



Universidade Federal do ABC

Sistema de vigilância por vídeo em Rede Mesh

Aluno: Breno Matos Carvalho Soares

Curso: TSI

Orientador: Mario Gazziro

Sistema de vigilância por vídeo em Rede Mesh

Trabalho de Conclusão de Curso apresentado ao curso de pós-graduação em TSI - Tecnologia e Sistemas da Informação, da Universidade Federal do ABC, como requisito parcial para obtenção do diploma no curso de Tecnologia.

Dedicatória

Dedico este trabalho a todas as pessoas que me deram suporte e que me “suportaram” durante esta jornada. Agradeço ao Prof. Mario Guazziro pela orientação e apoio e ao Cezar Augusto pela ajuda técnica na montagem do ambiente para o experimento.

Agradeço a minha esposa Cristiane pela paciência e sobrecarga que ele arcou para que eu pudesse me dedicar as disciplinas e ao projeto de conclusão e aos meu filhos Ulisses, Aquiles e Hércules que sempre são a minha fonte de inspiração e motivação.

Lista de Figuras

Figura 1 - Orange pi i96.....	11
Figura 2 - Logitech c920.....	11
Figura 3 - Ambiente do WrapperGate.....	12
Figura 4 - Fluxograma do módulo Wrapper Gate.....	13
Figura 5 - Caso de uso do módulo Member Watch - Todos os serviços web do MemberWatch implementam o padrão REST.....	14
Figura 6 - Modelo Entidade Relacionamento (WrapperGate e MemberWatch).....	15
Figura 7 - Topologia de rede.....	16
Figura 8 - Ambiente montado com as máquinas e suas especificações.....	16
Figura 9 - balthasar-2.....	17
Figura 10 - melchior-1.....	17
Figura 11 - Execução do agente da câmera.....	18
Figura 12 - Execução do WrapperGate.....	19
Figura 13 - Ferramenta (POSTMAN) utilizada para fazer uma requisição a API WrapperGate.....	19
Figura 14 - Pasta de imagens para armazenamento das imagens capturadas.....	20
Figura 15 - Banco de Dados - Inserção dos dados para teste.....	20
Figura 16 - Captura de imagens pelo WrapperGate.....	21
Figura 17 - Imagens Capturadas.....	22

Sumário

Resumo.....	6
Tema.....	7
Introdução.....	7
Motivação.....	8
Objetivo.....	10
Quais as vantagens.....	10
Metodologia.....	11
Resultados.....	16
Considerações Finais.....	22
Conclusão.....	23
Apêndices.....	24
CamCapture.java - Principal classe do agente de captura de imagens.....	24
WrapperGate.java - Classe de serviços Web, para comunicação com os agentes remotos das câmeras.....	24

Resumo

Com base na quantidade de dispositivos de vigilância conectados à internet e a grande aderência de pessoas a redes sociais, é possível compartilharmos os outputs destes dispositivos e compartilhar entre os membros de uma mesma rede social, com as devidas autorizações concedidas pelos proprietários dos dispositivos.

Comprovamos em laboratório a viabilidade deste modelo de software, permitindo a captura de imagens remotamente e acesso às mesmas para usuários que não sejam os proprietários, mas possuem um acesso.

Este modelo de software poderá oferecer grandes ganhos às comunidades, tornando os membros mais engajados com as mesmas e com uma ferramenta que auxilia na segurança das comunidades cadastradas.

Tema

Modelo de software para recebimento e disponibilização de dados captados por câmeras em Rede Mesh para criação de um rede de vigilância comunitária e colaboração mútua entre os indivíduos de uma comunidade.

Introdução

Muitas informações são coletadas por dispositivos de vigilância e disponibilizadas para os proprietários destes dispositivos através de serviços web, que recebem o output de dados dos aparelhos e disponibiliza o mesmo para os indivíduos autorizados (proprietários do dispositivo).

A algumas décadas no passado uma das poucas maneiras de obtermos este nível de informação era através da interação com outros membros da comunidade a qual os indivíduos estavam inseridos. Conforme o estudo de Andrea Kavanaugh, Ph.D. [1], podemos perceber que as comunidades nas últimas décadas ultrapassam as fronteiras geográficas, mas continuam sendo comunidades dentro das redes sociais e como o uso do IOT tem reconfigurado os meios de produção como discorrido no artigo de José Ignacio Rodríguez Molano, Juan Manuel Cueva Lovelle, Carlos Enrique Montenegro, J. Javier Rainer Granados e Rubén González Crespo [2] e modificado o mundo conforme o artigo de Asta Zelenkauskaite, Nik Bessis, Stelios Sotiriadis e Eleana Asimakopoulou [3]

Toda esta informação é utilizada pelo proprietário do dispositivo, que toma as suas decisões baseado no que foi capturado e disponibilizado. Além dos dispositivos de captura de imagens temos: microfones, sensores de movimento, termômetros, auto-falantes, servo-motores, etc. Todos dispositivos que produzem informação, enviam informações e muitas vezes executam comandos.

Toda esta capacidade depende apenas de um acesso a um link que expõe uma interface de serviços para um dispositivo que executa as instruções enviadas pelo mesmo.

Pode-se utilizar os dispositivos para controlar ambientes, como descrito no artigo de Gonçalo Marques e Rui Pitarma [4] , onde o IOT é utilizado para fazer a aferição da qualidade do ar dentro de edifícios.

As redes sociais também podem ser utilizadas como ferramentas úteis a comunidades agregando consciência social e preparando a mesma para eventuais desastres, como relatado no artigo de Yoko Akama, Susan Chaplin, Peter Fairbrother [5] ou apoio a questões de vulnerabilidade ou diversidade social dentro das comunidades como mencionado no artigo de Mark E. Feinberg, Nathaniel R. Riggs & Mark T. Greenberg [6].

Soluções inovadoras são propostas para o uso de IOT, barateando o custo da implementação e a facilidade de uso, como descrito no artigo de Husniza Razalli, Mohammed Hazim Alkawaz e Aizat Syazwan Suhai [7]

Com as redes sociais, combinada com as aplicações e informações disponibilizadas pelo IOT podemos ter uma quantidade considerável de informações conforme o artigo de Luigi Atzori, Antonio Iera, Giacomo Morabito e Michele Nitti [8] e conectar as comunidades a um universo de computação ubíqua, como descrito por Antonio M. Ortiz, Dina Hussein, Soochang Park, Son N. Han, Noel Crespi [9] e um clustering de Internet Of Things and Social Network como explanado no artigo de Lianhong Ding, Peng Shi e Bingwu Liu [10]

Motivação

Recuperar o espírito de cooperação entre os membros de uma comunidade, sem limites geográficos, pois podemos estar falando de pessoas em estados diferentes que possuem alguma afinidade e por conta desta proximidade de interesses se estabelecem vínculos como uma comunidade.



