

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS**  
**NÚCLEO DE EDUCAÇÃO A DISTÂNCIA**  
**Pós-graduação *Lato Sensu* em Arquitetura de Software Distribuído**

**Igor Galvão de Oliveira**

**SISTEMA DE GESTÃO INTEGRADA MUNICIPAL**

Belo Horizonte

2021

**Igor Galvão de Oliveira**

**SISTEMA DE GESTÃO INTEGRADA MUNICIPAL**

Trabalho de Conclusão de Curso de Especialização  
em Arquitetura de Software Distribuído como  
requisito parcial à obtenção do título de especialista.

Orientador: Dr. Pedro A. Oliveira

Belo Horizonte

2021

*Dedico este trabalho a Deus e a minha família.*

## **AGRADECIMENTOS**

A Deus por me permitir concluir minha especialização.

A minha família por me apoiar em todos os momentos.

A minha noiva por me apoiar e incentivar sempre a ser melhor.

A mim por nunca desistir e sempre persistir.

## RESUMO

Este projeto aborda uma solução para melhor atender o município de Bom Destino, utilizando dispositivos móveis ou *desktop*. Atualmente o município enfrenta problemas devido a alguns órgãos municipais não serem informatizados, os que são, enfrentam dificuldades de integração de sistemas entre as diferentes secretarias, autarquias e empresas municipais. Contudo, é interesse do município informatizar todos os órgãos municipais que ainda não são; integrar os sistemas das diferentes secretarias, autarquias e empresas municipais, a fim de agilizar a tramitação de processos e o atendimento aos cidadãos. Para solucionar o problema, o projeto propõe a integração de todos os sistemas legados utilizando Enterprise Service Bus (ESB), também será desenvolvido um novo sistema baseado em Web, que irá integrar novos módulos que irão auxiliar o município a prestar um melhor serviço nas áreas de: saúde, educação, agropecuária, segurança e meio ambiente. O novo sistema irá se comunicar com sistemas legados, evitando assim gastos com migração de sistema já existente. O projeto está estruturado da seguinte forma: definição dos requisitos com os *stakeholders*, modelagem e projeto arquitetural, prova de conceito funcional e avaliação da arquitetura. Espera-se ao final do projeto que as integrações sejam atendidas com a prova de conceito proveniente das modelagens/definições deste documento.

**Palavras-chave:** arquitetura de software, integração de sistemas, projeto de software, web, dispositivos móveis.

## SUMÁRIO

<b>1. Objetivos do trabalho.....</b>	<b>7</b>
<b>2. Descrição geral da solução .....</b>	<b>7</b>
2.1. Apresentação do problema .....	8
2.2. Descrição geral do software (Escopo).....	8
<b>3. Definição conceitual da solução .....</b>	<b>9</b>
3.1. Requisitos Funcionais .....	9
3.2 Requisitos Não-Funcionais.....	11
3.3. Restrições Arquiteturais .....	13
3.4. Mecanismos Arquiteturais .....	14
<b>4. Modelagem e projeto arquitetural .....</b>	<b>15</b>
4.1. Modelo de componentes .....	15
4.2. Modelo de implantação .....	17
4.3. Modelo de dados .....	19
<b>5. Prova de Conceito (POC) / protótipo arquitetural .....</b>	<b>20</b>
5.1. Implementação e Implantação .....	20
5.2. Interfaces/ APIs .....	39
<b>6. Avaliação da Arquitetura .....</b>	<b>52</b>
6.1. Análise das abordagens arquiteturais .....	52
6.2. Cenários .....	52
6.3. Avaliação.....	53
6.4. Resultado .....	70
7. Conclusão .....	72
<b>REFERÊNCIAS.....</b>	<b>73</b>
<b>APÊNDICES .....</b>	<b>74</b>

## **1. Objetivos do trabalho**

O objetivo deste projeto é apresentar uma proposta de arquitetura de criação de um novo sistema baseado em Web e dispositivos móveis e integrar os diversos sistemas já existentes no município. O projeto visa informatizar todo o município, integrando sistemas existentes com novos módulos para que o município consiga realizar uma gestão eficiente e inteligente.

Os objetivos específicos são:

1. Criar um módulo de serviço ao cidadão, para que o mesmo possa consultar informações de interesse do cidadão – Imposto Predial e Territorial (IPTU), Imposto Territorial Rural (ITR).
2. Criar módulo de integração geral, para que seja possível integrar todos os sistemas de diferentes tecnologias e aplicações existentes, de modo a prover transparência, disponibilidade e robustez para esta integração geral.
3. Criar módulo de informações municipais georreferenciadas, para que seja possível a prefeitura ter controle total sobre a expansão da ocupação do município sede e de seus distritos.
4. Criar módulo de qualidade de vida, para que seja possível a prefeitura disponibilizar serviços de saúde, inclusão social e de combate a pobreza.
5. Criar módulo de segurança, para que seja possível acompanhar dados relativos à segurança pública.
6. Criar módulo de gestão estratégica de projetos, para que possa prover a gestão de todos os projetos municipais, com indicadores do andamento individual e global dos projetos.

## **2. Descrição geral da solução**

Integrar os diversos sistemas existentes em secretarias, autarquias e empresas municipais e desenvolver sistema que permita o município fornecer serviços aos seus cidadãos em diversas áreas como: saúde, educação, segurança, agropecuária e meio ambiente.

## **2.1. Apresentação do problema**

Com a popularização do acesso à internet, o município de Bom Destino viu a necessidade de se modernizar, para que assim consiga prestar um serviço a sua população mais eficiente e eficaz.

O município de Bom Destino enfrenta algumas dificuldades para alcançar a modernização de seus sistemas, pois diversos sistemas existentes em suas secretarias, autarquias e empresas municipais não tem integração entre si. O uso de diversos sistemas sem integração faz com que o serviço prestado seja ineficiente ou em muitos casos com o tempo de espera acima do desejado.

Também é de desejo do município a criação de um novo sistema que possa disponibilizar novos recursos para os seus cidadãos. O sistema que será desenvolvido enfrenta algumas restrições onde o mesmo deve ser implantando em todos os órgãos municipais e deve ser projetado para ser modular e implantável por módulos, de acordo com a prioridade e necessidade da prefeitura.

## **2.2. Descrição geral do software (Escopo)**

A elaboração do sistema a ser desenvolvido tem por objetivo modernizar e aprimorar os serviços prestados pelo município de Bom Destino, com isso será possível realizar um melhor gerenciamento de recursos e serviços nas áreas de prioridade da atual gestão administrativa.

O sistema irá possibilitar que os cidadãos tenham acessos a diversos novos serviços, tais como: gerar e consultar IPTU e ITR, monitorar dados da segurança pública, combater a pobreza de forma eficiente e realizar programas de inclusão social, também será possível monitorar a expansão do município, com isso será possível o mapeamento completo do potencial econômico do município.



### **3. Definição conceitual da solução**

#### **3.1. Requisitos Funcionais**

##### **Módulo de Acesso**

- Realizar autenticação e autorização

O sistema deve permitir que os usuários realizem cadastro de forma única e garanta acesso a todos os sistemas do município, conforme matriz de acesso de cada usuário.

##### **Módulo de serviço ao cidadão**

- Gerar Impostos IPTU / ITR

O sistema deve permitir ao cidadão gerar o imposto (IPTU/ITR) conforme o tipo de pessoa: física, jurídica ou rural. Sistema deverá ter comunicação com serviço dos correios para recuperar endereço através do CEP e acesso ao serviço do sistema STUR para que possa realizar o processamento dos impostos municipais.

##### **Módulo de Georreferência**

- Visualizar mapa do município

O sistema deve permitir o cidadão do município visualizar o mapa do município para que seja possível a prefeitura ter controle sobre a ocupação do município.

##### **Módulo Qualidade de vida**

- Manter Projetos Sociais

O sistema deve permitir o usuário com o devido acesso, possam cadastrar e manter informações referente a projetos sociais que são realizados no município.

- Manter Inscrição em projetos sociais

O sistema deve permitir o cidadão do município realizar inscrição e manter a inscrição em projetos sociais que o município tenha disponível.

- Manter agendamento médico

O sistema deve permitir que o cidadão do município possa agendar e manter o atendimento médico nos postos de saúde do município.

- Manter Unidade Básica de Saúde

O sistema deve permitir o usuário com o devido acesso, possam manter os dados dos postos de atendimentos médico do município.

- Manter agendamento de vacinação

O sistema deve permitir que o cidadão do município possa manter o agendamento de vacinação.

- Visualizar acompanhamento escolar

O sistema deve permitir que o cidadão do município possa visualizar o acompanhamento escolar dos seus dependentes que estudem em escola públicas do município.

- Agendar coleta de lixo seletivo

O sistema deve permitir que o cidadão do município possa realizar o agendamento da coleta de lixo seletivo.

### **Módulo de segurança**

- Cadastrar Denúncia

O sistema deve permitir o cidadão cadastrar denúncia para que possa notificar as autoridades do município.

### **Módulo de integração geral**

- Recuperar prontuário eletrônico do SUS

O sistema deve recuperar informações do prontuário eletrônico do cidadão do SUS do sistema SASCi. Para alcançar o resultado será utilizado a arquitetura ESB – Enterprise Service Bus (ESB).

- Recuperar imposto processado

O sistema deve recuperar informações do processamento de imposto do cidadão do sistema STUR. Para alcançar o resultado será utilizado a arquitetura ESB – Enterprise Service Bus (ESB).

### 3.2 Requisitos Não-Funcionais

- Usabilidade – O sistema deve prover boa usabilidade

<b>Estímulo</b>	Cidadão realizando um agendamento médico
<b>Fonte do Estímulo</b>	Cidadão acessando e realizar agendamento médico
<b>Ambiente</b>	Funcionamento, carga normal
<b>Artefato</b>	Módulo de qualidade de vida
<b>Resposta</b>	O front-end apresenta opções intuitivas, simples e objetivas.
<b>Medida da Resposta</b>	O usuário conseguiu realizar o cadastro em no máximo 5 minutos.

- Acessibilidade – O sistema deve suportar ambientes Web responsivos e ambientes móveis

<b>Estímulo</b>	Consultar Atendimento Médico
<b>Fonte do Estímulo</b>	Usuário acessando o sistema de um celular
<b>Ambiente</b>	Funcionamento, carga normal
<b>Artefato</b>	Módulo qualidade de vida
<b>Resposta</b>	O front-end se adaptou as resoluções e tamanho das telas, mudando os componentes de posição de forma a ficar melhor a navegação do usuário.
<b>Medida da Resposta</b>	Identidade visual semelhante em todas as resoluções, com objetos redimensionados de acordo com a resolução e tamanho.

- Desempenho – O sistema deve ser rápido

<b>Estímulo</b>	Acessando a tela de gerar impostos IPTU/ITR
<b>Fonte do Estímulo</b>	Usuário gerando boleto de pagamento de imposto
<b>Ambiente</b>	Funcionamento, carga normal
<b>Artefato</b>	Módulo de serviço ao cidadão

<b>Resposta</b>	O sistema respondeu com os dados solicitados.
<b>Medida da Resposta</b>	O sistema respondeu em menos de 10 segundos.

- Segurança – O sistema deve apresentar altos padrões de segurança

<b>Estímulo</b>	Acessar uma página privada pela URL sem estar logado no sistema.
<b>Fonte do Estímulo</b>	Qualquer usuário.
<b>Ambiente</b>	Em funcionamento com carga normal.
<b>Artefato</b>	Todos os módulos
<b>Resposta</b>	O sistema deve redirecionar o usuário para a tela de login solicitando usuário e senha.
<b>Medida da Resposta</b>	O sistema não deve permitir o acesso às páginas privadas.

- Interoperabilidade – O sistema deve ser comunicar com vários sistemas, com tecnologias heterogêneas

<b>Estímulo</b>	Teste de conexão.
<b>Fonte do Estímulo</b>	Usuário agendando um atendimento médico
<b>Ambiente</b>	Funcionamento, carga normal
<b>Artefato</b>	Módulo de integração geral
<b>Resposta</b>	O serviço do sistema SASCi respondeu com sucesso a solicitação, recuperando o prontuário do SUS do paciente.
<b>Medida da Resposta</b>	Conexão efetuada.

- Disponibilidade – O sistema deve operar em qualquer período do dia e da noite.

<b>Estímulo</b>	Shutdown no cluster primário do servidor de aplicação.
<b>Fonte do Estímulo</b>	Administrador do Servidor de Aplicação
<b>Ambiente</b>	Diversos usuários estão utilizando o sistema
<b>Artefato</b>	Todos os módulos
<b>Resposta</b>	Todos os usuários logados na aplicação devem continuar utilizando o sistema sem perceber que houve uma queda de nós do servidor de aplicação.
<b>Medida da Resposta</b>	Todas as solicitações dos usuários devem ser atendidas, podendo

	haver um atraso de 3 segundos devido à queda de um dos nós.
--	---

- Testabilidade

<b>Estímulo</b>	Execução de testes no sistema.
<b>Fonte do Estímulo</b>	Analista Desenvolvedor
<b>Ambiente</b>	Ambiente de Desenvolvimento
<b>Artefato</b>	Todos os módulos
<b>Resposta</b>	O sistema testou todas as funcionalidades disponíveis.
<b>Medida da Resposta</b>	O sistema deve possibilitar efetuar os testes com scripts automatizados executando com apenas um comando.

- Manutenibilidade – O sistema deve apresentar manutenção facilitada.

<b>Estímulo</b>	O componente de negócio responsável por recuperar as informações no sistema SASCi precisa ser corrigido.
<b>Fonte do Estímulo</b>	Exceção ocorreu durante a recuperação do prontuário do cidadão.
<b>Ambiente</b>	O módulo de qualidade de vida utiliza esse componente. Na ocasião da falha, o componente poderia estar recebendo uma requisição com dados inválidos.
<b>Artefato</b>	Módulo de qualidade de vida
<b>Resposta</b>	Todos os envios devem ser atendidos.
<b>Medida da Resposta</b>	As informações devem ser enviadas em até 30 segundos.

### 3.3. Restrições Arquiteturais

- O sistema deve ser desenvolvido em Java (back-end) e Angular (front-end);
- O sistema deve utilizar o serviço dos correios para disponibilizar os dados do endereço;
- O sistema deve utilizar o serviço do IBGE para disponibilizar os dados referente ao município;

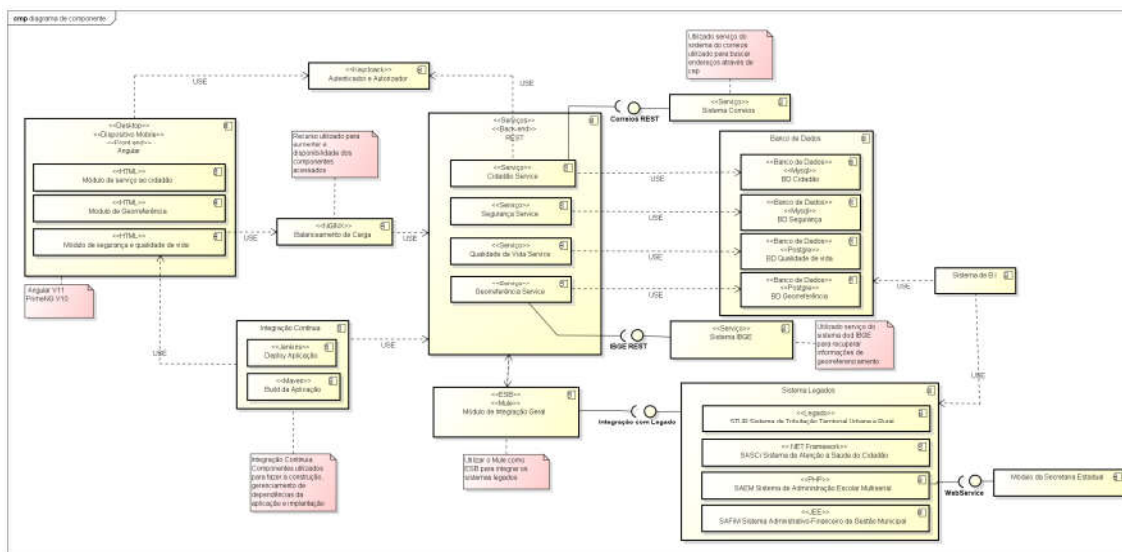
- O sistema deve abrir de forma responsiva em aparelhos menores, como celular e tablet;
- O sistema deve ser modular para facilitar a implantação;
- As integrações entre os sistemas devem utilizar o padrão REST ou SOAP;
- Deverá utilizar plataforma de autenticação e autorização única para acesso de todo o sistema.

### 3.4. Mecanismos Arquiteturais

Mecanismo de Análise	Mecanismo de Design	Mecanismo de Implementação
Front-end	Interface de comunicação com o usuário do sistema	Angular, PrimeNG
Back-end	Regra de negócios da aplicação	Java
Relatórios	Framework responsável por gerar relatórios	Jasper Reports
ESB	Integração de sistemas	Mule
Versionamento	Versionamento do código-fonte da aplicação	Git
Integração com outros módulos e sistema	Interface utilizando XML ou JSON	Webservices e Restful API's
Autenticação e Autorização	Verificação das credenciais e tentativas de conexão	Keycloak
Persistência	Banco de dados relacional	MySQL, Postgre
Persistência	Framework ORM	JPA
Log	Framework de Log	Log4J
Build	Ferramenta de compilação	Maven
Deploy	Deploy da aplicação no servidor e testes automatizados	Jenkins
Cache	Framework de cache	EhCache
Alta disponibilidade	Balanceamento de carga das aplicações	NGINX

## 4. Modelagem e projeto arquitetural

### 4.1. Modelo de componentes



A seguir é mostrado um detalhamento dos componentes e os módulos envolvidos no modelo de componentes. Nessa arquitetura devemos considerar um sistema que será desenvolvido chamado SGM e os sistemas existentes (legados). O sistema a ser desenvolvido será dividido em módulos implantáveis separadamente. A separação entre as aplicações front-end e back-end auxilia na independência entre componentes. Nas aplicações de front-end não será implementado regras negociais do sistema, sendo sua principal função fazer com que o usuário tenha uma experiência intuitiva, simples, objetiva e responsiva. Para alcançar o objetivo será utilizado a combinação de Angular e PrimeNG, com isso será possível o usuário interagir com o sistema independente do dispositivo utilizado. Isso garante o atendimento aos requisitos não funcionais de acessibilidade e usabilidade.

