**Universidade Europeia**

**Engenharia Informática**

**Engenharia de Software**

Manuel Beijinho

50037192​@iade.pt

Junho 2020

**Relatório de Engenharia de Software**

Conteúdo

[1. Descrição do Projeto de Software 3](#_Toc43906793)

[1.1. Visão Geral do Projecto 3](#_Toc43906794)

[1.2. Propósito do Projecto 3](#_Toc43906795)

[1.2.1. Sector de Negócio 3](#_Toc43906796)

[1.2.2. Objectivos Principais do Projeto 3](#_Toc43906797)

[1.3. Âmbito 3](#_Toc43906798)

[1.3.1. Contexto do trabalho 4](#_Toc43906799)

[1.3.2. Produtos Rivais e avaliação comparativa 4](#_Toc43906800)

[1.4. Esboços e protótipos iniciais 5](#_Toc43906801)

[1.5. Stakeholders 5](#_Toc43906802)

[1.5.1. Equipa de desenvolvimento 5](#_Toc43906803)

[1.5.2. Utilizador 5](#_Toc43906804)

[1.5.3. Outros stakeholders 6](#_Toc43906805)

[1.6. Constrangimentos gerais 6](#_Toc43906806)

[1.7. Convenções e definições – Conceitos chave 7](#_Toc43906807)

[2. Requisitos de Software 8](#_Toc43906808)

[2.1. Atores e use cases 8](#_Toc43906809)

[2.1.1. Atores 8](#_Toc43906810)

[2.1.2. Casos de Uso 8](#_Toc43906811)

[2.1.3. Diagrama de Casos de Uso 9](#_Toc43906812)

[2.2. Requisitos funcionais 10](#_Toc43906813)

[2.3. Requisitos não funcionais 11](#_Toc43906814)

[2.3.1. Requisitos de performance 11](#_Toc43906815)

[2.3.2. ​Requisitos de capacidade e armazenamento 11](#_Toc43906816)

[2.3.3. Requisitos de usabilidade 12](#_Toc43906817)

[2.3.4. Requisitos de Internacionalização 12](#_Toc43906818)

[2.3.5 Requisitos de adaptação 12](#_Toc43906819)

[3. Design de Software 13](#_Toc43906820)

[3.1. Proposed software architecture 13](#_Toc43906821)

[3.2. User Interface Design 14](#_Toc43906822)

[4. Construção de Software 18](#_Toc43906823)

[4.1. Ambiente de desenvolvimento integrado 18](#_Toc43906824)

[5. Testes de software 19](#_Toc43906825)

[5.1. Planos para testes e objetivos 19](#_Toc43906826)

[5.2 Funcionalidades testadas 19](#_Toc43906827)

[5.3 Critérios de avaliação dos testes 19](#_Toc43906828)

[5.4 Testes unitários 19](#_Toc43906829)

[5.5 Testes de sistema 20](#_Toc43906830)

[6. Feedback AgileFactor 20](#_Toc43906831)

# 1. Descrição do Projeto de Software

## 1.1. Visão Geral do Projecto

Este projeto teve como objetivo desenvolver uma aplicação móvel para tablets Android que permita executar de forma segura as tarefas de um agente de uma instituição financeira, considerando os pressupostos de mobilidade e de georreferenciação das operações executadas.

## 1.2. Propósito do Projecto

Este projeto foi desenvolvido para avaliação final de Project Factory, cadeira de 2º Semestre do 3º Ano do curso de Engenharia Informática da Universidade Europeia. Foi proposto pela AgileFactor e desenvolvido com vista nos requisitos da empresa.

A aplicação permite realizar operações que concretizam um “caixa-avançado” e que incluem as operações de levantamento, depósito, visualização de movimentos/consulta de saldos e pagamento de serviços/outros.

### 1.2.1. Sector de Negócio

A aplicação é direcionada ao setor financeiro/bancário. Uma entidade financeira adquire o software para poder utilizar dentro da sua organização.

