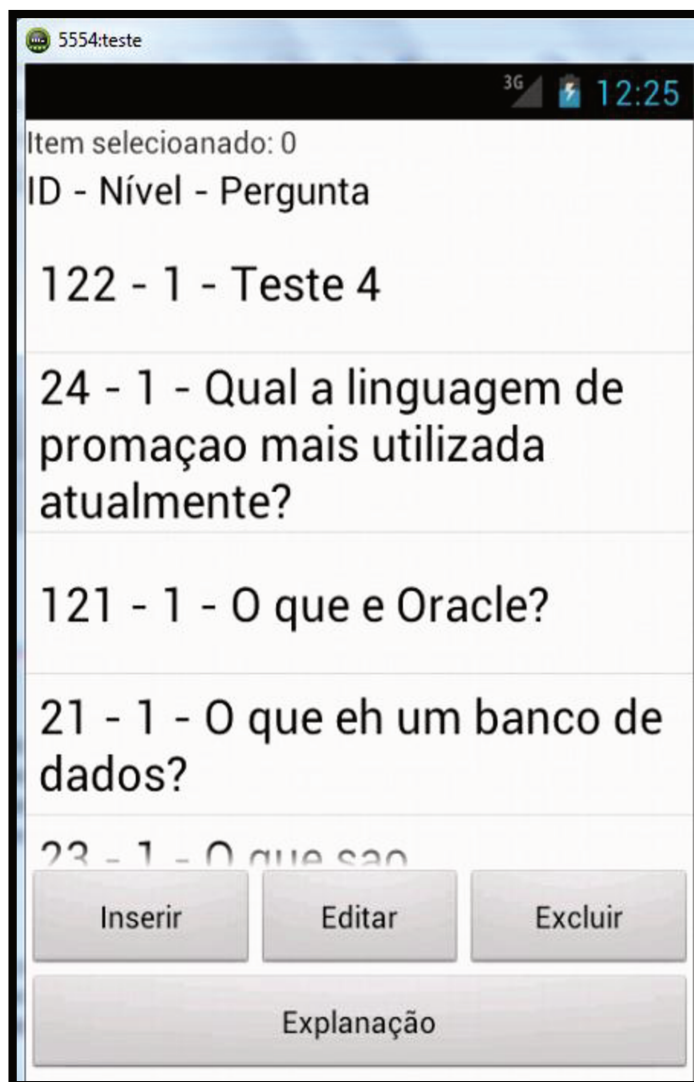


Figura 12: Tela de Manutenção de Perguntas.

Na tela de manutenção de perguntas o usuário também pode inserir uma explicação para a pergunta selecionada clicando no botão “Explicação”. A Figura 13 ilustra a tela de inserção de explicação.

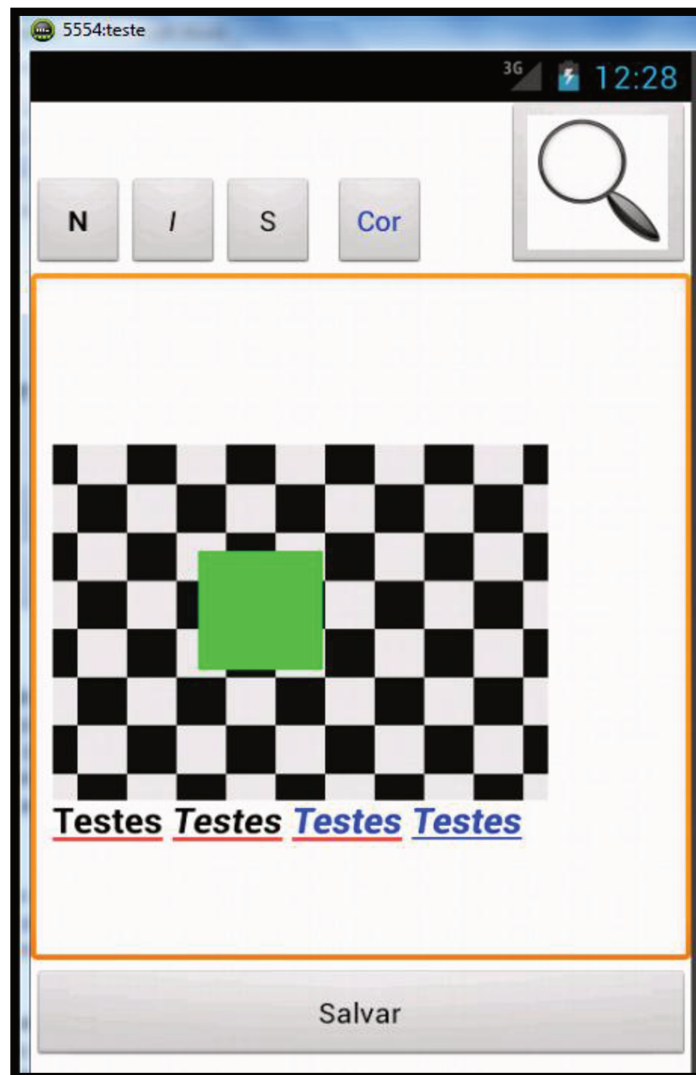


Figura 13: Tela de Inserção de Explicação.

Na tela de explicação de pergunta, o usuário pode montar uma explicação com um editor de texto. Clicando na lupa canto superior direito da tela, o usuário pode escolher uma imagem da galeria para inserir no editor. Clicando no botão “N”, o texto digitado fica negrito, o botão “I” deixa o texto itálico, e o botão “S” deixa o texto sublinhado. O botão “Cor” permite que o usuário troque a cor do texto.

Após montar a explicação, o usuário pode salvá-la no banco de dados. Ao clicar em “Salvar”, o sistema converte todos os dados do editor em uma página *html*. Conforme a quantidade de dados contidos no editor, o texto convertido para *html* pode ficar muito extenso, devido a essa possível extensão é necessário converter essa *string* em *bytes*, para que o seu tamanho seja reduzido e então o cliente consiga enviar esses dados para o servidor.

Ao receber esses dados em *bytes*, o servidor precisa convertê-lo novamente para *string*. Normalmente essa *string* irá ultrapassar o tamanho máximo de 4000 caracteres permitidos no tipo de campo *varchar2* do *Oracle*, então é necessário buscar alternativas para salvar estes dados. A primeira alternativa seria utilizar um campo *clob*. Um campo *clob* é um tipo primitivo do *Oracle*, que até a versão utilizada neste trabalho (10), permite armazenar textos de até 4GB. A segunda alternativa seria dividir essa página *html* em partes menores, e inserir essas partes em ordem para depois poder remontá-las novamente. Devido a menor complexidade, neste trabalho é utilizada a segunda alternativa.

O botão “Sair” encerra a conexão com o servidor e fecha o aplicativo, deixando a porta que o agente usuário estava ocupando livre para outro usuário poder se conectar ao servidor.

4.4.4.3- MÓDULO APRENDIZ

No módulo do aprendiz o usuário pode realizar a avaliação quantas vezes julgar necessário.

Ao entrar neste módulo o aplicativo abre uma tela que representa o “Menu do Aluno”, esta contém um botão para iniciar a avaliação e um botão para sair do aplicativo.

Ao clicar no botão “Iniciar Teste”, o aplicativo abre a tela para iniciar a avaliação. Esta tela possui: um cabeçalho com dados referentes ao usuário, uma pergunta com as opções, e um botão para responder conforme ilustra a Figura 14. Cada avaliação possui 10 (dez) perguntas, onde cada pergunta deve ser respondida em até 90 segundos, caso ultrapasse o limite de tempo, a questão é computada como errada. Na parte superior da tela através do cabeçalho, o usuário pode acompanhar: a quantidade de erros e acertos, o tempo restante para responder a questão da tela, o seu nível atual e a quantidade de perguntas já respondidas na mesma avaliação.

Todas as perguntas apresentadas na tela são perguntas objetivas com quatro opções, sendo apenas uma a correta.

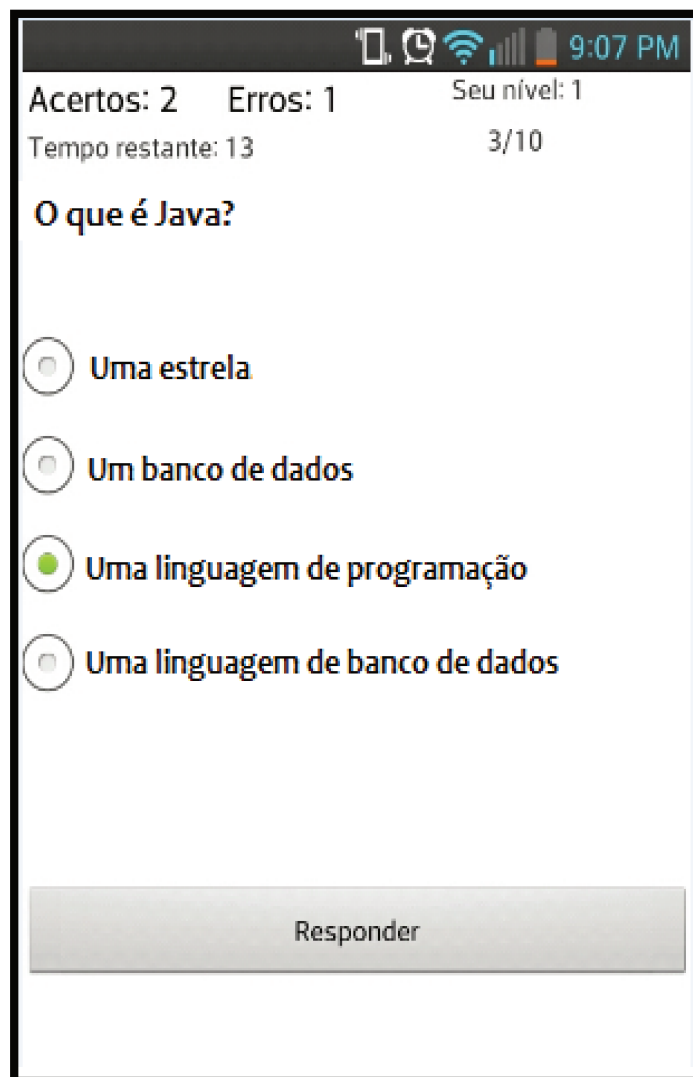


Figura 14: Tela da avaliação.