O SGM irá utilizar o autenticador e autorizador Keycloak, sendo um software de código fonte aberto e fornecido pela empresa Red Hat, a qual o município tem contrato vigente. O Keycloak tem como objetivo abstrair toda a complexidade de criação e manutenção de usuário do sistema, pois todo esse conjunto de funcionalidade é possível obter com a adoção do framework. O login será feito em um ponto único e conforme a matriz de acesso do usuário será permitido ou não o acesso as funcionalidades do sistema. Um dos pontos fortes para utilização do Keycloak é o fato de pode configurar de forma fácil e rápida o login utilizando contas de redes sociais como: Google, Facebook e Twitter. A adoção do framework garante o atendimento ao requisito não funcional de segurança.

As regras de negócios do sistema SGM serão desenvolvidas na camada de serviço utilizando a linguagem Java e utilizando o estilo de arquitetura REST. A utilização da linguagem Java se deve ao fato de ser uma linguagem de fácil aprendizado, orientada a objeto e que é amplamente utilizado em sistemas corporativos pelo mundo. A utilização do estilo de arquitetura REST se deve ao fato de tornar o desenvolvimento dos serviços fácil e o esforço de aprendizado ser pequeno em relação ao protocolo SOAP. Todos os serviços irão utilizar uma estrutura de balanceamento de carga. Isso garante que os recursos mais utilizados sejam escalados e assim as requisições tenham um tempo de resposta baixo.

A aplicação a ser desenvolvida irá adotar o conceito de microsserviços, com isso a camada de serviço será desenvolvida em módulos e cada módulo terá o seu respectivo banco de dados. A utilização desse conceito irá permitir que o sistema tenha uma alta escalabilidade, alta disponibilidade e flexibilidade para utilização de diferentes tecnologias ao mesmo tempo.

Uma das integrações prevista é com o sistema de B.I que irá realizar a mineração dos dados sócio-econômicos e de gestão, utilizando como fonte de dados os diversos bancos de dados do município.

O projeto irá permitir a integração do sistema a ser desenvolvido com todos os sistemas legados que o município já possui. Para alcançar tal objetivo será utilizado o modelo de arquitetura ESB (Enterprise Service Bus), utilizando o Mule, devido a ser um software altamente escalável e baseado em Java. O Mule irá permitir a integração de múltiplos sistemas com protocolos e tecnologias diferentes.

Todos os artefatos serão construídos, testados e publicados automaticamente utilizando a metodologia DevOps. Com a adoção da metodologia DevOps é possível construir, testar e publicar artefatos de forma automática, conseguindo alcançar até o ambiente de produção de forma rápida. Para alcançar esse objetivo serão utilizados o Jenkins e o Maven. A metodologia DevOps garante os requisitos não funcionais de testabilidade e manutenibilidade.



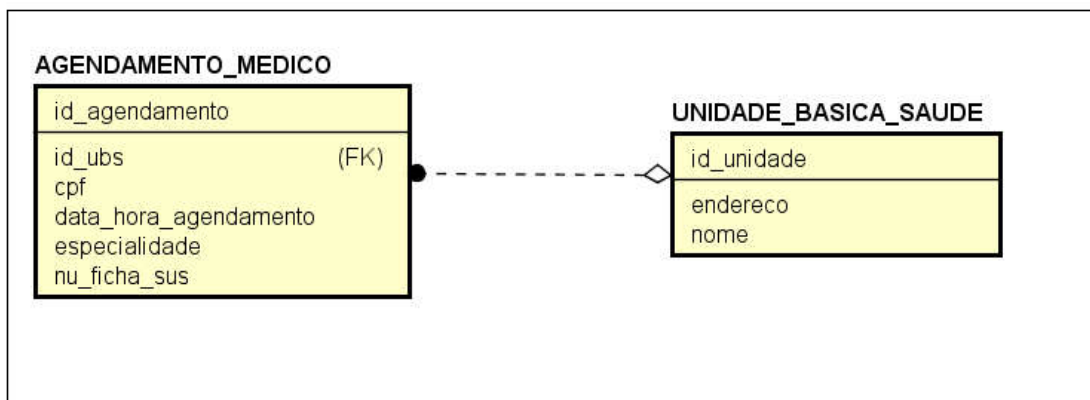


Servidor de aplicação	É o componente do modelo de implantação responsável por gerenciar o processamento das requisições. Nele serão instalados os artefatos e componentes de negócios. Na POC foi escolhida a hospedagem em nuvem utilizando os serviços disponíveis do Heroku.
Servidor de banco de dados	Responsável por armazenar os dados da aplicação. Na POC foi escolhida a hospedagem em nuvem utilizando os serviços disponíveis do Heroku. Conforme a necessidade é possível escolher serviços onde permite ter alta escalabilidade, replicação de dados e backup automáticos.
Servidor de BI	Responsável pelo sistema de Business Intelligence. Essa instancia deve ter permissão a todos bancos de dados do projeto.
Servidor de Integração ESB	Servidor responsável por hospedar a ferramenta que irá possibilitar a integração entre todos os sistemas da prefeitura.

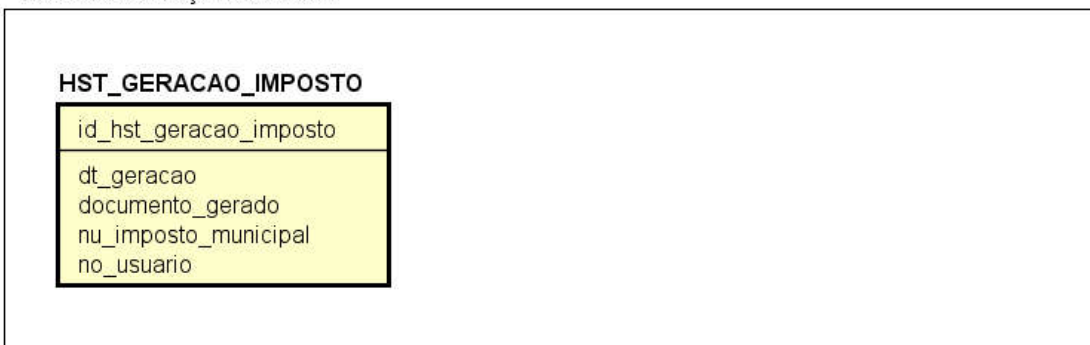
### 4.3. Modelo de dados

A aplicação utiliza vários bancos de dados, sendo um para cada módulo. Um modelo de dados foi gerado, para auxiliar o entendimento da solução.

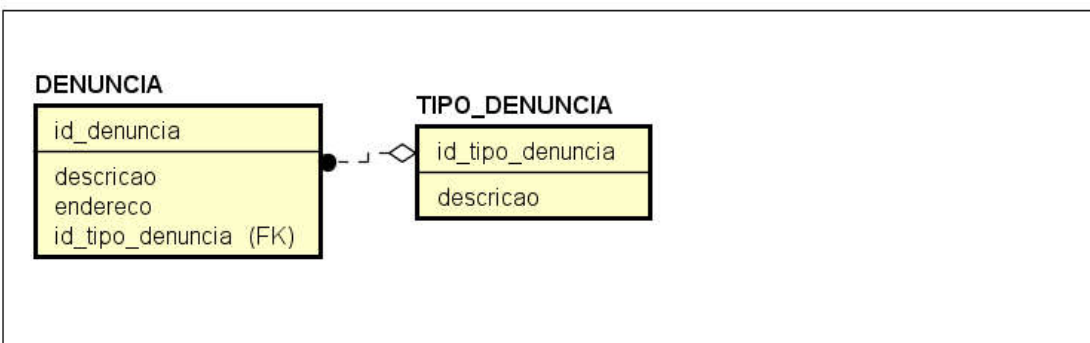
Módulo qualidade de vida



Módulo de serviço ao cidadão



Módulo de segurança



## **5. Prova de Conceito (POC) / protótipo arquitetural**

### **5.1. Implementação e Implantação**

#### **5.1.1 Requisitos não funcionais**

A prova de conceito desse projeto visa atender aspectos importantes da arquitetura que dizem respeito aos seguintes requisitos não funcionais:

- **Segurança – O sistema deve apresentar altos padrões de segurança**

Esse RNF foi escolhido devido à preocupação em manter dados seguros e evitar falhas de segurança do projeto.

Os critérios de aceite são:

- Não permitir que usuários possam acessar páginas privadas sem estar autenticados no sistema.
- Ao identificar que o usuário não está autenticado, o sistema deve redirecionar o usuário para tela de login.
- Não permitir que usuários possam acessar páginas sem ter o devido acesso a funcionalidade
- O sistema deve permitir que o usuário navegue em telas públicas sem precisar estar autenticado.

- **Usabilidade – O sistema deve prover boa usabilidade**

Esse RNF foi escolhido devido à importância em manter um sistema com boa usabilidade, e que possamos garantir uma navegação simples e objetiva.

Os Critérios de aceite são:

- As telas do sistema devem apresentar facilidade de navegação
- O cidadão não pode demorar mais que 5 minutos para efetuar um agendamento médico.
- O acesso as funcionalidades devem apresentar objetividade e não serem confusos.

- **Acessibilidade – O sistema deve suportar ambientes móveis e web responsivos.**

Esse RNF foi escolhido para garantir que atenda todas as exigências da arquitetura em ter um sistema responsivo e que se adapte em celulares, tablets e desktops.

Os Critérios de aceite são:

- A tela do sistema deve apresentar facilidade de navegação e os objetos da tela devem se adaptar de acordo com a resolução identificada, tanto para celulares como desktops.
- O sistema deve se manter com o mesmo padrão de cores e objetos.
- O sistema deve ser compatível com os principais browsers do mercado como: Internet Explore, Chrome, Firefox e Opera.

- **Desempenho – O sistema deve ser rápido**

Esse RNF foi escolhido para garantir uma boa performance na aplicação e poder determinar se o desempenho desse requisito não funcional será atendido.

Os Critérios de aceite são:

- As telas devem ser bem otimizadas para que não demorem mais que 10 segundos a serem renderizadas ao usuário.

- **Interoperabilidade – Cada módulo do sistema deve se comunicar com outros sistemas.**

Esse RNF foi escolhido devido a importância de se integrar o sistema a ser desenvolvido com os sistemas legado e futuros sistemas que possa ser adquirido.

Os Critérios de aceite são:

- O sistema deve expor serviço REST para que seja possível realizar a integração entre módulos e com futuros sistemas.
- O sistema deve conseguir se comunicar com tecnologias heterogêneas dos sistemas legados.

### 5.1.2 Casos de Uso

Para realizar a prova do conceito do projeto, vários casos de usos foram implementados visando fornecer insumos para validação dos requisitos não funcionais. São eles:

Módulo	Caso de Uso	Requisito não funcional
Qualidade de Vida	Consultar Unidade Básica de Saúde	Segurança, Acessibilidade, Usabilidade e Desempenho
	Cadastrar Unidade Básica de Saúde	Segurança, Acessibilidade, Usabilidade e Desempenho
	Alterar Unidade Básica de Saúde	Segurança, Acessibilidade, Usabilidade e Desempenho
	Excluir Unidade Básica de Saúde	Segurança, Acessibilidade, Usabilidade e Desempenho
Acesso	Realizar autenticação e autorização	Segurança, Usabilidade e Acessibilidade
Qualidade de Vida	Consultar Agendamento Médico	Segurança, Acessibilidade, Usabilidade, Desempenho e Interoperabilidade
	Cadastrar Agendamento Médico	Segurança, Acessibilidade, Usabilidade, Desempenho e Interoperabilidade
	Alterar Agendamento Médico	Segurança, Acessibilidade, Usabilidade, Desempenho e Interoperabilidade
	Excluir Agendamento Médico	Segurança, Acessibilidade, Usabilidade, Desempenho e Interoperabilidade
Cidadão	Gerar Imposto IPTU / ITR	Segurança, Acessibilidade, Usabilidade e Interoperabilidade
Segurança	Cadastrar Denúncia	Segurança, Acessibilidade, Usabilidade e Desempenho

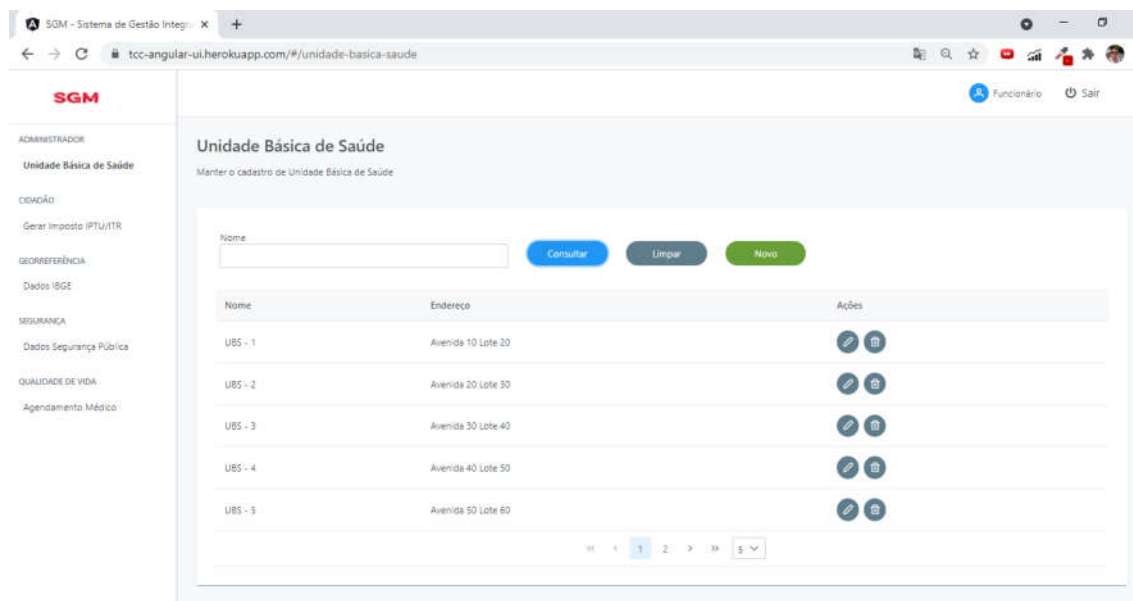
- **Módulo Qualidade de vida**

### Caso de Uso: Consultar Unidade Básica de Saúde

Descrição: Este caso de uso deve permitir os funcionários do município consultar uma unidade básica de saúde. Deve ser disponibilizado o campo nome para realizar a consulta. O acesso a funcionalidade deve ser restrito somente aos funcionários do município com o devido acesso.

ID	1
História do Usuário	Consultar Unidade Básica de Saúde
Eu como	Funcionário do município autenticado e com permissão
Quero	Consultar uma unidade básica de saúde
Para que	Seja possível visualizar, alterar ou excluir.
Prioridade	Média

### Protótipo:



### Caso de Uso: Cadastrar Unidade Básica de Saúde

Descrição: Este caso de uso deve permitir os funcionários do município cadastrar uma unidade básica de saúde. Deve ser disponibilizado os campos nome e endereço. O acesso a funcionalidade deve ser restrito somente aos funcionários do município com o devido acesso.

ID	2
História do Usuário	Cadastrar Unidade Básica de Saúde
Eu como	Funcionário do município autenticado e com permissão
Quero	Cadastrar uma nova unidade básica de saúde
Para que	Sejam listadas e tenha a possibilidade de ser escolhida ao realizar um agendamento médico.
Prioridade	Média

### Protótipo:

SGM - Sistema de Gestão Integrada Municipal

Unidade Básica de Saúde

Cadastrar Unidade Básica de Saúde

Nome \*

Endereço \*

Posto de Saúde N10

Avenida Brasil Lote 20

Limpar Salvar

Voltar

SGM - Sistema de Gestão Integrada Municipal



### Caso de Uso: Alterar Unidade Básica de Saúde

Descrição: Este caso de uso deve permitir os funcionários do município alterar uma unidade básica de saúde. Deve ser disponibilizado uma consulta onde possa escolher qual registro se quer alterar. O acesso a funcionalidade deve ser restrito somente aos funcionários do município com o devido acesso.

ID	3
História do Usuário	Alterar Unidade Básica de Saúde
Eu como	Funcionário do município autenticado e com permissão
Quero	Alterar uma unidade básica de saúde
Para que	Possa corrigir um possível erro de digitação ou atualizar as informações
Prioridade	Média

### Protótipo:

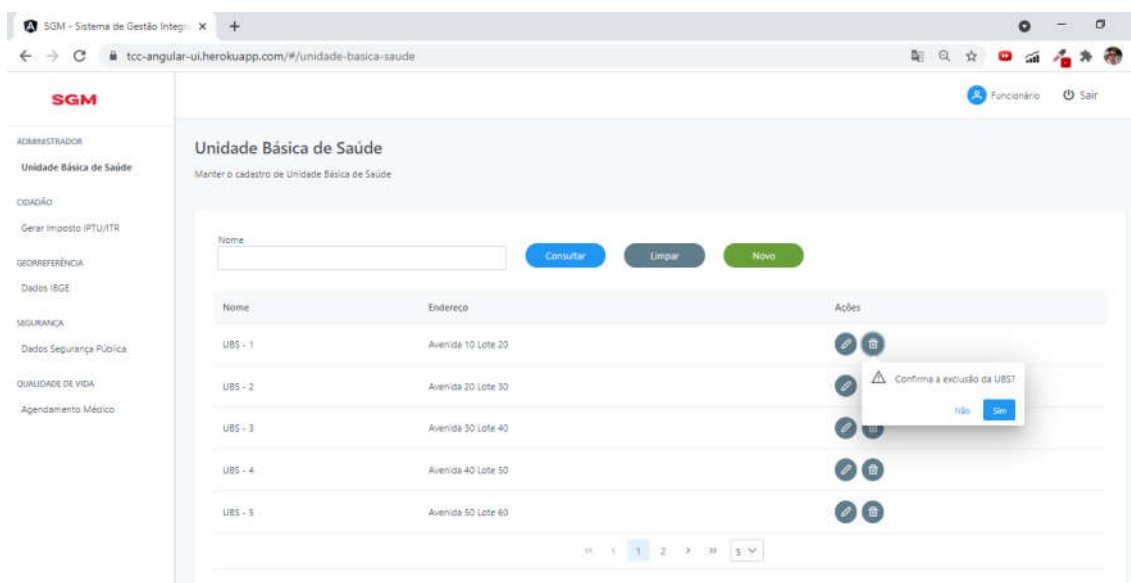
The screenshot shows a web browser window with the URL `tcc-angular-ui.herokuapp.com/#/unidade-basica-saude/incluirid=1`. The application is titled "SGM - Sistema de Gestão Integrada Municipal". On the left, there is a sidebar menu with the following items: ADMINISTRADOR, Unidade Básica de Saúde (highlighted), CIDADÃO, Gerar Imposto IPTU/ITR, GEOPERIFERÊNCIA, Dados IBGE, SEGURANÇA, Dados Segurança Pública, QUALIDADE DE VIDA, and Agendamento Médico. The main content area is titled "Unidade Básica de Saúde" and "Alterar Unidade Básica de Saúde". It contains a form with two input fields: "Nome" (containing "UBS -") and "Endereço" (containing "Avenida 10 Lote 20"). Below the "Nome" field are two buttons: "Limpar" and "Alterar". A "Voltar" button is located at the bottom right of the form area. The footer of the page reads "SGM - Sistema de Gestão Integrada Municipal".

