**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

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

Trabalho de Conclusão de Curso

Sistema de Prestação de Serviços ao Cidadão - DescomplicaCidadão

Bruno de Sá Freire Ponte

Belo Horizonte

Agosto / 2021.

# 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

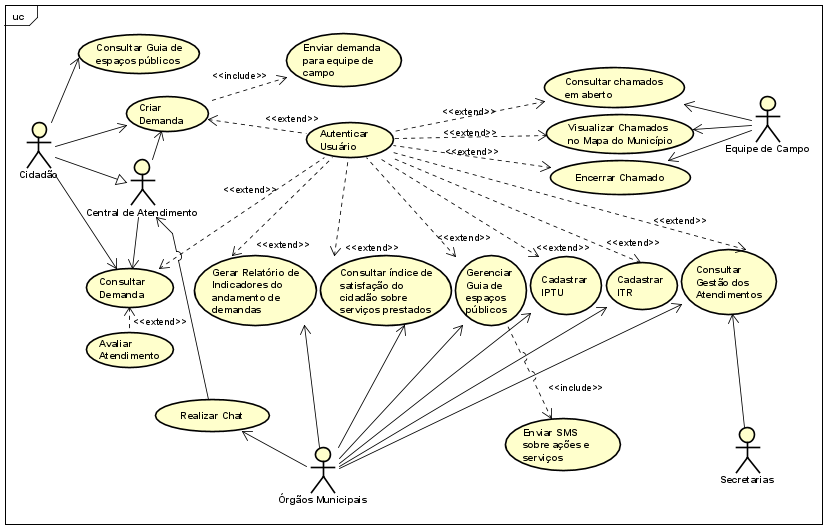
10. Informações da implementação 11

11. Referências 12

## Cronograma de trabalho

|  |  |  |  |
| --- | --- | --- | --- |
| **Datas** | | **Atividade / Tarefa** | **Produto / Resultado** |
| **De** | **Até** |
| 26/06/2021 | 26/06/2021 | 1. Entendimento geral do escopo e atividades do projeto. | Definição do tema do trabalho. |
| 27/06/2021 | 29/06/2021 | 2. Criação do cronograma de atividades. | Criação do cronograma de atividades documentado e definição de datas. |
| 30/06/2021 | 02/07/2021 | 3. Criação do Diagrama de Caso de Uso. | Diagrama de Caso de Uso criado. |
| 03/07/2021 | 05/07/2021 | 4. Especificação dos casos de uso Especificação de casos de usos documentado. | Especificação dos casos de uso Especificação de casos de usos documentado. |
| 06/07/2021 | 08/07/2021 | 5. Criação do modelo Entidade-Relacionamento. | Diagrama de Entidade-Relacionamento documentado. |
| 09/07/2021 | 11/07/2021 | 6. Criação do diagrama de componentes. | Diagrama de componentes definido. |
| 12/07/2021 | 14/07/2021 | 7. Criação do diagrama de classes de domínio. | Diagrama de classes de domínio criado. |
| 15/07/2021 | 17/07/2021 | 8. Criação do diagrama de implantação. | Diagrama de implantação documentado. |
| 18/07/2021 | 20/07/2021 | 9. Escolha dos 3 casos de uso para desenvolvimento. | Casos de uso selecionados. |
| 21/07/2021 | 23/07/2021 | 10. Detalhamento dos requisitos de interface para os casos de uso selecionados. | Requisitos de interface definidos. |
| 24/07/2021 | 26/07/2021 | 11. Criação do plano de testes para os casos de uso selecionados. | Plano de testes documentado. |
| 27/07/2021 | 29/07/2021 | 12. Estimativa de pontos de função. | Planilha de pontos de função documentada. |
| 30/07/2021 | 09/08/2021 | 13. Implementação e codificação do sistema (casos de uso selecionados e sistema básico). | Codificação realizada. |
| 10/08/2021 | 10/08/2021 | 14. Execução do primeiro ciclo de testes no sistema. | Testes realizados. |
| 11/08/2021 | 11/08/2021 | 15. Correção de bugs encontrados após o primeiro ciclo de testes. | Problemas corrigidos na aplicação. |
| 12/08/2021 | 12/08/2021 | 16. Realização de um segundo ciclo de testes na aplicação para verificar se as correções fizeram surgir novos bugs. | Testes realizados. |
| 13/08/2021 | 15/08/2021 | 17. Finalização e entrega do projeto. | Envio dos fontes ao GitHub e entrega do TCC na plataforma do AVA. |

## Diagrama de casos de uso



## Requisitos não-funcionais

Os requisitos não-funcionais listados abaixo são imprescindíveis para que a aplicação funcione adequadamente do ponto de vista arquitetural.

