* **8 O Projeto de Software**

A solução apresentada neste projeto tem como característica uma arquitetura distribuída, podendo ser segregada nas seguintes responsabilidades:

* Módulo para Captura de Imagens;
* Servidor para Reconhecimento Facial;
* Aplicação WEB;
* Base para Armazenamento de Dados.
* **8.1 Módulo para Captura de Imagens**

Para o módulo que possuí a responsabilidade de capturar as imagens dos presentes na sala, foi utilizado um microcontrolador chamado ESP32-CAM, um microcontrolador de baixo custo financeiro e baixo custo energético para o funcionamento.

A placa, produzida pela empresa chinesa *Espressif Systems,* é considerada uma SoC (do inglês “System on Chip”, ou “Sistema em um Chip”), ou seja, contém todos os recursos integrados na mesma placa, que possuí as funções Wi-Fi e Bluetooth, além da câmera OV2640, 512 KB de memória RAM, 32 Mb de memória Flash, e um processador ESP32-D0WD[referência].

Para o desenvolvimento do software embarcado, foi utilizada a IDE Arduino como ambiente de desenvolvimento, com a linguagem de programação C.

O funcionamento do módulo é dado pela disponibilização de um módulo por sala de aula, tendo um *endpoint* na rede local para que, ao receber uma requisição HTTP neste *endpoint*, o módulo então capture uma imagem, e a devolva para o *client* da requisição. Sua responsabilidade é delimitada por esta ação: retornar a imagem da sala de aula quando solicitada para que o servidor realize o reconhecimento facial.



Figura XX: Placa ESP32-CAM (Dimensões: 40 x 27,5 x 4.5 mm)

Fonte: XXX.

* **8.2 Servidor para Reconhecimento Facial**

Para o servidor que executará o processamento das imagens e o reconhecimento facial, utilizamos a linguagem de programação JavaScript com a plataforma de execução Node.js, que é a responsável por interpretar o código JavaScript no lado do servidor (*server-side*).

A responsabilidade do servidor é definida por duas partes, sendo a primeira, a função de intermediar as ações da *aplicação WEB* (Tópico 8.3), processando qualquer cadastro, atualização de registro, ou remoção feita na aplicação WEB por algum usuário, agindo como a camada de controle da arquitetura, baseando-se no padrão MVC (Model, View e Controller).

A segunda responsabilidade da aplicação do servidor, é a de executar o reconhecimento facial. Quando a aula for iniciada pelo professor (ação executada na aplicação WEB – Tópico 8.3), o servidor irá solicitar ao módulo a imagem, realizando uma requisição HTTP para o endpoint definido na rede local, e ao receber a imagem, fará o reconhecimento facial, checando se o aluno está presente na imagem feita. Este processo será feito diversas vezes durante a aula, para garantir que o aluno esteja em sala durante pelo menos 75% do tempo de aula, e após o final da aula, a aplicação do servidor irá registrar a presença para o aluno ou a falta dela.

* **8.2.1 Infraestrutura Conteinerizada**

Para o desenvolvimento da solução, a viabilidade foi bastante considerada, e buscando aumentar a facilidade de implantação da solução, foi definido o uso da tecnologia de conteinerização chamada Docker.

Com o uso desta tecnologia, a aplicação do servidor ganha em portabilidade, pois se torna independente de sistema operacional e de hardware, desta forma, é criada a imagem da aplicação do servidor sendo executada com o Node.js, e essa imagem pode ser executada em qualquer hardware capaz de executar o Docker.

Com esta característica, a definição do servidor se torna muito versátil, podendo-se definir como servidor desde supercomputadores robustos até placas com recursos computacionais mais reduzidos, como por exemplo uma Raspberry Pi, adequando assim o servidor à necessidade computacional, evitando gastos desnecessários.

* **8.3 Aplicação WEB**

A aplicação WEB desenvolvida tem o objetivo de apresentar uma interface gráfica intuitiva para que toda a solução seja administrada. Todas as interações dos usuários com a solução, com exceção das fotos tiradas pelo módulo, são dados através da aplicação WEB.

Na aplicação WEB o(a) Secretário(a) irá cadastrar o(a) Aluno(a) ou atualizar seus dados, o(a) Aluno(a) irá checar suas presenças e aulas, e o(a) Professor(a) irá iniciar ou cancelar suas aulas, além de definir o tempo de aula.

Esta aplicação foi desenvolvida utilizando como linguagem o TypeScript (superset da linguagem JavaScript) com o Framework Angular, facilitando o desenvolvimento do front-end com práticas como a componentização do front-end, e trabalhando com a aplicação do servidor aplicando o padrão *client-server*, utilizando o servidor como um conjunto de Restful APIs, ou seja, programas que vão ser manipulados e utilizados através dos métodos e requisições HTTP.

* **8.4 Base para Armazenamento de Dados**

O armazenamento dos dados foi centralizado em uma única base de dados MySQL, da empresa Oracle, que utiliza o motor MariaDB. Nesse banco de dados serão armazenadas as informações referentes aos alunos, professores, secretários, aulas, e salas de aula, além das presenças registradas, e todo o fluxo de utilização desta base de dados é controlado e passa pela aplicação do servidor, nenhuma outra aplicação tendo acesso à base. As definições da modelagem da base são descritas em um tópico específico para isto ao longo deste trabalho.

É válido pontuar que, somente o armazenamento das imagens é feito de forma isolada, sem o uso da base de dados, salvando as imagens por tempo determinado diretamente no armazenamento de disco principal.

* **8.5 Arquitetura da Solução**

A arquitetura da solução é definida pela imagem abaixo:

Figura XX: Arquitetura da Solução definida visualmente.

Fonte: Própria.

Na arquitetura definida, temos algumas definições:

* Um módulo é necessário por sala, assim tirando a foto para a checagem de presença de todas as aulas ministradas naquela sala;
* Um módulo é necessário para que o(a) Secretário(a) possa realizar o cadastro do(a) Aluno(a);
* A aplicação do servidor e a base de dados estão centralizados no mesmo servidor;
* Ambos, tanto o servidor quanto os módulos devem estar em uma mesma rede.

Seguindo o modelo de arquitetura, temos dois fluxos principais na solução, **o cadastro**, e a **checagem de presença**.

No cadastro(*1*), o(a) Secretário(a) através da aplicação WEB faz o preenchimento dos dados do usuário, e utiliza o módulo que é ativado para tirar a primeira foto do aluno, enviando todos os dados para a aplicação do servidor armazenar na base centralizada.

Na checagem de presença(2), o(a) Professor(a) inicia a aula, e então o servidor calcula o intervalo em que as fotos serão tiradas para a checagem da presença, e utilizando este intervalo, requisita que o módulo presente naquela sala devolva uma imagem. Após isso, a aplicação do servidor faz o reconhecimento facial, utilizando as fotos tiradas no cadastro do aluno, e então registra as presenças, ou a falta delas.

A arquitetura da solução foi definida desta maneira para que fosse distribuída escalável, podendo suportar um número não definido de salas de aula e alunos, sendo assim uma solução viável tanto para instituições de ensino menores, quanto para as maiores instituições de ensino.