**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO RIO GRANDE DO NORTE– IFRN**

**CAMPUS JOÃO CÂMARA**

**DIREÇÃO ACADÊMICA**

**CURSO INFORMÁTICA**

**IFQuestion - Banco de questões compartilhadas**

MARIA CAROLAINE FERREIRA AGUIAR

SABRINA DA SILVA TORRES

**JOÃO CÂMARA/RN**

**2016**

MARIA CAROLAINE FERREIRA AGUIAR

SABRINA DA SILVA TORRES

**IFQuestion – Banco de questões compartilhadas**

Trabalho de Conclusão de Curso apresentado à Diretoria Acadêmica do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte – Campus João Câmara, como requisito necessário à obtenção do título de Técnico em informática.

Orientador: Prof.Me.Diego Oliveira

João Câmara/RN

2016

MARIA CAROLAINE FERREIRA AGUIAR

SABRINA DA SILVA TORRES

**IFQuestion – Banco de questões compartilhadas**

Trabalho de Conclusão de Curso apresentado à Diretoria Acadêmica do Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte – Campus João Câmara, como requisito necessário à obtenção do título de Técnico em informática.

Orientador: Prof.Me. Diego Oliveira

Aprovado em \_\_/\_\_/\_\_\_\_\_

BANCA EXAMINADORA

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Me. Diego Henrique Oliveira de Souza - Orientador

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Igor Augusto de Carvalho Alves – Examinador

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Prof. Gileno Câmara de França - Examinador

Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte.

Agradeço a Deus pela sua infinita bondade, a minha família por todo apoio e carinho, ao meu namorado Rian Phellipe que sempre me incentivou.

**Maria Carolaine Ferreira Aguiar**

A Deus por ter me concedido saúde e força para superar as dificuldades, aos meus pais pelo apoio, ao meu namorado Wendeson Araújo por acreditar na minha capacidade.

**Sabrina da Silva Torres**

**AGRADECIMENTOS**

Ao Nosso Deus

Por guiar-nos, pela sua proteção e por todas as bênçãos alcançadas.

As nossas famílias

Que sempre nos apoiaram e nos incentivaram a chegar até aqui.

Aos nossos amigos

Pelo carinho e a força para realizarmos mais esta missão.

Ao nosso professor e orientador Diego Oliveira

Sem seu apoio, atenção e confiança, não teríamos meios de concretizar este trabalho.

E a todos os professores do IFRN-João Câmara, pela paciência, respeito e atenção. Obrigado!

**DEDICATÓRIA**

Dedicamos este trabalho em especial ao nosso Deus, pois nos concedeu as condições necessárias para chegarmos até o final desta jornada. Aos nossos pais por todo incentivo, amor e confiança. Aos professores do IFRN, em especial ao professor Diego Oliveira, pela dedicação e atenção que nos proporcionaram a obtenção de bons resultados.

“Que os vossos esforços desafiem as impossibilidades, lembrai-vos de que as grandes coisas do homem foram conquistadas do que parecia impossível ”.

Charles Chaplin

“Lembre-se que as pessoas podem tirar tudo de você, menos o seu conhecimento”.

Albert Einstein

**RESUMO**

Este trabalho teve como objetivo principal a criação de um sistema de armazenamento de questões e elaboração de provas, de diversificadas disciplinas, facilitando assim o trabalho dos professores que poderão armazenar questões de um determinado assunto, assim como suas respostas.

O sistema também possuiu uma funcionalidade muito útil: a criação dinâmica de provas, na qual o docente pode escolher as questões desejadas e gerar uma prova, com cabeçalho institucional, isso vai promover aos usuários um melhor aproveitamento de seu tempo, já que a criação será automática (com base nas suas escolhas).

O IFQuestion também conta com uma outra funcionalidade, a integração dos alunos com os professores, através desta função os discentes poderão visualizar as questões cadastradas pelos docentes e com isso estudar para a prova, porém com níveis de acessos diferentes, para assim garantir a segurança dos dados, por exemplo: os professores poderão cadastrar, pesquisar e alterar questões, e criar avaliações escritas, já os alunos irão somente visualizar e pesquisar questões.

O projeto foi desenvolvido utilizando a linguagem de programação *Java Server Pages* (JSP) que é uma tecnologia que ajuda os desenvolvedores de software a criarem páginas *web* geradas dinamicamente baseadas em HTML, além disto, foi utilizado SQL (*Structured Query Language*) - uma linguagem padrão universal para manipulação bancos de dados - as plataformas utilizadas foram a base de dados *MySQL* juntamente com a interface *Heidi*, *Netbeans,* e o *Tomcat*.

**Palavras-chave**: Armazenamento de questões, elaboração de provas, sistema.

**LISTA DE FIGURAS**

[Figura 1. Java Server Pages (JSP) no IFQuestion 1](#_Toc446711494)7

[Figura 2: Banco de dados MySQL com a interface Heidi](#_Toc446711496) 18

[Figura 3: DCU do Administrador 2](#_Toc446711497)3

[Figura 4: CDU do professor 2](#_Toc446711498)3

[Figura 5 CDU do aluno 2](#_Toc446711499)4

[Figura 6: Base de Dados do Banco de Questões 2](#_Toc446711500)5

[Figura 7: Resultados ao vivo 2](#_Toc446711501)7

[Figura 8: Telemetria Cadastro de aluno](#_Toc446711502) 28

[Figura 9: Tela inicial](#_Toc446711503) 28

[Figura 10: Cadastro de aluno](#_Toc446711504) 29

[Figura 11: Menu do professor 3](#_Toc446711505)0

[Figura 12: Menu de cadastro de questões 3](#_Toc446711506)1

[Figura 13: Cadastro de questão objetiva 3](#_Toc446711507)2

[Figura 14: Cadastro de questão subjetiva 3](#_Toc446711508)3

[Figura 15: Pesquisar 3](#_Toc446711509)4

[Figura 16: Resultado (s) da pesquisa de questões 3](#_Toc446711510)5

[Figura 17: Minhas questões 3](#_Toc446711511)5

[Figura 18: Alteração de questões 3](#_Toc446711512)6

[Figura 19: Alteração concluída](#_Toc446711513) 37

[Figura 20: Menu do aluno](#_Toc446711514) 38

[Figura 21: Resultado (s) da pesquisa feita pelo aluno 4](#_Toc446711515)0

[Figura 22: Modelo da avaliação gerada pelo IFQuestion 4](#_Toc446711516)0

**LISTA DE TABELAS**

Tabela 1 – Requisitos funcionais do IFQuestion 21

Tabela 2 – Requisitos não funcionais do IFQuestion 22

**LISTA DE SIGLAS E ABREVIATURAS**

IFRN – Instituto Federal do Rio Grande do Norte

CSS – Cascating Style Sheets

DAO - Data Access Object

DCU - Diagrama de Casos de Uso

UML – Unified Modeling Language

SGBD - Sistema Gerencial de Banco de Dados

SQL - Structured Query Language

JSP – Java Server Pages

MVC - Model-View-Controller

OO - Orientação a Objetos

HTML – Hyper Text Markup Language

CPU – Central Processing Unit

JVM – Java Virtual Machine

**SUMÁRIO**

**1** **INTRODUÇÃO** 13

1.1 MOTIVAÇÃO 14

1.2 OBJETIVOS 14

1.3 ORGANIZAÇÃO 15

**2** **FUNDAMENTAÇÃO TEÓRICA** 16

2.1 JAVA 16

2.2 JAVA SERVER PAGES 17

2.3 TOMCAT 18

2.4 MYSQL 19

**3** **METODOLOGIA** 21

3.1 BRAINSTORMING 22

3.2 BRIEFING 22

**4** **DESENVOLVIMENTO 28**

4.1 GRÁFICOS DE DESENVOLVIMENTO DO SISTEMA.........................................42

4.2 JAVA VIRTUAL MACHINE....................................................................................43

**5** **CONCLUSÕES 44**

5.1 CONCLUSÕES GERAIS 45

5.2 CONCLUSÕES ESPECÍFICAS 46

5.3 TRABALHOS FUTUROS 46

**REFERÊNCIAS** 47

**1 INTRODUÇÃO**

Desenvolvido a partir de uma necessidade dos professores do Instituto Federal de educação, ciências e tecnologia do Rio Grande do Norte (IFRN), campus João Câmara, em especial o professor de história Gileno França - que foi o idealizador do projeto – juntamente com o professor Diego Oliveira – Orientador – foi então criado um sistema de armazenamento de perguntas e respostas, no qual denominamos de IFQuestion.

A ideia surgiu de um problema enfrentado pelo docente Gileno, que se deparava com a necessidade de armazenar questões referentes a sua disciplina. O armazenamento feito pelo professor era realizado em um documento *word*, porém esse tipo de armazenamento possuía diversos problemas, dentre eles a pesquisa de questões, pois o usuário teria que olhar questão por questão até achar a desejada, e isto levaria a um grande desperdício de tempo, vale ressaltar também que esse não é um armazenamento seguro.

Diante destes problemas, foi necessário pensar em uma solução eficiente, e essa solução seria o IFQuestion. Este sistema tem como objetivo principal o armazenamento seguro e eficaz de questões, além de facilitar a vida de seus usuários, que poderão economizar seu tempo e investi-lo em outras atividades.

O IFQuestion é um sistema voltado principalmente aos professores, nele é possível que os usuários adicionem questões, classifique-as, identifique-as, utilizando temas e subtemas que facilitem posteriormente na busca das mesmas, e guardar as respostas esperadas e/ou corretas de cada questão. Além disto há um espaço especialmente voltado para os alunos, que poderão visualizar as questões cadastradas anteriormente pelos professores, e com isso estudar para as provas.

O sistema também conta com uma função de extrema importância, que é a elaboração de provas. Esta funciona da seguinte maneira: o usuário – neste caso o docente que tem acesso as questões – escolhe as perguntas do assunto que está sendo abordado em sala de aula, após selecioná-las as questões irão automaticamente para uma nova página, na qual já constará o nome da instituição de ensino, bem como os espaços de preenchimento de dados dos alunos e informações da disciplina em que a prova irá ser aplicada.

* 1. MOTIVAÇÃO

A partir da observação do cenário que o professor Gileno França enfrentava, sugeriu-se a criação de uma solução rápida e eficiente, que proporcionasse aos docentes mais segurança e aproveitamento de tempo, podendo assim, dedicar-se a realização de outras tarefas, que é o já citado IFQuestion.

