

1 2 9 0



UNIVERSIDADE D  
COIMBRA

Miguel Costa Monteiro

**DESENVOLVIMENTO DE UM SISTEMA DE APOIO AO  
ENSINO BASEADO EM DISPOSITIVOS MÓVEIS**

**Dissertação no âmbito da Especialização em Robótica, Controlo e Inteligência Artificial do Mestrado em Engenharia Eletrotécnica e de Computadores, orientada pelo Professor Doutor Jorge Miguel Sá Silva e pelo Professor Doutor André Miguel de Almeida Marrão Rodrigues e apresentada ao Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologias da Universidade de Coimbra**

julho de 2023





UNIVERSIDADE D  
**COIMBRA**

**Desenvolvimento de um sistema de apoio ao ensino  
baseado em dispositivos móveis**

**Miguel Costa Monteiro**

Dissertação para obtenção do Grau de Mestre em  
**Engenharia Electrotécnica e de Computadores**

**Supervisores:**

Doutor Jorge Sá Silva

Doutor André Rodrigues

**Júri**

Presidente: Doutor Paulo Menezes

Vogal: Doutor Mário João Santos

Vogal: Doutor Jorge Sá Silva

**Julho de 2023**



# Agradecimentos

Começo por agradecer aos meus orientadores Professor Jorge Sá Silva e ao Professor André Rodrigues pela oportunidade de integrar este projeto e por todo o apoio e disponibilidade ao longo da realização do mesmo.

Gostaria de agradecer a todos os meus colegas de laboratório e equipa de trabalho, principalmente ao José Oliveira e Guilherme Borges, pelo companheirismo, apoio e valioso contributo para o trabalho em equipa.

Quero expressar o meu especial agradecimento ao Marcelo Fernandes por partilhar conhecimentos técnicos indispensáveis para a conclusão deste trabalho e pela sua imensa disponibilidade ao longo deste ano.

Queria agradecer à minha família, em especial aos meus pais por toda a sua paciência, apoio e pela oportunidade de tirar um curso superior.

Por último, quero agradecer à Ana que sempre me motivou, apoiou e ajudou a alcançar os meus objetivos ao longo do meu percurso académico.

# Abstract

Currently, an increasing number of people use smart devices, such as smartphones in their daily life. These devices can integrate several applications which, in general, discard the Human as an active element of the system, generalising the content given to the user. However, smartphones are able to collect behavioural, physical and social data related to the user. That sort of information is essential to create applications of assistance and monitoring. For that reason, the concept of Human-in-the-Loop Cyber-Physical System (HitLCPS) had emerged, where the Human is considered as part of a Cyber-Physical System (CPS).

This study is part of the process of recreating and expanding the ISABELA platform. The aim of this Internet of Things (IoT) application is to improve university students' academic performance. Initially, ISABELA platform was only developed for Android operating system. For that reason, its use to monitor and guide students was limited. Thus, it was proposed the creation of an innovative platform developed simultaneously for both Android and iOS operating systems. The platform employs news technologies, such as IoT paradigm, Artificial Intelligence (AI), chatbots and smartphones that will provide data to a HitLCPS.

In this study, the forms that compose the ISABELA application were created from a library that had been developed. This library will allow making small changes to the forms or easily adding new ones.

The ISABELA application was also used as the basis for another application - MAORI - which will be used in the recruitment phase of the Portuguese Air Force.

## Keywords

Internet of Things, Cyber-Physical-System, Human-in-the-loop Cyber-Physical-System, Smartphone

# Resumo

Atualmente, grande parte da população possui dispositivos móveis inteligentes, dos quais se destacam os *smartphones*. Estes dispositivos permitem a integração de várias aplicações que, de uma forma geral, descartam o Ser Humano como um elemento ativo do sistema, generalizando o conteúdo fornecido aos utilizadores. No entanto, os *smartphones* possibilitam a recolha de dados comportamentais, físicos e sociais do utilizador, que são essenciais para a criação de aplicações de auxílio e monitorização. Assim, surge o conceito de HitLCPS, que considera o Ser Humano como parte integrante de um CPS.

