DESENVOLVIMENTO DE UM SISTEMA WEB – PATRIMÔNIO MAIS SEGURO – COM ACESSO SIMULTÂNEO E REMOTO A CÂMERAS DE SEGURANÇA E ARMAZENAMENTO EM NUVEM DE GRAVAÇÕES

DEVELOPMENT OF A WEB SYSTEM – PATRIMÔNIO MAIS SEGURO – WITH SIMULT NEO AND REMOTE ACCESS TO SECURITY CAMERAS AND CLOUD STORAGE OF RECORDINGS

Diogo Muniz Soares de Brito
dmsb2@discente.ifpe.edu.br

Marco Antônio de Oliveira Domingues
marcodomingues@recife.ifpe.edu.br

RESUMO

Mecanismos eficazes de monitoramento de segurança em patrimônios públicos e privados é algo que cada vez mais se faz prioritário para a preservação de imóveis e equipamentos contra arrombamentos e furtos. De acordo com este cenário, este artigo tem como objetivo apresentar o desenvolvimento de uma aplicação web que permite o monitoramento patrimonial de forma eficiente e com baixo custo. O sistema nomeado **Patrimônio+Seguro**, utilizará imagens de reproduções de câmeras IP, colhidas através de rede Wi-Fi por meio do protocolo RTSP, para que estas gravações possam ser acessadas remotamente, através de um sistema web. O armazenamento das imagens colhidas é algo essencial também para que se tenha o histórico das gravações e para isto o sistema realizará uma integração com o Google Drive para realizar o upload dos vídeos, no padrão de compressão H.264, de forma periódica numa conta de serviço. A aplicação poderá ser acessada mediante o e-mail e senha cadastrados na base de dados, MongoDB. Este banco de dados foi escolhido por sua simples configuração e adaptabilidade com a linguagem Python através da Mongo Engine.

Palavras-chave: Câmera IP. Monitoramento por imagens. Automatização. Python. Sistema web. Armazenamento em nuvem.

Instituto Federal de Pernambuco. Campus Recife. Curso Tecnólogo de Análise e Desenvolvimento de Sistemas. 30 de Novembro de 2022.

ABSTRACT

Effective security monitoring mechanisms in public and private assets is something that is increasingly becoming a priority for the preservation of properties and equipment against break-ins and theft. According to this scenario, this article aims to present the development of a web application that will help people to have an efficient way of monitoring these assets. The system named Patrimônio+Seguro will use images of reproductions from IP cameras, collected through a Wi-Fi network using the RTSP protocol, so that these recordings can be accessed remotely, through a web system. The storage of the collected images is also essential to have the recording history and for this the system will integrate with Google Drive to upload the videos, in the H.264 compression standard, periodically in a service. The application can be accessed through the email and password registered in the database, MongoDB. This database was chosen for its simple configuration and adaptability to the Python language through the Mongo Engine.

Keywords: IP camera. Image monitoring. Automation. Python. Web system. Cloud storage.

1. INTRODUÇÃO

A segurança patrimonial, tanto privada quanto pública, é um tema que cada vez mais tem sido discutido e aprimorado para que possamos ter mecanismos eficientes que protejam e minimizem as chances de assaltos e invasões. Não é de hoje que crimes ao patrimônio é um grande problema social, onde os infratores que cometem estes crimes têm como principal alvo locais com pouca segurança, luminosidade e monitoramento. Através de um relatório sobre a segurança no campus da Universidade Federal de Goiás, fruto de um estudo motivado pelos múltiplos assaltos em seus arredores, foi constatado que locais onde há vigilância e infraestrutura (câmeras, iluminação, passarelas, porteiros/ vigias), há sensação de segurança. Quando esse tipo de aparato diminui ou inexiste, a representação de perigo projeta-se sobre a instituição. (MARTINS et al., 2015).

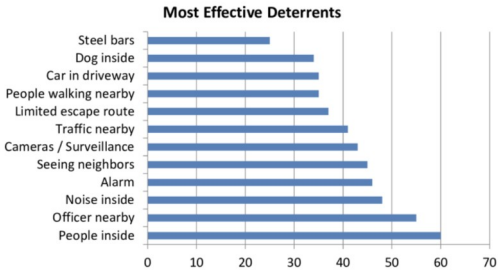
De acordo com CAPUTO (2014):

Se um criminoso está ciente da possibilidade de ser vigiado e registrado, ele pode determinar que o risco de detecção supera em muito os benefícios.

