

DIEGO CRISTIANO ANDRADE

DICA: o software em função do Ensino-Aprendizagem

Assis
2015

DIEGO CRISTIANO ANDRADE

DICA: o software em função do Ensino-Aprendizagem

Trabalho de Conclusão de Curso de Bacharelado em Ciência da Computação da Fundação Educacional do Município de Assis, como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Luiz Carlos Begosso

Área de Concentração: Informática

Assis
2015

FICHA CATALOGRÁFICA

ANDRADE, Diego Cristiano

DICA: o software em função do Ensino-Aprendizagem / Diego Cristiano Andrade. Fundação Educacional do Município de Assis – FEMA – Assis, 2015.
Número de páginas: 50.

Orientador: Luiz Carlos Begosso.

Trabalho de Conclusão de Curso – Instituto Municipal de Ensino Superior de Assis – IMESA.

1. Software Educacional.
2. Ensino-Aprendizagem.

CCD: 001.6
Biblioteca da FEMA

DICA: o software em função do Ensino-Aprendizagem

DIEGO CRISTIANO ANDRADE

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, analisado pela seguinte comissão examinadora:

Orientador: Prof. Dr. Luiz Carlos Begosso

Analisador: Prof. Me. Felipe Alexandre Cardoso Pazinatto

Assis
2015

DEDICATÓRIA

Dedico este trabalho ao meu pai Aparecido Joaquim Andrade (*in memoriam*) e minha mãe Maria José Andrade, pois eles são a base de quem sou hoje e sempre acreditaram em meus sonhos. Dedico também a toda minha família, namorada, amigos e ao meu orientador Prof. Dr. Luiz Carlos Begosso pelo apoio e incentivo para concluir este trabalho.

AGRADECIMENTOS

Agradeço primeiramente a Deus acima de tudo, pois é a luz que me guia e a força que me faz levantar todos os dias.

Aos meus pais por me apoiarem sempre que necessitei e a incentivar para continuar sempre em frente.

A toda minha família, por também me apoiarem.

A minha namorada Rayssa Fernandes, que ao longo deste trabalho me ajudou direta e indiretamente para conclusão do mesmo e sempre esteve ao meu lado.

Ao meu orientador, Prof. Dr. Luiz Carlos Begosso, pelos ensinamentos transmitidos e colaboração prestada.

A todos meus amigos e colegas de classe, pelo companheirismo e motivação para não desistir.

E a todos que direta ou indiretamente contribuíram para a realização deste trabalho.

“Aprendi que são os pequenos acontecimentos diários que tornam a vida espetacular.”

William Shakespeare (1564 – 1616)

RESUMO

Os softwares educacionais estão cada vez mais presentes em nossas vidas, direta ou indiretamente, seja com jogos digitais, softwares tutoriais ou mesmo softwares de ensino-aprendizagem, onde visa melhorar e inovar a forma de ensino.

Este trabalho tem por objetivo o desenvolvimento de um software educacional, onde proporcionará aos professores a criação de avaliações e testes e o oferecimento dessas atividades para os alunos.

Para a realização deste projeto, foi desenvolvida uma aplicação web, de modo que a linguagem de programação a ser utilizada seja Java com interface de desenvolvimento Eclipse IDE, e XHTML.

Palavras-Chave: Software Educacional; Ensino-Aprendizagem; Professor.

ABSTRACT

The educational software are increasingly present in our lives, directly or indirectly, either with digital games, software tutorials or even teaching-learning software, which aims to improve, the education.

This work aims at the development of educational software, which will provide teachers the creation of evaluations and tests and the offering these activities for students.

For this Project, a web application will be developed using the programming language Java, with development interface Eclipse IDE and XHTML.

Keywords: Educational Software; Teaching-Learning; teacher.

LISTA DE ILUSTRAÇÕES

Figura 1 - Ensino Tradicional (extraído de Guerra, 2010).	19
Figura 2 - Representação do paradigma instrucionista (extraído de Guerra, 2010).	21
Figura 3 - Representação do paradigma construcionista (extraído de Guerra, 2010).	21
Figura 4 - Exemplo de correção da avaliação.	27
Figura 5 - Exemplo de ranking.	27
Figura 6 - Diagrama de Classe.	28
Figura 7 - Diagrama de Caso de Uso.	29
Figura 8 - Tela de acesso ao sistema DICA.	30
Figura 9 - Tela de Cadastros.	31
Figura 10 - Consultas.	31
Figura 11 - Criação de avaliação.	32
Figura 12 - Tela para elaboração de perguntas.	33
Figura 13 - Respostas.	34
Figura 14 - Resultado final.	34
Figura 15 - Consulta de ranking dos alunos.	35
Figura 16 - Banco de dados DICA.	36
Figura 17 - Tabela aluno do Banco de Dados.	37
Figura 18 - Projeto DICA.	37
Figura 19 - Conexão com Banco de Dados.	38
Figura 20 - Classe do pacote Domain.	39
Figura 21 - Classe do pacote DAO.	40
Figura 22 - Classe do pacote Bean.	41
Figura 23 - Classe do pacote Test.	42
Figura 24 - Resultado do teste JUnit.	43
Figura 25 - Classe JSFÚtil.	43
Figura 26 - Páginas Web da aplicação.	44
Figura 27 - Página XHTML do Login.	44
Figura 28 - Interface gráfica da tela Login.	45
Figura 29 - Interface gráfica do Template.	45

LISTA DE ABREVIATURAS E SIGLAS

JSF	Java Server Faces
JSP	Java Server Pages
SGBD	Sistema de Gerenciamentos de Banco de Dados

SUMÁRIO

1. INTRODUÇÃO	14
1.1 OBJETIVOS.....	15
1.2 JUSTIFICATIVA.....	15
1.3 MOTIVAÇÃO	15
1.4 ESTRUTURA DO TRABALHO	16
1.5 PERSPECTIVAS DE CONTRIBUIÇÃO	16
1.6 METODOLOGIAS DE PESQUISA.....	16
1.7 RECURSOS NECESSÁRIOS.....	17
2. ENSINO-APRENDIZAGEM	18
2.1 INTRODUÇÃO AO ENSINO	18
2.1.1 Conhecendo a Informática e sua História	18
2.1.2 Ensino Tradicional	19
2.1.2.1 Vantagens e Desvantagens	20
2.1.3 Ensino com Informática	20
2.1.4 O uso de Softwares no Ambiente Educativo	20
2.1.4.1 As Duas Abordagens de Ensino	20
2.1.5 Tipos de Software Educacional	22
2.2 TRABALHOS CORRELATOS	22
2.2.1 Duolingo	22
2.2.2 Brainyoo	23
3. TECNOLOGIAS UTILIZADAS	24
3.1 JAVA.....	24
3.2 ECLIPSE IDE	24
3.3 APACHE TOMCAT.....	24
3.4 JSF E PRIMEFACES	25
3.5 POSTGRESQL.....	25
4. PROPOSTA DO TRABALHO	26
4.1 PROJETO DICA.....	26
4.1.1 Correção da Avaliação	26
4.1.2 Ranking dos Alunos	27

4.2 DIAGRAMA DE CLASSE E DIAGRAMA DE CASO DE USO.....	28
4.3 ACESSO AO SISTEMA.....	29
4.4 USUÁRIO PROFESSOR.....	30
4.4.1 Cadastros	30
4.4.2 Consultas	31
4.4.3 Elaborando a Avaliação	32
4.5 USUÁRIO ALUNO.....	33
4.5.1 Realizando a Avaliação	33
4.5.2 Consultando o Ranking dos Alunos	35
4.6. IMPLEMENTAÇÃO DO DICA.....	35
4.6.1 Banco de Dados	36
4.6.2 Desenvolvimento Web	37
4.6.2.1 Pacote Factory.....	38
4.6.2.2 Pacote Domain.....	39
4.6.2.3 Pacote DAO.....	40
4.6.2.4 Pacote Bean.....	41
4.6.2.5 Pacote Test.....	42
4.6.2.6 Pacote Útil.....	43
4.6.2.7 Páginas Web.....	44
4.6.2.7.1 <i>Template</i>	45
5. CONCLUSÃO	47
REFERÊNCIAS	48

1. INTRODUÇÃO

É de fundamental importância o uso de tecnologia na educação visando melhorar a concentração e o rendimento escolar, já que nos dias de hoje jovens passam um tempo notável em seus computadores e celulares. Um software educacional tem o intuito de chamar a atenção dos jovens por meio de jogos visando a aprendizagem (Falkembach, 2005).

Os softwares educativos estão cada vez mais presentes no ensino, e são utilizados desde chamadas à brincadeiras dentro das salas de aula até para o ensino de teorias. O objetivo desta prática tem relação com a possibilidade de potencializar atitudes pedagógicas e inovar o sistema de aprendizagem (Savi, 2008).

Outra forma comum nas escolas, hoje, é como as atividades extra classe estão utilizando os computadores para o ensino da informática, no entanto, sem mudar o método tradicional do ensino, essa abordagem não se encaixa nos intenderes da “informática na educação”. Esta “informática na educação” reforça o professor a utilizar da sua disciplina para ter conhecimento sobre os potenciais educacionais do computador e ser capaz de alternar as atividades tradicionais de ensino-aprendizagem com atividades que usam o computador (Valente, 1999).

Se o professor utilizar dessas novas práticas, os alunos serão atraídos a essa forma de ensino tornando-se cada vez mais cúmplices da tecnologia. Porém, em alguns adultos incomoda o fato de jovens estarem conectados a jogos, uma vez que, segundo a opinião dos adultos, os jogos possuem apenas o lado negativo. No entanto é necessário que os adultos aceitem o fato de que os jogos digitais trazem benefícios no desenvolvimento e no processo educativo dos jovens (Pietro, 2005).

De acordo com Aranha (2006), os jogos são vistos como ferramentas altamente atrativas aos estudantes e essenciais para o treinamento educacional e mental.

Essas constantes mudanças científicas e tecnológicas iguais o mundo tem passado ultimamente possibilitou o surgimento de uma nova cultura do saber, a cultura digital ou cibercultura (Haguenauer, 2007).

Já usufruindo dessa nova cultura, os jovens ao invés de utilizarem os livros ou cadernos para aprender, iriam por meio dessa aplicação, desenvolver suas atividades de forma digital, onde aprenderiam o conteúdo proposto pelo professor de maneira simples, com um

ambiente diferente do conhecido e uma forma prazerosa de aprender (Bittencourt et al, 2003).

1.1 OBJETIVOS

Este trabalho tem por objetivo o desenvolvimento de um software direcionado para educação, apto a enriquecer as aulas, atraindo a atenção dos jovens através de aplicação digital, auxiliando o docente no processo ensino-aprendizagem.

O software proporciona aos professores a criação e o oferecimento de atividades para os alunos, como forma de avaliações e testes. O sistema está pronto a atender a todas as disciplinas desde o ensino fundamental até o médio.

1.2 JUSTIFICATIVA

É de fundamental importância o uso do computador nas escolas, no qual ele se torna eficaz, aumentando a motivação dos alunos e criação de atividades que constituem oportunidades especiais para aprender e resolver enigmas (Moratori, 2003).

Como a forma de aprendizagem vem se adequando a tecnologia, o uso de um novo método de ensino se faz necessário, onde então os softwares educacionais entram como auxílio à esse método, pois possibilita o aumento da motivação dos alunos e diversão ao mesmo tempo em que eles aprendem algum conteúdo.

1.3 MOTIVAÇÃO

A principal motivação de fazer este trabalho é ter crescido no mundo dos games, tendo sempre a curiosidade de saber como são desenvolvidos, e o desejo de criar um software para ajudar pessoas, porém é necessário ter escolas com infraestrutura tecnológica e professores capacitados para se adequar a esse método de ensino.

