

**FUNDAÇÃO OSWALDO ARANHA**  
**CENTRO UNIVERSITÁRIO DE VOLTA REDONDA**  
**CURSO DE SISTEMAS DE INFORMAÇÃO**  
**TRABALHO DE CONCLUSÃO DE CURSO**

**IGOR MARÇAL BARBOSA DOS SANTOS**  
**MAX YURI FERREIRA**  
**ROGER PINTO DA SILVA**  
**RÔMULO DE LACERDA CORRÊA RÍMULO**

***INTERNSHIPIN***: Plataforma de preparação do universitário para o  
mercado de trabalho.

**VOLTA REDONDA**  
**2017**

**FUNDAÇÃO OSWALDO ARANHA**  
**CENTRO UNIVERSITÁRIO DE VOLTA REDONDA**  
**CURSO DE GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO**  
**TRABALHO DE CONCLUSÃO DE CURSO**

***INTERNSHIPIN***: Plataforma de preparação do universitário para o mercado de trabalho.

Trabalho de Conclusão de Curso, denominado como Projeto Integrado, apresentado no Curso de Sistema de Informação do Centro Universitário de Volta Redonda, como requisito para obtenção do grau de Bacharel em Sistema de Informação.

**Alunos:**

Igor Marçal Barbosa dos Santos

Max Yuri Ferreira

Roger Pinto da Silva

Rômulo de Lacerda Correa Rímulo

**Orientadores:**

Prof. Me. Venicio Siqueira Filho

Prof. Esp. Adilson Gustavo do Espírito Santo

Prof. Me. Rosenclever Lopes Gazoni

**Coorientadores:**

Prof. Me. Antônio Carlos da Silva

Prof. Dr. Carlos Eduardo Costa Vieira

Prof<sup>a</sup>. Dra. Sirlei Aparecida de Oliveira

**VOLTA REDONDA**

**2017**

# FOLHA DE APROVAÇÃO

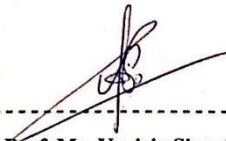
**INTERSHIPIN: PLATAFORMA DE PREPARAÇÃO DO UNIVERSITÁRIO PARA O  
MERCADO DE TRABALHO.**

**IGOR MARÇAL BARBOSA DOS SANTOS  
MAX YURI FERREIRA  
ROGER PINTO DA SILVA  
RÔMULO DE LACERDA CORRÊA RÍMULO**

TCC – Projeto Integrado apresentado no Curso de Sistemas de Informação, como requisito para obtenção do título Bacharel em Sistemas de Informação, pela Fundação Oswaldo Aranha.

Aprovado em 28 de Outubro de 2017.

Banca Examinadora



-----  
**Prof. Me. Venicio Siqueira Filho**  
**(Presidente da Banca Examinadora e Prof. Orientador)**



-----  
**Prof. Dr. Carlos Eduardo Costa Vieira**  
**(Coord. Curso de SI – Examinador Interno)**



-----  
**Profª. Me. Rita de Cássia Santos Carvalho**  
**(Examinadora Interna)**

Volta Redonda – RJ

## RESUMO

Um estudante de Tecnologia da Informação (TI), nos dias de hoje, ao tentar ingressar no mercado de trabalho, se depara com uma realidade: em certos casos, ele se encontra defasado em conhecimentos profissionais. Em muitos casos, o conhecimento adquirido em sala de aula não é suficiente para iniciar a sua carreira. As dificuldades começam durante as entrevistas de emprego, onde os estudantes não possuem confiança no conhecimento adquirido e caso ele consiga a vaga, outros problemas surgem para serem enfrentados pelos universitários quando procuram encaixar-se no mercado de trabalho, sem nenhuma experiência comprovada em alguma área ou projeto semelhante. O *INTERNSHIPIN* vem como uma proposta para sanar essa deficiência em experiência prática de um estudante da área de TI e tem como objetivo ser uma ferramenta intuitiva e responsiva para que os alunos troquem conhecimento e tenham acesso a diversos projetos para colocarem em prática o que aprenderam em sala de aula e ganharem confiança. O foco do projeto é propor um sistema amigável para que os usuários tenham facilidade ao usá-lo, além de ser eficaz e útil ao discente, procurando se tornar, assim, um ponto de encontro entre os universitários com interesses em comum em desenvolver seus conhecimentos e acumular experiências práticas. Através desse trabalho, pode-se acompanhar a experiência adquirida pela equipe ao aplicar a metodologia ativa *Design Thinking*, mostrando novas maneiras de organizar e de apresentar um serviço de qualidade. Aplicou-se também a técnica de *brainstorming*, gerando um processo de prototipagem de forma rápida para alcançar o resultado esperado. No desenvolvimento desta plataforma serão utilizadas diversas tecnologias, tais como: *HyperText Markup Language (HTML5)*, *Cascading Style Sheets 3 (CSS3)*, *JavaScript com JQuery* e *Ruby on Rails*.

**Palavras-chave:** Mercado de Trabalho, *Software Web*, Metodologia Ativa *Design Thinking*.

## **ABSTRACT**

*An IT student today, when trying to enter the job market, is faced with a reality: in some cases, he is lacking of professional knowledge. In many cases, the knowledge gained in classroom is not enough to start your career. Difficulties begin during job interviews, where students do not have confidence in the knowledge acquired and if they get the job, other problems arise to be faced by university students when they seek to fit into the job market, with no previous experience in any area or similar project. INTERNSHIPIN comes as a proposal to remedy this deficiency in practical experience of an IT student and aims to be an intuitive and responsive tool for students to exchange knowledge and have access to various projects to put into practice what they have learned in classroom and gain confidence. The focus of the project is to propose a friendly system so that users have an easy way to use it, as well as being effective and useful to the student, thus seeking to become a meeting point among university students with common interests in developing their knowledge and accumulate practical experiences. Through this work, one can follow the experience acquired by the team when applying the active methodology Design Thinking, showing new ways to organize and present a quality service. The brainstorming technique was also applied, generating a rapid prototyping process to achieve the expected result. In the development of this platform will be used various technologies such as HyperText Markup Language (HTML5), Cascading Style Sheets 3 (CSS3), JavaScript with JQuery and Ruby on Rails.*

**Keywords:** *Job Market, Web Software, Active Methodology, Design Thinking;*

## SUMÁRIO

1. INTRODUÇÃO.....	11
1.1 OBJETIVO GERAL .....	12
1.2 OBJETIVOS ESPECÍFICOS .....	12
1.3 MOTIVAÇÃO PARA DESENVOLVIMENTO DO PROJETO .....	12
1.4 ORGANIZAÇÃO DO PROJETO .....	13
1.5 RESULTADOS ESPERADOS .....	14
2 METODOLOGIA.....	15
3 REFERENCIAL TEÓRICO .....	18
3.1 METODOLOGIA ATIVA – <i>DESIGN THINKING</i> .....	18
3.2 EDUCAÇÃO E <i>T.I.</i> .....	20
3.3 <i>HTML5</i> .....	21
3.4 <i>CSS3</i> .....	22
3.5 <i>JAVASCRIPT E JQUERY</i> .....	22
3.6 <i>BOOTSTRAP</i> .....	23
3.7 <i>RESPONSIVE WEB DESIGN</i> .....	24
3.8 <i>RUBY</i> .....	25
3.9 <i>MVC</i> .....	25
3.10 <i>RUBY ON RAILS</i> .....	27
3.11 <i>POSTGRESQL</i> .....	29
3.12 <i>HEROKU</i> .....	30
3.13 <i>GIT</i> .....	31
3.14 <i>GITHUB</i> .....	32
4 GERENCIAMENTO DO PROJETO .....	33
4.1 LEVANTAMENTO DO ESCOPO .....	33
4.2 ESTRUTURA ANALÍTICA DO PROJETO – <i>EAP</i> .....	33
4.3 PROTÓTIPO .....	35
4.4 MATRIZ <i>SWOT</i> .....	37
4.5 MATRIZ DE RESPONSABILIDADE.....	37
5 ANÁLISE DE REQUISITOS.....	39
5.1 DESCRIÇÃO DOS PROCESSOS E PROCEDIMENTOS .....	39
5.2 REQUISITOS FUNCIONAIS .....	39
5.3 REQUISITOS NÃO FUNCIONAIS .....	39
5.4 DIAGRAMA DE ATIVIDADES.....	40
5.4.1 DIAGRAMA DE ATIVIDADES: APLICANDO UM <i>LOGIN</i> .....	40

5.4.2 DIAGRAMA DE ATIVIDADES: CADASTRAR TÓPICO.....	42
5.5 DESCRIÇÃO DOS PROBLEMAS E PROPOSTAS DE MELHORIAS....	43
5.6 DIAGRAMA DE PACOTES PROPOSTO.....	43
5.7 DIAGRAMA DE CONTEXTO DO CASO DE USO.....	44
5.8 TABELAS DE CASO DE USO.....	45
5.9 ESTUDO DE MERCADO.....	46
5.9.1 PESQUISA DE MERCADO.....	47
5.9.2 ANÁLISE COMPARATIVA.....	47
5.9.3 CONCLUSÃO DO ESTUDO DE MERCADO.....	48
5.10 ANÁLISE DE RISCOS.....	48
5.11 CRONOGRAMA.....	50
5.12 QUADRO DE FUNCIONALIDADES.....	51
6 ANÁLISE.....	52
6.1 DESENVOLVIMENTO DO ESTUDO ECONÔMICO.....	52
6.1.1 CUSTOS E DESPESAS.....	52
6.1.2 CALCULO DO FLUXO DE CAIXA.....	54
6.2 DIAGRAMA DE CLASSE DE DOMÍNIO.....	56
6.3 DIAGRAMA DE MÁQUINA DE ESTADO.....	56
7 PROJETO.....	58
7.1 DIAGRAMA DE COMUNICAÇÃO.....	58
7.1.1 DIAGRAMA DE COMUNICAÇÃO: CADASTRANDO UM TÓPICO.....	58
7.1.2 DIAGRAMA DE COMUNICAÇÃO: CONSULTANDO UM TÓPICO.....	58
7.2 DIAGRAMA DE CLASSE DO PROJETO.....	59
7.3 DIAGRAMA DE <i>BACHMAN</i> .....	60
7.4 PROJETO DE BANCO DE DADOS.....	61
7.4.1 <i>SCRIPT</i> DE BANCO DE DADOS.....	61
7.4.2 DIMENSIONAMENTO DO BANCO DE DADOS.....	62
7.5 LISTA DE TABELAS.....	63
7.6 RESTRIÇÃO DE INTEGRIDADE.....	66
8 IMPLEMENTAÇÃO.....	68
8.1 ESTUDO DE INFRA ESTRUTURA DE TI.....	68
8.2 ANÁLISE DE INFRAESTRUTURA.....	68
8.2.1 CONFIGURAÇÃO DE HARDWARE.....	68
8.2.2 VANTAGENS.....	69
8.2.3 DESVANTAGENS.....	69
8.2.4 INFRAESTRUTURA E CRESCIMENTO.....	69
9 IMPLANTAÇÃO.....	71

<b>9.1. TESTE UNITÁRIO .....</b>	<b>71</b>
<b>9.2. TESTE INTEGRADO .....</b>	<b>71</b>
<b>10 CONCLUSÃO .....</b>	<b>72</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>	<b>73</b>
<b>ANEXOS .....</b>	<b>76</b>
<b>APÊNDICES .....</b>	<b>77</b>
<b>APENCIDE I .....</b>	<b>77</b>
<b>APENCIDE III .....</b>	<b>85</b>
<b>APENDICE II .....</b>	<b>87</b>

## LISTA DE SIGLAS

<i>AJAX</i>	<i>Asynchronous JavaScript and XML</i>	JavaScript assíncrono e XML
<i>API</i>	<i>Application Programming Interface</i>	Interface de Programação de Aplicativos
<i>CSS</i>	<i>Cascading Style Sheets</i>	Folhas de estilo em cascata
<i>EAD</i>	Educação a Distância	
<i>EAP</i>	Estrutura Analítica do Projeto	
<i>HTML</i>	<i>HyperText Markup Language</i>	Linguagem de marcação de hipertexto
<i>MVC</i>	<i>Model, view and controller</i>	Modelo, visão e controle
<i>ORM</i>	<i>Object Relational Mapping</i>	Mapeamento objeto relacional
<i>PHP</i>	<i>Hypertext Preprocessor</i>	Linguagem Interpretada Livre
<i>SQL</i>	<i>Structured Query Language</i>	Linguagem de Consulta Estruturada
<i>SWOT</i>	<i>Strengths, Weaknesses, Opportunities and Threats</i>	Forças, Fraquezas, Oportunidades e Ameaças
<i>TCC</i>	Trabalho de Conclusão de Curso	
<i>TI</i>	Tecnologia da Informação	
<i>UML</i>	<i>Unified Modeling Language</i>	Linguagem de Modelagem Unificada
<i>UNIFOA</i>	Centro Universitário de Volta Redonda	
<i>URL</i>	<i>Uniform Resource Locator</i>	Localizador Padrão de Recursos
<i>WEB</i>	<i>World Wide Web</i>	Rede mundial de computadores

## LISTA DE FIGURAS

Figura 1 – Fluxo de Montagem da Metodologia Design Thinking .....	18
Figura 2 – Metodologia Design Thinking Aplicada no TCC do UniFOA .....	20
Figura 3 – HTML5.....	21
Figura 4 – CSS3.....	22
Figura 5 – JavaScript .....	23
Figura 6 – Bootstrap Framework.....	23
Figura 7 – Responsive Web Design .....	24
Figura 8 – Ruby.....	25
Figura 9 – Padrão MVC .....	26
Figura 10 – Ruby on Rails Framework.....	27
Figura 11 – Estrutura de projeto do Ruby on Rails .....	28
Figura 12 – Banco de Dados PostgreSQL .....	30
Figura 13 – Heroku.....	31
Figura 14 – GIT .....	31
Figura 15 – GitHub .....	32
Figura 16 – Estrutura Analítica do Projeto – EAP .....	34
Figura 17 – Tela de <i>Login</i> .....	35
Figura 18 – Tela de Perguntas e Respostas.....	35
Figura 19 – Tela inicial.....	36
Figura 20 – Tela de Perfil .....	36
Figura 21 – Matriz SWOT.....	37
Figura 22 – Diagrama de Atividade Realizar Login.....	41
Figura 23 – Diagrama de Atividade Cadastrar Tópico.....	42
Figura 24 – Diagrama de Pacotes Proposto .....	44
Figura 25 – Diagrama de Contexto do Caso de Uso .....	45
Figura 26 – Gráfico de Gantt .....	50
Figura 27 – Gráfico do Fluxo de Caixa.....	55
Figura 28 – Diagrama de Classe de Domínio.....	56
Figura 29 – Diagrama de Máquina de Estado .....	57
Figura 30 – Diagrama de Comunicação – Cadastrando um tópico.....	58
Figura 31 – Diagrama de Comunicação – Consultando um tópico.....	59
Figura 32 – Diagrama de Classe do Projeto (Manter Tópicos).....	59
Figura 33 – Diagrama de Bachman.....	60
Figura 34 – Como o Heroku funciona.....	70
Figura 35 – Topologia da Infraestrutura de Rede.....	70

## LISTA DE TABELAS

Tabela 1 – Matriz de Responsabilidade.....	38
Tabela 2 – Descrição dos problemas e soluções propostas.....	43
Tabela 3 – Manter Tópico (Cadastrar Tópico).....	46
Tabela 4 – Análise Comparativa.....	48
Tabela 5 – Análise de riscos.....	49
Tabela 6 – Quadro de funcionalidades.....	51
Tabela 7 – Custos e despesas.....	53
Tabela 8 – Custo por Homem/Hora.....	53
Tabela 9 – Planilha do Fluxo de Caixa.....	54
Tabela 10 – Dimensionamento do banco de dados.....	62
Tabela 11 – Lista de tabelas.....	63
Tabela 12 – Restrição de integridade.....	66

## 1. INTRODUÇÃO

O Cenário atual verificado no nosso curso possui algumas dificuldades na realização das suas atividades e que são oriundas principalmente pela falta de interação entre professores e alunos com relação às dificuldades pela falta de oportunidade em tirar dúvidas no decorrer da aula, gerada principalmente pela metodologia ainda empregada pelo modelo educacional. Muitos deles fazem os famosos “grupos de estudo”, visando compartilhar os conhecimentos e melhorar o aproveitamento para provas. O problema é que nem sempre os alunos com maior conhecimento ou os professores estão disponíveis para resolver as dúvidas rapidamente, deixando alguns alunos por conta própria para desenvolverem suas capacidades ou não. Por conta disso, alguns alunos simplesmente deixam de estudar o que era necessário, deixando sempre para o próximo semestre, ou para dependência e prova final.

Sabe-se da importância dos professores, dos alunos, enfim, de todos que se disponibilizam em ajudar uns aos outros, promovendo assim, uma motivação que venha contribuir na aquisição de conhecimentos e elevarem suas oportunidades no mercado de trabalho.

O *INTERNSHIPIN* propõe a solução para resolver este problema, ou seja, gerar uma plataforma que reúne todos os alunos e professores do mesmo curso de uma única universidade, compartilhando entre si conhecimentos, dúvidas, exercícios e projetos supervisionados pelo professor responsável.

As empresas conveniadas a universidade terão acesso aos projetos realizados pelos alunos específicos que se candidatarão a vaga, caso possua, e que tenham cumprido os requisitos necessários. Os projetos serão uma espécie de portfólio para que os estudantes exponham suas ideias e seus conhecimentos, contendo avaliações dadas pelo supervisor do projeto, e com isso adquirem experiências profissionais realizadas em equipes simulando o mercado de trabalho.

Os alunos também poderão criar chamadas de dúvida para que outros alunos do mesmo curso, independentemente do período possam ajuda-los, contando também com os professores que ministram matérias relacionadas para que após um prazo possam também entrar na discussão e fixar o conhecimento.

Afinal, quando você vê, discute, ensina e pratica, você agrega mais conhecimento para sua carreira.

### **1.1 OBJETIVO GERAL**

O objetivo principal é preparar o discente para a realidade do mercado de trabalho e facilitar seu ingresso no mesmo ao centralizar oportunidades de adquirir conhecimento teórico e experiência prática mesmo sem ter trabalhado na área.

### **1.2 OBJETIVOS ESPECÍFICOS**

Para os objetivos específicos do desenvolvimento desta aplicação, destacam-se:

- Desenvolver um aplicativo que facilite a interação do aluno com o mercado de trabalho.
- Reunir conhecimento e experiência para ingressar no mercado de trabalho;
- Incentivar o trabalho em equipe;
- Aprofundar o conhecimento dos alunos por meio de discussões ativas e resoluções de exercício ou questões inacabadas;
- Oferecer a oportunidade de participar de projetos que simulam o ambiente do mercado de trabalho.

### **1.3 MOTIVAÇÃO PARA DESENVOLVIMENTO DO PROJETO**

O desenvolvimento desta proposta de trabalho foi motivada para facilitar a consolidação do conhecimento adquirido em sala de aula e aplicar na realidade exigida no mercado de trabalho.

## 1.4 ORGANIZAÇÃO DO PROJETO

Neste projeto utilizaremos a metodologia estabelecida por Siqueira *et al.* (2017) para o Curso de Sistemas de Informação. Este projeto possui várias fases para seu desenvolvimento. A primeira delas é constituída pela construção do Referencial Teórico, com o objetivo de expor inicialmente a metodologia ativa empregada, onde destaca-se a aplicabilidade da Metodologia *Design Thinking*, apresentando como esta poderá ser aplicada no desenvolvimento do projeto, inclusive como o processo utilizado para construir uma ideia inovadora e transformá-la em realidade.

A próxima etapa é o Gerenciamento do Projeto onde busca-se realizar o planejamento do projeto, constando as técnicas aplicadas durante a execução das atividades referentes a um projeto, conhecimentos, ferramentas e habilidades, com o objetivo de atingir de maneira eficiente o escopo proposto do projeto. Nesta fase também serão construídos o Escopo do Projeto, uma Estrutura Analítica do Projeto (*EAP*), Análise de Risco, Matriz *SWOT* e o Cronograma do Projeto.

A fase de Análise de Requisitos, em seguida, é a etapa para compreensão e descrição dos processos visando construir o software. Também é nesta etapa onde serão criadas as representações gráficas, na forma de diagramas, para o que foi proposto, além de descrever o sistema e seus processos. Para isso serão criados diagramas diversos, tais como Diagrama de Pacotes, Diagramas e Tabelas de Casos de Uso, Diagrama de Contexto dos Casos de Uso. Além disso, também será realizado um Estudo de Gestão do Projeto e um Estudo do Mercado.

Há também a fase de Análise, que se define por ser onde serão examinados os requisitos levantados pelos usuários, realizando também a verificação dos mesmos e a definição das reais necessidades do sistema de informação. Aqui serão utilizadas as informações levantadas e coletadas durante a fase de Análise de Requisitos, que permitem a criação do Estudo Econômico do Projeto, do Diagrama de Máquina de Estado e o Diagrama de Classe de Domínio.