O uso de câmeras de segurança como recurso para coibir ações criminosas a patrimônios e ajudar na identificação de criminosos, é uma prática amplamente utilizada no mundo inteiro e vem crescendo cada dia mais. De acordo com um estudo realizado pelo Departamento de Justiça Criminal e Criminologia da Universidade da Carolina do Norte em Charlotte (EUA), foram analisados fatores que ajudam a entender sob a perspectiva dos infratores, dentre outras coisas, quais mecanismos os motivam ou convencem a desistir de cometer um crime de roubo a um patrimônio. Segundo o levantamento, para cerca de 45% das respostas fornecidas por estes infratores, a percepção de câmeras de segurança no local os faria desistir de cometer o

delito. (JOSEPH B. KUHNS, 2012)

Figura 1: Motivações para desistência de crime a um patrimônio.



Fonte: https://www.researchgate.net/figure/Perception-of-Effectiveness-of-Burglary-Deterrents-According-to-Burglars-of-sample_fig2_268444817

Este artigo tem como objetivo apresentar o desenvolvimento de um sistema web de código aberto, que de forma gratuita, possa permitir o monitoramento por câmeras IP e o acesso às imagens dessas câmeras em tempo real, através da web. A primeira motivação para a construção desta aplicação foi a necessidade constatada pelo Centro de Pesquisa do Campus Recife do Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco, onde se faz necessário possuir um mecanismo de monitoramento do prédio onde ficam localizados vários laboratórios de pesquisa da instituição.

Este trabalho seguirá uma organização mediante esta ordem: a seção 1 tem como objetivo introduzir o leitor ao tema proposto; a seção 2 apresenta a revisão de literatura para o aprofundamento do trabalho; as seções 3, 4 e 5 se referem à metodologia da pesquisa, na qual, a partir da revisão bibliográfica, foram definidas quais ferramentas, métodos e técnicas foram utilizados para compor o sistema proposto, além de algumas definições das tecnologias relacionadas; na seção 6 abordaremos as etapas adotadas para o desenvolvimento do sistema, englobando a modelagem do banco de dados, objetivos da aplicação e tecnologias utilizadas para construção do sistema web; na seção 7 serão exibidos os resultados e discussões obtidos a partir da aplicação das técnicas definidas na metodologia, incluindo as discussões gerais do trabalho; por fim, na seção 8 serão apresentadas as conclusões e indicações para trabalhos futuros.

O sistema ***Patrimonio+Seguro***, solução proposta fruto deste artigo, além de ter sido construído com o objetivo de atender a necessidade do IFPE em manter a segurança e o monitoramento dos bens de seus laboratórios de pesquisa, tem flexibilidade de poder ser implantado em qualquer estabelecimento, seja público ou privado, que possua câmeras IP conectadas a rede do servidor que terá o sistema implantado.

2. CÂMERAS IP

Internet Protocol Camera ou simplesmente Câmera IP, é um tipo de câmera que é largamente utilizado no mundo como ferramenta para monitoramento e segurança. Este tipo de câmera, diferentemente das tradicionais câmeras CCTV (closed-circuit television camera), possuem a capacidade de enviar e receber dados via internet. (TAKEOGLU; TOSUN, 2015)

Originalmente, câmeras IP possuem a capacidade de armazenar suas gravações localmente ou no caso de algumas fabricantes, em nuvem, como na fabricante EZVIZ Inc., um dos fabricantes das câmera IP utilizadas na implementação do artefato de software apresentado neste artigo..

Além disso, esse tipo de câmera por estar conectada a uma rede de internet, permite que estas câmeras possam ser acessadas tanto local quanto remotamente.

O modelo da câmera IP utilizada para o desenvolvimento da aplicação implementada neste artigo foi a descrita na imagem abaixo.

Figura 2: Câmera IP modelo CS-C1C-D0-1D1WFR, utilizada durante o desenvolvimento do sistema Patrimônio+Seguro

Fonte: https://www.ezviz.com/br/product/c1c/9686

No modelo CS-C1C-D0-1D1WFR, utilizado no desenvolvimento da solução proposta, permite a imprescindível característica de modo noturno, onde temos a possibilidade de realizar gravações à noite através de tecnologia infravermelho e com isso, de forma eficaz, se é possível visualizar nitidamente mesmo no escuro. Além desta câmera possuir uma lente grande-angular de 110°/130°, permitindo uma grande amplitude de gravação (EZVIZ, c2022).