1.4 ESTRUTURA DO TRABALHO

O presente trabalho está estruturado em cinco capítulos. O capítulo 1 estabelece os objetivos, a justificativa e a motivação para o seu desenvolvimento. O capítulo 2 apresenta as bases teóricas que fundamentam o trabalho e pesquisa do ensino-aprendizagem e que sustenta e justifica a realização do trabalho, apresentando alguns trabalhos correlatos que utilizam software educacional no ensino-aprendizado. O capítulo 3 discute todas as tecnologias necessárias para o desenvolvimento deste software. Para atender os objetivos estabelecidos, o capítulo 4 apresenta a proposta do trabalho e a implementação do projeto. Finalmente, o capítulo 5 apresenta a conclusão e trabalhos futuros.

1.5 PERSPECTIVAS DE CONTRIBUIÇÃO

A aplicação tem como objetivo auxiliar no ensino-aprendizado, aliando-se ao corpo docente com o intuito de aprimorar a educação, onde possa contribuir para o método de aprendizagem.

1.6 METODOLOGIAS DE PESQUISA

As metas e objetivos deste trabalho foram alcançados através de pesquisas teóricas, leitura de artigos acadêmicos, artigos científicos, livros, monografias, dissertações, teses, guias práticos e técnicos, livros e fontes digitais, tornando possível a elaboração dessa aplicação para buscar a comprovação e eficiência do mesmo.

A metodologia para a execução deste projeto de conclusão de curso é experimental, visto, que seu objetivo, além de apoiar a pesquisa acadêmica, é desenvolver um software que auxilie na educação.

1.7 RECURSOS NECESSÁRIOS

Para desenvolver a pesquisa foram necessários os recursos de hardware e software citados abaixo:

- **Hardware**
 - Computador
 - Processador Intel Core i3 2.40Ghz.
 - Disco Rígido *5400 RPM SATA de 750 GB.*
 - Memória de 1333 Mhz DDR3 4 GB.
- **Software**
 - ***Eclipse IDE***– Ambiente de desenvolvimento para a tecnologia Java. Será utilizada a versão Kepler.
 - ***Java Development Kit***– Conjunto de ferramentas que possibilita o desenvolvimento com a tecnologia Java.
 - ***Java Runtime Environment***– Conjunto de ferramentas que possibilita compilar o código Java em qualquer sistema operacional.

2. ENSINO-APRENDIZAGEM

Neste capítulo será abordada a teoria que sustenta o presente estudo deste trabalho, softwares em função do ensino-aprendizado, e também apresenta trabalhos correlatos.

2.1 INTRODUÇÃO AO ENSINO

2.1.1 Conhecendo a Informática e sua História

O uso dos computadores em escolas tem se tornado comum nos últimos tempos, desde escolas de ensino primário até ensino superior, tanto público como particulares. Um dos pontos mais discutidos em fóruns de debate educacional é a introdução dos computadores na educação.

Existem dois lados dessa moeda, os céticos e os otimistas: os primeiros acreditam que não se deva pensar em computadores em escolas, pois muitas delas ainda não possuem o material básico e não possuem estruturas para o ensino adequado. Por outro lado, os otimistas acreditam que sim, o ensino deva acompanhar o avanço da informática na educação, e os computadores devam fazer parte do cotidiano dos estudantes.

Se ficarmos em dúvida em qual dos lados apoiar não chegaremos a lugar algum, pois o computador direta ou indiretamente já se encontra presente em várias escolas, e mais, presente também na vida cotidiana dos estudantes.

Existem dois tipos de ensino envolvendo os computadores, sendo eles, ensino de computação ou ensino pelo computador. No primeiro, se adquire o conceito sobre computador, no outro, o docente usa os recursos computacionais para os estudantes terem conceitos sobre o domínio da matéria (Valente, 1993).

Decidido escolher pelo segundo, a possibilidade de discutir o uso é fundamental, já que no meio educacional, muitas das vezes o computador é utilizado para reproduzir o ensino em sala de aula, por meio de software educacional (Richter, 2006).

Vale ressaltar que a informática educacional tem seu elo através das máquinas. Pressey, em 1924, foi o pioneiro, inovando com uma máquina que corrigia textos. Logo após, Skinner na década de 50 criou uma outra máquina que ensinava usando instrução programada. No

Brasil os softwares baseados em Instrução Programadas (CAI's), são conhecidos como Instrução Auxiliada por Computador (Valente, 1998).

2.1.2 Ensino Tradicional

Vieira (2006) destaca que o ensino tradicional é centrado no professor onde seu dever com os alunos é de vigiar, aconselhar, ensinar o conteúdo e corrigi-lo. O ensino tradicional tem como base a transmissão de conhecimento através de aulas, em geral exposta, numa sequência predeterminada e fixa, dando valor ao conteúdo teórico e sua qualidade. A figura 1, mostra com simplicidade o ensino tradicional.

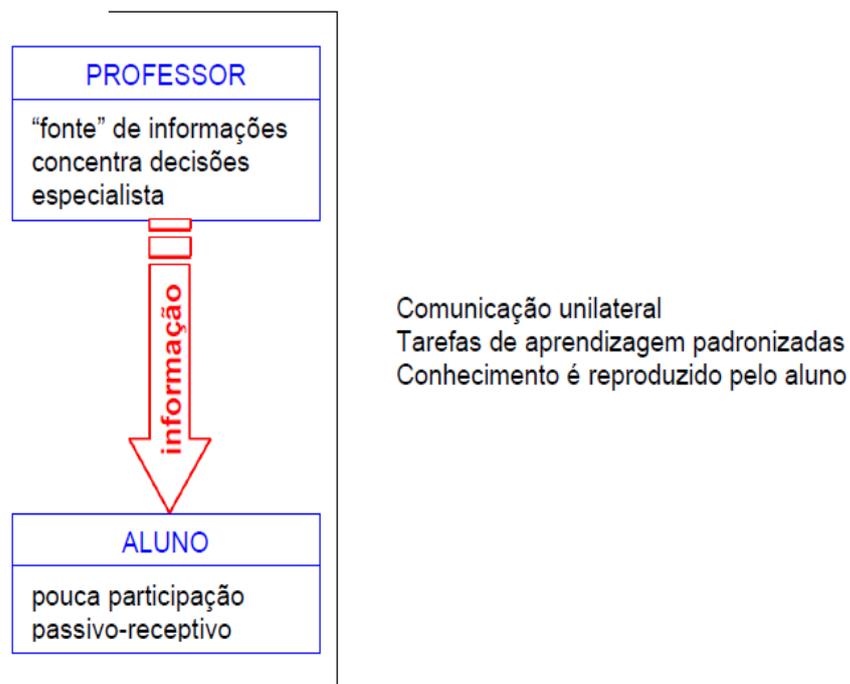


Figura 1 - Ensino Tradicional (extraído de Guerra, 2010).

A sequência de Aprendizagem é estática: o professor fala, o aluno ouve e aprende. Essa metodologia de ensino não faz do aluno elemento ativo na sua construção de conhecimento, geralmente desconsiderando o aprendizado fora da sala de aula, seus esforços e participação coletiva (Grzesiuk, 2008).

2.1.2.1 Vantagens e Desvantagens

A ligação de Professor e aluno funciona como uma forma de motivação e apoio, possibilitando a troca de ideias e conhecimentos. Porém o aluno tem pouco papel nesse ensino e menos responsabilidades, sempre se adequando as mudanças dos professores e baseando o aprendizado em memorização e imitação do professor (Vidal, 2002).

2.1.3 Ensino com Informática

O uso do computador na educação ocasionou uma revolução no conceito de ensino-aprendizagem, ensinando através de vários softwares educacionais em diferentes modos, isso comprova que o auxílio dessa ferramenta tecnológica pode ser útil no processo de aprendizagem (Grzesiuk, 2008).

Segundo Valente (2003), os softwares educacionais de antigamente caracterizavam-se por serem versões computadorizados dos métodos de ensino. No início a ideia era imitar as atividades que ocorriam em salas de aulas, mas com a dispersão do seu uso, outras modalidades foram desenvolvidas.

2.1.4 O uso de Softwares no Ambiente Educativo

Como Santana (2008) descreve, os softwares educacionais são de grande valia no processo de ensino-aprendizagem, os alunos se comunicam entre si e com as máquinas, facilita a troca de experiências, e ao surgirem dúvidas, buscar por soluções. Seu uso no ensino gera autoconfiança por parte dos alunos em resolver situações sozinho e resolver futuros problemas.

2.1.4.1 As Duas Abordagens de Ensino

Existem duas formas fundamentadas de ensino sendo elas:

+ Instrucionista – por este meio o ensino é reproduzido à imagem do ensino tradicional, o ensinamento é passado ao aluno de forma que o computador irá ensiná-lo, como ilustrado na Figura 2.

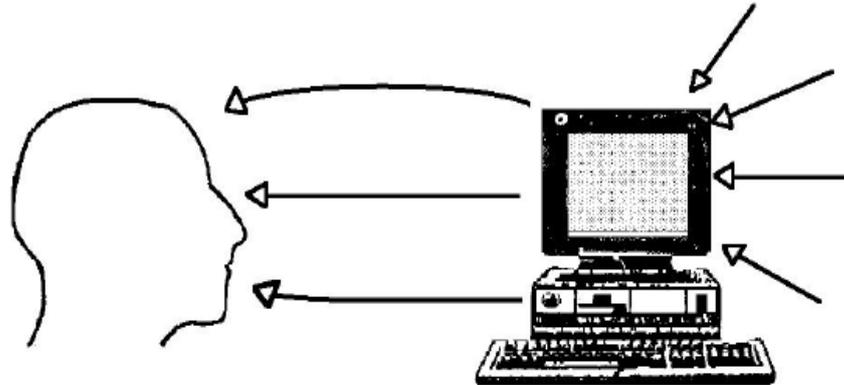


Figura 2 - Representação do paradigma instrucionista (extraído de Guerra, 2010).

+ Construcionista – por este meio o aluno interage com o software, cria soluções e toma decisões, conforme a Figura 3.

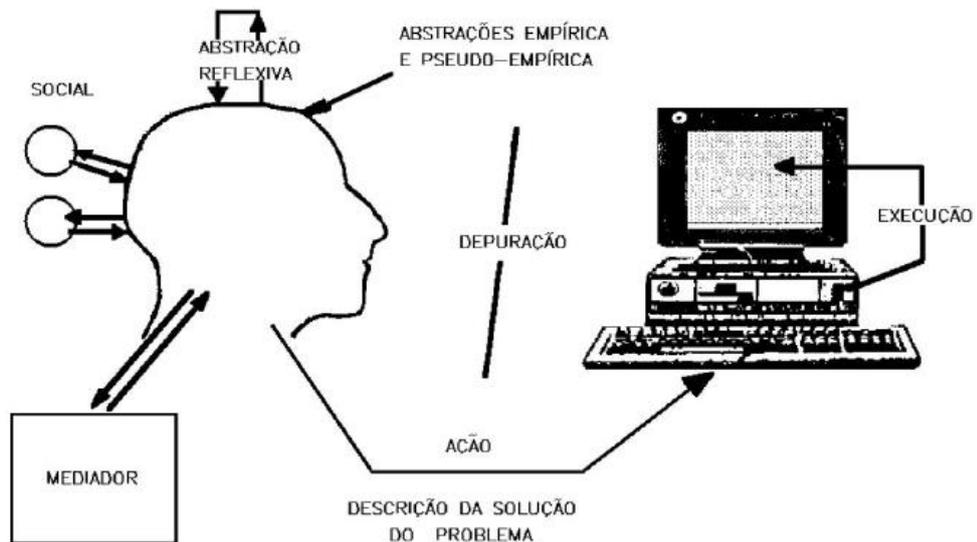


Figura 3 - Representação do paradigma construcionista (extraído de Guerra, 2010).