### Caso de Uso: Excluir Unidade Básica de Saúde

Descrição: Este caso de uso deve permitir os funcionários do município excluir uma unidade básica de saúde. Deve ser disponibilizado uma consulta onde possa escolher qual registro se quer excluir. O acesso a funcionalidade deve ser restrito somente aos funcionários do município com o devido acesso.

ID	4
História do Usuário	Excluir Unidade Básica de Saúde
Eu como	Funcionário do município autenticado e com permissão
Quero	Excluir uma unidade básica de saúde
Para que	Possa retirar unidade que foram desativas do sistema.
Prioridade	Média

### Protótipo:



### Caso de Uso: Consultar Agendamento Médico

Descrição: Este caso de uso deve permitir os cidadãos do município consultar um agendamento médico. Deve ser disponibilizado o campo CPF para realizar a consulta. O acesso a funcionalidade deve ser permitido a todo cidadão devidamente autenticado no sistema.

ID	5
História do Usuário	Consultar agendamento médico
Eu como	Cidadão do município
Quero	Consultar um agendamento médico
Para que	Seja possível visualizar, alterar ou excluir.
Prioridade	Alta

### Protótipo:

SGM - Sistema de Gestão Integrada Municipal

### Caso de Uso: Cadastrar Agendamento Médico

Descrição: Este caso de uso deve permitir os cidadãos do município realizar um agendamento de consulta médica. Deve ser disponibilizado o campo CPF e ao clicar no botão de consultar deve buscar informações do prontuário eletrônico do SUS no sistema SASCi e exibir nos seguintes campos: nome, cartão nacional de saúde, data nascimento e endereço. O cidadão deve informar os seguintes campos obrigatórios para realizar o agendamento da consulta, campos: unidade básica de saúde, data hora da consulta e especialidade. O acesso a funcionalidade deve ser permitido a todo cidadão devidamente autenticado no sistema.

ID	6
História do Usuário	Cadastrar Agendamento Médico
Eu como	Cidadão do município
Quero	Cadastrar um novo agendamento médico
Para que	Seja possível realizar uma consulta com data e hora marcada na devida especialidade.
Prioridade	Alta

### Protótipo:

The screenshot displays a web application interface for medical appointment registration. The sidebar on the left contains navigation links: CIDADÃO, Gerar Imposto IPTU/ITR, GEOPREFERÊNCIA, Dados IBGE, SEGURANÇA, Dados Segurança Pública, QUALIDADE DE VIDA, and Agendamento Médico. The main content area is titled 'Agendamento Médico' and includes the subtitle 'Realizar Agendamento Médico em Unidade Básica de Saúde'. The form contains the following fields and controls:

- CPF \***: Input field with value '019.833.171-10' and a blue button 'Consultar Dados SUS'.
- Nome**: Input field with value 'Luiz Bani Fino'.
- Cartão Nacional de Saúde**: Input field with value '908634043402417'.
- Data Nascimento**: Input field with value '25/03/1988'.
- Endereço**: Input field with value 'Santa Efigênia Belo Horizonte - MG CEP: 30260-900'.
- Unidade Básica de Saúde \***: Dropdown menu with value 'UBS - 1'.
- Endereço UBS**: Input field with value 'Avenida 10 Lote 20'.
- Data Hora da Consulta \***: Input field with value '26/10/2021 17:00'.
- Especialidade \***: Input field with value 'Endocrinologista'.

At the bottom of the form are two buttons: 'Limpar' (grey) and 'Agendar' (blue). A 'Voltar' button is located at the bottom right of the page.

### Caso de Uso: Alterar Agendamento Médico

Descrição: Este caso de uso deve permitir os cidadãos do município realizar uma alteração no agendamento de consulta médica. Deve ser disponibilizado uma consulta por CPF onde será exibido todos os agendamentos do cidadão e assim possa escolher qual registro se quer alterar. Será permitido alterar os seguintes campos: unidade básica de saúde, data hora da consulta e especialidade. O acesso a funcionalidade deve ser permitido a todo cidadão devidamente autenticado no sistema.

ID	7
História do Usuário	Alterar Agendamento Médico
Eu como	Cidadão do município
Quero	Alterar um agendamento médico
Para que	Possa corrigir um possível erro de digitação ou atualizar as informações do agendamento médico
Prioridade	Alta

### Protótipo:

SGM - Sistema de Gestão Integrada Municipal

Agendamento Médico

Realizar Agendamento Médico em Unidade Básica de Saúde

CPF \*  
010.202.171-10

Nome  
João Silva Filho

Cartão Nacional de Saúde  
9098040302207

Data Nascimento  
03/05/1988

Endereço  
Santa Efigênia Belo Horizonte - MG CEP: 30260-800

Unidade Básica de Saúde \*  
UBS - 1

Endereço UBS  
Avenida 10, Lote 20

Data Hora da Consulta \*  
26/10/2021 17:00

Especialidade \*  
Endocrinologista

Limpar Alterar

Voltar

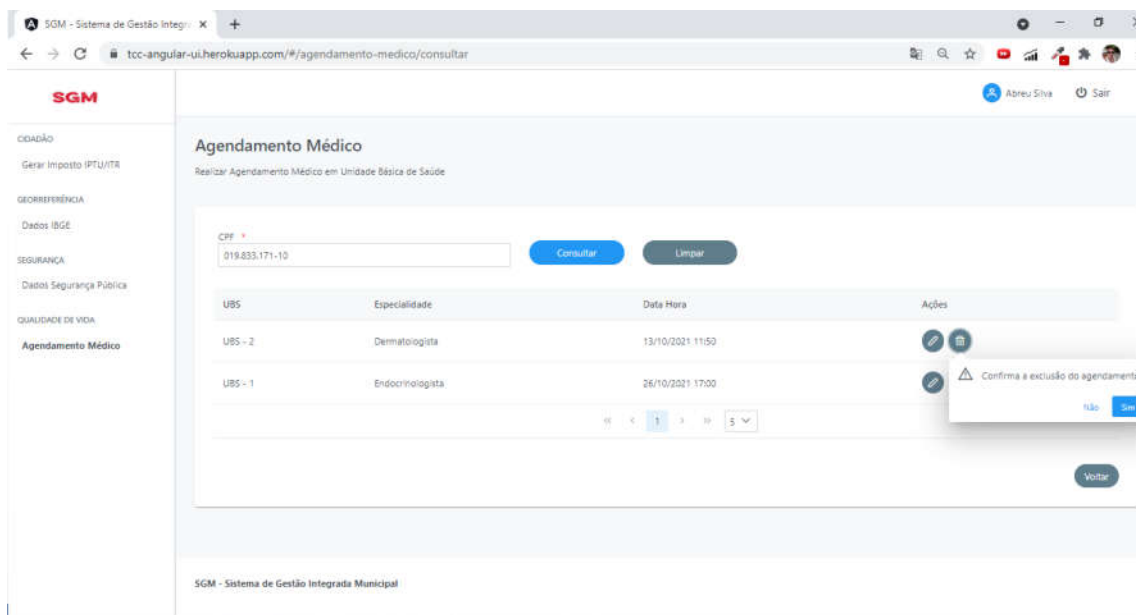
SGM - Sistema de Gestão Integrada Municipal

### Caso de Uso: Excluir Agendamento Médico

Descrição: Este caso de uso deve permitir os cidadãos do município excluir um agendamento de consulta médica. Deve ser disponibilizado uma consulta por CPF onde será exibido todos os agendamentos do cidadão e assim possa escolher qual registro se quer excluir. O acesso a funcionalidade deve ser permitido a todo cidadão devidamente autenticado no sistema.

ID	8
História do Usuário	Excluir Unidade Básica de Saúde
Eu como	Cidadão do município
Quero	Excluir um agendamento médico
Para que	Possa excluir agendamento médico que não será possível comparecer.
Prioridade	Alta

### Protótipo:



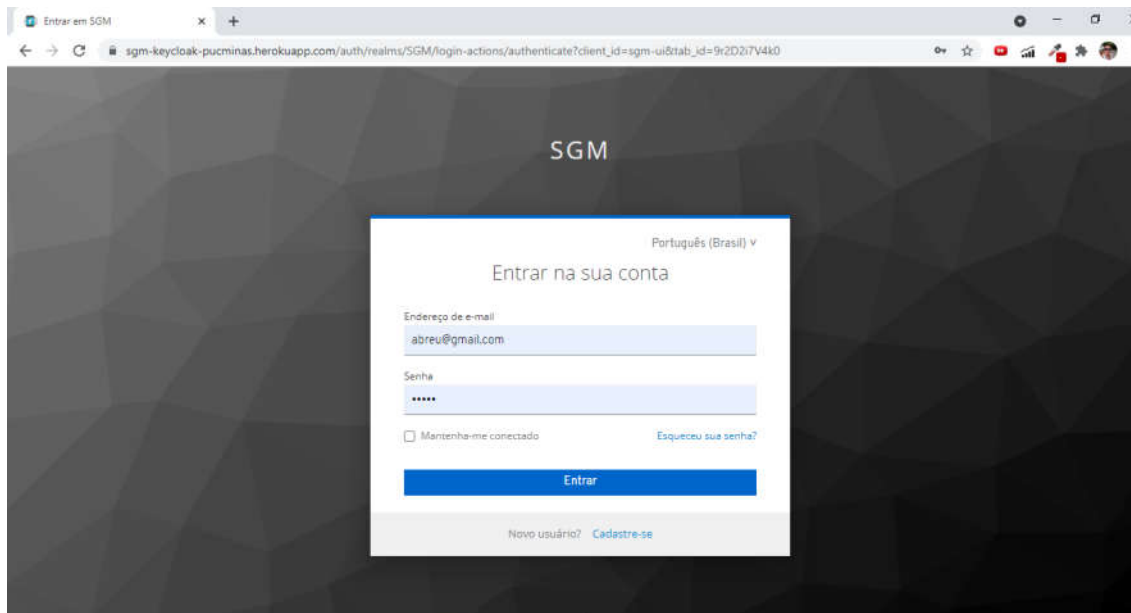
- **Módulo Acesso**

**Caso de Uso: Realizar Autenticação e Autorização**

Descrição: Este caso de uso deve permitir os cidadãos do município acessar a aplicação. Todo cidadão do município deverá realizar autenticação para acessar a aplicação, caso o mesmo não tenha conta será possível realizar o cadastro no link: Cadastre-se. O acesso a funcionalidade deve ser permitido a todo cidadão do município.

ID	9
História do Usuário	Efetuar Login
Eu como	Cidadão do município
Quero	Realizar autenticação no sistema
Para que	Seja possível utilizar as funcionalidades do sistema
Prioridade	Alta

Protótipo:



## ▪ Módulo serviço ao cidadão

### Caso de Uso: Gerar imposto IPTU / ITR

Descrição: Este caso de uso deve permitir os cidadãos do município gerar imposto IPTU/ITR. Deve ser disponibilizado os seguintes campos de preenchimento obrigatório: CPF, tipo pessoa, cep e nº. O sistema deve carregar a informação do endereço conforme o cep digitado, deverá ser feito uma consulta no sistema dos correios. Ao clicar em gerar imposto o sistema deve ser comunicar com o sistema STUR para que possa colher informações e gerar o boleto de pagamento do imposto. O acesso a funcionalidade deve ser permitido a todo cidadão devidamente autenticado no sistema.

ID	10
História do Usuário	Gerar imposto IPTU/ITR
Eu como	Cidadão do município
Quero	Gerar boleto de pagamento do imposto
Para que	Baixar o boleto de pagamento do imposto
Prioridade	Alta

### Protótipo:

The screenshot displays the 'Gerar Imposto IPTU/ITR' form within the SGM web application. The interface includes a sidebar with navigation links: 'CIDADÃO', 'Gerar Imposto IPTU/ITR', 'GEORREFERÊNCIA', 'Dados IBGE', 'SEGURANÇA', 'Dados Segurança Pública', 'QUALIDADE DE VIDA', and 'Agendamento Médico'. The main form area is titled 'Gerar Imposto IPTU/ITR' and 'Gerar boleto de pagamento do imposto'. It features a radio button selection for 'CPF' (selected) and 'CNPJ'. The 'CPF' field contains '019.833.171-10'. The 'Tipo Pessoa' dropdown is set to 'Física'. The 'Cep' field contains '72220-255'. The 'nº' field contains '14'. The 'Endereço' field displays 'QNV 24 Conjunto O - Celândia Sul (Celândia), DF - Brasília'. At the bottom of the form are two buttons: 'Limpar' and 'Gerar Imposto'. The footer of the application reads 'SGM - Sistema de Gestão Integrada Municipal'.




1 / 1 100% +

PDF viewer toolbar

PREFEITURA MUNICIPAL BOM DESTINO - IPTU

Pagador	João de Sousa
CEP	72220255
Data Geração	04/10/2021
Data Vencimento	09/10/2021
Valor Multa	R\$ 300,50
Valor Documento	R\$ 1.500,99

IPTU



SGM - Sistema de Gestão Integrada x ITR (18).pdf

tcc-angular-ui.herokuapp.com/#/gerar-imposto

SGM

Abreu Silva Sair

GERAÇÃO

Gerar Imposto IPTU/ITR

GEORREFERÊNCIA

Dados IBGE

SEGURANÇA

Dados Segurança Pública

QUALIDADE DE VIDA

Agendamento Médico

Gerar Imposto IPTU/ITR

Gerar boleto de pagamento do imposto

☒ CPF ☐ CNPJ

CPF \* 019.833.171-10

Tipo Pessoa \* Rural

Cep \* 72220-255

Nº \* 14

Endereço QVW 24 Conjunto D - Cerâmica Sul (Cerdândia) - DF - Brasília

Limpar Gerar Imposto

SGM - Sistema de Gestão Integrada Municipal


1 / 1 100% +

PDF viewer toolbar

PREFEITURA MUNICIPAL BOM DESTINO - ITR

Pagador	João de Sousa
CEP	72220255
Data Geração	04/10/2021
Data Vencimento	09/10/2021
Valor Multa	R\$ 300,50
Valor Documento	R\$ 1.500,99

ITR



- **Módulo Segurança**

### Caso de Uso: Cadastrar Denúncia

Descrição: Este caso de uso deve permitir os cidadãos do município realizar o cadastro de denúncias de vários tipos. Deve ser disponibilizado os seguintes campos de preenchimento obrigatório: Tipo Denúncia, Descrição, Endereço da Ocorrência. O acesso a funcionalidade deve ser permitido a todo cidadão devidamente autenticado no sistema.

ID	11
História do Usuário	Cadastrar Denúncia
Eu como	Cidadão do município
Quero	Cadastrar uma denúncia referente ao município
Para que	Seja possível notificar as autoridades do município
Prioridade	Média

### Protótipo:

The screenshot displays a web browser window with the URL `tcc-angular-ui.herokuapp.com/#/denuncia`. The application interface includes a sidebar menu on the left with the following items: CIDADÃO, Gerar Imposto IPTU/ITR, GEORREFERÊNCIA, Dados ISGE, SEGURANÇA, **Cadastrar Denúncia** (highlighted), QUALIDADE DE VIDA, and Agendamento Médico. The main content area is titled 'Cadastrar Denúncia' with the subtitle 'Realizar o cadastro de Denúncia'. The form contains three required fields: 'Tipo Denúncia' (a dropdown menu with 'Selecione' as the current selection), 'Descrição' (a text area), and 'Endereço da Ocorrência' (a text field). At the bottom of the form are two buttons: 'Limpar' (grey) and 'Salvar' (blue). The footer of the application reads 'SGM - Sistema de Gestão Integrada Municipal' and 'Prefeitura Municipal de Bom Destino'.

