

FasTuga Driver

Projeto de Tópicos Avançados de Engenharia de Software

Martim Silva
Curso de Engenharia Informática
Sistemas de Informação
Politécnico de Leiria
Leiria, Portugal
2200681@my.ipleiria.pt

Francisco Amaral
Curso de Engenharia Informática
Sistemas de Informação
Politécnico de Leiria
Leiria, Portugal
2170831@my.ipleiria.pt

Diogo Chen
Curso de Engenharia Informática
Sistemas de Informação
Politécnico de Leiria
Leiria, Portugal
2212590@my.ipleiria.pt

João Vieira
Curso de Engenharia Informática
Sistemas de Informação
Politécnico de Leiria
Leiria, Portugal
2191191@my.ipleiria.pt

Resumo

Este relatório tem como objetivo apresentar o desenvolvimento de uma aplicação de entregas ao domicílio chamada FasTuga Driver, realizada no âmbito da unidade curricular de Tópicos Avançados de Engenharia de Software.

No decorrer deste documento, será abordado as funcionalidades, metodologias usadas, arquitetura e testes realizados ao longo do desenvolvimento.

Palavras-chave—SCRUM, Extreme Programming, Sprint, Product Backlog

Abstract

This report aims to present the development of a home delivery application called FasTuga Driver, carried out within the course of Advanced Topics in Software Engineering.

Throughout this document, it will be discussed the functionalities, methodologies used, architecture and tests carried out throughout the development.

Keywords— SCRUM, Extreme Programming, Sprint, Product Backlog

I. INTRODUÇÃO

As diversas funcionalidades disponibilizadas pelo software dos dias de hoje contribuem para grande parte de como a nossa sociedade funciona, tendo o aumento da relevância e da variedade de software de qualidade ao longo dos anos contribuído para o desenvolvimento de diversas indústrias e setores. Múltiplas aplicações foram desenvolvidas como forma de suporte aos diferentes setores trazendo uma maior flexibilidade para os seus clientes, utilizadores, etc.

Sendo assim, foram também desenvolvidas aplicações que possibilitam serviços de entrega de comida, entre vários outros serviços de entregas removendo alguns dos problemas que os métodos tradicionais, têm tais como: o deslocamento até ao restaurante, filas de espera, etc.

Neste relatório será apresentada a aplicação desenvolvida no âmbito da disciplina de Tópicos Avançados de Engenharia de Software assim como a tecnologia usada, como esta

funciona, a sua arquitetura e o processo de desenvolvimento da mesma.

II. DESCRIÇÃO DO SISTEMA – FASTUGA DRIVER

FasTuga Driver é uma aplicação para dispositivos móveis com o simples objetivo de simplificar o processo de entrega de refeições ao domicílio realizado pelos condutores registados na plataforma.

A presente aplicação oferece a possibilidade ao condutor de ver as encomendas disponíveis para entrega, as encomendas que lhe estão atribuídas, e obter direções GPS em tempo-real, verificar o seu saldo e as suas estatísticas. Facilitando assim, todo o processo de entrega.

Por cada pedido entregue e dado como concluído, o utilizador é recompensado com o valor de uma taxa de entrega que está diretamente relacionada com a distância percorrida necessária para realizar a entrega. O valor da taxa é depois incrementado na conta do utilizador.

Na secção abaixo são explicadas as funcionalidades desenvolvidas.

A. Login

Neste primeiro ecrã, é onde o utilizador poderá navegar para o ecrã de registo, através do link “Sign up”, ou fazer a sua autenticação. Para além disso, também é possível selecionar, através da checkbox “Stay logged in”, se o utilizador pretende manter a sessão iniciada mesmo após fechar a aplicação.

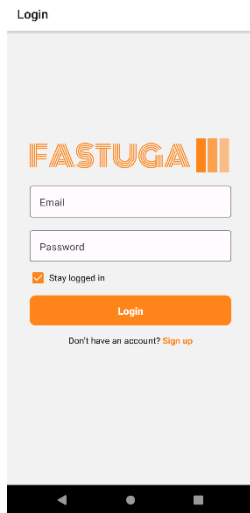


Figura 1 - Ecrã de Login

B. Registo (Register)

Caso o utilizador não tenha uma conta, será no ecrã de registo a que este se deverá dirigir. Neste ecrã, o utilizador terá de preencher todos os campos corretamente, para que assim possa criar uma conta com sucesso. Todos os campos presentes no ecrã estão protegidos por regras, de forma a manter a integridade dos dados.

