



Allergy Tracker

Aplicação Móvel para Gestão de Alergias

Rafael Granja

Junho de 2025

Agradecimentos

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetur id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Sumário

Agradecimentos	1
Lista de Siglas	5
1 Introdução	7
1.1 Enquadramento	7
1.2 Motivação	8
1.3 Objetivos	8
1.4 Organização do Documento	9
2 Estado da Arte	10
2.1 Aplicações de Gestão de Alergias	10
2.2 APIs Meteorológicas	12
2.3 Sistemas de Notificação Móvel	13
3 Tecnologias Utilizadas e Engenharia de Software	14
3.1 Android SDK e Jetpack	14
3.2 SQLite e DatabaseHelper	14
3.3 Location Services e AlarmManager	15
4 Implementação e Testes	16
4.1 Autenticação e Registro	16
4.1.1 MainActivity (Login)	16
4.1.2 RegisterActivity	16
4.2 Perfil de Usuário	17
4.2.1 ProfileActivity	17
4.2.2 Seleção de Foto e Alergias	17
4.3 Rastreador de Medicação	17
4.4 Lembretes Diários	17
4.5 Integração com Open-Meteo	18

5	Demonstração e Validação	19
5.1	Cenários de Uso	19
5.2	Resultados de Testes Funcionais	20
5.3	Testes de Usabilidade	20
6	Conclusões e Trabalhos Futuros	21
6.1	Principais Conclusões	21
6.2	Trabalhos Futuros	21

Lista de Figuras

2.1	Menu principal da aplicação	11
2.2	A API é usada para ir buscar informações meteorológicas, e dar-lhes um valor de 0-10 calculado, de menor a maior risco para o utilizador, consoante as alergias colocadas no perfil. . .	12
2.3	Como funciona o processo de <i>AlarmManager</i>	13

Lista de Siglas

API

Interface de Programação de Aplicações (Application Programming Interface)

Android SDK

Kit de Desenvolvimento de Software para Android (Android Software Development Kit)

Java

Linguagem de programação utilizada no desenvolvimento

SQLite

Linguagem de Consulta Estruturada Lite, um sistema de banco de dados leve

URI

Identificador Uniforme de Recursos (Uniform Resource Identifier)

CSV

Valores Separados por Vírgula (Comma-Separated Values)

Open-Meteo

API meteorológica utilizada para previsões climáticas

FusedLocationProviderClient

Serviço Android para obtenção de localização por fusão de sensores

AlarmManager

Serviço Android para agendamento de alarmes e notificações

PendingIntent

Conceito Android para intenções adiadas (deferred intents)

Toast

Elemento de interface Android para mensagens temporárias

RecyclerView

Componente Android para exibição eficiente de listas dinâmicas

Jetpack

Conjunto de bibliotecas modernas para desenvolvimento Android

AndroidX

Namespace das bibliotecas Jetpack para compatibilidade e atualizações

RTC

Relógio em Tempo Real (Real-Time Clock), usado para alarmes

UV

Radiação Ultravioleta (Ultraviolet)

REST

Estilo de arquitetura para APIs web (Representational State Transfer)

Capítulo 1

Introdução

1.1 Enquadramento

Devido ao aumento das alergias em todo o mundo, tornou-se cada vez mais importante controlar bem estes problemas. Os telemóveis são uma ótima ferramenta para criar soluções que ajudam as pessoas no seu dia a dia. Este projeto está focado em desenvolver uma aplicação simples e fácil de usar, que ajuda o utilizador a lembrar-se de tomar a medicação e a acompanhar as suas alergias. Também incorporei dados meteorológicos para enviar notificações para tomar remédios para que não agrave as alergias, beneficiando quem é mais sensível. Pensámos muito no design e na facilidade de utilização para que toda a gente consiga usar a aplicação sem dificuldades.