Este estudo faz parte do processo de recriação e expansão da plataforma ISABELA, uma solução baseada em IoT cujo objetivo é melhorar o desempenho académico dos estudantes universitários. A aplicação para *smartphone* foi desenvolvida inicialmente apenas para o sistema operativo *Android*, o que limitava a quantidade de estudantes que se conseguia monitorizar e orientar. Desta forma, propôs-se a criação de uma aplicação inovadora, desenvolvida em simultâneo para os sistemas operativos *Android* e *iOS*. A aplicação emprega novas tecnologias, como o paradigma da IoT, AI, *chatbots* e *smartphones*, que servirão para fornecer dados a um HitLCPS.

Neste estudo foi também desenvolvida uma biblioteca com o objetivo de facilitar a criação dos formulários que integram a aplicação ISABELA. Esta biblioteca permitirá fazer pequenas alterações nos formulários ou adicionar novos formulários com relativa facilidade. Por fim, efetuou-se uma adaptação da aplicação para utilização na fase de recrutamento da Força Aérea Portuguesa surgindo assim a aplicação MAORI.

## Palavras Chave

Internet of Things, Cyber-Physical-System, Human-in-the-loop Cyber-Physical-System, Smartphone

# Índice

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contextualização . . . . .	2
1.2	Motivação . . . . .	2
1.3	Objetivos . . . . .	3
1.4	Metodologia . . . . .	3
1.5	Estrutura da Dissertação . . . . .	4
<b>2</b>	<b>Tecnologias Base e Estado da Arte</b>	<b>6</b>
2.1	Tecnologias Base . . . . .	7
2.1.1	Android . . . . .	7
2.1.2	iOS . . . . .	7
2.1.3	.NET . . . . .	7
2.1.4	XAMARIN . . . . .	7
2.1.5	FIWARE . . . . .	8
2.2	Conceitos . . . . .	9
2.2.1	<i>Internet of Things</i> . . . . .	9
2.2.2	<i>Cyber-Physical System</i> . . . . .	9
2.2.3	<i>Human-in-the-Loop Cyber-Physical System</i> . . . . .	10
2.3	Plataformas atuais -ISABELA/BATINA e UCFrameworks . . . . .	10
2.4	Plataformas Semelhantes . . . . .	13
2.4.1	<i>BigBlueButton</i> . . . . .	13
2.4.2	<i>ALEKS</i> . . . . .	14
2.4.3	<i>Socrative</i> . . . . .	14
2.4.4	<i>StudentLife</i> . . . . .	15
2.5	Análise Comparativa . . . . .	16

## Índice

---

<b>3</b>	<b>Biblioteca SocialiteForms</b>	<b>18</b>
3.1	Contextualização . . . . .	19
3.2	Estrutura da biblioteca . . . . .	19
3.3	Características do ficheiro descritivo em <i>JSON</i> . . . . .	20
<b>4</b>	<b>Aplicação ISABELA</b>	<b>27</b>
4.1	Contextualização . . . . .	28
4.2	Descrição . . . . .	28
4.3	Requisitos . . . . .	29
4.3.1	Requisitos não funcionais . . . . .	29
4.3.2	Requisitos Funcionais . . . . .	29
4.4	Arquitetura . . . . .	30
4.4.1	Arquitetura geral . . . . .	31
4.4.2	Arquitetura da aplicação móvel . . . . .	33
4.5	Formulários da aplicação ISABELA . . . . .	34
4.5.1	Formulário de proximidade . . . . .	35
4.5.2	Formulário de Sono . . . . .	36
4.5.3	Formulário <i>POMS</i> . . . . .	37
4.5.4	Formulário emocional com imagens . . . . .	37
4.5.5	Formulário de transporte . . . . .	38
4.5.6	Formulário <i>HEXACO</i> . . . . .	40
4.5.7	Formulário de finalidade da aplicação . . . . .	41
<b>5</b>	<b>MAORI</b>	<b>43</b>
5.1	Contextualização . . . . .	44
5.2	Desenvolvimento . . . . .	44
5.2.1	Página do treino de Voo . . . . .	45
5.2.2	Formulário <i>POMS</i> . . . . .	45
<b>6</b>	<b>Avaliações e Testes</b>	<b>47</b>
<b>7</b>	<b>Conclusão e Trabalho Futuro</b>	<b>51</b>
7.1	Conclusão . . . . .	52
7.2	Trabalho Futuro . . . . .	52