### 1.2.2. Objectivos Principais do Projeto

Permitir às pessoas que tenham acesso a serviços financeiros de qualidade, a baixo custo e próximos de si, potenciando maiores níveis de igualdade e de acessibilidade às oportunidades para melhoria da qualidade de vida das pessoas.

## 1.3. Âmbito

Aplicação tablet que permite realizar operações como depósitos, levantamentos, consulta de movimentos/saldo para clientes interessados.

A aplicação deve ser segura e manter um *tracking* de todas as operações realizadas, bem como quem as realizou e onde realizou.

### 1.3.1. Contexto do trabalho

A necessidade de realizar operações financeiras em mobilidade tem-se manifestado nos últimos anos com o desenvolvimento da tecnologia e dos modos de vida em movimento, bem como pela falta de sistemas financeiros acessíveis à população, na maioria das regiões do mundo, tal como é suportado pelos relatórios do Banco Mundial.

Assim, a aplicação permite que as pessoas tenham acesso a serviços financeiros a baixo custo e próximos de si, combatendo a desigualdade social.

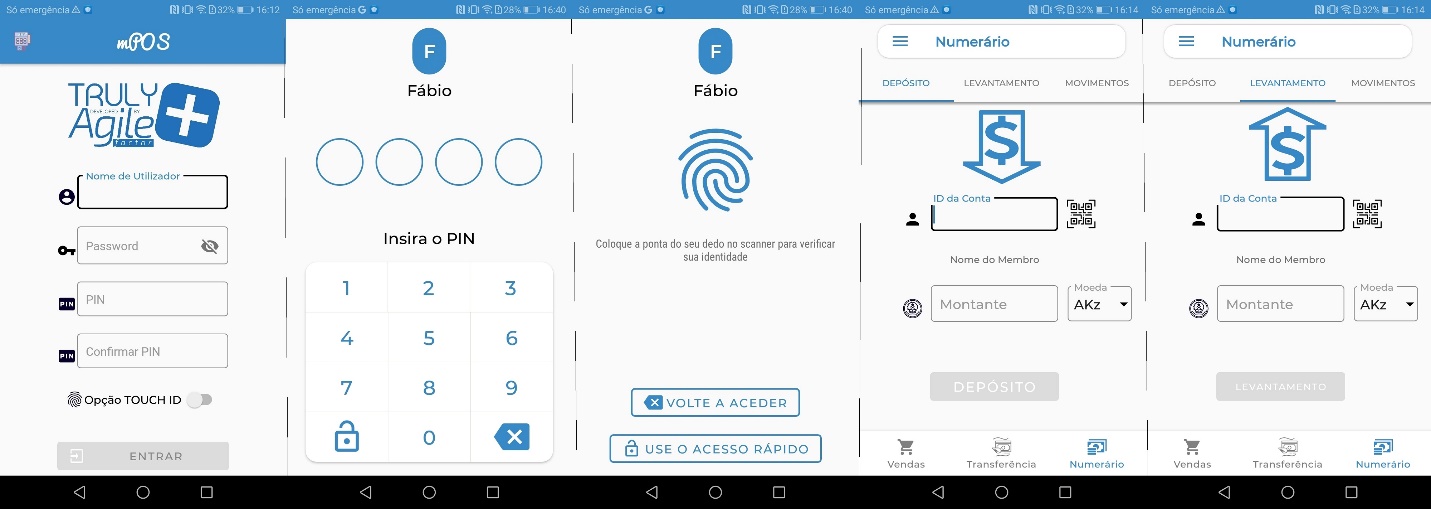
### 1.3.2. Produtos Rivais e avaliação comparativa

[CB Bank mobile agent banking](https://www.cbbank.com.mm/en/sme-banking/online-channels/mobile-agent-banking): Permite aos agentes aceitar depósitos, fazer transferências de fundos, pagar serviços.

