

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ**

**BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

FLOOD PREVENTION – software para PREVENÇÃO DE CATÁSTROFES AMBIENTAIS com ênfase em inundações baseado na detecção e resposta

**Curitiba**

**2024**

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ**

**BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

FLOOD PREVENTION – software para PREVENÇÃO DE CATÁSTROFES AMBIENTAIS com ênfase em inundações baseado na detecção e resposta

**Grupo de trabalho:**

Anabelly

João Vitor de Lima Antunes

Josiel da Silva Queiroz Júnior

Mateus Alves

Matheus

**Curitiba**

**2024**

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ**

**BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO**

FLOOD PREVENTION – software para PREVENÇÃO DE CATÁSTROFES AMBIENTAIS com ênfase em inundações baseado na detecção e resposta

Relatório Acadêmico apresentado ao curso de bacharelado em Ciência da Computação, como parte dos requisitos necessários à obtenção de nota na disciplina PCO127A - Resolução de Problemas de Natureza Discreta.

Orientadores: Prof. Andrey Cabral Meira e Prof. Andrey Cabral Meira

**Curitiba**

**2024**

**SUMÁRIO**

[1. INTRODUÇÃO 4](#_Toc169015826)

[2. FLOOD PREVENTION: QUEM SOMOS? 4](#_Toc169015827)

[2.1 Atividade 1 – Nome da atividade 4](#_Toc169015828)

[2.2 Atividade 2 – Nome da atividade 4](#_Toc169015829)

[2.x Atividade x – Nome da atividade 4](#_Toc169015830)

[3. PJBL FASE 1 - PROTÓTIPO DO SISTEMA EMBARCADO (RA1) 4](#_Toc169015831)

[3.1 Sensores 4](#_Toc169015832)

[3.1.1 Sensor DHT22 4](#_Toc169015833)

[3.1.2 Sensor HC - SR04 5](#_Toc169015834)

[3.2 Atuadores 5](#_Toc169015835)

[3.2.1 Atuador LED *(Light Emitting Diode)* 5](#_Toc169015836)

[3.2.2 Atuador *Piezzo Buzzer* 5](#_Toc169015837)

[3.3 Comunicação em tópicos 5](#_Toc169015838)

[3.3.1 Temperatura 5](#_Toc169015839)

[3.3.2 Umidade 5](#_Toc169015840)

[3.3.3 Distância 5](#_Toc169015841)

[3.3.4 Alerta 6](#_Toc169015842)

[3.3.5 Botão 6](#_Toc169015843)

[3.3.6 Mensagem 6](#_Toc169015844)

[4. PJBL FASE 2 - PROTÓTIPO DE APLICAÇÃO WEB (RA2) 6](#_Toc169015845)

[5. RA – 03: BANCO DE DADOS 6](#_Toc169015846)

[6. RA – 04: INTEGRAÇÃO 6](#_Toc169015847)

[7. RELATO DE EXPERIÊNCIA 6](#_Toc169015848)

[8. REFERÊNCIAS 7](#_Toc169015849)

# 1. INTRODUÇÃO

Deve contar uma breve introdução sobre o que se trata o portfólio.

# 2. FLOOD PREVENTION: QUEM SOMOS?

Nesta seção estará presente sobre a empresa.

## 2.1 Atividade 1 – Nome da atividade

Em cada uma dessas subseções deve ter um comprovante que a atividade foi feita. Pode ser anexado imagens, códigos, relatos pessoais de entendimento. O professor pode chamar para conversar e realizar perguntas sobre o que foi desenvolvido, para confirmar se o aluno de fato entendeu/realizou ou não. Caso não saiba responder, poderá impactar na nota.