Após as fases de Análise, a próxima fase a ser processada é a de Projeto, sendo a fase que procura estabelecer “como” o sistema fará o que foi

determinado na fase de análise, ou seja, qual será a solução para o problema identificado, nesta fase é onde será feita a maior parte da modelagem do software a ser desenvolvido, ou seja, onde será produzida a arquitetura do sistema em que serão construídos os diagramas representando os módulos proposto, sendo por fim criado o Projeto de Banco de Dados.

A fase seguinte é composta pelo Desenvolvimento de diversos itens como o Plano de Testes, Estudo de Infraestrutura de *TI*, Proposta de Segurança, de Confiabilidade e de Implantação, *Layout* das Telas e Relatórios, e por fim o Estudo de Segurança, é chamado de fase de Implementação, onde ocorre a codificação e integração dos módulos para obter-se o produto final.

Na fase final ocorre a implantação do produto de *software* desenvolvido através das fases anteriores, baseado na Metodologia *UniFOA*, ocorre a etapa de implementação liberando o *software* para uso, sendo as principais atividades desta fase os Testes de Aceitação e Marketing do Produto.

## **1.5 RESULTADOS ESPERADOS**

Com a criação do *INTERNSHIPIN* espera-se criar uma plataforma que seja capaz de contribuir e qualificar o discente ou um usuário qualquer para enfrentar o mercado de trabalho através da consolidação do conhecimento teórico inter-relacionado com a experiência prática, dadas as participações em projetos que simulam o ambiente do mercado de trabalho.

## 2 METODOLOGIA

A metodologia aplicada no desenvolvimento do *INTERNSHIPIN* seguirá as orientações de diversos dos modelos elaborados para o desenvolvimento de trabalhos acadêmicos estabelecidos como norma na instituição de ensino *UniFOA*, iniciando com uma metodologia ativa, na qual utilizou-se a *Design Thinking* cuja finalidade está em uma forma de pensar para dar vida a ideias, onde o aprendiz é o centro do processo do desenvolvimento intelectual.

Para estruturação do projeto adota-se na instituição *UniFOA* o seu manual de instrução que pode ser visto em Castro *et al.* (2009), material indispensável para formatação do *TCC* e de outras produções acadêmicas.

Em conformidade com Siqueira *et al.* (2017) o projeto em questão será desenvolvido de acordo com os materiais apresentados nos manuais de orientação para a conclusão do Projeto Integrado, esta metodologia se assemelha com a metodologia *design thinking* e tem como objetivo gerar versões intermediárias do *software* durante o ciclo de vida do desenvolvimento, assim minimizando os riscos de erros e levar os projetos dentro de um padrão normalizado para o desenvolvimento do sistema.

O desenvolvimento do sistema irá seguir padrões de construção *Web*, usando as linguagens deste meio. Para o desenvolvimento *Front-End* do projeto, usaremos a linguagem *HTML5* para criar a estrutura corporal do sistema, a linguagem *CSS3* para dar estilo (características) as estruturas *HTML5*, *JavaScript* pois será usada para a criação de eventos (funções e/ou procedimentos), a biblioteca *Jquery* usada para criar uma dinâmica melhor com o usuário, *Ajax* para criar “carregamentos” de estruturas sem a necessidade de atualização de páginas e buscar requisições por parte do servidor. Para o desenvolvimento *Back-End*, será utilizado a linguagem *Ruby*. Utilizaremos alguns *frameworks* para agilizar o desenvolvimento como *Rails* para complementar o *Ruby*; *Bootstrap* e *Material Design* para o *Front-End*. O Banco de Dados a ser implementado será do *MySQL*. A aplicação será rodada em um servidor de produção e outro para homologação na *Digital Ocean*.

### Referencial Teórico

- Metodologia *Design Thinking*;

- Educação e *T.I.*;
- *HTML5*;
- *CSS3*;
- *JavaScript e JQuery*;
- *Bootstrap*;
- *Responsive Web Design*;
- *Ruby*;
- *MVC*;
- *Ruby On Rails*;
- *PostgreSQL*;
- *HEROKU*;
- *GIT*;
- *GITHUB*.

### **Gerenciamento de Projeto**

- Levantamento do Escopo;
- Estrutura Analítica do Projeto – *EAP*;
- Dicionário *EAP*;
- Quadro de Funcionalidades;
- Matriz de Responsabilidades;
- Matriz *SWOT*;
- Cronograma de Projeto;
- Orçamento;
- Stakeholders;
- Análise de Risco;
- Relatórios de Desempenho.

### **Análise de Requisitos**

- Descrição do Sistema;
- Descrição dos Processos e Procedimentos;
  - Requisitos Funcionais;
  - Requisitos não Funcionais;
- Desenvolvimento do Diagrama de Atividade;
- Desenvolvimento do Diagrama de Pacotes;

- Desenvolvimento do Diagrama de Contexto;

### **Análise**

- Desenvolvimento do Estudo Econômico;
- Diagrama de Classe de Domínio;
- Diagrama de Máquina de Estado.

### **Projeto**

- Desenvolvimento do Diagrama de Classe de Projeto;
- Desenvolvimento do Diagrama de *Bachman*;
- Desenvolvimento do Diagrama de Comunicação;
- Desenvolvimento do Projeto de Banco de Dados.

### **Implementação**

- Desenvolvimento do Estudo de Segurança;
- Desenvolvimento da Programação;
- Elaboração dos *Layout* das Telas e Relatórios;
- Desenvolvimento do Plano de Teste.

### **Implantação**

- Conclusão do Projeto;
- Testes Integrados;
- Testes Unitários.

A Orientação do Projeto Integrado (*TCC*) será de acordo com o modelo de orientação criado no Curso de Sistemas de Informação, formado por professores especialistas em diversas áreas de *TI*, os quais pertencem ao Comitê de Orientação, cada um deles possuem atribuições para orientação em sua área, em detalhes sobre tais atribuições podem ser vistas no Anexo I.

Os encontros efetuados com estes professores do Comitê são registrados em Atas de Reuniões, que são produzidas pelos alunos e depois são entregues ao Coordenador de *TCC*, algumas destas podem ser vistas no Apêndice I.

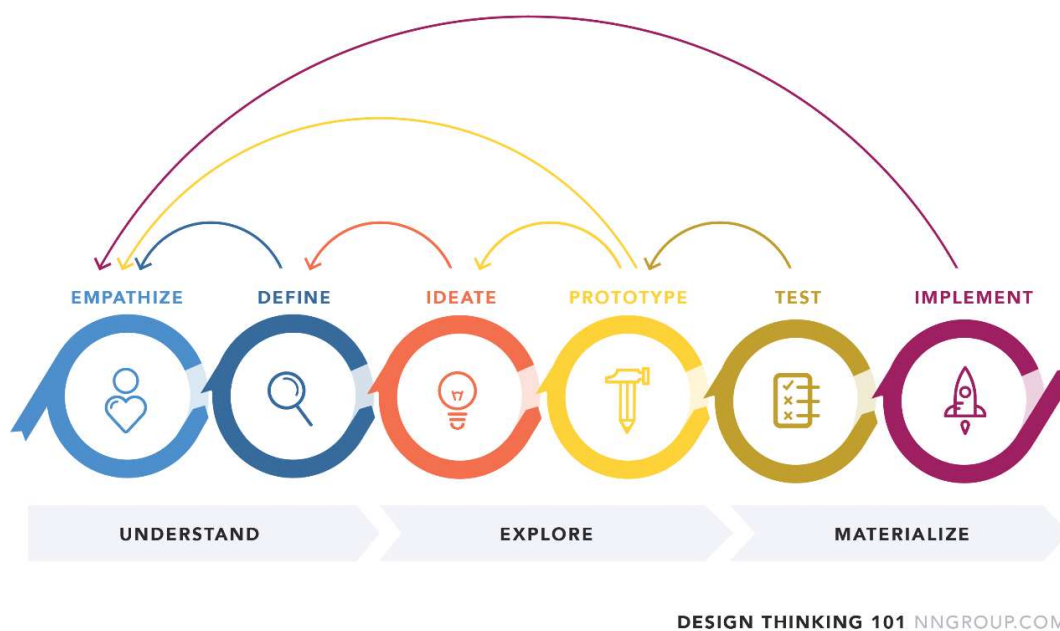
### 3 REFERENCIAL TEÓRICO

#### 3.1 METODOLOGIA ATIVA – *DESIGN THINKING*

Segundo Brown (2010) *Design Thinking* representa o próximo passo: colocar ferramentas nas mãos de pessoas que possivelmente não tenham pensado em si mesmas como “*designers*” e direcioná-las a uma diversidade muito mais ampla de problemas. O autor anterior, ainda cita que a inovação não se limita mais aos lançamentos de novos produtos físicos, mas abrange os novos tipos de processos. São essas as atividades centradas no ser humano que os *designers* trabalham todos os dias. De acordo com o mesmo autor, existe uma evolução que reflete o crescente reconhecimento por parte dos líderes de negócio de que a importância e contribuição do *design* é muito ampla para ser deixado de lado exclusivamente para os *designers*: fazer *design* para pensar *design*.

A Figura 01 representa as etapas de um processo com metodologia *Design Thinking*, que pode ser dividida em três etapas: Compreender (*Understand*), Explorar (*Explore*) e Materializar (*Materialize*).

Figura 1 – Fluxo de Montagem da Metodologia Design Thinking



Fonte: GIBBONS (2016)

Segundo Gibbons (2016) dentro de cada uma das três etapas já citadas, existem mais duas, totalizando seis passos para implantar com sucesso a metodologia *Design Thinking*. O mesmo autor dissemina as etapas minuciosamente:

- Empatia (*Empathize*): desenvolver conhecimento sobre o que seus usuários alvo fazem, dizem, pensam e sentem através de pesquisas que devem ser realizadas. Observar diretamente o que eles fazem, como pensam, e o que querem. O objetivo desta fase é começar a se empatizar com os usuários e suas perspectivas.
- Definir (*Define*): juntar as pesquisas realizadas e observar onde os problemas do usuário existem. Identificando esses problemas, use os dados da fase Empatia para obter informações. Existe uma interseção entre as dificuldades e problemas dos usuários observados?
- Idealizar (*Idealize*): um *brainstorm* é bem-vindo para criar soluções criativas que abordam as necessidades dos utilizadores não satisfeitos que foram identificados na fase Definir.
- Protótipo (*Prototype*): construa representações reais para um subconjunto de ideias. O objetivo é entender quais são as engrenagens das suas ideias e se funcionam ou não.  
Torne suas ideias reais. Pegue o *feedback* e reconstrua o protótipo com base no próprio retorno em código sujo e rápido. Divulgue para outros usuários.
- Teste (*Test*): após divulgar o protótipo, pegue o comentário deles para analisar: Essa solução atende as necessidades dos usuários? Eles melhoraram como se sentem, pensam ou fazem suas tarefas? Continue sempre testando para checar se o produto corresponde as necessidades dos clientes reais.
- Implementar (*Implement*): colocar em prática toda a visão adquirida sobre as fases anteriores. Certifique-se de que a solução é materializada (*materialized*) e facilita a vida dos seus usuários finais.

Segundo Brown (2010) a continuidade da inovação pode ser visto mais como espaços que se sobrepõem do que como uma série de etapas que devem ser cumpridas. Podemos ter em mente que tais espaços são como o problema, a oportunidade ou a inspiração que encoraja a busca por soluções; a idealização, desenvolver e testar ideias; e implementar, o caminho que é trilhado do papel ao mercado.

Figura 2 – Metodologia *Design Thinking* Aplicada no TCC do *UniFOA*



De acordo com Brown (2010) a metodologia *Design Thinking* se divide em três etapas. Entre elas estão: Compreender, Explorar e Materializar.

A metodologia *UniFOA* se divide em seis etapas. Entre elas estão: Gerenciamento, Análise de Requisitos, Análise, Projeto, Implementação e Implantação.

A relação entre elas estão em: Compreender que se relacionam com as etapas de Gerenciamento e Análise de Requisitos.

Explorar que se relacionam com as etapas de Análise e Projeto.

E por fim, Materializar que se relaciona com as etapas de Implementação e Implantação.

### 3.2 EDUCAÇÃO E T.I.

De acordo com Sé (2013) a tecnologia desperta interesses, curiosidades e permite um vasto campo de possibilidades infinitas, inclusive na resolução de conflitos, nacionais e internacionais.

Segundo o autor anterior, o estudo, no teor cru da palavra, está ultrapassado nas salas de aula, por exemplo, utilizando-nos apenas de quadros,

giz, apagadores, canetas, presença de professores. É preciso haver desenvolvimento de ideias para o aprendizado, este método é obsoleto e ultrapassado. Hoje, não é mais justificada esta metodologia de ensino, apenas. Toda escola, curso, universidade deveria acrescentar a modalidade *EAD*, além deste conceito tradicional.

### 3.3 HTML5

Segundo Mazza (2012) se você intenciona ou trabalha com tecnologia, boa parte dos seus projetos será utilizada através de um navegador, seja uma rede social, controle bancário, portal de notícias ou sites de campanha. Ou seja, o “habitat natural” do usuário é a *web*. Ter conhecimento sobre o *front-end* é essencial para qualquer programador.

O mesmo autor cita que todo o *HTML* que desenvolvemos ganha vida nos navegadores dos usuários.

Segundo Eis (2011) *HTML* é a linguagem básica da internet. Foi construída para ser entendida facilmente por seres humanos e máquinas.

Figura 3 – HTML5



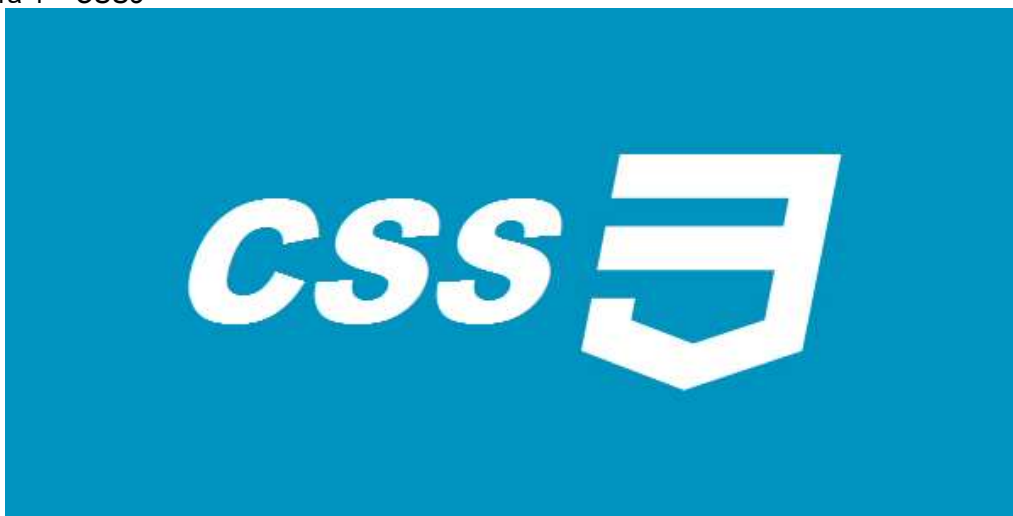
Fonte: SAWYER (2015)

### 3.4 CSS3

Segundo Pessoa (2016) CSS significa *Cascading Style Sheet*, em português: Folha de Estilo em Cascata. A definição mais simples segundo seu livro está no site da W3C: “Folha de estilo em cascata é uma linguagem para descrever a apresentação de páginas da *web* (como *backgrounds*, cores, espaçamento, fontes e tamanhos) a linguagem da *web*”.

De acordo com Pessoa (2016), o CSS tem como finalidade ajudar a estruturar e melhorar a visualização dos elementos *HTML* para o usuário.

Figura 4 – CSS3



Fonte: SAWYER (2015)

### 3.5 JAVASCRIPT E JQUERY

Segundo Bertoni (2015) *JavaScript* é uma linguagem com possibilidades de incrementar o *HTML* com elementos interativos. Sua página se torna capaz de responder a diversos eventos que são iniciados pelo usuário. Com o *JavaScript* você ainda é capaz de manter dados persistentes utilizando objetos, arquivos e banco de dado relacional.

De acordo com Balduino (2012) *jQuery* é uma biblioteca *JavaScript*, que manipula com mais facilidade os documentos *HTML*, eventos, interações e animações com *AJAX* para maior rapidez no desenvolvimento *web*.

Figura 5 – *JavaScript*



Fonte: SILVEIRA (2013)

### 3.6 **BOOTSTRAP**

Segundo Matos (2015) *Bootstrap* é uma solução desenvolvida em 2010 por dois desenvolvedores do *Twitter*. O propósito era criar uma solução para otimizar a produção e desenvolvimento de layouts responsivos. Antes de tomar grandes proporções e ser utilizado por grande parte dos desenvolvedores *web*, o *Bootstrap* era utilizado internamente no *Twitter*.

Figura 6 – *Bootstrap Framework*



Fonte: MARTÍNEZ (2013)

De acordo com Matos (2015) apesar de o *Bootstrap* agilizar muito o desenvolvimento do *layout*, existem coisas que ele não é capaz de fazer, como:

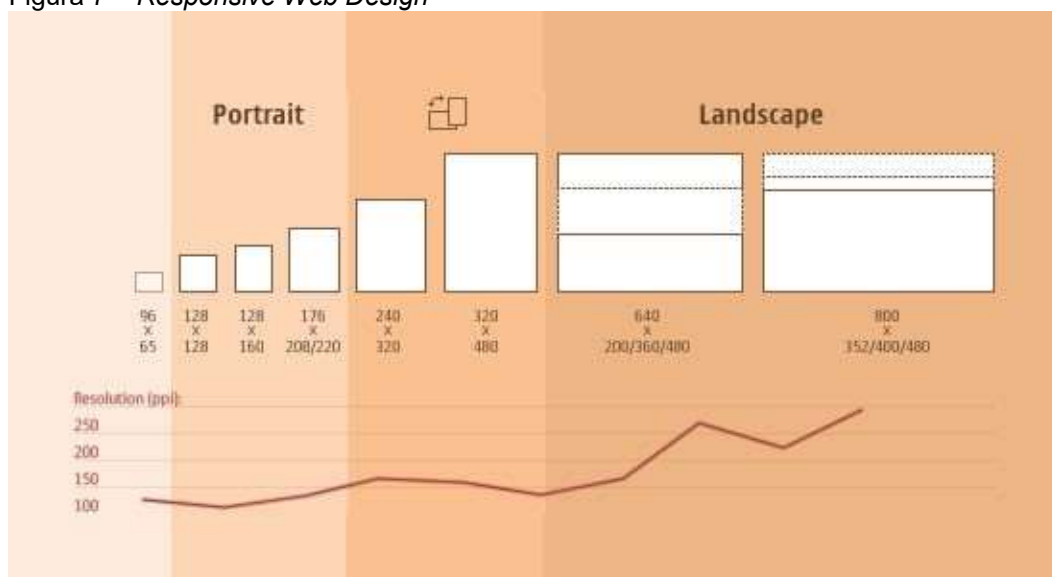
- Clonar um *layout*;
- Programar todos os elementos necessários para desenvolver um site ou sistema;

- Eliminar a necessidade de programar.

### 3.7 RESPONSIVE WEB DESIGN

Segundo Teixeira (2011) devido ao crescimento da variedade de dispositivos com acesso a sites (*tablets, smartphones, TVs, monitores diversos, etc*), seria inviável desenvolver diversas versões diferentes de um mesmo site ou sistema para cada tamanho de tela e resolução disponível no mercado. O *Responsive Web Design* é uma técnica para solucionar esse tipo de problema: desenvolver um site ou sistema de maneira que os elementos se adaptem e se ajustem de maneira automática de acordo com o tamanho da tela e resolução do dispositivo que está acessando a página.

Figura 7 – *Responsive Web Design*



Fonte: TEIXEIRA (2011)

De acordo com Teixeira (2011) um *design responsivo* abrange:

- Adaptar o *layout* de acordo com a resolução do dispositivo que está visualizando-a.
- Ajustar e redimensionar as imagens automaticamente para que caibam na tela.
- Ocultar elementos que são dispensáveis em dispositivos móveis e menores.

- Ajustar botões e *links* para *touch*, já que o raio de alcance do dedo é maior que o do ponteiro do mouse.
- Simplificar elementos para facilitar a navegação do usuário.

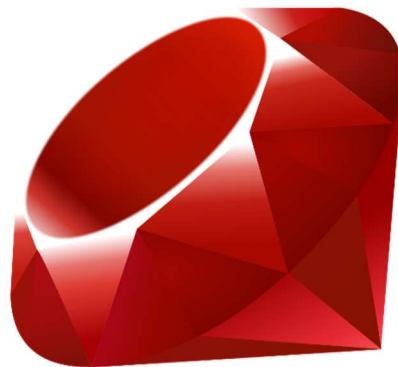
### 3.8 RUBY

De acordo com Brando (2010) Yukihiro Matsumoto, conhecido pela *web* como “*Matz*”, o criador do *Ruby*, que é uma linguagem orientada a objetos adequada para aplicativos de grande porte e para pequenos desenvolvimentos do dia-a-dia. O propósito do *Ruby* é ser uma linguagem mais produtiva e mais divertida de utilizar.

