

**Departamento de Engenharia Eletrotécnica**

Mestrado em Engenharia Eletrotécnica

**Projeto – Controlo e Monitorização Inteligentes na Agricultura**

**Unidade Curricular de Desenvolvimento de Aplicações Multiplataforma**

**Trabalho realizado por:**

Fernando Marques nº2202576

Nuno Silva nº2202578

**Docente:** Professor João Ramos

Índice

[1. Introdução 4](#_Toc1694057702)

[2. Arquitetura da Solução 5](#_Toc32476696)

[3. Principais módulos implementados 6](#_Toc997670539)

[4. Hardware utilizado 8](#_Toc319095397)

[5. Características e funcionalidades da solução 10](#_Toc638794463)

[6. Conclusões 11](#_Toc2044063933)

[7. Autoavaliação 12](#_Toc1120982624)

# Introdução

Este relatório tem como objetivo mostrar a aplicação dos conhecimentos adquiridos na UC de Desenvolvimento de Aplicações Multiplataforma. Pretende-se a construção de uma aplicação baseada em conteúdos HTML com recurso a ferramentas como o Bootstrap, Node.js, Vue.js, Firebase, entre outras. Esta aplicação permitirá, através de uma página principal (Monitorização), a visualização e o controlo de dispositivos em tempo real, a gestão de regras desses mesmos dispositivos, que gera limites de atuação, via página de regras, e a gestão dos mesmos sensores com recurso a uma página de settings que permitirá assim adicionar e remover dispositivos, estando a mesma conectada a uma base de dados online.

Os dispositivos acima indicados, estarão divididos em duas classes, sensores e atuadores. Os sensores serão de hardware físico que estarão ligados a um microcontrolador (ESP32), que por sua vez fará a aquisição dos sensores e registará o seu valor na base de dados. Os atuadores estarão disponíveis na interface de utilizador e que, através da mesma base de dados, permitirá a interação com o mesmo microcontrolador. Este microcontrolador terá um código próprio que fará a interação com os valores lidos nos sensores e base de dados, e com os estados dos atuadores e as suas saídas.

Será também tido em conta, a utilização de API’s, que mostrará informação de interesse para o utilizador.

# Arquitetura da Solução

A arquitetura do projeto rege-se pelo diagrama da Figura 1. O Esp32 controla a aquisição de sinais dos sensores de cada zona, neste caso da Estufa e do Jardim da Piscina, e controla também as ações das saídas ordenadas por intermédio dos atuadores (ações do utilizador). Estes dados fluem de forma bidirecional entre o Esp32 e a base de dados Firebase. Os mesmos dados estabelecidos também entre a Firebase e o dispositivo do utilizador (Servidor e Cliente) onde corre a aplicação.

