



## Manual de para desenvolvimento de novos Casos de Uso para o AIV

Esse documento mostra os conceitos e passos necessários para a criação de um novo caso de uso de Video Analytics para o AIV.

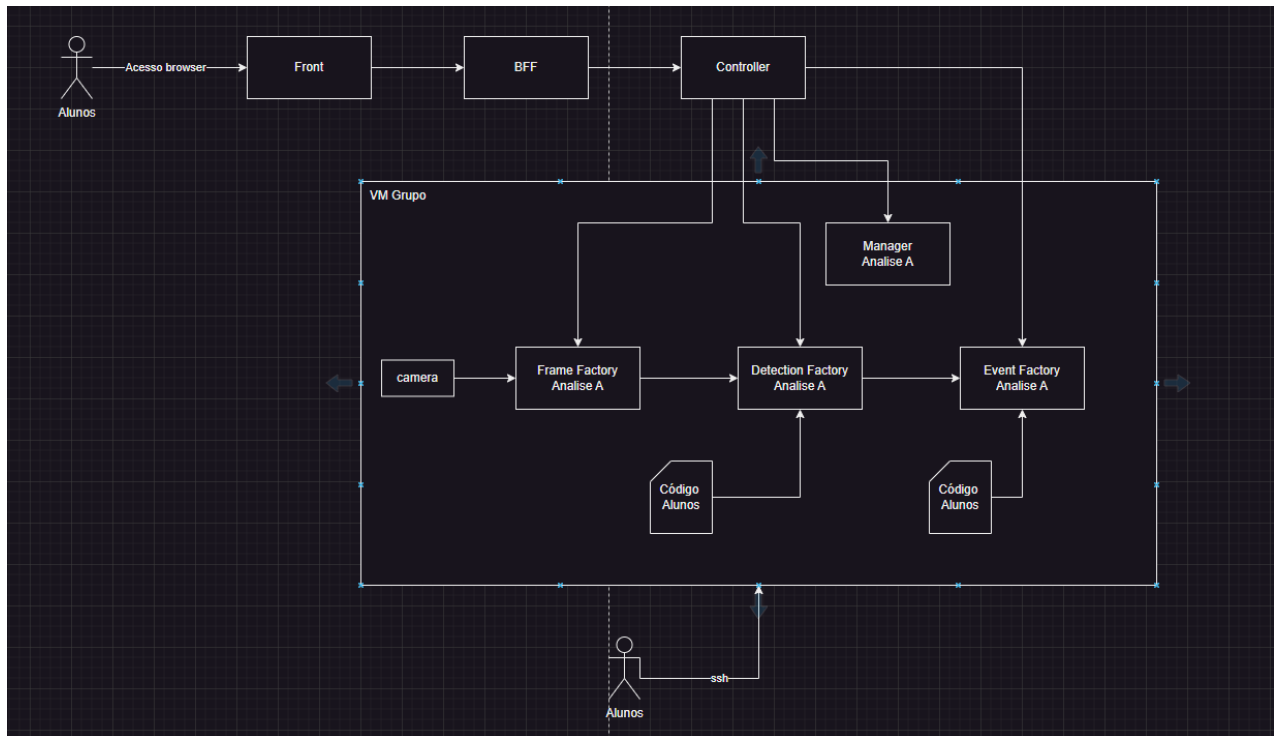
## Sumário

Manual de para desenvolvimento de novos Casos de Uso para o AIV.....	1
Sumário .....	2
Componentes do sistema .....	3
O fluxo de dados de uma análise .....	3
Captura.....	3
Processamento.....	4
Consolidação .....	4
Bibliotecas e contratos.....	5
Detection Factory.....	5
Event Factory .....	8
Passo a passo completo para os alunos.....	11
Acessar a máquina virtual do grupo .....	12
Incluir um vídeo novo na máquina para utilizar como stream de câmera .....	13
Para verificar se seu vídeo está na máquina.....	13
Criar uma análise .....	14
Passo 1 .....	14
Passo 2 .....	15
Passo 3 .....	15
Passo 4 .....	16
Passo 5 .....	17
Executar os componentes e códigos para iniciar a análise.....	19
Verificar se a análise está rodando .....	20
Conectar a IDE na VM .....	21
Visual Studio.....	21
VIM.....	23
Modificar componentes.....	25
Detection factory .....	25
Event factory.....	25
Cancelar análise .....	27

## Componentes do sistema

O aluno irá acessar o sistema de duas formas: pelo site da plataforma e por uma conexão com a máquina virtual de desenvolvimento.

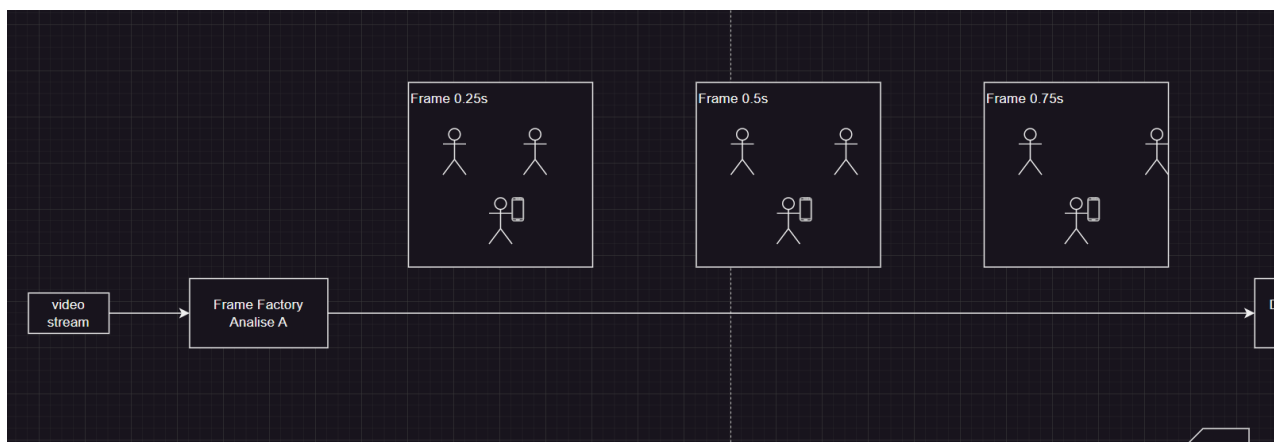
Dentro da máquina virtual, o aluno irá interagir com seus códigos fontes a partir do host, e os códigos serão executados dentro de containers docker, gerenciados pelo sistema.



## O fluxo de dados de uma análise

### Captura

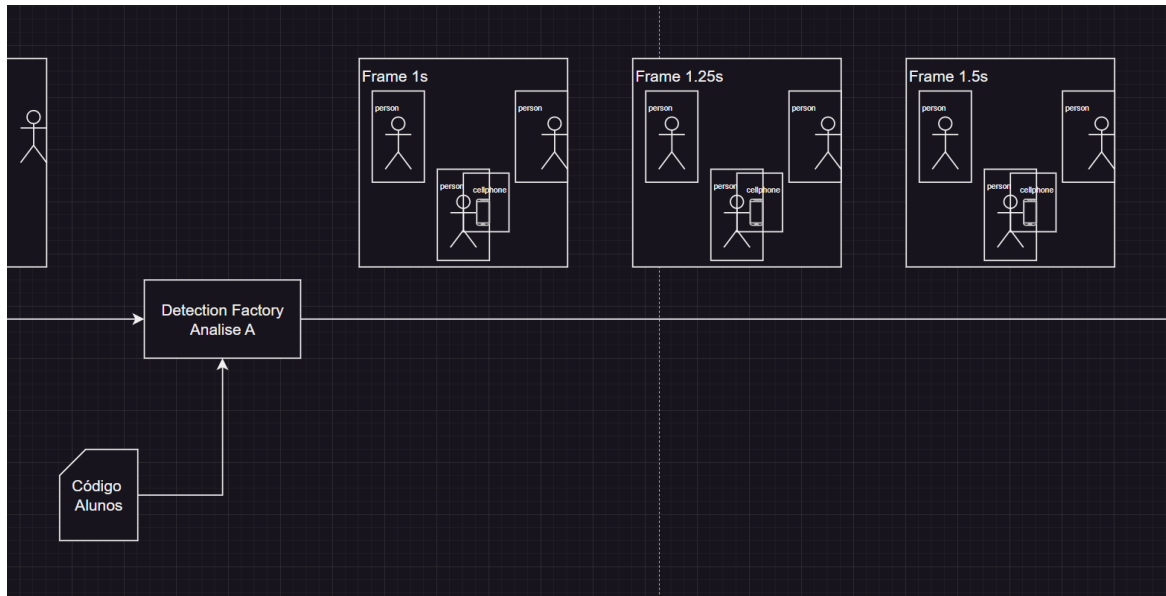
Uma análise se inicia com a captura de uma stream de vídeo simulado pelo sistema pelo componente que chamamos de Frame Factory, que corta o vídeo em frames e insere esses frames no pipeline de processamento da análise.



## Processamento

O frame então passa pelo componente que chamamos de Detection Factory, cuja responsabilidade é realizar inferências através de uma IA para determinar quais entidades estão presentes dentro do frame.

