



# Aplicação Móvel Balvia Para Dispensador de Medicamento

**Yúri dos Ramos Gué - a39358**

Relatório de Projeto apresentado à Escola Superior de Tecnologia e Gestão para  
obtenção do Grau de Mestre em Informática.

Trabalho realizado sob a orientação de:  
Prof. José Eduardo Fernandes

Bragança  
Maio 2025





# Aplicação Móvel Balvia Para Dispensador de Medicamento

**Yúri dos Ramos Gué - a39358**

Relatório de Projeto apresentado à Escola Superior de Tecnologia e Gestão para  
obtenção do Grau de Mestre em Informática.

Trabalho realizado sob a orientação de:

Prof. José Eduardo Fernandes

Este documento não inclui as sugestões feitas pelo conselho de administração.

Bragança

Maior 2025



# Dedicatória

Dedico este trabalho aos meus pais, que sempre me apoiaram incondicionalmente e me inspiraram a perseguir os meus objectivos com determinação.

Agradeço aos meus professores e orientadores académicos, pelo conhecimento partilhado e pelo apoio constante.

Aos meus orientadores e colegas da Balvia Ecosystems, que foram fundamentais para o desenvolvimento deste projecto, surgido no âmbito do estágio de inovação que realizei na empresa. A sua orientação, apoio e incentivo foram essenciais para a concretização deste trabalho.

Por fim, a todos os amigos que estiveram ao meu lado, oferecendo amizade e motivação ao longo desta jornada, deixo o meu sincero agradecimento.

# Agradecimentos

Este trabalho foi realizado com o apoio da Balvia Ecosystems, onde surgiu como parte do estágio de inovação. Gostaria de expressar a minha sincera gratidão a todos os que contribuíram para o sucesso deste projeto.

Agradeço ao Instituto Politécnico de Bragança, em especial ao Professor José Eduardo Fernandes, pela sua orientação académica, conhecimento e apoio contínuo.

Gostaria de manifestar a minha gratidão ao Bruno Costa, supervisor na Balvia Ecosystems, cuja orientação e supervisão foram essenciais para o desenvolvimento deste projeto. O meu agradecimento estende-se também ao Professor Rui Pedro Lopes, pelas suas valiosas contribuições, ao meu colega Isaac Marcelino, que esteve comigo ao longo desta jornada, e aos demais colegas que contribuíram para o sucesso deste trabalho.

Expresso a minha profunda gratidão à minha família, em especial ao meu pai Arlindo Gué, à minha mãe Nira Gué, aos meus irmãos Yara Gué, Arlito Gué e Recintia Gué, e à minha namorada Cleyde Pires dos Santos, pelo apoio incondicional, carinho e incentivo constante.

Aos meus amigos e demais familiares, que estiveram comigo nos momentos mais desafiantes, oferecendo palavras de encorajamento e apoio emocional, deixo o meu sincero agradecimento.

Sem o suporte e o envolvimento de todos vós, este trabalho não teria sido possível.

# Resumo

O presente relatório aborda o desenvolvimento de uma aplicação móvel que facilita a gestão e administração de medicamentos, oferecendo uma solução integrada e inteligente para pacientes, cuidadores e profissionais de saúde. A aplicação permite o acesso remoto a dados em tempo real, o envio de notificações e a emissão de alertas no momento da toma da medicação, promovendo assim a adesão ao tratamento e melhorando a qualidade de vida dos utilizadores. Denominado Balvia, o projeto combina um dispensador inteligente com uma plataforma digital, possibilitando o acompanhamento remoto e simplificando a administração de medicamentos. A aplicação disponibiliza quatro perfis de utilizadores, com níveis de acesso e funcionalidades adaptadas: paciente (pessoa que toma a medicação), cuidador informal (familiar ou pessoa próxima que acompanha o paciente), cuidador formal (profissional de saúde ligado a empresas especializadas) e empresa parceira (prestadores de serviços de apoio aos pacientes). A aplicação oferece ainda funcionalidades específicas para os cuidadores, como o acesso a calendários e rotas otimizadas para o reabastecimento de dispensadores localizados nas proximidades. Desta forma, Balvia propõe-se a revolucionar a gestão de medicamentos, garantindo uma experiência integrada e facilitando a comunicação entre os diversos intervenientes no processo terapêutico.

**Palavras-chave:** Aplicação Móvel, Gestão de Medicamento, Administração de Medicamento, Paciente, Cuidadores, Suporte aos Cuidadores, Alertas de Medicamento, Acompanhamento Remoto.

# Abstract

This report addresses the development of a mobile application that facilitates the management and administration of medicines, offering an integrated and intelligent solution for patients, caregivers and health professionals. The application allows remote access to real-time data, sending notifications and issuing alerts at the time of medication intake, thus promoting treatment adherence and improving users' quality of life. Called Balvia, the project combines an intelligent dispenser with a digital platform, enabling remote monitoring and simplifying drug administration. The application provides four user profiles, with access levels and adapted functionalities: patient (person taking medication), informal caregiver (family or close person accompanying the patient), formal caregiver (health professional connected to specialized companies) and partner company (providers of patient support services). The application also offers specific features for caregivers, such as access to calendars and routes optimized for replenishment of dispensers located nearby. In this way, Balvia proposes to revolutionize the management of medicines, ensuring an integrated experience and facilitating communication between the various stakeholders in the therapeutic process.

**Keywords:** Mobile Application, Medication Management, Drug Administration, Patient, Caregivers, Caregiver Support, Drug Alerts, Remote Monitoring.

# Conteúdo

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Introdução</b>                                  | <b>1</b> |
| 1.1      | Enquadramento . . . . .                            | 1        |
| 1.2      | Objetivo . . . . .                                 | 2        |
| 1.3      | Estrutura do Relatório . . . . .                   | 2        |
| 1.3.1    | Capítulo 1: Introdução . . . . .                   | 3        |
| 1.3.2    | Capítulo 2: Estado da Arte e Tecnologias . . . . . | 3        |
| 1.3.3    | Capítulo 3: Requisitos e Arquitetura . . . . .     | 3        |
| 1.3.4    | Capítulo 4: Desenvolvimento . . . . .              | 4        |
| 1.3.5    | Capítulo 5: Conclusão . . . . .                    | 4        |
| <b>2</b> | <b>Estado de Arte</b>                              | <b>5</b> |
| 2.1      | Introdução . . . . .                               | 5        |
| 2.2      | Trabalhos relacionados . . . . .                   | 5        |
| 2.2.1    | MediSafe . . . . .                                 | 6        |
| 2.2.2    | MyTherapy . . . . .                                | 6        |
| 2.2.3    | CareZone . . . . .                                 | 7        |
| 2.2.4    | SapoMed . . . . .                                  | 7        |
| 2.2.5    | Mediminder . . . . .                               | 7        |
| 2.3      | Comparação e Limitações Identificadas . . . . .    | 8        |
| 2.4      | Tecnologias . . . . .                              | 8        |
| 2.4.1    | Linguagem Dart . . . . .                           | 9        |

|          |   |           |
|----------|---|-----------|
| 2.4.2    | Framework Flutter . . . . .                       | 9         |
| 2.4.3    | SQLite . . . . .                                  | 9         |
| 2.4.4    | FlutterBlue BLE Plus . . . . .                    | 10        |
| 2.4.5    | Trello . . . . .                                  | 10        |
| 2.4.6    | Postman . . . . .                                 | 10        |
| 2.4.7    | Xcode . . . . .                                   | 11        |
| 2.4.8    | DataGrip . . . . .                                | 11        |
| 2.4.9    | Android Studio . . . . .                          | 11        |
| 2.4.10   | Visual Paradigm Community . . . . .               | 12        |
| 2.4.11   | Overleaf . . . . .                                | 12        |
| 2.4.12   | Mendeley . . . . .                                | 13        |
| 2.5      | Conclusão . . . . .                               | 13        |
| <b>3</b> | <b>Requisitos e Arquitetura</b>                   | <b>15</b> |
| 3.1      | Especificação e Validação de Requisitos . . . . . | 15        |
| 3.1.1    | Especificação de Requisitos . . . . .             | 16        |
| 3.1.2    | Validação de Requisitos . . . . .                 | 16        |
| 3.2      | Requisitos . . . . .                              | 16        |
| 3.2.1    | Diagrama de Casos de Uso . . . . .                | 17        |
| 3.2.2    | Histórias de Utilizador . . . . .                 | 21        |
| 3.2.3    | Diagrama de Domínio . . . . .                     | 27        |
| 3.3      | Arquitetura do Sistema . . . . .                  | 28        |
| 3.4      | Conclusão . . . . .                               | 31        |
| <b>4</b> | <b>Desenvolvimento</b>                            | <b>33</b> |
| 4.1      | Arquitetura da Aplicação . . . . .                | 34        |
| 4.1.1    | Estrutura Modular . . . . .                       | 34        |
| 4.1.2    | Padrão de Arquitetura . . . . .                   | 35        |
| 4.1.3    | Gestão de Estado . . . . .                        | 35        |
| 4.1.4    | Persistência de Dados . . . . .                   | 35        |

|          |  |           |
|----------|--|-----------|
| 4.1.5    | Comunicação com Dispositivos Bluetooth . . . . .                   | 36        |
| 4.1.6    | Navegação e Fluxo da Aplicação . . . . .                           | 36        |
| 4.2      | Mockups e Protótipos . . . . .                                     | 36        |
| 4.3      | Funcionalidades Implementadas . . . . .                            | 40        |
| 4.4      | Código Desenvolvido . . . . .                                      | 45        |
| 4.4.1    | Provider para Gestão de Autenticação . . . . .                     | 45        |
| 4.4.2    | Estrutura da Mensagem para Comunicação com o Dispensador . . . . . | 47        |
| 4.4.3    | Conexão e Comunicação via Bluetooth . . . . .                      | 48        |
| 4.4.4    | Serviço de Sincronização de Dados . . . . .                        | 50        |
| 4.4.5    | Verificação e Disparo de Alarmes . . . . .                         | 51        |
| 4.5      | Dificuldades Encontradas e Soluções . . . . .                      | 53        |
| 4.6      | Validação Técnica e Resultados . . . . .                           | 55        |
| 4.7      | Conclusão . . . . .  | 56        |
| <b>5</b> | <b>Conclusão</b>   | <b>57</b> |

# Lista de Figuras

|      |   |    |
|------|---|----|
| 3.1  | Diagrama de Casos de Uso: Autenticação e Gestão de Contas. . . . .  | 18 |
| 3.2  | Diagrama de Casos de Uso Subsistema Balvia Cliente. . . . .         | 19 |
| 3.3  | Diagrama de Casos de Uso Subsistema Balvia Empresa. . . . .         | 21 |
| 3.4  | Diagrama de Domínio . . . . .                                       | 29 |
| 3.5  | Arquitetura da Solução . . . . .                                    | 31 |
|      |   |    |
| 4.1  | Mockup de Página Inicial. . . . .                                   | 37 |
| 4.2  | Mockup de Página de Registo. . . . .                                | 37 |
| 4.3  | Mockup do Menu Lateral. . . . .                                     | 38 |
| 4.4  | Mockup de Página de Calendário. . . . .                             | 38 |
| 4.5  | Mockup de Página de Reabastecimento. . . . .                        | 39 |
| 4.6  | Mockup da Etapa de Reabastecimento de um Módulo Específico. . . . . | 39 |
| 4.7  | Mockup do Reabastecimento com Instruções de Encerramento. . . . .   | 40 |
| 4.8  | Mockup de Alarme para Notificação da Toma do Medicamento. . . . .   | 40 |
| 4.9  | Interface Inicial. . . . .  | 41 |
| 4.10 | Interface de Login. . . . .   | 41 |
| 4.11 | Interface de Registo. . . . .                                       | 42 |
| 4.12 | Interface Dashboard. . . . .  | 42 |
| 4.13 | Interface Menu lateral. . . . .                                     | 43 |
| 4.14 | Interface Calendário. . . . .                                       | 43 |
| 4.15 | Interface Dispensador. . . . .                                      | 44 |
| 4.16 | Interface de Horários e Medicamentos. . . . .                       | 44 |

|  |    |
|--|----|
| 4.17 Tela de Reabastecimento do Dispensador via Bluetooth. . . . . | 45 |
| 4.18 Interface da tela de Alarme para a toma da medicação. . . . . | 45 |



# Capítulo 1

## Introdução

Este capítulo proporciona o enquadramento e objetivos do trabalho desenvolvido, apresentando a organização estrutural deste relatório.

### 1.1 Enquadramento

A gestão e a adesão ao tratamento medicamentoso representam desafios significativos tanto para os pacientes como para os profissionais de saúde. A complexidade dos esquemas terapêuticos, especialmente em casos de doenças crónicas, aumenta a probabilidade de esquecimentos, erros na administração e abandono do tratamento, o que pode resultar em consequências graves para a saúde dos pacientes. Estes problemas tornam-se ainda mais críticos em populações idosas, que frequentemente necessitam de apoio de cuidadores informais ou formais para garantir a correta administração da medicação. Neste contexto, o projeto Balvia surge como uma solução inovadora para a gestão de medicamentos, combinando um dispensador inteligente com uma aplicação móvel que integra os principais intervenientes no processo terapêutico: o paciente, o cuidador informal, o cuidador formal e a empresa parceira. Esta plataforma digital tem como objetivo promover uma experiência de saúde conectada, facilitando o acompanhamento remoto e oferecendo uma série de funcionalidades específicas para cada tipo de utilizador. A aplicação móvel desenvolvida permite o acesso remoto a dados sobre as prescrições, disponibilizando notificações e

alertas de medicação em tempo real, que auxiliam na adesão ao tratamento e reforçam a segurança e a qualidade de vida dos pacientes. Para os cuidadores e empresas parceiras, a aplicação proporciona também a gestão de subscrições e dispensadores, além de incluir calendários e rotas otimizadas para o reabastecimento de dispensadores nas proximidades. Desta forma, o Balvia propõe-se a revolucionar a administração de medicamentos ao oferecer uma solução integrada, promovendo um acompanhamento mais eficaz e uma comunicação mais fluida entre todos os envolvidos, com foco na melhoria dos resultados terapêuticos e na experiência do paciente e dos cuidadores.

## **1.2 Objetivo**

O objetivo do projeto Balvia é desenvolver uma solução integrada e inteligente para a gestão e administração de medicamentos, promovendo a adesão ao tratamento e melhorando a qualidade de vida dos pacientes. Através de uma aplicação móvel combinada com um dispensador inteligente, pretende-se facilitar o acompanhamento remoto e fornecer uma plataforma digital que conecte pacientes, cuidadores informais, cuidadores formais e empresas parceiras. Esta solução visa simplificar o processo de administração de medicamentos, oferecendo funcionalidades como acesso remoto a dados de medicação, envio de notificações e alertas, bem como a gestão de subscrições e dispensadores. Adicionalmente, propõe-se proporcionar aos cuidadores um suporte mais eficiente, incluindo ferramentas de planeamento, como calendários e rotas otimizadas para o reabastecimento dos dispensadores, garantindo uma experiência de saúde conectada e integrada para todos os intervenientes no processo terapêutico.

## **1.3 Estrutura do Relatório**

Este relatório encontra-se estruturado em cinco capítulos, os quais são descritos resumidamente a seguir.

### **1.3.1 Capítulo 1: Introdução**

Este capítulo aborda os desafios relacionados com a gestão e administração de medicamentos, destacando a necessidade de soluções inovadoras que promovam a adesão ao tratamento e melhorem a qualidade de vida dos pacientes. Apresentam-se os propósitos do projeto Balvia, que combina um dispensador inteligente com uma aplicação móvel, evidenciando o seu papel enquanto solução integrada para o acompanhamento remoto e a simplificação do processo terapêutico. Adicionalmente, é fornecida uma visão geral dos tópicos subseqüentes, com o objetivo de facilitar a compreensão do leitor acerca do desenvolvimento e dos objetivos do projeto.

