

# Projeto Temático I

## MyGardenCare

Guilherme Kairê Morais Barea, Henrique Quissini Jacóbi, Maurício Negrele Knabben

Área do Conhecimento de Ciências Exatas e Engenharias

Universidade de Caxias do Sul (UCS) - Caxias do Sul - RS - Brasil

**Resumo.** *O artigo discute a proposta, desenvolvimento e funcionamento de um sistema de gestão de jardins, o MyGardenCare. Desenvolvido em C#, o MyGardenCare permite que o usuário cadastre suas plantas e os cuidados relacionados a elas, bem como acompanhe o cronograma de atividades que devem ser realizadas para o devido desenvolvimento das plantas. O sistema é composto por diferentes camadas na implementação, delegando determinadas responsabilidades para pontos distintos, assegurando a organização do código.*

## 1. Introdução

Nos dias de hoje a pauta do meio ambiente está cada vez mais presente no dia a dia das pessoas. As mudanças climáticas e eventos extremos estão se tornando cada vez mais comuns e servem de alerta para que cuidemos do meio ambiente. A fim de conectar-se com a natureza e promover os cuidados com o meio ambiente, a jardinagem é uma prática muito comum na sociedade moderna, sendo acessível tanto para aqueles que dispõem de terras e vivem no campo, como aqueles que vivem nas cidades e utilizam outras alternativas.

Apesar de acessível e amplamente realizada, a jardinagem exige uma série de cuidados diários e monitoramento constante. Manter todas as atividades organizadas e garantir que não serão negligenciadas pode ser um desafio, especialmente em jardins maiores ou em projetos colaborativos. Pensando nestas dificuldades o MyGardenCare foi desenvolvido, visando facilitar e organizar a realização de tarefas e gerenciamento do jardim.

Durante a disciplina de Projeto Temático I o sistema foi pensado, arquitetado e produzido, e neste presente artigo será descrito, abordando os objetivos principais, o levantamento de requisitos, o planejamento de arquitetura do projeto (estruturação do código e do banco de dados) e a implementação do sistema, com o fluxo de utilização das principais funcionalidades presentes na aplicação. Também serão abordados os desafios encontrados ao longo do projeto.

## 2. Compreendendo o Problema e Levantando os Requisitos

A primeira fase do projeto focou na compreensão do problema e na proposição de uma solução a ser desenvolvida. O objetivo geral do projeto era criar um aplicativo desktop, seguindo as etapas de um processo de desenvolvimento de software, tais como: análise de negócios, análise de requisitos, análise de sistemas, design de software e definição da arquitetura de software. O desafio apresentado foi desenvolver um software voltado para a gestão de jardins, aplicando princípios de programação orientada a objetos, fundamentos de banco de dados (persistência) e de engenharia de software. A ideia central do software foi refinada através de discussões durante o primeiro encontro semanal da disciplina, culminando na decisão de criar uma plataforma para gerenciamento e monitoramento de cuidados de jardinagem.

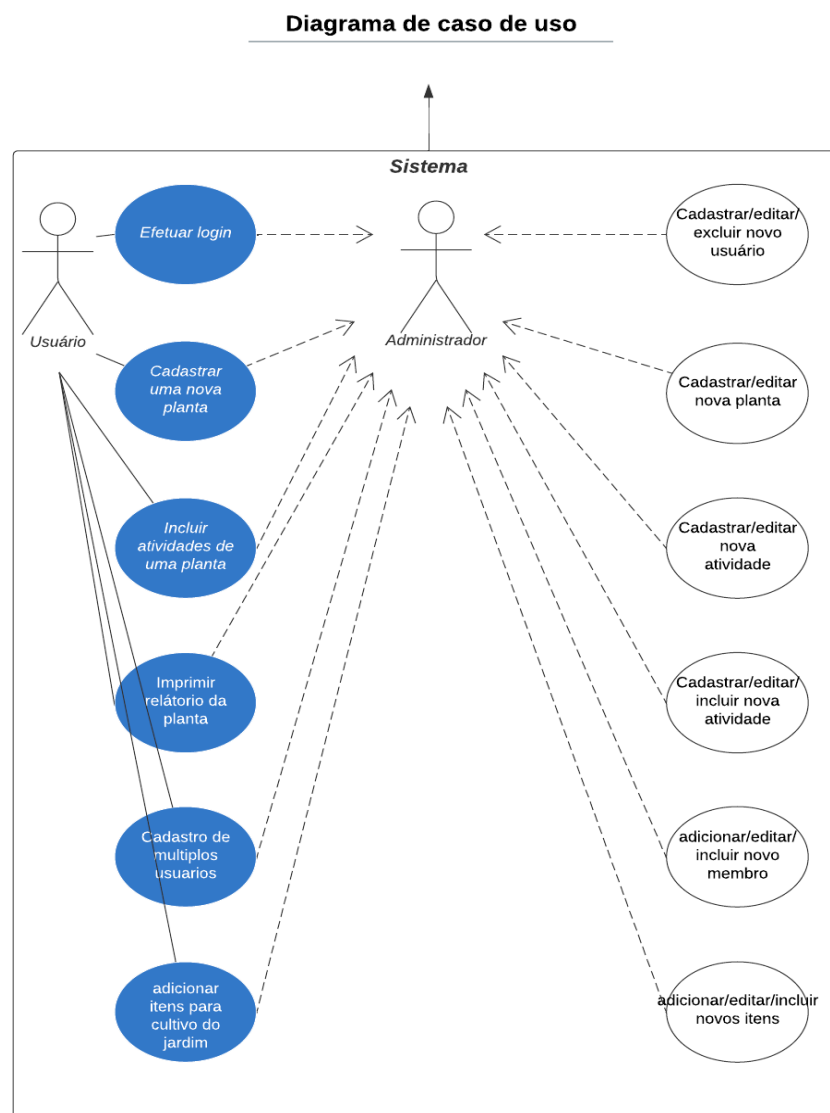
Ao longo dos encontros semanais, também foi realizado o levantamento de requisitos do projeto. O levantamento de requisitos é uma etapa essencial no processo de desenvolvimento de software, onde desenvolvedores e partes interessadas colaboram para entender as necessidades e expectativas do sistema a ser desenvolvido. Esta fase envolve coletar, analisar, documentar e validar as informações necessárias para o software, incluindo funcionalidades, desempenho, segurança, interfaces do usuário e outros requisitos técnicos. A agilidade é um fator importante na engenharia de requisitos, com a intenção de transferir ideias dos envolvidos para a equipe de desenvolvimento de forma eficiente e sem atrasos. É muito provável que novos requisitos continuem a surgir à medida que o desenvolvimento iterativo do artefato progride (PRESSMAN, 2021).