Utilizar as tecnologias de dispositivos de vigilância para estimular o engajamento e o interesse pelos assuntos da comunidade.

Estimular através do compartilhamento das informações dos dispositivos, aproximação entre os membros da comunidade, para que se preocupem uns com os outros e consigam se enxergar, como parte importante e ativa de um todo.

Objetivo

Proponho um modelo de software que permita a uma comunidade, implantar um serviço de vigilância comunitária , através da integração de dispositivos de observação (câmeras) que estejam conectados à internet e compartilhem seus outputs de dados coletados em um hub onde cada um dos dispositivos estaria desta forma conectado a uma rede mesh.

Neste hub os dispositivos teriam seus serviços de output de dados conectados ao hub, que deverá implementar uma interface para interação com a API do dispositivo. Após o dispositivo ter sido devidamente conectado e sua interface implementada no hub, todos os usuários conectados ao hub que possuam permissão por parte do proprietário do dispositivo, ou pertençam a um grupo definido pelo mesmo, poderão acessar as informações que aquele dispositivo provê através do próprio hub.

Quais as vantagens

Além do hub poder ser utilizado como uma ferramenta de vigilância comunitária, pode-se utilizá-lo como porteiro eletrônico, conectando a campainha a internet e desviando o sinal para algum usuário do hub previamente autorizado a receber este sinal e ao monitorar a câmera prosseguir com o atendimento ao interlocutor.

O hub poderia ser conectado a um serviço de análise de imagens para identificar possíveis padrões de invasão ou violência e notificar aos integrantes da comunidade ou a polícia sobre uma potencial ameaça.

O ponto principal nesta ideia é utilizar a tecnologia e as comunidades para criar um engajamento entre as pessoas pelo senso coletivo de segurança e ajuda mútua.

Dependendo do tamanho que estas comunidades assumirem podemos ter bairros inteiros monitorados, pessoas podendo viajar com tranquilidade , vizinhos ajudando vizinhos, um retorno ao antigo senso de comunidade que nossos avós tinham.

Metodologia

Para o desenvolvimento deste hub será utilizado o Spring Tools 4 como IDE, banco de dados PostgreSQL 16.2 e Java 11. O software será desenvolvido utilizando arquitetura de microserviços com SpringBoot 3.0. Todos os módulos serão colocados em imagens Docker, para abstrair as necessidades de configuração da máquina onde será instalado o hub. Basicamente serão desenvolvidos dois módulos:

- WrapperGate - Conexão aos dispositivos de vigilância e exposição da interface de serviços dos mesmos.
- MemberWatch - Gerenciador de membros e permissões aos dispositivos cadastrados

O WrapperGate deverá consumir e interagir com a interface de serviços de uma câmera Logitech c920 (figura 2) conectada a um Orange pi i96 (figura 1) . Toda a interface da c920 , deverá ser encapsulada pelo WrapperGate e este será responsável por abstrair todo fluxo de autenticação, bufferização e disponibilização para os usuários que possuírem autorização para consumir a saída de informações do dispositivo. O WrapperGate deverá expor a interface do dispositivo utilizando como padrão de serviços web o REST.

	
Figura 1 - Orange pi i96	Figura 2 - Logitech c920

O WrapperGate funcionará dentro do contexto definido conforme a figura 3.

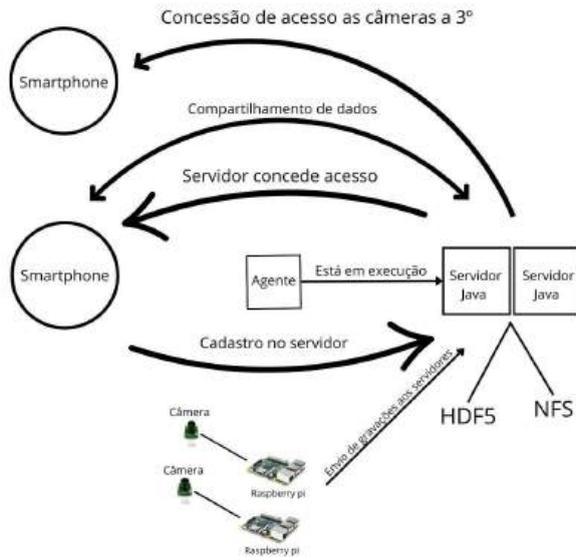


Figura 3 - Ambiente do WrapperGate

O módulo WrapperGate precisará abstrair do usuário requisitante das informações de output do dispositivo, todo o trabalho de chamada a API, autenticação (se possuir), ou qualquer outra atividade inerente ao consumo do serviço que expõem os dados de output do dispositivo. Para o escopo deste estudo utilizarei o padrão de projeto “Provider” para encapsular todos os detalhes inerentes ao serviço da câmera c920 e teremos uma interface única para acesso aos métodos da API do dispositivo:

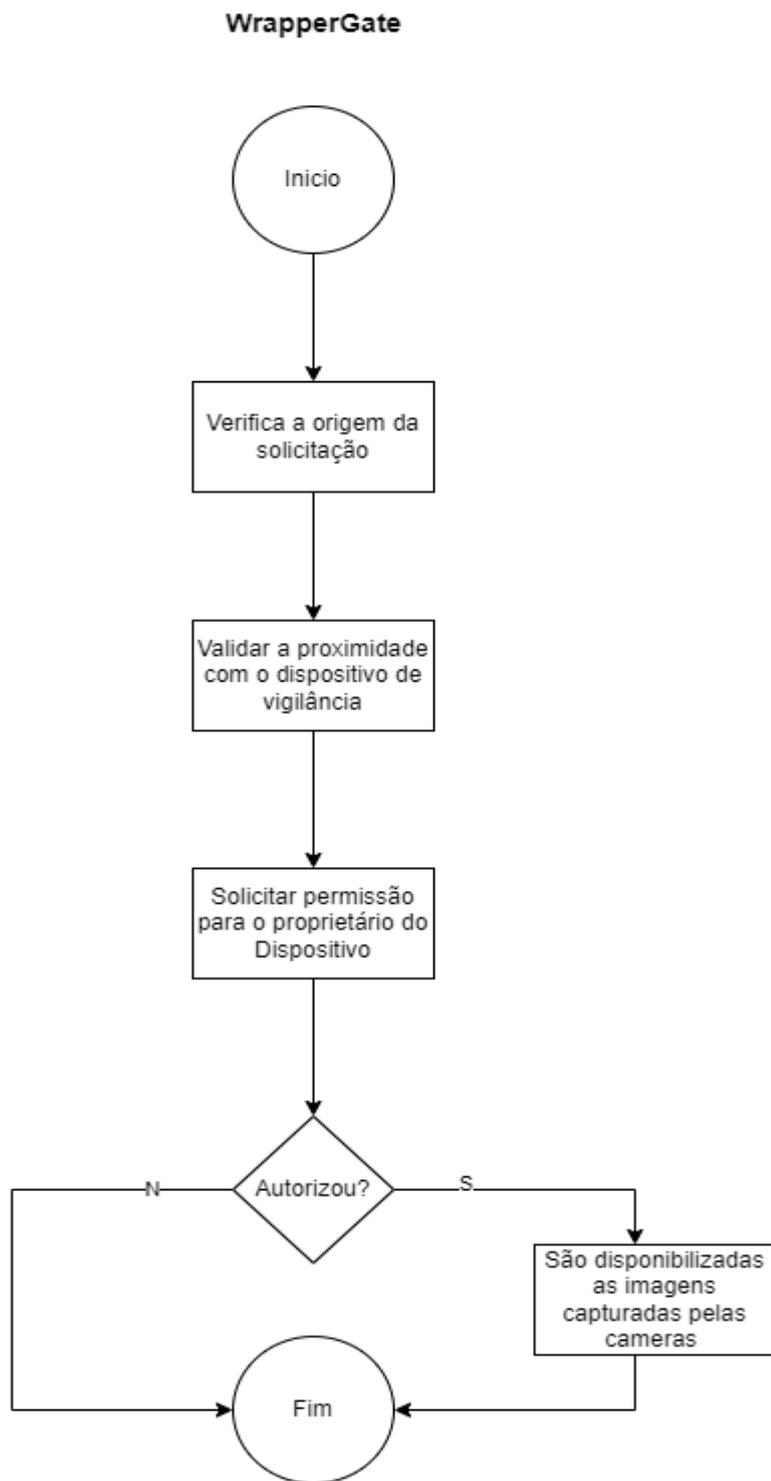


Figura 4 - Fluxograma do módulo Wrapper Gate

O MemberWatch será o módulo de gerenciamento de usuário e permissões. Neste primeiro momento iremos implementar as seguintes funções do caso de Uso:

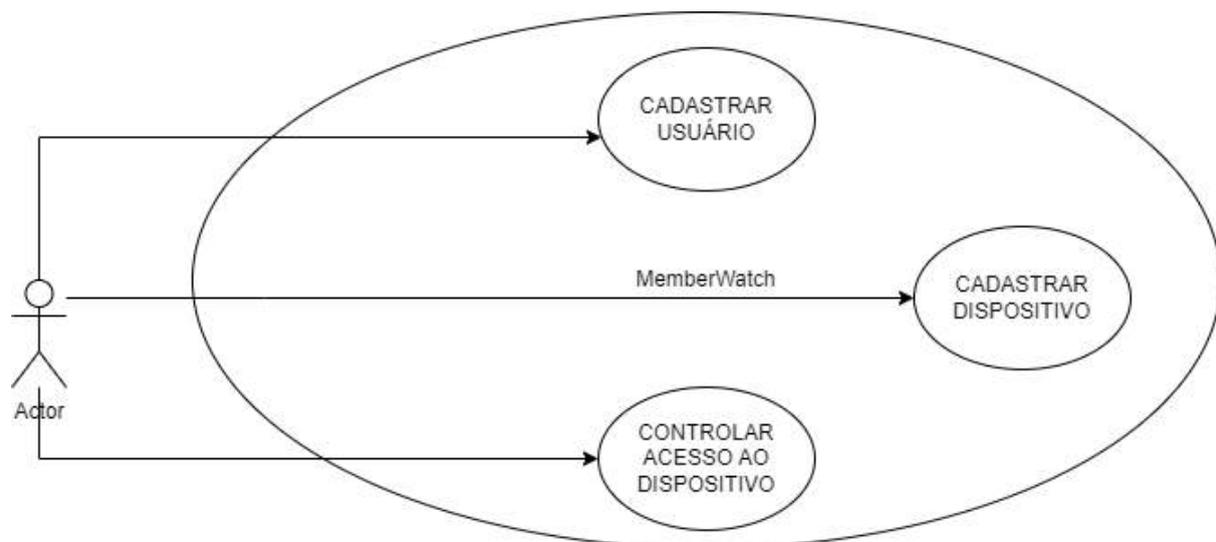


Figura 5 - Caso de uso do módulo Member Watch - Todos os serviços web do MemberWatch implementam o padrão REST.

