

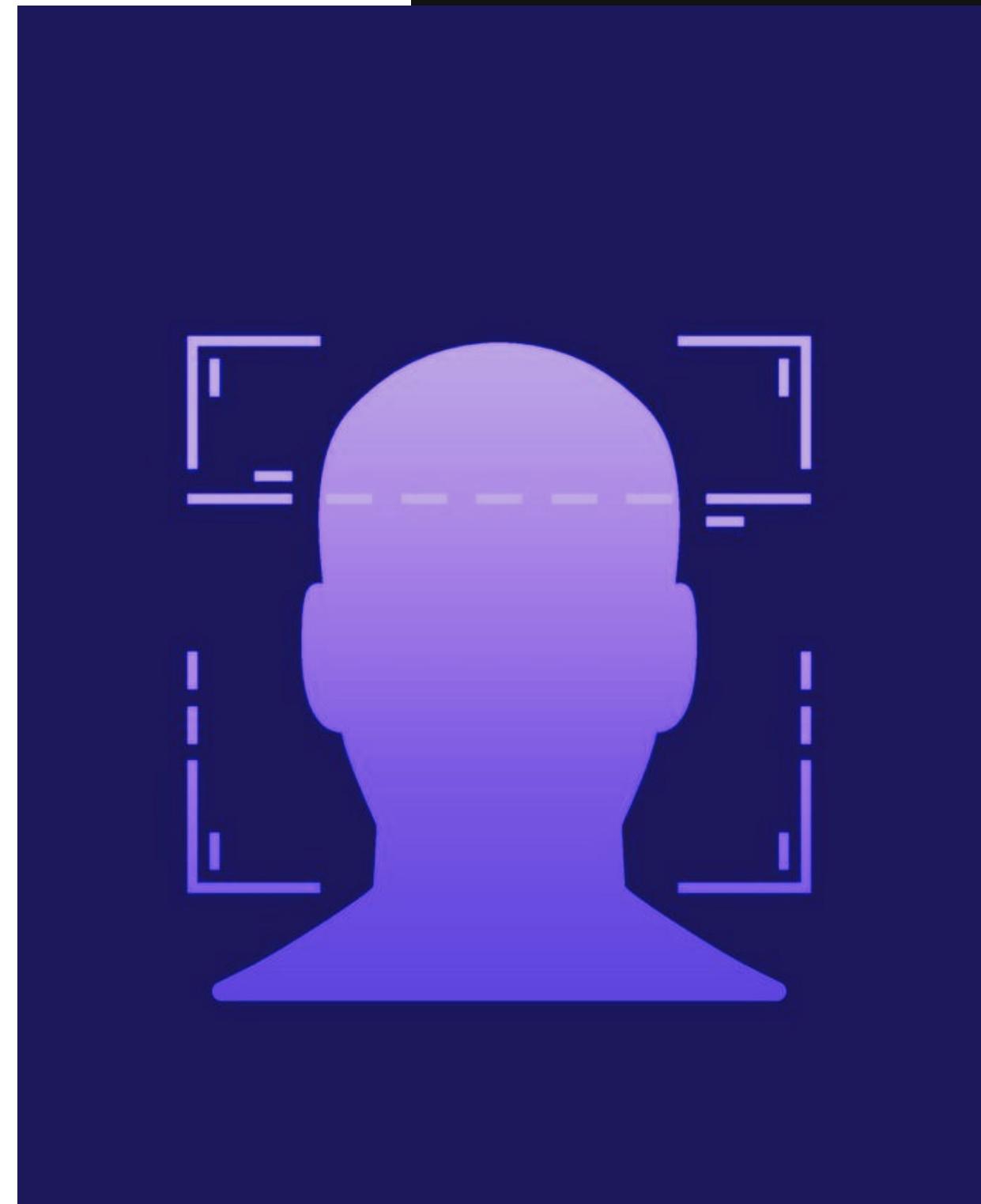
ZODAK

Sistema de reconhecimento facial

Benjamin Walter Mauss

Kéllen da Rocha Kern

Orientador: Rafael Dittgen

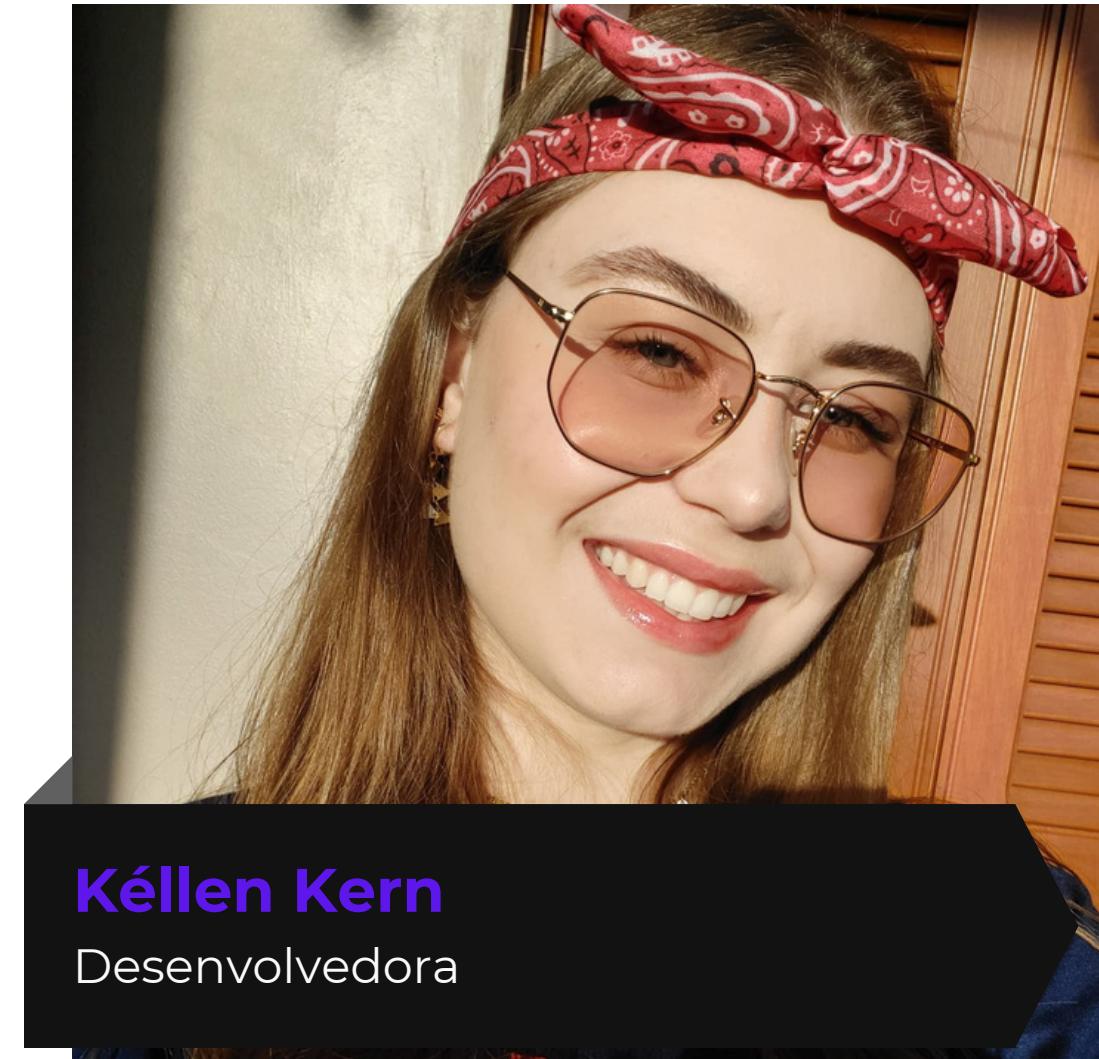


INTEGRANTES



Benjamin Mauss

Desenvolvedor



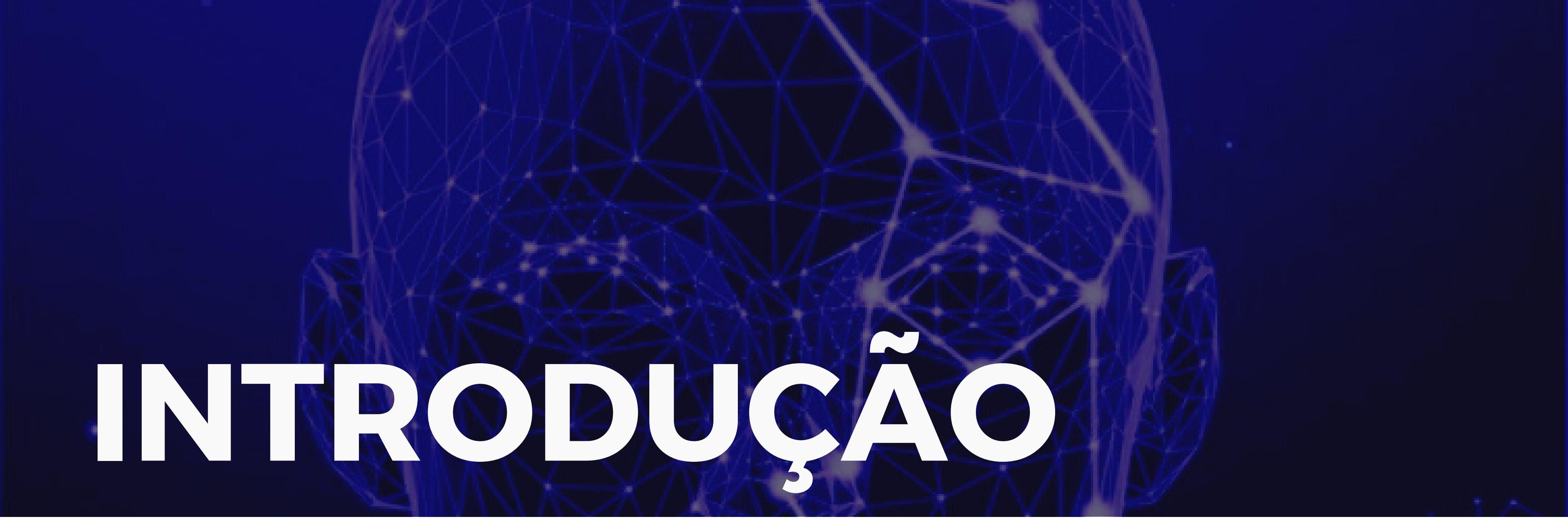
Kéllen Kern

Desenvolvedora

AGENDA:

- 04. Introdução**
- 05. Objetivos**
- 06. Justificativa**
- 08. Trabalhos Relacionados**
- 09. Tecnologias utilizadas**
- 10. Diagrama de Caso de Uso**
- 11. Diagrama de Banco de Dados**
- 12. Funcionalidades Implementadas**
- 13. Resultado**
- 23. Conclusão**
- 24. Perguntas**

```
31     def __init__(self, path=None):
32         self.file = None
33         self.fingerprints = set()
34         self.logdups = True
35         self.debug = debug
36         self.logger = logging.getLogger(__name__)
37         if path:
38             self.file = open(os.path.join(path, 'fingerprints.txt'), 'a')
39             self.file.seek(0)
40             self.fingerprints.update(self.file.read().splitlines())
41
42     @classmethod
43     def from_settings(cls, settings):
44         debug = settings.getbool('SUPERVISOR_DEBUG')
45         return cls(job_dir(settings), debug)
46
47     def request_seen(self, request):
48         fp = self.request_fingerprint(request)
49         if fp in self.fingerprints:
50             return True
51         self.fingerprints.add(fp)
52         if self.file:
53             self.file.write(fp + os.linesep)
54
55     def request_fingerprint(self, request):
56         return request_fingerprint(request)
```



INTRODUÇÃO

- IA ganhando mais espaço a cada dia
- Reconhecimento de Imagens através de visão computacional

Automatizar o processo de registro de presença

através de reconhecimento facial.