* 1. OBJETIVOS

O objetivo principal deste trabalho, é a criação de um software que facilite a vida de seus usuários, ao passo que auxilie na realização de tarefas feitas por eles de modo fácil e seguro. O sistema no qual denominamos de IFQuestion, foi implementado na plataforma *web*, e projetado para suprir as necessidades internas identificadas no campus IFRN João Câmara.

Na intenção de alcançar o objetivo geral, foram traçados alguns objetivos específicos, dentre os quais, podemos citar: colocar em prática os saberes adquiridos durante todo o curso e conduzi-los ao desenvolvimento do projeto. Todos os conhecimentos foram essenciais, dentre os quais:

* O levantamento dos requisitos e práticas envolvidas,
* A aplicação da orientação a objetos durante a programação do sistema;
* Conhecimento de novas tecnologias no decorrer de cada fase de implementação do projeto;
* Realizar a implementação do sistema;
* Desenvolver um protótipo do IFQuestion
* Avaliar e testar o sistema.
  1. ORGANIZAÇÃO

Este documento foi organizado da seguinte forma:

O capítulo 1 expõe de forma resumida do que este documento se trata, trazendo assim uma introdução sobre o tema, juntamente com a motivação, os objetivos e a organização.

O capítulo 2 aborda as principais ferramentas e tecnologias utilizadas no desenvolvimento do projeto.

O capítulo 3 traz uma visão geral do sistema IFQuestion, as metodologias utilizadas para levantar os requisitos e a listagem desses elementos. Além disso, apresenta a relação de casos de uso e o relacionamento desses com os requisitos do sistema.

O capítulo 4 abrange as particularidades do projeto proposto, neste capitulo iremos abordar o desenvolvimento do sistema, incluindo o levantamento de requisitos, modelagem do banco, programação das funcionalidades, *design* das telas e números (tempo investido, linhas de código, classes).

Por fim, o capítulo 5 contém as conclusões, que relatam os resultados da pesquisa em relação ao desenvolvimento do sistema e sugestões para trabalhos futuros que poderão ser feitos para deixar o sistema mais completo e eficiente.

1. **FUNDAMENTAÇÃO TEÓRICA**

Este capítulo tem como objetivo principal apresentar as tecnologias e as metodologias utilizadas durante o desenvolvimento do sistema. Dentro desta concepção abordaremos as seguintes ferramentas que foram aplicadas, agrupando-as do seguinte modo: tópico 2.1 apresentações da linguagem de programação JAVA; tópico 2.2 exposição da tecnologia de desenvolvimento de páginas *web* em linguagem *Java* com a junção de *HTML (Java Server Pages – JSP*); tópico 2.3 apresenta o *Apache Tomcat* que é um servidor *web Java* que permite a execução de aplicações *web*; e por fim o tópico 2.4*Structure Query Language* – Linguagem de Consulta Estruturada (SQL) que é utilizada pela base de dados *MySQL* para realizar consultas.

* 1. JAVA

O paradigma clássico ou estruturado, precursor do paradigma orientado a objetos, passou pouco tempo sendo considerado promissor, uma vez que, na época em que foi criado, não existiam *softwares* com uma grande quantidade de linhas de código, logo, apenas conseguiu suprir as necessidades desse período, afirma Schach (2009). Contudo, com a rápida evolução do mercado de *software* e com seu rápido crescimento, passaram a surgir sistemas de grande porte, e esse paradigma começou a apresentar problemas, como:

* As técnicas eram incapazes de lidar com o tamanho cada vez maior dos produtos de software que passaram a surgir;
* Durante a manutenção pós-entrega, esse paradigma não conseguia atender as expectativas.

Um dos principais motivos que gerou esses problemas diz respeito ao modelo das técnicas desse paradigma, que eram orientadas a operações ou orientadas a atributos, mas não aos dois simultaneamente. Em comparação com o paradigma da orientação a objetos, esta considera igualmente importantes tanto os atributos quanto as operações (Schach, 2009).

Utilizando o conceito de objeto, a Orientação a Objetos considera cada objeto do sistema formado por atributos e está sujeito a operações que podem ser realizadas com ele, de tal forma a alterar os valores de seus atributos. Preocupa-se em representar o mais fielmente possível as situações do mundo real em um sistema computacional, diz Farinelli (2007, p. 3).

O que possibilita reduzir a quantidade de código utilizados na programação do sistema é o conceito de herança, no qual, um objeto de nível superior pode ter seus atributos compartilhados por um objeto de classe inferior (analogamente às características de pai para filho). Outro fator está na segurança, obtida pelo encapsulamento, processo em que os detalhes que não contribuem para as características essenciais de um objeto são omitidos.

Todo programa Java consiste em pelo menos uma classe. É impossível escrever um programa Java que não use classes. Elas descrevem uma entidade lógica que contém dados, além de métodos (as instruções) para manipular estes dados. Um objeto é a instanciação física da classe que contém dados específicos (Anderson; Franceschi, 2010).

Os conceitos presentes foram aplicados no desenvolvimento do sistema, uma vez que, foi utilizada a linguagem Java voltada para o desenvolvimento *web*, *Java Server Pages*, sendo essa linguagem, em sua essência, orientada a objetos.

* 1. JAVA SERVER PAGES

*Java Server Pages* é uma tecnologia que auxilia os desenvolvedores na criação de páginas *web* dinâmicas.

Com JSP, os *designers* e programadores podem rapidamente incorporar elementos dinâmicos em páginas da *web*, utilizando Java embutido e algumas *tags* de marcação simples. Estas *tag*(s) fornecem ao designer de HTML um meio de acessar dados e lógica de negócio armazenados em objetos Java sem ter que dominar as complexidades do desenvolvimento de aplicações (Pittella; Felipe, 2013).

No JSP utiliza-se a arquitetura *Model-View-Controlle*r (MVC, em português Modelo-Visualização-Controle). É possível esclarecer esse padrão de arquitetura, de modo sintético, através dos tópicos listados a seguir (Metlapalli, 2010):

Este modelo a seguir foi utilizado durante o desenvolvimento do sistema IFQuestion.

* *Model* (Modelo) – corresponde às classes *Data Access Object* (DAO). Responsável pela parte lógica da aplicação. As ações encapsulam chamadas de classes da camada de negócio;
* *View* (Visual) – é representado pelas páginas JSP. Encarregado de apresentar o resultado;
* *Controller* (Controle) – corresponde aos *servlets.* Recebem as requisições decidindo que funções de lógica deverão ser executadas.



Figura : Java Server Pages (JSP) no IFQuestion

Na figura 1 podemos observar a tecnologia JSP sendo utilizada no sistema IFQuestion, como dito anteriormente JSP é uma ferramenta que possibilita a criação de páginas *web* dinâmicas. No **item 1** é possível visualizar o HTML, o mesmo é responsável pela criação dos formulários do sistema, já o **item 2** atribui ação/interatividade a esses formulários. Está imagem é um exemplo da arquitetura *view* do MVC, pois se trata de uma classe JSP que estará em contato direto com os usuários.

* 1. TOMCAT

Desenvolvido pela fundação *Apache*, o *software Tomcat* é um servidor que permite a execução de aplicações web, sua principal característica é estar centrada na linguagem de programação Java, mais especificamente nas tecnologias *de Servlets* e de *Java Server Pages* (JSP).

[*Apache Tomcat*](http://www.techtudo.com.br/tudo-sobre/s/apache-tomcat.html) é um programa que surgiu com dois grandes propósitos: ser mais leve e eficiente que os outros. Além de funcionar como um simples servidor *Web,* ***Apache Tomcat***ainda pode ser integrado a outros. É uma ferramenta escrita em código aberto, e é um programa gratuito (TechTudo, 2011).

* 1. MYSQL

O MySQL é um sistema de banco de dados relacional, que utiliza a linguagem SQL (*Structured Query Language*) para manipulação, definição e consulta de dados. É atualmente um dos bancos de dados mais populares, com mais de 10 milhões de instalações pelo mundo. Suas principais características são: portabilidade (suporta praticamente qualquer plataforma atual), compatibilidade de interface para diversas linguagens de programação, excelente desempenho e estabilidade, pouco exigente quanto a recursos de *hardware*, facilidade de uso, é um *Software* Livre e de replicação facilmente configurável.

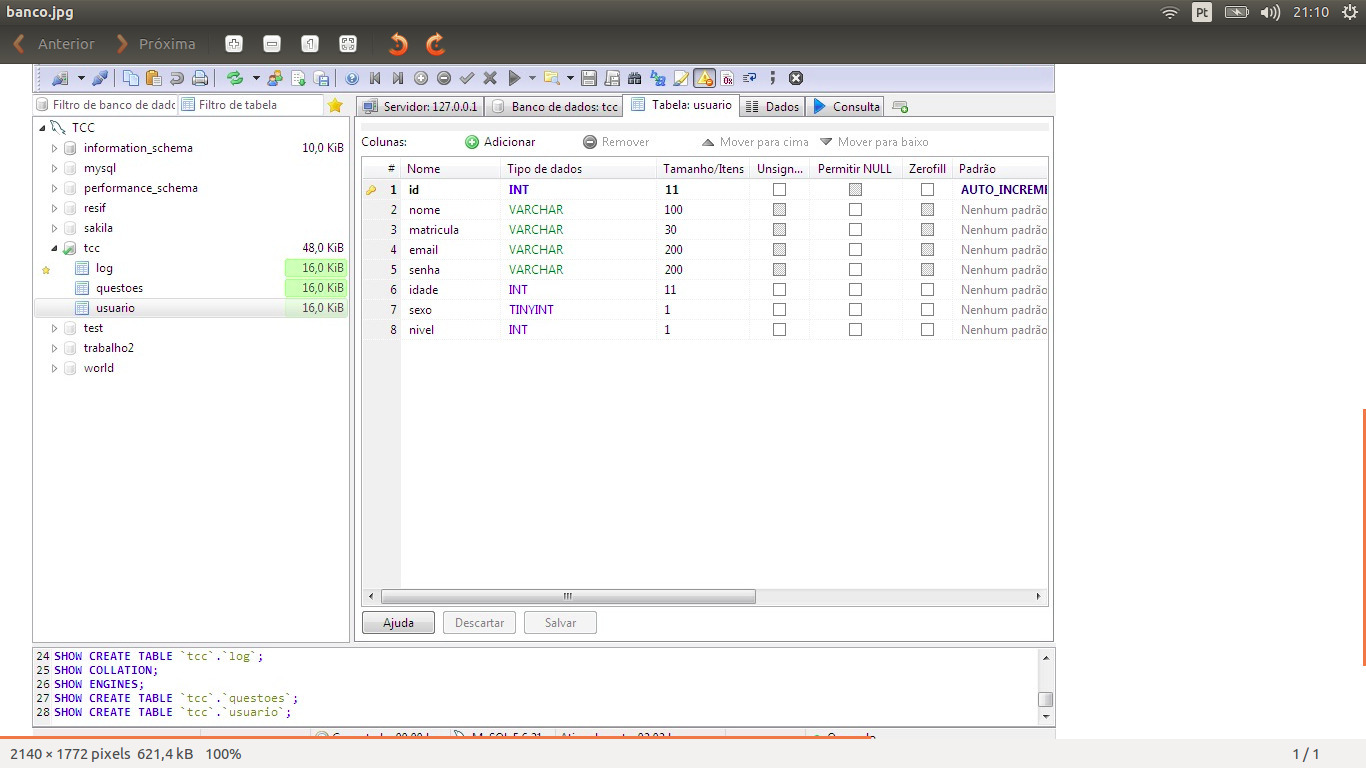


Figura : Banco de dados MySQL com a interface Heidi

A figura 2 mostra uma das tabelas do banco de dados do IFQuestion, nesta imagem é possível visualizar os nomes das colunas da tabela usuário como por exemplo: id, nome, matricula, email etc. Aqui também é possível visualizar os tipos de dados que serão cadastrados no caso de “INT” os valores que serão armazenados serão inteiros, melhor dizendo: números. Caso “VARCHAR” serão guardados textos (palavras).