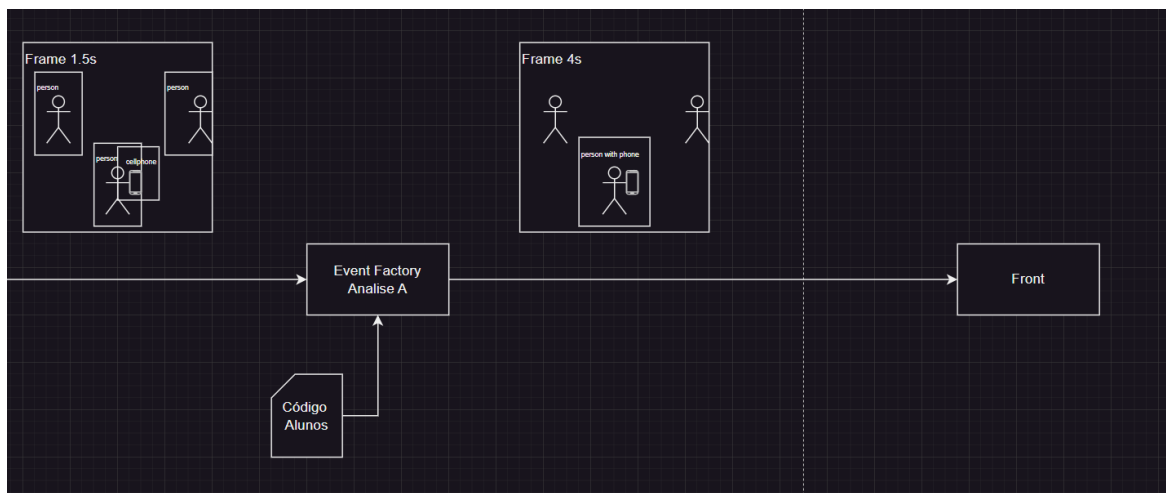
A lógica do componente é composta pelo código do aluno que chama uma biblioteca da infraestrutura do AIV, e irá lidar com toda comunicação do sistema.



## Consolidação

Após inferência da IA e coleta das entidades identificadas na imagem, os frames passam pelo Event Factory, cuja responsabilidade é avaliar o contexto de um frame processado recebido com todos os frames anteriores e decidir em qual momento (frame) um evento deve ser iniciado, e quais entidades devem ser exibidas.

Assim como na Detection Factory, a lógica do componente é composta pelo código do aluno que chama uma biblioteca da infraestrutura do AIV, e irá lidar com toda comunicação do sistema.



## Bibliotecas e contratos

Os Códigos da Detection Factory e Event Factory são iniciados a partir do código dos alunos, que devem preparar Classes e Funções que serão utilizadas pelo AIV através da execução de bibliotecas.

### Detection Factory

Na Detection Factory, os alunos devem criar uma função que recebe um array de objetos (InferenceFrameData) que contém os bytes de uma imagem e os dados do frame que são utilizados pelo sistema. Essa função deve retornar um array de objetos (InferencePredictionData) que contenha os resultados das inferências e os dados do frame.

Exemplo de payload recebido:

```
[
  {
    "frame": "bytes",
    "newFrameData": { "...": "..." }
  },
  {
    "frame": "bytes",
    "newFrameData": { "...": "..." }
  }
]
```

Exemplo de payload retornado:

```
[
  {
    "predictions": [
      {
        "classId": "person",
        "trackId": "",
        "confidence": 0.98,
        "boundingBox": {
          "type": "quadrilateral",
          "coordinates": [
            {"x": 0, "y": 0},
            {"x": 0, "y": 0},
            {"x": 0, "y": 0},
            {"x": 0, "y": 0}
          ]
        }
      }
    ]
  },
  {
    "predictions": [
      {
        "classId": "person",
        "trackId": "",
        "confidence": 0.98,
        "boundingBox": {
          "type": "quadrilateral",
          "coordinates": [
            {"x": 0, "y": 0},
            {"x": 0, "y": 0},
            {"x": 0, "y": 0},
            {"x": 0, "y": 0}
          ]
        }
      }
    ]
  },
  "misc": {},
  "newFrameData": { "...": "..." }
],
{
  "predictions": [
    {
      "classId": "person",
      "trackId": "",
      "confidence": 0.98,
      "boundingBox": {
        "type": "quadrilateral",
        "coordinates": [
          {"x": 0, "y": 0},
          {"x": 0, "y": 0},
          {"x": 0, "y": 0},
          {"x": 0, "y": 0}
        ]
      }
    }
  ]
},
  "misc": {},
  "newFrameData": { "...": "..." }
}
]
```

Após criar a função, o código do aluno deve iniciar a biblioteca passando essa função e as variáveis de configuração.

## \_\_main\_\_.py

[Contents](#) [History](#) [Compare](#) [Blame](#)

```
1 import json
2 import logging
3 from .yolo import YOLOv5
4
5 from detectionfactory.configuration import cfg
6 from detectionfactory.detection_factory import DetectionFactory
7
8 logging.basicConfig()
9 logging.info("Bootstrapping YOLO inference worker")
10
11 ai_cfg = json.loads(json.loads(cfg.ai))
12
13 model = YOLOv5(model_size=ai_cfg["model_size"],
14                img_size=ai_cfg["img_size"],
15                classes=ai_cfg["classes"],
16                conf=ai_cfg["conf_thres"])
17
18 df = DetectionFactory(cfg.library, model)
19
20 logging.info("Bootstrapping done.")
21
22 df.start()
23
```

## Event Factory

Na Event Factory, os alunos devem criar uma classe que herda a classe abstrata “EventPipeline”, que recebe uma Detecção, que contém um frame com suas predições. Essa função deve retornar um array contendo os inícios e fins de eventos ou retornar vazio caso a lógica decida que aquele frame não contém nenhum evento

Exemplo de payload recebido:

```
{
  "frame": { "...": "..."},
  "...": "...",
  "predictions": [
    {
      "classId": "person",
      "trackId": "",
      "confidence": 0.98,
      "boundingBox": {
        "type": "quadrilateral",
        "coordinates": [
          {"x": 0, "y": 0},
          {"x": 0, "y": 0},
          {"x": 0, "y": 0},
          {"x": 0, "y": 0}
        ]
      }
    }
  ]
},
  "misc": {}
}
```



Exemplo de payload retornado:

```
[
  {
    "event_id": "e451ac1c-5b91-4856-b681-60409700f6d4",
    "detection": {
      "frame": {"..." : "..."},
      "...": "...",
      "predictions": [
        {
          "classId": "person",
          "trackId": "",
          "confidence": 0.98,
          "boundingBox": {
            "type": "quadrilateral",
            "coordinates": [
              {"x": 0, "y": 0},
              {"x": 0, "y": 0},
              {"x": 0, "y": 0},
              {"x": 0, "y": 0}
            ]
          }
        }
      ]
    },
    "misc": {}
  }
]
```

Após criar a classe que fará o processamento, o código do aluno deve iniciar a biblioteca passando essa classe e as variáveis de configuração.

## \_\_main\_\_.py

[Contents](#) [History](#) [Compare](#) [Blame](#)

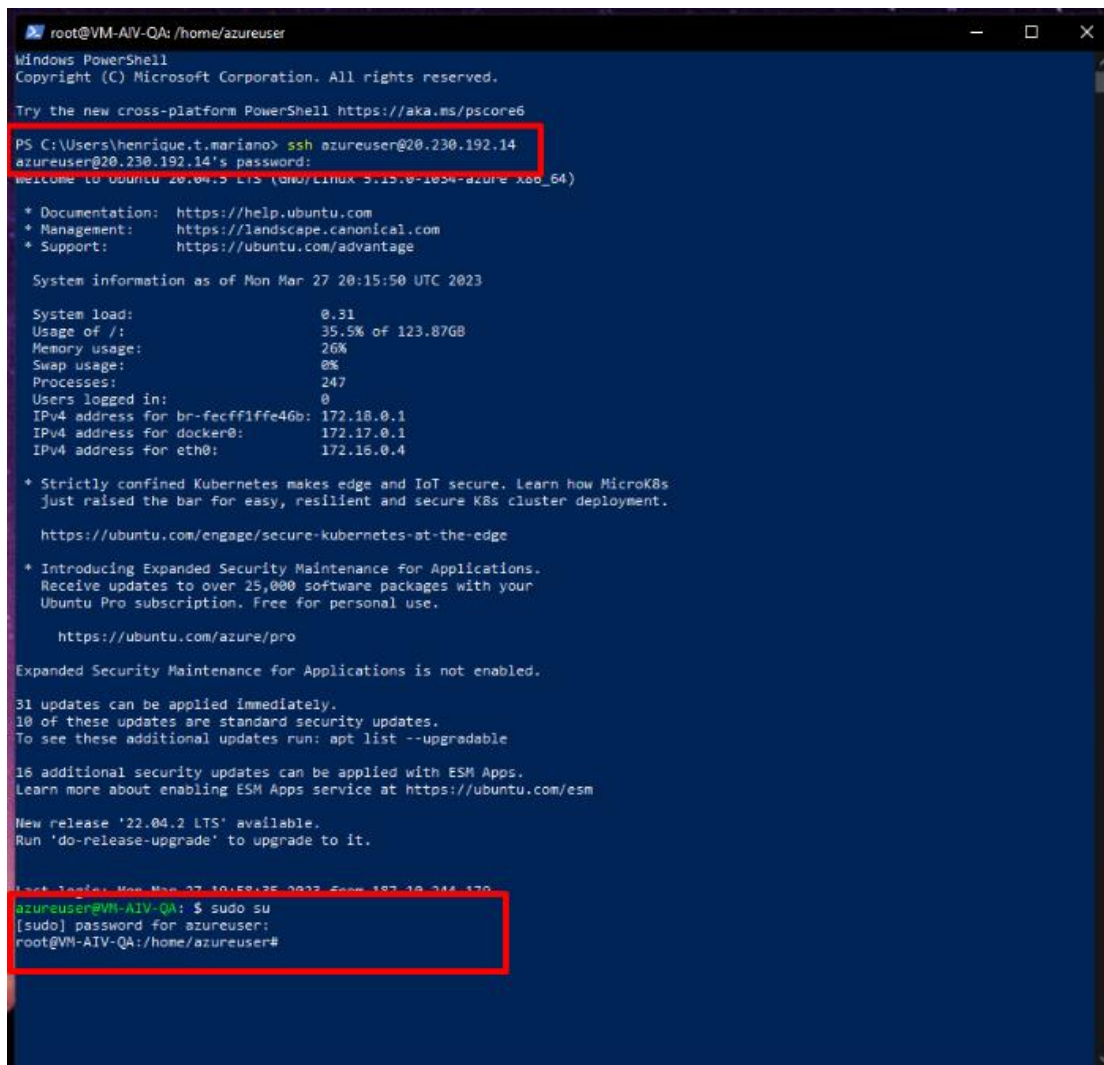
```
1 from eventfactory.configuration import cfg
2 from eventfactory import EventFactory
3 from .pipeline import Pipeline
4
5 lib_config = cfg.library
6
7 pipeline = Pipeline(lib_config)
8
9 eventFactory = EventFactory(lib_config,
10                             None,
11                             None,
12                             None,
13                             pipeline_overwrite=pipeline)
14
15 eventFactory.start()
16
17 |
```

