

# Sistema IOT para monitoramento de irrigação de cultura

## 1. OBJETIVO

O objetivo deste documento é descrever informações essenciais para o projeto, como as ferramentas utilizadas, seus requisitos e suas principais funções.

## 2. VISÃO GERAL DO CONTEXTO

A agricultura é uma atividade essencial para a sustentabilidade e segurança alimentar mundial, especialmente em países em desenvolvimento. A agricultura familiar é um importante componente desse setor, pois representa a maioria dos agricultores e pequenos produtores rurais e contribui significativamente para a produção de alimentos e geração de renda. No entanto, a agricultura familiar enfrenta desafios como a falta de recursos e acesso à tecnologia e conhecimento técnico, o que pode afetar a produtividade e rentabilidade. A implementação de sistemas de monitoramento de irrigação baseados em tecnologia e conhecimento técnico pode ajudar a melhorar a eficiência da produção, reduzir o desperdício de recursos e aumentar a rentabilidade.

## 3. MAPEAMENTO DOS PROBLEMAS

Analisando o contexto de pequenas áreas de plantio de milho, observa-se os seguintes problemas:

ID	PROBLEMA	DETALHAMENTO
PR01	Ausência de manejo técnico	Na agricultura familiar, a gestão das áreas de cultivo frequentemente é baseada em conhecimentos tradicionais, sem o uso de práticas e ferramentas técnicas que podem melhorar a produção.
PR02	Desperdício hídrico	A falta de precisão no cálculo do tempo de irrigação pode levar a um consumo excessivo de água, causando desperdício hídrico.
PR03	Desperdício energético	O uso inadequado do sistema de irrigação, quando além do necessário, causa o desperdício energético, aumentando os custos de operação.
PR04	Mal aproveitamento do potencial produtivo	O uso inadequado do sistema de irrigação, como a falta de ajustes do volume diário de água de acordo com as condições climáticas e o estágio de crescimento da cultura, prejudica no desenvolvimento da altura e frutos das plantas, uma vez que a água está relacionada com os processos metabólicos das mesmas, o que resulta em um mau aproveitamento do potencial produtivo.

4. VISÃO GERAL DA SOLUÇÃO PROPOSTA

A solução proposta é um sistema que permite o **monitoramento da irrigação** e de **informações climáticas**, baseado em tecnologia de **Internet das coisas (IoT)** que utiliza **dados meteorológicos coletados por uma estação** para calcular a necessidade hídrica de uma cultura, mediante o cálculo da **evapotranspiração (ETo)** proposto por **Penman-Monteith**. Pode-se acompanhar e gerenciar os dados coletados remotamente através de uma **plataforma web** responsiva. O sistema é projetado para ajudar os produtores a tomar decisões em seu manejo considerando o uso da água e energia de forma calculada, reduzindo o desperdício de recursos e maximizando a produtividade e rentabilidade da propriedade.

5. ENVOLVIDOS

FUNÇÃO/PAPEL	DESCRIÇÃO
Produtor	Pessoa responsável pela administração e operação da propriedade agrícola, além de planejar e executar as atividades de cultivo, incluindo a escolha das culturas, preparação do solo, plantio, colheita e comercialização dos produtos.
Estação Meteorológica	Conjunto de equipamentos eletrônicos responsáveis por coletar e disponibilizar os dados meteorológicos para o sistema realizar o cálculo da evapotranspiração (ETo). Deve ser instalada em uma posição estratégica para obter dados precisos e deve ser regularmente mantida e calibrada para garantir a precisão dos dados coletados.

