**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**

**PUC Minas Virtual**

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

Trabalho de Conclusão de Curso

Cidade Digital

Camila Martins Maschion

São Paulo

<Janeiro/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** |
| 17 / 08 / 21 | 18 / 08 / 21 | 1. Entendimento geral do escopo, atividades do projeto e criação do cronograma | Definição do tema do trabalho. |
| 19 / 08 / 21 | 20 / 08 / 21 | 2. Leitura de documentação de roteiro e escopo do TCC. | Esboço de planejamento de atividades. |
| 24 / 08 / 21 | 25 / 08 / 21 | 3. Diagrama de casos de uso | Diagrama de casos de uso de todo o sistema. |
| 28 /08 / 21 | 31/ 08 / 21 | 4. Requisitos não funcionais | Lista de requisitos não funcionais. |
| 07 / 09 / 21 | 08 / 09 / 21 | 5. Protótipo navegável | Protótipo Figma |
| 28 / 09 / 21 | 30 / 09 / 21 | 6. Diagrama de classes de domínio | Diagrama de classes de domínio criado |
| 10 / 10 / 21 | 15 / 10 / 21 | 7. Modelos de Componentes | Diagrama de componentes e descrição de componentes |
| 20 / 10 / 21 | 25 / 10 / 21 | 8. Modelo de implantação | Diagrama de modelo de implantação |
| 26 / 10 / 21 | 27 / 10 / 21 | 9. Planos de testes | Plano de testes dos 3 casos de uso. |
| 01 / 11 / 21 | 02 / 11 / 21 | 10.Desenvolvimento da lista de pontos de função e preenchimento da planilha | Planilha com pontos de função de todo o sistema. |
| 05 / 12 / 21 | 05 / 12 / 21 | 11. Revisão e ajustes | Github criado para armazenamento do projeto. |
| 15 / 01 / 22 | 18 / 01 / 22 | 12. Revisão e ajustes | Atualização do protótipo navegável e atualização do github. |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 13. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 14. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 15. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 16. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 17. |  |
| \_\_ / \_\_ / \_\_ | \_\_ / \_\_ / \_\_ | 18. |  |