1.2 Motivação

Muitas pessoas têm dificuldade em seguir corretamente os seus tratamentos, o que pode causar problemas de saúde. Ter notificações personalizadas e lembretes é fundamental para ajudar o utilizador a cumprir o regime de medicação. Como não existem muitas soluções completas e fáceis de usar para gerir alergias, este projeto veio preencher essa lacuna. Queremos também ajudar as pessoas a perceber melhor os fatores ambientais que podem aumentar o risco de alergias, para que possam agir com mais precaução. Este trabalho junta tecnologia móvel com recomendações de saúde para facilitar o dia a dia de quem tem alergias

1.3 Objetivos

- Desenvolver uma aplicação Android para gestão de perfis de alergias.
- Implementar rastreador de medicação recomendado por alergia.
- Criar sistema de lembretes diários baseado em dosagens.
- Integrar previsão meteorológica para cálculo de risco alérgico.
- Fornecer interface simples e acessível.

1.4 Organização do Documento

Este relatório está organizado em seis capítulos principais, seguidos pela bibliografia:

1. Introdução
2. Estado da Arte
3. Tecnologias Utilizadas e Engenharia de Software
4. Implementação e Testes
5. Demonstração e Validação
6. Conclusões e Trabalhos Futuros

Capítulo 2

Estado da Arte

Este capítulo apresenta as tecnologias, aplicações e interfaces que serviram de inspiração e base para o desenvolvimento deste projeto, bem como as decisões fundamentais tomadas em termos de interface gráfica e funcionalidades.

2.1 Aplicações de Gestão de Alergias

Atualmente, existem várias aplicações no mercado destinadas à gestão de alergias e condições médicas relacionadas. Muitas destas soluções procuram proporcionar interfaces acessíveis, intuitivas e que ofereçam lembretes oportunos ao utilizador sobre a sua medicação.

Neste contexto, o design visual e a usabilidade são aspetos chave. Foi com esta ideia que optei por desenvolver uma interface gráfica com cores vibrantes e contrastantes, fugindo ao uso comum de cores monotónicas que poderiam resultar numa experiência visual menos agradável e até deprimente.