1. **METODOLOGIA**

Tanto os estudos quanto as pesquisas são realizadas com base nos métodos, e a metodologia é o estudo destes métodos. Neste capítulo serão abordadas as técnicas e procedimentos que foram necessários para o levantamento de requisitos do sistema.

Inicialmente, foi observado as necessidades do nosso cliente, para entendermos melhor que o sistema deveria atender, a partir desta observação foram encontrados os seguintes problemas:

* As questões eram salvas em um documento Word.
* O armazenamento dessas questões requeria uma demanda de tempo muito grande.
* Este não é um armazenamento seguro, visto que, não há nenhuma proteção eficaz nos arquivos Word.
* Caso o usuário necessitasse buscar alguma questão, fazia-se necessário a busca detalhada por todo documento até encontrar a questão desejada.

As coletas de requisitos na fase inicial foram de grande importância para a entendimento do sistema. Esses requisitos refletem as necessidades dos clientes de um sistema que ajuda a resolver um problema, por exemplo, controlar um dispositivo, enviar um pedido ou encontrar informações, argumenta Sommerville (2007).

Para isso, diversas técnicas foram propostas, e vem sendo, para uso isolado ou combinado no processo de elicitação de requisitos, pelos analisas de requisitos. A seguir algumas dessas:

- **Entrevistas com os usuários:** interagir com os interessados no sistema em reuniões, com a intenção de conhecer o cenário e a problemática, além de gerar o *feedback*;

**- Cenários:** construções essenciais que explicam como o sistema poderá ser usado;

**- Prototipagem:** funcionam como ferramenta de elicitação de requisitos, possibilita ao usuário testar o sistema e mostrar os pontos positivos e negativos;

**- Estudo etnográfico:** baseia-se na observação direta do desempenho das atividades das pessoas.

Nesse projeto, utilizou-se para o levantamento de requisitos dos usuários, a entrevista, e técnica de prototipagem. Workshops de requisitos é uma atividade de levantamento de requisitos, onde todos os *stakeholders* são reunidos durante um período curto, mas intensivo e focado na discussão e definição dos requisitos. Segundo Sommerville (2007), o termo *stakeholders* refere-se a pessoa ou grupo afetado pelo sistema, direta ou indiretamente.

Para identificar melhor suas necessidades, foi-se necessário utilizar algumas das estratégias de levantamento de requisitos, dentre as quais o *Brainstorming* e o *brienfing*.

* 1. BRAINSTORMING

O *Brainstorn,* que tem como um dos principais significados: tempestade de ideias, ele é caracterizado como uma dinâmica em grupo que é utilizado por diversas empresas e/ou programadores para resolver problemas específicos, desenvolver novas ideias ou projetos, juntar informações e também para estimular o pensamento criativo, além de ser uma forma de entender as funcionalidades que o sistema deveria atender, para assim chegar o mais próximo o possível do que o cliente deseja.

Criado nos Estados Unidos, pelo publicitário Alex Osborn, a técnica de *brainstorming* propõe que um grupo de pessoas se reúnam e utilizem seus pensamentos e ideias para que possam chegar a um denominador comum, a fim de gerar ideias inovadoras que levem um determinado projeto adiante. É importante salientar que nenhuma ideia seja descartada ou julgada como errada ou absurda, todas as possíveis ideias devem ser anotadas e consultadas no decorrer da execução do projeto (BARBOSA, 2007).

* 1. BRIEFING

O *briefing* funciona como uma coleta de dados para o desenvolvimento do trabalho, seja ele a criação de um sistema ou de uma propaganda. Seu significado em português é “resumo”. É um documento em que pode conter a descrição da situação, os problemas encontrados bem como os objetivos e os possíveis recursos para solucioná-los. É a base de um processo de um planejamento, visto que, contém todo conjunto de informações.

Após estes processos, identificamos os requisitos que o sistema IFQuestion deveria atender e classificamos como funcionais e não-funcionais. Os funcionais descrevem os serviços do sistema enquanto os não-funcionais descrevem propriedades e restrições. A Tabela 1 expõe os requisitos funcionais, e a Tabela 2, os requisitos não funcionais. Os requisitos se enquadraram em uma das seguintes categorias: desejável (não obrigatório e recomendada para melhor desempenho) ou evidente (funcionalidade visível na aplicação).

As tabelas seguintes expõem as funcionalidades que o sistema deverá atender, bem como uma discrição prévia sobre elas.

* Na coluna “**Nome”** está o nome da funcionalidade/requisito.
* Na coluna “**Descrição**” está a descrição sobre a funcionalidade/requisito.
* Na coluna “**Categoria**” está a importância desta funcionalidade, que pode ser definida como dois tipos diferentes, desejável e evidente;

Tabela 1– Requisitos funcionais do IFQuestion.

|  |  |  |
| --- | --- | --- |
| **Nome** | **Descrição** | **Categoria** |
| Autenticar-se | Autenticar-se no sistema fornecendo login e senha. | Evidente |
| Gerenciar usuários | O administrador poderá cadastrar, alterar e excluir dados dos usuários (professores). | Evidente |
| Gerenciar questões | O usuário poderá cadastrar, alterar e excluir questões. | Evidente |
| Elaborar prova | O usuário do sistema (docente) pode elaborar uma prova/avaliação com as questões contidas em seu cadastro. | Evidente |
| Pesquisar questões | O cliente poderá pesquisar as questões inseridas por ele anteriormente em seu banco de dados. | Evidente |
| Visualizar questões | Os usuários do sistema poderão visualizar todas suas questões cadastradas por eles. | Evidente |

Tabela 2 – Requisitos não funcionais do IFQuestion.

|  |  |  |
| --- | --- | --- |
| **Nome** | **Descrição** | **Categoria** |
| Prazo | O sistema deve estar pronto no começo de março, tendo o tempo de aproximadamente dez (10) meses para o seu desenvolvimento. | Obrigatório |
| Usabilidade | Boa diagramação dos componentes das páginas Web dinâmicas | Obrigatório |
| Segurança | Deve haver confidencialidade das informações | Obrigatório |
| Robustez | O sistema deve atender ao objetivo de confiabilidade do sistema e executar sem erros recorrentes. | Desejável |
| Base de dados | MySQL | Obrigatório |
| Tecnologia de desenvolvimento | Deve ser desenvolvido usando JSP | Obrigatório |

Após o levantamento de requisitos, deu-se início ao estudo da Linguagem de Modelagem Unificada (*Unified Modeling Language* - UML) que se tornou a linguagem universalmente aceita para documentação de projetos de *software.* A UML oferece vários meios de ver como o sistema será modelado, permitindo analisá-lo e modelá-lo de diversos modos, tendo que fazer com que cada diagrama complemente o outro, para atingir o modelo do projeto final. Cada parte do sistema é analisada sobre uma certa visão, algumas com uma visão mais geral, como o Diagrama de Caso de Uso e outros com uma visão mais especifica (GUEDES, 2009).

A Figura 3 apresenta os casos de uso de ator Administrador pode realizar diversas ações no sistema, como por exemplo, Cadastrar usuários, Gerenciar questões, além de Gerenciar o programa, gerenciar o programa implica envolver diversas ações nas quais o administrador tem controle geral sobre o IFQuestion.