3. PROTOCOLO DE FLUXO CONTÍNUO EM TEMPO REAL (RTSP)

O acesso a uma câmera IP, através da Wi-Fi, pode se dar por meio de dois tipos de protocolos: o HTTP ou o RTSP. Para o desenvolvimento deste artigo foi utilizado o protocolo RTSP, que foi definido pela RFC 2326, onde pode ser caracterizado como um protocolo a nível de aplicação que foi implementado para executar junto a outros protocolos, RTV e RSVP por exemplo, para que seja possível a ter a completude de um serviço de internet. (KUROSE; ROSS, 2010)

Este protocolo foi criado e desenvolvido pelas empresas *RealNetworks* e *NetScape Communications* juntamente com a *Columbia University*, onde em 1996 foi apresentado para o IETF para ser considerado um novo padrão de protocolo para internet. Mas foi só em 1998 que após várias alterações foi aprovado pela IETF. (RFC2326, 1998)

Diferentemente do protocolo HTTP que comumente é utilizado para a transmissão de dados do tipo texto e imagens, o protocolo RTSP tem o objetivo de realizar serviços de streams de áudio e vídeo de forma contínua entre servidores e clientes. Estes serviços podem ser consumidos através de uma URL iniciando com rtps://. (KUROSE; ROSS, 2010)

Além dos pontos destacados acima que diferem os protocolos HTTP e RTSP, podemos destacar que enquanto o protocolo HTTP não possui um "estado", o RTSP mantém sessões para que possa interligar as subsequentes requisições para o stream de dados. (KUROSE; ROSS, 2010)

4. CODEC H.264

O armazenamento de gravações de câmeras de vídeo, principalmente quando envolve armazenamento em nuvem, é um

desafio que deve ser visto com bastante cuidado uma vez que caso não seja aplicado nenhuma tecnologia de compressão se pode algo que pode ser bem frustrante. Arquivos muito "pesados" exigem muito da largura de banda e em muitos casos podem haver perda de conexão devido ao limite desta largura. Além disso, a própria questão da quantidade de armazenamento fica bastante onerada, uma vez que o tamanho dos arquivos podem ser bastante extensos.

Para evitar os problemas citados anteriormente, a indústria criou diversos tipos de tecnologias, denominadas de codecs, que permitem realizar essas compressões de vídeo. Dentre os codecs mais populares, podemos citar: H.264 (MPEG-4 Part 10 AVC), MPEG-2, MPEG-4 Part 2 e o Windows Media Video (WMV). Cada codec tem suas especificações e aplicações mais recomendadas.

Para o software desenvolvido a partir deste artigo a tecnologia de compressão H.264 se mostrou uma excelente alternativa para que se possa realizar uma diminuição drástica dos arquivos de vídeo que serão transmitidos pela rede para a nuvem com pouca perda de qualidade.

A maior parte das técnicas de compressão de vídeo se dão através da disparidade entre quadros com o intuito de reduzir o tamanho do arquivo. Existem dois tipos de abordagens, utilizando o tratamento de diferenças único quadro ou entre quadros. No caso de um único quadro as técnicas de compressão trabalham na premissa de que os olhos humanos não conseguem captar determinadas diferenças entre algumas faixas de cores. Já quando a abordagem é voltada para o tratamento entre quadros, unicamente a transição de um quadro para o outro são atacadas. Sendo assim, ignorando redundâncias de pixels temos a compressão do vídeo e consequentemente a diminuição de seu tamanho.(apud BRAVO, 2015).

Dentre as principais características do codec H.264 podemos citar as seguintes:

- Aumento considerável da eficiência de codificação fornecendo uma

redução média de taxa de bits de 50%;

- Robustez de erro, ofertando uma série de ferramentas para prevenção, detecção e correção de erros;
- Codificação de vídeo com ótima qualidade, com recursos de baixa latência e melhor qualidade para maior latência;
- Documentação clara de sintaxe, com o objetivo de deixar mais simples as implementações deste codec. (apud BRAVO, 2015)

5. ARMAZENAMENTO EM NUVEM

Uma das definições mais importantes quando se pensa na construção de um software, seja ele mobile, desktop, um sistema web ou mesmo um jogo, é o local onde ficarão armazenados os dados que serão utilizados pela aplicação. Para o desenvolvimento do sistema proposto neste artigo, consideramos a opção de realizar o armazenamento dos dados em nuvem, através do Google Drive.

De acordo com AWS (c2022), o armazenamento na nuvem é um modelo de computação em nuvem que, através de provedores de computação, realiza o armazenamento de dados na internet. Esses servidores são responsáveis por gerenciar e

operar os dados como serviços que são disponibilizados.

Ainda segundo a AWS (c2022), "Os aplicativos acessam o armazenamento na nuvem por meio de protocolos de armazenamento tradicionais ou diretamente usando uma API. Muitos fornecedores oferecem serviços complementares criados para ajudar a coletar, gerenciar, proteger e analisar dados em escala massiva."