POR QUÊ?

- Sem falsificações;
- Sem interação entre aluno e a aplicação;
- Ganho de tempo:



CADA CHAMADA LEVA EM TORNO DE
7 minutos

Por Dia	35 minutos
Por semana	196 minutos

560
vezes por semestre
3920 minutos
65.33 horas

curso técnicos tem 1120 horas por semestre

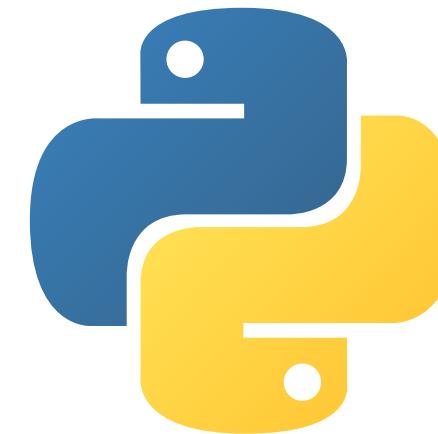
Fonte: Pesquisa de campo realizada no Instituto Federal de Sapiranga (IFSul)

TRABALHOS SIMILARES

Nome	Imagens banco de dados	Conexão	Público Alvo	Alteração Manual	Diferenciais
ChurchiX		Câmera IP	Hotéis/Igrejas		Projetos Personalizados
Fareclock		Dispositivo móvel	Empresas		Calculo de Folha de pagamento
Zodak		Câmera IP	Escolas		Cadastro de horários Cadastro de Ambientes/Salas Fácil Integração SUAP

