

FasTuga Driver

Projeto da Unidade Curricular de Tópicos Avançados de Engenharia de Software

Ricardo dos Santos Franco
Departamento de Engenharia
Informática
Instituto Politécnico de Leiria
Leiria, Portugal
2202314@my.ipleiria.pt

Marco António Perdigão Gaspar
Departamento de Engenharia
Informática
Instituto Politécnico de Leiria
Leiria, Portugal
2201708@my.ipleiria.pt

Professora Catarina Reis
Departamento de Engenharia
Informática
Instituto Politécnico de Leiria
Leiria, Portugal
catarina.reis@my.ipleiria.pt

Bruno Vieira Silva
Departamento de Engenharia
Informática
Instituto Politécnico de Leiria
Leiria, Portugal
2200670@my.ipleiria.pt

Abstract—O objetivo deste projeto era desenvolver uma aplicação fundamentalmente mobile que possibilitasse a entrega de encomendas criadas pelos clientes no website relacionado com o projeto de DAD (FasTuga website). Os utilizadores desta aplicação poderão registar-se, escolher as encomendas a entregar, e através do mapa integrado na aplicação, chegar ao destino indicado pelos clientes de forma a concluir a entrega. De forma a desenvolver esta aplicação foram utilizadas várias metodologias.

I. INTRODUCTION

O projeto FasTuga Driver foi-nos proposto pela professora Catarina Reis na Unidade Curricular de Tópicos Avançados de Engenharia de Software e tem como objetivo ser utilizada por distribuidores de encomendas de *fast food* que trabalhem para a empresa fictícia “FasTuga” onde podem escolher a encomenda que pretendam, e ver o caminho que precisam percorrer para a entregar.

A aplicação proposta foi desenvolvida para mobile (android) e utilizando a linguagem kotlin.

O presente relatório tem como propósito apresentar as funcionalidades da aplicação desenvolvida e as metodologias usadas.

Este documento encontra-se dividido da seguinte forma:

- II - FasTugaDriver
- III - PROCESSO DE DESENVOLVIMENTO DE SOFTWARE
- IV - EVOLUÇÃO
- V - ARQUITETURA DE SOFTWARE
- VI - VERIFICAÇÃO E VALIDAÇÃO
- VII - CONCLUSÃO
- VIII - REFERÊNCIAS

II. FASTUGADRIVER

Esta aplicação tem como principal objetivo a gerência de encomendas, escolha e entrega das mesmas (com base num mapa).

Ao abrir a aplicação pela primeira vez, o utilizador irá ver uma página de Login (Figura 1) onde pode entrar diretamente com a sua conta ou escolher a opção de se registar (Figura 2). Tanto o login como o registo irão enviar o utilizador para o funcionamento principal da aplicação.

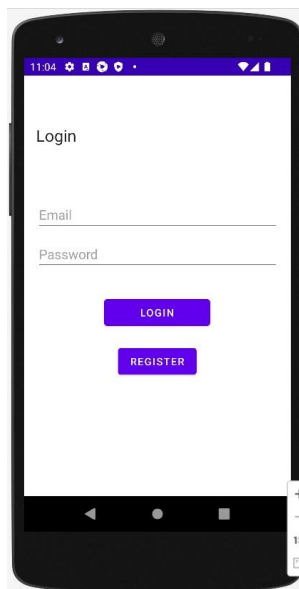


Figura 1 - Login

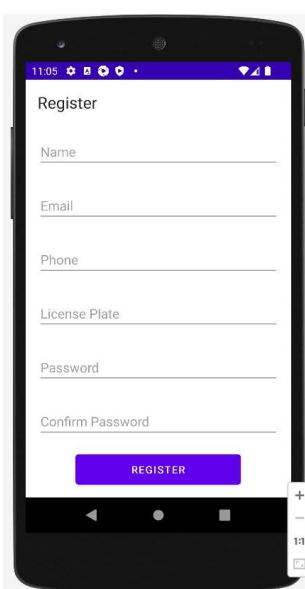


Figura 2 - Register

Se o utilizador não tiver as permissões ativas de localização será redirecionado para a página de autorização onde a pode ativar (Figura 3)

