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

João Vieira  
*Curso de Engenharia Informática*  
Sistemas de Informação *Politécnico de Leiria*Leiria, Portugal   
2191191@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

Resumo

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

Ao longo 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 from a software engineering perspective, carried out within the course of Advanced Topics in Software Engineering.

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

**Keywords— SCRUM, Extreme Programming, Sprint, Product Backlog**

# 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.

# Descrição do Sistema – FasTuga Driver

Realizada para uso mobile, a aplicação FasTuga Driver permite aos seus utilizadores (condutores) efetuarem entregas ao domicílio de pedidos realizados no website do restaurante FasTuga.

Esta aplicação facilita todo o processo de entrega permitindo ao utilizador ver os pedidos, escolher quais pretende entregar e obter direções GPS para a morada do pedido.

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.

## 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.

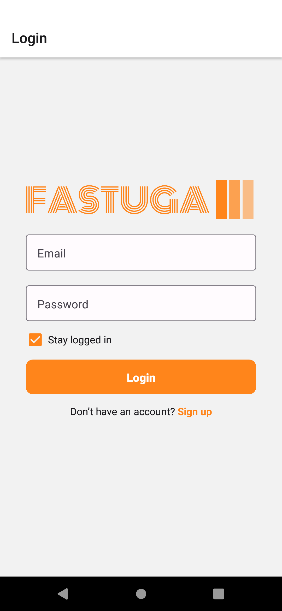


Ilustração 1 - Ecrã de Login

## 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.

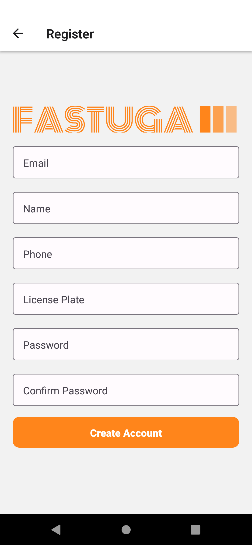


Ilustração 2 - Ecrã de Registo

## 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

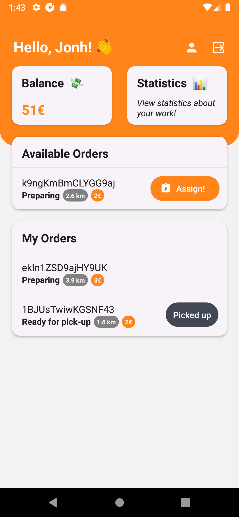


Ilustração 3 – Dashboard

## 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.

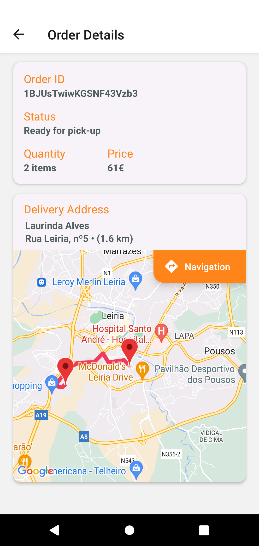


Ilustração 4 - Ecrã de Detalhes de Encomenda

## 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.

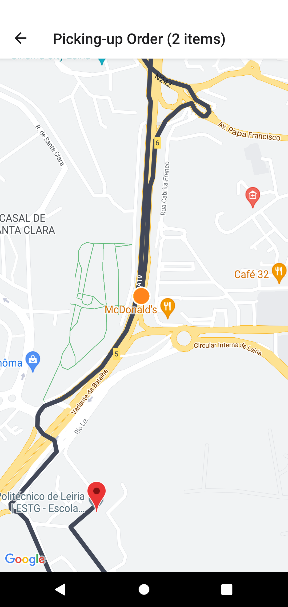


Ilustração 5 - Ecrã de Navegação

## 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.

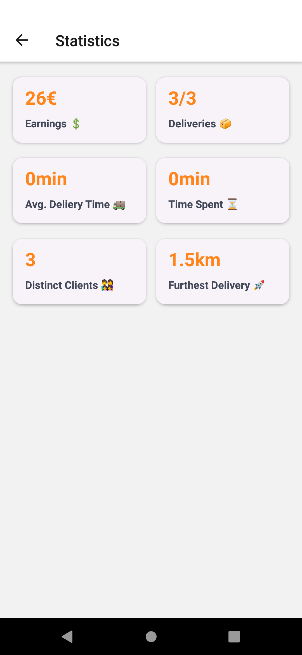


Ilustração 6 - Ecrã de Estatísticas

## 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.

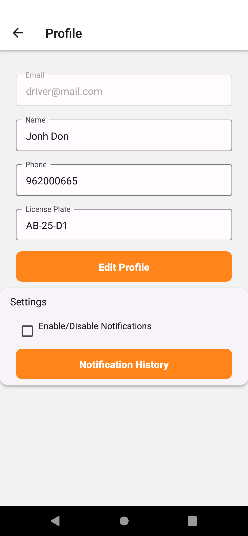


Ilustração 7 - Ecrã de Perfil

## Histórico de Notificações (Notification History)

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

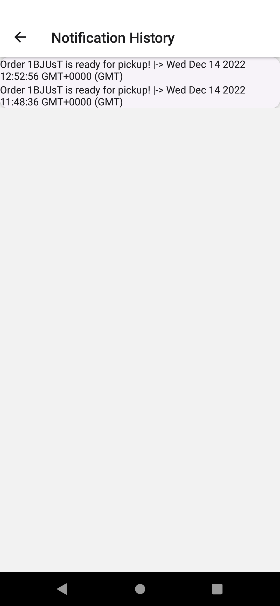


Ilustração 8 - Ecrã de Histórico de Notificações

# 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 nós comprometemos 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 váriadas funcionalidades numa linguagem informal e natural com o intuito de serem percetiveis pelo Utilizador Final*  Foi também utilizado a *SCRUM board* que permite acompanhar o estado de cada uma das *user stories* de maneira a facilmente ter noção que User Stories foram atribuídas, as User Stories que estão a ser desenvolvida, quais os user stories que ainda não tem testes concluidos e por ultimo as user stories que estão completas.

Para controlo de versoes usamos um software de controlo de versoes distribuido (DVCS) entitulado de git, este permitiu-no manter um registo das varios versionamentos do software e manter sempre uma linhagem de versoes(neste caso no ramo Master) que sabiamos que estavam a funcionar sem ter de sacrificar as linhagem mais recentes ou prototipos. O repositorio remoto usado está localizado no website entitulado de bitbucket, sendo este um hosting service de repositorios git. Foi adquirido recentemente pela atlassian pelo que tem tido melhores integrações com o Jira permitindo por exemplo concluir user stories automaticamente apos o ramo ao qual está associado ser merged.