TECNOLOGIAS UTILIZADAS

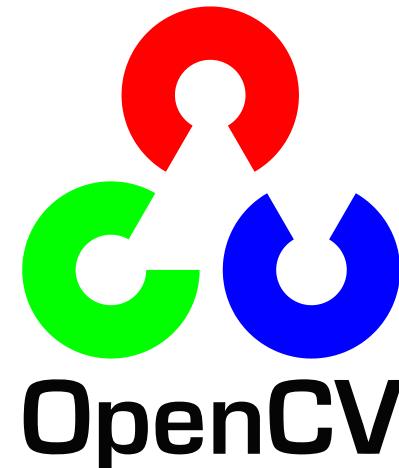
Linguagens



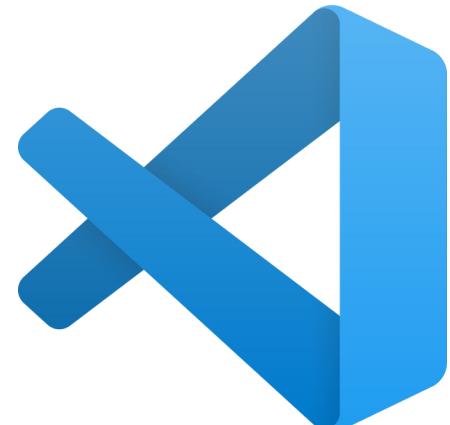
Banco de Dados



Biblioteca



IDE



Hardware



x2

DIAGRAMA DE BANCO DE DADOS

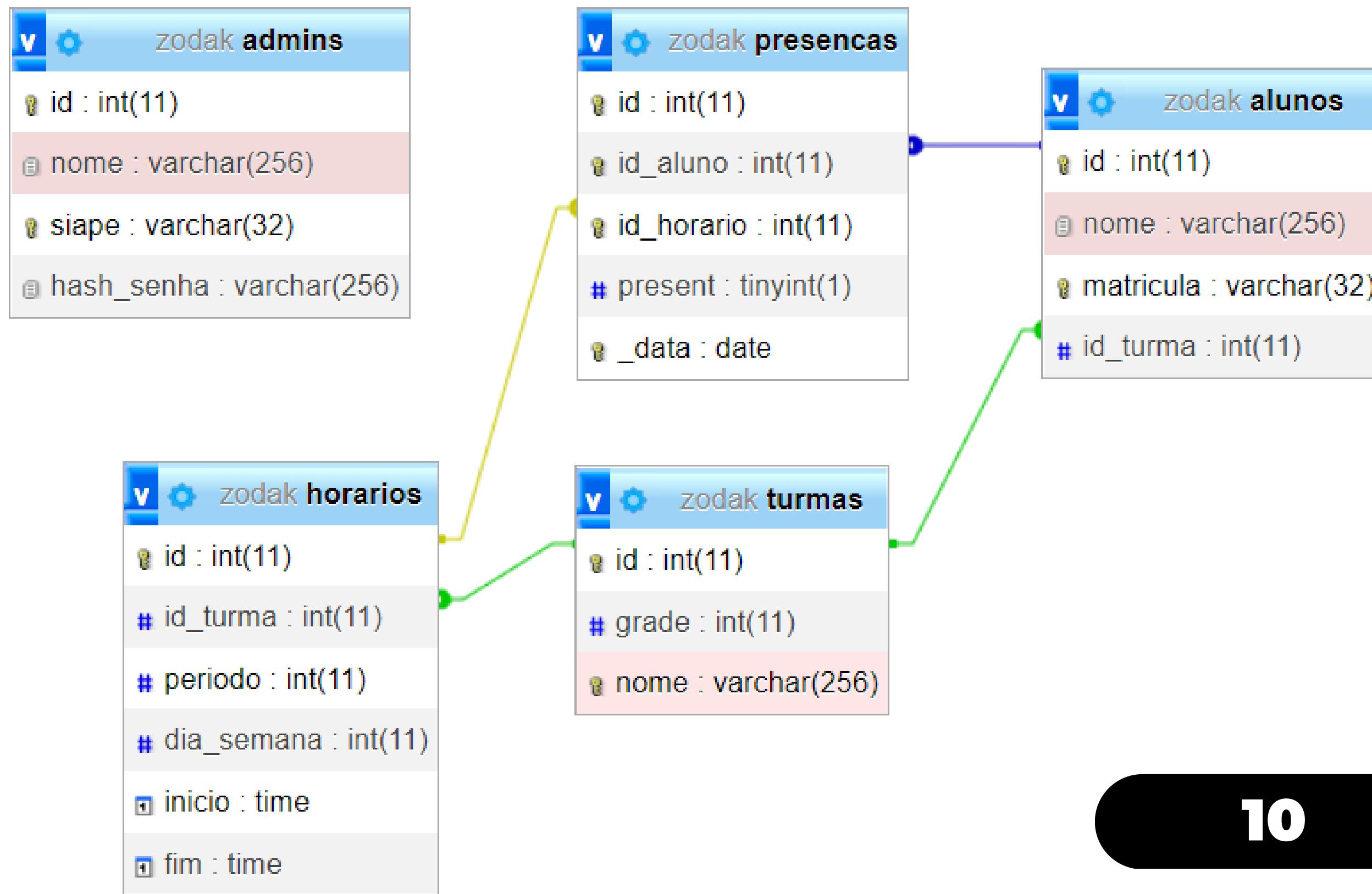
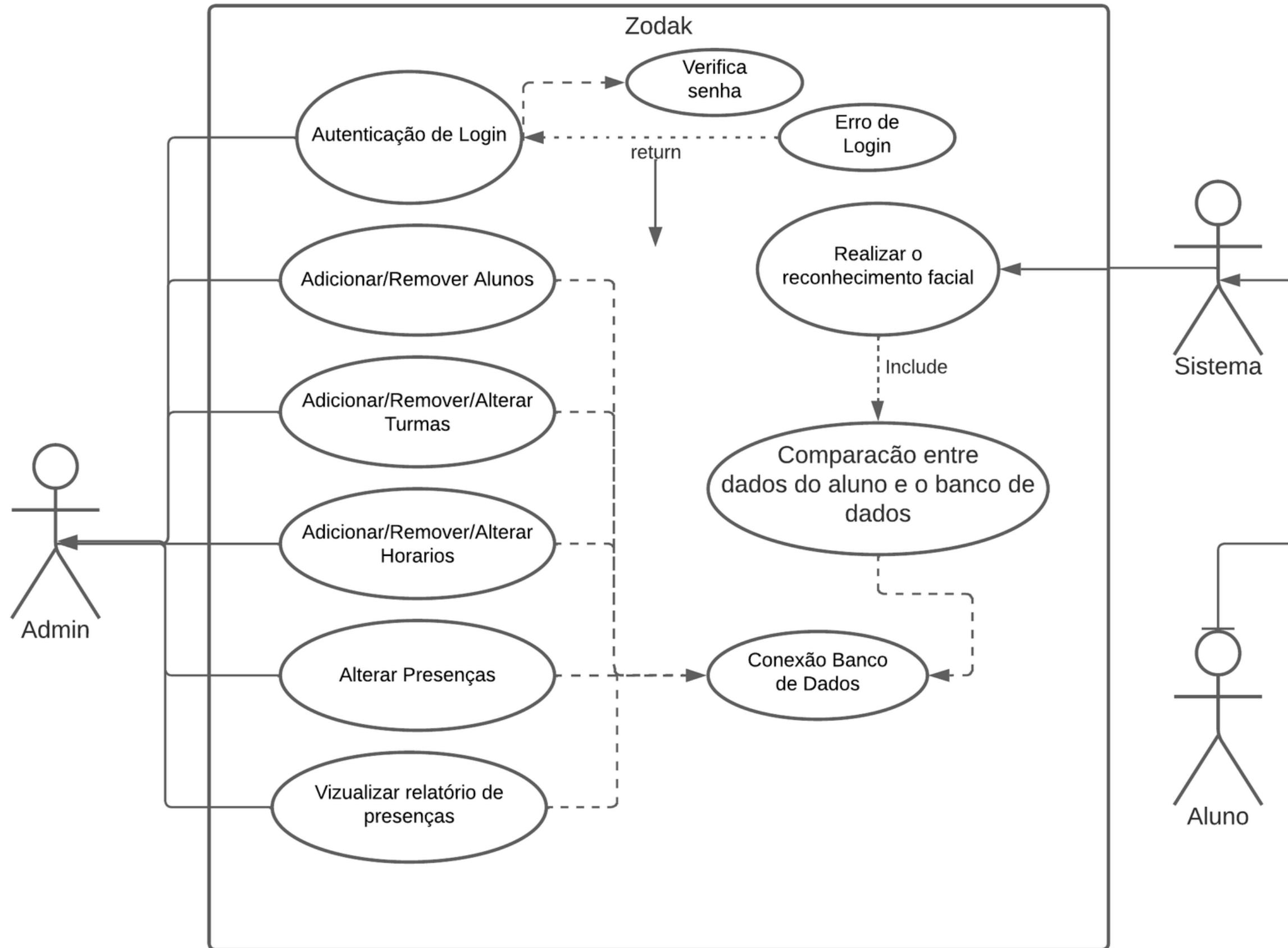


DIAGRAMA DE CASO DE USO



```
12  
13 if (! $_POST || !$POST["login"] || !$POST["senha"]){
14     header("Location: /v1/login?r=Acesso_direto");
15     die();
16 }
```

FUNCIONALIDADES PLANEJADAS:

-  Acesso de administrador (Professores)
-  Cadastro de alunos e seus dados faciais
-  Cadastro/exclusão/alteração de turmas
-  Cronograma de horários
-  Visualização do relatório das presenças
-  Alteração no registro da presença
-  Reconhecimento Facial