A aplicação está dividida em 3 fragmentos principais, são eles (figura 4):

- Orders
- Map
- Profile

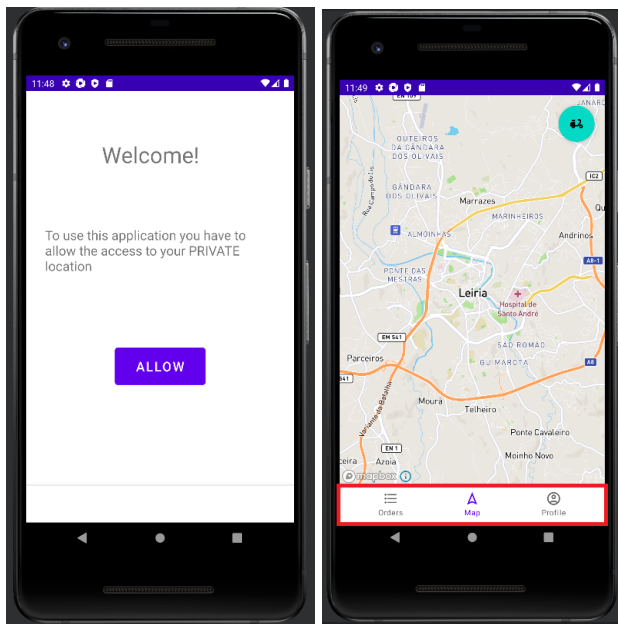


Figura 3 - Authorization Page Figura 4 - Main Page (Map)

Para o utilizador começar a entregar uma encomenda precisa de aceder às “Orders” onde encontra uma lista de encomendas detalhadas e disponíveis para serem entregues por algum “driver”, lista essa que pode ser filtrada por distância (“Nearest orders”, “Farthest orders”), tendo como pré definida a opção “Nearest orders” (figura 5).

Após a escolha da encomenda a querer entregar, será mostrado no fragmento “Map” o caminho desde o restaurante até à localização do cliente que fez a encomenda (Figura 6).

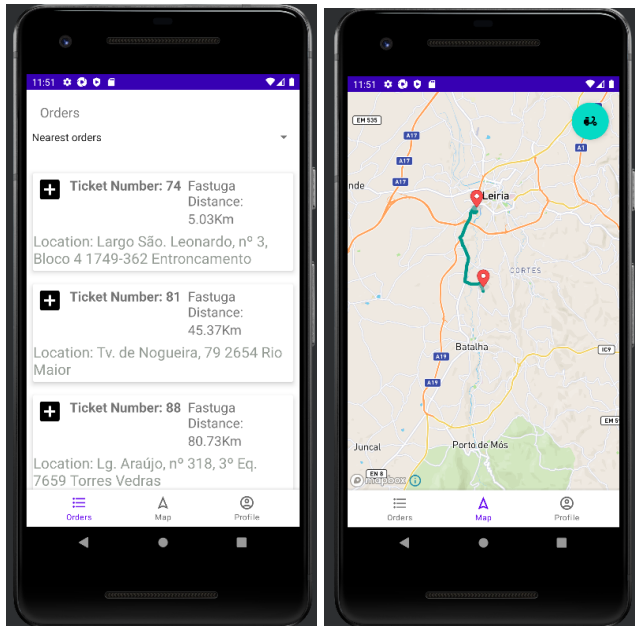


Figura 5 - Lista de “Orders” Figura 6 - Caminho para a encomenda

Para começar a entrega da encomenda, existe um botão no canto superior direito que contém informações para a entrega da encomenda, com o “ticket number”, a “tax fee” e a morada do cliente. Nesta página há também a possibilidade de cancelar a entrega da encomenda (tendo o

“driver” de indicar a razão pela qual quer cancelar) (Figura 7).