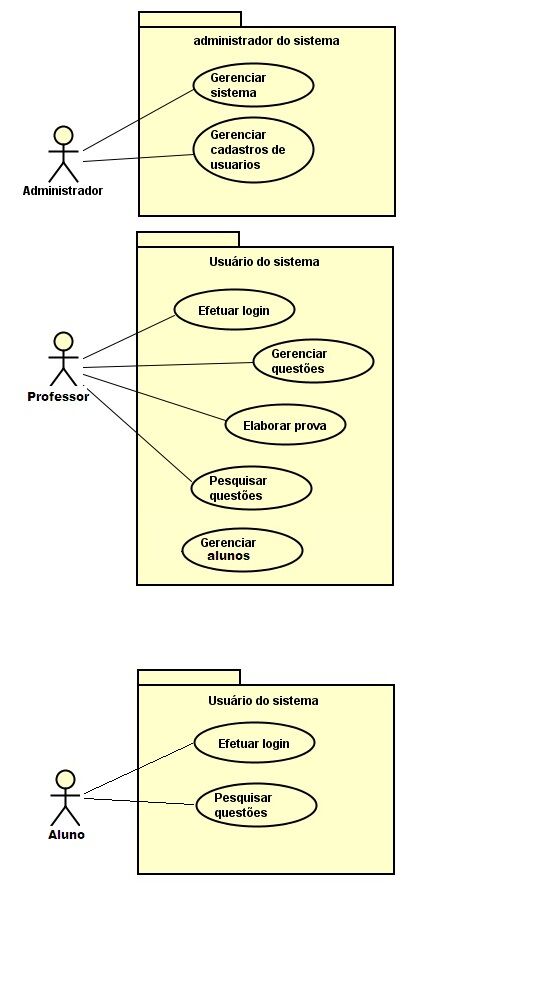


Figura 3: DCU do Administrador

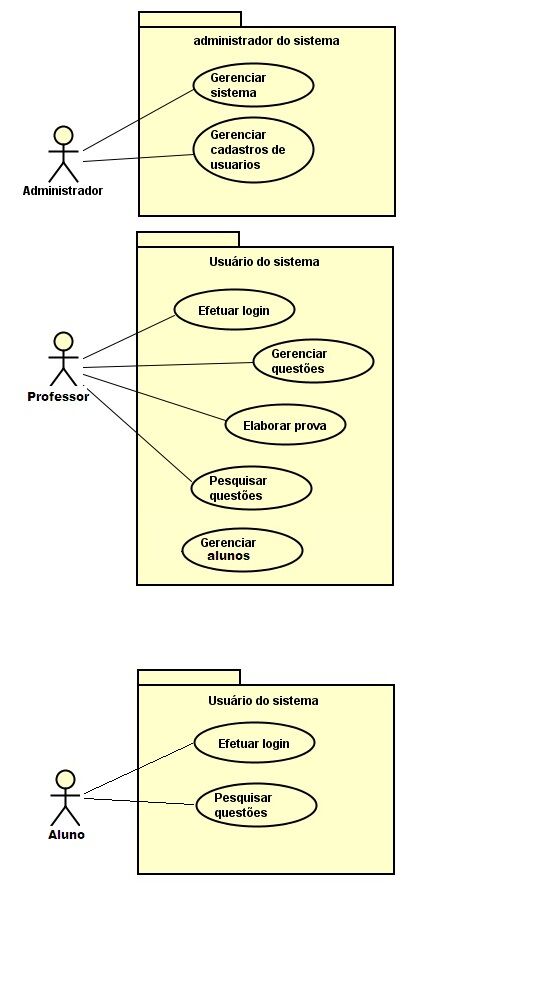


Figura 4: CDU do professor

Na figura 4 temos o CDU do professor, que poderá realizar *login* no sistema, gerenciar questões, elaborar prova, pesquisar questões, além de gerenciar alunos (podendo assim remover ou adicionar). Vale ressaltar também, a importância de somente o professor ou um técnico – que neste caso também pode ser um monitor da disciplina- cadastrar os alunos, pois assim evitará que o mesmo se cadastre como um professor e tenha acesso privilegiado no sistema.

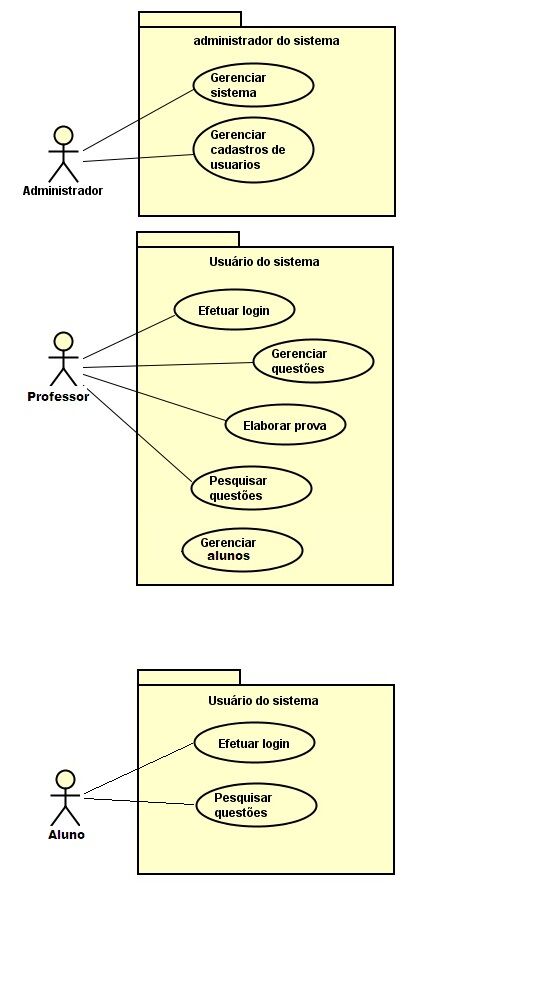


Figura 5: CDU do aluno

A Figura 5 apresenta o CDU do aluno, que poderá efetuar *login* e pesquisar/visualizar questões.

Antes da implementação propriamente dita do sistema, foram feitos diferentes diagramas que exemplificasse melhor o que deveria ser feito no projeto, dentre os quais podemos citar, o modelo do Banco de Dados, com base nas entidades, que poderá ser visualizado na figura seguinte (figura 6).

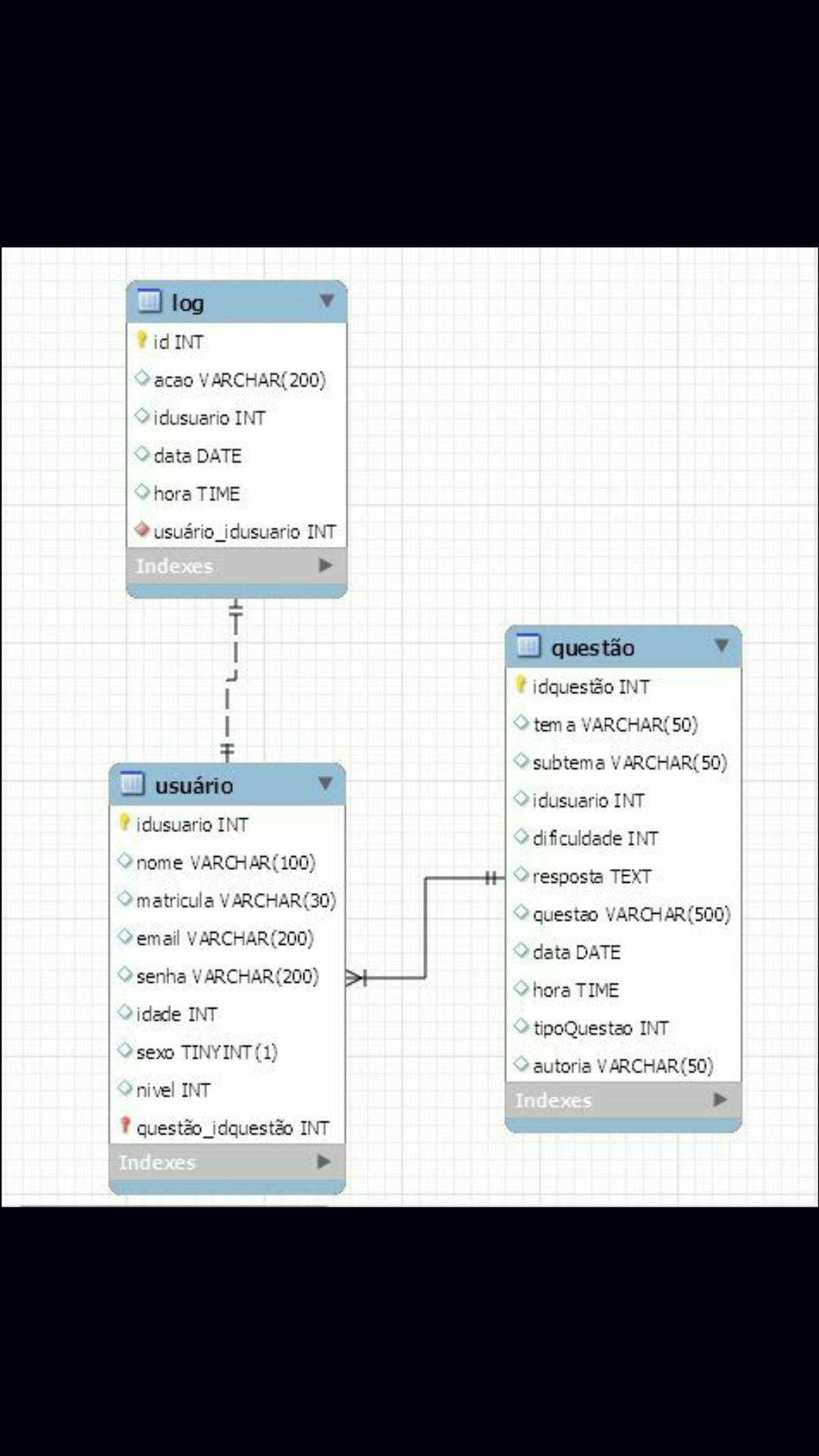


Figura 6: Base de Dados do Banco de Questões

1. **DESENVOLVIMENTO**

Como dito anteriormente, o sistema IFQuestion foi desenvolvido a partir de uma necessidade dos professores do IFRN, campus João Câmara. As principais necessidades eram o armazenamento eficaz e seguro de questões que são utilizadas em suas disciplinas, seja na elaboração de exercícios complementares ao assunto que estava sendo abordado no momento, ou até mesmo na criação de atividades avaliativas.

O projeto utilizou diferentes plataformas e linguagens de desenvolvimento, que já foram exemplificadas no decorrer do capítulo Fundamentação teórica. Agora no decorrer deste novo capitulo abordaremos como se deu o processo de desenvolvimento/implementação do sistema.

Para o IFQuestion adotamos a estrutura *Model View Controller* (MVC). No sistema o *Model* corresponde às classes DAO, que implementam a lógica da aplicação, o *View* é representado pelas páginas JSP, que é encarregado de apresentar o resultado, e o *Controller* corresponde aos *servlets*, que são os controladores das requisições.

**O modelo (*Model*)** é utilizado para manipular informações de forma mais detalhada. É o modelo que tem acesso a toda e qualquer informação sendo essa vinda de um banco de dados.

**A visão (*view*)** é responsável por tudo que o usuário final visualiza, toda a interface, informação, não importando sua fonte de origem, é exibida graças a camada de visão.

**A Controladora (*controller*),** como o nome já sugere, é responsável por controlar todo o fluxo de informação que passa pelo site/sistema. É na controladora que se decide “se”, “o que”, “quando” e “onde” deve funcionar. Define quais informações devem ser geradas, quais regras devem ser acionadas e para onde as informações devem ir, é na controladora que essas operações devem ser executadas. Em resumo, é a controladora que executa uma regra de negócio (modelo) e repassa a informação para a visualização (visão).