ACESSO ADMINISTRADOR

ZODAK

IFsul Campus Sapiranga

Username

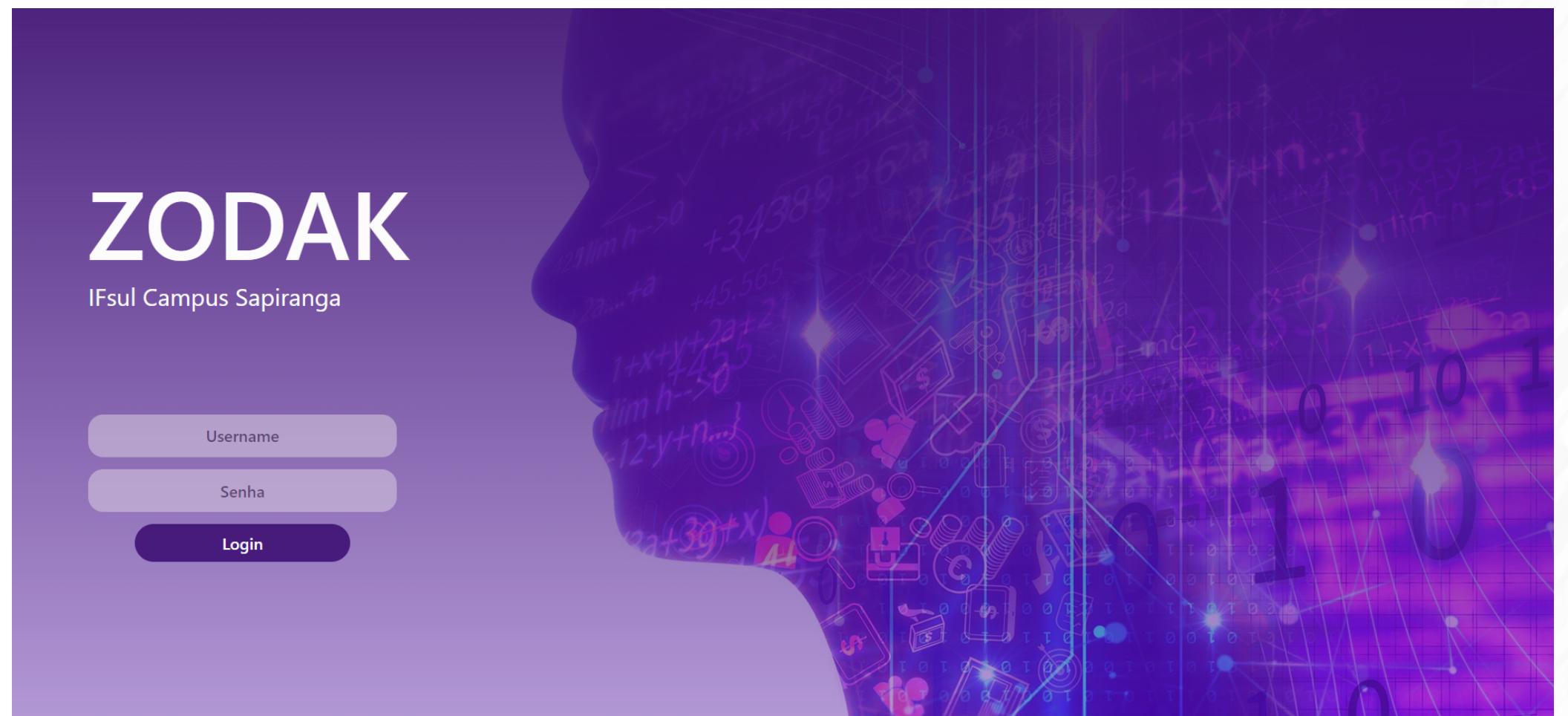
Senha

Login

13

ACESSO ADMINISTRADOR

```
18 $hash = md5($_POST['senha']);
19 $mat = strtolower($_POST["login"]);
20
21 $q = mysqli_query($conn, "SELECT * FROM zodak admins WHERE siape='$_POST["login"]' and hash_senha='$hash');");
22 if ($q && (mysqli_num_rows($q) == 1)){
23     $r=mysqli_fetch_assoc($q);
24     $_SESSION['id'] = $r['id'];
25     $_SESSION['siape'] = $r['siape'];
26     $_SESSION['nome'] = $r['nome'];
27     header("Location: /v1?r=senha_correta");
28     die();
29 }else{
30     header("Location: /v1?r=senha_errada");
31     die();
32 }
33
34     1 <?php
35
36     2
37     3 session_start();
38     4 session_destroy();
39
40     5
41     6 header("Location: /v1/login");
42
43     7
```