* 1. Requisitos de Produtos
     1. Usabilidade

Os usuários dos órgãos municipais e da equipe de campo deverão receber treinamento antes de operar a aplicação, a fim de obter o melhor uso da aplicação.

* + 1. Eficiência

A aplicação deverá permitir processar uma quantidade de demandas por dia usando como base a quantidade populacional do município.

* + 1. Confiabilidade

A aplicação deverá ter alta disponibilidade em todos os dias e horários da semana, inclusive feriados.

* + 1. Portabilidade

A aplicação deverá ser executada nas plataformas Windows, MAC e Linux em ambiente desktop e Android e Apple em aparelhos móveis.

* 1. Requisitos Organizacionais
     1. Entrega

Um relatório para acompanhamento do andamento das atividades deverá ser entregue toda segunda-feira.

* + 1. Implementação

A aplicação deverá ser desenvolvida na linguagem C# com o Visual Studio 2019.

* + 1. Padrões

A aplicação deverá usar programação orientada a objetos com o uso do padrão MVC.

* 1. Requisitos Externos
     1. Interoperabilidade

A aplicação sempre deverá se comunicar com o banco SQL Server.

* + 1. Éticos

A aplicação não apresentará dados sensíveis do usuário sem que ele esteja logado na aplicação.

* + 1. Legais

A aplicação deverá estar em conformidade com as leis municipais quanto a geração do IPTU e ITR.

## 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).

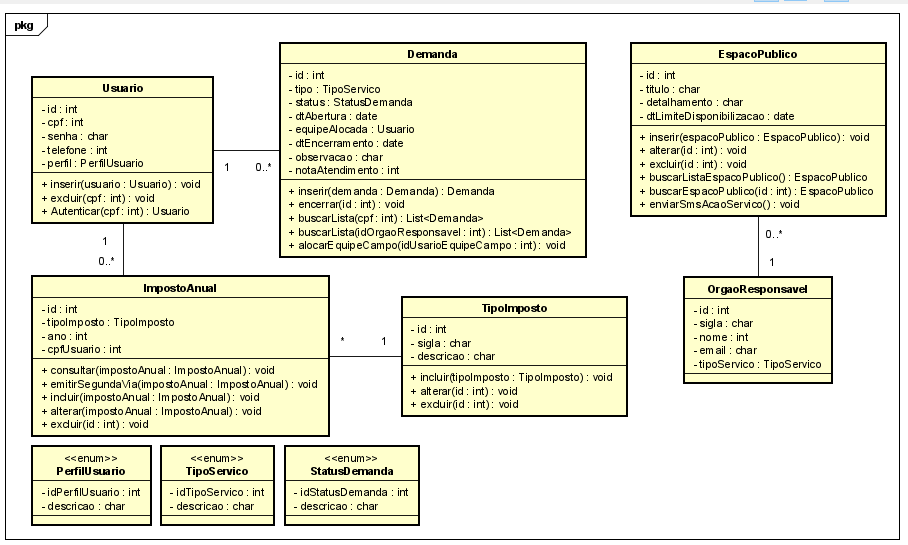
Faça um vídeo de apresentação do protótipo navegável desenvolvido e disponibilize-o de forma que a banca de avaliação do TCC possa visualizá-lo. Esse vídeo deve ter duração de, no máximo, 3 minutos. Utilize, preferencialmente, o formato AVI.

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.>

Link do vídeo de apresentação do protótipo navegável: xxxxxxxxx

Link do repositório onde se encontra o protótipo: xxxxxxxx

## Diagrama de classes de domínio



## Modelo de componentes

### Padrão arquitetural

<Nesta seção, você deve indicar o padrão arquitetural escolhido para o desenvolvimento da aplicação (por exemplo, MVC, MVVM, etc). Liste todas as **tecnologias** que serão utilizadas em sua implementação da proposta>.

### Diagrama de componentes

<Apresente uma imagem legível do diagrama de componentes da aplicação representando todos os componentes, indicando os elementos da arquitetura e as interfaces entre eles. Ele deve contemplar os principais elementos das tecnologias listadas no item anterior. >

### Descrição dos componentes

<Faça uma descrição sucinta dos componentes indicando o papel de cada um deles dentro do padrão arquitetural escolhido. Indique também quais componentes serão reutilizados (navegadores, SGBDs, *middlewares*, etc), quais componentes serão adquiridos por serem proprietários e quais componentes precisam ser desenvolvidos.>

|  |  |  |
| --- | --- | --- |
| **Número** | **Componente** | **Descrição** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## Diagrama de implantação

<Nesta seção, apresente o diagrama de implantação da aplicação, indicando o mapeamento dos elementos de *software* da arquitetura para os elementos de *hardware* onde eles executarão.>

## 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.>