[Emerico agent banking](https://emerico.biz/mobile-agent-banking/): Permite prestar vários serviços, como a aplicação acima mencionada, e outros: levantamentos, alocação de fundos, abertura de contas, empréstimos.

[BPC - SmartVista](https://www.bpcbt.com/smartvista-solutions/agent-banking/): Aplicação paralela à Emerico agente banking.

## 1.4. Esboços e protótipos iniciais



Uma imagem com captura de ecrã

Descrição gerada automaticamente

## 1.5. Stakeholders

### 1.5.1. Equipa de desenvolvimento

A equipa de desenvolvimento é constituída pelo programador da aplicação mobile - Manuel Beijinho, e pela equipa responsável pelo backend da aplicação – AgileFactor.

### 1.5.2. Utilizador

O utilizador da aplicação é uma entidade financeira que adquira o software para utilizar nos seus negócios.

### 1.5.3. Outros stakeholders

A equipa de gestores de Projetos é constituída por:

- AgileFactor

- Professor Jacinto Estima

- Professor José Vasconcelos

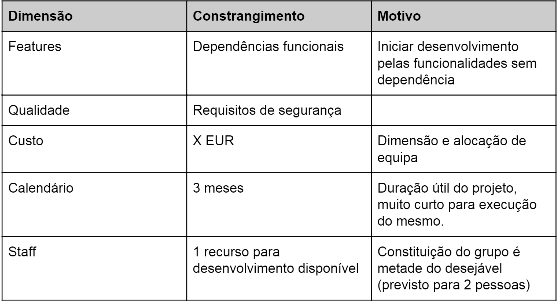
Os project owners são:

- AgileFactor

- Professor Jacinto Estima

- Professor José Vasconcelos

## 1.6. Constrangimentos gerais



A duração do projeto foi de quatro meses e isto implicava as seguintes tarefas: desenvolvimento do software, documentação e apresentações do projeto. O âmbito do projeto foi definido com base neste espaço de tempo para este conjunto de deliverables. De forma a entregar uma aplicação funcional e com serviços completos foi necessário reservar tempo para melhorar a lógica de implementação e limpar código.

## 1.7. Convenções e definições – Conceitos chave

Agente bancário – operador contratado pela entidade financeira para processar transações de clientes.

# 2. Requisitos de Software

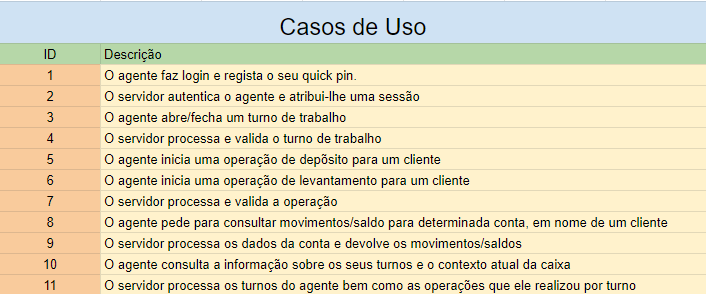
## 2.1. Atores e use cases

### 2.1.1. Atores

Agente bancário – utilizador da aplicação, responsável por processar as transações do cliente.

AgileFactor APIs – processa toda a informação e realiza todas as operações necessárias para que o serviço seja prestado ao cliente.

