

## **A Proposal Project for a PROJETO INTEGRADOR 6:**

Gustavo Souto Silva de Barros Santos (gustavo.santos@novaroma.edu.br)

João Lucas Camilo (lucascamilo373@gmail.com)

Luiz Felipe Silva (luiz.silva1@novaroma.edu.br)

Supervisor Professor: MSC Claudio Pereira da Silva

(profclaudiosilva@gmail.com)

Faculdade Nova Roma

### **1 Project Description**

O presente projeto propõe o desenvolvimento de um dispositivo inteligente e vestível, concebido na forma de um suporte inteligente para bonés, o qual se tornará capaz de proceder com o monitoramento em tempo real dos níveis de intensidade proporcionados pela exposição à radiação ultravioleta (UV).

Nesse prumo, o objetivo deste dispositivo é fornecer alertas aos usuários - com auxílio de sinais vibratórios ou SMS - quando os níveis de radiação luminosa ou temperatura estiverem acima dos limites seguros de modo a poder prevenir riscos à saúde humana decorrente da superexposição solar.

A finalidade deste dispositivo é proporcionar aos usuários uma maior gama de incentivos quanto a práticas preventivas de saúde, onde seu uso estará voltado para atividades ao ar livre, esportes ou funcionar como EPI (Equipamento de Proteção Individual) para trabalhadores ao ar livre sob radiação solar.

O sistema se funda no uso de placa de microcontrolador ESP8266, a fim de dar conectividade Wi-Fi independente com capacidade de comunicação sem fio - de forma barata e eficiente - em conjunto com o sensor UV GUVB-S12SD para poder detectar os graus de incidência da radiação, através da leitura analógica dos níveis de intensidade, onde a placa de microcontrolador não só processará os dados coletados como também procederá tanto com disparo SMS quanto a ativação do Módulo de Vibração para alertar o usuário.

As aplicações do boné inteligente abrangem desde cuidados pessoais no dia a dia, práticas desportivas ao ar livre, até campanhas de conscientização sobre doenças dermatológicas, como o câncer de pele, de modo a reforçar o uso de tecnologias vestíveis como ferramentas de bem-estar e prevenção.

### **2 Project Scope**

O escopo deste presente projeto trata de abranger o desenvolvimento de um suporte inteligente para bonés com o fito de monitorar a exposição solar do usuário, por meio do uso de tecnologia IoT e sensores ambientais.

O sistema será composto por um dispositivo vestível apto a medir e processar

dados relacionados à radiação ultravioleta, fornecendo alertas locais ao usuário quando os limites de segurança forem excedidos, e transmitindo os dados para uma plataforma digital para análise e visualização.

O escopo contempla o desenvolvimento de hardware, bem como o de software embarcado, software de backend e a integração do dispositivo com uma plataforma digital para visualização e análise dos dados.

O desenvolvimento do hardware incluirá os seguintes componentes:

- **Sensores de Radiação UV:** O protótipo do suporte irá ser equipado com sensores, permitindo a medição da radiação UV, para proceder com o monitoramento contínuo das condições de exposição solar.
- **Microcontrolador:** O dispositivo fará uso do módulo ESP8266 Node MCU como unidade central de processamento e comunicação. Este será responsável por processar os dados capturados pelo sensor, gerenciar a comunicação Wi-Fi e controlar os módulos de alerta.
- **Mecanismos de Alertas:** Um módulo de vibração irá ser integrado para emitir alertas táteis ao usuário, de modo eficiente e discreto, acerca dos níveis de exposição elevados.
- **Fonte de Alimentação:** O sistema será alimentado por uma fonte de energia portátil (bateria LiPo), dando autonomia ao dispositivo, e assim sendo eficiente para garantir o funcionamento contínuo.

O desenvolvimento do software abrangerá as seguintes funcionalidades:

- **Aquisição e Processamento de Dados:** O firmware no ESP8266 irá coletar todos os dados do sensor GUVB-S12SD, bem como calculará o índice UV em tempo real e ainda irá determinar a necessidade de alerta local. Os dados processados (nível UV, classificação de risco) serão formatados em JSON (JavaScript Object Notation), um padrão leve e legível para troca de dados.
- **Comunicação IoT:** Todos os dados formatados em JSON irão ser transmitidos pelo ESP8266 via Wi-Fi utilizando o protocolo MQTT (Message Queuing Telemetry Transport). Este protocolo é padrão em aplicações IoT em razão de seu alto nível de eficiência, bem como baixo consumo de banda e modelo publish/subscribe, adequado para poder proceder com o envio de dados de sensores.
- **Serviço de Backend:** Um serviço de backend, que foi desenvolvido na linguagem Go (Golang), irá ser implementado para poder atuar como receptor das mensagens MQTT. A linguagem Go foi escolhida por causa de sua alta performance, eficiência em concorrência e,

também, adequação para serviços de rede. Este serviço irá ser o responsável por subscrever aos tópicos MQTT relevantes, receber os dados JSON, validá-los e persisti-los no banco de dados.

- **Armazenamento de Dados:** O sistema fará uso do InfluxDB para o armazenamento dos dados históricos de exposição solar. InfluxDB é um banco de dados de séries temporais otimizado para armazenar e consultar dados com timestamp, como leituras de sensores, para assim poder oferecer uma alta performance tanto de escrita quanto de leitura para este tipo de carga de trabalho.
- **Sistemas de Alertas Locais:** O firmware irá permitir a configuração de limiares de segurança para o índice ultravioleta e assim irá emitir notificações ao usuário via módulo de vibração quando estes limites forem ultrapassados.
- **Interface do Usuário e Visualização de Dados:** Uma plataforma web, utilizando Grafana, irá ser configurada para visualização dos dados. O Grafana se conectará ao InfluxDB para exibir dashboards interativos, permitindo ao usuário visualizar os dados de exposição solar (atuais e históricos), tendências e alertas registrados. Nesse sentido, a interface será projetada para ser intuitiva e acessível.
- **Integração da Plataforma Digital:** A arquitetura tratará de assegurar a integração fluida entre o dispositivo (ESP8266), o protocolo de comunicação (MQTT), o serviço de backend (Go), o banco de dados (InfluxDB) e a plataforma de visualização (Grafana), permitindo assim o monitoramento remoto e análise dos padrões de exposição.