### **1.3.2 Capítulo 2: Estado da Arte e Tecnologias**

Este capítulo apresenta uma revisão abrangente das investigações e sistemas existentes relacionados com a gestão e administração de medicamentos, com o objetivo de proporcionar uma visão aprofundada do panorama atual. Esta análise procura identificar os principais desafios e avanços na área, estabelecendo uma base sólida para o desenvolvimento da aplicação móvel integrada ao dispensador de medicamentos. Adicionalmente, é realizada uma descrição detalhada das tecnologias selecionadas para a implementação da aplicação, incluindo as suas características, vantagens e limitações. São apresentadas justificações para cada escolha tecnológica, evidenciando de que forma estas contribuem para a eficácia e a eficiência do sistema proposto.

### **1.3.3 Capítulo 3: Requisitos e Arquitetura**

Este capítulo identifica os requisitos necessários para o desenvolvimento da aplicação móvel e a sua integração com o dispensador de medicamentos. São descritos os requisitos funcionais, como a gestão de perfis, o envio de notificações, o acesso remoto a dados e a integração com o dispensador. São igualmente abordados os requisitos não funcionais, relacionados com o desempenho, a segurança, a usabilidade e a escalabilidade, fundamentais para assegurar a eficiência e a fiabilidade do sistema. Por fim, apresenta-se a

arquitetura planeada e as tecnologias seleccionadas, explicando de que forma estas suportam a implementação das funcionalidades e contribuem para alcançar os objetivos do projeto.

### **1.3.4 Capítulo 4: Desenvolvimento**

Neste capítulo, é apresentada uma descrição completa do desenvolvimento do sistema, desde a conceção até a implementação. Destacam-se escolhas de design, métodos de desenvolvimento e os desafios superados durante o processo.

### **1.3.5 Capítulo 5: Conclusão**

O último capítulo oferece um sumário geral do projeto, destacando contribuições, limitações e sugestões para melhorias futuras. Essa seção revisita os objetivos do projeto e avalia se foram totalmente alcançados.

# Capítulo 2

## Estado de Arte

Neste capítulo, efetua-se uma revisão abrangente de aplicações informáticas e artigos para fundamentar o desenvolvimento da aplicação móvel, abordando igualmente as tecnologias utilizadas.

### 2.1 Introdução

A pesquisa para o desenvolvimento deste projeto incluiu a análise de artigos e aplicações similares relacionadas com a gestão e administração de medicamentos. Foram identificados projetos relevantes, que se destacam pelas suas funcionalidades, como alertas para a toma de medicação, partilha de relatórios com profissionais de saúde, monitorização de sintomas e registo de dados de saúde, bem como o armazenamento de prescrições e a gestão de contactos médicos.

### 2.2 Trabalhos relacionados

A revisão de sistemas e estudos existentes permitiu uma compreensão aprofundada das melhores práticas e das funcionalidades mais desejadas em aplicações destinadas à gestão e administração de medicamentos. Através dessa análise comparativa, foi possível identificar lacunas e oportunidades para o aprimoramento da aplicação desenvolvida. Destaca-se,

ainda, a relevância de uma interface de utilizador intuitiva e a necessidade de funcionalidades robustas para garantir a eficácia do sistema. Segue-se uma descrição detalhada das principais aplicações analisadas, destacando suas contribuições e limitações.

### **2.2.1 MediSafe**

O MediSafe [1] é uma das aplicações mais reconhecidas no domínio da gestão de medicamentos, oferecendo um conjunto abrangente de funcionalidades para promover a adesão terapêutica. Entre seus principais recursos, encontram-se lembretes personalizados, uma base de dados extensa de medicamentos e um sistema de alertas para cuidadores.

A aplicação diferencia-se ao disponibilizar informações detalhadas sobre os fármacos, incluindo interações medicamentosas e efeitos adversos. Nos Estados Unidos, destaca-se ainda pelo alerta de possíveis conflitos entre medicamentos. No entanto, uma limitação relevante do MediSafe é sua dependência da base de dados local, o que pode restringir a sua aplicabilidade em determinados países. Além disso, a interface, embora funcional, pode ser considerada complexa para utilizadores idosos ou com dificuldades tecnológicas.

### **2.2.2 MyTherapy**

O MyTherapy [2] apresenta uma abordagem integrada, combinando gestão de medicação com monitorização da saúde. Além dos lembretes para a administração de medicamentos, oferece um diário de sintomas e um sistema de relatórios para facilitar o acompanhamento clínico.

Uma vantagem significativa dessa aplicação é a possibilidade de registar indicadores de saúde, como tensão arterial e glicemia, proporcionando um panorama abrangente da condição do utilizador. No entanto, uma das principais limitações do MyTherapy é a necessidade de entrada manual de dados, o que pode representar um obstáculo para a adesão contínua, especialmente entre utilizadores menos experientes.

### **2.2.3 CareZone**

O CareZone [3] foca-se na organização de informações médicas e no suporte ao cuidador. Sua principal funcionalidade consiste na gestão de medicamentos, complementada pelo armazenamento de informações médicas relevantes, como contactos clínicos e detalhes de seguros de saúde.

Apesar de sua proposta abrangente, a aplicação apresenta algumas limitações, como a falta de suporte para monitorização ativa da adesão ao tratamento. Além disso, a funcionalidade de partilha de informações, embora útil, pode suscitar preocupações relacionadas à privacidade dos dados.

### **2.2.4 SapoMed**

O SapoMed [4] destaca-se pelo uso de tecnologia de reconhecimento de código de barras para facilitar o registo de medicamentos. Além disso, oferece notificações visuais, sonoras e vibratórias para reforçar a adesão terapêutica.

Embora apresente um diferencial interessante na captura automática de informações, a sua base de dados pode ser limitada em relação à diversidade de medicamentos disponíveis. Outra limitação reside na dependência de conexão à Internet para algumas funcionalidades, o que pode comprometer sua usabilidade em determinados contextos.

### **2.2.5 Mediminder**

O Mediminder [5] foi desenvolvido especificamente para apoiar pacientes com doenças crónicas, focando-se na minimização de esquecimentos e na prevenção de administrações incorretas. Suas funcionalidades incluem a gestão de tratamentos, lembretes inteligentes, registo de administrações e um diário de sintomas.

Uma das principais vantagens do Mediminder é a possibilidade de envolver terceiros, permitindo que cuidadores e profissionais de saúde sejam notificados em situações críticas. No entanto, algumas limitações podem ser apontadas, como a ausência de integração direta com dispositivos médicos para monitorização automatizada e a necessidade de

melhorias na personalização da interface para diferentes perfis de utilizadores.

## **2.3 Comparação e Limitações Identificadas**

As soluções existentes no mercado, como MediSafe, MyTherapy, CareZone, SapoMed e Mediminder, abordam a gestão de medicamentos e adesão ao tratamento, mas ainda apresentam várias limitações. Muitas dessas plataformas, como MediSafe e MyTherapy, oferecem lembretes e monitoramento, mas não permitem um acompanhamento remoto integrado entre pacientes, cuidadores e profissionais de saúde. O Balvia desenvolve um dispensador inteligente e uma aplicação móvel que oferece notificações personalizadas e acompanhamento remoto em tempo real, conectando todos os intervenientes no processo terapêutico. Além disso, aplicações como CareZone não oferecem integração com dispositivos inteligentes, o que limita a automação e a eficácia no acompanhamento do tratamento. O Balvia, ao integrar um dispensador inteligente, permite automatizar a administração de medicação, facilitando o controle e a adesão ao tratamento. No que diz respeito ao suporte aos cuidadores, plataformas como Mediminder oferecem funcionalidades básicas, mas não incluem ferramentas avançadas de planeamento e gestão de rotas para reabastecimento, algo que o Balvia oferece, simplificando o trabalho dos cuidadores. Em termos de privacidade e segurança, muitas soluções ainda enfrentam desafios em proteger os dados dos pacientes, mas o Balvia assegura uma proteção robusta, cumprindo as regulamentações de privacidade. Além disso, o Balvia se destaca por ser mais resistente à conectividade intermitente, garantindo que suas funcionalidades cruciais continuem operacionais, mesmo em locais com internet instável, uma limitação que outras plataformas não conseguem superar.

## **2.4 Tecnologias**

A escolha cuidadosa das tecnologias para o desenvolvimento desse projeto foi orientada pela necessidade de atender a requisitos específicos de escalabilidade e sustentabilidade

a longo prazo. Dada a natureza do sistema, centrada no armazenamento de dados e gestão eficiente, antecipa-se a sua utilização como referência para futuros projetos com características semelhantes. Por isso, priorizaram-se tecnologias modernas, amplamente reconhecidas e com um sólido suporte na comunidade de desenvolvimento.

### **2.4.1 Linguagem Dart**

A linguagem Dart [6] foi escolhida por ser moderna e otimizada para o desenvolvimento de interfaces de utilizador reativas e de alto desempenho. A sua compilação Ahead-Of-Time (AOT) permite que as aplicações Flutter alcancem um desempenho equivalente ao nativo em dispositivos móveis. Além disso, apresenta uma curva de aprendizagem suave e integra-se de forma eficiente com o Flutter, proporcionando uma experiência de desenvolvimento fluida e coesa.

### **2.4.2 Framework Flutter**

O Flutter [7] é um framework multiplataforma que permite o desenvolvimento de aplicações para iPhone Operating System (iOS), Android, Web e Desktop a partir de um único código-base. Oferece um conjunto abrangente de widgets personalizáveis, garantindo uma interface de utilizador consistente e adaptada a diferentes plataformas. Além disso, possui um ecossistema ativo e em constante evolução, assegurando o acesso a uma vasta gama de pacotes e plugins que ampliam as suas funcionalidades.

### **2.4.3 SQLite**

O SQLite [8] foi selecionado como solução para armazenamento local, devido à sua eficiência e fiabilidade, sendo ideal para aplicações que requerem persistência de dados offline. Proporciona um equilíbrio eficaz entre simplicidade e a capacidade de lidar com conjuntos de dados mais complexos.

#### **2.4.4 FlutterBlue BLE Plus**

O Bluetooth foi integrado na aplicação utilizando o pacote FlutterBlue BLE Plus [9], escolhido pela sua robustez e versatilidade no suporte a funcionalidades de comunicação com dispositivos Internet of Things (IoT), controlo de hardware e sincronização de dados. Este pacote oferece uma Application Programming Interface (API) bem estruturada, permitindo a descoberta, conexão e gestão de dispositivos Bluetooth de forma eficiente. A documentação detalhada e o suporte ativo da comunidade facilitam a implementação, reduzindo a complexidade associada a funcionalidades Bluetooth Low Energy (BLE). Adicionalmente, o FlutterBlue BLE Plus suporta notificações e leitura de características, garantindo uma comunicação bidirecional em tempo real com dispositivos conectados. A sua capacidade de funcionar em múltiplas plataformas torna-o ideal para projetos que necessitam de escalabilidade e integração confiável com dispositivos Bluetooth.

#### **2.4.5 Trello**

O Trello [10] foi escolhido como a plataforma de gestão de projeto para facilitar a aplicação da metodologia Scrum. Essa ferramenta proporciona uma visão visual clara das tarefas em andamento, permitindo o acompanhamento do progresso de cada sprint. Com a capacidade de criar listas, cartões e atribuir tarefas, o Trello oferece uma abordagem flexível e intuitiva para o acompanhamento do desenvolvimento do projeto e a gestão eficiente da equipa.

#### **2.4.6 Postman**

O Postman [11] foi escolhido para o testes da API do projeto Balvia devido à sua interface intuitiva e funcionalidades avançadas. A ferramenta permite testar requisições HyperText Transfer Protocol (HTTP), validar respostas e garantir uma comunicação eficiente entre a aplicação móvel e o servidor. Além disso, facilita a organização dos testes, a deteção de erros de forma precoce e acelera o ciclo de desenvolvimento, tornando-se uma ferramenta essencial para garantir a integridade da API e a eficiência do sistema Balvia.

### **2.4.7 Xcode**

O Xcode [12] foi escolhido para o desenvolvimento da aplicação móvel Balvia devido à sua integração total com as plataformas da Apple e à sua robustez enquanto ambiente de desenvolvimento. Através do Xcode, é possível criar, testar e depurar a aplicação de forma eficiente para dispositivos iOS. A ferramenta disponibiliza funcionalidades avançadas, como o Interface Builder, que facilita a criação de interfaces gráficas, e o simulador, que permite testar a aplicação em diferentes dispositivos. Para além disso, o Xcode assegura uma boa gestão do ciclo de vida da aplicação, desde a codificação até à publicação na App Store, sendo, por isso, uma escolha crucial para garantir a qualidade e a performance da aplicação Balvia.

### **2.4.8 DataGrip**

O DataGrip, desenvolvido pela JetBrains [13] foi escolhido devido à sua poderosa funcionalidade na gestão de bases de dados. Trata-se de uma ferramenta abrangente e altamente eficiente para trabalhar com diferentes sistemas de gestão de bases de dados (SGBDs), como MySQL, PostgreSQL, SQLite, entre outros. O DataGrip oferece uma interface intuitiva que facilita a criação, edição e consulta de bases de dados, além de permitir uma depuração eficaz de queries SQL. A sua integração com outras ferramentas de desenvolvimento e o suporte a múltiplos SGBDs tornam-no uma escolha estratégica para o desenvolvimento da aplicação Balvia, especialmente no que diz respeito à gestão e manipulação eficiente dos dados relacionados com a medicação e os utilizadores.

### **2.4.9 Android Studio**

O Android Studio [14] foi escolhido devido à sua robustez e integração nativa com o sistema operativo Android, oferecendo uma plataforma altamente otimizada para a criação de aplicações móveis. Este ambiente de desenvolvimento proporciona ferramentas avançadas para o design de interfaces, testes, depuração e otimização de performance das aplicações, além de ser amplamente suportado pela comunidade de desenvolvedores. A

sua compatibilidade com o Flutter e o Dart também garante a flexibilidade e escalabilidade necessárias para o desenvolvimento da aplicação Balvia, permitindo uma integração eficiente com outras tecnologias e APIs. Adicionalmente, o Android Studio oferece suporte para múltiplos dispositivos e emuladores, facilitando o processo de desenvolvimento e teste da aplicação em diferentes cenários de utilização.

#### **2.4.10 Visual Paradigm Community**

O Visual Paradigm Community [15] foi selecionado devido à sua capacidade de fornecer ferramentas robustas e acessíveis para a modelação e o design de software, essenciais para o planeamento e a documentação do desenvolvimento de aplicações. Este ambiente de desenvolvimento disponibiliza suporte para uma ampla gama de notações, como UML (Unified Modeling Language), BPMN (Business Process Model and Notation) e diagramas de casos de uso, que são cruciais para a representação clara e organizada dos requisitos e fluxos da aplicação. Além disso, a versão Community é gratuita para utilização não comercial, tornando-se uma solução acessível para equipas de desenvolvimento que procuram ferramentas de alto nível sem custos adicionais. A integração com outras ferramentas de desenvolvimento e a sua interface intuitiva promovem a produtividade, garantindo que as equipas possam colaborar e estruturar projetos de forma eficiente. Esta escolha foi ainda motivada pelo forte suporte da comunidade e pela extensa documentação disponível, que facilita a resolução de problemas e melhora a curva de aprendizagem para novos utilizadores.

#### **2.4.11 Overleaf**

O Overleaf [16] foi escolhido para criar o relatório da aplicação Balvia em LaTeX devido à sua facilidade de uso, ambiente colaborativo e integração com o formato LaTeX. A ferramenta simplifica a gestão de pacotes, permite visualização em tempo real e garante eficiência na formatação e revisão de conteúdos, sendo ideal para documentos técnicos e científicos.

### **2.4.12 Mendeley**

O Mendeley [17] foi selecionado devido à sua eficácia e integração com o processo de gestão de referências acadêmicas, proporcionando uma solução robusta para a organização, partilha e citação de fontes bibliográficas. Esta plataforma disponibiliza ferramentas avançadas para a criação de bibliotecas personalizadas, colaboração entre equipas de investigação e inserção automática de citações em diversos formatos, garantindo agilidade e precisão na produção de documentos académicos. A sua compatibilidade com softwares de processamento de texto, como o Microsoft Word e LaTeX, permite uma integração eficiente no fluxo de trabalho, reduzindo o tempo necessário para formatar referências. Adicionalmente, o Mendeley oferece suporte para sincronização em nuvem, permitindo o acesso às bibliotecas em múltiplos dispositivos e promovendo a mobilidade e continuidade do trabalho em diferentes contextos.