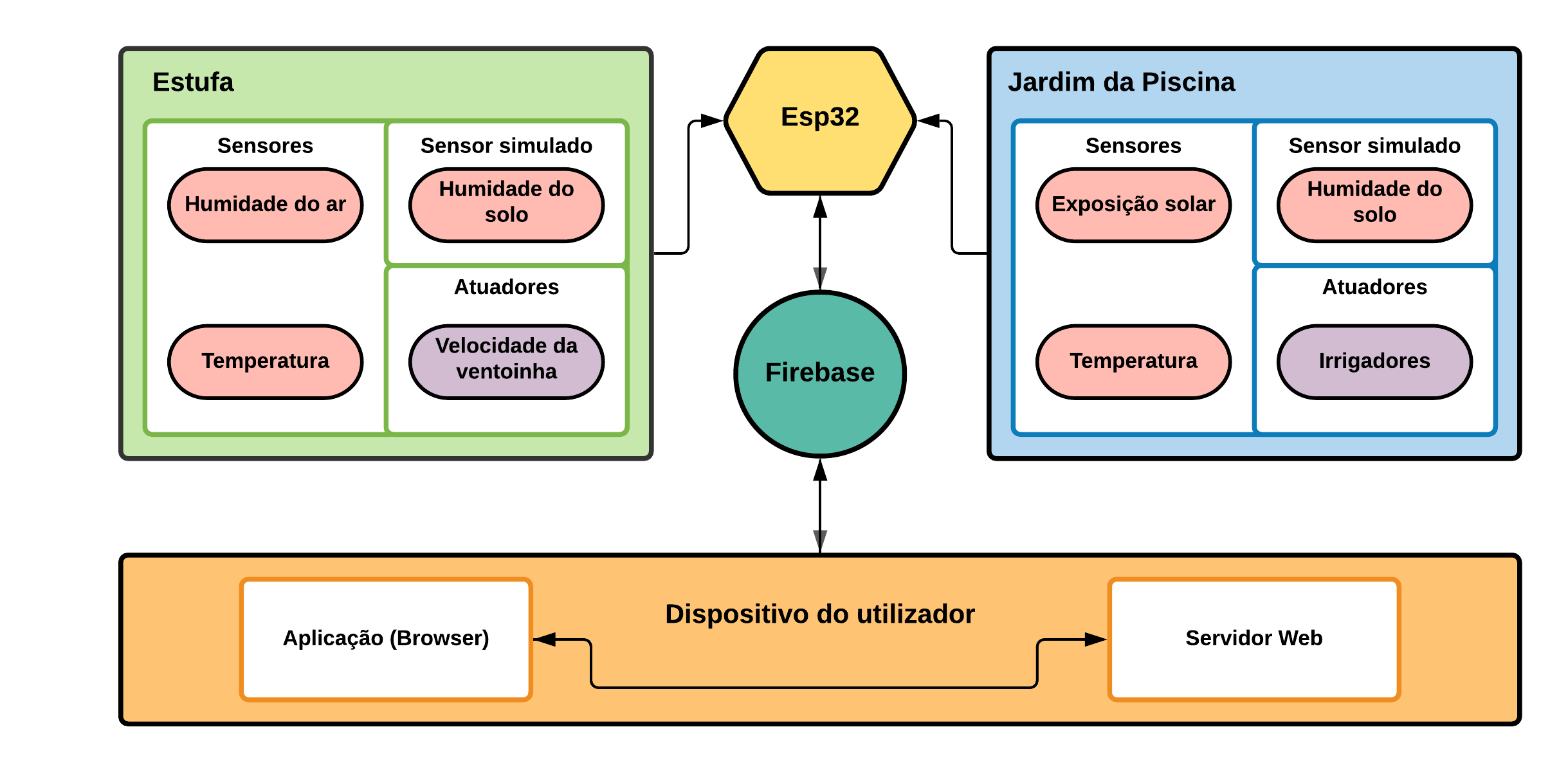


Figura 1 - Arquitetura da solução

A arquitetura da aplicação, Vue.js, segue a estrutura de componentes que é demonstrada no diagrama da Figura 2. É montado na página principal uma barra de navegação que contém o nome e, por intermédio de três links, faz a comutação entre as páginas principais (Monitorização, Regras e Settings). A página de monitorização é composta por quatro componentes onde são apresentadas as informações de interesse recolhidas das API’s, como informações climáticas, do boletim diário da pandemia Covid-19, assim como também uma listagem de sensores e atuadores disponíveis por cada zona (Estufa e Jardim da Piscina). Na página de regras, um componente de notificações constrói a estrutura onde são geridas as regras. Para cada sensor são criados notificações e limites, que podem ser definidos pelo utilizador, através dos componentes filhos do mesmo. Esta informação também se encontra organizada por zona. Na página de *settings*, o utilizador pode eliminar e/ou adicionar sensores/atuadores por zona, como poderemos ver numa descrição mais detalha seguidamente no **Principais módulos implementados**.

Figura 2 - Arquitetura da aplicação

# Principais módulos implementados

A aplicação do cliente, concebida em Vue.js como já mencionado anteriormente, é baseada em JavaScript. O Vue permite usar uma sintaxe baseada em HMTL e permite executar métodos tendo como dados as instâncias subjacentes do Vue. Para embelezar a aplicação é utilizado o Bootstrap que assim cria componentes de interface com o cliente no browser bastante apelativos.