6. REQUISITOS FUNCIONAIS

Produtor				
ID	DESCRIÇÃO DO REQUISITO	COMPLEXIDADE	CRITICIDADE	DEPENDÊNCIA
RF001	O sistema deve permitir que o <b>PRODUTOR &lt;&lt;se cadastre&gt;&gt;</b> por meio de <b>nome</b> , <b>email</b> e <b>senha</b> , tendo suas informações de cadastro armazenadas de forma segura e podendo alterá-las a qualquer momento.	Média ▾	Alta ▾	
RF002	O sistema deve exigir que o <b>PRODUTOR</b> se autentique por meio de <b>&lt;&lt;login&gt;&gt;</b> utilizando o <b>email</b> e a <b>senha</b> cadastrados. Caso contrário, o sistema deve exibir uma mensagem de erro informando que as informações de login estão incorretas. Além disso, o sistema deve permitir que os usuários façam	Média ▾	Alta ▾	RF001

	<<logout>> a qualquer momento, desconectando suas contas do sistema.			
RF003	O sistema deve permitir que o <b>PRODUTOR</b> <<cadastre>> sua cultura informando o <b>tipo de cultura</b> , a <b>vazão de água do sistema de irrigação</b> , a <b>data de plantio</b> , a <b>área da plantação</b> , e o <b>tempo diário de irrigação atual</b> .	Média ▾	Alta ▾	RF002
RF004	O sistema deve permitir que o <b>PRODUTOR</b> <<edite>> informações de sua cultura ou <<exclua>>.	Baixa ▾	Baixa ▾	RF002, RF003
RF005	O sistema deve permitir que o <b>PRODUTOR</b> <<visualize>> a <b>evapotranspiração de referência</b> , <b>evapotranspiração diária da cultura</b> , e as <b>médias diárias</b> dos valores climatológicos coletados pela estação.	Alta ▾	Alta ▾	RF002, RF007, RF009, RF010, RF011
	O sistema deve permitir que o <b>PRODUTOR</b> <<visualize>> o <b>período de tempo</b> que o sistema de irrigação deve estar ativo para suprir a demanda hídrica da cultura, e também visualizar a <b>economia de consumo de água</b> com base na comparação entre o tempo de irrigação anterior e o tempo de irrigação calculado.	Baixa ▾	Baixa ▾	RF002, RF012, RF013
RF006	O sistema deve permitir que o <b>PRODUTOR</b> desative a estação, podendo ligá-la a qualquer momento.	Baixa ▾	Baixa ▾	RF002
ESTAÇÃO				
RF007	A <b>ESTAÇÃO</b> deve <<enviar os dados>> coletados para o banco de dados, permitindo acesso remoto, análise, armazenamento e gerenciamento dos mesmos.	Média ▾	Alta ▾	

RF008	A <b>ESTAÇÃO</b> deve <<informar>> caso algum sensor não colete os dados necessários, ou ainda apresente mau funcionamento deduzido pelo salto na amplitude dos valores medidos, para que seja feita a verificação e manutenção da estação.	Média ▾	Média ▾	RF002
GERAIS				
RF009	O sistema deve <<calcular a média diária dos valores climáticos>>.	Média ▾	Baixa ▾	RF007
RF010	O sistema deve <<calcular a evapotranspiração de referência>> com base nos dados climatológicos coletados pela estação.	Média ▾	Alta ▾	RF007
RF011	O sistema deve <<calcular a evapotranspiração diária da cultura>>, com base no coeficiente de evaporação da cultura em questão e na evaporação de referência calculada.	Alta ▾	Alta ▾	RF002, RF003, RF010
RF012	O sistema deve <<calcular e informar o período de tempo>> que o sistema de irrigação deve estar ativo para suprir a demanda hídrica com base na evapotranspiração diária da cultura e demais características da plantação: <b>vazão de água do sistema de irrigação, data de plantio e área da plantação.</b>	Média ▾	Alta ▾	RF002, RF011
RF013	O sistema deve <<calcular e informar a economia de consumo de água>> com base na comparação entre o tempo de irrigação anterior e o tempo de irrigação calculado.	Média ▾	Média ▾	RF002, RF012

## 7. REQUISITOS NÃO FUNCIONAIS

REQUISITOS NÃO FUNCIONAIS			
ID	DESCRIÇÃO DO REQUISITO	COMPLEXIDADE	CRITICIDADE
RNF001	A interface deve ser clara e objetiva.	Baixa ▾	Alta ▾
RNF002	Ter mecanismos de autenticação, proteger dados sensíveis e ser seguro.	Média ▾	Alta ▾
RNF003	O sistema deve apresentar mensagem de erro, quando for o caso, de forma clara e objetiva.	Média ▾	Média ▾

## 8. ESTRUTURA DO BANCO DE DADOS

<https://estacao-meteorologic-default-rtdb.firebaseio.com/>

### Obs:

Alguns atributos, para fácil identificação na base de dados, tiveram seus nomes simplificados.

