

**UNIVERSIDADE FEDERAL DE SERGIPE
CAMPUS ALBERTO CARVALHO
DEPARTAMENTO DE SISTEMAS DE INFORMAÇÃO**

ANDRÉS SILVA MENÉNDEZ

**UM MÉTODO AUTOMÁTICO DE AVALIAÇÃO PARA
ENSINO DE PROGRAMAÇÃO EM BANCO DE DADOS**

**ITABAIANA
2016**

**UNIVERSIDADE FEDERAL DE SERGIPE
CAMPUS ALBERTO CARVALHO
DEPARTAMENTO DE SISTEMAS DE INFORMAÇÃO**

ANDRÉS SILVA MENÉNDEZ

**UM MÉTODO AUTOMÁTICO DE AVALIAÇÃO PARA
ENSINO DE PROGRAMAÇÃO EM BANCO DE DADOS**

Trabalho de Conclusão de Curso
submetido ao Departamento de Sistemas
de Informação da Universidade Federal de
Sergipe, como requisito parcial para a
obtenção do título de Bacharel em
Sistemas de Informação.

Orientador: Msc. ANDRÉ VINICIUS RODRIGUES PASSOS NASCIMENTO
Coorientador: Dr. METHANIAS COLAÇO RODRIGUES JÚNIOR

**ITABAIANA
2016**

MENENDEZ, Andrés Silva.

Um método automático de avaliação para ensino de programação em banco de dados/ Andrés Silva Menéndez – Itabaiana: UFS, 2015. 89f.

Trabalho de Conclusão de Curso em Bacharel em Sistemas de Informação – Universidade Federal de Sergipe, Curso de Sistemas de Informação, 2016.

1. Avaliação Automática.
2. Banco de Dados.
3. Sistemas de Informação.

I. Um método automático de avaliação para ensino de programação em banco de dados

ANDRÉS SILVA MENÉNDEZ

**UM MÉTODO AUTOMÁTICO DE AVALIAÇÃO PARA ENSINO
DE PROGRAMAÇÃO EM BANCO DE DADOS**

Trabalho de Conclusão de Curso submetido ao corpo docente do Departamento de Sistemas de Informação da Universidade Federal de Sergipe (DSIITA/UFS) como parte dos requisitos para obtenção do grau de Bacharel em Sistemas de Informação.

Itabaiana, 25 de maio de 2016.

BANCA EXAMINADORA:

Prof. André Vinicius Rodrigues Passos Nascimento, Mestre
Orientador
DSIITA/UFS

Prof. Methanias Colaço Rodrigues Júnior, Doutor
Coorientador
DSIITA/UFS

Prof. José Aélío de Oliveira Júnior, Doutor
DSIITA/UFS

DEDICATÓRIA

Dedico este trabalho aos meus pais, Andrés Menéndez e Kelly Chrystian, que sempre me apoiaram, me deram conselhos e torceram por mim na minha vida profissional. Sem este apoio provavelmente não estaria apresentando este trabalho. Meu amor e gratidão por vocês é incondicional. Portanto, nenhuma palavra irá conceituar meu agradecimento, entretanto vou deixar aqui meu muito obrigado a tudo que ambos fizeram e fazem por mim.

Aos meus irmãos Lucca e Túlio, que sempre estiveram comigo. À minha namorada, Alessandra Batista, que sempre me apoiou quando eu precisei, dividindo alegrias e decepções. Às minhas famílias. Aos meus amigos. Aos meus familiares que não estão mais presentes, Valter, Walter, Creuza, Orlando e Nardé. Dedico a todos que contribuíram de alguma forma para que esse trabalho fosse apresentado.

AGRADECIMENTOS

Agradeço primeiramente a Ele, pois sem ele nada disto seria possível, Deus. Que me concebeu vida, força, sabedoria, proteção e coragem para ter a felicidade de apresentar este trabalho. Serei eternamente grato ao maior de todos, o Senhor Deus.

Aos meus pais agradeço por todo o esforço de se doar ao máximo, abdicando de certas situações para que um dia eu pudesse chegar lá. Logo, não há palavras para descrever como eu agradeço por tudo, porém vou deixar aqui meu eterno obrigado a tudo que vocês dois fizeram e fazem tudo para que eu seja um profissional honesto.

Quero agradecer também à minha namorada Alessandra que sempre esteve comigo, me apoiando, me ouvindo, dando conselhos e acreditando em mim. Meu muito obrigado por você estar sempre ao meu lado. Amo você.

Agradecer a minha família Menéndez e minha família Pinto. Aos meus avós, tios, primos, “mãedrastra” e agregados que contribuíram para que eu concluísse minha graduação. Muito obrigado por fazerem de tudo para que eu chegasse até aqui.

Quero agradecer a todos os meus amigos, em especial, aos meus grandes e eternos amigos do Dinâmico, sem exceções. Aos meus amigos do Atlantis que sempre estão comigo nos momentos de lazer. Aos meus amigos do Salesiano. Amigos antigos como Dagoberto, Helton, Tércio, Yamara, Bibita e o velho Candjas. Aos meus amigos antigos da Code2, em especial, Thiers. Ao pessoal da Cencosud. A todos os amigos da UFS, em especial a Cleverton dos Santos, não só por ser, além de mim o único a se formar da minha turma, mas por dividir, angústias, aflições e alegrias. Você é um exemplo a todos, meu caro amigo, pois com toda dificuldade você também chegou lá!

Assim, quero deixar aqui meu agradecimento a todos os meus amigos e conhecidos no qual contribuíram de algum modo para minha formação.

Por último e não menos importante. Quero agradecer aos meus mestres por todo ensinamento, compreensão, caronas principalmente, e incentivo a nunca desistir. Assim, agradeço a todos os professores sem exceção, mas em especial a André Luís, Alcides Benicasa, Aélcio Oliveira, Adolfo Guimarães, Mai ly Faro e Eugênio Cardoso.

Agradecer ao prof. Marcos Dósea por toda a paciência, conselhos e grandes ensinamentos que levarei para sempre! Minha formação não seria a mesma sem você. Ao meu coorientador, Methanias Colaço, por todas as caronas, resenhas, infinitos ditados e por todas as suas aulas, que deveriam ser obrigatórias a todos os cursos. São aulas para a vida toda. Ao meu querido amigo, competentíssimo professor e excelente orientador André Vinicius, por todo seu tempo e paciência, toda sua generosidade, pelas caronas, pelos almoços, pelos conselhos, por tudo. Sou muito grato a você, que como um segundo pai, ficava chateado quando eu tirava nota ruim. Agradeço também ao melhor professor de todos, Andrés I. M. Menéndez, que não cabem palavras para descrever o quanto me ensinou desde que eu era criança. São infinitas qualidades para representar o professor que você é, e foi pra mim.

De uma forma geral, queria deixar meu agradecimento a todos que contribuíram de certa forma para minha formação. Muito Obrigado.

MENENDEZ, Andrés Silva. **Um Método Automático de Avaliação Para Ensino de Programação em Banco de Dados**. 2015. Trabalho de Conclusão de Curso – Curso de Sistemas de Informação, Departamento de Sistemas de Informação, Universidade Federal de Sergipe, Itabaiana, 2016.

RESUMO

A avaliação manual de exercícios de programação pode representar um obstáculo no processo de ensino e aprendizagem. Exercícios de programação exigem um tempo considerável do professor para que sejam efetivamente avaliados uma vez que muitos casos de testes precisam ser executados para verificar de forma completa a resposta de um aluno. Em turmas com número relativamente grande de alunos o professor pode não dispor do tempo necessário para validar por completo todas as soluções, e o *feedback* dado aos alunos pode não acontecer no tempo adequado. A solução para esse problema está na utilização de ferramentas de avaliação automática de exercícios de programação. Apesar da existência de vários trabalhos e ferramentas na área de avaliação automática, nenhum deles aborda a avaliação automática de exercícios de programação voltados para procedimentos armazenados em banco de dados. Logo, as turmas de programação em banco de dados continuam sofrendo dos problemas relacionados com a avaliação manual de soluções. Nesse trabalho, propomos um método de avaliação automática de procedimentos armazenados em banco de dados. Através de uma abordagem dinâmica, a ferramenta analisa, por meio da comparação de resultados, os procedimentos submetidos à avaliação.

Palavras-chave:

Avaliação Automática, Banco de Dados, Sistemas de Informação.

ABSTRACT

The manual evaluation of programming exercises can be an obstacle in the process of teaching and learning. Programming exercises require a considerable time of the teacher to assess effectively because many cases tests need to be executed to verify the response of a student. In classes with relatively high number of students, the teacher can't afford the time required to validate completely all the solutions, and the feedback given to students can't happen at the best time. The solution for this problem is use tools of automatic evaluation of programming exercises. Despite the existence of several studies and tools in this subject, none of them talk about the automatic evaluation of programming exercises with focus on stored procedures in the database. Therefore, the database programming classes still suffering from problems related to manual assessment solutions. In this paper, we propose an online tool for automatic evaluation of stored procedures in the database. Through a dynamic approach, the analysis tool the procedures evaluation by comparison of results.

Keywords:

Assessment Automatic, Databases, Information Systems.

LISTA DE FIGURAS

Figura 1: <i>Arquitetura da Ferramenta</i>	22
Figura 2: <i>Processo de Avaliação Automática</i>	24
Figura 3: <i>Diagrama de casos de uso da ferramenta</i>	30
Figura 4: <i>Fluxo de Efetuar Login</i>	51
Figura 5: <i>Fluxo de Criar Questão</i>	52
Figura 6: <i>Fluxo de Validar Questão</i>	52
Figura 7: <i>Fluxo de Ativar Perfil de Professor</i>	53
Figura 8: <i>Fluxo de Submissão de Respostas</i>	54
Figura 9: <i>Fluxo de Comentar Resultado</i>	54
Figura 10: <i>Esquema de dados da ferramenta</i>	56
Figura 11: <i>Esquema de dados da ferramenta – “Resposta”</i>	58
Figura 12: <i>Esquema de dados da ferramenta – “Usuário”</i>	59
Figura 13: <i>Tela Inicial da ferramenta</i>	61
Figura 14: <i>Tela de Registre-se da ferramenta</i>	62
Figura 15: <i>Tela Menu Principal da ferramenta</i>	63
Figura 16: <i>Tela de Solicitar Perfil de Professor</i>	63
Figura 17: <i>Tela de Ativar Perfil de Professor</i>	64
Figura 18: <i>Tela de Minha Turmas do Professor</i>	64
Figura 19: <i>Tela de Criar Turma</i>	65
Figura 20: <i>Tela de Detalhar Turma</i>	66
Figura 21: <i>Tela de Criar Atividade</i>	66
Figura 22: <i>Tela de Criar Questão- Parte 1</i>	67
Figura 23: <i>Tela de Criar Questão- Parte 2</i>	68
Figura 24: <i>Tela de Visualizar Questão</i>	70
Figura 25: <i>Tela de Questões</i>	71
Figura 26: <i>Tela de Validar Questão Particionada</i>	72
Figura 27: <i>Tela de Questão Visão Aluno</i>	73
Figura 28: <i>Tela de Responder Questão</i>	73
Figura 29: <i>Tela de Visualizar Resultado – Parte 1</i>	74
Figura 30: <i>Tela de Visualizar Resultado – Parte 2</i>	75
Figura 31: <i>Composição do Resultado – Nº Caso de Teste</i>	76
Figura 32: <i>Composição do Resultado – Caso de Teste</i>	76
Figura 33: <i>Composição do Resultado – Resultado Final</i>	76
Figura 34: <i>Histórico de Submissões</i>	77

LISTA DE QUADROS

Quadro 1: <i>Requisitos funcionais e não funcionais da ferramenta.</i>	29
Quadro 2: <i>Caso de Uso Efetuar Login</i>	33
Quadro 3: <i>Caso de Uso Manter Turmas – Criar Turma</i>	34
Quadro 4: <i>Caso de Uso Manter Esquemas – Editar Esquema</i>	35
Quadro 5: <i>Caso de Uso Manter Turmas– Detalhar Turma</i>	36
Quadro 6: <i>Caso de Uso Manter Turmas – Remover Turma</i>	36
Quadro 7: <i>Caso de Uso Matricular na Turma</i>	37
Quadro 8: <i>Caso de Uso Manter Turmas – Criar Turma</i>	38
Quadro 9: <i>Caso de Uso Manter Questões – Editar Questão</i>	39
Quadro 10: <i>Caso de Uso Manter Questões– Detalhar Questão</i>	40
Quadro 11: <i>Caso de Uso Manter Questões – Remover Questão</i>	41
Quadro 12: <i>Caso de Uso Solicitada Perfil de Professor</i>	42
Quadro 13: <i>Caso de Uso Validar Questão</i>	43
Quadro 14: <i>Caso de Uso Ativar Perfil de Professor</i>	44
Quadro 15: <i>Caso de Uso Visualizar Histórico de Submissões - Professor</i>	45
Quadro 16: <i>Caso de Uso Visualizar Histórico de Submissões - Aluno</i>	46
Quadro 17: <i>Caso de Uso Visualizar Resultado de Submissões - Professor</i>	47
Quadro 18: <i>Caso de Uso Visualizar Resultado de Submissões - Aluno</i>	48
Quadro 19: <i>Caso de Uso Submeter Resposta para Avaliação Automática</i>	49
Quadro 20: <i>Caso de Uso Comentar Resultado de Submissões - Professor</i>	50

SUMÁRIO

1. INTRODUÇÃO	11
1.1. Objetivos.....	12
1.1.1. Geral	12
1.1.2. Específicos	12
1.2. Relevância do trabalho.....	13
1.3. Metodologia	13
1.4. Estrutura do trabalho	13
2. REVISÃO BIBLIOGRÁFICA.....	15
2.1. Trabalhos relacionados	15
2.2. Trabalhos Relacionados no âmbito de Ambientes de Apoio ao Ensino de Programação SQL.	18
3. PROJETO DA FERRAMENTA.....	21
3.1. Requisitos funcionais e não funcionais	28
3.2. Casos de Uso	30
3.2.1. Casos de Uso – Nível de Usuário	30
3.2.2. Casos de Uso – Nível de Sistema	32
3.2.2.1. Caso de Uso Autenticar Usuário.....	33
3.2.2.2. Caso de Uso Manter Turmas	34
3.2.2.3. Caso de Uso Matricular na Turma.....	37
3.2.2.4. Caso de Uso Manter Questões.....	38
3.2.2.5. Caso de Uso Solicitar Perfil de Professor	42
3.2.2.6. Caso de Uso Validar Questão.....	43
3.2.2.7. Caso de Uso Ativar Perfil de Professor.....	44
3.2.2.8. Caso de Uso Visualizar Histórico de Submissões	45
3.2.2.9. Caso de Uso Visualizar Resultado de Submissão	47
3.2.2.10. Caso de Uso Submeter Resposta para Avaliação Automática	49
3.2.2.11. Caso de Uso Comentar Resultado de Submissão.....	50
3.3. Fluxo de Sequência de Atividades.....	50
3.4. Modelagem de Dados	56
3.4.1. Componentes do modelo de dados	56
4. FERRAMENTA	61
5. CONCLUSÕES	78
REFERÊNCIAS	81

1. INTRODUÇÃO

A utilização de sistemas de avaliação automática de programação tornou-se recentemente um método importante para ajudar professores e instrutores de cursos de programação a fornecer *feedbacks* rápidos e úteis sobre as soluções dos estudantes (ROMLI; SULAIMAN; ZAMLI, 2010).

Cursos em que há programação envolvida são geralmente conhecidos por serem cursos bastante práticos e têm como objetivo o desenvolvimento da compreensão dos princípios de programação dos alunos (ROMLI; SULAIMAN; ZAMLI, 2010). Habilidades de programação são necessárias para todos os estudantes de ciência da computação e afins. Estes estudantes podem se tornar especialistas em programação apenas por meio de práticas de exercícios intensivos (GUPTAL; DUBEY, 2012).

A avaliação manual de exercícios de programação representa uma extensa carga de trabalho para professores e assistentes, principalmente devido ao notável aumento do número de alunos nas turmas de programação (GUPTAL; DUBEY, 2012). Esta avaliação manual, aparentemente, promove dificuldades em garantir consistência e precisão dos resultados da avaliação. Além disso, o *feedback* tardio dado pelo professor dificulta o processo de ensino e aprendizagem. Portanto, a avaliação automática de exercícios de programação tornou-se um importante método de correção de exercícios de programação dos alunos, bem como um método rápido de prover *feedback* sobre suas soluções.

Sistemas de avaliação automática são usados em diversos cenários, com diferentes funções, como em ambientes de aprendizagem, provas de programação e sistemas de concurso, por exemplo.

Embora não represente uma abordagem nova no auxílio ao ensino de programação, os sistemas de avaliação automática são geralmente direcionados para turmas iniciais de programação e atendem somente a determinadas linguagens de programação.

Atualmente, muitos módulos de sistemas de informações são construídos com uma arquitetura que prioriza a utilização de procedimentos armazenados nos Sistemas Gerenciadores de Banco de Dados, principalmente para processamentos que não possuem interação com usuários. Nessa direção, vários cursos de programação têm

como objetivo ensinar programação utilizando extensões da linguagem sql a fim de produzir procedimentos e funções armazenadas. Esses cursos, por não fazerem uso de um sistema de avaliação automática, enfrentam as mesmas dificuldades que um curso de programação tradicional.

A não utilização de sistemas de avaliação automática em cursos de programação de banco de dados é resultado da inexistência de sistemas de avaliação automática direcionados para esse fim. Os poucos trabalhos direcionados para a área de banco de dados, a exemplo de (SILVA; LINO; FAVERO; 2009) e outros citados na seção de trabalhos relacionados, apresentam auxílio apenas na execução de comandos SQL independentes.

Nesse sentido, o objetivo desse trabalho é projetar e construir um sistema de avaliação automática de programação para ensino de programação em banco de dados.

Esperamos, com esse trabalho, fornecer um *feedback* automático de qualidade para os programadores em banco de dados e, conseqüentemente, aumentar a qualidade dos códigos escritos para estes ambientes. Além disso, será possível reduzir a carga de trabalho do professor/instrutor, visando uma melhoria no acompanhamento dos exercícios executados.

1.1. Objetivos

1.1.1. Geral

O objetivo desse trabalho é propor um método automático de verificação de exercícios de programação de procedimentos armazenados em banco de dados.