### 2.1.2. Casos de Uso



### 2.1.3. Diagrama de Casos de Uso

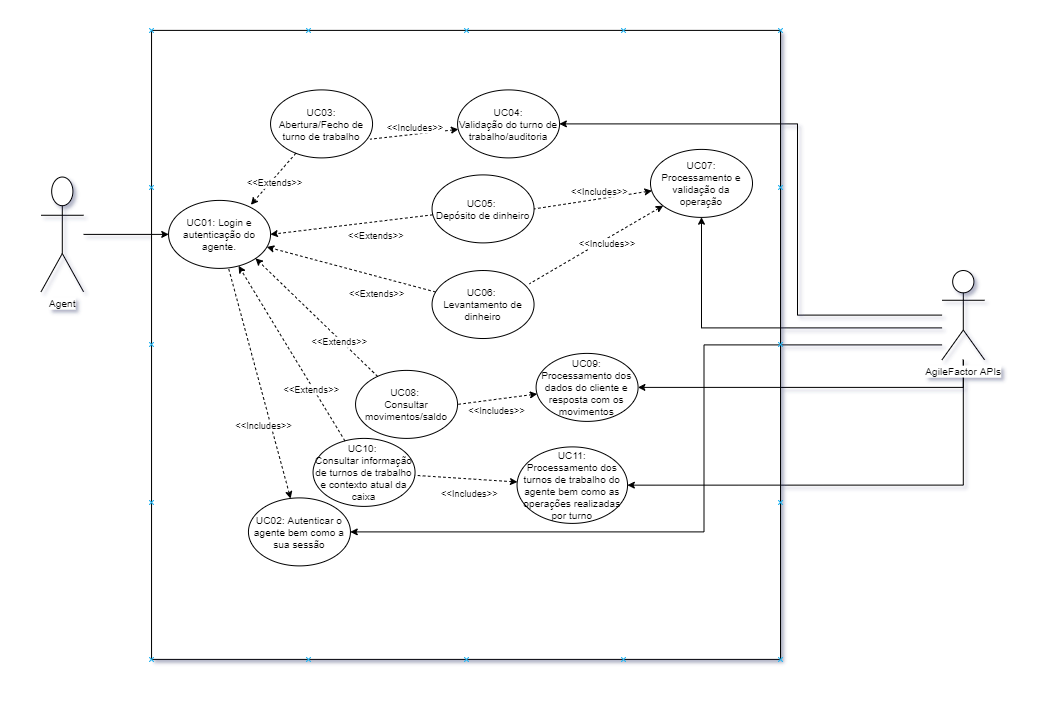


Figura 1 - Diagrama de Casos de Uso mAgent

## 2.2. Requisitos funcionais

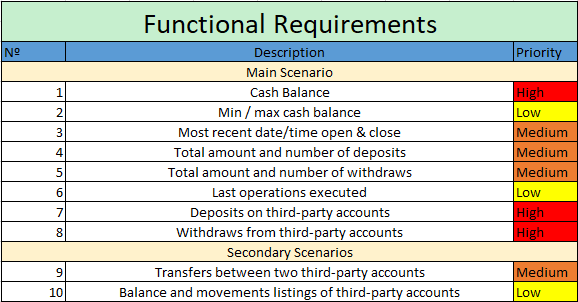


Figura 2- Tabela de requisitos funcionais

## 2.3. Requisitos não funcionais

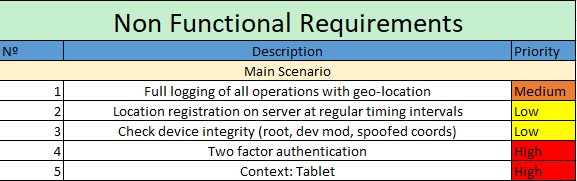


Figura 3 - Tabela de requisitos não funcionais

### 2.3.1. Requisitos de performance

O tempo de resposta da aplicação para tarefas locais é praticamente imediato, inferior a 1s.

Tarefas que precisem de pedir algo ao servidor podem demorar algum tempo, dependendo da qualidade da rede e/ou sobrecarga do servidor.

### 2.3.2. ​Requisitos de capacidade e armazenamento

É apenas guardado no armazenamento interno da aplicação alguns tokens do utilizador e o PIN estabelecido para a sessão.

O resto da informação está guardada numa base de dados e é pedida ao servidor quando requisitada.

### 2.3.3. Requisitos de usabilidade

A aplicação segue os princípios de design propostos pela Google: “Material Design”. A aplicação usa a mesma palete de cores em todos os ecrãs, os inputs são todos controlados e contêm várias ajudas e indicadores para o utilizador saber o que é necessário preencher. Cada ação realizada tem sempre um feedback.