## Passo a passo completo para os alunos

- Acesse a máquina virtual de desenvolvimento [\(vide documentação\)](#)
- [OPCIONAL] – Inclua um arquivo de vídeo para ser utilizado como stream [\(vide documentação\)](#)
- Crie uma análise no AIV [\(vide documentação\)](#)
- Execute os componentes e códigos [\(vide documentação\)](#)
- Verifique se a análise está rodando [\(vide documentação\)](#)
- Conecte a IDE [\(vide documentação\)](#)
- Modificar componentes [\(vide documentação\)](#)
- Cancelar a análise [\(vide documentação\)](#)

## Acessar a máquina virtual do grupo

- Abra o terminal de sua preferência (Neste caso utilizaremos o Windows Power Shell);
- Digite o comando para conectar via SSH com a máquina:
  - A sintaxe do comando é: `ssh <USUARIO_VM>@<IP_VM>`
  - Exemplo: `ssh azureuser@20.230.192.14`
  - OBS: Para todos os grupos o usuário será "azureuser"
- Digite a senha da máquina
  - OBS: Enquanto a senha é digitada, os caracteres pressionados não são exibidos!
- Habilite a permissão de administrador na máquina
  - Utilize o comando: `sudo su`
  - Digite a senha da máquina



```
root@VM-AIV-QA: /home/azureuser
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\henrique.t.mariano> ssh azureuser@20.230.192.14
azureuser@20.230.192.14's password:
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.15.0-1034-azure x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Mon Mar 27 20:15:50 UTC 2023

System load:          0.31
Usage of /:            35.5% of 123.87GB
Memory usage:         26%
Swap usage:           0%
Processes:            247
Users logged in:      0
IPv4 address for br-fecff1ffe46b: 172.18.0.1
IPv4 address for docker0: 172.17.0.1
IPv4 address for eth0: 172.16.0.4

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

 * Introducing Expanded Security Maintenance for Applications.
   Receive updates to over 25,000 software packages with your
   Ubuntu Pro subscription. Free for personal use.

   https://ubuntu.com/azure/pro

Expanded Security Maintenance for Applications is not enabled.

31 updates can be applied immediately.
10 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

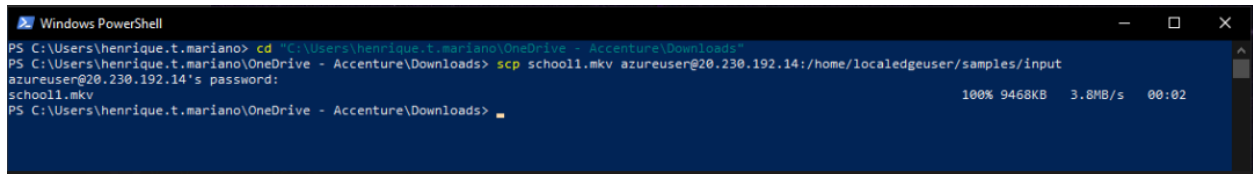
16 additional security updates can be applied with ESM Apps.
Learn more about enabling ESM Apps service at https://ubuntu.com/esm

New release '22.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

Last login: Mon Mar 27 10:59:35 2023 from 192.168.1.176
azureuser@VM-AIV-QA: $ sudo su
[sudo] password for azureuser:
root@VM-AIV-QA: /home/azureuser#
```

Incluir um vídeo novo na máquina para utilizar como stream de câmera

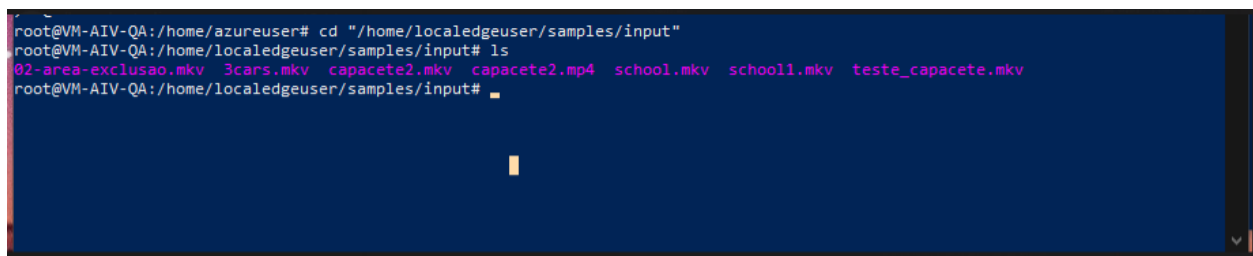
- Abra o terminal de sua preferência (Neste caso utilizaremos o Windows Power Shell)
- Vá até a pasta onde está localizado seu arquivo de vídeo
  - OBS: O formato do arquivo precisa ser .mkv
  - Para converter um vídeo para o formato .mkv utilize o site de sua preferência, por exemplo: <https://www.veed.io/convert/mp4-to-mkv>
- Dê o comando para transferir o arquivo:
  - A sintaxe do comando é: `scp <NOME_DO_ARQUIVO.MKV>`  
`<USUÁRIO_VM>@<IP_VM>:/home/localedgeuser/samples/input`
  - Exemplo: `scp school1.mkv azureuser@20.230.192.14:/home/localedgeuser/samples/input`
- Digite a senha da VM



```
Windows PowerShell
PS C:\Users\henrique.t.mariano> cd "C:\Users\henrique.t.mariano\OneDrive - Accenture\Downloads"
PS C:\Users\henrique.t.mariano\OneDrive - Accenture\Downloads> scp school1.mkv azureuser@20.230.192.14:/home/localedgeuser/samples/input
azureuser@20.230.192.14's password:
school1.mkv 100% 9468KB 3.8MB/s 00:02
PS C:\Users\henrique.t.mariano\OneDrive - Accenture\Downloads>
```

Para verificar se seu vídeo está na máquina

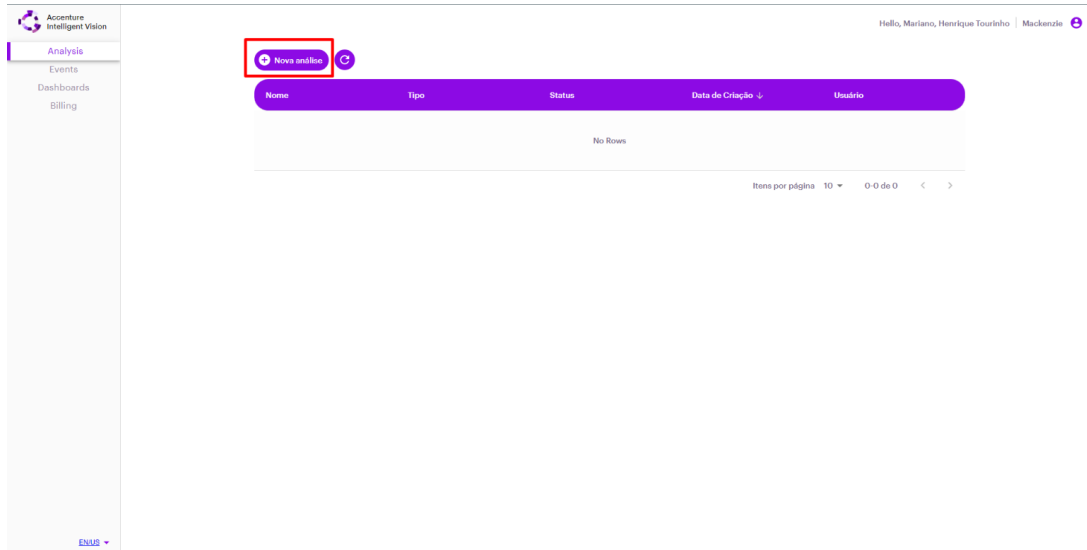
- Vá até o diretório especificado
  - Utilize o comando: `cd "/home/localedgeuser/samples/input"`
- Liste os arquivos existentes nessa pasta
  - Utilize o comando: `ls`
- Se o arquivo que você utilizou o comando (`scp ...`) estiver listado, a transferência ocorreu com sucesso



```
root@VM-AIV-QA:/home/azureuser# cd "/home/localedgeuser/samples/input"
root@VM-AIV-QA:/home/localedgeuser/samples/input# ls
02-area-exclusao.mkv  3cars.mkv  capacete2.mkv  capacete2.mp4  school.mkv  school1.mkv  teste_capacete.mkv
root@VM-AIV-QA:/home/localedgeuser/samples/input#
```

