Escola Superior de Tecnologia e Gestão

**Estufa Inteligente**

Daniel Filipe Silva Brás

Diogo Miguel Ferreira Gomes

Trabalho de Projeto da Unidade Curricular de Tecnologias de Internet

Leiria, junho de 2023

# Lista de Figuras

[Figura 1 - Arquitetura Estufa Inteligente 3](#_Toc137454608)

[Figura 2 – Diagrama de representação da ativação de alarme e câmara 4](#_Toc137454609)

[Figura 3 – Cenário de Teste 6](#_Toc137454610)

# Lista de tabelas

[Tabela 1 – Valores de temperatura medidos em situação de teste 8](#_Toc137454499)

[Tabela 2 – Valores de humidade medidos em situação de teste 8](#_Toc137454500)

# Lista de siglas e acrónimos

|  |  |
| --- | --- |
| ESTG | Escola Superior de Tecnologia e Gestão |
| IPLeiria | Instituto Politécnico de Leiria |
| RPi | Raspberry Pi |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Índice

[Lista de Figuras iv](#_Toc137417615)

[Lista de tabelas v](#_Toc137417616)

[Lista de siglas e acrónimos vi](#_Toc137417617)

[1. Introdução 1](#_Toc137417618)

[2. Arquitetura 2](#_Toc137417619)

[2.1. Arquitetura Geral 2](#_Toc137417620)

[2.2. Sensores Utilizados 2](#_Toc137417621)

[2.3. Atuadores Utilizados 3](#_Toc137417622)

[3. Implementação 4](#_Toc137417623)

[4. Cenário de Teste 5](#_Toc137417624)

[5. Resultados obtidos 7](#_Toc137417625)

[6. Conclusão 9](#_Toc137417626)

[7. Bibliografia 10](#_Toc137417627)

# Introdução

No âmbito da cadeira de Tecnologias de Internet, foi-nos solicitado que desenvolvêssemos um protótipo de uma solução baseada em Internet of Things (IoT) e optámos por criar um projeto com base no tema “Estufa Inteligente”.

Neste projeto temos como objetivo criar uma solução automatizada para uma estufa melhorando a eficiência do cultivo das suas plantas, fornecendo um ambiente controlado onde fatores como temperatura, humidade e luminosidade são monitorados e desencadeiam modificações significativas no seu ambiente, afetando o desenvolvimento das plantas e, consequentemente, a sua produtividade. Para tal utilizamos um cenário real de sensores, atuadores e controladores com capacidade de comunicação, de modo a interagir em tempo real com um servidor web, disponibilizando assim a informação Anytime *&* *Anywhere*.

# Arquitetura

## Arquitetura Geral

A arquitetura do nosso projeto consiste em três principais componentes: os sensores, os atuadores e os controladores, todos interligados por uma rede IoT. Os sensores são responsáveis por recolher informações ambientais, como temperatura, humidade do ar e intensidade da luz. Esses dados são enviados para os controladores, que interpretam as informações e acionam atuadores, como sistema de iluminação, alarme e câmara. Além disso, os controladores, Arduino MKR1000 e Raspberry Pi, enviam os dados recebidos para um servidor web, permitindo o acesso aos dados através de uma interface amigável (Dashboard) para o utilizador.

Na Dashboard é possível monitorar os valores registados pelos sensores, juntamente com a data e hora de atualização, o status geral da estufa, os atuadores em funcionamento, etc. Por meio da secção de histórico os utilizadores poderão ter acesso às últimas leituras realizadas pelos sensores assim como verificar se os atuadores estiveram ativos ou inativos caso já existam dados a apresentar. Na secção de estatíticas será possível visualizar dados estatísticos específicos de cada sensor ou atuador consoante o seu histórico. E finalmente, na secção dedicada à câmara, não só é possível visualizar a última imagem capturada e a respetiva hora de captura, mas também ter acesso a um histórico que armazena as três últimas imagens capturadas.

## Sensores Utilizados

Para garantir a precisão das medições físicas, decidimos utilizar um sensor de temperatura e humidade, que ligará ao Arduino MKR1000, para monitorar as condições do ar dentro da estufa. Optámos por escolher também um sensor de luminosidade para medir a intensidade da luz na estufa. Estes sensores são capazes de enviar os dados recebidos para os controladores de forma precisa e contínua, atualizando dinamicamente os valores apresentados na Dashboard.