O projeto ainda contemplará a execução de alguns testes em diferentes etapas de desenvolvimento para proceder com sua validação, inclusive:

- **Testes Físicos:** O protótipo será testado em condições reais de uso, com a verificação de seu desempenho na medição de radiação UV e temperatura, bem como precisão e efetividade dos alertas gerados.
- **Testes de Software:** Irão ser realizados testes unitários, bem como os de integração, para o firmware do ESP8266 e para o serviço de backend em Go. Testes de carga e performance serão aplicados à plataforma de dados (MQTT Broker, Go Service, InfluxDB) para poder validar sua escalabilidade e confiabilidade.
- **Testes de Usabilidade:** A interface do Grafana irá ser avaliada tanto à clareza e facilidade de uso quanto a utilidade das informações que forem apresentadas.

Destacar que, em relação às limitações e exclusões, o projeto não contemplará -

neste primeiro instante - as seguintes funcionalidades:

- **Análise Preditiva e Inteligência Artificial:** O sistema não irá conter recursos de inteligência artificial para fins de prognósticos analíticos, ou seja, para prever os padrões de exposição solar ou sugerir ações personalizadas com base no comportamento pretérito do usuário.
- **Integração com Sistemas de Saúde:** O dispositivo não se integrará a prontuários médicos eletrônicos ou sistemas clínicos.
- **Recursos Avançados de Conectividade:** As Funcionalidades para compartilhamento de dados do usuário em tempo real com terceiros (instituições, profissionais de saúde) não serão implementadas.

### 3 Project Team Members

A equipe de criação e desenvolvimento do projeto UVTrack é composta por 3 (três) membros, onde cada membro desempenhará funções estratégicas e complementares, assegurando, desta forma, uma execução eficiente, eficaz, coesa e tecnicamente sólida em todas as etapas do projeto:

- **Gustavo Souto Silva de Barros Santos (G)** - Graduando em Ciência da Computação pela Faculdade Nova Roma (FNR), é o responsável pela elicitación, modelagem e rastreabilidade dos requisitos funcionais (RFs) e não funcionais (RNFs), bem como pela produção e manutenção da documentação técnica (especificações, diagramas, relatórios). Atua na garantia do alinhamento entre escopo, requisitos e entregas técnicas. Adicionalmente, contribui na montagem física do protótipo, considerando critérios de ergonomia, segurança e usabilidade.
- **João Lucas Camilo (J)** – Graduando em Ciência da Computação pela Faculdade Nova Roma (FNR), é Encarregado pelo desenvolvimento do firmware embarcado (C/C++ no ESP8266), incluindo a lógica de aquisição de dados do sensor UV, processamento, formatação JSON, comunicação MQTT e acionamento de alertas. Responsável pela condução dos testes funcionais, de integração e de campo, calibração de sensores e otimização do sistema.
- **Luiz Felipe Silva (L)** – Responsável pela liderança técnica e arquitetura do sistema, incluindo a definição da arquitetura embarcada e da plataforma de software (MQTT, Go, InfluxDB, Grafana). Especifica componentes, valida a viabilidade de integração e supervisiona o desenvolvimento físico e lógico. Atua como elo entre os objetivos do projeto e as decisões de engenharia, garantindo a aderência aos requisitos e boas práticas.

## 4 Device Functionality and Design

Esta seção detalha a funcionalidade e o design do dispositivo da UVTrack, abrangendo a arquitetura de hardware e software, as tecnologias empregadas e os componentes específicos utilizados.

### 4.1 Modelagem Arquitetural

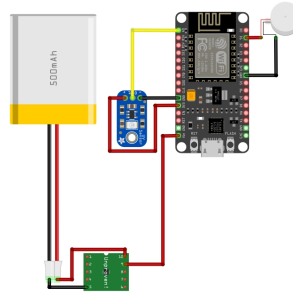
A modelagem arquitetural do UVTrack descreve a organização funcional dos componentes do sistema, evidenciando a interação entre o microcontrolador, sensores, atuadores e a plataforma de comunicação e dados. O objetivo é representar o fluxo de dados desde a captação (radiação UV), passando pelo processamento no microcontrolador (ESP8266), a geração de alertas locais e a transmissão de dados para o backend.

A arquitetura se estrutura nas camadas: sensoriamento, processamento embarcado, atuação local, comunicação e plataforma de dados/visualização.