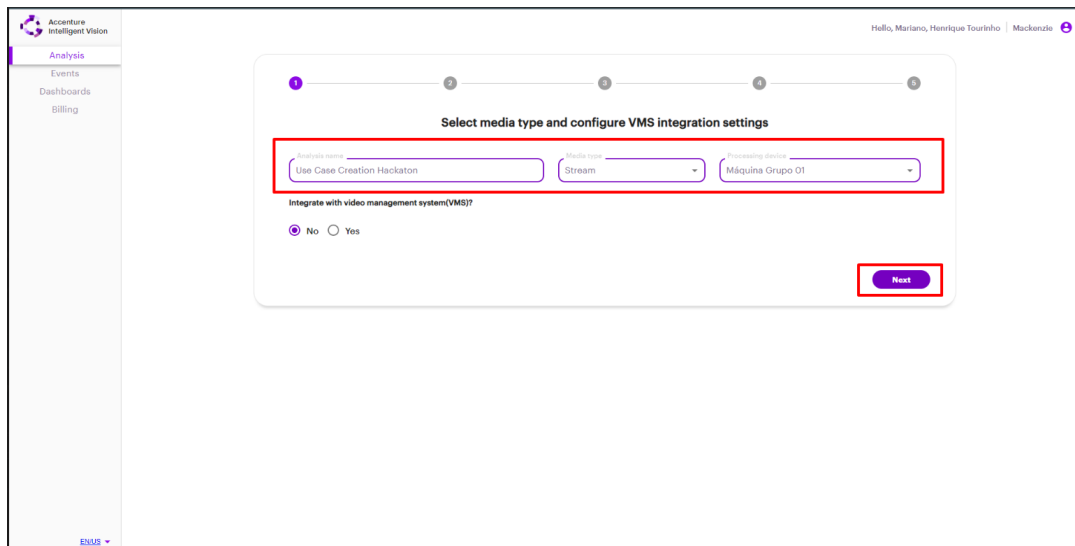
## Criar uma análise

- Acesse a plataforma (<https://aivqa.liquidstudiobr.com/>)
- Realize o login (utilize o e-mail pré-cadastrado)
- Clique no botão “Nova análise”



### Passo 1

- Digite um nome para sua análise
- Selecione a opção “Stream”
- Selecione o dispositivo de processamento do seu grupo.
- Clique em “Próximo”



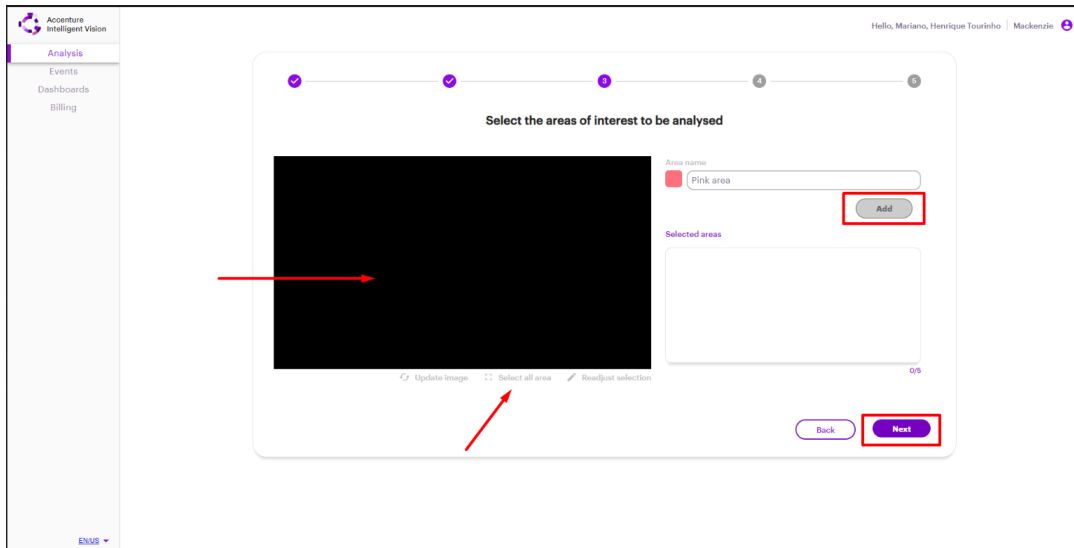
## Passo 2

- Digite o endereço do simulador de vídeo utilizando o padrão
  - A sintaxe do comando é: `rtsp://rtspsim:554/media/<NOME_DO_VIDEO>.mkv`
  - Exemplo: `rtsp://rtspsim:554/media/school.mkv`
- Clique em “Validar” e aguarde a plataforma validar a conexão com a stream
- Após validado, clique em “Próximo”



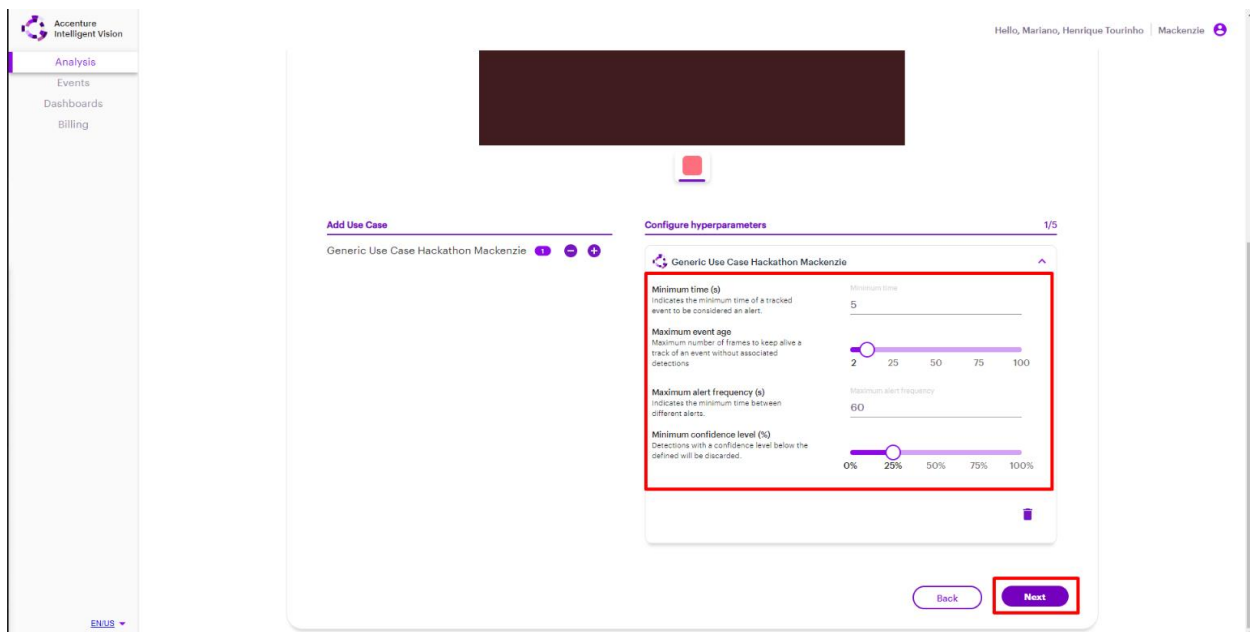
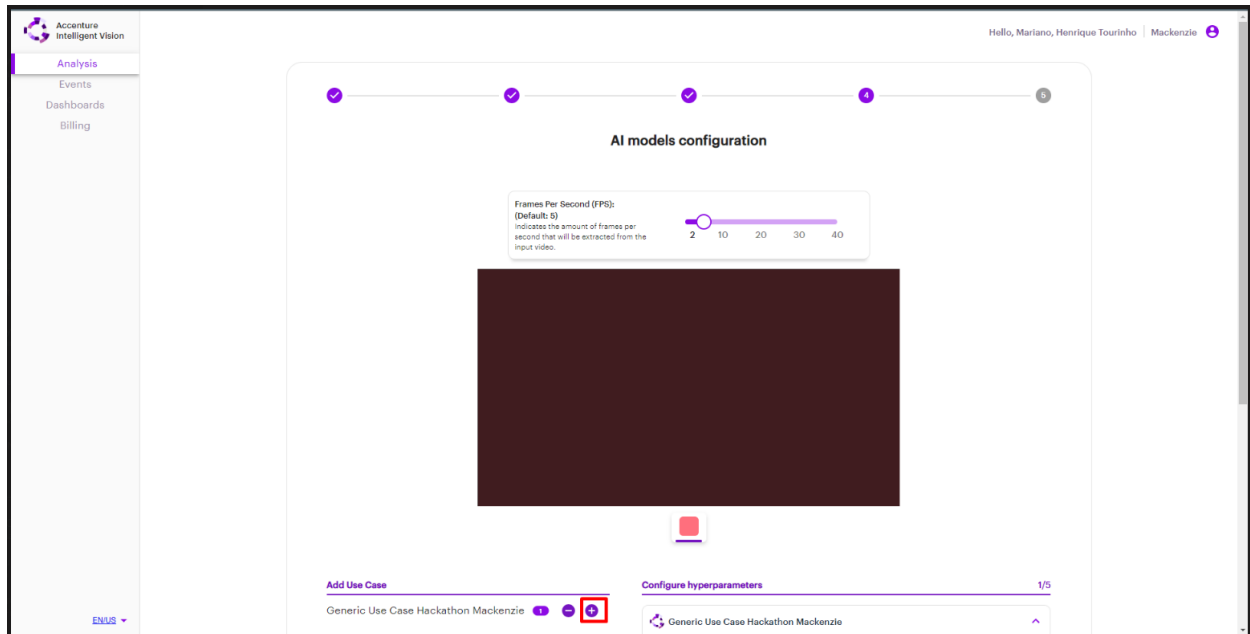
## Passo 3