A aplicação desenvolvida conta com as três páginas acima descritas.

A página de monitorização, contém informação geral do ambiente e dos dados gerais da pandemia, recolhidos de API’s, apresenta também informação referente às zonas onde estão os dispositivos IoT separa esta por *cards*. Cada *card* (zona), tem uma listagem de sensores/atuadores referentes a ela mesma, ou seja, os que estão definidos na base de dados dentro do *array* dessa mesma zona, seja Estufa e Jardim da Piscina.No caso da zona da Estufa, foi implementado um atuador do tipo *slider* que permite o ajuste de velocidade da ventoinha de circulação de ar numa gama de valores de 0 a 100%. O atuador do jardim da piscina é binário, ou seja, é utilizado um botão do tipo *toggle* para ligar ou desligar os irrigadores. No final do ecrã podem surgir mensagens de alerta, se assim estiverem definidos na págia de regras, como vamos explicar já de seguida, e assimquando os limiares dos sensores forem excedidos. Na Figura 3 é possível observar se o layout do primeiro ecrã.

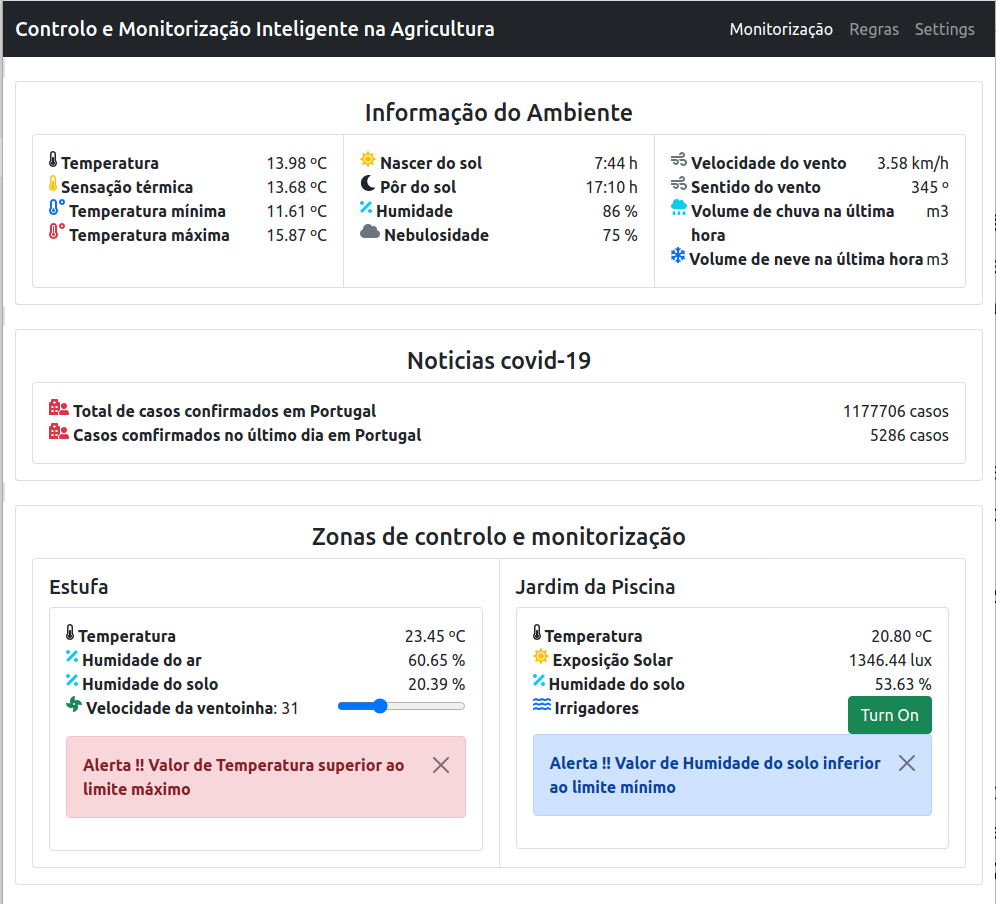


Figura 3 - Ecrã Monitorização

O ecrã das regras permite definir regras para todos os sensores. São utilizados botões para definir se uma certa notificação deve ou não estar ativa, estes mudam de cor consoante o estado do botão (*true* ou *false)*. De forma a definir os limiares dos sensores é utilizado o componente *input* que contém uma caixa de texto para capturar o valor inserido nesta. Na Figura 4 é possível observar se o ecrã das regras.