Segundo Cussó (2016) o *Ruby* é hoje em dia uma das linguagens de programação preferidas e que os desenvolvedores mais tem aceitação em utilizar, já que trabalha de maneira com a qual é possível de certa forma “antecipar” como irá se comportar.

De acordo com Cussó (2016) o *Ruby* também é a linguagem *back-end* mais utilizada no momento para desenvolvimento de startups. Alguns exemplos que utilizam *Ruby* são o *Twitter* e o *Basecamp*.

Figura 8 – *Ruby*



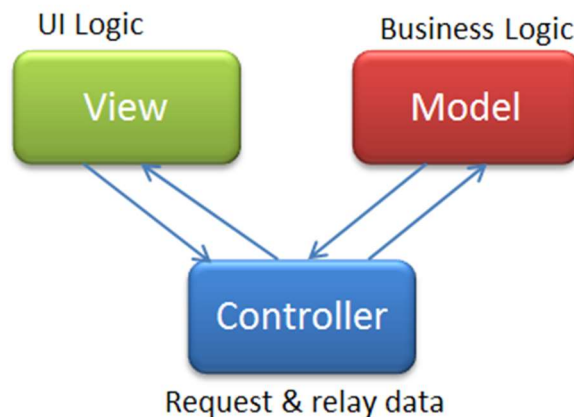
Fonte: ELIAS (2013)

### 3.9 MVC

Segundo Belem (2013) *MVC* é um modelo de arquitetura de *software*, cujo objetivo é separar as regras de negócio da interface que o usuário interage.

De acordo com Belem (2013), sua estruturação se baseia em dividir a interface de interação (*Views*), do controle de informação, que possui regras de negócio (*Models*), partindo de uma controladora (*Controllers*) que faz o “meio-de-campo” entre as duas primeiras camadas.

Figura 9 – Padrão MVC



Fonte: CHOWDHURRY (2013)

Belem (2013) faz uma breve introdução para o entendimento de cada camada:

- *Model* é a camada que representa as informações dos seus dados, oferecendo meios de leitura e escrita para os mesmos. Ou seja: o que diz respeito a validações, escritas e leituras dos dados pertence a camada *Model*.
- A camada *View* é onde o sistema interage com o usuário. Toda a parte de *front-end* deve ser exibido nessa camada. Ela é responsável apenas por exibir os dados.
- Já o *Controller* possui *public methods* que são chamados de *actions*, que são responsáveis por uma “página” do seu sistema. O *controller* é responsável por decidir qual *model* irá utilizar; o que pedir para o *model*; quais *views* serão utilizadas para exibição do retorno de dados da camada *Model*.

### 3.10 RUBY ON RAILS

Segundo Faria (2009) *Ruby on Rails* é um framework MVC que utiliza alguns recursos como *DRY*, ou seja, não é necessário repetir o código por todo o aplicativo. O *Ruby on Rails* utiliza alguns recursos como os métodos *helpers* e as *partials* para melhorar a leitura e diminuir a repetição de código pelo sistema.

Figura 10 – *Ruby on Rails Framework*



Fonte: MARTINS (2015)

De acordo com Faria (2009), um *helper* é um módulo que contém funções auxiliares para a camada *View*, eliminando lógicas complexas do código. Existem vários *helpers* úteis, como *helpers* que formatam moedas, datas, formulários em *HTML*, *Javascript*, entre outros.

O autor acima deixa claro que o *Ruby on Rails* possui suporte para os bancos de dados mais utilizados no mercado, como *SQLServer*, *Oracle*, *PostgreSQL*, *MySQL*, *SQLite*, e para indicar qual será utilizado, basta especificar os dados de acesso em um simples arquivo de conexão.

Segundo Faria (2009) existe outra característica interessante no *Rails*, que é o perfil de execução do aplicativo, podendo ter perfis personalizados para o ambiente de desenvolvimento, homologação e produção. É possível também você utilizar bancos de dados diferentes entre um perfil e outro, já que o *framework* possui a habilidade de sincronizar todos os dados entre os bancos de dados já citados.

O autor anterior ainda esclarece que *Ruby on Rails* possui 5 módulos independentes:

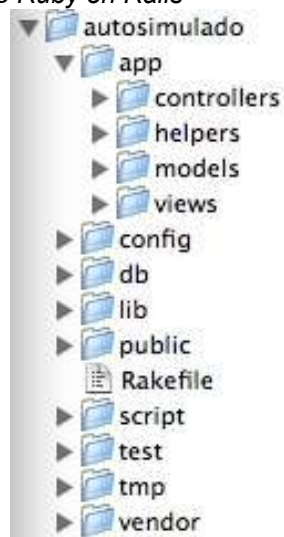
- Módulo *Active Record*: conecta os objetos de negócio com as *tables* do banco de dados criando um modelo onde os dados e a

lógica se apresentam em conjunto. É um *ORM* baseado em convenções.

- Módulo *Action Pack*: abrange o *Action Controller* e o *Action View*. O *Action Controller* é responsável pela interação entre o *model* e a *view*. Cuida também da parte de roteamento de *URLs* para as ações dos *controllers*, fazendo o gerenciamento personalizado com *URLs* de fácil leitura. Já o *Action View* compõe toda a funcionalidade fundamental para renderizar a *view*.
- Módulo *Action Support*: é um aglomerado de classes úteis e extensões de biblioteca padrão.
- Módulo *Action Mailer*: é um ótimo *framework* para *services* de entrega e recebimento de e-mails.
- Módulo *Action WebServices*: é responsável por prever uma forma de produzir *APIs* inter-operáveis com *Rails*. A partir da versão 2.0 este módulo foi eliminado, devido a implementação do módulo *Rest* no *Rails*.

Segundo Faria (2009) a estrutura de um *Rails Project* é feita na criação de diretórios:

Figura 11 – Estrutura de projeto do *Ruby on Rails*



Fonte: FARIA (2009)

- *App*: possui subpastas que armazenam *controllers*, *helpers*, *views* e *models*.

- *Config*: possui diversos arquivos de configuração, inclusive o arquivo que determina qual será o Banco de Dados utilizado.
- *Db*: é responsável pelas *Migrations*, que são um recurso onde o desenvolvedor escreve arquivos *Ruby* para manipulação do banco de dados. Através das *Migrations* é possível criar tabelas, colunas, realizar triggers sem a necessidade de escrever em *SQL*.
- *Lib*: se destina em receber as bibliotecas externas.
- *Log*: como o nome já diz, os logs gerados em tempo de execução são salvos nesta pasta.
- *Public*: armazena as imagens, *stylesheets* e os arquivos *javascript*.
- *Script*: tem os *scripts* utilizados para ajudar no desenvolvimento. Com tais *scripts*, o desenvolvedor pode gerar novos *controllers* e *models* através da linha de comando. Também possui um programa *server*, um servidor *web* para desenvolvimento.
- *Test*: armazena arquivos para realização de testes funcionais, integrados e unitários.
- *Vendor*: armazena os *plugins* do projeto, e o próprio *framework*, dependendo do caso.

### 3.11 POSTGRESQL

Segundo Milani (2008) o *PostgreSQL* é um Sistema Gerenciador de Banco de Dados Relacional que é utilizado para guardar informações e dados de soluções em informática em todos os seguimentos de negócio, assim como administrar tais informações.

Figura 12 – Banco de Dados PostgreSQL



### 3.12 HEROKU

Segundo Poojary (2015) Heroku é um serviço em nuvem *pay-as-you-go* que oferece assim como a *Amazon Web Service*, que empresas mudem todas as suas operações sem os data centers tradicionais. É uma plataforma com ambiente dinâmico que permite que os desenvolvedores simplesmente “joguem” seu código e deixe que a plataforma lide com o resto. Você perde um pouco de controle, mas ganha em simplicidade e em velocidade.

De acordo com Poojary (2015) o Heroku utiliza o controle de versão *Git* para gerenciar os *deploys* da sua aplicação. Tudo que você precisa fazer para dar *deploy* é puxar o repositório *Git* para os servidores do Heroku.

O autor acima ainda diz que o Heroku possui algumas características, como:

Rodar a aplicação através de um número pré-selecionado de *virtual servers*:

- Gerenciar *releases* da sua aplicação para diferentes ambientes;
- Garantia de que sua aplicação será recuperada automaticamente de falhas nos servidores;
- Controla o balanceamento de acessos para que você possa dimensionar sua aplicação para suportar milhões de usuários;
- Permite que você rapidamente possa adicionar ou remover partes da infraestrutura, como *caching servers* e banco de dados.

Segundo Poojary (2015), o Heroku tem suporte para as seguintes linguagens de programação: *Ruby, Node.js, Python, Java, Go, PHP e Scala*.

Figura 13 – Heroku



Fonte: LOGVINENKO (2015)

### 3.13 GIT

Segundo Dacal (2015) *Git* é um controlador de versão que foi criado por Linus Torvalds para desenvolver o *Kernel* do *Linux*. Com o *Git*, é possível que você possa salvar cada alteração do seu projeto e futuramente consultar ou reverter para uma determinada alteração. Funciona como um ponto de restauração do *Windows*.

Figura 14 – GIT



Fonte: LENNON (2017)

### 3.14 GITHUB

Segundo Dacal (2015) *GitHub* é um sistema online que funciona em cima do *Git*. É possível criar projetos e mantê-los online, mas ele também é utilizado como uma rede social, onde os desenvolvedores podem ver projetos de outros desenvolvedores e contribuir com eles. É utilizado também como portfolio para os desenvolvedores, já que olhar para o código que você escreve é a melhor forma de conhecer seu nível de trabalho.

Figura 15 – *GitHub*



Fonte: GENTLE (2016)

## **4 GERENCIAMENTO DO PROJETO**

Segundo registrado em PMBOK (2013) a gerência de projetos é a aplicação de conhecimentos, habilidades, ferramentas e técnicas para projetar atividades que visem satisfazer ou exceder as necessidades e expectativas das partes envolvidas, com relação ao projeto.

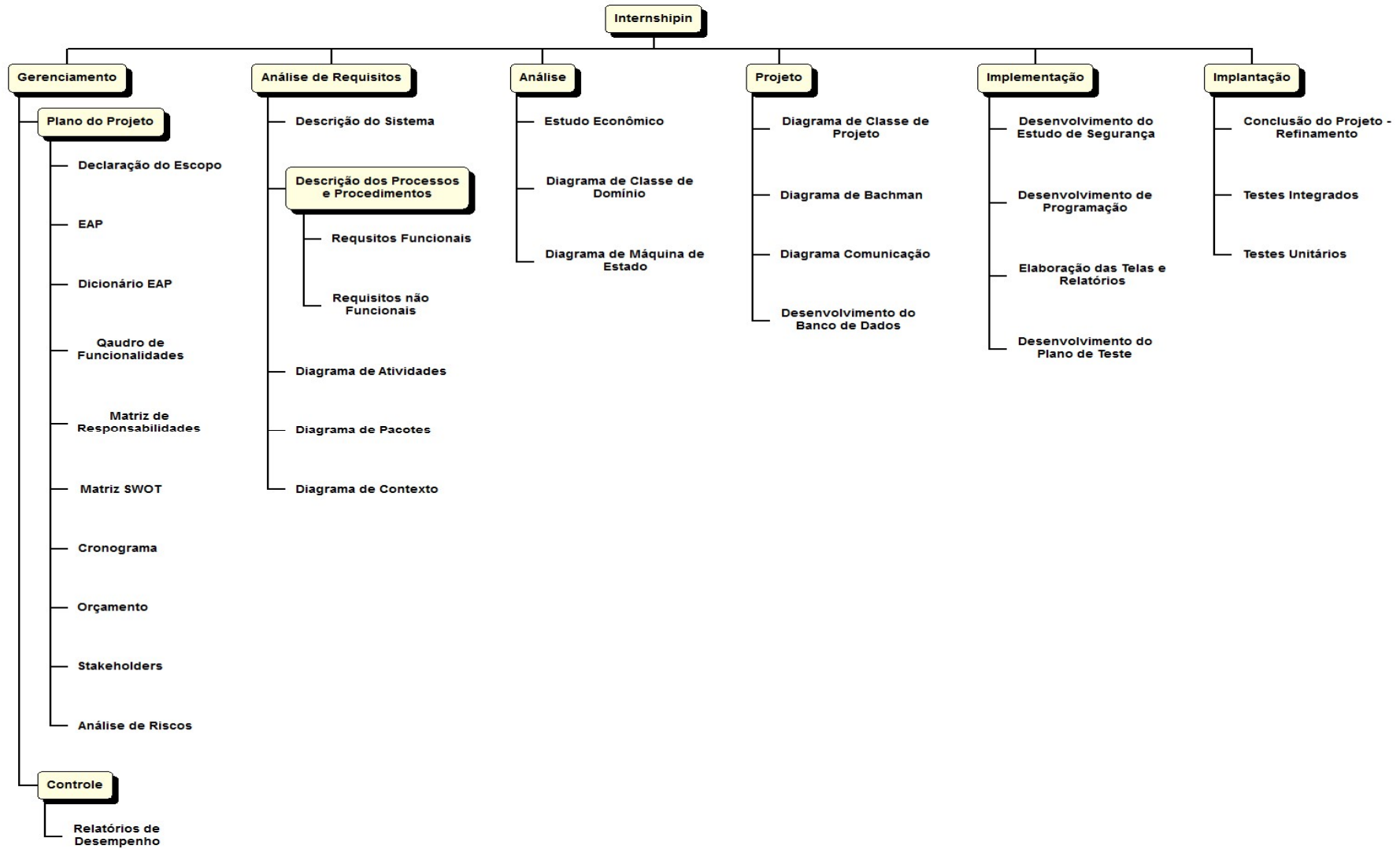
### **4.1 LEVANTAMENTO DO ESCOPO**

A expectativa é que o sistema desempenhe bem sua função de melhorar a preparação do aluno para a realidade do mercado de trabalho, através do incentivo ao compartilhamento de conhecimento entre os usuários (alunos e professores) sendo uma aplicação que será distribuída para a plataforma *Web*.

### **4.2 ESTRUTURA ANALÍTICA DO PROJETO – EAP**

Segundo Sabino (2015) a *EAP* é uma ferramenta visual que auxilia na definição do escopo, desmembrando as fases do projeto em partes manejáveis, permitindo ao gerente do projeto e as partes interessadas terem uma visão organizada, estruturada e clara de todas as entregas do projeto.

Figura 16 – Estrutura Analítica do Projeto – EAP



### 4.3 PROTÓTIPO

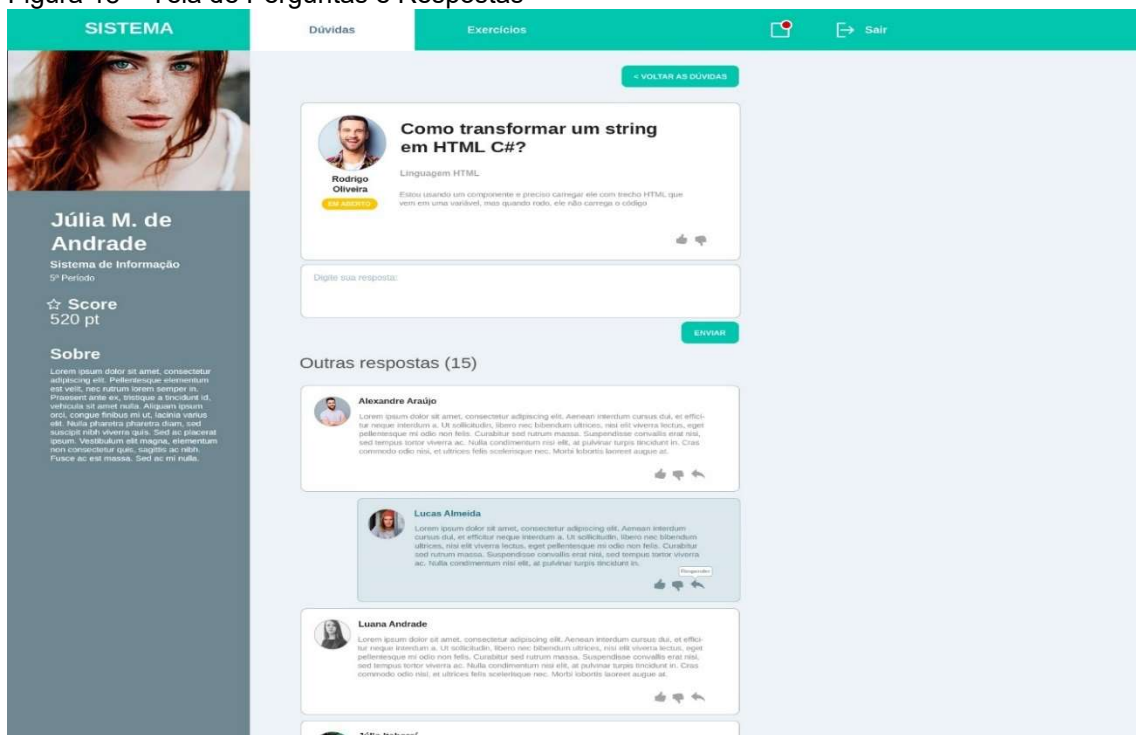
A tela a seguir apresenta os campos para o usuário logar com sua Matrícula e Senha.

Figura 17 – Tela de Login



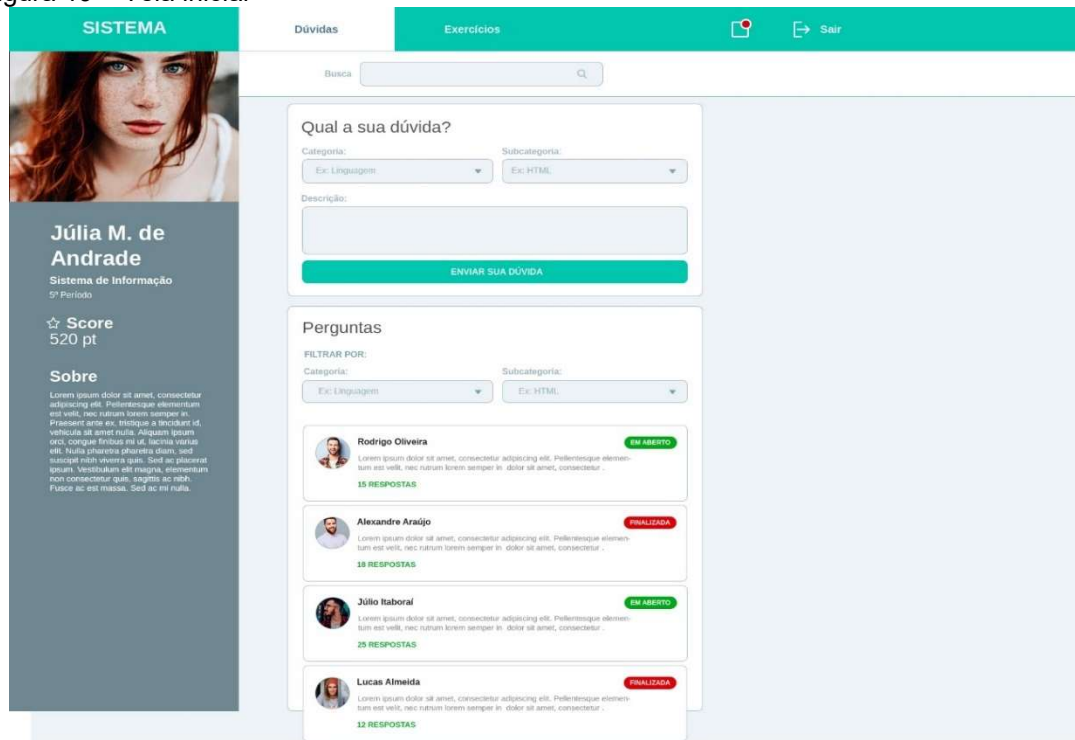
A tela a seguir mostra a tela de pergunta e respostas.

Figura 18 – Tela de Perguntas e Respostas



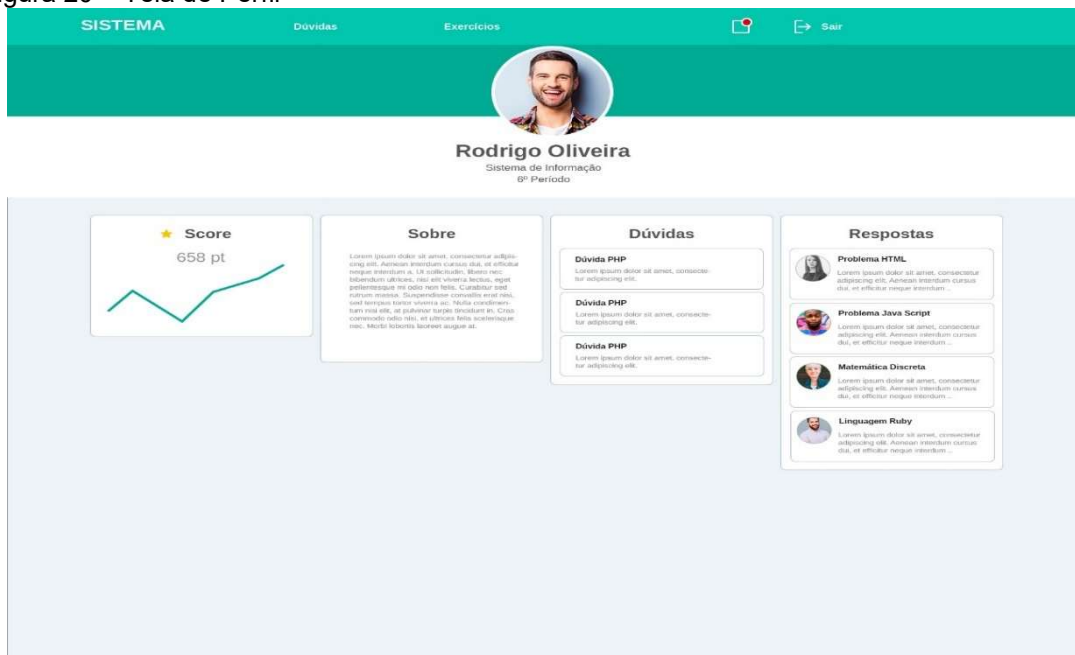
A imagem abaixo mostra a tela inicial da plataforma.