### 2.3.4. Requisitos de Internacionalização

A aplicação deve suportar multi-língua e multi-formatação. Atualmente a linguagem suportada é o inglês, mas a aplicação está preparada para suportar qualquer linguagem, assumido que o ficheiro com as traduções é preenchido.

A aplicação suporta atualmente multi-formatação: é feita uma verificação com o tipo de formatação e horário do sistema (tablet) do agente e é dessa forma que é apresentado na aplicação.

### 2.3.5 Requisitos de adaptação

● Diversos ecrãs

A aplicação suporta diversos tamanhos de ecrã tablet.

# 3. Design de Software

## 3.1. Proposed software architecture

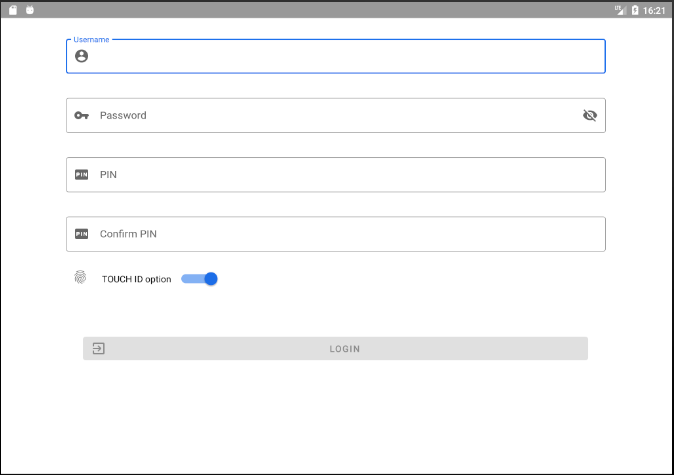
No diagrama abaixo está representada a arquitetura da solução proposta. A camada de apresentação inclui todos os ecrãs necessários ao agente para poder realizar as operações essenciais ao negócio. Na camada de negócio estão incluídos todos os serviços necessários a consumir para que cada operação seja realizada com sucesso. Estes dados e operações comunicam entre camadas através das APIs desenvolvidas pela AgileFactor, e são armazenados nas bases de dados da mesma.

