**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

**Pós-graduação *Lato Sensu* em Engenharia de *Software***

Trabalho de Conclusão de Curso

GESEP – Gestão de Serviços Públicos

Vinicius Reia Canalles

Belo Horizonte

<Abril/2022

# Trabalho de Conclusão de Curso

**Sumário**

Trabalho de Conclusão de Curso 3

1. Cronograma de trabalho 4

2. Diagrama de casos de uso 5

3. Requisitos não-funcionais 5

4. Protótipo navegável do sistema 5

5. Diagrama de classes de domínio 6

6. Modelo de componentes 7

6.1. Padrão arquitetural 7

6.2. Diagrama de componentes 7

6.3. Descrição dos componentes 7

7. Diagrama de implantação 8

8. Plano de Testes 9

9. Estimativa de pontos de função 10

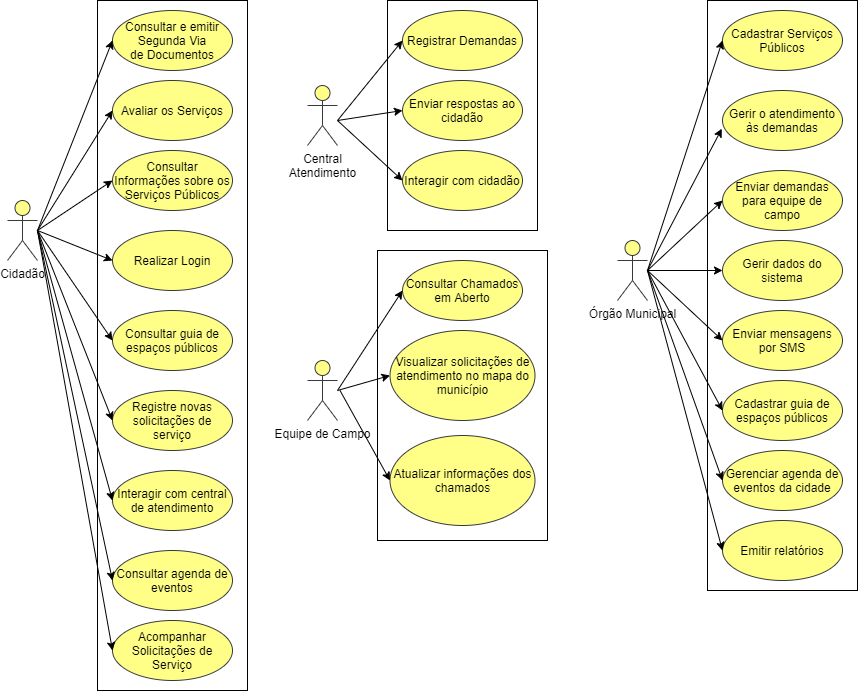
10. Informações da implementação 11

11. Referências 12

## Cronograma de trabalho

|  |  |  |  |
| --- | --- | --- | --- |
| **Datas** | | **Atividade / Tarefa** | **Produto / Resultado** |
| **De** | **Até** |
| 03 / 01 / 22 | 04 / 01 / 22 | 1. Estudo e definição de tecnologia FrontEnd | Utilização de ReactJS |
| 05 / 01 / 22 | 06 / 01 / 22 | 2. Estudo e definição de tecnologia e plataforma BackEnd | Utilização AWS Cloud e linguagem Java, com Spring Boot e MVC |
| 07 / 01 / 22 | 08 / 01 / 22 | 3. Criação do diagrama de casos de uso | Diagrama de casos de uso definido |
| 09 / 01 / 22 | 10 / 01 / 22 | 4. Análise e definição dos requisitos não funcionais | Requisitos não funcionais definidos |
| 11 / 01 / 22 | 12 / 01 / 22 | 5. Análise e definição do diagrama de classes | Diagrama de classe definido |
| 13 / 01 / 22 | 14 / 01 / 22 | 6. Análise e definição do modelo de componentes | Modelo de componentes definido |
| 15 / 01 / 22 | 16 / 01 / 22 | 7. Análise e definição do diagrama de implantação | Diagrama de Implantação definido |
| 17 / 01 / 22 | 18 / 01 / 22 | 8. Análise e definição dos pontos de função | Pontos de função definidos |
| 19 / 01 / 22 | 20 / 01 / 22 | 9. Análise e definição do plano de testes | Plano de testes definido |
| 21 / 01 / 22 | 12 / 02 / 22 | 10. Construção da aplicação FrontEnd | Telas de interação com o usuário criadas |
| 13 / 02 / 22 | 06 / 03 / 22 | 11. Construção da aplicação BackEnd | Criação da API concluída |
| 07 / 03 / 22 | 20 / 03 / 22 | 12. Execução dos cenários de teste definidos no plano de testes | Cenários de teste executados com sucesso |
| 21 / 03 / 22 | 09 / 04 / 22 | 13. Revisão da documentação a ser enviada no template estabelecido | Template preenchido e documentação finalizada |

## Diagrama de casos de uso



## Requisitos não-funcionais

* Uso de Design responsivo nas interfaces gráficas;
* O aplicativo deverá executar em plataforma Web, Android e Iphone;
* O sistema deve garantir a segurança das senhas dos usuários, criptografando-as ao serem inseridas no banco de dados;
* O aplicativo somente poderá ser utilizado por usuários cadastrados e autenticados;
* O aplicativo deve ser construído seguindo o padrão de arquitetura MVC;
* O aplicativo deve possuir alta disponibilidade;
* O aplicativo deverá ser desenvolvido utilizando Html, Css e JavaScript para o FrontEnd e Java com Spring Boot para o BackEnd;