1.1.2. Específicos

- Identificar requisitos necessários para uma abordagem de avaliação automática de procedimentos armazenados em banco de dados;
- Propor uma arquitetura para uma ferramenta de avaliação automática de programação em banco de dados;
- Projetar uma ferramenta *on-line* para utilização do método automático de avaliação;

- Construir uma ferramenta de avaliação automática de programação em banco de dados.

1.2. Relevância do trabalho

Ferramentas de avaliação automática representam atualmente um grande aliado no ensino de programação. Apesar da existência de muitas ferramentas de avaliação para diferentes linguagens de programação, são poucas as iniciativas voltadas para a área de banco de dados. Este trabalho procura preencher essa lacuna através do projeto e construção de uma abordagem voltada para procedimentos armazenados em banco de dados. Espera-se que essa ferramenta possa ser utilizada pelas disciplinas de programação em banco de dados, auxiliando professores e alunos, além de servir de incentivo para o surgimento de novas ferramentas e abordagens para a área de ensino em banco de dados.

1.3. Metodologia

Inicialmente, será realizada a revisão da literatura e pesquisa bibliográfica sobre avaliação automática de programação.

Após a revisão da literatura, será feito o levantamento de requisitos para a abordagem direcionada para procedimentos armazenados. Serão levantados requisitos funcionais e não funcionais que devem descrever todos os aspectos significativos do sistema.

Após o levantamento de requisitos, a ferramenta será projetada.

Após o projeto, será realizado o desenvolvimento da ferramenta contemplando os requisitos do escopo definido para o trabalho.

1.4. Estrutura do trabalho

O restante do trabalho está estruturado como segue. No capítulo 2 são apresentados os principais artigos e ferramentas encontrados relacionados com o tema do nosso trabalho. No capítulo 3 é apresentada a abordagem para avaliação automática de procedimentos armazenados. No capítulo 4, são descritas as principais atividades

realizadas durante o projeto da Ferramenta, levantamento de requisitos, diagrama de casos de uso, protótipos e esquema de dados lógico. O capítulo 5 descreve um estudo de caso para apresentação das funcionalidades da ferramenta. Finalmente, no capítulo 6 são apresentadas as conclusões do trabalho.

2. REVISÃO BIBLIOGRÁFICA

Sistemas de Avaliação Automática fornecem uma possibilidade para o aprendiz apresentar uma solução, testá-la sobre um conjunto de testes específicos, e receber um *feedback* instantâneo. Em turmas de programação, estes sistemas podem auxiliar professores na correção de exercícios, trabalhos e até provas dos alunos (SKUPAS, 2010). Em geral, os principais objetivos em promover uma ferramenta de avaliação automática são principalmente para reduzir a carga de trabalho dos professores e tutores e para melhorar a consistência da avaliação de exercícios de programação dos alunos (GUPTAL, DUBEY, 2005).

É possível perceber várias vantagens na avaliação automática de programação. Em (PIETERSE, 2013) são mencionados alguns fatores que se destacam na avaliação automática, como velocidade de resposta, disponibilidade, consistência e objetividade da avaliação. Outro importante destaque é a vantagem do *feedback* imediato para os alunos, especialmente para iniciantes que podem se beneficiar de equívocos iniciais. A vantagem de obter um resultado rápido para uma resposta é a principal vantagem da ferramenta em questão, pois facilita a aprendizagem, permitindo a prática dos alunos com uma avaliação a qualquer momento e em qualquer lugar, bem como a possibilidade de realizar um número maior de tarefas.

Nas subseções a seguir serão apresentados alguns trabalhos relacionados à programação em geral, bem como trabalhos relacionados no âmbito de ambientes de apoio ao ensino de programação SQL.

2.1. Trabalhos relacionados

Nesta seção apresentaremos algumas propostas que possuem objetivo semelhante ao nosso trabalho: avaliar automaticamente exercícios de programação. As propostas apresentadas representam trabalhos moderadamente e fortemente relacionados.

2.1.1. Automatic Assessment of Programming

Em (GUPTAL; DUBEY, 2005) é proposto um sistema de avaliação automática, utilizando a abordagem de análise dinâmica, para avaliar automaticamente e instantaneamente trabalhos de programação. Este sistema funciona da seguinte forma: o aluno submete sua função que representa um programa ou parte de um programa, e a

mesma é compilada, executada e verificada usando uma função inversa, que é a função considerada correta pelo professor. O sistema foi utilizado em um curso de programação, mas nenhum experimento foi realizado. Apenas foram coletados alguns números com relação ao total de submissões, total de programas com erros de compilação, total de programas com erros de lógica e total de programas corretos. O sistema proposto tem algumas limitações apontadas pelos próprios autores: a) a entrada de dados só pode ser feito por argumentos na linha de comando; b) as saídas precisam ter um formato pré-definido; c) o sistema não é web, portanto é necessário que todos os alunos instalem a ferramenta.

2.1.2. Programming Assessment and Test Data Generation

Em (GUPTAL; SULAIMAN; ZAMLI, 2010) é apresentado uma revisão sobre as abordagens que têm sido utilizadas em vários estudos no que diz respeito à avaliação automática de programação, a geração de dados de teste e integração entre ambos. Como parte do método, o processo de geração dos dados de teste desempenha parte integrante para realizar uma análise dinâmica nos programas dos alunos. Vários métodos automatizados para a geração de dados de teste, em particular no campo de testes de software estão disponíveis. Infelizmente, eles raramente são utilizados no contexto da área de avaliação automática de programação. A partir do resultado da avaliação, é percebido que ainda há uma lacuna na utilização de técnicas de geração de dados de teste nesta área. Os testes para a avaliação automática de programação, além de detectar erros de programação, também devem garantir que sejam alcançados os objetivos de ensino, mostrando aos alunos onde se deve melhorar. O estudo também destaca que testes caixa-preta são mais utilizados que teste caixa-branca nas ferramentas observadas. Assim, um conjunto de testes deve ser cuidadosamente concebido, incorporando os critérios adequados e suficientes para se obter uma boa cobertura de teste para avaliação de exercícios de programação.

2.1.3. Feedback Improvement in Automatic Program Evaluation Systems

Em (SKUPAS, 2010) são propostas várias ideias para possíveis melhorias no teste de caixa-preta e um método estruturado de avaliação automática, que podem levar a uma melhoria no *feedback* dado ao aluno. Depois de elaborar o método e apresentar as ideias foi possível discutir com um grupo de cinco alunos sobre vantagens e desvantagens do método. Os alunos esclareceram que as mensagens deixaram muito a

desejar e que necessitam de *feedback* mais claro sobre seus erros. Além disso, eles também gostariam de ter informações mais precisas sobre erros em tempo de execução.

2.1.4. The recent development of automated programming assessment

Em (LIANG; LI; XU, 2009) são discutidas vantagens e desvantagens das análises estática e dinâmica levando em consideração a prática instrucional em um sistema de avaliação automática. Embora muitos destes sistemas tenham sido provados serem de grande ajuda para os professores e alunos de programação, ainda existem vários problemas não solucionados. Como por exemplo, a falta de segurança nos sistemas web. Além disso, existe uma baixa precisão nas definições dos erros exibidos e a falta de uma análise estática eficaz. Os autores indicaram novos rumos no desenvolvimento de avaliação automática, como por exemplo, uma padronização das mensagens retornadas dos sistemas de avaliação automática de programação, uma forma de verificar o código-fonte e um sistema tutor inteligente. Além disso, um *feedback* apropriado para os estudantes sobre como o programa pode ser melhorado é uma direção considerável. No entanto, alguns componentes críticos de ensino e aprendizagem precisam ser cuidadosamente considerados. Logo, algumas abordagens ideais ainda estão sob investigação. A forma de utilização eficaz de análise dinâmica e estática na prática instrucional é sugerida, de modo que os professores possam utilizar os resultados deste estudo para escolher a abordagem que mais se adequa em seu contexto.

2.1.5. Automatic Test-Based Assessment of Programming – A Review

Em (DOUCE; LIVINGSTONE; ORWELL, 2005) é apresentada uma revisão da literatura dos trabalhos que envolvem projetos que avaliam automaticamente exercícios de programação. Três gerações de sistemas de avaliação foram identificadas: a) Os sistemas de primeira geração, que representam as tentativas iniciais para automatizar testes; b) Os sistemas de segunda geração, que são caracterizados pelo uso de ferramentas baseadas em linha de comando, às vezes em associação com interfaces GUI localmente criados e mantidos; c) Os sistemas de terceira geração, que fazem uso de tecnologias baseadas na web para fornecer apoio aos educadores. Os autores indicaram que diferentes educadores e instituições podem aplicar os sistemas de avaliação automática de diferentes maneiras. Uma das vantagens é que os avaliadores humanos são falíveis, enquanto avaliadores automáticos podem fornecer respostas objetivas (desde que sejam devidamente programadas). Existem, naturalmente, desvantagens. A

principal delas são as restrições que se aplicam ao que pode ser avaliado automaticamente. Por fim, os autores concluíram que sempre haverá a necessidade de avaliadores humanos. Em vez de substituir o tutor, estes sistemas podem fornecer suporte. Eles também podem apoiar os alunos, permitindo que adquiram maior confiança em seu trabalho. Os sistemas podem ajudar os tutores e educadores, permitindo-lhes identificar os equívocos dos alunos, facilitando o ensino da programação e da engenharia de software de forma mais eficaz.

2.2. Trabalhos Relacionados no âmbito de Ambientes de Apoio ao Ensino de Programação SQL.

Nesta seção apresentaremos algumas propostas de ferramentas voltadas à linguagem SQL. As propostas apresentadas representam trabalhos com ambientes voltados para consultas básicas da linguagem SQL. Não foram encontrados até o momento trabalhos que relatem ferramentas que tenham como abordagem a avaliação automática de procedimentos armazenados em banco de dados.

2.2.1. Um Laboratório Virtual de Ensino de Programação SQL no Ambiente Moodle.

É proposta por (SILVA; LINO; FAVERO; 2009) a implantação de um laboratório virtual de programação de SQL no ambiente Moodle. O laboratório corresponde a uma adaptação da ferramenta LabSQL (Lino; 2007), a qual foi desenvolvida para auxiliar no processo de ensino/aprendizagem de SQL num ambiente *on-line*. Este ambiente propõe uma solução para o desafio de estimular o aprendiz a aperfeiçoar a sua solução buscando, além de uma resposta que retorna o resultado correto, uma consulta com complexidade próxima da solução ótima. Por meio de métricas de engenharia de software e métodos estatísticos o LabSQL atribui uma nota na consulta SQL submetida pelo aprendiz, que em essência é a distância da consulta do aprendiz em relação à solução ótima, inicialmente cadastrada pelo professor. Quando o aprendiz realiza uma solução menos complexa, automaticamente essa consulta passa a ser a solução ótima.

2.2.2. Evaluating Effectiveness of Feedback in Sql-Tutor.

O SQL-Tutor (MITROVIC, 2000), que é um sistema tutor inteligente para programação SQL, aborda apenas consultas sobre o comando Select. Um módulo

pedagógico seleciona o conteúdo a ser apresentado ao aprendiz, e dependendo de seu modelo e das respostas dos exercícios prévios, o sistema retornará um *feedback* específico ao aprendiz.

O destaque desta ferramenta é seu *feedback*, que utiliza um componente inteligente, fundamentado na técnica de modelagem baseada em restrições para a codificação das regras; cada situação de erro é codificada em uma regra. Esse sistema apresenta versões de implementação com mais de 600 regras (MITROVIC; MARTIN, 2000). Observam-se, porém, dois problemas nesta abordagem: o primeiro é o custo para monitorar todas as situações e ficar codificando regras; o segundo está na dificuldade de garantir que as regras cobrem todas as situações.

2.2.3. Sqlator: an Online Sql Learning Workbench.

O SQLator (SADIQ; ORLOWSKA; SADIQ; LIN, 2004) é um ambiente que utiliza uma abordagem denominada de correção-pelo-retorno. Esta ferramenta dispõe de um tutorial integrado que apresenta conceitos fundamentais, oferecendo diversos bancos de dados para prática. O SQLator conta ainda com um conjunto de questões de testes, e permite a execução real das consultas sobre os bancos de dados. O SQLator possui textos e tutoriais sobre programação e coleta um conjunto de estatísticas detalhadas sobre cada aprendiz (tempo de login, número de execuções, dentre outras). A razão fundamental para o êxito da implantação do SQLator tem sido a sua capacidade de fornecer feedback aos alunos. Ele complementa o papel de um tutor, com a vantagem adicional de ser disponível 24/7, uma vez que está disponível na web. A capacidade de praticar de forma independente um grande número de consultas SQL com diferentes níveis de complexidade, e receber feedback instantâneo sobre correção de consulta fez com que o SQLator ganhasse certa popularidade entre os estudantes.

2.2.4. Backwash Effect on Sql Skills Grading

O AsseSQL (PRIOR; LISTER, 2004), assim como o SQLator é um ambiente que utiliza a abordagem correção-pelo-retorno. Esta é uma ferramenta web de treinamento e avaliação de consultas SQL. Apesar disto, o AsseSQL provê um conceito certo ou errado das consultas realizadas pelos aprendizes. As provas e testes são armazenados em um banco de dados, o qual contém lista de questões; classificação das questões; tipo de prova; duração; número de questões, dentre outras informações. Assim, as provas são criadas randomicamente por aprendiz. Durante a prática dos exercícios, o resultado

esperado da consulta é mostrado. Existem dois tipos de resultados: 1) os de erros de sintaxe; 2) se a resposta traz ou não o resultado correto.

3. ABORDAGEM PARA BANCO DE DADOS

Este capítulo apresenta uma abordagem de avaliação automática para exercícios de programação de banco de dados. É apresentada uma proposta de arquitetura e características essenciais a uma ferramenta de avaliação automática de procedimentos armazenados.

3.1. Avaliação Automática de Procedimentos Armazenados

Cursos de Computação, principalmente aqueles voltados para o desenvolvimento de sistemas, apresentam disciplinas no seu projeto pedagógico que têm como ementa a programação em banco de dados. O objetivo dessas disciplinas é fornecer ao aluno o conhecimento necessário sobre procedimentos armazenados e funções. Nessas disciplinas os alunos utilizam extensões da linguagem SQL para criar programas que mesclam instruções imperativas e comandos declarativos da linguagem SQL. O método de ensino nos cursos de programação em banco de dados segue o mesmo padrão dos cursos iniciais de programação. Após o estudo da sintaxe básica da linguagem, os alunos fazem exercícios de programação e, paralelamente, são apresentadas novas características da linguagem, novas funções e melhores práticas. Desse modo, os cursos de programação de banco de dados também podem se beneficiar de ferramentas de avaliação automática de programação.

Através da revisão da literatura apresentada no capítulo 2, percebe-se que uso de ferramentas de avaliação automática para cursos de programação não representa algo novo, apesar da recente motivação por parte de pesquisadores e educadores. No entanto, também pela revisão da literatura, percebe-se que o foco na área de Banco de Dados tem sido apenas a avaliação de consultas da linguagem SQL. Essa afirmação é sustentada pela inexistência de ferramentas e trabalhos que tenham como objetivo a criação de ferramentas de avaliação automática para procedimentos armazenados.

Esse trabalho busca preencher essa lacuna através da apresentação de uma proposta de arquitetura para uma ferramenta de avaliação voltada para banco de dados, da identificação das principais características que uma ferramenta dessa natureza deve oferecer, e da implementação de um protótipo de ferramenta que demonstre a viabilidade da arquitetura.

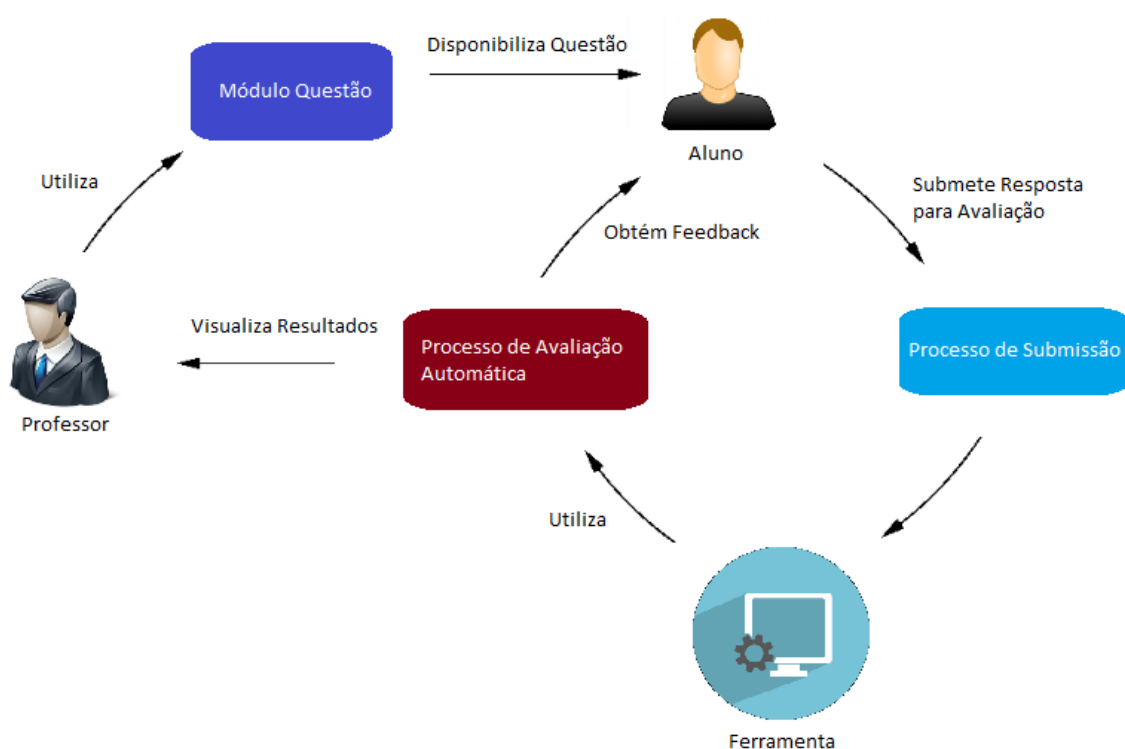


Figura 1: Arquitetura da Ferramenta

3.2. Módulos da Ferramenta

Esta seção descreve detalhadamente os módulos utilizados para criação da ferramenta. Como foi possível visualizar na figura 1, o fluxo começa com o professor utilizando o módulo da questão. Este disponibiliza uma questão para o aluno submeter a resposta no processo de submissão. Após isto, a ferramenta utiliza o processo de avaliação automática para gerar os resultados.

3.2.1. Módulo da Questão

Todo o fluxo da ferramenta (figura 1) se inicia no módulo da criação da questão. Este módulo apresenta dois principais processos: Criação e Validação.

