

**LIGA DE ENSINO DO RIO GRANDE DO NORTE
FACULDADE NATALENSE PARA O DESENVOLVIMENTO DO RIO GRANDE DO
NORTE
CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

JÓRIO LEONARDO GARCIA DE PAIVA

SISTEMA DE RESOLUÇÃO DE QUESTÕES

**Natal/RN
2010**

JÓRIO LEONARDO GARCIA DE PAIVA

SISTEMA DE RESOLUÇÃO DE QUESTÕES

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Faculdade Natalense para o Desenvolvimento do Rio Grande do Norte como requisito final para obtenção do grau de Bacharel em Sistemas de Informação.

ORIENTADOR: Ms. Alexandre Luiz Galvão Damasceno

Natal/RN

2010

Catálogo na Publicação – Biblioteca da FARN
Setor de Processos Técnicos

Paiva, Jório Leonardo Garcia de
Sistema de resolução de questões / Jório Leonardo Garcia de Paiva. –
Natal: FARN, 2010.
50f.

Orientador: Alexandre Luiz Galvão Damasceno (Msc.)
Monografia (Graduação em Sistemas da Informação) – Faculdade
Natalense para o Desenvolvimento do Rio Grande do Norte.

1. Sistemas de Informação - Monografia. 2. Desenvolvimento de Web -
Monografia. 3. Banco de Dados - Monografia. I. Título.

RN/FARN/BC

CDU 004.4

RESUMO

O trabalho proposto tem como objetivo principal o desenvolvimento de um sistema de resolução de questões no qual tem a finalidade de aplicar os conhecimentos de desenvolvimento WEB e de Banco de Dados, adquiridos durante a graduação de BSI. O sistema será disponibilizado na internet, com as funcionalidades de resolução de questões como também de visualização de um ranking de desempenho. No seu desenvolvimento foram feitas análises em UML de alguns sistemas já existentes na internet, que serviram de referencia para o desenvolvimento, e análise do próprio sistema em questão com o objetivo de apresentar as características dos aplicativos, logo em seguida serão apresentadas as tecnologias utilizadas. Na sistemática aplicada ao desenvolvimento do projeto apresentado, foi adotada uma metodologia não muito convencional na qual o banco de dados não é visto como um simples repositório de dados, explorando ao máximo as funcionalidades que a SQL proporciona, colocando as regras de negócios em *procedures*, *views* e *functions*. Destarte, essa monografia pretende servir de referência para discentes que desejem implementar novos sistemas, sugerindo o modelo adotado no desenvolvimento.

Palavras-Chave: Desenvolvimento web. Banco de dados. SQL.

ABSTRACT

The proposed work aims at the development of a system for resolving issues in which aims to apply knowledge of Web development and Database, acquired during their studies of BSI. The system will be available on the internet, with features resolution of issues as well as viewing of a performance ranking. In its development were analyzed in UML some existing systems on the Internet, which served as reference for the development and analysis of the system in question in order to present the characteristics of the applications, then immediately present the technologies used. In the scheme applied to the development of the project presented, was adopted a very unconventional method in which the database is not viewed as a single data repository, using the full functionality that SQL provides, putting business rules in procedures, views and functions. So, this monograph is intended as a reference for students wishing to implement new systems, suggesting the model used in development.

Keywords: Web Development Database. SQL.

LISTA DE FIGURAS

FIGURA 1 – Diagrama UML e seus elementos	12
FIGURA 2 – Hierarquia entre atores.....	13
FIGURA 3 – Representação incorreta de caso de uso.....	14
FIGURA 4 – Representação indicada para o caso de uso.....	14
FIGURA 5 – Fluxo normal e fluxos alternativos.....	15
FIGURA 6 – Instrução <i>select</i>	17
FIGURA 7 – Utilização do operador <i>like</i>	17
FIGURA 8 – Posição da “%” na instrução <i>like</i>	18
FIGURA 9 – Utilização da cláusula <i>like</i> no sistema desenvolvido.....	19
Figura 10 – Pci concursos.....	21
Figura 11- Diagrama de caso de uso do pci concursos.....	22
Figura 12 - Questões de concursos.....	25
Figura 13 - Diagrama de caso de uso do site Questões de concursos.....	26
Figura 14 – O sistema desenvolvido.....	30
Figura 15 – Relatório de desempenho dos usuários.....	31
Figura 16 – Diagrama de casos de uso do sistema desenvolvido	32
Figura 17 – Diagrama de entidade relacionamento do sistema.....	35
Figura 18 – Processamento dividido em servidores.....	36
Figura 19 – Tela de Cadastro de questões.....	38
Figura 20 – Função Manutenção feita em ASP que invoca a procedure.....	39
Figura 21 – Stored Procedure de inclusão de questão.....	40
Figura 22 – Stored Procedure de alteração de questão.....	41
Figura 23 – Stored Procedure de exclusão de questão.....	42
Figura 24 - Stored Procedure de consulta de questão.....	42

Figura 25 - Stored Procedure que retorna o desempenho dos usuários.	43
Figura 26 - Autenticação do sistema.....	44
Figura 27 - Desabilitando o acesso anônimo no IIS.....	45
Figura 28 - Ferramenta Enterprise.....	47
Figura 29 - Ferramenta Query Analyzer.....	48

LISTA DE ABREVIATURAS, SIGLAS E SÍMBOLOS

BSI = Bacharelado de Sistema de Informações

DFD = Diagrama de Fluxo de Dados

UML = *Unified Modeling Language*

ASP = *Active Server Pages*

SQL = *Structured Query Language*

SGBD = Sistema de gerenciamento de banco de dados

DML = *Data manipulation language*

DDL = *Data definition language*

CESPE = Centro de Seleção e Promoção de Eventos da Universidade de Brasília

FCC = Fundação Carlos Chagas

DER = Diagrama de Entidade Relacionamento

IIS = *Internet Information Services*

SUMÁRIO

1 INTRODUÇÃO	9
1.1 OBJETIVOS.....	10
1.2 JUSTIFICATIVA.....	10
2 REFERENCIAL TEÓRICO	11
3 METODOLOGIA	20
3.1 PCI CONCURSOS.....	21
3.1.1 Descrição dos casos de uso do Pci concursos.....	23
3.1.2 Vantagens do sistema.....	24
3.1.3 Desvantagens do sistema.....	24
3.2 QUESTÕES DE CONCURSO.....	25
3.2.1 Descrição alguns caso de uso do Questões de concursos.....	27
3.2.2 Vantagens do sistema.....	28
3.2.3 Desvantagens do sistema.....	29
4 O SISTEMA DESENVOLVIDO	29
4.1 DIAGRAMA DE CASOS DE USO.....	31
4.1.1 Descrição de alguns casos de uso do sistema desenvolvido	32
4.2 DER.....	34
4.3 MODELO DE IMPLEMENTAÇÃO.....	35

4.4 MAPEAMENTOS DOS CASOS DE USO, TELAS E PROCEDURES	36
4.5 SEGURANÇA.	43
5 JUSTIFICATIVA PARA AS TECNOLOGIAS UTILIZADAS.....	46
5.1 ASP	46
5.2 MICROSOFT SQL SERVER 2000	49
6 CONCLUSÃO.....	42
REFERÊNCIAS	43

1 INTRODUÇÃO

Durante a graduação em um curso de Sistema de Informação, são adquiridos conhecimentos necessários para o desenvolvimento de *softwares*, incluindo os que possuem funcionamento WEB. Dessa forma, este trabalho tem como objetivo a implementação de um sistema com a funcionalidade de resolução de questões que será disponibilizado na Internet.

Inicialmente é apresentada a avaliação de sistemas já existentes na WEB, explicitando os principais casos de uso e a sua descrição, utilizando elementos da UML. Esta abordagem permite a elaboração de idéias para a criação do sistema proposto.

Em seguida são disponibilizadas as características do aplicativo desenvolvido, apresentando os seus casos de uso, seu funcionamento, o modelo de desenvolvimento, o diagrama de entidade relacionamento e as questões de segurança. O modelo de desenvolvimento utilizado é caracterizado por colocar a regra de negócio no banco de dados, através do uso de *stored procedures*, *tiggers*, *views* e *functions*; esse modelo escolhido é atípico, pois tradicionalmente o banco de dados é visto como um simples repositório de dados.

Finalmente apresentam-se os fatores de foram determinantes na escolha das tecnologias utilizadas, que estão diretamente ligadas ao modelo de desenvolvimento escolhido. Essas tecnologias foram o banco de dados SQL Server 2000, a linguagem de programação ASP, todas pertencentes à empresa Microsoft®.

Através da dissertação da experiência obtida no desenvolvimento do sistema contida nesta monografia pretende-se criar uma documentação para auxiliar outros estudantes que desejam implementar aplicativos com funcionamento WEB, através do modelo adotado.

1. 1 OBJETIVOS

Geral

Desenvolver um sistema com funcionamento web, aplicando os conhecimentos adquiridos em algumas disciplinas do curso. Através da documentação deste, será possível criar uma referência para futuros estudantes que desejem implementar algo semelhante.

Específicos

- Apresentar como o sistema foi desenvolvido, através da análise de sistemas já existentes.
- Mostrar suas características, através de seus casos de uso e suas telas.
- Apresentar um modelo de desenvolvimento no qual o banco de dados não é um simples repositório de dados.
- Explicar sobre as tecnologias utilizadas na implementação e o motivo da sua escolha.
- Disponibilizar a ferramenta desenvolvida como estudo de caso na internet para permitir que outros possam usá-la no auxílio de seus estudos.

1.2 JUSTIFICATIVA

Na implementação de um programa é possível iniciar o desenvolvimento de diversas formas como através da coleta de requisitos com o cliente ou análise de sistemas já existentes com as características desejadas; em seguida podem ser escolhidas diversas tecnologias para implementação escolhendo também a metodologia utilizada.

Assim esse trabalho apresenta umas das possíveis formas para se desenvolver um sistema WEB, mostrando os procedimentos adotados no

desenvolvimento de um sistema que será disponibilizado na internet. Desta forma servirá como referência para futuras implementações de sistemas por outros estudantes.

2 REFERENCIAL TEÓRICO

Para um melhor entendimento dos termos utilizados neste trabalho, é necessária inicialmente a explanação de conceitos abrangidos no contexto.

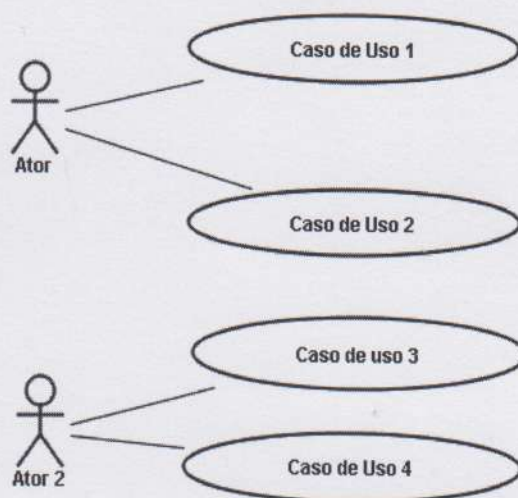
UML e seus elementos

A UML foi criada originalmente para modelagem de sistemas baseados em *software*. No entanto nos dias atuais é abrangida em vários domínios como em telecomunicações, sistemas eletrônicos, sistemas aeroespaciais entre outros. No ramo do desenvolvimento de *software* é utilizada para visualizar, especificar, construir e documentar os elementos de um sistema. Nela utilizamos os casos de uso para representar a interação entre o usuário e o sistema. (MARTINS, 2007)

Existem inúmeras vantagens em sua utilização, pois gera uma documentação para o sistema e evita a dependência em torno de um programador, facilitando futuras manutenções.

A UML é composta por inúmeros diagramas como classes, objetos, casos de uso, interação, sequência, estados, atividades, componentes e *deployment*. Neste trabalho serão apresentados os diagramas de caso uso dos sistemas que foram avaliados e do sistema desenvolvido. O diagrama de casos de uso tem o objetivo de apresentar os requisitos funcionais do sistema, não detalhando sobre a tecnologia utilizada. Apresentando o que o sistema irá fazer e não como irá fazer. Este diagrama é composto principalmente por ator, caso de uso e relacionamento, como pode ser visto na figura 1.