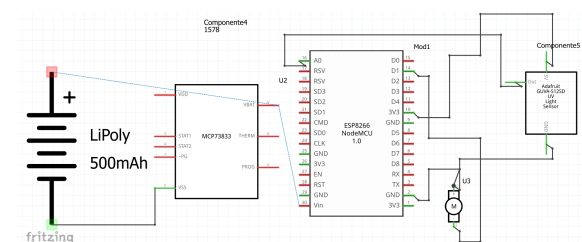
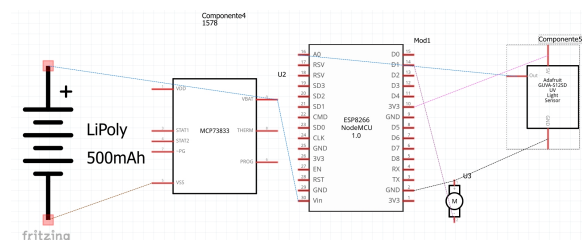
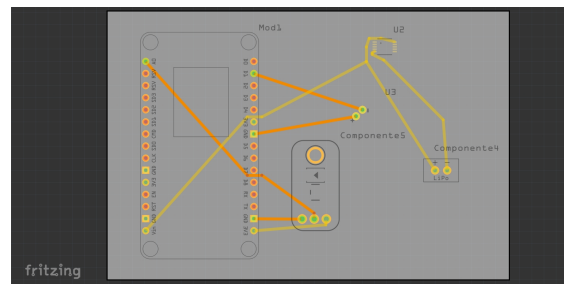
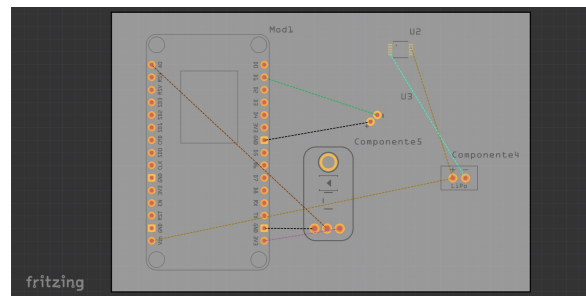
#### 4.1.1 Arquitetura do Circuito UVTrack

A implementação física do protótipo utiliza o módulo NodeMCU, baseado no System-on-Chip (SoC) ESP8266, como unidade central de processamento e conectividade. Este SoC, com CPU Xtensa LX106 de 32 bits e Wi-Fi integrado (IEEE 802.11 b/g/n), é ideal para aplicações IoT.

- **Sensoriamento:** O sensor de radiação UV GUVA-S12SD é conectado à entrada analógica (A0) do NodeMCU para medir os níveis dos raios UV.
- **Processamento:** O ESP8266 executa o firmware responsável por ler o sensor, calcular o índice UV, formatar os dados em JSON e gerenciar a comunicação MQTT.
- **Atuação:** Um módulo de vibração é conectado a um pino GPIO do NodeMCU para fornecer feedback tátil ao usuário.
- **Bateria:** Uma bateria de Polímero de Lítio (LiPo) de 3.7V, gerenciada por circuito de carga dedicado MCP73833, fornece energia ao sistema.
- **Interface:** A programação e depuração iniciais são feitas via interface UART/USB. Atualizações de firmware podem ser realizadas Over-The-Air (OTA) via Wi-Fi.



fritzing



#### 4.1.2 Arquitetura da Plataforma de Software

A plataforma de software complementa o hardware embarcado, permitindo a coleta, armazenamento e visualização dos dados de exposição UV:

- **Dispositivo (ESP8266):** Coleta dados do sensor UV GUA-S12SD, processa localmente, gera alertas vibratórios e publica dados (Índice UV,

classificação) em formato JSON via MQTT para um tópico específico.

- **Broker MQTT:** Um servidor MQTT (como Mosquitto) atua como intermediário, recebendo as mensagens publicadas pelo ESP8266 e entregando-as aos clientes subscritos.
- **Serviço Backend (GO):** Uma aplicação escrita em Go subscrive ao tópico MQTT uvtrack/data. Ao receber uma mensagem JSON, o serviço a valida, processa (se necessário) e a insere no banco de dados InfluxDB, associando um timestamp.
- **Banco de Dados (InfluxDB):** Armazena as séries temporais dos dados de UV recebidos, organizados por tags (ex: ID do dispositivo) e fields (ex: indice\_uv, classificacao).
- **Plataforma de Visualização (Grafana):** Conecta-se ao InfluxDB como fonte de dados. Dashboards configurados no Grafana permitem aos usuários visualizar gráficos do histórico de exposição UV, níveis atuais (com alguma latência), tabelas de dados e alertas, acessíveis via navegador web.

## 4.2 Principais Tecnologias

O protótipo da UVTrack trata de proceder com a integração de um conjunto de tecnologias chave para garantir funcionalidade, eficiência e usabilidade:

### 4.2.1 Sistemas inteligentes wearable (Computação Vestível)

O projeto trata de materializar o conceito de computação vestível ao integrar sensores, processamento e atuadores de forma discreta e miniaturizada em um acessório de uso pessoal (suporte para boné).

O design prioriza o conforto e a usabilidade, habilitando o monitoramento contínuo sem interferir significativamente na experiência do usuário.

### 4.2.2 Dispositivos Computacionais Autônomos (Sistemas Embarcados)

O núcleo do dispositivo inteligente é um sistema embarcado fundado no SoC ESP8266. Esta solução autônoma integra hardware e firmware otimizados para a tarefa específica de monitoramento UV, processamento em tempo real, geração de alertas locais e comunicação sem fio, caracterizando-se pelo baixo consumo energético e custo reduzido.

### 4.2.3 Codificação para Sistemas Integrados (Programação Embarcada)

O firmware do ESP8266 é desenvolvido em C/C++ utilizando o framework

Arduino ou o ESP-IDF. Esta abordagem permite controle de baixo nível do hardware, interação direta com periféricos (ADC, GPIO, Wi-Fi), otimização de performance e gerenciamento eficiente de energia. O código implementa a lógica de leitura do sensor, cálculo do índice UV, acionamento do módulo de vibração e comunicação via MQTT com payloads em formato JSON.

#### 4.2.4 Protocolo de Comunicação IoT (MQTT)

O Message Queuing Telemetry Transport (MQTT) é utilizado para a comunicação entre o dispositivo ESP8266 e o backend. Sua arquitetura publish/subscribe, leveza e eficiência o tornam ideal para ambientes com restrições de rede e energia, típicos de aplicações IoT. Garante a entrega desacoplada e assíncrona dos dados do sensor.