CADASTRO DE ALUNO

ZODAK

Alunos

Turmas

Horários

Presenças

SAIR

ADICIONAR UM NOVO ALUNO

Nome

Matricula

20191infm

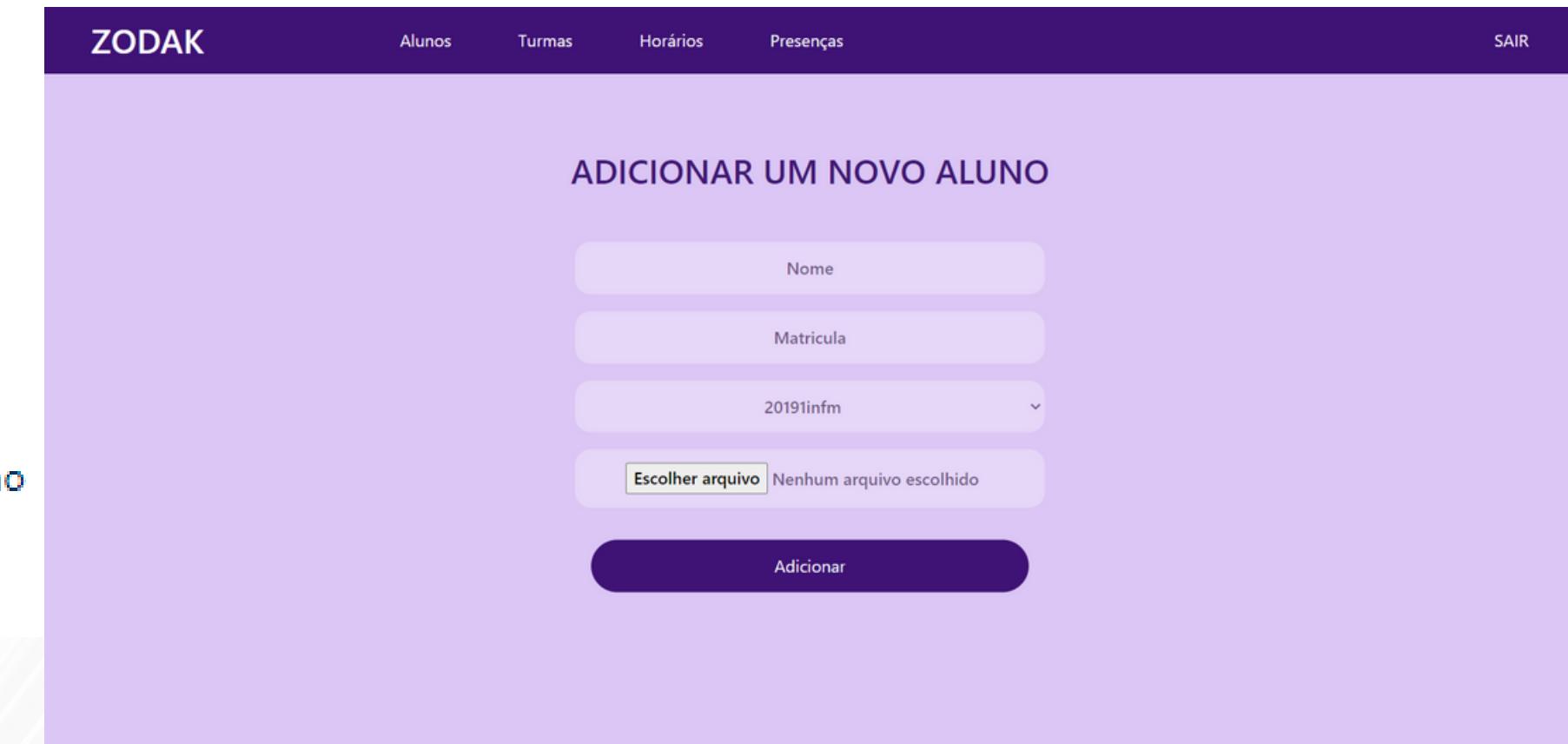
Escolher arquivo

Nenhum arquivo escolhido

Adicionar

CADASTRO DE ALUNO

```
147     $output = shell_exec('export LD_LIBRARY_PATH=""; python3 /opt/lampp/htdocs/v1/pyzin/extract_faceid.py ' . $id . ' 2>&1');
148     if(trim($output)== 1){
149         echo("looks good!\n");
150         // agora adicionar o usuario no banco :
151         $sql = "INSERT INTO zodak.alunos VALUES ('$id', '{$_POST["nome"]}', '{$_POST["matricula"]}', '{$_POST["turma"]}');";
152
153         $q = mysqli_query($conn, $sql);
154         if($q){
155             echo("alright, everything went fine!!!\n");
156         }else{
157             echo("se pa que deu errado!!!\n");
158         }
159     }else{
160         echo("deu erro no reconhecimento facial, então o aluno
161         unlink($target_file);
162     }
```



CRONOGRAMA DE HORÁRIOS

ZODAK

Alunos

Turmas

Horários

Presenças

SAIR

HORÁRIOS

id	turma	periodo	dia_semana	inicio	fim
1	20191infm ▾	1 ▾	Terça-Feira ▾	07:30:00	08:15:00
2	20191infm ▾	3 ▾	Sexta-Feira ▾	00:01:00	01:20:00
3	20191infm ▾	4 ▾	Sexta-Feira ▾	01:21:00	08:15:00

Atualizar horários

RELATÓRIO DE PRESENÇA

ZODAK

Alunos

Turmas

Horários

Presenças

SAIR

Nome:

Turma:

Since: dd/mm/aaaa

Until: dd/mm/aaaa

PESQUISAR

PRESENÇAS

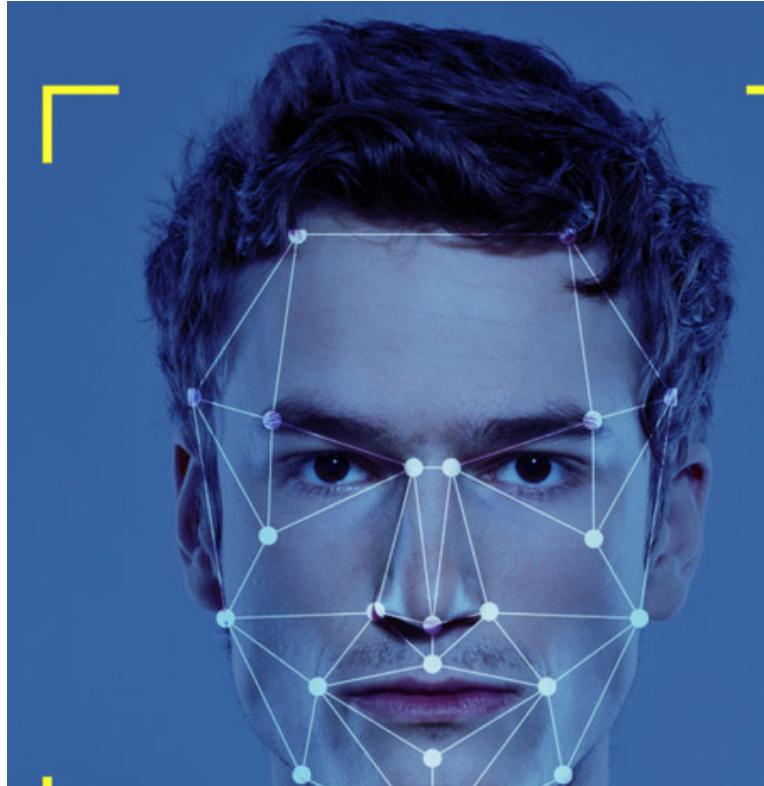
ID presença	Aluno	Matrícula	Turma	Período	Data	Início	Fim	Presente
92	Vitor	2021INF1658	20191infm	3	2022-07-29	00:01:00	01:20:00	<input checked="" type="checkbox"/>
93	Kellen	2022ETM2587	20191infm	3	2022-07-29	00:01:00	01:20:00	<input type="checkbox"/>
99	Vitor	2021INF1658	20191infm	4	2022-07-29	01:21:00	08:15:00	<input checked="" type="checkbox"/>
100	Kellen	2022ETM2587	20191infm	4	2022-07-29	01:21:00	08:15:00	<input type="checkbox"/>