De acordo com Pressman (2021), a importância do levantamento de requisitos pode ser destacada pelos seguintes pontos: compreensão do problema, base para o desenvolvimento, satisfação do cliente, controle de mudanças, redução de riscos e comunicação eficaz.

Dessa forma, foram identificados como requisitos do usuário os seguintes aspectos: o usuário pode cadastrar plantas e programar atividades de cuidado como rega, poda e adubação, fornecendo detalhes sobre cada atividade. Como requisitos do sistema, foram identificadas as funcionalidades de: calcular o tempo médio entre cada cuidado, exibir uma agenda de atividades, inserir, editar e excluir plantas e atividades, além de restringir o acesso à lista de usuários apenas para administradores. Os requisitos funcionais descrevem o comportamento esperado do sistema. No projeto, foram identificados nove requisitos funcionais: RF 1 - Incluir/Excluir/Alterar dados de uma planta no jardim, como informações da espécie e fotos. RF 2 - Incluir/Excluir/Alterar atividades a serem realizadas em uma ou mais plantas. RF 3 - Gerar relatório de cuidados a serem realizados em um período. RF 4 - Marcar atividades realizadas nas plantas, em forma de checklist. RF 5 - Cadastrar múltiplos usuários para colaboração nas atividades. RF 6 - Gerar um histórico de atividades realizadas em determinada planta e para o jardim. RF 7 - Consultar as condições climáticas em um determinado período. RF 8 - Incluir/Excluir/Alterar produtos a serem armazenados e

utilizados nos cuidados referente ao jardim. RF 9 - Incluir/Excluir/Alterar dados de usuário, como cadastro, login e informações do usuário.

Um caso de uso ilustra uma jornada estilizada sobre como um usuário (desempenhando um de vários possíveis papéis) interage com o sistema sob determinadas circunstâncias (PRESSMAN, 2021). Segundo Pressman (2021), um ator é qualquer entidade externa que interage com o sistema. Neste projeto, foram definidos dois atores, ou perfis de usuário, cada um com permissões específicas, sendo eles o perfil de jardineiro e de administrador. Apenas o administrador tem acesso a uma tela onde é possível realizar as operações de CRUD (Create, Read, Update, Delete) com as entidades do sistema, inclusive com a classe Usuário, podendo alterar o nível de permissão de outros usuários. O perfil de jardineiro é limitado às operações de registrar e acompanhar atividades de cuidado com as plantas e atualizar alguns dados de sua conta. Para um melhor entendimento usamos dois tipos de modelos de casos de uso, na figura 1 está o diagrama, secundamente temos as tabelas de casos de uso usando requisitos funcionais.



**Figura 1. Diagrama de casos de uso**

**Caso 1:**

<b>Caso de uso</b>	Cadastro do cliente
<b>Requisitos Funcionais Atendidos</b>	RF9
<b>Atores</b>	Cliente
<b>Pré-condições</b>	Cliente registrar-se
<b>Pós-Condições</b>	Cliente registrado.
<b>Cenário de sucesso (fluxo principal)</b>	<ol style="list-style-type: none"><li>1. O cliente clica na opção para se cadastrar.</li><li>2. O sistema exibe a tela de cadastro.</li><li>3. O cliente insere seus dados de cadastro.</li><li>4. O cliente clica na opção de concluir.</li><li>5. O sistema valida os dados informados e efetua o cadastro.</li></ol>
<b>Extensão (fluxos alternativos)</b>	<ol style="list-style-type: none"><li>6. O cliente não preenche todas as informações necessárias.</li><li>7. O sistema informa o erro e pede para o cliente inserir novamente os dados.</li></ol>
<b>Requisitos especiais</b>	não possui.
<b>Frequência</b>	rotineiramente

**Caso 2:**

<b>Caso de uso</b>	Cadastro de uma nova planta
<b>Requisitos Funcionais Atendidos</b>	RF1
<b>Atores</b>	Cliente
<b>Pré-condições</b>	Cliente estar registrado e logado.
<b>Pós-Condições</b>	Cliente cadastrar a planta.
<b>Cenário de sucesso (fluxo principal)</b>	<p>O cliente já logado clica na opção de registro de uma nova planta.</p> <p>O sistema exibe a tela para cadastro da planta.</p> <p>O cliente insere as informações da planta.</p> <p>O cliente clica na opção de salvar.</p> <p>O sistema valida os dados informados e efetua o cadastro da nova planta no jardim.</p>
<b>Extensão (fluxos alternativos)</b>	<p>O cliente não preenche todas as informações necessárias para o cadastro da planta.</p> <p>O sistema informa o erro e pede para o cliente inserir novamente as informações.</p>
<b>Requisitos especiais</b>	Não possui.
<b>Frequência</b>	rotineiramente

**Caso 3:**

<b>Caso de uso</b>	Cadastrar atividades de uma planta
<b>Requisitos Funcionais Atendidos</b>	RF2
<b>Atores</b>	Cliente
<b>Pré-condições</b>	Cliente estar registrado, logado e ter ao menos uma planta cadastrada.
<b>Pós-Condições</b>	Cliente visualizar o relatório da planta.
<b>Cenário de sucesso (fluxo principal)</b>	<ol style="list-style-type: none"><li>1. O cliente já logado clica na opção para adicionar atividades da planta.</li><li>2. O sistema exibe a tela para incluir informações das atividades da planta.</li><li>3. O usuário salva as modificações feitas.</li></ol>
<b>Extensão (fluxos alternativos)</b>	Não possui.
<b>Requisitos especiais</b>	Não possui.
<b>Frequência</b>	rotineiramente