Figura 19 – Tela inicial



A tela a seguir mostra a página de Perfil, onde será possível visualizar informações sobre a atuação do usuário no *Internshipin*:

Figura 20 – Tela de Perfil



#### 4.4 MATRIZ SWOT

A Matriz SWOT apresentada na figura 21 a seguir, registra os pontos positivos e negativos tanto externos quanto os internos com relação ao desenvolvimento do projeto, servindo para que a equipe tenha uma ideia do que precisa ser aperfeiçoado.

Figura 21 – Matriz SWOT

	POSITIVO	NEGATIVO
INTERNOS	<p>Integrante da equipe já trabalha com a tecnologia utilizada no projeto. Equipe unida Equipe com disponibilidade de horário para o projeto</p> <p><b>Forças (S)</b></p>	<p>Pouco conhecimento da linguagem de programação utilizada Pouco conhecimento na parte de Educação/TI</p> <p><b>Fraquezas (W)</b></p>
EXTERNOS	<p>Pouca concorrência Grande interesse dos alunos da área de exatas Estimular o trabalho em equipe entre os alunos Aquisição de conhecimento Adaptar para cursos de humanas ou biológicas</p> <p><b>Oportunidades (O)</b></p>	<p>Mercado de atuação restrito Disponibilidade dos docentes</p> <p><b>Ameaças (T)</b></p>

#### 4.5 MATRIZ DE RESPONSABILIDADE

A seguir é apresentada a Tabela 1, cuja finalidade é em demonstrar as atribuições que cada profissional possui no desenvolvimento do sistema;

Tabela 1 – Matriz de Responsabilidade

<b>Matriz de Responsabilidades</b>	<b>Gerente de Projeto</b>	<b>Analista de Requisitos</b>	<b>Analista de Banco de Dados</b>	<b>Programador</b>	<b>Analista de Teste</b>	<b>Designer</b>
Declaração do Escopo	AP/R	E				
EAP	AP/R	E				
Dicionário EAP	AP/R	E				
Quadro de Funcionalidades	AP/R	E				
Matriz de Responsabilidades	AP/R	E				
Matriz SWOT	AP/R	E				
Cronograma	AP/R	E				
Orçamento	AP/R	E				
Stakeholders	AP/R	E				
Análise de Riscos	AP/R	E				
Relatório de Desempenho	AP/R	E				
Descrição do Sistema	AP/R	E				
Requisitos Funcionais	AP/R	E				
Requisitos não funcionais	AP/R	E				
Diagrama de Atividades	AP/R		E			
Diagrama de Pacotes	AP/R		E			
Diagrama de Contexto	AP/R		E			
Diagrama de Classe de Domínio	AP/R		E			
Diagrama de Máquina de Estado	AP/R		E			
Estudo Econômico	AP/R	E				
Diagrama de Classe de Projeto	AP/R		E			
Diagrama de Bachman	AP/R		E			
Diagrama de Comunicação	AP/R		E			
Desenvolvimento de Banco de Dados	AP		E	R		
Desenvolvimento do Estudo de Segurança	AP/R			E		
Desenvolvimento de Programação	AP/R			E		
Elaboração das Telas e Relatórios	AP			R		E
Desenvolvimento do Plano de Teste	AP			R	E	
Teste Integrado	AP			R	E	
Testes Unitários	AP			R	E	
Conclusão do Projeto - Refinamento	AP/R	E				

<b>Legenda</b>	
E	Executa
R	Revisa
AP	Aprova

## 5 ANÁLISE DE REQUISITOS

### 5.1 DESCRIÇÃO DOS PROCESSOS E PROCEDIMENTOS

A plataforma irá conter uma área de login e após isto, o usuário poderá criar um tópico (pergunta) relacionado as suas dúvidas. Poderá também, responder tópicos de outros usuários e utilizando de filtros poderá realizar pesquisas dinâmicas e/ou objetivas.

Ao lado esquerdo da tela, o usuário terá sua foto de perfil, pontuação e descrição do perfil. No menu superior terá as abas de “exercícios, tarefas e projetos” e um campo de notificações e logoff.

### 5.2 REQUISITOS FUNCIONAIS

Os requisitos funcionais de acordo com o *Internshipin*:

- Gerenciar alunos e professores;
- Disponibilizar Informações de alunos;
- Gerenciar projetos entre alunos e professores;
- Gerenciar exercícios postados pelos professores;
- Gerenciar tópicos com dúvidas e questões sobre matérias;
- Gerenciar perguntas e respostas feitas pelos alunos e professores;
- Categorizar os alunos mais frequentes na plataforma através de um *ranking*.

### 5.3 REQUISITOS NÃO FUNCIONAIS

Os Requisitos Não-Funcionais de acordo com o *Internshipin*:

- **Disponibilidade:** Os usuários serão habilitados ao cadastro e utilização do sistema desde que sejam matriculados na instituição de ensino cujo *Internshipin* esteja vinculado;

- **Segurança:** As informações pessoais como CPF, RG e Matrícula e Senha não serão divulgadas a público;
- **Portabilidade:** O Sistema deve ser responsivo respeitando regras de user experience, sendo possível acessá-lo na sua plenitude com o desktop através do *Google Chrome, Internet Explorer e Mozilla Firefox*;
- **Usabilidade:** A Aplicação deve ser intuitiva e responsiva para os alunos e professores;
- **Desempenho:** O sistema deverá estar online 24 horas por dia em um servidor reservado para ele.

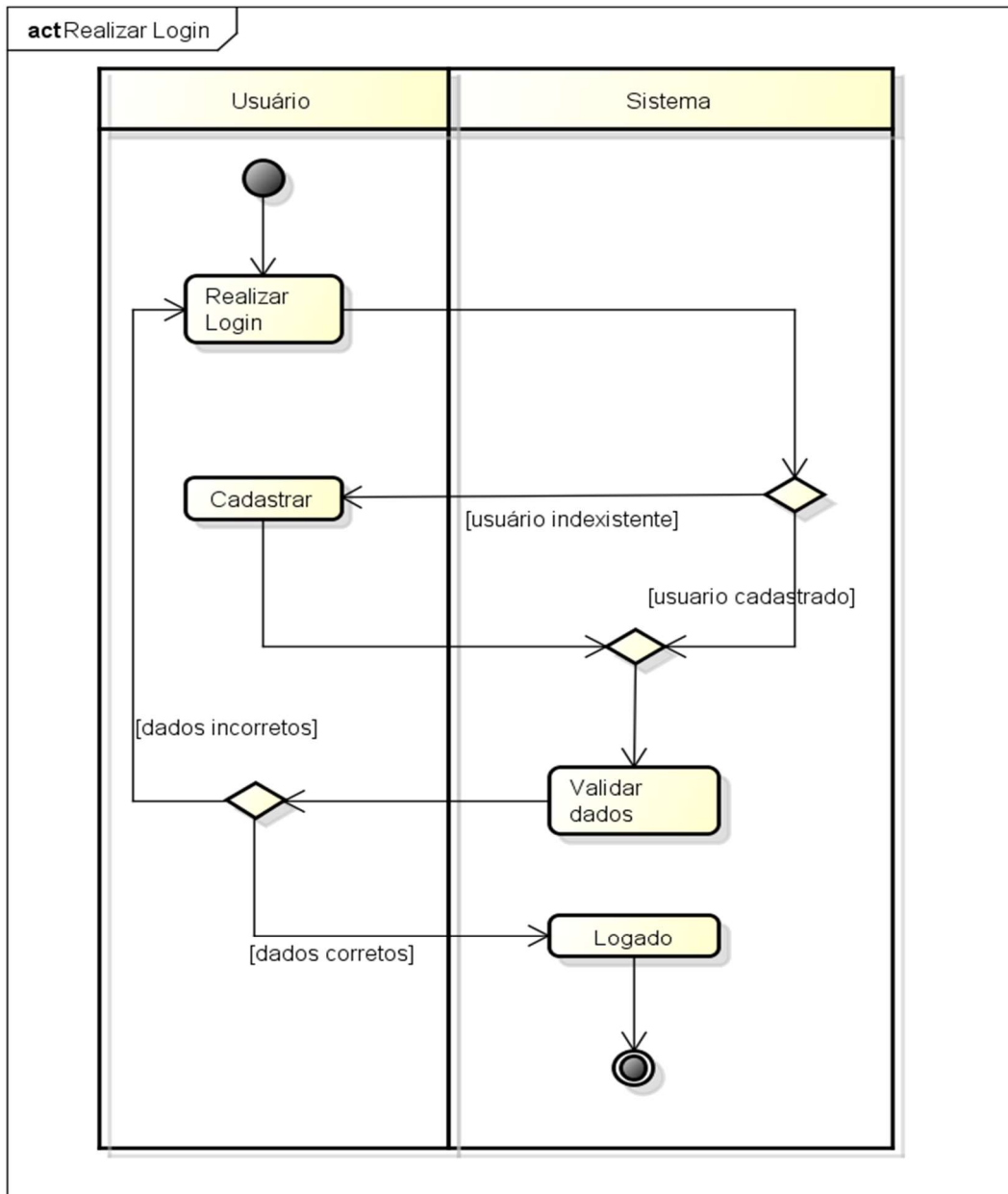
## 5.4 DIAGRAMA DE ATIVIDADES

Segundo Guedes (2009) o diagrama de atividades descreve os passos a serem percorridos para a finalização de uma atividade específica e concentra-se na representação do fluxo de controle de um evento.

### 5.4.1 DIAGRAMA DE ATIVIDADES: APLICANDO UM *LOGIN*

As etapas apresentadas no diagrama de atividade, seguem a sequência de atividades para o usuário realizar o *login* na plataforma.

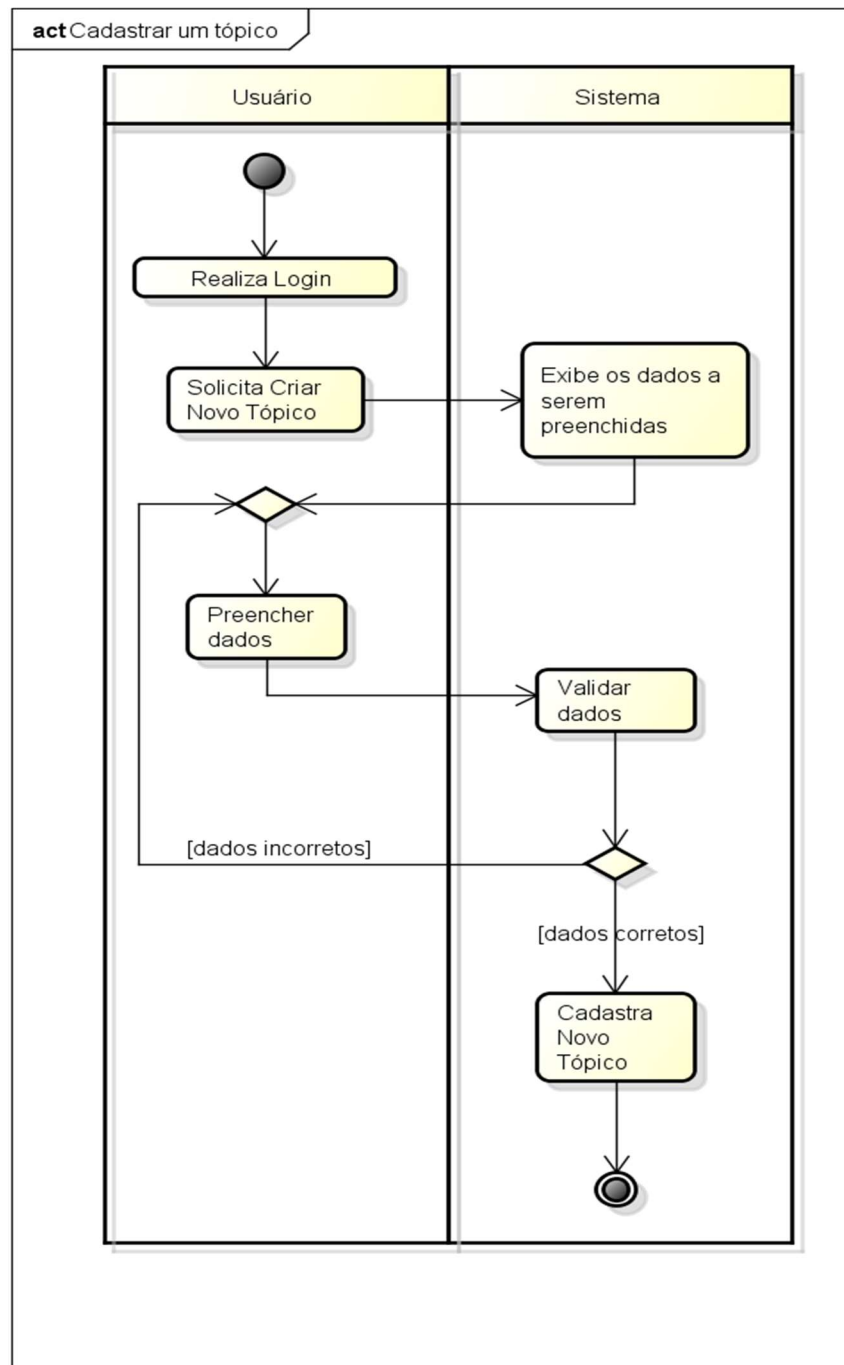
Figura 22 – Diagrama de Atividade Realizar Login



## 5.4.2 DIAGRAMA DE ATIVIDADES: CADASTRAR TÓPICO

As etapas apresentadas no diagrama de atividade, seguem a sequência de atividades para o usuário cadastrar um novo tópico na plataforma.

Figura 23 – Diagrama de Atividade Cadastrar Tópico



## 5.5 DESCRIÇÃO DOS PROBLEMAS E PROPOSTAS DE MELHORIAS

Ao longo do projeto foram identificados problemas que possivelmente ocorrerão com a aplicação. Para tais problemas que já foram identificados, foi também idealizadas algumas soluções.

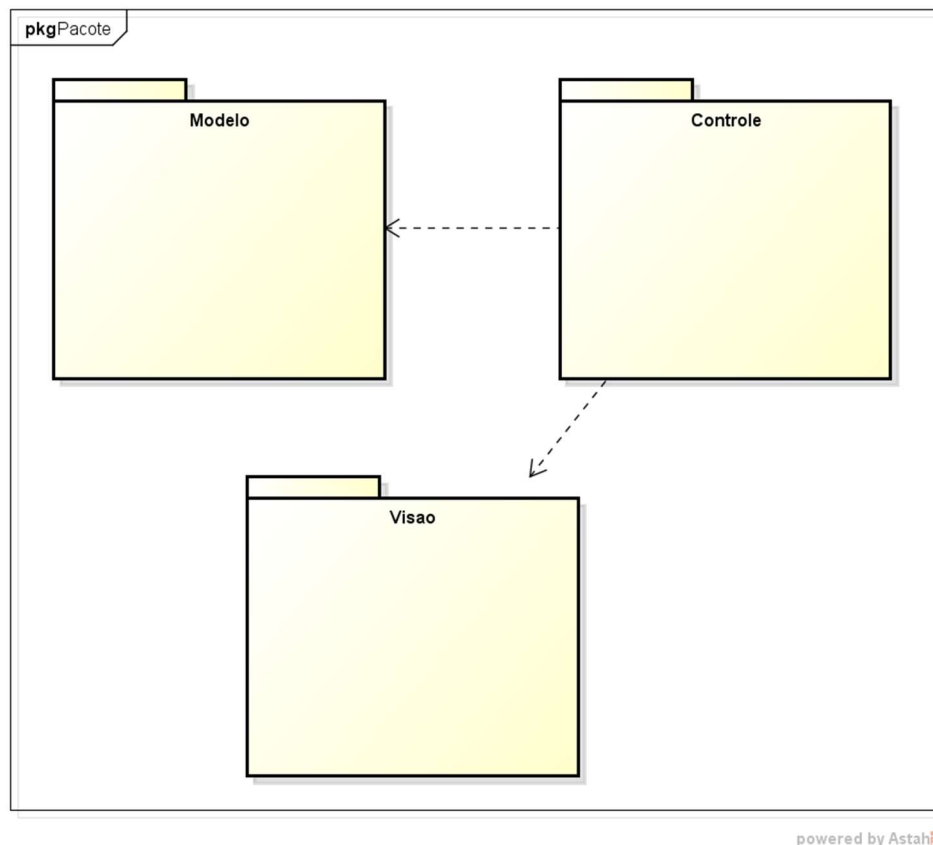
Tabela 2 – Descrição dos problemas e soluções propostas

Descrição dos Problemas	Soluções Propostas
O aluno não tinha retorno sobre os exercícios realizados	Através do <i>INTERNSHIPIN</i> a resolução dos exercícios já está disponível para o aluno consultar ao terminar o mesmo.
Falta de experiência prática	A funcionalidade de projetos irá simular o mercado de trabalho durante o desenvolvimento de um projeto por uma equipe de alunos, supervisionado por um professor.
Conhecimento adquirido não é fixado	O <i>INTERNSHIPIN</i> apresenta projetos para os alunos desenvolverem e fixar o conhecimento adquirido.
Falta de interação entre alunos e professores	Através da funcionalidade de projetos, os alunos trabalharão em equipe durante o desenvolvimento, melhorando assim a interação.
Dúvidas não são sanadas em sala	Os alunos poderão criar tópicos onde postarão suas dúvidas e outros alunos ou professores irão responder ou discutir sobre o assunto.

## 5.6 DIAGRAMA DE PACOTES PROPOSTO

De acordo com Guedes (2009) o diagrama de pacotes é um diagrama estrutural que tem por objetivo representar os subsistemas ou submódulos englobados por um sistema de forma determinar as partes que o compõem que pode ser usado de maneira independente ou associado com outros diagramas.

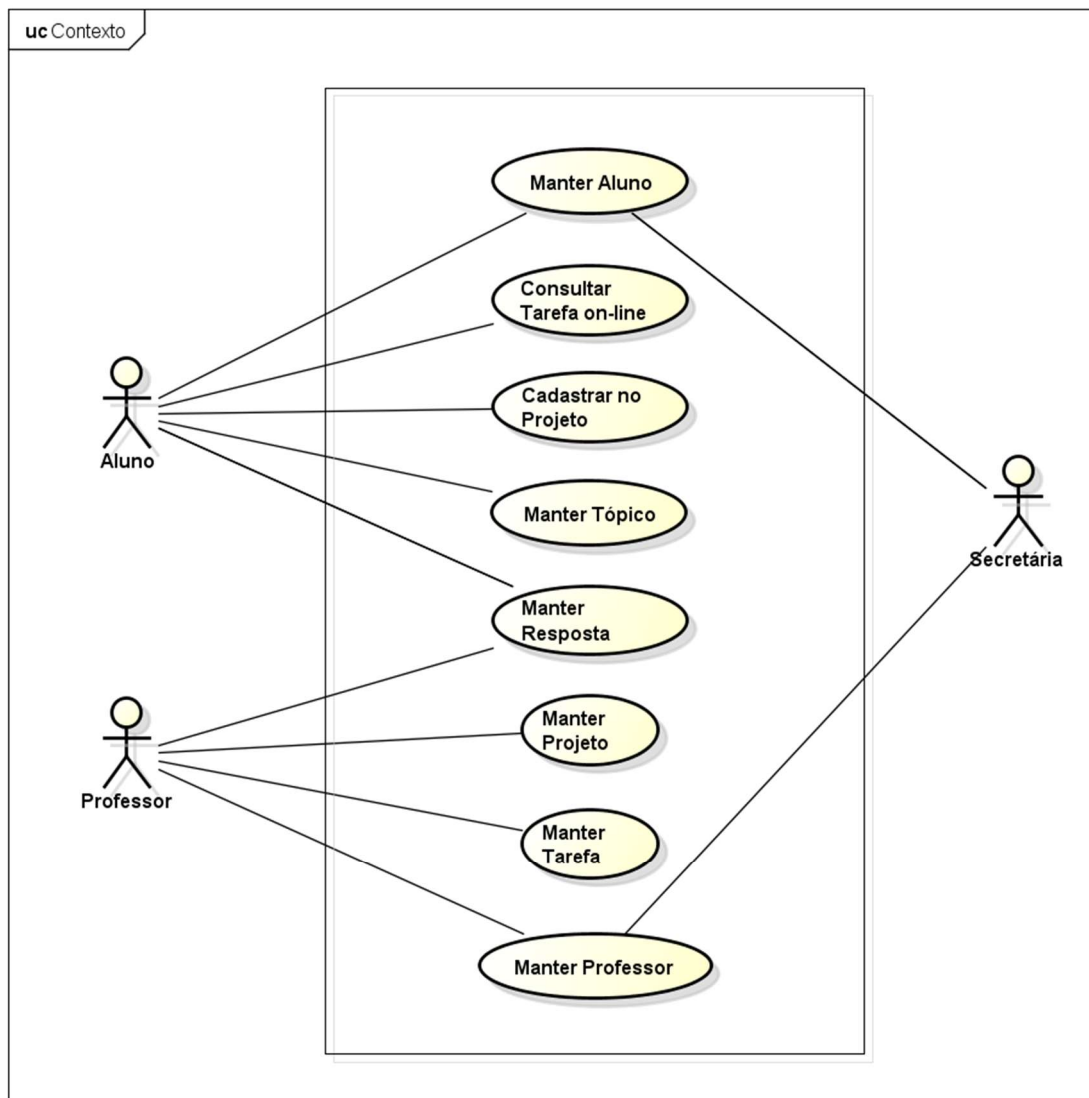
Figura 24 – Diagrama de Pacotes Proposto



## 5.7 DIAGRAMA DE CONTEXTO DO CASO DE USO

Segundo Guedes (2009) o diagrama de caso de uso é o mais informal da *UML*, utilizado normalmente nas fases de levantamento e análise de requisitos do sistema, embora venha a ser consultado durante todo o processo de modelagem e possa servir de base para outros diagramas.