De acordo com Odun-Ayo (2017), uma das grandes vantagens de se armazenar dados em nuvem é que esses dados estão em diversos locais espalhados pelo mundo, em servidores, assim aumentando a segurança desses dados contra catástrofes que podem vir a ocorrer caso estes dados fossem armazenados localmente num HD.

A opção de realizar o armazenamento em nuvem para as gravações das câmeras IP foi definida considerando que a instituição IFPE Campus Recife possui parceria com o Google e com isso possuindo uma grande capacidade de armazenamento de dados no Google Drive, a partir de contas institucionais. Dentre as principais vantagens que temos com essa abordagem é evitar que haja uma grande quantidade de dados localmente nos servidores do Instituto uma vez que estamos visando ter um histórico de gravações de vídeo, mesmo que compactadas.

6.ETAPAS ADOTADAS

No desenvolvimento do sistema web **Patrimônio+Seguro**, foram adotadas as seguintes etapas descritas abaixo:

a) Revisão bibliográfica sobre os sistemas existentes de câmeras IPs tradicionais de mercado com a finalidade de entender o escopo destes sistemas e se eles já atenderiam a necessidade da aplicação resultante deste artigo;

b) Estudo sobre as principais tecnologias atuais de mercado que atenderiam aos requisitos do sistema, considerando reproduções simultâneas de câmeras IP de forma segura, realizando o armazenamento temporário localmente e integrando periodicamente os vídeos armazenados no Google Drive através da Google Drive API, onde a linguagem escolhida foi Python utilizando o framework Flask juntamente com o OpenCV;

c) Análise, planejamento das APIs necessárias para serem consumidas do Google Drive API a fim de realizar o upload dos vídeos, criação de pastas e expurgo de dados de forma periódica;

d) Estudo sobre o banco de dados utilizado para realizar o armazenamento dos usuários do sistema e dos dados das câmeras IPs. O banco de dados escolhido foi o MongoDB.

7 TECNOLOGIAS UTILIZADAS

A escolha das tecnologias utilizadas no desenvolvimento do sistema web **Patrimônio+Seguro** foi pensada considerando quais ferramentas atenderiam de forma mais precisa aos requisitos do sistema. Segue a descrição sumária das principais tecnologias utilizadas na aplicação web nas próximas sessões.

7.1 PYTHON

O sistema web foi desenvolvido através da linguagem orientada a objeto, Python. A escolha desta

tecnologia se deu pela sua baixa curva de aprendizado comparado a outras linguagens como Java e por ter bibliotecas que atenderiam de forma eficiente aos requisitos do sistema web.

"Python é uma linguagem fácil de aprender e poderosa. Ela tem estruturas de dados de alto nível eficientes e uma abordagem simples mas efetiva de programação orientada a objetos. A elegância de sintaxe e a tipagem dinâmica de Python aliadas com sua natureza interpretativa, o fazem a linguagem ideal para programas e desenvolvimento de aplicações rápidas em diversas áreas e na maioria das plataformas.", (PYTHON, 2022c)

De acordo com AWS (2022c), a linguagem python entre seus principais benefícios, incluem:

- Fácil entendimento e sintaxe básica semelhante a do inglês;
- Alta produtividade pois comparado a outras linguagens de programação, utiliza de poucas linhas de código para resolução de algoritmos;
- Riqueza de bibliotecas para que desenvolvedores não precisem "reinventar a roda" em suas soluções;
- Grande e ativa comunidade ao redor do mundo, o que facilita para pesquisas sobre as principais features e problemas que desenvolvedores possam vir a ter em suas implementações;
- Fácil portabilidade entre Sistemas Operacionais;
- Apesar de ser orientada a objetos, a linguagem também aceita outros tipos de programação, como estruturada e funcional.

7.2 FLASK

Flask é um *framework* Python voltado para deixar o desenvolvimento web mais simples e rápido.

De acordo com GRINBERG (2018), o Flask, comparado a outros frameworks é tão pequeno que pode ser chamado de "*microframework*", onde uma vez que se trabalhe com ele, um desenvolvedor aprenderá todo o funcionamento de seu código-fonte.

Porém, este pequeno tamanho não significa que ele não seja uma ferramenta poderosa. Flask está entre as bibliotecas Python mais utilizadas no mundo, sendo o principal concorrente ao Django.

Ainda segundo GRINBERG (2018), Flask foi desenhado para ser uma ferramenta extensível, ou seja, que tenha apenas as configurações básicas para o seu funcionamento e uma vez que seja necessário realizar algo mais complexo, sejam feitas as instalações necessárias. Esta característica ajuda a garantir que aplicações Python que rodem através do Flask sejam o máximo de enxutas quanto possível.

