

Precisamos modificar as urls RTSP, as configurações da câmera e o formato do json que o sauron envia pro MongoDB.

## URL RTSP

No arquivo **sauron.py**, logo no começo da main tem a variável **urls**:

```
def main():
    ##### INPUTS
    #####
    # Load configurations from external file
    idxname, _, ip, user, pwd, pcname, includebox, line_thickness,
min_score_thresh = loadconfig(
    'config.ini')
    vapix = VapixRequests(ip, user, pwd)

    ##### ADDED #####
    urls =
['rtsp://altave:altave@172.19.170.112/axis-media/media.amp?fps=10&video
codec=h264&audio=0&text=1',

'rtsp://altave:altave@172.19.170.122/axis-media/media.amp?fps=10&videoc
odec=h264&audio=0&text=1']
```

É só adicionar à **urls** as urls que você quer.

## Config Cameras

No arquivo **sauron.py**, basta buscar pela função **setup\_camera**

```
def setup_camera(cam_num):
    if cam_num == 1:
        # setup of Camera 1
        cam_name          = 'Camera1'
        cam_ip            = '172.19.170.113'
        cam_user          = 'altave'
        cam_passwd        = 'altave'
        cam_altitude      = 50
        cam_installation_LLA = [-23.156128, -45.791352, 784]
        cam_LLA_ref       = [-23.1679369, -45.831635, 500]
```

```

        cam_PTZ_ref          = [-20, 0.0, 0.0]
        cam      = ch.Camera(
                                cam_name, cam_ip, cam_user, cam_passwd,
cam_altitude,
                                cam_instalation_LLA, cam_LLA_ref, cam_PTZ_ref
                                )

    elif cam_num == 2:
        pass

    elif cam_num == 3:
        pass

    return cam

```

que se encontra logo acima da main. Você tem que adicionar os cam\_num até que tenha um cam\_num para cada câmera. E as configurações devem ser colocadas na mesma ordem das urls RTSP.

## Modificação SJSON

Vamos precisar deixar o json no seguinte formato (conversado com o Gui Martins):

```

sjson = {
    'idx': idx, # inutil
    'idxname': idxname, # inutil
    'sensor': sensor, # inutil
    'coordinates': {
        "latitude": -23.151996577530838,
        "longitude": -45.788515727619185}, # inutil
    'alerts': mapsauron2alerts(sauronobjs),
    'lla': lla_dict, # inutil
    'timestamp': timestamp,
    'seen': False,
    'datetime': {"$date":
datetime.now(pytz.timezone("Brazil/East")).isoformat()},
    'sauron': sauronobjs, # inutil
    'imageUrl': url,
    'firebase_img_url': "replace_web_here"
    'thumb_up': False,
    'thumb_down': False
}

```

No sauron.py, a função que cria o json é chamada com os seguintes argumentos

```
sjson = sauronlib.gensauronjson(ptz, lla, curr_time, imgname, tower,  
boxes_filtered, imgpath)
```

Teremos que modificar o **sauronlib.py** para gerar um json no formato desejado, usando esses argumentos.