## Diagrama de casos de uso

|  |
| --- |
|  |

## Requisitos não-funcionais

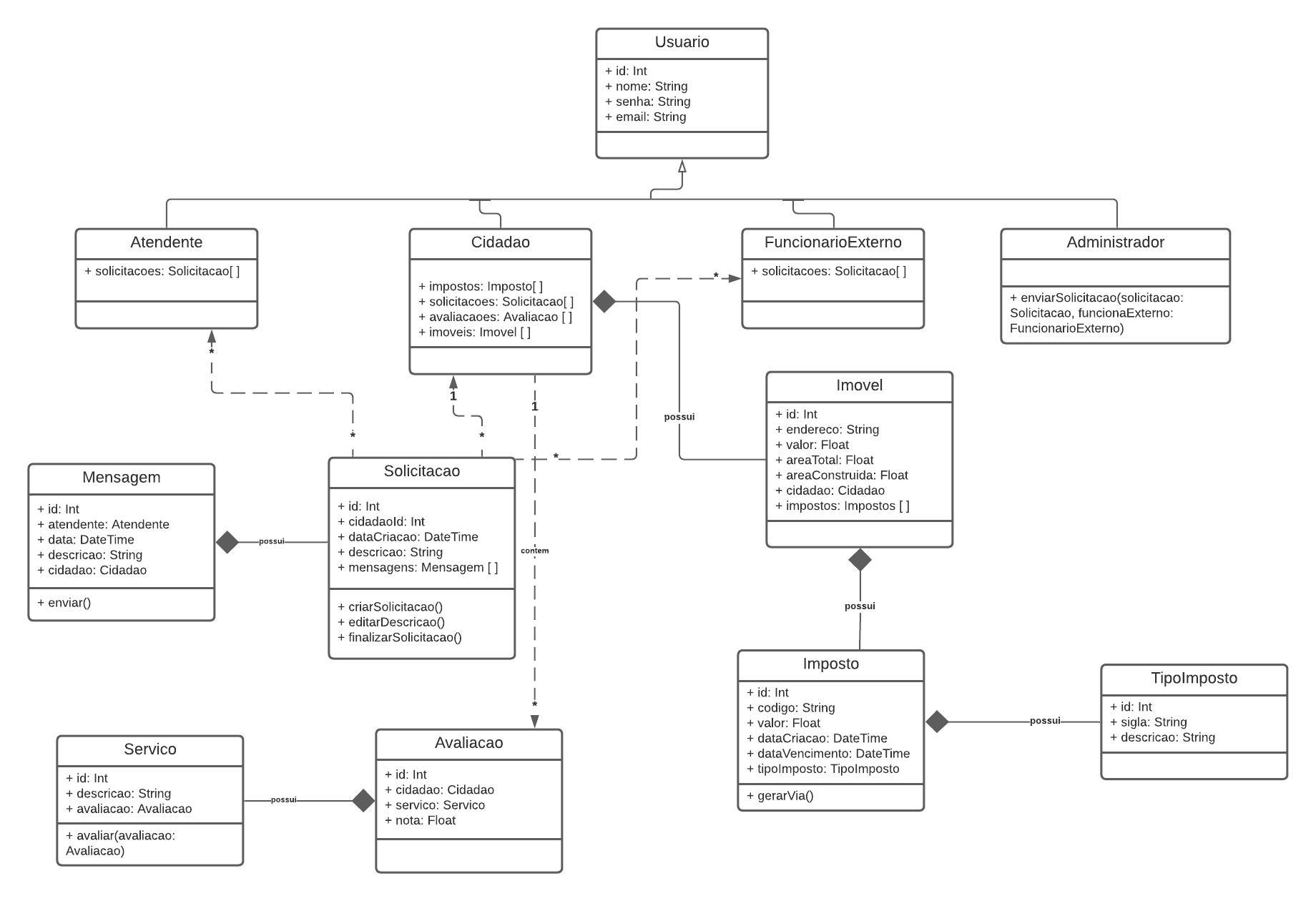
* A aplicação deve conter acessibilidade de forma que todas as informações possam ser lidas via aplicativos de leitura de tela;
* A aplicação deve conter um modo que facilite o uso para pessoas daltônicas.
* A aplicação deve responder as requisições do usuário em no máximo 1000 milissegundos;
* A aplicação deve comportar 50% da população online ao mesmo tempo.
* A aplicação deve executar em plataforma Android e iOS, tanto em smartphones quanto em tablets;
* A aplicação deve ser responsiva em todos os tamanhos de tela de forma que o layout não seja afetado;
* As informações dos usuários devem estar asseguradas conforme a Lei Geral de Proteção de Dados;
* A aplicação deve garantir funcionamento 24 horas por dia, 7 dias da semana durante 365 dias do ano;
* A aplicação deve garantir um canal seguro para troca de requisições entre as aplicações Android e iOS de forma segura e encriptada.

## Protótipo navegável do sistema

* Protótipo navegável em formato AVI:  
  Link: [Protótipo navegável](https://drive.google.com/drive/folders/1GsF8o7WLA0DeJ8LX58TfU3ErM19LgssN?usp=sharing)

Link: [GitHub](https://github.com/CaMaschion/cidade-digital)