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

# Arquitetura do Sistema

A nossa aplicação funciona essencialmente com 3 componentes, a aplicação em si (A), visível para o cliente e a API (B) e a base de dados (C).

# 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).

De seguida procedemos a copiar as intruçoes previamente escritas nos vários

# Conclusão e Trabalho Futuro

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

Foram usadas ferramentas de controle de versões (Git), uma ferramenta de planejamento e monitorização de software (Jira) o que contribuiu para um desenvolvimento fluído 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.

# references

1. https://www.npmjs.com/package/fcm-notification
2. https://www.npmjs.com/package/@react-native-firebase/messaging
3. https://www.npmjs.com/package/expo
4. https://www.npmjs.com/package/expo-location
5. https://www.npmjs.com/package/react-native-maps-directions

# Appendix

No decorrer do desenvolvimento deste trabalho de grupo houve uma grande entreajuda 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

ISTO É REALMENTE PRECISO?

Para construir o apk é necessário usar a ferramenta eas, cujo se instala pelo npm usando o comando npm install -g eas-cli o comando a executar é o eas build -p android --profile preview.

Infelizmente, devido a natureza das dependências das push up notifications relacionadas ao firebase perdemos a habilidade de usar o comando que usamos na maioria do desenvolvimento da aplicação (npx) expo start, sendo que o prefixo npx é necessário em linux.devido ao comando expo por si só ter sido declarado obsoleto.

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 e necessario um ficheiro privatekey.json que falaremos no futuro, npm run serve que irá simular um cenário em que o Fastuga restaurante mudou o estado de alguma receita e pretende notificar os respectivos condutores e o comando npm run notifications Clean que é um comando mais para o lado do desenvolvedor que permite retirar a propriedade que informa a função anteriormente referida que certa order já foi notificada e que nao precisa de o ser novamente, esta também limpa todo o histórico de notificações pois notamos que era ficava difícil de ver se realmente uma notificação tinha sido disputada quando no histórico já estavam 10 exatamente iguais.

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 novo existem instruções de como obter este ficheiro nas references, estando no link fcm-notification.

Existe no entanto um erro cujo não conseguimos resolver por estar fora do nosso controle que é o facto do typescript recentemente ter provavelmente atualizado os tipos causando um monte de erros ao executar os dois comandos, no entanto os ficheros iram ser transcritos para javascript sendo possível os executar manualmente usando node <ficheiro>.