Considerando o modelo MVC (*Model-View-Controller*), o *framework* Flask foi utilizado com a finalidade de reduzir o tempo de desenvolvimento dos serviços da camada controladora no back-end da aplicação web. Estes serviços são consumidos pela camada de visão da aplicação, através de requisições HTTP entre o front-end e o back-end em Python.

7.3 OPEN SOURCE COMPUTER VISION LIBRARY (OPENCV)

A biblioteca OpenCV é um framework amplamente utilizado no

desenvolvimento de software, promovendo uma série de funcionalidades que deixam muito mais simples a realização de conexões, armazenamento de vídeos e transmissões ao vivo com câmeras IP, através de protocolos HTTP ou RTS. O OpenCV é uma ferramenta que possui adaptações para diferentes linguagens de programação, inclusive Python, através da biblioteca *opencv-python*.

De acordo com OpenCV (2022c), o OpenCV (Open Source Computer Vision Library) é uma biblioteca de software voltada para imagens computacionais e aprendizado de máquina de código aberto. O OpenCV foi construído para fornecer uma infraestrutura comum para aplicativos de visão computacional e acelerar o uso da percepção da máquina nos produtos comerciais. Além disso, possui interfaces para C++, Python, Java e MATLAB e suporta Windows, Linux, Android e Mac OS. O OpenCV se inclina principalmente para aplicativos de imagens em tempo real.

O OpenCV, foi uma ferramenta crucial para o desenvolvimento das funcionalidades do projeto que envolviam o tratamento das imagens das câmeras e do armazenamento dos vídeos. Anexando este framework ao fato de que Python é uma linguagem de programação de sintaxe simples, através de poucas linhas de código, a solução final pôde ser alcançada com pouco esforço satisfatoriamente.

Figura 3: Método criado para gerar os frames das câmeras, utilizando OpenCV

```python
def gen_frames_by_ip_to_view(cap, camera_matrix_size):
    while True:
        success, frame = cap.read()
        if success:
            try:
                resized_frame = resize_img_from_matriz_size(frame, camera_matrix_size)
                ret, buffer = cv2.imencode('.jpg', resized_frame)
                resized_frame = buffer.tobytes()
                yield (b'--frame\r\n'
                    b'Content-Type: image/jpeg\r\n\r\n' + resized_frame + b'\r\n')
            except Exception as e:
                logging.error('Error getting frames to store >> %s', e)
                logging.exception(e)
                print(e)
                pass
        else:
            pass
```

Fonte: https://github.com/dmsb/security-camera-ifpe/blob/main/securityCameraServices.py

Figura 4: Método criado para armazenar os frames das câmeras, utilizando OpenCV

```python
out = cv2.VideoWriter(file_location + file_name, fourcc, 20, (frame_width, frame_height))

current_seconds = time.time()

while True:
    success, frame = cap.read()
    if success:
        try:
            if time.time() - current_seconds <= RECORDING_TIME_IN_SECONDS:
                out.write(frame)
            else:
                out.release()
                service_acc_parameters = (file_location, file_name)
                thread_to_upload_saved_video = Thread(target=upload_videos_to_google_drive, args=(service_acc_parameters,))
                thread_to_upload_saved_video.start()
                __get_frames_to_store(cap, camera)
        except Exception as e:
            print(e)
            break
    else:
        logging.info('Error getting frames to store >> %s' % (camera.mac_address))
        break
```

Fonte: https://github.com/dmsb/security-camera-ifpe/blob/main/videoLocalStorer.py

7.4 BOOTSTRAP

O Bootstrap é uma ferramenta amplamente utilizada no mundo inteiro para desenvolvimento de telas para sites e sistemas web. Através de um conjunto de estilizações e funções Javascript, podemos criar de forma otimizada e com menor tempo de desenvolvimento, telas complexas e com bastante fluidez. Para o projeto Patrimônio+Seguro, o Bootstrap foi a tecnologia base por trás do front-end, onde suas dependências foram utilizadas a partir de uma extensão do Flask: *Bootstrap-Flask*.

Segundo LI (2017c), "Bootstrap-Flask é uma coleção de macros Jinja para Bootstrap e Flask. Ele ajuda você a renderizar dados e objetos relacionados ao Flask para HTML de marcação Bootstrap mais facilmente."

Figura 5: Importação do flask-bootstrap do Flask

```
from flask_bootstrap import Bootstrap5
from flask import Flask

app = Flask(__name__)

bootstrap = Bootstrap5(app)
```

Fonte: https://bootstrap-flask.readthedocs.io/en/stable/basic/#installation

Figura 6: importações das estilizações CSS e do JS no front-end

```
<head>
....
{{ bootstrap.load_css() }}
</head>
<body>
...
{{ bootstrap.load_js() }}
</body>
```

Fonte: https://bootstrap-flask.readthedocs.io/en/stable/basic/#installation

7.4 MONGODB

O MongoDB é um banco de dados NoSQL baseado em documentos, que dentre suas principais características podemos citar alta performance, API para consultas, alta disponibilidade e escalabilidade horizontal (MONGODB, 2022c).