2.1.5 Tipos de Software Educacional

Os Softwares para educação são de tipos diferentes, onde cada um pode mostrar uma característica diferente do outro, auxiliando no processo de construção de conhecimento. Veremos três delas.

+ Tutoriais – software onde a informação organizada é produzida por uma sequência pedagógica particular e passada ao aluno. O estudante é quem determinará se segue ou não essa sequência. Os tutoriais possuem as seguintes características: pode ser um conteúdo eletrônico animado ou vídeo interativo, onde o computador conduz o ensino ao aluno, não permitindo que o aluno veja o processo, somente o final dele. Esse software não mostra se o aluno absorveu o conhecimento (Jucá, 2006).

+Exercícios e Prática – destaca-se em apresentar lições ou exercícios. O aluno tem a função de apenas passar de uma atividade para outra e o resultado é avaliado pelo software (Jucá, 2006).

+Jogos – desafiam e motivam os alunos, pois possuem um fim educacional, envolvendo-os em competição com a máquina ou amigos de classe (Jucá, 2006).

2.2 TRABALHOS CORRELATOS

Esta seção tem a objetividade de apresentar trabalhos anteriores que possuem alguma correlação ao presente trabalho.

2.2.1 Duolingo

Criado por Luis von Ahn em 2011, o Duolingo foi feito com propósito de ajudar outras pessoas na área da educação. Ahn percebeu que cerca de 1,2 bilhão de pessoas estavam aprendendo outra língua, foi então onde criou um site que futuramente seria também um aplicativo para smartphone que ensinava inglês, de forma com que os usuários aprendessem como em um jogo, onde, em cada nível é possível se aprofundar mais no idioma (Texeira, 2014).

2.2.2 Brainyoo

O Brainyoo, é um software educativo de fichas que auxilia seus usuários a aprender e armazenar conteúdos na memória de longo prazo. Devido ao seu modo variado de aprendizagem e seu sistema de fichamentos, nele é possível estudar questões de concursos abertos à simulados de Enem. Desenvolvido para proporcionar uma aprendizagem ativa e totalmente participativa. O Brainyoo possui 3 métodos de aprendizagem: modo de memória de longo prazo, onde o foco é guardar os conteúdos estudados de maneira duradoura; modo aleatório, onde o programa baseia a chance de repetição de cada fichamento em probabilidades, que podem ser definidas pelo usuário; e o modo de exame, muito útil antes de provas, quando o foco é apenas em um tema definido pelo usuário para uma revisão (Brainyoo, 2012).

3. TECNOLOGIAS UTILIZADAS

Neste capítulo serão apresentadas todas as tecnologias utilizadas para realizar o desenvolvimento do software.

3.1 JAVA

Java é uma linguagem que foi introduzida no mercado em 1995 pela *Sun Microsystems*, apresentado em grande entusiasmo aos programadores e analistas de software. Possui características importantes que a diferenciam de outras linguagens, tais como:

- **Orientação à objetos:** tudo com exceção dos seus dados primitivos são classes ou métodos o que se resume na principal função de orientação a objeto que é abstração, encapsulamento e herança. Além de outras características como, independência de plataforma, não possui ponteiros, performance, segurança, permite multithreading (Jandl, 1999).

3.2 ECLIPSE IDE

Iniciado seu projeto em 2001 pela IBM e suportado por fornecedores de softwares o eclipse IDE é um ambiente de desenvolvimento integrado baseado em Java, uma estrutura e um conjunto de serviços para o desenvolvimento de aplicações Desktop e Web, entende várias linguagens de programação e aceita instalações de pluinjs para simular o desenvolvimento da plataforma (Gallardo, 2012).

3.3 APACHE TOMCAT

Um software que foi desenvolvido pela Fundação Apache que permite a execução de aplicações web, juntamente com a linguagem de programação Java, mas especificamente nas tecnologias Servlets e JSP (Porta, 2013).

3.4 JSF E PRIMEFACES

É um framework web baseado em componentes pré-definidos. O JSF permite criar aplicações Java Web de uma maneira favorável aos programadores que não precisam se preocupar com Javascript nem HTML, para facilitar ainda mais a vida do programador existem bibliotecas como Primefaces que possuem muitas funcionalidades prontas e componentes sofisticados (Oliveira, 2013).

3.5 POSTGRESQL

Um SGBD objeto-relacional de código aberto, utilizado como forma de armazenamento de informações, com vários recursos como, consultas complexas, chaves estrangeiras, integridade transacional, controle de concorrência multi-versão, suporte ao modelo híbrido objeto-relacional, triggers, visões, linguagem procedural em várias linguagens para procedimentos armazenados, indexação por texto, estrutura para guardar dados georeferenciados PostGIS (Milani, 2008).

4. PROPOSTA DO TRABALHO

Neste capítulo será apresentado o projeto do trabalho, destacando as características que o software-educativo Dica tem a oferecer, detalhando seu uso como usuário professor e como usuário aluno.

4.1 PROJETO DICA

A principal ideia do projeto é a criação de um software-educativo, onde o professor produzirá um método de avaliação diferente aos alunos. Esta avaliação consiste em ter 10 perguntas, para cada pergunta deve-se conter a categoria, inserir 4 alternativas e fornecer uma dica com base na pergunta.

Além desses requisitos a avaliação deverá possuir 2 benefícios aos alunos, sendo eles:

- Dica – revelará ao aluno a dica da pergunta.
- Eliminador – eliminará uma alternativa incorreta.

A quantidade desses benefícios é definida pelo professor no momento da criação da avaliação.

4.1.1 Correção da Avaliação

O método de correção para gerar a nota e a pontuação da avaliação é feita da seguinte forma: cada pergunta respondida corretamente soma-se 1 de nota e 100 pontos, porém, ao usar algum benefício será descontada nota e acrescentado pontos, caso o usuário utilize a “dica” desconta-se 0,25 de nota e acrescenta-se 50 pontos, caso utilize o “eliminador” desconta-se 0,50 de nota e acrescenta-se 100 pontos. Ao final das 10 perguntas respondidas as somas da nota e pontuação do usuário são gerados. A Figura 04 mostra um exemplo de correção da avaliação.

Exemplo de Correção da Avaliação

Requisitos da Avaliação		Valores			
10 Perguntas		Pergunta Correta = 1 Nota e 100 Pontos			
3 Dica		Dica = -0,25 Nota e 50 Pontos			
2 Eliminador		Eliminador = -0,50 Nota e 100 Pontos			
Resultados dos Alunos					
Aluno 1		Aluno 2		Aluno 3	
8 Respostas Corretas		10 Respostas Corretas		9 Respostas Corretas	
2 Uso de Dica		3 Uso de Dica		0 Uso de Dica	
1 Uso de Eliminador		2 Uso de Eliminador		0 Uso de Eliminador	
Nota e Pontuação Final					
Aluno 1		Aluno 2		Aluno 3	
8 x R.C. = 8 Nota e 800 Pontos		10 x R.C. = 10 Nota e 1000 Pontos		9 x R.C. = 9 Nota e 900 Pontos	
2 x Dica = -0,50 Nota e 100 Pontos		3 x Dica = -0,75 Nota e 150 Pontos		0 x Dica = 0 Nota e 0 Pontos	
1 x Eliminador = -0,50 e 100 Pontos		2 x Eliminador = -1 e 200 Pontos		0 x Eliminador = 0 e 0 Pontos	
Nota Final = 7 Nota e 1000 Pontos		Nota Final = 8,25 Nota e 1350 Pontos		Nota Final = 9 Nota e 900 Pontos	

Figura 4 - Exemplo de correção da avaliação.

4.1.2 Ranking dos Alunos

O DICA gera um ranking geral dos alunos, classificado pela pontuação alcançada na realização da prova.

Além da nota conquistada na prova, o aluno pode ganhar um benefício extra oferecido pelo professor: os 3 primeiros colocados no ranking ganharão 2 pontos na média final de um bimestre, incentivando assim os alunos a sempre buscar as primeiras posições. A Figura 05 exemplifica como seria o ranking com base no exemplo da Figura 04.

Exemplo de Ranking

Classificação				
Posição	Nome	Avaliação	Pontuação	Nota
1	Aluno 2	Prova 1	1350	8,25
2	Aluno 1	Prova 1	1000	7,00
3	Aluno 3	Prova 1	900	9,00

Figura 5 - Exemplo de ranking.

4.2 DIAGRAMA DE CLASSE E DIAGRAMA DE CASO DE USO

Tendo em vista a concepção do projeto elaborou-se o diagrama de classes e diagrama de casos de uso do software DICA. A Figura 6 ilustra o Diagrama de Classes e a Figura 7 o diagrama de Casos de Uso.

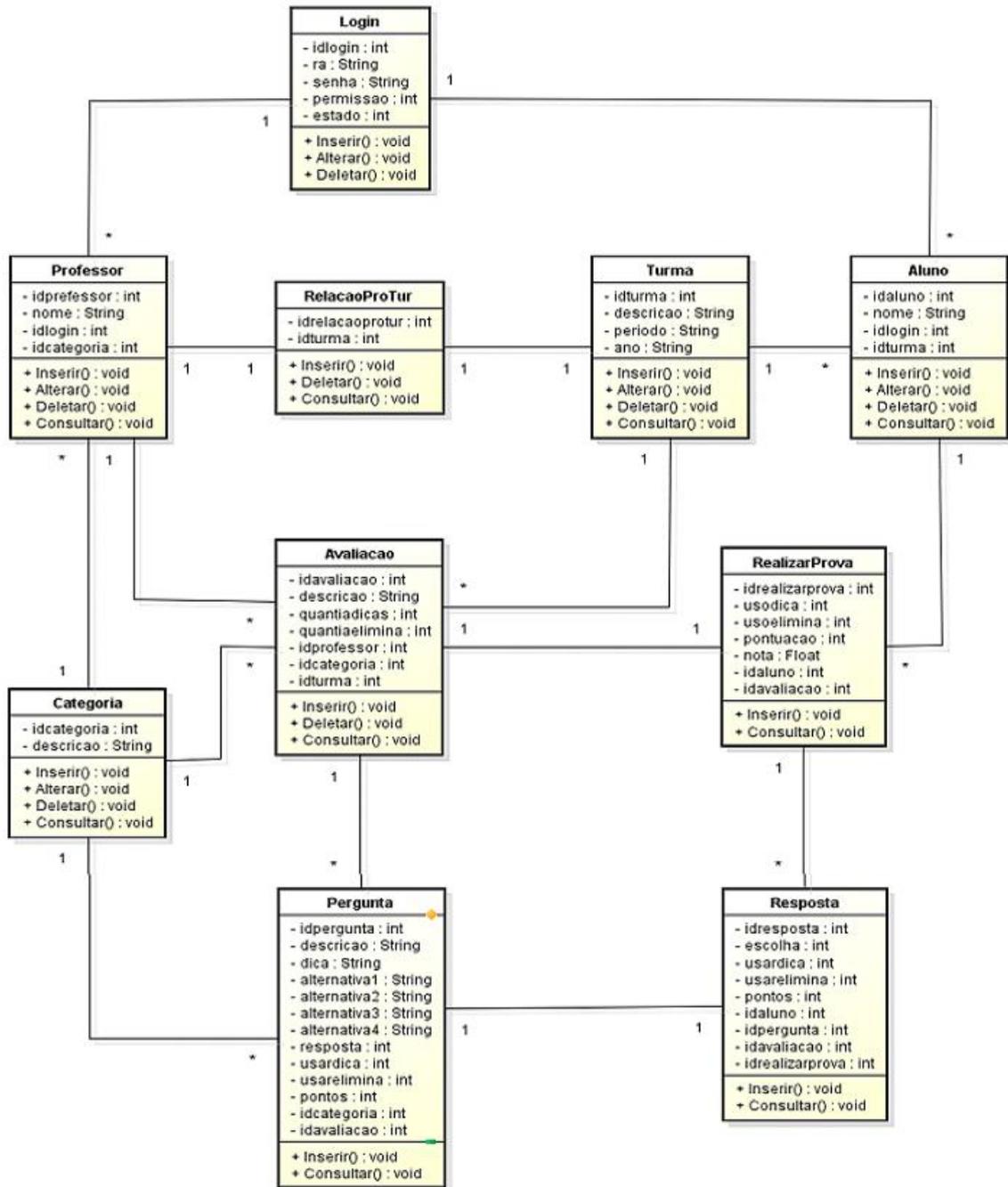


Figura 6 - Diagrama de Classe.