## Atuadores Utilizados

Como atuadores, implementámos um sistema de alarme, onde são ativados Leds, um no Arduino MKR1000 e outro no Raspberry Pi, caso nenhum dos sensores meça as condições físicas desejáveis. Caso o utilizador pretenda, pode também ligar um sistema de iluminação da estufa através de um botão na dashboard, sistema esse que é composto também por 2 Leds dispostos da mesma maneira que os Leds de alarme, isto é, em ambos os controladores.

Conforme solicitado, incluímos também uma câmara de segurança no projeto que será ativada simultaneamente com a ativação do alarme. Essa funcionalidade também será acionada quando nenhum dos sensores detetar as condições físicas desejadas.

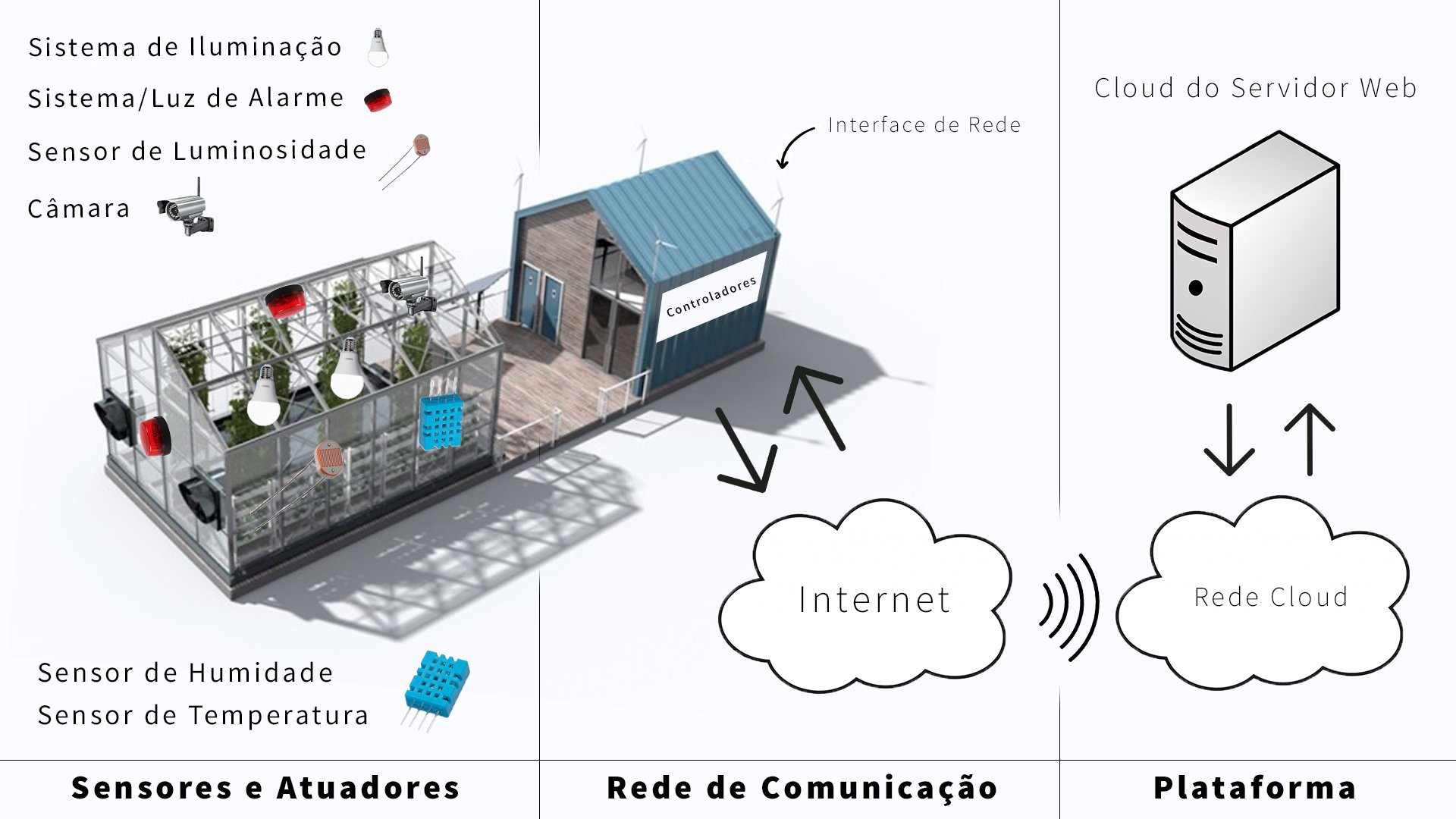


Figura - Arquitetura Estufa Inteligente

# Implementação

Na implementação do projeto, uma das funcionalidades cruciais incorporadas é a capacidade de controlar a iluminação dentro da estufa por meio da Dashboard. Nesta interface, os utilizadores têm a opção de manipular Leds que são responsáveis pela iluminação apropriada para o ambiente da estufa.

Uma imagem com texto, diagrama, Paralelo, Tipo de letra

Descrição gerada automaticamenteAdicionalmente, integrámos mais Leds que têm a finalidade de sinalizar situações de emergência. Estes Leds são acionados quando as condições ambientais na estufa se desviam da faixa ótima pré-definida. Especificamente, são ativados quando a temperatura medida está abaixo de 15°C ou excede 35°C. Simultaneamente, a humidade relativa do ar deve estar abaixo de 50% ou acima de 70%, e a intensidade da luz captada pelo sensor deve ser consideravelmente baixa ou inexistente.

Figura – Diagrama de representação da ativação de alarme e câmara

# Cenário de Teste

No nosso cenário de teste, utilizámos dois dispositivos principais: um Arduino MKR1000 e um Raspberry Pi. Para o Arduino MKR1000 utilizámos os seguintes equipamentos e portas:

* Sensor de Temperatura e Humidade: ligado ao VCC, GND e pino digital 7.