Ainda segundo MONGODB (2022c), um banco de dados baseado em documentos é caracterizado quando um registro é classificado como um documento. Um documento é uma estrutura de dados composta por pares de campo e valor que são parecidos com objetos do tipo *JSON*.

Figura 7: Exemplificação de um documento no MongoDB

Fonte: https://www.mongodb.com/docs/manual/introduction/

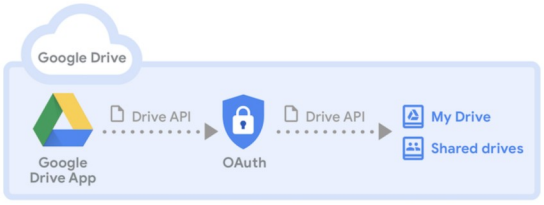
A escolha de usar o MongoDB foi definida pela necessidade de se ter um banco de dados seguro, simples de configurar e de fácil integração com a tecnologia Python.

Para a modelagem do sistema Patrimônio+Seguro, se fez necessário a criação de dois documentos no MongoDB, um para representar os usuários e outro para representar as câmeras.

7.5 GOOGLE DRIVE API

O Google Drive foi a plataforma escolhida para ter as gravações das câmeras armazenadas em nuvem. Através da Google Drive API, foi possível concretizar todas as funcionalidades necessárias para o gerenciamento das gravações dentro do período de 30 dias na plataforma do Google.

Figura 8: Arquitetura do Google Drive API



Fonte: https://developers.google.com/drive/api/guides/about-sdk

Segundo GOOGLE (2022c), A API do Google Drive é uma ferramenta

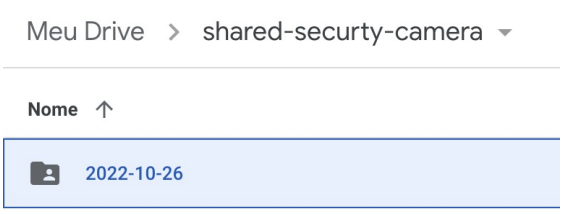
que permite que desenvolvedores criem aplicações que possam acessar o armazenamento na nuvem do Google Drive, através de funcionalidades robustas. Dentre as funcionalidades, as utilizadas no sistema web Patrimônio+Seguro foram: criação de pastas, upload de arquivos de grandes tamanhos e deleção de arquivos e pastas.

Realizar o upload de grandes através de requisições HTTP pode ser bastante desafiador para muitas aplicações, uma vez que requisições podem se tornar mais demoradas de forma proporcional ao quão maior for o tamanho do payload de cada requisição. Para o sistema proposto neste artigo, o upload de vídeos para o Google Drive se dará a cada 30 minutos de gravação, tendo assim, em média, arquivos de 500 MB.

Para que fosse possível realizar o upload destes arquivos foi necessário realizar o envio por partes de cada arquivo de vídeo, utilizando a propriedade "*resumable*" do Google Drive para a API de upload de arquivos. Com esta propriedade, conseguimos, em múltiplas requisições, enviar o arquivo que queremos realizar o upload delimitando a quantidade de bytes que cada requisição irá ter até que sejam enviados todos os bytes do arquivo em questão. A quantidade máxima de bytes que os payloads do Google Drive aceita é de 5 MB.

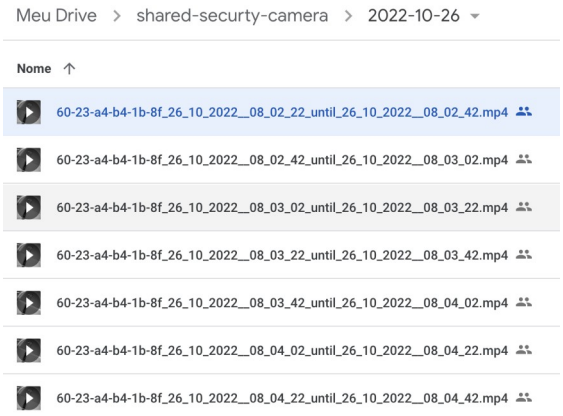
O sistema web diariamente irá criar uma pasta no Google Drive representando as gravações daquele dia. Nesta pasta, será realizado o upload de cada arquivo de vídeo representando a gravação de cada câmera cadastrada no sistema.