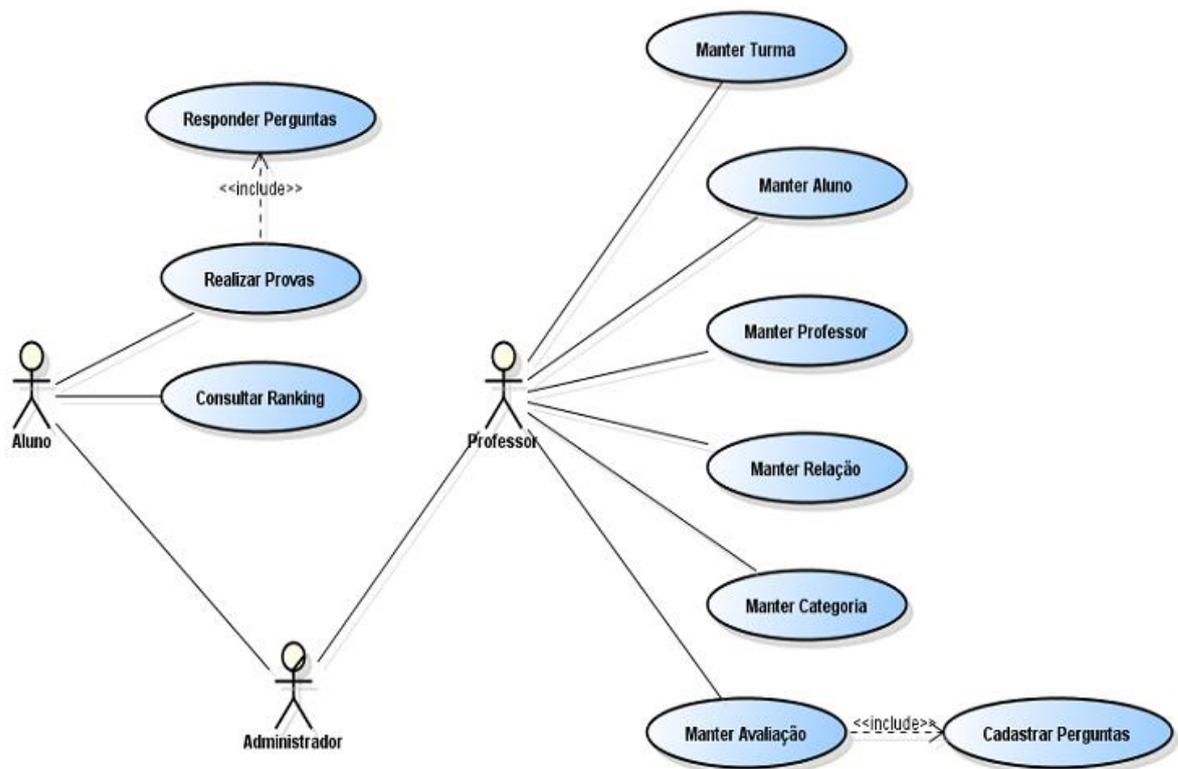


Figura 7 - Diagrama de Caso de Uso.

4.3 ACESSO AO SISTEMA

O acesso ao sistema é feito após o login, onde o usuário informa seu RA e sua senha e é verificado qual tipo de usuário está acessando, pois existe o usuário professor e usuário aluno, cada qual com sua parte acessível no sistema. O usuário professor será responsável por fazer os cadastros de alunos, turmas, categorias e de outros professores, além do principal que é a criação da avaliação para os alunos. O usuário aluno irá apenas realizar a avaliação gerada pelos professores no sistema, fazer consultas e verificar a classificação no ranking dos alunos. A Figura 08 mostra a tela de login do sistema Dica.

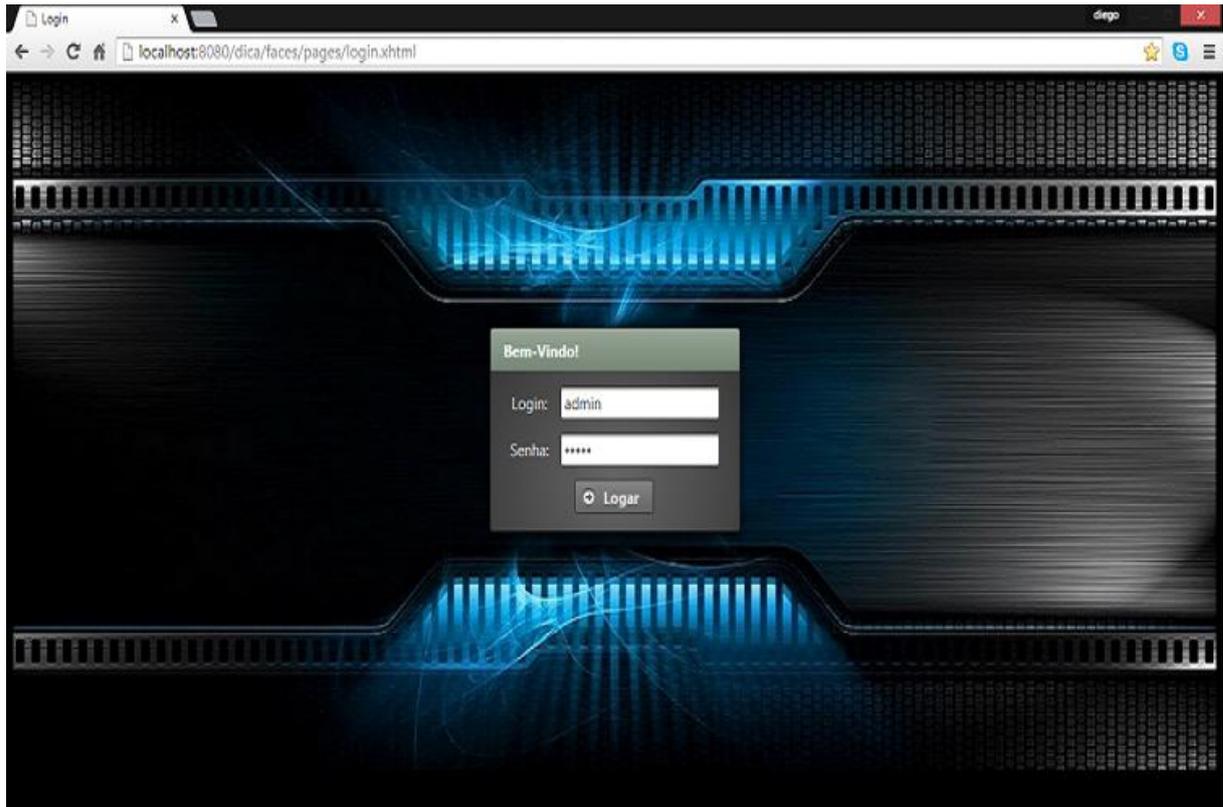


Figura 8 - Tela de acesso ao sistema DICA.

4.4 USUÁRIO PROFESSOR

O professor possui acesso a maior parte do sistema, o que inclui realizar todos os cadastros, fazer consultas e o principal, fazer a criação da avaliação.

4.4.1 Cadastros

O responsável pela parte de gerar cadastros é o professor, pois o sistema permite que apenas ele possua acesso a esta parte. Dentre os cadastros do sistema, destacam-se: cadastros de outros professores, alunos, turmas, categorias e professores em turmas. A Figura 09 mostra a parte onde se encontra no sistema os cadastros para serem acessados.

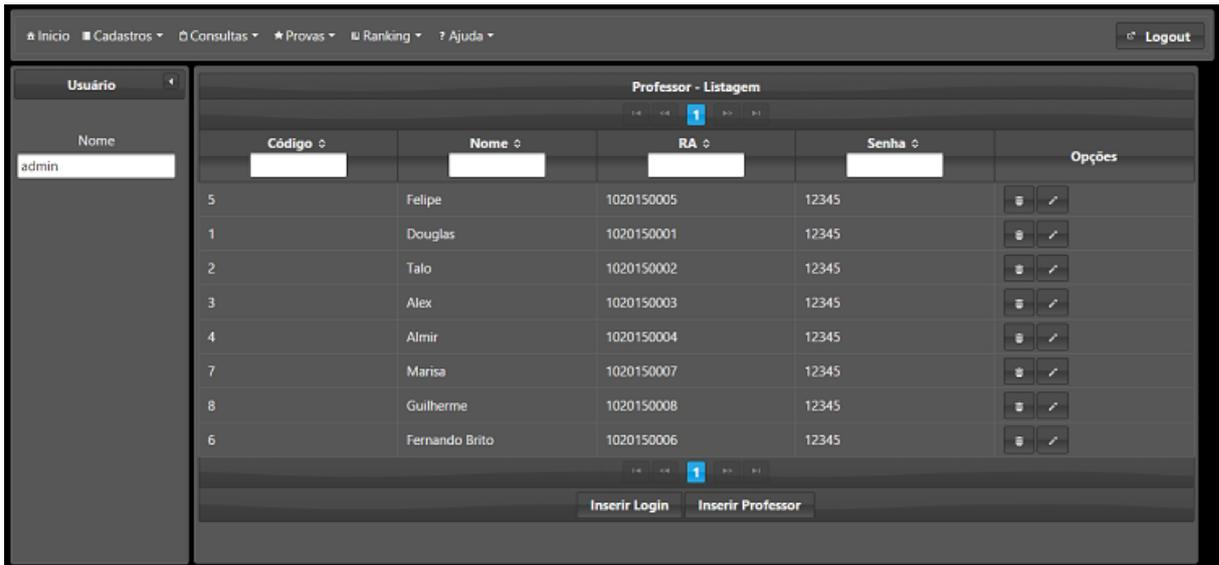


Figura 9 - Tela de Cadastros.

4.4.2 Consultas

As consultas podem ser acessadas tanto pelo professor como pelo aluno. Elas são referentes aos seus cadastros, e a única diferença é que não se podem inserir novos cadastros, havendo então as consultas de professores, alunos, turmas, categorias e professores em turma. A Figura 10 mostra a parte de consultas no sistema.