Na tabela “InfoClima”, alguns atributos foram nomeados com base os valores da equação de Penman-Monteith:

- **Es** equivale a pressão de vapor de saturação.
- **Ea** equivale a pressão parcial de vapor.
- **y** equivale ao coeficiente psicrométrico.
- **Delta** equivale a declividade da curva de pressão de vapor de saturação.

## 9. DEFINIÇÃO DA ARQUITETURA DA SOLUÇÃO

### a. ARQUITETURA FÍSICA

**Interface (Front-end):** A interface da aplicação será disponibilizada tanto para desktop quanto para smartphones, permitindo que os produtores acessem e gerenciem os dados coletados de forma conveniente.

**Back-end:** O back-end será instalado e configurado em uma máquina local, com a possibilidade de ser migrado para um servidor, garantindo que os dados coletados estejam sempre disponíveis e seguros.

**Hardware:** A solução inclui o uso de sensores como o TEMT6000 para medir a luminosidade, o BMP280 para medir a pressão atmosférica, o DHT22 para medir a temperatura e umidade do ar e um anemômetro caseiro com reed switch para medir a velocidade do vento. Estes sensores coletam dados precisos e confiáveis para o cálculo da evapotranspiração e do tempo de irrigação.

### b. ARQUITETURA LÓGICA

## **FRONT END:**

### **Flet**

Flet é uma estrutura que permite a construção de aplicativos interativos multi-usuário web, desktop e móveis. Atualmente funciona como uma biblioteca python. Através dela, a interface é desenvolvida por meio de controles Flet que são baseados no Flutter do Google. Nesse sentido, a mesma "envolve" o código python para widgets do Flutter, refinando-os, combinando widgets menores, ocultando complexidades, implementando as melhores práticas da interface do usuário e aplicando padrões razoáveis.

## **BACK END:**

### **Python**

Python é uma linguagem de programação criada em 1991 na Holanda por Guido Von Rossum. Python foi desenvolvida na linguagem ABC, possui uma sintaxe inspirada em C. Uma linguagem de alto nível, interpretada de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte sendo a linguagem mais popular do ano de 2020.

## **HARDWARE:**

### **C/C++**

C é uma linguagem de programação compilada de propósito geral, estruturada, imperativa, procedural, padronizada pela Organização Internacional para Padronização, criada em 1972 por Dennis Ritchie na empresa AT&T Bell Labs para desenvolvimento do sistema operacional Unix. Além disso, serviu como base para outras linguagens e pode ser usada no desenvolvimento de sistemas operacionais, aplicativos de todos os tipos, drivers e outros controladores de dispositivos, programar microcontroladores, etc. Enquanto C++ é uma linguagem de alto nível, com recursos adicionais de orientação a objetos, sendo uma evolução da linguagem C, criada pelo mesmo autor, que adiciona recursos de orientação a objetos.

## **BANCO DE DADOS:**

### **Firebase**

Lançado em 2011, o Firebase é um BaaS pertencente à Google que provê um conjunto gratuito de serviços de hospedagem diversos, como banco de dados realtime noSQL – com armazenamento em forma de árvores ou documento JSON – sistema de autenticação e outros mecanismos de segurança, envio de arquivos etc. Além disso, atua também como serviço API REST, com protocolo HTTPS embutido.

## **CONTROLE DE VERSÃO:**

### **Github**

O controle de versão é um processo fundamental para qualquer projeto de software, pois permite rastrear as alterações realizadas no código ao longo do tempo e garantir que o projeto esteja sempre atualizado. O projeto utilizará a plataforma Github como ferramenta de versionamento, permitindo que todos os envolvidos no projeto possam acessar e contribuir com o código de forma colaborativa. Além disso, o Github permite a criação de branches (ramificações) do projeto, permitindo que as alterações possam ser testadas antes de serem integradas à versão principal do código. Isso garante uma maior segurança e estabilidade do projeto, além de facilitar a manutenção e a correção de eventuais bugs.