



Universidade Federal da Paraíba

Centro de Informática

Métodos de Projeto de Software

# Estude Aqui

*Plataforma para vestibulandos*

Nycholas de Sousa - 11228201

Victor Batista Maia - 11228399

# Histórico de Revisões

<b>Data</b>	<b>Versão</b>	<b>Descrição</b>	<b>Autor(es)</b>
31/08/2017	0.01	Versão Inicial	Nycholas de Sousa Victor Batista Maia
15/11/2017	0.02	Maiores informações sobre Projeto	Nycholas de Sousa Victor Batista Maia
17/11/2017	0.03	Revisão proposta pelo Professor - 1	Nycholas de Sousa Victor Batista Maia
23/11/2017	0.04	Brainstorm 3	Nycholas de Sousa Victor Batista Maia
24/11/2017	0.05	Refeito documento - Usando linguagem Java	Nycholas de Sousa Victor Batista Maia
25/11/2017	0.06	Revisão proposta pelo Professor - 2	Nycholas de Sousa Victor Batista Maia
29/11/2017	1.00	Versão Final do Projeto	Nycholas de Sousa Victor Batista Maia
03/12/2017	1.01	Correção no diagrama e inserção da descrição dos padrões de projeto propostos	Nycholas de Sousa Victor Batista Maia

# Sumário

<b>Sumário</b>	<b>3</b>
<b>1. Introdução</b>	<b>7</b>
1.1. Propósito do Documento	7
1.2. Visão Geral do Problema	7
1.3. Informações sobre o Documento	7
1.4. Documentos Relacionados	8
<b>2. Descrição Geral</b>	<b>9</b>
2.1. Motivação	9
2.2. Problemas Identificados	9
2.3. Visão Geral do Sistema	9
2.3.1. Nome da aplicação	9
2.3.2. Descrição	9
2.3.3. Github	10
2.4. Usuários do Sistema	10
<b>3. Elicitação de Requisitos</b>	<b>11</b>
3.1. Brainstorming	11
3.1.1 [BST001] - Primeira sessão de brainstorm.	11
3.1.2. [BST002] - Segunda sessão de brainstorm.	12
3.1.3. [BST003] - Terceira sessão de brainstorm.	12
3.2. Questionário	13
3.3. Comparação com Softwares Concorrentes	14
<b>4. Análise de Requisitos</b>	<b>15</b>
4.1. Requisitos Funcionais	15

4.1.1. [RF001] – Usuário para Administração	15
4.1.2. [RF002] – Relatório de Administração	15
4.1.3. [RF003] – Cadastro de Usuário	16
4.1.4. [RF004] – Exibição de Questionários	16
4.1.5. [RF005] - Relatório de Usuário	16
4.1.6. [RF006] – Login	16
4.1.7. [RF007] – Notificação	16
4.1.8. [RF008] – Importar Questões	17
4.1.9. [RF009] – Responder Questionário	17
4.2. Requisitos Não Funcionais	17
4.2.1. [RNF001] - Persistência	17
4.2.2. [RNF002] - Tecnologia para Implementação	17
4.2.3. [RNF003] - Tecnologia para Relatórios	18
4.2.4. [RNF004] - Finalizar Sessão	18
4.2.5. [RNF005] - Plataforma	18
4.2.6. [RNF006] - Tempo de Resposta	18
4.2.7. [RNF007] - Privacidade das Informações	18
4.2.8. [RNF008] - Acesso Simultâneo	19
4.2.9. [RNF009] - Disponibilidade	19
4.2.10. [RNF010] - Tecnologia para Notificações	19
<b>5. Casos de Uso</b>	<b>20</b>
5.1. Diagrama de Casos de Uso	20
5.2. Descrição dos Casos de Uso	21
5.2.1. [UC001] Cadastrar Usuário	21
5.2.2. [UC002] Exibir Questionário	22
5.2.3. [UC003] Importar Questões	23

<b>6. Diagrama de Classes</b>	<b>24</b>
<b>7. Diagrama Entidade Relacionamento</b>	<b>25</b>
<b>8. Diagrama Arquitetural</b>	<b>26</b>
8.1. Diagrama de Alto Nível	26
8.2. Diagrama de Baixo Nível	26
<b>9. Padrões de Projeto</b>	<b>27</b>
9.1. Padrões de Projeto Obrigatórios	27
9.1.1. Adapter	27
9.1.2. Template Method	27
9.1.3. Facade	27
9.1.4. Factory	28
9.1.5. Command	28
9.1.6. Memento	28
9.2. Padrões de Projeto Adicionais	29
9.2.1. Singleton	29
9.2.2. Observer	29
9.2.3. State	29
9.2.4. Composite	30
<b>10. Glossário</b>	<b>31</b>
A	31
B	31
J	31
M	31
N	31
P	32
S	32



# 1. Introdução

Esta introdução fornece as informações necessárias para fazer um bom uso deste documento, explicitando seus objetivos e as convenções que foram adotadas na escrita do texto, além de conter uma lista de referências para outros documentos relacionados.

## 1.1. Propósito do Documento

Este documento descreve as técnicas, bem como os requisitos e casos de uso utilizados para elicitar, analisar e especificar a construção do software Estude Aqui. Software utilizado para juntar conteúdos utilizados por vestibulandos.

## 1.2. Visão Geral do Problema

Encontrar conteúdo e simulados para o vestibular às vezes pode ser desgastante, visto que o vestibulando muitas vezes não encontra tudo em apenas um local. O Estude Aqui tem como intuito erradicar essa dificuldade, tornando-se uma plataforma onde contenha os questionários/simulados necessários para que o vestibulando possa estudar.