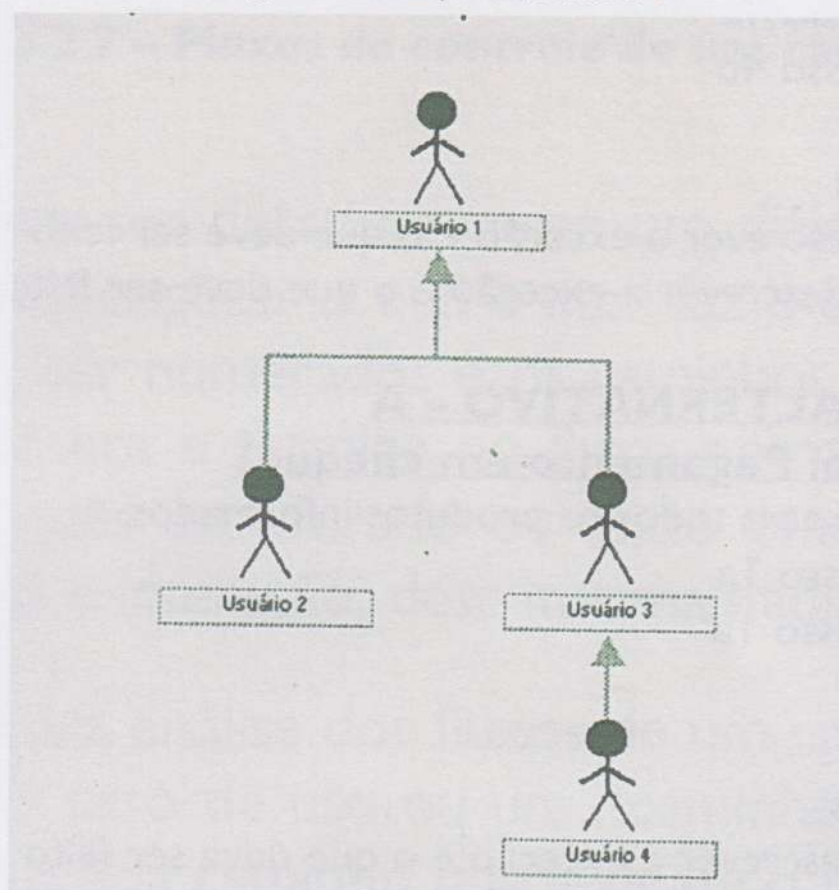
Figura 1 – Diagrama UML e seus elementos



Fonte: Documentação do sistema

Os atores são todos os elementos externos ao sistema que interage com o mesmo, eles podem ser usuários ou máquinas. Poderá existir uma hierarquia entre eles indicando que um ator herda os poderes e responsabilidades de outro ator, como visto na figura 2.

Figura 2 – Hierarquia entre atores.

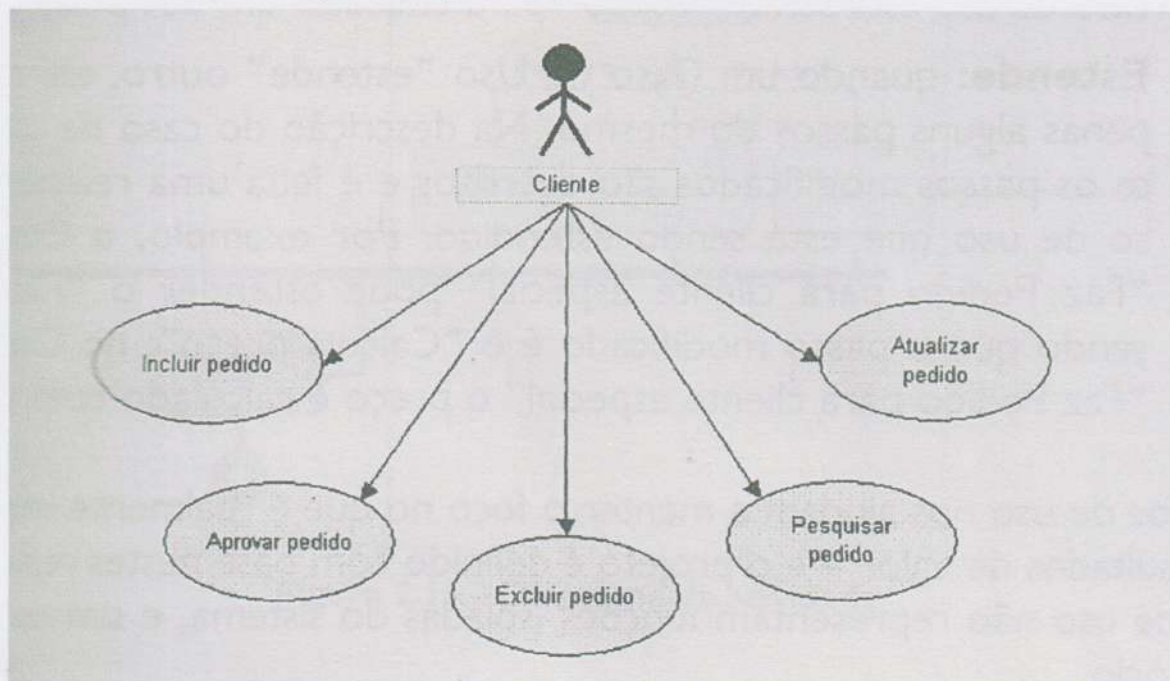


Fonte: MARTINS (2007, p.165)

O caso de uso é um elemento do diagrama que representa a especificação de uma sequência completa de interações entre um sistema e um ou mais agentes externos a esse sistema (BEZERRA, 2007). O nome para representar um caso de uso deverá ser escolhido de forma que destaque o seu valor para o usuário, por convenção utiliza-se um verbo seguido de um objeto. Bons exemplos são "Fazer um saque" e "Fazer um depósito" (MARTINS, 2007).

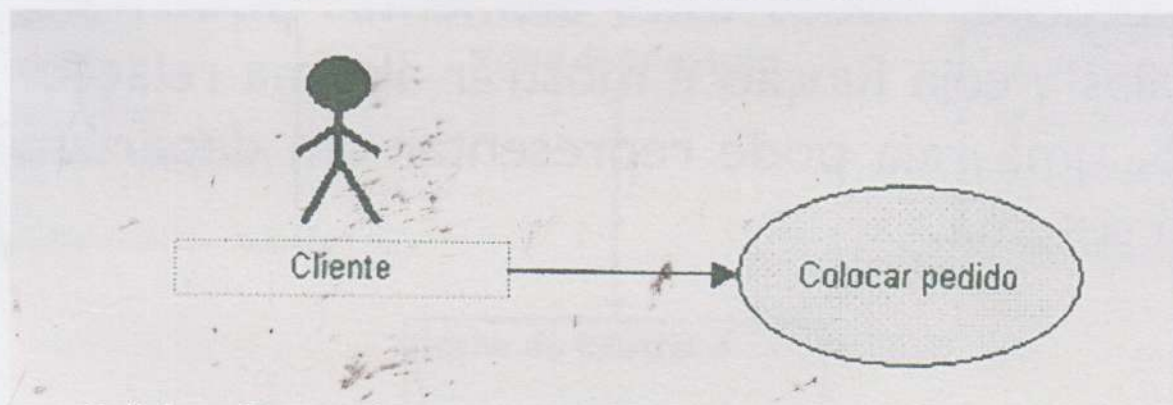
Na sua definição não se deve pensar como na criação de um DFD, que é uma decomposição de funcionalidades do sistema (MARTINS, 2007). Na figura 3 estão representados vários casos de uso que deveriam estar contidos apenas em um como visto na figura 4.

Figura 3 - Representação incorreta de caso de uso



Fonte: MARTINS (2007, p.173)

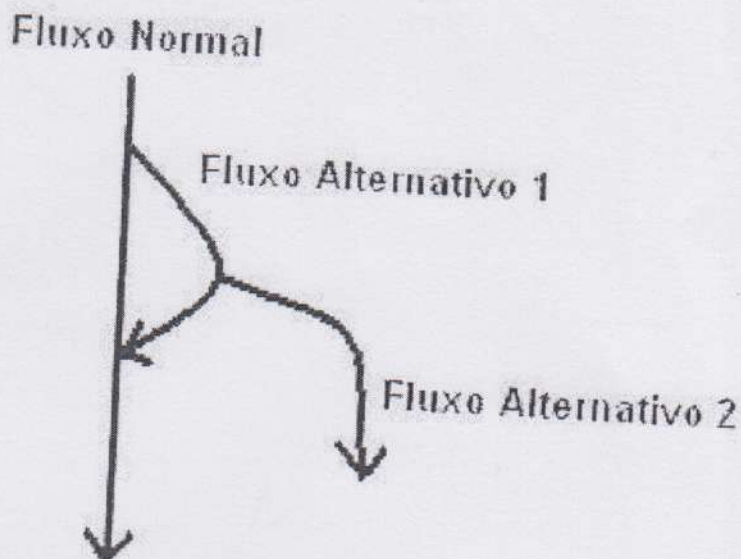
Figura 4 - Representação indicada para o caso de uso



Fonte: MARTINS (2007, p. 173)

A descrição funcional de um caso de uso é composta por um fluxo normal mostrando o que geralmente se espera que aconteça, como também nenhum ou vários fluxos alternativos mostrando o comportamento opcional como pode ser visto na figura 5.

Figura 5 - Fluxo normal e fluxos alternativos



Fonte: Martins (2007, p. 167)

Por fim existem os relacionamentos que segundo Bezerra (2007, p. 61),

Casos de uso e atores não existem sozinhos. Além desses últimos, o modelo de casos de uso possui um terceiro componente, cuja função é relacionar os atores e casos de uso. Esse componente corresponde aos relacionamentos.

Na UML são definidos os relacionamentos de comunicação, inclusão, extensão e generalização. O de comunicação tem a finalidade de indicar a quais casos de uso o ator interage.

A utilização do relacionamento de inclusão é indicada quando dois ou mais casos de uso utilizam uma sequência comum de interações, onde elas se transformam em outro caso de uso, assim vários casos de uso poderão incluir o comportamento comum evitando a descrição de uma mesma sequência de interações mais de uma vez (BEZERRA, 2007).

O relacionamento de extensão é utilizado para modelagem de situações em que diferentes sequências de interações podem ser inseridas em um mesmo caso de uso. No relacionamento de generalização poderá ocorrer entre casos de uso e entre atores (BEZERRA, 2007).

Os elementos da UML apresentados nesta seção, serão utilizados para modelagem dos sistemas apresentados nesta monografia, possibilitando uma visão de alto nível do sistema, em seguida serão abordadas as tecnologias utilizadas apresentado um pouco de como fazer a implementação dos casos de uso.

SQL

Implementada inicialmente pela IBM® na década de 1970, e aceita nos dias atuais pela grande maioria dos SGBD, a SQL é a linguagem padrão para lidar com banco de dados relacionais, nela encontramos operações de definição de dados e operações de manipulação de dados. O sistema de gerenciamento de banco de dados (SGBD) é o software que trata de todos os acessos ao banco de dados (DATE, 2004). Ela é conceituada como um conjunto de comandos de manipulação de banco de dados utilizado para criar e manter a estrutura desse banco de dados, além de incluir, excluir, modificar e pesquisar informações nas tabelas dele (Oliveira, 2002).

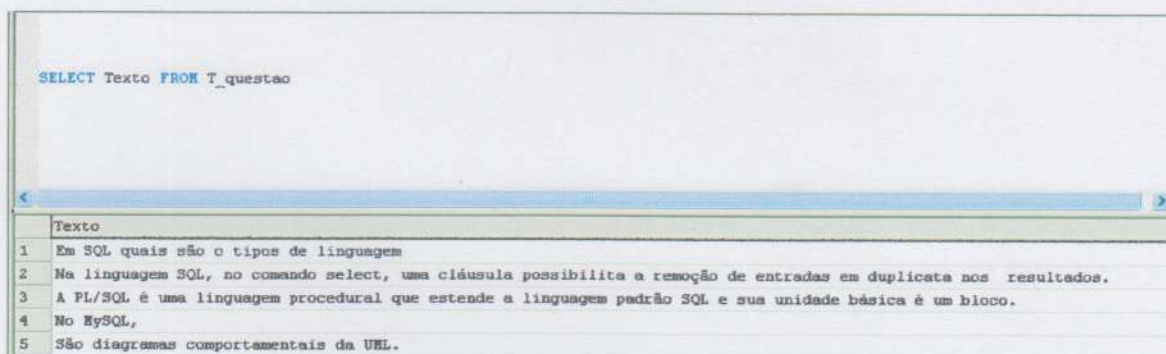
As operações de definição de dados são conhecidas por DDL sendo responsável por definir os detalhes de implementação dos esquemas de banco de dados, sendo armazenada no dicionário de dados, o qual é considerado uma tabela especial que só pode ser manipulada pelo próprio SGBD (SILBERSCHATZ et al, 2006). Instruções com *create type* e *create table* exemplificam essas operações.

