

Technical Project Report - Android Module

# Clean House

Subject: Computação Móvel

Date: Aveiro, 25/01/2022

Students: 89077: Renato Valente  
89162: Jacinto Lufilakio

Project abstract: Aplicação para contratar pessoas interessadas em prestar serviços de limpeza de casas. Pode ser utilizada pelos consumidores que pretendem ter as suas casas limpas e pelos prestadores de serviços que estão interessados em fazer o serviço de limpeza de casa.

Conteúdo do relatório:

## [1 Conceito do aplicativo](#)

## [2 Soluções implementadas](#)

[Visão geral da arquitetura](#)

[Interações implementadas](#)

[Aplicação do consumidor \(CleanApp\)](#)

[Aplicação do prestador de serviços \(HouseClean\)](#)

[Limitações do projeto](#)

[Novos recursos e alterações após a apresentação do projeto](#)

## [3 Conclusões e recursos de apoio](#)

[Lições aprendidas](#)

[Recursos do projeto](#)

[Contributions](#)

[Reference materials](#)

## 1 Conceito do aplicativo

O principal objetivo da aplicação House Clean é contratar pessoas interessadas em limpar a casa de um determinado cliente. Os utilizadores interessados nesta app são todas as pessoas que precisarem de contratar alguém para limpar a sua casa. Esta aplicação foi contemplada com outra aplicação House Cleaner, cujo principal interesse é das pessoas da limpeza, onde estas podem se candidatar para a limpeza das moradias nos arredores.

A aplicação, vem assim, ajudar quem tem necessidade de limpar a sua casa mas não dispõe de tempo ou até mesmo vontade e quem está a precisar de ganhar dinheiro a limpar casas e encontra dificuldade em arranjar clientes ou até mesmo acordar os termos dos serviços.

## 2 Soluções implementadas

### Visão geral da arquitetura

Quando a aplicação é iniciada é nos apresentado a atividade de login. Esta atividade está conectada ao Firebase e portanto é nesta mesma que, tanto podemos efetuar login bem como nos registarmos na plataforma. Foi implementado persistência local através da API SharedPreferences, permitindo assim guardar as credenciais do utilizador, de maneira a fazer login automático quando se inicia a aplicação ou fazer o auto preenchimento dos campos de login caso o utilizador efetue log out.

Como componente principal da nossa aplicação, tanto a dos clientes como a dos prestadores de serviços, temos uma actividade principal com uma bottom navigation bar e um fragment. Este fragment consoante o item selecionado da bottom navigation bar apresenta diferentes layouts conectados à base de dados, cada um com a sua funcionalidade.

Layouts dos clientes:

Map→ Neste layout foi implementado um mapa através da API Maps da Google. Este mapa permite, ao cliente, saber a localização dos prestadores de serviços quando estes se dirigem a sua casa.

Inbox→ Neste layout estão disponíveis, numa recycler view, todas as transações efetuadas pelo cliente, estejam elas em qualquer estado (à espera do prestador de serviço, terminadas, canceladas, etc.). Esta recycler view é populada por um recycler view adapter que suporta cliques de longa duração que permite ao utilizador cancelar a transação.

Houses→ Neste layout estão disponíveis numa recycler view todas as casas do cliente, populada por um recycler view adapter que também suporta cliques de longa duração. Estes cliques permitem anunciar a casa para ser limpa, sendo então disponibilizada a transação associada à mesma no inbox layout. Dispõe também de um floating button que nos permite adicionar casas. Clicando neste botão é iniciada outra atividade que permite aceder à câmara ou à galeria para adicionar uma foto da casa. Permite também adicionar a localização e o endereço da casa através da API Maps da Google.

Perfil→ Neste layout está disponível a informação básica do utilizador (nome e email) bem como uma fotografia do mesmo retirada da base de dados e o botão de log

Out. A foto pode ser alterada, clicando na mesma.

Layouts dos prestadores de serviço:

Map→ Neste Layout está disponível um mapa que após a candidatura à limpeza de uma casa ser aceite, este mostra o percurso até à mesma

Inbox→ Este layout é semelhante ao dos clientes, exceto que o clique de longa duração permite informar o cliente que o prestador de serviços já se encontra à porta de casa