- Selecione a região de interesse formando um polígono na tela de seleção ou clicando no botão “Selecionar toda área”
- Adicione a região de interesse com o botão “Adicionar”
- Clique em “Próximo”



## Passo 4

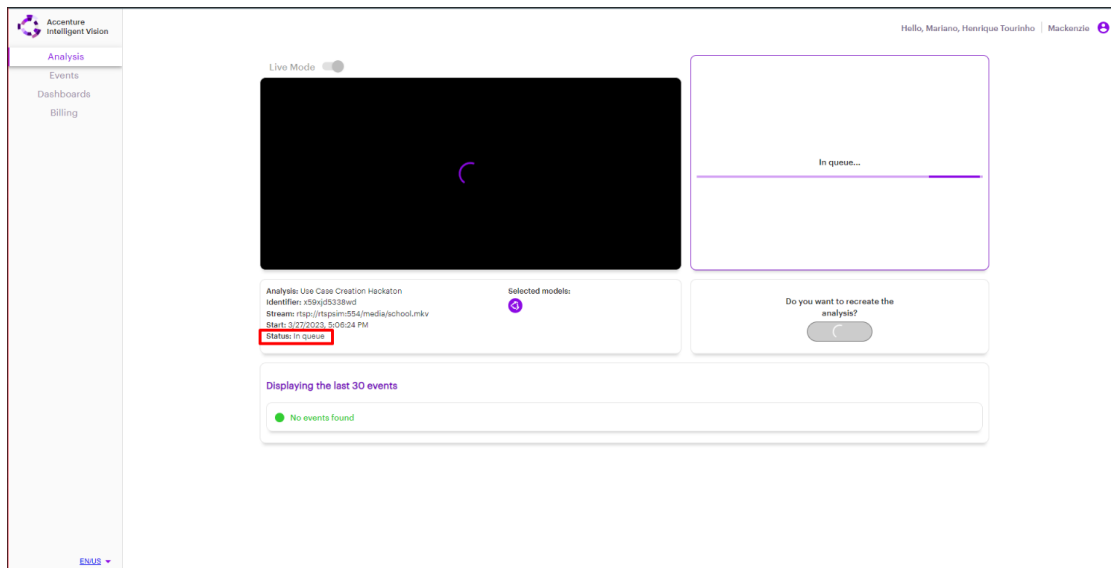
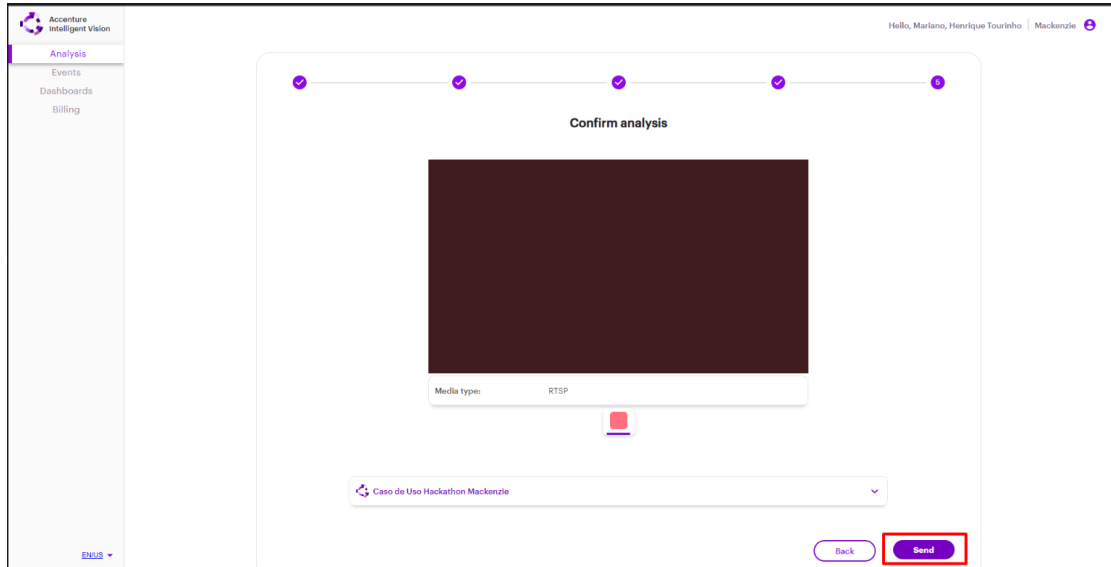
- Seleccione o caso de uso do hackaton
- Configure os parâmetros de acordo com a necessidade do use case





## Passo 5

- Clique em “Enviar”
- Certifique-se que a análise passou do status “Na fila” para “Provisionando”



Accenture  
Intelligent Vision

Analysis

Events

Dashboards

Billing

Live Mode

Analysis: Use Case Creation Hackaton  
Identifier: x59xj85338wd  
Stream: rtp://192.168.1.104/media/school.mkv  
Start: 3/22/2024, 5:06:24 PM  
**Status: Provisioning - The analysis components are being provisioned**

Selected models:

Provisioning...

Do you want to stop the analysis?

Stop

Displaying the last 30 events

No events found

ESUS

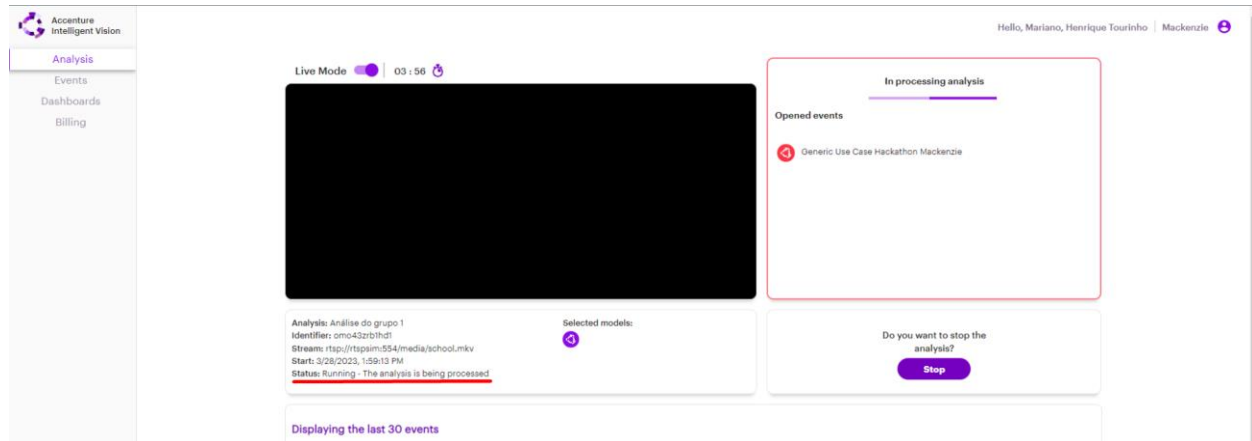
Hello, Mariano, Henrique Tourinho | Mackenzie