Para que o processo de criação seja concluído com sucesso, é preciso que primeiramente, o professor se dedique a descrever a questão em um grau elevado de especificidade, para evitar ambiguidade. Posteriormente, o professor tem a possibilidade de construir um ambiente onde será aplicado sua questão. Porém, isto não é obrigatório. Entretanto, é necessário que o professor desenvolva e preencha sua resolução sobre a

questão. Esta resolução será de fundamental importância no módulo posterior de Processo de Avaliação. Por último e não menos importante, é necessário que o professor preencha os casos de teste. O preenchimento dos casos de teste é livre, ou seja, a ferramenta coopera para que o professor possa criar inúmeros casos de teste para uma mesma questão.

O processo da validação consiste em validar as informações que foram inseridas no processo de criação. Caso haja alguma divergência entre o ambiente projetado, a resolução e os casos de teste, a mesma será informada ao professor. É importante mencionar que a questão criada pelo professor somente é visível para o aluno após a etapa da validação.

3.2.2. Processo de Submissão

No processo de submissão de uma resposta para avaliação automática, o aluno deve responder a questões anteriormente criadas pelo professor ou instrutor. A resposta do aluno é submetida através de um formulário no qual o aluno inclui o nome e o código do procedimento.

O foco deste processo não é exibir erros sintáticos encontrados nos procedimentos de resposta dos alunos. Pelo contrário, o objetivo deste módulo é preparar o procedimento do aluno para o módulo posterior de avaliação. Logo, é importante que o aluno não utilize a ferramenta com a finalidade de teste, e sim, quando acreditar que seu procedimento está correto com testes em outros ambientes de banco de dados.

3.2.3. Processo de Avaliação Automática

Para o processo de avaliação automática de procedimentos armazenados foi utilizada a estratégia de analisar dinamicamente os resultados obtidos entre a resolução do professor e a resolução do aluno com a finalidade de exibir os resultados que obtiverem igualdade e resultados que diferem.

Este módulo tem origem com a execução do ambiente preparatório criado pelo professor, se houver. A execução deste ambiente é importante, caso a questão necessite de um ambiente para que seja elaborada a questão. Após isto, temos a execução do procedimento de resolução do professor para a questão. A resolução do professor será

utilizada na comparação dos resultados dos alunos. Depois, temos as execuções dos casos de teste. Com estas execuções dos casos de teste sobre a resolução do professor serão obtidos os resultados do professor que serão armazenados para comparação com os resultados dos alunos. Para cada caso de teste é gerado um resultado.

Com os resultados do professor sobre essa questão, agora é o momento de executar o procedimento de resposta do aluno sobre os casos de teste cadastrados pelo professor. Caso haja um ambiente preparatório para a questão, este será criado para o aluno. Após isto, temos a execução da resposta do procedimento armazenado do aluno sobre os mesmos casos de teste que foram utilizados na execução do professor. Assim, com a obtenção dos resultados das execuções dos casos de teste, tanto do aluno como do professor, é possível realizar a comparação que é o ponto chave deste módulo. Para finalizar o processo, os resultados das comparações são exibidos para o aluno e professor.

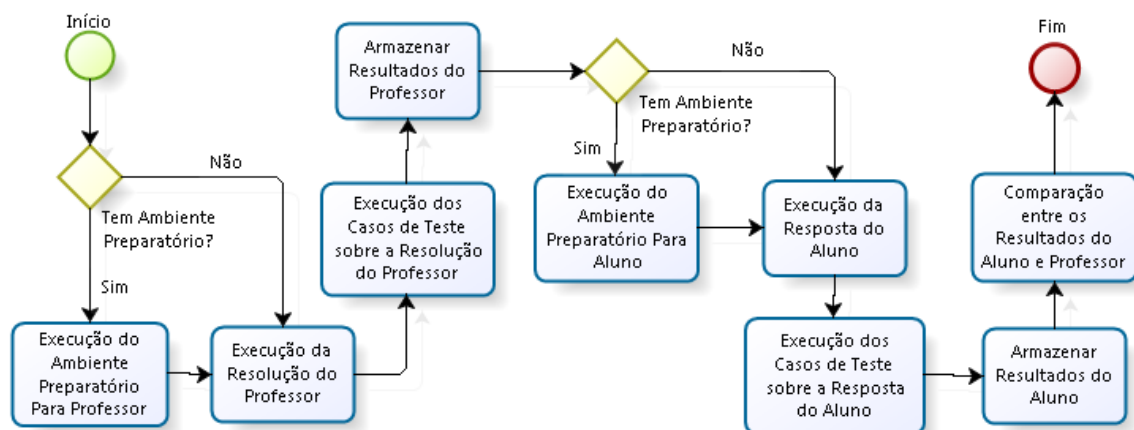


Figura 2: Processo de Avaliação Automática

3.2.4. Casos de Teste

O procedimento do estudante é testado e executado sobre os casos de testes especificados pelo professor. Os mesmos casos de teste para determinada questão serão executados tanto no procedimento de resolução do professor, como no procedimento do aluno. A ferramenta dispõe da possibilidade da criação de diversos casos de teste para uma mesma questão. O formato definido pela ferramenta deixa a cargo do professor a cobertura dos testes realizados sobre a resposta para a questão no formato mais genérico possível. Assim, a ferramenta somente irá testar aquilo que o professor acreditar que é

necessário. No capítulo 5 é descrito um estudo de caso onde é possível visualizar na seção Criar Questão, como criar os casos de teste de forma ideal.

3.3. Características Adicionais

Nesta seção é apresentada uma discussão sobre algumas características que contribuem para o processo de avaliação automática de exercícios de programação em bancos de dados. Estas características foram identificadas através da revisão da literatura, juntamente com a necessidade de adaptação da ferramenta em questão. Alguns trabalhos influenciaram diretamente a identificação dessas características (ILANTOLA, AHONIEMI, KARAVIRTA, SEPPALA, 2010).

3.3.1. Especificação Clara de Questões

A necessidade de especificar a descrição da questão é nítida, pois caso a especificação seja ambígua ela permitirá diferentes interpretações, o que prejudica bastante o processo de avaliação.

Desse modo, com a finalidade de evitar ambiguidades, é necessário que o professor especifique exatamente o que deve ser feito na questão. Além disto, nomes de procedimentos, parâmetros de entrada, parâmetros de saída também devem ser especificados. Essas restrições são necessárias para que a ferramenta possa executar e avaliar as respostas submetidas.

3.3.2. Feedback

O ensino autodidata entre os mais diversos alunos está se tornando cada vez mais popular. Nesse cenário, uma ferramenta que avalie automaticamente e instantaneamente exercícios de programação em banco de dados retornando para o aluno uma avaliação com resultados é de fundamental importância. Quando os alunos sabem exatamente o que estava errado com suas respostas e onde seus programas falharam, eles podem usar as informações para aprender com seus erros.

Como o principal objetivo da ferramenta em questão não é analisar sintaticamente os procedimentos armazenados, o *feedback* exibe os casos de teste que foram utilizados para obter os resultados, além de exibir se os resultados estão em igualdade com a resolução ideal considerada, que foi o procedimento do professor ou instrutor. Ou seja, a ferramenta não exibe para o aluno os resultados do procedimento do professor com a

finalidade de que ele possa refazer seu procedimento, caso necessário, com o propósito de chegar ao resultado do professor ocultado pela ferramenta. A ferramenta ainda conta com um campo texto onde o professor pode comentar cada resultado do aluno, auxiliando assim na próxima submissão do aluno.

3.3.3. Os Casos de Teste

Na maioria dos sistemas que avaliam automaticamente programas computacionais, o programa é testado através da execução do programa sobre conjuntos de dados para obtenção dos resultados. O grande desafio com certeza é que todos os casos de teste possam cobrir o programa avaliado. Ou seja, a avaliação do programa depende do projeto dos casos de teste criados pela ferramenta para que todos os testes possam ser realizados no programa em questão.

No caso específico da nossa ferramenta, os casos de teste ficam a cargo do professor. Logo, o procedimento do aluno será testado apenas com os testes que o professor julga necessário, evitando que algum procedimento seja testado de forma equivocada. Em resumo, o sucesso ou não da avaliação e o seu *feedback* dependem diretamente do conjunto de casos de testes preparados pelo professor. A criação de casos de testes não é uma tarefa simples e pode consumir bastante tempo, mas a reutilização dos mesmos em diferentes cursos cria uma compensação a médio prazo.

3.3.4. Submissões Ilimitadas

A utilização de ferramentas de avaliações automáticas está evoluindo cada vez mais na formação acadêmica. Se uma ferramenta não possui a possibilidade do aluno refazer ou refletir sobre uma questão com resultados insatisfatórios, será difícil uma formação adequada para este aluno. Logo, é essencial que os alunos sejam autorizados a reenviar programas de melhoria para avaliação. Entretanto, quando um número ilimitado de submissões é permitido, os alunos não são forçados a refletir sobre suas soluções. Eles podem simplesmente usar a ferramenta de avaliação como uma ferramenta de teste, o que não é o objetivo. Logo, a facilidade para submissões ilimitadas deve ser configurável e monitorada, evitando, assim, interpretações erradas sobre a utilização da ferramenta e perfil dos alunos ou questões.

3.3.5. Suporte Adicional

No modelo de aprendizagem tradicional os instrutores fornecem conselhos de ajustes e apoiam os estudantes para interpretar o *feedback* automático do sistema. Sendo assim, a ferramenta fornece uma base a este suporte adicional proporcionado pelo professor. Como citado anteriormente, a ferramenta dispõe de um campo reservado a comentários que o professor pode descrever algum auxílio para o estudante.

4. PROJETO DA FERRAMENTA

Após a revisão bibliográfica, iniciou-se o projeto da ferramenta, no qual foram analisados e definidos quais seriam seus requisitos funcionais e não funcionais. Além disso, foram modelados o digrama de casos de uso e o projeto lógico de dados.

3.1. Requisitos funcionais e não funcionais

Os requisitos funcionais da ferramenta, resumidos no Quadro 1, são: 1) Manter Usuário. Esse requisito permite que usuários sejam cadastrados e que os cadastros possam ser alterados; 2) Autenticar Usuário. Toda e qualquer interação com a ferramenta só pode ser efetuada por usuários autenticados; 3) Manter Turmas. Esse requisito permite ao usuário com perfil de professor criar, alterar, buscar e remover turmas; 4) Manter Atividades. Esse requisito permite ao usuário com perfil de professor criar, alterar, buscar e remover atividades; 5) Manter Questões. Esse requisito permite ao usuário com perfil de professor criar, alterar, buscar e remover questões; 6) Matricular na Turma. Esse requisito permite ao usuário com perfil de aluno se matricular em qualquer turma sem restrição; 7) Solicitar Perfil de Professor. Esse requisito permite ao usuário com perfil de aluno solicitar um perfil de professor ao Administrador do sistema; 8) Ativar Perfil de Professor. Esse requisito permite que o administrador do sistema adicione um perfil de professor a um usuário com perfil de aluno; 9) Validar Questão. Esse requisito permite que o usuário com perfil de professor valide uma questão anteriormente criada; 10) Submeter Resposta Para Avaliação Automática. Esse requisito permite que o usuário com perfil de aluno submeta sua resposta para a avaliação automática do sistema; 11) Visualizar Histórico de Submissões. Esse requisito permite ao usuário visualizar seu histórico de submissões. 12) Visualizar Resultado de Submissão. Esse requisito permite ao usuário visualizar o resultado de alguma submissão específica.

Requisitos Funcionais
R1: Manter Usuário (Inserir, Alterar e Buscar)
R2: Efetuar Login
R3: Manter Turmas (Inserir, Alterar, Buscar e Remover)
R4: Manter Atividades (Inserir, Alterar, Buscar e Remover)
R5: Manter Questões (Inserir, Alterar, Buscar e Remover)
R6: Matricular na Turma
R7: Solicitar Perfil de Professor
R8: Ativar Perfil de Professor
R9: Validar Questão
R10: Submeter Resposta Para Avaliação Automática
R11: Visualizar Histórico de Submissões
R12: Visualizar Resultado de Submissão
R13: Comentar Resultado de Submissão
Requisitos Não-Funcionais
R12: O Sistema só deve permitir acesso através de login e senha.
R13: O Sistema deve ser desenvolvido utilizando a linguagem C#.
R14: O Sistema deve ser desenvolvido para Web.
R15: O Sistema deve ser executado em qualquer plataforma (Linux, Unix, Windows).
R16: O Sistema deve utilizar o Ambiente Visual Studio 2010 integrado ao Banco de Dados MySQL
R17: O Sistema deve executar seus scripts no Banco de Dados SQL SERVER

Quadro 1: *Requisitos funcionais e não funcionais da ferramenta.*

Além desses requisitos citados anteriormente, a ferramenta foi desenvolvida para Web, na linguagem C# e utiliza o MySQL como banco de dados. O sistema deve executar seus scripts utilizando o banco de dados SQL SERVER. O Sistema deve ser executado em qualquer plataforma e o seu acesso só deve ser permitido através de usuário e senha.

4.2. Casos de Uso

Esta seção descreve os casos de uso a nível de usuário exibindo diagramas que ajudam a entender os atores destinados para cada funcionalidade do sistema. Além disso, ainda nesta seção é descrito os casos de uso a nível de sistema, que auxiliam na compreensão de cada funcionalidade.

4.2.1. Casos de Uso – Nível de Usuário

Após a identificação e análise inicial dos requisitos essenciais para a ferramenta, foi modelado o diagrama de casos de uso, que é mostrado na Figura 2. Com ele, é possível verificar quais são as funções do sistema.



Figura 3: Diagrama de casos de uso da ferramenta

O caso de uso Efetuar Cadastro tem como objetivo realizar o cadastro de um usuário qualquer no sistema. Independentemente, se o usuário é aluno ou professor, ao realizar o cadastro, o usuário recebe o perfil de aluno. Este caso de uso está associado ao requisito R1.

O caso de uso Efetuar Login valida os dados do usuário, nome de usuário e senha, para permitir seu acesso ao sistema. Somente com a autenticação válida é possível ao usuário acessar as demais funcionalidades da ferramenta. Associa-se ao requisito R2.

O caso de uso Ativar Perfil de Professor tem o objetivo de definir quem são os professores da turma e conceder acesso as funcionalidades destinadas ao professor. Este caso de uso está associado ao ator Administrador, pois é ele quem tem acesso a esta funcionalidade. O caso de uso Ativar Perfil de Professor refere-se ao requisito R8.

O caso de uso manter Turmas refere-se ao requisito R3 e tem como objetivo realizar manutenções das turmas no sistema. Sejam elas, inserir, alterar, buscar e remover. Este caso de uso está associado ao ator Professor, pois é ele quem tem acesso a esta funcionalidade.

O caso de uso Manter Atividades refere-se ao requisito R4 e tem como objetivo realizar manutenções das atividades dentro de uma determinada turma no sistema. Este caso de uso está associado ao ator Professor.

O caso de uso Manter Questões refere-se ao requisito R5 e tem como objetivo realizar manutenções das questões dentro de uma atividade que se localiza-se em uma determinada turma no sistema. O ator Professor está associado a este caso, pois é ele quem tem acesso a esta funcionalidade.

Já o caso de uso Validar Questão refere-se ao requisito R9. Este caso tem o objetivo de realizar uma validação na questão criada, ou seja, se houver algum erro em algum script da criação da questão, este caso de uso realizará esta verificação. Este caso de uso está associado ao ator Professor.

O caso de uso Solicitar Perfil de Professor tem como objetivo realizar uma solicitação de ativação de perfil para o administrador do sistema. Este caso de uso está

associado ao requisito R7. O ator Aluno está associado a este caso de uso, pois é ele quem realiza esta ação e tem acesso a essa funcionalidade.

Matricular na Turma é o caso de uso que objetiva realizar a matrícula de um aluno em uma determinada turma. Este caso de uso está associado ao requisito R6. O ator Aluno está associado a este caso de uso, pois é ele quem realiza esta ação e tem acesso a essa funcionalidade.

O caso de uso Submeter Resposta Para Avaliação Automática é considerado um dos casos de usos principais do sistema, pois este tem o objetivo submeter a resposta do aluno para avaliação automática, logo o ator deste caso de uso é o Aluno. O requisito associado é o R10.

Visualizar Histórico de Submissões permite a visualização do histórico das respostas que foram submetidas para avaliação pelo aluno no sistema. Os atores desse caso de uso são o Aluno e o Professor. Este caso de uso está associado ao requisito R11.

Visualizar Resultado de Submissão permite a visualização do resultado de uma submissão específica submetida pelo aluno no sistema. Os atores desse caso de uso são, o Aluno e o Professor. Este caso de uso está associado ao requisito R12.

O caso de uso Comentar Resultado de Submissão tem como objetivo o registro de um comentário do professor sobre um resultado de uma submissão específica do aluno. O ator Professor está associado a este caso de uso, pois é ele quem realiza esta ação e tem acesso a essa funcionalidade. Este caso de uso está associado ao requisito R13.

4.2.2. Casos de Uso – Nível de Sistema

Nesta seção apresentaremos os casos de uso em nível de sistema. Nestes são apresentados o objetivo de algumas funções, os usuários indicados, pré-condição e pós-condição para realização do caso de uso. Além disso, os fluxos principais, alternativos e de exceção são detalhados para conhecimento das interações de cada funcionalidade do sistema.

4.2.2.1. Caso de Uso Efetuar Login

Quadro 2: *Caso de Uso Efetuar Login*

Objetivo	Efetuar login no sistema para usar funcionalidades disponíveis no sistema.	
Ator	Professores e Alunos	
Pré-Condição	O Ator precisa estar cadastrado no sistema	
Pós-Condição	Usuário deve estar autorizado para acessar as funcionalidades do sistema.	
Fluxo Principal		
Ação do Ator		Resposta do Sistema
1. Ator clica no link “Login” no canto superior direito da tela.		2. Sistema exibe tela para preenchimento dos dados do usuário.
3. Ator informa login e senha e clica em Entrar		4. Sistema verifica preenchimento e validade dos dados.
		5. Sistema redireciona para tela de Home do sistema e permite o acesso do usuário às funcionalidades disponíveis para cada perfil.
Fluxo Alternativo		
Passo 3: Ator clica em Retornar. Retorna ao Menu Principal.		
Fluxo Exceção		
Passo 4. Dados não válidos. Sistema exibe mensagem “Usuário/Senha incorretos” e exibe os campos que estão com problemas retornando ao passo 3.		
Passo 4: Campos vazios. Sistema exibe mensagem “O campo * é obrigatório”, retornando ao passo 3.		