## **2.5 Conclusão**

Este capítulo proporcionou uma revisão abrangente de sistemas e artigos relacionados com a gestão e administração de medicamentos, destacando as principais funcionalidades oferecidas pelas aplicações móveis disponíveis no mercado. Embora todos os sistemas analisados apresentem características relevantes, como lembretes de medicação, agendamentos flexíveis e rastreamento de múltiplas prescrições, nenhum deles atende integralmente às necessidades específicas do projeto Balvia.

A aplicação móvel Balvia distingue-se por integrar funcionalidades inovadoras, como o acesso remoto a dados de medicação, a conexão Bluetooth com um dispensador inteligente e a gestão automatizada do reabastecimento do dispensador. Assim, o Balvia propõe uma solução mais completa e personalizada, que visa melhorar a adesão ao tratamento, simplificar a administração de medicamentos e proporcionar um acompanhamento mais eficaz, garantindo uma experiência integrada para os pacientes, cuidadores e empresas parceiras. A escolha cuidadosa das tecnologias, como Flutter [7], Dart [6] e SQLite [8], foi fundamental para garantir a escalabilidade e sustentabilidade do projeto. Além disso, a plataforma

Trello [10] foi incorporada para assegurar uma gestão eficiente do ciclo de vida do desenvolvimento. Dessa forma, o capítulo forneceu uma base sólida para o desenvolvimento da aplicação móvel Balvia, preparando o caminho para uma solução inovadora que atenda de forma eficaz às exigências contemporâneas na gestão e administração de medicamentos.

# Capítulo 3

## Requisitos e Arquitetura

Este capítulo tem como objetivo especificar os requisitos e a arquitetura da aplicação móvel Balvia. A definição clara e detalhada dos requisitos é fundamental para o sucesso de qualquer projeto de desenvolvimento de software, pois permite uma compreensão aprofundada das necessidades dos utilizadores e das funcionalidades que o sistema deve disponibilizar para satisfazer essas necessidades.

A arquitetura do sistema, por sua vez, desempenha um papel crucial na definição de como a aplicação irá operar, garantindo a integração adequada entre os diversos componentes e a interação eficiente com os utilizadores e outros sistemas.

Nesta secção, serão abordadas duas componentes essenciais: a especificação e validação dos requisitos da aplicação, que asseguram a adequação do sistema às expectativas do utilizador final, e a descrição da arquitetura, que define a estrutura técnica subjacente ao seu funcionamento.

### 3.1 Especificação e Validação de Requisitos

Seguindo as melhores práticas da engenharia de requisitos [18], o processo definido para a aplicação Balvia é delineado da seguinte forma:

### **3.1.1 Especificação de Requisitos**

Na fase de especificação de requisitos, serão documentados os requisitos funcionais e não funcionais da aplicação móvel Balvia. Este processo incluirá a elaboração de um diagrama de casos de uso, histórias de utilizador e um diagrama de domínio do sistema. Cada funcionalidade será descrita de forma detalhada, abrangendo o seu fluxo de funcionamento, as interações com outros componentes e quaisquer restrições associadas.

### **3.1.2 Validação de Requisitos**

Após a especificação, será realizada a validação dos requisitos com o intuito de assegurar que estes satisfazem as necessidades identificadas dos utilizadores e os objetivos do negócio. Esta etapa visa também identificar e corrigir eventuais erros, lacunas ou omissões na documentação dos requisitos, garantindo que o sistema esteja em conformidade com as expectativas. O processo de validação poderá ainda resultar na descoberta de novas funcionalidades relevantes para o projecto.

## **3.2 Requisitos**

Para garantir que a aplicação Balvia responda eficazmente às necessidades dos seus utilizadores, é fundamental definir e estruturar os seus requisitos de forma clara e rigorosa. Esta secção apresenta os principais elementos que compõem os requisitos do sistema, nomeadamente os diagramas de casos de uso, as histórias de utilizador e o diagrama de domínio. Os perfis de utilizadores considerados na aplicação incluem: visitantes (utilizadores não autenticados), pacientes, cuidadores informais, cuidadores formais e empresas parceiras. Estes atores representam os diferentes tipos de interação com a aplicação e estão refletidos nos diversos artefactos apresentados. A inclusão destes perfis nos diagramas e histórias permite compreender as funcionalidades esperadas da aplicação, o comportamento dos diferentes utilizadores e a estrutura conceptual subjacente ao seu

funcionamento. A definição cuidada destes requisitos contribui para assegurar a qualidade, fiabilidade e usabilidade da solução desenvolvida.

### 3.2.1 Diagrama de Casos de Uso

Os casos de uso representam a interação dos diferentes utilizadores com o sistema Balvia, descrevendo as principais funcionalidades disponíveis e o fluxo de operações. Estes diagramas permitem visualizar de forma clara a forma como cada tipo de utilizador, sejam visitantes (utilizadores não autenticados), pacientes, cuidadores informais, cuidadores formais ou empresas parceiras, interage com a plataforma, proporcionando uma melhor compreensão dos processos envolvidos. Cada caso de uso ilustra um conjunto específico de ações realizadas no sistema, identificando os atores e os seus respetivos papéis na gestão da medicação. Através destes diagramas, é possível compreender as principais funcionalidades da aplicação, tais como o registo e autenticação de utilizadores, a criação e gestão de prescrições, o acompanhamento remoto da administração da medicação, o envio de notificações e alertas, bem como a sincronização com o dispensador inteligente. Nos subcapítulos seguintes, são apresentados e descritos os principais diagramas de caso de uso, detalhando o fluxo de interações e as funcionalidades associadas a cada utilizador do sistema.

O diagrama de caso de uso apresentado na Figura 3.1 ilustra as funcionalidades principais relacionadas com o acesso e a gestão de contas no sistema. Os visitantes (utilizadores não autenticados) têm acesso a funções básicas, como criar conta, fazer login e recuperar senha. O paciente dispõe de funcionalidades específicas, incluindo recuperar e alterar a senha, visualizar e editar o perfil, ativar a biometria, configurar alertas e notificações e cancelar a conta. Os cuidadores formais, cuidadores informais e o gestor da empresa herdam todas as funcionalidades do paciente, podendo realizar as mesmas operações de gestão de conta.

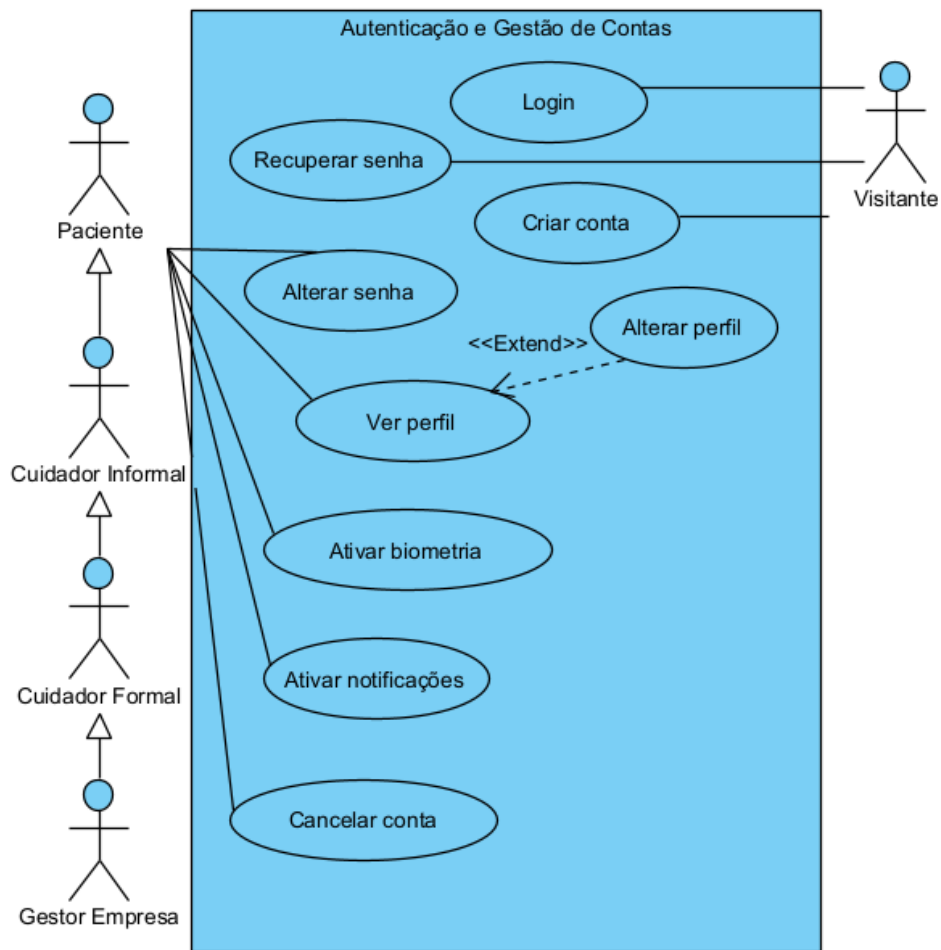


Figura 3.1: Diagrama de Casos de Uso: Autenticação e Gestão de Contas.

O caso de uso apresentado na Figura 3.2 ilustra as funcionalidades principais do Sub-sistema Balvia Cliente e descreve as operações disponíveis para o paciente e o cuidador informal. O paciente pode aceder ao caso de uso Aceder a Informações, que lhe permite consultar o calendário, visualizar a medicação, ver a prescrição médica, aceder a acessórios, visualizar o dispensador e consultar informações sobre o Balvia Ecosystem. Adicionalmente, pode utilizar o caso de uso Saúde e Alertas, que possibilita a monitorização do diário de saúde e o recebimento de alarmes de medicação. O cuidador informal herda todas as funcionalidades atribuídas ao paciente, dispondo ainda de casos de uso adicionais no módulo de monitorização e notificações, que permitem a gestão de pacientes,

bem como o envio e recepção de notificações. Este diagrama evidencia as funções partilhadas entre os dois perfis de utilizador, bem como as operações exclusivas do cuidador informal, de acordo com as suas responsabilidades no sistema.

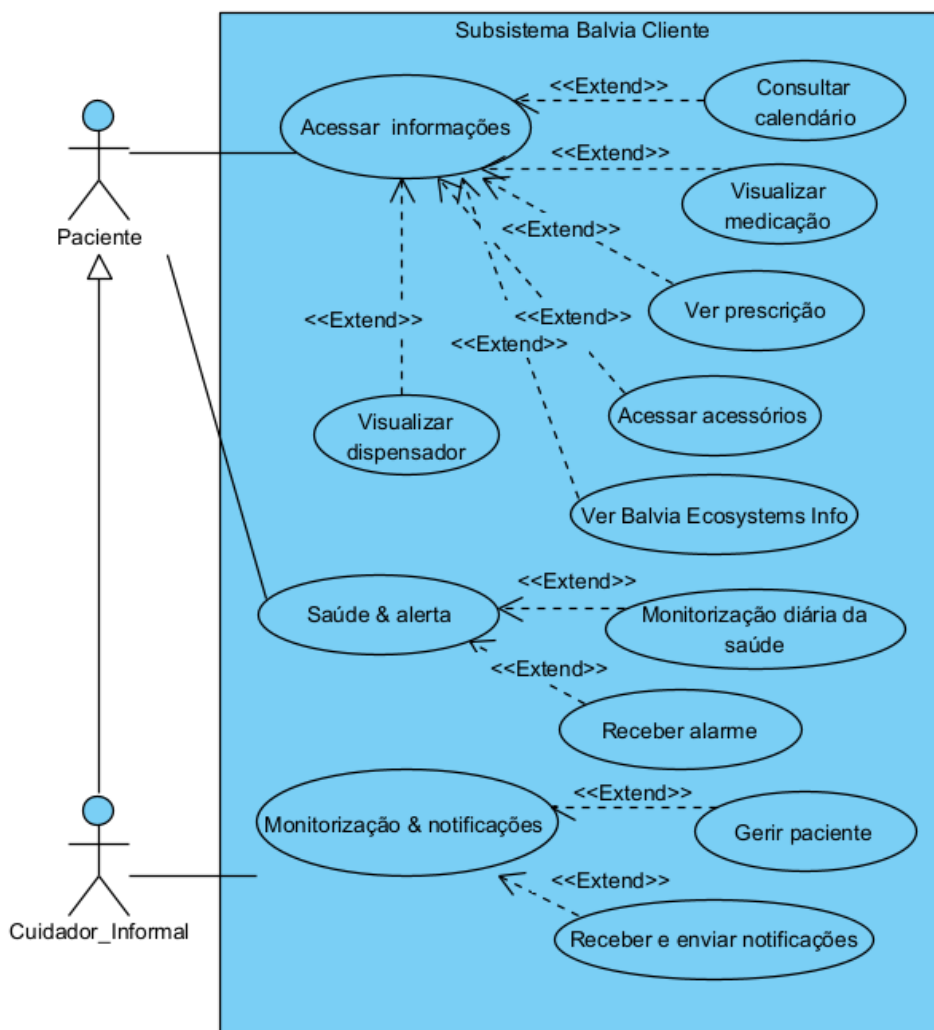


Figura 3.2: Diagrama de Casos de Uso Subsistema Balvia Cliente.

A Figura 3.3 apresenta os principais casos de uso do **Subsistema Balvia Empresa**, destacando as funcionalidades atribuídas a três perfis distintos: **Cuidador Formal**, **Gestor da Empresa** e **Administrador**.

O **Cuidador Formal** tem acesso a três áreas funcionais principais:

- **Gestão de Pacientes e Dispensadores:** permite gerir pacientes, visualizar e reabastecer dispensadores, organizar rotas de reabastecimento, gerir prescrições médicas e definir horários de administração da medicação.
- **Informações do Balvia Ecosystems:** proporciona acesso a conteúdos informativos, como tutoriais, manuais de utilizador, ajuda e suporte, contactos e a possibilidade de partilhar a experiência de utilização.
- **Monitorização e Notificação:** possibilita a consulta de relatórios de saúde e de medicação, bem como o envio e receção de notificações relevantes.

O **Gestor da Empresa** possui todas as permissões atribuídas ao Cuidador Formal, acrescidas das seguintes funcionalidades:

- **Gestão de Cuidadores Formais:** permite adicionar, editar ou remover cuidadores formais associados à empresa.
- **Gestão de Serviços e Subscrições:** possibilita a criação, edição e remoção de subscrições, bem como a associação de serviços e pacotes a essas subscrições.

O **Administrador** dispõe de todas as funcionalidades do Gestor da Empresa, bem como de permissões adicionais:

- **Gestão de Dispensadores:** permite a criação, edição e eliminação de dispensadores no sistema.
- **Administração e Gestão:** possibilita a gestão integral de utilizadores da plataforma.

Este diagrama de casos de uso evidencia a hierarquia de permissões no sistema, distinguindo claramente as funcionalidades comuns e exclusivas de cada perfil, de acordo com o respetivo nível de responsabilidade.