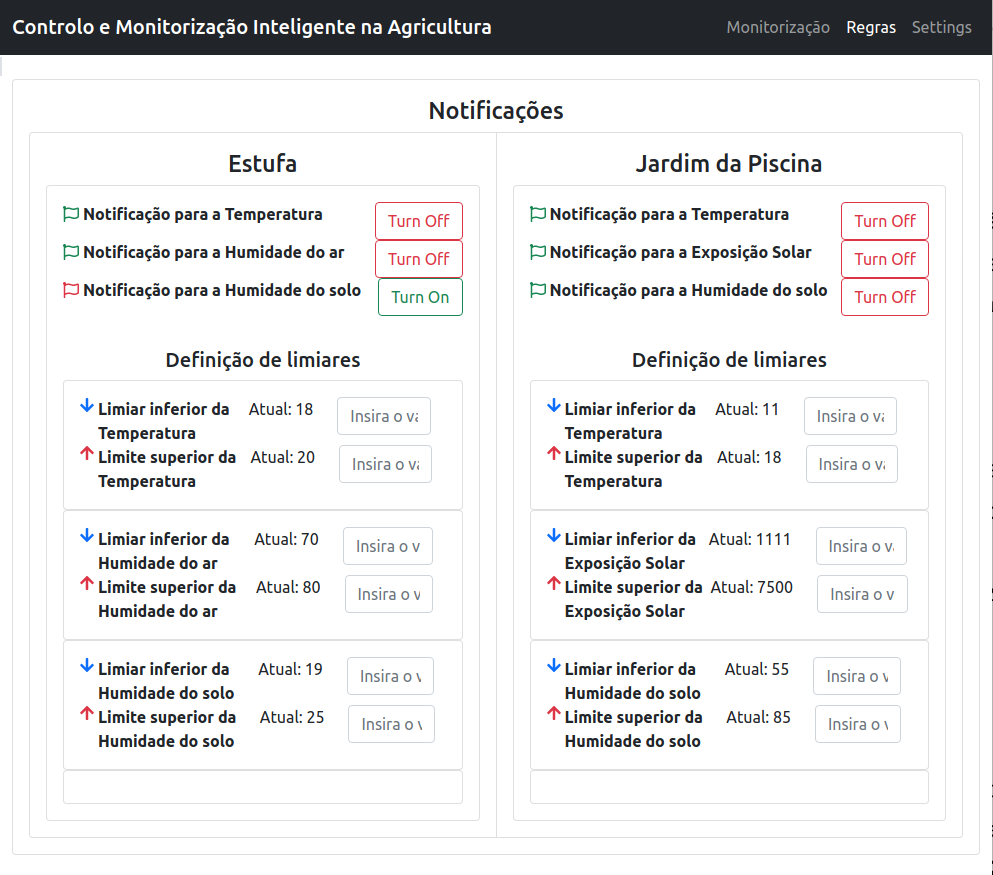


Figura 4 - Ecrã Regras

O ecrã das *settings* contém 2 conjunto de *cards*, um para remover sensores/atuadores e outro para adicionar sensores/atuadores. Para remover um sensor é necessário pressionar o ícone do “lixo” que está associado a esse item. Quando pressionado, surge uma mensagem num *popUp* de alerta, que informa o utilizador que está prestes a eliminar o item a que o botão corresponde. No modal que surge, só quando se clica no botão que confirmação é que o sensor ou atuador é eliminado da base de dados *firebase*, e só desta forma é que o mesmo deixa de aparecer na página de monitorização assim como as suas regras também deixam de existir. Para adicionar sensores ou atuadores é utilizado um sistema análogo. A característica dos mesmos tem de ser inserida nos componentes de *input*, de forma interpretável para o sistema, para assim adquirir as várias definições do sensor/atuador. Da mesma forma que no