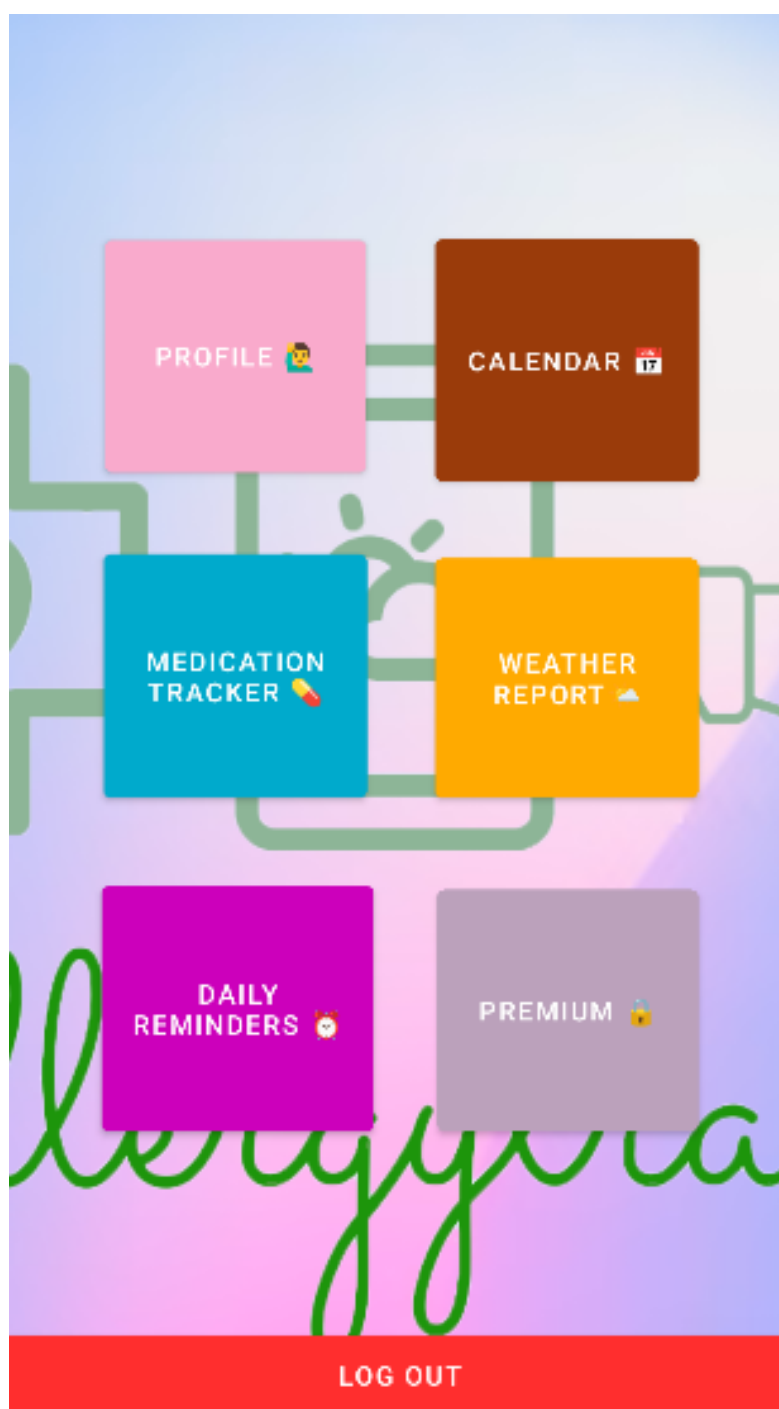


Figura 2.1: Menu principal da aplicação

2.2 APIs Meteorológicas

Uma componente importante da aplicação é o uso de APIs meteorológicas para fornecer previsões relevantes a utilizadores com alergias relacionadas ao clima. Neste projeto, foi utilizada a API **Open-Meteo**, escolhida pela sua simplicidade de uso, fiabilidade e precisão na previsão meteorológica detalhada.

```
1 usage
private List<DailyForecast> mapToDailyForecasts(MeteoResponse resp) {
    List<DailyForecast> out = new ArrayList<>();
    MeteoResponse.Daily d = resp.daily;
    for (int i = 0; i < d.time.size(); i++) {
        DailyForecast f = new DailyForecast();
        f.date          = d.time.get(i);
        f.tempMax       = d.temperature2mMax.get(i);
        f.tempMin       = d.temperature2mMin.get(i);
        f.windGust      = d.windgusts10mMax.get(i);
        f.uvIndex       = d.uvIndexMax.get(i);
        f.precipitation = d.precipitationSum.get(i);
        out.add(f);
    }
    return out;
}
```

Figura 2.2: A API é usada para ir buscar informações meteorológicas, e dar-lhes um valor de 0-10 calculado, de menor a maior risco para o utilizador, consoante as alergias colocadas no perfil.

2.3 Sistemas de Notificação Móvel

A notificação móvel constitui um recurso crítico para garantir que os utilizadores sigam corretamente os seus tratamentos e recebam avisos imediatos. Assim, foram implementadas funcionalidades baseadas no **AlarmManager**, uma tecnologia nativa do Android. Com a implementação dos lembretes, o utilizador recebe notificações específicas de acordo com os horários recomendados para cada medicamento, tornando o tratamento personalizado e eficaz.

```
private void scheduleAlarms() {
    AlarmManager am = (AlarmManager) getSystemService(Context.ALARM_SERVICE);
    if (am == null) return;

    int reqCode = 0;
    for (ReminderItem ri : reminderMap.values()) {
        for (Calendar cal : ri.times) {
            Intent i = new Intent( packageContext: this, ReminderReceiver.class);
            i.putExtra( name: "medText", value: ri.medName + " - " + ri.dosage);
            PendingIntent pi = PendingIntent.getBroadcast(
                context: this, reqCode++, i,
                flags: PendingIntent.FLAG_UPDATE_CURRENT | PendingIntent.FLAG_IMMUTABLE
            );
            am.setInexactRepeating(
                AlarmManager.RTC_WAKEUP,
                cal.getTimeInMillis(),
                AlarmManager.INTERVAL_DAY,
                pi
            );
        }
    }
}
```

Figura 2.3: Como funciona o processo de *AlarmManager*

Capítulo 3

Tecnologias Utilizadas e Engenharia de Software

3.1 Android SDK e Jetpack

O aplicativo foi desenvolvido sobre o **Android SDK** (API nível 35), utilizando a linguagem **Java 11** e as bibliotecas **AndroidX/Jetpack** para compatibilidade e componentes modernos. Em particular, aproveitamos:

- **AppCompat** e **Material Components** para estilos e widgets responsivos.
- **ConstraintLayout** para layouts flexíveis que se adaptam a diferentes resoluções.
- **RecyclerView** para exibição eficiente de listas dinâmicas (medicações, previsões).