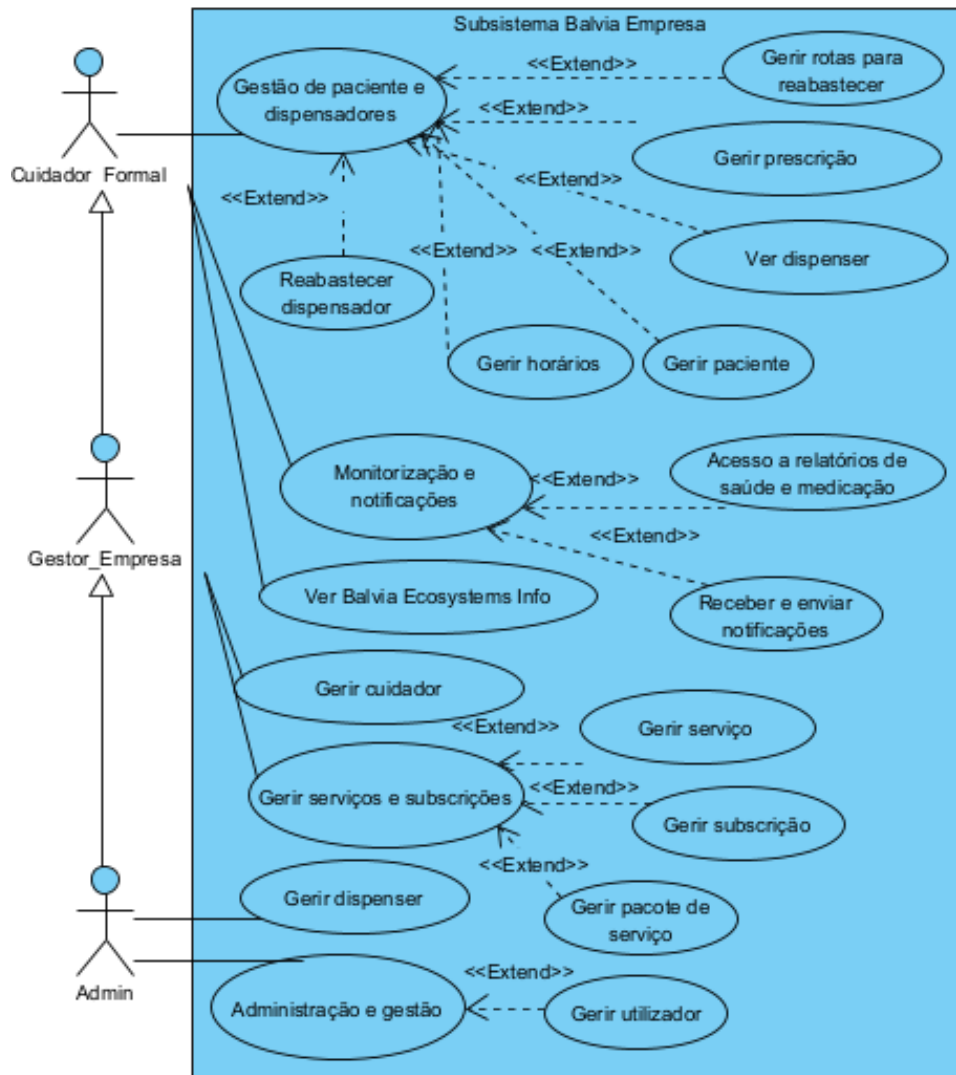


Figura 3.3: Diagrama de Casos de Uso Subsistema Balvia Empresa.

### 3.2.2 Histórias de Utilizador

A história de utilizador é uma técnica utilizada para descrever, de forma clara e objetiva, as necessidades e expectativas dos diferentes utilizadores do sistema Balvia. Através desta abordagem, é possível compreender o contexto de utilização da aplicação, as ações que os utilizadores pretendem realizar e os benefícios que esperam obter.

Cada história de utilizador é elaborada a partir da perspetiva de um interveniente específico, como visitante (utilizadores não autenticados), o paciente, o cuidador informal,

o cuidador formal ou a empresa parceira, destacando os seus objetivos e interações dentro da plataforma. Esta metodologia facilita a definição dos requisitos do sistema e assegura que este responde adequadamente às necessidades dos utilizadores.

Nos subcapítulos seguintes, são apresentadas e descritas as principais histórias de utilizador, ilustrando os diferentes cenários de utilização do sistema Balvia.

### Histórias de Utilizador do Tipo Visitante

A Tabela 3.1 apresenta as histórias de utilizador do tipo visitante no sistema, com uma descrição clara de cada funcionalidade, respetivo módulo, prioridade e critérios de aceitação.

| ID    | Módulo  | P | Utilizador | História   | Crítérios de Aceitação  |
|-------|---------|---|------------|--|---|
| UV-01 | Registo | 1 | Visitante  | Como novo utilizador, quero registar-me na aplicação, escolhendo entre Cliente ou Empresa, definir o meu papel e criar credenciais para obter permissões corretas. | Definir tipo de utilizador e papel; validação de e-mail e palavra-passe; verificação de campos obrigatórios; envio de e-mail de ativação. |
| UV-02 | Login   | 1 | Visitante  | Como utilizador registado, quero autenticar-me para aceder à minha conta.  | Validação correta de credenciais; mensagens de erro; opção de recuperação de palavra-passe.   |

Tabela 3.1: Histórias de Utilizador do Tipo Visitante

### Histórias de Utilizador do Tipo Paciente

A Tabela 3.2 apresenta as histórias de utilizador do tipo Paciente, detalhando funcionalidades específicas relacionadas com a gestão da sua conta e do seu tratamento. Estas funcionalidades permitem ao paciente acompanhar as suas medicações, acessórios, dispensador e calendário de forma clara e acessível. Cada história está associada a um módulo, a uma prioridade e a critérios de aceitação, garantindo uma experiência consistente e orientada para o apoio ao tratamento.

| ID    | Módulo            | P | Utilizador | História  | Crítérios de Aceitação   |
|-------|-------------------|---|------------|---|--|
| UP-01 | Visualizar Perfil | 2 | Paciente   | Como paciente, quero aceder ao meu perfil para visualizar as minhas informações pessoais. | Acesso pelo menu ou foto; Exibição de dados pessoais; Sem opção de edição. |

| ID    | Módulo              | P | Utilizador | História   | CrITÉrios de Aceitação  |
|-------|---------------------|---|------------|--|---|
| UP-02 | Editar Perfil       | 2 | Paciente   | Como paciente, quero editar as minhas informações pessoais para mantê-las atualizadas. | Acesso com dados pré-preenchidos; Validação de campos; Mensagem de sucesso após salvar. |
| UP-03 | Alterar Senha       | 1 | Paciente   | Como paciente, quero alterar a minha palavra-passe para garantir segurança.            | Validação da senha atual e nova; Nova senha robusta; Confirmação com sucesso.           |
| UP-04 | Ativar Biometria    | 3 | Paciente   | Como paciente, quero ativar o login biométrico para facilitar o acesso.                | Verificar suporte; Ativar opção; Login funcional por biometria.                         |
| UP-05 | Ativar Notificações | 3 | Paciente   | Como paciente, quero ativar/desativar notificações para controlar alertas.             | Acesso às definições; Push e e-mail configuráveis; Funcionalidade conforme escolha.     |
| UP-06 | Cancelar Conta      | 1 | Paciente   | Como paciente, quero cancelar a conta para remover os meus dados.                      | Confirmação com palavra-passe; Remoção total dos dados; Logout automático.              |
| UP-07 | Ajuda e Suporte     | 2 | Paciente   | Como paciente, quero aceder ao suporte para esclarecer dúvidas.                        | Acesso via menu; FAQ visível; Envio de pedidos e feedback possível.                     |
| UP-08 | Página Contacto     | 2 | Paciente   | Como paciente, quero contactar a equipa da aplicação facilmente.                       | Acesso via menu de ajuda; Exibir contactos e formulário.                                |
| UP-09 | Informações         | 2 | Paciente   | Como paciente, quero saber mais sobre a app e empresa.                                 | Versão da app; Info da empresa; Termos e política disponíveis.                          |
| UP-10 | Medicações          | 1 | Paciente   | Como paciente, quero ver minha lista de medicações detalhadamente.                     | Lista com status; Alertas de renovação; Mensagens em caso de erro.                      |
| UP-11 | Acessórios          | 2 | Paciente   | Como paciente, quero gerir os meus acessórios de forma eficaz.                         | Lista com status; Alertas de manutenção; Mensagens em caso de erro.                     |
| UP-12 | Dispensador         | 2 | Paciente   | Como paciente, quero ver o meu dispensador com detalhes.                               | Visualização dos módulos; Alertas de reposição; Mensagens de erro.                      |
| UP-13 | Calendário          | 1 | Paciente   | Como paciente, quero consultar meu calendário de medicação.                            | Ícones de doses; Histórico e próximas doses; Atualização automática.                    |

Tabela 3.2: Histórias de Utilizador do Tipo Paciente

### Histórias de Utilizador do Cuidador Informal

A Tabela 3.3 apresenta as histórias de utilizador do tipo Cuidador Informal, detalhando funcionalidades que permitem a gestão da conta, a administração de familiares, o monitorização do dispensador e o acompanhamento do calendário de medicação. Cada história

está associada a um módulo, a uma prioridade e a critérios de aceitação, assegurando clareza e apoio ao cuidado dos pacientes.

| ID    | Módulo              | P | Utilizador | História  | Critérios de Aceitação   |
|-------|---------------------|---|------------|---|--|
| UI-01 | Visualizar Perfil   | 2 | C.Informal | Aceder à minha página de perfil para visualizar as minhas informações.  | Acesso pelo menu; exibição correta das informações; leitura sem edição.                  |
| UI-02 | Editar Perfil       | 2 | C.Informal | Editar as minhas informações pessoais para manter os dados atualizados. | Acesso à edição com dados preenchidos; validação; mensagem de sucesso.                   |
| UI-03 | Alterar Senha       | 1 | C.Informal | Alterar a palavra-passe para garantir a segurança da conta.             | Validação da palavra-passe atual e nova; mensagem de sucesso.                            |
| UI-04 | Ativar Biometria    | 3 | C.Informal | Ativar login por biometria para aceder de forma rápida e segura.        | Verificação do dispositivo; ativação na app; login funcional.                            |
| UI-05 | Ativar Notificações | 3 | C.Informal | Ativar ou desativar notificações conforme a minha preferência.          | Acesso via menu; escolha de notificações push e e-mail; funcionamento conforme definido. |
| UI-06 | Cancelar Conta      | 1 | C.Informal | Cancelar a conta e remover os meus dados da aplicação.                  | Confirmação e autenticação; remoção de dados; logout automático.                         |
| UI-07 | Ajuda e Suporte     | 2 | C.Informal | Aceder à página de ajuda para esclarecer dúvidas ou contactar suporte.  | Acesso via menu; exibição de FAQ; envio de pedidos de suporte.                           |
| UI-08 | Página Contacto     | 2 | C.Informal | Contactar a equipa da aplicação através da página de contacto.          | Acesso pelo menu de ajuda; formulário de contacto funcional.                             |
| UI-09 | Informações         | 2 | C.Informal | Ver informações sobre a aplicação e a empresa responsável.              | Página “Sobre a App”; versão, descrição, política de privacidade e termos.               |
| UI-10 | Familiar            | 1 | C.Informal | Gerir familiares e o histórico de medicação para monitorizar a saúde.   | Acesso à página “Família”; exibição e edição dos dados (se autorizado); notificações.    |
| UI-11 | Dispensador         | 2 | C.Informal | Consultar o estado dos módulos do dispensador.                          | Acesso via menu; exibição do estado; alertas e mensagens de erro.                        |
| UI-12 | Calendário          | 1 | C.Informal | Ver o calendário de medicação com tomas passadas e futuras.             | Exibição por período do dia; alertas e detalhes; mensagens de erro se necessário.        |

Tabela 3.3: Histórias de Utilizador do Tipo Cuidador Informal

## Histórias de Utilizador do Tipo Cuidador Formal

A Tabela 3.4 apresenta as histórias de utilizador do tipo Cuidador Formal, detalhando funcionalidades que permitem a gestão da conta, a gestão de prescrições, o acompanhamento de dispensadores, o reabastecimento de medicamentos e a gestão de clientes. Cada história está associada a um módulo, a uma prioridade e a critérios de aceitação, garantindo clareza e suporte às atividades de cuidado e gestão.

| ID    | Módulo              | P | Utilizador | História  | Critérios de Aceitação   |
|-------|---------------------|---|------------|---|--|
| UF-01 | Visualizar Perfil   | 2 | C.Formal   | Aceder à minha página de perfil para visualizar as minhas informações.  | Acesso pelo menu; exibição correta; só leitura.                  |
| UF-02 | Editar Perfil       | 2 | C.Formal   | Editar as minhas informações pessoais para manter os dados atualizados. | Edição com dados preenchidos; validação; sucesso.                |
| UF-03 | Alterar Senha       | 1 | C.Formal   | Alterar a palavra-passe para garantir a segurança da conta.             | Validação da senha atual e nova; robustez; sucesso.              |
| UF-04 | Ativar Biometria    | 3 | C.Formal   | Ativar login por biometria para aceder de forma rápida e segura.        | Verificação do dispositivo; ativação e login biométrico.         |
| UF-05 | Ativar Notificações | 3 | C.Formal   | Ativar ou desativar notificações conforme a minha preferência.          | Acesso via menu; notificações push e e-mail; funcionamento.      |
| UF-06 | Cancelar Conta      | 1 | C.Formal   | Cancelar a conta e remover os meus dados da aplicação.                  | Confirmação com senha; remoção de dados; logout.                 |
| UF-07 | Ajuda e Suporte     | 2 | C.Formal   | Aceder à página de ajuda para esclarecer dúvidas ou contactar suporte.  | Acesso via menu; FAQ e envio de pedidos.                         |
| UF-08 | Página Contacto     | 2 | C.Formal   | Contactar a equipa da aplicação através da página de contacto.          | Acesso pelo menu; formulário funcional.                          |
| UF-09 | Informações         | 2 | C.Formal   | Ver informações sobre a aplicação e a empresa responsável.              | Página “Sobre a App”; versão, descrição, política e termos.      |
| UF-10 | Prescrição          | 1 | C.Formal   | Gerir prescrições médicas com horários e detalhes dos medicamentos.     | Adição, edição e visualização; permissões; organização por data. |
| UF-11 | Calendário          | 1 | C.Formal   | Ver o calendário de medicação com tomas passadas e futuras.             | Exibição por período; alertas; detalhes das tomas.               |

Tabela 3.4: Histórias de Utilizador do Tipo Cuidador Formal

## Histórias de Utilizador do Tipo Gestor de Empresa

A Tabela 3.5 apresenta as histórias de utilizador do tipo Gestor de Empresa, detalhando funcionalidades que permitem a gestão da conta, bem como a gestão eficiente de dispensadores, clientes, prescrições, cuidadores e subscrições. Estas funcionalidades asseguram que o gestor possa supervisionar o funcionamento dos dispositivos, garantir o reabastecimento correto e manter um registo atualizado de clientes e colaboradores. Os critérios de aceitação definidos garantem clareza na implementação e verificação de cada funcionalidade.

| ID    | Módulo              | P | Utilizador | História  | Critérios de Aceitação                                       |
|-------|---------------------|---|------------|---|--|
| UG-01 | Visualizar Perfil   | 2 | G.Empresa  | Aceder à minha página de perfil para visualizar as minhas informações.  | Acesso pelo menu; exibição correta; só leitura.              |
| UG-02 | Editar Perfil       | 2 | G.Empresa  | Editar as minhas informações pessoais para manter os dados atualizados. | Edição com dados preenchidos; validação; sucesso.            |
| UG-03 | Alterar Senha       | 1 | G.Empresa  | Alterar a palavra-passe para garantir a segurança da conta.             | Validação da senha atual e nova; robustez; sucesso.          |
| UG-04 | Ativar Biometria    | 3 | G.Empresa  | Ativar login por biometria para aceder de forma rápida e segura.        | Verificação do dispositivo; ativação e login biométrico.     |
| UG-05 | Ativar Notificações | 3 | G.Empresa  | Ativar ou desativar notificações conforme a minha preferência.          | Acesso via menu; notificações push e e-mail; funcionamento.  |
| UG-06 | Cancelar Conta      | 1 | G.Empresa  | Cancelar a conta e remover os meus dados da aplicação.                  | Confirmação com senha; remoção de dados; logout.             |
| UG-07 | Ajuda e Suporte     | 2 | G.Empresa  | Aceder à página de ajuda para esclarecer dúvidas ou contactar suporte.  | Acesso via menu; FAQ e envio de pedidos.                     |
| UG-08 | Página Contacto     | 2 | G.Empresa  | Contactar a equipa da aplicação através da página de contacto.          | Acesso pelo menu; formulário funcional.                      |
| UG-09 | Informações         | 2 | G.Empresa  | Ver informações sobre a aplicação e a empresa responsável.              | Página “Sobre a App”; versão, descrição, política e termos.  |
| UG-10 | Dispensador         | 3 | G.Empresa  | Acompanhar os dispensadores dos clientes e ver os seus detalhes.        | Listar e detalhar dispensadores; alertas de módulos vazios.  |
| UG-11 | Cliente             | 5 | G.Empresa  | Gerir os clientes da empresa para acompanhar e atualizar dados.         | Listar, editar, filtrar; gerir associação com dispensadores. |
| UG-12 | Prescrição          | 1 | G.Empresa  | Gerir prescrições dos clientes com horários e medicamentos.             | Adição, edição e visualização; organização e permissões.     |