#### 4.2.4 Protocolo de Comunicação IoT (MQTT)

O Message Queuing Telemetry Transport (MQTT) é utilizado para a comunicação entre o dispositivo ESP8266 e o backend. Sua arquitetura publish/subscribe, leveza e eficiência o tornam ideal para ambientes com restrições de rede e energia, típicos de aplicações IoT. Garante a entrega desacoplada e assíncrona dos dados do sensor.

#### 4.2.5 Formato de Serialização de Dados (JSON)

O JavaScript Object Notation (JSON) é o formato escolhido para estruturar os dados enviados pelo ESP8266 via MQTT. Sua simplicidade, legibilidade humana e facilidade de parsing por diversas linguagens de programação (incluindo Go no backend) o tornam um padrão de fato para APIs e troca de dados na web e em IoT.

#### 4.2.6 Tecnologia de Backend (Go - Golang)

A linguagem Go é empregada no desenvolvimento do serviço de backend responsável por receber e processar os dados MQTT. Suas características, como tipagem estática, compilação nativa, excelente suporte à concorrência (goroutines, channels) e bibliotecas padrão robustas para rede, a tornam altamente adequada para construir serviços performáticos, escaláveis e confiáveis para ingestão de dados IoT.

#### 4.2.7 Banco de Dados de Séries Temporais (InfluxDB)

Para a persistência dos dados de monitoramento UV, utiliza-se o InfluxDB. Como um banco de dados especificamente projetado para séries temporais, ele oferece alta performance de escrita e consulta para dados com timestamp, compressão eficiente e funcionalidades nativas para análise temporal (agregações, downsampling), sendo ideal para armazenar métricas e eventos de sensores IoT.



#### 4.2.8 Plataforma de Visualização de Dados (Grafana)

Grafana é a ferramenta escolhida para a criação de dashboards interativos e visualização dos dados armazenados no InfluxDB. Sua flexibilidade na criação de gráficos, tabelas e alertas, suporte a diversas fontes de dados (incluindo InfluxDB) e interface web amigável permitem aos usuários monitorar facilmente os níveis de exposição UV históricos e recentes.

#### 4.3 Componentes Eletrônicos

Esta seção trata de detalhar todos os componentes eletrônicos específicos selecionados para a implementação do protótipo UVTrack.

##### 4.3.1 Microcontrolador: **ESP8266 NodeMCU**

**O módulo NodeMCU v1.0 foi escolhido como cérebro do dispositivo. Ele integra o SoC ESP8266, que oferece:**

- Processador Xtensa LX106 de 32 bits.
- Memória Flash (tipicamente 4MB).
- Conectividade Wi-Fi 802.11 b/g/n integrada.
- Pinos GPIO (General Purpose Input/Output).
- Conversor Analógico-Digital (ADC) de 10 bits (1 pino utilizável).
- Interfaces de comunicação (UART, SPI, I2C).
- Regulador de tensão e interface USB (programação ou alimentação). sua popularidade e baixo custo, bem como sua ampla documentação e suporte da comunidade facilitam o desenvolvimento.

##### 4.3.2 Sensor de Radiação UV: **GUVA-S12SD**

Este sensor é projetado para detectar a intensidade da luz ultravioleta (UV), especificamente na faixa UVA e UVB. Características:

- Fotodiodo sensível a UV.
- Saída analógica de tensão proporcional à intensidade UV (0-1V para 0-10 mW/cm<sup>2</sup> já que é seu típico).
- Faixa de operação de tensão compatível com o ESP8266 (3.3V).

- Permite que seja feito o cálculo do Índice UV (requer calibração e fórmula de conversão apropriada).
- Conversor Analógico-Digital (ADC) de 10 bits (1 pino utilizável).
- Interfaces de comunicação (UART, SPI, I2C).
- Regulador de tensão e interface USB (programação ou alimentação). sua popularidade e baixo custo, bem como sua ampla documentação e suporte da comunidade facilitam o desenvolvimento.

#### 4.3.3 Atuador: Módulo de Vibração

Um pequeno motor de vibração (tipo “pancake” ou cilíndrico) é utilizado para fornecer feedback tátil. Características:

- Operação em baixa tensão (compatível com saídas de 3.3V do ESP8266, mas geralmente requer um transistor driver).
- Consumo de corrente moderado (necessita de um transistor NPN ou MOSFET para acionamento seguro a partir de um pino GPIO, evitando sobrecarga).
- Fornece um alerta discreto e eficaz diretamente ao usuário.

#### 4.3.4 Gerenciamento de Energia: Bateria LiPo e Carregador

O sistema é alimentado por uma bateria de Polímero de Lítio (LiPo) de célula única (3.7V nominal). Características:

- **Bateria LiPo:** Alta densidade energética, leveza e formato flexível. Capacidade (ex: 500mAh) a ser definida com base nos requisitos de autonomia.
- **Módulo Carregador:** Um módulo dedicado baseado em ICs como TP4056 ou MCP73831/3 é essencial para carregar a bateria LiPo de forma segura via USB e assim fornecer proteção contra sobrecarga, descarga excessiva e curto-circuito.
- **Regulador de Tensão:** Pode ser necessário um regulador LDO (Low-Dropout) para fornecer 3.3V estáveis ao NodeMCU e ao sensor a partir da tensão variável da bateria (tipicamente 3.0V a 4.2V), embora os módulos NodeMCU já incluam um regulador on-board.

#### 4.3.5 Conectividade e Fiação

- **Módulo Wi-Fi (Integrado do ESP8266):** A conectividade sem fio

principal do sistema é estabelecida por meio do módulo Wi-Fi integrado no microcontrolador ESP8266. Isso permite que o dispositivo se conecte a redes locais (Wi-Fi) para enviar dados de telemetria (níveis de UV e temperatura ambiente) para um broker MQTT e, depois, para serviços como o InfluxDB e o Grafana.

