UNIVERSIDADE FEDERAL DE ITAJUBÁ
INSTITUTO DE MATEMÁTICA E COMPUTAÇÃO

COM242 - Sistemas Distribuídos

Prof. Rafael Frinhani

PROJETO

Sistema de serviços públicos

Grupo:

Arthur Reznik - 2016001460
Caio Noriyuki Sasaki - 2016009842
Guilherme Delgado de Carvalho da Costa Braga - 2019003806
Ivan Leoni Vilas Boas - 2018009073
Karen de Souza Pompeu - 2016001610
Rodrigo Filippo Dias - 2016001479

01 de Dezembro, 2020

Sumário

---

1 Introdução

O projeto tem como objetivo dar o suporte necessário para melhorar o sistema de comunicação e gestão na ocorrência de incidentes servindo de plataforma para Defesa Civil. Em casos de desastres ambientais, de criminalidade, requisição entre entidades de saúde e segurança entre outros, o sistema irá possibilitar o monitoramento de riscos e atuação sobre eventos naturais e acidentes humanos, a partir da gestão de ordem de serviço para agentes públicos e privados. A plataforma permitirá a disseminação da informação para os cidadãos cadastrados na aplicação sobre avisos, alertas que tenham alguma relevância. Para isto, foram criadas APIs utilizando protocolo HTTP para que realizem a troca de informações, sendo criadas de maneira distribuídas e com características distintas.

2 Viabilidade

Segundo a LEI Nº 13.460, DE 26 DE JUNHO DE 2017, todo cidadão brasileiro tem direito aos serviços públicos prestados direta ou indiretamente pela administração pública. Analisando os percalços naturais, como cidades sendo devastadas por deslizamentos de terra, inundações, desabamentos de edifícios e centenas de pessoas mortas e, adicionado a isso a falta de comunicação entre as entidades públicas, que culmina em medidas atrasadas para o atendimento, que deveria ser imediato por parte das entidades (polícia, SAMU, CEMIG) para os grupos (regiões afetadas) populacionais. Esse cenário acaba resultando na descontinuidade, precariedade e atrasos de serviços que são essenciais à população, como corpo de bombeiros, polícia, telefonias fixa e móvel, transporte público, escolas, hospitais, energia elétrica, água potável, trânsito, abastecimento de alimentos, e muitos outros.

Neste sentido, propomos o sistema de serviço Público - SSP a ser aplicado a uma prefeitura qualquer para melhorar sua gestão, onde algumas de suas organizações público-privadas, através da trocas de mensagens com a prefeitura, garantiriam maior qualidade de prestação de serviço e melhor atendimento aos cidadãos. Uma vez que a prefeitura conseguiria concentrar melhor as informações e repassá-las as demais entidades, que por sua vez, em posse das informações otimizaria os seus serviços prestados.

Ademais, o sistema permitirá ainda que os cidadãos cadastrados também recebam informações, avisos e alertas para salvaguardar suas vidas contra catástrofes naturais. Assim o sistema auxiliaria em medidas tanto mediativas por parte das entidades quanto em preditivas para a população.

3 Tecnologias Utilizadas

As tecnologias utilizadas na construção do sistema foram Python e Node como linguagem de programação. Para o banco de dados empregou-se o banco MongoDB e o protocolo http para a realização da comunicação entre os sistemas do projeto. A utilização do mongodb deve-se ao fato de ser mais escalável que um banco de dados relacional, fazendo com que, futuramente, outras APIs possam se conectar e guardar suas informações. As linguagens utilizadas não são as mesmas para representar os serviços públicos. Isto foi feito para melhor simular uma situação real, em que diferentes APIs, feitas em diferentes linguagens, poderiam se comunicar através do ponto central: a prefeitura.

4 Estrutura do projeto

O projeto conta com módulos independentes para cada órgão. Estes são a polícia, o SAMU e a CEMIG. A Prefeitura é o controlador central que poderá acionar serviços de qualquer um desses órgãos através do envio de uma json. Os serviços aguardam receber uma comunicação e são capazes de responder.

Os módulos desenvolvidos que compõem o sistema distribuído controle de serviços públicos são:

- Prefeitura
- Policia
- SAMU
- CEMIG

4.1 Funcionalidades

As funcionalidades da interface de cada um dos prestadores de serviços podem variar em diferentes aspectos, de acordo com o papel empenhado. No caso deste trabalho, não existe uma interface amigável para o usuário, já que somente o núcleo da aplicação foi desenvolvido. Cada entidade representada neste trabalho realiza a troca de mensagens seguindo suas próprias necessidades e particularidades.