Os principais motivos para utilizar esta tecnologia são:

* Facilita o reaproveitamento de código;
* Facilidade na manutenção e adição de recursos;
* Maior integração da equipe e/ou divisão de tarefas;
* Facilidade em manter o seu código sempre limpo;
  1. GRÁFICOS DE DESENVOLVIMENTO DO SISTEMA

O *Netbeans* conta com uma ferramenta de determinação de perfil poderosa que pode fornecer informações importantes sobre o comportamento das aplicações corporativas. Usando a ferramenta de determinação de perfil, é possível monitorar facilmente os estados dos dados sobre o desempenho da CPU e uso de memória de uma aplicação. Para que a determinação do perfil seja feita é necessário que a implantação seja feita em um *Tomcat* local registrado ou *GlassFish*.

A seguir, iremos examinar o gráfico da memória para ver qual objeto está ocupando a maior parte da memória. No campo 'Bytes ao-vivo', o gráfico mostra o tamanho da memória usado por cada objeto. De acordo com os mesmos, nós podemos ver qual objeto ocupa a memória.

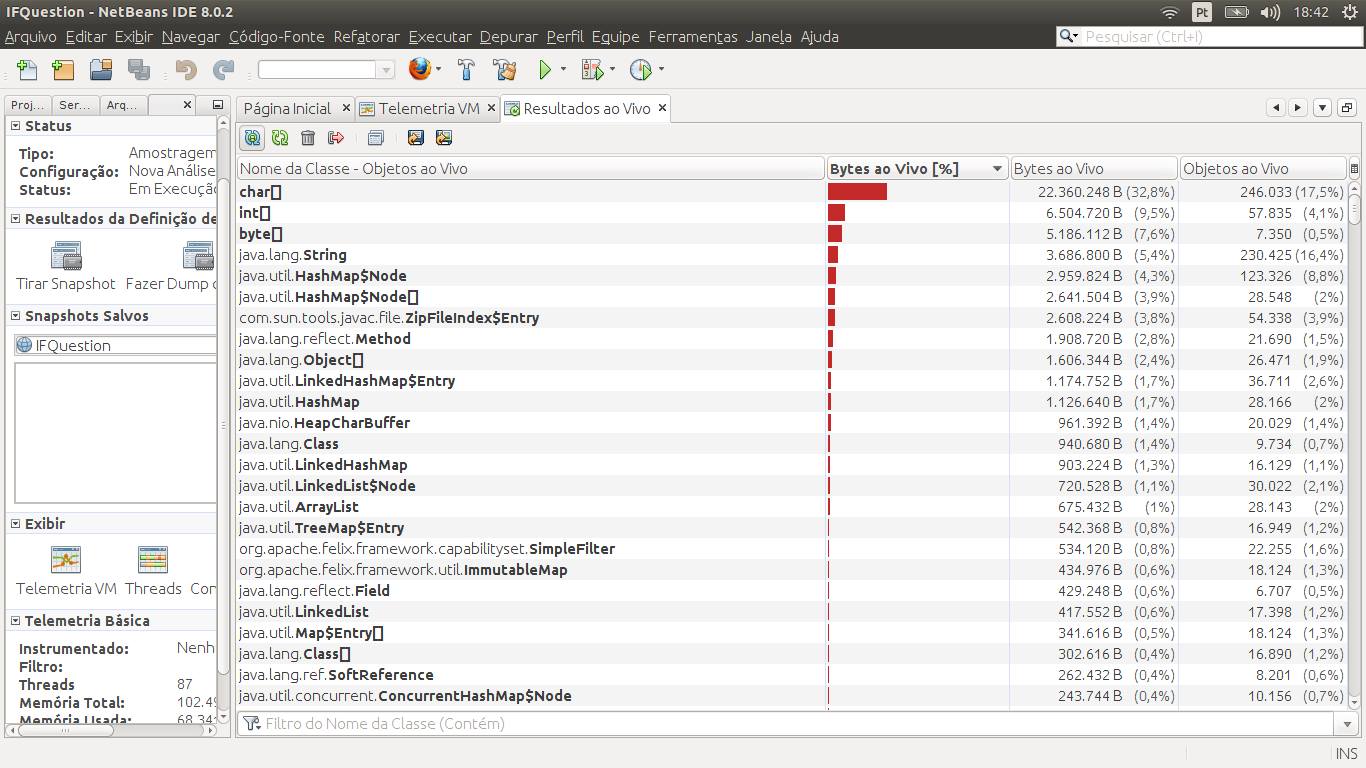


Figura 7: Resultados ao vivo

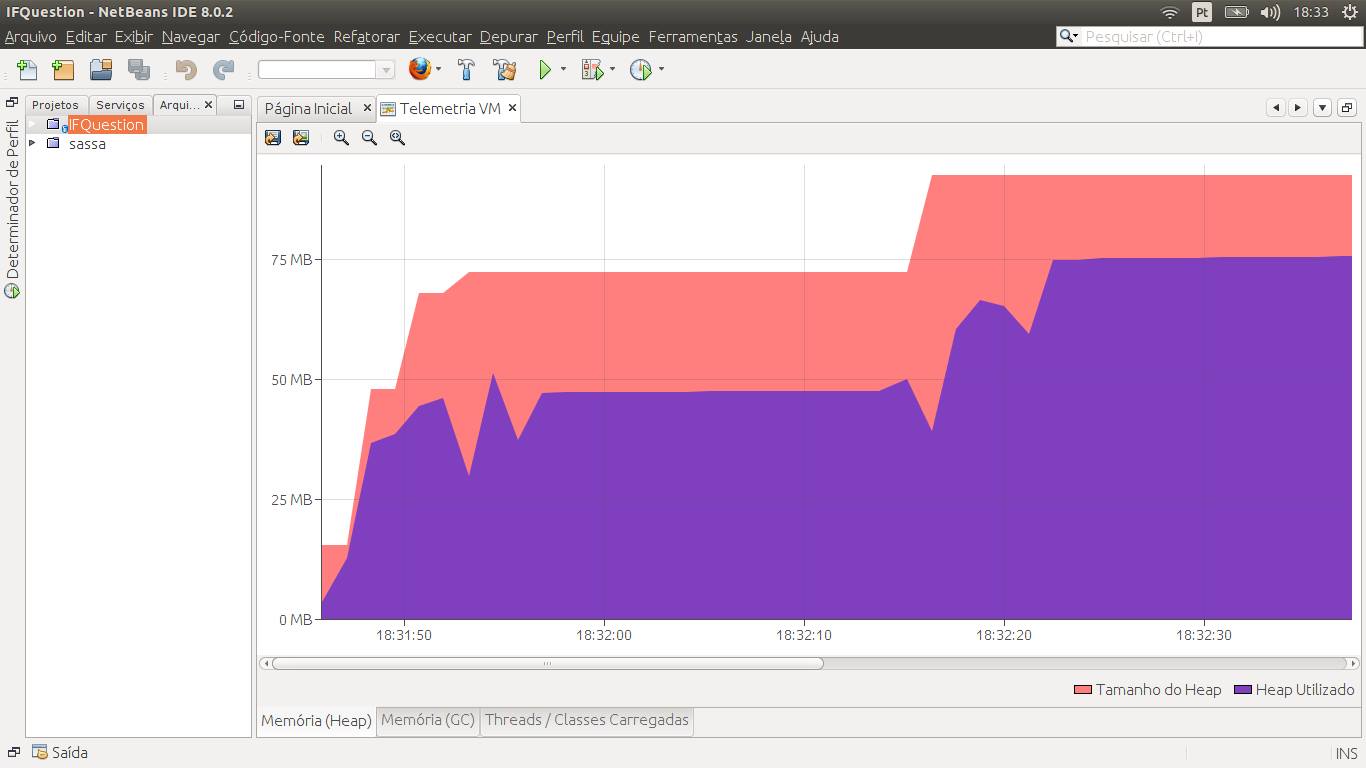


Figura 8: Telemetria

A Telemetria é usada para obter rapidamente uma visão geral dos dados de monitoramento em tempo real. A figura acima explica sucintamente como o Java Virtual Machine (JVM) usa a memória. Neste caso, o gráfico da figura 14 é relativamente instável, pois continua a crescer ao longo das aplicações desenvolvidas.

* 1. JAVA VIRTUAL MACHINE (JVM)

Java Virtual Machine é um aspecto do software Java envolvido na interação Web. A Java Virtual Machine é criada diretamente no download de software Java e ajuda a executar aplicações Java.

JVM é uma máquina virtual, como seu próprio nome já diz. E uma das funcionalidades dela é de servir como interpretador para o Java.  
Resumindo, o funcionamento é bem simples. Você escreve o código em um arquivo *.java*, compila, e é gerado um arquivo “*.class”,* esse arquivo “*.class”* é conhecido como  *bytecode*. A JVM irá ler/interpretar esse *bytecode* e executá-lo (FROTA, 2013).

A seguir será demostrado o IFQuestion através de *screenshots* obtidos de algumas das principais telas do sistema. A Figura 8, demonstra a página inicial do IFQuestion, uma vez que, para que o usuário tenha acesso ao sistema é necessário que ele esteja cadastrado, possuindo assim, uma matricula e uma senha. Nessa página, o usuário tem que informar os dois campos já citados.



Figura 9: Tela inicial

Esta é a tela inicial do sistema (figura 7), no **item 1** o usuário já cadastrado anteriormente deverá informar sua matricula bem como sua senha do sistema. O **item 2** é especialmente voltado para pessoas que ainda não se cadastraram no IFQuestion. No item 3 temos a área resevada para o administrador, que desempenhará funções mais complexas no sistema, como por exemplo, cadastrar alunos, já no **item 4** temos algumas imagens mostrando como o *software* deve ser utilizado - uma espécie de manual-.



Figura 10: Cadastro de aluno

(figura 8) No **item 1** tem-se o espaço para preenchimento com os dados do aluno, como por exemplo: nome, matricula, e-mail, senha, idade e sexo, para realização do cadastro todos os campos devem ser informados; no **item 2** temos o botão de Cadastrar, que confirma os dados e realiza o cadastro do usuário. Caso algo esteja incorreto ou o administrador deseje cancelar, deverá ser utilizado o **item 3**.

