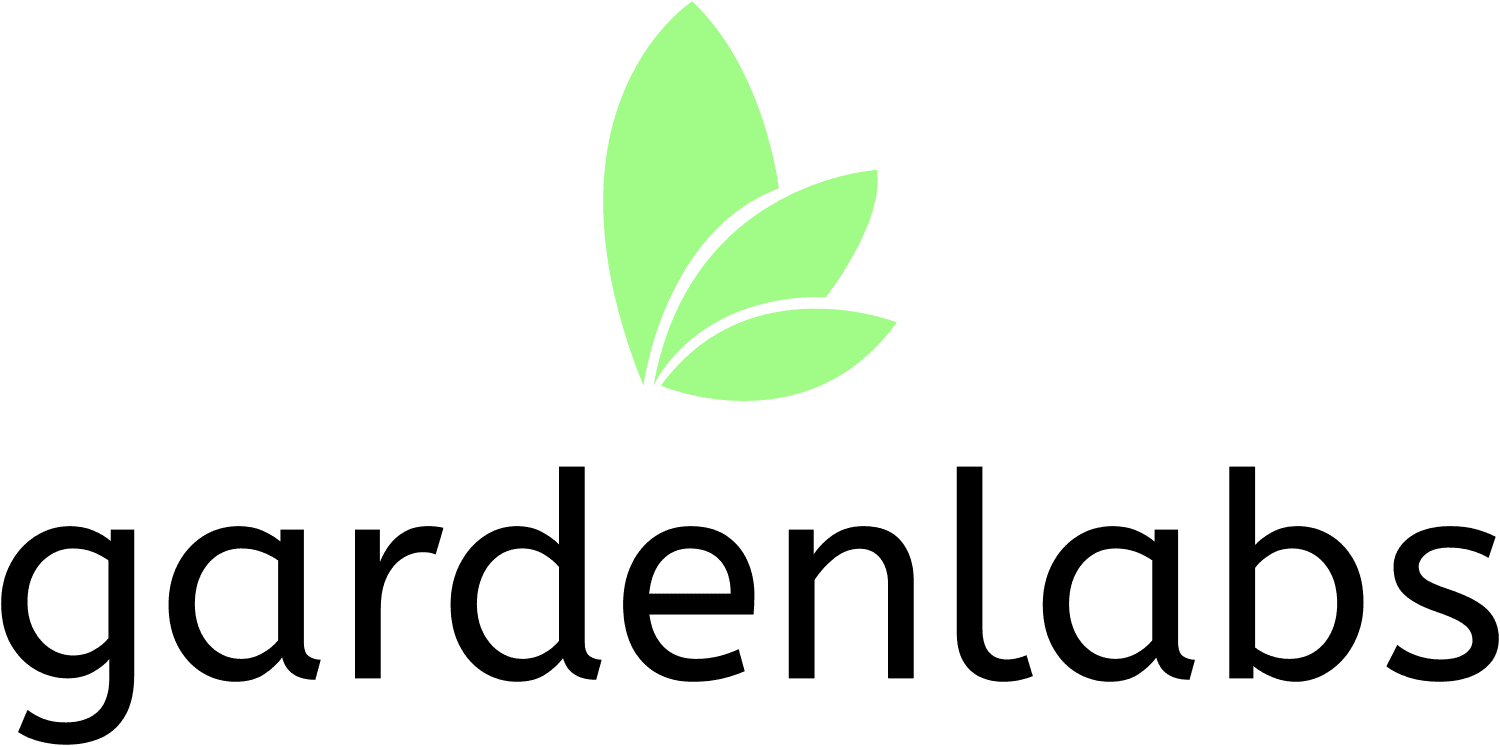
|  |  |  |  |
| --- | --- | --- | --- |
|  | | Curso de TeSP em  Programação de Sistemas de Informação  Projeto em  Sistemas de Informação (PSI) | |
|  |  | |  |
| **Ano Letivo 2024/2025** |  | | **2º Ano, 1º Semestre** |
|  | | | |
| **Projeto em Sistema de Informação** | | | |