## Diagrama de classes de domínio

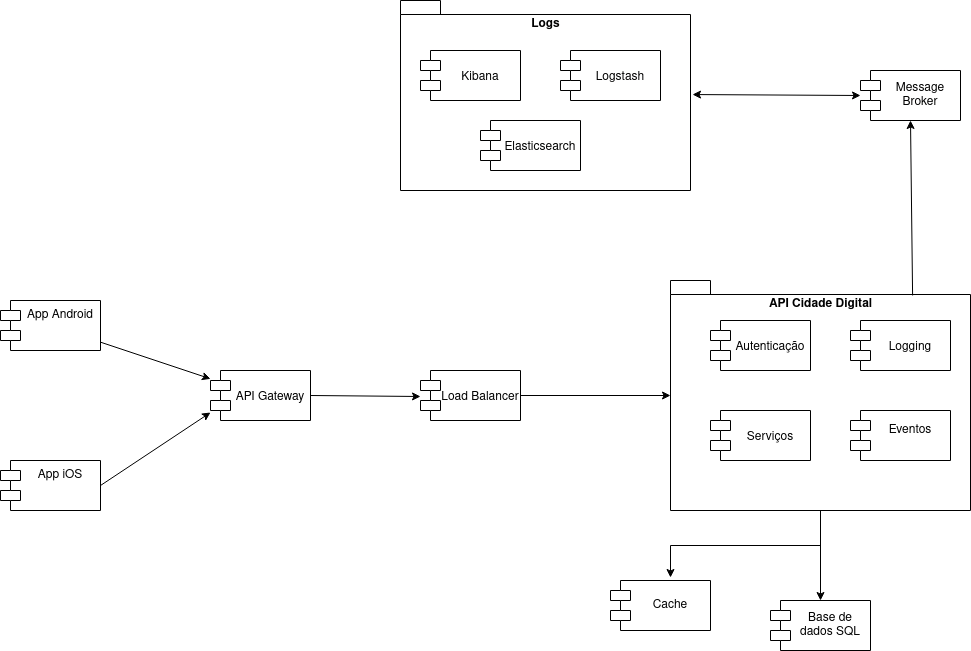


## Modelo de componentes

### Padrão arquitetural

O padrão arquitetural escolhido para o desenvolvimento da aplicação é o MVC para as APIs e MVVM para o Android. As **tecnologias** utilizadas na implementação serão: ASP.NET Core, Entity Framework, SQL Server, RabbitMQ, Kubernetes, Elasticsearch, Kibana, Logstash, Android SDK, Retrofit e SQLite.