3.2 SQLite e DatabaseHelper

Para persistência local, usamos o banco **SQLite** nativo do Android, encapsulado em nossa classe **DatabaseHelper** (que estende **SQLiteOpenHelper**).

- **Tabela Users**: armazena **USERNAME**, **PASSWORD**, **NAME**, **ALLERGY (CSV)** e **IMAGE_URI**. **Tabela Medications**: mapeia cada alergênico a um remédio.

- Métodos principais:
 - `insertData(username, password)` – registra novo usuário.
 - `checkLogin(username,password)` – valida credenciais.
 - `getProfile(username)` – obtém alergias e URI de foto.
 - `getMedicationsForAllergy(allergen)` – retorna medicações pre-definidas.

3.3 Location Services e AlarmManager

Location Usamos `FusedLocationProviderClient` (Google Play Services) para obter latitude/longitude do dispositivo em tempo real, respeitando permissões `ACCESS_FINE_LOCATION`.

AlarmManager Para lembretes de medicação:

- Criamos `PendingIntent` para cada medicação.
- Agendamos `setExactAndAllowWhileIdle()` ou `setInexactRepeating()` conforme a dosagem:
 - Cetirizine 10 mg — 1×/dia às 08:00
 - Fexofenadine 120 mg — 1×/dia às 20:00
 - Diphenhydramine 25 mg — 4×/dia às 08:00, 12:00, 16:00, 20:00

Capítulo 4

Implementação e Testes

4.1 Autenticação e Registro

4.1.1 MainActivity (Login)

A `MainActivity` apresenta campos de usuário e senha. Ao clicar em “Login”:

1. Chama `DatabaseHelper.checkLogin(username, password)`.
2. Se retornado não nulo, grava sessão em `SharedPreferences` e abre `HomeActivity`.
3. Caso contrário, exibe `Toast` de credenciais inválidas.

4.1.2 RegisterActivity

A `RegisterActivity` permite cadastro usando `DatabaseHelper.insertData()`. Valida:

- Campos não vazios.
- Usuário não duplicado com `checkUsername()`.
- Exibe mensagens de sucesso ou erro via `Toast`.

4.2 Perfil de Usuário

4.2.1 ProfileActivity

Exibe foto e alergias atuais do usuário:

- Carrega CSV de alergias e URI da foto via `getProfile(username)`.
- Preenche `ImageView` e texto de alergias.

4.2.2 Seleção de Foto e Alergias

- `Intent.ACTION_PICK` para foto da galeria, retorna `Uri`.
- Diálogo `AlertDialog.Builder.setMultiChoiceItems(...)` para múltipla seleção de alergias.
- Concatena seleções em string “Pollen, Mold, ...” e salva com `updateProfile()`.

4.3 Rastreador de Medicação

`MedicationTrackerActivity` consulta alérgenos do CSV do perfil e, para cada um, executa:

```
Cursor c = db.getMedicationsForAllergy(allergen);
```

Popula um `RecyclerView` com pares (medName — dosage).

4.4 Lembretes Diários

`DailyRemindersActivity` lê a mesma consulta de medicações, mas:

- Para cada par (med — dosage), gera horário(s) recomendados:
 - Once daily* → 08:00
 - Every 4–6h as needed* → 08:00,12:00,16:00,20:00
 - Auto-injector* → alerta único via botão
- Usa `AlarmManager.setExactAndAllowWhileIdle()` para cada hora.
- Exibe lista “Cetirizine — 10 mg @ 08:00” no `ListView`.

4.5 Integração com Open-Meteo

Na `WeatherActivity`:

1. Obtém localização com `FusedLocationProviderClient.getLastLocation()`.
2. Chama `MeteoApi.getForecast(lat,lon,fields,timezone)`.
3. Mapeia `MeteoResponse.daily` para lista de `DailyForecast`:

*`f.date = resp.daily.time[i]`**`f.tempMax = resp.daily.temperature_2m_max[i]`**`f.tempMin`*

4. Calcula `f.severity` (0–10) com regras de vento, umidade, UV, chuva e temperatura.
5. Exibe em `RecyclerView` 7 dias com “2025-06-25: 27°/13°, wind 38.2 m/s, risk 9/10”.