## 1.3. Informações sobre o Documento

**Introdução** - fornece informações sobre o documento de requisitos, apresentando seu propósito e sua visão geral.

**Descrição Geral** - fornece uma explicação breve sobre o produto de software a ser produzido e suas restrições gerais.

**Elicitação de Requisitos** - lista e comenta sobre as técnicas de levantamento de requisitos utilizadas para a elaboração deste documento.

**Análise de Requisitos** - apresenta os requisitos funcionais e não funcionais que devem ser atendidos no desenvolvimento da aplicação.

**Especificação de Requisitos** - determina os casos de uso, faz uma descrição dos principais casos de uso e apresenta o diagrama de casos de uso do sistema.

**Análise de Casos de Uso (Diagrama de Classes de Análise)** - descreve os vários tipos de objetos (classes) no sistema e o relacionamento entre eles.

**Glossário** - fornece uma lista de palavras encontradas neste documento, visando seu melhor entendimento e esclarecimento.

## 1.4. Documentos Relacionados

1. KENSKI, Vani Moreira. Educação e tecnologias – 2ª Edição. Local Campinas, Papirus, 2007.
2. Requisitos Funcionais e Não Funcionais. Acessado em Setembro de 2017. Disponível em: <https://sites.google.com/site/nyxsilva/requisitosfuncionaisen%C3%A3ofuncionais>
3. MELO, Ana Cristina de Souza Luiz. Exercitando modelagem em UML. Editora Brasport.
4. Notas de aula do Professor Raoni Kulesza.
5. Padrões de Projeto. Acessado em Novembro de 2017. Disponível em: <https://brizenowordpress.com/>



## **2. Descrição Geral**

A comunicação digital é o conjunto de normas relacionadas com o comportamento adequado e responsável no uso das tecnologias. Ela institui uma nova forma de comunicação afetando o conjunto das relações sociais, não apenas as estritamente comunicacionais, mas em todos os níveis, na comunicação relações pessoais, interpessoais, no trabalho, nas instituições, na indústria... Ainda segundo Antonio B Duarte Jr, diretor da Arth Informática, "A tecnologia digital oferece todas as possibilidades já exploradas na imprensa escrita, no rádio e na televisão, com duas vantagens: a velocidade e a interação. O indivíduo não fica somente no papel de receptor passivo, há a possibilidade de escolha, há decisões a serem tomadas." Não há hoje uma única força produtiva que não esteja, direta ou indiretamente, engajada em algum tipo de relação de comunicação digital.

### **2.1. Motivação**

Estudar através de questionários sempre foi uma ótima forma de estudo, trazendo mais benefícios do que apenas lendo os assuntos. Devido os vestibulares, algumas pessoas têm tendência a estudar dessa forma, assim, à medida que vai estudando, já pode conferir o seu aproveitamento no estudo e notar em qual área precisa dar mais atenção.

### **2.2. Problemas Identificados**

Com o grande fluxo de informações na internet, separar apenas o que será necessário para estudar sem com que se repita a fonte ou que uma fonte use outra se torna algo trabalhoso e cansativo. Por isso hoje em dia, estudar utilizando a internet pode tomar mais tempo separando os assuntos e os organizando do que estudando realmente, além do mais, devido a quantidade de informações na internet, não chegar na informação desejada o quanto antes pode acabar distraindo o usuário.

### **2.3. Visão Geral do Sistema**

#### **2.3.1. Nome da aplicação**

O nome escolhido para a aplicação é "Estude Aqui".

#### **2.3.2. Descrição**

Através de seu computador, o usuário pode usar o software para estudar para vestibulares sem ter que recorrer à várias abas ou sites na internet, redirecionando o tempo que gastaria pesquisando em uma sessão de estudo.

### 2.3.3. Github

O projeto está sendo armazenado em um site de repositórios, o github. Para mais informações sobre o projeto, assim como seu código, acesse o seguinte link: [https://github.com/nycholassousa/MPS\\_EstudeAqui](https://github.com/nycholassousa/MPS_EstudeAqui)

## 2.4. Usuários do Sistema

O sistema possui apenas dois tipos de usuários:

- **Vestibulando:** é aquele que usará o sistema para responder os questionários. Ele é capaz de criar conta, logar-se responder questionários quantas vezes achar necessário e ver um relatório de seu aproveitamento.
- **Administrador:** é aquele que irá cuidar de algumas questões internas do sistema, principalmente de adicionar novas questões ao banco de dados. O administrador também pode ser um vestibulando.

## 3. Elicitação de Requisitos

Para elicitar os requisitos do software, foram utilizadas duas técnicas: o brainstorming e a entrevista. Estas técnicas foram utilizadas por serem ágeis, uma vez que o tempo para elicitação teve de ser otimizado.

### 3.1. Brainstorming

Foram realizadas três reuniões presenciais de brainstorming com o intuito de definir o escopo e as tecnologias que devem ser utilizadas para o desenvolvimento. As notas das reuniões podem ser encontradas abaixo.

#### 3.1.1 [BST001] - Primeira sessão de brainstorm.

**Participantes:** Nycholas de Sousa, Victor Batista Maia

**Data:** 14/08/2017

**Foco da Discussão:**

- Definição do problema e qual seu escopo
- Decisão sobre quais métodos de elicitação utilizar
- Elaboração do Questionário
- Divisão de Atividades

**Resumo:**

Nesta primeira reunião o principal objetivo foi decidir com qual problema iríamos trabalhar como poderíamos solucioná-lo. Após alguns problemas citados e lembrar de experiências passadas, optamos por criar um software que pudesse ajudar vestibulandos, para que não ocorresse as mesmas dificuldades que encontramos antes de entrar na graduação. Com isso, analisamos a proposta e suas limitações, assim como o escopo do projeto. Após isso, analisamos quais seriam as melhores formas de elicitação de requisitos, optando então pelo brainstorming e questionário. Após isso, as atividades foram divididas entre os membros da equipe.