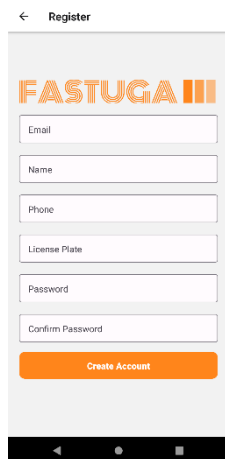


Figura 2 - Ecrã de Registo

C. Dashboard

Na Dashboard é onde o condutor poderá observar todas as suas encomendas, todas as encomendas sem condutores atribuídos e o seu saldo atual. Para além disso, é na Dashboard onde o condutor poderá atribuir encomendas a si mesmo e alterar o estado das mesmas

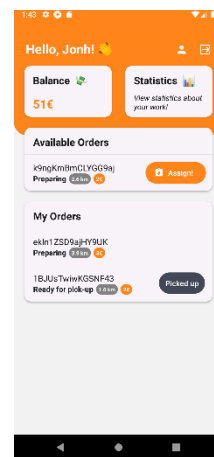


Figura 3 – Dashboard

D. Detalhes de Encomenda (Order Details)

Na página de detalhe de encomendas, o utilizador terá acesso ao: identificador distinto da encomenda, o estado atual da encomenda, preço referente ao pedido, quantidade e um mapa com o trajeto mais curto entre o restaurante e o local de entrega. Adicionalmente, caso o pedido esteja atribuído ao condutor, é apresentado um botão adicional que permite passar para o ecrã de navegação em tempo-real.

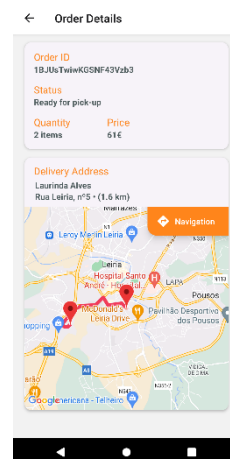


Figura 4 - Ecrã de Detalhes de Encomenda

E. Navegação (Navigation)

O ecrã de navegação disponibiliza direções em tempo-real, permitindo ao condutor visualizar o trajeto mais curto possível até ao seu destino enquanto conduz.

Caso o pedido se encontre em estado de preparação ou levantamento, as direções apresentadas são para o restaurante. Caso contrário, é apresentado as direções até ao local de entrega.

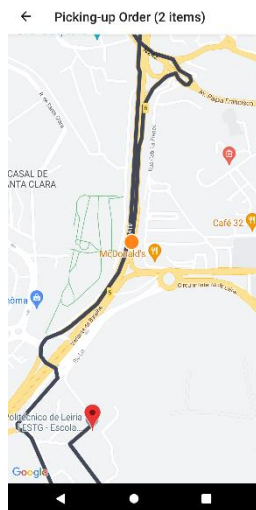


Figura 5 - Ecrã de Navegação

F. Estatísticas (Statistics)

Na página de estatísticas é onde o utilizador terá acesso a diversos indicadores, tais como:

- Ganhos totais;
- Número total de encomendas entregues;
- Número total de encomendas canceladas;
- Média de tempo por entrega com sucesso;
- Tempo total na entrega de encomendas;
- Número de clientes distintos;
- Maior distância percorrida numa encomenda.

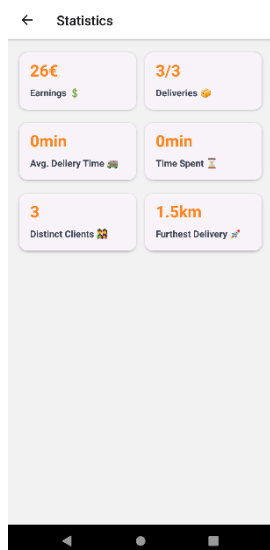


Figura 6 - Ecrã de Estatísticas

G. Perfil (Profile)

No ecrã de Perfil é onde o utilizador poderá ver e editar algumas das suas informações pessoais.

Tal como no ecrã de registo, neste ecrã todos os campos encontram-se protegidos por regras, de forma a manter a integridade dos dados.

Neste ecrã, também é possível ativar/desativar as notificações.