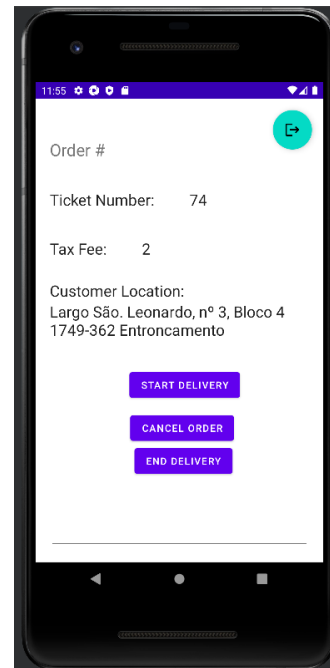


Figura 7 - Detalhes da encomenda seleccionada

Ao clicar em “Start Delivery” o utilizador receberá instruções visuais e textuais dos caminhos que tem de percorrer para chegar ao cliente e é possível observar a sua deslocação no mapa (Figura 8), na página profile o saldo (“balance”) fica com a adição condicional da taxa “tax fee”. Quando chegar ao destino e pressionando “End delivery” o “balance” é atualizado e estará pronto para uma nova entrega.

No caso do utilizador querer ver os seus dados, saldo, histórico de encomendas realizadas, notificações, estatísticas e dar *logout* poderá aceder ao fragmento “Profile” (Figura 9). Nesse fragmento, o utilizador consegue também aceder ao “Edit profile” para poder alterar os seus dados.

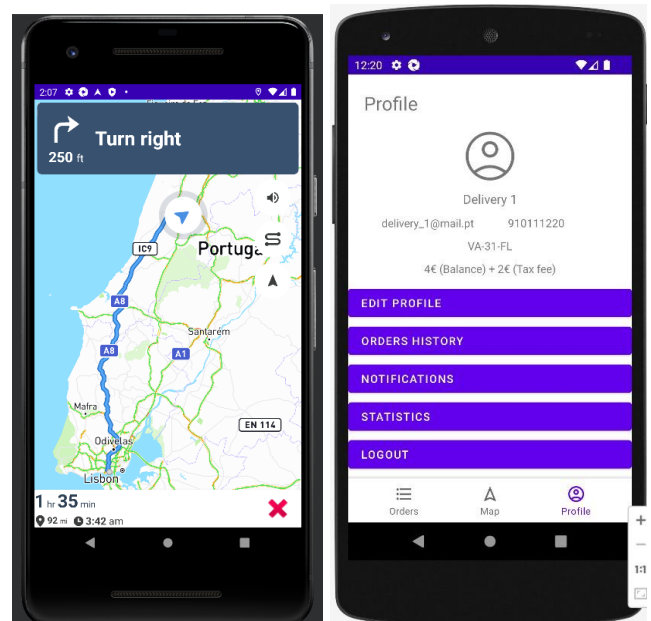


Figura 8 - Starting delivery

Figura 9 - Statistics Page

Na página do perfil, ao clicar em:

- “Orders History”, aparecerá uma lista com as encomendas que foram realizadas pelo utilizador (Figura 10).
- “Notifications”, verá as notificações que recebeu ao ter uma nova “order” no estado “ready”, ou seja, pronta para levantar, e quando outro “driver” cancelou a sua entrega de encomenda(Figura 11).
- “Statistics”, estarão as estatísticas relativas ao utilizador. São elas as encomendas que já entregou, tempo médio que demora para as entregar, tempo total que já esteve a entregar e para quantos clientes diferentes (Figura 12).