As perguntas que são apresentadas na tela são buscadas no banco de dados aleatoriamente filtrando por nível do usuário e perguntas ativas, caso não possuir nenhuma pergunta ativa para o nível do usuário, o filtro por nível é desconsiderado. Por exemplo, caso um usuário de nível 2 (dois) inicie a avaliação, o aplicativo solicita ao servidor uma pergunta de nível 2 (dois), caso o servidor não encontre uma pergunta ativa de nível 2 (dois), então ele procura uma pergunta ativa para qualquer nível, dando preferência a níveis menores que o do usuário. Conforme o número de erros e acertos do usuário, o seu nível e o da pergunta são alterados em tempo real.

Ao clicar no botão “Responder Pergunta”, ou o tempo de limite para cada pergunta for ultrapassado, uma nova pergunta é apresentada na tela, e os dados do cabeçalho são atualizados, controlando para que uma mesma

pergunta não seja apresentada na tela em uma mesma avaliação mais que uma vez.

Caso o usuário errar a pergunta respondida, e a mesma possuir uma explicação cadastrada, o sistema emite uma mensagem perguntando se o usuário deseja visualizar a explicação da pergunta conforme ilustra a Figura 15. Clicando em “Sim”, o sistema busca a explicação para a pergunta respondida, e se encontrar, abre uma tela com a explicação.

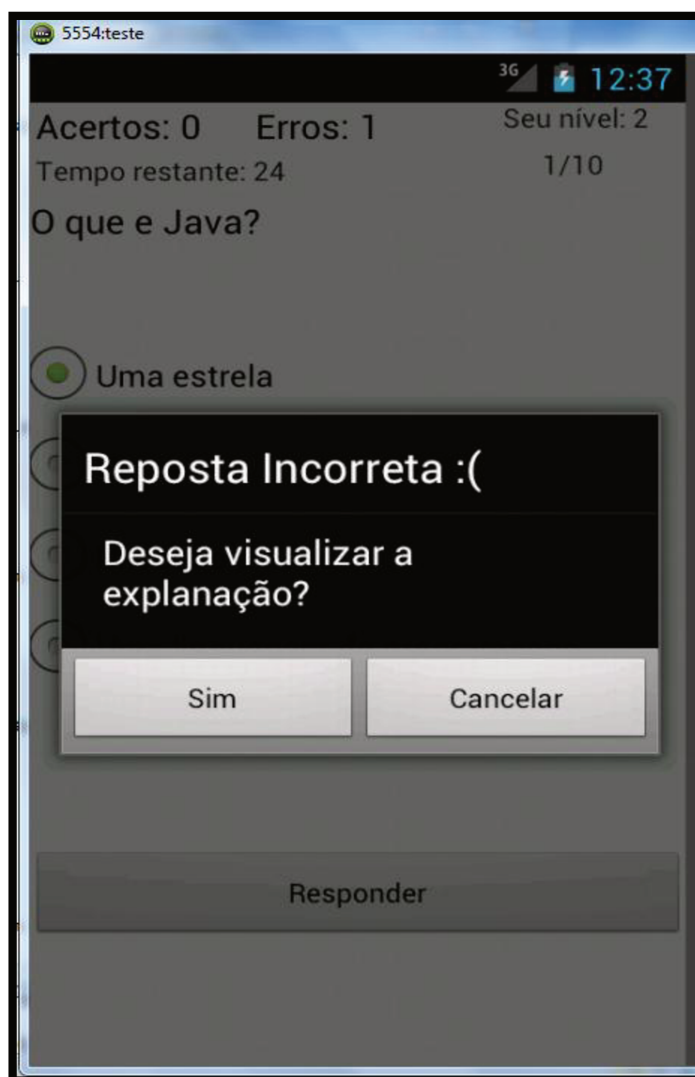


Figura 15: Questionamento sobre visualização da explicação.

Ao responder as dez perguntas da avaliação, o aplicativo abre uma tela mostrando o seu desempenho, o aprendiz pode visualizar a quantidade de erros, acertos e o nível atingido conforme ilustra a Figura 16. Nesta tela o usuário pode escolher entre iniciar uma nova avaliação ou encerrar o aplicativo.

Caso escolha iniciar uma nova avaliação, o sistema volta para a tela anterior, limpa o cabeçalho e reinicia o teste. Caso escolha sair do aplicativo, o mesmo é encerrado.

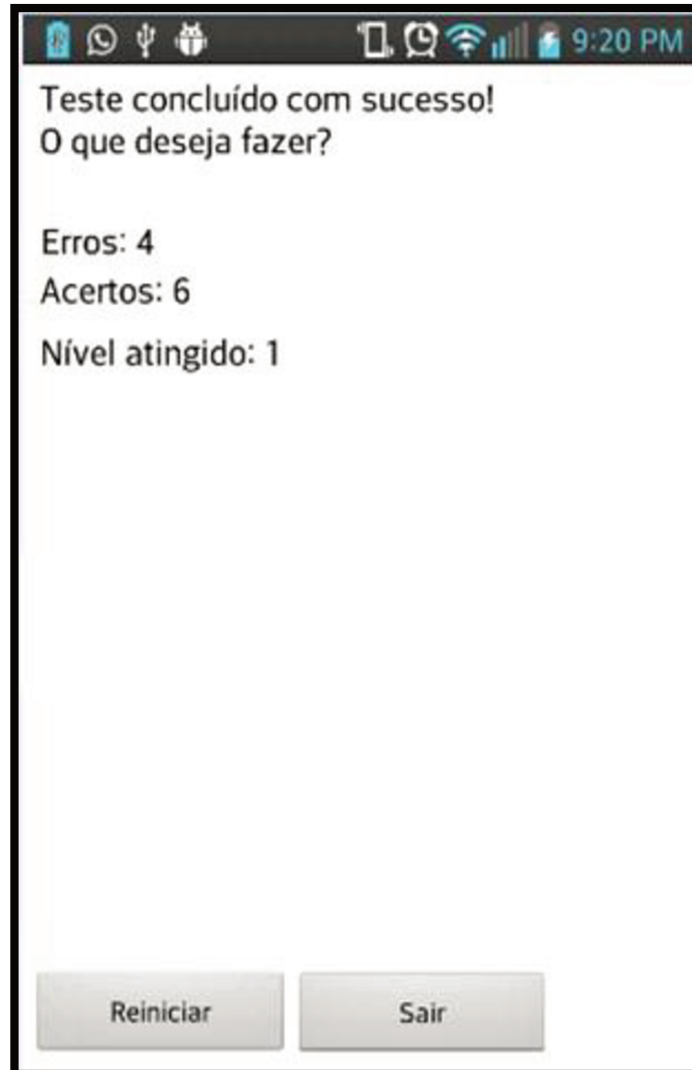


Figura 16: Tela do resultado do teste.

4.4.5- NIVELAMENTO DINÂMICO

A alteração de nível de pergunta e do usuário funciona de forma dinâmica, a cada pergunta respondida é salvo um histórico na tabela de perguntas respondidas, gravando quem respondeu, qual a pergunta, se acertou ou errou, a data que foi respondida, e duas marcações, uma marcando se esse histórico já foi utilizado no nivelamento de perguntas e outra marcando se esse

mesmo histórico já foi utilizado no nivelamento de usuário, ambos desmarcados por padrão.

Após salvar este histórico o sistema invoca uma *procedure* do banco de dados que é responsável por fazer o nivelamento dinâmico. Essa *procedure* faz dois procedimentos, primeiro faz o nivelamento do usuário e logo após o nivelamento da pergunta.

Nivelamento do usuário: A *procedure* consulta os três últimos registros inseridos na tabela de perguntas respondida, que ainda não foram utilizadas no nivelamento do usuário. Caso possua três registros ainda não computados, ela verifica se a quantidade de erros ou de acertos é maior nestes três últimos registros, caso a quantidade de erros for maior, o usuário perde um nível, caso a quantidade de acertos for maior, o usuário ganha um nível, controlando para que o nível fique entre 1 (um) e 5 (cinco). Após computar esses registros, a *procedure* os marca como computados, para que não sejam utilizados novamente para este mesmo fim.

Nivelamento de pergunta: Similar ao nivelamento de usuários, a *procedure* consulta os três últimos registros inseridos na tabela de perguntas respondida, que ainda não foram utilizadas no nivelamento de perguntas. Caso possua três registros ainda não computados, ela verifica se a quantidade de erros ou de acertos é maior nestes três últimos registros, caso a quantidade de erros for maior, a pergunta ganha um nível, ou seja, como a maioria dos usuários está errando essa questão, então ela é uma pergunta com um maior nível de dificuldade, caso a quantidade de acertos for maior, a pergunta perde um nível, passando a ser uma pergunta com menor nível de dificuldade, controlando para que o nível fique entre 1 (um) e 5 (cinco). Após computar esses registros, a *procedure* os marca como computados, para que não sejam utilizados novamente para este mesmo fim.

4.5- TECNOLOGIAS UTILIZADAS

Nesta sessão será discutido sobre a principal tecnologia utilizada no desenvolvimento do protótipo, que foi necessária para fazer a comunicação entre os agentes, os *sockets*.

4.5.1- SOCKETS

O STI desenvolvido conta com três elementos fundamentais para seu funcionamento, um banco de dados, um servidor e um protótipo. O banco de dados é responsável por armazenar os dados do STI, o servidor é responsável por coordenar os usuários e intermediar a comunicação do protótipo com o banco de dados, e o protótipo representa o usuário do STI.

O servidor e o banco de dados estão instalados em um computador localizado no Laboratório de Tecnologias da UFSC (*LabTec*), e o protótipo é um aplicativo para dispositivos móveis com o SO *Android* a partir da versão 2.3.

O servidor e o protótipo ficam em locais físicos diferentes, o que resultou na necessidade de adotar um mecanismo de comunicação para estes elementos, os *sockets*.