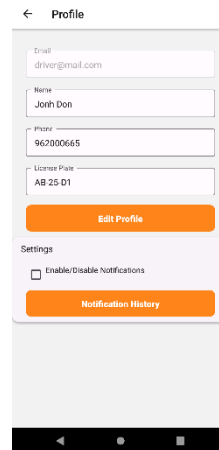


Figura 7 - Ecrã de Perfil

H. Histórico de Notificações (Notification History)

No ecrã de notificações o condutor terá acesso a uma lista de notificações recentes.

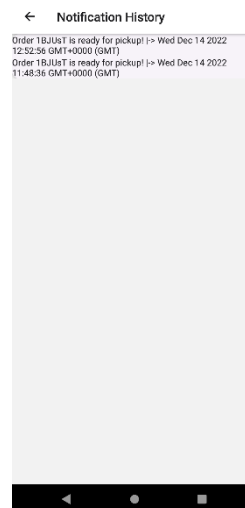


Figura 8 - Ecrã de Histórico de Notificações

III. PROCESSO DE DESENVOLVIMENTO

Foram usados alguns processos e técnicas da *framework SCRUM*, nomeadamente a parte relacionada com o *Sprint Backlog* que nos permitiu ter uma ideia geral do trabalho por realizar a cada semana, que foi a duração adotada para as Sprints de forma a nos comprometermos com o cliente a entregar partes da aplicação desenvolvidas e a funcionar. Foi utilizado também algumas características provenientes da metodologia ágil, *Extreme Programming*, como por exemplo: o conceito de *user story* que permite traduzir as variadas funcionalidades numa linguagem informal e natural com o intuito de serem perceptíveis ao utilizador final. Foi também utilizado a *SCRUM board* que permite acompanhar o estado de cada uma das *user stories* de maneira a observar facilmente quais *user stories* foram atribuídas, as *user stories* que estão a ser desenvolvidas, quais *user stories* pendentes de testes e por último as *user stories* que estão completas.

Foi usado um *software* de controlo de versões distribuído (DVCS) intitulado de *git*, este *software* permitiu manter um registo dos vários versionamentos do *software*.

O repositório remoto está localizado no website *bitbucket*, sendo este um *hosting service* de repositórios *git*. Foi adquirido recentemente pela *atlassian* pelo que tem tido melhores integrações com o *Jira* permitindo, por exemplo concluir *user stories* automaticamente após o ramo ao qual está associado ser “merged”.

Estes automatismos permitiram-nos ser mais eficientes e tornar o planeamento do projeto menos trabalhoso.

IV. ARQUITETURA DO SISTEMA

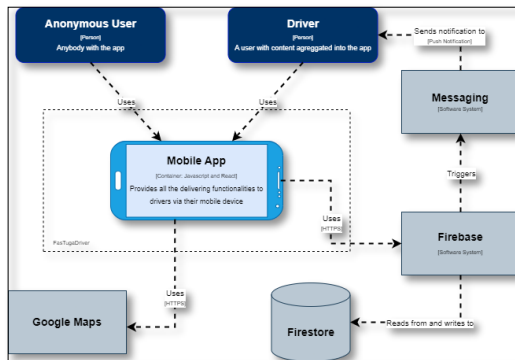


Figura 9 - Arquitetura do Sistema

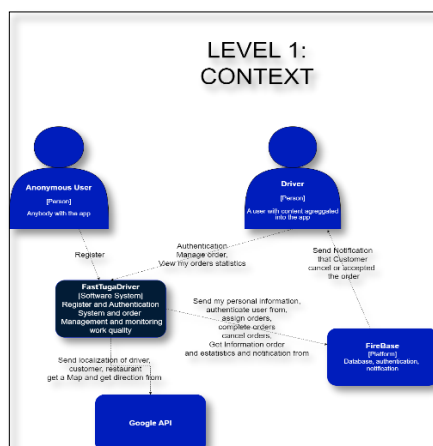


Figura 10 – Nível 1 (Contexto)

