

PROJEÇÃO DE DESENVOLVIMENTO-PROJETO  
INTERDISCIPLINAR  
SMART ACADEMY

## RESUMO

Este relatório tem como objetivo demonstrar o desenvolvimento, organização, Script e cronograma, baseado no projeto apresentado 'Smart Academy' do Projeto Interdisciplinar 2023.1.

Smart Academy é uma solução voltada para melhor aproveitamento do fluxo de pessoas na academia e monitoramento da disponibilidade dos equipamentos. Se utilizando da metodologia de Auto-Gestão as academias possam oferecer aos seus usuários uma forma de se organizarem de acordo com o fluxo de pessoas e melhor aproveitamento de seu tempo, visando a redução de ocupação da mesma.

## CRONOGRAMA

Aa Tarefas	Status	Date	Descrição	Platform
 <u>Divisão de tarefas</u>	Posted	25 de julho de 2023	BACK END : -Dev ESP = Ícaro -Dev Câmera = Allan, Thiago, Raul -Dev API - Ícaro FRONT: -Dev site = Raiane	python C++ Java SQL JavaScript
 <u>ESP</u>	doing	1 de agosto de 2023 → 4 de setembro de 2023	Desenvolvimento do Script em ESP - Modelagem, Diagram de	C++ SQL
<u>Câmeras &amp; API</u>	doing	1 de setembro de 2023 → 18 de setembro de 2023	Utilização de biblioteca OpenCV para câmeras(Python) e desenvolvimento API	python SQL Java
 <u>Documentação - Primeira entrega</u>	doing	19 de agosto de 2022 → 18 de setembro de 2022	Arranjo e registro do projeto	word notion trello
 <u>Desenvolvimento Front e Correções</u>	Backlog	1 de setembro de 2022 → 19 de setembro de 2022	Desenvolvimento do Front End	JavaScript
<u>Correções</u>	Backlog	19 de setembro de 2023 → 13 de novembro de 2023	Correções do desenvolvimento	

### #Divisão de tarefas:

Para melhor desenvolvimento houve a separação de tarefas, onde:

ESP – Recurso opcional para implementação da solução

Câmeras – Recurso primário para a implementação da solução

Front – Recurso de usabilidade da solução

Back – Recurso de usabilidade da solução

Outras tarefas:

SQL – Uso comum pelas equipes

### Recursos

ESP: implementação utilizando as linguagens C, C++ :

Gerente do projeto – Ícaro Aurich

\*Captar a presença de maneira física da utilização do equipamento

Câmeras: Implementação utilizando a linguagem Python- biblioteca - OpenCv:

Gerente do projeto – Allan Pinheiro, Thiago Portugal

\*Captar a presença de maneira digital utilização câmeras

\*Desempenho de hardware(câmera) – Raul Filho

Front: Implementação utilizando JS

Gerente do projeto – Raiane Caroline

\*Usabilidade e consumo da API

Back: Implementação utilizando Java

Gerente do projeto – Ícaro Aurich

\*Desenvolvimento da API

## DESENVOLVIMENTO ATUAL - ESP

***Arquivo credenciais***

```
#ifndef Credentials_h
#define Credentials_h

char ssid[] = "Aurich";
char pass[] = "12345678";

char user[]      = "usuario"; // "espressif" foi o usuário criado no início
char password[]  = "senha";    // senha criada com o usuário

#endif //Credentials_h
```

Armazena as informações para o ESP se conectar com a rede Wifi e credenciais para acesso a rede de dados

***Iniciação da credenciais***

```
1  #include "Credentials.h"
2
3  #define MYSQL_DEBUG_PORT    Serial
4  #define _MYSQL_LOGLEVEL_    1
5
6  #include <MySQL_Generic.h>
7
8  IPAddress server(192, 168, 183, 226); //Trocar pelo IP da máquina PELO IP DA SUA BASE SQL
9
10 uint16_t server_port = 4407;
11
12 MySQL_Connection conn((Client *)&client);
13 MySQL_Query *query_mem;
14
```

Parâmetros iniciais, importa a biblioteca MySQL, define o endereço IP e porta do banco de dados e instancia o Objeto de conexão

***Parametrização inicial***

```

void setup(){
  Serial.begin(115200);
  while (!Serial && millis() < 5000);

  MYSQL_DISPLAY1("\nIniciando ", ARDUINO_BOARD);
  MYSQL_DISPLAY(MYSQL_MARIADB_GENERIC_VERSION);
  MYSQL_DISPLAY1("Conectando a rede: ", ssid);

  WiFi.begin(ssid, pass);

  while (WiFi.status() != WL_CONNECTED){
    delay(500);
    MYSQL_DISPLAY0(".");
  }

  MYSQL_DISPLAY1("Conectado na rede. Meu endereço de IP é: ", WiFi.localIP());
  MYSQL_DISPLAY3("Tentando conexão em @", server, ", Porta =", server_port);
}

```

Inicia o serviço no serial/frequência 115200 e conecta a rede Wifi e servidor SQL

***Método de inserção no banco de dados***

```

void runInsert(){
  long rand_val = random(0,500);

  MySQL_Query query_mem = MySQL_Query(&conn);

  if (conn.connected()){
    // Comando SQL insert teste
    String sql_query = "insert into academy.teste(name,value) values('esp32'," + String(rand_val) + ");";
    MYSQL_DISPLAY(sql_query);

    if (!query_mem.execute(sql_query.c_str())){
      MYSQL_DISPLAY("Erro :(");
    }
    else{
      MYSQL_DISPLAY("Dados inseridos com sucesso.");
    }
  }
  else{
    MYSQL_DISPLAY("Desconectado do servidor. Não foi possível inserir.");
  }
}

```

Método de inserção no banco de dados. Inicia conexão e inserir nome do dispositivo (ESP)

## Controle de registro

```
void loop(){
    MYSQL_DISPLAY("Conectando...");

    if (conn.connectNonBlocking(server, server_port, user, password) != RESULT_FAIL){
        delay(500);
        runInsert();
        conn.close();
    }
    else{
        MYSQL_DISPLAY("\nFalha na conexão. Tentando novamente em 1 minuto.");
    }

    MYSQL_DISPLAY("\nSleep delay...");
    MYSQL_DISPLAY("=====");

    delay(60000);
}
```

Executa o método *Insert*, e Controla o registro

GITHUB+API : <https://github.com/icaroaurich/SmarthAcademy/tree/main>