- **Jumpers Macho e Fêmea:** Trata-se de cabos flexíveis especialmente feitos para poder estabelecer conexões seguras entre os módulos do circuito de modo a permitir que haja a transmissão eficiente de sinais elétricos. Isso quer dizer que eles são essenciais para a montagem ou modularidade do sistema, pois facilita e agiliza a substituição e a manutenção dos componentes. As cores dos fios utilizados seguem a convenção de padrão de boas práticas, quais sejam:
  - Vermelho – Alimentação (VCC/5V);
  - Preto – Terra (GND);
  - Azul - Sinais de entrada/saída.

#### 4.4 Arquitetura de Software e Plataforma de Dados:

Conforme introduzido na seção 4.1.2, a arquitetura de software é fundamental para coletar, transmitir, armazenar e visualizar os dados de exposição UV. Esta seção detalha o fluxo e os componentes dessa arquitetura.

##### 4.4.1 Fluxo de Dados

O fluxo de dados ponta a ponta opera da seguinte maneira:

1. **Coleta (ESP8266):** O firmware lê a tensão analógica do sensor GUVVA-S12SD.
2. **Processamento Local (ESP8266):** A tensão é convertida para um valor de Índice UV estimado. O firmware verifica se o índice ultrapassa limiares pré-definidos.
3. **Alerta Local (ESP8266):** Se um limiar for excedido, o módulo de vibração é ativado.
4. **Formatação (ESP8266):** Os dados relevantes (Índice UV, classificação de risco, timestamp ou ID do dispositivo) são formatados em uma string JSON.
5. **Publicação (ESP8266):** A string JSON é publicada em um tópico MQTT específico (ex: uvtrack/device/data) via Wi-Fi, utilizando a biblioteca AsyncMqttClient.
6. **Transmissão (Rede):** A mensagem MQTT trafega pela rede Wi-Fi

até o Broker MQTT.

7. **Recepção (Broker MQTT):** O Broker recebe a mensagem e a encaminha para todos os clientes subscritos ao tópico.
8. **Subscrição e Ingestão (Serviço Go):** O serviço backend, subscrito ao tópico, recebe a mensagem JSON.
9. **Processamento Backend (Serviço Go):** O serviço parseia o JSON, valida os dados e os prepara para inserção no banco de dados.
10. **Persistência (InfluxDB):** O serviço Go escreve os dados (Índice UV, classificação) como um novo ponto na measurement apropriada no InfluxDB, associando o timestamp atual ou o fornecido pelo dispositivo.
11. **Consulta (Grafana):** A plataforma Grafana, configurada com o InfluxDB como datasource, consulta periodicamente ou sob demanda os dados armazenados.
12. **Visualização (Grafana):** Os dados são apresentados ao usuário final em dashboards web interativos, exibindo gráficos de séries temporais, medidores, tabelas, etc.

#### 4.4.2 Componentes da Plataforma

O fluxo de dados ponta a ponta opera da seguinte maneira:

- **Firmware ESP8266 (C++/Arduino):** Trata-se do responsável pela lógica embarcada, leitura de sensor, alertas locais e comunicação MQTT/JSON.
- **Broker MQTT (Mosquitto):** Software intermediário que gerencia a publicação e subscrição de mensagens MQTT. Pode ser hospedado localmente, em nuvem ou utilizar um serviço gerenciado.
- **Serviço Backend (Go):** Aplicação customizada que atua como ponte entre o MQTT e o InfluxDB. Processa, valida e armazena os dados. Geralmente executado como um serviço contínuo.
- **Banco de Dados (InfluxDB):** Sistema de gerenciamento de banco de dados otimizado para séries temporais, responsável pela persistência eficiente dos dados.
- **Plataforma de Visualização (Grafana):** Ferramenta web para criar e exibir dashboards a partir dos dados do InfluxDB.

#### 4.4.3 Orquestração (Docker)

Para ter um maior nível de facilidade quanto à implantação e gerenciamento dos componentes de backend (Broker MQTT, Serviço Go, InfluxDB, Grafana), pode-se utilizar Docker e Docker Compose.

Um arquivo docker-compose.yml, ora situado no repositório sensoruv, define os serviços, suas imagens, dependências, volumes e configurações de rede, permitindo iniciar toda a pilha de backend com um único comando.

#### 4.5 Mechanical Design (Design Mecânico)

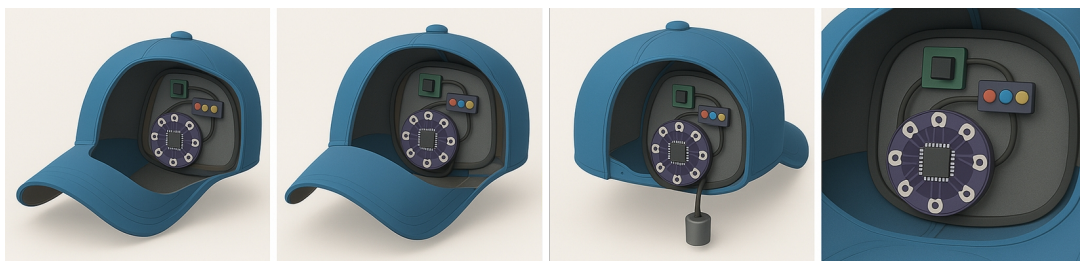
O design mecânico do dispositivo de suporte inteligente de radiação UV para boné projetado e desenvolvido com enfoque na ergonomia e usabilidade, bem como na integração eficiente dos componentes eletrônicos.

A disposição do sensor GUVA-S12SD foi planejada para garantir exposição ideal à luz solar, sem quaisquer obstruções, enquanto o módulo de vibração foi posicionado estrategicamente para poder proporcionar feedback tátil claro e imediato ao usuário.

Todos os módulos (NodeMCU, sensor, módulo de vibração, bateria, circuito de carga) são fixados de maneira a não comprometer o conforto ergonômico nem a mobilidade, utilizando um suporte leve e discreto integrado ao boné.