Tabela 3.5: Histórias de Utilizador do Tipo Gestor da Empresa

## Histórias Utilizador do Tipo Administrador

A Tabela 3.6 apresenta as histórias de utilizador do tipo Administrador, proporcionando uma visão geral das funcionalidades que permitem a gestão dos utilizadores e dos dispensadores, bem como o controlo do processo de reabastecimento. O Administrador tem autonomia para gerir o acesso dos utilizadores, manter os dispensadores atualizados e garantir o correto funcionamento da aplicação. Os critérios de aceitação definidos asseguram que cada funcionalidade seja implementada de forma clara e verificável.

| ID    | Módulo             | P | Utilizador | História  | Critérios de Aceitação   |
|-------|--------------------|---|------------|---|--|
| UA-01 | Reabastecer        | 4 | Admin      | Reabastecer dispensadores para garantir medicamentos disponíveis. | Ligação Bluetooth; inserir info por módulo; sequência com confirmação. |
| UA-02 | Utilizadores       | 5 | Admin      | Gerir utilizadores da aplicação para controlar acessos.           | Listar com filtros; adicionar, editar, ativar/desativar; histórico.    |
| UA-03 | Dispensador        | 3 | Admin      | Gerir dispensadores com visualização e edição de dados.           | Listagem; adicionar/editar/eliminar com validação; histórico de ações. |
| UA-04 | Calendário         | 2 | Admin      | Ver dispensadores próximos e rota eficiente de reabastecimento.   | Exibir rota e info; ajustar rota; permissões e mensagens de erro.      |
| UA-05 | Subscrição         | 5 | Admin      | Criar e gerir subscrições e associar serviços a utilizadores.     | Criar/editar subscrição; associar serviços e pacotes a clientes.       |
| UA-06 | Relatório de Saúde | 4 | Admin      | Aceder a relatórios de saúde e medicação dos pacientes.           | Indicadores de adesão; histórico de toma; exportar PDF; filtros.       |

Tabela 3.6: Histórias de Utilizador do Tipo Administrador

### 3.2.3 Diagrama de Domínio

O Diagrama de Domínio apresentado na Figura 3.4 representa as principais entidades e os respetivos relacionamentos no contexto do sistema Balvia, tendo por base a estrutura previamente definida pela empresa. O seu principal objetivo é proporcionar uma visão abrangente das entidades envolvidas na gestão e administração de medicamentos, facilitando a compreensão da organização e do fluxo de dados no sistema.

As principais entidades incluem Pessoa, Subscrição, Pacote de Serviço, Serviço, Horários, Medicamento, Empresa, Dispensador, Prescrição, Papel e Endereço, que se interligam para assegurar um funcionamento eficiente e seguro da plataforma. Estas entidades são

responsáveis por estruturar:

- a **gestão de utilizadores e permissões**, permitindo a diferenciação de papéis, tais como paciente, cuidador formal, cuidador informal e gestor da empresa;
- a **gestão de prescrições e medicação**, garantindo que todas as informações relativas aos tratamentos estejam devidamente organizadas e acessíveis;
- a **integração com dispositivos**, como dispensadores de medicamentos e smartwatches, assegurando a correta administração dos medicamentos;
- a **sincronização com o servidor e o acesso remoto aos dados**, permitindo um suporte contínuo aos utilizadores.

Adicionalmente, a aplicação dispõe de uma base de dados local, essencial para armazenar informações relativas à administração da medicação. Esta base de dados garante que:

- os **horários das tomas (Schedule)** sejam mantidos e consultados mesmo sem ligação à internet;
- os **alarmes e notificações** sejam ativados no momento adequado, assegurando o cumprimento do tratamento.

### 3.3 Arquitetura do Sistema

A arquitetura do sistema Balvia define uma estrutura robusta e flexível que suporta o funcionamento eficiente da aplicação, garantindo a integração segura e escalável entre os seus diversos componentes. Esta arquitetura foi concebida para assegurar uma comunicação eficaz entre a aplicação móvel, o dispensador inteligente e os diferentes intervenientes no processo terapêutico.

O desenvolvimento da arquitetura teve como princípio a criação de um sistema confiável, seguro e adaptável, capaz de gerir a administração de medicamentos de forma

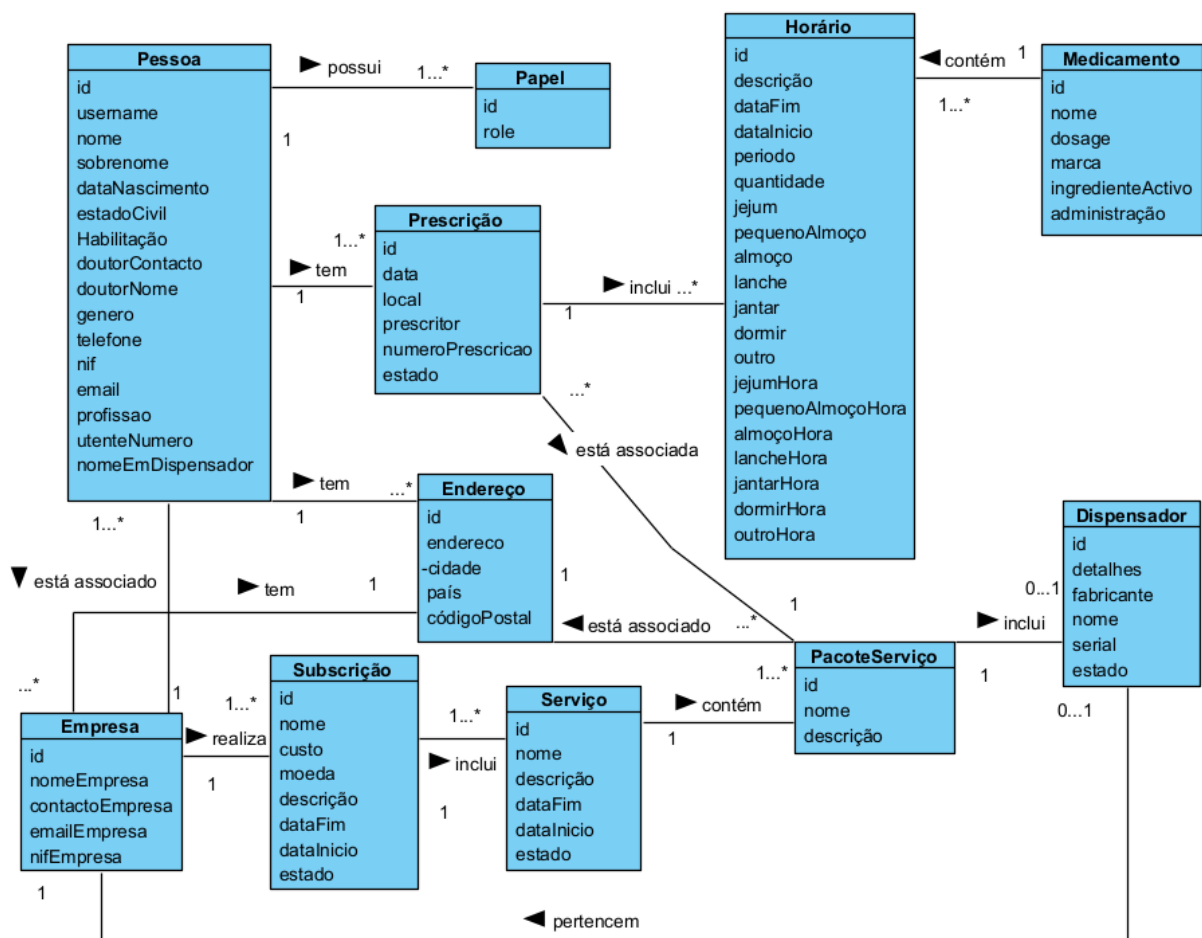


Figura 3.4: Diagrama de Domínio

automatizada e eficiente. Além disso, foram adotadas abordagens que garantem a integridade dos dados, a proteção da privacidade do utilizador e o desempenho adequado da aplicação, independentemente do contexto de utilização.

Nos tópicos seguintes, são apresentados os principais elementos da arquitetura do Balvia, destacando a sua estrutura, os métodos de comunicação e as decisões técnicas essenciais para o seu funcionamento.

A arquitetura da aplicação móvel Balvia, ilustrada na Figura 3.5, foi projetada para garantir uma comunicação eficiente entre os diferentes componentes do sistema, assegurando uma experiência de utilização contínua, mesmo em cenários de conectividade limitada. Esta arquitetura é composta pelos seguintes elementos principais:

- API RESTful e Sincronização Remota: A aplicação interage com um servidor remoto através de uma API RESTful, responsável por sincronizar dados críticos, como prescrições médicas, horários de medicação e alertas. Esta abordagem garante que todas as alterações realizadas na aplicação sejam refletidas na base de dados remota, assegurando consistência e atualização em tempo real.
- Comunicação com Dispositivos Externos via Bluetooth BLE: A aplicação comunica com dispositivos externos, como o dispensador de medicamentos e um relógio inteligente, utilizando tecnologia Bluetooth Low Energy (BLE). Esta conexão permite o controlo remoto do dispensador, incluindo o reabastecimento e a organização dos medicamentos, bem como a recolha de dados de saúde em tempo real através do relógio inteligente (frequência cardíaca, passos diários, calorias gastas e qualidade do sono).
- Base de Dados Local com SQLite: A aplicação possui uma base de dados local baseada em SQLite, garantindo o funcionamento em modo offline. Nesta base de dados são armazenadas informações essenciais, como prescrições médicas e horários de medicação. Em situações de falta de conectividade, a aplicação continua a funcionar normalmente, emitindo alertas locais para lembrar o paciente de tomar a medicação. Quando a ligação à internet é restabelecida, os dados locais são automaticamente sincronizados com o servidor.

- **Gestão de Alertas e Notificações:** A aplicação emite lembretes de medicação e alertas de saúde diretamente no dispositivo do utilizador, garantindo que o paciente esteja sempre informado sobre os horários de toma dos medicamentos e qualquer alteração importante no tratamento.

Com essa estrutura modular e resiliente, o sistema Balvia assegura uma integração eficaz entre os dispositivos, promovendo uma monitorização contínua e confiável da saúde do paciente, além de oferecer uma experiência de utilização intuitiva e segura. Os detalhes técnicos da implementação da aplicação móvel, incluindo a sua estrutura interna e arquitetura de software, serão abordados no Capítulo 4.



Figura 3.5: Arquitetura da Solução

### 3.4 Conclusão

Neste capítulo, foram apresentados os requisitos e a arquitetura do sistema Balvia, estabelecendo as bases para o seu desenvolvimento e funcionamento. A definição dos requisitos

permitiu identificar as necessidades dos utilizadores, assegurando que a solução seja eficaz e responda às expectativas dos diferentes perfis de utilizadores, incluindo pacientes, cuidadores e gestores.

A arquitetura do sistema foi concebida para garantir uma comunicação fluida e segura entre a aplicação móvel, o dispensador inteligente e os diversos intervenientes no processo terapêutico. Esta estrutura robusta e escalável facilita a gestão da administração de medicamentos, promove a adesão ao tratamento e melhora significativamente a experiência de utilização.

Com os requisitos claramente definidos e uma arquitetura sólida, o sistema Balvia está preparado para oferecer uma solução tecnológica confiável e eficiente. A aplicação garante não apenas o acompanhamento remoto e automatizado da administração de medicamentos, mas também uma experiência de utilização intuitiva e segura, contribuindo para uma gestão mais eficaz da saúde dos pacientes.

# Capítulo 4

## Desenvolvimento

Neste capítulo são descritas as principais etapas do desenvolvimento da aplicação móvel Balvia, desenvolvida no âmbito do presente projeto. São evidenciados os aspetos técnicos mais relevantes, bem como as dificuldades enfrentadas ao longo do processo de implementação e as respetivas decisões e soluções adotadas.

Para além da descrição funcional e técnica, são apresentados mockups que serviram de base para o desenvolvimento da interface gráfica da aplicação, bem como capturas de ecrã da versão final da aplicação móvel. Cada interface é devidamente contextualizada, explicando-se o seu propósito, os fluxos de interação com o utilizador e a forma como cada componente se integra no sistema global.

Adicionalmente, é abordada a arquitetura da aplicação, as tecnologias utilizadas, as bibliotecas externas incorporadas e os principais mecanismos de comunicação com dispositivos e sistemas externos, nomeadamente o servidor backend e o dispositivo de dispensação via ligação Bluetooth.

Este capítulo tem, assim, como objetivo fornecer uma visão abrangente e fundamentada do processo de desenvolvimento técnico da solução proposta.

## 4.1 Arquitetura da Aplicação

Após a definição da arquitetura global do sistema no Capítulo 3, esta secção descreve a arquitetura interna da aplicação móvel Balvia, detalhando a sua implementação técnica. A aplicação móvel Balvia foi desenvolvida utilizando o framework Flutter e a linguagem Dart, com o objetivo de proporcionar uma experiência fluida, multiplataforma e de elevada performance. A estrutura da aplicação foi concebida com base em princípios de modularidade, separação de responsabilidades e reutilização de componentes, o que facilitou a escalabilidade e a manutenção do código ao longo do desenvolvimento.

### 4.1.1 Estrutura Modular

O projeto foi organizado em diferentes camadas e pastas, cada uma com uma responsabilidade clara:

- **models** – Contém as classes que representam os modelos de dados utilizados na aplicação, como *User*, *Schedule*, *Medication*, entre outros. Estes modelos são utilizados para mapear e manipular dados provenientes da base de dados local ou de APIs.
- **services** – Inclui a lógica de comunicação com o servidor e com dispositivos externos. Aqui estão implementadas funcionalidades como chamadas HTTP para a API, bem como o serviço de comunicação com dispositivos Bluetooth via o pacote `flutter_blue_plus`.
- **providers** – Contém os gestores de estado da aplicação, implementados com o padrão *Provider*. Cada provider é responsável por gerir o estado de uma funcionalidade específica, como autenticação, calendário ou dispensador.
- **screens** – Contém as diferentes páginas da interface gráfica da aplicação (ecrãs). Cada *screen* representa uma funcionalidade distinta, como o registo de utilizador, o painel do cuidador, o calendário, entre outros.

- **widgets** – Inclui componentes reutilizáveis da interface de utilizador, como botões personalizados, caixas de diálogo, e listas de elementos com design consistente.

### 4.1.2 Padrão de Arquitetura

A arquitetura segue uma abordagem baseada no padrão MVVM (Model-View-ViewModel), em que:

- **Model** – Representa os dados e regras de negócio, definidos nas classes dentro de `models`.
- **View** – Refere-se à interface gráfica (UI), construída com os `widgets` Flutter nas pastas `screens` e `widgets`.
- **ViewModel** – Implementado nos `providers`, é responsável por mediar a interação entre a interface gráfica e os dados, reagindo a alterações de estado e atualizando a UI em tempo real.

Esta separação contribuiu para um código mais limpo e testável, com menor acoplamento entre as diferentes camadas.

### 4.1.3 Gestão de Estado

A gestão de estado foi implementada com o pacote `provider`, que oferece uma forma eficiente de partilhar dados entre `widgets` e reagir a alterações de forma reativa. Cada módulo funcional da aplicação possui o seu próprio `provider`, o que garante que os dados e a lógica estejam encapsulados e centralizados.