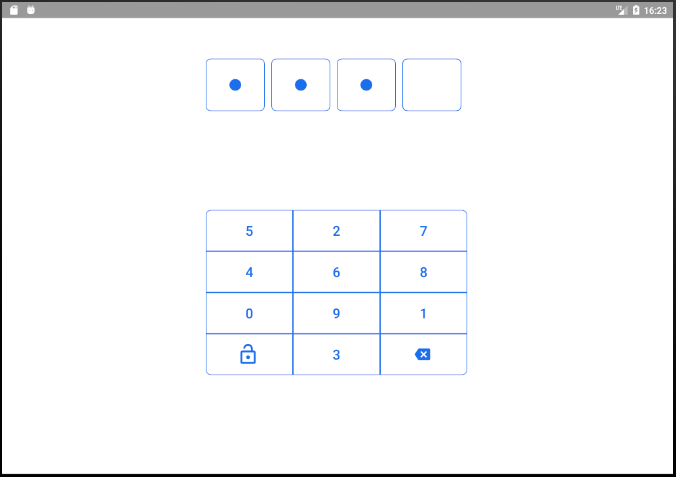
Uma imagem com captura de ecrã

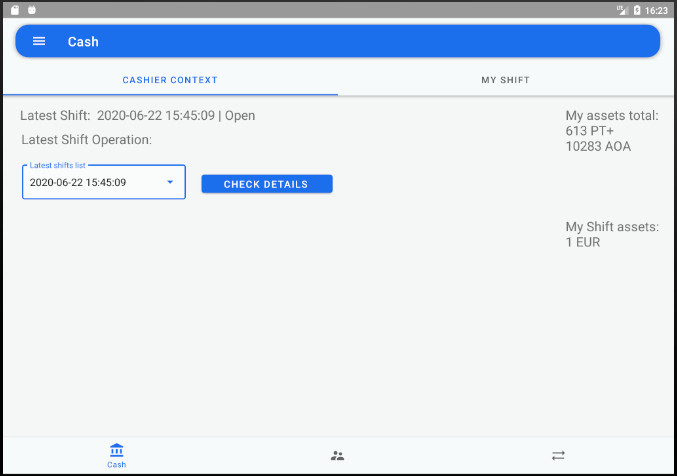
Descrição gerada automaticamente

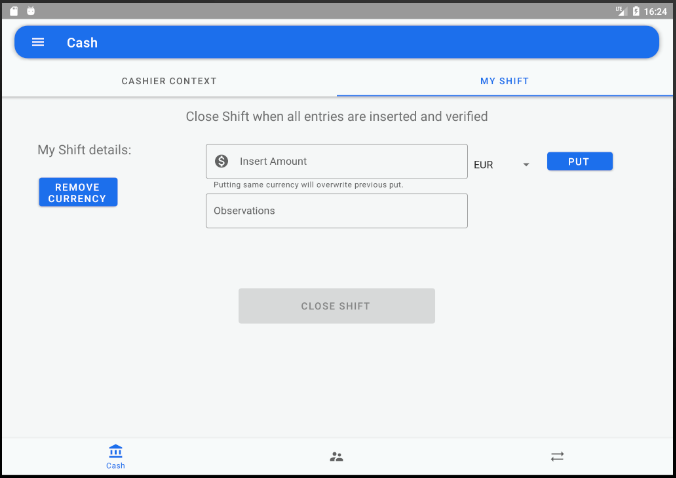
Figura 4 - Diagrama da arquitetura da solução