O encapsulamento dos circuitos eletrônicos tem por finalidade proporcionar uma proteção aos componentes contra impactos suaves e condições do meio ambiente (umidade, poeira), respeitando a flexibilidade do tecido. Além disso, o suporte tem um regulador traseiro e isso permite que o usuário possa fazer o ajuste de tamanho de modo a estar compatível para si.

Modelos tridimensionais podem ser utilizados para validar o posicionamento e dimensionamento dos elementos embarcados, garantindo que o sistema se mantenha funcional e esteticamente integrado à peça de vestuário.



#### 4.6 Safety and Security Items (Itens de Segurança e Confiabilidade)

Esta seção trata de descrever os mecanismos implementados no protótipo da

UVTrack para poder garantir tanto a segurança operacional do sistema quanto a confiabilidade das informações coletadas e transmitidas.

Considerando a operação contínua em ambientes externos e a natureza dos alertas relacionados à saúde, foram adotadas medidas para poder assegurar o funcionamento preciso do sensor, a integridade dos dados, a segurança da comunicação e a proteção dos componentes.

As práticas incluem verificações de inicialização, estratégias de economia de energia, bem como proteção contra variações elétricas e monitoramento da conectividade. Adicionalmente, o sistema pode oferecer feedback a respeito de seu estado operacional e assim permitir controle sobre todos os alertas, garantindo, portanto, uma operação estável e confiável.

#### 4.6.1 Autoteste na Inicialização

Na inicialização, o firmware do ESP8266 pode executar verificações básicas para confirmar a comunicação com o sensor GUVVA-S12SD (ex: leitura inicial dentro de uma faixa esperada) e o estado da conexão Wi-Fi. Falhas podem ser indicadas localmente (ex: padrão de vibração específico) ou reportadas via MQTT, se a conexão for estabelecida.

#### 4.6.2 Calibração Periódica dos Sensores

A precisão da leitura do sensor UV GUVVA-S12SD depende bastante de sua correta calibração e da fórmula de conversão para Índice UV utilizada no firmware. Validações periódicas comparando as leituras do dispositivo com instrumentos de referência podem ser necessárias para ajustar a fórmula ou detectar degradação do sensor.

#### 4.6.3 Otimização do Consumo Energético

O firmware trata de implementar estratégias para poder reduzir o consumo de energia, como colocar o ESP8266 em modos de baixo consumo (*deep sleep*) entre as leituras e transmissões. A frequência de leitura e transmissão é um parâmetro configurável para balancear a granularidade dos dados com a autonomia da bateria. O módulo de vibração só ativa quando necessário.

#### 4.6.4 Proteção Elétrica contra Picos de Tensão

O circuito de gerenciamento de carga da bateria LiPo (MCP73833) oferece proteções essenciais. O uso de reguladores de tensão adequados garante alimentação estável para os componentes sensíveis (ESP8266, sensor).

#### 4.6.5 Controle de Alertas

O firmware implementa a lógica para ativar o módulo de vibração com base

nos limiares de Índice UV configurados. Pode haver a possibilidade de configurar esses limiares ou desativar temporariamente os alertas através da plataforma Grafana (enviando um comando via MQTT para o dispositivo).

#### 4.6.6 Condições Precisas com Margem de Erro Controlada

O firmware trata de proceder com o gerenciamento da conexão Wi-Fi e MQTT, implementando lógicas de reconexão automática em caso de falha. O uso de QoS (Quality of Service) no MQTT pode ser configurado para garantir diferentes níveis de entrega das mensagens.

#### 4.6.7 Segurança da Plataforma

A segurança da plataforma de backend envolve proteger o Broker MQTT (ex: autenticação de clientes) e o serviço Go, bem como o acesso ao banco de dados do InfluxDB e a plataforma de dashboard do Grafana (ex: senhas fortes, firewalls, TLS/SSL para comunicação criptografada).

#### 4.6.8 Testes Funcionais e de Campo

Testes rigorosos são essenciais. Isso inclui testes unitários e de integração para o firmware e o backend, testes de comunicação ponta-a-ponta, testes de autonomia da bateria e testes de campo em condições reais de uso para validar a precisão das leituras, bem como o nível de eficácia dos alertas e a robustez geral do sistema desenvolvido.

### 4.7 Diagramas

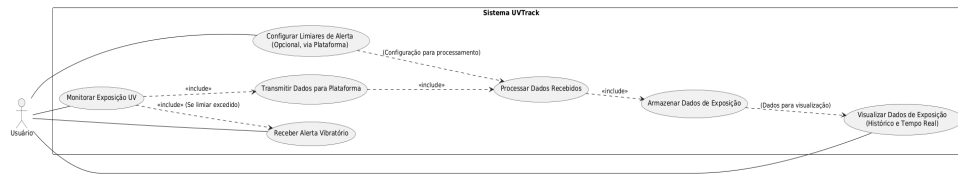
Os diagramas desenvolvidos para o projeto do suporte inteligente da UVTrack visam representar claramente a estrutura, o comportamento e as interações do sistema, tanto do hardware quanto do software.

Eles são ferramentas essenciais para o design, documentação, validação e comunicação deste presente projeto.

#### 4.7.1 Diagrama de Caso de Uso

Ilustra as principais interações entre os atores (Usuário, Sistema UVTrack) e as funcionalidades oferecidas. Casos de uso incluem: "Monitorar Exposição UV", "Receber Alerta Vibratório", "Visualizar Histórico de Exposição (via Grafana)", "Configurar Limiares (via Grafana)".