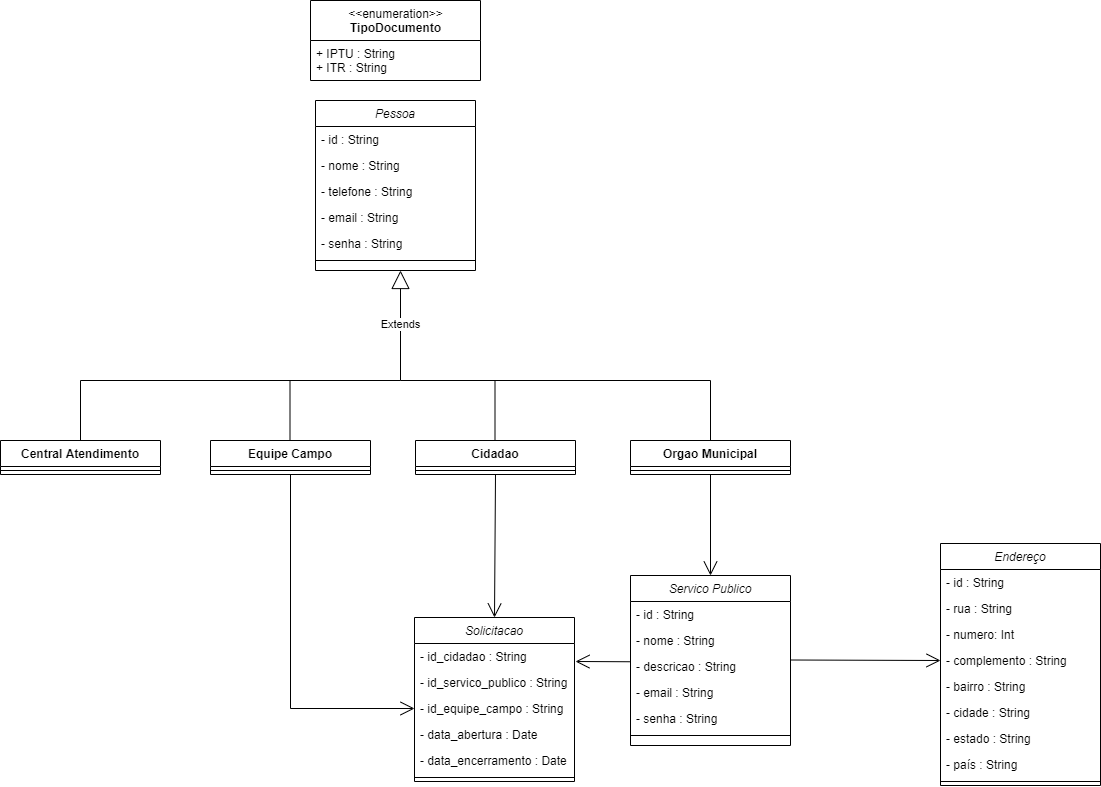
## Protótipo navegável do sistema

<Faça um protótipo navegável do sistema mostrando a tela inicial da aplicação e as de três casos de uso principais. A navegação entre as telas também precisa ser apresentada. Deve-se utilizar alguma ferramenta para a criação dos *wireframes* (como [Figma](http://figma.com) ou [Balsamiq](https://balsamiq.com/wireframes/), por exemplo).

Faça um vídeo de apresentação do protótipo navegável desenvolvido e disponibilize-o de forma que a banca de avaliação do TCC possa visualizá-lo. Esse vídeo deve ter duração de, no máximo, 3 minutos. Utilize, preferencialmente, o formato AVI.

Nesta seção, indique o *link* desse vídeo e do repositório (como o [GitHub](http://github.com), [Bitbucket](https://bitbucket.org/product/), etc) onde seu protótipo navegável está disponível.>