### **3.1.2. [BST002] - Segunda sessão de brainstorm.**

**Participantes:** Nycholas de Sousa, Victor Batista Maia

**Data:** 21/08/2017

**Foco da Discussão:**

- Análise do questionário respondido
- Refinamento da ideia
- Elencar requisitos funcionais do sistema

**Resumo:**

Com o questionário respondido entre os estudantes, foi-se decidido qual seria a linguagem de programação que seria utilizado no software, assim, foi-se optado por Javascript. Para melhorarmos a ideia, foi-se decidido como funciona o sistema, se seria apenas questionários, ou se o software faria conexão com sites que possuem conteúdos para vestibulandos. Assim, decidiu-se que seria apenas questionários, para que não se torne algo tão extenso. No que diz respeito a elencar os requisitos funcionais, elencamos de início apenas requisitos básicos, como responder questionários, cadastro/login, relatório de usuário.

### **3.1.3. [BST003] - Terceira sessão de brainstorm.**

**Participantes:** Nycholas de Sousa, Victor Batista Maia

**Data:** 23/11/2017

**Foco da Discussão:**

- Mudança da linguagem de software
- Adição de requisitos funcionais e não-funcionais

**Resumo:**

Após uma série de discussões, notou-se que fazer o sistema proposto em web seria difícil, tendo em vista tanto o prazo final, e que os padrões de projetos propostos são mais difíceis em Javascript. Com isso, foi decidido manter a ideia proposta, mas mudar apenas a linguagem de programação inicial, que tinha Javascript como base. Agora, a linguagem é Java para fazer o software. As mudanças que ocorreram com tais decisões foram: Não é mais um website, e sim um programa de computador. Sobre a adição de mais requisitos, foi decidido que seria necessário mais informações para o software, assim, gerando mais requisitos funcionais e não-funcionais.

## 3.2. Questionário

Com essas ideias em mãos foi selecionado um vestibulando para entrevistar.

**Data:** 23/08/2017

**Local:** Colégio Paraíso – Juazeiro do Norte, Ceará

**Perfil dos usuários:** Alunos do 3º ano do ensino médio e alunos de cursos pré-vestibular

Indicação de data	Você tem dificuldade de juntar os assuntos necessários para o vestibular?	O quão inclinado você está em estudar em apenas um programa, juntando os assuntos possíveis através de questionários ou provas passadas?	Com que frequência você utilizaria um programa que disponibiliza tal serviço? Classifique de 1 a 5.	Você compartilharia a aplicação com seus amigos?
31/08/2017	3	5	3	Sim
31/08/2017	4	3	3	Sim
31/08/2017	4	5	5	Sim
31/08/2017	3	5	3	Sim
31/08/2017	3	3	4	Sim
31/08/2017	3	3	5	Sim
31/08/2017	4	5	5	Sim
31/08/2017	3	4	3	Sim
31/08/2017	5	2	2	Sim

Resultando no seguinte gráfico:

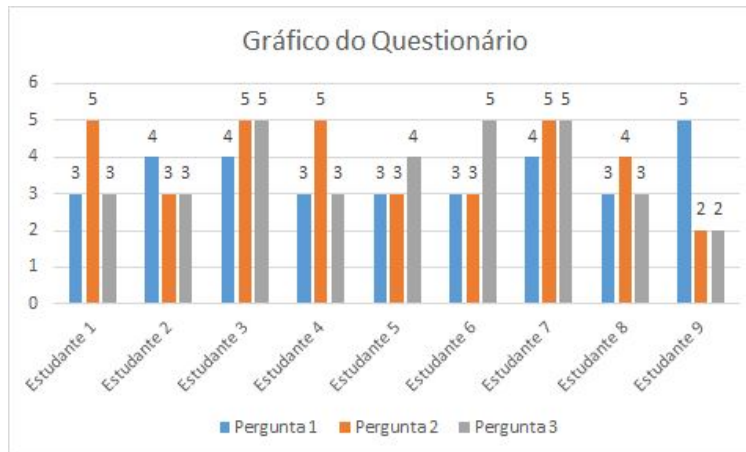


FIGURA 1 - Gráfico Representativo do Questionário

A partir desses levantamentos concluímos que existe demanda para um sistema nos moldes que pensamos. Um programa que disponibilize os assuntos usados em vestibular através de questionários, utilizando provas de universidades ou simulados.

Apesar que algumas das respostas do questionário indicaram que alguns usuários não fariam uso da aplicação, eles estariam fortemente inclinados à compartilhar a aplicação com outras pessoas.

### 3.3. Comparação com Softwares Concorrentes

Para melhor desenvolvimento do programa, foi realizado uma série de comparações entre as idéias listadas aqui e entre 3 aplicações concorrentes, os tópicos de comparação foram alguns requisitos funcionais que estão listados na seção seguinte.

	Estude Aqui	Concorrente 1	Concorrente 2	Concorrente 3
RF003	Sim	Não	Não	Não
RF004	Sim	Sim	Sim	Sim
RF005	Sim	Sim	Sim	Sim
RF006	Sim	Não	Não	Sim
RF007	Sim	Não	Sim	Não

Com isso, pode-se notar que o software “Estude Aqui” seria algo diferente e interessante, proporcionando mais um meio de estudo para vestibulandos.

## 4. Análise de Requisitos

A presente seção descreve e classifica os principais requisitos funcionais e não funcionais presentes no sistema em questão. Para estabelecer a prioridade dos requisitos, foram adotadas as denominações essencial, importante e desejável.

**Essencial:** é o requisito com o qual o sistema não entrará em funcionamento ou não atende minimamente a intenção do projeto. Isto é, são requisitos fundamentais e que precisam ser implementados.

**Importante:** é o requisito sem o qual o sistema entra em funcionamento, mas de forma não satisfatória. Requisitos importantes devem ser implantados o mais rápido possível, mas, se não forem, parte do sistema poderá ser implantada mesmo assim.