## Executar os componentes e códigos para iniciar a análise

- Acesse a máquina [\(vide documentação\)](#)
- Verifique se todos os componentes estão rodando
  - Utilize o comando: `docker ps`
  - OBS: Os componentes são:
    - `detection-factory-<ID_ANALISE>-<ID_CONTAINER>;`
    - `event-factory-<ID_ANALISE>-<ID_CONTAINER>;`
    - `model-control -<ID_ANALISE>-<ID_CONTAINER>;`
    - `analysis-manager -<ID_ANALISE>-<ID_CONTAINER>;`
    - `frame-factory-<ID_ANALISE>-<ID_CONTAINER>;`
- Execute o container da `detection-factory`
  - A sintaxe do comando é: `docker exec -it <CONTAINER_ID> bash`
  - Exemplo: `docker exec -it 42c14db2f850 bash`
- Dentro do container da `detection-factory`, execute o código
  - Utilize o comando: `python3 -um yolov5`
- Abra outro terminal
- Acesse a máquina [\(vide documentação\)](#)
- Execute o container da `event-factory`
  - A sintaxe do comando é: `docker exec -it <CONTAINER_ID> bash`
  - Exemplo: `docker exec -it 42c14db2f850 bash`
- Dentro do container da `event-factory`, execute o código
  - Utilize o comando: `python3 -um roi`

Verificar se a análise está rodando

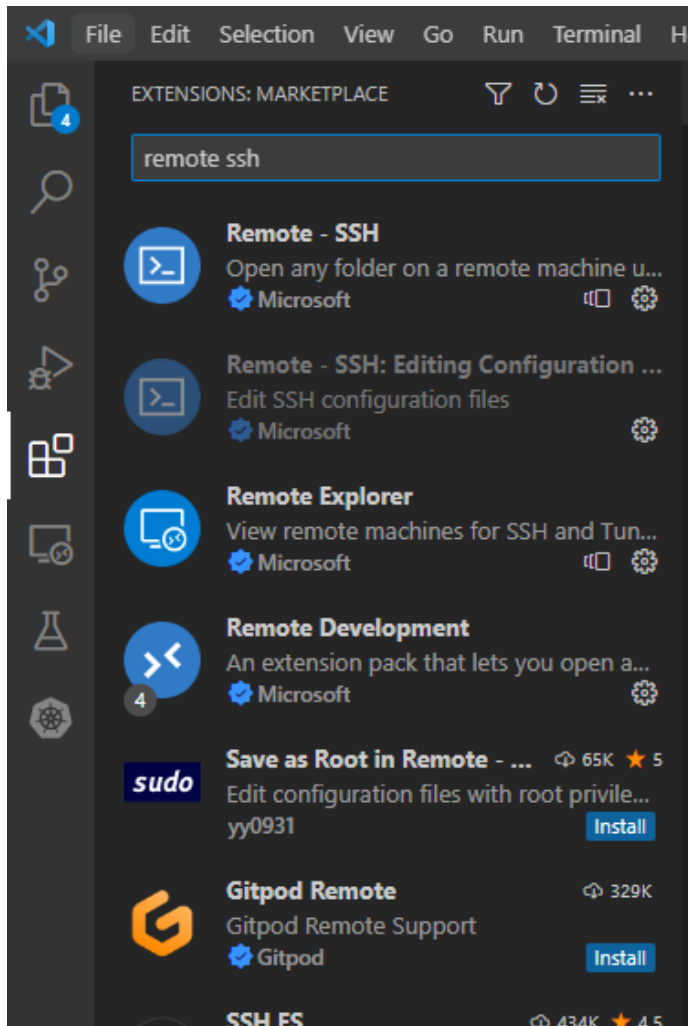
- Acesse a plataforma
- Acesse sua análise
- Verifique se o campo “status” está como “Em processamento/Running”



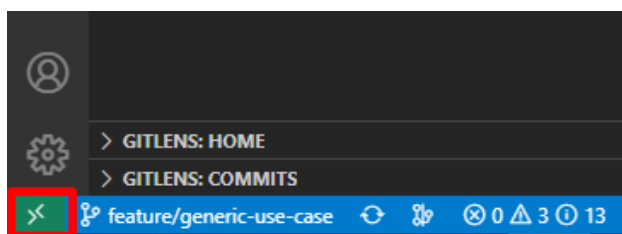
## Conectar a IDE na VM

### Visual Studio

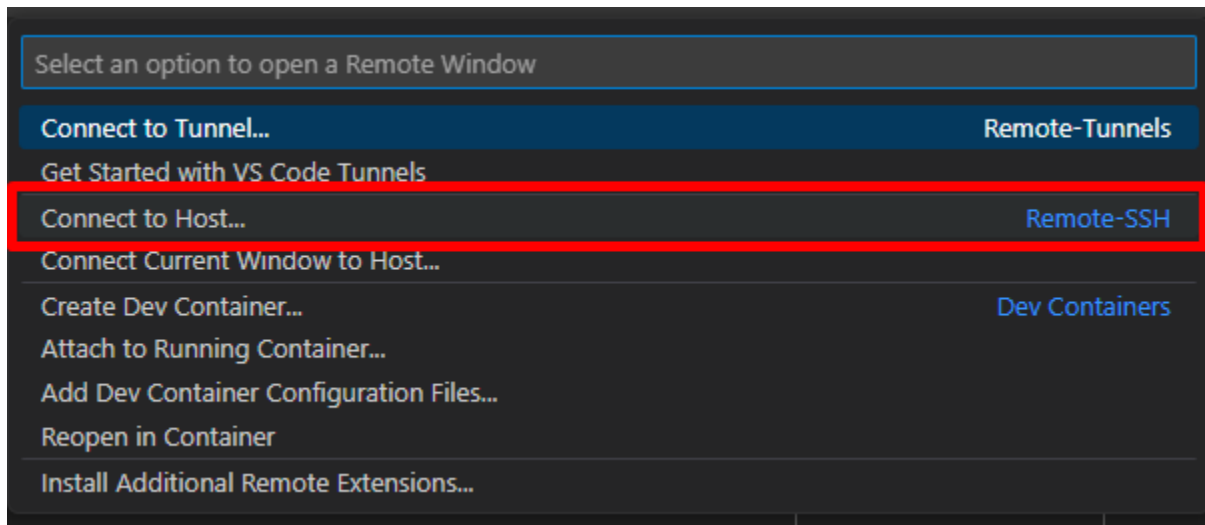
- Para editar um arquivo remotamente utilizando o VSCode, será necessário possuir uma extensão instalada, o Remote – SSH



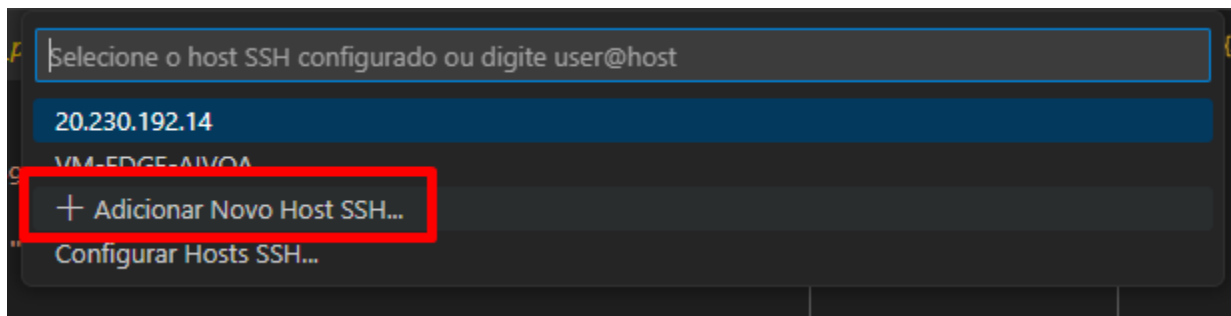
- Uma vez instalado, é possível se conectar a máquina virtual usando o botão no canto inferior esquerdo da IDE



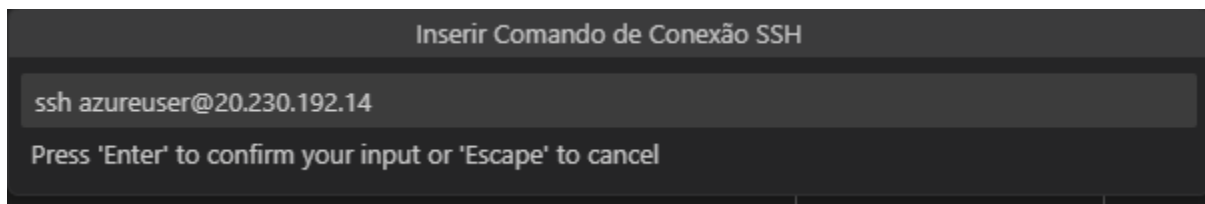
- Selecione "Connect to Host"



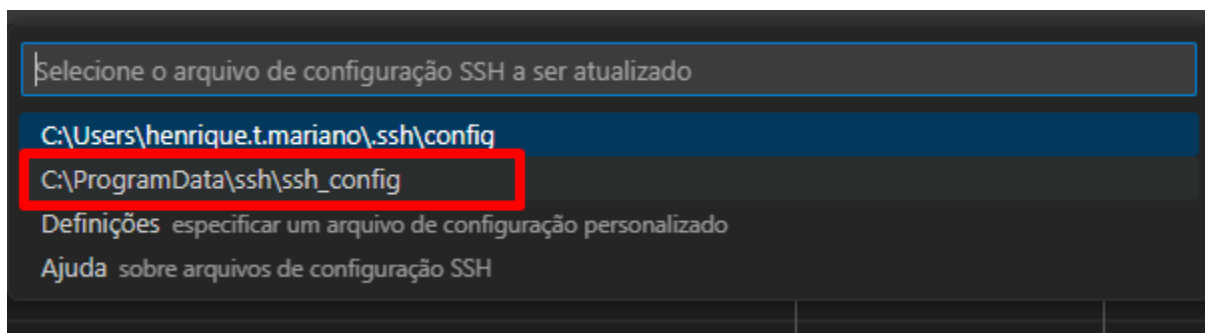
- Caso seja a primeira conexão, clique em "Adicionar Novo Host SSH..."