## Diagrama de classes de domínio



## Modelo de componentes

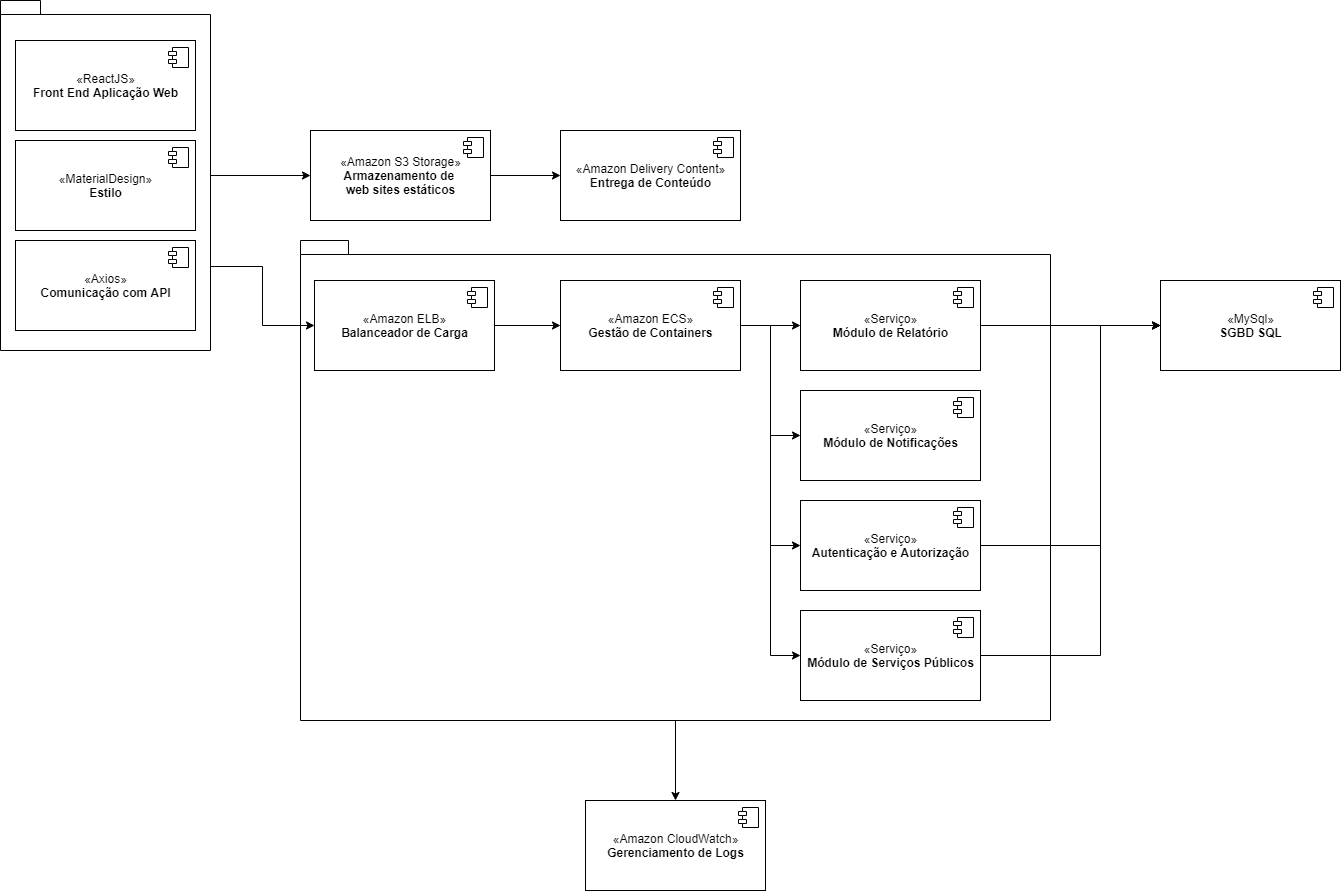
### Padrão arquitetural

- Padrão arquitetural: MVC (Model – View – Controller);

- Tecnologia Front End: Html, CSS e JavaScript com ReactJS;

- Tecnologia Backend: Java com Spring Boot;