**Caso 4:**

<b>Caso de uso</b>	Relatório de uma planta cadastrada
<b>Requisitos Funcionais Atendidos</b>	RF3
<b>Atores</b>	Cliente
<b>Pré-condições</b>	Cliente estar registrado, logado e ter ao menos uma planta cadastrada.
<b>Pós-Condições</b>	Cliente visualizar o relatório da planta.
<b>Cenário de sucesso (fluxo principal)</b>	<ol style="list-style-type: none"><li>1. O cliente já logado clica na opção de consultar o relatório de uma planta.</li><li>2. O sistema exibe a tela do relatório da planta com seus respectivos dados.</li><li>3. O cliente verifica o relatório para tirar suas conclusões sobre o jardim.</li></ol>
<b>Extensão (fluxos alternativos)</b>	Não possui.
<b>Requisitos especiais</b>	Não possui.
<b>Frequência</b>	rotineiramente

**Caso 5:**

<b>Caso de uso</b>	<b>Checklist de atividades</b>
<b>Requisitos Funcionais Atendidos</b>	RF4
<b>Atores</b>	Cliente
<b>Pré-condições</b>	Cliente estar registrado, logado e ter ao menos uma planta cadastrada e atividades completas.
<b>Pós-Condições</b>	Cliente marcar com sucesso o seu checklist
<b>Cenário de sucesso (fluxo principal)</b>	<ol style="list-style-type: none"><li>1. O cliente já logado clica na opção de consultar as atividades de uma planta.</li><li>2. O sistema exibe a tela das atividades da planta com seus respectivos dados.</li><li>3. O cliente verifica as atividades completadas e marca como feita no checklist.</li></ol>
<b>Extensão (fluxos alternativos)</b>	Não possui.
<b>Requisitos especiais</b>	Não possui.
<b>Frequência</b>	rotineiramente



**Caso 6:**

<b>Caso de uso</b>	Cadastro de múltiplos usuários
<b>Requisitos Funcionais Atendidos</b>	RF5
<b>Atores</b>	Cliente
<b>Pré-condições</b>	Cliente está registrado e logado.
<b>Pós-Condições</b>	Cliente ter sucesso no cadastro da colaboração
<b>Cenário de sucesso (fluxo principal)</b>	<ol style="list-style-type: none"><li>1. O cliente já logado clica na opção para convidar um usuário a sua atividade no jardim.</li><li>2. O sistema exibe a tela para colocar as informações necessárias</li><li>3. O cliente salva as informações adicionadas e termina o processo do cadastro.</li></ol>
<b>Extensão (fluxos alternativos)</b>	Não possui.
<b>Requisitos especiais</b>	Não possui.
<b>Frequência</b>	rotineiramente

**Caso 7:**

<b>Caso de uso</b>	Gerar um histórico de atividades realizadas de uma planta ou jardim.
<b>Requisitos Funcionais Atendidos</b>	RF6
<b>Atores</b>	Cliente
<b>Pré-condições</b>	Cliente estar registrado, logado e ter ao menos uma planta cadastrada e atividades completas.
<b>Pós-Condições</b>	Cliente ter sucesso ao gerar o histórico de atividades.
<b>Cenário de sucesso (fluxo principal)</b>	<ol style="list-style-type: none"><li>1. O cliente já logado clica na opção para gerar o histórico das atividades.</li><li>2. O cliente escolhe sua opção (gerar histórico do jardim ou de uma só planta).</li><li>3. O cliente gera com sucesso o histórico selecionado.</li></ol>
<b>Extensão (fluxos alternativos)</b>	Não possui.
<b>Requisitos especiais</b>	Não possui.
<b>Frequência</b>	rotineiramente

**Caso 8:**

<b>Caso de uso</b>	Inserir produtos a serem usados no jardim
<b>Requisitos Funcionais Atendidos</b>	RF8
<b>Atores</b>	Cliente
<b>Pré-condições</b>	Cliente estar registrado, logado e ter ao menos uma planta cadastrada.
<b>Pós-Condições</b>	Cliente inserir com êxito o produto.
<b>Cenário de sucesso (fluxo principal)</b>	<ol style="list-style-type: none"><li>1. O cliente já logado clica na opção para adicionar produtos.</li><li>2. O sistema exibe a tela para incluir produtos necessários para a planta.</li><li>3. O usuário salva as modificações feitas.</li></ol>
<b>Extensão (fluxos alternativos)</b>	Não possui.
<b>Requisitos especiais</b>	Não possui.
<b>Frequência</b>	rotineiramente

**Caso 9:**

<b>Caso de uso</b>	Cadastro do cliente
<b>Requisitos Funcionais Atendidos</b>	RF9
<b>Atores</b>	Cliente
<b>Pré-condições</b>	Cliente registrar-se
<b>Pós-Condições</b>	Cliente registrado.
<b>Cenário de sucesso (fluxo principal)</b>	<ol style="list-style-type: none"><li>1. O cliente clica na opção para se cadastrar.</li><li>2. O sistema exibe a tela de cadastro.</li><li>3. O cliente insere seus dados de cadastro.</li><li>4. O cliente clica na opção de concluir.</li><li>5. O sistema valida os dados informados e efetua o cadastro.</li></ol>
<b>Extensão (fluxos alternativos)</b>	<ol style="list-style-type: none"><li>6. O cliente não preenche todas as informações necessárias.</li><li>7. O sistema informa o erro e pede para o cliente inserir novamente os dados.</li></ol>
<b>Requisitos especiais</b>	não possui.
<b>Frequência</b>	rotineiramente

**Caso 10:**

<b>Caso de uso</b>	Login do cliente
<b>Requisitos Funcionais Atendidos</b>	RF9
<b>Atores</b>	Cliente
<b>Pré-condições</b>	Cliente está registrado.
<b>Pós-Condições</b>	Cliente efetuar o login.
<b>Cenário de sucesso (fluxo principal)</b>	<ol style="list-style-type: none"><li>1. O cliente clica na opção para logar.</li><li>2. O sistema exibe a tela de login.</li><li>3. O cliente insere seus dados de login.</li><li>4. O cliente clica na opção de fazer login.</li><li>5. O sistema valida os dados informados e efetua o login.</li></ol>
<b>Extensão (fluxos alternativos)</b>	<ol style="list-style-type: none"><li>1. O cliente não preenche todas as informações necessárias para o login.</li><li>2. O sistema informa o erro e pede para o cliente inserir novamente os dados.</li></ol>
<b>Requisitos especiais</b>	Não possui.
<b>Frequência</b>	rotineiramente

### 3. Arquitetura e Organização de Software

O MyGardenCare foi desenvolvido como uma aplicação desktop utilizando o ecossistema da Microsoft. A seguir, são detalhadas as tecnologias e a arquitetura do projeto.