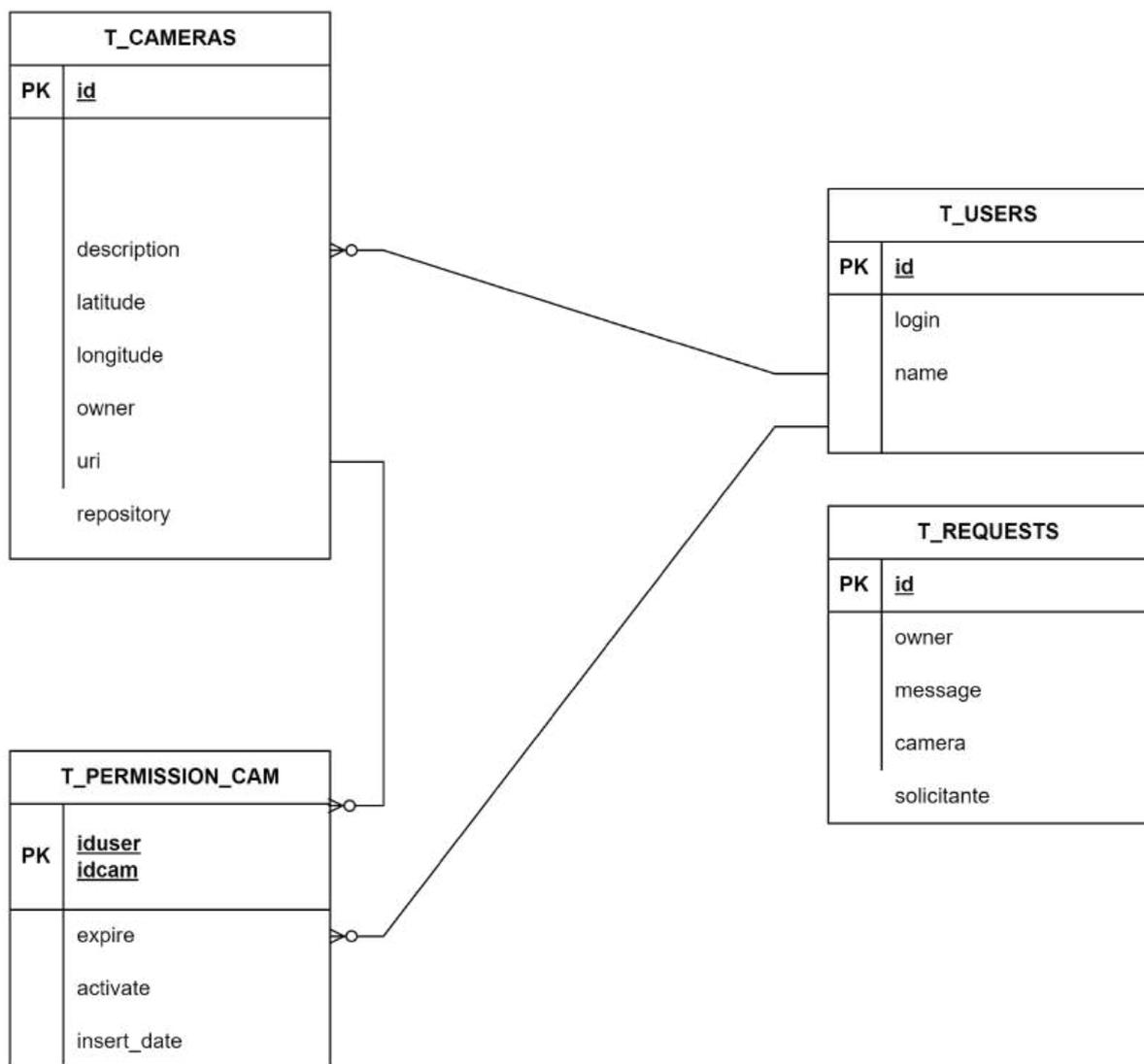


Figura 6 - Modelo Entidade Relacionamento (WrapperGate e MemberWatch)

Devido ao curto espaço de tempo serão implementadas as funcionalidades básicas do hub, para comprovar a viabilidade deste modelo do software.

Resultados

Na figura 7 é mostrada a topologia de hardware montada para suportar a aplicação e as câmeras.

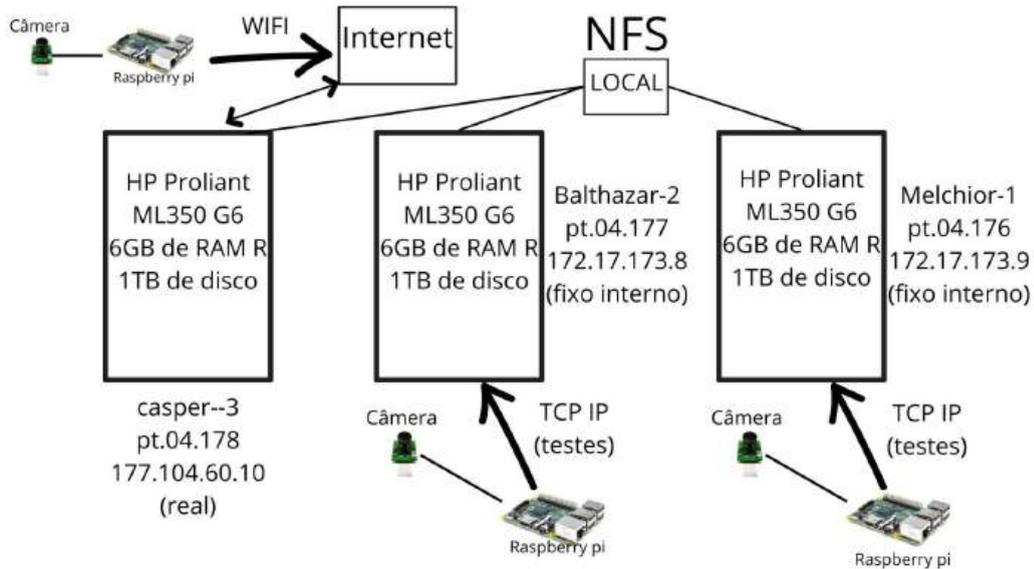


Figura 7 - Topologia de rede

Nas figura 8 temos as imagens reais das máquinas definidas na topologia



Figura 8 - Ambiente montado com as máquinas e suas especificações

Nas figuras 9 e 10 exibe os detalhes das máquinas conectadas a rede local, sem acesso a internet

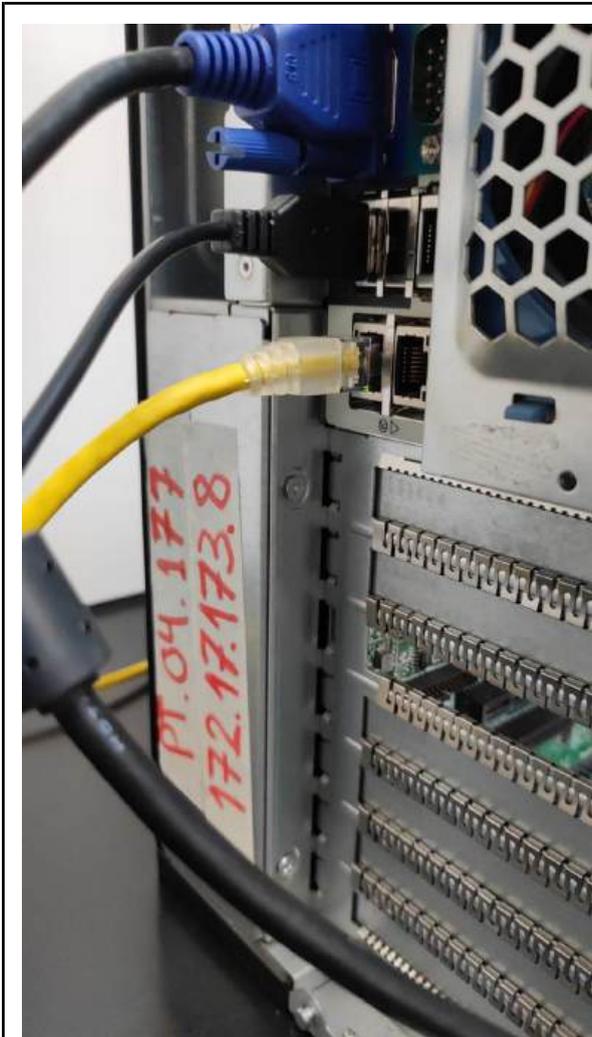


Figura 9 - balthasar-2



Figura 10 - melchior-1

Conforme os resultados obtidos nos ensaios de laboratório, foram efetuadas com sucesso na plataforma windows o acesso a duas câmeras distintas através do servidor WrapperGate.