4.2.2.2. Caso de Uso Manter Turmas

Seção: Criar Turma

Objetivo	Criar uma nova turma		
Ator	Professores		
Pré-Condição	O Ator precisa estar autenticado no sistema		
Pós-Condição	A turma deve ser inserida no sistema		
Fluxo Principal			
Ação do Ator		Resposta do Sistema	
1. Ator clica na aba Turmas.		2. Sistema exibe uma lista com turmas já cadastradas. Também exibe o link “Criar Turma”.	
3. Ator clica no link “Criar Turma”.		4. Sistema exibi o formulário de preenchimento com o botão “Criar” e o link para retornar para a lista de Turmas.	
5. Ator preenche o campo destinado a Descrição da Turma e clica em “Criar”.		6. Sistema valida os dados preenchidos.	
		7. Sistema redireciona para a tela de listas com a nova turma inserida.	
Fluxo Alternativo			
Passo 5: Ator clica no link “Retornar para Turmas”. Retorna a lista de Turmas cadastradas.			
Fluxo Exceção			
Passo 6: Campos vazios. Sistema exibe mensagem “O campo Descrição da Turma é obrigatório.”, retornando ao passo 4.			

Quadro 3: *Caso de Uso Manter Turmas – Criar Turma*

Seção: Editar Turma

Objetivo	Altera os dados da turma no sistema.		
Ator	Professor		
Pré-Condição	O Ator precisa estar autenticado no sistema		
Pós-Condição	Turma deve ter sido atualizada no sistema.		
Fluxo Principal			
Ação do Ator		Resposta do Sistema	
1. Ator clica na aba Turmas.		2. Sistema exibe uma lista com turmas já cadastradas. Também exibe o link “Criar Turma”.	
3. Ator clica no link “Editar” destinado a turma que ele deseja editar os dados.		4. Sistema exibe o formulário já preenchido com os dados originais, Também exibe o botão “Editar” e o link para retornar para a lista de Turmas.	
5. Ator altera os dados conforme ele deseja e clica no botão “Editar”.		6. Sistema valida os novos dados alterados.	
		7. Sistema redireciona para a lista de Turmas, contendo os dados atualizados da Turma editada.	
Fluxo Alternativo			
Passo 2: Ator clica em Retornar. Retorna ao Menu Principal.			
Passo 5: Ator clica em Retornar. Retorna ao passo 2.			
Fluxo Exceção			
Passo 5. Campos vazios. Sistema exibe mensagem “O campo Descrição da Turma é obrigatório.”, retornando ao passo 5.			

Quadro 4: *Caso de Uso Manter Turmas – Editar Turma*

Seção: Detalhar Turma

Objetivo	Visualizar os dados de uma determinada turma.
Ator	Professores
Pré-Condição	O Ator precisa estar autenticado no sistema
Fluxo Principal	
Ação do Ator	Resposta do Sistema
1. Ator clica na aba Turmas.	2. Sistema exibe uma lista com turmas já cadastradas. Também exibe o link “Criar Turma”.
3. O Ator clica no link “Detalhes” destinado a turma que ele deseja visualizar os dados.	4. Sistema exibe todos os dados da Turma selecionada. Também exibe o link para retornar a lista de turmas.
Fluxo Alternativo	
Passo 3: Ator clica em Retornar. Retorna ao Menu Principal.	

Quadro 5: *Caso de Uso Manter Turmas– Detalhar Turma***Seção: Remover Turma**

Objetivo	Remover uma determinada turma do sistema.
Ator	Professores
Pré-Condição	O Ator precisa estar autenticado no sistema
Pós-Condição	A turma deve ser removida do sistema.
Fluxo Principal	
Ação do Ator	Resposta do Sistema
1. Ator clica na aba Turmas no Menu.	2. Sistema exibe uma lista com turmas já cadastradas. Também exibe o link “Criar Turma”.
3. O Ator clica no link “Remover” destinado a turma que ele deseja remover do sistema.	4. Sistema exibe todos os dados da Turma selecionada. Também exibe o botão “Remover” e o link para retornar a lista de turmas.
5. Ator clica em Remover.	6. Sistema redireciona para lista de turmas, sem a turma removida.
Fluxo Alternativo	
Passo 3: Ator clica em Retornar. Retorna ao Menu Principal.	
Fluxo Exceção	
Passo 5: Turmas com Atividades. O sistema exibe a mensagem: “Esta turma não pode ser removida, pois existem atividades associadas.” Retornar ao passo 4.	

Quadro 6: *Caso de Uso Manter Turmas – Remover Turma*

4.2.2.3. Caso de Uso Matricular na Turma

Objetivo	Matricular o aluno na turma para usar funcionalidades disponíveis da turma.
Ator	Alunos
Pré-Condição	O Ator precisa estar autenticado no sistema
Pós-Condição	Usuário deve estar autorizado para acessar as funcionalidades da turma.
Fluxo Principal	
Ação do Ator	Resposta do Sistema
1. Ator clica na aba Turmas.	2. Sistema exibe uma lista com as turmas cadastradas.
3. Ator seleciona o link “Matricular” da turma destinada.	4. Sistema exibe mensagem de confirmação para se matricular na turma destinada.
5. Ator clica na opção de “OK”.	6. Sistema permanece na tela, porém com o aluno matriculado na turma e a opção “Visualizar” para a turma destinada.

Quadro 7: *Caso de Uso Matricular na Turma*

4.2.2.4. Caso de Uso Manter Questões

Seção: Criar Questão

Objetivo	Criar uma nova questão
Ator	Professores
Pré-Condição	O Ator precisa estar autenticado no sistema
Pós-Condição	A questão deve ser inserida no sistema
Fluxo Principal	
Ação do Ator	Resposta do Sistema
1. Ator clica na aba Turmas.	2. Sistema exibe uma lista com turmas já cadastradas. Também exibe o link “Criar Turma”.
3. Ator clica no link “Atividades” para a Turma que deseja criar uma questão.	4. Sistema exibe uma lista com as atividades da turma destinada. Também exibe o link “Criar Atividade” e o link “Retornar”.
5. Ator clica no link “Visualizar” para a Atividade que deseja criar uma questão.	6. Sistema exibe uma lista com as questões da atividade destinada. Também exibe o link “Criar Questão” e o link “Retornar”.
7. Ator clica no link “Criar Questão”.	8. Sistema exibe o formulário de criação de questão. Também exibe o botão “Criar Questão” e o link “Retornar”.
9. Ator preenche todos os campos necessários e/ou obrigatórios e clica no botão “Criar Questão”.	10. Sistema valida os dados preenchidos.
	11. Sistema redireciona para a tela de listas de questões com a nova questão inserida, com status de “Criada”.
Fluxo Alternativo	
Passo 3: Ator clica no link “Retornar”. Retorna a lista de Turmas cadastradas.	
Passo 5: Ator clica no link “Retornar”. Retorna a lista de Atividades cadastradas.	
Passo 7: Ator clica no link “Retornar”. Retorna a lista de Questões cadastradas.	
Fluxo Exceção	
Passo 9: Campos vazios. Sistema exibe mensagem “O campo * é obrigatório.”, retornando ao passo 4.	

Quadro 8: *Caso de Uso Manter Turmas – Criar Turma*

Seção: Editar Questão

Objetivo	Editar uma nova questão	
Ator	Professores	
Pré-Condição	O Ator precisa estar autenticado no sistema	
Pós-Condição	A questão deve ser inserida no sistema	
Fluxo Principal		
Ação do Ator		Resposta do Sistema
1. Ator clica na aba Turmas.		2. Sistema exibe uma lista com turmas já cadastradas. Também exibe o link “Criar Turma”.
3. Ator clica no link “Atividades” para a Turma que deseja criar uma questão.		4. Sistema exibe uma lista com as atividades da turma destinada. Também exibe o link “Criar Atividade” e o link “Retornar”.
5. Ator clica no link “Questões” para a Atividade que deseja criar uma questão.		6. Sistema exibe uma lista com as questões da atividade destinada. Também exibe o link “Criar Questão” e o link “Retornar”.
7. Ator clica no link “Editar” destinada a questão desejada.		8. Sistema exibe o formulário de edição de questão com os campos já preenchidos. Também exibe o botão “Criar Questão” e o link “Retornar”.
9. Ator altera todos os campos necessários e/ou obrigatórios e clica no botão “Editar”.		10. Sistema valida os dados preenchidos.
		11. Sistema redireciona para a tela de listas de questões com a nova questão editada, com status de “Criada”.
Fluxo Alternativo		
Passo 3: Ator clica no link “Retornar”. Retorna a lista de Turmas cadastradas.		
Passo 5: Ator clica no link “Retornar”. Retorna a lista de Atividades cadastradas.		
Passo 7: Ator clica no link “Retornar”. Retorna a lista de Questões cadastradas.		
Fluxo Exceção		
Passo 7: Campos vazios. Sistema exibe mensagem “O campo * é obrigatório.”, retornando ao passo 4.		

Quadro 9: Caso de Uso Manter Questões – Editar Questão

Seção: Detalhar Questão

Objetivo	Visualizar dados de uma determinada questão no sistema.
Ator	Professores
Pré-Condição	O Ator precisa estar autenticado no sistema
Fluxo Principal	
Ação do Ator	Resposta do Sistema
1. Ator clica na aba Turmas.	2. Sistema exibe uma lista com turmas já cadastradas. Também exibe o link “Criar Turma”.
3. Ator clica no link “Atividades” para a Turma que deseja criar uma questão.	4. Sistema exibe uma lista com as atividades da turma destinada. Também exibe o link “Criar Atividade” e do link “Retornar”.
5. Ator clica no link “Questões” para a Atividade que deseja criar uma questão.	6. Sistema exibe uma lista com as questões da atividade destinada. Também exibe o link “Criar Questão” e o link “Retornar”.
7. Ator clica no link “Detalhar” destinada a questão desejada.	8. Sistema exibe todos os campos relacionados a questão desejada já preenchidos. Também exibe o link “Retornar”.
Fluxo Alternativo	
Passo 3: Ator clica no link “Retornar”. Retorna a lista de Turmas cadastradas.	
Passo 5: Ator clica no link “Retornar”. Retorna a lista de Atividades cadastradas.	
Passo 7: Ator clica no link “Retornar”. Retorna a lista de Questões cadastradas.	

Quadro 10: *Caso de Uso Manter Questões– Detalhar Questão*

Seção: Remover Questão

Objetivo	Remover uma determinada questão do sistema.		
Ator	Professores		
Pré-Condição	O Ator precisa estar autenticado no sistema		
Fluxo Principal			
Ação do Ator		Resposta do Sistema	
1. Ator clica na aba Turmas.		2. Sistema exibe uma lista com turmas já cadastradas. Também exibe o link “Criar Turma”.	
3. Ator clica no link “Atividades” para a Turma que deseja criar uma questão.		4. Sistema exibe uma lista com as atividades da turma destinada. Também exibe o link “Criar Atividade” e do link “Retornar”.	
5. Ator clica no link “Questões” para a Atividade que deseja criar uma questão.		6. Sistema exibe uma lista com as questões da atividade destinada. Também exibe o link “Criar Questão” e do link “Retornar”.	
7. Ator clica no link “Remover” destinada a questão desejada.		8. Sistema exibe uma lista com as questões da atividade destinada sem a questão removida. Também exibe o link “Criar Questão” e do link “Retornar”.	
Fluxo Alternativo			
Passo 3: Ator clica no link “Retornar”. Retorna a lista de Turmas cadastradas.			
Passo 5: Ator clica no link “Retornar”. Retorna a lista de Atividades cadastradas.			
Passo 7: Ator clica no link “Retornar”. Retorna a lista de Questões cadastradas.			
Fluxo Exceção			
Passo 7: Erro inesperado. Caso ocorra algum erro inesperado, será exibida a mensagem “Não foi possível remover esta turma, favor contatar o Administrador do Sistema.”.			

Quadro 11: *Caso de Uso Manter Questões – Remover Questão*

4.2.2.5. Caso de Uso Solicitar Perfil de Professor

Objetivo	Solicitar um Perfil de Professor para o Administrador.		
Ator	Alunos		
Pré-Condição	O Ator precisa estar cadastrado no sistema		
Pós-Condição	Usuário deve ter uma solicitação pendente para avaliação do administrador.		
Fluxo Principal			
Ação do Ator		Resposta do Sistema	
1. Ator clica na aba de Solicitar Perfil de Professor no Menu.		2. Sistema exibe lista com as solicitações realizadas pelo usuário para o administrador	
3. Ator seleciona o link “Solicitar” e confirma as mensagens de confirmação que aparecerão.		4. Sistema envia uma solicitação para o administrador do sistema.	
Fluxo Alternativo			
Passo 3: Ator não confirma as mensagens de confirmação. Retorna ao passo 2.			

Quadro 12: *Caso de Uso Solicitada Perfil de Professor*

4.2.2.6. Caso de Uso Validar Questão

Objetivo	Validar uma questão criada.
Ator	Professores
Pré-Condição	O Ator precisa estar cadastrado no sistema
Pós-Condição	A questão deve ser validada e ficar visível para o aluno.
Fluxo Principal	
Ação do Ator	Resposta do Sistema
1. Ator clica na aba de Turmas no Menu.	2. Sistema exibe uma lista com turmas já cadastradas. Também exibe o link “Criar Turma”.
3. Ator clica no link “Atividades” para a Turma que deseja validar uma questão.	4. Sistema exibe uma lista com as atividades da turma destinada. Também exibe o link “Criar Atividade” e do link “Retornar”.
5. Ator clica no link “Questões” para a Atividade que deseja validar uma questão.	6. Sistema exibe uma lista com as questões da atividade destinada. Também exibe o link “Criar Questão” e do link “Retornar”.
7. Ator clica no link “Validar”.	8. Sistema exibe um formulário apenas para visualização dos campos com o link para “Ir para Editar Questão” e “Retornar”, Também exibe o botão “Validar” ao final.
9. Ator clica no botão “Validar”.	10. Sistema valida os dados do formulário.
	11. Sistema redireciona para lista de questões.
Fluxo Alternativo	
Passo 3: Ator clica no link “Retornar”. Retorna a lista de Turmas cadastradas.	
Passo 5: Ator clica no link “Retornar”. Retorna a lista de Atividades cadastradas.	
Passo 7: Ator clica no link “Retornar”. Retorna a lista de Questões cadastradas.	
Fluxo Exceção	
Passo 9. Dados não válidos. Sistema exibe a mensagem de erro destinada ao erro ocasionado e retorna ao passo 8.	

Quadro 13: *Caso de Uso Validar Questão*

4.2.2.7. Caso de Uso Ativar Perfil de Professor

Objetivo	Ativar um perfil de professor para um aluno.		
Ator	Administrador		
Pré-Condição	O Ator precisa estar cadastrado no sistema		
Pós-Condição	Usuário destinado com perfil de aluno deve ter um novo perfil de professor no sistema.		
Fluxo Principal			
Ação do Ator		Resposta do Sistema	
1. Ator clica na aba de Ativar Perfil de Professor no Menu.		2. Sistema exibe lista com todas as solicitações dos alunos.	
3. Ator clicar no link “Ativar” destinado ao usuário com perfil de aluno que deseja ativar.		4. Sistema exibe lista com todas as solicitações, porém, com o aluno destinado o status de Ativo.	
Fluxo Exceção			
Passo 3. Professor já ativo. Sistema exibe mensagem “Este professor já está ativo” e retorna ao passo 2.			

Quadro 14: *Caso de Uso Ativar Perfil de Professor*

4.2.2.8. Caso de Uso Visualizar Histórico de Submissões

Seção: Visão Professor

Objetivo	Visualizar o histórico de submissões realizadas por um determinado aluno.	
Ator	Professores	
Pré-Condição	O Ator precisa estar autenticado no sistema	
Fluxo Principal		
Ação do Ator		Resposta do Sistema
1. Ator clica na aba Turmas.		2. Sistema exibe uma lista com turmas já cadastradas. Com link “Matricular” ou “Visualizar” a depender da matrícula do aluno na turma.
3. Ator clica no link “Visualizar” para a Turma que deseja visualizar uma questão.		4. Sistema exibe uma lista com as atividades da turma destinada. Também exibe o link “Visualizar” e do link “Retornar”.
5. Ator clica no link “Visualizar” para a Atividade que deseja visualizar uma questão.		6. Sistema exibe uma lista com as questões da atividade destinada. Também exibe o link “Visualizar Resposta” e o link “Retornar”.
7. Ator clica no link “Visualizar Resposta” referente à questão que deseja.		8. Sistema exibe uma lista de todas as submissões realizadas pelo aluno, além das opções “Script Resposta”, “Visualizar Resultado”, “Comentar Resultado”.
Fluxo Alternativo		
Passo 3: Ator clica no link “Retornar”. Retorna a lista de Turmas cadastradas.		
Passo 5: Ator clica no link “Retornar”. Retorna a lista de Atividades cadastradas.		
Passo 7: Ator clica no link “Retornar”. Retorna a lista de Questões cadastradas.		

Quadro 15: *Caso de Uso Visualizar Histórico de Submissões - Professor*

Seção: Visão Aluno

Objetivo	Visualizar o histórico de submissões realizadas por um determinado aluno.		
Ator	Alunos		
Pré-Condição	O Ator precisa estar autenticado no sistema		
Fluxo Principal			
Ação do Ator		Resposta do Sistema	
1. Ator clica na aba Turmas.		2. Sistema exibe uma lista com turmas já cadastradas. Também exibe o link “Criar Turma”.	
3. Ator clica no link “Atividades” para a Turma que deseja criar uma questão.		4. Sistema exibe uma lista com as atividades da turma destinada. Também exibe o link “Criar Atividade” e do link “Retornar”.	
5. Ator clica no link “Questões” para a Atividade que deseja criar uma questão.		6. Sistema exibe uma lista com as questões da atividade destinada. Também exibe o link “Criar Questão” e do link “Retornar”.	
7. Ator clica no link “Respostas” referente à questão que deseja.		8. Sistema exibe uma lista de todos os alunos que realizaram uma submissão para aa questão.	
9. Ator clica no link “Visualizar Respostas” do aluno que deseja.		10. Sistema exibe lista contando todas as submissões que o aluno realizou, com as opções “Script Resposta”, “Editar Script” e Visualizar Resultado.	
Fluxo Alternativo			
Passo 3: Ator clica no link “Retornar”. Retorna a lista de Turmas cadastradas.			
Passo 5: Ator clica no link “Retornar”. Retorna a lista de Atividades cadastradas.			
Passo 7: Ator clica no link “Retornar”. Retorna a lista de Questões cadastradas.			