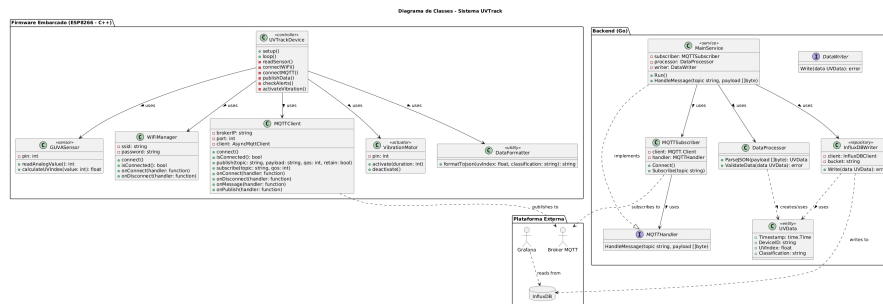
Este diagrama de caso de uso trata de ajudar a definir o escopo funcional e os requisitos do sistema do ponto de vista do usuário.



#### 4.7.2 Diagrama de Classes

Representa a estrutura lógica do software, tanto embarcado quanto backend. Para o firmware (C++), pode incluir classes para o Sensor UV, Cliente MQTT, Gerenciador de Wi-Fi, Controle de Vibração.

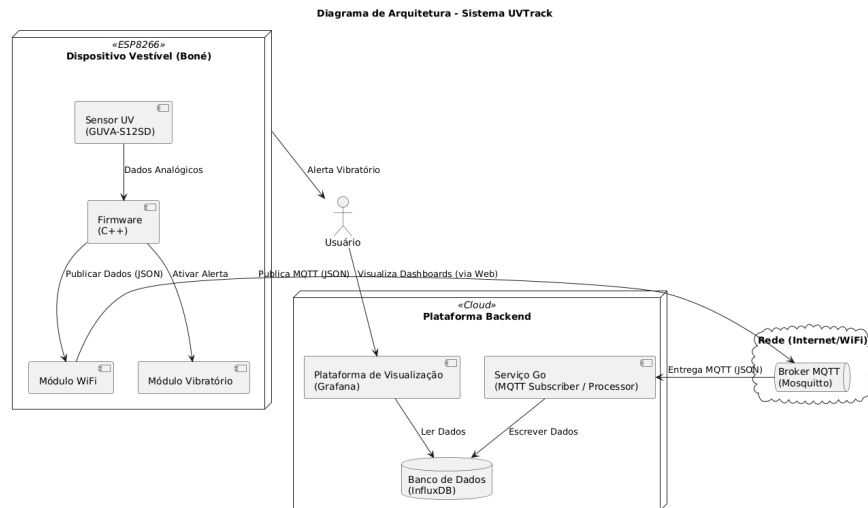
Para o backend (Go), pode incluir estruturas (structs) para poder representar os dados recebidos, clientes MQTT e InfluxDB, e pacotes para diferentes funcionalidades (ex: mqtt\_handler, influx\_writer, api). Isso ajuda a organizar o código e entender bem as dependências.



#### 4.7.3 Diagrama de Arquitetura (Hardware e Software)

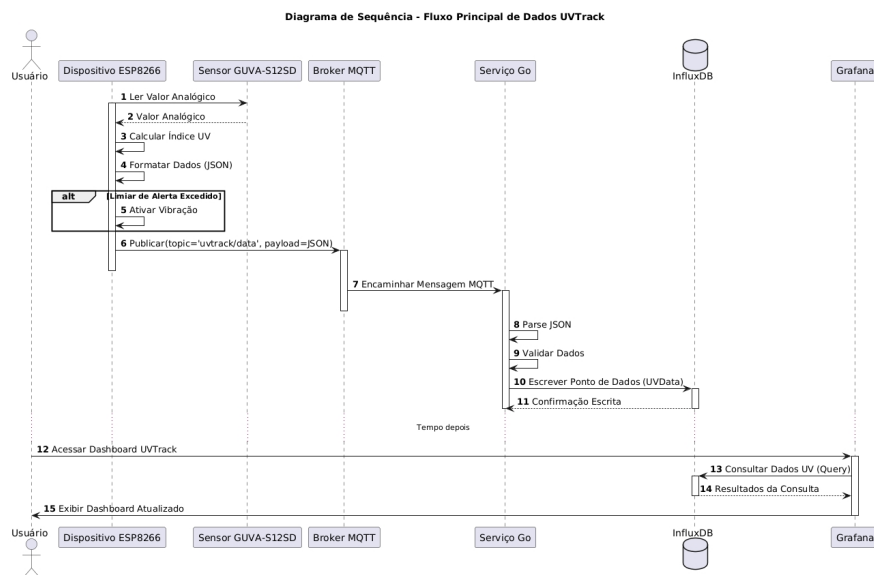
Conforme p que fora descrito e recomendado nas seções 4.1.1 e 4.1.2 deste presente documento, há diagramas visuais que mostram a interconexão dos componentes de hardware e o fluxo de dados através dos componentes de software (ESP8266, MQTT Broker, Go Service, InfluxDB, Grafana) são cruciais para a compreensão global do sistema.





#### 4.7.3 Diagrama de Sequência

Pode ser utilizado para poder detalhar interações específicas ao longo do tempo, como o fluxo de publicação de uma mensagem MQTT desde o ESP8266 até o armazenamento no bando de dados do InfluxDB, ou até mesmo o processo de consulta e exibição de dados pelo Grafana.