**Relatório de Especificação do SI**

***GardenLabs***

|  |  |
| --- | --- |
|  |  |
| **Grupo:** PL1-F | **Data de entrega:** 2024-02-?? |
| **Nº 2231652** | Diogo Nunes Ruivo |
| **Nº 2231752** | Duarte André Amado Fragoso |
| **Nº 2231659** | Tomás Azenha Monteiro |

ÍNDICE

[Índice de Figuras Erro! Marcador não definido.](#_Toc182944894)

[Índice de Tabelas 3](#_Toc182944895)

[1 Introdução 4](#_Toc182944896)

[1.1 Sumário executivo 4](#_Toc182944897)

[2 *Roles* e Requisitos do SI 5](#_Toc182944898)

# Índice de Tabelas

[Tabela 5 – Roles dos diferentes end-users do SI 5](#_Toc182945086)

[Tabela 6 – Requisitos funcionais Front-Office 6](#_Toc182945087)

[Tabela 7 – Requisitos funcionais Back-Office 7](#_Toc182945088)

[Tabela 9 – Cruzamento dos requisitos funcionais e respetivos roles 8](#_Toc182945089)

[Tabela 10 – Requisitos Não funcionais 11](#_Toc182945090)

# Introdução

Neste projeto, foram planeadas uma *web app* e uma *mobile app* no âmbito das disciplinas de Acesso Móvel a Sistemas de Informação, Plataformas de Sistemas de Informação, Serviços e Interoperabilidade de Sistemas e Projeto em Sistemas de Informação, sendo *GardenLabs* o nome atrbuído. Este foi desenvolvido utilizando o Android Studio e a *yii2 framework* (utilizada no desenvolvimento da *web app*), sendo planeada com recurso à metolodogia ágil *Scrum*.

Este projeto tem como objetivos principais produzir uma loja web operacional com uma grande variedade de produtos categorizados (Front-office e Back-office) e uma app mobile que permite agendar os serviços de jardinagem (Semanalmente, mensalmente, anualmente, etc...).

Também é necessário desenvolver uma API de serviços HTTP para o cliente mobile e uma base de dados relacional de suporte a todos estes elementos. A aplicação móvel deverá comunicar com a API para o cumprimento dos respetivos requisitos funcionais e não funcionais.

Este projeto será desenvolvido em 8 *Sprints*, tendo cada uma a duração de duas semanas. Em cada uma destas *Sprints*, é efetuado o *Sprint Planning* e o *Sprint Retrospective*, sendo também avaliado o progresso recorrendo ao *weekly scrum*.

Para conseguir alcançar um desenvolvimento em equipa mais eficaz, foi utilizado o *GitHub*, permitindo incrementar o produto de forma mais ágil e também foi utilizado o *Jira* no apoio do *Scrum* do projeto.

## Sumário executivo

Na presente secção, é feita uma pequena contextualização e descrição dos principais objetivos deste projeto. Na seguinte secção, é feita a especificação do sistema, ou seja, a sua definição da lógica de negócio, a análise de impacto, a análise concorrencial e a definição dos Roles e Requisitos de SI. Posteriormente, são descritos todos os *User Stories*, é apresentado o diagrama de entidade-relacionamento da base de dados partilhada entre a *web app* e a *mobile app*, são divulgados os *Wireframes* de todas as páginas e *screens* da *web app* e *mobile app* e, por fim, são listados todos os serviços que o sistema de incorporar.

# *Roles* e Requisitos do SI

As próximas tabelas descrevem os roles e requisitos do sistema.