No processo de software, existe a comunicação, o socket permite isso de forma a ser indispensável nos sistemas distribuídos, pois existe uma ligação entre os processos do servidor com o cliente. Um socket é a interface entre a camada de aplicação e a de transporte. (SANTOS, 2014)

Socket é uma interface de comunicação Cliente/Servidor que está também disponível em Java, podendo ser utilizada através do protocolo TCP (Transmission Control Protocol) ou do protocolo UDP (*User Datagram Protocol*), ambos sobre o protocolo IP (*Internet Protocol*). Os protocolos TCP e UDP possuem suas vantagens e desvantagens. Tacla (2014) apresenta suas principais características destacando suas vantagens e desvantagens:

UDP – Canal não confiável

- Não garante a entrega dos datagramas;
- Pode entregar datagramas duplicados;
- Não garante a ordem de entrega dos datagramas;

- Não tem estado de conexão (escuta estabelecida);
- Mais rápido que o modo orientado a conexão (TCP).

No protocolo UDP as mensagens são enviadas através de datagramas. Um datagrama é um pacote de dados que contém um cabeçalho e o conteúdo da mensagem, no cabeçalho devem constar o IP e a porta de origem, o IP e a porta de destino, o tamanho do pacote, e um *checksum*. Como não há uma conexão estabelecida entre o Cliente e o Servidor, é necessário todas essas informações para o envio de cada pacote.

TCP – Canal confiável

- Fluxo contínuo (*stream*);
- São fragmentados pelo TCP em segmentos;
- Garante a entrega;
- Não há duplicação;
- Garante a ordem de entrega dos segmentos
- Possui conexão e controla o estado de conexão (escuta, estabelecida, fechada);
- Ponto a ponto (*sender e receiver*).
- Controle de congestionamento – controla o *sender* quando a rede congestionada;
- Controle de fluxo – controla o *sender* para não sobrecarregar a rede.
- Mais lento que o modo não orientado a conexão (UDP).

No protocolo TCP o Cliente e o Servidor estabelecem uma conexão, e enviam mensagens através de *streams*. *Stream* é uma sequência de *bytes* transmitida durante uma conexão, neste protocolo as *streams* podem ser fragmentadas em segmentos quando necessário (quando uma mensagem é muito grande, por exemplo), garantindo a ordem de envio e recebimento.

O processo de comunicação do protocolo TCP ocorre da seguinte forma: O servidor abre uma porta determinada pelo usuário e fica aguardando conexões de clientes. O cliente precisa saber qual a porta aberta pelo servidor,

e então solicita uma conexão com o mesmo. A Figura 17 ilustra a comunicação do protocolo TCP.

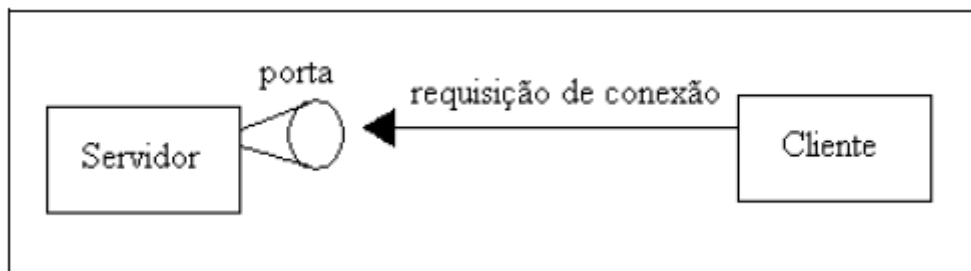


Figura 17: Comunicação do protocolo TCP. Extraído de: (SANTOS, 2014).

Caso não houver nenhum problema e o servidor aceitar a conexão, o cliente é transferido para outra porta qualquer livre criando assim um canal de comunicação entre o cliente e o servidor, e desocupa a porta aberta inicialmente pelo servidor, conforme ilustra a Figura 18.

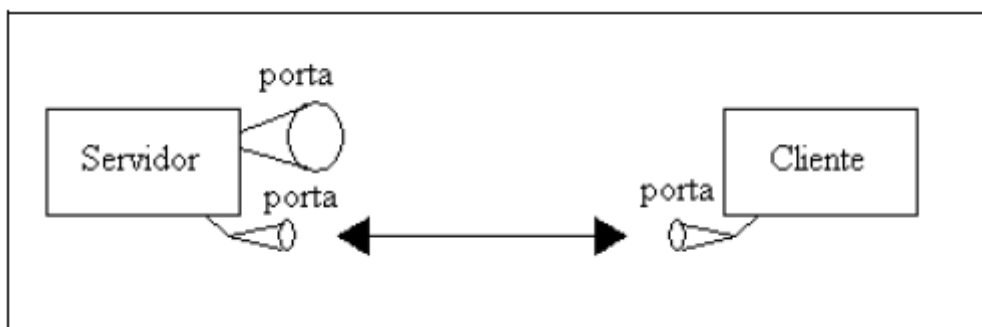


Figura 18: Transferência das portas do protocolo TCP. Extraído de: (SANTOS, 2014).

Na comunicação entre as entidades do STI foi optado pela utilização dos *sockets* com o protocolo TCP devido a maior confiabilidade da entrega das mensagens.

5- VALIDAÇÃO DO PROTÓTIPO

Nesta seção são apresentados os resultados da validação do protótipo. Na validação foram utilizadas a observação direta e um questionário. A observação direta objetiva prover suporte aos usuários durante a utilização do aplicativo. E o questionário com objetivo de avaliar se o STI cumpriu com os requisitos descritos no início do projeto bem como a satisfação dos usuários quanto à utilização do aplicativo.

5.1- ESTRATÉGIAS DE VALIDAÇÃO

A validação tem o objetivo de avaliar se o que foi construído atende as expectativas do usuário. A validação do Mazk STI (protótipo desenvolvido) foi realizada por alunos de três escolas, sendo duas da cidade Araranguá e uma da cidade de Sombrio, ambas participantes do projeto Programação no Ensino Médio (PEM) da UFSC – Araranguá, e com os professores do projeto. As escolas participantes são:

- Escola de Educação Básica de Araranguá (Colégio Estadual) – Araranguá
- Escola de Educação Básica Professora Maria Garcia Pessi – Araranguá
- Escola de Ensino Médio Macario Borba (Escola Jovem) – Sombrio

A validação do aplicativo aconteceu na UFSC – Araranguá, unidade do Mato Alto, na sala 110 do LabTeC, onde 50 alunos do 2º e 3º ano do ensino médio, divididos em duas turmas, utilizaram o aplicativo e responderam a um questionário. A primeira turma participou da validação das 9:00 às 12:00 horas e a segunda turma participou da validação das 14:00 as 17:00 horas, ambas no dia 5 de julho de 2014.

No decorrer da mesma semana, os professores do projeto inseriram os conteúdos na base de dados através do módulo tutor do aplicativo.

Para validar os resultados obtidos com o desenvolvimento do Mazk STI foi adotada a seguinte estratégia:

- a) O aplicativo foi instalado em quatro *tablets* com sistema operacional *Android*, disponibilizados pela UFSC.
- b) Foi elaborado um questionário de validação do Mazk STI contendo perguntas de múltipla escolha e espaços para sugestões ou observações relevantes quanto ao uso do aplicativo. O questionário aplicado encontra-se no **APENDICE 1**, no final do documento.
- c) Quatro alunos eram chamados simultaneamente para utilizar o aplicativo, e com o acompanhamento de dois tutores os alunos utilizavam o aplicativo. Cada aluno era orientado a efetuar um cadastro de usuário e a realizar a avaliação quantas vezes julgasse necessário.
- d) Ao final da utilização do aplicativo, o aluno respondia ao questionário através de um computador localizado nessa mesma sala.
- e) Após todos os alunos utilizarem o aplicativo, os professores do projeto foram chamados para utilizar o aplicativo como um aprendiz e responder o mesmo questionário.

5.2- RESULTADOS DA OBSERVAÇÃO DIRETA

Durante a utilização do aplicativo, alguns alunos solicitavam ajuda quando encontravam alguma dificuldade ou algum problema. Quando surgia alguma dificuldade na utilização por parte do aluno, ele recebia a orientação de um tutor para prosseguir. Quando foram encontrados problemas como erros, eles foram solucionados imediatamente.

Os principais casos observados foram:

- a) Ao tocar no botão “Responder”, na tela de responder uma pergunta, alguns alunos não esperavam carregar a próxima pergunta e davam um segundo toque não intencional no botão, então respondia a próxima pergunta sem vê-la.

Solução: Quando o aluno responde uma pergunta, uma próxima pergunta é consultada no banco para ser mostrado na tela, neste intervalo de tempo, o aluno não esperava carregar e tocava novamente no botão. Para este caso, seria necessário um ajuste na programação, adicionando uma “Barra de Progresso” na tela impedindo que o aluno pressionasse o botão novamente

antes de carregar a próxima pergunta. Como este ajuste demandava um tempo maior, o aluno apenas foi orientado a aguardar carregar a próxima pergunta após tocar no botão “Responder”.

b) Alguns alunos reclamaram da lentidão do aplicativo.

Solução: A turma do período matutino reclamou do desempenho do aplicativo. No momento da reclamação uma análise do aplicativo foi feita, e de fato havia uma lentidão excessiva na utilização. O aplicativo estava lançando *threads* (Processos) inúteis que estavam deixando a aplicação lenta, durante o intervalo entre a chamada de outros alunos para realizar o teste, foi feita uma pausa e rapidamente foi criada uma nova versão do aplicativo eliminando essas *threads* que não estavam sendo utilizadas. Após a alteração não houve mais reclamações de lentidão.

c) Ao virar a orientação do *tablet*, o aplicativo encerrava inesperadamente, com isso o aluno precisava reiniciar o aplicativo e iniciar uma nova sequência de perguntas.

Solução: Quando um aluno virava a orientação do *tablet* o aplicativo era encerrado, isso ocorre devido a uma configuração do aplicativo. A solução imediata para o problema foi bloquear a orientação dos *tablets* que estavam sendo utilizados para testar o aplicativo.

d) Os alunos não preenchiam os campos “CPF e Telefone” no cadastro de usuário.