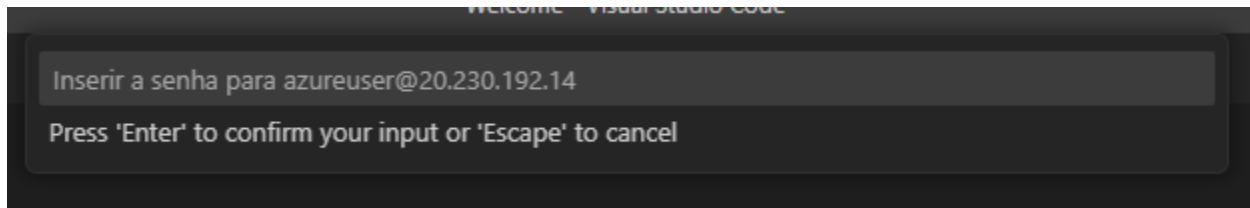
- Utilize o comando usado para acessar a máquina virtual



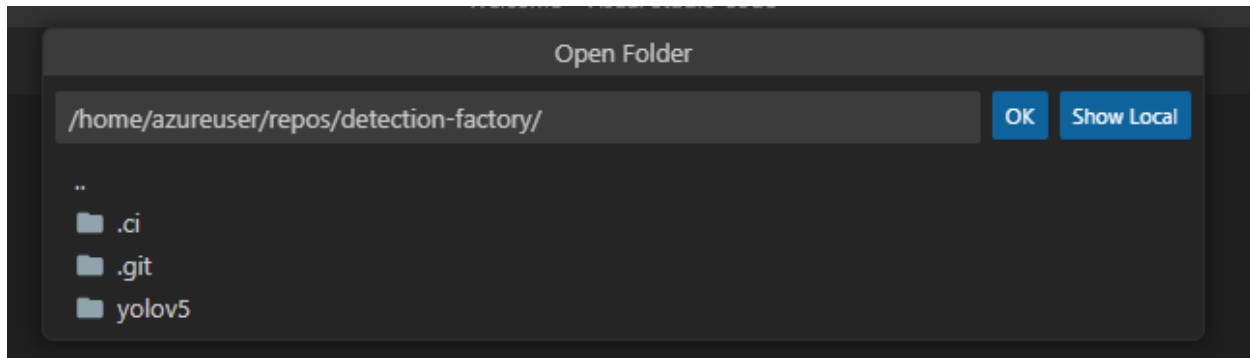
- Utilize o caminho onde as configurações de ssh ficam armazenadas em sua máquina



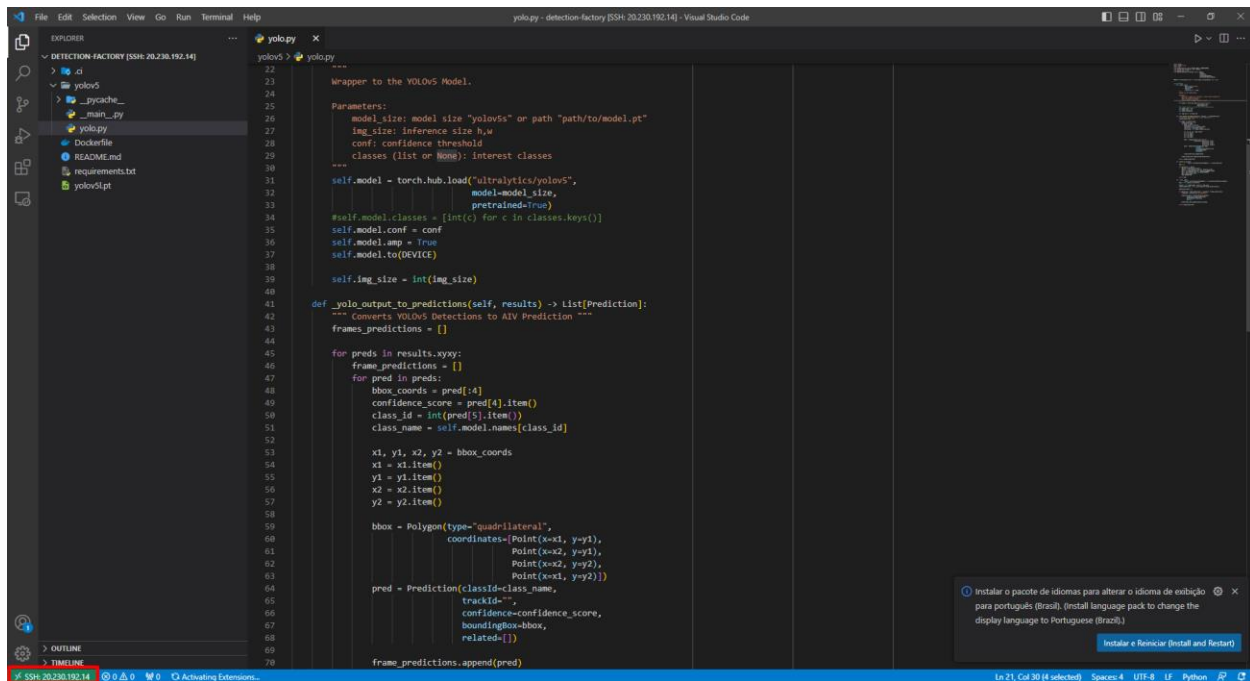
- A operação irá abrir uma nova janela do vsCode, logo de início, o programa irá pedir a senha



- Após conectado, o programa irá pedir para indicar a pasta do código.



- Uma vez selecionado, a ferramenta estará pronta para receber a atualização.



## VIM

Para editar um arquivo remotamente utilizando o vim local, basta executar o comando:

```
> vim scp://<IP_MAQUINA>/<CAMINHO_ARQUIVO>
```

Exemplo:

> vim scp://22.24.133.220//home/user/Repositories/project/code.py

```

12  area_of_interest_b64 = cfg.use_case.area_of_interest
11  area_of_interest = json.loads(base64.b64decode(area_of_interest_b64))
10  region_coords = area_of_interest["polygon"]["coordinates"]

    self._region_of_interest = RegionOfInterest(region_coords)

    params_b64 = cfg.use_case.params
    params = json.loads(base64.b64decode(params_b64).decode("utf8"))
    min_occurrences = params["minOccurrences"]
    max_outliers = params["maxOutliers"]

    self._business_logic = RoIBusinessLogic(min_occurrences, max_outliers)
    self._filter_classes = ClassFilter(params["classes"])

29  def process_detection(self, detection: Detection) -> Union[
    None,
    EventEndedSignal,
    EventStartedSignal]:

    detection = self._region_of_interest.process(detection)
    detection = self._filter_classes.process(detection)
    event = self._business_logic.process(detection)

    return event

5  from eventfactory import Detection, PipelineStep
4
3
2  class ClassFilter(PipelineStep):
1  def __init__(self, classes: dict) -> None:
6  self.classes = classes.keys()

    def process(self, detection: Detection) -> Detection:
3  predictions = filter(self._filter_classes, detection["predictions"])
4
5  detection["predictions"] = list(predictions)
6
7  return detection
8
9  def _filter_classes(self, prediction) -> None:
10 return True if prediction["classId"] in self.classes else False

1  lucas@CPX-CVTPUUSK35P

```



## Modificar componentes

- Acessar a máquina ([vide documentação](#))
- Conectar a IDE na VM ([vide documentação](#))