anterior, ao clicar no ícone verde com um símbolo mais, surge um *popUp* do tipo modal que transmite a mensagem que está prestes a adicionar um item. Só após a confirmação, este fica registado na base de dados *firebase* e assim disponível para monitorização e aplicação de regras. O utilizador pode cancelar a ação clicando numa outra área a página que não seja o botão de confirmação. Depois de adicionado o item, as caixas de texto usadas para dar entrada das propriedades dos itens são limpas. Na Figura 5 é possível observar se o ecrã das *settings*.

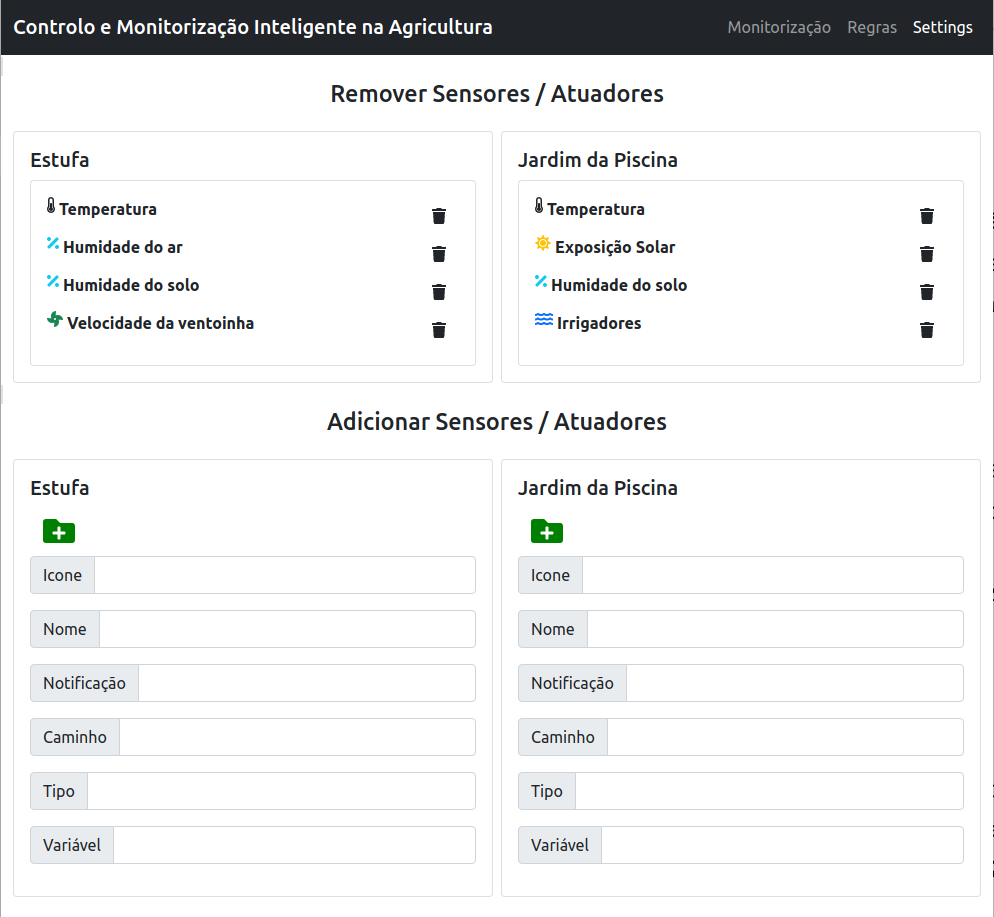


Figura 5 - Ecrã Settings

# Hardware utilizado

O objetivo principal do presente projeto é o de executar controlo e monitorização inteligente na Agricultura. Desta forma para simular um ambiente de agricultura são utilizadas duas zonas distintas. São obtidos sensores e atuadores para obter informação de cada zona e atuar sobre essa informação, como já abordado anteriormente.