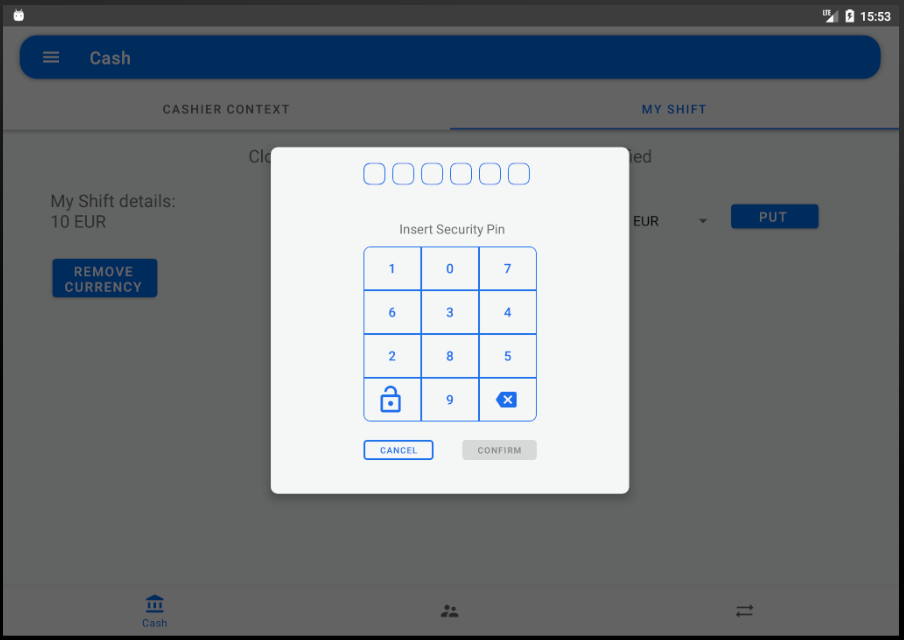
## 3.2. User Interface Design

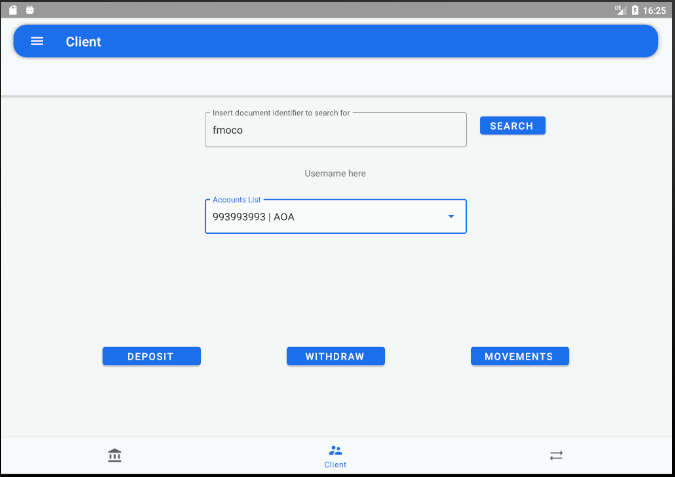


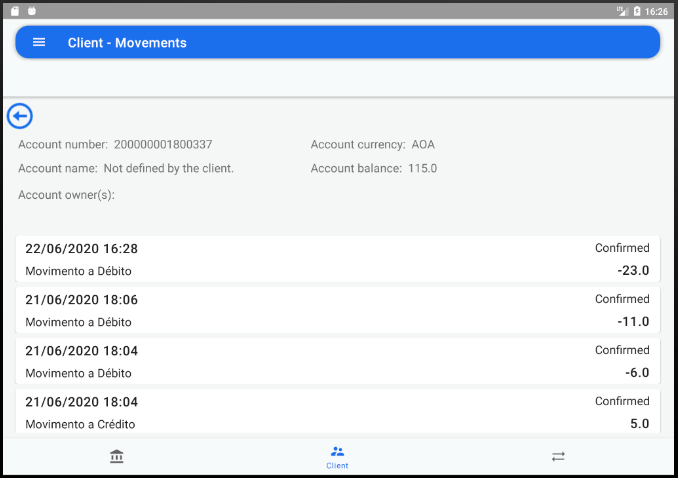


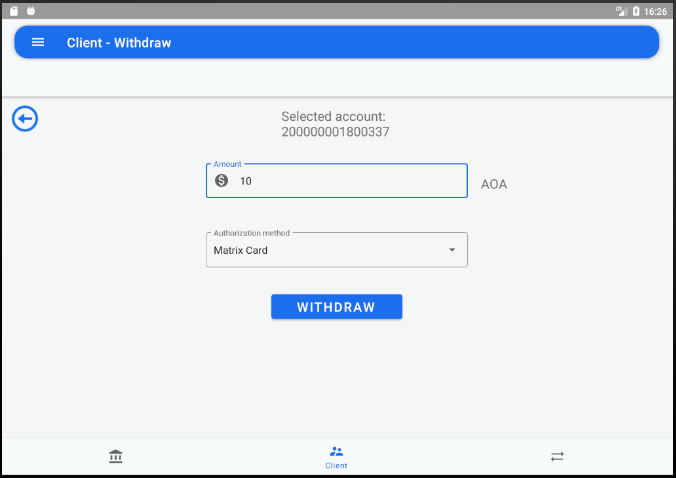




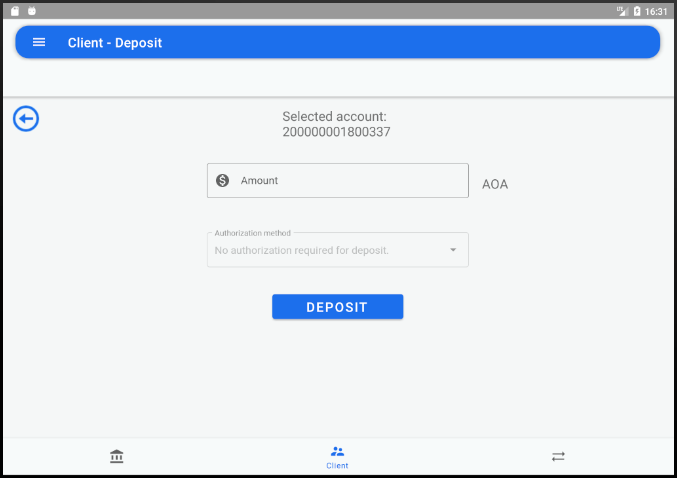












# 4. Construção de Software

## 4.1. Ambiente de desenvolvimento integrado

Projeto desenvolvido usando o Android Studio

Controlo de versões através do Git na aplicação GitKraken

# 5. Testes de software

## 5.1. Planos para testes e objetivos

A aplicação foi desenvolvida com vista nos requisitos apresentados pela AgileFactor. Deste modo, o objetivo era desenvolver uma aplicação coesa e que conseguisse, de forma segura e otimizada, executar todas as operações necessárias ao contexto da aplicação. Assim, cada funcionalidade necessária foi considerada um objetivo independente, de modo que a cada momento de integração da funcionalidade na aplicação eram realizados testes de performance e qualidade a todo o sistema.

O objetivo era conseguir integrar todos os módulos necessários na aplicação sem pôr em risco funcionalidades já implementadas.

Os testes foram realizados pelo programador, pela empresa que propôs o projeto e por colegas do curso que se voluntariaram para realizar os mesmos.

## 5.2 Funcionalidades testadas

Todas as funcionalidades foram testadas individualmente. Após aprovação do funcionamento da mesma esta era integrada no sistema inteiro, para que pudesse ser testada como um todo (para averiguar eventuais incompatibilidades entre módulos). Assim foi possível garantir que o desenvolvimento da aplicação estava controlado e não iria ser posto em causa o bom funcionamento da aplicação.

Estes testes foram realizados pelo programador e pela AgileFactor (através de demonstrações).

