RELATÓRIO FINAL

Projeto Arduino

Tiago Freitas a2021153107

I. Introdução

O objetivo deste projeto foi desenvolver um sistema completo de “Internet das Coisas” (IoT) para fazer a monitorização da luminosidade de um ambiente em tempo real. O sistema permite não só a leitura de dados através de um sensor de luz, mas também o controlo remoto de um LED, demonstrando a comunicação bidirecional entre o hardware físico e uma interface web.

Este trabalho consolida os conhecimentos adquiridos sobre como conectar dispositivos físicos (Arduino) à internet, utilizando uma API e uma Base de Dados para guardar e visualizar a informação.

II. Arquitetura Final

A solução final está dividida em três camadas principais, conforme o planeamento inicial:

Camada Física (Embebido):

- **Hardware:** Arduino Uno, Sensor de Luz (TEMT6000) e um LED indicador.
- **Função:** O Arduino lê constantemente o nível de luz. Se estiver em "Modo Automático", acende o LED quando a luz está fora dos níveis normais (ambiente muito escuro ou muito claro). Também recebe comandos manuais para ligar/desligar o LED ou reativar o modo automatico.

Camada de Servidor (Backend e Base de Dados):

- **Tecnologia:** Python com a biblioteca Flask (API), alojada na plataforma Vercel, e PostgreSQL (Base de Dados).

- **Função:** A API funciona como o intermediário do sistema. Tem endereços específicos (/luz e /led) para tratar os dados, e uma rota principal que entrega a interface visual ao utilizador.

Camada de Interface (Frontend):

- **Tecnologia:** HTML, CSS e JavaScript simples.
- **Função:** Um site que mostra o histórico de leituras numa tabela e possui botões para o utilizador controlar o LED remotamente.

III. Detalhes de Implementação

---

1. Sistema Embebido (Arduino)

O código do Arduino foi programado para ler a porta analógica A0 onde está ligado o sensor. Foi criada uma conversão matemática para transformar o valor bruto do sensor (0 a 1023) numa percentagem mais fácil de interpretar (0% a 100%).

Foi implementada uma lógica de controlo: o LED acende automaticamente se a luz estiver abaixo de 30% ou acima de 60%. A comunicação é feita via porta Série (USB), enviando os valores de luz e ficando à escuta de comandos.

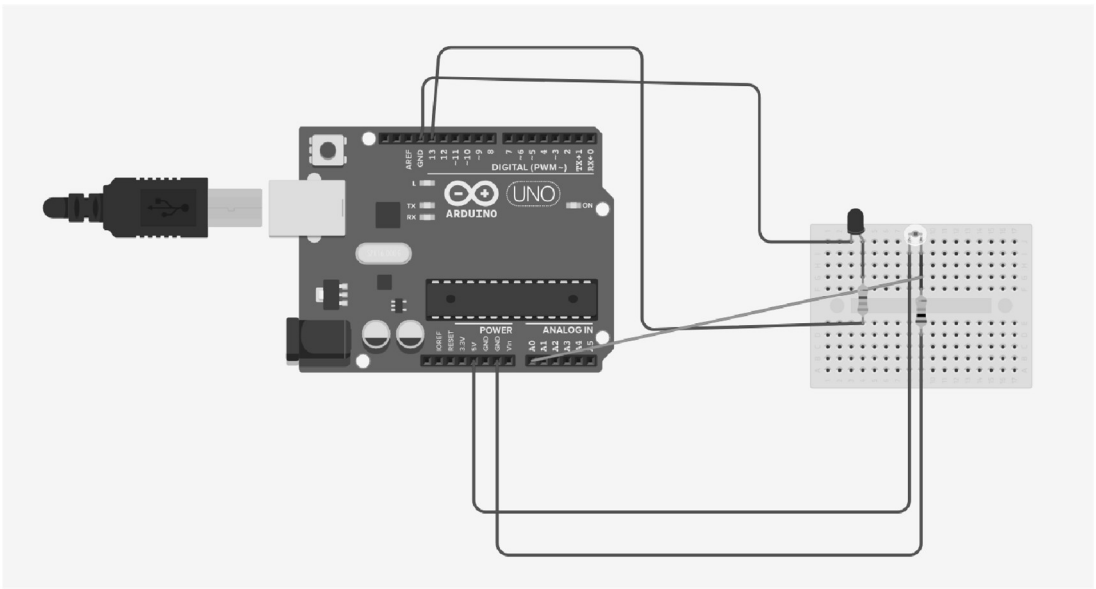

Fig. 1 - Esquema de ligação do circuito (Arduino, Sensor e LED).

2. Backend e Base de Dados

No lado do servidor (api/index.py), utilizou-se Python Flask e o projeto foi configurado para ser deployed na Vercel (através do ficheiro vercel.json).

Rota de Renderização: Foi implementada uma rota de raiz (/) que é responsável por fazer o render do template HTML (luz.html). Isto permite que, ao aceder ao link da API, o utilizador veja imediatamente a página visual em vez de apenas dados em bruto.

Foi criada uma ligação segura à base de dados PostgreSQL da escola. As funções da API (/luz e /led) geram a receção dos dados do sensor e os pedidos de controlo do LED.

3. Frontend (Interface Web)

A página luz.html foi desenhada para ser simples e funcional. Utiliza JavaScript (fetch) para comunicar com a API sem precisar de recarregar a página inteira, tornando a experiência mais fluida para o utilizador.

IV. Testes e Validação

O sistema foi testado em várias etapas para garantir o correto funcionamento:

- **Teste de Sensor:** Ao tapar o sensor com a mão, verificou-se que o valor no Arduino descia e o LED acendia automaticamente.
- **Teste de verificação de acesso:** Confirmou-se que a aplicação está acessível publicamente através do link da Vercel e que a página carrega corretamente.
- **Teste de Comunicação:** Os dados enviados foram confirmados na Base de Dados e aparecem na tabela do site.

V. Conclusão e Autoavaliação

Este projeto permitiu criar um sistema funcional que cumpre todos os requisitos definidos.

O que aprendi:

Compreendi melhor o conceito de API:, Código que permite que o Arduino e o Site comuniquem. Aprendi como fazer o deploy de uma aplicação Python na cloud usando o Vercel.

Desafios:

A parte mais complexa foi integrar as três peças (Arduino, Python e HTML) e garantir que a comunicação funcionava bem, especialmente após colocar o servidor online no vercel.

Futuro:

Como melhoria futura, o sistema poderia incluir gráficos visuais em vez de apenas uma tabela de texto.