Documento de Arquitetura

Versão 1.0

Preparado por Thaís Tavares de Amorim

19/10/2017

Tabela de Conteúdo

[1. Introdução 1](#_Toc494397737)

[2. Escopo do Projeto 1](#_Toc494397738)

[3. Organização do Projeto 2](#_Toc494397739)

[4. Processo e medidas de desenvolvimento 2](#_Toc494397740)

[5. Marcos do Projeto e Objetivos 2](#_Toc494397741)

[6. Desenvolvimento / Implementação 2](#_Toc494397742)

[7. Lições Aprendidas 2](#_Toc494397743)

[Aprovações 3](#_Toc494397744)

|  |  |  |
| --- | --- | --- |
| ACADEMICCI - Gestão de Monitoria | | |
| Plano de Projeto  *Project Plan* | | |
| Preparado por | Thaís Tavares de Amorim | Versão 1 |
| Aprovado por | Alexandre Rangel P Ribeiro | 19/10/2017 |
| Aprovador por | Wanderson Inácio dos Santos | 19/10/2017 |

# Introdução

O sistema ACADEMICCI tem como finalidade gerir as atividades de monitoria das Instituições de Ensino Superior em seu processo de inscrição, desenvolvimento de monitória e certificação, bem como informações de controle de monitória.

## Finalidade

Este documento oferece uma visão geral arquitetural abrangente do sistema, usando diversas visões arquiteturais para representar diferentes aspectos do sistema. O objetivo deste documento é capturar e comunicar as decisões arquiteturais significativas que foram tomadas em relação ao sistema.

## Escopo

ACADEMICCI é um sistema para gerenciamento de monitorias realizadas atualmente pelos próprios alunos, seja de forma presencial ou a distância. Este sistema advém da necessidade de controle e regulamentação do processo de seleção, acompanhamento e emissão de resultados das monitorias que poderão levar a certificação. ACADEMICCI visa melhor o gerenciamento de modo a automatizar as funcionalidades de seleção e certificação dos monitores, no sistema terá possibilidade de interagir de forma real e direta com seu Aluno-Orientador ou com seu Professor para tirar dúvidas constantes com relação a disciplina em questão. O sistema disponibilizará controle de horas de acesso para que o Aluno-Orientador possa receber o certificado ao final do semestre, se o mesmo atingir a quantidade de horas definida em edital de monitória.

## Definições, Acrônimos e Abreviações

Utilizar ao final do documento

## Visão Geral

São apresentados ainda neste documento diferentes visões arquiteturais de como o sistema deve se comportar em diferentes processos, como deve ser implantado e implementado e restrições de desempenho e qualidade.

# Representação Arquitetural

* Visão de Caso de Uso
  + Apresenta as funcionalidades arquiteturais importantes e os usuários do sistema.
* Visão Lógica
  + Descreve classes e sua organização e apresenta o padrão de arquitetura que deverá ser utilizado para desenvolvimento do sistema.
* Visão de Processos
  + Mostra o padrão de comportamento do sistema diante de diferentes ações do usuário.
* Visão de Implantação
  + Descreve a estrutura do ambiente onde o software será instalado.
* Visão de Implementação
  + Ilustra a distribuição do processo em um conjunto de nós do sistema, incluindo a distribuição física de processos e threads.

# Metas e Restrições da Arquitetura

Existem algumas restrições de requisito e de sistema principais que têm relação significativa com a arquitetura, sendo elas:

* O sistema deverá ser Web;
* Estrutura MVC;

# Visão de Casos de Uso

O caso de uso do sistema ACADEMICCI serão listados abaixo:

* Realizar Inscrição
* Realizar Login
* Manter Fórum
* Manter Mensagens
* Manter monitória

## Realizações de Casos de Uso

Disponibilizar Caso de Uso

# Visão Lógica

## Visão Geral

A visão lógica define a estrutura da arquitetura. Abaixo será especificado o padrão utilizado para o desenvolvimento do sistema, no caso, MVC.

Disponibilizar Diagrama de Classes (Do modelo MVC)

Onde:

* *View:* Pacote que contém as visões do projeto, ou seja, as interfaces, formulários et al;
* *Controller:* Pacote que recebe as informações e requisições do pacote *view* e os despacha para devida classe de controle, o pacote *controller* implementa o pacote *BusinessLogic* o qual contém as regras de negócio do sistema;
* *Action:* Pacote que recebe as informações e requisições do pacote *controller* e os atribui às respectivas classes do pacote *bean* e do pacote *persistence;*
* *Bean:* Pacote que recebe e armazena as informações referentes a seus respectivos objetos;
* *Persistence:* Pacote que recebe as requisições e realiza as operações relacionadas ao Banco de Dados, utilizando dados providos do pacote *bean.* O pacote *persistence* também pode enviar dados ao pacote *view;*

## Pacotes de Design Significativos do Ponto de Vista da Arquitetura

Disponibilizar Diagrama de Classes Geral do Sistema

# Visão de Implantação

# Visão de Implementação

# Tamanho e Desempenho

# Qualidade

# Aprovações

|  |  |  |
| --- | --- | --- |
| Título | Nome e Assinatura | Data |
| Product Owner | Thaís Tavares de Amorim | 28/09/2017 |
| Scrum Master | Raphael Guedes | 28/09/2017 |
| Time | Alexandre Rangel P Ribeiro | 28/09/2017 |
| Time | Wanderson Inácio dos Santos | 28/09/2017 |
| Time | Adriana Leticia | 28/09/2017 |
| Time | Claudio Filho | 28/09/2017 |

***Nota: Quaisquer alterações neste documento deverão ser submetidas ao processo de controle de projeto para aprovações antes de serem incorporadas a este documento.***

Histórico de Revisões

|  |  |  |  |
| --- | --- | --- | --- |
| Nome | Data | Razão da Mudança | Versão |
| Thaís Tavares | 19/10/2017 | Criação do documento | 1.0 |