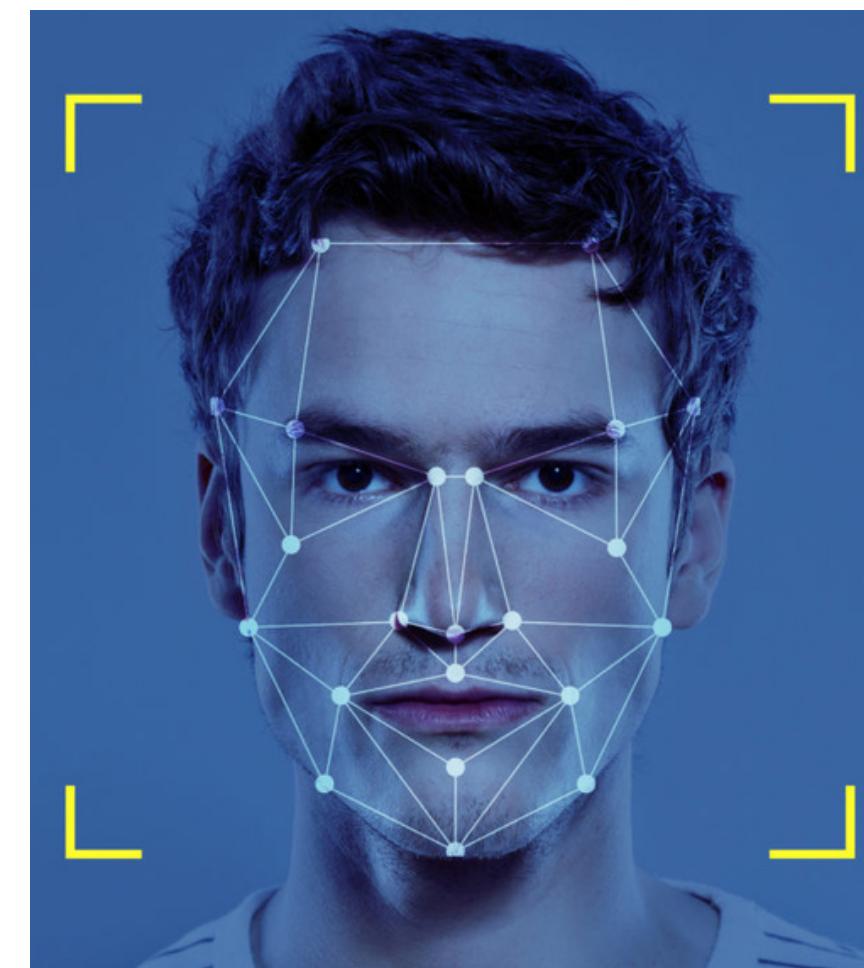
```
if (!$qq) {  
    echo 'Could not run query: ';  
    exit;  
}  
$row = mysqli_fetch_assoc($qq);  
while($row){  
    $id= $row["id"];  
    $turma = $row["turma"];  
    $id_turma = $row["id_turma"];  
    $periodo = $row["periodo"];  
    $dia_semana = $row["dia_semana"];  
    $inicio = $row["inicio"];  
    $fim = $row["fim"];  
    $dias = ["Domingo", "Segunda-Feira", "Terça-Feira", "Quarta-Feira",  
    $tt = "";  
    for($q =0; $q<sizeof($dias); $q++){
```

RECONHECIMENTO FACIAL

RECONHECIMENTO FACIAL

```
8
9  PATH = "/opt/lampp/htdocs/uploads/"
10 if len(sys.argv) == 2:
11     faceid = face_recognition.load_image_file(PATH+"faces_imagens/"+sys.argv[1]+".png")
12
13     if faceid is not None:
14         faceid = face_recognition.face_encodings(faceid)[0]
15         faceid.dump(PATH+"faces_encodes/"+sys.argv[1] + ".npy")
16         print("1")
17     else:
18         print("2")
19 else:
20     print("3")
21     print("Usage: python3 extract_faceid.py <image_name>")
22     sys.exit(1)
```





MODULO DO RECONHECIMENTO FACIAL

```
face_recognition.api.face_encodings(face_image, known_face_locations=None, num_jitters=1,  
model='small') [source]
```