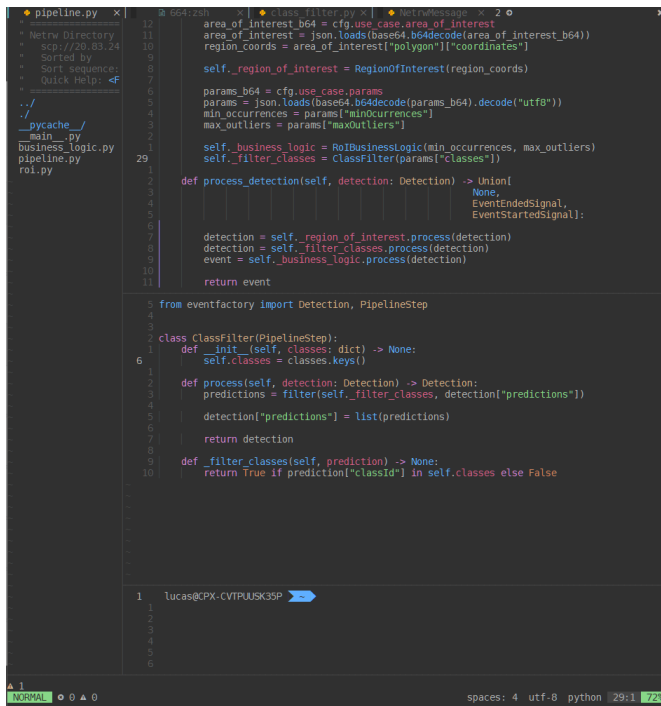
## Detection factory

- Utilizar o caminho absoluto: /home/azureuser/repos/detection-factory

## Event factory

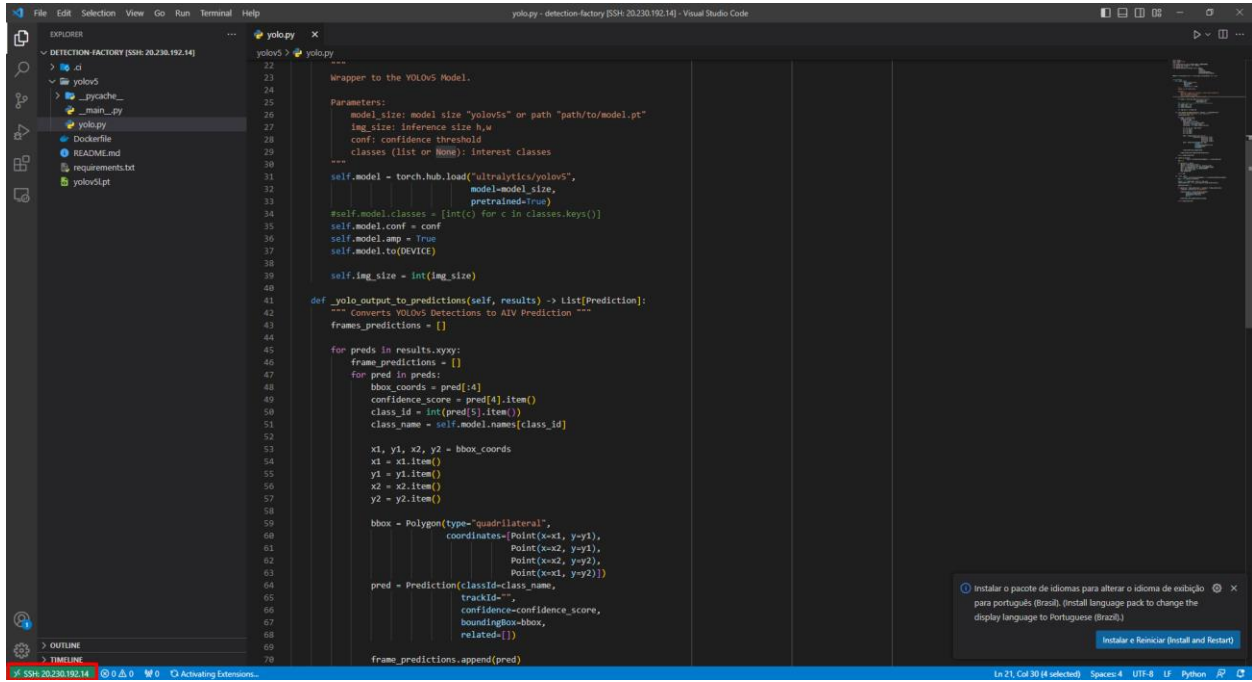
- Utilizar o caminho absoluto: /home/azureuser/repos/event-factory

## Exemplo VIM:



```
12 area_of_interest_b64 = cfg.use_case.area_of_interest
13 area_of_interest = json.loads(base64.b64decode(area_of_interest_b64))
14 region_coords = area_of_interest["polygon"]["coordinates"]
15
16 self._region_of_interest = RegionOfInterest(region_coords)
17
18 params_b64 = cfg.use_case.params
19 params = json.loads(base64.b64decode(params_b64).decode("utf8"))
20 min_occurrences = params["minOccurrences"]
21 max_outliers = params["maxOutliers"]
22
23 self._business_logic = RaBusinessLogic(min_occurrences, max_outliers)
24 self._filter_classes = ClassFilter(params["classes"])
25
26 def process_detection(self, detection: Detection) -> Union[
27     None,
28     EventEndedSignal,
29     EventStartedSignal]:
30
31     detection = self._region_of_interest.process(detection)
32     detection = self._filter_classes.process(detection)
33     event = self._business_logic.process(detection)
34
35     return event
36
37 from eventfactory import Detection, PipelineStep
38
39 class ClassFilter(PipelineStep):
40     def __init__(self, classes: dict) -> None:
41         self.classes = classes.keys()
42
43     def process(self, detection: Detection) -> Detection:
44         predictions = filter(self._filter_classes, detection["predictions"])
45         detection["predictions"] = list(predictions)
46         return detection
47
48     def _filter_classes(self, prediction) -> None:
49         return True if prediction["classId"] in self.classes else False
50
51 1 Lucas@CPX-CVTPUUSK35P
```

## Exemplo VS Code:



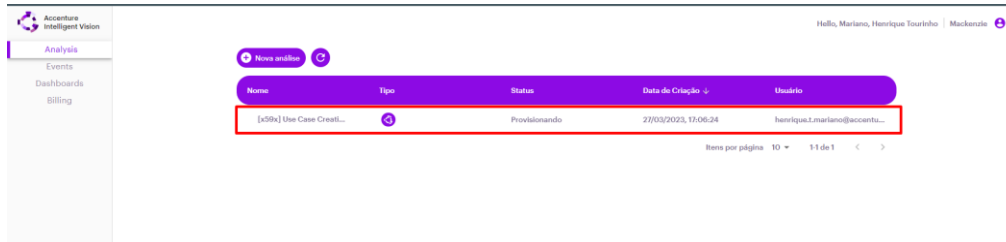
```

22  ---
23  Wrapper to the YOLOv5 Model.
24
25  Parameters:
26  model_size: model size "yolov5s" or path "path/to/model.pt"
27  img_size: inference size h,w
28  conf: confidence threshold
29  classes (list or None): interest classes
30  ---
31  self.model = torch.hub.load("ultralytics/yolov5",
32                               model=model_size,
33                               pretrained=True)
34  self.model.classes = [int(c) for c in classes.keys()]
35  self.model.conf = conf
36  self.model.amp = True
37  self.model.to(DEVICE)
38
39  self.img_size = int(img_size)
40
41  def _yolo_output_to_predictions(self, results) -> List[Prediction]:
42  """ Converts YOLOv5 Detections to AIV Prediction """
43  frames_predictions = []
44
45  for preds in results.xyxy:
46  frame_predictions = []
47  for pred in preds:
48  bbox_coors = pred[:4]
49  confidence_score = pred[4].item()
50  class_id = int(pred[5].item())
51  class_name = self.model.names[class_id]
52
53  x1, y1, x2, y2 = bbox_coors
54  x1 = x1.item()
55  y1 = y1.item()
56  x2 = x2.item()
57  y2 = y2.item()
58
59  bbox = Polygon(type="quadrilateral",
60                coordinates=(Point(x=x1, y=y1),
61                             Point(x=x2, y=y1),
62                             Point(x=x2, y=y2),
63                             Point(x=x1, y=y2)))
64  pred = Prediction(classId=class_name,
65                    trackId="",
66                    confidence=confidence_score,
67                    boundingBox=bbox,
68                    related=[])
69
70  frame_predictions.append(pred)

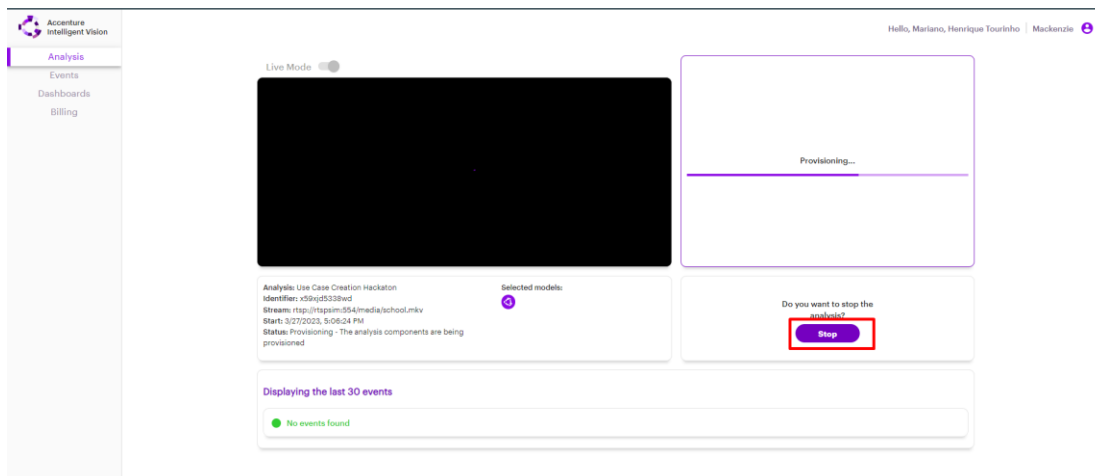
```

## Cancelar análise

- Acesse a plataforma (<https://aivqa.liquidstudiobr.com/>)
- Realize o login (utilize o e-mail pré-cadastrado)
- Selecione a análise desejada na listagem de análises.



- Clique em “encerrar”



- Logo em seguida, o sistema irá mostrar o status de cancelado