#### 3.1. Linguagem de programação

A linguagem de programação utilizada no desenvolvimento do sistema foi C#, desenvolvida pela Microsoft, ela é usada para criar uma variedade de aplicativos, desde programas de computador e aplicativos web até jogos. Conhecida por ser fácil de aprender, especialmente para quem já tem alguma familiaridade com outras linguagens de programação, como Java ou C++. Optamos por essa escolha devido a seu fundamento orientado a objetos e sua robustez e pela capacidade de desenvolver aplicações ricas em funcionalidades e com boa performance, além do amplo suporte da Microsoft. A versão utilizada é a última disponível no presente momento deste projeto, versão 8.0 do .NET.

**IDE (Integrated Development Environment)**, ou Ambiente de Desenvolvimento Integrado, é um software que ajuda programadores a escrever, testar e depurar (corrigir erros) seus códigos de forma mais eficiente. Pense na IDE como uma "caixa de ferramentas" completa para programadores. Dentro dessa "caixa" você encontra:

- **Editor de Código:** Onde você escreve o código.
- **Compilador/Interpretador:** Converte o código que você escreveu em uma linguagem que o computador possa entender e executar.
- **Depurador (Debugger):** Ajuda a encontrar e corrigir erros no código.
- **Gerenciamento de Projetos:** Organiza todos os arquivos e recursos que fazem parte do seu projeto de software.
- **Ferramentas de Autocompletar e Sugestões:** Auxiliam na escrita do código mais rapidamente e com menos erros.

O ambiente de desenvolvimento utilizado foi o Visual Studio Community 2022, ambiente nativo do C# e outras ferramentas de desenvolvimento, uma versão gratuita da IDE Visual Studio, desenvolvida pela Microsoft. É uma das IDE's mais populares e poderosas para desenvolver software, especialmente para Windows. Explicando de uma forma melhor algumas das razões pela qual ela se torna tão especial: Contém suporte para várias linguagens, como C#, C++, Python, JavaScript, entre outras; Ferramentas de depuração avançadas facilitando encontrar e corrigir erros no seu código; Editor de código inteligente que oferece autocompletar, sugestões e destaca erros à medida que você escreve o seu código; Interface de arrastar e soltar facilitando a criação de interfaces gráficas (como com Windows Forms); Integração com banco de dados incluindo ferramentas para trabalhar facilmente com bancos de dados como o SQLite; Gerenciamento de versionamento que integra-se com sistemas de controle de versão, como o Git, para que você possa gerenciar alterações no seu código ao longo do tempo. Podemos pensar no Visual Studio Community como uma "Super-ferramenta" que reúne tudo o que você precisa para criar, testar e lançar programas,

tudo em um único só lugar. E o melhor de tudo isso é que se disponibiliza totalmente gratuito para uso individual, para propósitos educacionais e profissionais.

### 3.2. Interface gráfica

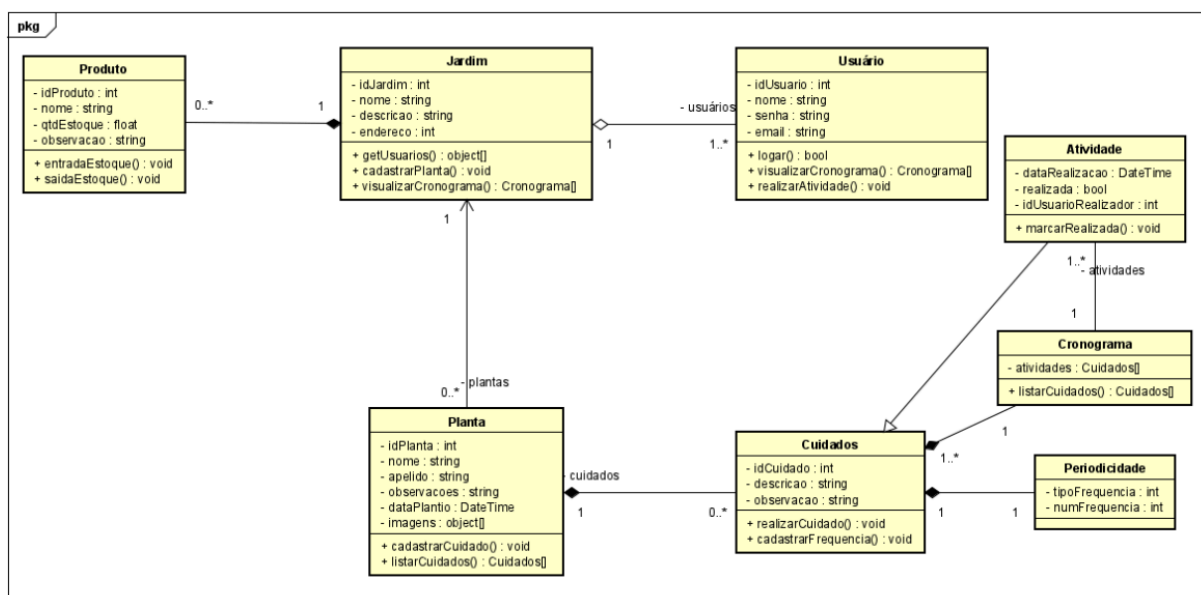
Nos dedicamos em proporcionar uma experiência agradável e intuitiva ao usuário, a interface visual do sistema deveria ser agradável e intuitiva como as demais aplicações já existentes no mercado. Pensando nisso, foi escolhida a tecnologia embarcada do C# .NET para criação de interfaces gráficas: O Windows Forms, uma plataforma fornecida também pela Microsoft que permite criar aplicativos de interface gráfica (GUI) para o sistema operacional Windows. Imagine que você quer criar um programa que tenha janelas, botões, caixas de texto e outros elementos que as pessoas possam clicar e interagir. Ele fornece as ferramentas e componentes necessários para construir essas interfaces visuais de forma mais fácil e rápida. É como um kit de construção para fazer programas que as pessoas usam no dia a dia.

A biblioteca proporciona rapidez e agilidade na geração de telas, combinado com a praticidade de visualização estilo *drag and drop*, onde o desenvolvedor pode arrastar os componentes e trabalhar com eles visualmente antes mesmo de compilar a aplicação.