Solução: Alguns alunos do período matutino questionavam da obrigatoriedade em preencher os campos CPF e Telefone. Ao perceber o incomodo e a insatisfação, esses campos foram removidos, e já não estavam mais na versão criada para o item “b”.

5.3- RESULTADOS DO QUESTIONÁRIO APLICADO

Uma das técnicas para acompanhar a satisfação do usuário quanto à utilização do aplicativo foi o questionário. O questionário foi elaborado com 9 (nove) questões de múltipla escolha permitindo comentar em cada questão,

além de um espaço no final do questionário onde permitia ao usuário descrever alguma crítica, sugestão ou elogio ao aplicativo.

O questionário foi aplicado com 50 (cinquenta) alunos após a utilização do aplicativo, e com os professores do projeto. A análise e interpretação dos resultados obtidos na aplicação do questionário são apresentadas a seguir:

A primeira questão do questionário é quanto à utilização de tecnologias móveis no processo educacional, visto que essa é uma tecnologia emergente e vem conquistando um grande público, principalmente os jovens, nos últimos anos. No Gráfico 1 é possível verificar a opinião dos usuários quanto a questão. Nas respostas eram permitidas as seguintes opções: “Ótimo”, “Bom”, “Razoável” e “Ruim”.

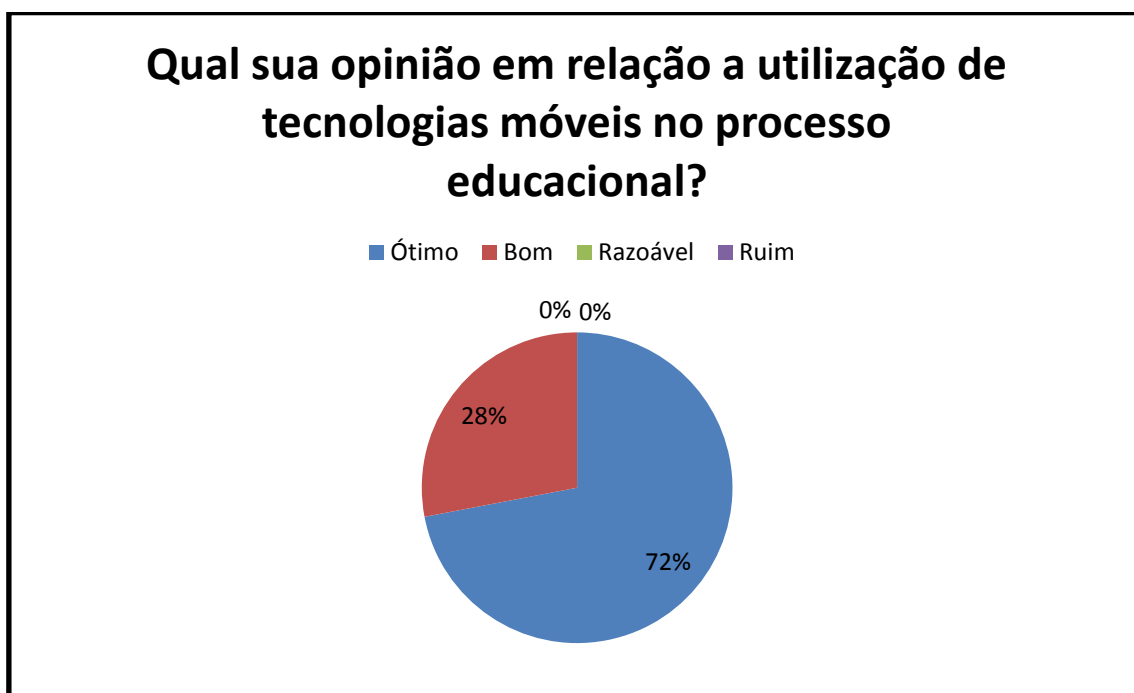


Gráfico 1: Opinião dos usuários quanto à utilização de tecnologias móveis no processo educacional.

Nos resultados, 72% dos usuários responderam que as tecnologias móveis são ótimas ferramentas para o processo educacional, enquanto 28% responderam que elas são apenas boas, além do comentário: “quando bem direcionada e planejada”, ou seja, segundo o usuário, as tecnologias móveis podem ser úteis no processo educacional quando bem direcionada e aplicada.

Com estas respostas pode-se inferir que os usuários acreditam que essas tecnologias são ótimas aliadas no processo educacional, porém é importante direcioná-las para determinados segmentos, e aplicar com eficiência.

Outra questão importante a ser avaliada é a opinião dos usuários quanto à utilização de uma técnica de inteligência em uma avaliação, onde os níveis das perguntas e dos alunos eram modificados dinamicamente conforme o desempenho dos mesmos. No Gráfico 2 é possível verificar a opinião dos usuários quanto a questão. Nas respostas eram permitidas as seguintes opções: “Ótimo”, “Bom”, “Razoável” e “Ruim”.

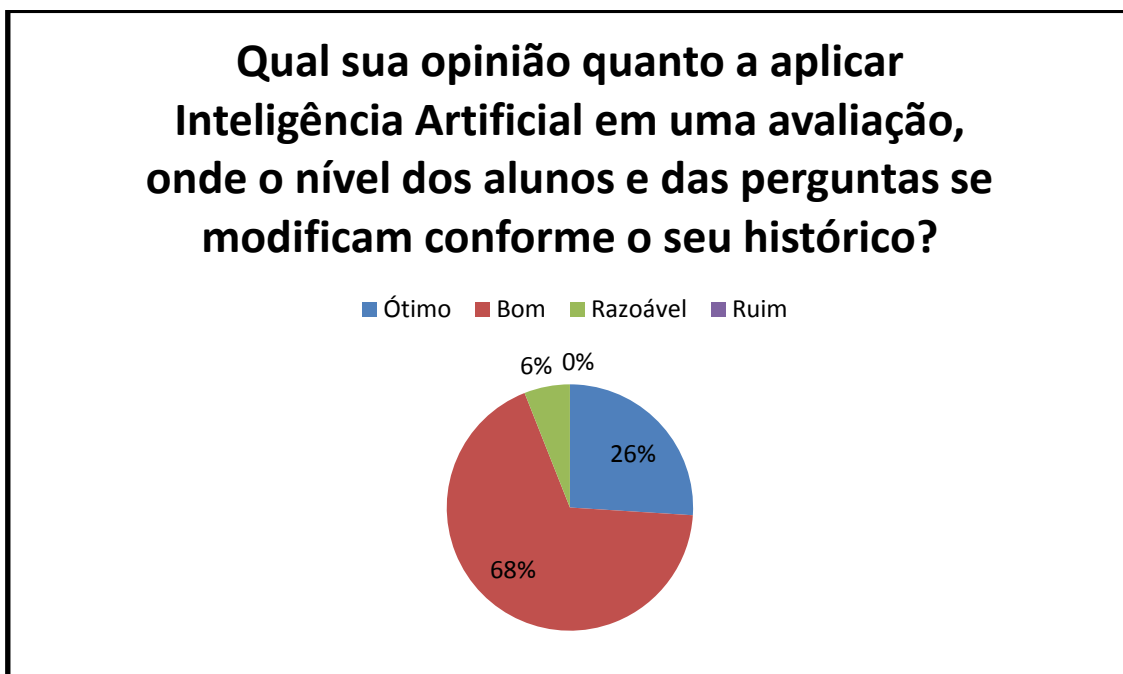


Gráfico 2: Opinião dos usuários quanto a aplicação de IA em uma avaliação.

Nos resultados, 26% dos usuários responderam que a Inteligência Artificial é uma ótima técnica para avaliação, enquanto 68% responderam que a técnica é apenas boa, e os outros 6% responderam razoável.

Um usuário fez o seguinte comentário sobre a questão: “dependendo do uso, pois deste modo faremos distinção dos alunos, se esta distinção não for bem utilizada resultará em uma exclusão dos alunos com menor nível.”.

Com base nas respostas obtidas pode-se inferir que a grande maioria dos usuários acredita que utilizar Inteligência Artificial em uma avaliação é uma

boa técnica, enquanto uma pequena minoria acredita que a utilização da técnica seja razoável.

Outra questão abordada no questionário foi quanto à interface do sistema. A interface é o cartão de visita de um software e é um quesito importante quanto à satisfação durante a utilização do aplicativo. No Gráfico 3 é possível verificar a opinião dos usuários quanto a questão. Nas respostas eram permitidas as seguintes opções: “Ótimo”, “Bom”, “Razoável” e “Ruim”.

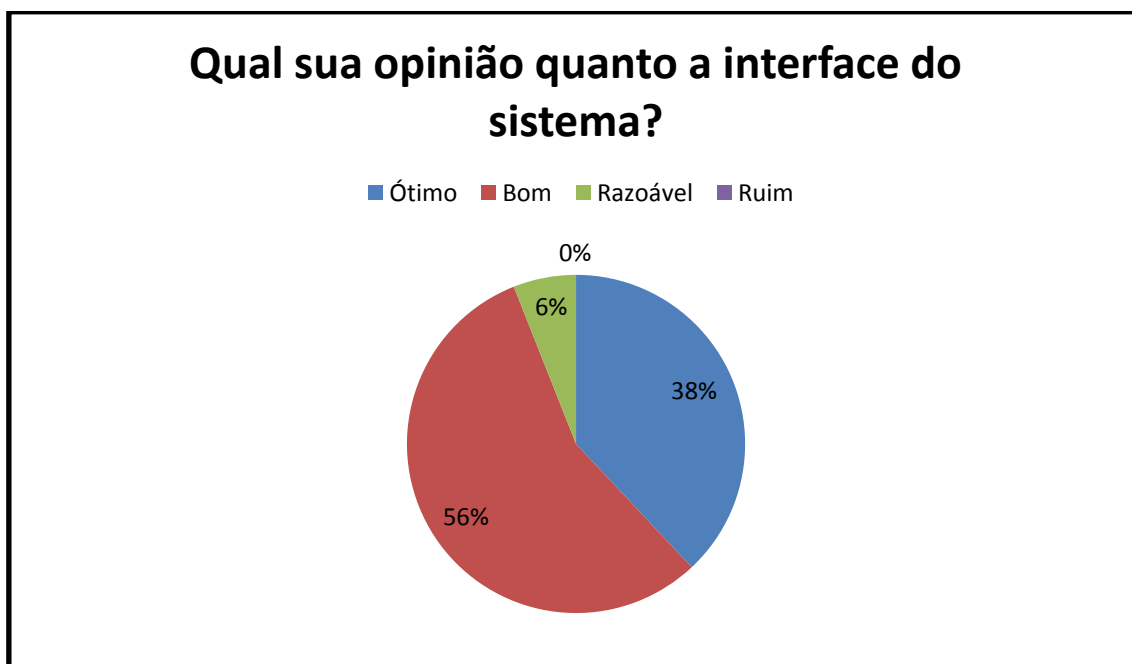


Gráfico 3: Opinião dos usuários quanto à interface do STI.