### 4.1.4 Persistência de Dados

Para o armazenamento local, foi utilizado o **SQLite**, permitindo que a aplicação funcionasse em modo offline. A persistência dos dados é feita através de modelos que interagem

com a base de dados de forma estruturada e segura. Esta abordagem é particularmente relevante para garantir que os dados relacionados com a medicação e a agenda do utilizador se mantenham acessíveis mesmo sem ligação à internet.

#### **4.1.5 Comunicação com Dispositivos Bluetooth**

A comunicação com os dispositivos de dispensação foi realizada através de Bluetooth Low Energy (BLE), utilizando o pacote `flutter_blue_plus`. Esta integração foi encapsulada num serviço específico na camada `services`, o que facilitou a reutilização do código e a manutenção da lógica de conexão, leitura e escrita de dados com o hardware.

#### **4.1.6 Navegação e Fluxo da Aplicação**

A navegação entre ecrãs foi gerida através do sistema de rotas nomeadas do Flutter. Adicionalmente, foram implementados controlos de autenticação e permissões que definem o acesso a determinadas áreas da aplicação consoante o perfil do utilizador (paciente, cuidador informal, cuidador formal ou gestor).

### **4.2 Mockups e Protótipos**

A Figura 4.1 e a Figura 4.2 apresentam os mockups iniciais da interface da aplicação, onde o utilizador pode aceder às opções de autenticação, nomeadamente o login e o registo. A primeira figura ilustra a página inicial da aplicação, enquanto a segunda mostra o ecrã de registo de novos utilizadores.

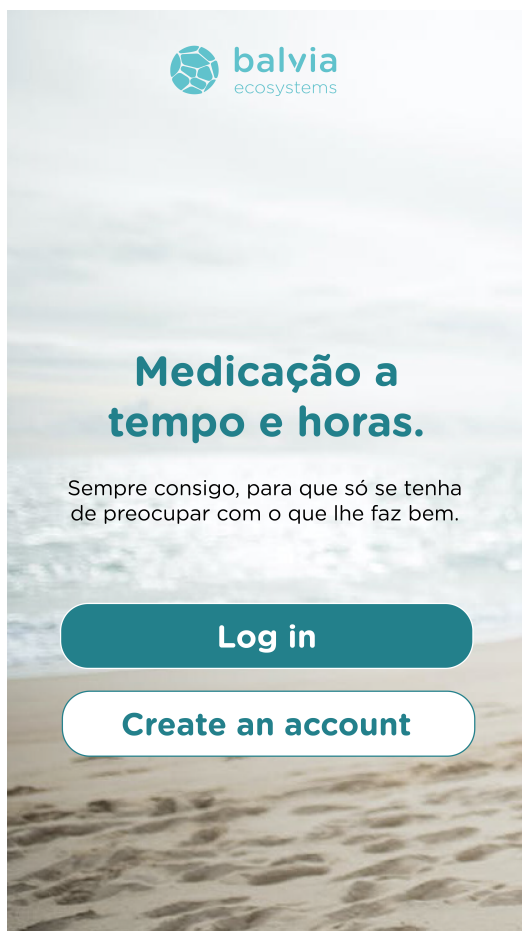


Figura 4.1: Mockup de Página Inicial.

Figura 4.2: Mockup de Página de Registo.

A Figura 4.3 e a Figura 4.4 apresentam os mockups do menu lateral (drawer) e da página de calendário da aplicação. O menu é adaptado ao perfil do utilizador, ao passo que o calendário possibilita o acompanhamento das tomas de medicação previstas.

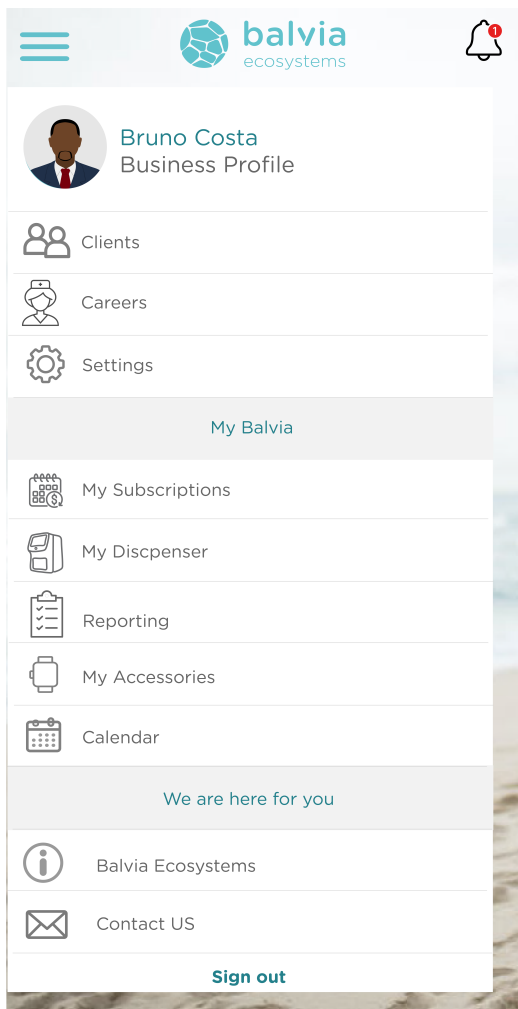


Figura 4.3: Mockup do Menu Lateral.

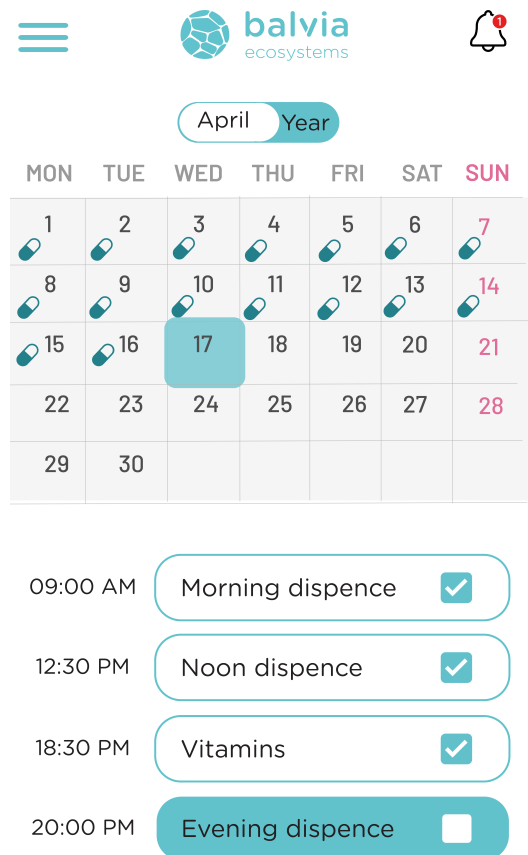
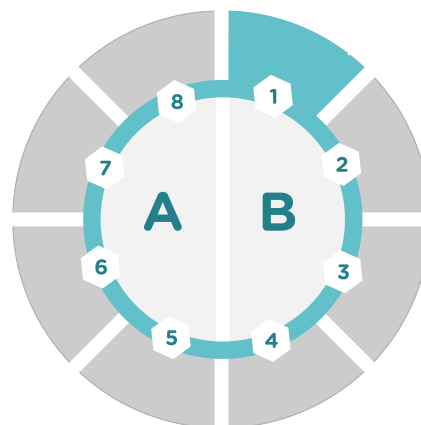
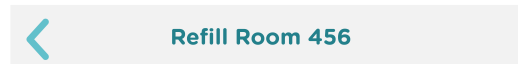


Figura 4.4: Mockup de Página de Calendário.

As Figuras 4.5 e 4.6 apresentam os mockups correspondentes às etapas iniciais do processo de reabastecimento, nomeadamente a ligação ao dispensador e a instrução visual para o reabastecimento de um módulo específico.



Figura 4.5: Mockup de Página de Reabastecimento.



**Please remove and refill  
MODULE 1**



Figura 4.6: Mockup da Etapa de Reabastecimento de um Módulo Específico.

Por fim, as Figuras 4.7 e 4.8 apresentam os mockups correspondentes à conclusão do processo de reabastecimento e à notificação ao utilizador, através do ecrã de alarme, para a toma do medicamento.

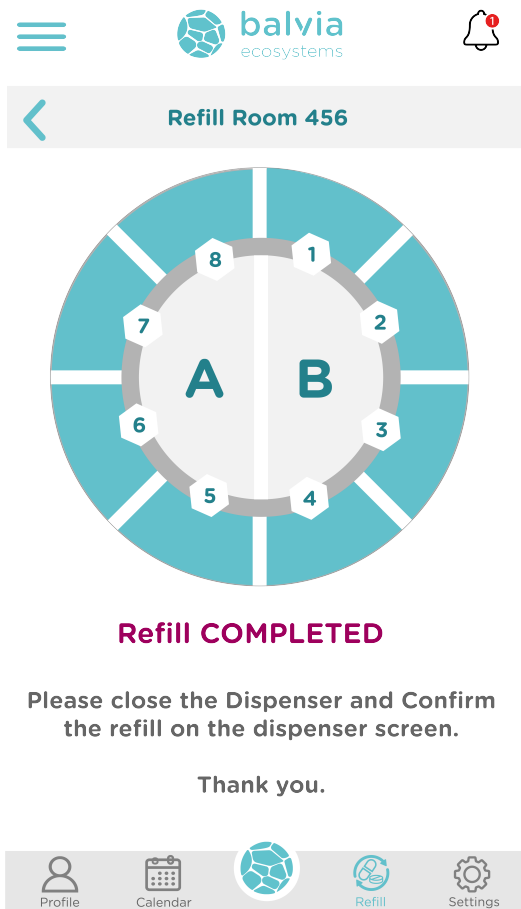


Figura 4.7: Mockup do Reabastecimento com Instruções de Encerramento.

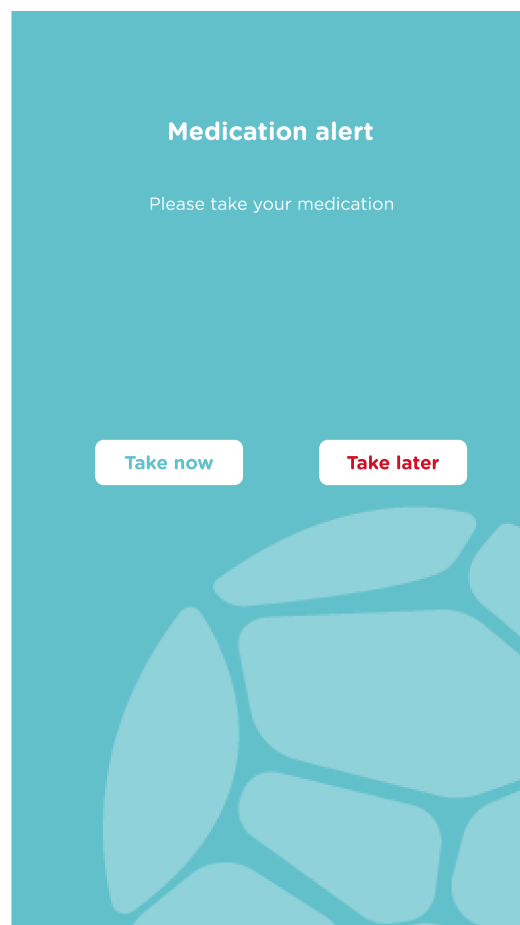


Figura 4.8: Mockup de Alarme para Notificação da Toma do Medicamento.

### 4.3 Funcionalidades Implementadas

As Figuras 4.9 e 4.10 apresentam as interfaces iniciais da aplicação, incluindo o ecrã principal com opções de login e registo, bem como o ecrã de login onde o utilizador insere as suas credenciais.



Figura 4.9: Interface Inicial.

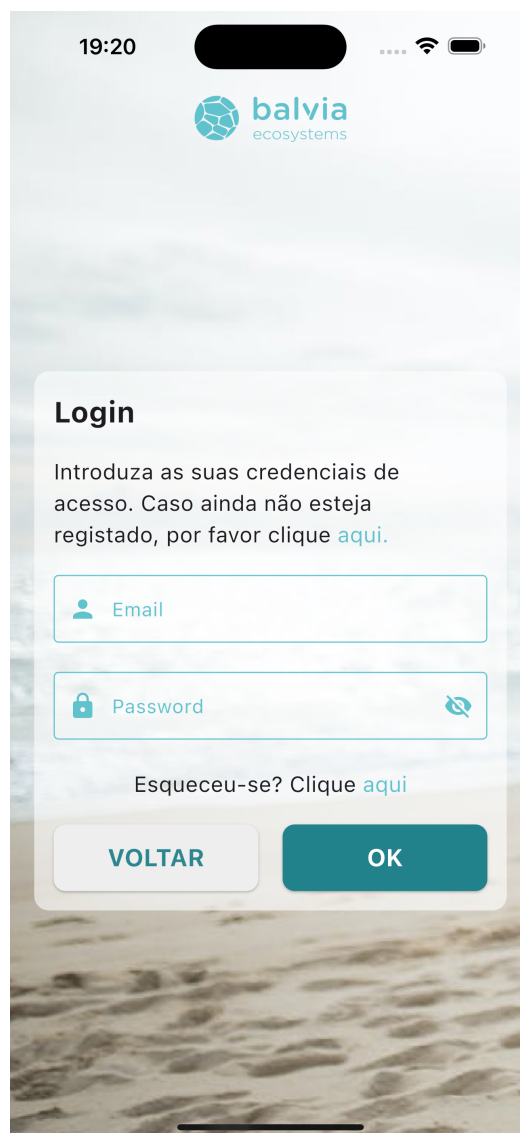


Figura 4.10: Interface de Login.

As Figuras 4.11 e 4.12 apresentam, respetivamente, as interfaces de registo e do painel principal (dashboard) da aplicação. A interface de registo permite ao utilizador criar uma nova conta, introduzindo os seus dados pessoais de forma estruturada. Já o dashboard oferece uma visão geral das funcionalidades disponíveis, adaptadas ao perfil do utilizador, como o acesso ao calendário de medicação, histórico de utilização e definições da conta.

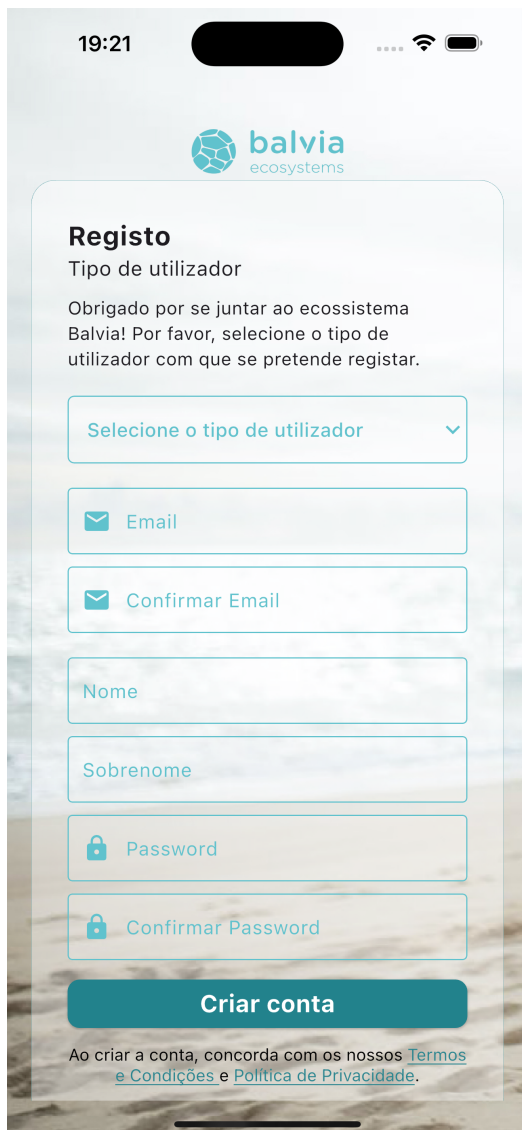


Figura 4.11: Interface de Registo.