As operações de manipulação de dados DML são operações de manipulação de dados, temos as instruções *select*, *insert*, *update* e *delete*. O sistema desenvolvido possui a característica de explorar ao máximo a utilização de procedimentos escritos em SQL, pois seu emprego é utilizado na elaboração das *stored procedures*.

Operador LIKE

Este operador da linguagem SQL tem a finalidade de indicar a existência de uma substring em uma string a ser buscada, ele é utilizado principalmente em combinação com o caractere especial "%". Na figura 6 é apresentada uma instrução DML para retornar todas as tuplas de uma determinada relação.

Figura 6 – Instrução *select*



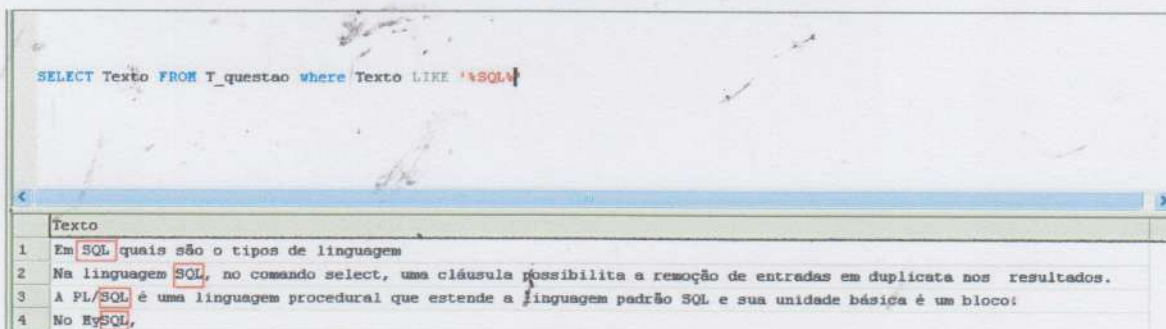
The screenshot shows a SQL query execution window. The query entered is `SELECT Texto FROM T_questao`. Below the query, the results are displayed in a table with one column, 'Texto', and five rows of data.

	Texto
1	Em SQL quais são o tipos de linguagem
2	Na linguagem SQL, no comando select, uma cláusula possibilita a remoção de entradas em duplicata nos resultados.
3	A PL/SQL é uma linguagem procedural que estende a linguagem padrão SQL e sua unidade básica é um bloco.
4	No MySQL,
5	São diagramas comportamentais da UML.

Fonte: Documentação do sistema

Para restringir a quantidade de linhas retornadas utiliza-se a cláusula *where* que pode ser utilizada em combinação com o operador like como visto na figura 7.

Figura 7 – Utilização do operador LIKE



The screenshot shows a SQL query execution window. The query entered is `SELECT Texto FROM T_questao where Texto LIKE '%SQL%'`. Below the query, the results are displayed in a table with one column, 'Texto', and four rows of data. The fifth row from the previous figure is omitted because it does not contain the substring 'SQL'.

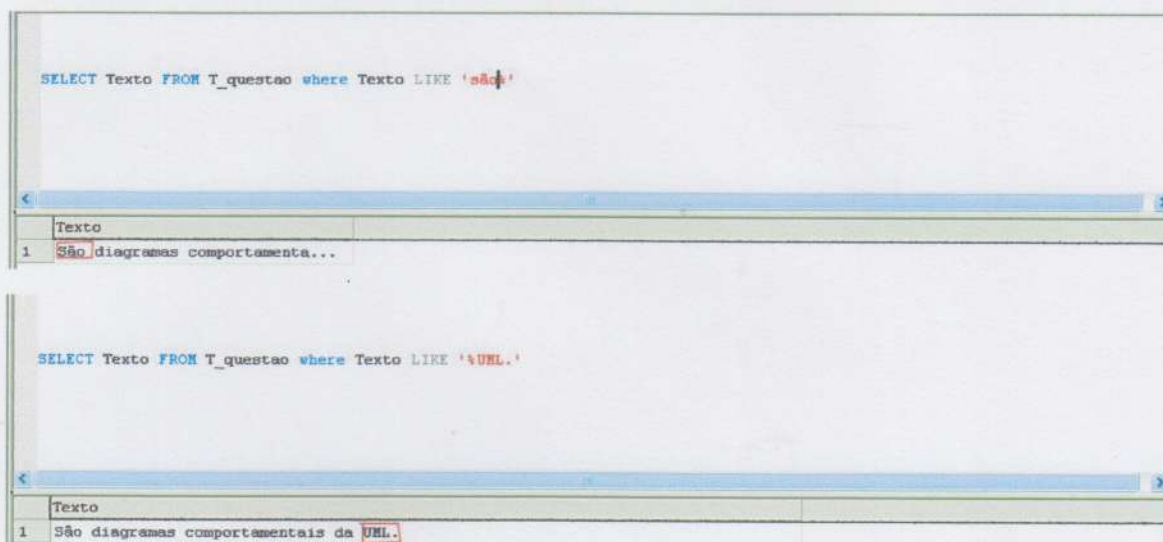
	Texto
1	Em SQL quais são o tipos de linguagem
2	Na linguagem SQL, no comando select, uma cláusula possibilita a remoção de entradas em duplicata nos resultados.
3	A PL/SQL é uma linguagem procedural que estende a linguagem padrão SQL e sua unidade básica é um bloco.
4	No MySQL,

Fonte: Documentação do sistema

Ao ser utilizado o operador *like*, foi omitida a ultima tupla, pois esta não continha a substring "SQL" em seu conteúdo. A forma de utilização do "%" influencia no resultado, pois caso seja colocado apenas no início da *substring*, indicará que se

deseja que a mesma esteja contida apenas no final da *string*, caso se coloque apenas no final se deseja que esteja contida apenas no início da *string*, o que pode ser visualizado na figura 8.

Figura 8 – Posição da “%” na instrução LIKE.



Fonte: Documentação do sistema

O operador *like* foi utilizado no sistema desenvolvido para localizar questões de acordo com a palavra chave informada pelo usuário, o código encontra-se disponível na figura 9. Nele é pretendido retornar as tuplas que possuam a palavra chave no enunciado da questão ou em algumas das alternativas.

Figura 9 – Utilização da cláusula like no sistema desenvolvido

```

SELECT *
FROM T_Questao
WHERE ( Texto like '%' + @PalavraChave + '%' )
OR ( OpcaoA like '%' + @PalavraChave + '%' )
OR ( OpcaoB like '%' + @PalavraChave + '%' )
OR ( OpcaoC like '%' + @PalavraChave + '%' )
OR ( OpcaoD like '%' + @PalavraChave + '%' )
OR ( OpcaoE like '%' + @PalavraChave + '%' )

```

Fonte: Documentação do sistema

Stored Procedures

As *Stored Procedures* têm a característica de conter uma coleção de comandos SQL com o objetivo de encapsular tarefas repetitivas, elas são identificadas por um nome que pode ser invocado tanto pelo próprio SGBD quanto por um sistema que faça interface com o mesmo. Nela podemos utilizar estruturas de decisão e controle, típicas de linguagem de programação, o que possibilita inserirmos a lógica de negócio no banco de dados.

A sua utilização proporciona a vantagem de se reduzir o tráfego na rede, como também aumentar a segurança do banco de dados, pois os usuários não terão acesso diretamente às tabelas e sim acesso as procedures que interagiram com as tabelas. Seu uso também impede o ataque do tipo SQL *injection*, no qual o usuário malicioso insere um comando SQL em um formulário de entrada, alterando a integridade do SGBD.

Linguagem ASP

A linguagem ASP, proporciona o desenvolvimento de sistemas com funcionamento WEB, diferentemente de outras linguagens, esta não é compilada diretamente para geração de um executável. O seu código fonte deve ser colocado no servidor para responder a solicitações dos clientes através do navegador WEB. O responsável pela publicação de um sistema em ASP é o IIS, disponível em sistemas operacionais como no Windows® 2003 Server.

Veja a definição de Mitchell (2000, p 4)

Active Server Pages contêm duas partes: código programático e HTML embutido. O código programático pode ser escrito em um número de linguagens de script. Uma linguagem de script é uma sintaxe particular utilizada para executar comandos em um computador.

Na utilização da linguagem é necessária a definição de qual script será utilizado, as possibilidades são: VBScript, JScript, PerlScript, e Phyton. As páginas do sistema desenvolvido foram todas implementadas em asp, servindo basicamente como uma interface gráfica com usuários.

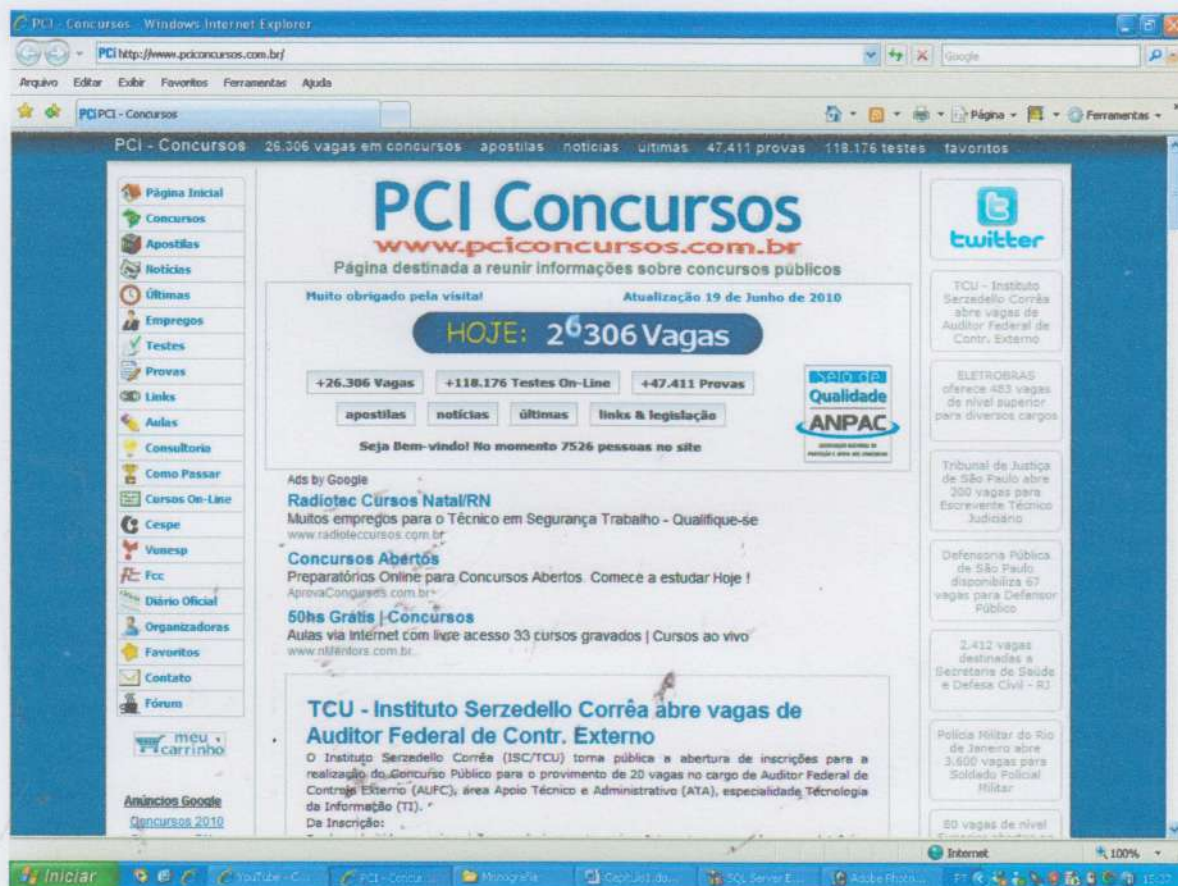
3 METODOLOGIA

Para atingir a meta de desenvolver o sistema de resolução de questões, foi necessário inicialmente avaliar as funcionalidades de sistemas já disponíveis na internet, os quais possuem comportamentos semelhantes ao do objetivo deste trabalho, estes foram utilizados como referencia para o desenvolvimento do sistema proposto. Foram escolhidos dois sistemas para análise, os quais são considerados os mais populares da internet e possuem bastantes usuários que os utilizam. Em seguida foram criados os diagramas de caso de uso e as suas descrições, com o objetivo de apresentar o seu comportamento. São apresentadas também as vantagens e desvantagens de cada sistema.