Nos resultados, 38% dos usuários responderam que acharam a interface do aplicativo ótima, enquanto 56% acharam a interface boa, e os outros 6% acharam a interface razoável. Com base nas respostas, pode-se inferir que de modo geral, o visual do aplicativo agradou de forma satisfatória aos usuários.

Ao final de cada avaliação, o aplicativo apresentava um resumo com a quantidade de erros, acertos e o nível atingido do aluno. No questionário foi perguntado se o usuário achou justo o nível que ele alcançou. No Gráfico 4 é possível verificar a opinião dos usuários quanto a questão. Nas respostas eram permitidas as seguintes opções: “Sim” e “Não”.

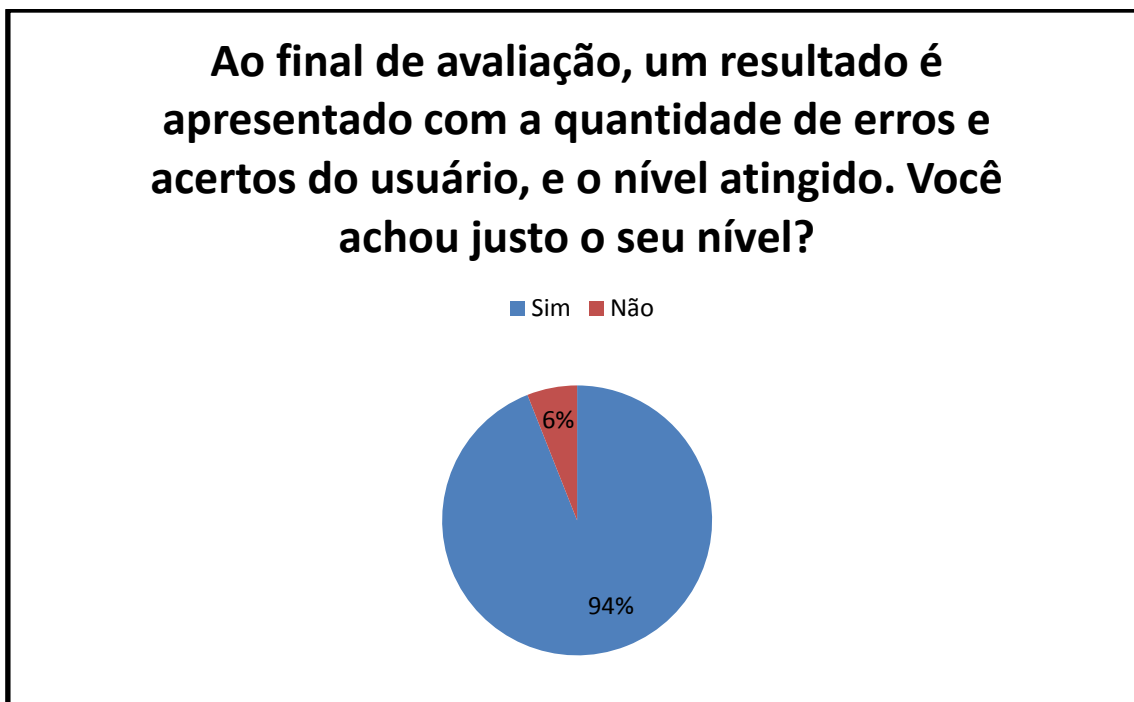


Gráfico 4: Opinião dos usuários quanto ao nível obtido na avaliação.

Nos resultados, 94% dos usuários acharam justo o nível atingido, enquanto 6% acharam que seu nível não foi justo.

Ao inserir cada questão no aplicativo, independente da complexidade da mesma, essa questão recebe nível 1 (um), e conforme os usuários a respondem, o seu nível pode variar de 1 (um) a 5 (cinco). Então, é compreensível a insatisfação de alguns usuários quanto ao seu nível, pois dependendo da sua sorte, no período inicial da utilização do aplicativo poderia ser sorteada uma pergunta com um nível de dificuldade mais alto, mesmo o usuário tendo um nível mais baixo.

Outra questão abordada no aplicativo foi quanto à clareza e a objetividade das perguntas. Um dos cuidados ao inserir as perguntas foi em procurar deixar a questão mais clara e objetiva possível, para que o usuário consiga compreender rapidamente a questão e não se confunda ao ler o enunciado. No Gráfico 5 é possível verificar a opinião dos usuários quanto a questão. Nas respostas eram permitidas as seguintes opções: “Sim”, “Não” e “As vezes”.

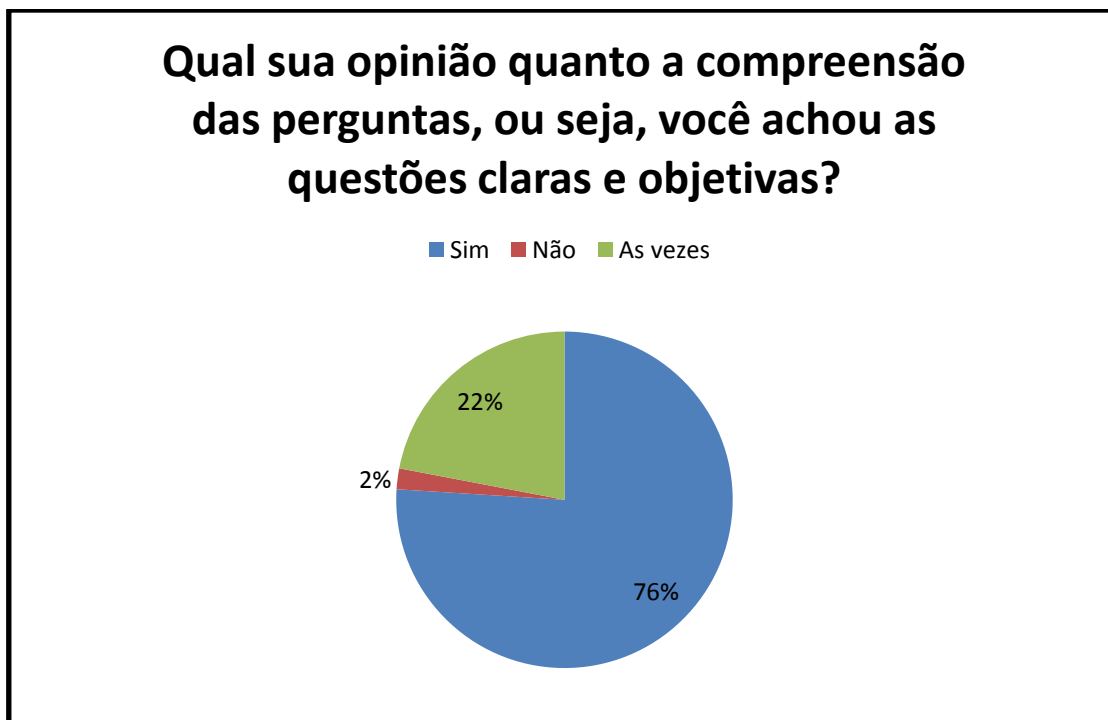


Gráfico 5: Opinião dos usuários quanto à clareza e objetividade das questões.

Nos resultados, 76% dos usuários acharam todas as questões claras e objetivas, 2% acharam as questões confusas, e os outros 22% acharam as questões confusas apenas em alguns momentos. Com base nas repostas, pode-se inferir que as perguntas inseridas no aplicativo são de fácil entendimento e não confundiram o usuário na realização do teste.

Uma questão importante para avaliar a satisfação do usuário, é o desempenho do aplicativo durante a sua utilização. Um software lento e com baixo desempenho, atrapalha na navegação e muitas vezes até impossibilita a sua utilização, pensando nisso, foi elaborada uma questão para avaliar as opiniões dos usuários quanto ao desempenho do aplicativo. No Gráfico 6 é possível verificar a opinião dos usuários quanto a questão. Nas repostas eram permitidas as seguintes opções: “Ótimo”, “Bom”, “Razoável” e “Ruim”.

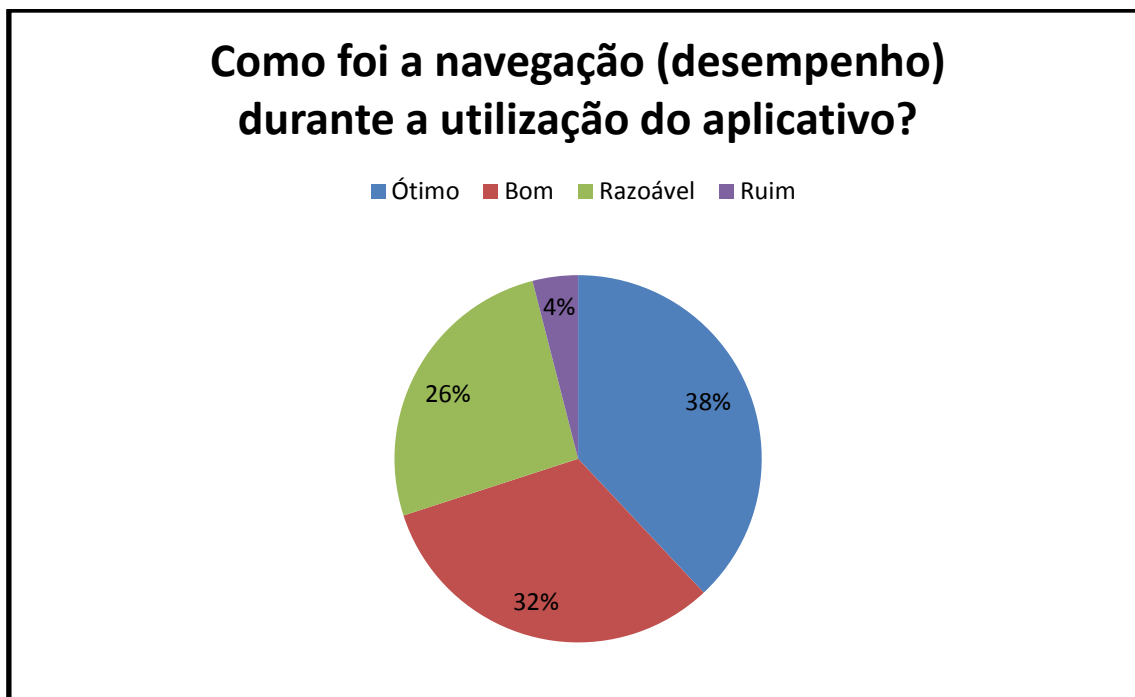


Gráfico 6: Opinião dos usuários quanto ao desempenho do aplicativo.