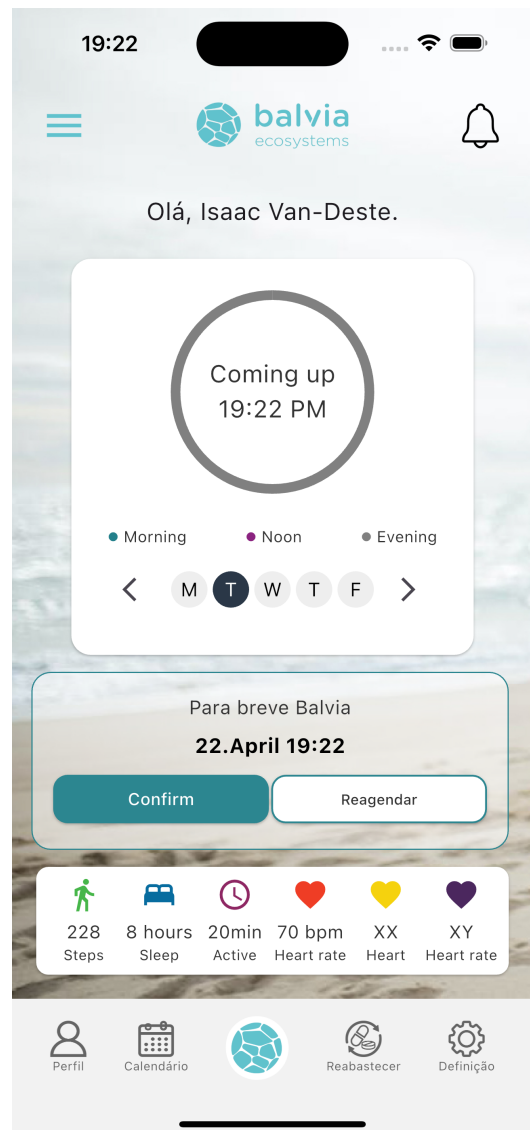


Figura 4.12: Interface Dashboard.

A Figura 4.13 e a Figura 4.14 apresentam, respetivamente, o menu lateral (drawer) e a interface de calendário da aplicação. O menu lateral ajusta-se consoante o perfil do utilizador, disponibilizando funcionalidades adequadas ao seu tipo de acesso. Já a interface de calendário permite ao utilizador visualizar e acompanhar de forma organizada as tomas de medicação agendadas, facilitando a gestão do tratamento.

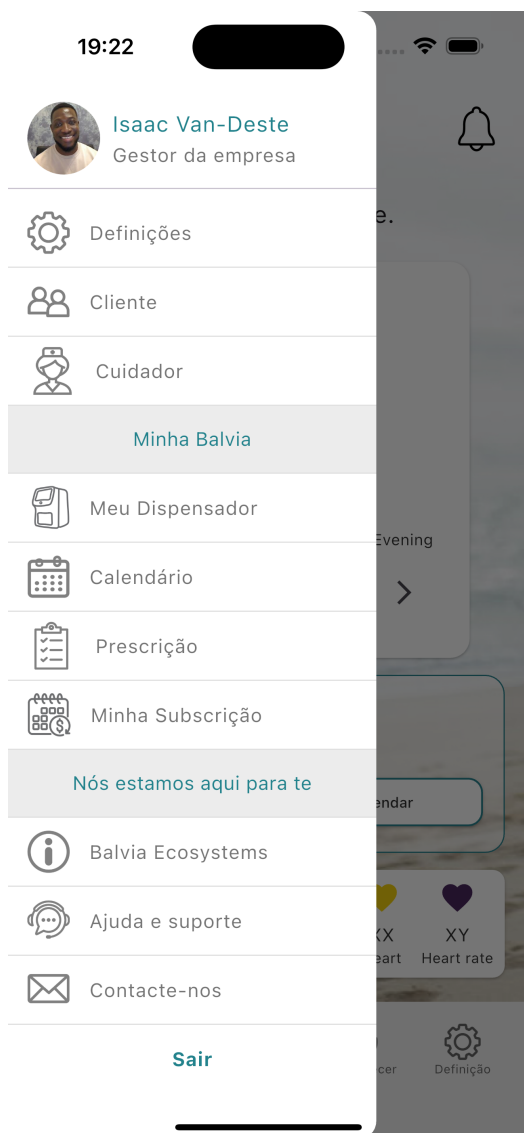


Figura 4.13: Interface Menu lateral.

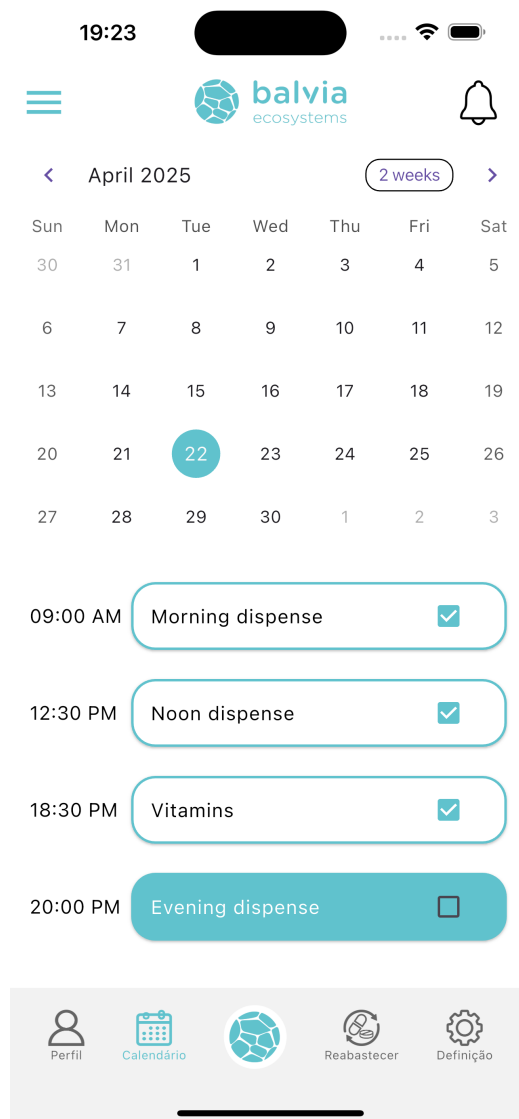


Figura 4.14: Interface Calendário.

A Figura 4.15 e a Figura 4.16 apresentam, respetivamente, a interface de gestão dos dispensadores e a interface de configuração dos horários e medicação. Através destas funcionalidades, o utilizador pode consultar informações detalhadas sobre os dispositivos associados, bem como configurar e gerir as tomas de medicação de forma eficiente e personalizada.



Figura 4.15: Interface Dispensador.

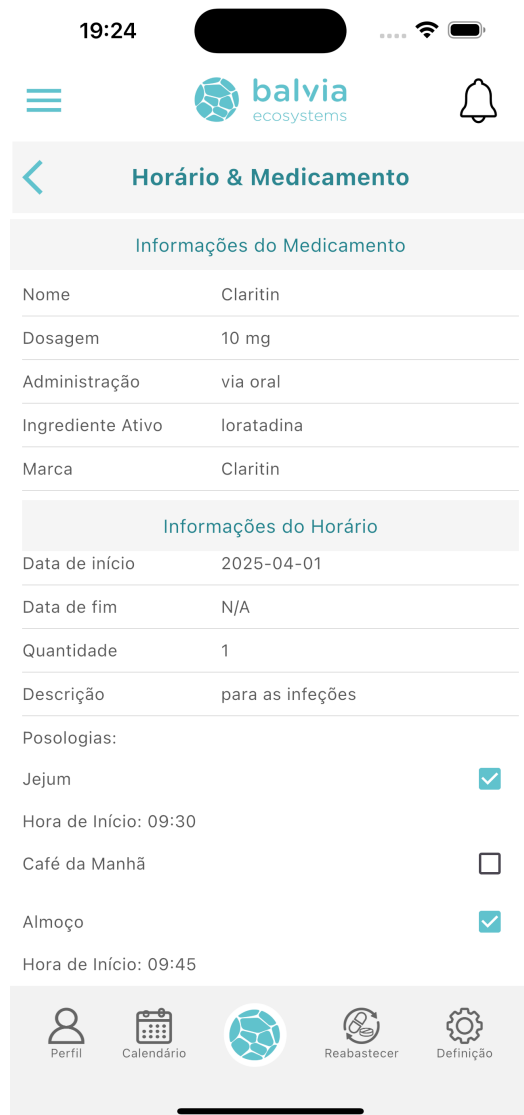


Figura 4.16: Interface de Horários e Medicamentos.

Por fim, a Figura 4.17 e a Figura 4.18 apresentam, respetivamente, a funcionalidade de reabastecimento do dispensador e a interface de alarme exibida no momento da toma da medicação. Estas interfaces permitem ao utilizador concluir o processo de reabastecimento de forma orientada e garantem a notificação atempada para a administração correta dos medicamentos.

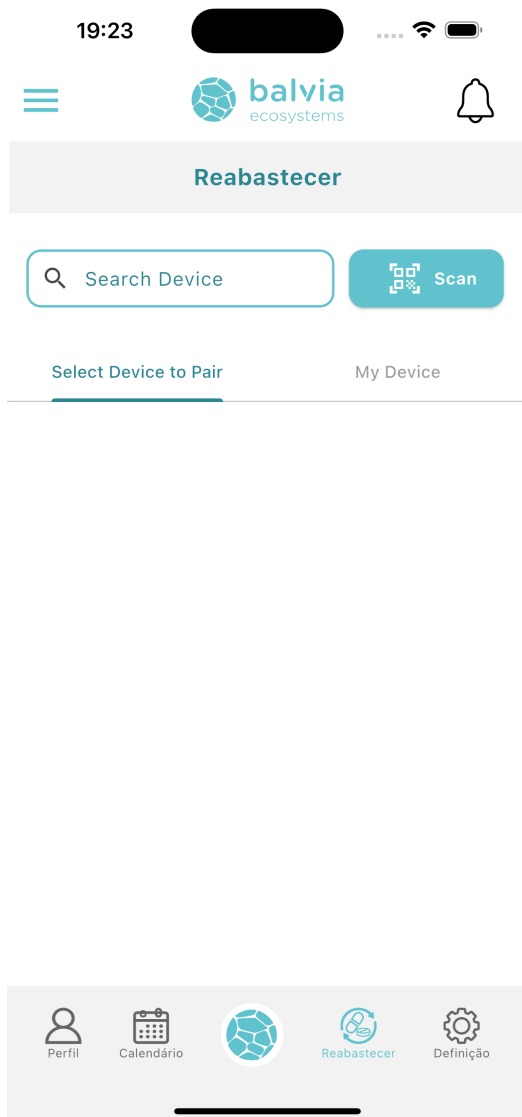


Figura 4.17: Tela de Reabastecimento do Dispensador via Bluetooth.

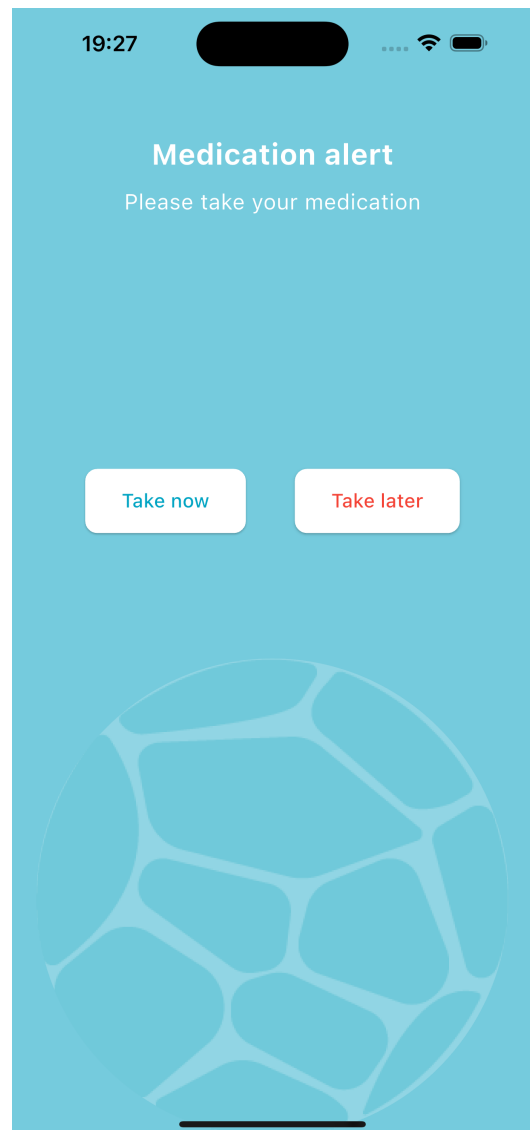


Figura 4.18: Interface da tela de Alarme para a toma da medicação.

## 4.4 Código Desenvolvido

Nesta secção apresenta-se um exemplo representativo da implementação do padrão *Provider* para a gestão do estado de autenticação na aplicação móvel Balvia.

### 4.4.1 Provider para Gestão de Autenticação

```

// Classe singleton que gere o estado de autentica o
class AuthProvider extends ChangeNotifier {
    static final AuthProvider _instance = AuthProvider._internal();
    bool _isLoggedIn = false;
    String? _token;

    bool get isLoggedIn => _isLoggedIn;
    String? get token => _token;

    factory AuthProvider() => _instance;
    AuthProvider._internal();

    // Efetua login e guarda token localmente
    Future<void> login(String token) async {
        _isLoggedIn = true;
        _token = token;
        final prefs = await SharedPreferences.getInstance();
        await prefs.setString('auth_token', token);
        notifyListeners();
    }

    // Efetua logout e remove token
    Future<void> logout() async {
        _isLoggedIn = false;
        _token = null;
        final prefs = await SharedPreferences.getInstance();
        await prefs.remove('auth_token');
        notifyListeners();
    }
}

```

```

// Verifica estado de autenticação ao iniciar a app
Future<void> checkLoginStatus() async {
    final prefs = await SharedPreferences.getInstance();
    final savedToken = prefs.getString('auth_token');
    _isLoggedIn = savedToken != null;
    _token = savedToken;
    notifyListeners();
}
}

```

Esta classe implementa um *singleton* que mantém o estado de autenticação do utilizador, armazenando o token de acesso de forma persistente usando *SharedPreferences*. O uso do *ChangeNotifier* permite que a interface reaja automaticamente às alterações do estado, promovendo uma arquitectura reativa e eficiente.

#### 4.4.2 Estrutura da Mensagem para Comunicação com o Dispensador

```

// Classe que representa a mensagem trocada com o dispensador
class Message {
    int? version = 2;           // Versão do protocolo
    int? command;              // Código do comando
    int? length;               // Comprimento dos parâmetros
    String? parameters;        // Parâmetros da mensagem

    // Construtor para criar mensagem a partir de comando e parâmetros
    Message(this.command, this.parameters) {
        length = parameters?.length ?? 0;
    }

    // Construtor para criar mensagem a partir de string formatada

```

```

Message.fromRaw(String raw) {
    var parts = raw.split('-');
    version = int.parse(parts[0]);
    command = int.parse(parts[1]);
    length = int.parse(parts[2]);
    parameters = parts[3];
}

// Converte a mensagem para o formato de string esperado
@override
String toString() => '$version-$command-$length-$parameters';
}

```

Esta classe encapsula a estrutura da mensagem usada para comunicação entre a aplicação e o dispensador, adotando um formato simples em texto separado por hífenes. A versão do protocolo, o comando a executar, o comprimento dos parâmetros e os parâmetros propriamente ditos são armazenados e manipulados, facilitando o envio e recepção de mensagens. A conversão entre a representação interna e a string formatada garante interoperabilidade clara e consistente no protocolo de comunicação.