## 2.2 Atividade 2 – Nome da atividade

## 2.x Atividade x – Nome da atividade

# 3. PJBL FASE 1 - PROTÓTIPO DO SISTEMA EMBARCADO (RA1)

O hardware do sistema de prevenção foi desenvolvido através do simulador online Wokwi, composto por dois sensores (DHT22 e HC - SR04) e dois atuadores (LED e *Piezzo Buzzer*). O software desenvolvido nesse módulo pode ser acessado a partir do link abaixo:

*https:/LinkDoRepositorioDaPastaDoRA1*

## 3.1 Sensores

### 3.1.1 Sensor DHT22

Trata-se de um sensor de temperatura e umidade, o qual permite realizar leituras de temperaturas entre -40 e 80 graus Celsius e umidade entre 0% e 100%. Na especificidade do projeto, estes dados são utilizados para realizar a predição das condições climáticas a fim de, junto a outras variáveis, verificar a possibilidade de enchentes.

### 3.1.2 Sensor HC - SR04

Trata-se de um sensor ultrassônico, dispositivo que emite uma onda de alta frequência e mede o tempo que essas ondas levam para elas serem refletidas em um objeto. Na especificidade do projeto, os dados colhidos por este sensor são utilizados para medir o nível de água do rio, através de suas ondas ultrassônicas.

## 3.2 Atuadores

### 3.2.1 Atuador LED *(Light Emitting Diode)*

É um componente que converte eletricidade em luz. Nesse contexto, é utilizado como um sinal visual de presença de risco na região.

### 3.2.2 Atuador *Piezzo Buzzer*

Componente de geração de ruído sonoro a partir da excitação elétrica. Nesse contexto, esse componente serve como um aviso sonoro caso haja risco de enchente, previsto pelos dados colhidos pelos sensores.

## 3.3 Comunicação em tópicos

O código desenvolvido em C comunica-se com outro código em Python através de uma requisição MQTT *(Message Queuing Telemetry Transport).* Para a realização da comunicação, houve a utilização de alguns tópicos, os quais serão descritos nas subseções a seguir.

**Figura 1: tópicos para comunicação MQTT.**

A screenshot of a computer

Description automatically generated

### 3.3.1 Temperatura

Este tópico é referente ao dado de temperatura recolhido pelo sensor DHT22, o valor recolhido é armazenado na variável **myTopicTemperatura** e em seguida é atribuído ao tópico nomeado como **/Temperatura**, então é publicado.

### 3.3.2 Umidade

Este tópico é referente ao dado de umidade recolhido também pelo sensor DHT22, o valor recolhido é armazenado na variável **myTopicUmidade** e em seguida é atribuído ao tópico nomeado como **/Umidade**, então é publicado no *Broker*.

### 3.3.3 Distância

Este tópico é referente ao dado de Distância recolhido pelo sensor HC-SR04. O valor recolhido é armazenado na variável **myTopicDistancia**, em seguida é atribuído ao tópico nomeado como **/Distancia**, então é publicado.

### 3.3.4 Alerta

Este tópico refere-se a variável **myTopicAction**, o qual é responsável por portar os dados recebidos pela conexão MQTT que será objeto de comparadores a fim de determinar uma ação no circuito. Em específico, o alerta aciona ou desativa os dispositivos de saída do circuito através do tópico **/Action/alerta**.

### 3.3.5 Botão

Este tópico refere-se a **myTopicButton**. Definido como **/Botao/alerta** representa a resposta devolvida pelo dashboard *(Node-red)* através da interação com um componente que abre ou fecha o fluxo de corrente em um circuito, o botão. O tópico preserva o estado do botão e este estado é publicado para o servidor.

### 3.3.6 Mensagem

Este tópico refere-se a variável **myTopicMensagem**. Tal tópico representa a mensagem que será enviada a depender dos dados colhidos pelos sensores, após passarem por um filtro condicional, dessa forma o resultado que é atribuído ao tópico varia e ao fim a função no qual o tópico está inserido é responsável por publicar uma mensagem ao servidor baseada no corpo do tópico **/Mensagem/alerta**.

