

---

## **Proposta de Projeto**

### **Epicurius - Aplicação Móvel de Receitas**

49412 - Vasco Costa; a49412@alunos.isel.pt  
49465 - Carolina Tavares; a49465@alunos.isel.pt

**Orientador:** Artur Ferreira; artur.ferreira@isel.pt

---

10 de Março de 2025

## **1. Introdução**

A gastronomia é essencial ao dia a dia e a tecnologia tem revolucionado a forma como exploramos, preparamos e compartilhamos receitas. O desenvolvimento da aplicação Epicurius visa tornar a experiência culinária mais acessível, dinâmica e personalizada.

O projeto permitirá aos utilizadores encontrarem receitas de forma simples através da captura dos ingredientes que tiver disponível. A aplicação identifica os ingredientes presentes na imagem e sugere possíveis pratos, otimizando o aproveitamento dos alimentos e reduzindo desperdícios, enquanto potencia a experiência culinária.

A aplicação conta ainda com um *feed* interativo onde é possível interagir com receitas publicadas por outros utilizadores e partilhar receitas. Possui também uma interface intuitiva com diversas funcionalidades, como sugestão de menu diário, planeamento de refeições e gestão do frigorífico com notificações inteligentes de alerta para produtos em fim de validade.

Para garantir um vasto banco de dados de receitas e uma gestão eficiente dos ingredientes, a aplicação utilizará [Spoonacular API](#) para fornecimento de informações nutricionais e sugestões de pratos, e a [Google Vision API](#) para identificação de ingredientes a partir de imagens.

## **2. Descrição do Sistema**

A Figura 1 apresenta a arquitetura do sistema a ser implementado, constituída pelos componentes: **Aplicação Móvel**, **Google Cloud Platform** composta pelo **Backend**, **Cloud Storage**, **Cloud Function** e a **Vision API**. Possui ainda duas **Bases de Dados** e a **Spoonacular API**.

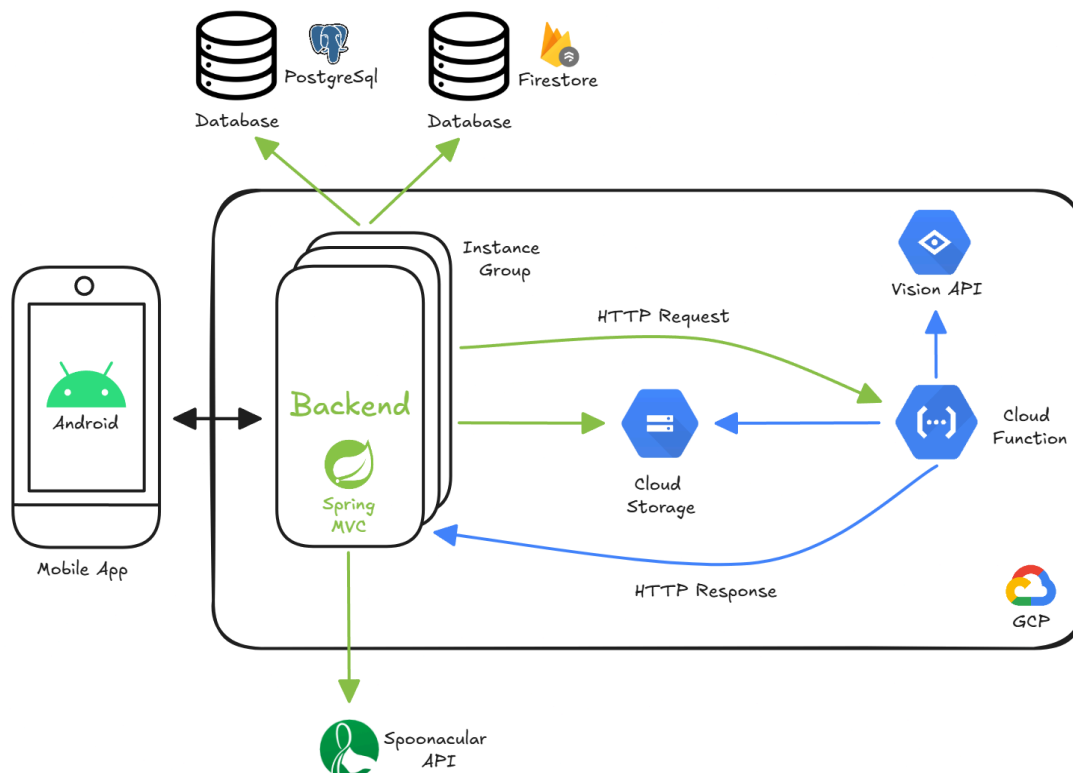


Figura 1 - Arquitetura do Sistema

## 2.1 Frontend

Através de uma aplicação móvel, desenvolvida utilizando **Jetpack Compose**, o utilizador consegue interagir com todas as funcionalidades do sistema.

## 2.2 Backend

Usando o **Spring MVC**, é implementada uma API REST que permitirá ao utilizador interagir com os seus dados pessoais e de receitas. Quando uma imagem é submetida, é a API que a guarda no **Cloud Storage** e realiza um pedido HTTP para a **Cloud Function**, de modo a ativá-la, para que esta possa processar caso se trate de uma imagem de ingredientes.

## 2.3 GCP

### 2.3.1 Cloud Storage

O Cloud Storage é utilizado para armazenamento de imagens relacionadas com determinada receita. As imagens de ingredientes que o utilizador pretende utilizar para pesquisa e posterior análise da **Vision API**, são armazenadas de forma temporária.