Figura 9: Exemplo de pasta criada no Google Drive:



Fonte: Google Drive

Figura 10: Exemplo de arquivos de vídeos criados no Google Drive:



Fonte: Google Drive

7.6 GOOGLE SERVICE ACCOUNT

Contas de serviço do Google é um tipo de conta criado para possibilitar o consumo das APIs do Google de forma não personificada, característica ideal para aplicações que precisam consumir APIs do Google sem a necessidade de autenticação de um usuário personificado Google, evitando a necessidade de passar pela etapa da tela de consentimento.

Tela de consentimento é uma página web, processada em tempo de autenticação em aplicações que estão tentando acessar algum recurso do Google através de suas APIs, por meio de um usuário Google personificado tradicional.

8. IMPLEMENTAÇÃO

O desenvolvimento da aplicação **Patrimônio+Seguro** se deu a partir dos seguintes passos:

a) Criação da base de dados local utilizando o MongoDB. Nesta base de dados temos 2 estruturas de dados, uma representando os usuários, com todas as informações pertinentes para o sistema, incluindo os dados de login e senha para autenticação. Além da estrutura de dados para usuários, temos uma para representar as câmeras IP, contendo informações como localização da câmera, MAC e se ela está ativa ou não;

b) Implementação da solução do sistema web, utilizando Python como linguagem de programação do servidor juntamente com o framework para desenvolvimento web, Flask, além do OpenCV para a conexão, renderização e armazenamento com as câmeras IP cadastradas. Com isto, o sistema permitirá que os usuários devidamente cadastrados e autenticados na aplicação possam realizar a visualização através web de forma remota, das câmeras IP cadastradas e ativas na mesma rede de internet que o sistema web está sendo executado;

c) Implementação das threads que executam em background para o armazenamento em nuvem das gravações. Uma vez que o sistema web Patrimônio+Seguro está rodando, se é capturado através do endereço MAC de cada câmera, a listagem de todos os IPs das câmeras IP cadastradas na mesma rede de internet. Esses endereços MAC estão armazenados no banco de dados MongoDB, juntamente com as demais informações de cada câmera cadastrada. Com os IPs resgatados, é feita a conexão via RTSP com cada câmera e assim a aplicação resgatará cada frame e escreverá localmente os vídeos num arquivo com a extensão mp4, utilizando a tecnologia de compressão H.264. Uma vez feito o armazenamento de cada vídeo no formato mp4, será iniciada uma integração com o Google através da API do Google Drive para realizar o upload de cada arquivo e após a conclusão do upload, realizar a deleção do vídeo localmente.

d) Implementação do expurgo de dados do Google Drive, onde se foi implementada uma verificação para deleção das gravações em nuvem mais antigas, considerando que uma vez que o sistema identifique que existam 30 dias de gravações, o primeiro dia destes 30 se é deletado, para que o limite de armazenamento da nuvem não seja comprometido.

9. CONCLUSÕES E FUTUROS TRABALHOS

Neste trabalho foi construído um sistema web baseado em Python para viabilizar tanto o acesso de forma remota a câmeras IP conectadas quanto as suas gravações, disponibilizando-as em nuvem. Com isto, a aplicação web Patrimônio+Seguro tem o objetivo de ser uma ferramenta que potencializará o monitoramento do laboratório de informática do Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco. De acordo com a arquitetura da solução proposta e as tecnologias utilizadas nesta solução, o software construído tem a capacidade de ser implantando em diferentes ambientes, não se resumindo apenas ao IFPE.

Para trabalhos futuros com fins de melhoria do software construído, poderia ser relevante o uso mais sofisticado do framework OpenCV, possibilitando ações como

identificação facial e integração com alarmes e/ou setores policiais para que as ações de reação a possíveis crimes possam ser feitas de maneira mais rápida possível.

REFERÊNCIAS

NSC TOTAL, **Com monitoramento eletrônico, 94% das tentativas de roubo a propriedades protegidas terminam em fracasso**. Disponível em https://www.nsctotal.com.br/noticias/com-monitoramento-eletronico-94-das-tentativas-de-roubo-a-propriedades-protegidas-terminam. Acesso em 4 out. 2022.

JOSEPH B. KUHNS, Understanding decisions to burglarize from the offender's perspective, 2012. Disponível em <https://www.researchgate.net/publication/268444817_Understanding_Decisions_to_Burglarize_from_the_Offender's_Perspective>. Acesso em 8 de out. 2022.

https://www.ufg.br/n/82009-divulgado-relatorio-sobre-a-seguranca-nos-campus-da-ufg?atr=pt-BR&locale=pt-BR

MARTINS, M., **Violência, conflitos e crimes nos Campus Universitários: Subsídios para a política de segurança da UFG**. Disponível em https://files.cercomp.ufg.br/weby/up/789/o/Relat%C3%B3rio_Sint%C3%A9tico_NECRIVI___revisado.pdf. Acesso em 3 out. 2022.