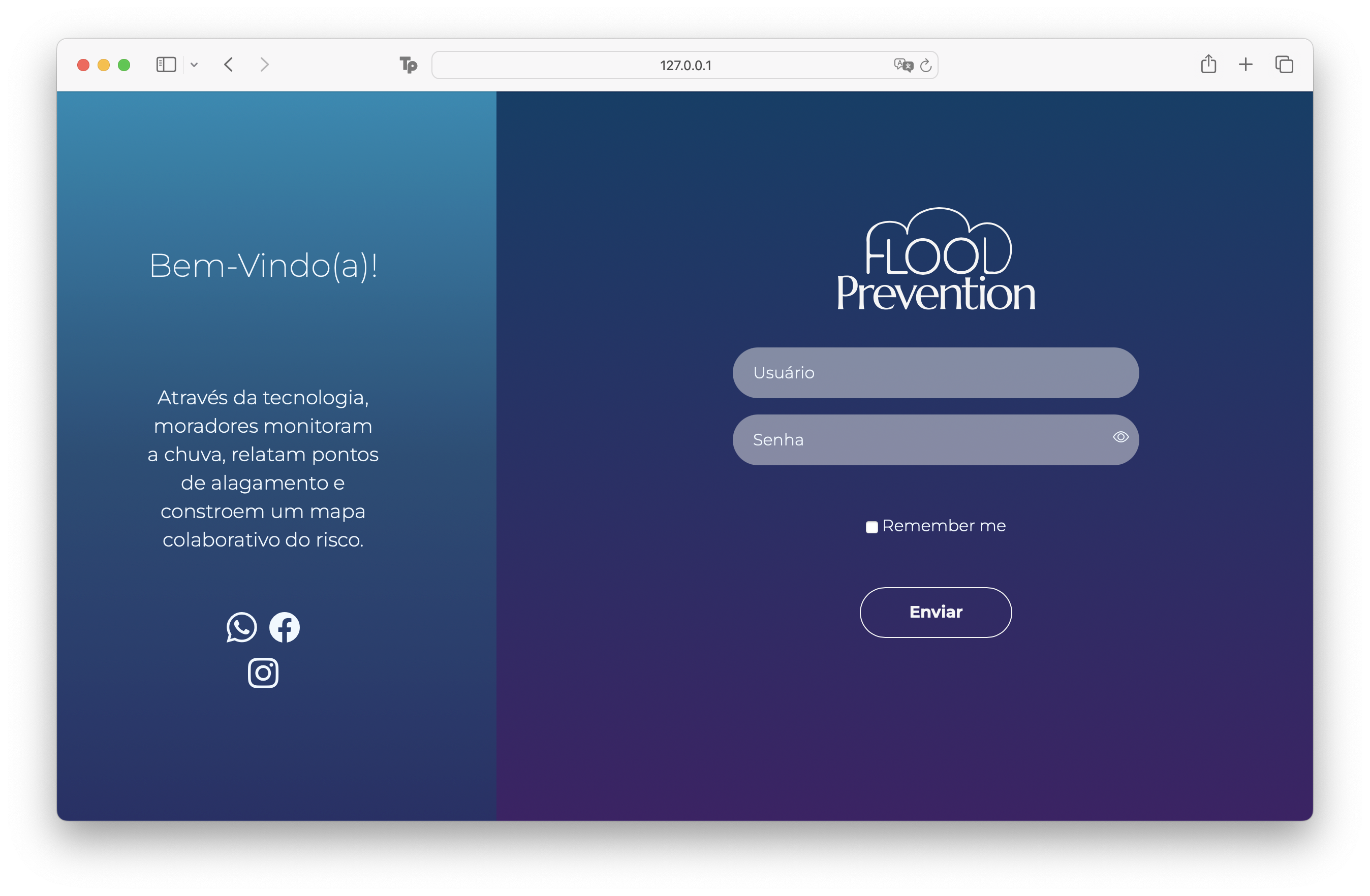
# 4. PJBL FASE 2 - PROTÓTIPO DE APLICAÇÃO WEB (RA2)

A fase de prototipação do projeto na fase de construir a aplicação Web é constituída de vários requisitos, principalmente referente a possibilidade de *CRUD*, ou seja, a partir dessa etapa desenvolvemos uma aplicação no qual é permitido criar, ler, atualizar e deletar. Dessa forma, através dessas funcionalidades os usuários previamente autorizados, após realizarem login, têm acesso as telas para cada uma das ações que se deseja realizar.