3.1 PCI CONCURSOS

O primeiro sistema encontra-se disponível na internet, através do endereço eletrônico <http://www.pciconcursos.com.br/testes/>, o qual é bastante conhecido pelos pretendentes de uma carreira no serviço público, e pode ser visualizado na figura 10.

Figura 10 – Pci concursos



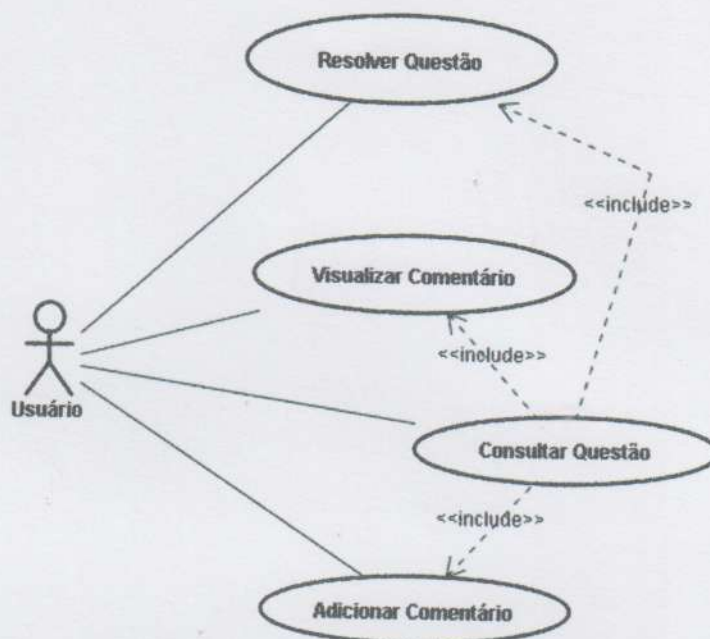
Fonte: www.pciconcursos.com.br

Além da funcionalidade de resolução de questões, o site também disponibiliza diversos outros serviços úteis, como a listagem dos concursos que estão com inscrição em aberto, divididos por regiões e estados.

Para a resolução de questões o usuário deverá inicialmente escolher o link teste disponível na página inicial do site, logo em seguida são disponibilizados os assuntos disponíveis de questões cadastradas, o usuário escolhe o assunto e o sistema apresenta as questões disponíveis no qual o mesmo pode resolvê-las,

comenta-lás ou visualizar os comentários já existentes; ao resolver as questões o sistema exibe gabarito e o seu desempenho. Em termos de UML o sistema está representado na figura 11.

Figura 11- Diagrama de caso de uso do pci concursos



Fonte: Documentação do sistema

No diagrama apresentado é possível visualizar 4 casos de uso e um ator. Para resolver questão, visualizar comentário e adicionar comentário é necessário inicialmente efetuar a consulta a qual é feita através da escolha do assunto e esta representada no diagrama pelo relacionamento de `<<include>>`. As linhas apresentadas no diagrama que não possuem direção unindo o ator ao caso de são do tipo comunicação, indicando que o ator efetua interações com sistema através dos casos de uso que estão relacionados, ou seja no diagrama apresentado o usuário interage com todos os casos de uso apresentados, podendo efetuar apenas a consulta caso deseje.

3.1.1 Descrição dos casos de uso do Pci concursos

Para uma melhor compreensão dos casos de uso é apresentada a descrição funcional dos principais casos de uso do sistema em avaliação, a UML não definiu padrão para a documentação de casos de uso; logo cada desenvolvedor poderá utilizar o seu modelo (MARTINS, 2007). Abaixo se encontra um modelo que poderá ser utilizado na descrição dos casos de uso.

Consultar Questão

A. Fluxo normal

- A.1) O sistema exibe assuntos agrupados por matérias.
- A.2) O usuário escolhe uma matéria.
- A.3) O sistema disponibiliza dez questões da referida matéria, com a opção de resolver, comentar e visualizar os comentários.

Resolver Questão

- B.1) Após efetuar a consulta o usuário escolhe a opção resolver.
- B.2) O sistema verifica as questões corretas mostrando os erros do usuário, a porcentagem de acerto e os botões voltar, gerar outro teste e as opções dos assuntos disponíveis.
- B.3) O usuário escolhe a opção gerar outro teste.
- B.4) O sistema retorna ao passo A.3.

Visualizar comentário

- C.1) No passo A.3 o usuário escolhe a opção visualizar comentário.
- C.2) O sistema disponibiliza todos os comentários já inseridos por outros usuários
- C.3) O usuário vai para a próxima questão.

3.1.2 Vantagens do sistema

- O sistema disponibiliza a opção de visualizar comentários, o que pode auxiliar nos estudos do usuário, pois ao errar uma questão ele tem a opção de verificar comentários de outros estudantes e possivelmente verificar qual o foi o seu erro.

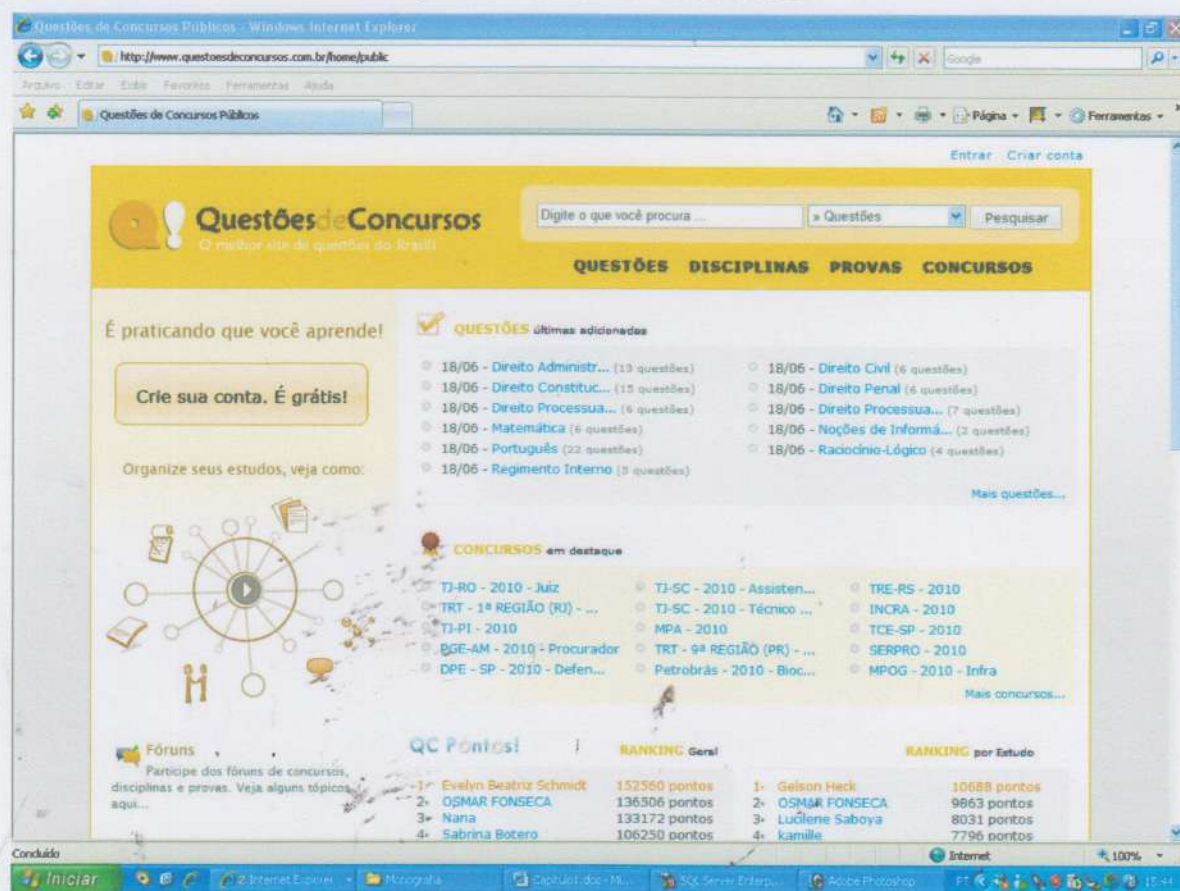
3.1.3 Desvantagens do sistema

- Não é possível escolher questões de acordo com a banca organizadora, o que impossibilita um estudo direcionado, já que as mesmas fazem suas elaborações de formas distintas, caso o usuário pretenda fazer um concurso realizado pela banca examinadora CESPE, ele gostará de resolver questões apenas dela.

3.2 QUESTÕES DE CONCURSO

Na figura 12, vemos o site que disponibiliza o segundo sistema a ser avaliado que se encontra disponível no endereço <http://www.questoesdeconcursos.com.br>.

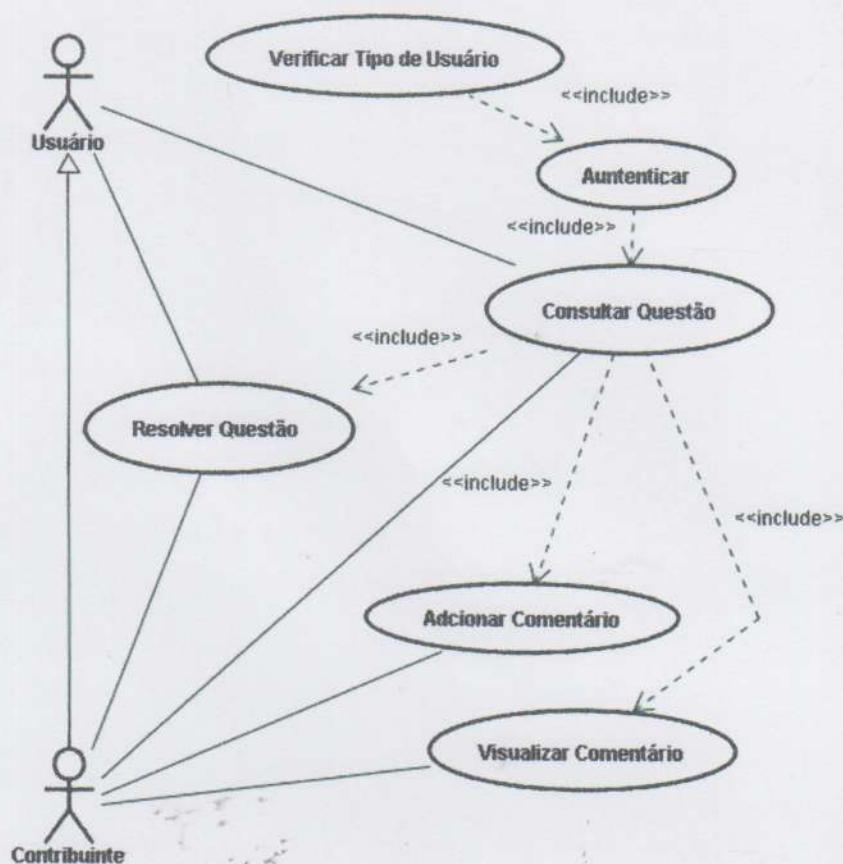
Figura 12 - Questões de concursos



Fonte: www.questoesdeconcursos.com.br

Ao contrário do sistema anterior, este possui dois tipos de atores, o usuário comum e o contribuinte, este precisa aderir a algum plano comercial oferecido pelo site para poder utilizar todas as funcionalidades do sistema como também resolver questões de forma ilimitada. O sistema encontra-se representado no diagrama da figura 13.

Figura 13- Diagrama de caso de uso do Questões de concursos



Fonte: Documentação do sistema

Nele podemos visualizar os atores através de uma hierarquia de atores, logo o Contribuinte herda os poderes e responsabilidades do Usuário, ambos se relacionam com os mesmos casos de uso porém com comportamentos um pouco diferentes. O usuário tem a limitação de só poder resolver 10 questões por dia, como também não pode utilizar de alguns filtros disponibilizados no caso de uso de resolução de questão. Para efetuar a consulta inicial é necessário estar autenticado, ao efetuar a autenticação o sistema verifica o tipo de usuário para definir as restrições necessárias.

3.2.1 Descrição alguns caso de uso do Questões de concursos

Seguindo a mesma linha de raciocínio da análise feita no sistema anterior, serão apresentados a descrição de alguns casos de uso.

Consultar Questão

A. Fluxo normal

A.1) O sistema exibe um campo para usuário digitar o que procura e algumas de opções de filtro como banca e ano da questão.