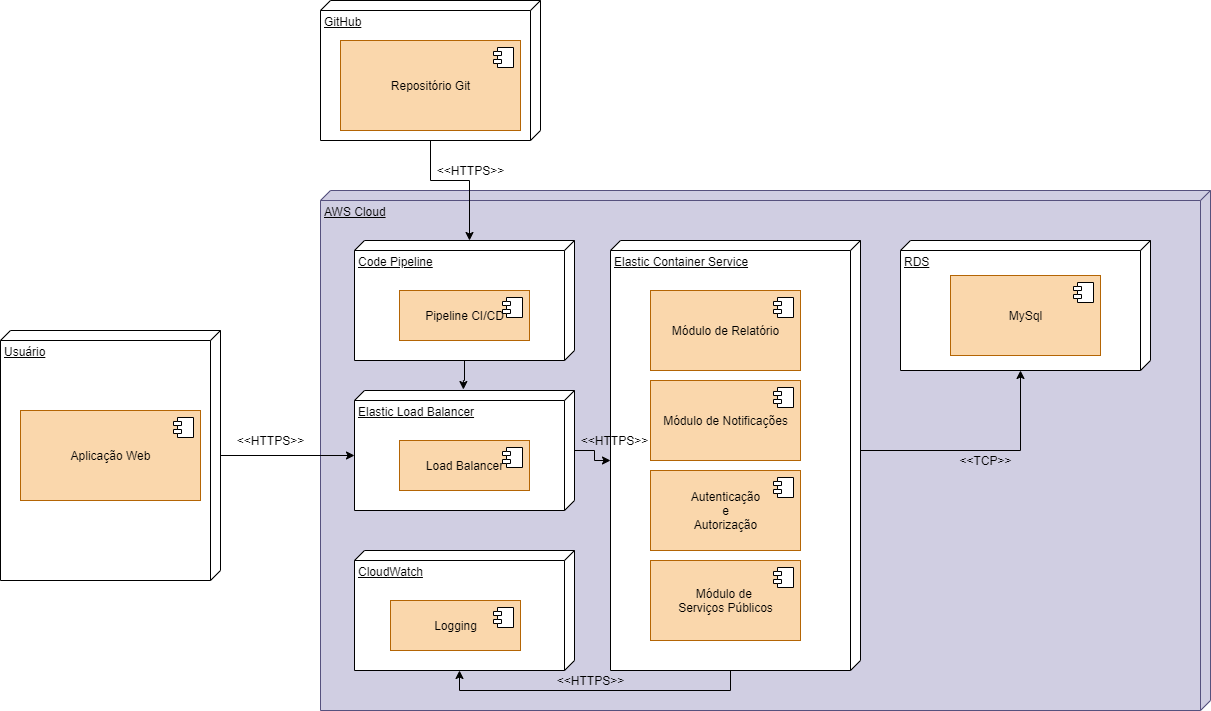
### Diagrama de componentes



### Descrição dos componentes

|  |  |  |
| --- | --- | --- |
| **Número** | **Componente** | **Descrição** |
|  | MySql | Armazenará as informações cadastradas pelos usuários e será a base para a geração dos relatórios – será reutilizado |
|  | Módulo de Relatório | Serviço responsável pela geração dos relatórios – a ser desenvolvido |
|  | Módulo de Notificações | Serviço responsável por enviar notificações aos usuários – a ser desenvolvido |
|  | Módulo de Autenticação e Autorização | Serviço responsável por realizar a autenticação e autorização dos usuários para utilização do sistema – a ser desenvolvido |
|  | Módulo de Serviços Públicos | Serviço responsável por gerenciar as ações referentes aos serviços públicos do sistema (cadastro, consulta, edição e remoção) – a ser desenvolvido |
|  | Gestão de Containers | Serviço responsável por gerenciar os containers onde será executada a aplicação – será reutilizado |
|  | Balanceador de Carga | Serviço responsável por distribuir as requisições recebidas dentro dos containers – será reutilizado |
|  | Gerenciamento de Logs | Serviço responsável por armazenar os logs das aplicações para consulta e análise posterior – será reutilizado |
|  | Armazenamento de websites estáticos | Serviço responsável por armazenar os arquivos que contém a configuração da aplicação frontend – será reutilizado |
|  | Entrega de Conteúdo | Serviço responsável por realizar a distribuição e entrega do conteúdo para os usuários – será reutilizado |