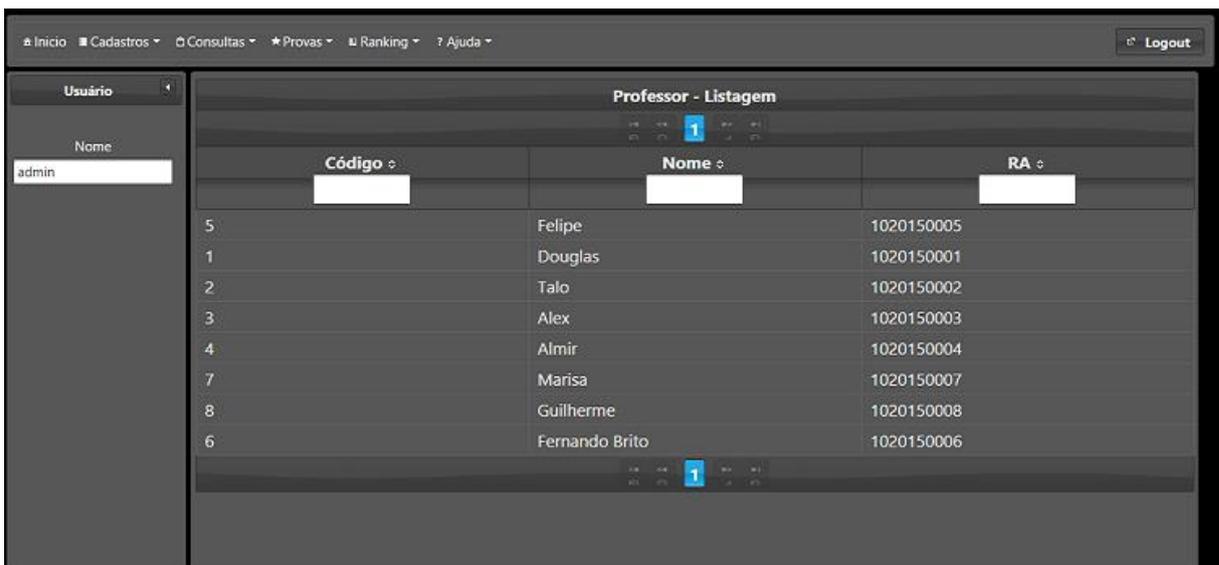


Figura 10 - Consultas.

4.4.3 Elaborando a Avaliação

A principal função do professor no sistema é a elaboração da avaliação para os alunos, ela é realizada da seguinte maneira: o professor possui um local próprio no sistema para elaborar uma avaliação. A Figura 11 mostra a tela inicial da criação de uma avaliação.

A imagem mostra a interface de usuário para a criação de uma avaliação. No topo, há uma barra de navegação com links para Início, Cadastros, Consultas, Provas, Ranking e Ajuda, além de um botão Logout. À esquerda, há um menu lateral com o título 'Usuário' e um campo 'Nome' com o valor 'admin'. O formulário principal contém os seguintes campos:

- Descrição: Avaliação Geral
- Professor: Felipe
- Categoria: Geral
- Turma: 3º Ensino Médio A
- Quantia de Dicas: 3
- Quantia de Eliminador: 2

Um botão 'Continuar' está localizado abaixo dos campos de entrada.

Figura 11 - Criação de avaliação.

Após a definição dos requisitos necessários, o professor deve clicar na opção continuar para ir ao próximo passo da criação da avaliação, que é a elaboração das perguntas e respostas. Nesta fase é importante que o professor escreva a pergunta e suas alternativas nas partes referentes a elas, além de ser obrigado a fornecer a resposta dessa pergunta e uma dica para a mesma. Serão necessárias 10 perguntas para se concluir a avaliação. A Figura 12 mostra a tela onde se elabora uma pergunta da avaliação.

Figura 12 - Tela para elaboração de perguntas.

Após concluir todas as 10 perguntas o professor deve finalizar a avaliação para que ela possa ser acessada pelos alunos.

4.5 USUÁRIO ALUNO

Assim como o professor o aluno possui acesso limitado ao sistema, podendo também acessar as consultas, realizar as avaliações e verificar o ranking dos alunos.

4.5.1 Realizando a Avaliação

Ao acessar o sistema o aluno deve verificar se algum professor disponibilizou avaliações para serem realizadas.

Ao iniciar uma avaliação o aluno é direcionado para a tela de pergunta para respondê-las, a tela é a mesma que a de pergunta, apenas com detalhes de resposta, onde o aluno encontra a opção de uso de dica e uso de eliminação. A Figura 13 mostra a tela de resposta da avaliação.

Usuário

Nome
admin

Pergunta 1

Realizar Prova Realizar Prova Avaliação Avaliação

Aluno Aluno Pergunta Pergunta

Descrição
Qual a Formula da Água?

Dica

1 H3O

2 H2O3

3 H2O

4 H5O1

Usar Dica Dica

Usar Elimina Eliminador

Escolha 1

Errada

Continuar

Figura 13 - Respostas.

Após o aluno responder as 10 perguntas e concluir a avaliação uma tela com os resultados de nota e pontuação da avaliação é exibida para ele. A Figura 14 representa a tela de resultado final.

Usuário

Nome
admin

Resultado Final

Respostas Corretas 10 / 10

Uso de Dica 3 / 3

Uso de Eliminador 2 / 2

Nota 8.25

Pontuação 1350

Parabéns!

Figura 14 - Resultado final.

4.5.2 Consultando o Ranking dos Alunos

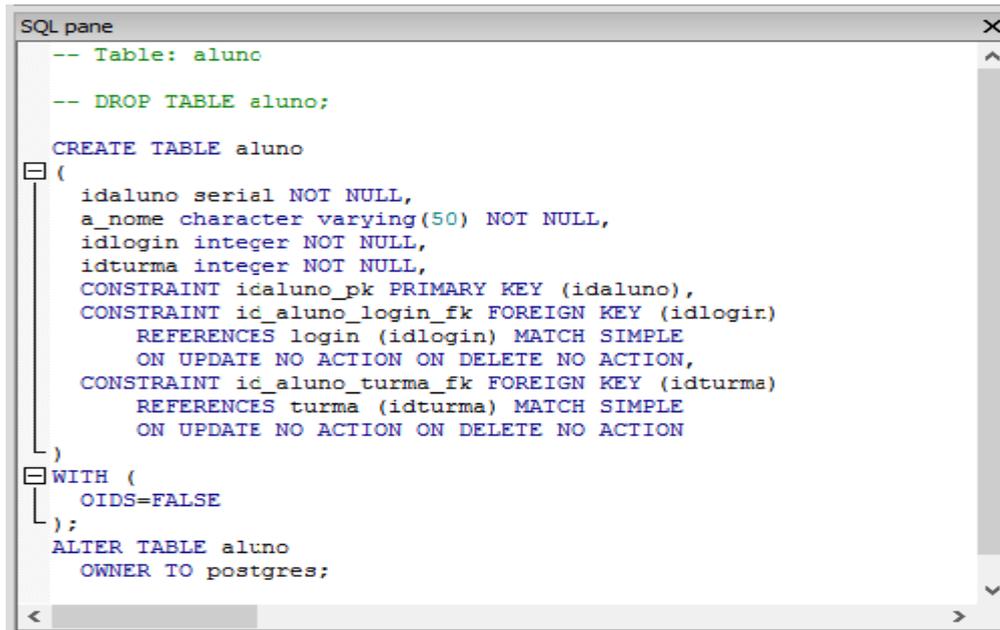
Ao acessar o sistema o aluno poderá também consultar o ranking dos demais alunos. A classificação é definida pela maior pontuação alcançada na avaliação e determina os 3 primeiros alunos que receberão o benefício proposto pelo professor. A Figura 15 mostra a tela de consulta de ranking dos alunos.

Nome	Avaliação	Uso de Dica	Uso de Eliminador	Nota	Pontuação
Gabriel	Avaliacao Geral	3	2	8.25	1350
Alexandre	Avaliacao Geral	2	2	7.5	1200
Diego	Avaliacao Geral	2	1	7.0	1000
Roger	Avaliacao Geral	0	0	9.0	900

Figura 15 - Consulta de ranking dos alunos.

4.6. IMPLEMENTAÇÃO DO DICA

Nesta seção será apresentada como foi a construção deste projeto detalhando o desenvolvimento da aplicação Web.



```
SQL pane
-- Table: aluno
-- DROP TABLE aluno;

CREATE TABLE aluno
(
    idaluno serial NOT NULL,
    a_nome character varying(50) NOT NULL,
    idlogin integer NOT NULL,
    idturma integer NOT NULL,
    CONSTRAINT idaluno_pk PRIMARY KEY (idaluno),
    CONSTRAINT id_aluno_login_fk FOREIGN KEY (idlogin)
    REFERENCES login (idlogin) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION,
    CONSTRAINT id_aluno_turma_fk FOREIGN KEY (idturma)
    REFERENCES turma (idturma) MATCH SIMPLE
    ON UPDATE NO ACTION ON DELETE NO ACTION
);
WITH (
    OIDS=FALSE
);
ALTER TABLE aluno
    OWNER TO postgres;
```

Figura 17 - Tabela aluno do Banco de Dados.

4.6.2 Desenvolvimento Web

Para desenvolver esse ambiente Web foi utilizado a linguagem de programação Java na IDE Java EE junto do servidor Apache Tomcat, com o banco de dados PostgreSQL, além de outras tecnologias. A Figura 18 apresenta o projeto e os pacotes da parte Web da aplicação.