RIBEIRO, L. T., **OLHARES VIVOS EM OLHOS DE VIDRO: A VIGILÂNCIA POR MEIO DE CÂMERAS DE MONITORAMENTO NO BAIRRO DE BOTAFOGO**. Disponível em https://periodicos.ufjf.br/index.php/csonline/article/view/17433. Acesso em 3 out. 2022.

A. Tekeoglu and A. S. Tosun, "Investigating Security and Privacy of a Cloud-Based Wireless IP Camera: NetCam," 2015 24th International Conference on Computer Communication and Networks (ICCCN), 2015, pp. 1-6.

Popovic, Gradimirka & Arsić, Nebojša & Jakšić, Branimir & Gara, Boris & Petrovic, Mile. (2013). Overview, Characteristics and Advantages of IP Camera Video Surveillance Systems Compared to Systems with other Kinds of Camera. 2. 356-362.

Caputo, A., **Digital Video Surveillance and Security**, 2. ed. Oxford: The Butterworth-Heinemann. 2014.

http://consad.org.br/wp-content/uploads/2013/02/A-IMPORT%C3%82NCIA-DA-MELHORIA-DA-QUALIDADE-DO-GASTO-P%C3%9ABLICO-NO-BRASIL-PROPOSTAS-PR%C3%81TICAS-PARA-ALCAN%C3%87AR-ESTE-OBJETIVO1.pdf

EZVIZ. EZVIZ Creating Easy Smart Homes, c2022. Página EZVIZ C1C - Câmera Wi-Fi interna com resolução HD. Disponível em https://www.ezviz.com/br/product/c1c/9686. Acesso em 17 out. 2022.

BRAVO, R. **INFRAESTRUTURA DE MONITORAMENTO REMOTO POR VÍDEO UTILIZANDO CÂMERAS IP E UM DISPOSITIVO ARM**, 2015. Página EZVIZ C1C - Câmera Wi-Fi interna com resolução HD. Disponível em http://repositorio.utfpr.edu.br/jspui/handle/1/23137. Acesso em 17 out. 2022.

VERINT. **Video Compression Standards: Selecting the Right Video Codec**. Disponível em http://www.bgwt.com.au/assets/files/Video-CompressionStandards.pdf. Acesso em 18/10/2022.

KUROSE, James F.; ROSS, Keith W. Redes de Computadores e a Internet: Uma Abordagem Top-Down. 5ª Edição, São Paulo, SP: Pearson Addison-Wesley, 2010.

STANIMIROVIC, U. Brid.TV, 2021. Página What Is H.264 Video Codec and How It's Shaping the Online Video World. Disponível em https://www.brid.tv/what-is-h264-video-codec. Acesso em 18 out. 2022.

Odun-Ayo, Isaac & Ajayi, Olasupo & Akanle, Matthew & Ahuja, Ravin. (2017). An Overview of Data Storage in Cloud Computing. 29-34. 10.1109/ICNGCIS.2017.9.

AWS. Amazon Web Services, c2022. Página O que é armazenamento na nuvem?. Disponível em https://aws.amazon.com/pt/what-is-cloud-storage/. Acesso em 21 out. 2022.

AWS. Amazon Web Services, c2022. Página O que é Python. Disponível em https://aws.amazon.com/pt/what-is/python/#:~:text=O%20Python%20%C3%A9%20uma%20linguagem,executada%20em%20muitas%20plataformas%20diferentes. Acesso em 21 out. 2022.

PYTHON. Python Software Foundation, c2022. Página O Tutorial de Python. Disponível em https://docs.python.org/pt-br/3/tutorial/index.html#tutorial-index. Acesso em 21 out. 2022.

GRINBERG, M., **Flask Web Development**, 2. ed. O'Reilly Media. 2018.

OpenCV. OpenCV team, c2022. Página "About". Disponível em https://opencv.org/. Acesso em 22 out. 2022.

LI, G. Bootstrap-Flask, c2017. Página Inicial. Disponível em https://bootstrap-flask.readthedocs.io/en/stable/#. Acesso em 23 out. 2022.

MONGODB. MongoDB Inc, c2022. Página Documentação. Disponível em https://www.mongodb.com/ Acesso em 24 out. 2022.

GOOGLE. Google, c2022. Página Documentação. Disponível em https://developers.google.com/drive. Acesso em 24 out. 2022.