* Led Vermelho (para representar o Alarme de Emergência): ligado ao GND e pino digital 9.
* Led Verde (para representar a Iluminação): ligado ao GND e pino digital 8.

O software utilizado para programar o Arduino foi o Arduino IDE, onde escrevemos um código em C++ para ler os dados dos sensores, acionar os atuadores e comunicar com o servidor via Wi-Fi. O Led de iluminação é ligado quando o valor do "atuadordeLuminosidade" é igual a 1 e desligado quando o valor é igual a 0. O Led de alarme, por sua vez, depende tanto dos sensores presentes no Arduino MKR1000 quanto do botão de pressão no RPi.

No Raspberry Pi, foram utilizados os seguintes equipamentos e portas:

* Botão de Pressão (para representar um Sensor de Luminosidade): ligado ao GND e pino físico 12.
* Led Vermelho (para representar o Alarme de Emergência): ligado ao GND e pino físico 5.
* Led Verde (para representar a Iluminação): ligado ao GND e pino físico 3.

A linguagem de programação utilizada para codificar o RPi foi Python, utilizando a biblioteca RPi.GPIO para controlar os pinos GPIO. A comunicação com o servidor foi feita através da biblioteca requests, para fazer requisições HTTP ao servidor. O sistema operativo utilizado foi o Raspberry Pi OS. Pretendíamos utilizar um sensor de luminosidade para medir a intensidade da luz mas como este não possui canais analógicos, em vez de medirmos um valor de 0 a 1024 com um LDR, decidimos então optar por escolher um botão de pressão que irá funcionar como sensor. Caso o botão esteja pressionado (o valor enviado seja 0), significa que não teremos luz e caso não esteja (sendo o valor enviado 1), significa que existirá luz.

O Led de iluminação funciona da mesma maneira que o Led correspondente no Arduino MKR1000 (quando o utilizador clica num botão na dashboard). O Led de alarme também segue o mesmo princípio, sendo ativado quando não há luz (botão pressionado) e quando os sensores de temperatura e humidade presentes no Arduino MKR1000 não lerem os valores desejados.