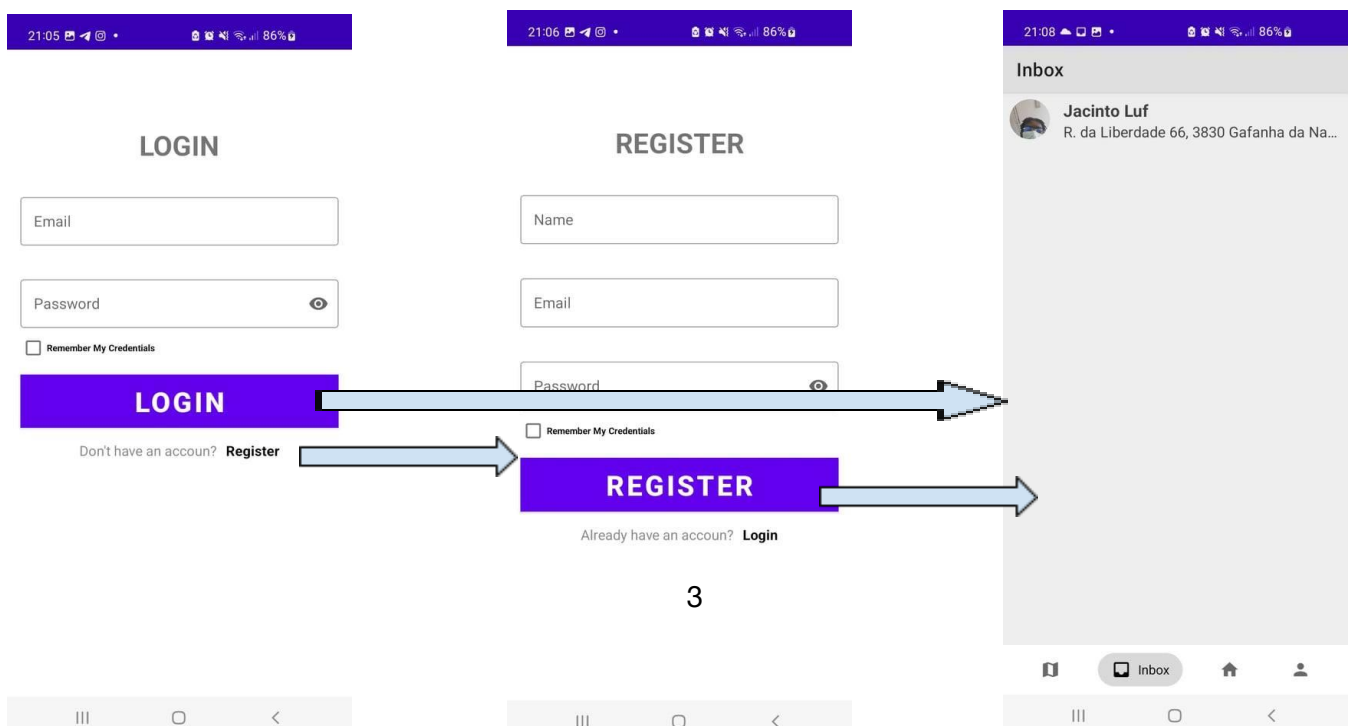
Houses→ Neste layout é mostrado todas as casas dentro de uma área, tendo a localização atual como centro, definida pelo prestador de serviços. O clique de longa duração permite ao prestador de serviços se candidatar à limpeza da casa.

Perfil→ Este layout é semelhante ao do cliente, com a única diferença que no caso dos prestadores de serviços está disponível uma seek bar que permite definir o raio da área em que o mesmo pode limpar.