Com este resultado é possível provar a viabilidade de um hub o de acesso para várias câmeras conectadas a este mesmo ponto.

O agente criado para acessar a câmera permitiu a captura de imagens remotamente através de uma interface de serviço com o hardware.

```
Prompt de Comando - java - j x + v
D:\Breno\TCC_Breno\project\agentwebcam\target>java -jar agentwebcam-0.0.1.jar

:: Spring Boot :: (v3.3.1)

2024-07-24T21:12:37.218-03:00 INFO 8628 --- [main] com.ithaka.ufabc.agentcam.AgentCam : Starting AgentCam v0.0.1 using Java 17.0.1
with PID 8628 (D:\Breno\TCC_Breno\project\agentwebcam\target\agentwebcam-0.0.1.jar started by specr in D:\Breno\TCC_Breno\project\agentwebcam\targ
et)
2024-07-24T21:12:37.227-03:00 INFO 8628 --- [main] com.ithaka.ufabc.agentcam.AgentCam : No active profile set, falling back to 1 d
efault profile: "default"
2024-07-24T21:12:40.098-03:00 INFO 8628 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8080 (http)
2024-07-24T21:12:40.118-03:00 INFO 8628 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-07-24T21:12:40.118-03:00 INFO 8628 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10
.1.25]
2024-07-24T21:12:40.181-03:00 INFO 8628 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicatio
nContext
2024-07-24T21:12:40.183-03:00 INFO 8628 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization
completed in 2710 ms
2024-07-24T21:12:41.238-03:00 INFO 8628 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8080 (http) with co
ntext path '/'
2024-07-24T21:12:41.416-03:00 INFO 8628 --- [main] com.ithaka.ufabc.agentcam.AgentCam : Started AgentCam in 5.571 seconds (process
running for 9.261)
```

Figura 11 - Execução do agente da câmera

O WrapperGate roteou as requisições conforme o resultado para as câmeras cadastradas em sua base de teste.

```
Prompt de Comando - java - j x + v
:: Spring Boot :: (v3.3.1)

2024-07-24T21:17:59.547-03:00 INFO 22928 --- [main] com.ithaka.ufabc.openwebcam.App : Starting App v0.0.1 using Java 17.0.11 wi
th PID 22928 (D:\Breno\TCC_Breno\project\openwebcam\target\openwebcam-0.0.1.jar started by specr in D:\Breno\TCC_Breno\project\openwebcam\target)
2024-07-24T21:17:59.552-03:00 INFO 22928 --- [main] com.ithaka.ufabc.openwebcam.App : No active profile set, falling back to 1
default profile: "default"
2024-07-24T21:18:00.727-03:00 INFO 22928 --- [main] s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repository
s in DEFAULT mode
2024-07-24T21:18:00.986-03:00 INFO 22928 --- [main] s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning
in 161 ms. Found 2 JPA repository interfaces.
2024-07-24T21:18:01.798-03:00 INFO 22928 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port 8081 (http)
2024-07-24T21:18:01.799-03:00 INFO 22928 --- [main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
2024-07-24T21:18:01.799-03:00 INFO 22928 --- [main] o.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/10
.1.25]
2024-07-24T21:18:01.769-03:00 INFO 22928 --- [main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplicatio
nContext
2024-07-24T21:18:01.769-03:00 INFO 22928 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initializatio
n completed in 2082 ms
2024-07-24T21:18:01.818-03:00 INFO 22928 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2024-07-24T21:18:02.079-03:00 INFO 22928 --- [main] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection conn8: ur
l=jdbc:h2:mem:wrappergatedb;user=SA
2024-07-24T21:18:02.087-03:00 INFO 22928 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2024-07-24T21:18:02.193-03:00 INFO 22928 --- [main] o.s.b.a.h2.H2ConsoleAutoConfiguration : H2 console available at '/h2-console'. Da
tabase available at 'jdbc:h2:mem:wrappergatedb'
2024-07-24T21:18:02.315-03:00 INFO 22928 --- [main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo
[name: default]
2024-07-24T21:18:02.429-03:00 INFO 22928 --- [main] org.hibernate.Version : HHH000412: Hibernate ORM core version 6.5
.2.Final
2024-07-24T21:18:02.484-03:00 INFO 22928 --- [main] o.h.c.internal.RegionFactoryInitiator : HHH00026: Second-level cache disabled
```

Figura 12 - Execução do WrapperGate

A aplicação cliente exibe o resultado esperado, convertendo a imagem em trânsito como uma string em Base64 para um arquivo de imagem armazenado no diretório da aplicação cliente.