Figura 25 – Diagrama de Contexto do Caso de Uso



powered by Astah

## 5.8 TABELAS DE CASO DE USO

Segundo Guedes (2009) o caso de uso representa as interações do usuário com o sistema, descrevendo as ações em que o sistema poderá executar ou não através da comunicação com usuários externos do sistema.

Na tabela 3 a seguir, é apresentado a descrição no modelo da tabela o processo Manter Tópico, exemplificando a ação de Cadastrar Tópico.

Tabela 3 – Manter Tópico (Cadastrar Tópico)

<b>Código</b>	<b>Nome</b>
<b>1</b>	<b>Manter Tópico</b>
<b>Caso de Uso Geral</b>	
<b>Ator Principal</b>	<b>Ator Secundário</b>
<b>Aluno</b>	
<b>Pré-condições</b>	
O aluno deve estar logado na plataforma para efetuar estas ações.	
<b>Pós-condições</b>	
Um novo tópico será cadastrado.	
<b>Resumo</b>	
Este caso de uso descreve as etapas percorridas pelo aluno na criação de um tópico.	
<b>Fluxo Principal – Cadastrar Tópico</b>	
<b>Atores</b>	<b>Sistema</b>
1- O aluno inicia a criação de um novo tópico.	
	2- Sistema solicita os dados para cadastro: nome do tópico, categoria, descrição da pergunta.
3- O aluno informa os dados para cadastro.	
	4- O sistema valida os dados. Se os dados forem inválidos, apresentar mensagem de erro e retornar ao passo 2.
	5- Sistema grava os dados.
	6- O sistema informa que a inclusão foi feita com sucesso.
<b>Restrições/Validações</b>	
<b>1. O aluno deve estar regularmente matriculado na instituição de ensino.</b>	

## 5.9 ESTUDO DE MERCADO

Basicamente o Estudo de Mercado é um conjunto de atividades orientadas para antever as vendas e os preços de certo produto com a finalidade de estimar as receitas futuras e examinar as possibilidades de obter resultados compensadores tanto para investimentos empresariais quanto para investimentos públicos.

### **5.9.1 PESQUISA DE MERCADO**

Ao realizar a pesquisa foi constatado que não existe uma plataforma com os objetivos do *INTERNSHIPIN*, foram encontradas algumas aplicações *web* com funcionalidades parecidas, por exemplo, o *StackOverflow* que possui tópicos com perguntas e respostas, servindo como uma forma de retirar dúvidas.

Também foi considerada a plataforma *GUJ* que possui a mesma funcionalidade da aplicação anterior, contudo o site é em português. Ambos são voltados para a área de *TI* porém nenhum deles oferece oportunidades de experiência prática.

Já a plataforma de ensino do *ALURA*, assim como o *INTERNSHIPIN*, também é uma plataforma especializada em ensino na área de *TI* e oferece algumas oportunidades de experiência prática, através dos exercícios presentes em seus cursos. Porém os seus pacotes de assinatura são somente para no mínimo 1 ano, o que acaba desanimando alguns alunos.

Por outro lado foram encontrados algumas aplicações desenvolvidas para instituições de ensino, mas estes assim como os anteriores não possuem a funcionalidade de organizar estudantes em grupos para a realização de projetos simulando o mercado de trabalho.

### **5.9.2 ANÁLISE COMPARATIVA**

A Tabela 3 mostra uma análise do mercado, mostrando através de um quadro as diferenças e semelhanças entre o sistema proposto e os sistemas existentes.

Tabela 4 – Análise Comparativa

<b>Análise Comparativa</b>				
<b>Características</b>	<i>Stackoverflow</i>	<i>ALURA</i>	<i>GUJ</i>	<i>INTERNSHIPIN</i>
<b>Desenvolvido para Instituições de Ensino</b>	Não	Sim	Não	SIM
<b>Possuí Exercícios Práticos</b>	Não	Sim	Não	SIM
<b>Estimula Trabalho em Equipe</b>	Não	Não	Não	SIM
<b>Controle de Tarefas</b>	Não	Sim	Não	SIM
<b>Controle de Projetos</b>	Não	Não	Não	SIM
<b>Controle de Tópicos</b>	Sim	Sim	Sim	SIM

### 5.9.3 CONCLUSÃO DO ESTUDO DE MERCADO

Concluimos que o *INTERNSHIPIN*, comparado com outras presentes no mercado, é a plataforma que melhor supre as necessidades dos alunos de TI presentes em instituições de ensino, pois foca na comunicação dos aluno entre si e com professores, para melhorar a troca de conhecimentos e experiências necessários para ingressar no mercado de trabalho. Além de incentivar o trabalho em equipe e a experiência prática através do desenvolvimento de projetos presentes na plataforma.

### 5.10 ANÁLISE DE RISCOS

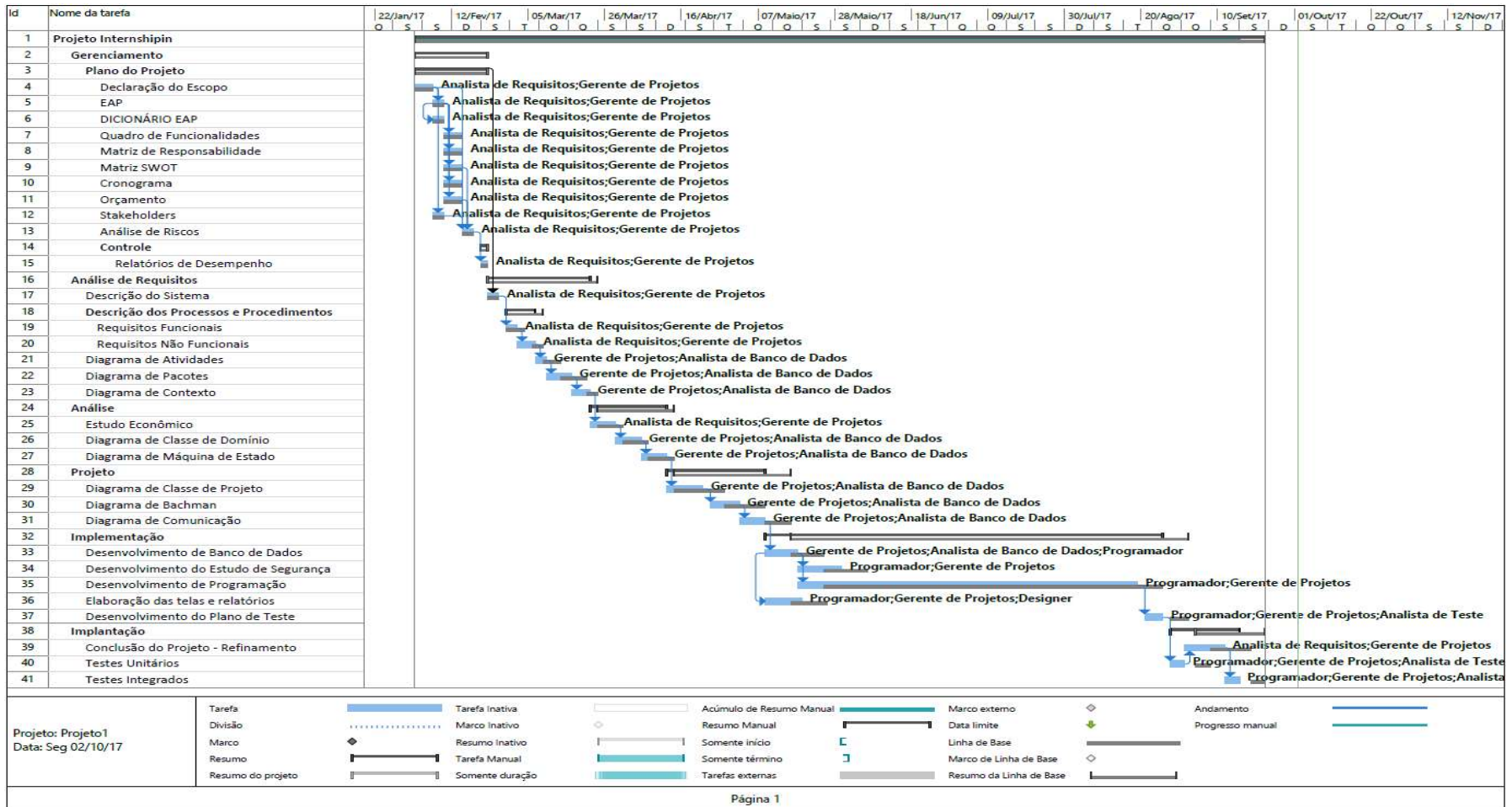
A análise de riscos é um estudo técnico que consiste em identificar e analisar os possíveis eventos que podem ser colocados como riscos presentes no ambiente de trabalho, considerando o meio ambiente e presença de terceiros.

Tabela 5 – Análise de riscos

Riscos - Análise Qualitativa										
Id	Evento	Probabilidade	Impacto nos objetivos do projeto							
			Escopo		Custo		Prazo		Qualidade	
1	Mercado de atuação restrito	20%	3	Médio	1	Baixo	1	Baixo	1	Baixo
2	Dsiponibilidade dos docentes	20%	2	Médio	1	Baixo	2	Médio	3	Médio
3	Pouco conhecimento da linguagem de programação utilizada	50%	3	Médio	2	Médio	3	Médio	4	Alto
4	Pouco conhecimento na parte de Educação/TI	60%	4	Alto	2	Médio	3	Médio	4	Alto
5	Mudança nos custos	15%	2	Médio	4	Alto	2	Médio	1	Baixo
6	Atraso na Entrega dos layouts	20%	2	Médio	2	Médio	4	Alto	3	Médio
7	Dsiponibilidade dos alunos	70%	3	Médio	2	Médio	4	Alto	3	Médio
8	Concorrentes	10%	1	Baixo	1	Baixo	1	Baixo	1	Baixo
9	Mudanças no cronograma	80%	3	Médio	1	Baixo	4	Alto	4	Alto
10	Alteração do escopo	25%	4	Alto	2	Médio	3	Médio	2	Médio

## 5.11 CRONOGRAMA

Figura 26 – Gráfico de Gantt



## 5.12 QUADRO DE FUNCIONALIDADES

Tabela 6 – Quadro de funcionalidades

<b>Nº</b>	<b>Processos a serem contemplados</b>	<b>Sistema Proposto</b>
<b>1</b>	Controlar Categorias	<b>X</b>
<b>2</b>	Controlar Subcategorias	<b>X</b>
<b>3</b>	Controlar Tópicos	<b>X</b>
<b>4</b>	Controlar Respostas	<b>X</b>
<b>5</b>	Controlar Projetos	<b>X</b>
<b>6</b>	Controlar Usuários	<b>X</b>
<b>7</b>	Controlar Registros	<b>X</b>
<b>Pontos de função ajustados</b>		<b>146,64</b>
<b>Quantidade de horas por pontos de função ajustado</b>		<b>4,4h</b>
<b>Custo do pontos de função ajustado (R\$)</b>		<b>150</b>
<b>Total de custo de pontos de função</b>		<b>21.996,00</b>
<b>Prazo de execução (meses)</b>		<b>5 Meses (4,73)</b>

Os cálculos do desenvolvimento dos pontos de função podem ser encontrados no Apêndice II.

## **6 ANÁLISE**

### **6.1 DESENVOLVIMENTO DO ESTUDO ECONÔMICO**

O retorno financeiro é esperado com 12 parcelas mensais. O valor da venda direta da plataforma será de R\$7.000,00. Para o controle financeiro das entradas e saídas do projeto, foi desenvolvido o fluxo de caixa.

As saídas do fluxo de caixa será com a manutenção do projeto, que será levado em conta o valor/hora do programador do sistema, que irá realizar manutenções por 23 horas mensais.

A entrada será de acordo com a venda e manutenção da plataforma, no valor de R\$7.000,00 e R\$1.000,00 respectivamente.

#### **6.1.1 CUSTOS E DESPESAS**

Para contabilizar os custos e despesas do desenvolvimento do projeto, com o auxílio do site [rinfo.com.br](http://rinfo.com.br), [copel.com](http://copel.com), [netcombo.com.br](http://netcombo.com.br) e [lovemondays.com.br](http://lovemondays.com.br), calculamos os valores de homem/hora e simulamos os gastos com energia e internet, conforme a tabela abaixo.

Tabela 7 – Custos e despesas

<b>Custos do Projeto</b>											
Custos Diretos	Fevereiro	Março	Abril	Maiο	Junho	Julho	Agosto	Setembro	Outubro	Novembro	Total
Gerente de Projetos	R\$ 1.082,80	R\$ 1.082,80	R\$ 1.082,80	R\$ 1.082,80	R\$ 1.082,80	R\$ 1.082,80	R\$ 1.082,80	R\$ 1.082,80	R\$ 1.082,80	R\$ 1.082,80	R\$ 10.828,00
Analista de Requisitos	R\$ 2.539,84	R\$ 2.539,84	R\$ 2.539,84	R\$ 2.539,84	R\$ 2.539,84	R\$ 2.539,84					R\$ 15.239,04
Analista de Banco de Dados						R\$ 1.582,38	R\$ 1.582,38	R\$ 1.582,38	R\$ 1.582,38	R\$ 1.582,38	R\$ 7.911,90
Programador						R\$ 2.000,00	R\$ 2.000,00	R\$ 2.000,00	R\$ 2.000,00	R\$ 2.000,00	R\$ 10.000,00
Analista de Teste									R\$ 1.667,58	R\$ 1.667,58	R\$ 3.335,16
Designer					R\$ 256,60	R\$ 256,60	R\$ 256,60				R\$ 769,80
<b>Total dos Custos Diretos:</b>											<b>R\$ 48.083,90</b>
Custos Indiretos	Fevereiro	Março	Abril	Maiο	Junho	Julho	Agosto	Setembro	Outubro	Novembro	Total
Internet	R\$70,00	R\$70,00	R\$70,00	R\$70,00	R\$70,00	R\$70,00	R\$70,00	R\$70,00	R\$70,00	R\$70,00	R\$700,00
Energia	R\$178,55	R\$178,55	R\$178,55	R\$178,55	R\$178,55	R\$178,55	R\$178,55	R\$178,55	R\$178,55	R\$178,55	R\$1.785,50
Transporte	R\$7,20	R\$7,20	R\$7,20	R\$7,20	R\$7,20	R\$7,20	R\$7,20	R\$7,20	R\$7,20	R\$7,20	R\$72,00
Alimentação	R\$22,00	R\$22,00	R\$22,00	R\$22,00	R\$22,00	R\$22,00	R\$22,00	R\$22,00	R\$22,00	R\$22,00	R\$220,00
<b>Total dos Custos Indiretos:</b>											<b>R\$2.777,50</b>
<b>Total do Projeto</b>											<b>R\$ 50.861,40</b>
<a href="http://www.copel.com/hpcopel/simulador/index.htm">http://www.copel.com/hpcopel/simulador/index.htm</a>											
<a href="http://www.netcombo.com.br/recargas-banda-larga-sem-mensalidade">http://www.netcombo.com.br/recargas-banda-larga-sem-mensalidade</a>											
<a href="http://www.rhinfo.com.br/sal-ti.htm">http://www.rhinfo.com.br/sal-ti.htm</a>											
<a href="https://www.lovedmondays.com.br/salarios/cargo/salario-desenvolvedor-ruby-on-rails">https://www.lovedmondays.com.br/salarios/cargo/salario-desenvolvedor-ruby-on-rails</a>											

Tabela 8 – Custo por Homem/Hora

Gerente de Projetos	R\$ 27,07 H/H
Analista de Requisitos	R\$ 15,11 H/H
Analista de Banco de Dados	R\$ 9,41 H/H
Programador	R\$ 11,90 H/H
Analista de Teste	R\$ 9,92 H/H
Designer	R\$ 6,46 H/H

## 6.1.2 CALCULO DO FLUXO DE CAIXA

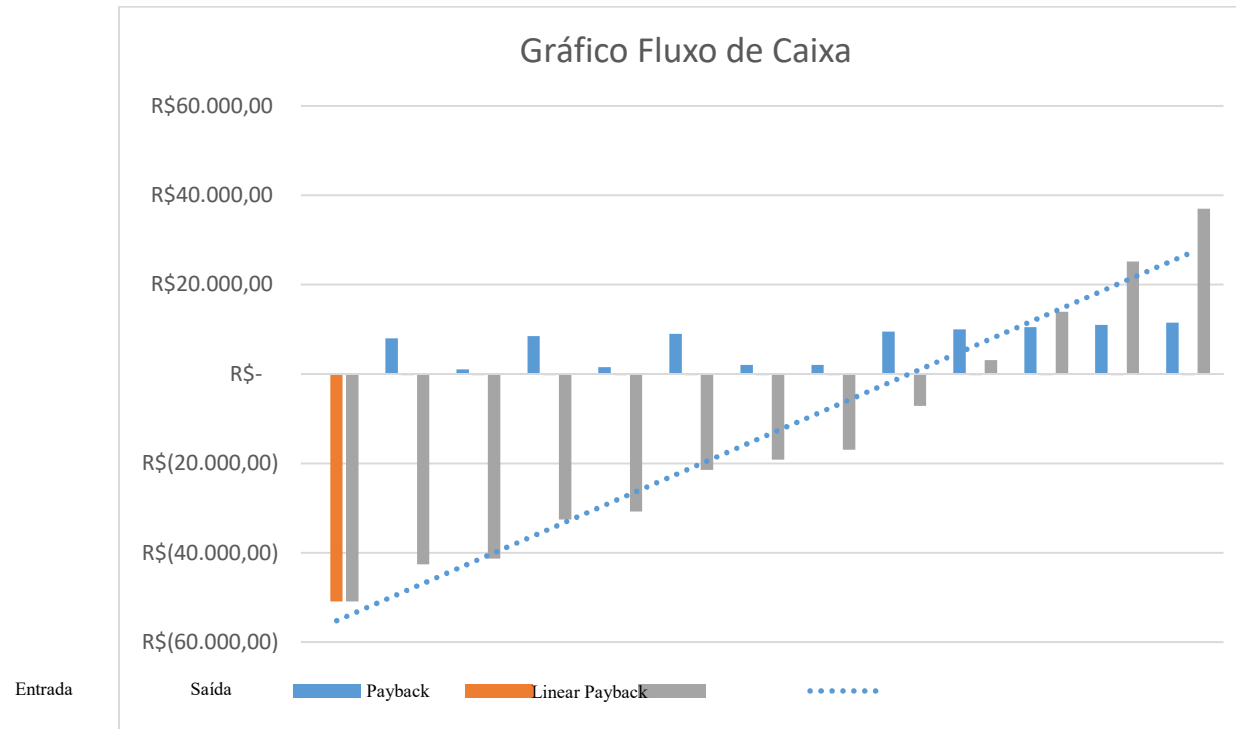
Para controlar as entradas e saídas do capital do projeto, foi elaborado o fluxo de caixa, que estão listados as entradas como a venda e manutenção do projeto cobrado do cliente e as saídas baseado no valor homem/hora do programador do projeto.