Para recolher a informação dos vários sensores e poder-se atuar em cada zona é utilizado um microcontrolador com ligação á internet. É utilizado um ESP32 que se conecta á base de dados do projeto de forma a fornecer e receber várias informações. Uma forma mais real de simular ambas as zonas seria a de utilizar um ESP32 para cada zona, mas para testes um microcontrolador foi suficiente.

Na zona da estufa são utilizados como sensores o bme280, para obter a humidade do ar presente na estufa, e o sensor si7021, para obter a temperatura presente dentro da estufa. A ligação destes sensores ao ESP32 é feita por i2c. Para além destes 2 sensores é gerado um sensor virtual para simular a humidade do solo. Como esta zona se trata de uma estufa é utilizada uma ventoinha de computador como atuador com o objetivo de controlar a temperatura e humidade do ar. De forma a se puder controlar a velocidade da ventoinha é utilizado um MOSFET que recebe um sinal PWM do microcontrolador. Por fim é utilizado um LED como indicador de temperatura elevada, função das regras definidas. Na Figura 6 é possível observar o hardware utilizado nesta zona.

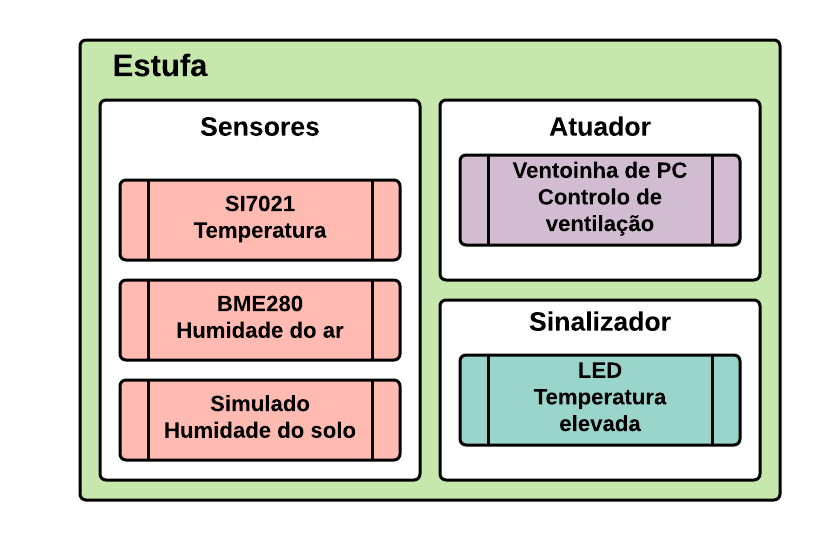


Figura 6 - Componentes Estufa

Na zona do jardim da piscina são também utilizados 2 sensores, 1 atuador e 1 sinalizador. Como sensores temos o LM35, para obter a temperatura da zona do jardim, e o sensor LDR, para obter a intensidade luminosa desta zona. Para além destes 2 sensores é também utilizado um sensor virtual para simular a humidade do solo do jardim. Para esta zona decidimos utilizar um irrigador como atuador de forma que fosse possível regar as plantas do jardim. Para acionar o irrigador é utilizado um relé de potência e para controlar este relé é utilizado um MOSFET. Foi necessário utilizar um MOSFET pois a saída do microcontrolador não consegue fornecer corrente suficiente para controlar a bobine do relé. Tal como na zona da estufa é utilizado um LED como indicador de temperatura elevada. Na Figura 7 é possível observar se o hardware utilizado nesta zona.