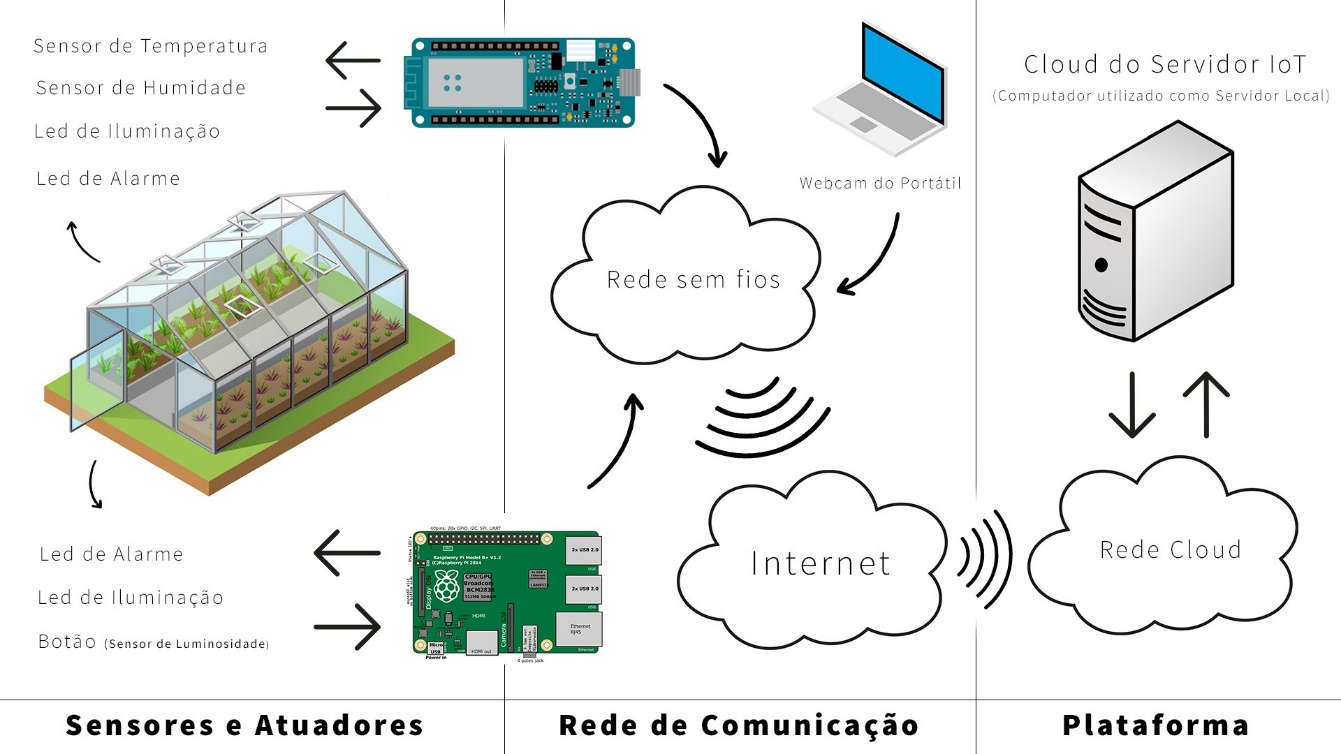
Para a câmara, criamos um código em Python que é compilado diretamente no portátil, permitindo o acesso à câmara padrão (webcam). A biblioteca utilizada para acessar a câmara foi a cv2, e para armazenar as imagens utilizamos o sistema de arquivos do próprio sistema operativo do portátil. Caso o alarme seja ativado devido a condições físicas desfavoráveis dentro da estufa, a câmara captura uma imagem a cada segundo. No entanto, é importante notar que pode haver um pequeno atraso ao ligar e desligar a câmara, o que pode resultar em variações no intervalo entre as capturas das imagens.

Figura – Cenário de Teste

# Resultados obtidos

Os procedimentos para realizar os testes e chegar aos resultados obtidos envolveram o seguinte:

1. Iniciar o servidor Uniform Server para executar um servidor web localmente no nosso portátil Windows.
2. Conectar o Arduino MKR1000 e o Raspberry Pi à rede Wi-Fi.
3. Compilar e carregar o código no Arduino MKR1000 e no RPi.
4. Compilar o código Python para a Câmara.
5. Abrir a Dashboard no navegador e confirmar se os dados dos sensores são exibidos corretamente.
6. Acionar os atuadores através da Dashboard e confirmar se funcionam corretamente.
7. Pressionar o botão de pressão no RPi e verificar se o alarme e a câmara são ativados.
8. Analisar as imagens capturadas pela câmara.
9. Verificar se o histórico e as estatísticas dos sensores e atuadores são armazenados corretamente no servidor.