A.2) O usuário digita uma palavra chave e clica em filtrar.

A.3) O sistema disponibiliza 5 questões por página, utilizando-se de paginação caso necessário.

B. Fluxo alternativo (Usuário não está logado)

B.1) No passo A.2 o sistema detecta que o usuário não está logado e oferece duas opções: Faça o seu *login* ou crie sua conta de acesso.

B.2) O usuário escolhe a opção de fazer o *login*.

B.3) O sistema exibe a tela com o campo Usuário ou *E-mail* e o campo senha.

B.4) O usuário preenche os campos e clica em entrar.

B.5) Ir para o passo A.1

Resolver Questão

B. Fluxo normal

B.1) Após efetuar a consulta o usuário escolhe a opção resolver.

B.2) Se a opção esteve correta o sistema exibe "Parabéns você acertou a questão" se não exibe "Você errou" e indica a opção correta.

C. Fluxo alternativo (Limite ultrapassado)

C.1) No passo B.1 o sistema verifica que usuário autenticado não é contribuinte e já realizou dez questões e exibe a mensagem "Você só pode resolver dez questões por dia, para acesso ilimitado é necessário se tornar contribuinte.

C.2) Retorna ao passo A.3.

3.2.2 Vantagens do sistema

- Existem várias opções de filtros como banca, instituição, ano, cargo, nível, modalidade, disciplina dificuldade, área de formação e área de atuação.
- Possui filtro por dificuldade apresentando as opções muito fácil, fácil, médio, difícil e muito difícil, o qual permite um estudo de acordo com nível de conhecimento podendo inicialmente resolver as questões mais fáceis e à medida que for evoluindo o seu estudo mudar o nível das questões.

- O sistema disponibiliza as estatísticas de .resoluções informando a quantidade total de usuários que resolveram a questão, dividido pelas alternativas disponíveis, podendo assim ver o índice de erros dos demais usuários, havendo a possibilidade de verificar os erros comuns dos candidatos.

3.2.3 Desvantagens do sistema

- O usuário só pode resolver dez questões por dia, para liberar essa limitação é necessário se tornar um colaborador contribuinte, onde é necessário um pagamento mensal de R\$ 8,90.
- Não são sugeridas outras opções de chave de busca caso não seja encontrada nenhuma questão, por exemplo se usuário digitar a palavra chave PLSQL o sistema retorna a mensagem de que não há nenhum registro, porém se buscarmos PL/SQL o sistema retorna 17 questões. Logo deveria existir uma mensagem de do tipo Você quis dizer PL/SQL indicando ao usuário a forma correta de busca, semelhante ao comportamento da busca do google.

4 O SISTEMA DESENVOLVIDO

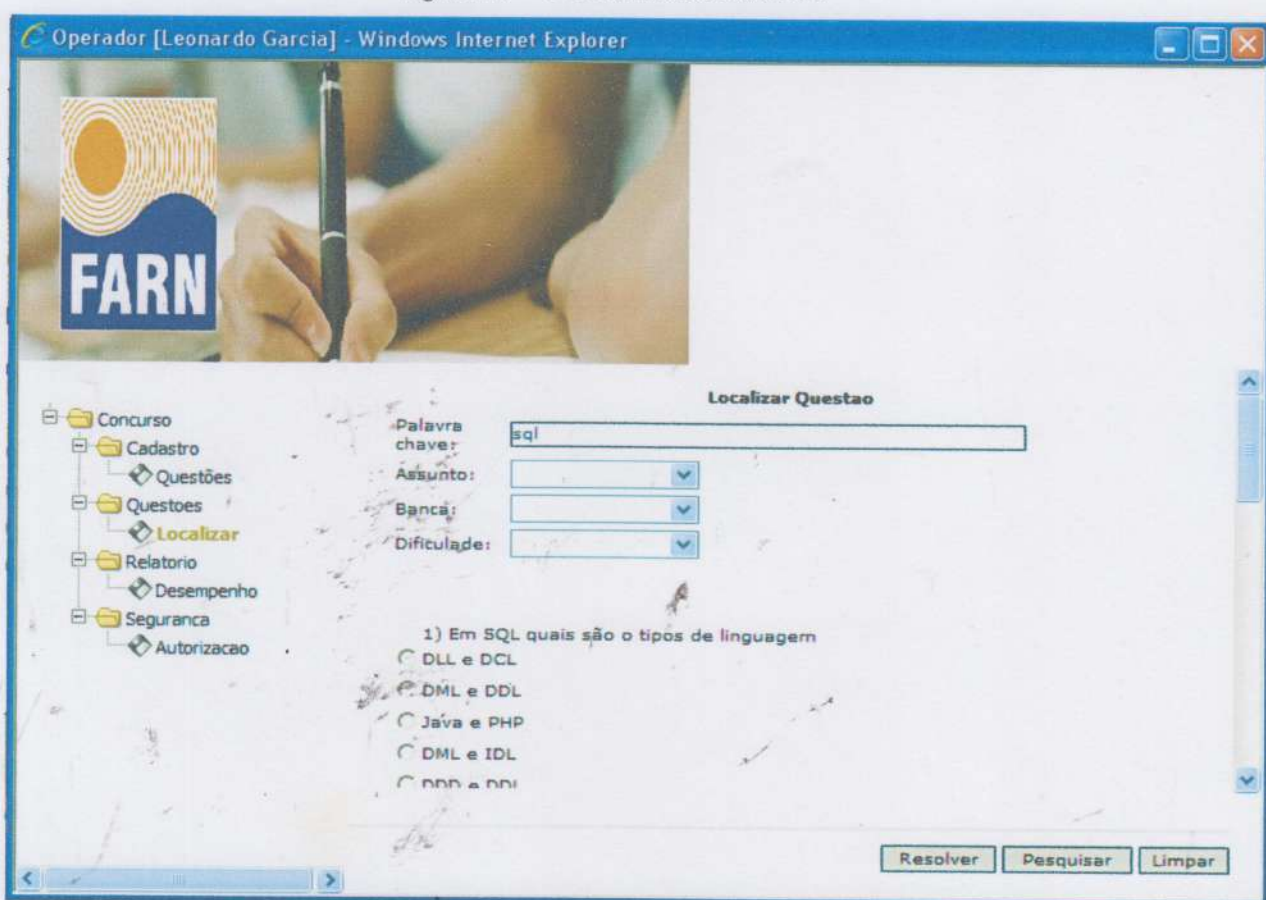
O sistema desenvolvido para esta monografia, tem como principal funcionalidade a resolução de questões, ele será disponibilizado na internet e poderá ser utilizado por estudantes que desejem avaliar seus conhecimentos através da aplicação prática de como determinados assuntos são exigidos. Será oferecida também a opção de visualização de desempenho, possibilitando a comparação entre todos os usuários do sistema.

Assim estudantes que desejem se preparar para concursos públicos ou apenas para realização de uma prova da universidade poderão utilizar o sistema

complementando o seu estudo, ele poderá estudar uma determinada disciplina e em seguida verificar como provavelmente será cobrado aquele assunto na hora da prova, através da resolução de questões da disciplina.

As funcionalidades existentes no sistema desenvolvido, foram elaboradas principalmente a partir da análise efetuada nos sistemas anteriores, o aplicativo desenvolvido para resolução de questões pode ser visualizado na figura 14, ele apresenta quatro telas de interface com o usuário: cadastro de questões, localizar questões, relatório de desempenho e autorização.

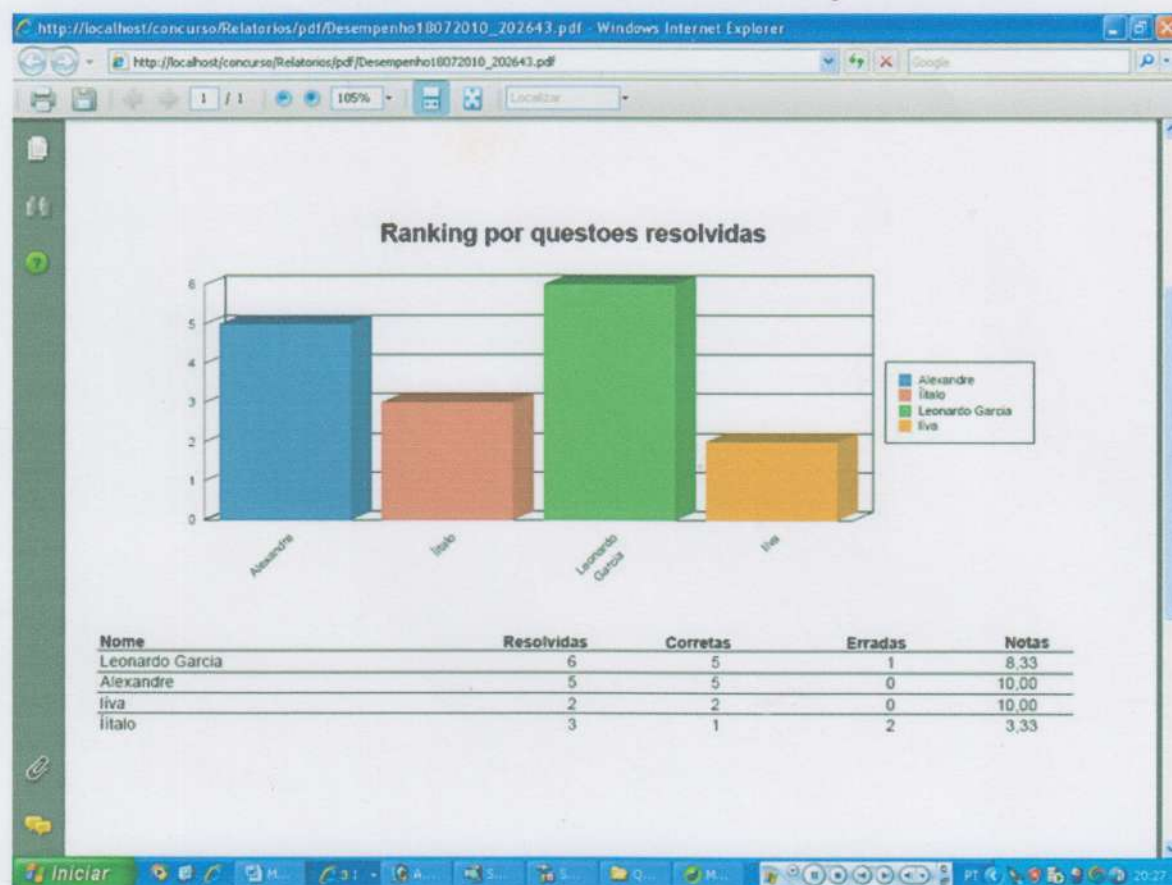
Figura 14 – O sistema desenvolvido



Fonte: Documentação do sistema

O sistema tem a funcionalidade de localizar as questões cadastradas de acordo com a palavra chave informada, como também de exibir um relatório de avaliação de desempenho mostrando todos os usuários que resolveram questões e suas respectivas notas em um gráfico que está exposto na figura 1.

Figura 15- Relatório de desempenho dos usuários



Fonte: Documentação do sistema

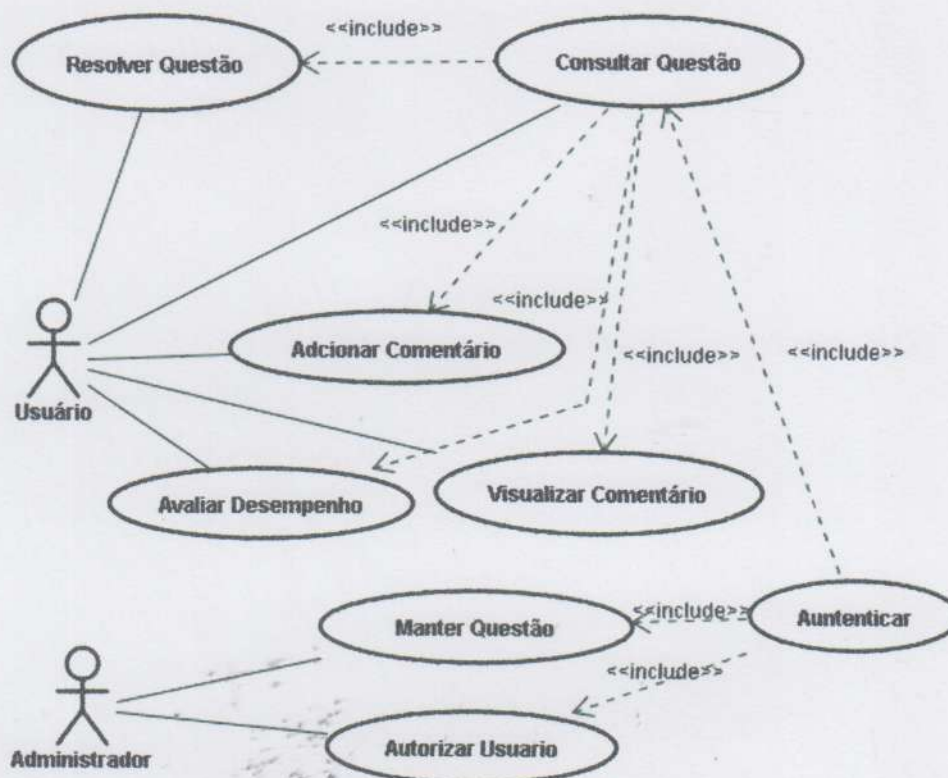
Neste relatório de Avaliação de Desempenho gerado pelo sistema é exibido um gráfico mostrando os usuários e a quantidade de questões resolvidas e abaixo existe uma lista ordenada pela quantidade de questões corretas. Na tela de autorização o ator “administrador” define o que os usuários terão direito a utilizar no sistema.