## 5.3 Critérios de avaliação dos testes

Os critérios de avaliação foram: averiguar se a funcionalidade causava algum tipo de incompatibilidade com o restante sistema; se a funcionalidade apresentava o comportamento esperado; se o utilizador conseguia utilizar a aplicação sem esforço de compreensão; se a aplicação se comportava da forma que o utilizador esperava que se comportasse.

## 5.4 Testes unitários

Os testes unitários foram realizados pelo programador e pela AgileFactor através de demonstrações. Cada funcionalidade era testada individualmente (como uma unidade) e de seguida era integrada no sistema.

## 5.5 Testes de sistema

Os testes de sistema foram realizados ao longo do desenvolvimento pelo programador principal e pela equipa da AgileFactor de modo a garantir o bom funcionamento da aplicação e que a mesma mostrava o comportamento esperado.

No final do desenvolvimento a aplicação foi também testada por 2 colegas de curso que tiveram de seguir um guião (que correspondia ao que um agente teria de lidar caso estivesse a usar a aplicação num caso real) de modo a dar um feedback sobre a usabilidade da app. Ambos os colegas deram um feedback, no geral, positivo sobre a aplicação, não mostrando dificuldade a utilizar a mesma.

Os testes foram pertinentes pois permitiram ao programador aperceber-se de alguns módulos onde, apesar do sucesso, os “software-testers” se mostraram mais confusos sobre como utilizar a app, melhorando então o conhecimento do programador para projetos futuros.

# 6. Feedback AgileFactor

A empresa deu um feedback bastante positivo relativamente ao projeto de estágio. A app foi cumprida face aos requisitos propostos e todos os obstáculos apresentados pelas condições de pandemia e pela falta de experiência com a tecnologia Android Studio foram ultrapassados com sucesso.

A AgileFactor destacou as seguintes “características de excelência” do programador:

- Capacidade de compreender/analisar/solucionar o negócio.

- Autonomia.