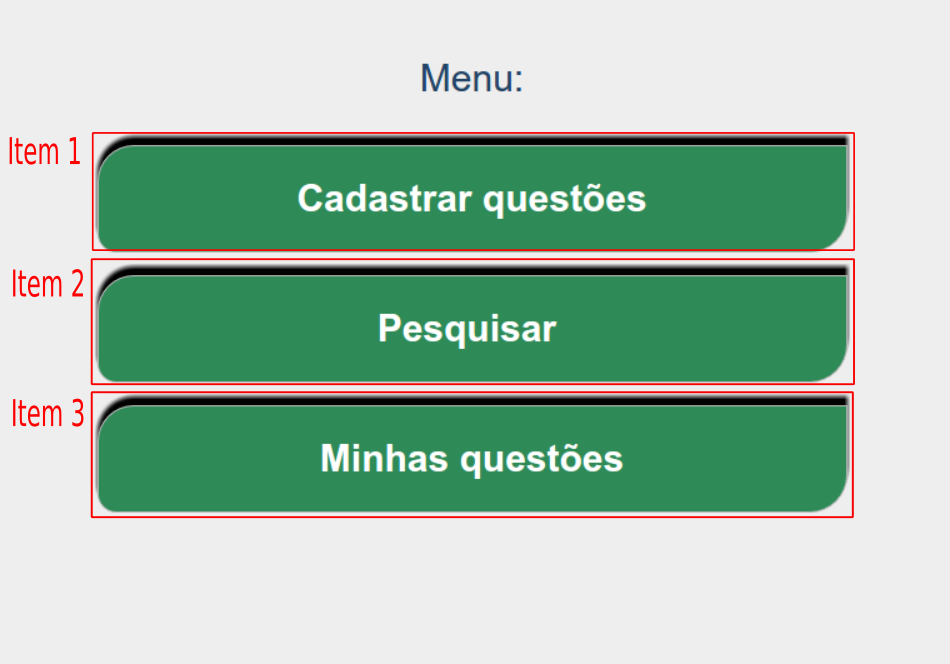


Figura 11: Menu do professor

Nesta imagem (figura 9) tem-se o menu do professor, no **item 1** é possível ver o usuário que está logado no momento bem como a opção de sair do sistema; no **item 2** tem-se a opção de cadastrar questões; no **item 3** pesquisar questões; **item 5** visualizar as questões cadastradas por aquele docente.

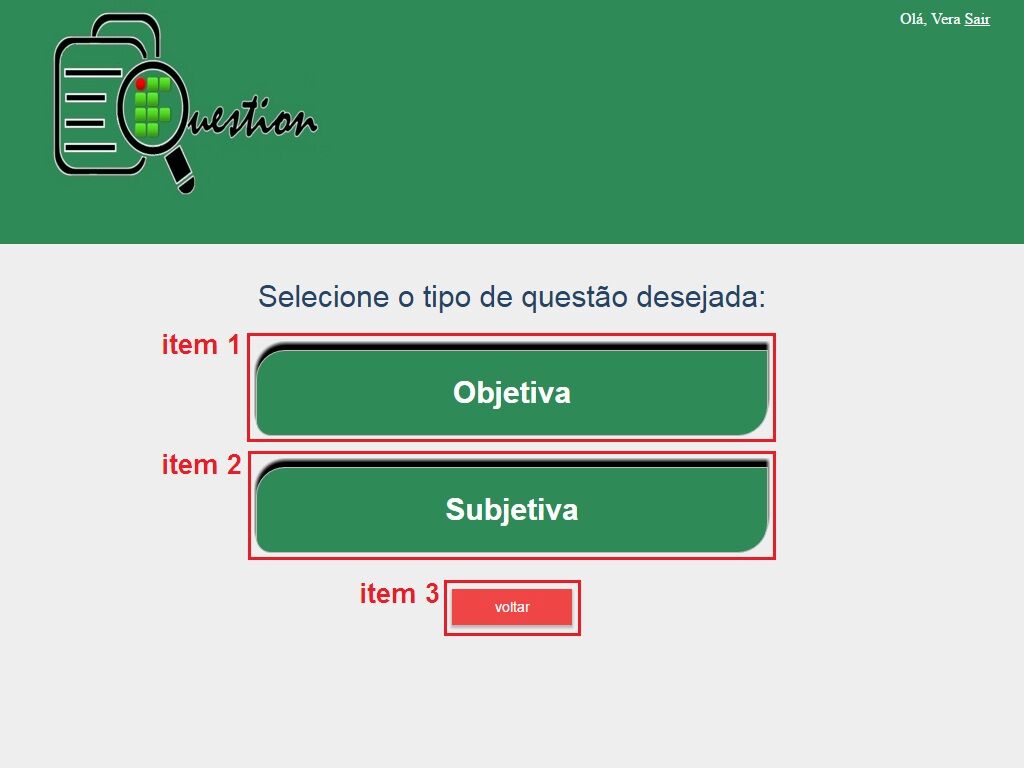


Figura 12: Menu de cadastro de questões

Ao selecionar o botão Cadastrar questões, o usuário será direcionado para esta tela (figura 10), na qual deverá escolher o tipo de questão que deseja cadastrar, se objetiva ou subjetiva; no caso do cadastro de questão objetiva deverá selecionar o **item 1**, e de questão subjetiva **item 2**; caso o usuário deseje retornar ao menu anterior deverá clicar no botão voltar (**item 3**).

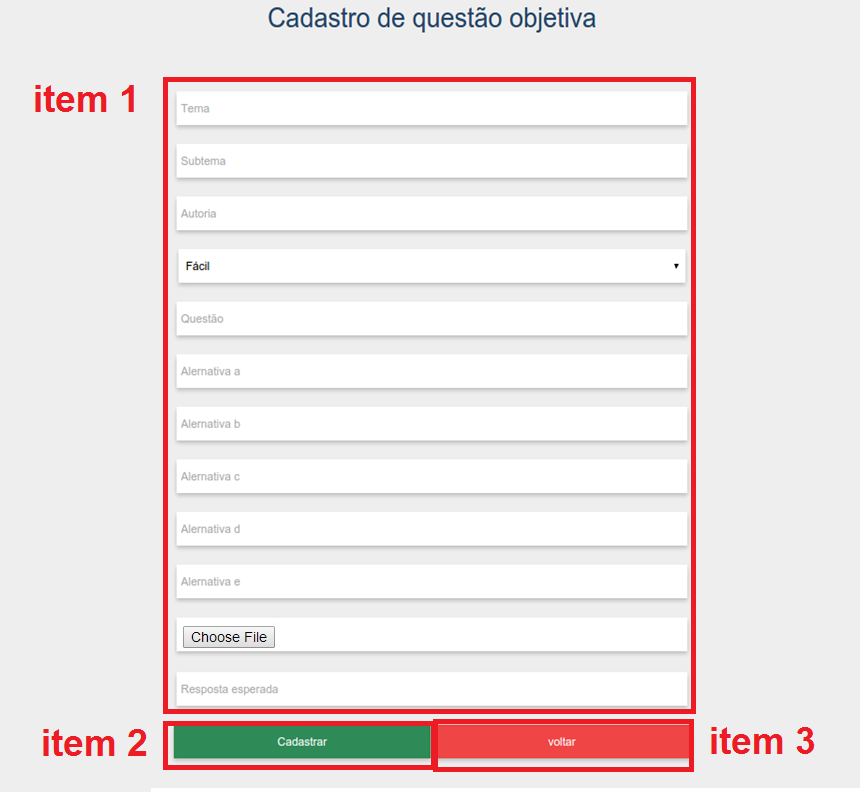


Figura 13: Cadastro de questão objetiva

Na figura 11 tem-se o cadastro de questões objetiva, no **item 1** temos o espaço para preenchimento de dados da questão como tema, subtema, autoria da questão, nível, questão, alternativa A, B, C, D e E, além da alternativa que responde corretamente a questão, caso seja necessário utilização de imagens na questão o usuário poderá adicionar a figura desejada usando o formulário “*Choose File*”. **Item 2** efetuar o cadastro da questão; **item 3** cancelar e retornar ao menu anterior.



Figura 14: Cadastro de questão subjetiva

Na figura 12 será mostrado o cadastro de questões subjetivas, **item 1** tem-se o conteúdo/tema de que a questão se trata, subtema da questão, autor/entidade autora da questão, nível de dificuldade da questão, que pode ser classificado como fácil, médio e difícil, o enunciado da questão, a resposta correta esperada e assim como no cadastro de questão objetiva neste cadastro também é possível inserir imagens utilizando o formulário “*Choose File*”; no **item 2** há a opção de efetuar o cadastro da questão, e no **item 3** cancelar e retornar ao menu anterior.



Figura 15: Pesquisar

A tela pesquisa está demostrada na figura 13. No **item 1** temos os dados que podemos filtrar as questões, como tema, subtema, autoria, data e grau de dificuldade; no **item 2** temos o botão pesquisar e no **item 3** cancelar a pesquisa e retornar ao menu anterior.

