FECAP

**PROJETO INTERDISCIPLINAR**

**GRUPO CGKM**

**Requisitos da disciplina Modelagem de Software e Arquitetura de Sistemas**

São Paulo

2024

**INTEGRANTES DO PROJETO e RA’S**

## Caua William Barbieri Brandao - 24225752

## Gabriel Orlandi Portes - 24026407

## Karoline Lemos Avelar - 24036456

## Matheus Santoro Veiga - 9020057

Sumário

[**1. INTRODUÇÃO 3**](#_Toc178795916)

[**2. DOCUMENTO DE ABERTURA DO PROJETO 3**](#_Toc178795917)

[3. REQUISITOS DE SISTEMA 6](#_Toc178795927)

[**3.1 REQUISITOS FUNCIONAIS DE SOFTWARE 6**](#_Toc178795928)

[**3.2 REQUISITOS NÃO FUNCIONAIS DE SOFTWARE 8**](#_Toc178795929)

[**4. CASOS DE USO 10**](#_Toc178795930)

[**5. ARQUITETURA DO SISTEMA 11**](#_Toc178795931)

[**6. REFERÊNCIAS BIBLIOGRÁFICAS 11**](#_Toc178795932)

## 1. INTRODUÇÃO

O desenvolvimento da agricultura urbana é uma iniciativa importante para promover a segurança alimentar, a sustentabilidade e a melhoria da qualidade de vida em áreas urbanas. Esse projeto consiste em um site interativo que visa disseminar o conhecimento sobre a agricultura urbana e incentivar o cultivo de alimentos em espaços urbanos. O site também incluirá uma calculadora de rendimento potencial, que permitirá aos usuários estimarem o volume de produção agrícola de acordo com o espaço disponível e as características dos cultivos.

O projeto está alinhado com os Objetivos de Desenvolvimento Sustentável (ODS) da ONU, com ênfase no **ODS 2** - Fome Zero e Agricultura Sustentável, e no **ODS 12** - Consumo e Produção Responsáveis



## 2. DOCUMENTO DE ABERTURA DO PROJETO

### Prefácio

Este documento é destinado a desenvolvedores, gestores de projeto e outros stakeholders interessados no desenvolvimento de soluções tecnológicas para a promoção da agricultura urbana. Ele apresenta uma visão geral do sistema, define os requisitos de usuário e sistema, e descreve a arquitetura de alto nível que será implementada. A evolução do sistema será acompanhada com atualizações contínuas baseadas no feedback dos usuários e na adaptação a novos desafios.

### Introdução

O projeto foi concebido para atender à necessidade de uma plataforma educativa e interativa que auxilie no planejamento e execução de atividades agrícolas em áreas urbanas, especialmente em espaços restritos. O sistema oferece uma área de login para que os usuários possam se cadastrar e acessar funcionalidades exclusivas, como a **Calculadora de Rendimento Potencial de Cultivos**, que calcula estimativas de produção agrícola com base em variáveis como área disponível e tipo de planta.

O sistema também contribui para os objetivos estratégicos de promover práticas sustentáveis em áreas urbanas e aumentar a conscientização sobre a importância da agricultura urbana como um meio de garantir a segurança alimentar e o consumo responsável de recursos.

### Glossário

**Agricultura Urbana**: Prática de cultivo, processamento e distribuição de alimentos em áreas urbanas.

**ODS**: Objetivos de Desenvolvimento Sustentável propostos pela ONU para equilibrar crescimento econômico, proteção ambiental e bem-estar social.

**Cadastro de Usuário**: Processo de registro de novos usuários, que inclui e-mail e senha.

**Calculadora de Rendimento**: Ferramenta que permite ao usuário calcular o rendimento estimado de diferentes tipos de cultivos com base em variáveis fornecidas.

### Definição de Requisitos de Usuário

O sistema deve fornecer aos usuários as seguintes funcionalidades:

**Cadastro e Login de Usuário**: O usuário poderá criar uma conta fornecendo e-mail e senha para acessar áreas restritas do site.