4.1 DIAGRAMA DE CASOS DE USO

O sistema desenvolvido possui dois atores o “usuário” e o “administrador”, o segundo é responsável por cadastrar as questões como também autorizar as funcionalidades que o usuário terá permissão. O usuário poderá resolver questões, para isso será necessária inicialmente a consulta ao banco de dados da palavra chave informada e os filtros desejados, retornando as questões desejadas

para resolução. O diagrama contido na figura 16 apresenta os seus casos de uso e atores.

Figura 16 – Diagrama de casos de uso do sistema desenvolvido



Fonte: Documentação do sistema

4.1.1 Descrição alguns casos de uso do sistema desenvolvido

Consultar Questão

A. Fluxo normal

A.1) O sistema exibe um campo para usuário digitar a palavra chave que procura e algumas de opções de filtro como banca e ano da questão e dificuldade.

A.2) O usuário digita uma palavra chave e clica em pesquisar.

A.3) O sistema disponibiliza as questões encontradas para resolução.

B. Fluxo alternativo (Questão não encontrada)

B.1) No passo A.2 o sistema verifica que não existe questão correspondente a palavra chave e exibe a mensagem que nenhuma questão foi encontrada e sugere opções de busca.

B.2) Ir para o passo A.1

Resolver Questão

A.1) no passo A.3 o usuário escolhe as questões que deseja e seleciona a opção resolver.

A.2) O sistema efetua a correção disponibilizando o resultado e atualizando o desempenho do usuário.

Avaliação de desempenho

A. Fluxo normal

A.1) O Usuário escolhe a opção "gerar relatório" na opção de Avaliação de desempenho.

A.2) O sistema exibe um relatório no qual é possível visualizar a quantidade de questões resolvidas, corretas, erradas e a respectiva nota como também um gráfico exibindo o ranking dos usuários.

Autorizar

A. Fluxo normal

A.1) O Administrador informa o usuário a ser dada autorização escolhe a opção buscar.

A.2) O sistema exibe todas as autorizações concedidas ao usuário.

A.3) O Administrador acrescenta alguma opção que o usuário não tem permissão e clica em "Autorizar".

A.4) O sistema atualiza as permissões do usuário e exibe a mensagem "Autorização concedida com sucesso".

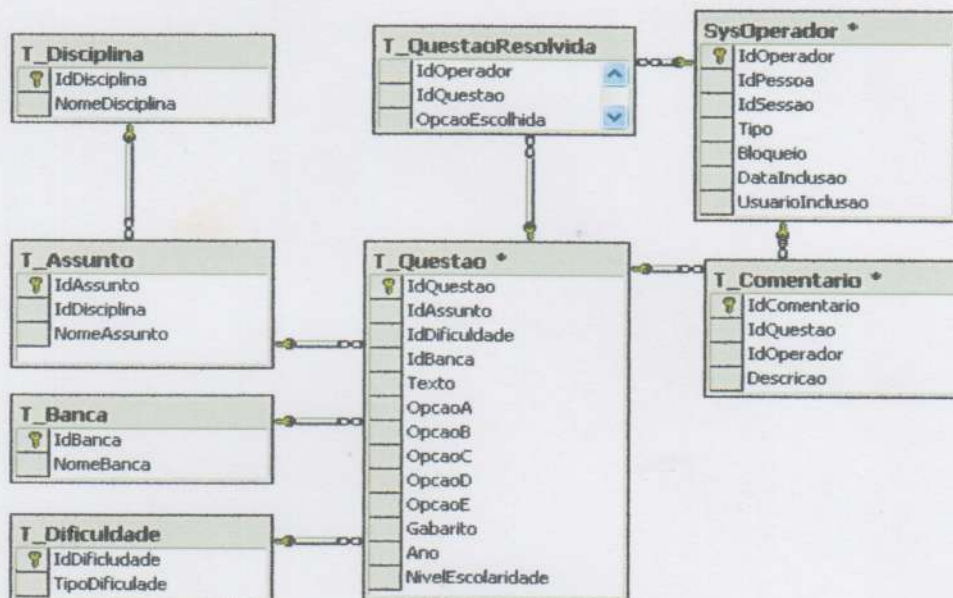
4.2 DER

O modelo de dados entidade-relacionamento (E-R) foi desenvolvido para facilitar o projeto de banco de dados, pois gera uma especificação de um esquema que representa a estrutura lógica geral de um banco de dados. São empregados três conceitos primordiais nesse modelo: conjuntos de entidades, conjuntos de relacionamentos e atributos. As entidades representam uma "coisa" ou "objeto" do mundo real, que é distinguível de todos os outros objetos, o relacionamento representa uma associação entre várias entidades, por fim o atributo representa uma função que mapeia o conjunto de entidades para um domínio (SILBERSCHATZ et al, 2006).

Para o sistema desenvolvido foram criados as entidades, relacionamentos e atributos exibidos na figura 17. Nele podemos visualizar que as entidades

possuem chaves primárias e chaves estrangeiras que garantem a integridade referencial do banco.

Figura 17 – Diagrama de entidade relacionamento do sistema



Fonte: Documentação do sistema

4.3 MODELO DE IMPLEMENTAÇÃO

Para o modelo escolhido para o desenvolvimento foram exploradas ao máximo as funcionalidades oferecidas pelo SGBD, de tal forma que as páginas desenvolvidas em ASP servem, basicamente, como de *interface* com o usuário, pois a maior parte do processamento ocorre no banco de dados, onde estão incluídas as regras de negócio, através de Stored Procedures, Views, etc. Para um melhor desempenho do sistema é possível dividir o processamento em servidores distintos conforme figura 18.

Figura 18 – Processamento dividido em servidores



Fonte: Documentação do sistema

Essa abordagem utilizada neste trabalho é atípica ao desenvolvimento tradicional onde o Banco de Dados é visto apenas como um simples repositório de dados.

Uma das vantagens desse modelo é a possibilidade de uma futura migração do sistema para outra linguagem de programação como ASP.NET ou PHP, utilizando um esforço menor de trabalho pois como as regras estão no banco, apenas as telas precisaram ser criadas. As regras de negócio correspondem a Stored Procedures, o administrador define quais procedures o usuário terá privilégio de utilizar, através das instruções *grant execute* e *revoke execute* é possível dar permissão e revogação. Essa metodologia garante que mesmo que um usuário com um pouco mais de conhecimento em TI, que tente acessar o banco diretamente, burlando o sistema, estará limitado a fazer apenas o que teria autorização, se efetuasse o procedimento correto de utilização.

4.4 MAPEAMENTOS DOS CASOS DE USO, TELAS E STORED PROCEDURES

A partir dos casos de uso do sistema desenvolvido foram elaboradas as telas e as procedures correspondentes para oferecer as funcionalidades pretendidas, o mapeamento desses componentes encontra-se relacionado na tabela 1.

Tabela 1 - Mapeamento dos casos de uso, telas e stored procedure

TELA:	Cadastro de questão	
CASO DE USO	PROCEDURE	OBJETIVO DA PROCEDURE
Manter questão	stp_Questao_Ins	Incluir Questão.
	stp_Questao_Upd	Alterar Questão.
	stp_Questao_Del	Excluir Questão.
	stp_Questao_Sel	Selecionar Questão para ser excluída ou alterada.
TELA:	Localizar Questão	
CASO DE USO	PROCEDURE	OBJETIVO DA PROCEDURE
Consultar questão	stp_LocalizarQuestao_Sel	Retornar as questões localizadas no banco de dados de acordo com a palavra chave e os filtros informados.
Resolver questão	stp_ResolverQuestao_Upd	Inseção no histórico de questões resolvidas e retornar o gabarito ao usuário.
Visualizar comentário	stp_Comentario_Sel	Retorna todos os comentários cadastrados para uma determinada questão.
Adicionar comentário	stp_Comentario_Ins	Incluir novo comentário para a questão.
TELA:	Relatório de desempenho	
CASO DE USO	PROCEDURE	OBJETIVO DA PROCEDURE
Avaliar Desempenho	stp_Desempenho_Rel	Retornar os dados para ser gerado um gráfico mostrando os usuários e a quantidade de questões resolvidas.
TELA:	Autorização	
CASO DE USO	PROCEDURE	OBJETIVO DA PROCEDURE
Autorizar Usuário	stp_Autorizacao_Ins	Incluir autorização, é necessário o comando GRANT.
	stp_Autorizacao_Del	Excluir autorização, é necessário o comando REVOKE.
	stp_Autorizacao_Sel	Selecionar as autorizações que o usuário possui.
TELA:	Autenticação	
CASO DE USO	PROCEDURE	OBJETIVO DA PROCEDURE
Autenticar	stp_Autenticacao_Upd	Autentica o usuário na aplicação.

Para a demonstração da ligação existente entre as telas, casos de uso e stored procedures; será apresentado o funcionamento interno do sistema quando o se deseja incluir uma nova questão. Para ser inserida uma questão é necessário que o "Administrador" informe os campos disponíveis no formulário da tela, que pode ser visto na figura 19. Logo após o "Administrador" preencher as informações referentes a uma determinada questão, ele selecionará o botão Incluir, deste modo a página desenvolvida em ASP receberá estas informações preenchidas e invocará a função

Manutenção, disponível na figura 20, que determinará a procedure referente à inclusão "stp_Questao_Ins" e montará a string a ser enviada ao banco de dados.

Figura 19 – Tela de Cadastro de questões.

Cadastro de Questões

	Enunciado:	
	Opção A:	
	Opção B:	
	Opção C:	
	Opção D:	
	Opção E:	
	Gabarito:	
	Disciplina:	
	Assunto:	
	Banca:	
	Dificuldade:	
	Ano:	
	Nível de Escolaridade:	

Fonte: Documentação do sistema

Figura 20 – Função Manutenção feita em ASP que invoca a procedure.

```

Function Manutencao
'NESTE MOMENTO É DEFINIDO AS PROCEDURE A SER INVOCADA
if Acao = "I" then      NovoCmd("stp_Questao_Ins")      end if
if Acao = "A" then      NovoCmd("stp_Questao_Upd")      end if
if Acao = "E" then      NovoCmd("stp_Questao_Del")      end if
'PARAMETROS DA PROCEDURE
Cmd.Parameters.Item("@IdQuestao").value                = Checanulo(IdQuestao)
Cmd.Parameters.Item("@IdAssunto").value                = Checanulo(IdAssunto)
Cmd.Parameters.Item("@IdDificuldade").value            = Checanulo(IdDificuldade)
Cmd.Parameters.Item("@IdBanca").value                  = Checanulo(IdBanca)
Cmd.Parameters.Item("@Texto").value                    = Checanulo(Erunciado)
Cmd.Parameters.Item("@OpcaoA").value                   = Checanulo(OpcaoA)
Cmd.Parameters.Item("@OpcaoB").value                   = Checanulo(OpcaoB)
Cmd.Parameters.Item("@OpcaoC").value                   = Checanulo(OpcaoC)
Cmd.Parameters.Item("@OpcaoD").value                   = Checanulo(OpcaoD)
Cmd.Parameters.Item("@OpcaoE").value                   = Checanulo(OpcaoE)
Cmd.Parameters.Item("@Gabarito").value                 = Checanulo(Gabarito)
Cmd.Parameters.Item("@Ano").value                      = Checanulo(Ano)
Cmd.Parameters.Item("@NivelEscolaridade").value        = Checanulo(NivelEscolaridade)
'CASO A PROCEDURE RETORNE UM MENSAGEM DE ERRO, EXIBA A MENSAGEM AO USUÁRIO
On Error Resume Next
    Cmd.Execute
    If Err.Number <> 0 Then
        Manutencao = "<font color='#FF0000'>" & Err.Description & "</font>"
    End If
On Error Goto 0
Set Cmd = Nothing
End Function

```

Fonte: Documentação do sistema

Em seguida o banco de dados receberá os valores passados na string pela aplicação e começará a sua execução. Na figura 21 está contido o código comentado da procedure "stp_questao_ins", nela inicialmente estão apresentados o seu nome e os seus parâmetros que correspondem aos enviados pela página ASP, em seguida existem críticas que indicaram a inserção ou não da questão de acordo com a regra de negócio, caso seja aprovada a inserção do registro, será executado um comando DML para inserção de uma tupla.