### 2.3.2 Cloud Function

A Cloud Function possui a funcionalidade de recolha de imagens do **Cloud Storage** que contenham apenas ingredientes e consequente submissão na **Vision API**. É ativada com um [trigger http](#) através de um pedido HTTP feito pelo **Backend**.

### 2.3.3 Vision API

A Vision API da Google para visão computacional será usada, neste contexto, para a deteção de ingredientes numa imagem.

## 2.4 Base de Dados

O sistema utiliza duas bases de dados distintas para otimizar o armazenamento e a pesquisa de informações. O **PostgreSQL** será responsável por armazenar os dados dos utilizadores e das receitas, limitando-se a informações essenciais e de menor tamanho, como o nome e os ingredientes, garantindo assim consultas mais eficientes. Para dados mais extensos, utilizar-se-á o **Firestore**, ideal para armazenar conteúdos volumosos, como por exemplo, os procedimentos das receitas.

Esta abordagem foi adotada para preservar o desempenho das pesquisas no **PostgreSQL**, ao mesmo tempo que permite armazenar grandes volumes de informação numa base de dados mais adequada a esse propósito, neste caso o **Firestore**.

## 3. Requisitos

### 3.1. Requisitos Obrigatórios

- *Feed* interativo com receitas publicadas pelos utilizadores seguidos, ordenadas segundo o tempo de publicação;
- Perfil do utilizador com informação das suas receitas publicadas, dos seus seguidores, utilizadores seguidos e organização personalizada de receitas publicadas;
- Perfil da receita que contém toda a informação relativa a uma dada receita, nomeadamente, autor, tipo de cozinha, tipo de refeição, ingredientes e preparação.
- A possibilidade de editar uma receita pertence apenas ao utilizador autor da mesma, já a opção de acompanhamento passo a passo fica disponível para toda a comunidade;
  - Na hipótese de o utilizador não possuir todos os ingredientes necessários à receita pretendida a realizar, existe a opção de sugestão de ingredientes substitutos, ou a criação de uma lista de compras com os mesmos;
- Publicação de uma receita no perfil do utilizador;
- Pesquisa por receitas através de:
  - Nome;
  - Formulário;
  - Foto captada pelo telemóvel ou carregada pelo utilizador que é analisada pela **Google Vision API** para detecção de ingredientes;
- Sugestão diária, onde é proposto um menu completo ao utilizador, que inclui, pequeno-almoço, almoço, lanche, jantar e sobremesa;
- Menu de utilizador que fornece acesso a receitas guardadas, denominadas de favoritos, possibilidade de alteração de informações pessoais, tais como, nome, password, email e privacidade do perfil. Possui também uma secção de planificação de refeições para determinados dias da semana, por exemplo, definir um limite calórico para o dia;
- Funcionalidade de frigorífico, com a possibilidade de colocar produtos, com as respetivas quantidades e datas de validade. Em caso de determinado produto se aproximar da sua data de expiração, o utilizador é notificado.

### 3.2. Requisitos Opcionais

- Apresentar sugestões de receitas por etiquetas seguidas pelos utilizadores;
- Recomendação de vinhos para acompanhar refeições principais;

- Adição de vídeos para acompanhamento passo a passo de uma receita, ação apenas realizada pelo proprietário da receita, guardados no **Google Cloud Storage**;
- Mini-cursos com diferentes dificuldades e tipos de cozinha ou demonstração de procedimentos básicos, de forma a incitar aprendizagem e exploração da cozinha, guardados no **Google Cloud Storage**;
- Em caso de notificação de expiração de um produto, são apresentadas ao utilizador sugestões de receitas para o mesmo, de forma a evitar o desperdício;
- Opção de download de receitas, possibilitando o acesso às mesmas na aplicação em modo offline.

#### 4. Riscos

O desenvolvimento deste projeto envolve certos riscos, tais como, a detecção parcialmente incorreta dos ingredientes a identificar por parte da **Vision API** e o limite de pedidos a APIs externas devido à dependência das mesmas. Outra preocupação são os custos da infraestrutura poderão ser aumentados caso seja atingido o valor total do cupão em uso da **Google Cloud Platform**.

De forma a diminuir os riscos associados, os erros na deteção de alimentos, serão combatidos adicionando um meio de verificação na aplicação móvel para que o utilizador possa adicionar ou remover ingredientes detectados (ex.: *pop up* de confirmação). Para prevenir o excesso de pedidos, será usada uma réplica local da API reduzindo a realização de pedidos. Mantendo o mesmo objetivo, inicialmente, o **Backend** é mantido em execução localmente reduzindo os custos de infraestrutura do **GCP**.

#### 5. Planeamento

O diagrama da **Figura 2** mostra o planeamento para a realização do projeto. Ao longo do desenvolvimento do mesmo, as tarefas vão sendo distribuídas pelos elementos do grupo.

#### 6. Referências

- Spoonacular API <https://spoonacular.com/food-api/docs>
- Vision API <https://cloud.google.com/vision/docs>
- Jetpack Compose <https://developer.android.com/develop/ui/compose/documentation>
- Spring MVC <https://docs.spring.io/spring-framework/reference/web/webmvc.html>
- Cloud Storage <https://cloud.google.com/storage/docs>
- Cloud Functions <https://cloud.google.com/functions/docs>
- PostgreSQL <https://www.postgresql.org/>
- Firestore <https://firebase.google.com/docs/firestore>

## DIAGRAMA DE GANTT



Figura 2 - Diagrama de Gantt