Nos resultados, 38% dos usuários acharam a navegação ótima, 32% acharam a navegação boa, outros 26% acharam razoável, enquanto 4% acharam a navegação ruim. Com base nas respostas, de modo geral, pode-se inferir que os usuários ficaram satisfeitos com o desempenho do aplicativo. Uma parte dos usuários achou o desempenho razoável, e uma pequena parte ficou insatisfeita. É compreensível que uma pequena parte dos usuários tenha avaliado a navegação como ruim, pois uma parte da turma do período matutino utilizou o aplicativo antes das correções encontradas na observação direta, estas correções influenciaram diretamente no desempenho do aplicativo. Porém, vale lembrar, que após as correções, não houve mais reclamações quanto ao desempenho do mesmo.

Quando o usuário errava uma questão, caso houvesse uma explicação cadastrada o aplicativo permitia a sua visualização. Foi elaborada uma questão sobre este tema no questionário, com o objetivo de avaliar o que o usuário achou desta funcionalidade. O Gráfico 7 é possível verificar a opinião dos usuários quanto a questão. Nas respostas eram permitidas as seguintes opções: “Ótimo”, “Bom”, “Razoável” e “Ruim”.

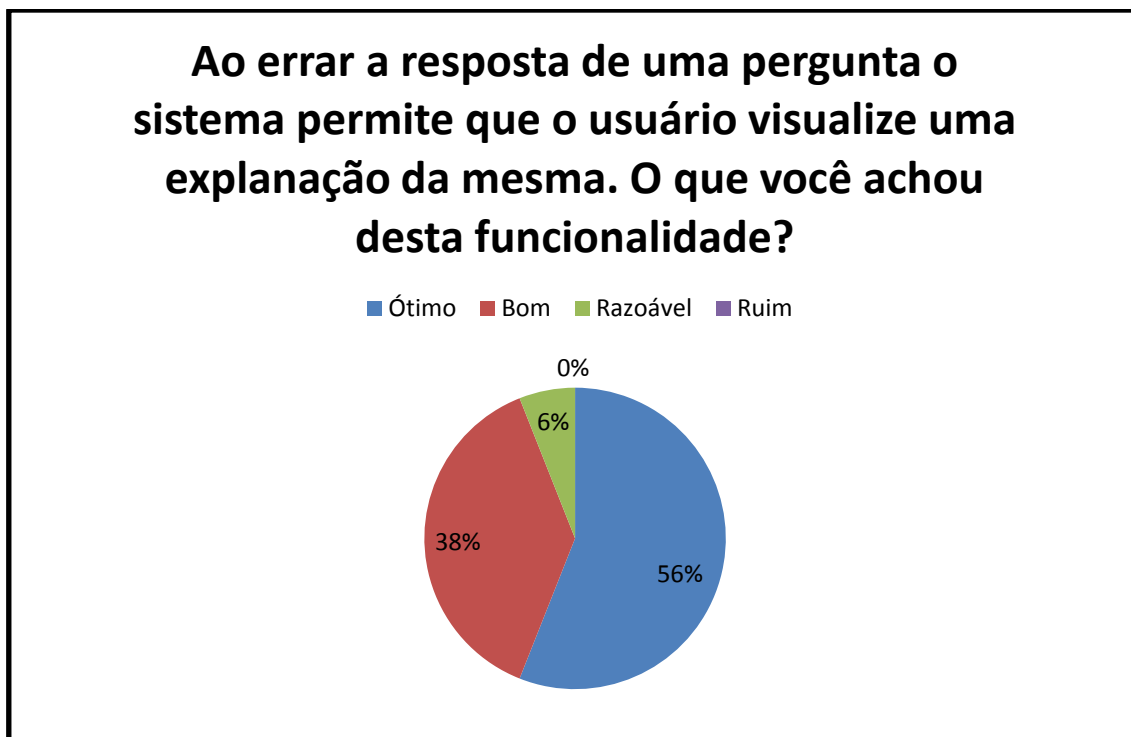


Gráfico 7: Opinião dos usuários quanto às explicações das questões.

Nos resultados, 56% dos usuários acharam ótima a opção de poder visualizar a explicação de uma pergunta ao errar uma questão, 38% acharam funcionalidade boa, enquanto 6% acharam razoável. Com base nas respostas, pode-se inferir que a opção de visualizar uma explicação teve uma boa aceitação por parte dos usuários, pois através dela é possível entender melhor sobre o tema, e porque errou a questão.

Outra questão abordada no questionário foi relacionada a usabilidade do sistema, esta questão tem como objetivo avaliar se o usuário achou o aplicativo fácil de manusear, ou teve dificuldades na utilização do mesmo. No Gráfico 8 é possível verificar a opinião dos usuários quanto a questão. Nas respostas eram permitidas as seguintes opções: “Ótimo”, “Bom”, “Razoável” e “Ruim”.

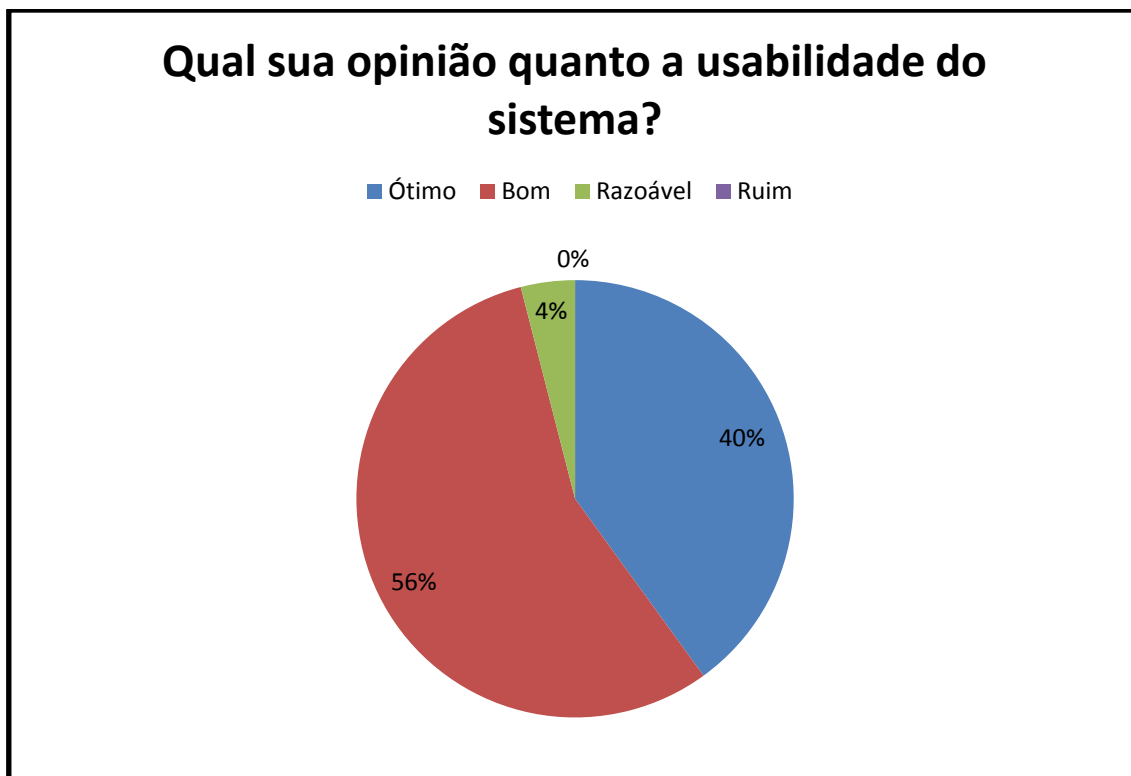


Gráfico 8: Opinião dos usuários quanto a usabilidade do STI.

Nos resultados, 40% dos usuários acharam ótima a usabilidade do aplicativo, outros 56% acharam boa, enquanto 4% acharam razoável. Com base nas respostas, pode-se inferir que de forma geral, os usuários ficaram satisfeitos quanto a usabilidade do aplicativo.

A última questão do questionário tinha como objetivo avaliar a satisfação na utilização do usuário de forma geral. Nesta questão foi perguntado se o usuário voltaria a utilizar o aplicativo após o término do curso. No Gráfico 9 é possível verificar a opinião dos usuários. Nas respostas eram permitidas as seguintes opções: “Sim”, “Não” ou “Talvez”.

