

**Documentação de um**

**Produto de Software**

**EurekaMap**

**Nome dos Alunos:**

| **RA** | **Nome** |
| --- | --- |
| **23.01500-4** | **Julio Yukio Matsumoto Nagawa** |
| **23.01001-0** | **Murilo Molina Barone** |
| **23.00591-2** | **Vinícius Parelho de Oliveira** |
| **23.01603-5** | **Vitor de Oliveira Maia** |
|  |  |

**2024**

**ÍNDICE DETALHADO**

**1.** **Levantamento dos Requisitos do Sistema de Software 3**

1.1. Extração de Requisitos 3

1.2. Análise da Coleta de Requisitos 3

**2.** **Especificação dos Requisitos do Sistema de Software 3**

2.1. Requisitos Funcionais 3

2.2. Requisitos Não-Funcionais 3

**3.** **Análise/Projeto 4**

3.1. Diagrama de Classes 4

3.2. Diagrama de Sequência 4

3.3. Modelo de Banco de Dados 4

3.4. Diagrama de Atividades (opcional) 4

3.5. Diagrama de estados (opcional) 4

**4.** **Implementação 5**

**5.** **Testes 5**

**6.** **Resultados e Considerações 5**

**Apêndice I 5**

**Descrição/Resumo do Projeto**

| A Mauá, deseja um aplicativo mobile para planejar o espaço para as bancadas da Eureka. Além de divulgar esse planejamento pelo aplicativo para os visitantes se guiarem no dia das apresentações com mais facilidade.  O projeto utilizará o kit de desenvolvimento de software, Flutter, que utiliza a linguagem Dart para criar o aplicativo tanto para Android quanto IOS.  O projeto incluirá um sistema de login para administradores planejarem os espaços para as bancadas na Eureka. Visitantes também poderão fazer login para visualizar o planejamento feito pelos organizadores.  O sistema utilizará um banco de dados não relacional salvo na nuvem pelo MongoDB. |
| --- |

1. **Levantamento dos Requisitos do Sistema de Software**

Este capítulo tem como objetivo apresentar o levantamento dos requisitos do Sistema de Software e a forma de extração dos Requisitos.

* 1. **Extração de Requisitos**

Neste item deve ser descrita a extração de requisitos utilizada, por exemplo: Questionário, Entrevista, Brainstorming e em seguida deve ser apresentado o questionário, roteiro de entrevistas ou roteiro brainstorming.

*Disciplina de Apoio: Frameworks for Low Code Development.*

* 1. **Análise da Coleta de Requisitos**

Neste item deve ser apresentado o resultado da extração de requisitos, respostas do questionário de forma sumarizada ou resposta das entrevistas.

*Disciplina de Apoio: Frameworks for Low Code Development.*

1. **Especificação dos Requisitos do Sistema de Software**

Este capítulo tem como objetivo especificar os requisitos funcionais e não funcionais.

Os requisitos especificados devem ter como base a análise da coleta de requisitos, de acordo, com o resultado da técnica de extração de requisitos.

* 1. **Requisitos Funcionais**

Neste item devem ser descritos os requisitos a serem atendidos funcionalmente pelo sistema de uma forma simples, possibilitando a compreensão do comportamento do sistema pela perspectiva do usuário. Utilizar Modelo de Caso de Uso (diagrama e especificações de atores e casos de uso).

*Disciplina de Apoio: Modelagem Orientada a Objetos.*

* 1. **Requisitos Não-Funcionais**

Neste item devem ser apresentados os requisitos não funcionais, que especificam restrições sobre os serviços ou funções providas pelo sistema.

*Disciplina de Apoio: Modelagem Orientada a Objetos.*

1. **Análise/Projeto**
   1. **Diagrama de Classes**

Neste item deve ser apresentado o modelo do domínio, visão de negócio, que representa um primeiro modelo conceitual do diagrama de classes.

O diagrama de classes deve possuir todas as classes identificadas do sistema, deve conter os atributos e métodos de cada classe, e os relacionamentos entre elas.

*Disciplina de Apoio: Modelagem Orientada a Objetos.*

* 1. **Diagrama de Sequência**

Neste item deve ser apresentado os diagramas de sequência com maior valor de negócio ao sistema. A escolha é realizada por caso de uso. Não há necessidade de realizar o diagrama de sequência para <<crud>>.

*Disciplina de Apoio: Modelagem Orientada a Objetos.*

* 1. **Modelo de Banco de Dados**

Neste item deve ser apresentado o modelo lógico relacional de banco de dados como proposta de solução.

Também deve ser representado o link de acesso ou repositório dos scripts físicos de banco de dados.

*Disciplina de Apoio: Banco de Dados Relacional.*

* 1. **Diagrama de Atividades (opcional)**

O diagrama de atividades representa o detalhamento de tarefas e o fluxo de uma atividade para outra de um sistema, geralmente utilizado para os métodos que contém regras de negócio.

*Esse diagrama deverá ser elaborado se houver necessidade e agregar valor ao projeto.*

*Disciplina de Apoio: Modelagem Orientada a Objetos.*

* 1. **Diagrama de estados (opcional)**

O diagrama de estados especifica as sequências de estados pelas quais o objeto pode passar durante seu ciclo de vida em resposta a eventos.

*Disciplina de Apoio: Modelagem Orientada a Objetos.*

1. **Implementação**

Neste item indicar o link de acesso ou repositório de toda implementação de programação orientada a objetos Java e Python (opcional).

*Disciplina de Apoio: Programação Orientada a Objetos, Lógica de Programação e Banco de Dados Relacional.*

1. **Testes**

Neste item indicar o link de acesso ou repositório de todas as evidências de testes unitários realizados no projeto, de acordo, com os casos de uso especificados.

*Disciplina de Apoio: Programação Orientada a Objetos.*

1. **Resultados e Considerações**

Neste item devem ser apresentados os principais “prints” das telas do sistema de software desenvolvido, com uma breve explicação de cada tela e ao final as considerações gerais do projeto, sob o ponto de vista dos requisitos que foram implementados e os resultados obtidos.

1. **Registro da Apresentação ao Parceiro**

Neste item devem ser apresentados os registros firmados com os parceiros do projeto.

**Apêndice I**

Neste item deve ser anexado o roteiro de entrevista ou questionário respondido.

*Disciplina de Apoio: Modelagem Orientada a Objetos.*