**Desejável:** é o requisito que não compromete as funcionalidades básicas do sistema, isto é, o sistema pode funcionar de forma satisfatória sem ele. Requisitos desejáveis são requisitos que podem ser implantados por último, sem comprometer o funcionamento do sistema.

### 4.1. Requisitos Funcionais

A partir da elicitación foram levantados os seguintes requisitos funcionais para a aplicação:

#### 4.1.1. [RF001] – Usuário para Administração

O sistema terá um usuário que será identificado como administrador. Para identificar tal administrador, será feita uma comparação no banco de dados.

**Prioridade:** Importante

**Casos de Uso Relacionados:** Cadastrar Usuário; Realizar Login

#### 4.1.2. [RF002] – Relatório de Administração

O administrador terá acesso a relatórios do sistema, que podem ser: dados sobre usuários, dados sobre as questões e logs feitos no sistema.

**Prioridade:** Importante

**Casos de Uso Relacionados:** Exibir Relatório

#### **4.1.3. [RF003] – Cadastro de Usuário**

O sistema deve cadastrar usuários e salvá-los no banco de dados MySQL.

**Prioridade:** Essencial

**Casos de Uso Relacionados:** Cadastrar Usuário

#### **4.1.4. [RF004] – Exibição de Questionários**

Será exibido os questionários aos usuários para que respondam e caso queiram, refaçam.

**Prioridade:** Essencial

**Casos de Uso Relacionados:** Ver Questionário

#### **4.1.5. [RF005] - Relatório de Usuário**

O sistema deve gerar relatórios para o usuário, assim, o mesmo pode ter conhecimento de como está sendo suas sessões de estudos e o aproveitamento de suas seções.

**Prioridade:** Essencial

**Casos de Uso Relacionados:** Exibir Relatório

#### **4.1.6. [RF006] – Login**

O usuário deverá fazer o login utilizando seu email para entrar no sistema, caso o mesmo já possua cadastro.

**Prioridade:** Essencial

**Casos de Uso Relacionados:** Cadastrar Usuário; Realizar Login

#### **4.1.7. [RF007] – Notificação**

O usuário receberá uma notificação por email ao entrar no programa, caso novas questões sejam adicionadas no sistema, possibilitando novas sessões de estudo.

**Prioridade:** Desejável

**Casos de Uso Relacionados:** Realizar Login; Notificar Novas Questões



#### **4.1.8. [RF008] – Importar Questões**

O administrador poderá importar novas questões para o banco de dados, utilizando dois tipos de arquivo: JSON ou CSV.

**Prioridade:** Desejável

**Casos de Uso Relacionados:** Importar Questões; Notificar Novas Questões

#### **4.1.9. [RF009] – Responder Questionário**

O usuário após estar logado no sistema, poderá responder o questionário, caso queira.

**Prioridade:** Essencial

**Casos de Uso Relacionados:** Ver Questionário

### **4.2. Requisitos Não Funcionais**

A partir da elicitação foram levantados os seguintes requisitos não funcionais para a aplicação:

#### **4.2.1. [RNF001] - Persistência**

O sistema deve ser capaz de armazenar informações das questões e dos usuários, utilizando banco de dados em MySQL, versão 5 ou superior.

**Prioridade:** Essencial

**Classificação:** Requisito de interoperabilidade

#### **4.2.2. [RNF002] - Tecnologia para Implementação**

A aplicação deve utilizar Java versão 1.8.0\_131 ou superior. Além do mais, para o funcionamento, é necessário que tenha o Java Runtime Environment na versão 1.8.0\_131-b11 ou superior, assim como o Java Hotspot Client VM, versão 25.131-b11 ou superior.

**Prioridade:** Essencial

**Classificação:** Requisito de implementação

### **4.2.3. [RNF003] - Tecnologia para Relatórios**

O sistema de relatórios fará uso da biblioteca PDFOne, versão 5.4.863.533 versão trial.

**Prioridade:** Essencial

**Classificação:** Requisito de implementação

### **4.2.4. [RNF004] - Finalizar Sessão**

Após ter o aplicativo fechado pelo usuário, o sistema deve finalizar a sessão inicializada anteriormente pelo usuário.

**Prioridade:** Desejável

**Classificação:** Requisito de implementação

### **4.2.5. [RNF005] - Plataforma**

O aplicativo deve ser executável sem problemas em sistemas operacionais Windows Vista ou mais recente, ou no Linux com Kernel 4.4.8 ou superior.

**Prioridade:** Essencial

**Classificação:** Requisito de portabilidade

### **4.2.6. [RNF006] - Tempo de Resposta**

O aplicativo deve retornar ao usuário de forma rápida qualquer estado requisitado (nova questão para responder, relatório, fazer login, etc). O usuário não deve aguardar mais que 3 segundos.

**Prioridade:** Essencial

**Classificação:** Requisito de desempenho

### **4.2.7. [RNF007] - Privacidade das Informações**

Os dados fornecidos pelo usuário não deverão ser transmitidos a terceiros, com seu uso restrito ao próprio aplicativo.

**Prioridade:** Essencial

**Classificação:** Requisito de privacidade

#### **4.2.8. [RNF008] - Acesso Simultâneo**

O sistema deve ser capaz de atender até 50 requisições simultaneamente.

**Prioridade:** Essencial

**Classificação:** Requisito de confiabilidade

#### **4.2.9. [RNF009] - Disponibilidade**

O banco de dados deve estar disponível 95% do tempo para o uso do sistema.