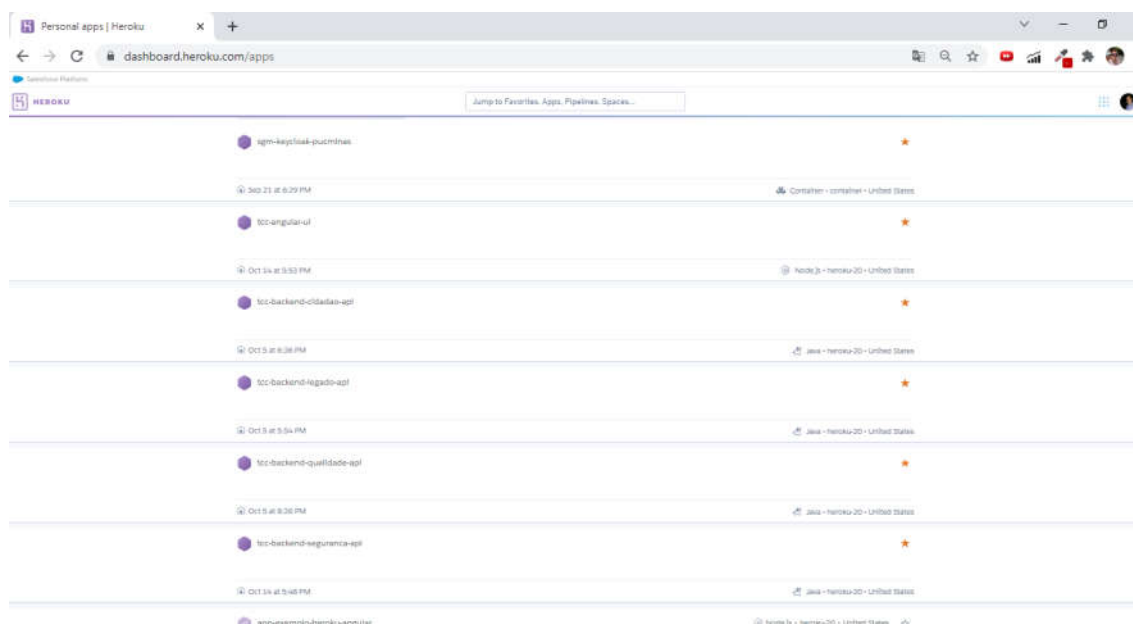
### 5.1.3 Tecnologias Utilizadas

Caso de Uso	Tecnologias
Efetuar Login	Keycloak
Consultar Unidade Básica de Saúde	Keycloak, Angular, PrimeNG, REST, Java, Postgre e Spring boot
Cadastrar Unidade Básica de Saúde	Keycloak, Angular, PrimeNG, REST, Java, Postgre e Spring boot
Alterar Unidade Básica de Saúde	Keycloak, Angular, PrimeNG, REST, Java, Postgre e Spring boot
Excluir Unidade Básica de Saúde	Keycloak, Angular, PrimeNG, REST, Java, Postgre e Spring boot
Consultar Agendamento Médico	Keycloak, Angular, PrimeNG, REST, Java, Postgre e Spring boot
Cadastrar Agendamento Médico	Keycloak, Angular, PrimeNG, REST, Java, Postgre, Mule, Spring Cloud e Spring boot
Alterar Agendamento Médico	Keycloak, Angular, PrimeNG, REST, Java, Postgre, Mule, Spring Cloud e Spring boot
Excluir Agendamento Médico	Keycloak, Angular, PrimeNG, REST, Java, Postgre e Spring boot
Gerar Imposto IPTU / ITR	Keycloak, Angular, PrimeNG, REST, Java, Mysql, Mule, Spring Cloud, Spring boot e Jasper Reports
Cadastrar Denúncia	Keycloak, Angular, PrimeNG, REST, Java, Mysql e Spring boot

### 5.1.4 Implantação

Toda a prova de conceito desenvolvida foi implantada na nuvem, foi utilizado Heroku para realizar a hospedagem dos serviços. A tabela a seguir detalha a implementação:

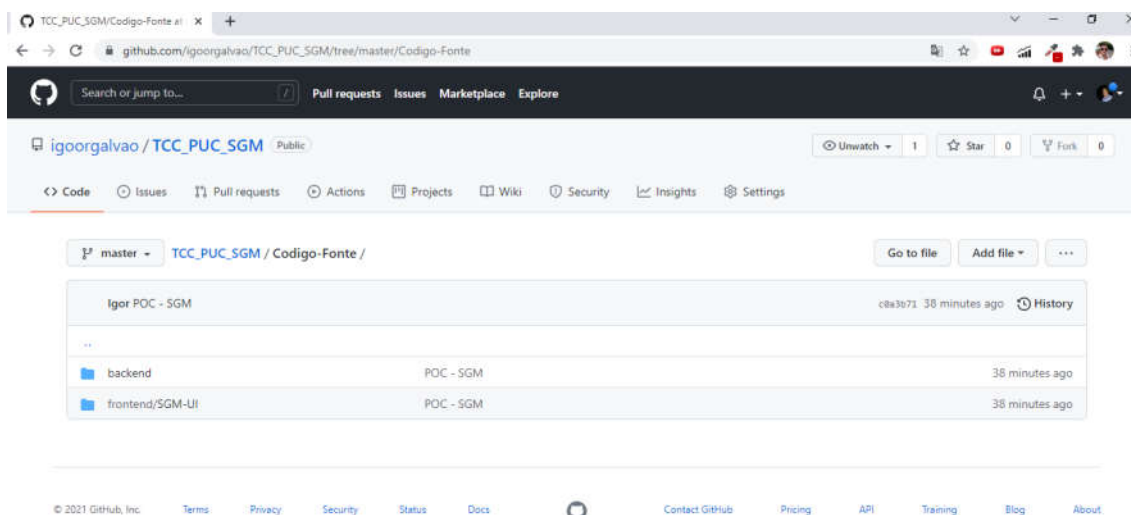
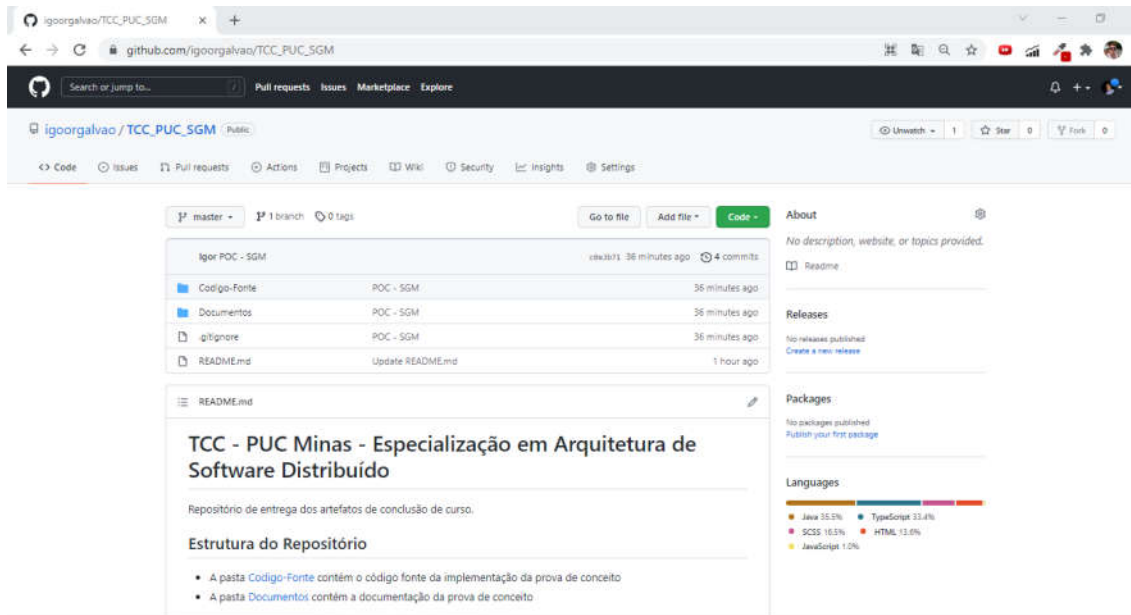
Nome	Componente	Fra- mework	Complemento
sgm-keycloak-pucminas	Keycloak	Container	Heroku-postgresql
tcc-angular-ui	Angular (Front-end)	Node.js	Nenhum
tcc-backend-cidadao-api	Service Cidadão (back-end)	Java	ClearDB Mysql
tcc-backend-seguranca-api	Service Segurança (back-end)	Java	ClearDB Mysql
tcc-backend-qualidade-api	Service qualidade de vida (back-end)	Java	Heroku-postgresql
tcc-backend-legado-api	Service legado	Java	Nenhum

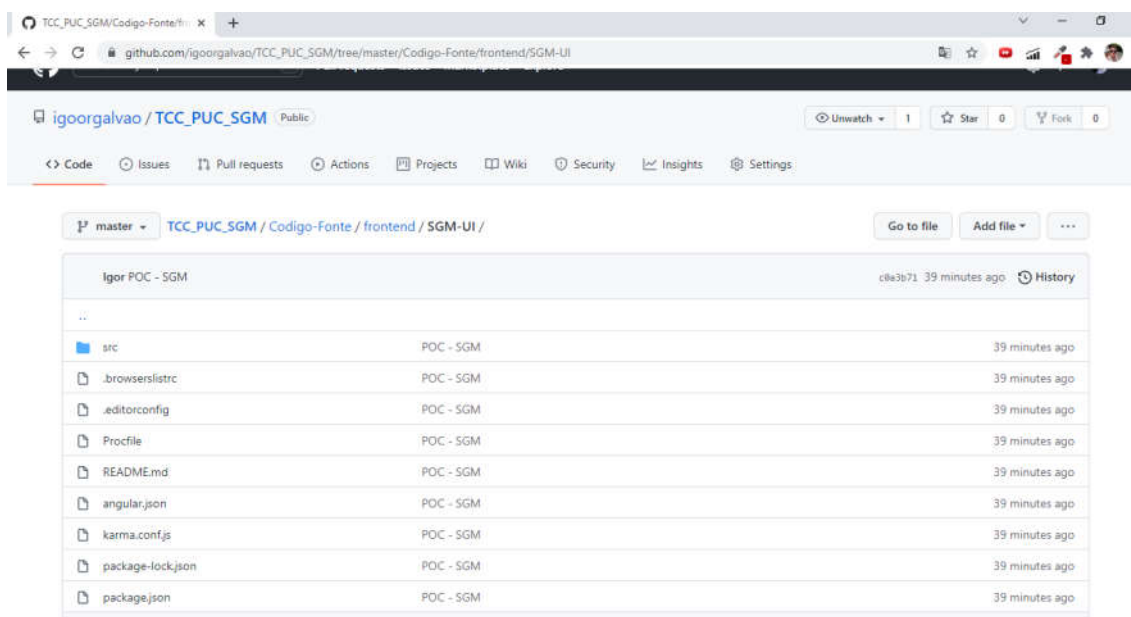
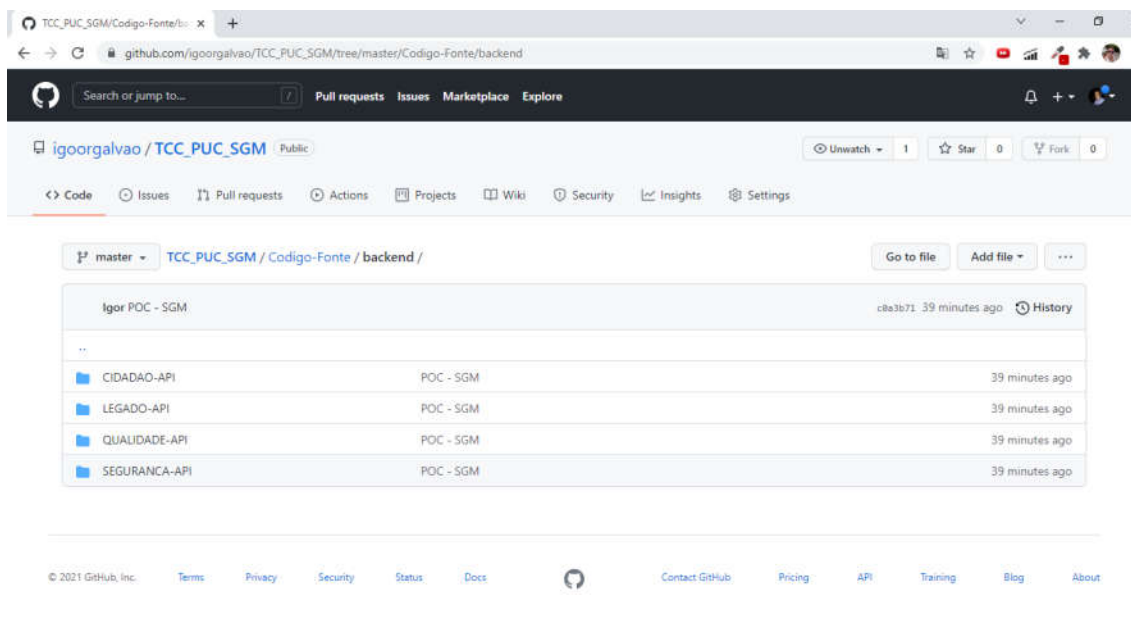


Os links para acesso a plataforma encontram-se no apêndice desse documento.

### 5.1.5 Repositório de Código

Todo o código fonte foi disponibilizado no Github. As imagens a seguir evidenciam o repositório do código fonte.





Os links para acesso ao Git encontram-se no apêndice desse documento.

## 5.2. Interfaces/ APIs

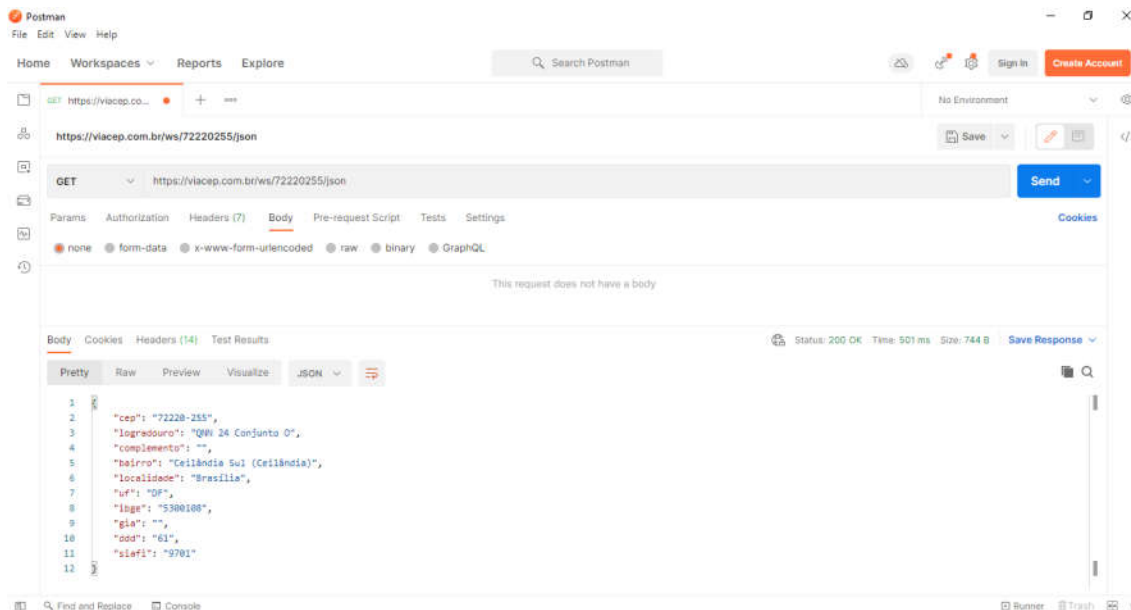
### 5.2.1. Consumidas

#### Interface: Correios (Via Cep)

Para a implementação da prova de conceito foi realizado a simulação de um consumo de API dos correios baseado no protocolo REST recebendo um JSON. A interface tem como objetivo fornecer o endereço baseado no cep informado. Para acessar a interface basta utilizar a URI: `viacep.com.br/ws/{CEP}/{Formato}` utilizando o método GET.

Entrada			
Parâmetro	Tipo	Obrigatoriedade	Descrição
CEP	String	Sim	Número do cep sem hífen
Formato	String	Sim	Tipo de retorno (JSON, XML)

Retorno		
Parâmetro	Tipo	Descrição
CEP	String	CEP com formatação
Logradouro	String	Logradouro do CEP pesquisado
Complemento	String	Complemento do CEP pesquisado
Bairro	String	Bairro do CEP pesquisado
Localidade	String	Localidade do CEP pesquisado
UF	String	UF do CEP pesquisado
IBGE	String	Código IBGE do CEP pesquisado
Gia	String	Gia do código pesquisado, utilizado somente para UF de São Paulo
DDD	String	DDD do Cep pesquisado
Siafi	String	Código Siafi do CEP pesquisado



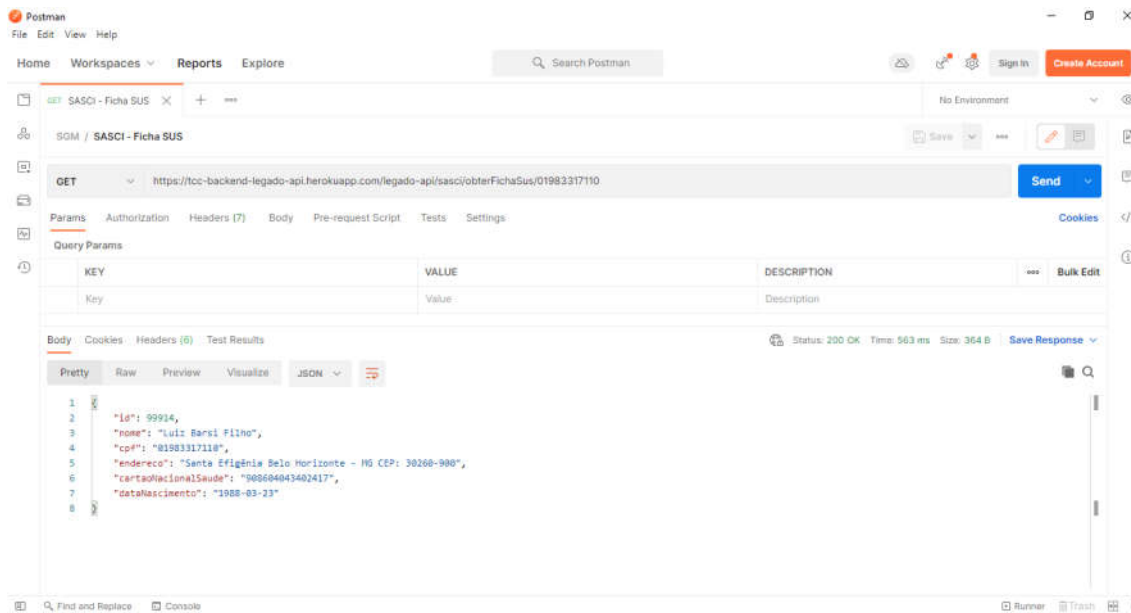
### Interface: SASCi (legado)

Para a implementação da prova de conceito foi realizado a simulação de um consumo de API de um sistema legado, o SASCi, baseado no protocolo REST recebendo um XML. A interface tem como objetivo fornecer o prontuário eletrônico do SUS do CPF informado. Para acessar a interface basta utilizar a URI: `tcc-backend-legado-api.herokuapp.com/legado-api/sasci/obterFichaSus/{CPF}` utilizando o método GET.