Tabela 9 – Planilha do Fluxo de Caixa

Fluxo de Caixa				
Período	Entrada	Saída	Payback	LUCRO
0	R\$ -	-R\$ 50.861,40	-R\$ 50.861,40	-R\$ 50.861,40
1	R\$ 8.000,00	-R\$ 273,70	-R\$ 42.587,70	R\$ 8.273,70
2	R\$ 9.000,00	-R\$ 273,70	-R\$ 33.314,00	R\$ 9.273,70
3	R\$ 10.000,00	-R\$ 273,70	-R\$ 23.040,30	R\$ 10.273,70
4	R\$ 11.000,00	-R\$ 273,70	-R\$ 11.766,60	R\$ 11.273,70
5	R\$ 12.000,00	-R\$ 273,70	R\$ 507,10	R\$ 12.273,70
6	R\$ 13.000,00	-R\$ 273,70	R\$ 13.780,80	R\$ 13.273,70
7	R\$ 14.000,00	-R\$ 273,70	R\$ 28.054,50	R\$ 14.273,70
8	R\$ 15.000,00	-R\$ 273,70	R\$ 43.328,20	R\$ 15.273,70
9	R\$ 16.000,00	-R\$ 273,70	R\$ 59.601,90	R\$ 16.273,70
10	R\$ 17.000,00	-R\$ 273,70	R\$ 76.875,60	R\$ 17.273,70
11	R\$ 18.000,00	-R\$ 273,70	R\$ 95.149,30	R\$ 18.273,70
12	R\$ 19.000,00	-R\$ 273,70	R\$ 114.423,00	R\$ 19.273,70
TMA	8%			R\$ 114.423,00
VPL	R\$ 46.123,76			
TIR	24%			

Taxa foi baseada na taxa SELIC do dia 28/09/2017 de 8,25%

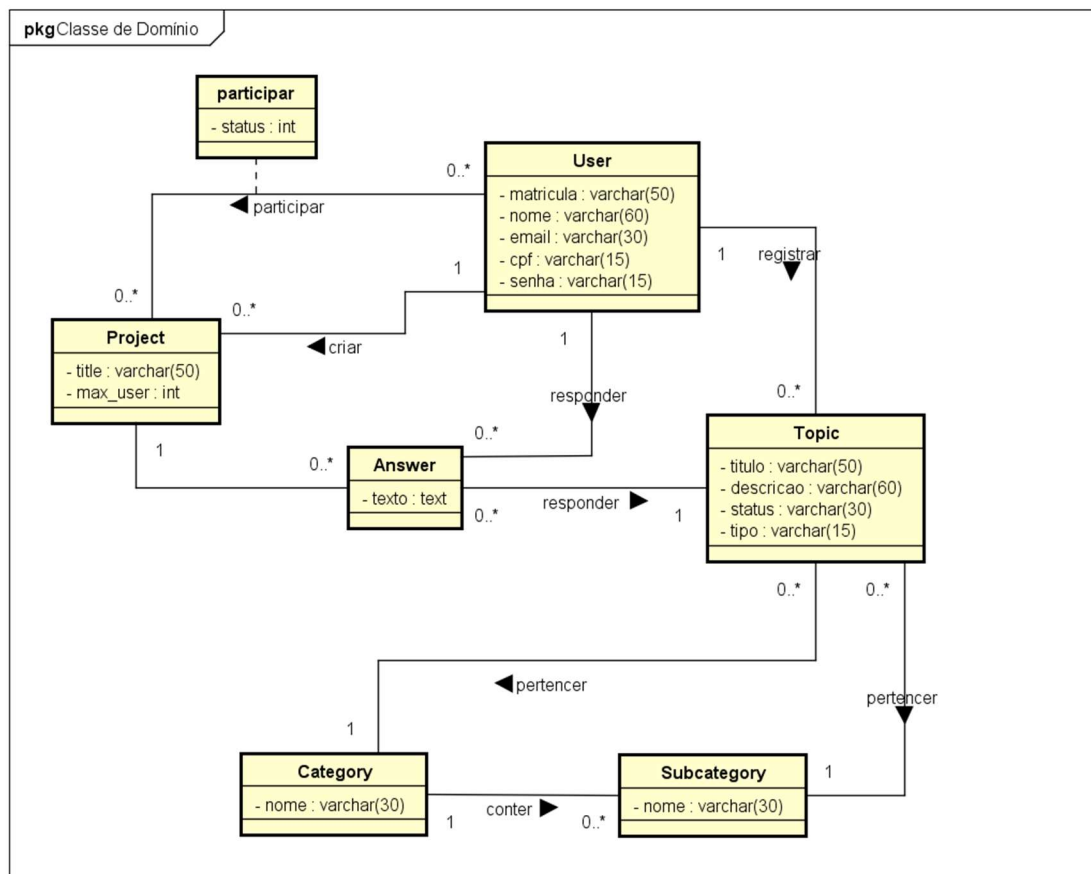
Figura 27 – Gráfico do Fluxo de Caixa



## 6.2 DIAGRAMA DE CLASSE DE DOMÍNIO

O Diagrama de Classe de Domínio conforme dito por Guedes (2009) provavelmente é o mais utilizado e o mais importante dos diagramas da *UML*. Serve de apoio para a maioria dos demais diagramas. Como o próprio nome diz, ele define a estrutura das classes utilizadas pelo sistema, e, também determina os atributos e os métodos pertencente às classes.

Figura 28 – Diagrama de Classe de Domínio

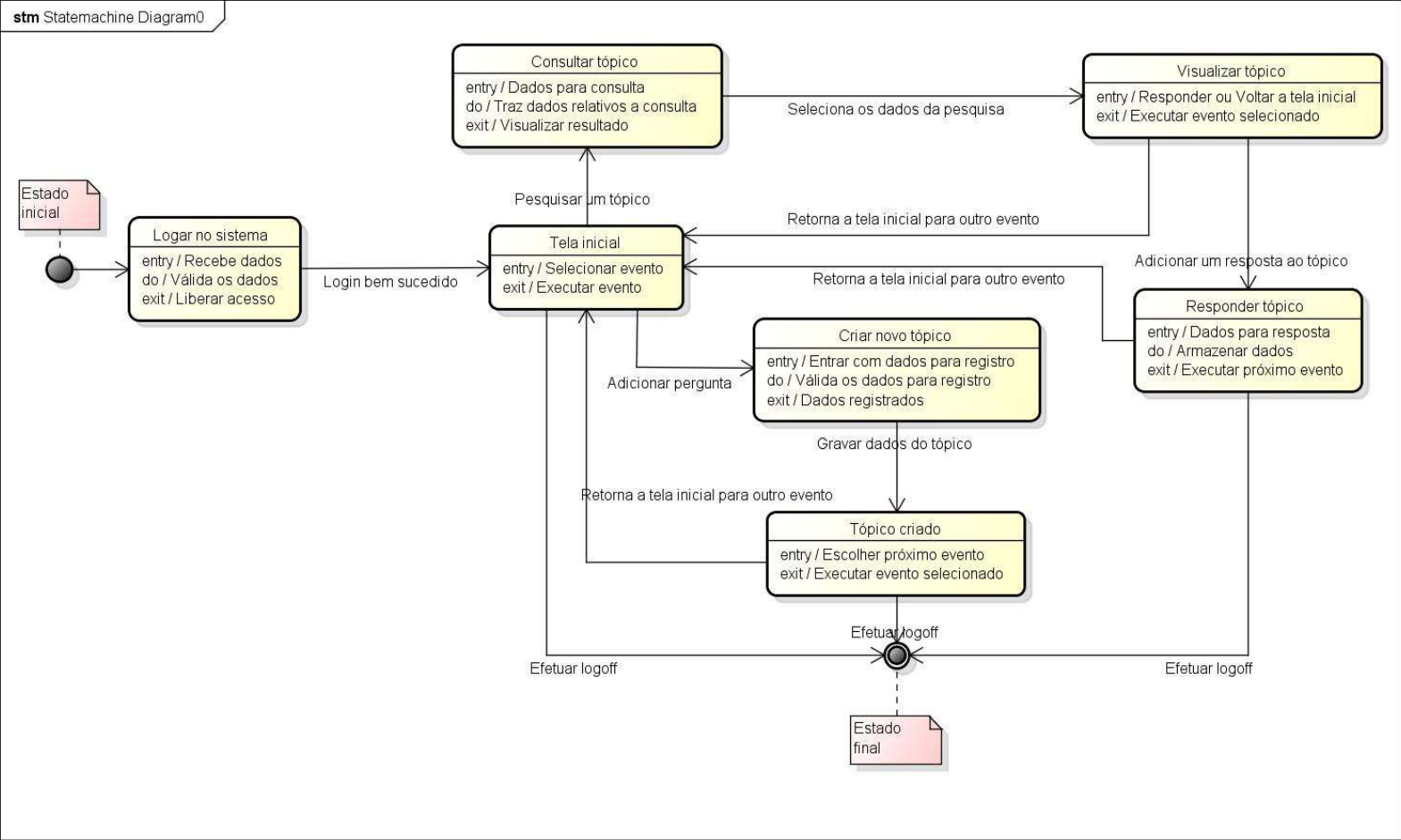


powered by Astah

## 6.3 DIAGRAMA DE MÁQUINA DE ESTADO

De acordo com Guedes (2009) o diagrama de máquina de estados demonstra o comportamento de um elemento por meio de um conjunto finito de transições de estado, ou seja, uma máquina de estados.

Figura 29 – Diagrama de Máquina de Estado



powered by Astah

## 7 PROJETO

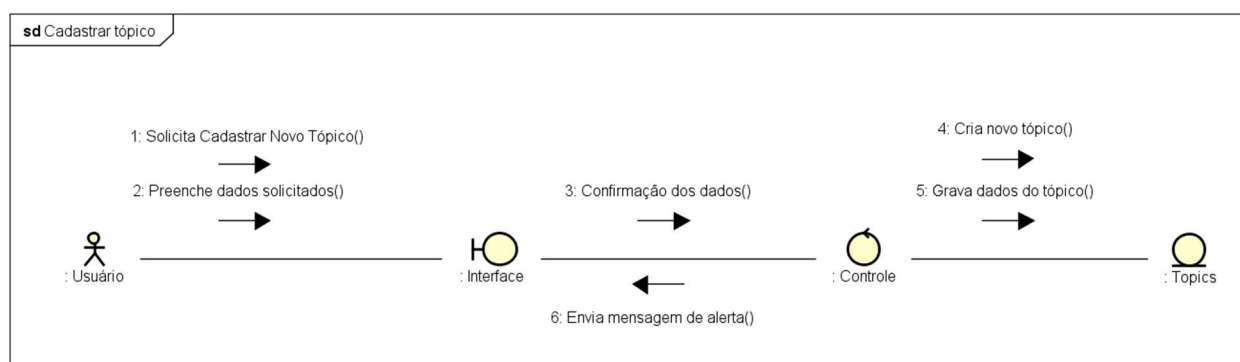
### 7.1 DIAGRAMA DE COMUNICAÇÃO

De acordo com Guedes (2009) o diagrama de comunicação é um diagrama que apresenta a interação dos objetos de uma forma sequencial simples, porém com um enfoque distinto, diferente do diagrama de sequência que não se preocupa com a temporalidade do processo, concentrando-se em como os objetos estão vinculados e quais mensagens trocam entre si durante o processo.

#### 7.1.1 DIAGRAMA DE COMUNICAÇÃO: CADASTRANDO UM TÓPICO

As etapas enumeradas seguem a ordem de como os objetos serão tratados pela plataforma para realizar a criação de um novo tópico.

Figura 30 – Diagrama de Comunicação – Cadastrando um tópico

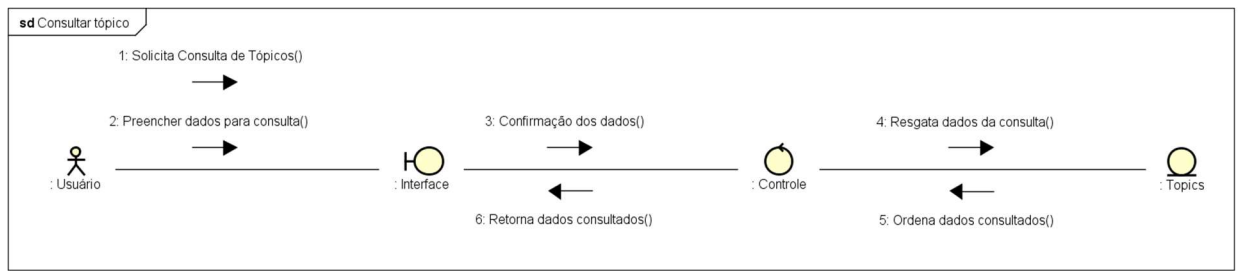


powered by Astah

#### 7.1.2 DIAGRAMA DE COMUNICAÇÃO: CONSULTANDO UM TÓPICO

As etapas enumeradas seguem a ordem de como os objetos serão tratados pela plataforma para realizar a consulta de um tópico.

Figura 31 – Diagrama de Comunicação – Consultando um tópico

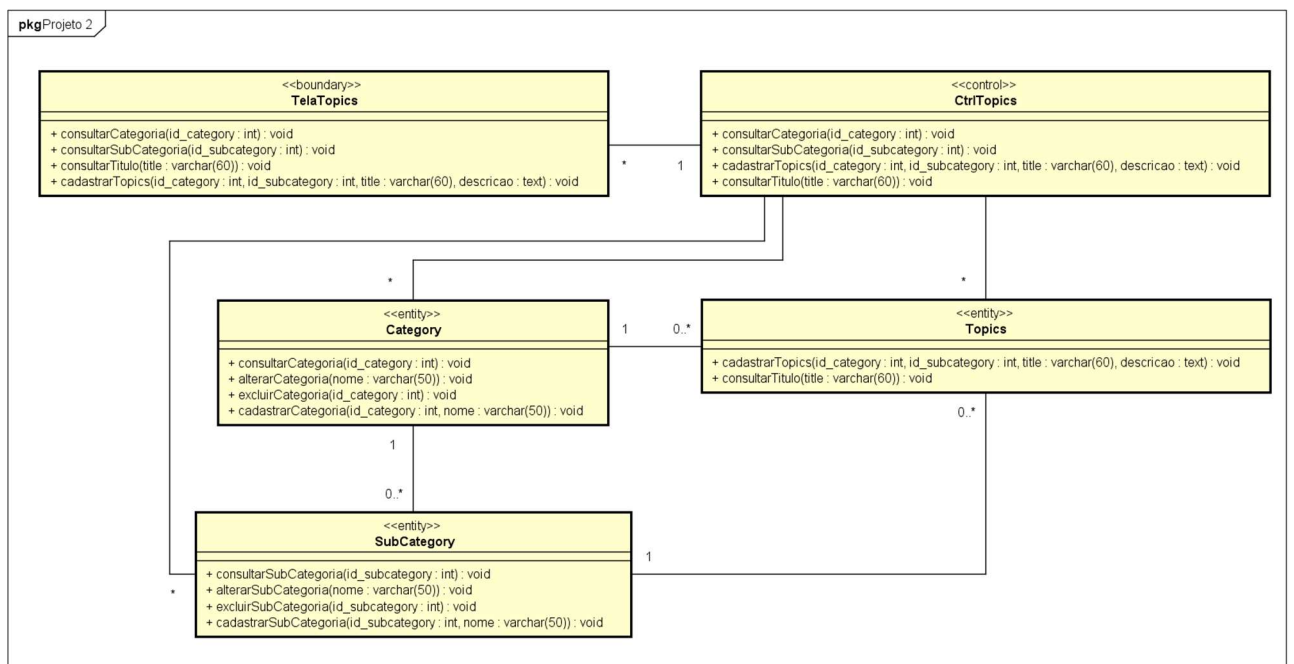


powered by Astah

## 7.2 DIAGRAMA DE CLASSE DO PROJETO

De acordo com Guedes (2009) o diagrama de classe de projeto é o diagrama final que apresenta todas as informações necessárias para criação do software e interfaces da aplicação. Como: atributos, classes, cardinalidades, etc.

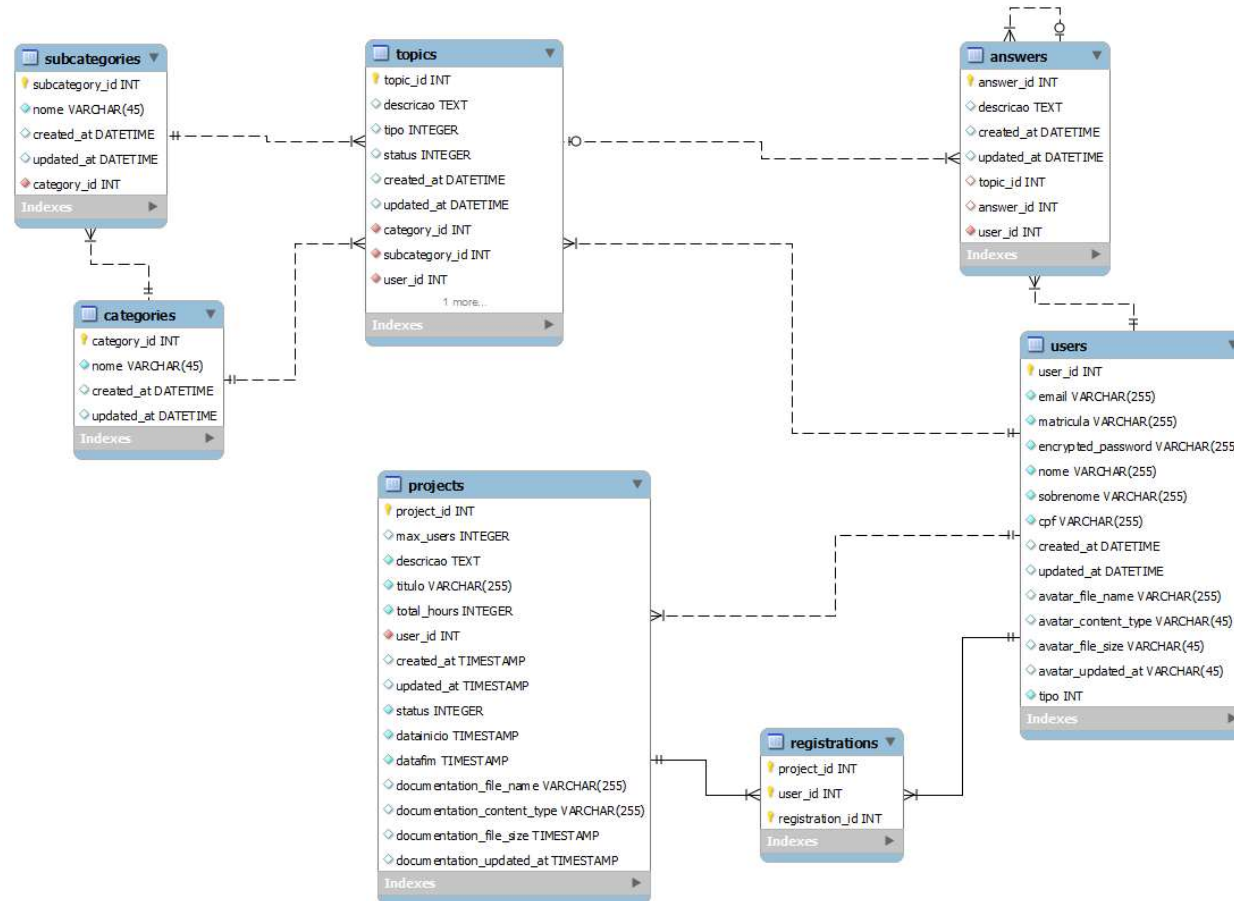
Figura 32 – Diagrama de Classe do Projeto (Manter Tópicos)



powered by Astah

### 7.3 DIAGRAMA DE BACHMAN

Figura 33 – Diagrama de Bachman



## **7.4 PROJETO DE BANCO DE DADOS**

O projeto a ser desenvolvido possuirá um banco de dados baseado no modelo Objeto-Relacional (ORM).

O objetivo de escolhermos este modelo é suprir as discrepâncias entre o paradigma Orientado a Objetos e o Modelo Entidade-Relacional, criando um mapeamento entre o modelo Orientado a Objetos e o modelo relacional.

Com o uso de tal modelo, estamos priorizando um código mais limpo e elegante, facilitando futuras manutenções na aplicação que estamos a desenvolver.

### **7.4.1 SCRIPT DE BANCO DE DADOS**

O script do banco de dados pode ser visualizado no Apêndice III da documentação.