Tabela – Roles dos diferentes end-users do SI

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Role | Descrição detalhada | Back-office | Front-office | App mobile |
| Clientes | No front-office, o cliente interage diretamente com a interface da app mobile e da loja online, ao poder navegar por diferentes serviços de jardinagem, visualizar informações sobre os profissionais disponíveis, agendar serviços, e comprar produtos da loja online. A experiência do cliente é crucial, com um design intuitivo que facilita a navegação, seleção de serviços, e finalização de compras. O cliente pode também receber notificações sobre lembretes de serviços, promoções e dicas de jardinagem. |  | x | x |
| Administradores | Os administradores são responsáveis pela manutenção e atualização da *web app* e *app mobile*, implementando novas funcionalidades e corrigindo bugs, podem adicionar novos produtos, atualizar preços e monitorar o inventário disponível, eles podem também analisar dados de uso para entender melhor as preferências dos clientes e otimizar a experiência. A comunicação com a equipa de suporte é fundamental para resolver problemas rapidamente e garantir a satisfação do cliente. | x | x | x |
| Prestadores de Serviço | Os jardineiros contratados interagem diretamente com a app mobile e, em alguns casos, com o website. Eles podem visualizar agendamentos de serviços, ter acesso a detalhes sobre as tarefas a ser realizadas e comunicar com os clientes para confirmar horários ou esclarecer dúvidas. Além disso, podem receber feedback dos clientes e ajustar seus serviços conforme necessário. |  |  | x |
| Gestores de loja e Serviços | No back-office, os gestores de loja e serviços têm acesso a ferramentas de gestão de dados, onde monitorizam agendamentos, controlam o *stock* de produtos, e analisam o desempenho das vendas. Eles gerenciam o CRM (gestão de relacionamento com o cliente) e utilizam análises para melhorar as estratégias de marketing e a eficiência operacional. | x |  |  |

Tabela – Requisitos funcionais Front-Office

|  |  |  |
| --- | --- | --- |
| # | Requisito | Prioridade |
| RF-FO-01 | O sistema deve disponibilizar uma página de carrinho de compras. | Alta |
| RF-FO-02 | O sistema deve apresentar os produtos da loja em *cards*. | Média |
| RF-FO-03 | O sistema deve permitir guardar produtos nos favoritos. | Média |
| RF-FO-04 | O sistema deve permitir ver os detalhes de cada produto. | Alta |
| RF-FO-05 | O sistema deve permitir aceder a todas as páginas através da página principal. | Média |
| RF-FO-06 | O sistema deve permitir selecionar a quantidade desejada na compra de produtos. | Alta |
| RF-FO-07 | O sistema deve permitir filtrar os produtos através de categorias. | Média |
| RF-FO-08 | O sistema deve permitir filtrar os produtos com base no resultado da barra de pesquisa. | Média |
| RF-FO-09 | O sistema deve disponibilizar um *check-out* do carrinho de comprasque permita efetuar o pagamento dos produtos escolhidos. | Alta |
| RF-FO-10 | O sistema deve permitir escolher o método de expedição ao efetuar o *check-out* do carrinho de compras. | Alta |
| RF-FO-11 | O sistema deve permitir escolher o método de pagamento ao efetuar o *check-out* do carrinho de compras. | Alta |
| RF-FO-12 | O sistema deve fornecer uma página de histórico de compras a cada utilizador. | Alta |