Entrada			
Parâmetro	Tipo	Obrigatoriedade	Descrição
CPF	String	Sim	CPF do paciente sem formatação

Retorno		
Parâmetro	Tipo	Descrição
ID	Inteiro	Identificador único do prontuário eletrônico do SUS
Nome	String	Nome do paciente
Endereço	String	Endereço do paciente
Cartão nacional de Saúde	String	Número do cartão nacional de saúde do paciente
Data de nascimento	Date	Data de nascimento do paciente





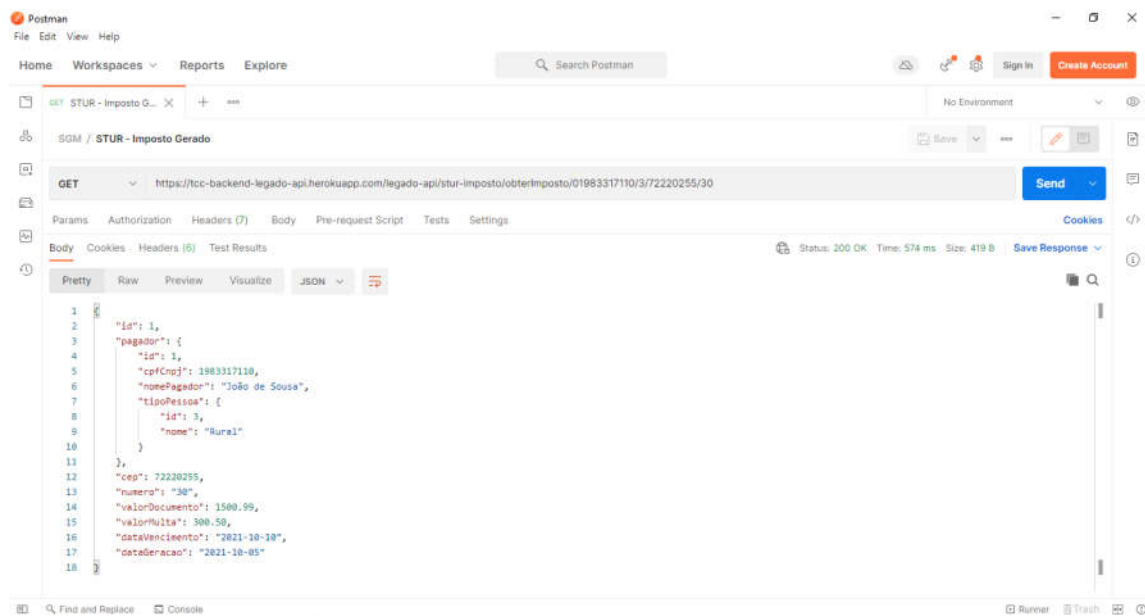
### Interface: STUR (legado)

Para a implementação da prova de conceito foi realizado a simulação de um consumo de API de um sistema legado, o STUR, baseado no protocolo REST recebendo um XML. A interface tem como objetivo fornecer o processamento do imposto municipal do cidadão. Para acessar a interface basta utilizar a URI: `tcc-backend-legado-api.herokuapp.com/legado-api/stur-imposto/obterImposto/${CPF/CNPJ}/${TIPO_PESSOA}/${CEP}/${NUMERO}` utilizando o método GET.

Entrada			
Parâmetro	Tipo	Obrigatoriedade	Descrição
CPFCNPJ	String	Sim	Número do CPF ou CNPJ do cidadão
TIPO_PESSOA	Inteiro	Sim	Tipo pessoa: 1- Física, 2- Jurídica e 3- Rural
CEP	Inteiro	Sim	Número do cep sem hífen
NUMERO	String	Sim	Número da residência do cidadão

Retorno		
Parâmetro	Tipo	Descrição
ID	Inteiro	Identificador único do imposto municipal

IdPagador	Inteiro	Identificador único do cidadão pagador
CpfCnpj	Inteiro	CPF ou CNPJ do cidadão pagador
NomePagador	String	Nome do cidadão pagador
IdTipoPessoa	Inteiro	Identificador único do tipo pessoa
Nome	String	Descrição do tipo pessoa
Cep	Inteiro	Cep do cidadão pagador
Numero	String	Número da residência do cidadão pagador
ValorDocumento	Decimal	Valor do imposto gerado
ValorMulta	Decimal	Valor da multa do imposto gerado
DataVencimento	Date	Data de vencimento do imposto gerado
DataGeracao	Date	Data de geração do imposto gerado

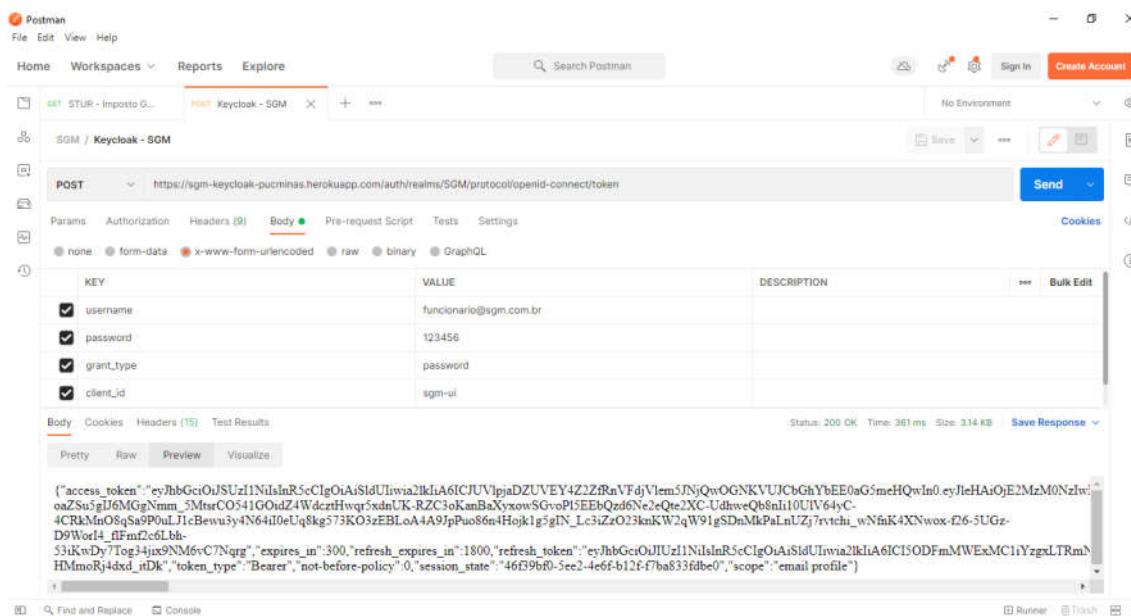


### Interface: Autenticação

Para a implementação da prova de conceito foi realizado a simulação de um consumo de API de autenticação, o Keycloak, baseado no protocolo REST. A interface tem como objetivo fornecer um token de acesso ao sistema. Para acessar a interface basta utilizar a URI: `sgm-keycloak-pucminas.herokuapp.com/auth/realms/SGM/protocol/openid-connect/token` utilizando o método POST.

Entrada			
Parâmetro	Tipo	Obrigatoriedade	Descrição
username	String	Sim	Login de acesso ao sistema. Utilizado o e-mail do cidadão
password	String	Sim	Senha de acesso ao sistema.
grant_type	String	Sim	Maneira que o Keycloak possui para obter o acesso
client_id	String	Sim	Nome do cliente da aplicação cadastrada no Keycloak

Retorno		
Parâmetro	Tipo	Descrição
access_token	String	Token de acesso a aplicação
expires_in	String	Tempo de vida do token
refresh_expires_in	String	Tempo de vida da atualização do token
refresh_token	String	Token de atualização
token_type	String	Tipo do token
not-before-policy	Inteiro	Política de segurança para invalidar tokens
session_state	String	Estado da sessão
scope	String	Escopo do token

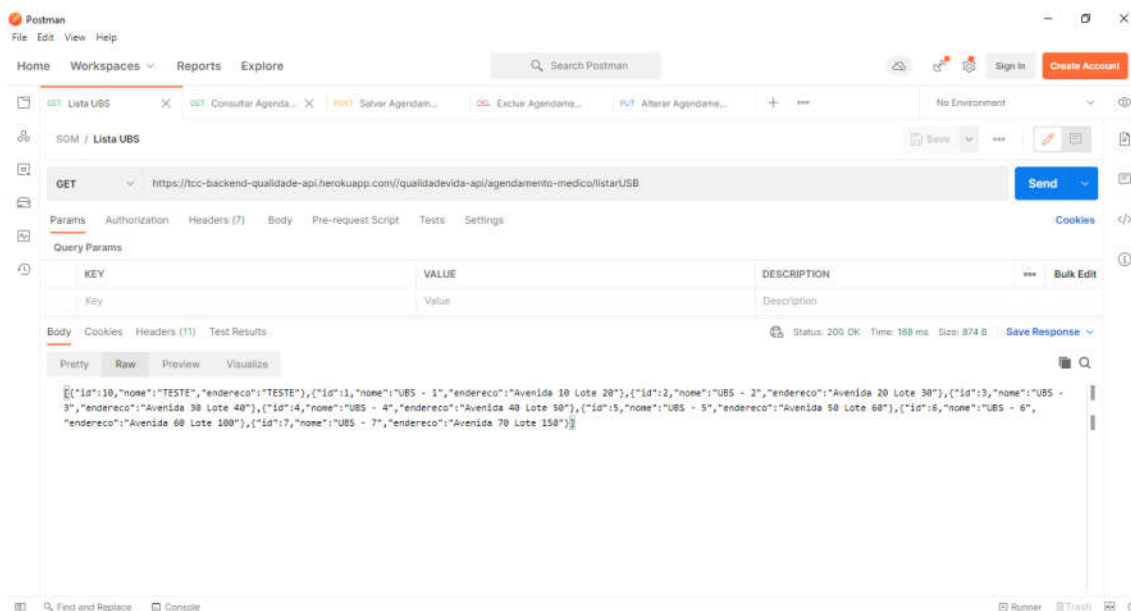


### 5.2.2. Fornecidas

#### Interface: Listar unidade básica de saúde (UBS)

Para a implementação da prova de conceito foi criada a API de listar unidade básica de saúde, baseada no protocolo REST retornando um JSON. A interface tem como objetivo fornecer a lista de UBS cadastrada no município. Para acessar a interface basta utilizar a URI: `tcc-backend-qualidade-api.herokuapp.com//qualidadevida-api/agendamento-medico/listarUSB` utilizando método GET.

Retorno		
Parâmetro	Tipo	Descrição
Id	Inteiro	Identificador único da UBS
Nome	String	Nome da UBS
Endereço	String	Endereço da UBS



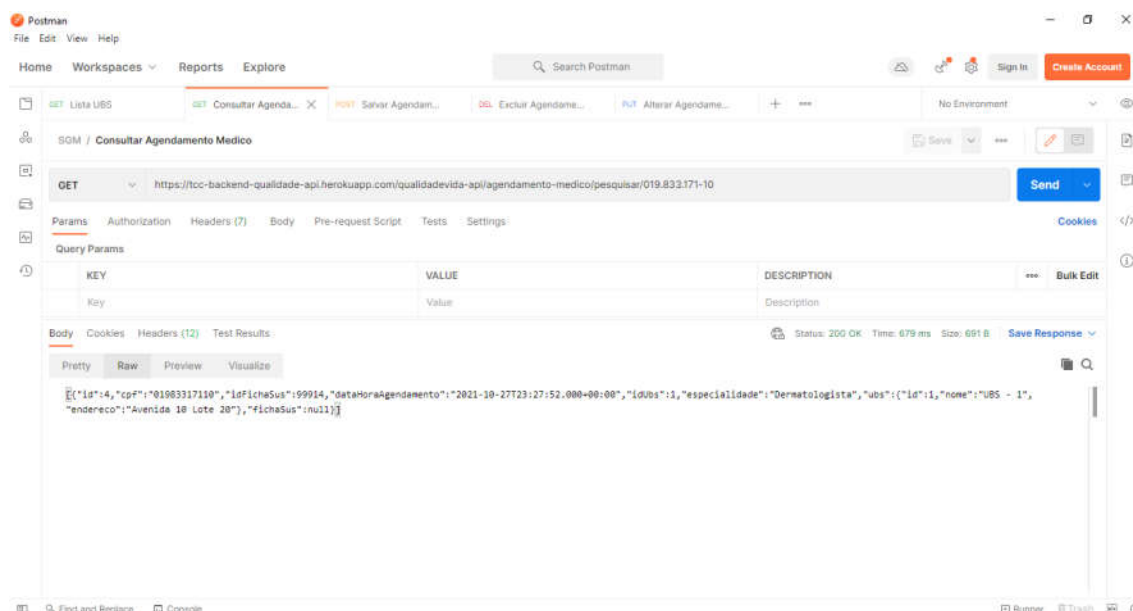
### Interface: Consultar agendamento médico

Para a implementação da prova de conceito foi criada a API de consultar agendamento médico, baseada no protocolo REST retornando um JSON. A interface tem como objetivo fornecer a lista de agendamento médicos de um paciente. Para acessar a interface basta utilizar a URI: `tcc-backend-qualidade-api.herokuapp.com/qualidadevida-api/agendamento-medico/pesquisar/{CPF}` utilizando método GET.

Entrada			
Parâmetro	Tipo	Obrigatoriedade	Descrição
CPF	String	Sim	CPF do paciente sem formatação

Retorno		
Parâmetro	Tipo	Descrição
ID	Inteiro	Identificador único do agendamento médico
CPF	String	CPF do paciente
IdFichaSus	Inteiro	Identificador único do prontuário do SUS
DataHoraAgendamento	Date	Data e hora do agendamento da consulta

IdUbs	Inteiro	Identificador único da UBS escolhida
Especialidade	String	Especialidade escolhida pelo paciente

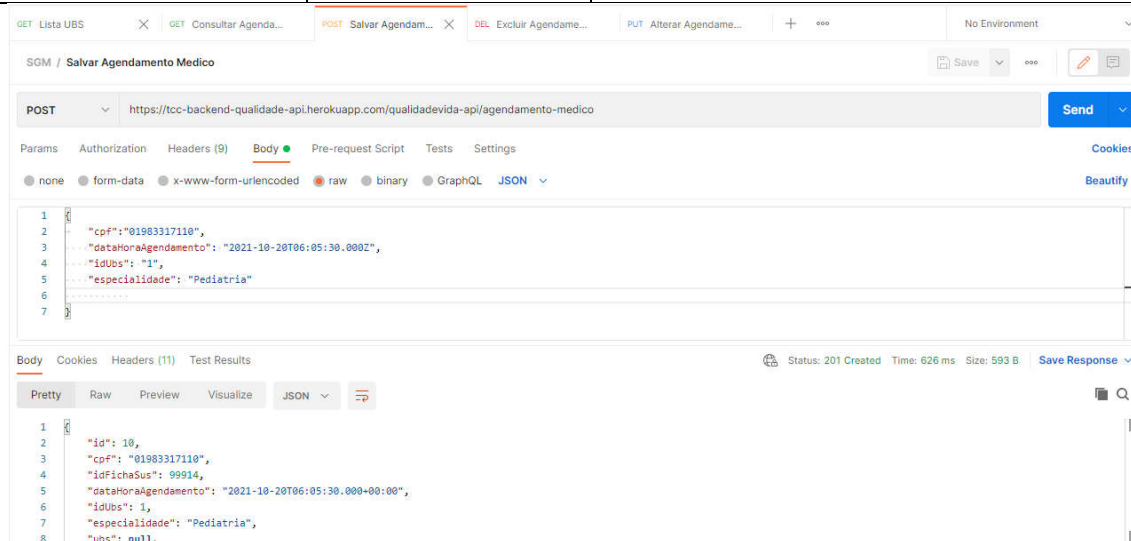


### Interface: Realizar agendamento médico

Para a implementação da prova de conceito foi criada a API de realizar agendamento médico, baseada no protocolo REST. A interface tem como objetivo realizar o agendamento médico do paciente. Para acessar a interface basta utilizar a URI: `tcc-backend-qualidade-api.herokuapp.com/qualidadevida-api/agendamento-medico` utilizando método POST.

Entrada			
Parâmetro	Tipo	Obrigatoriedade	Descrição
CPF	String	Sim	CPF do paciente sem formatação
DataHoraAgendamento	Date	Sim	Data e hora do agendamento médico
IdUbs	Inteiro	Sim	Identificador único da UBS
Especialidade	String	Sim	Especialidade a marcar o agendamento

Retorno		
Parâmetro	Tipo	Descrição
ID	Inteiro	Identificador único do agendamento médico
CPF	String	CPF do paciente
IdFichaSus	Inteiro	Identificador único do prontuário do SUS
DataHoraAgendamento	Date	Data e hora do agendamento da consulta
IdUbs	Inteiro	Identificador único da UBS escolhida
Especialidade	String	Especialidade escolhida pelo paciente



### Interface: Alterar agendamento médico

Para a implementação da prova de conceito foi criada a API de alterar agendamento médico, baseada no protocolo REST. A interface tem como objetivo alterar um agendamento médico do paciente. Para acessar a interface basta utilizar a URI: `tcc-backend-qualidade-api.herokuapp.com/qualidadevida-api/agendamento-medico` utilizando método PUT.

Entrada			
Parâmetro	Tipo	Obrigatoriedade	Descrição
ID	Inteiro	Sim	Identificador único do agendamento a ser alterado

CPF	String	Sim	CPF do paciente sem formatação
DataHoraAgendamento	Date	Sim	Data e hora do agendamento médico a ser alterado
IdUbs	Inteiro	Sim	Identificador único da UBS a ser alterado
Especialidade	String	Sim	Especialidade a ser alterada

Retorno		
Parâmetro	Tipo	Descrição
ID	Inteiro	Identificador único do agendamento médico
CPF	String	CPF do paciente
IdFichaSus	Inteiro	Identificador único do prontuário do SUS
DataHoraAgendamento	Date	Data e hora do agendamento da consulta
IdUbs	Inteiro	Identificador único da UBS escolhida
Especialidade	String	Especialidade escolhida pelo paciente

GET Lista UBS   GET Consultar Agenda...   POST Salvar Agendame...   DEL Excluir Agendame...   PUT Alterar Agendame... X   +   No Environment

SGM / Alterar Agendamento Medico

PUT   https://tcc-backend-qualidade-api.herokuapp.com/qualidadevida-api/agendamento-medico   Send

Params   Authorization   Headers (9)   Body   Pre-request Script   Tests   Settings   Cookies   Beautify

● none   ● form-data   ● x-www-form-urlencoded   ● raw   ● binary   ● GraphQL   JSON

```

1  {
2    "id": 5,
3    "cpf": "01983317110",
4    "dataHoraAgendamento": "2021-10-21T06:05:30.000Z",
5    "idUbs": "3",
6    "especialidade": "Pediatria"
  }

```

Body   Cookies   Headers (11)   Test Results   Status: 200 OK   Time: 190 ms   Size: 587 B   Save Response

Pretty   Raw   Preview   Visualize   JSON

```

1  {
2    "id": 5,
3    "cpf": "01983317110",
4    "idFichaSus": 99914,
5    "dataHoraAgendamento": "2021-10-21T06:05:30.000+00:00",
6    "idUbs": 3,
7    "especialidade": "Pediatria",
8    "ubs": null,
9    "fichaSus": null
10 }

```



### Interface: Excluir agendamento médico

Para a implementação da prova de conceito foi criada a API de excluir um agendamento médico, baseada no protocolo REST. A interface tem como objetivo excluir um agendamento médico do paciente. Para acessar a interface basta utilizar a URI: `tcc-backend-qualidade-api.herokuapp.com/qualidadevida-api/agendamento-medico/${idAgendamento}` utilizando método DELETE.