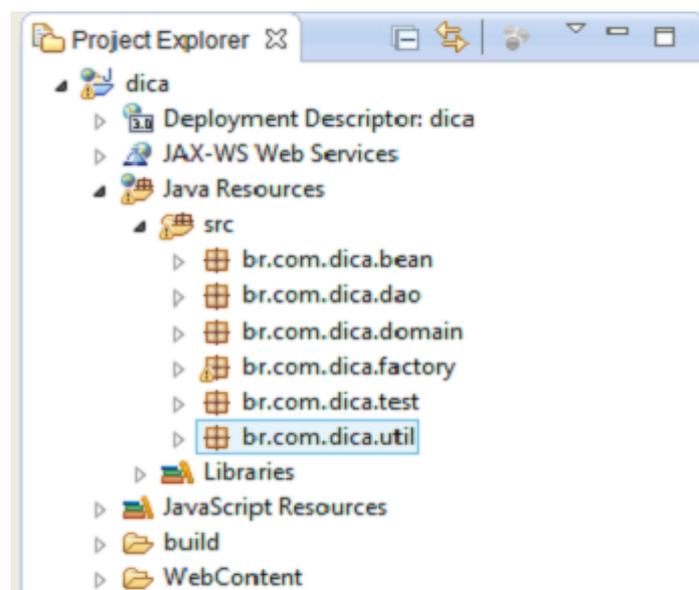


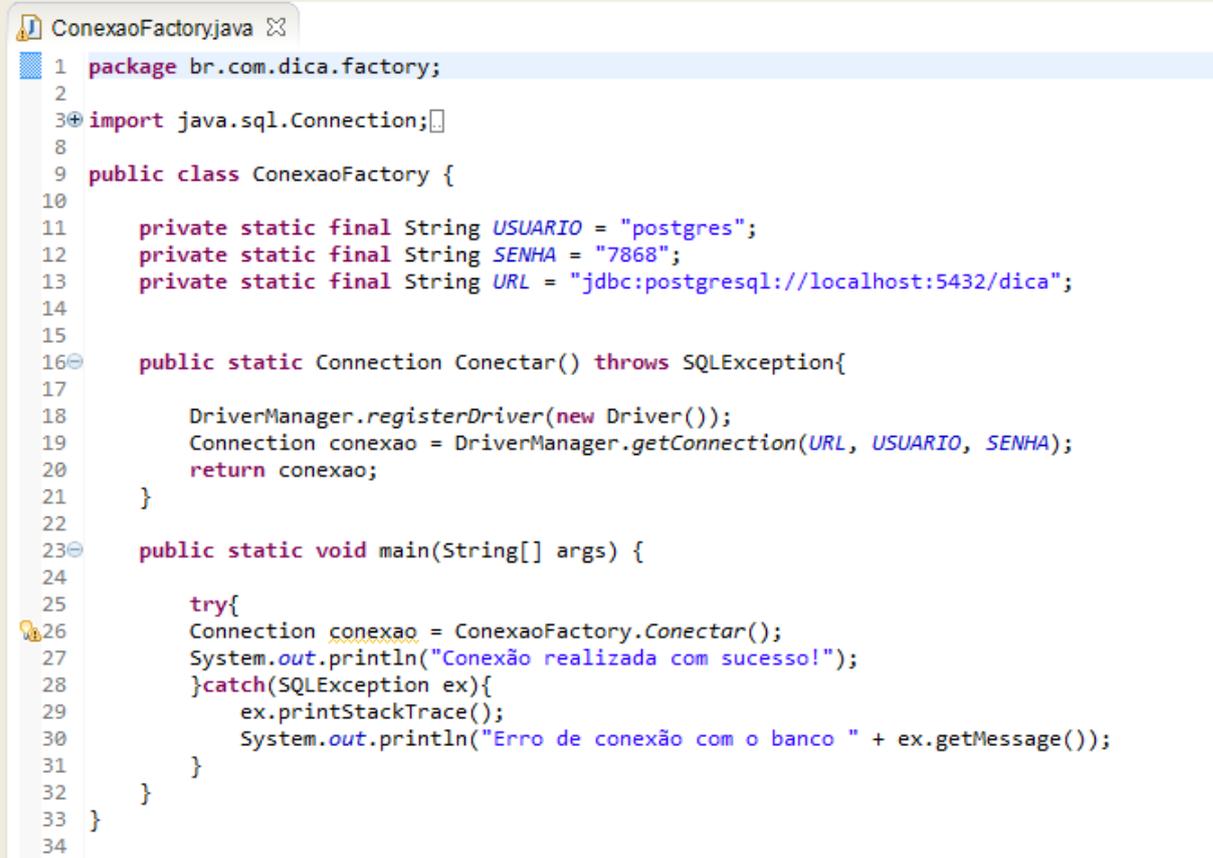
Figura 18 - Projeto DICA.

Como ilustrado na Figura 18, o projeto encontra-se dividido em seis pacotes para melhor organização no seu desenvolvimento.

As seções seguintes descrevem o conteúdo de cada um dos pacotes da aplicação.

4.6.2.1 Pacote Factory

Este pacote possui apenas uma classe, responsável pela conexão com o banco de dados PostgreSQL. A Figura 19 exhibe o código utilizado para fazer essa conexão.



```
ConexaoFactory.java
1 package br.com.dica.factory;
2
3 import java.sql.Connection;
4
5
6
7
8
9 public class ConexaoFactory {
10
11     private static final String USUARIO = "postgres";
12     private static final String SENHA = "7868";
13     private static final String URL = "jdbc:postgresql://localhost:5432/dica";
14
15
16     public static Connection Conectar() throws SQLException{
17
18         DriverManager.registerDriver(new Driver());
19         Connection conexao = DriverManager.getConnection(URL, USUARIO, SENHA);
20         return conexao;
21     }
22
23     public static void main(String[] args) {
24
25         try{
26             Connection conexao = ConexaoFactory.Conectar();
27             System.out.println("Conexão realizada com sucesso!");
28         }catch(SQLException ex){
29             ex.printStackTrace();
30             System.out.println("Erro de conexão com o banco " + ex.getMessage());
31         }
32     }
33 }
34
```

Figura 19 - Conexão com Banco de Dados.

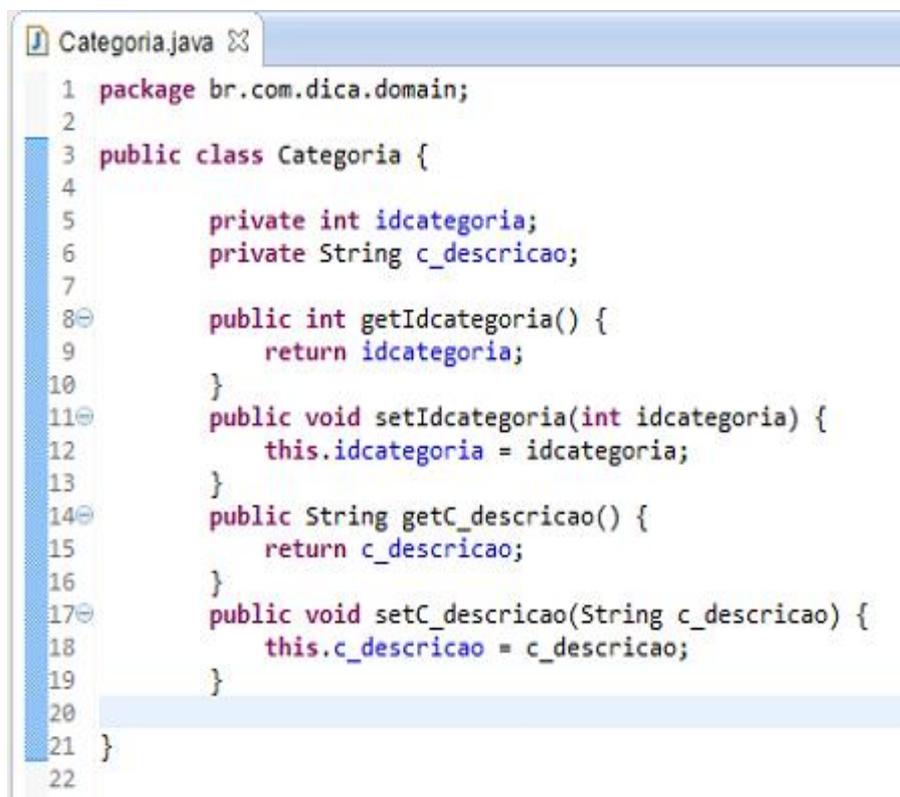
A Figura 19 apresenta a classe “*ConexaoFactory*” onde se cria três variáveis *String*, sendo informado o usuário do banco, senha e url, em seguida o método “*Conectar*” chama uma nova conexão recebendo o usuário, senha e url e retornando uma conexão aberta.

O método “*main*” desta classe foi implementado apenas para testar a conexão, fica dentro de um *try catch*, onde, caso a conexão for executada com sucesso é exibida na tela de

console do Java a mensagem “Conexão realizada com sucesso!”, porém, caso ocorra uma exceção é exibida na tela a mensagem “Erro de conexão com o banco” junto com linhas de códigos de erro com a conexão.

4.6.2.2 Pacote Domain

Neste pacote encontram-se as classes responsáveis por conter as variáveis, construtores e os *Getters* e *Setters* referente a cada tabela do banco de dados. A Figura 20 mostra uma parte do código de uma classe do pacote Domain.



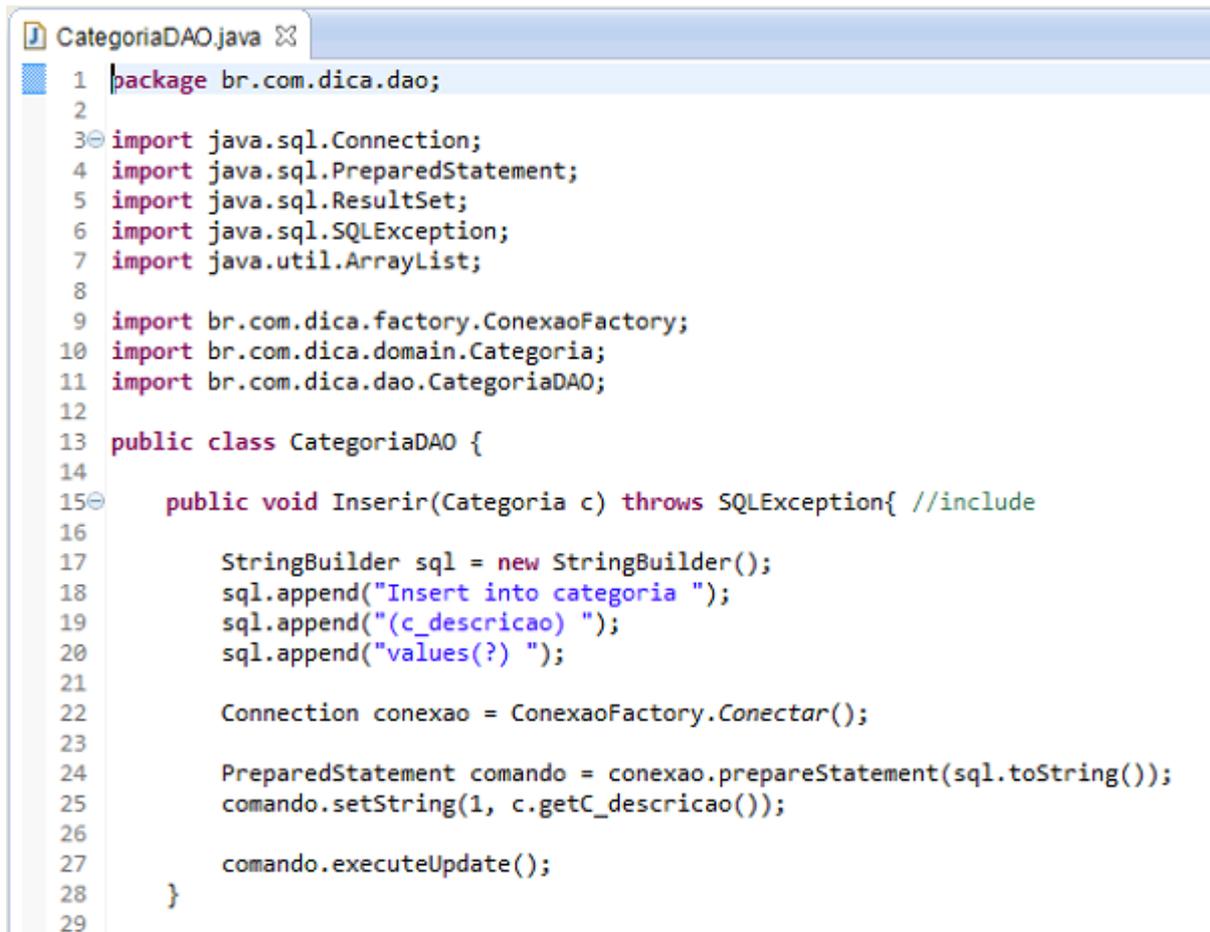
```
1 package br.com.dica.domain;
2
3 public class Categoria {
4
5     private int idcategoria;
6     private String c_descricao;
7
8     public int getIdcategoria() {
9         return idcategoria;
10    }
11    public void setIdcategoria(int idcategoria) {
12        this.idcategoria = idcategoria;
13    }
14    public String getC_descricao() {
15        return c_descricao;
16    }
17    public void setC_descricao(String c_descricao) {
18        this.c_descricao = c_descricao;
19    }
20
21 }
22
```

Figura 20 - Classe do pacote Domain.

A Figura 20 mostra a classe “*Categoria*” da aplicação no pacote Domain, onde se ve as variáveis e logo em seguida seus construtores *Getters* e *Setters* que posteriormente serão utilizados nas classes do pacote DAO. Com o intuito de melhor organizar os códigos, os nomes das variáveis são os mesmos nomes das tabelas do banco de dados.