Figura 21 – Stored Procedure de inclusão de questão, as palavras com a coloração verde representam comentário.

```
--nome da procedure
alter procedure stp_questao_ins
-- informa-se os parametros de entrada e os seus respectivos tipos de dados
(
    @idquestao int, @idassunto int, @iddificuldade int, @idbanca int,
    @texto varchar(1000), @opcaoa varchar(200), @opcaob varchar(200),
    @opcaoc varchar(200), @opcaod varchar(200), @opcae varchar(200),
    @gabarito char(1), @ano int, @nivelescolaridade char(1)
)
as
begin
    -----
    -- criticas
    -----

    --obriga o usuário a informar o texto
    if @texto is null
    begin
        --caso o texto não tenha sido informado, será retornada a mensagem abaixo para a aplicação asp
        -- e será interrompida a execução da procedure
        raiserror 13000 'informe o texto da questão.'
        return
    end

    -- neste momento é inserida uma regra de negócio, que não permite a inclusão de mais de uma questão com o mesmo
    -- texto
    if exists( select * from t_questao where texto = @texto)
    begin
        raiserror 13000 'não é possível incluir a questão com o texto informado, pois já existe outra
        questão cadastrada.'
        return
    end

    -----
    -- inseri a questão no banco de dados, com os valores passados como parametros
    -----

    insert into t_questao
    (
        idquestao, idassunto, iddificuldade, idbanca, texto, opcaoa, opcaob, opcaoc, opcaod, opcae, gabarito,
        ano, nivelescolaridade
    )
    values
    (
        @idquestao, @idassunto, @iddificuldade, @idbanca, @texto, @opcaoa, @opcaob, @opcaoc, @opcaod, @opcae, @gabarito,
        @ano, @nivelescolaridade
    )
end
```

Fonte: Documentação do sistema.

Assim como na inserção, a alteração e a exclusão funcionam de maneira semelhante, porém utilizarão procedures diferentes, que estão disponíveis nas figuras 22 e 23. Na procedure vista na figura 24 “stp_Cad_Questao_Sel”, diferentemente das anteriores, esta retorna dados para aplicação através da instrução *select*, que retornará todos os dados de uma determinada questão, para que a mesma possa ser alterada ou excluída.

Na estrutura de uma *stored procedure*, é opcional a existência de parâmetros de entrada, que pode ser visto na procedure “stp_Desempenho_Rel”, a qual não possui parâmetros, e é utilizada para geração do relatório de desempenho de todos os usuários, ela pode ser visualizada na figura 25

Figura 22 – Stored Procedure de alteração de questão.

```

--nome da procedure
alter procedure stp_questao_upd
-- informa-se os parametros de entrada e os seus respectivos tipos de dados
(
    @idquestao int, @idassunto int, @iddificuldade int, @idbanca int,
    @texto varchar(1000), @opcaoa varchar(200), @opcaob varchar(200),
    @opcaoc varchar(200), @opcaod varchar(200), @opcae varchar(200),
    @gabarito char(1), @ano int, @nivelescolaridade char(1)
)
as
begin

-----
-- criticas
-----

--obriga o usuário a informar o texto
if @texto is null
begin
    --caso o texto não tenha sido informado, será retornada a mensagem abaixo para a aplicação asp
    -- e será interrompida a execução da procedure
    raiserror 13000 'informe o texto da questão.'
    return
end

-- neste momento é inserida uma regra de negócio, que não permite a inclusão de mais de uma questão com o mesmo
-- texto
if exists( select * from t_questao where texto = @texto and idquestao <> @idquestao)
begin
    raiserror 13000 'não é possível incluir a questão com o texto informado, pois já existe outra
    questão cadastrada.'
    return
end

-----
-- altera a questão no banco de dados, com os valores passados como parametros
-----

update t_questao
set idassunto = @idassunto, iddificuldade = @iddificuldade, idbanca = @idbanca, texto = @texto, opcaoa = @opcaoa,
    opcaob = @opcaob, opcaoc = @opcaoc, opcaod = @opcaod, opcae = @opcae, gabarito = @gabarito, ano = @ano,
    nivelescolaridade = @nivelescolaridade
where idquestao = @idquestao

end

```

Fonte: Documentação do sistema.

Figura 23 – Stored Procedure de exclusão de questão.

```

--nome da procedure
alter procedure stp_questao_del
-- informa-se os parametros de entrada e os seus respectivos tipos de dados
(
    @idquestao int, @idassunto int, @iddificuldade int, @idbanca int,
    @texto varchar(1000), @opcaoa varchar(200), @opcaob varchar(200),
    @opcaoc varchar(200), @opcaod varchar(200), @opcae varchar(200),
    @gabarito char(1), @ano int, @nivelescolaridade char(1)
)
as
begin

-----
-- criticas
-----

-- caso a questão já tenha sido resolvida por algum usuário, ela não poderar ser excluida
if exists( select * from t_questaoresolvida where idquestao <> @idquestao)
    begin
        raiserror 13000 'não é possível excluir questão, pois ela já foi resolvida por algum usuário.'
        return
    end

-----
-- exclui a questão no banco de dados
-----

delete t_questao

where idquestao = @idquestao

end

```

Fonte: Documentação do sistema.

Figura 24 – Stored Procedure de consulta de questão.

```

--nome da procedure
create procedure stp_questao_sel
(
    @idquestao int
)
as
begin

--retorna os dados da questão de acordo com a sua chave
select *

from t_questao

where idquestao = @idquestao

end

```

Fonte: Documentação do sistema.

Figura 25 - Stored Procedure que retorna o desempenho dos usuários.

```
--nome da procedure
alter procedure stp_desempenho_rel
as
begin

--efetua o processamento para indicar a quantidade de
--questões resolvidas, corretas e erradas
select pe.nome,
       ( select isnull(count(idquestao),0)
         from t_questaoresolvida qr
         where qr.idoperador = op.idoperador
       ) as questoesresolvidas,
       ( select isnull(count(idquestao),0)
         from t_questaoresolvida qr
         where qr.idoperador = op.idoperador
         and opcaoescolhida = opcao correta
       ) as questoescorretas,
       ( select isnull(count(idquestao),0)
         from t_questaoresolvida qr
         where qr.idoperador = op.idoperador
         and opcaoescolhida <> opcao correta
       ) as questoeserradas

from sysoperador op

inner join genpessoa pe on pe.idpessoa = op.idpessoa

where exists( select idquestao
              from t_questaoresolvida
              where idoperador = op.idoperador)

order by questoescorretas desc
```

Fonte: Documentação do sistema.

4.5 SEGURANÇA

O modelo de desenvolvimento utilizado preocupou-se com a segurança da informação, pois, apesar de não ser um fator crucial no sistema apresentado, este mesmo modelo poderá ser utilizado em novas aplicações em que envolvam transações comerciais.

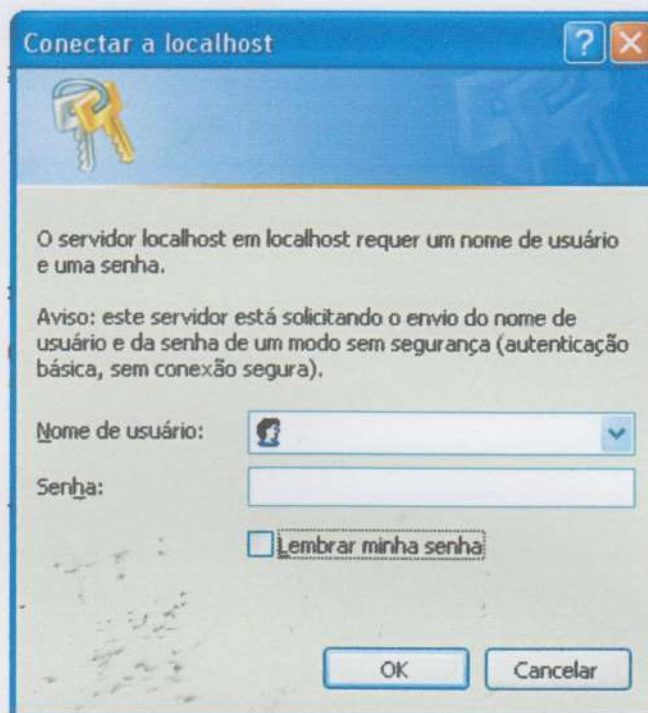
Para acessar o sistema é necessário que o usuário esteja cadastrado de três formas:

- No Active Directory - AD.
- No SQL Server.

- No banco de dados.

Ao digitar o endereço do sistema no navegador é solicitado o nome de usuário e senha (ver figura 26), os quais devem estar previamente cadastrados no AD que é um serviço de diretório nas redes Windows®.

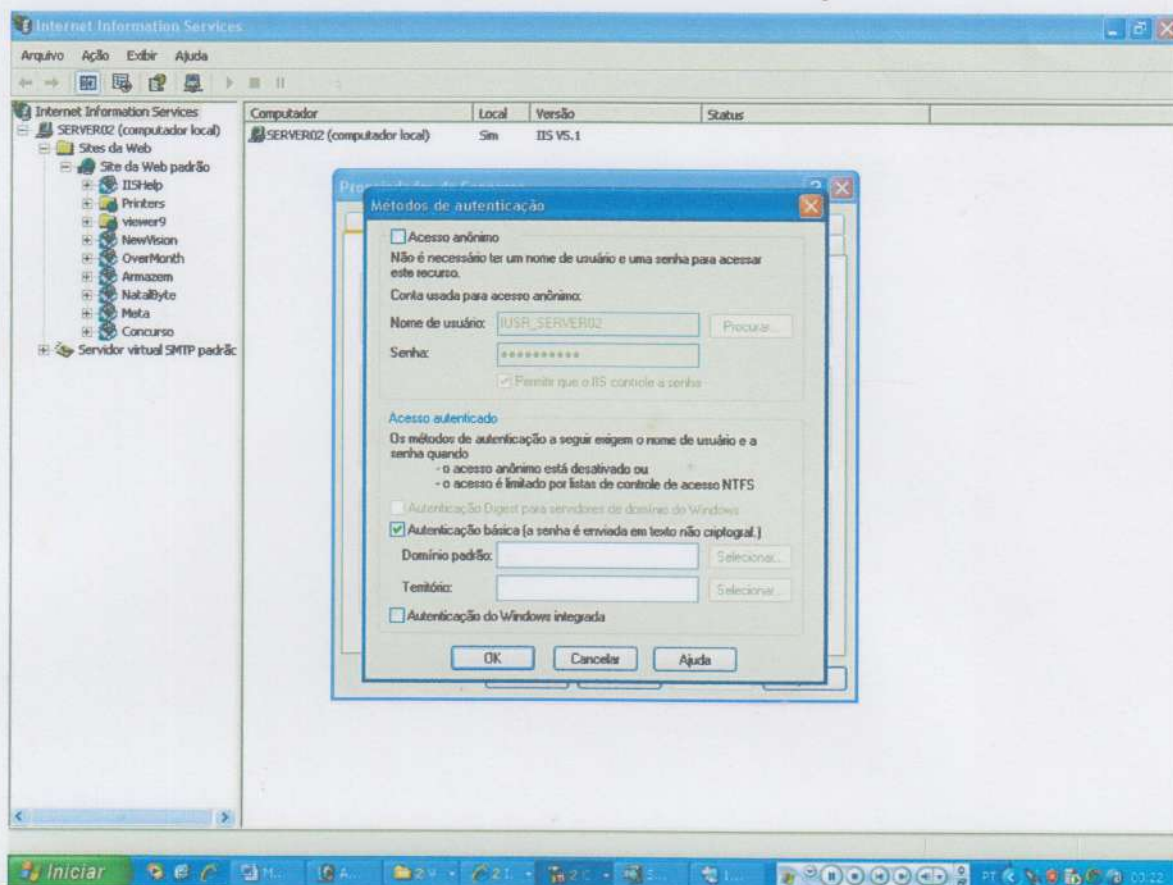
Figura 26 – Autenticação do sistema



Fonte: Documentação do sistema

Ao se desejar implantar em um sistema esse tipo de autenticação é necessário desabilitar o acesso "anônimo" e selecionar a opção "autenticação" básica no site cadastrado no IIS como pode ser visto na figura 27.