Entrada			
Parâmetro	Tipo	Obrigatoriedade	Descrição
IdAgendamento	Inteiro	Sim	Identificador único do agendamento a ser excluído

Retorno	
Status	Descrição
Code: 204	Este código significa que o servidor processou a solicitação com sucesso, mas não vai devolver nenhum conteúdo

The screenshot shows a REST client interface with a tab for 'DEL. Excluir Agendame...'. The request method is 'DELETE' and the URL is 'https://tcc-backend-qualidade-api.herokuapp.com/qualidadevida-api/agendamento-medico/5'. The 'Send' button is visible. Below the URL bar, the 'Params' tab is selected, showing a table with columns 'KEY', 'VALUE', and 'DESCRIPTION'. The table contains one row with 'Key' and 'Value'. The 'Body' tab is also visible. At the bottom, the 'Test Results' tab shows the response status: 'Status: 204 No Content', 'Time: 173 ms', and 'Size: 388 B'. The 'Save Response' button is also present.

### Interface: Cadastrar Denúncia

Para a implementação da prova de conceito foi criada a API de cadastrar denúncia, baseada no protocolo REST. A interface tem como objetivo permitir o cadastro de denúncias para notificar as autoridades do município. Para acessar a interface basta utilizar a URI: `tcc-backend-qualidade-api.herokuapp.com/qualidadevida-api/agendamento-medico` utilizando método POST.

Entrada			
Parâmetro	Tipo	Obrigatoriedade	Descrição
TipoDenuncia{id}	Inteiro	Sim	Identificador único do objeto tipo de denúncia
EnderecoOcorrencia	String	Sim	Endereço da ocorrência da denúncia
Descrição	String	Sim	Descrição do que de fato aconteceu na denúncia

Retorno		
Parâmetro	Tipo	Descrição
ID	Inteiro	Identificador único do Objeto Denúncia
TipoDenuncia{id}	Inteiro	Identificador único do objeto tipo de denúncia
TipoDenuncia{descricao}	String	Descrição do objeto tipo de denúncia
EnderecoOcorrencia	String	Endereço da ocorrência da denúncia
Descrição	String	Descrição do que de fato aconteceu na denúncia

The screenshot displays a REST client interface with the following details:

- Request:**
  - Method: **POST**
  - URL: `https://tcc-backend-seguranca-api.herokuapp.com/seguranca-api/denuncia`
  - Body (raw):

```
1 {
2   "tipoDenuncia": {"id": 1},
3   "endereçoOcorrência": "Rua Branca",
4   "descricao": "Denúncia do município"
5 }
```
- Response:**
  - Status: **201 Created**
  - Time: **800 ms**
  - Size: **549 B**
  - Body (pretty):

```
1 {
2   "id": null,
3   "tipoDenuncia": {
4     "id": 1,
5     "descricao": null
6   },
7   "endereçoOcorrência": "Rua Branca",
8   "descricao": "Denúncia do município"
9 }
```

## 6. Avaliação da Arquitetura

### 6.1. Análise das abordagens arquiteturais

A arquitetura proposta neste projeto é baseada em microserviços que trabalham de forma independentes e se comunicam utilizando o protocolo REST. Para a criação telas foram escolhidos os frameworks Angular e o PrimeNG, por serem frameworks de fácil aprendizado, possui uma comunidade muito forte e permitir a arquitetura atingir o requisito não funcional de acessibilidade tornando toda a parte de interfaces responsivas a qualquer dispositivo. Os serviços foram criados baseados em protocolo REST utilizando o framework Spring boot com Java. Todos os componentes propostos neste projeto são possíveis de serem implementados em nuvem.

### 6.2. Cenários

**Cenário 1:** Ao realizar o acesso a uma URL ou página, o sistema deve apresentar altos padrões de segurança necessário, garantindo que o usuário possa acessar as páginas apenas autenticado no sistema. O sistema deve redirecionar o usuário para a tela de autenticação quando o mesmo tentar acessar uma página sem estar autenticado no sistema. O sistema deverá garantir que as páginas públicas possam ser acessadas sem estar autenticado, garantindo assim a segurança e confidencialidade das informações estando em conformidade com o requisito não funcional de segurança.

**Cenário 2:** Ao navegar na tela, o sistema deve apresentar boa usabilidade. A navegação deve apresentar facilidade e o acesso às funcionalidades deve ser intuitivo e objetivo. O usuário deve conseguir realizar um agendamento médico em no máximo 5 minutos, assim garantindo a agilidade e a usabilidade para ficar de acordo com o requisito não funcional de usabilidade.

**Cenário 3:** Ao acessar o sistema através de um dispositivo móvel ou desktop com resolução reduzida, utilizando os browsers principais como IE, Chrome, Firefox e Opera, a tela do usuário deverá se adaptar automaticamente, redimensionando seus links, botões, padrão de cores e tabela de dados de acordo com a resolução. Esta é a garantia de que o requisito não funcional de acessibilidade foi satisfeito.

**Cenário 4:** Ao realizar um acesso em alguma tela do sistema, o sistema deve ter um desempenho aceitável e responder em no máximo 10 segundos a renderização dos objetos na tela. Esta é a garantia de que o requisito não funcional de desempenho foi satisfeito.

Na priorização foi utilizado o método de Árvore de Utilidade reduzida e com prioridades. Foi categorizado de acordo os atributos de qualidade a que estão relacionados e então classificados em função de sua importância e complexidade, considerando a percepção de negócio e arquitetura. As duas variáveis de priorização "Importância" (IMP) e "Complexidade" (COM), foram classificadas em alta (A), média (M) e baixa (B) de acordo com as características do requisito.

Atributos de Qualidade		Cenários	IMP.	COM.
Funcionalidade	Segurança	Cenário 1: O sistema deve apresentar altos padrões de segurança	A	M
	Usabilidade	Cenário 2: O sistema deve prover boa usabilidade.	M	B
	Acessibilidade	Cenário 3: O sistema deve suportar ambientes web responsivos e ambientes móveis	M	A
Eficiência	Desempenho	Cenário 4: O sistema deve ser rápido	A	A

### 6.3. Avaliação

Processo de avaliação dos cenários identificados item [6.3](#) são analisados. O objetivo é determinar os riscos, não riscos, pontos de sensibilidade e tradeoffs e as evidências mostrando o requisito de qualidade sendo atendido.

**Cenário 1**

Atributo de Qualidade:	Segurança
Requisito de Qualidade:	O sistema deve apresentar altos padrões de segurança
Preocupação	
Impossibilitar o acesso a páginas privadas do sistema sem autenticação no sistema	
Cenário (s):	
Cenário 1	
Ambiente	
Sistema em operação normal	
Estímulo:	
Usuário tenta acessar uma página privada do sistema sem estar autenticado no sistema.	
Mecanismo:	
Criar um mecanismo de validação de credenciais e suas permissões associadas para acessar recursos protegidos.	
Medida de resposta:	
O usuário deve ser redirecionado para tela de autenticação	
Considerações sobre a arquitetura:	
Riscos:	O gerenciamento de sessões e de autenticação apropriados são críticos para segurança web. Falhas nessa área frequentemente envolvem falha ao proteger credenciais e sessões durante o ciclo de vida. A utilização do autenticador Keycloak é uma maneira simples de proteger as API's de acesso indevidos.
Pontos de Sensibilidade:	Servidor de aplicação operando em modo HTTPS
Tradeoff:	Não existe

## Evidências do cenário 1

Usuário acessando a página pública do sistema SGM.



O cidadão efetua autenticação no sistema para acesso as páginas privadas do sistema. O sistema gera um token de acesso onde permite o usuário navegar entre as páginas privadas do sistema, se o mesmo tentar acessar alguma página privada sem um token de acesso válido o sistema irá redirecionar para a página de login.

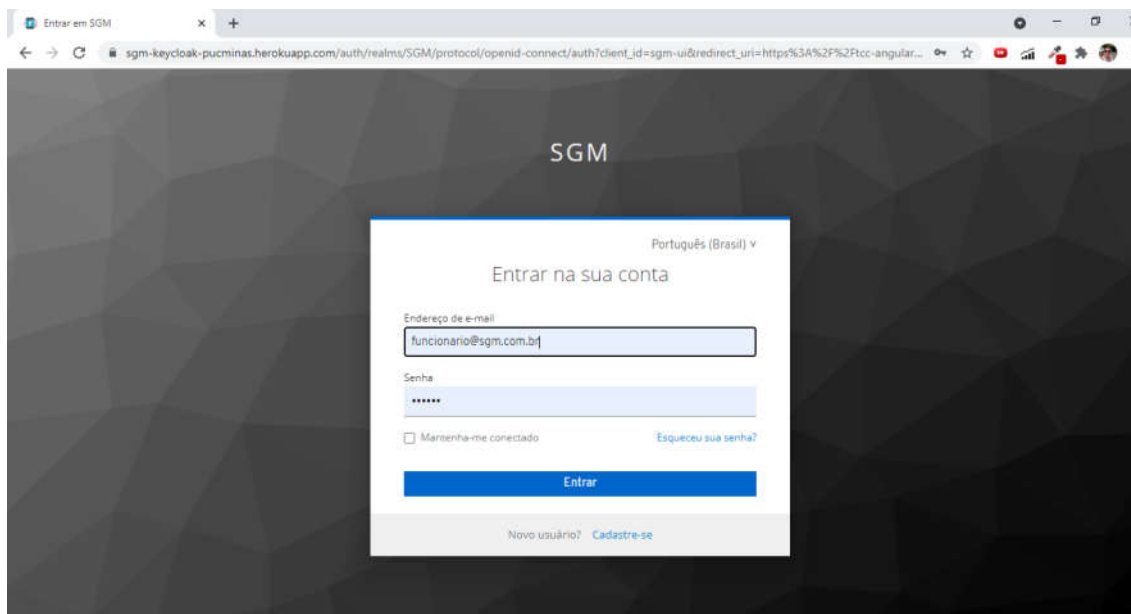
A captura de tela mostra a página de login do sistema SGM. O formulário "Entrar na sua conta" está centralizado sobre um fundo escuro com uma textura geométrica. O formulário contém:

- Seletor de idioma: "Português (Brasil)" com uma seta para baixo.
- Título: "Entrar na sua conta".
- Campo de "Endereço de e-mail" com o valor "abreu@gmail.com".
- Campo de "Senha" com caracteres ocultos por pontos.
- Caixa de seleção "Mantenha-me conectado" desmarcada.
- Link "Esqueceu sua senha?".
- Botão "Entrar" em azul.
- Links "Novo usuário?" e "Cadastre-se" na base.

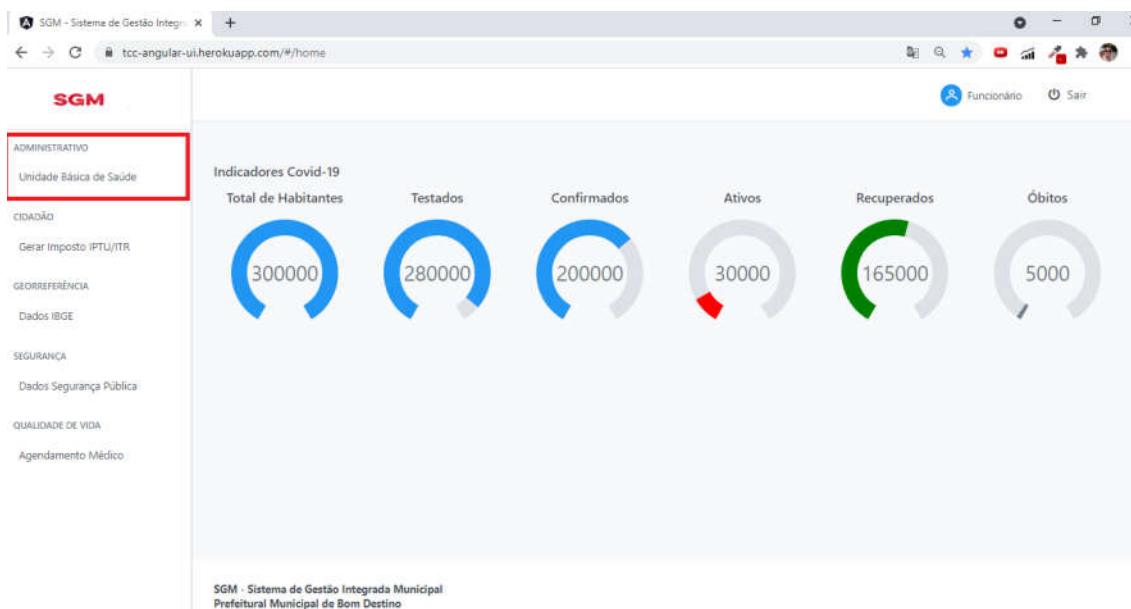




Um funcionário efetua autenticação no sistema para acesso as páginas privadas do sistema. O sistema gera um token de acesso onde permite o funcionário navegar entre as páginas privadas do sistema, se o mesmo tentar acessar alguma página privada sem um token de acesso válido o sistema irá redirecionar para a página de login.



Um funcionário do município consegue realizar o login e recebe um token de acesso que irá permitir navegar nas páginas privadas do sistema. O acesso feito pelo funcionário permite acessar funcionalidade administrativa do município.

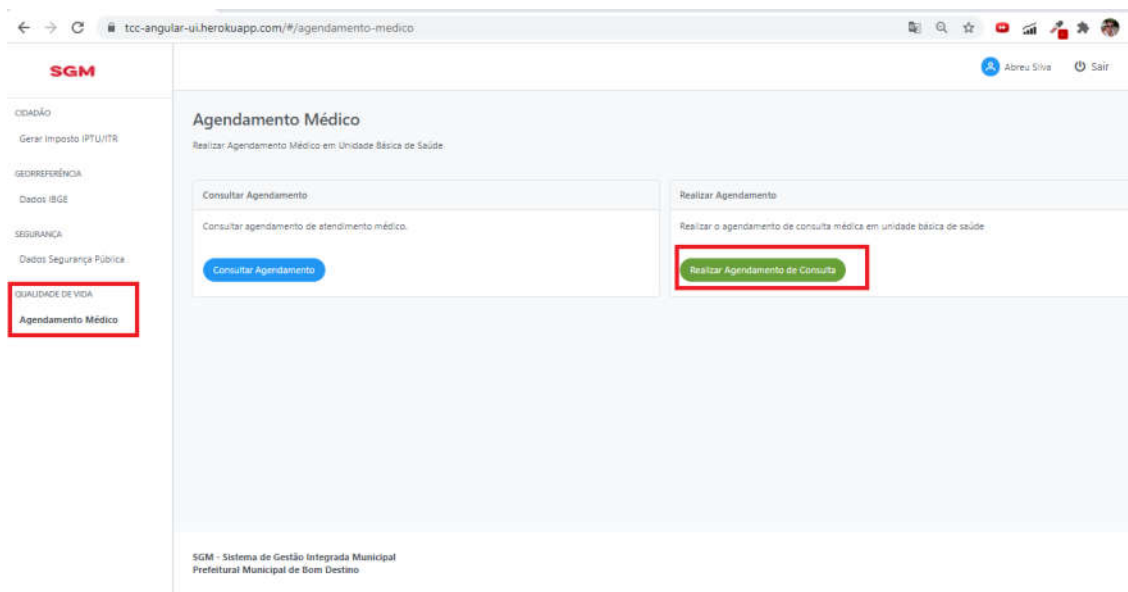


## Cenário 2

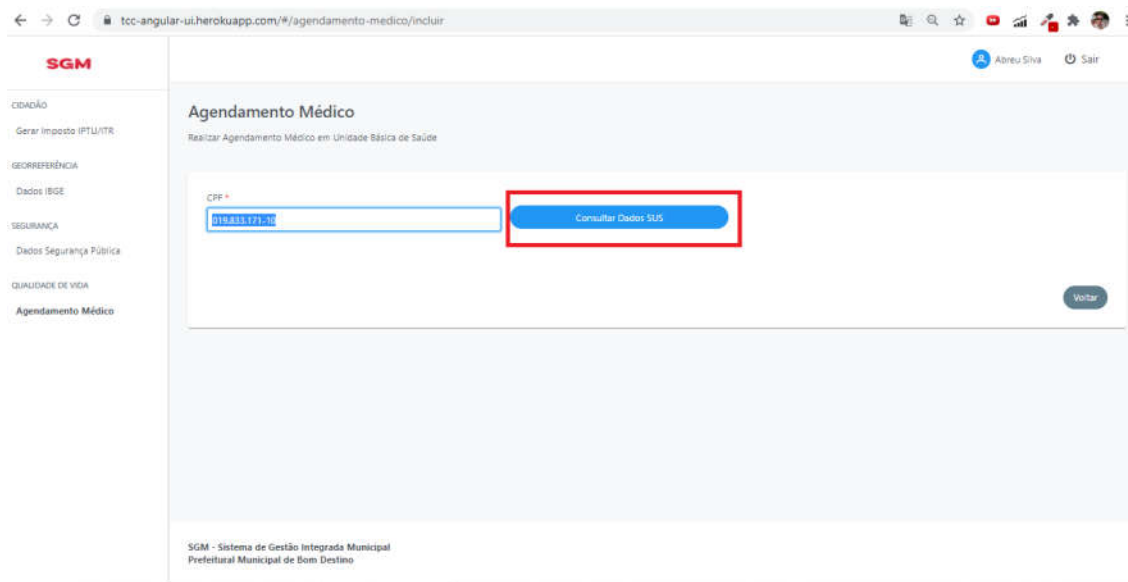
Atributo de Qualidade:	Usabilidade
Requisito de Qualidade:	O sistema deve prover boa usabilidade.
Preocupação	
O sistema deve fornecer interfaces simples de navegação e tornar a experiência do usuário rápida e objetiva.	
Cenário (s):	
Cenário 2	
Ambiente	
Sistema em operação normal	
Estímulo:	
Usuário navegando no site e efetuando um agendamento médico, devendo apresentar objetividade e um agendamento médico rápido de no máximo 5 minutos.	
Mecanismo:	
Telas simples e objetivas sem muitos componentes a serem carregados, possibilitando o servidor de aplicação a responder de forma rápida as requisições. O sistema trabalha com template padrão para todas as telas, assim alterando apenas o conteúdo principal.	
Medida de resposta:	
O cidadão não deve demorar mais que 5 minutos para efetuar um agendamento médico e no final o sistema deve exibir uma mensagem de confirmação ao cidadão.	
Considerações sobre a arquitetura:	
Riscos:	Pode ocorrer um pico de memória no servidor ou um número de usuários muito grande pode ocasionar sobrecarga no servidor de aplicação, fazendo com que os processamentos fiquem mais lentos.
Pontos de Sensibilidade:	Balanceamento de carga ativo
Tradeoff:	Não existe

## Evidências do cenário 2

O usuário se autentica no sistema escolhe a opção de menu agendamento médico e escolher a opção realizar agendamento médico.