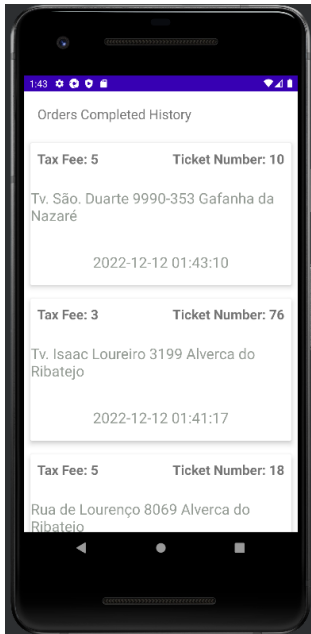


Figura 10 - Orders History

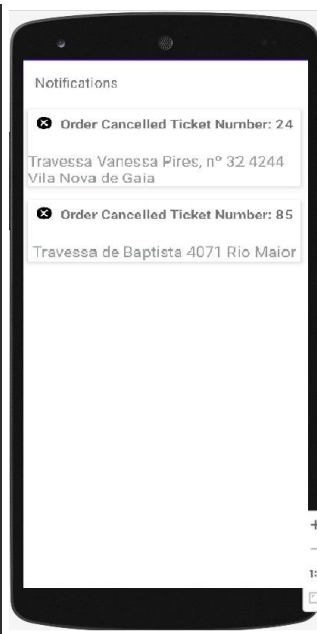


Figura 11 - Notifications Page

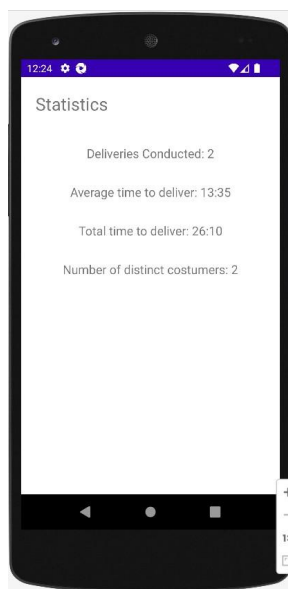


Figura 12 - Statistics Page

III. PROCESSO DE DESENVOLVIMENTO DE SOFTWARE

No desenvolvimento do projeto FasTugaDriver foi usada a framework Scrum, Quadros Kanban e uma adaptação da metodologia Extreme Programming.

Durante o processo realizaram-se as Sprint Planning no início de cada semana onde ficava definido o Sprint Backlog, também se concretizaram as Sprint Retrospective onde se concluía a Sprint e definia novas estratégias para aumentar o desempenho da equipa.

Todas as Sprint tiveram a duração de sete dias (sprints semanais).

No início de cada semana o quadro kanban recebia novos cartões que representavam as User Stories a realizar durante a nova sprint.

Em relação à metodologia Extreme Programming, foram implementadas as práticas Informative Workspace e Weekly Cycle.

A prática Informative Workspace foi aplicada nos quadros brancos da sala L.E.S, nos mockups feitos em figma anexados a cada User Story no Jira Software. A prática de ciclos semanais foi implementada quando reunimos no início de cada semana, quando o cliente escolheu que User Stories implementar em cada semana e planeamos o trabalho de cada semana em cada semana.