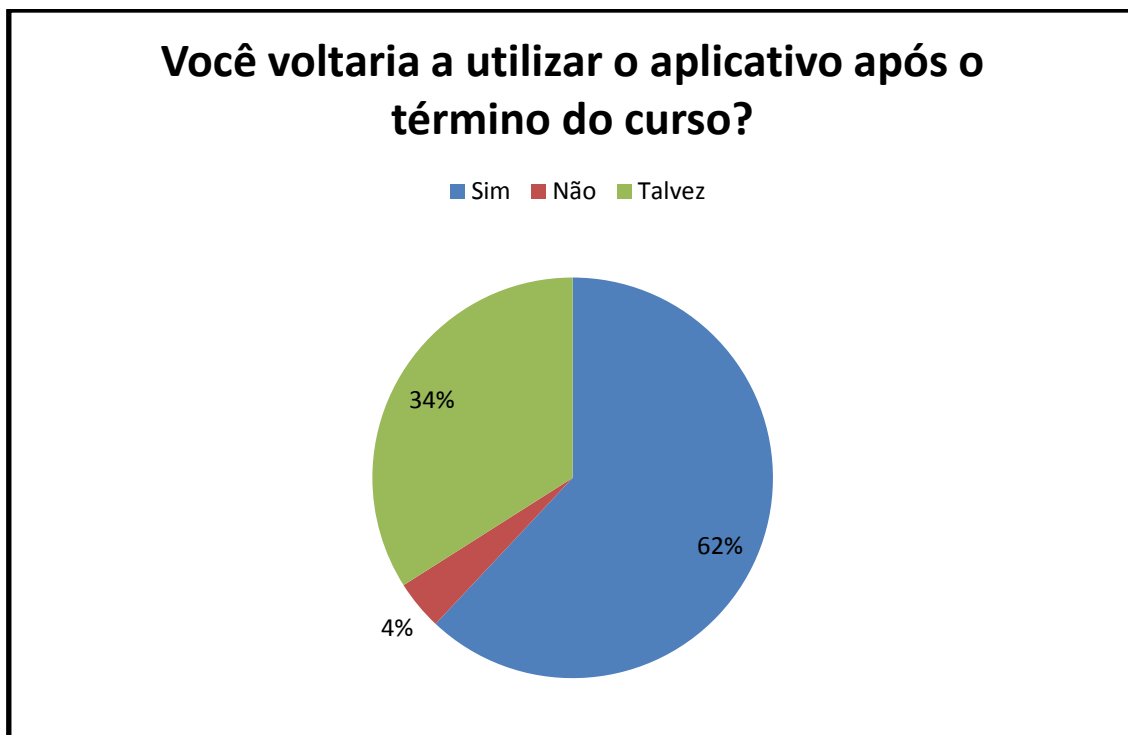


Gráfico 9: Opinião dos usuários quanto à reutilização do aplicativo após o término do curso.

Nos resultados, 62% dos usuários responderam que voltariam a utilizar o aplicativo, mesmo após terminar o curso, outros 34% responderam que talvez voltassem a utilizar, enquanto apenas 4% responderam que não voltariam a utilizá-lo. Com base nas respostas, pode-se inferir que a maior parte dos usuários tiveram uma boa experiência com o aplicativo e voltariam a utilizá-lo, e uma pequena parte mostrou-se insatisfeito com a experiência na utilização.

Ao final do questionário, havia um espaço para o usuário fazer alguma crítica, sugestão ou elogio ao aplicativo, alguns usuários utilizaram este espaço para expressar sua opinião quanto ao aplicativo:

Na observação direta também foi notado que alguns alunos achavam 60 (sessenta) segundos insuficientes para responder a questão, com base nessas observações o tempo para responder cada questão foi aumentado para 90 (noventa) segundo com a turma vespertina.

- “mais tempo para responder”.

Alguns usuários destacaram a característica do software em poder avaliar os alunos que fizeram o curso de programação, tanto por parte do tutor

quanto por parte do aluno, ou seja, ambos podem acompanhar o desempenho do aluno através do aplicativo.

- “gostei pois nos permite ver como se saímos no curso.”
- “É muito bom pois permite poder analisar o aluno.”
- “Foi bom eu utilizar para ver minha capacidade e ver o que eu aprendi do curso.”

Também houve uma crítica ao aplicativo, na observação direta um aluno teve dificuldades em efetuar o cadastro do usuário. Após algumas tentativas, foi verificado que o que estava impossibilitando o cadastro era o caractere “apóstrofo”(‘) contido no sobrenome do aluno, que por algum motivo, o sistema não estava aceitando. Após remover o caractere em questão, tudo ocorreu normalmente.

- “Melhorias na admissão de caracteres no cadastro. Ex: Meu nome é Gabriel D'Ávila, meu cadastro dava erro por causa do apóstrofo. Mas fora isso, tudo certo.”

5.4- CONSIDERAÇÕES DA VALIDAÇÃO

As técnicas do questionário e da observação direta foram consideradas satisfatórias para avaliar o protótipo, pois possibilitou avaliar se o STI proposto apresentou alguma contribuição positiva ao usuário. Como podemos acompanhar nos resultados do questionário, a grande maioria dos usuários avaliou positivamente o aplicativo, tanto as questões de usabilidade e desempenho, como também o conteúdo e a funcionalidade do aplicativo.

6- CONSIDERAÇÕES FINAIS

Neste trabalho foi desenvolvido um protótipo chamado Mazk STI com a finalidade de automatizar os conteúdos em um sistema tutor inteligente. Neste protótipo foi desenvolvido um módulo tutor para a automatização de conteúdos e os demais módulos que completam o sistema: módulo administrador, módulo aprendiz. Além do protótipo, também foi desenvolvido um servidor para o armazenamento dos dados, que é onde reside o agente inteligente.

O protótipo desenvolvido oferece ao professor uma interface simples e amigável para a automatização de conteúdos do STI. Quando o usuário do STI for do tipo tutor, o módulo tutor é carregado e neste módulo, o professor pode inserir, editar ou excluir uma pergunta, e inserir ou substituir uma explicação. No editor da explicação é permitido carregar imagens, colocar textos negritos, sublinhados e itálicos para facilitar o entendimento da questão.

O protótipo do STI desenvolvido conta com funcionalidades que estimulam os alunos e professores a o utilizarem. O aluno se sente estimulado a utilizar um aplicativo quando existe um desafio, como podemos acompanhar no dia dos testes, onde a maioria dos alunos faziam os testes duas ou mais vezes, para poderem alcançar um nível maior do que já haviam conseguido. Para os professores, além de acompanhar o rendimento dos alunos, o STI também conta com a colaboração do aluno ao nivelar as perguntas da base de dados. As perguntas são inseridas na base com o nível mínimo 1 (um), conforme os alunos as respondem, os níveis das perguntas também alteram dinamicamente, desta forma o professor pode conhecer a real dificuldade daquela pergunta para a sua turma, assim, quanto mais se utiliza o aplicativo, mais precisos serão os níveis obtidos pelas perguntas.

Todos os objetivos propostos no início do trabalho foram alcançados satisfatoriamente. Todas as perguntas que foram respondidas nas avaliações, e suas respectivas explicações, foram inseridas com o Mazk STI. Com a utilização do aplicativo, as perguntas do banco de dados, que antes possuíam todas nível 1 (um), agora possuem níveis mais próximos da realidade, variando de 1 (um) a 5 (cinco). Os níveis dos usuários também estão mais próximos dos reais, no momento do cadastro todos usuários recebem o nível 1 (um), porém, atualmente os níveis registrados variam de 1 (um) a 5 (cinco).

As críticas recebidas no dia da validação do protótipo foram consideradas positivas e certamente irão contribuir para futuras melhorias e evoluções do protótipo. Levando em considerações as expectativas do projeto, conclui-se que o protótipo desenvolvido atendeu as expectativas esperadas de forma satisfatória.

Trabalhos Futuros:

- Criar uma tela de parametrização, ou seja, uma tela onde se possa configurar as informações que hoje são estáticas, como o tempo para responder cada pergunta, a quantidade de perguntas respondida em cada teste, etc.
- Criar uma tela para acompanhar o desempenho do aluno.
- Organizar os dados do STI por disciplinas, permitindo a utilização do STI para outras disciplinas além de programação.

7- REFERÊNCIAS

- ALVARES, L. O.; SICHMAN, J. S. **Introdução aos sistemas multiagentes**. In: MEDEIROS, C. M. B. (Ed.). *Jornada de Atualização em Informática (JAI'97)*. Brasília: UnB, 1997. cap. 1, p. 1–38.
- ANDROID. **Developers**. Disponível em: <<http://developer.android.com/sdk/index.html?hl=i/>>. Acesso em: 19 nov. 2014.
- ANDROIDPIT. **AndroidPIT Internacional**. Disponível em: <<http://www.androidpit.com.br/androidpit>>. Acesso em: 20 nov. 2014.
- ARTERO, Almir Olivette. *Inteligência Artificial: Teórica e Prática*. São Paulo: Livraria da Física, 2009. 230 p.
- COPPIN, B. **Inteligência Artificial**. 1a Edição. Rio de Janeiro: LTC, 2004.
- DA SILVA, Daniel Rodrigues. **Desenvolvimento de um sistema especialista para auxílio e diagnóstico de doenças em macieiras, baseado no ciclo de vida de um programa**. *Repositório de TCCs-Sistemas de Informação2* (2014).
- DA SILVA, Juarez B. et al. **Uso de dispositivos móveis para acesso a Experimentos Remotos na Educação Básica**, 2013.
- DA SILVA, Luiz Fernando et al. **A Utilização de Dispositivos Móveis como Ferramenta de Ensino Aprendizagem**. *Revista Pedagogia em Foco/Faculdade Aldete Maria R454 Alves-FAMA. Iturama, MG: Fama*, v. 1, n. 1, 2010-[on-line]. Jan./dez. 2013. Anual., p. 81, 2013.
- DANTAS, Teresa Cristina et al. **AprendEAD: Ambiente para Educação à Distância Apoiado em Agentes**. *Cadernos do IME-Série Informática*, v. 23, p. 13-20, 2013.
- DE LACERDA, Estéfane GM; DE CARVALHO, ACPLF. **Introdução aos algoritmos genéticos**. *Sistemas inteligentes: aplicações a recursos hídricos e ciências ambientais*, v. 1, p. 99-148, 1999.
- EBELL, Mark H. **Artificial neural networks for predicting failure to survive following in-hospital cardiopulmonary resuscitation**. *The Journal of family practice*, v. 36, n. 3, p. 297-303, 1993.
- ECLIPSE. **The Eclipse Foundation**. Disponível em: <<http://eclipse.org/downloads/>>. Acesso em: 19 nov. 2014.
- EDUCATORS, Connecting Computing. **CS 10K Initiative**. Disponível em: <<http://www.computingportal.org/cs10k>>. Acesso em: 19 nov. 2014.