### 3.3. Diagrama de Classes

A fim de endereçar o problema mais facilmente e entender quais as entidades responsáveis pelo funcionamento do sistema, foi produzido um diagrama de classes na Linguagem de Modelagem Unificada (UML).

O diagrama UML do MyGardenCare é denotado pela **Figura 2.** onde é possível observar as relações entre as classes lógicas da aplicação. O diagrama foi concebido com o auxílio da ferramenta Astah Community.



**Figura 2. Diagrama de Classes**

Após a definição dos requisitos e da elaboração dos casos de uso, o diagrama foi gerado com as seguintes classes: Jardim, Produto, Usuário, Planta, Cuidados, Cronograma, Periodicidade e Atividade. Cada classe possui seus atributos distintos e contribui para que os requisitos sejam atendidos corretamente.

### 3.4. Banco de Dados

Para o devido funcionamento do sistema é fundamental que o usuário possa entrar e sair da aplicação conforme suas necessidades sem perder dados. Para que seja possível o devido controle das informações, o sistema consta com um banco de dados relacional.

A tecnologia utilizada para esta solução foi o SQLite, podemos imaginá-lo como um grande arquivo de fichas onde você pode armazenar, organizar e acessar informações rapidamente. Popular por ser simples de usar, não requer uma instalação complicada e pode ser integrado diretamente em aplicativos, o que torna perfeito para projetos menores como o nosso ou aplicativos que precisam de um banco de dados local. É como um caderno onde você pode anotar e buscar informações facilmente sem precisar de um servidor grande e complexo. Um banco de dados leve e eficiente, ideal para aplicações desktop que não requerem um servidor de banco de dados robusto. Para a devida visualização dos dados foi utilizado também o SQLite Studio, uma ferramenta que permite que o desenvolvedor faça interações com o banco de dados por meio de queries, a fim de validar o devido funcionamento da aplicação.

As tabelas definidas foram estruturadas com base nas classes lógicas apresentadas no diagrama de classes, permitindo as relações descritas e a devida organização dos dados antes de seu processamento.

Para viabilizar o acesso aos dados dentro do projeto, foi utilizada a ferramenta *ADO.NET Entity Framework*. O *Entity Framework* utiliza a presença de *Object-Relational Mappers* (ORMs) e permite que a equipe de desenvolvimento siga filosofias do relacionamento de código com banco, a utilizada no projeto foi a *code-first*.

### 3.5. Arquitetura do projeto

Um projeto com a robustez de um gerenciador de jardins precisa possuir uma arquitetura condizente com sua proposta. A fim de organizar as implementações e facilitar as possíveis manutenções, o sistema foi dividido em quatro camadas diferentes: *UI*, *Application*, *Domain* e *Persistence*.

- **Camada de User Interface (UI):** É a camada responsável por conter toda a interface de usuário, ou seja, as telas de cadastro, o menu principal e todas as chamadas para as funcionalidades que o usuário solicitar.
- **Camada de *Application*:** Implementa as regras de negócio por meio dos *Services* e realiza chamadas para o domínio visando o acesso ao banco de dados. Nesta camada,



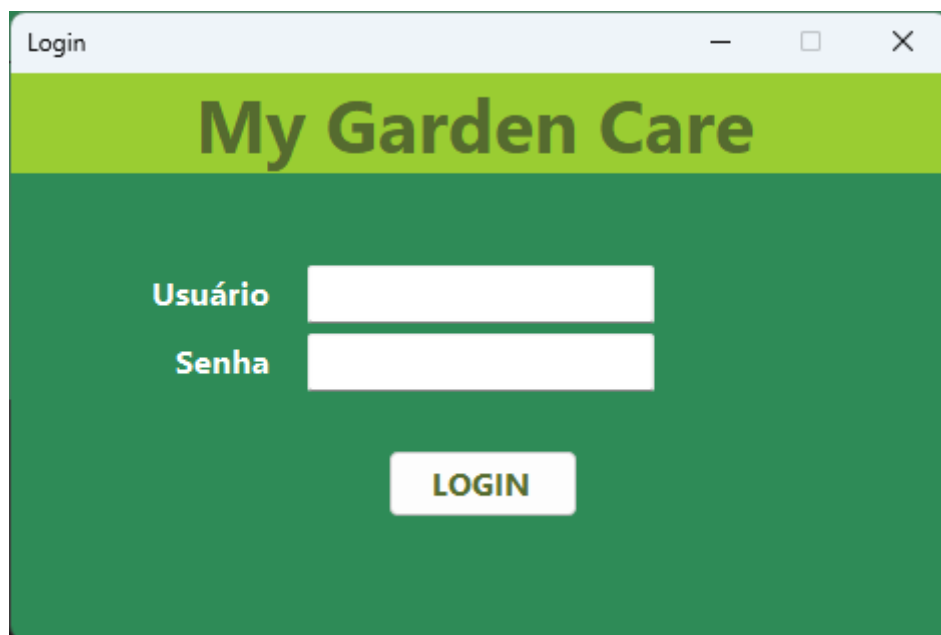
são definidas as *Controllers* que possuem as operações que as telas da UI podem chamar. Também é aqui que estão localizadas os *Data Transfer Objects* (DTOs), classes responsáveis por transportar informações por dentro do sistema.

- **Camada de *Domain*:** Implementa as classes relacionadas às tabelas no banco de dados. É nesta camada que os objetos são definidos e os mapeamentos são realizados.
- **Camada de *Persistence*:** Camada responsável por efetivar a relação entre código e banco de dados, por meio do contexto de banco de dados e das *migrations*, gerados pelo *Entity Framework*.