**Acesso à Calculadora de Rendimento**: Após o login, o usuário terá acesso à calculadora para estimar o rendimento de cultivos com base em diferentes variáveis.

**Interface Intuitiva e Responsiva**: O site deve ser fácil de navegar e adaptável a diferentes dispositivos, incluindo smartphones e tablets.

**Ferramenta de Contato**: Deve haver um formulário de contato para que os usuários possam enviar dúvidas e sugestões.

### Arquitetura do Sistema

A arquitetura proposta é composta por três camadas:

**Camada de Apresentação**:  
Desenvolvida com HTML, CSS e JavaScript para o front-end, garantindo uma experiência visual agradável e responsiva para o usuário.

**Camada de Lógica de Negócio**:  
Implementada em **C#**, essa camada será responsável por executar os cálculos de rendimento e gerenciar a autenticação dos usuários.

**Camada de Persistência de Dados**:  
Utilizará um banco de dados **SQL Server** para armazenar informações dos usuários, como credenciais e histórico de cálculos.

### Especificação de Requisitos do Sistema

Os requisitos do sistema são divididos em requisitos funcionais e não funcionais.

#### **Requisitos Funcionais de Software**

1. O sistema deve permitir que os usuários se cadastrem usando um e-mail e senha.
2. O sistema deve autenticar os usuários cadastrados por meio de um formulário de login.
3. Após a autenticação, o sistema deve redirecionar o usuário para a área de cálculo de rendimento.
4. A calculadora deve aceitar a área disponível para cultivo, tipo de planta e outras variáveis para estimar o rendimento potencial.
5. O sistema deve exibir os resultados de forma gráfica e textual para facilitar a compreensão.
6. Deve ser possível recuperar senhas esquecidas por meio do e-mail de cadastro.

#### **3.2 Requisitos Não Funcionais de Software**

1. O tempo de resposta do sistema deve ser inferior a 2 segundos para cada operação de cálculo.
2. A interface deve ser compatível com os navegadores mais utilizados, como Google Chrome e Mozilla Firefox.
3. As senhas dos usuários devem ser criptografadas para garantir a segurança dos dados.
4. O sistema deve suportar até 10.000 usuários simultâneos sem degradação de desempenho.
5. O sistema deve estar disponível 99,9% do tempo.
6. Deve haver logs de auditoria para registrar tentativas de login e falhas de autenticação.

### Modelos do Sistema

**Modelo de Dados**: Representação de tabelas de usuários e cálculos realizados.

**Modelo de Fluxo de Dados**: Mostra a interação entre o usuário, o front-end e o back-end.

**Modelo de Interface**: Esboço das principais telas, incluindo login, cadastro, calculadora e resultados.

### Evolução do Sistema

O sistema está preparado para incorporar novas funcionalidades no futuro, como:

**Monitoramento de Cultivos**: Adição de um dashboard para monitorar o crescimento dos cultivos e sugerir melhorias.

**Previsão Climática**: Integração com APIs meteorológicas para fornecer previsões que afetam o rendimento.

### Apêndices

**Requisitos de Hardware**: O sistema deve ser capaz de rodar em servidores com um mínimo de 4 GB de RAM e 100 GB de espaço em disco.

**Configuração do Banco de Dados**: A estrutura do banco de dados incluirá tabelas para usuários, cálculos e logs de acesso.