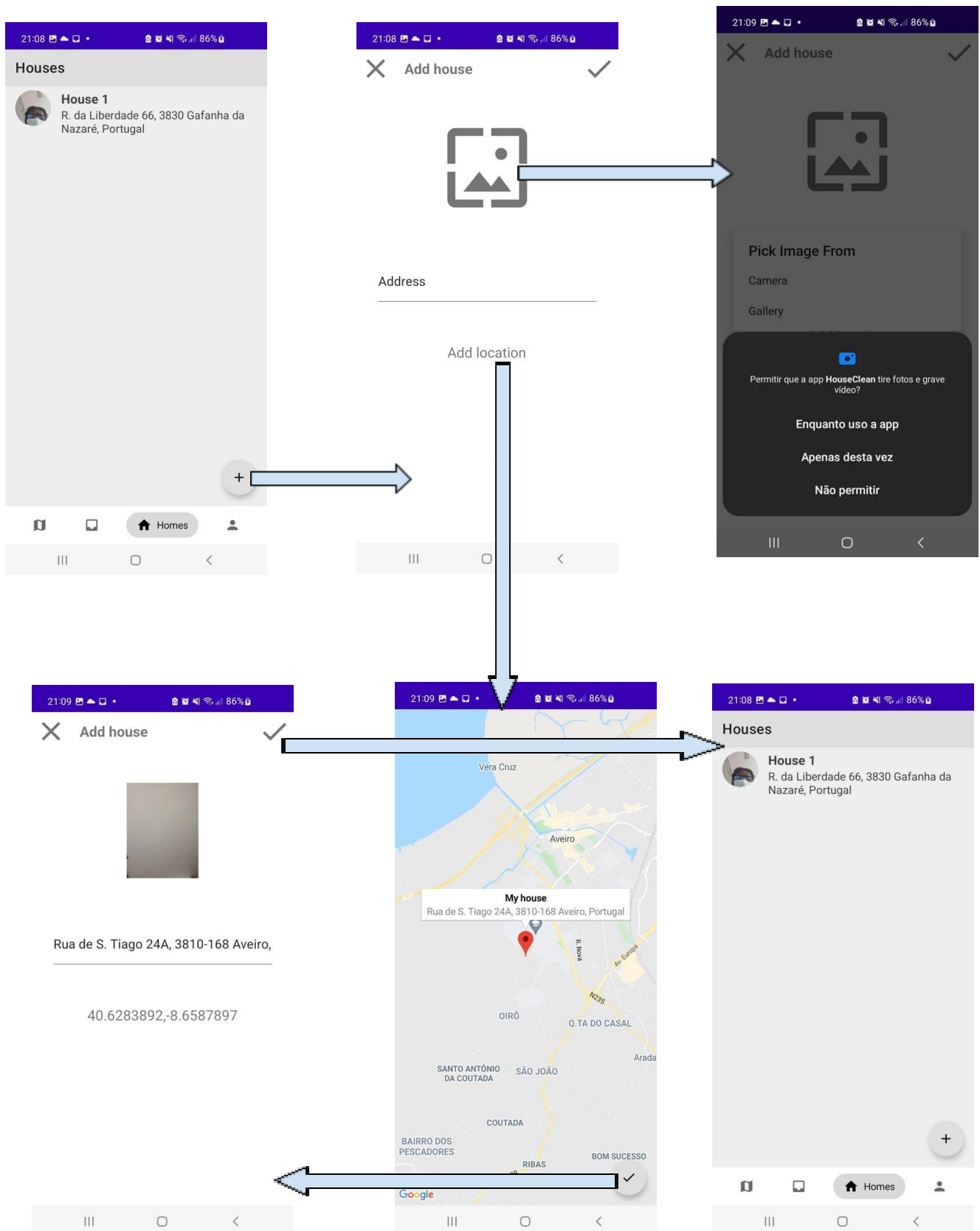
A atividade principal suporta também as alterações feitas na base de dados, sendo responsável por notificar o cliente quando algum prestador de serviço se candidata à limpeza da casa, bem como quando o mesmo chega, gerando um código QR que simula o desbloqueio da porta da casa.