## 7.4.2 DIMENSIONAMENTO DO BANCO DE DADOS

Tabela 10 – Dimensionamento do banco de dados

X	Tabela	Users	Topics	Answers	Categories	Subcategories	Projects	Registrations
A	Quantidade de registro das tabelas	100	1000	3000	20	50	20	100
B	Somatório dos tamanhos das colunas	200	1000	1000	20	20	100	10
C	Quantidade de colunas	11	9	7	4	4	15	3
D	Quantidade de colunas de tamanho variável	6	1	0	1	1	3	0
E	Cabeçalho da linha = E = 3 bytes(tamanho fixo - constante)	3	3	3	3	3	3	3
F	Percentual livre dos blocos (padrão é 10)	10	10	10	10	10	10	10
G	Tamanho do bloco (padrão é 8192 = 8k)	8192	8	8	8	8	8	8
H	Padrão da alocação da Tablespace (65536 = 64k)	65536	65536	65536	65536	65536	65536	65536
TFT1	Tamanho físico da tabela em bytes	21260088115200000,00	141557760000000,00	0,00	25165824000,00	62914560000,00	1415577600000,00	0,00
TFT2	Tamanho físico da tabela em kb	21260088115200,00	141557760000,00	0,00	25165824,00	62914560,00	1415577600,00	0,00
<b>Somatório dos tamanhos das tabelas em KB</b>		21403149533184,00						
<b>Percentual de segurança (10% a 30%)</b>		25,00%						
<b>Área de dados (Somatório + Acréscimo) em KB</b>		26753936916480,00						

## 7.5 LISTA DE TABELAS

Tabela 11 – Lista de tabelas

Campo	Tipo/Tamanho	Tipo de Chave	Obrigatório	Restrição de Integridade	Tabela Referenciada	Extra
Users						
user_id	integer	PK	X	NOT NULL		Auto Incremento
matricula	varchar(255)		X	NOT NULL		
nome	varchar(255)		X	NOT NULL		
sobrenome	varchar(255)		X	NOT NULL		
email	varchar(255)		X	NOT NULL		
avatar_file_name	varchar(255)					
avatar_content_type	varchar(255)					
avatar_file_size	integer					
avatar_updated_at	datetimeoffset					
tipo	integer		X	NOT NULL		
cpf	varchar(255)		X	NOT NULL		
encrypted_password	varchar(255)					
Topics						
topic_id	integer	PK	X	NOT NULL		Auto Incremento
titulo	varchar(255)		X	NOT NULL		
descricao	text		X	NOT NULL		
status	integer			NOT NULL		
created_at	datetimeoffset					
updated_at	datetimeoffset					
user_id	integer	FK	X	NOT NULL	Users	

category_id	integer	FK	X	NOT NULL	Categories	
subcategory_id	integer	FK	X	NOT NULL	Subcategories	
Answers						
answer_id	integer	PK	X	NOT NULL		Auto Incremento
descricao	text		X	NOT NULL		
created_at	datetimeoffset					
updated_at	datetimeoffset					
user_id	integer	FK	X	NOT NULL	Users	
topic_id	integer	FK		NOT NULL	Topics	
answer_id	integer	FK		NOT NULL	Answers	
Categories						
category_id	integer	PK	X	NOT NULL		Auto Incremento
nome	varchar(255)		X	NOT NULL		
created_at	datetimeoffset					
updated_at	datetimeoffset					
Subcategories						
subcategory_id	integer	PK	X	NOT NULL		Auto Incremento
category_id	integer	FK	X	NOT NULL		
nome	varchar(255)		X	NOT NULL		
created_at	datetimeoffset					
updated_at	datetimeoffset					
Projects						
project_id	integer	PK	X	NOT NULL		Auto Incremento
max_users	integer					
titulo	varchar(255)		X	NOT NULL		
descricao	text		X	NOT NULL		
total_hours	integer		X	NOT NULL		

user_id	integer	FK	X	NOT NULL	Users	
created_at	datetimeoffset					
updated_at	datetimeoffset					
status	integer			NOT NULL		
datainicio	datetime		X	NOT NULL		
datafim	datetime		X	NOT NULL		
documentation_file_name	varchar(255)					
documentation_content_type	varchar(255)					
documentation_file_size	integer					
documentation_updated_at	datetimeoffset					
Registrations						
registration_id	integer	PK	X	NOT NULL		Auto Incremento
project_id	integer	FK	X	NOT NULL	Projects	
user_id	integer	FK	X	NOT NULL	Users	

## 7.6 RESTRIÇÃO DE INTEGRIDADE

Tabela 12 – Restrição de integridade

ENTIDADE	INSERÇÃO	ALTERAÇÃO	DELEÇÃO
<b>USERS</b>	Para a inserção de um novo User, os campos Matrícula, Nome, Email, Tipo e Password deverão estar preenchidos.	Para alteração de usuário, ele precisa estar autenticado como ele próprio.	Não há deleção de users.
<b>TOPICS</b>	Para a inserção de um novo Topic, os campos Título, Descrição, Categoria e Subcategoria devem estar selecionados.	Para alteração de um tópico, o criador dele ou o admin devem estar autenticados.	Para deleção de topic, o criador dele ou o admin devem estar autenticados.
<b>CATEGORIES</b>	Para a inserção de uma nova categoria, o campo nome deve estar selecionado e o tipo de User deve ser Admin.	Para a alteração de uma category, o admin deve estar autenticado.	Para a deleção de uma category, o admin deve estar autenticado.
<b>SUBCATEGORIES</b>	Para a inserção de uma nova subcategoria, o campo nome deve estar selecionado e o tipo de User deve ser Admin.	Para a alteração de uma subcategory, o admin deve estar autenticado.	Para a deleção de uma subcategory, o admin deve estar autenticado.

<b>PROJECTS</b>	Para a inserção de um novo Project os campos titulo, descricao, total_hours e datainicio e datafim devem estar preenchidos e o usuário precisa ter permissão de Professor.	Para a alteração de um project, o professor criador ou o admin devem estar autenticados	Para a deleção de uma subcategory, o admin deve estar autenticado.
<b>REGISTRATIONS</b>	Para a inserção de um novo Registration, o usuário deve ter permissão de Aluno e o Project deve ter pelo menos uma vaga disponível.	Para a alteração de um Registration, o professor criador do project ou o admin ou o aluno que se inscreveu devem estar autenticados.	Para a deleção de um Registration, o admin ou o aluno que se inscreveu devem estar autenticados.

## 8 IMPLEMENTAÇÃO

### 8.1 ESTUDO DE INFRA ESTRUTURA DE TI

O sistema estará hospedado no Heroku, podendo ser acessado através de qualquer dispositivo que possua acesso a navegadores como *Google Chrome*, *Internet Explorer* e *Mozilla Firefox*, desde que haja conexão de internet.

### 8.2 ANÁLISE DE INFRAESTRUTURA

De acordo com Poojary (2015) Heroku é um serviço em nuvem *pay-as-you-go*, oferecendo que empresas mudem suas operações sem os tradicionais *data centers*. Possui ambiente dinâmico possibilitando ao desenvolvedor simplesmente “jogar” o código neste ambiente e deixar com que o próprio Heroku faça o resto. Há um pouco de perda de controle, porém há ganho em simplicidade e velocidade.

#### 8.2.1 CONFIGURAÇÃO DE HARDWARE

Para acessar a aplicação é necessário utilizar qualquer dispositivo com as seguintes configurações:

- Possuir um navegador instalado (ex.: *Google Chrome*, *Mozilla Firefox*, *Microsoft Edge*);
- Ter acesso à internet (por cabo ou *wireless*);
- As restrições de Memória RAM variam de acordo com o dispositivo;
- Computador/Notebook: 1gb;
- *Smartphone*: 512mb.

## 8.2.2 VANTAGENS

O Heroku possui algumas vantagens interessantes, principalmente para os desenvolvedores.

- Possui interface gráfica;
- Alinhamento de *deploys* é feito juntamente ao GIT, facilitando a manipulação da aplicação através do próprio controle de versão;
- Plano básico é gratuito;
- Possui suporte para diversos bancos de dados e aplicações de diversas linguagens.

## 8.2.3 DESVANTAGENS

- Os planos mais simples oferecem limitação de acesso;
- Os preços são mais altos do que servidores compartilhados.

## 8.2.4 INFRAESTRUTURA E CRESCIMENTO

O Heroku funciona de maneira fácil e intuitiva para aumentar a potência do servidor. Tudo é feito rapidamente através da interface gráfica, podendo com um simples clique aumentar a quantidade de RAM, por exemplo, para sua aplicação, desde que você possua um plano com essa possibilidade de crescimento.

Figura 34 – Como o Heroku funciona

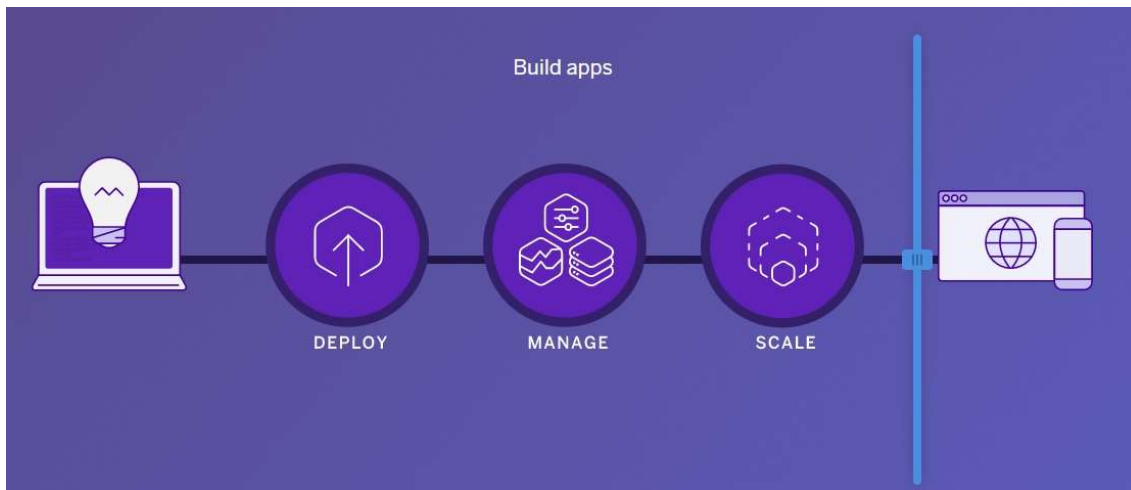
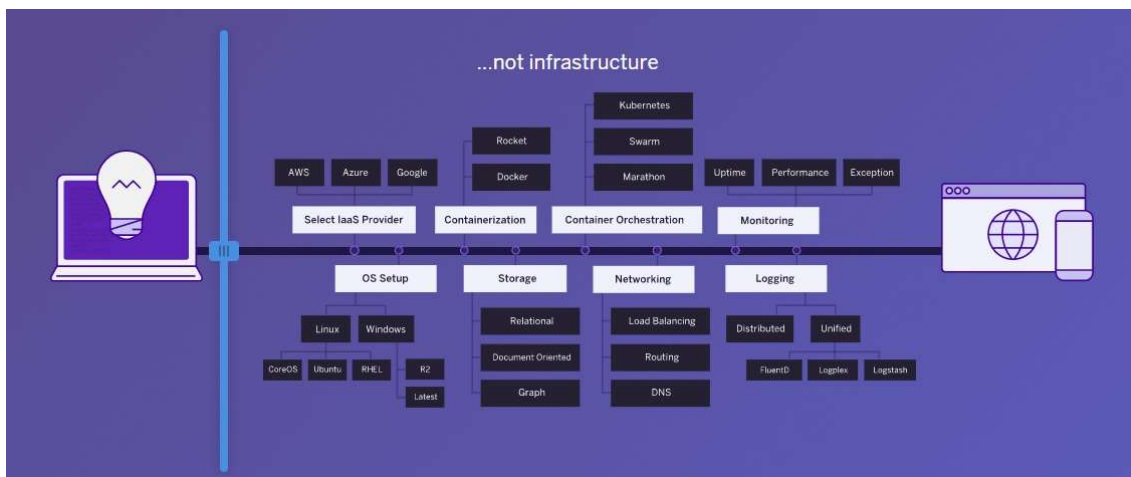


Figura 35 – Topologia da Infraestrutura de Rede



## **9 IMPLANTAÇÃO**

O *Internshipin* será implementado através do domínio .com.br, que poderá ser acessado em todos os dispositivos e navegadores.

Utilizamos o Registro BR para realizar a disponibilidade da plataforma, por um valor de R\$40,00 por ano.

### **9.1. TESTE UNITÁRIO**

Foi desenvolvido teste unitário nos métodos, funcionalidades e módulos do projeto. Para resolver os problemas encontrados o quanto antes e também facilidade em uma possível mudança de unidade.

### **9.2. TESTE INTEGRADO**

O teste integrado foi realizado a fim de testar a combinação das unidades em conjunto. Para assegurar o desempenho, confiabilidade, além de toda a sua funcionalidade. E ajudam a garantir que o trabalho do desenvolvedor não está afetando o trabalho de outro.

## 10 CONCLUSÃO

Durante o desenvolvimento deste projeto nos deparamos com as diversas dificuldades de inserir um aluno de *TI* no mercado de trabalho e construímos esta plataforma, através da utilização de metodologias ativas como o *Design Thinking*, com a capacidade de simular um ambiente do mercado de trabalho para os estudantes dentro de uma instituição de ensino superior.

Ao simular a estrutura do mercado de trabalho provendo experiência prática aos alunos esperamos assim facilitar o seu ingresso no mercado de trabalho, tendo assim a esperança de que no futuro projetos desenvolvidos dentro da plataforma *INTERNSHIPIN* possam ser considerados atividades extra curriculares ou até mesmo como uma forma de estágio.

## REFERÊNCIAS BIBLIOGRÁFICAS

BALDUINO, Plínio. Dominando **JavaScript** com **jQuery**. 1. Ed. Casa do Código, 2012. 192.

BELEM, Thiago. **O que é o MVC**. 2013. Disponível no site: <http://blog.thiagobelem.net/o-que-e-o-mvc>. Acessado em: 22 Mai 2017.

BERTONI, Nilson. **O que é JavaScript, para que serve, onde usar?** 2015. Disponível no site: <http://nilsonbertoni.blogspot.com.br/2015/03/oque-e-javascript-para-que-serve-onde.html>. Acessado em: 22 Mar 2017.

BRANDO, Carlos. **A Filosofia do Ruby**. 2010. Disponível no site: <http://nomedojogo.com/2010/01/20/a-filosofia-do-ruby/>. Acessado em: 07 Mai 2017.

BROWN, Tim. **Design Thinking: Uma metodologia poderosa para decretar o fim das velhas ideias**. 2. ed. Elsevier, 2010. 249.

CASTRO, Flávia Lages et al. **Manual para Elaboração de Trabalhos Acadêmicos. UniFOA - Volta Redonda/ RJ. 2ª Edição**, 2009. Disponível no Site: [http://www.unifoa.edu.br/microsigadocumentos/PortariaReitoria/manual\\_tcc\\_2edicao.pdf](http://www.unifoa.edu.br/microsigadocumentos/PortariaReitoria/manual_tcc_2edicao.pdf). Acessado em: 05 Fev 2017.

CHOWDHURY, Shahriar Iqbal. **Why(s) & How(s) of Asp.Net MVC Part 1**. Disponível no site: <https://www.codeproject.com/KB/aspnet/552846/mvc-10.png>. Acessado em: 22 Mai 2017.

CUSSÓ, Anna. **Ruby é a linguagem da moda, descubra por quê**. 2016. Disponível no site: <http://blog.prolancer.com.br/ruby-linguagem-moda-descubra-por-que/>. Acessado em: 07 Mai 2017.

DACAL, Vinicius. **O básico do básico do Git e do Github**. 2015. Disponível no site: <http://www.viniciusdocal.com/pt/iniciante/2015/01/29/o-basico-do-basico-do-git-e-do-github.html>. Acessado em: 31 Mai 2017.

EIS, Diego. **O básico: O que é HTML?**. Disponível no site: <https://tableless.com.br/o-que-html-basico/>. Acessado em: 15 Mar 2017.

ELIAS, Diego. **Instalação Ruby on Rails Windows 7**. Disponível no site: <http://rubyonrailsbr.blogspot.com.br/2013/07/instalacao-ruby-on-rails-windows-7.html>. Acessado em: 24 Mai 2017.

FARIA, Leonardo. **O que é Ruby on Rails**. 2009. Disponível no site: <http://leonardofaria.net/2009/11/09/o-que-e-ruby-on-rails/>. Acessado em: 22 Mai 2017.

GENTLE, Anne. **Learning the Vocabulary of GitHub for Docs**. Disponível no site: <https://justwriteclick.com/2016/11/12/learning-the-vocabulary-of-github-for-docs/>. Acessado em: 31 Mai 2017.

GIBBONS, Sarah. **Design Thinking 101**. 2016. Disponível no site: <https://www.nngroup.com/articles/design-thinking/>. Acessado em: 15 Fev 2017.

GUEDES, Gilleanes T. A. **UML 2 – Uma abordagem prática**. 2009. Disponível no site: <https://s3.novatec.com.br/capitulos/capitulo-9788575221938.pdf>. Acessado em: 24 Mai 2017.

LENNON, Ackeley. **Git: versionamento e produtividade sem medo**. Disponível no site: <http://blogdoscursos.com.br/git-versionamento-e-produtividade-sem-medo/>. Acessado em: 31 Mai 2017.

LOGVINENKO, Dmitri. **Sails.js on Heroku**. Disponível no site: <https://vort3x.me/sailsjs-heroku/>. Acessado em: 24 Mai 2017.

MARTÍNEZ, Manuel. **Twitter Bootstrap: ¿Qué hicimos antes sin tí?**. Disponível no site: <http://www.freel.es/es/twitter-bootstrap-diseno-web>. Acessado em: 24 Mai 2017.

MARTINS, Alberto. **Instalando Ruby on Rails no seu Cloud Ubuntu**. Disponível no site: <http://blog.infolink.com.br/instalando-ruby-on-rails-no-seu-cloud-ubuntu/>. Acessado em: 24 Mai 2017.

MATOS, Thierry Rene. **Introdução ao Bootstrap para iniciantes**. 2015. Disponível no site: <http://websocialdev.com/introducao-ao-bootstrap-para-iniciantes-em-desenvolvimento/>. Acessado em: 22 Mai 2017.

MAZZA, Lucas. **HTML5 e CSS3: Domine a web do futuro**. 1. ed. Casa do Código, 2012. 217.

MILANI, André. **PostgreSQL – Guia do Programador**. 1. ed. Novatec, 2008. 392.

PESSOA, Fabio. **O que é CSS e qual sua finalidade**. 2016. Disponível no site: <http://fabiopessoa.com.br/css/o-que-e-css-e-qual-sua-finalidade/>. Acessado em: 15 Mar 2017.

PMBOK. **Guia PMBOK 5ª Edição**. 2013. Disponível no site: <https://onedrive.live.com/?authkey=%21ACKA-->

[2kWKOP0QI&cid=F35DDEC7817F7405&id=F35DDEC7817F7405%21111&parId=root&o=OneUp](https://www.cloudacademy.com/blog/what-is-heroku/). Acessado em: 24 Mai 2017.

POOJARY, Nitheesh. **What is Heroku: getting started with PaaS development**. 2015. Disponível no site: <https://cloudacademy.com/blog/what-is-heroku/>. Acessado em: 31 Mai 2017.

SABINO, Gabriela. **O que é uma EAP?**. 2015. Disponível no site: <http://www.radardeprojetos.com.br/2015/03/o-que-e-uma-eap.html>. Acessado em: 24 Mai 2017.

SAWYER, Maggie. **A Complete Guide about the Advantages of HTML5 and CSS3**. 2013. Disponível no site: <http://www.webdevelopmenthelp.net/2015/09/advantages-of-html5-and-css3.html>. Acessado em: 24 Mai 2017.

SÉ, José. **A importância da tecnologia na educação**. 2013. Disponível no site: <https://www.tiespecialistas.com.br/2013/04/a-importancia-da-tecnologia-na-educacao>. Acessado em: 18 Mar 2017.

SILVEIRA, Paulo. **Começar a programar é com JavaScript**. Disponível no site: <http://blog.caelum.com.br/comecar-a-programar-e-com-javascript/>. Acessado em: 24 Mai 2017.

SIQUEIRA Filho, Venício *et al.*. **Manuais de Orientação do Projeto Integrado**. UniFOA – Volta Redonda/RJ. Disponível em: <http://portal.unifoa.edu.br/PortalSagres/Modules/Portal/Services /inicial.geral.srvx>> Acesso em: 15 Fev. 2017.

TEIXEIRA, Fabricio. **O que é Responsive Web Design**. 2011. Disponível no site: <https://brasil.uxdesign.cc/o-que-%C3%A9-responsive-web-design-ab292eb616b7>. Acessado em: 22 Mai 2017.

THOMAS, Peter. **1 Minute PostgreSQL Config Optimization: Make Sure Your Hardware Is Working For You**. Disponível no site: <http://www.tivix.com/blog/postgresql-no-effort-config-optimization/>. Acessado em: 24 Mai 2017.

## **ANEXOS**

### **ANEXO I**

#### **Relação das atribuições dos Orientadores e Coorientadores do TCC**

##### **Prof. Esp. Adilson Gustavo do Espirito Santo**

Professor responsável pelos itens de Gerenciamento de Projeto, Estudo de Mercado e Estudo Econômico.

##### **Prof. Me. Antônio Carlos da Silva**

Professor responsável pelos itens de Estudo de Arquitetura e Estudo de Segurança do Sistema.

##### **Prof. Dr. Carlos Eduardo Costa Vieira**

Professor responsável por orientações aos TCCs e auxiliando na produção dos artigos.

##### **Prof. Me. Rosenclever Lopes Gazoni**

Professor responsável pelo estudo do Banco de Dados do Sistema e Programação.

##### **Prof. Me. Venício Siqueira Filho**

Professor responsável pela disciplina de Projeto Integrado, acompanha e orienta o desenvolvimento do TCC como um todo, orientando as equipes nos diversos produtos que devem ser entregues, responsável pela Análise e Projeto de Sistemas.