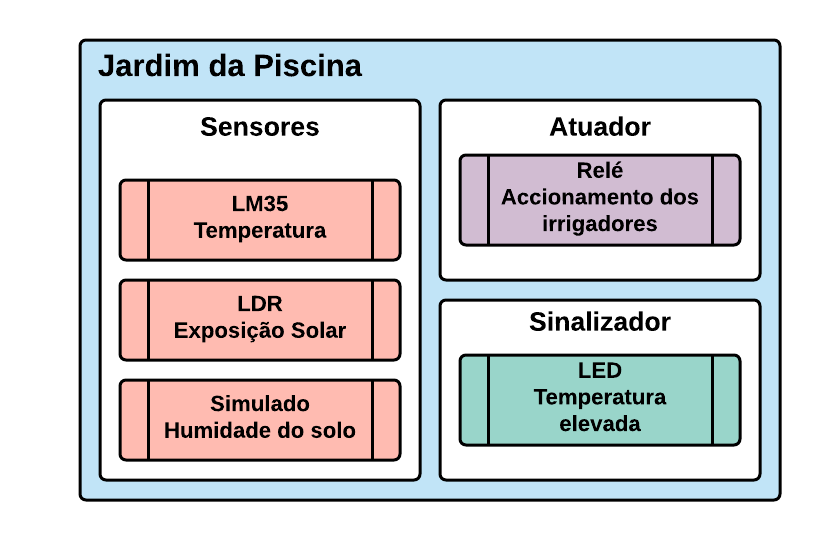


Figura 7 - Componentes Jardim da Piscina

Para além destes componentes é utilizado uma fonte de 12v para alimentar a ventoinha de computador e um regulador de tensão para 5v para alimentar a bobine do relé e o sensor LM35. Na Figura 8 é possível observar o circuito completo desta solução.

Falta Figura 8 com o circuito.

# Outras Características e funcionalidades da solução

A solução tem como principal objetivo monitorizar e controlar sensores/atuadores das duas zonas (Estufa e Jardim da piscina), mas para além deste objetivo a solução apresenta 12 informações sobre o ambiente da localização de Leiria, esta informação é composta por dados meteorológicos. Ainda na página de monitorização é ilustrada informação sobre o total de casos de covid em Portugal e os casos de covid em Portugal, verificamos que estes dados estavam coerentes com a informação dada pela Direção Geral de Saúde.

Para os alertas surgem mensagens na página com cores alusivas, utilizando assim a cor vermelha para quando o limite superior é ultrapassado e azul para o limite inferior. Para esta característica estar a funcionar é necessário que a notificação esteja ativada na página de regras, como falado anteriormente. Existe uma notificação independente para cada sensor e logo é possível ativar a notificação para cada sensor.

Como medida de proteção é definido um tamanho máximo de dígitos que podem ser inseridos na definição dos limiares dos sensores.

Para que sejam carregados ícones quando se adiciona um item, é necessário inserir o código fonte desse ícone que está disponível na página do FontAwesome V5.7.2.

# Conclusões

As aplicações IoT estão cada vez mais presentes no nosso dia a dia. Neste trabalho foi possível desenvolver um uma aplicação suportada em html e vue.js que controla um dispositivo IoT (ESP32), programado para recolher informação de sensores e desencadear ações solicitadas pelo utilizador. Vue.js mostrou ser uma forma de programação bastante simples e versátil, permitindo, com pouco código, passar informação de componentes pais para componentes filhos de forma fácil. Permite também fazer uma programação baseada numa estrutura que depois é utilizada para vários componentes, como por exemplo, nos *cards* um único cartão foi programado e usado várias vezes em função da zona que estávamos a interagir/monitorizar. Da mesma forma que para a apresentação dos sensores/atuadores um único componente foi criado, que depois é executado numa função do tipo *for – v-for*, e dessa forma apresenta todos os sensores/atuadores em lista, mudando os seus atributos (leitura do sensor ou botão) em função das suas propriedades na base de dados *firebase*.

# Autoavaliação

Todos os requisitos foram compridos, a apresentação de informações gerais de ambiente, duas zonas com dois ou mais elementos de diferentes tipos de sensores e atuadores, a implementação de regras na aplicação, utilização de dois API’s e apresentação de dados em tempo real. A pesquisa de exemplos mostrou ser uma grande ajuda no desenvolvimento do projeto e apesar de algumas dificuldades encontradas durante a programação e testes, estes foram ultrapassados com a ajuda do docente.