4.6.2.3 Pacote DAO

No pacote DAO estão as classes que são responsáveis por toda transação que acontece com o banco de dados, como criação de conexão com o banco, os métodos *insert*, *delete*, *update* e *list* e encerramento da conexão com o banco de dados, além de outras. A Figura 21 exibe uma parte do código de uma classe da aplicação no pacote DAO.



```
1 package br.com.dica.dao;
2
3 import java.sql.Connection;
4 import java.sql.PreparedStatement;
5 import java.sql.ResultSet;
6 import java.sql.SQLException;
7 import java.util.ArrayList;
8
9 import br.com.dica.factory.ConexaoFactory;
10 import br.com.dica.domain.Categoria;
11 import br.com.dica.dao.CategoriaDAO;
12
13 public class CategoriaDAO {
14
15     public void Inserir(Categoria c) throws SQLException{ //include
16
17         StringBuilder sql = new StringBuilder();
18         sql.append("Insert into categoria ");
19         sql.append("(c_descricao) ");
20         sql.append("values(?) ");
21
22         Connection conexao = ConexaoFactory.Conectar();
23
24         PreparedStatement comando = conexao.prepareStatement(sql.toString());
25         comando.setString(1, c.getC_descricao());
26
27         comando.executeUpdate();
28     }
29 }
```

Figura 21 - Classe do pacote DAO.

A Figura 21 apresenta o método *Inserir* que representa o *insert* em um banco de dados. Esse método recebe as variáveis da classe do pacote Domain e cria uma *StringBuild* fazendo uma nova sql para retornar ao banco de dados, em seguida inicia-se uma conexão e prepara uma parametrização onde essa sql recebe as variáveis como *string*, por fim o *executeUpdate* executa a inserção na tabela do banco.

4.6.2.4 Pacote Bean

As classes desse pacote são classes comuns de Java e contêm as regras de negócios, mapeadas através de anotações ou *Java Annotations*. Geralmente são classes Java Bean registrada com JSF gerenciados pelo *framework* JSF, com isso o Managed Bean pode ser acessado em uma página JSF. A Figura 22 apresenta uma parte do código da classe Categoria no pacote Bean e os métodos *prepararNovo()* e *Novo()* e suas anotações.

```

CategoriaBean.java
1 package br.com.dica.bean;
2
3 import java.sql.SQLException;

@ManagedBean(name = "MBCategoria")
@ViewScoped
public class CategoriaBean {

    private Categoria categoria;
    private ArrayList<Categoria> itens;
    private ArrayList<Categoria> itensFiltrados;

    public Categoria getCategoria() {
        return categoria;
    }

    public void setCategoria(Categoria categoria) {
        this.categoria = categoria;
    }

    //inserir
    public void prepararNovo(){

        categoria = new Categoria();
    }

    public void novo(){

        try{
            CategoriaDAO cdao = new CategoriaDAO();
            cdao.Inserir(categoria);

            itens = cdao.Listar();
            JSFUtil.adicionarMensagemSucesso("Categoria salvo com sucesso!");
        }catch(SQLException ex){
            ex.printStackTrace();
            JSFUtil.adicionarMensagemErro(ex.getMessage());
        }
    }
}

```

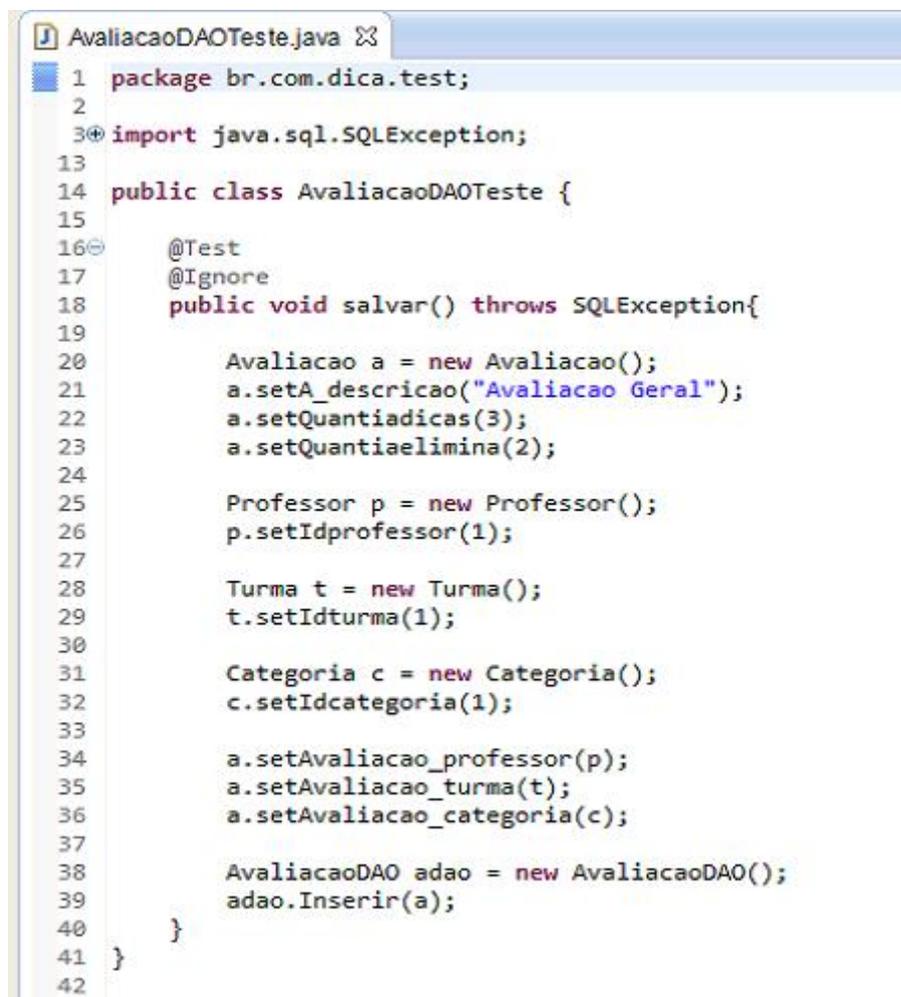
Figura 22 - Classe do pacote Bean.

A Figura 22 apresenta código no qual existe uma anotação “@ManagedBean(name = “MBCategoria”)”, para representar o nome do atributo. Caso não seja especificado um nome

para esse atributo, por padrão, será utilizado o nome da classe, e a anotação “@ViewScoped” é um bean ativo enquanto o usuário interage com a página JSF na Web.

4.6.2.5 Pacote Test

Neste pacote encontram-se as classes de teste da aplicação, responsável por testar as classes DAO. Tais testes são feitos a partir do *framework* JUnit, muito útil quando ainda não se tem as páginas Web elaboradas, sendo então possível testar as classes e os métodos para prosseguir na aplicação enquanto as telas ainda não foram criadas. A Figura 23 apresenta uma classe teste.



```
AvaliacaoDAOTeste.java
1 package br.com.dica.test;
2
3 import java.sql.SQLException;
13
14 public class AvaliacaoDAOTeste {
15
16     @Test
17     @Ignore
18     public void salvar() throws SQLException{
19
20         Avaliacao a = new Avaliacao();
21         a.setA_descricao("Avaliacao Geral");
22         a.setQuantiadicadas(3);
23         a.setQuantiaelimina(2);
24
25         Professor p = new Professor();
26         p.setIdprofessor(1);
27
28         Turma t = new Turma();
29         t.setIdturma(1);
30
31         Categoria c = new Categoria();
32         c.setIdcategoria(1);
33
34         a.setAvaliacao_professor(p);
35         a.setAvaliacao_turma(t);
36         a.setAvaliacao_categoria(c);
37
38         AvaliacaoDAO adao = new AvaliacaoDAO();
39         adao.Inserir(a);
40     }
41 }
42
```

Figura 23 - Classe do pacote Test.

Como ilustrada na Figura 23 esta classe teste possui uma anotação “@Test”, responsável por testar os métodos da classe. Observa-se que também existe a anotação “@Ignore” para ignorar o teste de método. Na tela do JUnit é mostrada a quantidade de métodos testados representado por “runs” e também quantos erros e falhas possuem caso ocorram. Após a execução do teste com JUnit é exibido uma barra com cor verde ou vermelho, onde respectivamente representam teste funcional ou não-funcional, a Figura 24 mostra o resultado do teste da Figura 23.

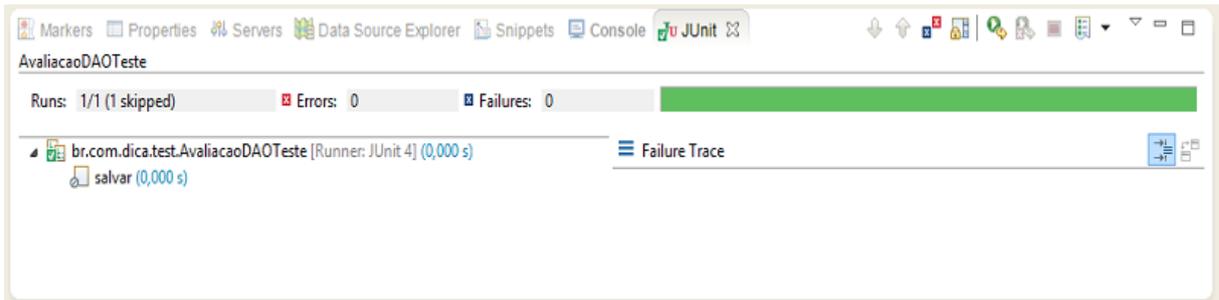


Figura 24 - Resultado do teste JUnit.

4.6.2.6 Pacote Útil

Esse pacote possui apenas uma classe “JSFUtil” que serve para utilitários JSF em páginas Web. Possui 2 métodos de mensagem, uma caso ocorra erro e outra de sucesso na página. A Figura 25 mostra essa classe.

```

1 package br.com.dica.util;
2
3 import javax.faces.application.FacesMessage;
4 import javax.faces.context.FacesContext;
5
6 public class JSFUtil {
7
8     public static void adicionarMensagemSucesso(String mensagem){
9
10         FacesMessage msg = new FacesMessage(FacesMessage.SEVERITY_INFO, mensagem, mensagem);
11         FacesContext contexto = FacesContext.getCurrentInstance();
12         contexto.addMessage(null, msg);
13     }
14
15     public static void adicionarMensagemErro(String mensagem){
16
17         FacesMessage msg = new FacesMessage(FacesMessage.SEVERITY_ERROR, mensagem, mensagem);
18         FacesContext contexto = FacesContext.getCurrentInstance();
19         contexto.addMessage(null, msg);
20     }
21 }

```

Figura 25 - Classe JSFUtil.

4.6.2.7 Páginas Web

A interface de interação com o usuário ocorre através de páginas Web, como descrito anteriormente no projeto. As páginas Web, implementadas em JSF, possuem uma extensão “.xhtml”, onde ocorre a junção de XML e HTML. Estas páginas encontram-se armazenadas na pasta “WebContent/pages”, como na Figura 26.

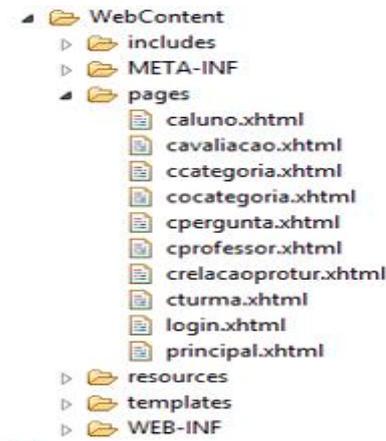


Figura 26 - Páginas Web da aplicação.