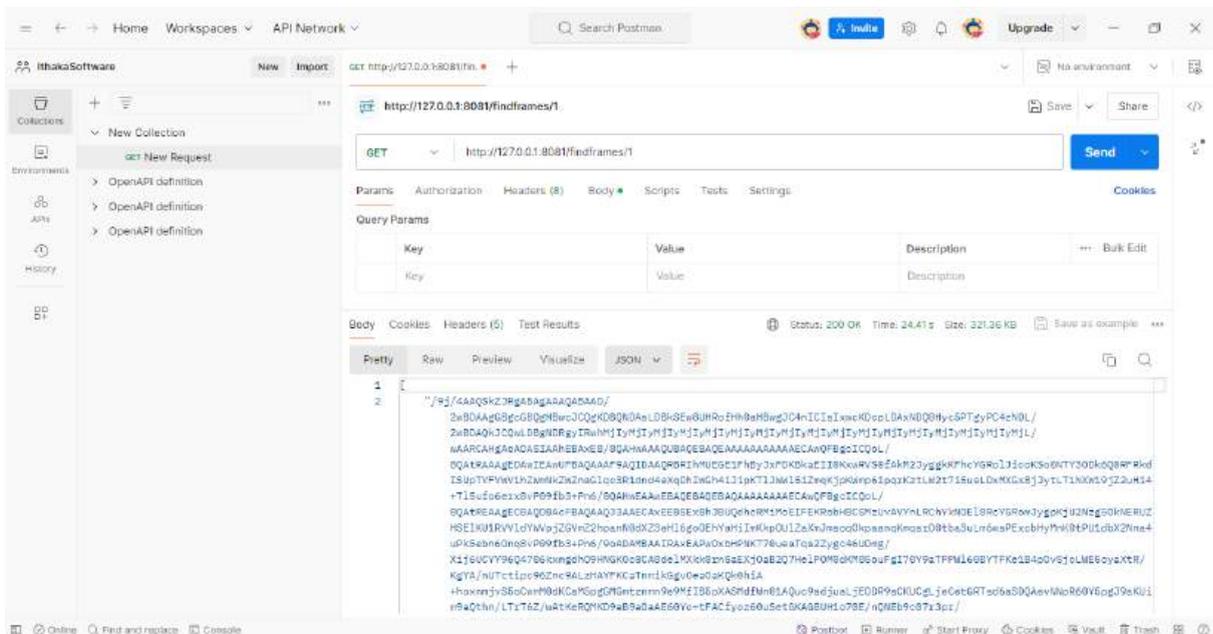


Figura 13 - Ferramenta (POSTMAN) utilizada para fazer uma requisição a API WrapperGate

O WrapperGate mostrou-se apto a manter requisições as cameras conectadas a ele para captura de frames de maneira periódica e as armazenando em uma hierarquia de pastas.