**Prioridade:** Essencial

**Classificação:** Requisito de confiabilidade

#### **4.2.10. [RNF010] - Tecnologia para Notificações**

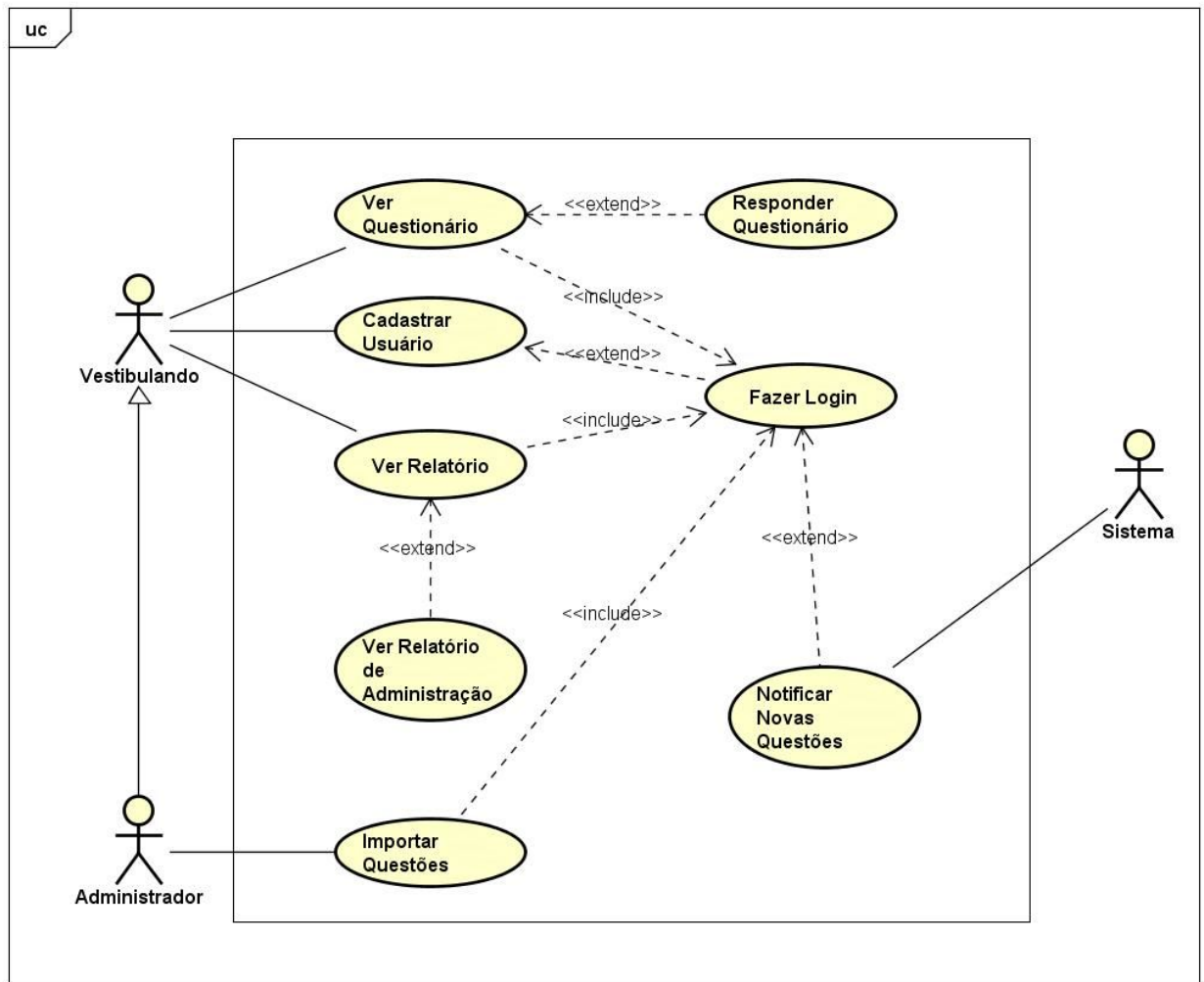
O sistema de notificações usará a API JavaMail versão 1.4.5

**Prioridade:** Importante

**Classificação:** Requisito de implementação

# 5. Casos de Uso

## 5.1. Diagrama de Casos de Uso



powered by Astah

FIGURA 2 - Diagrama dos Casos de Uso do Sistema

Caso queira ver a imagem com uma melhor qualidade, acesse o seguinte link:  
[https://raw.githubusercontent.com/nycholassousa/MPS\\_EstudeAqui/master/imgs/use.jpg](https://raw.githubusercontent.com/nycholassousa/MPS_EstudeAqui/master/imgs/use.jpg)

## 5.2. Descrição dos Casos de Uso

### 5.2.1. [UC001] Cadastrar Usuário

<b>Identificador</b>	[UC001]
<b>Nome</b>	Cadastrar Usuário
<b>Descrição</b>	Criar uma conta de usuário
<b>Requisitos</b>	RF003
<b>Ator(es)</b>	Usuário
<b>Prioridade</b>	Essencial
<b>Include</b>	
<b>Extend</b>	1. Fazer login - Após Cadastrado, o usuário pode ou não realizar o login no sistema
<b>Pré-condições</b>	Ser a primeira vez que usa o software
<b>Pós-condições</b>	Os dados de login do usuário serão salvos no banco de dados
<b>Frequência de Uso</b>	Normal
<b>Fluxo Principal [FP]</b>	<ol style="list-style-type: none"><li>1. O usuário abre o software.</li><li>2. Sistema exibe tela inicial</li><li>3. Usuário seleciona a opção de Realizar Cadastro</li><li>4. Sistema solicita dados obrigatórios a serem preenchidos pelo usuário</li><li>5. Usuário autoriza o seu cadastro, inserindo os dados solicitados</li><li>6. O usuário é redirecionado para a tela principal sobre o software</li></ol>
<b>Fluxo Secundário [FS]</b>	<ol style="list-style-type: none"><li>1. Usuário preenche formulário de cadastro com dados inválidos e/ou não preenche um ou mais espaços obrigatórios</li><li>2. Sistema informa quais espaços foram preenchidos de forma inválida e irá solicitar que Usuário preencha novamente</li><li>3. Sistema reseta a tela para que Usuário preencha tudo novamente.</li></ol>