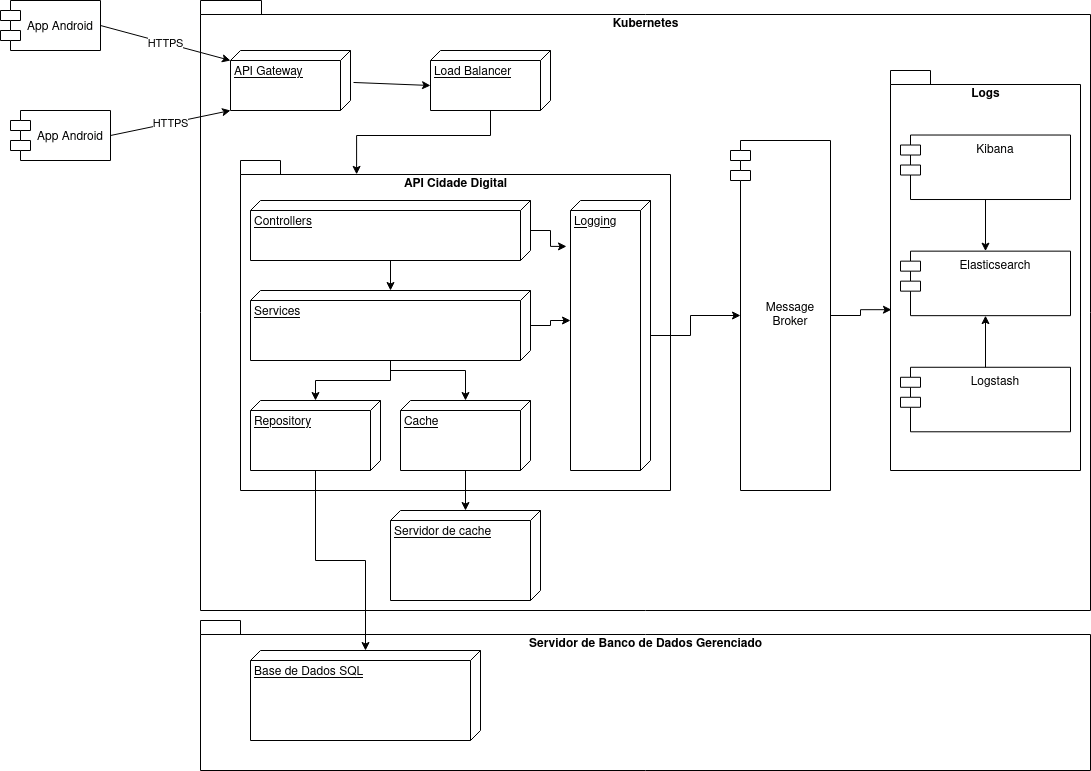
### Diagrama de componentes



### Descrição dos componentes

|  |  |  |
| --- | --- | --- |
| **Número** | **Componente** | **Descrição** |
|  | App Android | Aplicativo para plataforma Android para acessar os serviços do Cidade Digital. |
|  | App iOS | Aplicativo para plataforma iOS para acessar os serviços do Cidade Digital. |
|  | API Gateway | Componente de gerenciamento de apis que fica entre o cliente e a coleção de serviços. |
|  | Load Balancer | Recurso utilizado para aumentar a disponibilidade dos componentes acessados. |
|  | API Cidade Digital | Componentes utilizados para expor as operações que serão invocadas por outros módulos e clientes do sistema. |
|  | Base de Dados | Componente utilizado para armazenar os dados da aplicação. |
|  | Message Broker | Componente utilizado para realizar integração entre serviços de forma assíncrona. |
|  | Logs | Componentes utilizados para indexar, armazenar e visualizar logs da aplicação. |
|  | Cache | Recurso utilizado para diminuir a latência das requisições. |

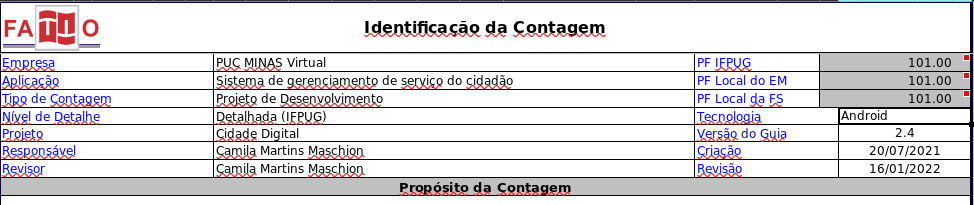
## Diagrama de implantação



## Plano de Testes

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Número** | **Caso de uso** | **Objetivo do caso de teste** | **Entradas** | **Resultados esperados** |
|  | Listar minhas solicitações. | Exibir todas as solicitações do usuário. | - Clicar em “Minhas Solicitações” na tela principal . | A lista de solicitações deve ser exibida com a informação de data do registro e status da solicitação. |
|  | Adicionar nova solicitação. | Inserir nova solicitação no sistema | - Clicar em “Minhas Solicitações” na tela principal .  - Clicar no botão    para adicionar nova solicitação.  - Selecionar o tipo de solicitação através do “dropdown”  - Inserir a descrição da solicitação e clicar no botão ENVIAR. | Deve ser gerado um número de protocolo com a mensagem “Sua solicitação foi registrada com sucesso. Seu protocolo é: XXXXXXXXXX-XXXX”. |
|  | Gerar segunda via de IPTU. | Gerar uma segunda via de boleto do IPTU. | - Clicar no botão “Consultar Impostos” na tela principal.  - Clicar no botão “Emitir 2° IPTU” e inserir os dados nos campos.  - Clicar no botão “Gerar 2ª via”. | Deve aparecer uma mensagem “Segunda via de IPTU emitida com sucesso, verifique seu e-mail. Seu protocolo é: XXXXXXXXXX-XXXX” |

## Estimativa de pontos de função



*Link* do repositório <https://github.com/CaMaschion/cidade-digital> - planilha de estimativa de pontos de função.