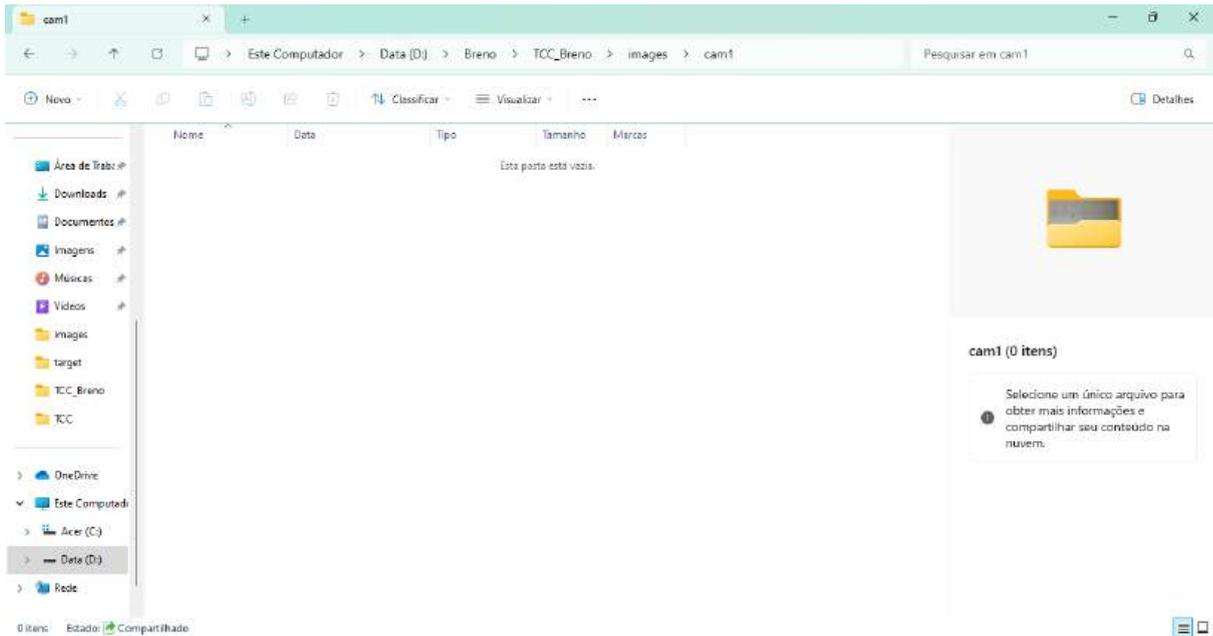


Figura 14 - Pasta de imagens para armazenamento das imagens capturadas

Carregando as informações de câmeras e usuários na base de dados H2

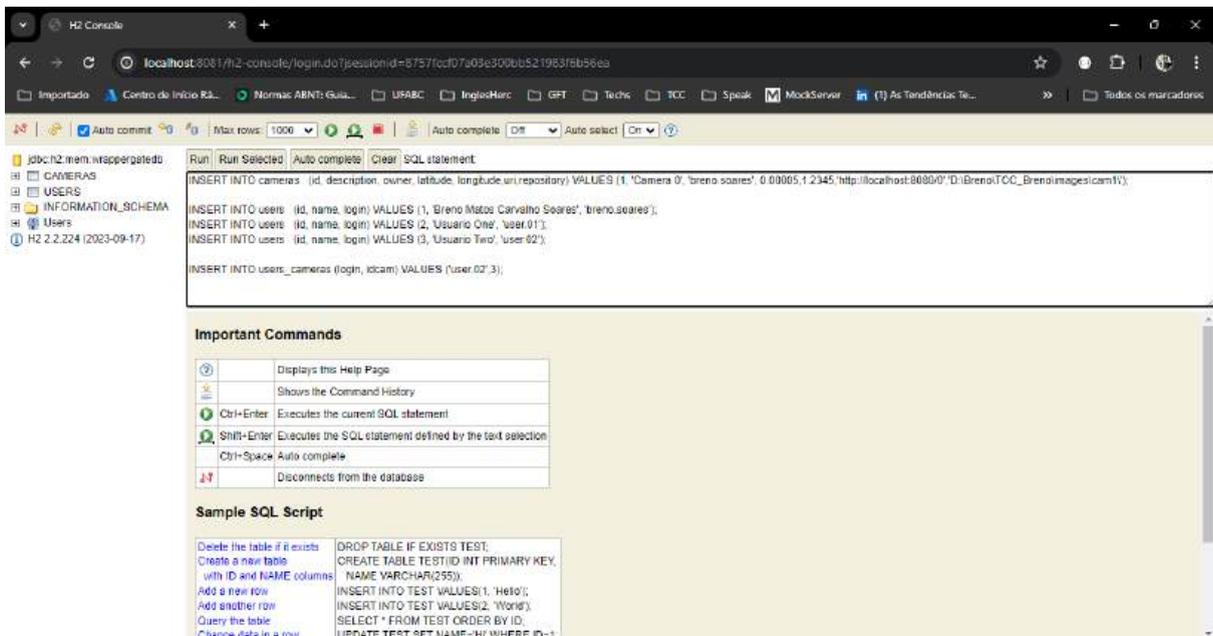


Figura 15 - Banco de Dados - Inserção dos dados para teste

Servidor em execução

```
Prompt de Comando - java - j X + v
nContext
2024-07-25T22:02:32.930-03:00 INFO 6968 --- [main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initialization
completed in 1858 ms
2024-07-25T22:02:33.104-03:00 INFO 6968 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
2024-07-25T22:02:33.398-03:00 INFO 6968 --- [main] com.zaxxer.hikari.pool.HikariPool : HikariPool-1 - Added connection conn0: url
=jdbc:h2:mem:wrappergatedb user=SA
2024-07-25T22:02:33.401-03:00 INFO 6968 --- [main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
2024-07-25T22:02:33.406-03:00 INFO 6968 --- [main] o.s.b.a.h2.H2ConsoleAutoConfiguration : H2 console available at '/h2-console'. Dat
abase available at 'jdbc:h2:mem:wrappergatedb'
2024-07-25T22:02:33.696-03:00 INFO 6968 --- [main] o.hibernate.jpa.internal.util.LogHelper : HH000204: Processing PersistenceUnitInfo
[name: default]
2024-07-25T22:02:33.776-03:00 INFO 6968 --- [main] org.hibernate.Version : HH000412: Hibernate ORM core version 6.5.
2.Final
2024-07-25T22:02:33.825-03:00 INFO 6968 --- [main] o.h.c.internal.RegionFactoryInitiator : HH000076: Second-level cache disabled
2024-07-25T22:02:34.395-03:00 INFO 6968 --- [main] o.s.o.j.p.SpringPersistenceUnitInfo : No LoadTimeWeaver setup: ignoring JPA clas
s transformer
2024-07-25T22:02:34.458-03:00 WARN 6968 --- [main] org.hibernate.orm.deprecation : HH09000025: H2Dialect does not need to be
specified explicitly using 'hibernate.dialect' (remove the property setting and it will be selected by default)
2024-07-25T22:02:35.381-03:00 INFO 6968 --- [main] o.h.e.t.j.p.i.JtaPlatformInitiator : HH000489: No JTA platform available (set
'hibernate.transaction.jta.platform' to enable JTA platform integration)
2024-07-25T22:02:35.470-03:00 INFO 6968 --- [main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for p
ersistence unit 'default'
2024-07-25T22:02:35.902-03:00 WARN 6968 --- [main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by defa
ult. Therefore, database queries may be performed during view rendering. Explicitly configure spring.jpa.open-in-view to disable this warning
2024-07-25T22:02:36.326-03:00 INFO 6968 --- [main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port 8081 (http) with co
ntext path '/'
2024-07-25T22:02:36.373-03:00 INFO 6968 --- [main] com.ithaka.ufabc.openwebcam.App : Started App in 6.268 seconds (process runn
ing for 7.283)
2024-07-25T22:04:22.210-03:00 INFO 6968 --- [nio-8081-exec-5] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring DispatcherServlet 'dis
patcherServlet'
2024-07-25T22:04:22.211-03:00 INFO 6968 --- [nio-8081-exec-5] o.s.web.servlet.DispatcherServlet : Initializing Servlet 'dispatcherServlet'
2024-07-25T22:04:22.214-03:00 INFO 6968 --- [nio-8081-exec-5] o.s.web.servlet.DispatcherServlet : Completed initialization in 3 ms
2024-07-25T22:04:22.227-03:00 WARN 6968 --- [nio-8081-exec-5] o.s.web.servlet.PageNotFound : No mapping for GET /favicon.ico
2024-07-25T22:04:22.235-03:00 WARN 6968 --- [nio-8081-exec-5] o.s.web.servlet.PageNotFound : No endpoint GET /favicon.ico.
```

Figura 16 - Captura de imagens pelo WrapperGate

Resultado na pasta:

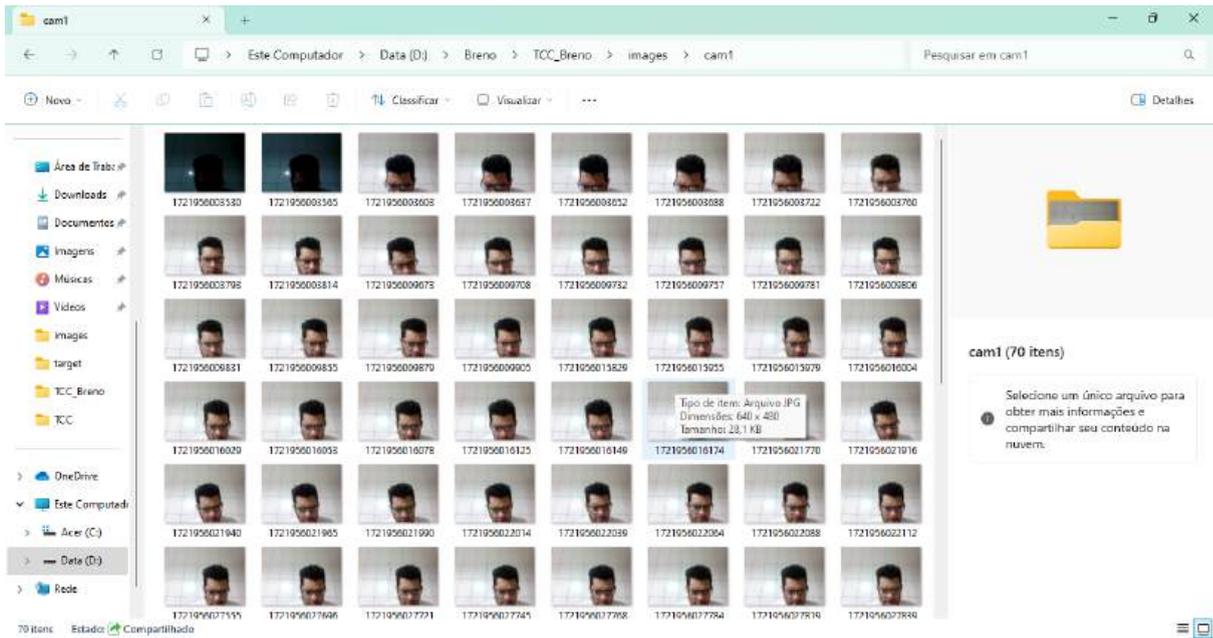


Figura 17 - Imagens Capturadas

Considerações Finais

O Experimento comprova a viabilidade de um software para comunicação com múltiplas câmeras pertencentes a diferentes usuários, dentro de uma rede social.

O compartilhamento das imagens capturadas pelos dispositivos conectados ou acesso/captura diretamente na interface de serviço de uma câmera conectada é realizada após a autorização prévia que concede acesso ao dispositivo ou as informações capturadas até o momento pelo mesmo.