### 5.2.2. [UC002] Exibir Questionário

<b>Identificador</b>	[UC002]
<b>Nome</b>	Iniciar Questionário
<b>Descrição</b>	Iniciar um questionário ao usuário, solicitando as respostas corretas de uma pergunta de cada vez
<b>Requisitos</b>	RF004; RF006; RF009
<b>Ator(es)</b>	Usuário
<b>Prioridade</b>	Essencial
<b>Include</b>	1. Realizar Login
<b>Extend</b>	1. Responder Questionário
<b>Pré-condições</b>	O usuário precisa estar logado
<b>Pós-Condições</b>	Caso responda uma questão: 1. Se houver mais questões a serem respondidas, o usuário será solicitado para responder uma nova questão. 2. Caso não haja mais questões, informar ao usuário que este finalizou todas as perguntas e caso o usuário queira, ver o relatório de desempenho.
<b>Frequência de uso</b>	Alta
<b>Fluxo Principal [FP]</b>	1. O usuário, com login realizado, solicita iniciar o questionário 2. O usuário responde ou pula, uma após a outra, as questões mostradas na tela 3. Quando não houver mais perguntas a serem respondidas, o Sistema irá informar este fato ao Usuário. 4. Caso o usuário deseje, ele poderá ver um relatório do seu aproveitamento em relação ao questionário

### 5.2.3. [UC003] Importar Questões

<b>Identificador</b>	[UC003]
<b>Nome</b>	Importar Questões
<b>Descrição</b>	Adicionar uma ou mais questões ao questionário acessado pelos usuários.
<b>Requisitos</b>	RF006; RF008
<b>Ator(es)</b>	Administrador
<b>Prioridade</b>	Importante
<b>Include</b>	1. Fazer Login
<b>Extend</b>	
<b>Pré-condições</b>	<ol style="list-style-type: none"><li>1. O administrador precisa estar logado em uma conta de administrador do sistema</li><li>2. O administrador deve ter o arquivo .JSON ou .CSV na formatação correta</li></ol>
<b>Pós-condições</b>	<ol style="list-style-type: none"><li>1. Após adicionar uma questão, o administrador deve informar ao sistema se deseja adicionar mais uma questão ou não. Caso escolha a segunda opção, volta para a página inicial do sistema.</li></ol>
<b>Frequência de Uso</b>	Média
<b>Fluxo Principal [FP]</b>	<ol style="list-style-type: none"><li>1. O administrador faz seu login no aplicativo</li><li>2. Solicita para adicionar mais questões</li><li>3. O administrador envia ao sistema o arquivo .JSON ou o arquivo .CSV correspondente à questão</li><li>4. O administrador escolhe se decide importar mais questões</li><li>5. Caso queira importa mais questões, volta pro passo 3</li><li>6. Caso não deseje importar mais questões, volta para a tela inicial do programa</li></ol>
<b>Fluxo Secundário [FS]</b>	<ol style="list-style-type: none"><li>1. Administrador envia um arquivo de formato diferente de .JSON ou de .CSV</li><li>2. Sistema informa ao administrador que o arquivo é inválido</li><li>3. Administrador escolhe se deseja enviar novamente ou cancelar a importação da questão.</li></ol>