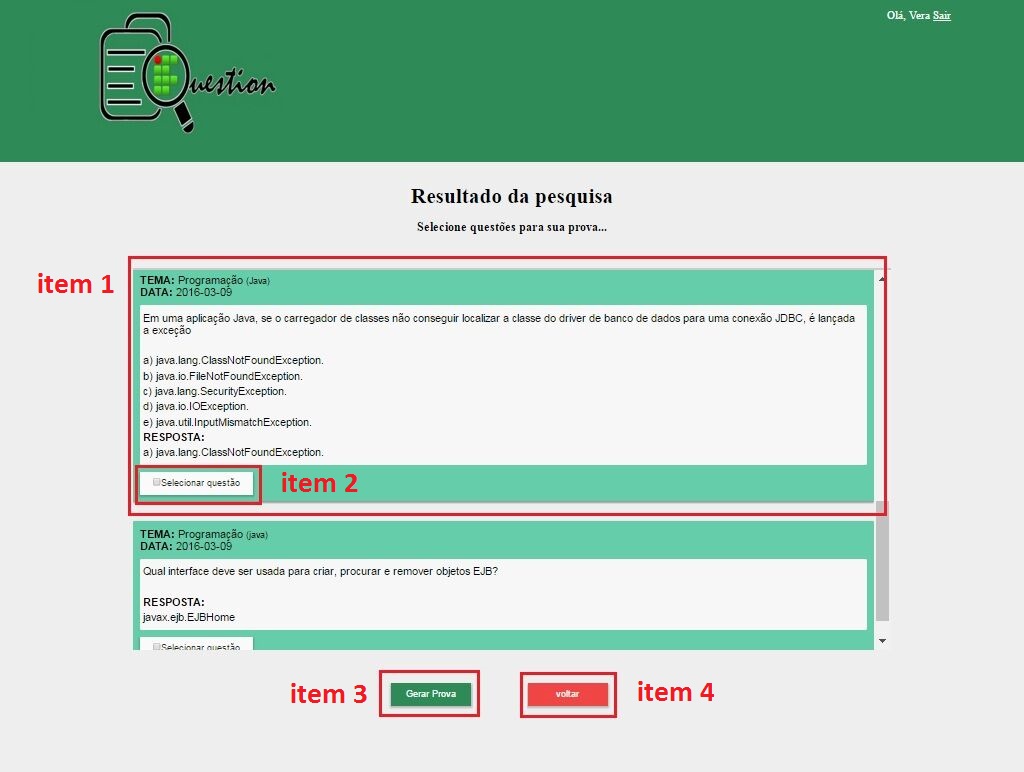


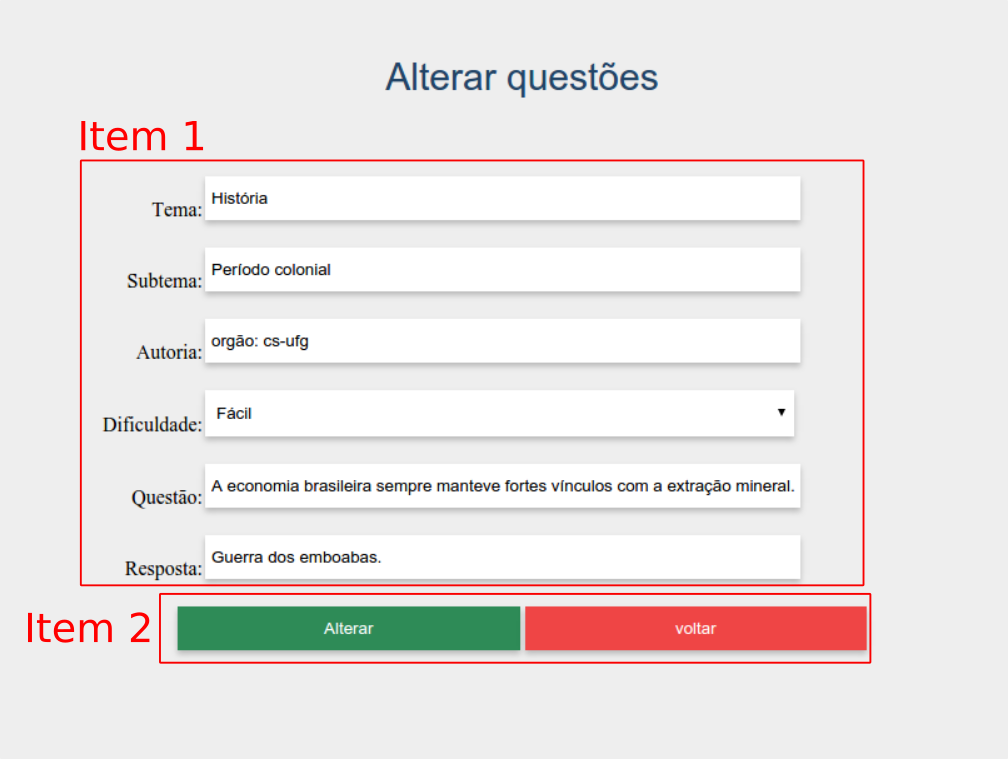
Figura 16: Resultado (s) da pesquisa de questões

Na figura 14 tem-se o resultado de uma pesquisa feita por um professor; no **item 1** temos o tema e a data da questão, bem como a questão propriamente dita e a resposta correta; no **item 2** é possível selecionar a questão; **item 3** gerar prova, que irá conter todas as questões que foram selecionadas; **item 4** voltar ao menu anterior.



Figura 17: Minhas questões

A figura 15 representa a listagem de questões de um determinado docente. No **item 1** temos os dados da questão listada; no **item 2** é possível selecionar a questão; **item 3** alterar dados da questão; **item 4** gerar prova a partir das questões selecionadas e **item 5** retornar ao menu anterior.

Figura 18: Alteração de questões

Na figura 16 tem-se a tela alteração de questão, no **item 1** é possível visualizar os dados da questão, como o tema, subtema, autoria, dificuldade, enunciado da questão bem como a resposta correta/esperada, para que assim o docente altere somente o que é necessário; no **item 2** confirmar os dados corretos e alterar a questão ou cancelar alteração e retornar a tela anterior.

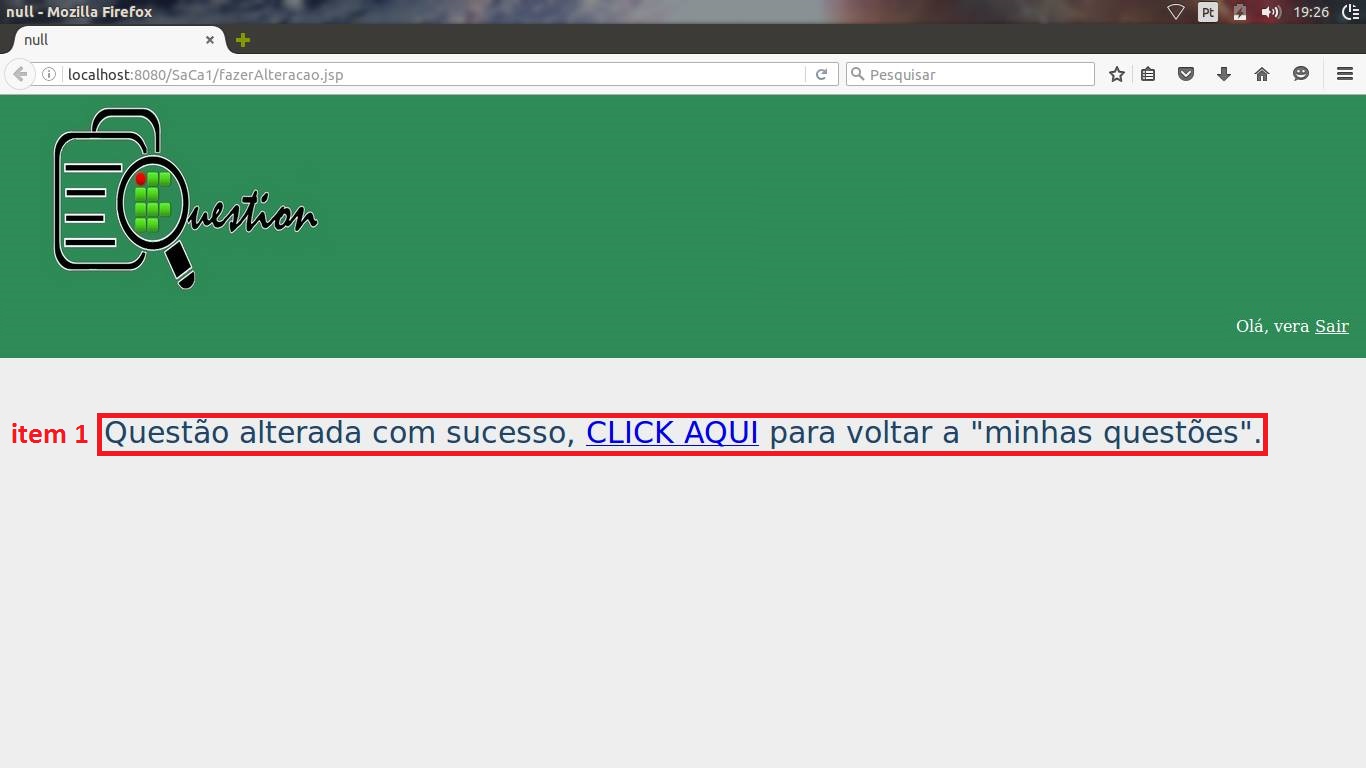


Figura 19: Alteração concluída

Na figura 17 tem-se a alteração concluída. No **item 1** o usuário poderá conferir se a alteração foi concluída e retornar a página “Minhas questões”.



Figura 20: Menu do aluno

Na figura 18 tem-se o menu do aluno, que diferentemente do docente só possui uma opção que é pesquisar questão; **no item 1** o usuário poderá pesquisar questões de determinado assunto, ano, nível de dificuldades e entre outras para que tenha mais uma ferramenta de estudo para as avaliações.



Figura 21: Modelo da avaliação

Figura 19: Resultado (s) da pesquisa feita pelo aluno, **item 1** tem-se o(s) resultado(s) do assunto pesquisado pelo aluno, que poderá visualizar a questão bem como sua resposta.

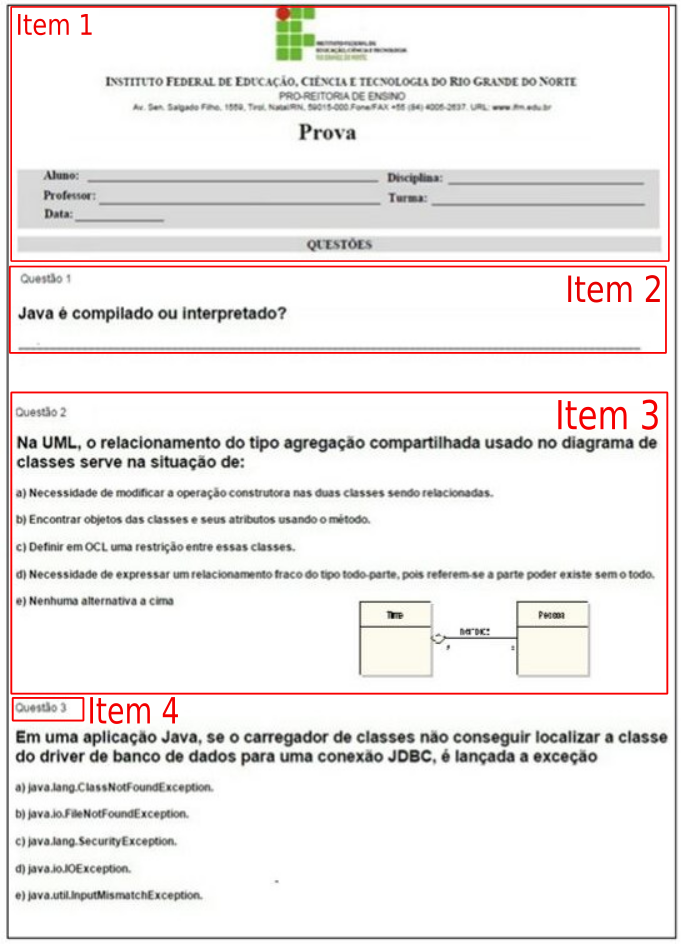


Figura 22: Modelo da avaliação gerada pelo IFQuestio

A figura 20 representa o modelo da prova que é criada pelo IFQuestion; no **item 1**: Cabeçalho institucional é gerado automaticamente, este possui o símbolo da instituição de ensino, neste caso o IFRN e campo destinado ao aluno, que preencherá como seus dados pessoais, como nome e turma além disto há também os dados da disciplina em que a prova está sendo aplicada; **item 2**: Questão subjetiva (enunciado) e neste caso, o sistema gera um espaço para a resposta do aluno; **item 3**: Questão do tipo objetiva (enunciado) bem como as alternativas. **item 4**: Número da questão (é gerado pelo sistema);

1. **CONCLUSÃO**

Com a criação do sistema IFQuestion, os docentes ganharam mais uma ferramenta de grande potencial para armazenarem suas questões de trabalho e elaborarem suas provas. A partir da criação deste projeto ficará mais rápido, prático e fácil realizar estas tarefas ao passo que as aplicações aconteceram de forma segura e eficiente.

