

Assistente Virtual para Monitoramento de Consumo Energético

Descrição

O **Assistente Virtual para Monitoramento de Consumo Energético** é um sistema inteligente projetado para interagir com os usuários sobre o consumo de energia. Ele oferece dicas de economia de energia e gera relatórios detalhados sobre o uso de dispositivos conectados. Com integração IoT, permite feedback em tempo real sobre o consumo de energia de dispositivos monitorados.

Esse assistente utiliza **Inteligência Artificial** para fornecer recomendações e otimizar o consumo de energia, ajudando os usuários a reduzir custos e impactos ambientais.

Tecnologias

- **Backend:**
 - **.NET:** Para APIs e processamento de dados.
 - **Java:** Para processamento de IA e integração com o backend.
 - **Banco de Dados Relacional:** MySQL ou PostgreSQL para armazenar históricos de uso de energia.
- **Frontend:**
 - Aplicativo móvel com **integração de voz**.
 - **React Native** ou **Flutter** para o desenvolvimento do aplicativo.
- **IoT:**
 - **ESP32** ou outro microcontrolador com sensores de corrente.
 - **MQTT** para comunicação em tempo real entre dispositivos.
- **DevOps:**
 - **GitLab CI/CD**, **Jenkins** ou **GitHub Actions** para automação de deploy.
- **Segurança de Dados:**
 - Conformidade com a **LGPD** (Lei Geral de Proteção de Dados) e **GDPR**.

Funcionalidades

- **Monitoramento em Tempo Real:** Controle do consumo de energia de dispositivos conectados em tempo real.

- **Dicas de Economia:** O assistente fornece dicas personalizadas com base no uso de energia.
- **Relatórios de Consumo:** Geração de relatórios sobre o consumo de energia, com base no histórico dos dispositivos.
- **Integração de Voz:** O aplicativo móvel tem a capacidade de responder a comandos de voz para interagir com o assistente.
- **Feedback Imediato:** Receba alertas e notificações quando os dispositivos estiverem consumindo energia excessiva.

Requisitos

- **.NET Core:** Para o desenvolvimento do backend.
- **Java:** Para processamento de IA e controle de dispositivos.
- **Banco de Dados Relacional:** MySQL ou PostgreSQL.
- **Node.js:** Para desenvolvimento do aplicativo móvel.
- **ESP32:** Para integração com dispositivos IoT.
- **MQTT Broker:** Para comunicação entre dispositivos.

Como Rodar o Projeto

1. Configuração do Backend

.NET

1. Clone o repositório:

```
bash
git clone https://github.com/Bernas01/GS-IOT.git
```

2. Navegue até o diretório:

```
bash
cd assistente-energia
```

3. Restaure as dependências:

```
bash
dotnet restore
```

4. Execute a aplicação:

```
bash
dotnet run
```

2. Configuração do Frontend

Aplicativo Móvel

1. Clone o repositório do aplicativo:

```
bash
Copiar código
git clone https://github.com/Bernas01/GS-IOT.git
```

2. Instale as dependências:

```
bash
npm install
```

3. Inicie o aplicativo:

```
bash
npm run start
```

3. Configuração do MQTT

1. Caso não tenha um broker MQTT configurado, você pode usar um broker público como **HiveMQ**:
 - a. Endereço: `broker.hivemq.com`
 - b. Porta: `1883`
2. Se preferir, instale um broker MQTT local, como o **Mosquitto**.

4. Conectando o IoT (ESP32)

1. Faça o upload do código para o seu **ESP32** utilizando a IDE do Arduino ou PlatformIO.
2. Conecte o sensor de corrente no pino especificado (exemplo: **pino 34**).
3. Teste a conexão com o **MQTT Broker** e com o backend.