O usuário digita o CPF e clica em Consultar Dados SUS, com isso será feito a busca do prontuário eletrônico do SUS do paciente. A busca será feita no sistema SASCi (legado).



O sistema recupera os dados do usuário. O usuário digita os dados necessários para realizar o agendamento e clica no botão Agendar.

**SGM**

**Agendamento Médico**  
Realizar Agendamento Médico em Unidade Básica de Saúde

CPF \*  
019.833.171-10 [Consultar Dados SUS](#)

Nome  
Luiz Sani Figue

Cartão Nacional de Saúde  
90863403402417

Data Nascimento  
23/03/1968

Endereço  
Santa Efigênia Belo Horizonte - MG CEP: 30260-908

Unidade Básica de Saúde \*  
UBS - 1

Endereço UBS  
Avenida 10 Lote 20

Data Hora da Consulta \*  
28/10/2021 17:08

Especialidade \*  
Dermatologista

[Limpar](#) [Agendar](#) [Voltar](#)

O sistema emite notificação avisando que o agendamento foi realizado com sucesso. Todo o processo de agendamento foi realizando somente com 4 cliques e o preenchimentos dos campos obrigatórios.

**SGM**

**Agendamento Médico**  
Realizar Agendamento Médico em Unidade Básica de Saúde

CPF \*  
[Consultar Dados SUS](#)

[Voltar](#)

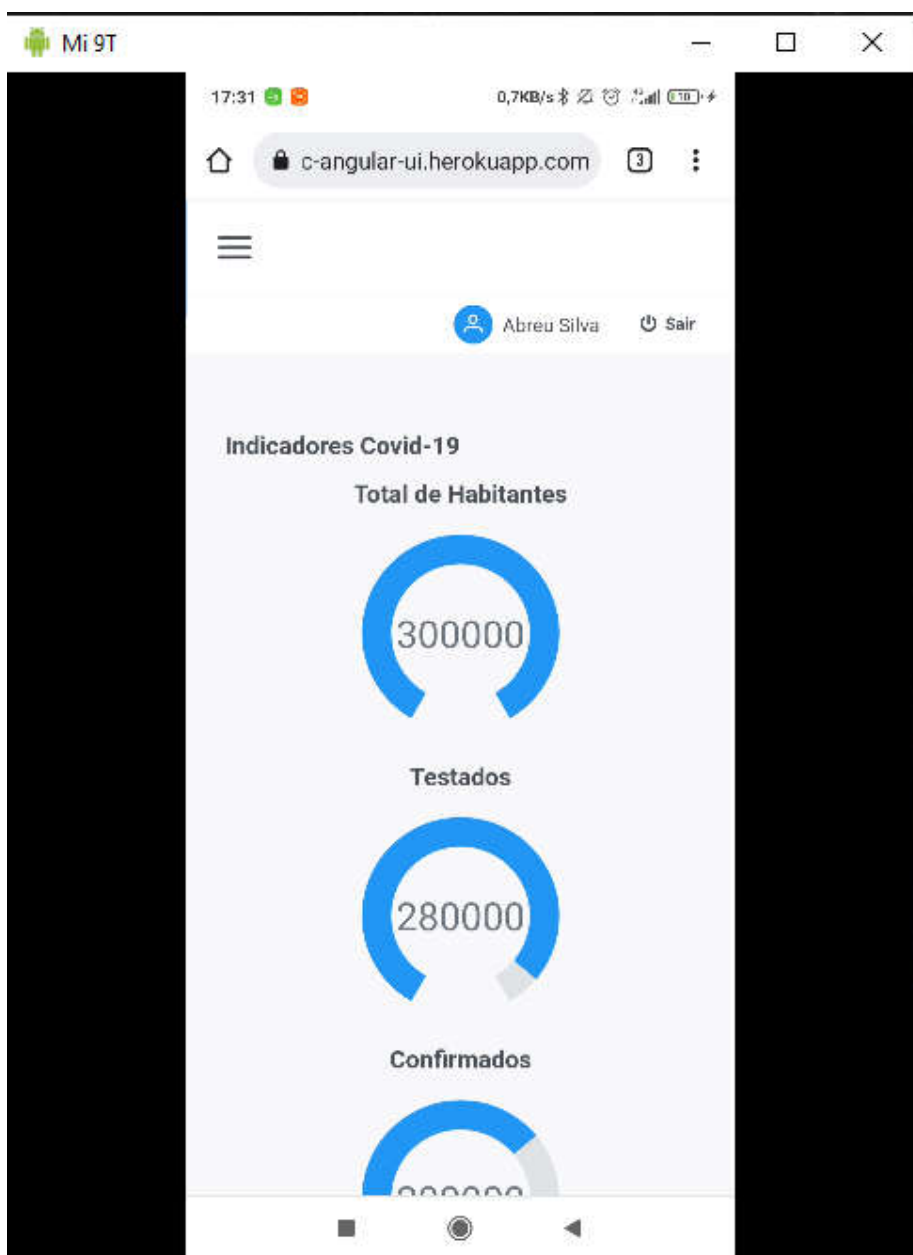
SGM - Sistema de Gestão Integrada Municipal  
Prefeitura Municipal de Bom Destino

### Cenário 3

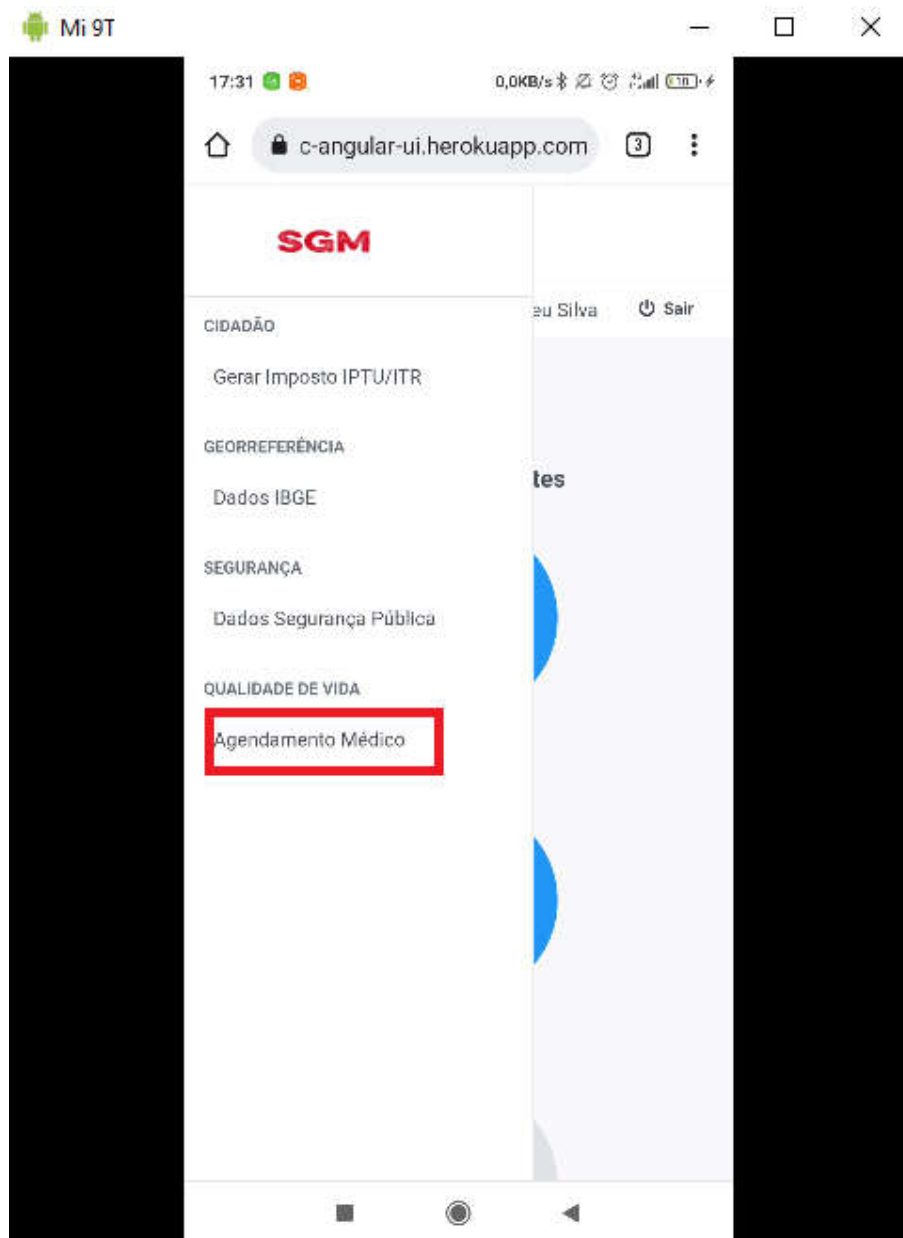
Atributo de Qualidade:	Acessibilidade
Requisito de Qualidade:	O sistema deve suportar ambientes web responsivos e ambientes móveis.
Preocupação	
O sistema deve se adaptar a interfaces de diversos tamanhos sem perda de funcionalidade e sem causar impactos de qualidade de navegação.	
Cenário (s):	
Cenário 2	
Ambiente	
Sistema em operação normal	
Estímulo:	
Usuário consultar os agendamentos médicos realizados.	
Mecanismo:	
Criação de telas utilizando mecanismos de design responsivos e ajustáveis, movimentando os componentes para que caibam em dispositivos diferentes.	
Medida de resposta:	
O sistema deve se adaptar a resoluções de tela dos diversos dispositivos, sem perder funcionalidade.	
Considerações sobre a arquitetura:	
Riscos:	A experiência do usuário pode ser altamente impactada devido a qualidade da rede que estão acessando. Além disso, resoluções extremamente pequenas poderão causar alguns deslocamentos indesejáveis, porém sem perda das funcionalidades.
Pontos de Sensibilidade:	Não existe
Tradeoff:	Não existe

### Evidências do cenário 3

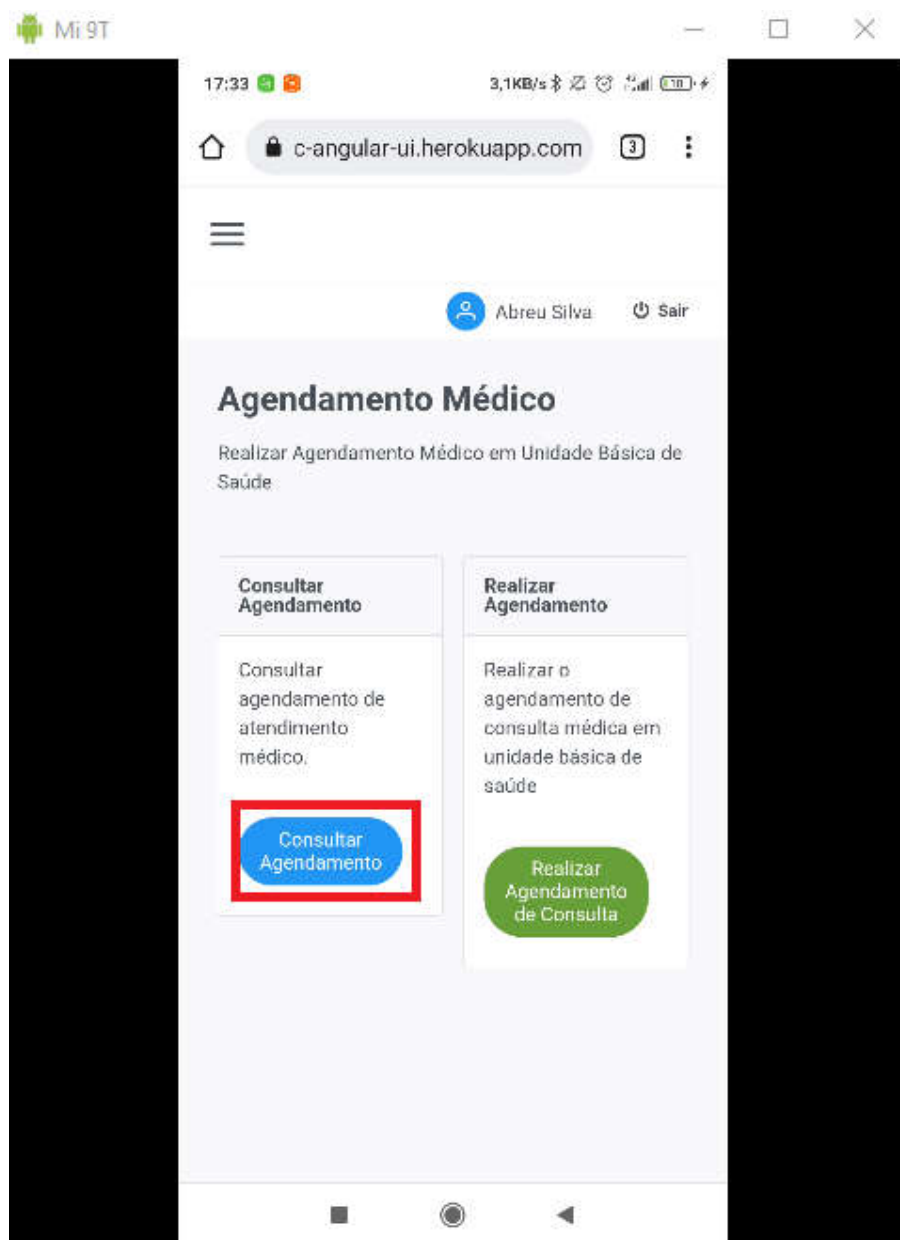
Usuário acessa a aplicação por um dispositivo móvel. O sistema é exibido de forma responsiva se adaptando a resolução do dispositivo de acesso.



Usuário expande os itens de menu e seleciona a opção Agendamento médico.



O usuário seleciona a opção consultar agendamento.





O usuário digita o CPF que deseja consultar os agendamentos.

The screenshot shows a mobile application interface on a Mi 9T device. The app is titled "Agendamento Médico" and has a subtitle "Realizar Agendamento Médico em Unidade Básica de Saúde". The main form contains a CPF input field with the value "019.833.171-10". Below the input field are three buttons: "Consultar" (blue), "Limpar" (grey), and "Voltar" (grey). The footer of the app displays "SGM - Sistema de Gestão Integrada Municipal" and "Prefeitura Municipal de Bom Destino". The device's status bar at the top shows the time as 17:34, a battery level of 61%, and a data speed of 0,2KB/s.