|  | FrontEnd Aplicação Web | Serviço responsável pela criação dos componentes que serão utilizado para interação com o usuário – a ser desenvolvido |
|  | Estilo | Serviço responsável por aplicar os padrões de estilo de design nas páginas da aplicação web – a ser desenvolvido |
|  | Comunicação com API | Serviço responsável por enviar requisições para o BackEnd conforme ação do usuário – a ser desenvolvido |

## Diagrama de implantação



## Plano de Testes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número** | **Caso de uso** | **Objetivo do caso de teste** | **Entradas** | **Resultados esperados** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

## Estimativa de pontos de função

<Indique o *link* do repositório (como o [GitHub](http://github.com), por exemplo) onde sua planilha de estimativa de pontos de função está disponível.>

## Informações da implementação

<Este é um item **não obrigatório**, mas desejável.

Informe aqui o *link* para seu repositório de código e quaisquer informações necessárias para seu acesso.>

## Referências

<Esse trabalho não requer revisão bibliográfica e, por isso, a inclusão das referências não é obrigatória, embora seja recomendada. Caso você deseje incluir referências empregadas em seu trabalho, relacione-as de acordo com as normas ABNT, disponíveis em [www.pucminas.br](http://www.pucminas.br), no *link*: <http://portal.pucminas.br/imagedb/documento/DOC_DSC_NOME_ARQUI20160217102425-n.pdf>.

Exemplo:

SOBRENOME DO AUTOR, Nome do autor. **Título do livro ou artigo.** Cidade: Editora, ano.>