4.1.1 Prefeitura

A prefeitura será o centro de informações, cada uma das entidades deverá se reportar a ela, sendo a mesma responsável por disseminar essa informação para os usuários. Ela também, possibilitará a comunicação entre os diferentes prestadores de serviço. Para exemplificar a funcionalidade da comunicação entre eles, imaginamos o seguinte cenário: há um incêndio ocorrendo em uma subestação de roteamento de energia elétrica da CEMIG, não se sabe se há feridos no incidente, o SAMU/Bombeiros são requisitados, a causa do incêndio pode ter sido acidental ou criminosa além de colocar em risco os moradores e quem estiver transitando pela estrada nos arredores; assim a POLÍCIA também é acionada.

4.1.2 Polícia

4.1.3 SAMU

A implementação da entidade Samu foi feita utilizando a linguagem Python, diferente das outras entidades que foram utilizados o Node.js. Na Figura 1 abaixo é possível ver a estrutura da classe Samu, onde seus atributos são: ocorrência, carro e notificar e seus métodos são somente para se pegar as informações necessárias para o envio

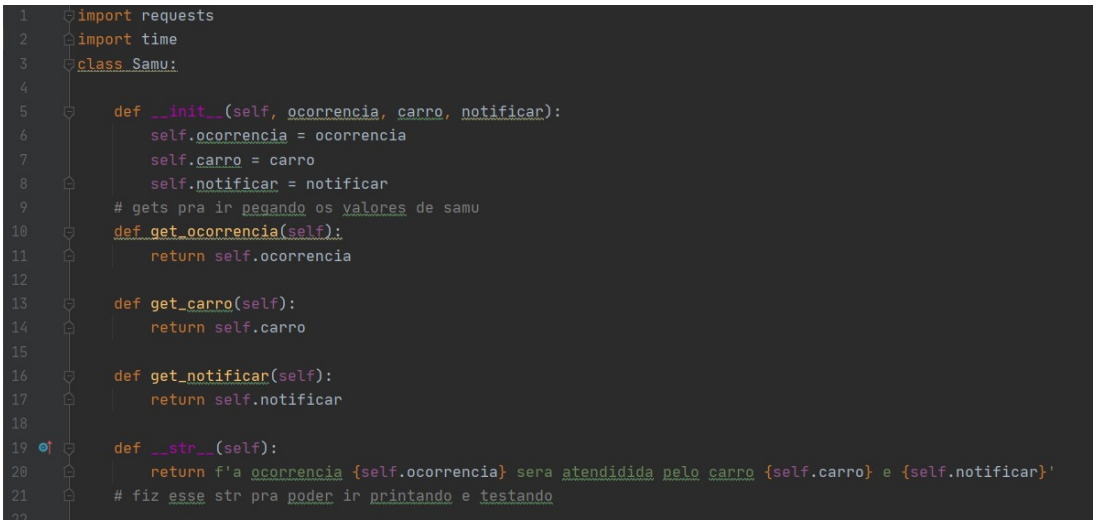


Figura 1: Estrutura da classe samu

A parte de envio das mensagens se da na Figura 2, após conexão com um servidor criado através do python flask, as mensagens podem ser trocadas. No caso, enquanto o programa estiver rodando, o usuário pode continuamente digitar as mensagens e estas serão enviadas, neste caso, um dicionário foi criado para o envio das mensagens:

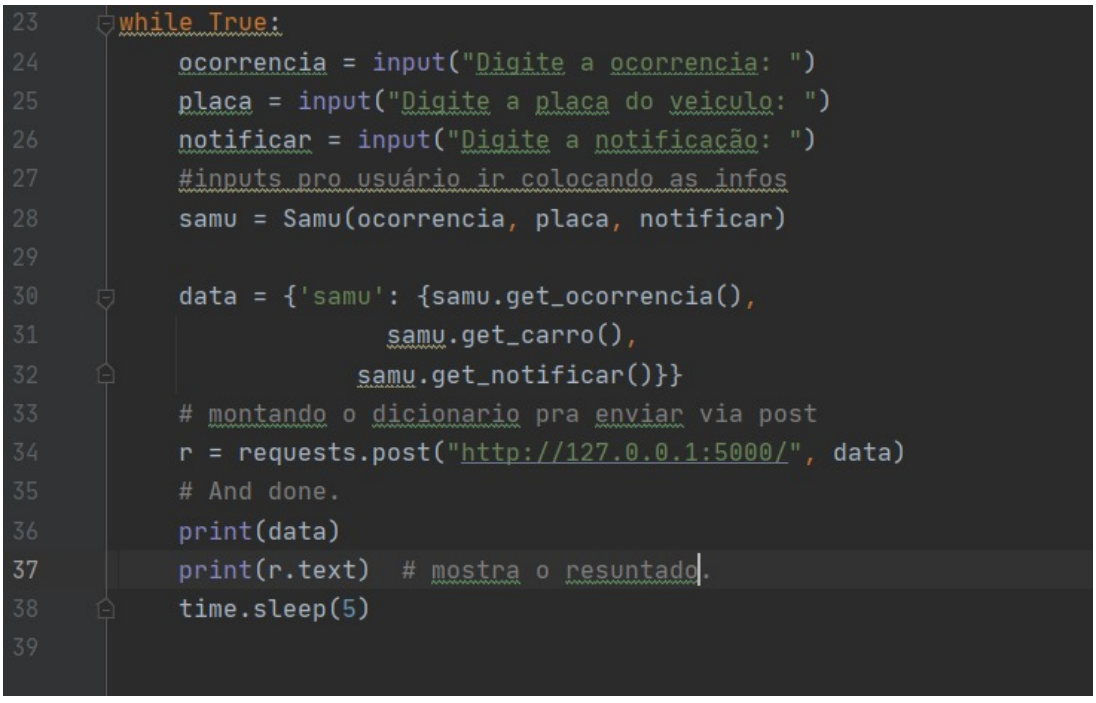
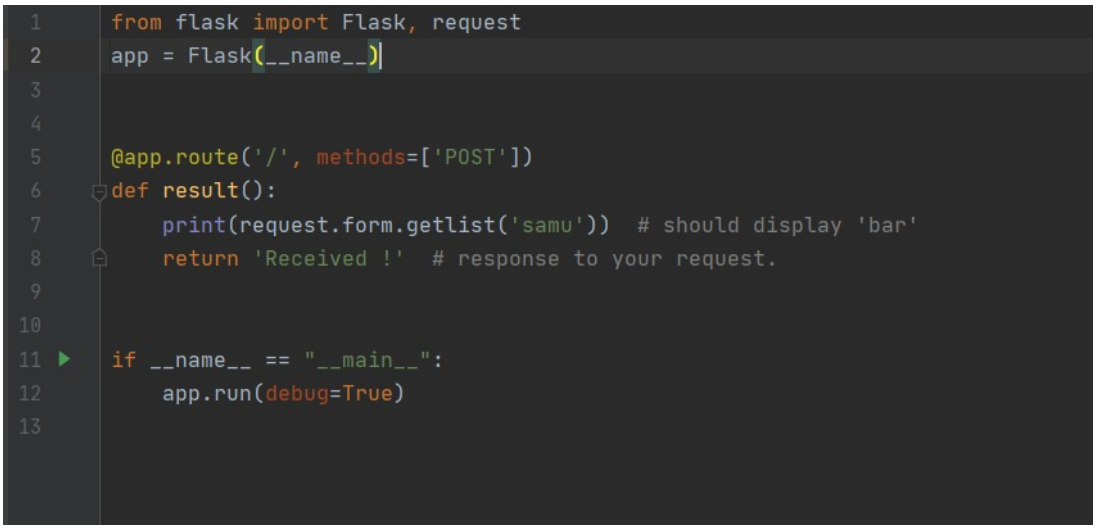


Figura 2: Envio da mensagem

Na Figura 3 vemos a estrutura do receptor das mensagens, que irá recebe-las e imprimi-las na tela. Vale ressaltar que o método utilizado para a transferência das mensagens foi o Post. Caso a mensagens seja recebida corretamente, uma mensagem de recebimento é enviada.



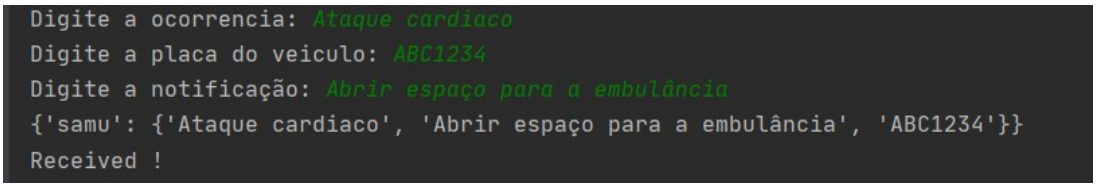
```
from flask import Flask, request
app = Flask(__name__)


@app.route('/', methods=['POST'])
def result():
    print(request.form.getlist('samu'))  # should display 'bar'
    return 'Received !'  # response to your request.


if __name__ == "__main__":
    app.run(debug=True)
```