Capítulo 5

Demonstração e Validação

5.1 Cenários de Uso

A aplicação desenvolvida foi testada e validada através de múltiplos cenários de uso práticos, incluindo:

- **Registo e Login:** o utilizador consegue criar uma conta e autenticar-se com sucesso, com validações apropriadas para campos vazios e credenciais incorretas.
- **Configuração do Perfil:** permite ao utilizador carregar uma foto pessoal e seleccionar alergias específicas através de um menu intuitivo e claro.
- **Gestão de Medicação:** após definir as alergias, a aplicação fornece recomendações automáticas de medicamentos com horários de administração claramente definidos e personalizados.
- **Consulta Meteorológica:** permite ao utilizador aceder a previsões meteorológicas detalhadas que incluem informações relevantes para o risco de alergias associadas às condições climatéricas.
- **Contactos de Emergência:** o utilizador pode configurar contactos para emergências e acioná-los facilmente através de um botão específico na interface principal.

5.2 Resultados de Testes Funcionais

Os testes funcionais realizados confirmaram o funcionamento correto das principais funcionalidades, nomeadamente:

- **Autenticação:** todos os cenários de registo e login passaram com sucesso, garantindo robustez e segurança dos dados do utilizador.
- **Base de Dados:** a integração com SQLite e as operações via DatabaseHelper mostraram-se estáveis e eficientes, sem falhas de integridade ou persistência de dados.
- **Alarmes e Notificações:** os lembretes diários configurados através do AlarmManager funcionaram corretamente, respeitando as diferentes frequências indicadas para cada medicamento recomendado.
- **Integração com Open-Meteo:** os pedidos REST para a API externa mostraram-se fiáveis, devolvendo dados consistentes e relevantes.

5.3 Testes de Usabilidade

Realizaram-se testes de usabilidade com um grupo de pessoas de idades variadas, que destacaram os seguintes pontos fortes:

- Interface gráfica intuitiva e apelativa.
- Facilidade em configurar o perfil e gerir os dados pessoais.
- Eficácia dos lembretes automáticos para medicação.

As sugestões de melhoria incluem:

- Melhorar o contraste visual em certas páginas, particularmente no calendário.
- Introduzir funcionalidades adicionais, como alertas mais específicos para diferentes níveis de risco de alergia.

Capítulo 6

Conclusões e Trabalhos Futuros

6.1 Principais Conclusões

Ao concluir este projeto, destacam-se os seguintes resultados obtidos:

- Desenvolvimento bem-sucedido de uma aplicação Android completa, que integra funcionalidades essenciais como autenticação, gestão de perfil, rastreamento de medicação e previsão meteorológica personalizada.
- Validação prática da solução, comprovando a viabilidade técnica e a eficácia do conceito.
- Obtenção de feedback positivo por parte dos utilizadores de teste, reforçando a utilidade e usabilidade da aplicação em contexto real.
- A demarcação de funções que tiveram que ser removidas, ou adiadas, deixaram tristeza, mas confiança para um trabalho autónomo futuro.

6.2 Trabalhos Futuros

Com vista à evolução contínua da aplicação, identificam-se várias oportunidades para trabalho futuro:

- **Funcionalidades Avançadas de IA:** implementação de uma funcionalidade premium que utiliza inteligência artificial para análise preditiva de sintomas e recomendações personalizadas.

- **Melhorias na Interface Gráfica:** otimizar ainda mais a interface, proporcionando melhor experiência visual e interação mais fluida com o utilizador.
- **Integração com Serviços Externos Adicionais:** considerar o uso de APIs complementares para obter informação mais detalhada sobre alergias e fatores ambientais.
- **Testes de Longa Duração e Escalabilidade:** conduzir testes mais extensivos que simulem utilização prolongada para assegurar estabilidade e desempenho em larga escala.

Referências Bibliográficas

- [1] Open-Meteo API – <https://open-meteo.com/>
- [2] Square, Inc. Retrofit – <https://square.github.io/retrofit/>
- [3] Android Developers – <https://developer.android.com/>
- [4] Github do Trabalho - <https://github.com/RMaestro123/AllergyTrackerProject>