Na Figura 26 mostra vários arquivos “.xhtml”, cada uma sendo uma página da aplicação Web, a seguir é apresentada a Figura 27 que ilustra uma parte do código da página “login.xhtml”.

```

login.xhtml
8
9
10 <h:head>
11   <title>Login</title>
12   <h:outputStylesheet library="css" name="estilos.css"/>
13   <h:outputScript library="js" name="scripts.js" />
14   <h:graphicImage library="imagens" name="Login.png" width="100%" height="100%" />
15 </h:head>
16
17 <h:body>
18   <h:form>
19     <p:dialog header="Bem-Vindo!" visible="true"
20       closable="false" resizable="false" draggable="false"
21       styleClass="usuario-estilo" widgetVar="dlg">
22       <h:panelGrid columns="2" cellpadding="5">
23         <h:outputLabel for="login" value="Login:" />
24         <p:inputText id="login" value="#{MBLogin.acao.ra}"
25           size="20" maxLength="10"
26           required="true" requiredMessage="Digite o Login!" />
27
28         <h:outputLabel for="senha" value="Senha:" />
29         <p:password id="senha" value="#{MBLogin.acao.senha}"
30           size="20" maxLength="8"
31           required="true" requiredMessage="Digite o Login!" />
32       </h:panelGrid>
33     </p:dialog>
34   </h:form>
35 </h:body>
36 </html>

```

Figura 27 - Página XHTML do Login.

Observando os códigos da Figura 27 podemos notar as *tags* que formam a interface gráfica na página Web apresentada na Figura 28. Tal página é a de login, onde o usuário informa nos campos de digitação seu login e senha, e ao clicar em logar é feita a validação para verificar a existência do usuário no banco de dados para que possa ter acesso ao sistema.

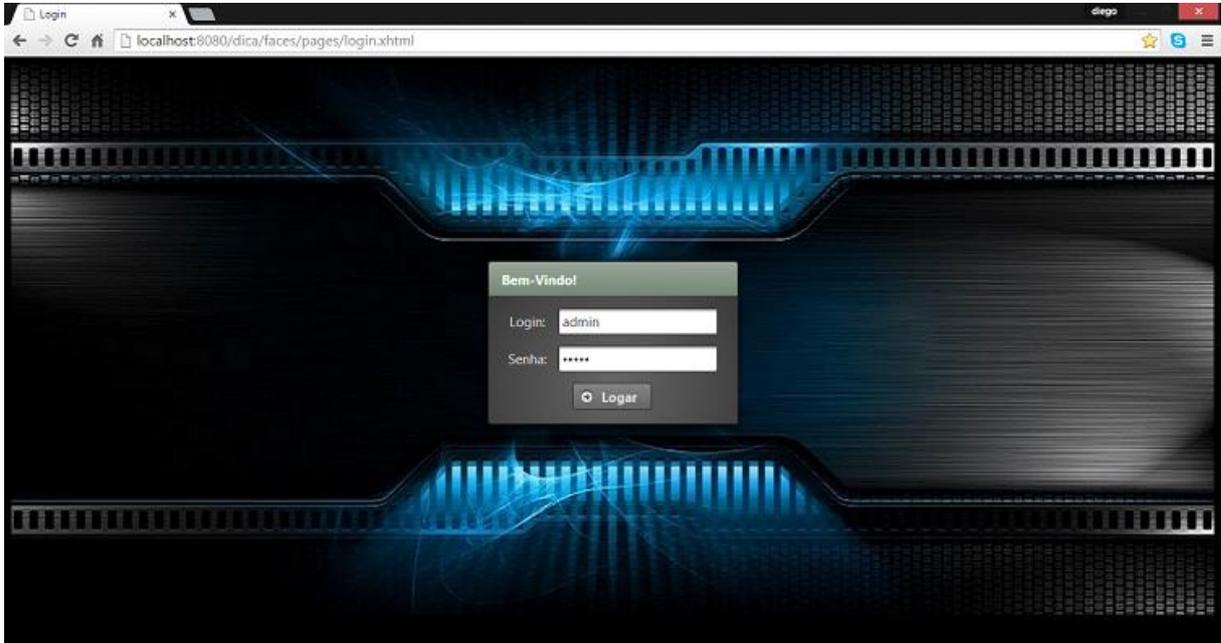


Figura 28 - Interface gráfica da tela Login.

4.6.2.7.1 Template

Para facilitar ainda mais as páginas foi utilizado um *template* como modelo para as páginas da aplicação Web. Esse modelo possui estruturas pré-definidas com intuito de poupar trabalho ao programador. A seguir é apresentado na Figura 29 cada uma das partes do modelo utilizado nessa aplicação.

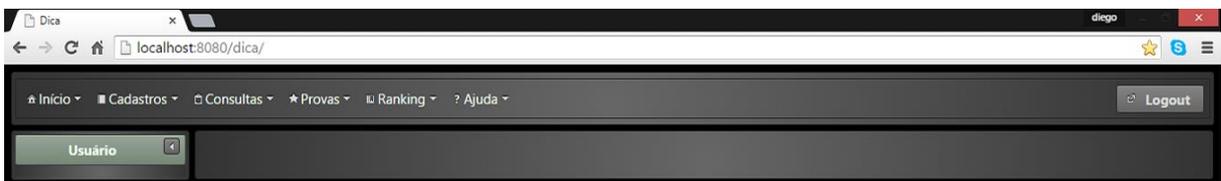


Figura 29 - Interface gráfica do Template.

Na Figura 29 são observadas três partes pré-definidas: no topo encontra-se o menu do sistema; na parte inferior esquerda contém as informações do usuário; e na parte inferior a direita aparece o conteúdo de uma página do sistema. Vale ressaltar que todas as páginas utilizadas neste projeto possuem esse modelo de *template*.

Foi apresentado nesse capítulo toda a estrutura do desenvolvimento deste projeto, visando colocar em prática todo material teórico pesquisado e a proposta do projeto.

5. CONCLUSÃO

O presente trabalho teve como proposta desenvolver uma aplicação Web sobre ensino-aprendizado, visando fornecer aos professores e alunos um novo método de conhecimento educacional. A execução deste trabalho se tornou importante por passar os conhecimentos de ensino-aprendizado com softwares educacionais nos dias atuais, tendo em vista que muitas escolas já utilizam o método de ensino com computação. Acredita-se que o objetivo deste trabalho foi alcançado com a elaboração da aplicação Web descrito no trabalho.

Espera-se em trabalhos futuros, que este projeto possa cada vez mais se aperfeiçoar ou servir de inspiração para outros métodos acadêmicos de ensino-aprendizado com softwares-educativos.

REFERÊNCIAS

ARANHA, Gláucio. **Jogos Eletrônicos como um conceito chave para o desenvolvimento de aplicações imersivas e interativas para o aprendizado.** Disponível em <<http://www.cienciasecognicao.org/pdf/v07/m31685.pdf>> Acesso em 08/10/2014.

BITTENCOURT, José Ricardo; GIRAFFA, Lucia Maria Martins. **A Utilização dos Role-Playing Games Digitais no Processo de Ensino-Aprendizagem.** 2003. 62p. Dissertação. Faculdade de Informática PUCRS. Porto Alegre. 2003.

FALKEMBACH, Gilse. **Concepção e Desenvolvimento de Material Educativo Digital.** Disponível em <<http://seer.ufrgs.br/renote/article/view/13742/7970>> Acesso em 09/10/2014.

GALLARDO, David. **Introdução à Plataforma Eclipse.** Disponível em <<http://www.ibm.com/developerworks/br/library/os-eclipse-platform/>> Acesso em 18/07/2015.

GUERRA, João Henrique Lopes. **Utilização do Computador no Processo de Ensino-Aprendizagem: Uma Aplicação em Planejamento e Controle da Produção.** 2000. 159p. Dissertação. Escola de Engenharia de São Carlos – Universidade de São Paulo. São Carlos. 2000.

GRZESIUK, Diorgenes Felipe. **O uso da Informática na Sala de Aula como Ferramenta de Auxílio no Processo Ensino-Aprendizagem.** 2008. 39p. Monografia. Universidade Tecnológica Federal do Paraná – Campus Medianeira. Medianeira. 2008.

HAGUENAUER, Cristina. **Uso de Jogos na Educação Online: a Experiência do LATEC/UFRJ.** Disponível em <<http://www.latec.ufrj.br/revistas/index.php?journal=educaonline&page=article&op=view&path%5B%5D=143>> Acesso em 09/10/2014.

JANDL, Peter Jr. **Introdução ao Java.** São Francisco. Núcleo de Educação a Distância: 1999.

JUCÁ, Sandro César Silveira. A relevância dos softwares educativos na educação profissional. **Ciências e Cognição**. v8. p.22-28. 2006.

MILANI, André. **PostgreSQL – Guia do Programador**. São Paulo: Novatec, 2008.

MORATORI, Patrick Barbosa. **Por que Utilizar Jogos Educativos no Processo de Ensino Aprendizagem?**. 2003. 28p. Trabalho de Conclusão de Curso. Universidade Federal do Rio de Janeiro. Rio de Janeiro. 2003.

OLIVEIRA, Hêlbert Coelho de. **JSF Eficaz: As melhores práticas para o desenvolvedor Web Java**. São Paulo: Casa do Código, 2013.

PORTA, Giovana La. **O Software Apache Tomcat**. Disponível em <<http://www.frameworksystem.com/blog/o-software-apache-tomcat/>> Acesso em 18/07/2015.

PRIETO, Lilian. **Uso das Tecnologias Digitais em Atividades Didáticas nas Séries Iniciais**. Disponível em <<http://seer.ufrgs.br/renote/article/view/13934/7837>> Acesso em 09/10/2014.

RICHTER, Denis. Informática no Processo Ensino-Aprendizagem: Contribuindo para uma Nova Escola. **Revista Formação**. v2. n13. 2006.

SANTANA, Juliana Cristina de. **A Utilização de Novas Tecnologias no Ensino de Ciências**. Disponível em <http://www.senept.cefetmg.br/galerias/Arquivos_senept/anais/terca_tema1/TerxaTema1Artigo14.pdf> Acesso em 22/02/2015.

SAVI, Rafael. **Jogos Digitais Educacionais: Benefícios e Desafios**. Disponível em <<http://seer.ufrgs.br/renote/article/view/14405/8310>> Acesso em 09/10/2014.

TEXEIRA, Rafael Farias. **Criador do Duolingo conta como voltou a empreender mesmo milionário**. Disponível em <<http://revistapegn.globo.com/Noticias/noticia/2014/02/criador-do-duolingo-conta-como-voltou-empreender-mesmo-milionario.html>> Acesso em 01/03/2015.

VALENTE, José Armando. **Computadores e Conhecimento Repensando a Educação**. Campinas: Editora SP:UNICAMP/NIED. 1993.

VALENTE, José Armando. **Computadores e Conhecimento Repensando a Educação**. 2. ed. Campinas: Editora SP:UNICAMP/NIED. 1998.

VALENTE, José Armando. **O Computador na Sociedade do Conhecimento**. Campinas: Editora SP:UNICAMP/NIED. 1999.

VIDAL, Elisabete. **Ensino à Distância vs Ensino Tradicional**. 2002. 76p. Disponível em <http://homepage.ufp.pt/~lmbg/monografias/evidal_mono.pdf> Acesso em 22/02/2015.

VIEIRA, Zacarias Nascimento de Lima. **A Informática na Educação**. 2006. 46p. Monografia. Universidade Candido Mendes. Rio de Janeiro. 2006.