Quadro 16: Caso de Uso Visualizar Histórico de Submissões - Aluno

4.2.2.9. Caso de Uso Visualizar Resultado de Submissão

Seção: Visão Professor

Objetivo	Visualizar um resultado específico de uma submissão realizada por um determinado aluno.	
Ator	Professores	
Pré-Condição	O Ator precisa estar autenticado no sistema	
Fluxo Principal		
Ação do Ator		Resposta do Sistema
1. Ator clica na aba Turmas.		2. Sistema exibe uma lista com turmas já cadastradas. Com link “Matricular” ou “Visualizar” a depender da matrícula do aluno na turma.
3. Ator clica no link “Visualizar” para a Turma que deseja visualizar uma questão.		4. Sistema exibe uma lista com as atividades da turma destinada. Também exibe o link “Visualizar” e do link “Retornar”.
5. Ator clica no link “Visualizar” para a Atividade que deseja visualizar uma questão desta atividade.		6. Sistema exibe uma lista com as questões da atividade destinada. Também exibe o link “Visualizar Resposta” e do link “Retornar”.
7. Ator clica no link “Visualizar Resposta” referente à questão que deseja.		8. Sistema exibe uma lista de todas as submissões realizadas pelo aluno, além das opções “Script Resposta”, “Visualizar Resultado”, “Comentar Resultado”.
9. Ator clica no link “Visualizar Resultado” da submissão específica desejada.		10. Sistema exibe o formulário inalterado da submissão com o resultado. Além da opção de “Retornar”.
Fluxo Alternativo		
Passo 3: Ator clica no link “Retornar”. Retorna a lista de Turmas cadastradas.		
Passo 5: Ator clica no link “Retornar”. Retorna a lista de Atividades cadastradas.		
Passo 7: Ator clica no link “Retornar”. Retorna a lista de Questões cadastradas.		

Quadro 17: *Caso de Uso Visualizar Resultado de Submissões - Professor*

Seção: Visão Aluno

Objetivo	Visualizar o histórico de submissões realizadas por um determinado aluno.	
Ator	Alunos	
Pré-Condição	O Ator precisa estar autenticado no sistema	
Fluxo Principal		
Ação do Ator		Resposta do Sistema
1. Ator clica na aba Turmas.		2. Sistema exibe uma lista com turmas já cadastradas. Também exibe o link “Criar Turma”.
3. Ator clica no link “Atividades” para a Turma que deseja criar uma questão.		4. Sistema exibe uma lista com as atividades da turma destinada. Também exibe o link “Criar Atividade” e do link “Retornar”.
5. Ator clica no link “Questões” para a Atividade que deseja criar uma questão.		6. Sistema exibe uma lista com as questões da atividade destinada. Também exibe o link “Criar Questão” e do link “Retornar”.
7. Ator clica no link “Respostas” referente à questão que deseja.		8. Sistema exibe uma lista de todos os alunos que realizaram uma submissão da questão destinada.
9. Ator clica no link “Visualizar Respostas” do aluno que deseja.		10. Sistema exibe lista contando todas as submissões que o aluno destinado realizou, com as opções “Script Resposta”, “Editar Script” e Visualizar Resultado.
11. Ator clica no link “Visualizar Resultado” da submissão específica desejada.		12. Sistema exibe o formulário inalterado da submissão com o resultado, também a opção de “Retornar”.
Fluxo Alternativo		
Passo 3: Ator clica no link “Retornar”. Retorna a lista de Turmas cadastradas.		
Passo 5: Ator clica no link “Retornar”. Retorna a lista de Atividades cadastradas.		
Passo 7: Ator clica no link “Retornar”. Retorna a lista de Questões cadastradas.		

Quadro 18: *Caso de Uso Visualizar Resultado de Submissões - Aluno*

4.2.2.10. Caso de Uso Submeter Resposta para Avaliação Automática

Objetivo	Realizar uma submissão de uma resposta para avaliação automática do sistema.	
Ator	Alunos	
Pré-Condição	O Ator precisa estar autenticado no sistema	
Pós-Condição	Sistema deve ter gerado um resultado para esta submissão.	
Fluxo Principal		
Ação do Ator	Resposta do Sistema	
1. Ator clica na aba Responder Questão.	2. Sistema exibe três combos selecionáveis de Turma, Atividades e Questão.	
3. Ator seleciona a Turma que deseja responder uma questão.	4. Sistema mantém a opção selecionada pelo ator no combo de Turma e carrega o combo de Atividade com as atividades desta turma.	
5. Ator seleciona a Atividade que deseja responder uma questão.	6. Sistema mantém as opções selecionadas pelo ator no combo de Turma e Atividade, além de carregar o combo de Questão com as questões válidas desta atividade.	
7. Ator seleciona a Questão que deseja responder.	8. Sistema mantém as opções selecionadas e exibe um novo campo inalterado com a descrição da questão, também exibe o botão “Responder”.	
9. Ator clica no botão “Responder”.	10. Sistema redireciona para tela de Responder Questão e exibe o formulário para submissão da resposta. Também exibe o botão de “Submeter Resposta para Avaliação Automática”.	
Fluxo Alternativo		
Passo 3: Ator seleciona outra Turma. Os combos são reinicializados e a opção da turma é mantida.		
Passo 5: Ator seleciona outra Atividade. Os combos são reinicializados e as opções da turma e da atividade são mantidas.		

Quadro 19: *Caso de Uso Submeter Resposta para Avaliação Automática*

4.2.2.11. Caso de Uso Comentar Resultado de Submissão

Objetivo	Comentar um resultado específico de uma submissão realizada por um determinado aluno.	
Ator	Professores	
Pré-Condição	O Ator precisa estar autenticado no sistema	
Fluxo Principal		
Ação do Ator	Resposta do Sistema	
1. Ator clica na aba Turmas.	2. Sistema exibe uma lista com turmas já cadastradas. Com link “Matricular” ou “Visualizar” a depender da matrícula do aluno na turma.	
3. Ator clica no link “Visualizar” para a Turma que deseja visualizar uma questão.	4. Sistema exibe uma lista com as atividades da turma destinada. Também exibe os links “Visualizar” e “Retornar”.	
5. Ator clica no link “Visualizar” para a Atividade que deseja visualizar uma questão desta atividade.	6. Sistema exibe uma lista com as questões da atividade destinada. Também exibe o link “Visualizar Resposta” e do link “Retornar”.	
7. Ator clica no link “Visualizar Resposta” referente à questão que deseja.	8. Sistema exibe uma lista de todas as submissões realizadas pelo aluno. Além das opções “Script Resposta”, “Visualizar Resultado”, “Comentar Resultado”.	
9. Ator clica no link “Comentar Resultado” da submissão específica desejada.	10. Sistema exibe o formulário inalterado da submissão com o resultado e o campo Comentar Resultado, também a opção de “Retornar” e o botão “Comentar”.	
11. Ator preenche o campo Comentar Resultado e clica no botão “Comentar”.	12. Sistema redireciona para tela de lista de submissões.	
Fluxo Alternativo		
Passo 3: Ator clica no link “Retornar”. Retorna a lista de Turmas cadastradas.		
Passo 5: Ator clica no link “Retornar”. Retorna a lista de Atividades cadastradas.		
Passo 7: Ator clica no link “Retornar”. Retorna a lista de Questões cadastradas.		

Quadro 20: Caso de Uso Comentar Resultado de Submissões - Professor

4.3. Fluxo de Sequência de Atividades

Em um sistema é fundamental ter o conhecimento do fluxo das atividades que serão desempenhadas por ele. O principal objetivo de haver fluxos de sequências de atividades especificadas é conhecer o fluxo correto de funcionalidades do sistema. O fluxo de atividades é desenhado pela sequência das atividades de um processo de negócio. Um diagrama de atividade é essencialmente um gráfico de fluxo, mostrando o fluxo de controle de uma atividade para outra.

4.3.1. Fluxo de Efetuar Login

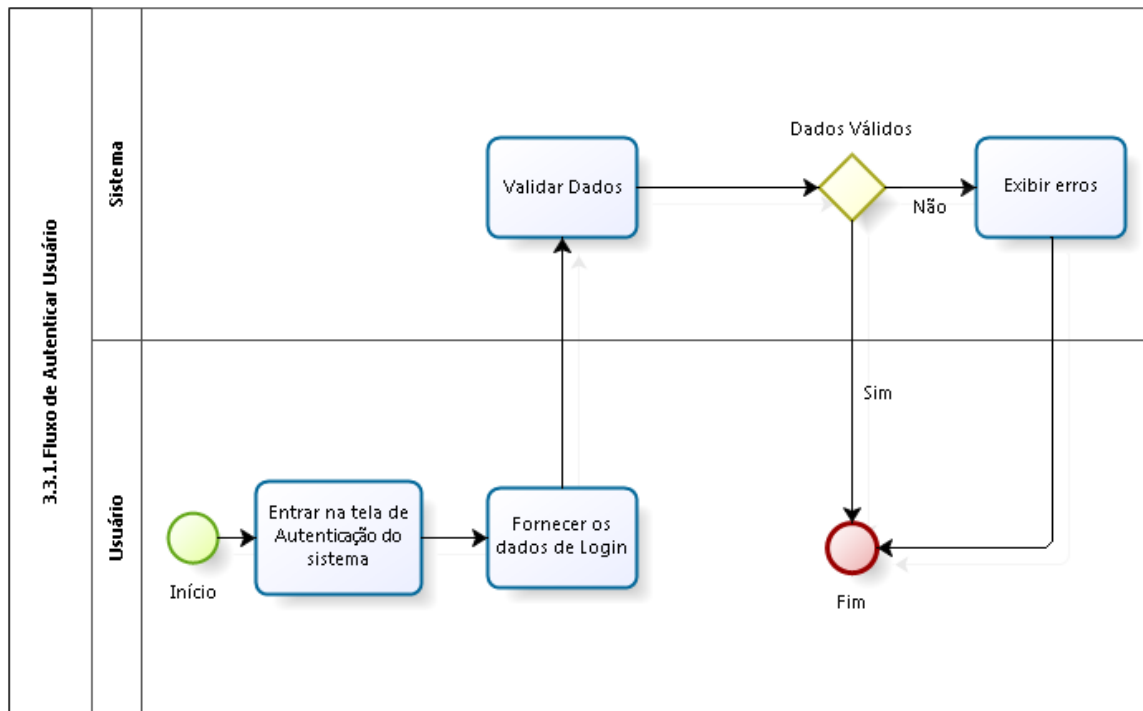


Figura 4: Fluxo de Efetuar Login

Este fluxo se inicia com o usuário entrando no sistema para que realize a autenticação no sistema. O usuário deve fornecer seus dados de usuário e senha. Após isso, o sistema deve validar os dados. Caso os dados sejam válidos, o acesso ao sistema deve ser liberado e o fluxo é encerrado. Caso contrário, o sistema deve exibir os erros em tela e o fluxo também é encerrado.

4.3.2. Fluxo de Criar Questões

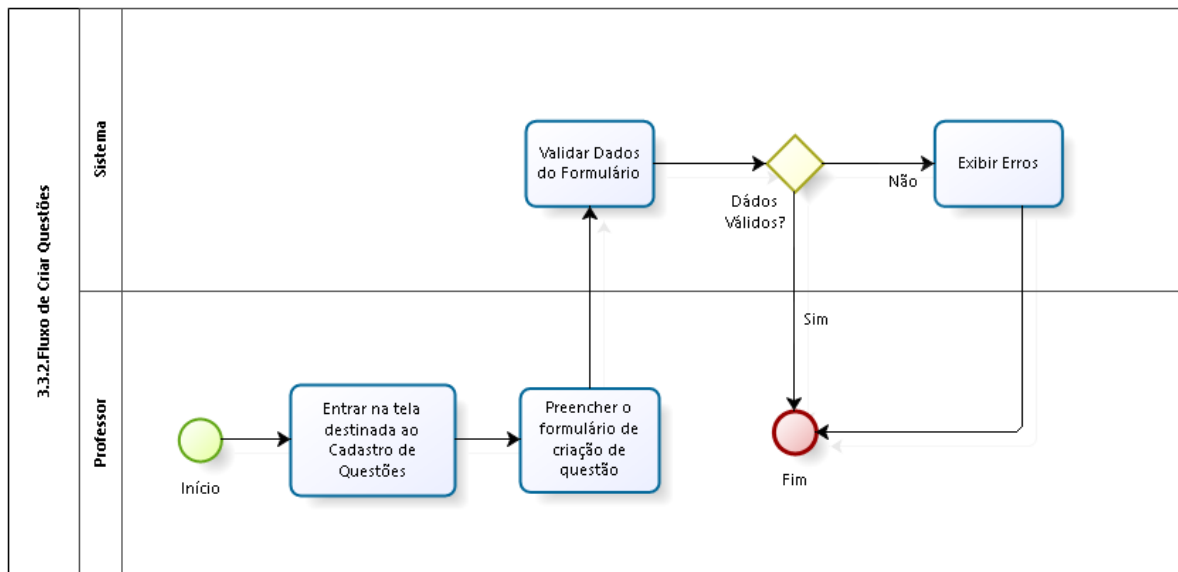


Figura 5: Fluxo de Criar Questão

O fluxo de criar questão se inicia com o professor entrando no sistema e se encaminhando para tela destinada ao cadastro das questões. O professor deve preencher o formulário de criação de questão, não esquecendo de preencher todos os seus campos obrigatórios. Após isso, o sistema deve validar os dados preenchidos pelo professor. Caso os dados sejam válidos, a questão é criada com sucesso e o fluxo é encerrado. Caso contrário, o sistema deve exibir os erros em tela e o fluxo também é encerrado.

4.3.3. Fluxo de Validar Questão

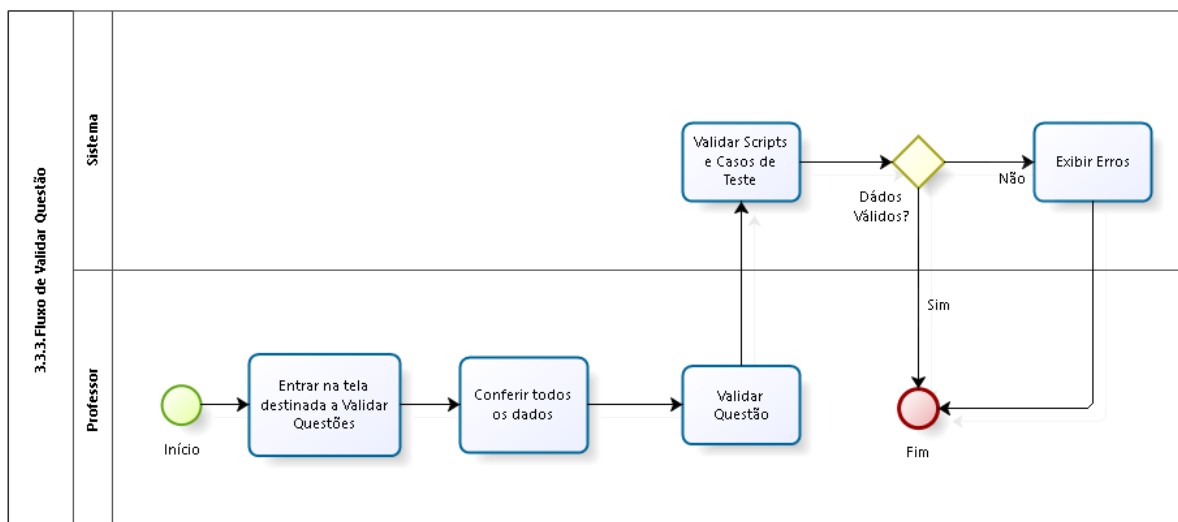


Figura 6: Fluxo de Validar Questão

Para que o fluxo de validar questão seja válido, a questão já deve estar cadastrada no sistema seguindo o fluxo anterior de criar questão, como exibido na figura 5.

O fluxo de validar a questão se inicia com o professor entrando no sistema e se encaminhando para tela destinada a validação das questões. Assim, o professor deve conferir todos os dados que foram preenchidos no fluxo de criar questão. Depois, o professor deve validar a questão. Após isso, o sistema deve validar os Scripts e Casos de Teste. Caso os Scripts e Casos de Teste sejam válidos, a questão é validada com sucesso, aparecerá na visão do aluno, e o fluxo é encerrado. Caso contrário, o sistema deve exibir os erros em tela e o fluxo também é encerrado.

4.3.4. Fluxo Ativação do Perfil do Professor

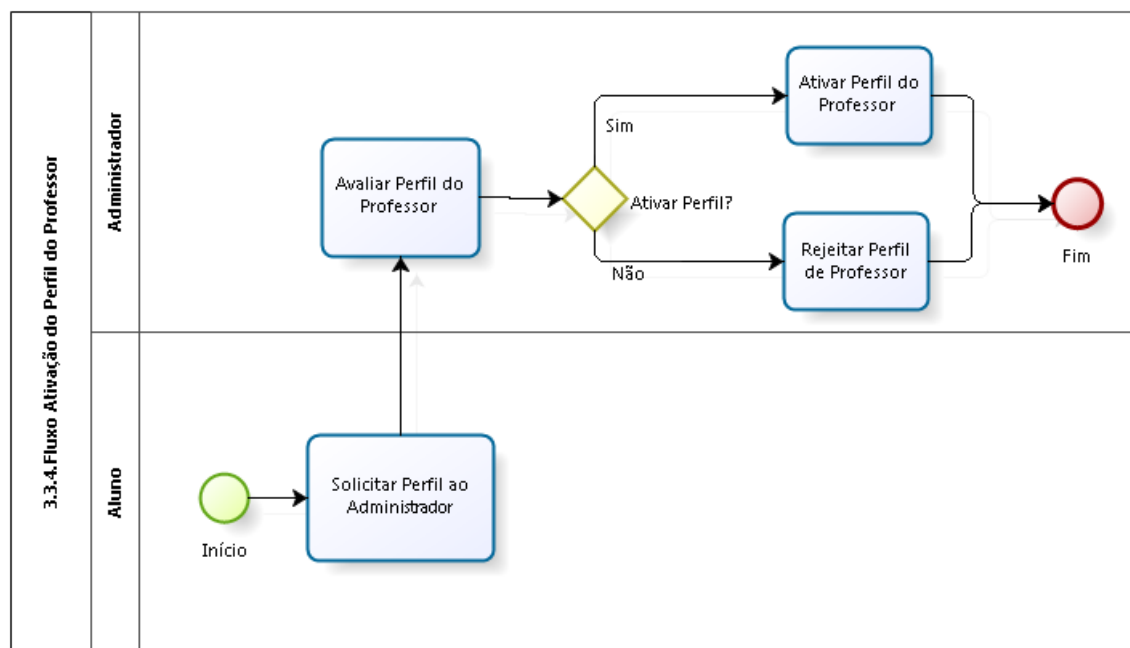


Figura 7: Fluxo de Ativar Perfil de Professor

O fluxo de ativar perfil de professor no sistema se inicia com o aluno entrando no sistema e se encaminhando para tela destinada a solicitação do novo perfil de professor ao Administrador do sistema. Assim, o administrador avalia o usuário que está solicitando o novo perfil de professor. Depois o administrador tem dois caminhos que é ativar ou rejeitar a solicitação de ativação do novo perfil do usuário solicitante e o fluxo é encerrado.