- FEIGENBAUM, Edward A., Avron Barr, and Paul R. Cohen, eds. **The handbook of artificial intelligence**. New York: Addison-Wesley, 1989.
- FOGEL, L.J., OWENS, A.J. & WALSH, M.J.(1966). **Artificial Intelligence through Simulated Evolution**. John Wiley
- FRIGO, Luciana Bolan; POZZEBON, Eliane; BITTENCOURT, Guilherme. **O papel dos agentes inteligentes nos sistemas tutores inteligentes**. In: World Congress on Engineering and Technology Education. 2004. p. 86.
- GIRAFFA, L. M. M. **Uma arquitetura de tutor utilizando estados mentais**. Porto Alegre: CPGCC da UFRGS, 1998. Tese de Doutorado.
- GOMIDE, Fernando Antonio Campos; GUDWIN, Ricardo Ribeiro. Modelagem, controle, sistemas e lógica fuzzy. *SBA Controle & Automação*, v. 4, n. 3, p. 97-115, 1994.
- GOOGLE. **Self-DrivingCar Test**. 2014. Disponível em: <<http://www.google.com/about/careers/lifeatgoogle/self-driving-car-test-steve-mahan.html>>. Acesso em: 16 set. 2014.
- HERPICH, Fabrício; FRANCISCATTO, Roberto. **Desenvolvimento de um Protótipo Utilizando Técnicas Mobile Learning voltadas ao Ensino**. In: Anais dos Workshops do Congresso Brasileiro de Informática na Educação. 2013.
- HUBNER, Jomi Fred. **Um Modelo de Reorganização de Sistemas Multiagentes**. São Paulo, 2003. 224 p
- IROBOT. **I Robot Roomba**: Vacuum Cleaning Robot. Disponível em: <<http://www.irobot.com/us/learn/home/roomba.aspx>>. Acesso em: 16 set. 2014.
- KOVÁCS, Zsolt L. **Redes neurais artificiais**. Editora Livraria da Física, 2002.
- KUKULSKA-HULME, Agnes; TRAXLER, John (Ed.). **Mobile learning: A handbook for educators and trainers**. Psychology Press, 2005.
- LESSER, Victor. **Cooperative Multi-Agent Systems: A Personal View of the State of the**
- MACORATTI, José Carlos. **O processo de Software**. Disponível em: <http://www.macoratti.net/proc_sw1.htm>. Acesso em: 18 nov. 2014.
- MARGHESCU, G.; CHICIOREANU, T.D.; MARGHESCU, I., "An Alternative to the Traditional Methods in Education - M-Learning: a Glance into the Future," *EUROCON, 2007. The International Conference on "Computer as a Tool"*, vol., no., pp.2410,2414, 9-12 Sept. 2007. doi:

10.1109/EURCON.2007.4400623

MEC (Org.). **Ministério distribuirá tablets a professores do ensino médio**. Disponível em:

<http://portal.mec.gov.br/index.php?option=com_content&id=17479>. Acesso em: 02 fev. 2012.

MILLINGTON, Ian; FUNGE, John. **Artificial intelligence for games**. CRC Press, 2009.

MINATTO, Samuel Ghisleri. **Ambiente Inteligente de Aprendizagem para Dispositivos Móveis**. Trabalho de Conclusão de Curso – Universidade Federal de Santa Catarina, v. 1 135 p. 2013.

NETBEANS. **NetBeans IDE**. Disponível em: <<http://dlc-cdn.sun.com/netbeans/7.1/rc1/>>. Acesso em: 19 nov. 2014.

NORVIG, Peter; RUSSELL, Stuart. **Inteligência Artificial**, 3ª Edição. Elsevier Brasil, 2014.

ORACLE. **Oracle Technology Network**. Disponível em: <<http://www.oracle.com/technetwork/developer-tools/developer-suite/downloads/101202winsoft-087370.html>>. Acesso em: 19 nov. 2014.

PAIVA, Lilian R. Mendes et al. **Aplicação para dispositivos móveis utilizando tecnologias interativas**: a Realidade Virtual e Aumentada aplicada ao estudo da anatomia humana. X Encontro Anual de Computação-EnAComp-IFTM, 2013.

PAULA FILHO, Wilson de Pádua. **Engenharia de Software**: fundamentos, métodos e padrões. São Paulo: LTC Editora, 2000.

PLACEMENT, School Advanced. **AP Computer Science Principles**. Disponível em: <<http://csprinciples.cs.washington.edu/>>. Acesso em: 19 nov. 2014.

PORTAL BRASIL. **Professores do ensino médio de escolas públicas receberão tablets no segundo semestre**. Disponível em: <<http://www.brasil.gov.br/noticias/arquivos/2012/02/03/professores-do-ensino-medio-de-escolas-publicas-receberao-tablets-no-segundo-semester>>. Acesso em 20 out 2014.

POZZEBON, Eliane ; MACHADO, G. ; MINATTO, S. G. ; IZE, M. J. ; FRIGO, Luciana B.. **Programação de Computadores no Ensino Médio**. In: International Conference on Interactive Computer, 2013, Florianópolis-SC-Brasil. International Conference on Interactive Computer. Florianópolis-SC:

ICBL, 2013. v. 1. p. 203-206.

RAABE, Andre Luis Alice. **Uma proposta de arquitetura de sistema tutor inteligente baseada na teoria das experiências de aprendizagem mediadas**. 2005.

RICKEL, J. W. **Intelligent Computer-Aided Instruction: A Survey Organized Around System Components**. S.1.: IEEE Transactions on Systems, Man, and Cybernetics, v. 19, n.1, p.40-57, 1989.

ROCHADEL, Willian et al. **Utilization of remote experimentation in mobile devices for education**. In: Global Engineering Education Conference (EDUCON), 2012 IEEE. IEEE, 2012. p. 1-6.

ROSA, João Luis Garcia. **Fundamentos da Inteligência Artificial**. Editora LTC. Rio de Janeiro, 2011.

RUIC, Gabriela. **3 em cada 10 brasileiros são donos de smartphones**. Disponível em: <<http://exame.abril.com.br/tecnologia/noticias/3-em-cada-10-brasileiros-sao-donos-de-smartphones>>. Acesso em: 26 jun. 2013.

RUSSELL, Stuart; NORVIG, Peter. **Inteligência Artificial**. Ed. Campus, tradução da 2ª edição, 2004.

SANTOS, Daniel Tenório dos. **Principais Aplicativos de Socket**. Disponível em: <<http://danieltenorio411.blogspot.com.br/2012/06/principais-aplicativos-de-socket.html>>. Acesso em: 19 nov. 2014.

SANTOS, Giovanni Almeida; RISSOLI, Vandor Roberto Vilardi. **Benefícios no Uso de um Assistente Inteligente no Ensino-Aprendizagem de Programação Computacional**.

SCHWAB, Brian. **AI game engine programming**. Cengage Learning, 2009.

SILVA, Maurílio; TELES, Vivianny Duarte. **O aprendizado de língua inglesa em dispositivos móveis através da aplicação de um sistema tutor inteligente**. In: Anais do Simpósio Brasileiro de Informática na Educação. 2011.

SILVEIRA, Ricardo A. **Inteligência Artificial em Educação: um modelo de sistema tutorial inteligente para microcomputadores**. Porto Alegre, FAGED/PUCRS, 1992.

SIMÕES, Priscyla Waleska Targino de Azevedo ET AL. **Perspectivas da Utilização da Inteligência Computacional Médica a partir das Redes Bayesianas e Agentes Tutores no Ensino de Lombalgia**. Inova Saúde, v. 2, n. 1, 2013.

SOMMERVILLE, Ian. **Engenharia de Software**. 8. ed. São Paulo: Pearson Education do Brasil, 2007. 568 p.

SOUZA, Marcelo; VAHLICK, Adilson. **Influência dos jogos no campo da inteligência artificial**. REAVI-Revista Eletrônica do Alto Vale do Itajaí, v. 2, n. 2, p. 157-160, 2013.

TACLA, Cesar Augusto. **Sockets UDP, TCP e MULTICAST**. Disponível em: <<http://www.inf.ufsc.br/~bosco/ensino/ine5645/2014-1/0050-Sockets.pdf>>.

Acesso em: 20 nov. 2014.

TAROUCO, L.; ROCKENBACH M.. **Objetos de aprendizagem para M-Learning**. In: CONGRESSO INTERNACIONAL SOBRE O USO DE OBJETOS DE APRENDIZAGEM, 1., 2004, São Paulo.

WHITBY, Blay. **Inteligência artificial**: um guia para iniciantes. Madras, 2004.

APENDICE 1

Questionário de Validação do Mazk STI

Qual sua opinião em relação a utilização de tecnologias móveis no processo educacional?

- Ótimo
- Bom
- Razoável
- Ruim

Qual sua opinião quanto a aplicar Inteligência Artificial em uma avaliação, onde o nível dos alunos e das perguntas se modificam conforme o seu histórico?

- Ótimo
- Bom
- Razoável
- Ruim

Qual sua opinião quanto a interface do sistema?

- Ótimo
- Bom
- Razoável
- Ruim

Ao final de cada avaliação, um resultado é apresentado com a quantidade de erros e acertos do usuário, e o nível atingido. Você achou justo o seu nível?

- Sim
- Não

Qual sua opinião quanto a compreensão das perguntas, ou seja, você achou as perguntas claras e objetivas?

- Sim
- Não

- As vezes

Como foi a navegação (desempenho) durante a utilização do aplicativo?

- Ótimo
- Razoável
- Bom
- Ruim

Ao errar a resposta de uma pergunta o sistema permite que o usuário visualize uma explicação da mesma. O que você achou desta funcionalidade?

- Ótimo
- Bom
- Razoável
- Ruim

Qual sua opinião quanto a usabilidade do sistema?

- Ótimo
- Bom
- Razoável
- Ruim

Você voltaria a utilizar o aplicativo após o término do curso?

- Sim
- Não
- Talvez

Escreva alguma crítica, sugestão ou elogio ao aplicativo.