#### 4.7.4 Diagrama de Gantt

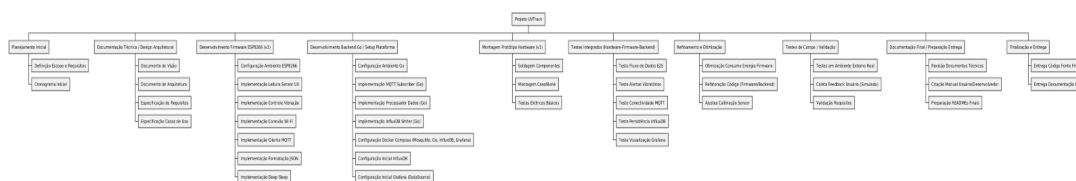
O diagrama apresentado sintetiza o cronograma do projeto UVTrack, um sistema embarcado para monitoramento de radiação ultravioleta, estruturado em dez etapas sequenciais. Inicia-se com o planejamento e definição do escopo, seguido pela elaboração da documentação técnica, incluindo visão do sistema, arquitetura e requisitos.

O desenvolvimento do firmware para a placa de microcontrolador ESP8266 contempla leitura de sensores UV, controle de vibração, comunicação via Wi-Fi

e MQTT, além de otimizações energéticas. Em paralelo, desenvolve-se o backend em Go, com integração a InfluxDB, Grafana e orquestração via Docker. Em seguida, a montagem do protótipo e testes elétricos são feitos.

A fase de testes integrados garante a comunicação fim a fim entre hardware, firmware e backend. Com o sistema funcional, procede-se à etapa de refinamento, com ajustes de consumo, código e calibração. Os testes de campo validam o desempenho em ambiente real e coletam os feedback.

Finda-se, portanto, com a documentação técnica e entrega oficial do projeto, incluindo código-fonte e materiais de apoio. Trata-se de uma abordagem integrada e iterativa, orientada a sistemas ciberfísicos e aplicações IoT.



## 5 Project Schedule

O cronograma de execução do projeto de suporte inteligente da UVTrack foi estruturado para poder abranger, de forma sequencial e integrada, todas as etapas necessárias ao desenvolvimento do dispositivo vestível e da plataforma de dados associada.

Resumo do Cronograma:

Fase	Descrição Resumida	Semanas
<b>Fase 1 – Elaboração e Planejamento Inicial</b>	Definição de objetivos, escopo, requisitos, arquitetura e plano de trabalho	Semana 1
<b>Fase 2 – Desenvolvimento da Documentação Técnica</b>	Especificações técnicas, diagramas, banco de dados, APIs via MQTT/JSON	Semanas 2 e 3
<b>Fase 3 – Desenvolvimento do Firmware e Backend</b>	Código para ESP8266 e serviço em Go, InfluxDB, Grafana, integração com Docker	Semanas 4 a 6
<b>Fase 4 – Implementação e Montagem do Protótipo</b>	Aquisição dos componentes, montagem eletrônica e integração mecânica	Semanas 7 a 11
<b>Fase 5 – Testes e Validações</b>	Testes de bancada e campo, verificação de funcionalidades e robustez	Semana 12
<b>Fase 6 – Entrega Final e Apresentação</b>	Relatório final, apresentação, entrega do protótipo funcional e documentação completa	Semana 13

### 5.1 Fase 1 - Elaboração e Planejamento Inicial

Definição de objetivos, escopo, requisitos, arquitetura inicial (hardware e software), tecnologias e plano de trabalho.

## 5.2 Fase 2 - Desenvolvimento da Documentação Técnica

Elaboração das especificações detalhadas, diagramas de arquitetura, design de banco de dados, APIs (implícitas via MQTT/JSON).

## 5.3 Fase 3 - Desenvolvimento do Firmware e Backend

Implementação do código C++ para o ESP8266 (leitura de sensor, MQTT, JSON, alertas) e do serviço Go (recepção MQTT, escrita no InfluxDB). Configuração inicial do InfluxDB e Grafana (possivelmente com Docker).

## 5.4 Fase 4 - Implementação e Montagem do Protótipo Físico

Aquisição de componentes, montagem do circuito eletrônico e integração mecânica no suporte do boné.

## 5.5 Fase 5 - Testes e Validações

Testes de bancada e de campo para verificar a funcionalidade completa do sistema: coleta de dados, transmissão via MQTT, persistência no bando de dados InfluxDB, visualização no dashboard do Grafana, alertas vibratórios, autonomia da bateria e robustez geral.

## 5.6 Fase 6 - Entrega Final e Apresentação

Consolidação dos resultados, elaboração do relatório técnico final, preparação da apresentação e entrega do protótipo funcional e documentação associada.

# 6 Budget

A estimativa de custos para os componentes eletrônicos do protótipo UVTrack é essencial para o planejamento financeiro do projeto.

## 6.1 Tabela de Valores

Componente	Qtd.	PUO (R\$)	Fabricante	Fornecedor	Descrição	CT (R\$)
NodeMCU ESP8266	1	30,00	Espressif	M. Livre	Placa	30,00
Sensor GUVAS12SD	1	20,00	Keyes	Aliexpress	Sensor	20,00
Módulo de Vibração	1	8,40	—	Nefeltech	Sensor	8,40
Bateria LiPo 3.7V LC802530	1	39,90	Rontek	M. Livre	Bateria	39,90

Módulo de Carregamento MCP73833	1	18,99	—	Shopee	Carregamento	18,99
Cabos e Jumpers	1 kit	11,50	—	Nefeltech	Fios	11,50
Suspensório Turtle	1	47,90	Steelfex	F. Costa	Base Suporte	47,90
<b>Total Geral (CTG)</b>	—	—	—	—	—	<b>176,69</b>
<b>Legenda</b>	<b>Qtd.</b> = Quantidade / <b>PUO</b> = Preço Unitário Orçado / <b>CT</b> = Custo Total / <b>CTG</b> = Custo Total Geral					

## 7 Appendix

### 7.1 Links importantes

- Repositório GitHub – UVTrack ([Click Aqui!](#))
- Repósitoio GitHub – sensoruv ([Click Aqui!](#))
- Checklist para Revisão de Especificação de Requisitos ([Click Aqui!](#))
- Vídeos de Apresentação do Protótipo via Drive: ([01](#) | [02](#) | [03](#))