Tabela – Requisitos funcionais Back-Office

|  |  |  |
| --- | --- | --- |
| # | Requisito | Prioridade |
| RF-BO-01 | O sistema deve atualizar a quantidade de stock existente ao apresentar os produtos na loja. | Alta |
| RF-BO-02 | O sistema deve validar a autenticação de utilizadores na plataforma. | Alta |
| RF-BO-03 | O sistema deve permitir adicionar novos produtos à loja. | Alta |
| RF-BO-04 | O sistema deve permitir alterar as características de produtos existentes na loja. | Alta |
| RF-BO-05 | O sistema deve permitir remover produtos existentes da loja. | Alta |
| RF-BO-06 | O sistema deve permitir a alteração das *passwords* dos utilizadores. | Alta |
| RF-BO-07 | O sistema deve permitir processar todas as avaliações (1-5) dos produtos e conseguir estabelecer uma média para cada um. | Média |
| RF-BO-08 | O sistema deve conseguir processar o *check-out* do carrinho de compras após escolher os produtos desejados. | Alta |
| RF-BO-09 | O sistema deve permitir adicionar novos métodos de expedição na efetuação do *check-out* do carrinho de compras. | Alta |
| RF-BO-10 | O sistema deve permitir editar métodos de expedição existentes na efetuação do *check-out* do carrinho de compras. | Alta |
| RF-BO-11 | O sistema deve permitir remover métodos de expedição existentes na efetuação do *check-out* do carrinho de compras. | Alta |
| RF-BO-12 | O sistema deve permitir adicionar novos métodos de pagamento na efetuação do *check-out* do carrinho de compras. | Alta |
| RF-BO-13 | O sistema deve permitir editar métodos de pagamento existentes na efetuação do *check-out* do carrinho de compras. | Alta |
| RF-BO-14 | O sistema deve permitir remover métodos de pagamento existentes na efetuação do *check-out* do carrinho de compras. | Alta |

Tabela – Cruzamento dos requisitos funcionais e respetivos roles

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # | Requisito | Administradores | Clientes | Gestores | Prestadores de Serviço |
| RF-FO-01 | O sistema deve disponibilizar uma página de carrinho de compras. | x | x |  |  |
| RF-FO-02 | O sistema deve apresentar os produtos da loja em *cards*. | x | x | x |  |
| RF-FO-03 | O sistema deve permitir guardar produtos nos favoritos. | x | x |  |  |
| RF-FO-04 | O sistema deve permitir ver os detalhes de cada produto. | x | x | x |  |
| RF-FO-05 | O sistema deve permitir aceder a todas as páginas através da página principal. | x | x |  |  |
| RF-FO-06 | O sistema deve permitir selecionar a quantidade desejada na compra de produtos. | x | x |  |  |
| RF-FO-07 | O sistema deve permitir filtrar os produtos através de categorias. | x | x |  |  |
| RF-FO-08 | O sistema deve permitir filtrar os produtos com base no resultado da barra de pesquisa. | x | x |  |  |
| RF-FO-09 | O sistema deve disponibilizar um *check-out* do carrinho de comprasque permita efetuar o pagamento dos produtos escolhidos. | x | x |  |  |
| RF-FO-10 | O sistema deve permitir escolher o método de expedição ao efetuar o *check-out* do carrinho de compras. | x | x |  |  |
| RF-FO-11 | O sistema deve permitir escolher o método de pagamento ao efetuar o *check-out* do carrinho de compras. | x | x |  |  |
| RF-FO-12 | O sistema deve fornecer uma página de histórico de compras a cada utilizador. | x | x |  |  |
| RF-BO-01 | O sistema deve atualizar a quantidade de stock existente ao apresentar os produtos na loja. | x |  | x |  |
| RF-BO-02 | O sistema deve validar a autenticação de utilizadores na plataforma. | x | x |  |  |
| RF-BO-03 | O sistema deve permitir adicionar novos produtos à loja. | x |  | x |  |
| RF-BO-04 | O sistema deve permitir alterar as características de produtos existentes na loja. | x |  | x |  |
| RF-BO-05 | O sistema deve permitir remover produtos existentes da loja. | x |  | x |  |
| RF-BO-06 | O sistema deve permitir a alteração das *passwords* dos utilizadores. | x | x |  |  |
| RF-BO-07 | O sistema deve permitir processar todas as avaliações (1-5) dos produtos e conseguir estabelecer uma média para cada um. | x |  |  |  |
| RF-BO-08 | O sistema deve conseguir processar o *check-out* do carrinho de compras após escolher os produtos desejados. | x |  |  |  |
| RF-BO-09 | O sistema deve permitir adicionar novos métodos de expedição na efetuação do *check-out* do carrinho de compras. | x |  | x |  |
| RF-BO-10 | O sistema deve permitir editar métodos de expedição existentes na efetuação do *check-out* do carrinho de compras. | x |  | x |  |
| RF-BO-11 | O sistema deve permitir remover métodos de expedição existentes na efetuação do *check-out* do carrinho de compras. | x |  | x |  |
| RF-BO-12 | O sistema deve permitir adicionar novos métodos de pagamento na efetuação do *check-out* do carrinho de compras. | x |  | x |  |
| RF-BO-13 | O sistema deve permitir editar métodos de pagamento existentes na efetuação do *check-out* do carrinho de compras. | x |  | x |  |
| RF-BO-14 | O sistema deve permitir remover métodos de pagamento existentes na efetuação do *check-out* do carrinho de compras. | x |  | x |  |