Figura 3: Estrutura do receptor das mensagens

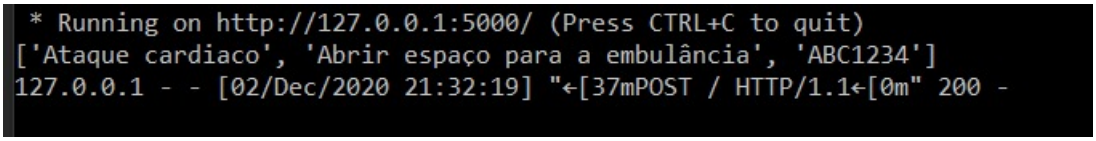
A partir do momento que se estabelece a conexão, neste exemplo, local, as mensagens podem ser trocadas. O envio das mensagens pode ser feito por outras maneiras, neste exemplo, o próprio usuário digita as mensagens que serão enviadas, ou seja, ele mesmo digita a ocorrência, a placa do veículo e a notificação.



```
Digite a ocorrencia: Ataque cardiaco
Digite a placa do veiculo: ABC1234
Digite a notificação: Abrir espaço para a embulância
{'samu': {'Ataque cardiaco', 'Abrir espaço para a embulância', 'ABC1234'}}
Received !
```

Figura 4: Envio das mensagens

Após o envio das mensagens, o recebimento é feito e a mensagens é imprimida na tela para o usuário de outra ponta possa vê-la, assim como é mostrado na Figura 5:



```
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
['Ataque cardiaco', 'Abrir espaço para a embulância', 'ABC1234']
127.0.0.1 - - [02/Dec/2020 21:32:19] "←[37mPOST / HTTP/1.1←[0m" 200 -
```

Figura 5: Recebimento das mensagens

4.1.4 CEMIG

4.2 Comunicação

A comunicação no sistema se inicia com a requisição de um serviço feito pela prefeitura, o que será enviado para o determinado serviço que retornará uma resposta. A Figura 6 ilustra os caminhos e direções do envio de pacotes entre os sistemas desenvolvidos. O envio consiste em um pacote do tipo json enviado utilizando o protocolo http.

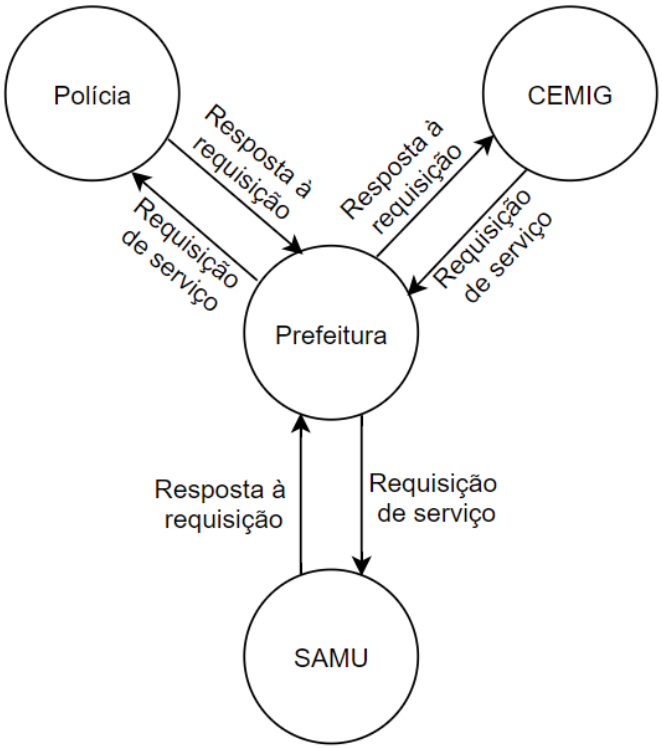


Figura 6: Comunicação entre prefeitura e serviços

5 Implementação do Sistema

O sistema foi implementado, conforme foi descrito nos capitulos anteriores e seu codigo está disponivel e pode ser encontrado no seguinte endereço: `https://github.com/arthur-reznik/Trabalho-Sistemas-Distribuidos`

6 Conclusão

Um sistema dedicado à disseminação de informações relevantes à segurança dos habitantes de uma cidade, além da rápida possibilidade de intercomunicação entre os prestadores de serviços essenciais, potencializará o acionamento dessas entidades. Sendo um trabalho distribuído e baseado em APIs, qualquer linguagem de programação poderá usufruir deste serviço. A utilização do protocolo HTTP também se provou bastante eficaz, principalmente por sua facilidade em ser trabalhado.