# 3. REQUISITOS DE SISTEMA

## 3.1 REQUISITOS FUNCIONAIS DE SOFTWARE

Necessários 6 requisitos

|  |  |
| --- | --- |
| **RFS01** Cadastro de Usuário | |
| **Função** | Permitir o cadastro de novos usuários. |
| **Descrição** | O sistema deve permitir que o usuário se registre fornecendo seu e-mail e criando uma senha. |
| **Entradas** | E-mail e senha. |
| **Fonte** | Formulário de cadastro na interface do usuário. |
| **Saídas** | Mensagem de confirmação de cadastro. |
| **Ação** | Armazena os dados do usuário no banco de dados e envia um e-mail de confirmação. |

|  |  |
| --- | --- |
| **RFS02** Login de Usuário | |
| **Função** | Autenticar usuários. |
| **Descrição** | O sistema deve permitir que o usuário insira seu e-mail e senha para acessar funcionalidades restritas. |
| **Entradas** | E-mail e senha |
| **Fonte** | Formulário de login na interface do usuário. |
| **Saídas** | Redirecionamento para a área de usuário autenticado. |
| **Ação** | Verifica as credenciais no banco de dados e concede acesso ao usuário. |

|  |  |
| --- | --- |
|  |  |
| **RFS03** Calculadora de Rendimento | |
| **Função** | Estimar o rendimento de cultivos. |
| **Descrição** | O usuário insere parâmetros como área, tipo de cultivo e tempo de cultivo para calcular o rendimento. |
| **Entradas** | Área (m²), tipo de cultivo, tempo de cultivo (semanas). |
| **Fonte** | Dados fornecidos pelo usuário. |
| **Saídas** | Relatório do rendimento estimado. |
| **Ação** | Processa as informações e aplica os cálculos. |

|  |  |
| --- | --- |
|  |  |
| **RFS04** Recuperação de Senha | |
| **Função** | Permitir recuperação de senhas. |
| **Descrição** | O sistema deve permitir que o usuário insira seu e-mail para receber um link de redefinição de senha. |
| **Entradas** | E-mail de cadastro. |
| **Fonte** | Formulário de recuperação de senha. |
| **Saídas** | E-mail com link para redefinição. |
| **Ação** | Verifica o e-mail no banco de dados e envia o link. |

|  |  |
| --- | --- |
|  |  |
|  |  |
| **RFS05** Formulário de Contato | |
| **Função** | Receber mensagens dos usuários. |
| **Descrição** | O sistema deve permitir que usuários enviem dúvidas ou sugestões. |
| **Entradas** | Nome, e-mail, mensagem. |
| **Fonte** | Formulário de contato. |
| **Saídas** | Confirmação de envio. |
| **Ação** | Armazena a mensagem e a encaminha ao suporte. |

|  |  |
| --- | --- |
|  |  |
| **RFS06** Edição de Perfil | |
| **Função** | Permitir atualização de informações do usuário. |
| **Descrição** | O usuário deve poder atualizar e-mail, senha e preferências. |
| **Entradas** | Novos valores dos campos de perfil. |
| **Fonte** | Formulário de edição de perfil. |
| **Saídas** | Mensagem de confirmação de atualização. |
| **Ação** | Atualiza os dados no banco de dados. |

## 3.2 REQUISITOS NÃO FUNCIONAIS DE SOFTWARE

Necessários 6 requisitos

|  |  |
| --- | --- |
|  |  |
| **RFS01** Segurança de Dados | |
| **Função** | Proteger dados dos usuários. |
| **Descrição** | O sistema deve criptografar as senhas usando algoritmos de segurança, como BCrypt. |
| **Entradas** | Senhas dos usuários. |
| **Fonte** | Banco de dados. |
| **Saídas** | Senhas criptografadas. |
| **Ação** | Aplica criptografia antes de armazenar. |
|  |  |

|  |  |
| --- | --- |
|  |  |
| **RFS02** Desempenho | |
| **Função** | Garantir tempos de resposta rápidos. |
| **Descrição** | O sistema deve responder a operações em menos de 3 segundos. |
| **Entradas** | Requisições do usuário. |
| **Fonte** | Sistema. |
| **Saídas** | Respostas das requisições. |
| **Ação** | Otimiza o desempenho das operações. |