#### 4. Funcionamento do Software

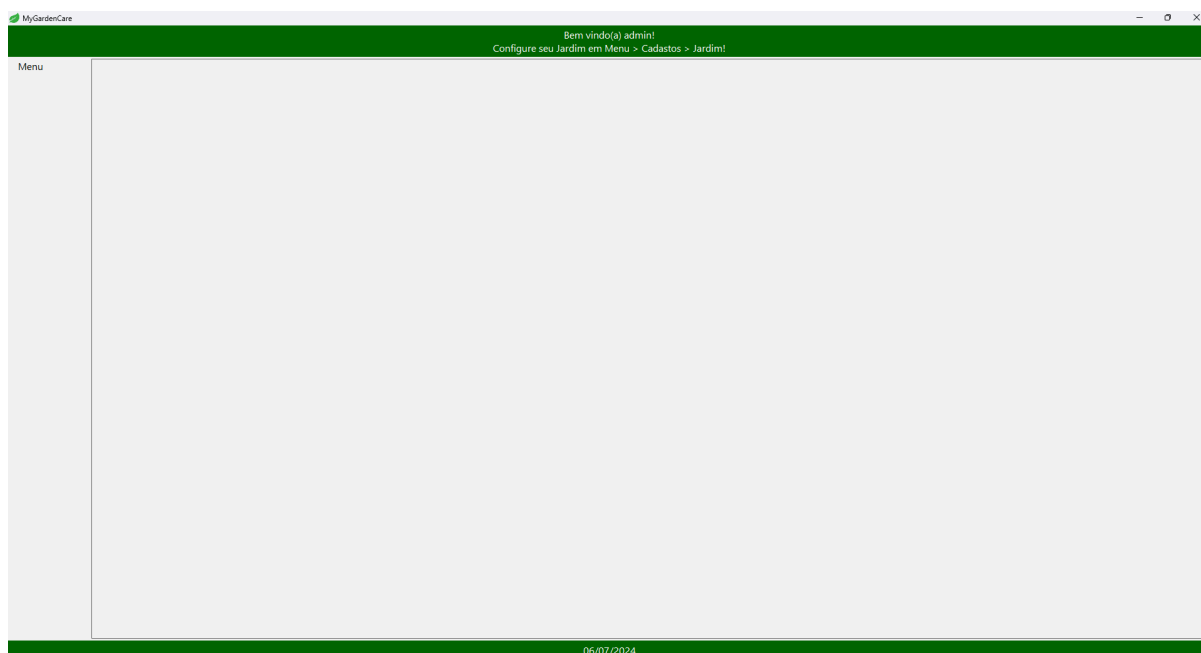
Nesta seção serão apresentadas as telas do sistema, bem como o fluxo de utilização do sistema.

Ao acessar o sistema, o usuário será apresentado à tela de login, onde é necessário informar usuário e senha de um usuário já cadastrado. Sendo o primeiro acesso ao sistema, deve-se fazer o processo de login com o usuário ‘admin’, reservado ao usuário administrador máximo do sistema.



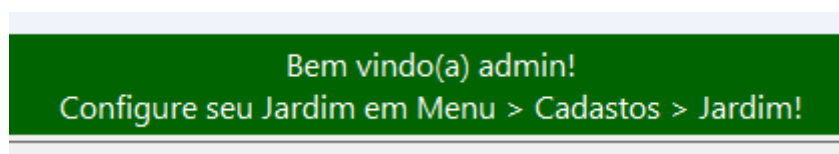
**Figura 3. Tela de login**

Após o processo de login bem sucedido, o usuário será apresentado ao menu principal, onde poderá navegar no menu lateral esquerdo entre as demais telas do sistema. Sendo o primeiro acesso ou o ambiente não possuir um jardim configurado, o sistema indicará para que o mesmo realize esta configuração.



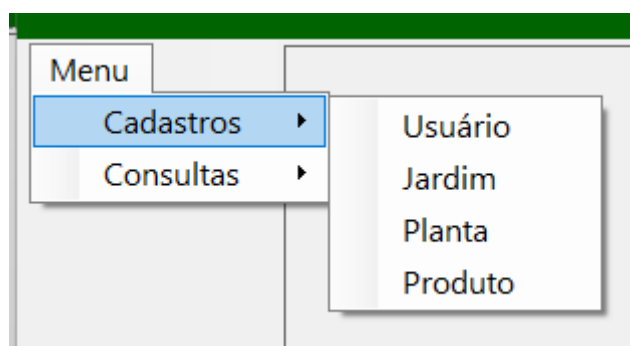
**Figura 3. Menu principal após o login**

Ao acessar o sistema pela primeira vez, ou não possuindo um Jardim configurado, o sistema apresentará o aviso da **Figura 4** ao usuário, indicando-o para fazer a configuração inicial.



**Figura 4. Indicação para que o usuário configure o jardim no cabeçalho**

Ao acessar o menu de opções, o usuário poderá escolher entre algumas opções para navegar entre o sistema. Inicialmente é permitido acessar o cadastro de usuários e o cadastro de jardim (antes da configuração inicial de jardim). Após a criação de um Jardim, já é possível acessar o cadastro de plantas e produtos.



**Figura 5. Opções do Menu de Cadastros**

Ao acessar a opção de Cadastros > Usuário, o utilizador poderá cadastrar um novo usuário, podendo informar os dados pessoais do usuário e configurar seu acesso. Este cadastro pode ser acessado antes mesmo de cadastrar o Jardim.

The screenshot shows a web application window titled 'MyGardenCare'. The header bar is green and contains the text 'Bem vindo(a) admin!'. On the left, there is a 'Menu' sidebar. The main content area has a green background with a light green header for the form titled 'CADASTRO DE USUÁRIO'. The form includes input fields for 'Nome', 'CPF', 'Telefone', 'Email', and 'Senha'. Below the 'Senha' field is a checkbox labeled 'Administrador'. At the bottom right of the form are two buttons: 'SALVAR' and 'FECHAR'. The footer of the application window displays the text 'Trabalhando no Jardim de Casa - 06/07/2024'.

**Figura 5. Tela de cadastro de usuários**

Ao acessar a opção de Cadastros > Jardim, o utilizador poderá configurar os dados do seu jardim, podendo informar um nome, descrição e endereço do jardim.

The screenshot shows a web application window titled 'MyGardenCare'. The header bar is green and contains the text 'Bem vindo(a) admin!'. On the left, there is a 'Menu' sidebar. The main content area has a green background with a light green header for the form titled 'CADASTRO DE JARDIM'. The form includes input fields for 'Nome', 'Descrição', and 'Endereço'. At the bottom right of the form are two buttons: 'SALVAR' and 'FECHAR'. The footer of the application window displays the text 'Trabalhando no Jardim de Casa - 06/07/2024'.