4.3.5. Fluxo de Submissão de Respostas

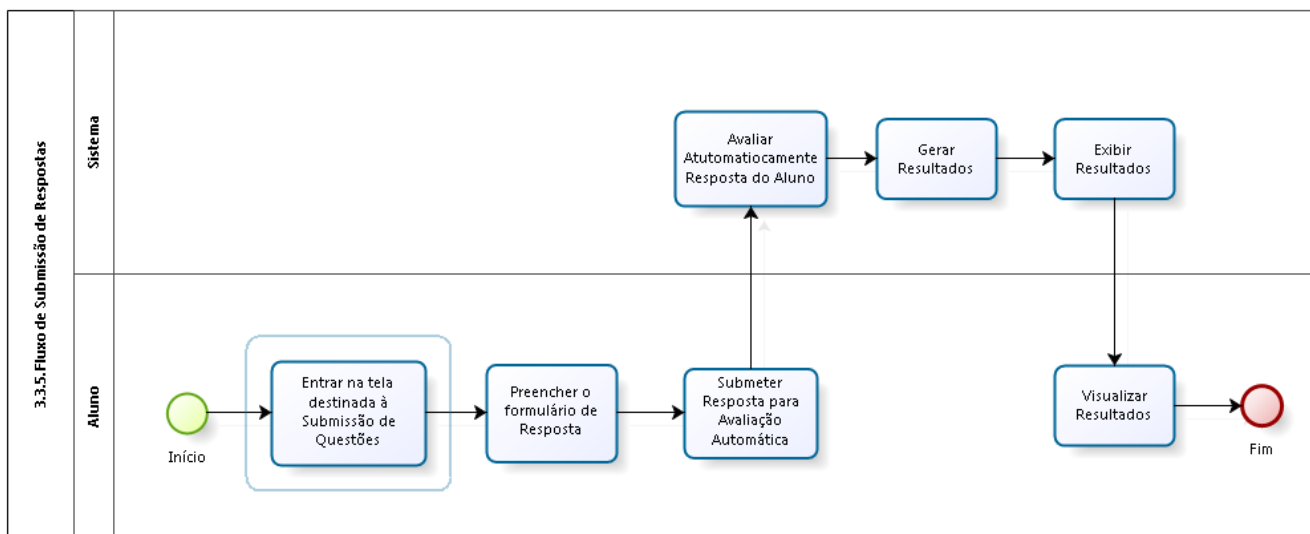


Figura 8: Fluxo de Submissão de Respostas

Este fluxo se inicia com o aluno entrando no sistema e se encaminhando para tela destinada à submissão de respostas para uma determinada questão. O aluno preenche o formulário de respostas, depois submete para avaliação automática do sistema. Assim, o sistema vai avaliar automaticamente a resposta do aluno gerando resultados e os exibindo. Por último o aluno visualiza os resultados obtidos pelo sistema e o fluxo é encerrado.

4.3.6. Fluxo de Comentar um Resultado

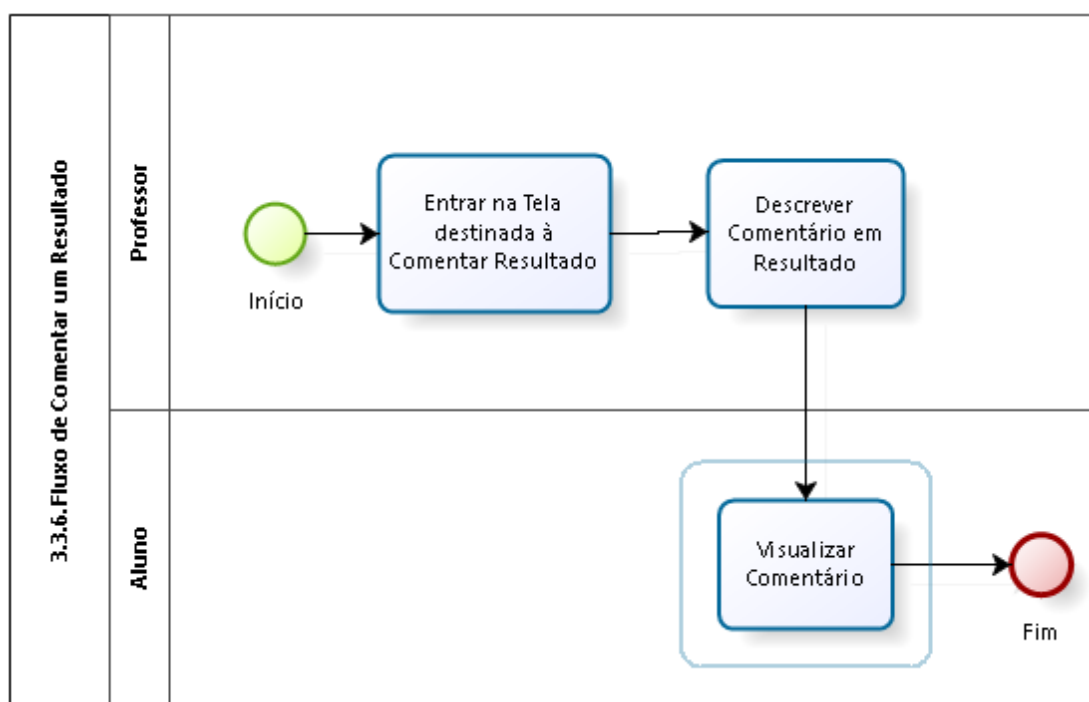


Figura 9: Fluxo de Comentar Resultado

Para que o fluxo de comentar um resultado seja válido, o resultado já deve ter sido gerado pelo sistema seguindo o fluxo anterior através do perfil de aluno para submeter resposta para avaliação.

O fluxo de comentar resultado se inicia com o professor entrando no sistema e se encaminhando para tela destinada à comentar um resultado para uma determinada resposta. O professor preenche o campo destinado ao seu comentário. Assim, o aluno visualiza o comentário do professor e o fluxo se encerra.

4.4. Modelagem de Dados

Após a criação dos fluxos de sequência de atividades, foi modelado o projeto lógico de dados do Sistema. A Figura 9 mostra o esquema de dados da ferramenta. Nesta seção detalharemos as tabelas do esquema.

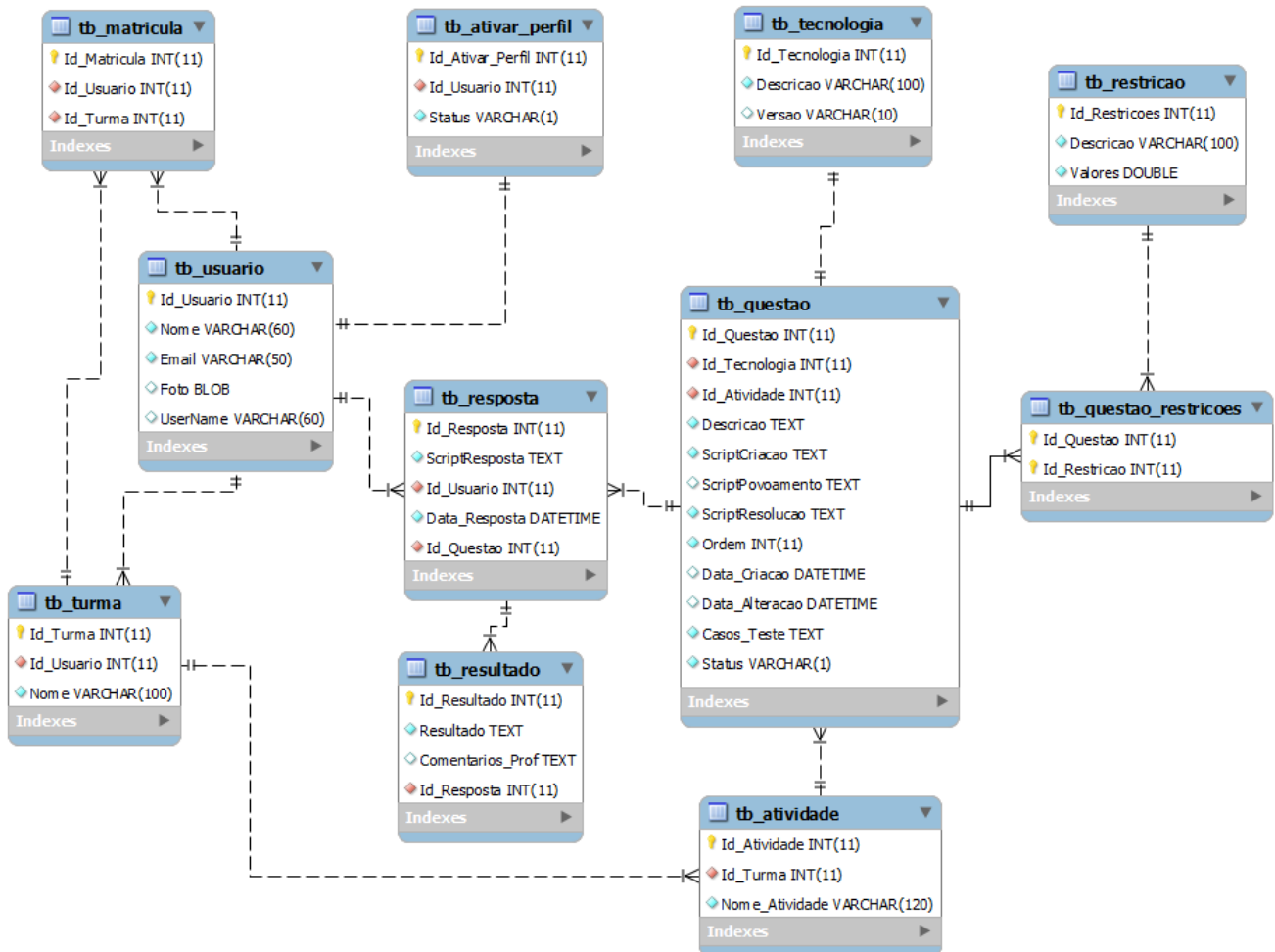


Figura 10: Esquema de dados da ferramenta

4.4.1. Componentes do modelo de dados

Tabela tb_turma: Todas as informações sobre as turmas são persistidas na tabela tb_turma. Esta tabela é necessária, pois irá armazenar todas as turmas que os professores criarem. Logo, esta turma tem a obrigatoriedade de armazenar um Id_Usuario, atributo que compõe uma restrição de integridade referencial, e um Nome

para a turma. A tabela `tb_turma` ainda possui uma ligação N-1 com a tabela `tb_usuario`, isto se deve pelo fato de um usuário poder criar N turmas. A tabela tem uma restrição de chave primária composta pelo atributo `Id_Turma` que é um *surrogate key* nesta tabela.

Tabela `tb_atividade`: A tabela `tb_atividade` foi criada para o armazenamento das atividades de uma determinada turma. Assim, é possível perceber que há uma ligação N-1 com a tabela `tb_turmas`. Logo existe um atributo `Id_Turma` que é uma restrição de integridade referencial. Nesta tabela ainda podemos observar os seguintes atributos: `Id_Atividade`, que é um atributo chave *surrogate key*, logo obrigatório; E o `Nome_Atividade`, que se refere a descrição da atividade.

Tabela `tb_questao`: A tabela `tb_questao` foi criada para o armazenamento das questões de uma determinada atividade. Assim, é possível perceber que há uma ligação N-1 com a tabela `tb_atividade`, logo existe um atributo `Id_Atividade` que é uma restrição de integridade referencial. Esta tabela ainda possui ligações com as tabelas de `tb_questao_restricoes`, `tb_tecnologia`, e `tb_resposta`. Podemos observar ainda, os seguintes atributos: `Id_Questao`, que é um atributo chave *surrogate key*, logo obrigatório; Adicionalmente, temos outros parâmetros nesta tabela: `Descricao`, `ScriptCriacao`, `ScriptPovoamento`, `ScriptResolucao`, `Ordem`, `Data_Criacao`, `Data_Alteracao`, `Casos_Teste` e `Status`.

Tabela `tb_resposta`: A tabela `tb_resposta` foi criada para o armazenamento das respostas de uma determinada questão. A `tb_resposta` tem uma ligação importante N-1 para a `tb_resultado` e recebe uma ligação da tabela de questões, como pode ser visualizado na figura 10. Nesta tabela ainda podemos observar os seguintes atributos: `Id_Resposta`, que é um atributo chave *surrogate key*, logo obrigatório; `ScripResposta`, `Data_Resposta`, atributos obrigatórios. Além dos atributos `Id_Usuario`, que é uma restrição de integridade referencial da tabela `tb_usuario` e `Id_Questao`, que é uma restrição de integridade referencial da tabela `tb_questao`.

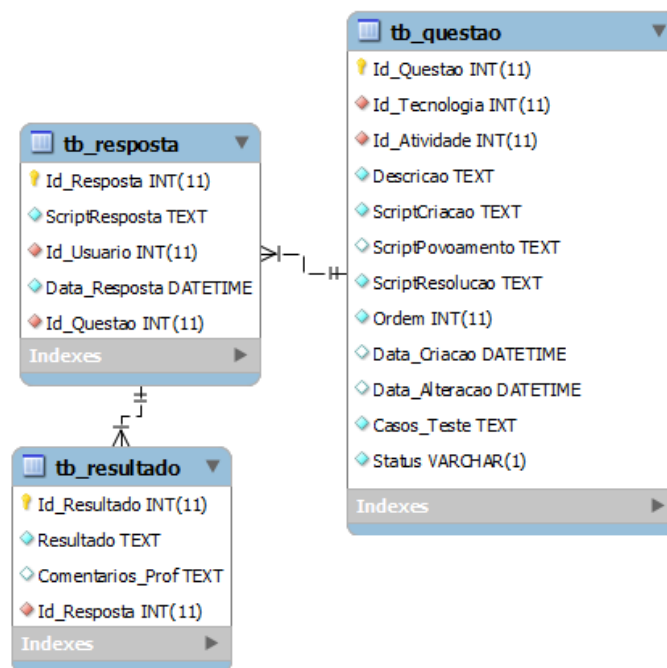


Figura 11: Esquema de dados da ferramenta – “Resposta”

Tabela tb_resultado: A tabela `tb_resultado` foi criada para o armazenamento dos resultados de uma determinada resposta. Nesta tabela é possível encontrar os seguintes atributos: `Id_Resultado`, que é um atributo chave *surrogate key*, logo obrigatório. `Resultado`, que é o atributo onde será armazenado os resultados das respostas submetidas pelos alunos, logo é obrigatório o preenchimento deste atributo. Além disso, é possível observar o atributo `Comentarios_Prof`, que é destinado aos comentários do professor para determinado resultado do aluno. Por último, temos o atributo `Id_Resposta` que é uma restrição de integridade referencial da tabela `tb_resposta`.

Tabela tb_usuario: A tabela `tb_usuario` foi criada para o armazenamento dos usuários que se cadastrem no sistema. Esta tabela há ligações importantes que pode ser visualizado na figura 11, como: Uma ligação 1-N com a tabela `tb_matricula`, pois cada usuário pode se matricular em diversas turmas. Outra ligação é a tabela `tb_ativar_perfil` 1-1, pois cada usuário pode ter mais um perfil de professor, além do seu de usuário (aluno). Ainda outras duas ligações 1-N podem ser observadas desta tabela, são as `tb_matricula` e `tb_turma`.

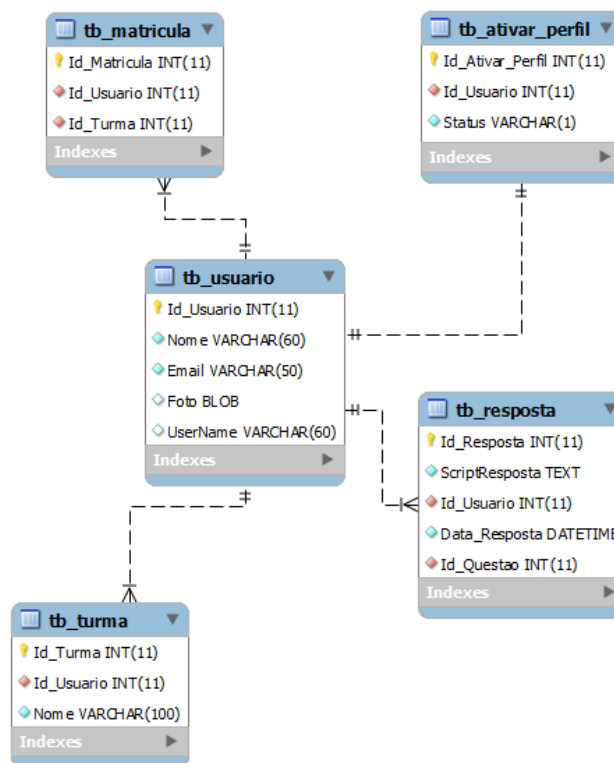


Figura 12: Esquema de dados da ferramenta – “Usuário”

Tabela tb_matricula: A tabela tb_matricula foi criada para o armazenamento das matrículas dos usuários. Esta tabela é resultado de uma associação N-N entre as tabelas tb_usuario e tb_turma. A tb_matricula tem como atributos o Id_Matricula que é um atributo chave *surrogate key*, logo obrigatório. E as chaves estrangeiras Id_Usuario e Id_Turma que também são obrigatórias.

Tabela tb_ativar_perfil: A tabela tb_ativar_perfil foi criada para o armazenamento dos perfis de professor dos usuários destinados a este perfil. Nesta tabela existe uma ligação 1-1 com a tb_usuario. Esta tabela tem o atributo Id_Ativar_Perfil que é um atributo chave *surrogate key*, logo obrigatório. Ainda há também uma restrição de integridade referencial Id_Usuario provido da tabela tb_usuario, além do Status que é obrigatório.