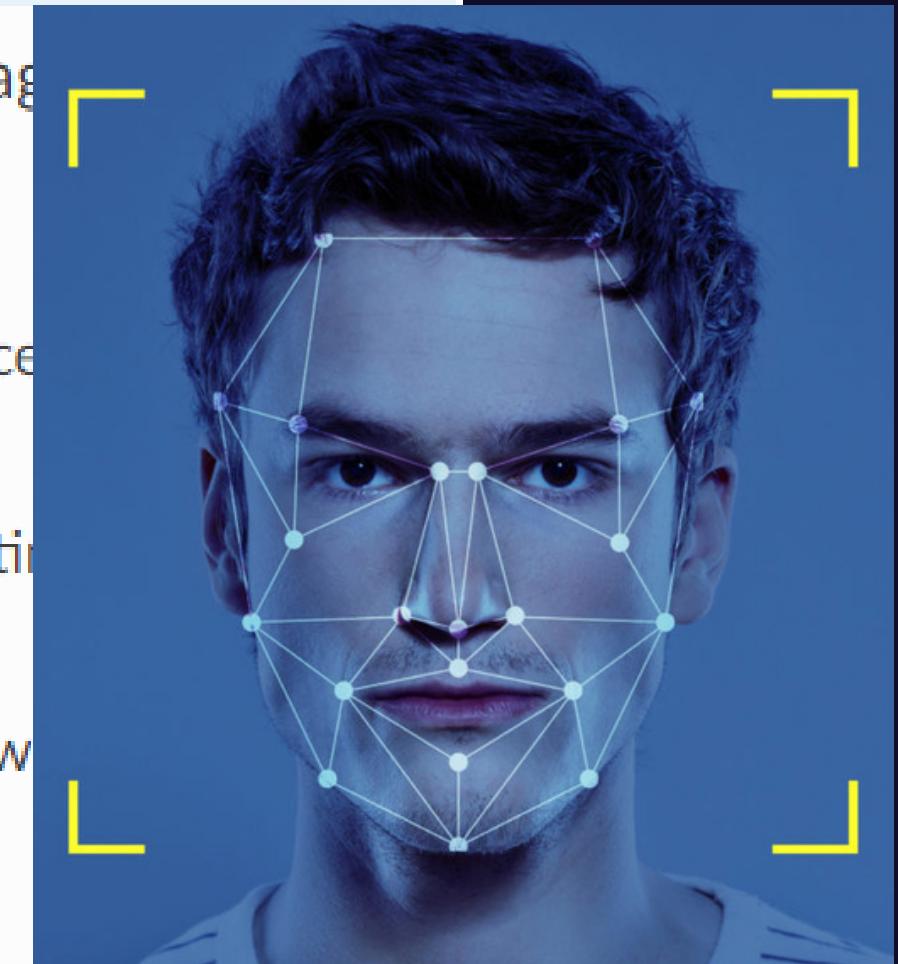
Given an image, return the 128-dimension face encoding for each face in the image.

Parameters:

- `face_image` – The image that contains one or more faces
- `known_face_locations` – Optional - the bounding boxes of each face you already know them.
- `num_jitters` – How many times to re-sample the face when calculating encodings. Higher is more accurate, but slower (i.e. 100 is 100x slower)
- `model` – Optional - which model to use. “large” or “small” (default) where “large” returns 5 points but is faster.

Returns:

A list of 128-dimensional face encodings (one for each face in the image)



COMPARAÇÃO DOS ENCODINGS

```
# process once per 10 frames
if c % 10 ==0:

    # Resize frame of video to 1/4 size for faster face recognition processing
    small_frame = cv2.resize(frame, (0, 0), fx=(10/RES)/10, fy=(10/RES)/10)

    # Convert the image from BGR color (which OpenCV uses) to RGB color (which face_recognition uses)
    rgb_small_frame = small_frame[:, :, ::-1]

    # Find all the faces and face encodings in the current frame of video
    face_encodings = face_recognition.face_encodings(rgb_small_frame)

for face_encoding in face_encodings:
    face_distances = face_recognition.face_distance(known_face_encodings, face_encoding)
    if len(face_distances) == 0: # avoid some errors, although it could be better implemented
        continue
    best_match_index = np.argmin(face_distances)
    # print(alunos_turma[best_match_index]["nome"], " - ", face_distances[best_match_index])

    if face_distances[best_match_index] < 0.55: # and alunos_turma[best_match_index] in alunos_turma:

        # print(alunos_turma[best_match_index])
        id = alunos_turma[best_match_index]["id"]
        # removing both face and aluno from the list
        alunos_turma.pop(best_match_index)
        known_face_encodings.pop(best_match_index)
        cursor.execute("update presencas set present = 1 where id_aluno = %s and id_horario = %s and _data=%s", (id, id_horario, now.strftime("%Y-%m-%d")))
        cursor.execute("commit")
```

```
1  <?php  
2  
3  require_once ("../database/connect.php");  
4  
5  
6  session_start();  
7  if (isset($_SESSION['id'])){  
8      header("Location: /v1/turmas?r=Ja_logado");  
9      die();  
10 }  
11  
12  
13 if (! $_POST || !$POST["login"] || !$POST["senha"]){  
14     header("Location: /v1/login?r=Acesso_desbloqueado");  
15     die();  
16 }  
17  
18 $hash = md5($_POST['senha']);  
19 $mat = strtolower($_POST["login"]);  
20  
21 var_dump($conn);  
22 $q = mysqli_query($conn, "SELECT * FROM alunos WHERE matricula = '$mat' AND senha = '$hash'");  
23  
24 if ($q && (mysqli_num_rows($q) == 1)){  
25     $r=mysqli_fetch_assoc($q);  
26     $_SESSION['id'] = $r['id'];  
27     $_SESSION['matricula'] = $r['matricula'];  
28     $_SESSION['nome'] = $r['nome'];  
29     header("Location: /v1/turmas?r=senha_cadastrada");  
30     die();  
31 }else{  
32     header("Location: /v1/login?r=senha_errada");  
33     die();  
34 }
```

Conclusão?

- Fase de Teste na FECITI foi um sucesso
- Um aluno em diversas turmas

Obrigada!

- waltermaussbenjamin@gmail.com
- kellenkern.sg454@academico.ifsul.edu.br