**Figura 5. Tela de cadastro de Jardim**

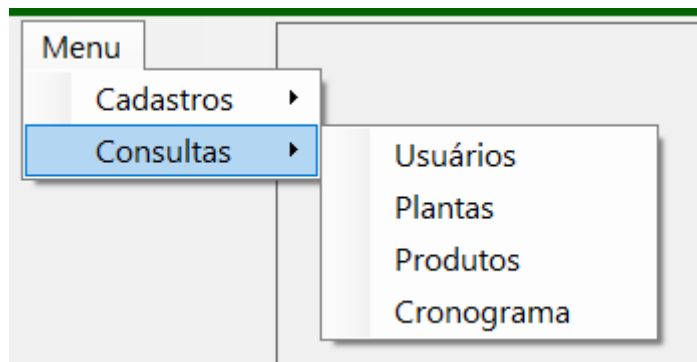
Ao acessar a opção de Cadastros > Planta, o utilizador poderá cadastrar uma nova planta, nesta tela é possível informar os dados de um planta (nome, apelido, observações e data de plantio), bem como os cuidados relacionados a esta planta. Na seção dos cuidados, é possível informar a descrição deste cuidado, bem como observações e a periodicidade deste cuidado. Os cuidados são listados em forma de tabela na parte inferior da página.

The screenshot shows a web application window titled 'MyGardenCare'. The main header is green with the text 'Bem vindo(a) admin!'. A left sidebar contains a 'Menu' button. The main content area has a green background and is titled 'CADASTRO DE PLANTAS' in a light green banner. Below the banner, there are input fields for 'Nome', 'Data do Plantio' (with a calendar icon and the date '03/07/2024'), 'Apelido', and 'Observações'. A section titled 'Cuidados' contains input fields for 'Descrição', 'Observações', and 'Periodicidade' (a dropdown menu). To the right of these fields are two buttons: 'Adicionar Cuidado' and 'Remover Cuidado'. Below these fields is a table with three columns: 'Descrição', 'Observações', and 'Periodicidade'. At the bottom right of the form are two buttons: 'SALVAR' and 'FECHAR'. The footer of the application shows the text 'Trabalhando no Jardim de Casa - 07/07/2024'.

Descrição	Observações	Periodicidade

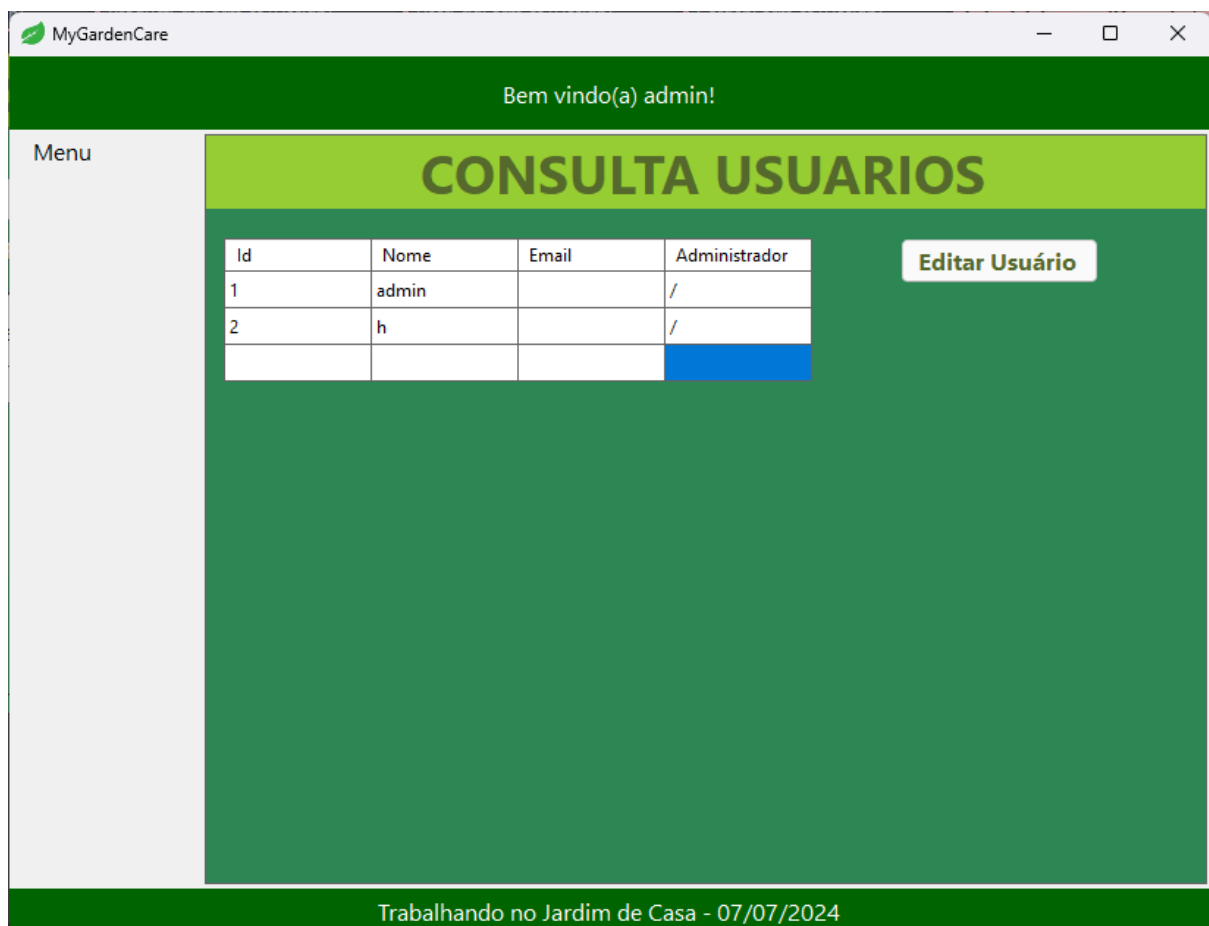
**Figura 6. Tela de Cadastro de Plantas**

Ao acessar o submenu Consultas, o usuário poderá acessar as telas de consulta do sistema: Consulta de usuários, consulta de plantas, consulta de produto e cronograma.



**Figura 7. Opções do Menu de Consultas**

Ao acessar a opção de Consultas > Usuários, o usuário poderá visualizar todos os usuários cadastrados no sistema. Também há a possibilidade de editar um usuário, utilizando o botão “Editar Usuário”.