Tabela tb_tecnologia: A tabela tb_tecnologia foi criada para o armazenamento da tecnologia (MYSQL, SQL SERVER etc.) utilizada para responder a questão. Nesta tabela existe uma ligação 1-1 com a tb_questao. Esta tabela tem o atributo

Id_Tecnologia que é um atributo chave *surrogate key*, logo obrigatório. Ainda há também o atributo Descricao da tecnologia e a Versão.

Tabela tb_restricao: A tabela tb_restricao foi criada para o armazenamento das restrições das questões. Esta tabela tem uma ligação 1-N com a tabela tb_questao_restricao. A tb_restricao tem o atributo Id_Restricoes que é um atributo chave *surrogate key*, logo obrigatório. Ainda há também o atributo Descricao da restrição e a Valores.

Tabela tb_questao_restricoes: A tabela tb_questao_restricoes foi criada para o armazenamento da relação entre as restrições que serão empregadas e as questões que receberão essas restrições. Esta tabela é resultado de uma associação N-N entre as tabelas tb_restricao e tb_questao. A tb_questao_restricoes tem os seguintes atributos obrigatórios que são: Id_Questao e Id_Restricao.

5. FERRAMENTA

A ferramenta *on-line* desenvolvida tem como objetivo avaliar de forma automática procedimentos armazenados em banco de dados.

Nesta seção apresentaremos um estudo de caso com uma turma fictícia, porém uma questão real utilizada em turmas da disciplina de Sistemas de Informação, na parte de Programação TSQL.

Tela Inicial

A Figura 12 apresenta a tela inicial da ferramenta. As opções do menu *Home* e *Sobre* redireciona para tela de início e para uma tela de informações básicas sobre a ferramenta, respectivamente. Já no canto superior direito, é possível visualizar os links de *Registre-se* e *Entrar*, que veremos adiante.



Figura 13: Tela Inicial da ferramenta

Registrar-se

A Figura 13 apresenta a tela para criar uma nova conta de usuário na ferramenta. O usuário terá que fornecer informações como *Nome Completo*, *Nome de Usuário*, *E-mail* e *Senha* para ter acesso à ferramenta.

A imagem mostra a interface de registro de usuário da ferramenta SQL Escola. No topo, há uma barra azul com o logo "SQL Escola" e links para "[Registrar-se]" e "[Entrar]". Abaixo, há dois botões: "Home" e "Sobre". O título principal da seção é "Criar uma nova conta de usuário". Abaixo do título, há duas linhas de texto: "Utilize o formulário abaixo para criar uma nova conta." e "São necessários 7 caracteres para validar a senha.". O formulário, intitulado "Informações da Conta", contém campos para: "Nome Completo", "Nome de Usuário", "E-mail", "Senha" e "Confirmação de Senha". Um botão verde "Registrar" está localizado abaixo do formulário.

Figura 14: Tela de Registre-se da ferramenta

Menu Principal

Ao se registrar na ferramenta, o usuário recebe o perfil de usuário, ou seja, aluno. A figura 14 apresenta a tela em que um usuário fictício de nome *João Pedro* é um aluno que se inscreveu na ferramenta. Assim, este perfil tem as opções de *Turmas*, *Responder Questão* e *Solicitar Perfil de Professor*, além das opções iniciais *Home* e *Sobre*.



Figura 15: Tela Menu Principal da ferramenta

Solicitar Perfil de Professor

Para que o usuário se torne professor é necessário que o mesmo solicite uma ativação em seu perfil. Para realizar esta operação é necessário que ele entre nesta tela de solicitar perfil de professor.



Figura 16: Tela de Solicitar Perfil de Professor

Ativar Perfil de Professor

O administrador do sistema é responsável por ativar o perfil de professor do usuário. Assim, o administrador utilizará esta tela de ativar perfil.

Perfil: Administrador
 Usuário: admin! [Sair]

[Home](#)
[Ativar Professores](#)
[Sobre](#)

Ativar Solicitação de Professores

Nome	E-mail do Usuário	Status da Solicitação	Ativar	Desativar
Andrés Silva Menéndez	andres.s.menendez@gmail.com	Ativo	Ativar	Desativar
João Pedro	joao@gmail.com	Aguardando Ativação do Adm	Ativar	Desativar

Figura 17: Tela de Ativar Perfil de Professor

Turmas

Após o professor ter seu acesso como o perfil adequado, ele tem acesso às turmas. Assim, ele irá se deparar com a tela de turmas, exibida na figura 17. Nesta tela, além dos nomes das turmas cadastradas, também é exibido o nome do professor, alunos matriculados e as opções de *Atividades*, *Editar*, *Detalhes* e *Remover*.

Perfil: Professor
 Usuário: andres! [Sair]

[Home](#)
[Turmas](#)
[Minhas Turmas](#)
[Alternar Perfil](#)
[Sobre](#)

Turmas

[Criar Nova Turma](#)
[Exibir Todas as Turmas](#)

Nome da Turma	Professor	Alunos Matriculados	Atividades	Editar	Detalhes	Remover
Programação TSQL	Andrés Silva Menéndez	2	Atividades	Editar	Detalhes	Remover
Turma do TCC	Andrés Silva Menéndez	2	Atividades	Editar	Detalhes	Remover

Figura 18: Tela de Minha Turmas do Professor

Criar Turma

Na figura 18 é possível visualizar a tela que realiza a criação de turma. Esta tela é composta pelo campo destinado a descrição do nome da turma e o botão para a criação.

SQL Escola

Perfil: Professor
Usuário: andres! [[Sair](#)]

[Home](#) [Turmas](#) [Minhas Turmas](#) [Alternar Perfil](#) [Sobre](#)

Criar Turma

Criar Turma

Nome da Turma

[Criar Turma](#)

[Retornar](#)

Figura 19: Tela de Criar Turma

Detalhar Turma

Na figura 19 é possível visualizar a tela de detalhes da turma. Esta tela é composta pelos alunos que estão matriculados na turma. O objetivo é o próprio professor ter o conhecimento de quem está matriculado em sua turma.

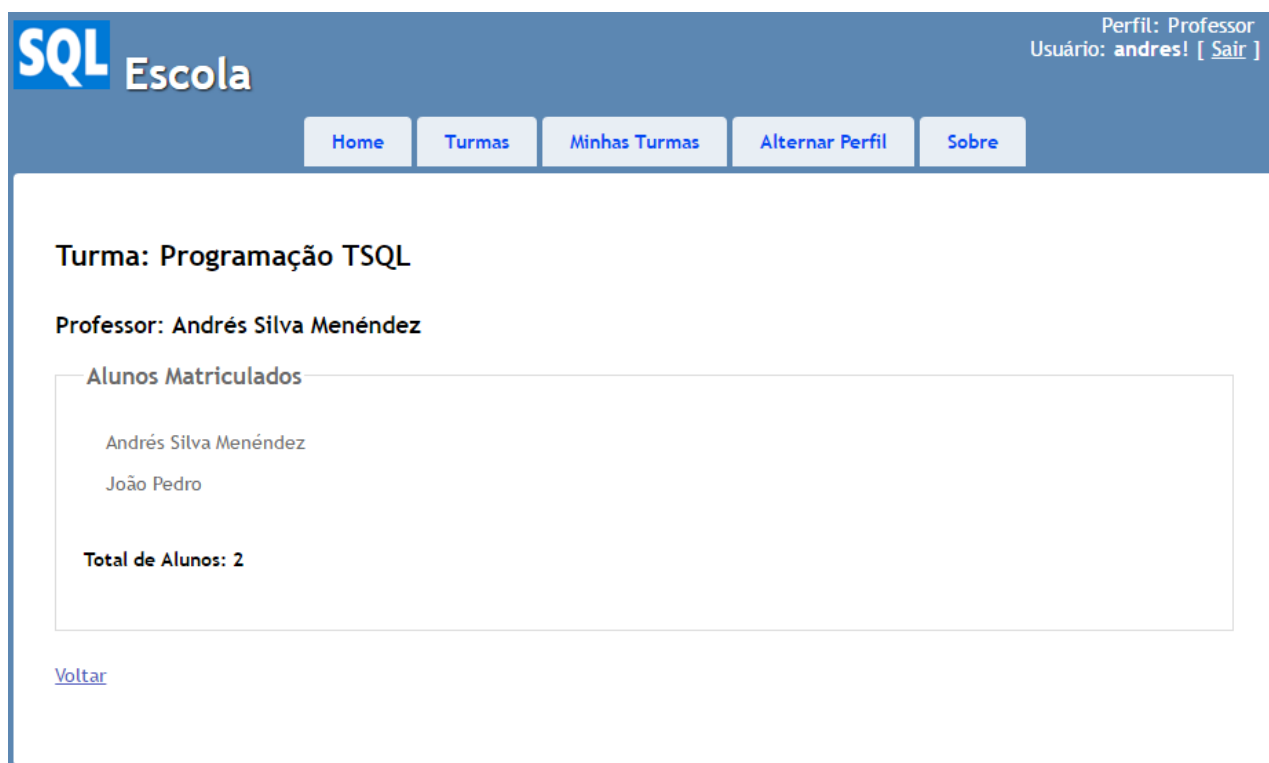


Figura 20: Tela de Detalhar Turma

Criar Atividade

Na figura 20 é possível visualizar a tela que realiza a criação de uma nova atividade. Esta tela é composta pelo campo destinado a descrição do nome da atividade e o botão para a criação.

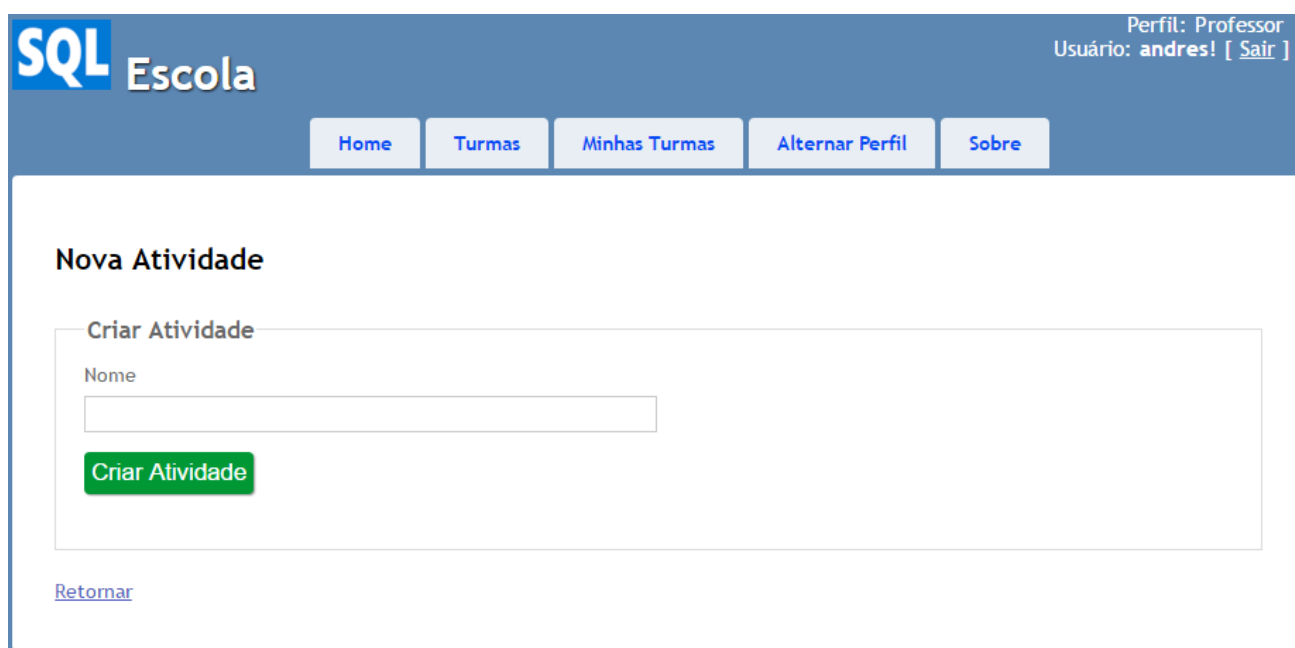


Figura 21: Tela de Criar Atividade

Criar Questão

Para o estudo de caso, vamos criar a seguinte questão utilizada em turmas de programação TSQL.

1. *Criar um procedimento SP_CLASSIFICA_CLIENTE para classificar um Cliente como PARCEIRO ou ALVO. O procedimento deve receber como parâmetros de entrada a data de inclusão do Cliente e a quantidade de produtos que um Cliente já comprou. O procedimento deve apresentar um parâmetro de saída informando se o Cliente é PARCEIRO ou ALVO. O cliente será classificado de acordo com a seguinte regra:*

*Se a data de inclusão do Cliente é maior ou igual a '01/03/2010' ou a Quantidade de Produtos já comprados pelo Cliente é maior que 50
O Cliente é PARCEIRO
Caso Contrário o Cliente é ALVO.*

Assim, vamos adaptar esta questão original ao sistema. Primeiramente, temos as figuras 21 e 22, que se complementam formando a tela de criar questão completa.

SQL Escola

Perfil: Professor
Usuário: andres! [Sair]

Home Turmas Minhas Turmas Alternar Perfil Sobre

Nova Questão

Criar Questão

Número de Ordem

Descrição

Carregamento de Arquivo (PDF ou DOC)
 Nenhum arquivo selecionado

Figura 22: Tela de Criar Questão- Parte 1

Script de Criação

Script de Povoamento

Nome do Procedimento do Script de Resolução

Script de Resolução

Casos de Teste

Criar Questão

Figura 23: Tela de Criar Questão- Parte 2

Para adaptar a questão à ferramenta é necessário observar que os campos *Ordem*, *Descrição*, *Nome do Procedimento do Script de Resolução*, *Script Resolução* e *Casos de Teste* são obrigatórios. Assim, é necessário que o professor, além de cadastrar a descrição como vimos acima, também preencha o Script de Resolução com a sua resposta que julga ser ideal, pois será necessária no momento de avaliar automaticamente a resposta do aluno.

Portanto, temos o seguinte Script de Resolução:

```
CREATE PROCEDURE SP_CLASSIFICA_CLIENTE (@DT_INCLUSAO
DATETIME @QTDE_COMPRAS INT, @INFO VARCHAR(8) OUTPUT )
As begin
    if @DT_INCLUSAO >= '20100301' OR @QTDE_COMPRAS > 50
        set @INFO = 'PARCEIRO'
    else
        set @INFO = 'ALVO'
end
```

Ainda que tenhamos o Script de Resolução enquadrado na questão, precisaremos ainda cadastrar os casos de teste que serão utilizados para avaliar a resposta do aluno. Deste modo, temos os seguintes casos de teste:

```
DECLARE @INFO VARCHAR(8)
EXEC SP_CLASSIFICA_CLIENTE '20090409', 25, @INFO OUTPUT
SELECT @INFO;
```

```
DECLARE @INFO VARCHAR(8)
EXEC SP_CLASSIFICA_CLIENTE '20100309', 50, @INFO OUTPUT
SELECT @INFO;
```

```
DECLARE @INFO VARCHAR(8)
EXEC SP_CLASSIFICA_CLIENTE '20160409', 30, @INFO OUTPUT
SELECT @INFO;
```

```
DECLARE @INFO VARCHAR(8)
EXEC SP_CLASSIFICA_CLIENTE '20100227', 60, @INFO OUTPUT
SELECT @INFO;
```

```
DECLARE @INFO VARCHAR(8)
EXEC SP_CLASSIFICA_CLIENTE '20100301', 10, @INFO OUTPUT
SELECT @INFO;
```

Para cada caso de teste, será gerado um resultado. Neste caso um resultado pode ter duas opções de acordo com o procedimento do estudo de caso, são elas: PARCEIRO ou ALVO.

Mediante o exposto, temos os devidos preenchimentos de acordo com a figura 23:



Figura 24: Tela de Visualizar Questão

Questões

A Figura 24 apresenta a tela de questões da ferramenta. Nesta tela são apresentadas todas as questões cadastradas para esta turma e atividade. Os campos apresentados são os de *Ordem*, *Descrição da Questão* e *Status*. As opções para cada questão são *Respostas*, *Visualizar*, *Validar*, *Editar* e *Remover*.

Questões

Turma: Programação TSQL / Professor: Andrés Silva Menéndez / Atividade: Atividade 10

[Criar Nova Questão](#)

Ordem	Descrição da Questão	Status	Respostas dos Alunos	Visualizar	Validar Questão	Editar	Remover
1	Criar um procedimento SP_CLASSIFICA_CLIENTE para classificar um Cliente como PARCEIRO ou ALVO. O procedimento deve receber como parâmetros de entrada a data de inclusão do Cliente e a quantidade de produtos que um Cliente já comprou. O procedimento deve apresentar um parâmetro de saída informando se o Cliente é PARCEIRO ou ALVO. O cliente será classificado de acordo com a seguinte regra: Se a data de inclusão do Cliente é maior ou igual a '01/03/2010' ou a Quantidade de Produtos já comprados pelo Cliente é maior que 50 O Cliente é PARCEIRO Caso Contrário o Cliente é ALVO.	Criada	Respostas	Visualizar	Validar	Editar	Remover

[Retornar para Atividades](#)**Figura 25: Tela de Questões****Validar Questão**

Com o objetivo de validar a questão em um banco de dados, ou seja, verificar se os scripts e casos de teste estão de acordo, é necessário que o professor utilize a opção de validar questão. A opção validar questão representa um link para a tela de validação de questão (figura 25). Esta tela do sistema é de fundamental importância, uma vez que se houver problemas na criação da questão, não será possível a avaliação automática por parte da ferramenta.

Número de Ordem

1

Descrição

Criar um procedimento SP_CLASSIFICA_CLIENTE para classificar um Cliente como PARCEIRO ou ALVO. O procedimento deve receber como parâmetros de entrada a data de inclusão do Cliente e a quantidade de produtos que um Cliente já comprou. O procedimento deve apresentar um parâmetro de saída informando se o Cliente é PARCEIRO ou ALVO. O cliente será classificado de acordo com a seguinte regra:

Se a data de inclusão do Cliente é maior ou igual a '01/03/2010' ou a Quantidade de Produtos já comprados pelo Cliente é maior que 50,
O Cliente é PARCEIRO
Caso Contrário o Cliente é ALVO.