Os benefícios de utilizar o IFQuestion são:

* Não é necessário fazer *download* do sistema nem instalar um software especifico para utilizar o sistema.
* Facilidade no uso: as telas são simples e intuitivas. Uma das principais características deste software é a facilidade de manuseio.
* Praticidade e conforto: usuário poderá acessar suas questões em tempo real, em qualquer lugar, desde que o mesmo possua acesso à Internet.
* Facilidade de acesso ao sistema pode ser feito utilizando qualquer dispositivo móvel (celular, tablet ou computador).
* Segurança: A integridade dos dados e segurança das informações é garantida por acesso restrito de usuários autorizados.
* Designer: o IFQuestion também conta com uma boa diagramação de páginas web.
* Fácil de usar: docentes de qualquer área poderão utilizar o sistema.
* O IFQuestion possui uma funcionalidade totalmente nova, a elaboração de prova, que permite ao usuário criar uma avaliação a partir de suas questões de maneira extremamente rápida e fácil.
  1. CONCLUSÕES GERAIS

A partir do desejo de aprimorar nossos conhecimentos que foram aprendidos no decorrer deste curso, bem como todas as tecnologias de programação estudada durante ele e que foram de grande importância para o desenvolvimento deste trabalho demos início a criação do sistema IFQuestion.

* 1. CONCLUSÕES ESPECÍFICAS

Com o término deste trabalho podemos colocar em prática diversos conhecimentos e aprimorar outros, que levaremos por toda a vida, entretanto, durante o desenvolvimento do sistema Banco de Questões foram encontradas algumas dificuldades, neste tópico iremos destacar algumas delas, dentre as quais:

* Uma das principais dificuldades encontradas no desenvolvimento deste projeto foi a pequena base na linguagem de programação Java, um problema que não é pessoal, e que atingiu toda a turma de informática 2012.1.
* A utilização de *Java Server Pages* (JSP) nos proporcionou interação nas páginas web, ao passo que, dinamizou a interação com os usuários, tornando-as mais intuitivas, contudo, a dificuldade com Java tornou mais complicada esta utilização e acarretou na demora da finalização deste sistema.
* Durante a etapa de levantamento de requisitos foram encontradas diversas dificuldades para entender as necessidades dos clientes. Para compreensão, foi pensado como o sistema funcionará para atender as necessidades como, por exemplo, o armazenamento de dados importantes que antes eram salvos em um arquivo *Word*.
* O IFQuestion inicialmente possuía mais funções, como o cadastro de alunos, que por sua vez poderiam interagir com os professores e estudar através das questões cadastradas pelos docentes, entretanto, com o passar dos dias vimos que isso seria inviável devido ao curto período de tempo e a dificuldade encontrada para desenvolver esta funcionalidade.

A seguir, serão apresentadas as principais tecnologias e os conhecimentos adquiridos sobre elas, apontando também sua importância e funcionalidade no proceder do trabalho.

* JSP – Tecnologia exposta em sala de aula, na disciplina de programação e desenvolvimento web, sua utilização trouxe muitas vantagens durante a construção do projeto. Entretanto, para aplicá-la no desenvolvimento do sistema foi necessário realizar uma pesquisa aprofundada sobre essa tecnologia, por exemplo, validação de formulário, criação de sessões, conexão com o banco de dados *MySQL*, entre outras.
* Arquitetura MVC – permitiu a organização do código, proporcionando mais agilidade no desenvolvimento do sistema, além de facilitar as modificações, bem como a divisão de tarefas.
* UML – Estudar essa linguagem visual de diagramação, permitiu construir diagramas que facilitaram a compreensão do sistema. Com ela foi possível descrever etapas da construção, o que foi fundamental no desenvolvimento do sistema.
  1. TRABALHOS FUTUROS

O sistema ainda necessita de muitas melhorias e reparos, então sugere-se que haja uma continuação do trabalho, com a finalidade de melhorar as funções já desenvolvidas.

É necessário também que os próximos discentes que derem continuidade a este projeto tentem suprir tantos as deficiências quanto criar novas funções para o sistema, tornando-o assim mais completo.

A seguir serão destacados alguns pontos que poderão contribuir para a melhoria do sistema, podendo assim aumentar seu público alvo e torna-lo cada vez mais eficiente e seguro.

* Avaliação do serviço – Permitir que o usuário avalie e sugira melhorias para o sistema.
* Envio de e-mails de confirmação de cadastro e alteração de senha.
* Aumentar a interação dos alunos e professores – permitir a criação de simulados para os alunos, promovendo assim uma maior interação entre eles.
* Permitir o compartilhamento de questões entre os docentes cadastrados.
* Interação de usuários – o sistema poderá permite a interação de um usuário com os outros, ao passo que poderão avaliar questões, além de poderem comentar.

Com o fim deste trabalho pode-se concluir que o mesmo foi de extrema importância para a vida acadêmica e profissional, ao passo que foi possível colocar em prática conhecimentos adquiridos no curso, além de aumentar o leque de aprendizados, na medida em que foram realizadas as pesquisas.

**REFERÊNCIAS**

LARMAN, Craing. **Utilizando UML e Padrões**. 3 Ed. São Paulo: Bookman Companhia Ed, 2007.

LUFTMAN, J.N.; LEWIS, P.R.; OLDACH, S.H.: Transforming The Enterprise: The Alignment Of Business And Information Technology Strategies. **IBM Systems Journal**, v.32, n.1, p.198-221, 1993.

METLAPALLI. **Páginas Java Server(jsp)**. Rio de Janeiro: Livros Técnicos e Científicos, 2010.

SOMMERVILLE, I. **Engenharia de Software.** 8. ed. São Paulo: Pearson Addison-Wesley, 2007. ISBN 978-85-88639-28-7.

GUEDES, Gilleanes T. A**. UML 2 - Uma abordagem prática**. 1. Ed. Novatec Editora LTDA: Rubens Prates, 2009. ISBN: 978-85-7522-193-8.

DEMETRIO, A. Paula; VIEIRA, J. Apolinario; SOARES, St. Batista. **Central do TCC Informática**. 2012.

ORACLE. **Java Software**. Disponível em: <https://www.oracle.com/java/index.html> Acesso em: 03 de jan. 2016.

MARTINEZ, Mariana. **UML**. Disponível em: <http://www.infoescola.com/engenharia-de-software/uml/> Acesso em: 07 de jan. 2016.

NETBEANS. **NetBeans IDE - A Forma Mais Inteligente e Rápida de Codificar**. Disponível em: <https://netbeans.org/features/index\_pt\_BR.html>. Acesso em: 5 de setembro de 2015.

TECHTUDO. **Apache Tomcat; Ferramenta ajuda o usuário a criar sites** Disponível em: <http://www.techtudo.com.br/tudo-sobre/apache-tomcat.html>. Acesso em: 5 de janeiro de 2015.

FARINELLI, F. **Conceitos Básicos De Programação Orientação a Objetos.** Disponível em: <http://sistemas.riopomba.ifsudestemg.edu.br/dcc/materiais/1662272077POO.pdf>. Acesso em: 10 out. 2015.

# SIGNIFICADOS. **Significado de Brainstorming.** Disponível em: <http://www.significados.com.br/brainstorming/> acesso em 15 de out. 2015.

# PORTAL EDUCAÇÃO. **Fundamentação Teórica.** Disponível em: <http://www.portaleducacao.com.br/pedagogia/artigos/31156/fundamentacao-teorica> Acesso em 15 de out. 2015.

# BARBOSA, Igor. **O QUE É BRAINSTORM?** Disponivel em: < http://issomesmo.com/o-que-e-brainstorm/>. Acesso em: 4 jan. 2016.

# BELEM, Thiago. **Mas afinal, o que é o MVC?**. Disponível em: <http://blog.thiagobelem.net/o-que-e-o-mvc>Acesso em 12 de jan. 2016.

ALTER, S. **Information Systems:** a management perspective. 3 Ed. San Francisco. Addison-Wesley Publishing Co. Massachusetts, 1992.

ANDERSON, J.: FRANCESCHI, H. **JAVA 6 – Uma Abordagem Ativa de Aprendizagem.** 2. Ed. Rio de Janeiro: LTC, 2010.

HENDERSON, J.C.; VENKATRAMAN, N. Strategic Alignment: Leveraging Information Technology For Transforming Organizations**. IBM Systems Journal,** v.32, n.1, p.4-16, 1993.

KEEN, P.G.W.: Information Technology And The Management Theory: The Fusion Map. **IBM Systems Journal**, v.32, n.1, p.17-38, 1993.

SCHACH, S. R.**Engenharia de Software:** Os Paradigmas Clássicos Orientado A

Objetos. 7 Ed. São Paulo: McGraw-Hill, 2009**.**

SPÍNOLA, M.M; PESSÔA, M.S.P. Tecnologia da Informação. In: CONTADOR, J.C. (coord.). **Gestão de Operações** – a Engenharia de Produção a serviço da modernização da empresa. São Paulo: Editora Edgard Blücher, 1997.

STONER, J. A. F. **Administração**. 5. ed. Rio de Janeiro: LTC, 1999.

MORAN, José Manuel, MASETTO, Marcos, BEHRENS, Marilda. **Novas tecnologias e mediação pedagógica**. 6. ed. São Paulo: Papirus, 2003.

# NETBEANS. **Determinando um perfil de uma Aplicação Corporativa no NetBeans IDE**. Disponível em: < https://netbeans.org/kb/docs/javaee/profiler-javaee\_pt\_BR.html> Acesso em: 24 de mar. 2016.

# NETBEANS. **Estudo de caso do perfilador NetBeans**. Disponível em: <https://netbeans.org/competition/win-with-netbeans/case-study-nb-profiler\_pt\_BR.html>. Acesso em: 24 de mar. 2016.