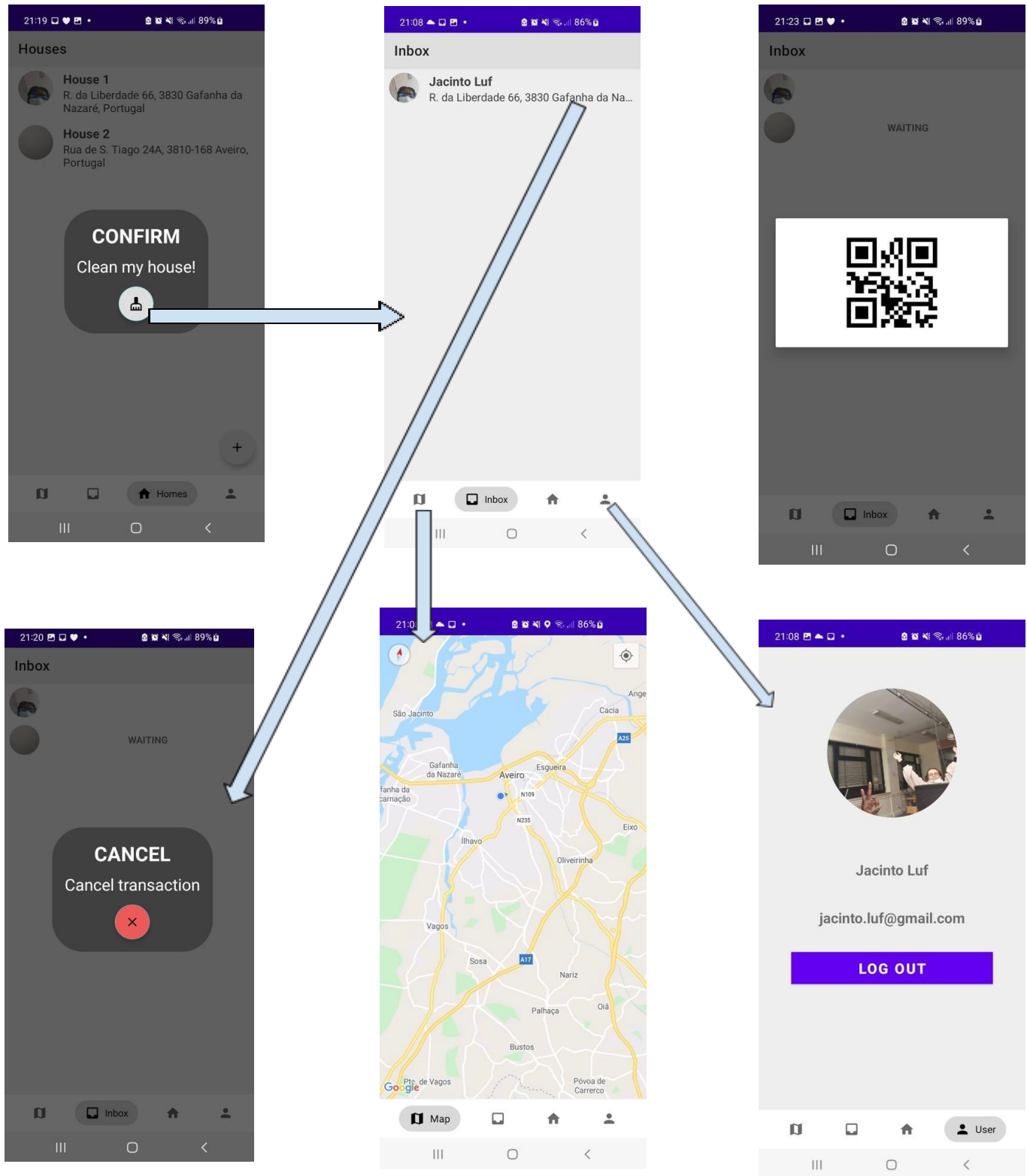
Nesta aplicação estão presentes várias dependências, que nos ajudaram no desenvolvimento da mesma. Foram usadas dependências da Google, tais como, Maps que nos permitiram gerar os mapas usados na aplicação. Temos também dependências do Firebase bem como o Auth que permite a autenticação do utilizador, a Database que permite ter conexão com uma base de dados em tempo-real e a Storage que permite armazenar ficheiros, nomeadamente as fotos dos utilizadores e das casas. Também usamos a dependência Chip-navigation-bar para os efeitos na bottom navigation bar e o Glide que usamos para popular as Image Views e as Circle Image Views. Por último usamos as dependências Coroutines, Lifecycle Viewmodel e Lifecycle Runtime, que permite que o layout espere que a busca dos dados à base de dados seja concluída para poder popular o mesmo.