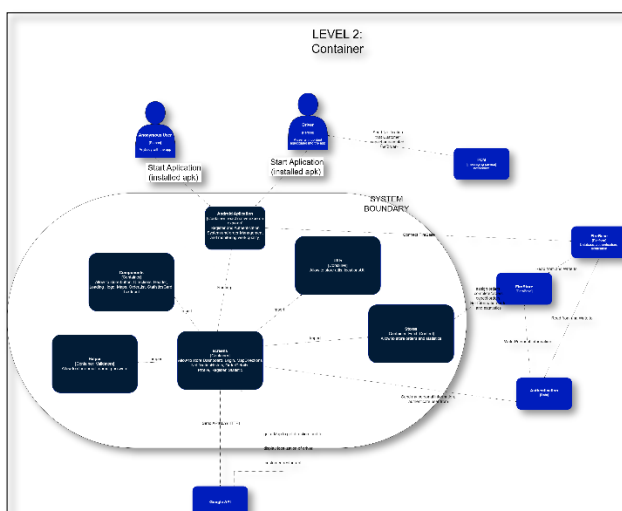


Figura 10 – Nível 2 (Container)

A. Utilizador (Driver)

O utilizador usa a aplicação móvel, onde cada interação é realizada sob a forma de pedidos *HTTP*.

B. Aplicação Mobile

A aplicação mobile, desenvolvida em *React Native*, representa a camada de apresentação que permite a interação do utilizador com o sistema.

C. Firebase

Todos os pedidos efetuados pela aplicação móvel, são respondidos pelo *Firebase*, plataforma escolhida para armazenar dados (*Firestore*), efetuar autenticação de utilizadores (*Authentication*) e gerar notificações (*Messaging*).

V. VERIFICAÇÃO E VALIDAÇÃO

Para a realização dos testes automatizados de cada funcionalidade, foi utilizado o *Katalon* que é uma plataforma de testes automatizados. Para cada *user story* foi criado uma *feature file* (*gherkin*). A Cada *feature file* está também associado pelo menos um ficheiro *groovy*, que contém o código que permite executar cada teste.

VI. CONCLUSÃO E TRABALHO FUTURO

A realização deste projeto permitiu-nos aprofundar conhecimentos sobre o desenvolvimento de software em equipa, dando uso a metodologias, ferramentas e frameworks presentes no mercado de trabalho.

Foram usadas ferramentas de controle de versões (*Git*), uma ferramenta de planeamento e monitorização de software (*Jira*) o que contribuiu para um desenvolvimento fluido e organizado.

Por fim, podemos concluir que com a realização deste projeto os elementos deste grupo adquiriram conhecimentos valiosos para aplicar no mercado de trabalho.

APPENDIX

A. Contribuição de cada elemento

No decorrer do desenvolvimento deste trabalho de grupo houve uma grande entajada e cumplicidade entre os membros do grupo. Por exemplo, caso um membro do grupo estivesse com dificuldades em devolver alguma funcionalidade a solicitação por ajuda era rapidamente respondida.

Em termos de funcionalidades todos os membros do grupo participaram na mesma funcionalidade, porém houve alguns com mais destaque. Especificamos abaixo o contributo de cada uma das funcionalidades implementadas:

- **Login:**
 - Francisco, João, Martim
- **Registo:**
 - Martim
- **Perfil:**
 - Diogo
- **Dashboard:**
 - Martim, João, Diogo, Francisco
- **Detalhes de Encomenda:**

- João, Martim
- **Navegação:**
 - Martim
- **Statistics:**
 - Francisco, Martim
- **Notificações:**
 - João

B. Executar o projeto.

Os comandos necessários para executar o projeto são os seguintes:

1. **npm install**
2. **expo start**

C. Gerar apk

Para gerar a apk, será necessário instalar primeiro o *aes* e criar uma conta na plataforma *expo*.

1. **npm install -g eas-cli**
2. **eas init**
3. **eas build -p android --profile preview**

D. Credenciais de acesso

- Email: driver@mail.com
- Password: 123456

E. Executar sistema de notificações

Para disparar as notificações é necessário aceder à pasta *firebase-fun* onde se encontram dois ficheiros *typescript* após ser feito **npm install**. Existem dois comandos a usar com o *npm*, no entanto é necessário um ficheiro *privatekey.json*, **npm run serve** que irá simular um cenário em que o restaurante mudou o estado de alguma encomenda e pretende notificar os respetivos condutores e o comando **npm run notifications clean**, que é um comando que permite retirar a propriedade que informa que certo utilizador já foi notificado para certa encomenda. Esta comando também limpa todo o histórico de notificações.

Como referido anteriormente é necessário um ficheiro *privatekey.json* este não foi incluído no repositório pois permitiria um acesso por completo as várias componentes administrativas do projeto (*admin-sdk*) o ficheiro irá ser fornecido apesar de caso seja necessário migrar para um projeto *firebase*.