Tudo isto utilizando um único ponto de acesso fornecido pelo WrapperGate através de um login ao mesmo.

Conclusão

Com o compartilhamento das informações capturadas pelos dispositivos de segurança, podemos aumentar a segurança dentro de nossas comunidades simplesmente permitindo que membros da comunidade tenham acesso às imagens capturadas pelas câmeras de qualquer membro que autorize o acesso e desta forma possam monitorar os entornos da sua comunidade e prevenir ou inibir ameaças.

Podemos estender esta cobertura a uma cidade e permitir acesso às autoridades para que possam atuar de maneira assertiva e preventiva para aumentar a segurança das comunidades. Conectar Algoritmos de Análise de imagens para identificar situações perigosas ou ameaças aos entornos, recebendo como input de dados as imagens capturadas por qualquer dispositivo conectado ao WrapperGate.

Apêndices

CamCapture.java - Principal classe do agente de captura de imagens

```
1 package com.ithaka.ufabc.agentscam;
2
3 import java.awt.image.BufferedImage;
4
5 public class CamCapture {
6
7     public List<String> captureFrame(Integer idCam, Integer seconds) {
8
9         CanvasFrame canvas = new CanvasFrame("Job Cam");
10        canvas.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
11
12        var imagesEncoded64 = new ArrayList<String>();
13
14        try {
15            OpenCVFrameGrabber grabber = GrabberStart.getInstance();
16            Frame frame = null;
17            OpenCVFrameConverter.ToIplImage converter = new OpenCVFrameConverter.ToIplImage();
18
19            for (int i = 0; i<seconds;i++) {
20                frame = grabber.grabFrame();
21
22                IplImage img = converter.convert(frame);
23
24                opencv_imgcodecs.cvSaveImage("imageCam.jpg", img);
25
26                File fileImage = new File("./imageCam.jpg");
27                BufferedImage originalImage=ImageIO.read(fileImage);
28                ByteArrayOutputStream baos=new ByteArrayOutputStream();
29                ImageIO.write(originalImage, "jpg", baos );
30
31                byte[] imageInByte = baos.toByteArray();
32
33                imagesEncoded64.add(Base64.getEncoder().encodeToString(imageInByte));
34
35                frame.close();
36
37                frame = null;
38
39                Thread.sleep(500);
40            }
41        }
42    }
43}
```

WrapperGate.java - Classe de serviços Web, para comunicação com os agentes remotos das câmeras.

```
1 package com.ithaka.ufabc.openwebcam.rest;
2
3 import java.io.File;
4
5 @RestController
6 public class WrapperGate {
7
8     @Autowired
9     WebCamServices webCamServices;
10
11     @SuppressWarnings("unchecked")
12     @GetMapping("/{FindFrames/{idcam}")
13     public List<String> findIdCamAndCaptureFrames(@PathVariable Integer idcam){
14
15         var camera = webCamServices.findCamera(idcam);
16
17         RestClient restClient = RestClient.create();
18
19         return restClient.get()
20             .uri(camera.getUri())
21             .retrieve()
22             .body(List.class);
23     }
24
25     @GetMapping("/listcams")
26     public List<CameraDTO> listIdCams() {
27
28         return webCamServices.listAllCameras();
29     }
30
31     @Scheduled(fixedRate = 3000)
32     public void captureImagesInCameras() {
33         var cameras = webCamServices.listAllCameras();
34
35         captureFramesInAllCameras(cameras);
36     }
37
38     private void captureFramesInAllCameras(List<CameraDTO> cameras) {
39         cameras.parallelStream().forEach(cam->{
40
41             RestClient restClient = RestClient.create();
42
43             @SuppressWarnings("unchecked")
44             List<String> frames = restClient.get()
45                 .uri(cam.getUri())
46                 .retrieve()
47                 .body(List.class);
48         });
49     }
50 }
```

Referências

[1] The Impact of Computer Networking on Community: A Social Network Analysis Approach1, Andrea Kavanaugh, Ph.D., Setembro 27-29, 1999.

<http://www.bev.net/sites/default/files/researchDocs/TPRC.UserStudy.Kavanaugh.pdf>

[2] Metamodel for integration of Internet of Things, Social Networks, the Cloud and Industry 4.0, José Ignacio Rodríguez Molano, Juan Manuel Cueva Lovelle, Carlos Enrique Montenegro, J. Javier Rainer Granados & Rubén González Crespo, 27 Fevereiro de 2017. <https://link.springer.com/article/10.1007/s12652-017-0469-5>

[3] Interconnectedness of Complex Systems of Internet of Things through Social Network Analysis for Disaster Management, 2012 Fourth International Conference on Intelligent Networking and Collaborative Systems, Asta Zelenkauskaite, Nik Bessis, Stelios Sotiriadis e Eleana Asimakopoulou, [19-21 Setembro 2012](#). <https://ieeexplore.ieee.org/abstract/document/6337971>

[4] Using IoT and Social Networks for Enhanced Healthy Practices in Buildings, Gonçalo Marques e Rui Pitarma, Outubro 2018.

https://link.springer.com/chapter/10.1007/978-3-030-03577-8_47

[5] Role of social networks in community preparedness for bushfire, International Journal of Disaster Resilience in the Built Environment , ISSN: 1759-5908, Setembro 2014,

<https://www.emerald.com/insight/content/doi/10.1108/IJDRBE-01-2014-0010/full/html>

[6] Social Networks and Community Prevention Coalitions, Springer Link, Volume 26, páginas 279-298, Julho 2005,

<https://link.springer.com/article/10.1007/s10935-005-5390-4>

[7] Smart IOT Surveillance Multi-Camera Monitoring System, IEEE 7th Conference on Systems, Process and Control (ICSPC 2019), 13–14 Dezembro 2019.

<https://ieeexplore.ieee.org/abstract/document/9067984>

[8] The Social Internet of Things (SIoT) – When social networks meet the Internet of Things: Concept, architecture and network characterization, Computer Networks Volume 56, Issue 16, 14 Novembro 2012, Páginas 3594-3608. <https://www.sciencedirect.com/science/article/abs/pii/S1389128612002654>

[9] The Cluster Between Internet of Things and Social Networks: Review and Research Challenges, IEEE Volume 1, Issue 3, 18 Abril 2014, páginas 206-215. <https://ieeexplore.ieee.org/abstract/document/6802330>

[10] The clustering of Internet, Internet of Things and social network, IEEE 2010 Third International Symposium on Knowledge Acquisition and Modeling, 20-21 October 2010. <https://ieeexplore.ieee.org/abstract/document/5646274>



Assinatura do Orientador