IV. EVOLUÇÃO

Sprint 1

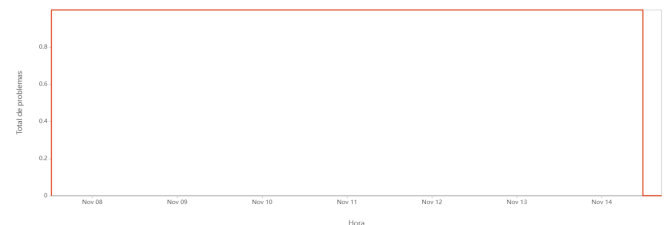


Figura 13 - Burndown chart Sprint 1

Sprint 2

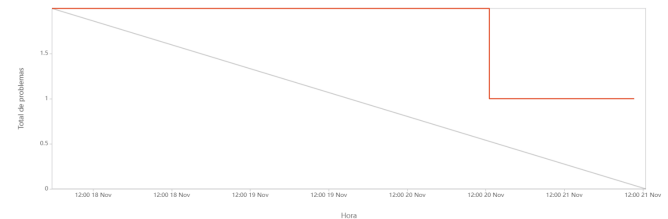


Figura 14 - Burndown chart Sprint 2

Sprint 3

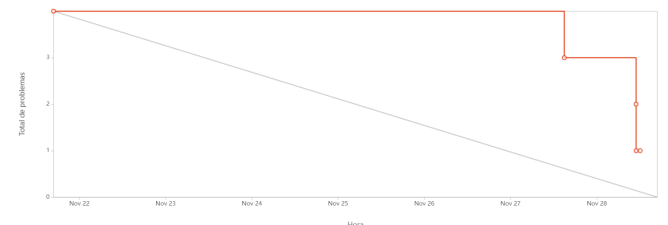


Figura 15 - Burndown chart Sprint 3

Sprint 4

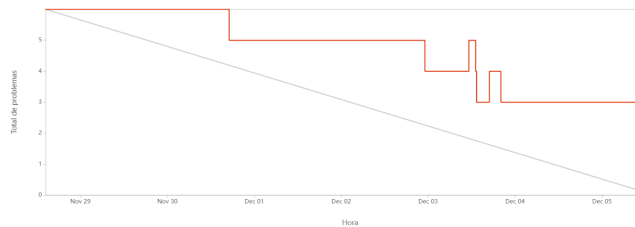


Figura 16 - Burndown chart Sprint 4

Sprint 5

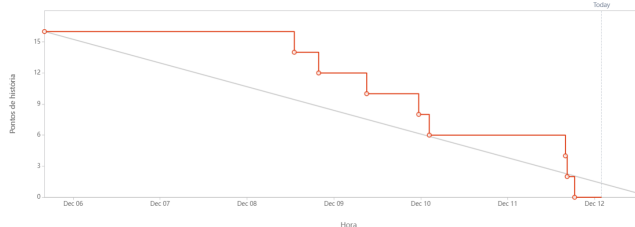


Figura 16 - Burndown chart Sprint 5

Os gráficos apresentados anteriormente mostram a evolução do projeto e o processo de desenvolvimento. Estes gráficos mostram a variação dos User Stories a fazer ao longo de cada sprint e ao longo do desenvolvimento.

Conseguimos analisar que nas Sprint dois, três e quatro houve passagem de User Stories para a Sprint Seguinte.

Na Sprint 4 o número de problemas aumenta em dois instantes temporais, o que indica que algo foi declarado como pronto ou terminado, mas voltou a estar em progresso.

Na Sprint 5 a linha vermelha passa para baixo da linha cinzenta, porque o projeto foi totalmente terminado antes do tempo restante da Sprint.

Relação entre o business value atribuído e o business value entregue.

Sprint	BV a entregar	BV entregue	Story Points
Sprint 1	100	100	2
Sprint 2	50	50	2
Sprint 3	225	225	8
Sprint 4	225	225	12
Sprint 5	1625	1625	16
Total	2000	2000	32

V. SYSTEM ARCHITECTURE

O sistema como um todo é composto por vários componentes. Vendo o sistema com uma granularidade reduzida temos 3 sistemas a comunicar. A aplicação android

FasTugaDriver, a API FasTuga e as entidades externas em que confiamos.

Fazendo zoom no contentor da aplicação FasTugaDriver, temos uma aplicação que interage com a API FasTuga utilizando uma API interna.

Fazendo zoom no contentor da API FasTuga temos uma aplicação que trata da lógica de negócio e do acesso a dados.

As entidades externas são usadas para servir problemas específicos. Usamos a entidade externa geocod para obter as coordenadas de uma morada, navigations para indicar o caminho e a entidade directions para mostrar o caminho entre 2 pontos.