## 6. Diagrama de Classes

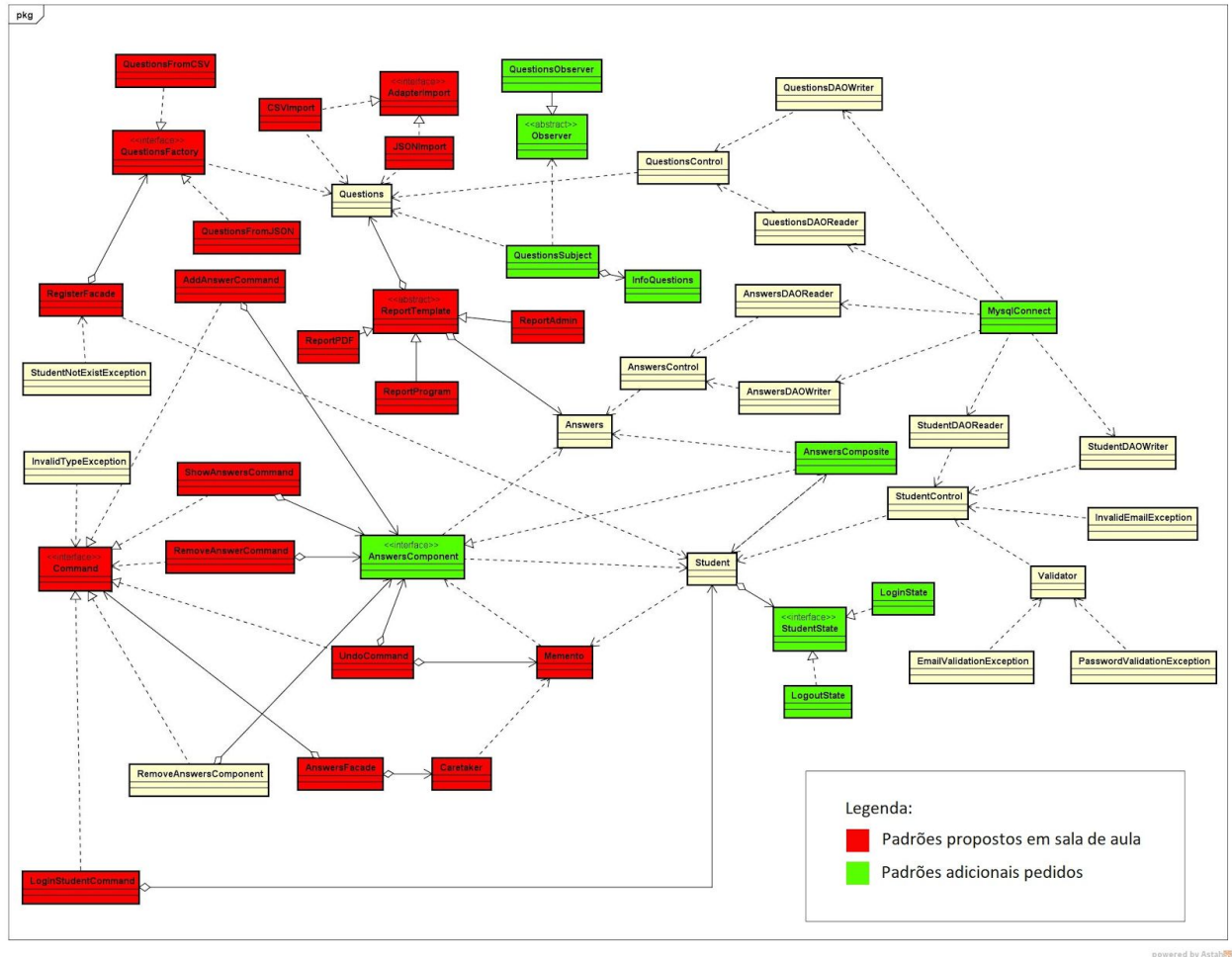


FIGURA 3 - Diagrama de Classes do Sistema

Caso queira ver a imagem com uma melhor qualidade, acesse o seguinte link:  
[https://raw.githubusercontent.com/nycholassousa/MPS\\_EstudeAqui/master/imgs/class.jpg](https://raw.githubusercontent.com/nycholassousa/MPS_EstudeAqui/master/imgs/class.jpg)



## 7. Diagrama Entidade Relacionamento

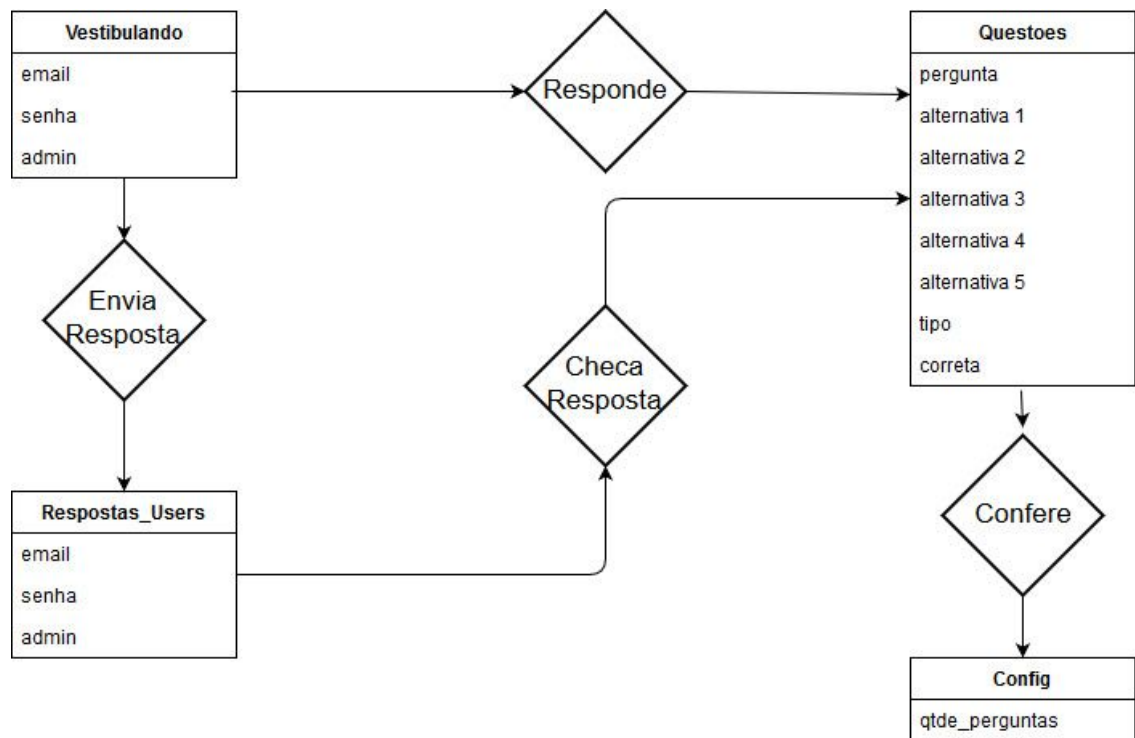


FIGURA 4 - Diagrama de Entidade Relacionamento do Sistema

## 8. Diagrama Arquitetural

### 8.1. Diagrama de Alto Nível

O diagrama de alto nível é simples, o usuário, através do aplicativo, fará conexão com a internet, onde através dela, se conecta ao banco de dados, assim, podendo responder as questões ou apenas ver o relatório de desempenho.