Tabela – Requisitos Não funcionais

|  |  |  |  |
| --- | --- | --- | --- |
| # | Requisito | Tipo | Prioridade |
| RNF-01 | O sistema deve ter uma interface intuitiva e fácil de utilizar. | Usabilidade | Alta |
| RNF-02 | A interface gráfica do sistema deve ter proporções que permitam a exibição correta de todo o texto. | Usabilidade | Média |
| RNF-03 | O sistema deve fornecer feedback claro e imediato ao utilizador sobre o estado das ações que este realiza. | Usabilidade | Média |
| RNF-04 | O sistema deve ser tolerante e deve conseguir responder a eventuais falhas. | Fiabilidade | Alta |
| RNF-05 | O sistema deve realizar backups dos dados frequentemente para evitar perda de dados | Fiabilidade | Alta |
| RNF-06 | O sistema deve ter capacidade de tolerar uma grande quantidade de pedidos | Fiabilidade | Média |
| RNF-07 | O sistema deve codificar os dados dos utilizadores para manter a segurança dos dados pessoais | Segurança | Alta |
| RNF-08 | O sistema deve assegurar proteção contra *cross-site-scripting.* | Segurança | Alta |
| RNF-09 | O sistema deve limitar o acesso a dados sensíveis de utilizadores apenas a pessoas autorizadas com permissões específicas. | Segurança | Alta |
| RNF-10 | O sistema deve conseguir responder às solicitações dos clientes de forma rápida e eficaz. | Eficiência | Alta |
| RNF-11 | O sistema deve ser otimizado, o uso de recursos deve ser minimizado e o desempenho maximizado. | Eficiência | Alta |
| RNF-12 | O sistema deve conseguir suportar as várias informações de cada cliente de forma eficaz. | Eficiência | Alta |
| RNF-13 | O sistema deve estar disponível durante 98% do seu tempo de funcionamento diário. | Disponibilidade | Alta |
| RNF-14 | O sistema deve ser projetado de modo a minimizar o impacto de falhas individuais e garantir uma alta disponibilidade do serviço. | Disponibilidade | Alta |
| RNF-15 | O sistema deve ter manutenções programadas durante períodos em que haja baixa atividade por parte dos utilizadores. | Disponibilidade | Média |
| RNF-16 | O sistema deve ser compatível com os diversos sistemas operativos existentes. | Ambiente | Alta |
| RNF-17 | O sistema deve ser compatível com diferentes versões de sistemas operativos que possam ser usadas. | Ambiente | Alta |
| RNF-18 | O sistema deve ser compatível com diferentes sistemas de hardware (Computadores, telemóveis, etc...). | Ambiente | Alta |
| RNF-19 | O sistema deve ser projetado de forma que os testes durante a fase de desenvolvimento sejam fáceis de executar. | Desenvolvimento | Média |
| RNF-20 | O sistema deve ser projetado de forma que o código esteja bem organizado para facilitar possíveis manutenções. | Desenvolvimento | Alta |
| RNF-21 | O sistema deve seguir o padrão MVC para promover a modularidade. | Desenvolvimento | Alta |