|  |
| --- |
|  |
| **RFS03** Otimiza o desempenho das operações. | |
| **Função** | Assegurar que o sistema funcione em diversos navegadores. |
| **Descrição** | O sistema deve ser compatível com navegadores modernos, como Chrome, Firefox e Safari. |
| **Entradas** | Acesso via navegador. |
| **Fonte** | Usuários. |
| **Saídas** | Interface do sistema. |
| **Ação** | Testa e corrige para diferentes navegadores. |

|  |  |
| --- | --- |
|  |  |
| **RFS04** Interface Responsiva | |
| **Função** | Adaptar o layout do sistema. |
| **Descrição** | O site deve ser responsivo, ajustando-se a diferentes tamanhos de tela. |
| **Entradas** | Resolução de tela. |
| **Fonte** | Dispositivos dos usuários. |
| **Saídas** | Layout ajustado. |
| **Ação** | Aplica CSS responsivo. |

|  |  |
| --- | --- |
|  |  |
| **RFS05** Disponibilidade | |
| **Função** | Garantir alta disponibilidade. |
| **Descrição** | O sistema deve estar disponível 99,9% do tempo durante horários comerciais. |
| **Entradas** | Conexões dos usuários. |
| **Fonte** | Servidores. |
| **Saídas** | Acesso contínuo ao sistema. |
| **Ação** | Monitora e mantém servidores. |

|  |  |
| --- | --- |
|  |  |
| **RFS06** Usabilidade | |
| **Função** | Proporcionar uma experiência intuitiva. |
| **Descrição** | A interface deve ser fácil de navegar e entender, com documentação disponível. |
| **Entradas** | Interação do usuário. |
| **Fonte** | Design de interface. |
| **Saídas** | Satisfação do usuário. |
| **Ação** | Realiza testes de usabilidade e ajusta conforme feedback. |

## 4. CASOS DE USO

**Caso de Uso 1: Cadastro de Usuário**

**Descrição**: Um usuário se cadastra no sistema fornecendo seu e-mail e criando uma senha.

**Ator**: Visitante (não autenticado).

**Fluxo de Evento**:

1. O usuário acessa a página de cadastro.
2. Insere e-mail e senha.
3. O sistema verifica se o e-mail já está cadastrado.
4. Se não estiver, armazena os dados e envia um e-mail de confirmação.
5. O usuário é redirecionado para a página de login.

**Caso de Uso 2: Login e Autenticação**

**Descrição**: Um usuário faz login na plataforma usando e-mail e senha.

**Ator**: Usuário cadastrado.

**Fluxo de Evento**:

1. O usuário acessa a página de login.
2. Insere o e-mail e a senha.
3. O sistema verifica as credenciais.
4. Se corretas, o usuário é redirecionado para a página principal.

**Caso de Uso 3: Uso da Calculadora de Rendimento**

**Descrição**: Um usuário autenticado acessa a calculadora e insere variáveis para estimar o rendimento de cultivos.

**Ator**: Usuário autenticado.

**Fluxo de Evento**:

1. O usuário acessa a página da calculadora.
2. Insere área de cultivo, tipo de planta e tempo de cultivo.
3. Clica em "Calcular".
4. O sistema processa as informações e exibe o resultado.

## 5. ARQUITETURA DO SISTEMA

O sistema utiliza uma **arquitetura em três camadas**:

1. **Camada de Apresentação (Front-end)**:

Desenvolvida com HTML, CSS e JavaScript, proporcionando uma interface intuitiva e responsiva.

1. **Camada de Lógica de Negócio (Back-end)**:

Implementada em **C#**, gerenciando os cálculos de rendimento e autenticação de usuário.

1. **Camada de Persistência de Dados**:

Utiliza um banco de dados **SQL Server** para armazenamento de informações de cadastro, logs de acesso e cálculos realizados.

## 6. REFERÊNCIAS BIBLIOGRÁFICAS

ONU - Objetivos de Desenvolvimento Sustentável. Disponível em: https://brasil.un.org/pt-br/sdgs/2

SOMMERVILLE, I. **Engenharia de Software.** 11ª Edição. São Paulo: Pearson Addison-Wesley, 2017.