17:34 0,2KB/s 61% 3

Agendamento Médico

Realizar Agendamento Médico em Unidade Básica de Saúde

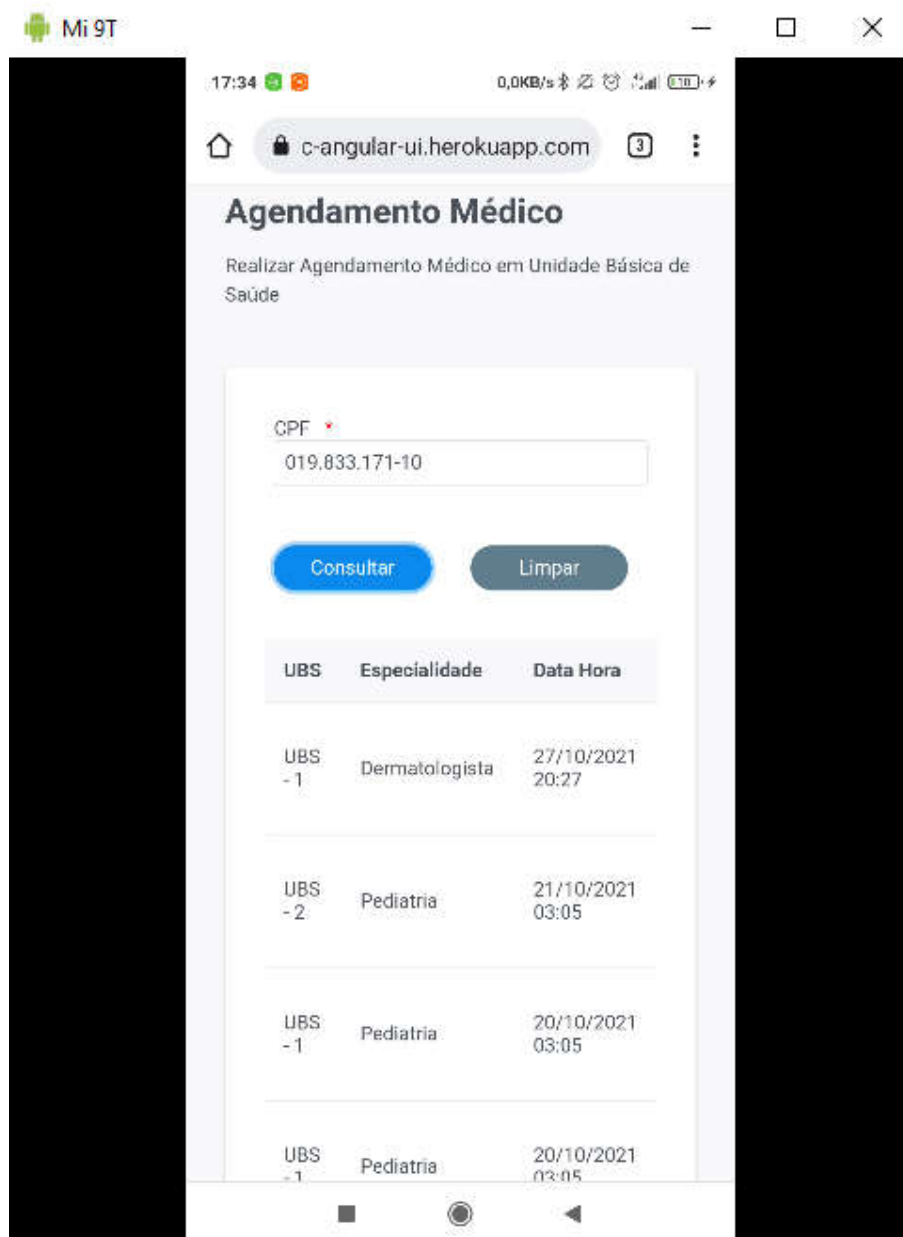
CPF \*

019.833.171-10

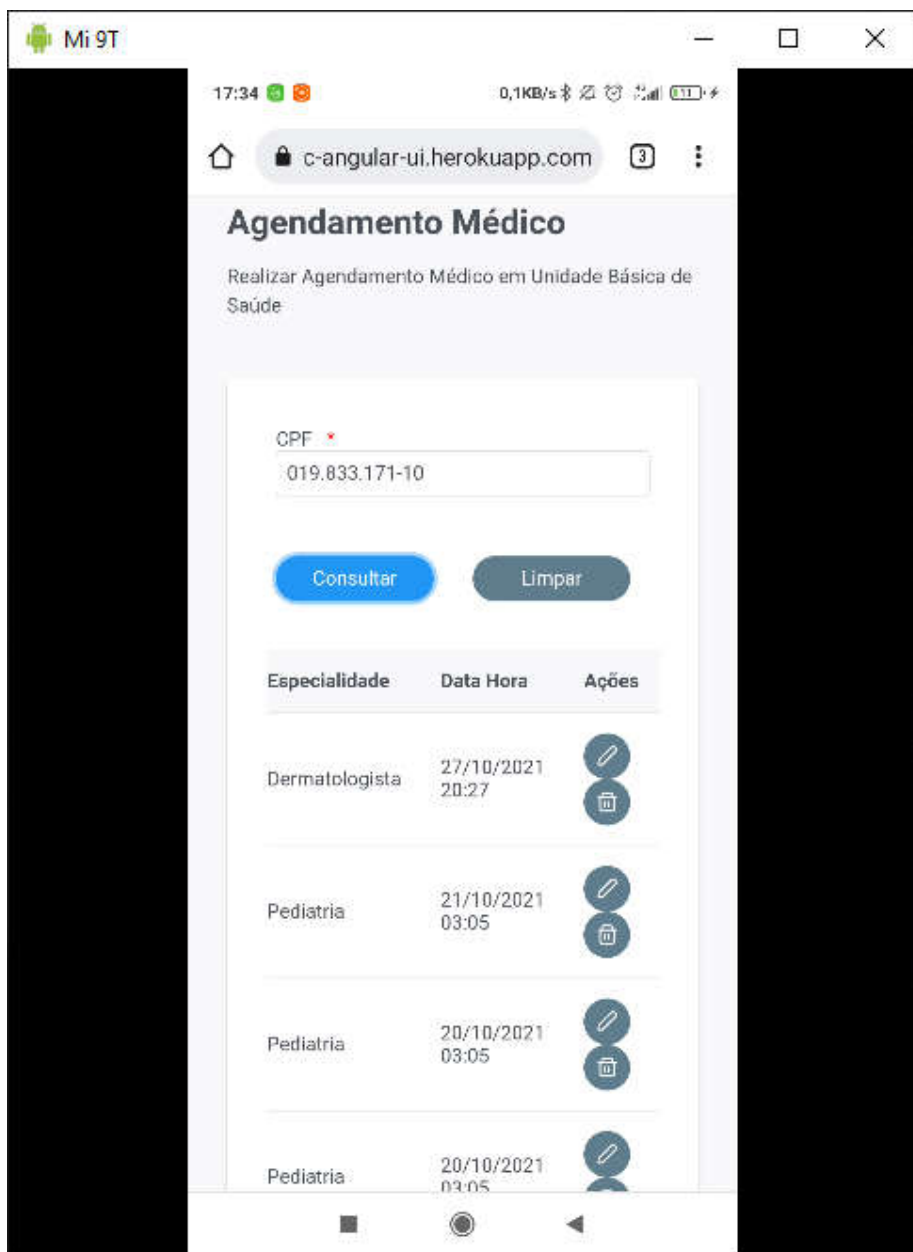
Consultar Limpar Voltar

SGM - Sistema de Gestão Integrada Municipal  
Prefeitura Municipal de Bom Destino

O sistema exibe a lista de consultas agendadas do paciente. A tabela com o resultado redimensionada de forma automática a resolução do dispositivo.



Caso o usuário deseje alterar o ou excluir o agendamento, será rolar a tabela para o lado, assim os botões serão exibidos.

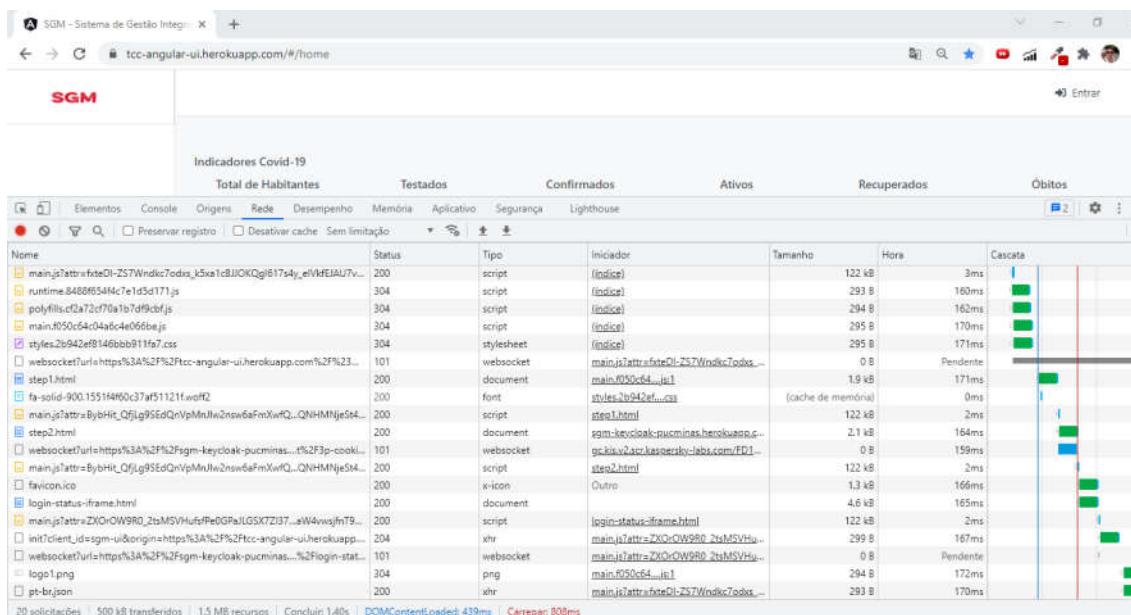


**Cenário 4**

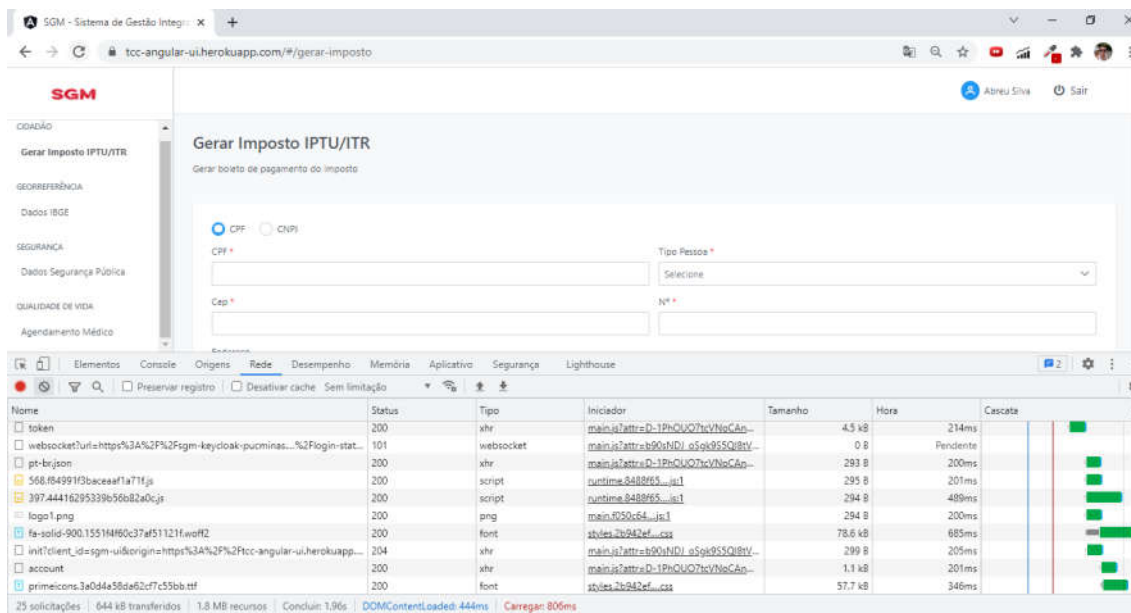
Atributo de Qualidade:	Desempenho
Requisito de Qualidade:	O sistema deve ser rápido.
Preocupação	
O sistema deve apresentar desempenho satisfatório dentro dos limites aceitável.	
Cenário (s):	
Cenário 4	
Ambiente	
Sistema em operação normal	
Estímulo:	
O sistema deve ser rápido e renderizar as páginas em no máximo 10 segundos.	
Mecanismo:	
Interface criada com telas objetivas, utilizando componentes com boa performance de uso e utilizando poucos componentes para que a renderização seja feita de forma rápida	
Medida de resposta:	
As páginas devem apresentar rapidez e bom desempenho	
Considerações sobre a arquitetura:	
Riscos:	Pode ocorrer um pico de memória no servidor ou um número de usuários muito grande pode ocasionar sobrecarga no servidor, fazendo com que os processamentos fiquem mais lentos.
Pontos de Sensibilidade:	Balanceamento de carga ativo
Tradeoff:	Não existe

## Evidências do cenário 4

Usuário acessando a tela pública do sistema. O sistema mostrou rapidez na renderização da tela, sendo carregada em menos de 2 segundo.



Usuário acessando a tela privada do sistema. O sistema mostrou rapidez na renderização da tela, sendo carregada em menos de 2 segundos.



#### 6.4. Resultado

Considerando os atributos de qualidade, o objetivo da validação arquitetural foi analisar esses atributos. Foi verificado que a arquitetura proposta atende as necessidades do projeto com algumas possíveis melhorias. A validação permitiu executar cenários para definir os pontos fortes e fracos da proposta do projeto. Os seguintes requisitos de qualidade foram validados:

Requisito Não Funcional	Testado	Homologado
O sistema deve apresentar altos padrões de segurança	Sim	Sim
O sistema deve prover boa usabilidade	Sim	Sim
O sistema deve suportar ambientes Web responsivos e ambientes móveis	Sim	Sim
O sistema deve ser rápido	Sim	Sim
O sistema deve ser comunicar com vários sistemas, com tecnologias heterogêneas	Sim	Sim

Avaliando a arquitetura proposta para esse projeto, foi possível identificar alguns pontos fortes e algumas limitações. O front-end foi desenvolvido utilizando o framework Angular proporcionado ao projeto uma estrutura bastante robusta e que segue padrões de componentes e serviços do próprio framework. A utilização do PrimeNG agiliza o desenvolvimento de interfaces, pois vários componentes já estão implementados e pronto para uso. A junção de Angular e PrimeNG permite que o usuário possa ter interfaces completamente responsivas se ajustando em qualquer dispositivo. Porém existe uma limitação quando ao front-end, onde caso tenha necessidade de componentes mais complexos será necessária uma equipe especializada em UX/UI.

A utilização do autenticador e autorizador Keycloak permite que o projeto tenha um ponto único de acesso e controle de usuário. Esse recurso permite abstrair toda a complexidade da parte de manter usuários, autenticação e autorização. Um dos pontos fortes do Keycloak

é a possibilidade de configurar a autenticação utilizando redes sociais como: Google, Facebook e Twitter.

O back-end foi dividido em vários módulos onde cada um possui o seu banco de dados, com isso cada módulo tem o seu objetivo específico o que os tornam coesos, com suas responsabilidades, possibilitando outras aplicações de utilizarem esses serviços, facilitando o reuso e integrações. Um ponto de melhoria, pode ser a utilização de vários bancos de dados para um mesmo módulo, podendo utilizar bancos relacionais e bancos NoSQL em um mesmo módulo.

No quesito de integração a arquitetura proposta consegue atender o desejo do município de integrar sistemas legados com novas aplicações, mesmo com a variação de tecnologias (REST e SOAP). Para alcançar este resultado foi utilizado um ESB – Enterprise Service Bus para gerenciar toda a comunicação entre legado e novas aplicações. Para os novos serviços desenvolvidos foi utilizado o padrão REST devido a sua facilidade de compreensão e implementação.

No quesito de implementação, foi utilizado a solução de implantação em nuvem. O Heroku foi escolhido pois é um ambiente de desenvolvimento e implantação completo em nuvem (PaaS), permitindo utilizar diversos serviços em diversas linguagens de programação conforme a necessidade da aplicação. O Heroku disponibiliza os serviços na forma gratuita ou da forma paga, conforme a necessidade de uso dos componentes.

Analisando a arquitetura final foi possível chegar à conclusão que a mesma oferece muitos pontos fortes e poucas limitações ou pontos de melhorias. A arquitetura mostra-se com um grande potencial de crescimento do projeto, divisão entre módulos e integração entre todos os sistemas que o município possui.

## **7. Conclusão**

Este trabalho apresentou um projeto arquitetural para uma plataforma que irá fornecer diversos serviços ao cidadão do município de Bom Destino. O projeto arquitetural tinha alguns objetivos a serem alcançados, vale destacar os seguintes objetivos: suportar ambientes web e móveis, se comunicar com sistemas externos via Restful, ser hospedado em nuvem e ser modular e implantável por módulo. Entende-se que os objetivos propostos foram alcançados, embora exista alguns pontos de melhorias que podem ser explorados em uma próxima versão.



## REFERÊNCIAS

Heroku Documentation. Site oficial. Disponível em: <<https://devcenter.heroku.com/>>. Acesso em: 01 set. 2021.

Mule ESB. Site Oficial. Disponível em: <<https://www.mulesoft.com/pt/resources/esb/what-mule-esb>>. Acesso em: 01 set. 2021

PrimeNG. Site Oficial. Disponível em: <<https://primefaces.org/primeng/showcase/#/setup>>. Acesso em: 01 set. 2021

Angular Docs. Google. Disponível em: <<https://angular.io/docs>>. Acesso em: 01 set. 2021

Keycloak with Angular. Blog. Disponível em: <<https://blog.bitsrc.io/integrating-keycloak-with-angular-for-sso-authentication-9d1c6c2d2742>>. Acesso em: 05 set. 2021.

Keycloak-Angular. NPM. Disponível em: <<https://www.npmjs.com/package/keycloak-angular>>. Acesso em: 05 set. 2021.

## APÊNDICES

URL do repositório do projeto: [https://github.com/igoorgalvao/TCC\\_PUC\\_SGM](https://github.com/igoorgalvao/TCC_PUC_SGM)

URL de apresentação da POC: <https://www.youtube.com/watch?v=Pv-CCJOG4uw>

URL da POC da aplicação: <https://tcc-angular-ui.herokuapp.com/#/home>

*(Observação: Devido ao Heroku ser utilizado em modo free é necessário aguardar alguns segundos após clicar no link para que todos os serviços sejam iniciados).*

Dados de Acesso (POC)		
Perfil	Login	Senha
Administrador	funcionario@sgm.com.br	123456
Cidadão	abreu@gmail.com	abreu
Cidadão	cidadao@gmail.com	123456

URL Serviços Heroku	
Nome	URL
tcc-angular-ui	<a href="https://tcc-angular-ui.herokuapp.com/#/home">https://tcc-angular-ui.herokuapp.com/#/home</a>
sgm-keycloak-pucminas	<a href="https://sgm-keycloak-pucminas.herokuapp.com/auth/">https://sgm-keycloak-pucminas.herokuapp.com/auth/</a>
tcc-backend-cidadao-api	<a href="https://tcc-backend-cidadao-api.herokuapp.com/cidadao-api/teste/listar">https://tcc-backend-cidadao-api.herokuapp.com/cidadao-api/teste/listar</a>
tcc-backend-legado-api	<a href="https://tcc-backend-legado-api.herokuapp.com/legado-api/sasci/obterFichaSus/01983317110">https://tcc-backend-legado-api.herokuapp.com/legado-api/sasci/obterFichaSus/01983317110</a>
tcc-backend-legado-api	<a href="https://tcc-backend-legado-api.herokuapp.com/legado-api/stur-imposto/obterImposto/01983317110/3/72220255/30">https://tcc-backend-legado-api.herokuapp.com/legado-api/stur-imposto/obterImposto/01983317110/3/72220255/30</a>
tcc-backend-qualidade-api	<a href="https://tcc-backend-qualidade-api.herokuapp.com/qualidadevida-api/teste/listar">https://tcc-backend-qualidade-api.herokuapp.com/qualidadevida-api/teste/listar</a>
tcc-backend-seguranca-api	<a href="https://tcc-backend-seguranca-api.herokuapp.com/seguranca-api/teste/listar">https://tcc-backend-seguranca-api.herokuapp.com/seguranca-api/teste/listar</a>