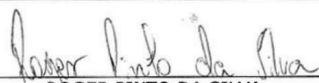
# APÊNDICES

## APENCIDE I

### Atas de encontro com os Professores Orientadores

#### Ata de Reunião

REUNIÃO DO COMITÊ DE ORIENTAÇÃO DO TCC – PROJETO INTEGRADO		
Curso de Sistema de Informação		
<b>OBJETIVO:</b> Análise dos conteúdos e itens que constam do Projeto Integrado, visando tornar o TCC dentro da realidade que o mercado de trabalho de desenvolvimento de sistemas exige.		
<b>ASSUNTOS A SEREM TRATADOS:</b> • Correção de itens na documentação.		
<b>Data:</b> 11/08/2017	<b>Horário:</b> 20:30 ~ 21:00	<b>Local:</b> Centro Universitário Fundação Oswaldo Aranha
PARTICIPANTES		
Nome	Função	
ROGER PINTO DA SILVA	Integrante do projeto	
CARLOS EDUARDO SOUZA VIEIRA	Professor do Comitê de Orientação	
ASSUNTOS TRATADOS		
✓ Sanar dúvidas e questões em relação a documentação necessária para entrega final do Trabalho de Conclusão de Curso.		
PENDÊNCIAS		
COMPROMISSOS EM ABERTO NO ÚLTIMO PERÍODO	PRAZO PARA CONCLUSÃO	RESPONSÁVEL
Alterações definidas pelo Professor Carlos Eduardo.	18/08/2017	Alunos integrantes do projeto.
<b>PRÓXIMA REUNIÃO</b> 18 / 08 / 2017		



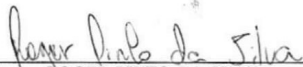
ROGER PINTO DA SILVA  
Integrante do projeto




CARLOS EDUARDO SOUZA VIEIRA  
Professor do Comitê de Orientação

## Ata de Reunião

REUNIÃO DO COMITÊ DE ORIENTAÇÃO DO TCC – PROJETO INTEGRADO		
Curso de Sistema de Informação		
<b>OBJETIVO:</b> Análise dos conteúdos e itens que constam do Projeto Integrado, visando tornar o TCC dentro da realidade que o mercado de trabalho de desenvolvimento de sistemas exige.		
<b>ASSUNTOS A SEREM TRATADOS:</b> <ul style="list-style-type: none"><li>Correção de itens na documentação.</li></ul>		
<b>Data:</b> 18/08/2017	<b>Horário:</b> 20:30 ~ 21:00	<b>Local:</b> Centro Universitário Fundação Oswaldo Aranha
PARTICIPANTES		
Nome	Função	
ROGER PINTO DA SILVA	Integrante do projeto	
CARLOS EDUARDO SOUZA VIEIRA	Professor do Comitê de Orientação	
ASSUNTOS TRATADOS		
✓ Sanar dúvidas e questões em relação a documentação necessária para entrega final do Trabalho de Conclusão de Curso.		
PENDÊNCIAS		
COMPROMISSOS EM ABERTO NO ÚLTIMO PERÍODO	PRAZO PARA CONCLUSÃO	RESPONSÁVEL
Alterações definidas pelo Professor Carlos Eduardo.	18/08/2017	Alunos integrantes do projeto.
PRÓXIMA REUNIÃO		
25 / 08 / 2017		

  
\_\_\_\_\_  
ROGER PINTO DA SILVA  
Integrante do projeto

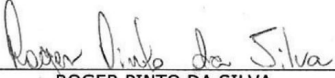
  
\_\_\_\_\_  
CARLOS EDUARDO SOUZA VIEIRA  
Professor do Comitê de Orientação


## Ata de Reunião

REUNIÃO DO COMITÊ DE ORIENTAÇÃO DO TCC – PROJETO INTEGRADO		
Curso de Sistema de Informação		
<b>OBJETIVO:</b> Análise dos conteúdos e itens que constam do Projeto Integrado, visando tornar o TCC dentro da realidade que o mercado de trabalho de desenvolvimento de sistemas exige.		
<b>ASSUNTOS A SEREM TRATADOS:</b>		
• Correção de itens na documentação.		
<b>Data:</b> 25/08/2017	<b>Horário:</b> 20:30 ~ 21:00	<b>Local:</b> Centro Universitário Fundação Oswaldo Aranha
PARTICIPANTES		
Nome	Função	
ROGER PINTO DA SILVA	Integrante do projeto	
CARLOS EDUARDO SOUZA VIEIRA	Professor do Comitê de Orientação	

ASSUNTOS TRATADOS
✓ Sanar dúvidas e questões em relação a documentação necessária para entrega final do Trabalho de Conclusão de Curso.

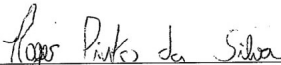
PENDÊNCIAS		
COMPROMISSOS EM ABERTO NO ÚLTIMO PERÍODO	PRAZO PARA CONCLUSÃO	RESPONSÁVEL
Alterações definidas pelo Professor Carlos Eduardo.	25/08/2017	Alunos integrantes do projeto.

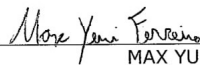
  
\_\_\_\_\_  
ROGER PINTO DA SILVA  
Integrante do projeto

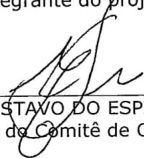
  
\_\_\_\_\_  
CARLOS EDUARDO SOUZA VIEIRA  
Professor do Comitê de Orientação

## Ata de Reunião

REUNIÃO DO COMITÊ DE ORIENTAÇÃO DO TCC - PROJETO INTEGRADO		
Curso de Sistema de Informação		
<b>OBJETIVO:</b> Análise dos conteúdos e itens que constam do Projeto Integrado, visando tornar o TCC dentro da realidade que o mercado de trabalho de desenvolvimento de sistemas exige.		
<b>ASSUNTOS A SEREM TRATADOS:</b>		
• Mudanças Cronograma		
<b>Data:</b> 24/08/2017	<b>Horário:</b> 20:30 ~ 21:00	<b>Local:</b> Centro Universitário Fundação Oswaldo Aranha
<b>PARTICIPANTES</b>		
<b>Nome</b>	<b>Função</b>	
MAX YURI FERREIRA	Integrante do projeto	
ROGER PINTO DA SILVA	Integrante do projeto	
ADILSON GUSTAVO DO ESPÍRITO SANTO	Professor do Comitê de Orientação	
<b>ASSUNTOS TRATADOS</b>		
✓ Sanar dúvidas e questões em relação ao Cronograma.		
<b>PENDÊNCIAS</b>		
<b>COMPROMISSOS EM ABERTO NO ÚLTIMO PERÍODO</b>	<b>PRAZO PARA CONCLUSÃO</b>	<b>RESPONSÁVEL</b>
Apresentar Cronograma conforme mudanças solicitadas pelo Professor Adilson.	31/08/2017	Alunos integrantes do projeto.
<b>PRÓXIMA REUNIÃO</b> 31 / 08 / 2017		

  
ROGER PINTO DA SILVA  
Integrante do projeto

  
MAX YURI FERREIRA  
Integrante do projeto

  
ADILSON GUSTAVO DO ESPÍRITO SANTO  
Professor do Comitê de Orientação

## Ata de Reunião

REUNIÃO DO COMITÊ DE ORIENTAÇÃO DO TCC - PROJETO INTEGRADO		
<b>Curso de Sistema de Informação</b>		
<b>OBJETIVO:</b>		
Análise dos conteúdos e itens que constam do Projeto Integrado, visando tornar o TCC dentro da realidade que o mercado de trabalho de desenvolvimento de sistemas exige.		
<b>ASSUNTOS A SEREM TRATADOS:</b>		
<ul style="list-style-type: none"> <li>• Linhas de Base e Cronograma</li> </ul>		
<b>Data:</b>	<b>Horário:</b>	<b>Local:</b>
12/09/2017	20:30 ~ 21:00	Centro Universitário Fundação Oswaldo Aranha
PARTICIPANTES		
Nome	Função	
MAX YURI FERREIRA	Integrante do projeto	
ROGER PINTO DA SILVA	Integrante do projeto	
ROMULO DE LACERDA CORREA RIMULO	Integrante do projeto	
IGOR MARÇAL BARBOSA DOS SANTOS	Integrante do projeto	
ADILSON GUSTAVO DO ESPÍRITO SANTO	Professor do Comitê de Orientação	
ASSUNTOS TRATADOS		
✓ Sanar dúvidas e questões em relação ao Cronograma.		
PENDÊNCIAS		
COMPROMISSOS EM ABERTO NO ÚLTIMO PERÍODO	PRAZO PARA CONCLUSÃO	RESPONSÁVEL
Elaboração do Cronograma com alterações definidas pelo Professor Adilson.	14/09/2017	Alunos integrantes do projeto.
PRÓXIMA REUNIÃO		
14 / 09 / 2017		

*Roger Pinto da Silva*

ROGER PINTO DA SILVA  
Integrante do projeto

*Max Yuri Ferreira*

MAX YURI FERREIRA  
Integrante do projeto

*Rômulo de Lacerda Correa Rimulo*

ROMULO DE LACERDA CORREA RIMULO  
Integrante do projeto


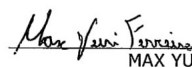


*Igor Marçal*

IGOR MARÇAL BARBOSA DOS SANTOS  
Integrante do projeto

*Adilson Gustavo do Espírito Santo*  
ADILSON GUSTAVO DO ESPÍRITO SANTO  
Professor do Comitê de Orientação

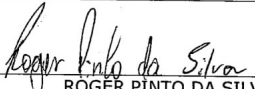
## Ata de Reunião

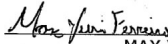
REUNIÃO DO COMITÊ DE ORIENTAÇÃO DO TCC – PROJETO INTEGRADO		
<b>Curso de Sistema de Informação</b>		
<b>OBJETIVO:</b>		
Análise dos conteúdos e itens que constam do Projeto Integrado, visando tornar o TCC dentro da realidade que o mercado de trabalho de desenvolvimento de sistemas exige.		
<b>ASSUNTOS A SEREM TRATADOS:</b>		
<ul style="list-style-type: none"> <li>• Confeção da Matriz SWOT</li> </ul>		
<b>Data:</b>	<b>Horário:</b>	<b>Local:</b>
16/03/2017	20:30 ~ 21:00	Centro Universitário Fundação Oswaldo Aranha
PARTICIPANTES		
Nome	Função	
IGOR MARÇAL BARBOSA DOS SANTOS	Integrante do projeto	
MAX YURI FERREIRA	Integrante do projeto	
ROGER PINTO DA SILVA	Integrante do projeto	
ADILSON GUSTAVO DO ESPÍRITO SANTO	Professor do Comitê de Orientação	
ASSUNTOS TRATADOS		
✓ Sanar dúvidas e questões em relação à Matriz SWOT.		
PENDÊNCIAS		
COMPROMISSOS EM ABERTO NO ÚLTIMO PERÍODO	PRAZO PARA CONCLUSÃO	RESPONSÁVEL
Elaboração da Matriz SWOT com alterações definidas pelo Professor Adilson.	16/03/2017	Alunos integrantes do projeto.
PRÓXIMA REUNIÃO		
23 / 03 / 2017		

 _____ ROGER PINTO DA SILVA Integrante do projeto	 _____ MAX YURI FERREIRA Integrante do projeto
 _____ ADILSON GUSTAVO DO ESPÍRITO SANTO Professor do Comitê de Orientação	 _____ IGOR MARÇAL BARBOSA DOS SANTOS Integrante do projeto

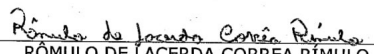
**Ata de Reunião**

<b>REUNIÃO DO COMITÊ DE ORIENTAÇÃO DO TCC - PROJETO INTEGRADO</b>		
<b>Curso de Sistema de Informação</b>		
<b>OBJETIVO:</b>		
Análise dos conteúdos e itens que constam do Projeto Integrado, visando tornar o TCC dentro da realidade que o mercado de trabalho de desenvolvimento de sistemas exige.		
<b>ASSUNTOS A SEREM TRATADOS:</b>		
<ul style="list-style-type: none"> <li>• Confeção da Matriz SWOT</li> </ul>		
<b>Data:</b>	<b>Horário:</b>	<b>Local:</b>
23/03/2017	20:30 ~ 21:00	Centro Universitário Fundação Oswaldo Aranha
<b>PARTICIPANTES</b>		
Nome	Função	
MAX YURI FERREIRA	Integrante do projeto	
ROGER PINTO DA SILVA	Integrante do projeto	
RÔMULO DE LACERDA CORREA RÍMULO	Integrante do projeto	
ADILSON GUSTAVO DO ESPÍRITO SANTO	Professor do Comitê de Orientação	
<b>ASSUNTOS TRATADOS</b>		
<ul style="list-style-type: none"> <li>✓ Sanar dúvidas e questões em relação à Estrutura Analítica de Projeto (EAP).</li> </ul>		
<b>PENDÊNCIAS</b>		
COMPROMISSOS EM ABERTO NO ÚLTIMO PERÍODO	PRAZO PARA CONCLUSÃO	RESPONSÁVEL
Elaboração da Matriz SWOT com alterações definidas pelo Professor Adilson.	16/03/2017	Alunos integrantes do projeto.

  
 \_\_\_\_\_  
 ROGER PINTO DA SILVA  
 Integrante do projeto

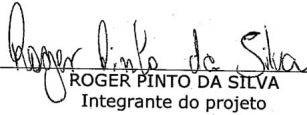
  
 \_\_\_\_\_  
 MAX YURI FERREIRA  
 Integrante do projeto


  
 \_\_\_\_\_  
 ADILSON GUSTAVO DO ESPÍRITO SANTO  
 Professor do Comitê de Orientação

  
 \_\_\_\_\_  
 RÔMULO DE LACERDA CORREA RÍMULO  
 Integrante do projeto

## Ata de Reunião

REUNIAO DO COMITÊ DE ORIENTAÇÃO DO TCC - PROJETO INTEGRADO		
Curso de Sistema de Informação		
<b>OBJETIVO:</b> Análise dos conteúdos e itens que constam do Projeto Integrado, visando tornar o TCC dentro da realidade que o mercado de trabalho de desenvolvimento de sistemas exige.		
<b>ASSUNTOS A SEREM TRATADOS:</b> • Confeção da Matriz SWOT		
<b>Data:</b> 18/09/2017	<b>Horário:</b> 20:30 ~ 21:00	<b>Local:</b> Centro Universitário Fundação Oswaldo Aranha
PARTICIPANTES		
Nome	Função	
ROGER PINTO DA SILVA	Integrante do projeto	
ROSENCLEVER LOPES GAZONI	Professor do Comitê de Orientação	
ASSUNTOS TRATADOS		
✓ Sanar dúvidas e questões em relação aos diagramas solicitados.		
PENDÊNCIAS		
COMPROMISSOS EM ABERTO NO ÚLTIMO PERÍODO	PRAZO PARA CONCLUSÃO	RESPONSÁVEL
Correção dos diagramas para a próxima semana.	20/09/2017	Aluno integrante do projeto.
<b>PRÓXIMA REUNIÃO</b> 20 / 09 / 2017		

  
ROGER PINTO DA SILVA  
Integrante do projeto

  
ROSENCLEVER LOPES GAZONI

**Script do banco de dados:**

```
CREATE TABLE users(
user_id SERIAL PRIMARY KEY,
matricula VARCHAR(255) NOT NULL,
nome VARCHAR(255) NOT NULL,
email VARCHAR(255) NOT NULL,
cpf VARCHAR(255) NOT NULL,
avatar_file_name VARCHAR(255),
avatar_content_type VARCHAR(255),
avatar_file_size INTEGER,
avatar_updated_at TIMESTAMP,
tipo INTEGER NOT NULL,
encrypted_password VARCHAR(255) NOT NULL
);
```

```
CREATE TABLE topics(
topic_id SERIAL PRIMARY KEY,
titulo VARCHAR(255) NOT NULL,
descricao TEXT NOT NULL,
status INTEGER NOT NULL,
tipo INTEGER NOT NULL,
category_id INTEGER NOT NULL REFERENCES categories (category_id),
subcategory_id INTEGER NOT NULL REFERENCES subcategories
(subcategory_id),
user_id INTEGER NOT NULL REFERENCES users (user_id),
created_at TIMESTAMP,
updated_at TIMESTAMP
);
```

```
CREATE TABLE categories(
category_id SERIAL PRIMARY KEY,
nome VARCHAR(255) NOT NULL
created_at TIMESTAMP,
updated_at TIMESTAMP
);
```

```
CREATE TABLE subcategories(
subcategory_id SERIAL PRIMARY KEY,
category_id INTEGER NOT NULL REFERENCES categories (category_id),
nome VARCHAR(255) NOT NULL,
created_at TIMESTAMP,
updated_at TIMESTAMP
);
```

```
CREATE TABLE answers(
answer_id SERIAL PRIMARY KEY,
topic_id INTEGER REFERENCES topics (topic_id),
```

```
user_id INTEGER NOT NULL REFERENCES users (user_id),
answer_id INTEGER REFERENCES users (answer_id),
descricao TEXT NOT NULL,
created_at TIMESTAMP,
updated_at TIMESTAMP
```

```
);
```

```
CREATE TABLE projects(
project_id SERIAL PRIMARY KEY,
max_users INTEGER,
titulo VARCHAR(255) NOT NULL,
descricao TEXT NOT NULL,
total_hours INTEGER NOT NULL,
user_id INTEGER NOT NULL REFERENCES users (user_id),
created_at TIMESTAMP,
updated_at TIMESTAMP,
status INTEGER NOT NULL,
datainicio TIMESTAMP NOT NULL,
datafim TIMESTAMP NOT NULL,
documentation_file_name VARCHAR(255),
documentation_content_type VARCHAR(255),
documentation_file_size INTEGER,
documentation_updated_at TIMESTAMP
);
```

```
CREATE TABLE registrations(
registration_id SERIAL PRIMARY KEY,
project_id INTEGER NOT NULL REFERENCES projects (project_id)
user_id INTEGER NOT NULL REFERENCES users (user_id)
);
```

APENDICE II

<b>Total de Pontos de Funções não Ajustados</b>				
<b>Tipo de Função</b>	<b>Complexidade Funcional</b>	<b>Pontos de Função</b>	<b>Total por Complexidade</b>	<b>Total por Função</b>
			<b>Serviço Proposto</b>	<b>Total</b>
<b>Arquivo Lógico Interno</b>	<b>Simple</b>	<b>7</b>	<b>49 (7)</b>	<b>49</b>
	<b>Média</b>	<b>10</b>	<b>0</b>	
	<b>Complexa</b>	<b>15</b>	<b>0</b>	
<b>Arquivo de Interface Externa</b>	<b>Simple</b>	<b>7</b>	<b>0</b>	<b>0</b>
	<b>Média</b>	<b>10</b>	<b>0</b>	
	<b>Complexa</b>	<b>15</b>	<b>0</b>	
<b>Entrada Externa</b>	<b>Simple</b>	<b>3</b>	<b>36 (12)</b>	<b>36</b>
	<b>Média</b>	<b>4</b>	<b>0</b>	
	<b>Complexa</b>	<b>6</b>	<b>0</b>	
<b>Saída Externa</b>	<b>Simple</b>	<b>4</b>	<b>8 (2)</b>	<b>35</b>
	<b>Média</b>	<b>5</b>	<b>20 (4)</b>	
	<b>Complexa</b>	<b>7</b>	<b>7 (1)</b>	
<b>Consulta Externa</b>	<b>Simple</b>	<b>3</b>	<b>9 (3)</b>	<b>21</b>
	<b>Média</b>	<b>4</b>	<b>12 (3)</b>	
	<b>Complexa</b>	<b>6</b>	<b>0</b>	
<b>Total de Pontos de Funções não Ajustados</b>				<b>141</b>
<b>Fator de Ajuste</b>				<b>1,04</b>
<b>Total de Pontos de Funções Ajustados</b>				<b>146,64</b>

Características Gerais dos Sistemas	Nível de Influência
	NI
<b>1 – Comunicação dos dados</b>	<b>5</b>
<b>2 – Processamento distribuído</b>	<b>3</b>
<b>3 – Desempenho</b>	<b>1</b>
<b>4 – Utilização de Equipamento</b>	<b>0</b>
<b>5 – Volume de Transações</b>	<b>3</b>
<b>6 – Entrada de dados “on-line”</b>	<b>5</b>
<b>7 – Eficiência do usuário final</b>	<b>4</b>
<b>8 – Atualização “on-line”</b>	<b>4</b>
<b>9 – Processamento complexo</b>	<b>0</b>
<b>10 – Reutilização de código</b>	<b>1</b>
<b>11 – Facilidade de implantação</b>	<b>3</b>
<b>12 – Facilidade operacional</b>	<b>5</b>
<b>13 – Múltiplos locais</b>	<b>0</b>
<b>14 – Facilidade de mudanças</b>	<b>5</b>
<b>Nível de Influência (NI)</b>	<b>39</b>
<b>Fator de Ajuste = (NI * 0,01) + 0,65</b>	<b>1,04</b>