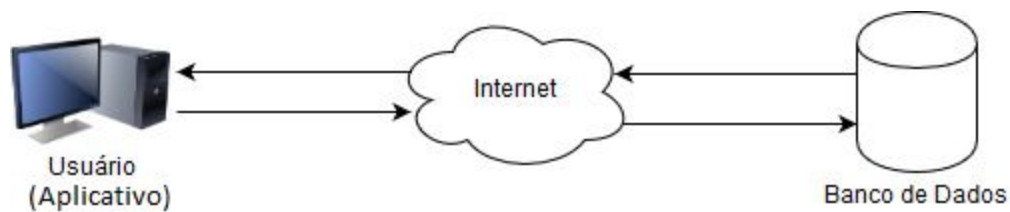


FIGURA 5 - Diagrama Arquitetural de Alto Nível

### 8.2. Diagrama de Baixo Nível



FIGURA 6 - Diagrama Arquitetural de Baixo Nível

## 9. Padrões de Projeto

### 9.1. Padrões de Projeto Obrigatórios

#### 9.1.1. Adapter

<b>Pacote</b>	infra.adapter
<b>Classe(s)</b>	AdapterImport; JsonImport; CsvImport
<b>Funcionamento</b>	Padrão de projeto usado utilizado para fazer a importação baseado no arquivo, como JSON e CSV diferem entre si mas tem o mesmo intuito, o padrão de projeto neste caso torna-se conveniente.

#### 9.1.2. Template Method

<b>Pacote</b>	business.model.report
<b>Classe(s)</b>	ReportTemplate; ReportAdmin; ReportPDF; ReportProgram
<b>Funcionamento</b>	Padrão utilizado para criar os relatórios. Como cada tipo de relatório possui uma base, essa base pode ser utilizada como “esqueleto” e apenas as coisas adicionais serem usadas por cada classe. Assim, pode-se usar o template method.

#### 9.1.3. Facade

<b>Pacote</b>	business.control.facade
<b>Classe(s)</b>	AnswersFacade, RegisterFacade
<b>Funcionamento</b>	Padrão utilizado para iniciar um conjunto de instruções.

#### 9.1.4. Factory

<b>Pacote</b>	business.control.questions.factory
<b>Classe(s)</b>	QuestionsFactory; QuestionsFromCsv; QuestionsFromJson
<b>Funcionamento</b>	Utilizado para criar o objeto Questions, porém, a subclasse que instancia dizendo qual a origem desse objeto, se veio por um arquivo JSON ou por um arquivo CSV.

#### 9.1.5. Command

<b>Pacote</b>	business.control.commands
<b>Classe(s)</b>	AddAnswerCommand; Command; LoginStudentCommand RemoveAnswerCommand; ShowAnswersCommand; UndoCommand
<b>Funcionamento</b>	Padrão de projeto utilizado para uso de comandos, tais como: desfazer ação, adicionar respostas, remover respostas, login de usuário.

#### 9.1.6. Memento

<b>Pacote</b>	business.model.memento
<b>Classe(s)</b>	Caretaker; Memento
<b>Funcionamento</b>	Funcionando lado a lado com o padrão Command, usado para desfazer ação.

## 9.2. Padrões de Projeto Adicionais

### 9.2.1. Singleton

<b>Pacote</b>	infra.database
<b>Classe(s)</b>	MysqlConnect
<b>Funcionamento</b>	Com o padrão, apenas uma instância da classe é utilizada em todo o projeto. Como a classe lida apenas com a conexão SQL, o uso desse padrão torna-se algo conveniente.

### 9.2.2. Observer

<b>Pacote</b>	business.control.questions.observer
<b>Classe(s)</b>	Observer; InfoQuestions; QuestionsObserver; QuestionsSubject
<b>Funcionamento</b>	Este padrão tem como objetivo notificar o usuário caso novas questões sejam adicionadas ao banco de dados. Tal notificação acontece no momento do login, onde é feita uma comparação com um arquivo local que contém a quantidade de questões da última vez que foi aberto. Caso a quantidade seja diferente da quantidade do banco de dados, a notificação é lançada e o arquivo é atualizado.

### 9.2.3. State

<b>Pacote</b>	business.model.student.state
<b>Classe(s)</b>	StudentState; LoginState; LogoutState
<b>Funcionamento</b>	O uso deste padrão foi para definir qual o estado em que o usuário se encontra, ou seja, se está logado no sistema ou não.

### 9.2.4. Composite

<b>Pacote</b>	business.model.answers
<b>Classe(s)</b>	AnswersComponent; AnswersComposite
<b>Uso</b>	A ideia do Composite é criar uma classe base que contém toda a interface necessária para todos os elementos e criar um elemento especial que agrega outros elementos.

# 10. Glossário

## A

**API** – A Interface de Programação de Aplicação é um conjunto de rotinas e padrões estabelecidos por um software para a utilização das suas funcionalidades por aplicativos que não pretendem envolver-se em detalhes da implementação do software, mas apenas usar seus serviços.

**Ator** – Alguém ou algo fora do sistema que interage com ele.

## B

**Banco de dados** – Uma coleção de dados relacionados armazenados juntos, com redundância controlada de acordo com um esquema, para ser utilizado por um ou mais aplicativos.

## J

**Java Hotspot Client VM** – Máquina virtual Java para desktops e servidores.

**Java Runtime Environment** – Junção de APIs e a Máquina Virtual Java, utilizados para executar aplicações Java.

**Java** – Linguagem de programação orientada a objetos convencional da Plataforma Java.

**Javascript** – Linguagem de programação interpretada originalmente feita para criação de scripts em navegadores web.

## M

**Mysql** – Sistema de Gerenciamento de Banco de Dados (SGBD) popular que utiliza a linguagem SQL.

**Máquina Virtual Java** – Programa que carrega e executa os aplicativos Java.

## N

**NodeJS** – Interpretador de código Javascript que facilita na criação de aplicações mais complexas em navegadores web.

## P

**PDFOne** – Conjunto de API's para criar, manipular e organizar documentos PDF dentro de aplicações Java.

## S

**SGBD** – É um sistema cujo objetivo principal é gerenciar o acesso e a correta manutenção dos dados armazenados em um ou mais bancos de dados

**Software** – Programa; reunião dos procedimentos e/ou instruções que determinam o funcionamento de um computador.

**SQL** – Linguagem de pesquisa declarativa padrão para banco de dados.

## V

**Vestibulando** – Estudante que se prepara para prestar o exame vestibular.