Script de Criação

Script de Povoamento

Nome do Procedimento do Script de Resolução

SP_CLASSIFICA_CLIENTE

Script de Resolução

```
CREATE PROCEDURE SP_CLASSIFICA_CLIENTE (@DT_INCLUSAO  
DATETIME, @QTDE_COMPRAS INT, @INFO VARCHAR(8) OUTPUT )  
As begin  
    if @DT_INCLUSAO >= '20100301' OR @QTDE_COMPRAS > 50  
        set @INFO = 'PARCEIRO'  
    else  
        set @INFO = 'ALVO'  
end
```

Casos de Teste

```
DECLARE @INFO VARCHAR(8)  
EXEC SP_CLASSIFICA_CLIENTE '20090409', 25, @INFO  
OUTPUT  
SELECT @INFO;  
DECLARE @INFO VARCHAR(8)  
EXEC SP_CLASSIFICA_CLIENTE '20100309', 50, @INFO  
OUTPUT  
SELECT @INFO;  
DECLARE @INFO VARCHAR(8)  
EXEC SP_CLASSIFICA_CLIENTE '20160409', 30, @INFO  
OUTPUT  
SELECT @INFO;  
DECLARE @INFO VARCHAR(8)  
EXEC SP_CLASSIFICA_CLIENTE '20100227', 60, @INFO  
OUTPUT  
SELECT @INFO;  
DECLARE @INFO VARCHAR(8)  
EXEC SP_CLASSIFICA_CLIENTE '20100301', 10, @INFO  
OUTPUT  
SELECT @INFO;
```

Figura 26: Tela de Validar Questão Particionada

Submeter Resposta

Depois que a questão é validada pelo professor, ela fica visível na visão do aluno. Desta forma, o aluno poderá submeter sua resposta para avaliação automática da ferramenta. Para isto, o aluno deve entrar na tela de questões (figura 26). Nesta tela, é possível observar os campos de exibição *Ordem* e *Descrição da Questão*, além do link *Responder* nas opções.

Questões

Turma: Programação TSQL / Professor: Andrés Silva Menéndez / Atividade: Atividade 10

Ordem	Descrição da Questão	Opções
1	Criar um procedimento SP_CLASSIFICA_CLIENTE para classificar um Cliente como PARCEIRO ou ALVO. O procedimento deve receber como parâmetros de entrada a data de inclusão do Cliente e a quantidade de produtos que um Cliente já comprou. O procedimento deve apresentar um parâmetro de saída informando se o Cliente é PARCEIRO ou ALVO. O cliente será classificado de acordo com a seguinte regra: Se a data de inclusão do Cliente é maior ou igual a '01/03/2010' ou a Quantidade de Produtos já comprados pelo Cliente é maior que 50 O Cliente é PARCEIRO Caso Contrário o Cliente é ALVO.	Responder

[Retornar para Atividades](#)

Figura 27: Tela de Questão Visão Aluno

Após isto, temos a tela de Responder a questão propriamente dita. Na figura 27 é possível observar a descrição da questão juntamente com o número de ordem. Logo depois, percebemos dois campos: *Script de Resposta* e *Nome do Procedimento no Script de Resposta*.

Responder Questão

Questão: Criar um procedimento SP_CLASSIFICA_CLIENTE para classificar um Cliente como PARCEIRO ou ALVO. O procedimento deve receber como parâmetros de entrada a data de inclusão do Cliente e a quantidade de produtos que um Cliente já comprou. O procedimento deve apresentar um parâmetro de saída informando se o Cliente é PARCEIRO ou ALVO. O cliente será classificado de acordo com a seguinte regra: Se a data de inclusão do Cliente é maior ou igual a '01/03/2010' ou a Quantidade de Produtos já comprados pelo Cliente é maior que 50 O Cliente é PARCEIRO Caso Contrário o Cliente é ALVO.

Ordem: 1

Submeter Questão

Script de Resposta

Nome do Procedimento no Script de Resposta

Submeter Resposta para Avaliação Automática

Figura 28: Tela de Responder Questão

Visualizar Resultado

Após a etapa de submeter a questão, é gerado o resultado da avaliação automática pelo sistema que é exibido nas telas de resultado. As figuras 28 e 29 são complementares. A figura 28 é a primeira parte da tela de visualizar resultado. Nela é

possível observar que é exibido o script de resposta do aluno e uma parte do resultado das execuções dos casos de teste.

Visualizar Resultado

Questão: Criar um procedimento SP_CLASSIFICA_CLIENTE para classificar um Cliente como PARCEIRO Cliente e a quantidade de produtos que um Cliente já comprou. O procedimento deve apresentar um acordo com a seguinte regra: Se a data de inclusão do Cliente é maior ou igual a '01/03/2010' ou a Q Contrário o Cliente é ALVO. / Ordem: 1 / Data e Hora da Resposta: 17/04/2016 16:53:09

Resultado

Script Resposta

```
CREATE PROCEDURE SP_CLASSIFICA_CLIENTE (@DT_INCLUSAO VARCHAR(10), @QTDE_COMPRAS INT, @INFO VARCHAR(8)
OUTPUT )
as
begin
    DECLARE @DATA DATETIME
    SET @DATA = CONVERT(datetime, @DT_INCLUSAO, 103)
    if @DATA >= CONVERT(datetime, '01/03/2010', 103) OR @QTDE_COMPRAS > 50
        set @INFO = 'PARCEIRO'
    else
        set @INFO = 'ALVO'
end
```

Resultado

```
Caso de Teste 1
DECLARE @INFO VARCHAR(8)
EXEC SP_CLASSIFICA_CLIENTE '09/04/2009', 25, @INFO OUTPUT
SELECT @INFO

Resultado Correto:
alvo

Caso de Teste 2

DECLARE @INFO VARCHAR(8)
EXEC SP_CLASSIFICA_CLIENTE '09/03/2010', 50, @INFO OUTPUT
SELECT @INFO
```

Figura 29: Tela de Visualizar Resultado – Parte 1

Resultado

```
Caso de Teste 1
DECLARE @INFO VARCHAR(8)
EXEC SP_CLASSIFICA_CLIENTE '09/04/2009', 25, @INFO OUTPUT
SELECT @INFO

Resultado Correto:
alvo

Caso de Teste 2

DECLARE @INFO VARCHAR(8)
EXEC SP_CLASSIFICA_CLIENTE '09/03/2010', 50, @INFO OUTPUT
SELECT @INFO

Resultado Correto:
parceiro

Caso de Teste 3

DECLARE @INFO VARCHAR(8)
EXEC SP_CLASSIFICA_CLIENTE '09/04/2016', 30, @INFO OUTPUT
SELECT @INFO

Resultado Correto:
parceiro

Caso de Teste 4

DECLARE @INFO VARCHAR(8)
EXEC SP_CLASSIFICA_CLIENTE '27/02/2010', 60, @INFO OUTPUT
SELECT @INFO

Resultado Correto:
parceiro

Caso de Teste 5

DECLARE @INFO VARCHAR(8)
EXEC SP_CLASSIFICA_CLIENTE '01/03/2010', 10, @INFO OUTPUT
SELECT @INFO

Resultado Correto:
parceiro
```

Comentários do Professor

Figura 30: Tela de Visualizar Resultado – Parte 2

Já na figura 29, é exibida a segunda parte da tela de visualizar resultado. Nela é possível observar que é exibido o campo para resultado e outro campo destinado aos comentários do professor sobre este resultado.

A composição do campo Resultado é dado da seguinte forma:

- Número do Casos de Teste em questão:

Resultado

Caso de Teste 1

```
DECLARE @INFO VARCHAR(8)
EXEC SP_CLASSIFICA_CLIENTE '09/04/2009', 25, @INFO OUTPUT
SELECT @INFO
```

Resultado Correto:
alvo

Figura 31: Composição do Resultado – N° Caso de Teste

- Casos de Teste em questão cadastrado pelo professor:

Resultado

Caso de Teste 1

```
DECLARE @INFO VARCHAR(8)
EXEC SP_CLASSIFICA_CLIENTE '09/04/2009', 25, @INFO OUTPUT
SELECT @INFO
```

Resultado Correto:
alvo

Figura 32: Composição do Resultado – Caso de Teste

- Resultado final:

Resultado

Caso de Teste 1

```
DECLARE @INFO VARCHAR(8)
EXEC SP_CLASSIFICA_CLIENTE '09/04/2009', 25, @INFO OUTPUT
SELECT @INFO
```

Resultado Correto:
alvo

Figura 33: Composição do Resultado – Resultado Final

Esta etapa de resultados pode ter o resultado correto, como é exibido no exemplo, ou pode ter o resultado que difere do resultado do professor. De modo independente do resultado da avaliação, esta etapa de resultado final (figura 32) irá exibir o resultado da avaliação do aluno, nunca a do professor. O número do caso de teste em questão (figura 30) e o caso de teste cadastrado pelo professor (figura 31) compõem o resultado final.

Histórico de Submissões

Na figura 33 é exibida a tela de histórico de submissões. Esta tela contém todas as submissões do aluno para esta questão. Ela exibe a data em que a resposta foi submetida, além das opções de visualizar o script de resposta submetido, adicionalmente com a possibilidade de editar o script e por fim, visualizar o resultado.

SQL Escola

Perfil: Usuário
Usuário: joao! [Sair]

HomeTurmasResponder QuestãoSolicitar Perfil de ProfessorSobre

Histórico de Respostas

Questão: Criar um procedimento SP_CLASSIFICA_CLIENTE para classificar um Cliente como PARCEIRO ou ALVO. O procedimento deve receber como parâmetros de entrada a data de inclusão do Cliente e a quantidade de produtos que um Cliente já comprou. O procedimento deve apresentar um parâmetro de saída informando se o Cliente é PARCEIRO ou ALVO. O cliente será classificado de acordo com a seguinte regra: Se a data de inclusão do Cliente é maior ou igual a '01/03/2010' ou a Quantidade de Produtos já comprados pelo Cliente é maior que 50 O Cliente é PARCEIRO Caso Contrário o Cliente é ALVO.

Ordem: 1

Data	Script Resposta	Editar Script	Resultado
17/04/2016 16:54:26	Visualizar	Editar	Visualizar Resultado
17/04/2016 16:53:09	Visualizar	Editar	Visualizar Resultado

Figura 34: Histórico de Submissões

6. CONCLUSÕES E TRABALHOS FUTUROS

Os cursos de programação em banco de dados possuem as mesmas características dos cursos iniciais de programação. Nesses cursos o *feedback* do professor para os exercícios passados em sala de aula constitui fator crucial para o aprendizado, principalmente para aqueles que estão vendo uma linguagem ou tecnologia pela primeira vez. É através desse *feedback* que o aluno pode resolver erros impeditivos e avançar no conteúdo. A inexistência de um *feedback* ou um *feedback* tardio, que na prática traz as mesmas consequências que o primeiro, produz um desestímulo pelo curso em função da incapacidade de resolver alguns exercícios.

Entre os vários fatores que contribuem para um *feedback* demorado do professor está, principalmente, o dia a dia ocupado do professor. Além disso, uma avaliação manual de exercícios de programação numa turma grande de programação em banco de dados representa uma extensa carga de trabalho, pois muitos exercícios exigem, não apenas a sua execução, mas a preparação de um ambiente inicial e a execução de muitos casos de testes para uma avaliação efetiva das respostas dos alunos.

Na literatura foram encontradas diversas ferramentas, sistemas e ambientes que realizam uma avaliação automática de exercícios de programação. No entanto, até o presente momento, não foi encontrada nenhuma ferramenta de avaliação automática para programação em extensões da linguagem SQL. Avaliar automaticamente os procedimentos armazenados dos alunos numa turma grande de programação em banco de dados pode trazer benefícios tanto para o professor, quanto para os alunos envolvidos. Portanto, uma ferramenta *on-line*, disponível a qualquer momento, que auxilia professores no ensino de programação em banco de dados retornando para os alunos um *feedback* instantâneo dos seus exercícios é o grande objetivo deste trabalho.

Dentre os principais desafios para a avaliação automática de procedimentos armazenado está a abrangência de questões que a ferramenta pode avaliar. Existem muitas possibilidades e tipos de questões a serem validadas e respostas a serem avaliadas. Portanto, é de suma importância que a ferramenta torne-se o mais genérica possível, ou seja, tenha o poder de avaliar o maior número possível de questões. Assim, através de revisão da literatura, pesquisa de usuários e muitas discussões, chegamos a

alguns requisitos necessários e satisfatórios para a realização da avaliação automática pela ferramenta.

Outros trabalhos moderadamente relacionados expõem seus ideais sobre avaliação automática de respostas. Em (GUPTAL; DUBEY, 2005) é proposto um sistema que avalia automaticamente trabalhos de programação em geral. Em (SILVA; LINO; FAVERO; 2009) é proposta uma implantação de um laboratório virtual de consultas básicas de SQL no ambiente Moodle. Em (MITROVIC, 2000), é proposto um sistema tutor inteligente para programação SQL, porém aborda apenas consultas sobre o comando *Select*. Em (SADIQ; ORLOWSKA; SADIQ; LIN, 2004) é proposto um ambiente *on-line* que dispõe de um tutorial integrado que apresenta conceitos fundamentais, oferecendo diversos bancos de dados para prática de consultas básicas. Em (PRIOR; LISTER, 2004), é proposto uma ferramenta web de treinamento e avaliação de consultas básicas SQL provendo uma lista de questões já integrada.

O presente trabalho difere dos trabalhos anteriores na medida em que apresenta uma abordagem para a avaliação automática de procedimentos armazenados, e não de consultas básicas SQL. Para isto, foi utilizada a abordagem dinâmica. A análise dinâmica avalia um programa por executá-lo com base nos dados de teste que são gerados automaticamente ou manualmente fornecidos. Para uma análise eficaz, o programa final deve ser executado com entradas de teste suficientes para produzir um *feedback* importante (GUPTAL, DUBEY, 2005).

Logo, com o uso da técnica de caixa-preta são utilizados testes funcionais para garantir um resultado rápido sobre os procedimentos armazenados. Esta técnica foi escolhida, pois numa avaliação manual é difícil descobrir os resultados de um programa, uma vez que o programa pode ter muitos caminhos diferentes para um mesmo resultado. Portanto, esta técnica se baseia em comparar os resultados finais de uma possível solução ótima, com os resultados do aprendiz.

A arquitetura e abordagem propostas podem servir de base para trabalhos futuros. A ferramenta ainda precisa ser validada através de experimentos em sala de aula para comprovar os benefícios alcançados por ferramentas de avaliação automática não voltadas para banco de dados. Outros módulos também podem estender as funcionalidades da ferramenta. A avaliação por abordagem estática, por exemplo, pode complementar a abordagem dinâmica adotada. Após a utilização da ferramenta por um número considerável de turmas, os dados armazenados também podem servir de insumo

para trabalhos na área de Mineração de Dados a fim de potencializar o aprendizado a respeito dos alunos e do próprio curso (ALLEVATO, THORNTON, EDWARDS, QUINONES; 2008).

REFERÊNCIAS

ALLEVATO, Anthony. THORNTON, Matthew. EDWARDS, Stephen. QUINONES, Perez. **Mining Data from an Automated Grading and Testing System by Adding Rich Reporting Capabilities.** Department of Computer Science, Virginia Tech. EDM 8, Pág. 167-176. 2008.

DOUCE, Christopher; LIVINGSTONE, David; ORWELL, James; **Automatic test-based assessment of programming: A review.** Journal on Educational Resources in Computing. Volume 5 Issue 3, September 2005.

FAVERO, E. L. **Avaliação automática de consultas SQL em ambiente virtual de ensino-aprendizagem.** CISTI 2007 - 2ª Conferência Ibérica de Sistemas e Tecnologias de Informação 2007.

FAVERO, A. Lobato and E. L. **Aplicando rubrica para avaliar qualitativamente o estudante no labsql.** in CLEI'08: Proceedings of the XXXIV Conferencia Latinoamericana de Informática, 2008.

GUPTAL, Surendra; DUBEY, Kumar; **Automatic Assessment of Programming assignment.** Department of Computer Engineering. Shri G. S. Institute of Technology & Science 23, Park Road Indore, India. Pág. 315-323. 2012.

ILANTOLA, Petri. AHONIEMI, Tuukka. KARAVIRTA, Ville. SEPPALA, Otto. **Review of Recent Systems for Automatic Assessment of Programming Assignments.** Koli Calling '10, Pág. 28-31, 2010.

LIANG, Yingli; LIU, Quanbo; XU, Jun. **The recent development of automated programming assessment.** International Conference on Computational Intelligence and Software Engineering. Informatics in Education, Department Of Educational Information And Technology South China Normal University, Guangzhou. 2009.

LINO, A. D. P., SILVA, A. S., SANTOS, T.L.T. HARB, M.P.A.H., FAVERO, E.L., BRITO, S.R. **Avaliação automática de consultas SQL em ambiente virtual de ensino-aprendizagem.** Conferencia Ibérica de Sistemas y Tecnologías de la Información. CISTI. 2007.

MITROVIC, A; Martin, B; **Evaluating Effectiveness of Feedback in Sql-Tutor.** Iwalt 2000, Palmerston North, P. 143-144, 2000.

PIETERSE, Vreda. **Automated Assessment of Programming Assignments.** 3rd Computer Science Education Research Conference on Computer Science Education Research. Pág. 45-56. Vol. 3. 2013.

PRIOR, J. C. Lister, R. **The Backwash Effect on Sql Skills Grading.** in: Proceedings of The 9th Annual Sigcse Conference on Innovation and Technology in Computer Science Education (Leeds, United Kingdom). Acm Press, New York, P. 32-36., 2004.

SILVA, Osiel Marlon Negrão da; LINO, Adriano D. P.; FAVERO, Eloi Luiz. **Um Laboratório Virtual de Ensino de Programação SQL no Ambiente Moodle.** XX Simpósio Brasileiro de Informática na Educação. Universidade Federal do Pará (UFPA), Pará. 2009.

SADIQ, S., Orlowska, M., Sadiq, W. & Lin, J. **Sqlator: an Online Sql Learning Workbench.** in Proceedings of The 9th Annual Sigcse Conference on Innovation and Technology in Computer Science Education (Leeds, United Kingdom). Acm Press, New York, P. 223-227, 2004.

SKUPAS, B. **Feedback Improvement in Automatic Program Evaluation Systems.** Institute of Mathematics and Informatics, Vilnius. Vol. 9 Issue 2. Pág. 229-237. 9p. 2010.

TRUONG, N., Bancroft, P. & Roe, P. Elp. **A Web Environment for Learning to Program.** in: Proceedings the 19th Annual Conference of The Australasian Society for Computers in Learning in Tertiary Education, Auckland, New Zealand, 19:661-670, 2002.