### Interações implementadas

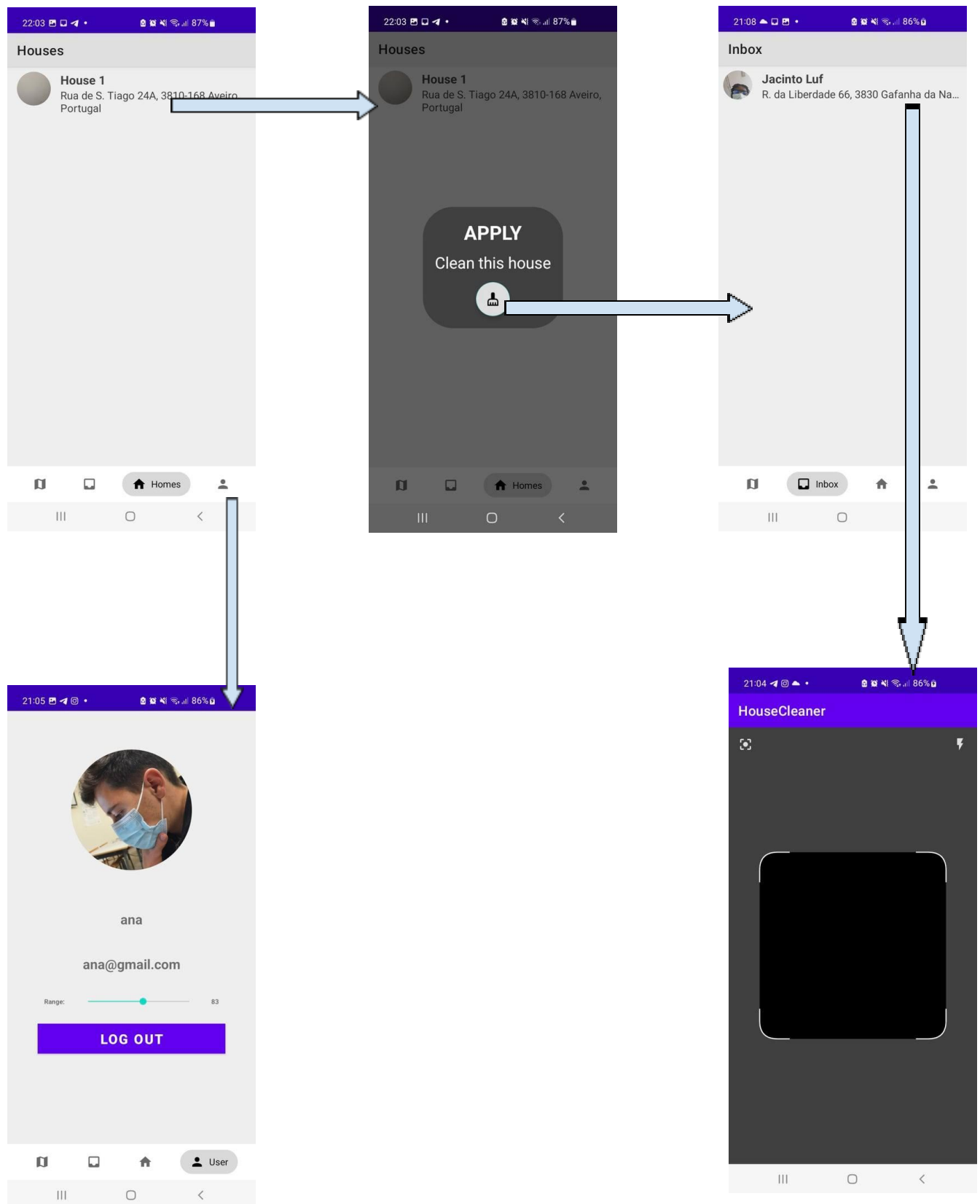


## Aplicação do consumidor (CleanApp)





## Aplicação do prestador de serviços (HouseClean)



## Limitações do projeto

Uma das funções não implementadas foi o mapa com o percurso entre o prestador de serviços e a casa. O leitor de códigos QR da parte do prestador de serviços não funciona como o esperado, o que torna impossível ler os códigos gerados pelos clientes. Uma das limitações está no facto de não ser possível ao cliente eliminar casas que tenha adicionado.

## Novos recursos e alterações após a apresentação do projeto

Num trabalho futuro, além de implementarmos o que não foi implementado, seria ótimo se pudessemos passar a gestão da troca de mensagens entre cliente e prestador de serviços para um servidor, por forma a garantir a integridade dos dados. Deveria também ser implementado um chat entre cliente e prestador de serviços para facilitar a comunicação entre ambos. Visto que se trata de uma aplicação cujo objetivo é gerar renda poderíamos também introduzir um método de pagamento na aplicação.

## 3 Conclusões e recursos de apoio

### Lições aprendidas

Achamos fácil e prático o uso de dialogs e coroutines, que nos facilitou bastante trabalho. Consistência e comunicação entre o cliente e o prestador de serviços foi o que se tornou mais difícil.

### Recursos do projeto

Recurso:	Disponível a partir de:
Repositório do código:	<a href="https://github.com/renatovalente5/CleanHouseApp">https://github.com/renatovalente5/CleanHouseApp</a>
APK pronto para implantar:	CleanApp.apk: <a href="https://drive.google.com/file/d/1n3SRpWuaqbq83zq0vRpBEGpx3y5RN4C/view?usp=sharing">https://drive.google.com/file/d/1n3SRpWuaqbq83zq0vRpBEGpx3y5RN4C/view?usp=sharing</a> , HouseCleaner.apk: <a href="https://drive.google.com/file/d/1sleH24DrWO5GSCmzqeMndT_92BDCz-vM/view?usp=sharing">https://drive.google.com/file/d/1sleH24DrWO5GSCmzqeMndT_92BDCz-vM/view?usp=sharing</a>

### Contribuições

O projeto foi dividido de modo a que ambos os participantes contribuíssem com 50% no trabalho.

### Materiais de referência

[Estrutura de fragments](#)

[Notificações](#)

[Coroutines](#)

[Mapa num fragment](#)

[Ligar as definições de localização](#)

[Navegar entre atividades](#)