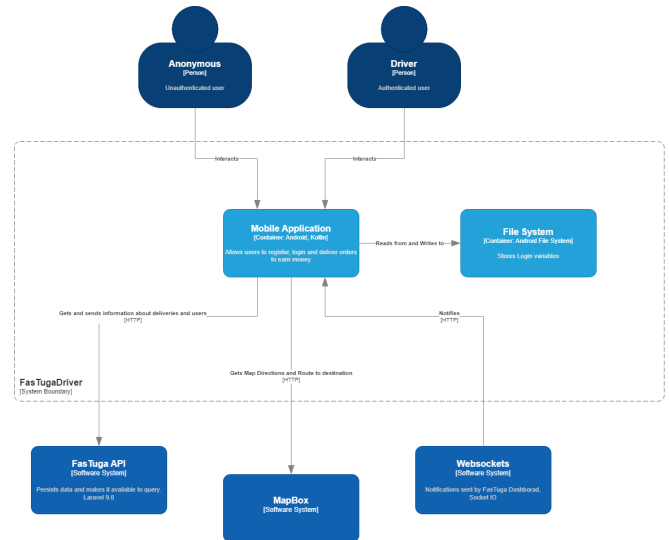


Figura 17 - System Architecture C4 model level 2

VI. VERIFICAÇÃO E VALIDAÇÃO

Na aplicação FasTugaDriver, para testar o desempenho e correto funcionamento das User Stories e Tasks pedidas pelo cliente, foram criados testes, com o objetivo de descobrir situações de defeito da aplicação e verificar se o sistema funciona da forma esperada.

No decorrer de cada teste diferente nas User Stories, existiram testes de forma a verificar o correto funcionamento das features (happy path).

Os testes foram realizados através do software de *automation testing* Katalon. Na realização dos testes, usámos maioritariamente o método *Black-Box*, que examina as funcionalidades realizadas.

As User Stories apenas eram consideradas completas quando os resultados da execução de todos os testes eram os esperados.

VII. CONCLUSÃO

Concluindo, as metodologias e frameworks usadas foram cumpridas.

Relembrando os principais objetivos de selecionar encomendas, ver o caminho até ao local de entrega, iniciar e terminar o processo de entrega, consultar estatísticas sobre o trabalho que o utilizador realizou, concluo que as funcionalidades foram satisfeitas.

As vantagens da aplicação FasTugaDriver para os utilizadores são:

- Facilidade no registo e acesso à aplicação;
- Fácil visualização das encomendas disponíveis para entregar uma vez que podem ser ordenadas por distância ao restaurante FasTugaDriver;
- Fácil escolha da encomenda a entregar;
- O distribuidor recebe notificação quando uma encomenda está pronta para levantar, o que evita atrasos ou descontentamento dos clientes do restaurante.
- Conseguir dar início ao processo de distribuição ao clicar num botão presente na janela onde aparece a morada onde entregar, o valor a ganhar com essa entrega e outras funcionalidade descritas a seguir.
- O utilizador consegue rastrear o caminho até ao destino, clicando também no botão “Start Delivery”
- O utilizador consulta diretamente na aplicação o valor a receber nos detalhes do seu perfil.
- A janela principal é simples para que o utilizador tenha todas as funcionalidades à mão não deixando

de ver o mapa com o caminho a seguir que é o principal.

Com este projeto aprofundamos o conhecimento e experiência em desenvolvimento agile, tivemos oportunidade de desenvolver para Mobile usando a linguagem de programação Kotlin, o que nos desafiou na descoberta de diferentes formas de resolver problemas.

Appendix

O trabalho proposto foi realizado por todos os elementos da equipa, em cada etapa do trabalho cada elemento do grupo esteve presente, no caso, no planeamento de tarefas, nos testes e no projeto em si.

Na parte de servidor utilizámos o docker (que de momento já está no servidor da máquina virtual), e a API feita em laravel