## 4.1 Login e presença de Blueprints

Na ilustração abaixo temos o nosso modelo de tela de login, no qual é preciso para que sejam acessados outros locais do site.

**Figura 2: página de login.**



Nessa página podemos exemplificar através da ilustração que contém o código logo a seguir. Nela encontram-se o dicionário (necessário nesse módulo para armazenar as credenciais dos usuários), *blueprint* e rotas criadas para validação na página de login, especificamente.

**Figura 3: printscreen do código com dicionário de administrador, rotas e suas funções.**

A screen shot of a computer

Description automatically generated

**Figura 4: printscreen do código com dicionário de usuário e definição do *blueprint*.**

**A screen shot of a computer

Description automatically generated**

No entanto, no decorrer do projeto passaram a serem utilizados métodos mais eficazes para definição de *blueprints*, como podemos analisar na imagem abaixo, no qual todos passam a serem chamados no arquivo app\_controller.py e assim sejam processados ao iniciar a aplicação através do app.py.

**Figura 5: printscreen do código com definição dos *blueprints*.**

A screenshot of a computer

Description automatically generated

## 4.2 Views (HTMLs)

O diretório nomeado como ***views***contém todas as páginas HTML que serão renderizadas para os usuários da solução. Como mecanismo do Flask utilizamos uma página como base para que as outras telas herdem suas configurações e personalização. É possível identificar na ilustração abaixo a construção desse mecanismo na página **base.html** com um trecho do código.

**Figura 6: printscreen com o código da página base.html, neste trecho contém a lógica que será herdada pelas outras páginas referentes à barra de navegação.**

**A screenshot of a computer

Description automatically generated**

**Figura 7: printscreen com o código da página base.html, neste trecho está exposto a lógica para inserir os dados vindos de outra página.**

**A screenshot of a computer

Description automatically generated**

Para possibilitar esse desempenho foi necessário utilizar o Jinja2, no qual essa ferramenta permite a construção de páginas HTML com código em Python incorporados em seu corpo. Como exemplo pode-se citar a figura 7, nela está ilustrada uma condição, neste caso o “IF”.

# 5. RA – 03: BANCO DE DADOS (RA3)

# 6. RA – 04: INTEGRAÇÃO (RA4)

# 7. RELATO DE EXPERIÊNCIA

Nos conte como está sendo a sua experiencia nesta matéria. Acredita que está conseguindo entender o conteúdo? Está gostando de aprender coisas novas (Hardware)?, está sendo uma boa experiencia criativa?

# 8. REFERÊNCIAS

Caso houver, todas as citações e referências devem seguir um padrão único, sendo que o proponente pode decidir entre as normas vigentes da ABNT, VANCOUVER ou periódico indexado.

Para as referências utilizar espaçamento entre linhas simples e texto alinhado esquerda.

Todo material acessado online deve ter link e data de acesso.

Você pode utilizar programas organizadores de citação e referência (Ex: EndNote, Mendeley etc.), mas não se esqueça de conferir as referências mesmo assim para evitar transtornos.

ELECTRONICS WORKSHOP**. IOT based flood monitoring onitoring system using ESP 32 and ultrasonic sensor**. Disponível em: https://www.instructables.com/IOT-BASED-FLOOD-MONITORING-SYSTEM-USING-ESP-32-AND-1/. Acesso em: 29/03/2024.

WOKWI. **wokwi-buzzer**. Disponível em: https://docs.wokwi.com/pt-BR/parts/wokwi-buzzer. Acesso em: 01/04/2024.

WOKWI. **wokwi-dht22**. Disponível em: https://docs.wokwi.com/pt-BR/parts/wokwi-dht22. Acesso em: 28/03/2024.

WOKWI. **wokwi-led**. Disponível em: https://docs.wokwi.com/pt-BR/parts/wokwi-led. Acesso em: 28/03/2024.

WOKWI. **wokwi-hc-sr04**. Disponível em: https://docs.wokwi.com/pt-BR/parts/wokwi-hc-sr04. Acesso em: 28/03/2024.