Realizámos testes no Raspberry Pi para avaliar o funcionamento do botão presente na dashboard, utilizado para controlar a iluminação, bem como o Led de Alarme. Para testar o Led de Iluminação, ligámos o circuito e, uma vez que o botão envia o novo estado para a API, verificámos que tudo transcorreu sem problemas. No caso do Led de Alarme, optámos por enviar requisições POST para definir a temperatura em 100 ºC e a humidade em 100%. Dessa forma, conseguimos observar a diferença entre o alarme ativado (com a luminosidade a 0) e o alarme desativado (com a luminosidade a 1), bastando pressionar ou não o botão.

Com alguns testes no Arduino MKR1000, conseguimos obter os seguintes valores de temperatura e humidade:

|  |  |
| --- | --- |
| **Data de Atualização** | **Temperatura** |
| *2023-06-07 14:56:53* | *33.40* |
| *2023-06-07 14:56:59* | *33.50* |
| *2023-06-07 14:57:04* | *33.50* |
| *2023-06-07 14:57:10* | *33.60* |
| *2023-06-07 14:57:20* | *33.70* |
| *…* | *…* |
| *2023-06-07 14:58:44* | *34.40* |
| *2023-06-07 14:58:49* | *34.30* |

Tabela – Valores de temperatura medidos em situação de teste

|  |  |
| --- | --- |
| **Data de Atualização** | **Humidade** |
| *2023-06-07 14:56:53* | *37.00* |
| *2023-06-07 14:56:59* | *36.00* |
| *2023-06-07 14:57:04* | *36.00* |
| *2023-06-07 14:57:10* | *36.00* |
| *2023-06-07 14:57:15* | *36.00* |
| *…* | *…* |
| *2023-06-07 14:58:39* | *35.00* |
| *2023-06-07 14:58:44* | *34.00* |

Tabela – Valores de humidade medidos em situação de teste

Assim, finalizámos os testes com sucesso, uma vez que os resultados de temperatura e humidade estavam dentro dos parâmetros normais.

# Conclusão

Após a fase de desenvolvimento e implementação do projeto "Estufa Inteligente", pudemos verificar a eficácia e a relevância de soluções baseadas em Internet of Things (IoT) para a agricultura moderna. Com o monitoramento preciso e automatizado de variáveis cruciais para o desenvolvimento de plantas, tais como a temperatura, a humidade e a luminosidade, torna-se possível criar um ambiente controlado e otimizado para a maximização da produtividade agrícola.

Através da utilização de sensores, atuadores e controladores interligados numa rede IoT, fomos capazes de recolher e analisar dados ambientais em tempo real, permitindo intervenções imediatas e precisas sempre que necessário. Em relação à Dashboard, podemos dizer que forneceu uma visualização clara e intuitiva dos dados recolhidos, permitindo uma monitorização e controle eficazes da estufa.

Ainda que o protótipo tenha sido bem-sucedido, identificamos possíveis melhorias para o futuro, como a incorporação de um ar condicionado e um humidificador, que permitiriam um controle mais sofisticado do ambiente da estufa. A inclusão de um buzzer para o alarme de emergência também poderia aumentar a segurança e eficiência da estufa. Para além disso, consideramos que a secção de estatísticas poderia ser enriquecida com a inclusão de mais indicadores e análises, permitindo um melhor entendimento do comportamento e do desempenho da estufa ao longo do tempo.

Em suma, os objetivos enunciados na introdução do projeto foram alcançados com sucesso. Através do desenvolvimento deste projeto, ficou clara a potencialidade da utilização da Internet of Things na agricultura, assim como a importância da evolução constante e da adaptação às novas tecnologias para a melhoria da produtividade e eficiência no setor. Este projeto marcou apenas o início de um caminho que promete ser longo e repleto de novas descobertas e avanços.

# Bibliografia

Documentação do Arduino MKR 1000 - <https://docs.arduino.cc/> ;

Documentação do Raspberry Pi - <https://www.raspberrypi.com/documentation/> ;

Youtube - <https://www.youtube.com/> ;

ChatGPT - <https://chat.openai.com/> ;