Figura 27 – Desabilitando o acesso anônimo no IIS



Fonte: Documentação do sistema

Após a primeira validação o sistema verifica se existe um usuário cadastrado no SQL Server. Para efetuar o cadastro de novos usuários, é necessária a utilização de duas Stored Procedures já fornecidas pelo SGBD: a `sp_grantlogin` que é utilizada informando o domínio e nome do usuário a ser criado. Em seguida é necessário a procedure `sp_grantdbaccess` a qual associa o usuário a determinado Banco de Dados.

O terceiro passo é verificar se existe cadastro do usuário em uma tabela gerada na criação do banco de dados.

5 JUSTIFICATIVA PARA AS TECNOLOGIAS UTILIZADAS

Nesta seção serão apresentados os motivos que levaram a escolha das tecnologias utilizadas para o desenvolvimento do sistema.

5.1 ASP

Esta linguagem apresenta a vantagem de possuir bastante referência disponível em livros e na internet. O que facilita o desenvolvimento e possíveis soluções de problemas que venham surgir. Ela é utilizada por muitos desenvolvedores no Brasil e no mundo, os quais disponibilizam, na internet soluções para problemas por eles enfrentados, que servem de referências para outros programadores.

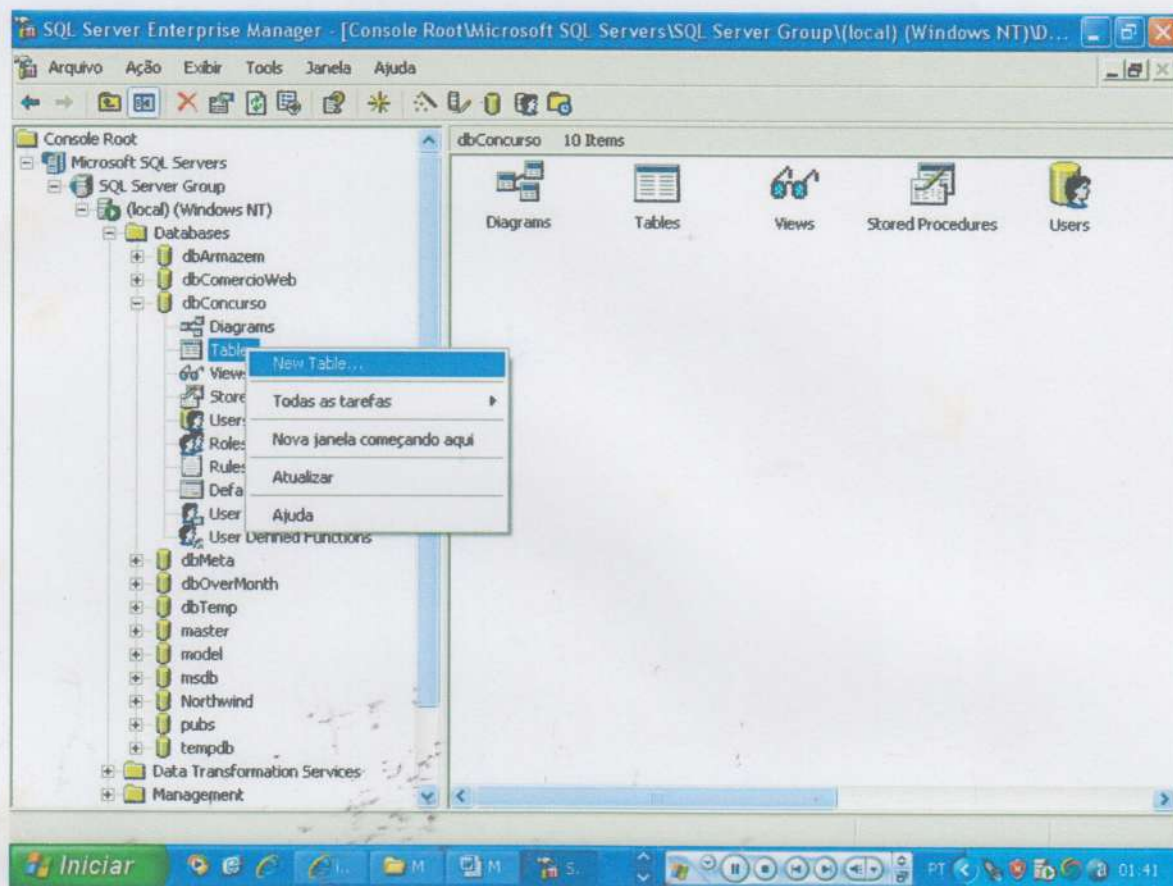
Outra vantagem é que o sistema operacional Windows® 2003 Server já disponibiliza o IIS o qual é necessário para sua utilização.

5.2 MICROSOFT SQL SERVER 2000

O SGBD escolhido é bastante robusto e confiável; garantindo controle de concorrência, restauração de falhas a partir do *log*, além de possuir bastante compatibilidade com a linguagem ASP e o Windows® 2003 Server. Nele encontram-se disponíveis duas ferramentas: o SQL *Enterprise Manager* e o SQL *Query Analyzer*. Com a utilização do *Enterprise Manager*, é possível a criação de objetos como *stored procedures*, *tables* e *views* utilizando o modo gráfico, o que torna esta criação mais ágil do que na utilização da ferramenta *Query Analyzer*, que gera os objetos através da escrita de comandos SQL. Porém, apesar de ser mais demorado o desenvolvimento através de instruções SQL, podemos explorar ao máximo as funcionalidades oferecidas por esta linguagem, o que permite a inclusão de regras

de negócio no banco de dados. Na figura 28 é possível a visualização da ferramenta *Enterprise Manager*.

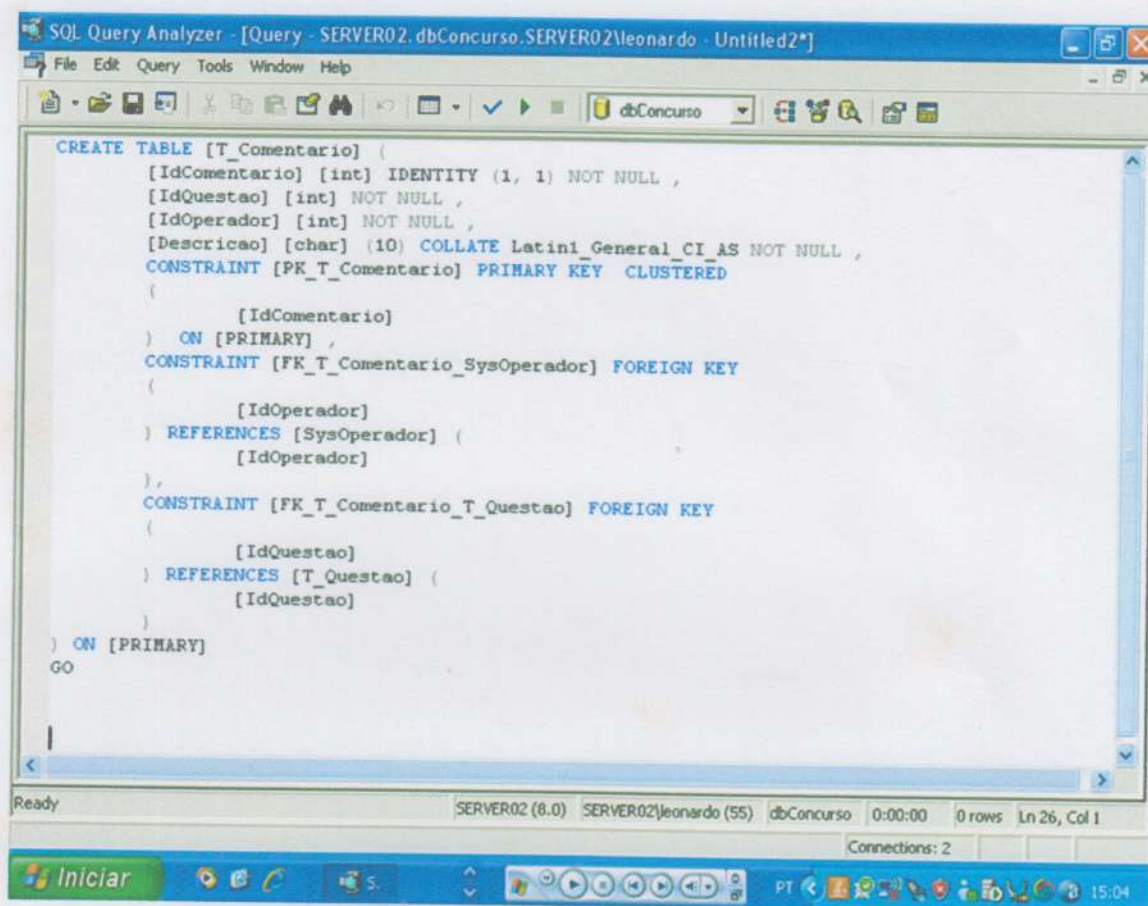
Figura 28 - Ferramenta Enterprise



Fonte: Documentação do sistema

O SQL Query Analyzer pode visualizado na figura 29, onde encontra-se um exemplo de criação de uma tabela que pertence ao sistema desenvolvidos.

Figura 29 – Ferramenta Query Analyzer



Fonte: Documentação do sistema

6 CONCLUSÃO

Através do desenvolvimento do sistema de resolução de questões foi possível aplicar os conhecimentos de desenvolvimento web e de banco de dados, inicialmente através da análise em UML de sistemas já disponíveis, foram apresentadas as características deles, utilizando-as como motivação para a criação das funcionalidades do sistema proposto. Essa abordagem inicial apresenta uma possível forma de se definir as características de um sistema que se deseje implementar, não se devendo copiar sistemas, mas extrair as vantagens de cada um deles para desenvolver um produto com qualidade .

A UML também se mostrou bastante importante no desenvolvimento, pois gera um recurso gráfico no qual é possível apresentar as características de um sistema em um nível mais alto, que pode ser compreendido por diversas partes como o analista, o programador e o cliente. Através dos diagramas de casos de uso foram exibidas as funcionalidades dos sistemas analisados. Após essa documentação das características do sistema é importante a escolha das tecnologias mais adequadas e o modelo de desenvolvimento, este trabalho mostrou a vantagem no modelo escolhido, pois além de explorar as funcionalidades oferecidas pelo SGBD também facilita a migração do sistema para uma nova linguagem de programação.

Por fim o sistema desenvolvido para esta monografia será disponibilizado na internet, oferecendo uma ferramenta de auxílio para os estudantes como também poderá ser analisado em conjunto com esta monografia por futuros discentes que queiram implementar sistemas acadêmicos ou até mesmo comerciais utilizando o modelo adotado, através das tecnologias apresentadas eles poderão disponibilizá-las na internet.

Referências

- BEZERRA, Eduardo. **Princípios de análise e projeto de sistemas com UML**. Rio de Janeiro: Campus, 2007.
- BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML Guia do usuário**. Rio de Janeiro: Campus, 1999.
- DATE, C. J. **Introdução a sistemas de bancos de dados**. Rio de Janeiro: Campus, 2003.
- MARTINS, José Carlos Cordeiro. **Gerenciando projetos de desenvolvimento de software com PMI, RUP e UML**. Rio de Janeiro: Brasport, 2007.
- MITCHELL, Scott & ATKINSON, James, **Aprenda em 21 dias ASP Active Server Pages 3.0**. Rio de Janeiro: Campus, 2000.
- OLIVEIRA, Celso Henrique Poderoso. **SQL curso prático**. São Paulo: Novatec, 2002.
- SELTZER, W. Valdemar; SILVA, Flávio Soares Corrêa. **Banco de Dados**. São Paulo: Edgar Blucher, 2005.
- SILBERSCHATZ, Abraham; KORTH, Henry F. & SUDARSHAN, S. **Sistema de banco de dados**. Rio de Janeiro: Campus, 2006.