### 4.4.3 Conexão e Comunicação via Bluetooth

```

Future<void> connectToDevice(BluetoothDevice device, BuildContext context)
async {
    if (_connectedDevices[device] == true) return;

    try {
        await device.connect();
        _connectedDevices[device] = true;
        notifyListeners();
    }
}

```

```

// Descubre servi os e caracter sticas
List<BluetoothService> services = await device.discoverServices();
for (var service in services) {
  for (var characteristic in service.characteristics) {
    if (characteristic.properties.write) {
      _writeCharacteristic = characteristic;

      // Envia mensagem inicial ao m dulo 1
      if (currentModule == 1) {
        Message message = Message(403, "$currentModule");
        await sendString(message.toString());

        Navigator.push(context,
          MaterialPageRoute(
            builder: (_) => RefillPage(deviceName: device.name,
              currentModule: currentModule)
          )
        );
      }
    }

    if (characteristic.properties.read) {
      _readCharacteristic = characteristic;
      await characteristic.setNotifyValue(true);

      // Escuta notifica es recebidas
      characteristic.value.listen((value) {
        String receivedString = utf8.decode(value);
        if (receivedString.isNotEmpty) {

```

```

        handleReceivedMessage(receivedString, context, device.name);
    }
    });
}
}
}
} catch (e) {
    print('Erro na conexão: $e');
}
}

```

Esta função gere a conexão a um dispositivo Bluetooth, evita múltiplas conexões ao mesmo dispositivo e, após estabelecer a ligação, descobre os serviços e características oferecidas. Identifica características que permitem escrita e leitura, armazenando-as para comunicação posterior. Envia uma mensagem inicial ao dispositivo (especificamente ao módulo 1) e navega para uma página de reabastecimento. Também ativa notificações para receber mensagens do dispositivo, tratando-as conforme chegam, permitindo comunicação bidirecional em tempo real entre a aplicação e o dispensador.

#### 4.4.4 Serviço de Sincronização de Dados

```

// Serviço que sincroniza dados locais com o servidor remoto
class SyncService {
    final DatabaseHelper dbHelper = DatabaseHelper();

    Future<void> syncData(BuildContext context) async {
        final authProvider = Provider.of<AuthProvider>(context, listen: false);

        // Sincroniza itens não sincronizados da tabela Schedule
        List<Map<String, dynamic>> unsyncedSchedules =
            await dbHelper.retrieveUnsyncedItems();
    }
}

```

```

for (var schedule in unsyncedSchedules) {
    bool success = await _sendItemToServer(schedule, 'schedule',
        authProvider);
    if (success) {
        await dbHelper.markItemAsSynced(schedule['id']);
    }
}

// Sincroniza itens n o sincronizados da tabela Log
List<Map<String, dynamic>> unsyncedLogs =
await dbHelper.retrieveUnsyncedItems();
for (var log in unsyncedLogs) {
    bool success = await _sendItemToServer(log, 'log', authProvider);
    if (success) {
        await dbHelper.markItemAsSynced(log['id']);
    }
}
}
}

```

O `SyncService` é responsável por garantir que os dados registados localmente na aplicação sejam sincronizados com o servidor remoto. Através da função `syncData`, são recuperados itens que ainda não foram sincronizados das tabelas `Schedule` e `Log`. Para cada item, é efetuada uma tentativa de envio para o servidor utilizando as credenciais providenciadas pelo `AuthProvider`. Após confirmação do sucesso, o item é marcado localmente como sincronizado para evitar envios duplicados. Esta abordagem permite manter a integridade e a atualização dos dados entre o dispositivo do utilizador e o backend, mesmo em cenários de conectividade intermitente.

#### 4.4.5 Verificação e Disparo de Alarmes

```

void checkAndSetAlarms(ScheduleModel schedule,
DateTime now, BuildContext context) {
    if (schedule.startDate != null &&
        schedule.startDate!.isBefore(now) &&
            (schedule.endDate == null || !schedule.endDate!.isBefore(now))) {
        _triggerAlarmsForPosologies(schedule, now, context);
    }
}

// Dispara notificações para cada posologia ativa e no horário correto
void _triggerAlarmsForPosologies(ScheduleModel schedule,
DateTime now, BuildContext context) {
    if (schedule.fast == true && compareTime(
        schedule.fastTime, now)) {
        _showNotification(
            'Alarme de Jejum', 'hora de tomar o remédio em jejum', context);
    }
    if (schedule.breakfast ==
true && compareTime(schedule.breakfastTime, now)) {
        _showNotification('Alarme de Café da Manhã',
            'hora de tomar o remédio no café da manhã', context);
    }
    // ... demais posologias seguem o mesmo padrão
}

```

O método `checkAndSetAlarms` avalia se o horário atual está dentro do período válido definido entre `startDate` e `endDate` para o agendamento (`ScheduleModel`). Caso afirmativo, chama-se `_triggerAlarmsForPosologies`, que verifica para cada tipo de posologia (jejum, café da manhã, almoço, etc.) se está ativa e se o horário coincide com o atual (usando a função `compareTime`). Para cada posologia ativa e no horário correto, é

disparada uma notificação local para alertar o utilizador para tomar o medicamento.

Esta implementação garante que os alarmes sejam disparados apenas dentro do intervalo válido do agendamento e respeitando os horários específicos de cada tipo de posologia, promovendo uma gestão precisa e personalizada das notificações de medicação.

## 4.5 Dificuldades Encontradas e Soluções

Durante o desenvolvimento da aplicação móvel em Flutter, surgiram diversos desafios técnicos relacionados com a gestão de estados, interface dinâmica, comunicação com dispositivos via Bluetooth e integração com serviços externos. Abaixo descrevem-se algumas das principais dificuldades encontradas, bem como as soluções aplicadas:

- **Atualização de Agendamentos com Múltiplas Posologias:** A estrutura inicial não permitia representar várias posologias e respetivos horários por agendamento. Este problema foi resolvido através da criação de um modelo auxiliar (`AuxScheduleModel`) que permitiu mapear diferentes tipos de toma (jejum, pequeno-almoço, almoço, etc.) com os respetivos horários, facilitando tanto a visualização como a edição.
- **Inicialização Correta da Interface com Dados Existentes:** Foi identificada uma dificuldade ao carregar os dados de posologias e horários já existentes na interface de edição. A solução passou pela correta implementação da lógica de `initState`, garantindo a marcação automática das checkboxes e a associação dos campos de horário aos dados previamente guardados.
- **Gestão de Overflow em Dispositivos iOS:** Em dispositivos com ecrãs menores (como o iPhone 12), a aplicação apresentava erros de `RenderFlex overflowed`. Esse problema foi resolvido com o uso de widgets como `Flexible` e `Expanded`, bem como ajustes nos layouts com margens e tamanhos relativos.
- **Envio e Receção de Mensagens via Bluetooth (ESP32):** Durante a comunicação com o dispensador, houve dificuldade em distinguir as mensagens enviadas

daquelas recebidas. A lógica foi ajustada para que a aplicação apenas processasse mensagens recebidas a partir da característica de notificação, ignorando os dados ecoados localmente.

- **Navegação Condicionada Após Conexão Bluetooth:** Inicialmente, ao tentar realizar múltiplas navegações seguidas após conectar ao dispositivo, o fluxo não funcionava como esperado. Este problema foi resolvido centralizando a navegação numa única função acionada após a receção de confirmação do dispositivo.
- **Gestão de Seções de Reabastecimento:** A implementação de um sistema de reabastecimento por módulo (1 a 8) exigiu a criação de lógica sequencial com envio de mensagens (403, *n*) e receção de respostas específicas para cada módulo. O controlo do estado atual do reabastecimento foi feito com incremento de um índice de módulo e controlo por *Provider*.
- **Exibição e Edição de Dados de Medicamento:** A integração de campos como unidade, quantidade, e tipo de secção exigiu revisão nos modelos de dados e na interface. Foram aplicados *Dropdowns*, *Checkboxes* e *Controllers* para permitir uma edição intuitiva e segura.
- **Conectividade Bluetooth com o Dispensador:** Inicialmente, a aplicação apresentava instabilidade na conexão Bluetooth com o dispositivo ESP32. Este problema foi resolvido através da otimização do fluxo de descoberta de serviços e características, bem como do tratamento adequado de permissões em dispositivos Android e iOS.
- **Tratamento de Notificações e Alarmes:** Houve dificuldades na gestão de notificações em segundo plano, especialmente em dispositivos Android mais recentes com restrições de background. Recorreu-se à biblioteca `flutter_local_notifications` e à configuração adequada de canais de notificação para garantir a fiabilidade dos alertas.

- **Adaptação da Interface ao Perfil do Utilizador:** A necessidade de personalizar menus e funcionalidades consoante o tipo de utilizador (paciente, cuidador, gestor) exigiu uma arquitetura modular e reutilizável. Foi implementado um sistema de controlo de acesso com base no perfil autenticado, permitindo a exibição dinâmica das opções relevantes.
- **Tratamento de Erros e Experiência do Utilizador:** Identificou-se a necessidade de fornecer feedback claro ao utilizador em caso de falhas (ex.: falha de sincronização com o dispensador). Foram introduzidas mensagens de erro, ícones informativos e estados de carregamento que melhoraram a usabilidade da aplicação.
- **Integração com API Externa:** A comunicação com o backend implicou a conversão correta de modelos em JSON, tratamento de respostas assíncronas e configuração de cabeçalhos de autenticação. Bibliotecas como `http` e `dio` foram utilizadas para gerir essas integrações de forma segura e eficiente.

## 4.6 Validação Técnica e Resultados

Durante as diversas fases do desenvolvimento da aplicação Balvia, foram realizados testes técnicos e validações funcionais que permitiram aferir a robustez e a qualidade das funcionalidades implementadas. Esta secção descreve os principais resultados obtidos ao longo desse processo, bem como os pontos fortes e as melhorias identificadas durante o ciclo de desenvolvimento.

- **Testes funcionais e integração:** As funcionalidades principais, como a gestão de prescrições, o reabastecimento de dispensadores e a navegação entre perfis de utilizador, foram testadas com sucesso, revelando estabilidade e usabilidade.
- **Interação com dispositivos:** A integração com o dispensador via Bluetooth revelou-se funcional, permitindo comunicação bidirecional eficaz. No entanto, algumas instabilidades ocasionais exigiram tratamentos adicionais de exceções.

- **Feedback preliminar:** Utilizadores-teste destacaram a simplicidade da interface e a fluidez da navegação como pontos positivos. O processo de autenticação e registo foi considerado intuitivo.
- **Melhorias identificadas:** Foi reconhecida a necessidade de incluir mensagens de erro mais claras, bem como indicadores visuais de progresso durante operações críticas, como o reabastecimento.

## 4.7 Conclusão

O desenvolvimento da aplicação constituiu uma etapa fundamental deste projeto, durante a qual foram abordadas e ultrapassadas diversas dificuldades técnicas e conceptuais. Através da definição de uma arquitetura robusta e modular, foi possível estruturar a aplicação de forma escalável e sustentável, facilitando futuras extensões e operações de manutenção.

As funcionalidades implementadas refletem os requisitos previamente definidos, com especial ênfase na interação com o dispensador via Bluetooth, na gestão dinâmica de agendamentos com múltiplas posologias e na personalização da interface consoante o perfil do utilizador. Estas capacidades foram validadas com base em protótipos e mockups, concebidos para garantir uma experiência intuitiva e acessível.

As soluções encontradas para os desafios enfrentados, como o tratamento de notificações, a comunicação em tempo real com o dispositivo ou a adaptação da interface a diferentes ecrãs, contribuíram significativamente para o amadurecimento do sistema e para a consolidação das competências técnicas envolvidas.

Em suma, este capítulo evidenciou a complexidade do projeto e a capacidade de integração entre diferentes componentes tecnológicos, resultando numa aplicação funcional, coesa e alinhada com os objetivos traçados.

# Capítulo 5

## Conclusão

Neste capítulo, apresentam-se as conclusões do projeto Balvia, sintetizando os principais resultados alcançados e refletindo sobre o impacto da solução desenvolvida. O sistema Balvia foi concebido com o objetivo de otimizar o processo de gestão e administração de medicamentos, proporcionando maior segurança e eficiência a pacientes e cuidadores.

Ao longo do desenvolvimento, foram ultrapassados desafios relacionados com a comunicação Bluetooth com o dispensador de medicamentos, assegurando uma ligação estável e fiável. Adicionalmente, o sistema demonstrou ser capaz de operar eficazmente em modo offline, com sincronização automática dos dados quando a ligação é restabelecida, garantindo assim a continuidade do serviço.

Os testes funcionais e de usabilidade realizados confirmaram que a aplicação satisfaz os requisitos definidos, oferecendo uma interface intuitiva e funcionalidades robustas. O sistema mostrou-se eficiente na gestão de prescrições, no reabastecimento do dispensador, na monitorização de sinais vitais e no envio de notificações relacionadas com a toma da medicação.

Entre as lições aprendidas, destaca-se a importância de uma interface clara e intuitiva para facilitar a utilização por diferentes perfis de utilizadores, nomeadamente pacientes e cuidadores. O processo de desenvolvimento evidenciou ainda a relevância da sincronização offline/online como elemento essencial para assegurar a continuidade do serviço em diversos contextos de utilização.

Para trabalhos futuros, recomenda-se a exploração da integração com outros dispositivos de monitorização de saúde, o aprimoramento da personalização das notificações e a expansão do suporte a diferentes idiomas. Para além disso, seria pertinente investigar o potencial da inteligência artificial na análise dos dados dos pacientes, com vista à geração de recomendações personalizadas.

Desta forma, o sistema Balvia cumpre o seu propósito de oferecer uma solução eficiente e segura para a gestão de medicamentos, promovendo a adesão ao tratamento e contribuindo para a melhoria da qualidade de vida dos seus utilizadores.

# Bibliografia

- [1] *Digital Health • Platform • Adherence • Persistence • Solutions / Medisafe*. URL: <https://www.medisafe.com/>.
- [2] *Medication Reminder and Pill Tracker App ??[Clock Emoji]??– MyTherapy*. URL: <https://www.mytherapyapp.com/>.
- [3] *CareZone | Easily manage multiple medications and health info*. URL: <https://carezone.com/>.
- [4] B. M. Silva, I. M. Lopes, M. B. Marques, J. J. P. C. Rodrigues e M. L. Proença, “A mobile health application for outpatients medication management,” em *2013 IEEE International Conference on Communications (ICC)*, 2013, pp. 4389–4393. DOI: 10.1109/ICC.2013.6655256.
- [5] F. Oprea, D. Rosner, F. Popescu e A. Scrab, “Mediminder – Medication Management and Reminder Application,” em *2021 20th RoEduNet Conference: Networking in Education and Research (RoEduNet)*, 2021, pp. 1–6. DOI: 10.1109/RoEduNet54112.2021.9637712.
- [6] *Dart overview | Dart*. URL: <https://dart.dev/overview>.
- [7] *Flutter - Build apps for any screen*. URL: <https://flutter.dev/>.
- [8] *SQLite Home Page*. URL: <https://www.sqlite.org/>.
- [9] *flutter\_blue\_plus|Flutterpackage*. URL: [https://pub.dev/packages/flutter\\_blue\\_plus](https://pub.dev/packages/flutter_blue_plus).
- [10] *Trello | Atlassian*. URL: <https://www.atlassian.com/br/software/trello>.

- [11] *Postman: The World's Leading API Platform | Sign Up for Free*. URL: <https://www.postman.com/>.
- [12] *Xcode - Apple Developer*. URL: <https://developer.apple.com/xcode/?cid=ht-xcode>.
- [13] *DataGrip: The Cross-Platform IDE for Databases & SQL by JetBrains*. URL: <https://www.jetbrains.com/datagrip/>.
- [14] *Fazer o download do Android Studio e das ferramentas de apps: desenvolvedores Android | Android Developers*. URL: <https://developer.android.com/studio?hl=pt-br>.
- [15] *Free UML, BPMN and Agile Tutorials - Learn Step-by-Step*. URL: <https://www.visual-paradigm.com/tutorials/>.
- [16] *LaTeX editor features benefits | Overleaf - Overleaf, Editor LaTeX Online*. URL: <https://pt.overleaf.com/about/features-overview>.
- [17] *Mendeley | Free reference manager | Elsevier*. URL: <https://www.elsevier.com/products/mendeley>.
- [18] K. Wiegers e J. Beatty, *Software Requirements, Third Edition*.