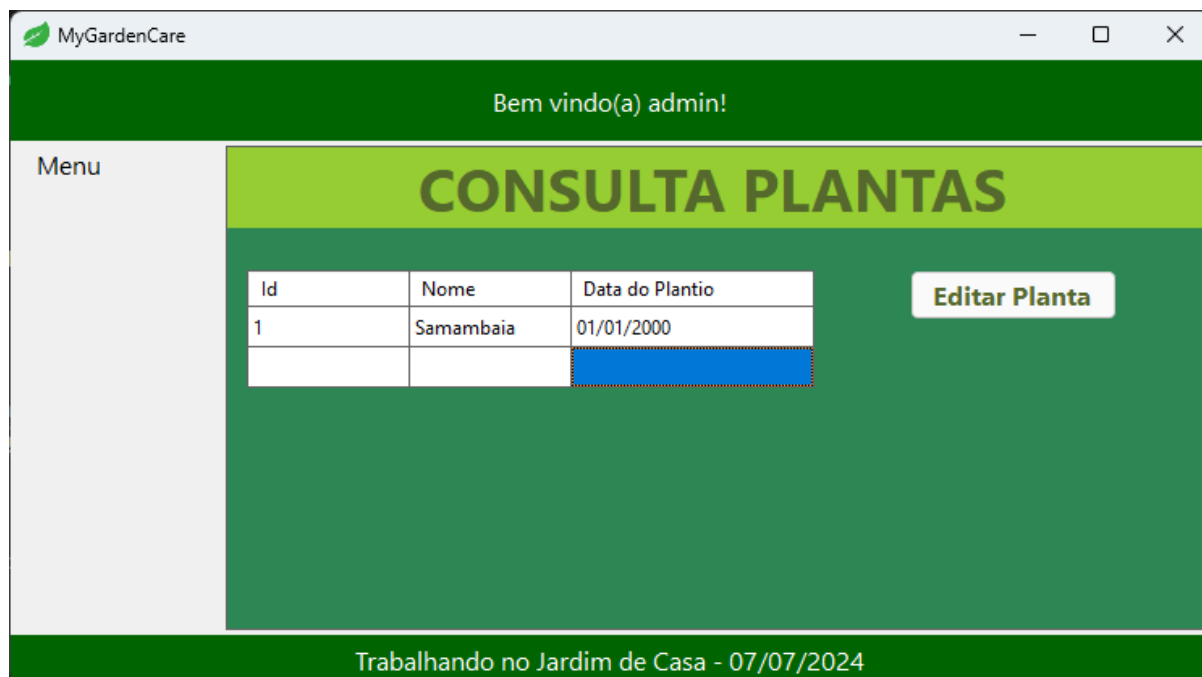
**Figura 8. Tela de consulta de usuários**

Em qualquer opção do menu de consultas, ao clicar no botão Editar, o usuário será apresentado com uma tela modal sobre o sistema, com as informações do registro selecionado exibidas na tela de cadastro.



**Figura 9.** Tela modal de cadastro de usuário após o usuário utilizar a opção “Editar Usuário” na tela de consulta de usuários.

Ao acessar a opção de Consultas > Plantas, o usuário poderá visualizar todas as plantas cadastradas no sistema. Também há a possibilidade de editar uma planta e seus cuidados, utilizando o botão “Editar Planta”.



**Figura 10.** Tela de Consulta de plantas

Ao acessar a opção de Consultas > Produtos, o usuário poderá visualizar todos os produtos cadastrados no sistema. Também há a possibilidade de editar um produto utilizando o botão “Editar Produto”.



**Figura 11. Tela de consulta de produtos**

Ao acessar a opção de Consultas > Cronograma, o usuário será apresentado à tela mais importante do sistema. Nesta tela é onde o usuário poderá visualizar todas as atividades a fazer em um determinado período, podendo visualizar as atividades a fazer e já feitas para aquele período ou até listar somente as atividades feitas, para verificar em forma de relatório os dados salvos no sistema. Uma atividade já feita mostrará o usuário realizador e a data de realização, bem como o checkbox “Feito” marcado. Para marcar uma atividade como concluída o usuário deverá marcar o checkbox “Feito”.

MyGardenCare

Bem vindo(a) admin!

Menu

## CRONOGRAMA

Período:  a

Planta: 
☒ Somente Feitas

Feito	Planta	Atividade	Observações	Usuario	Data Realizacao
<input type="checkbox"/>					

Trabalhando no Jardim de Casa - 07/07/2024

**Figura 12. Tela de Cronograma**

## 5. Considerações finais

Durante o desenvolvimento do projeto foi possível aplicar uma enorme quantidade de conceitos que temos estudado durante as aulas. O projeto contou desde o começo com uma ampla utilização do que se entende por Engenharia de Software, por meio dos processos de levantamento de requisitos, elaboração de casos de uso e diagramas de classe, até a finalização com mockups e o desenvolvimento do projeto. A partir da liberdade proporcionada para: identificar um problema, gerar uma ideia de solução e de fato implementar foi possível entender o propósito daquilo que estudamos durante os anos de curso.

Na parte de codificação foi possível identificar conceitos e práticas já comentadas em experiências prévias porém com a necessidade um grau de seriedade maior, afinal é necessário produzir uma solução funcional no fim das contas. Muitas das práticas exercidas são padrão no ambiente profissional e sem dúvidas essa experiência é edificante para os desenvolvedores do projeto.

O desenvolvimento do Projeto Temático I foi um desafio para o Grupo 4 em diversos aspectos e proporcionou para os componentes da equipe uma experiência de colaboração entre pensamentos distintos, algo muito comum no mercado de trabalho. Com toda certeza foram tiradas valiosas lições dos erros cometidos e também ideias novas para a solução de problemas relacionados ao desenvolvimento de software.



## **6. Referências**

PRESSMAN, Roger S.; MAXIM, Bruce R. Engenharia de software. Grupo A, 2021. E-book. ISBN 9786558040118. Disponível em:

<https://integrada.minhabiblioteca.com.br/#/books/9786558040118/>.

LEDUR, Cleverson L. Desenvolvimento de sistemas com C#. Grupo A, 2018. E-book. ISBN 9788595023147. Disponível em:

<https://integrada.minhabiblioteca.com.br/books/9788595023147/>.