---

# Lista de Figuras

2.1	Arquitetura de uma aplicação em <i>Xamarin</i> . Adaptado de [1] . . . . .	8
2.2	Processo de um sistema Human-in-the-loop (HiTL). Adaptado de [2]	10
2.3	Alguns <i>layouts</i> da aplicação ISABELA . . . . .	11
2.4	Alguns layouts da aplicação BATINA . . . . .	12
2.5	Página inicial da plataforma <i>BigBlueButton</i> [3] . . . . .	13
2.6	Relatório de progressos fornecido aos alunos pela <i>ALEKS</i> . Adaptado de [4] . . . . .	14
2.7	Página inicial da aplicação <i>Socrative</i> para professores. Adaptado de [5] . . . . .	15
2.8	Arquitetura da aplicação <i>StudentLife</i> . Adaptado de [6] . . . . .	16
3.1	Fluxograma que representa os passos necessários desde o pedido de criação de formulário à biblioteca até à sua criação. . . . .	20
3.2	Fluxograma do Controlo de Questões. . . . .	20
3.3	<i>TitleControls</i> - Código <i>JSON</i> (esquerda) e representação na aplicação (direita). . . . .	21
3.4	<i>DescriptionControl</i> - Código <i>JSON</i> (esquerda) e representação na aplicação (direita). . . . .	21
3.5	<i>DateControl</i> - Código <i>JSON</i> (esquerda) e representação na aplicação (direita). . . . .	22
3.6	<i>SliderControl</i> - Código <i>JSON</i> (esquerda) e representação na aplicação (direita). . . . .	23
3.7	<i>TextInputControl</i> - Código <i>JSON</i> (esquerda) e representação na aplicação (direita). . . . .	23
3.8	<i>ConditionalPickerControl</i> - Código <i>JSON</i> (esquerda) e representação na aplicação (direita). . . . .	24

## Lista de Figuras

---

3.9	<i>CheckBoxControl</i> - Código <i>JSON</i> (esquerda) e representação na aplicação (direita). . . . .	24
3.10	<i>PickerHorizontalImagesControl</i> - Código <i>JSON</i> (esquerda) e representação na aplicação (direita). . . . .	25
3.11	<i>HorizontalImagesControl</i> - Código <i>JSON</i> (esquerda) e representação na aplicação (direita). . . . .	26
3.12	<i>RadioButtonControl</i> - Código <i>JSON</i> (esquerda) e representação na aplicação (direita). . . . .	26
4.1	Arquitetura da aplicação ISABELA para recolha e processamento de dados. Adaptado de [7] . . . . .	31
4.2	Nuvem ISABELA. Adaptado de [7] . . . . .	32
4.3	Arquitetura da aplicação móvel ISABELA. . . . .	34
4.4	Formulário de proximidade (esquerda). Tipo de teclado utilizado no formulário de proximidade (direita). . . . .	35
4.5	Formulário de Sono . . . . .	36
4.6	Formulário <i>POMS</i> da aplicação ISABELA. . . . .	37
4.7	Formulário emocional com imagens. . . . .	38
4.8	Formulário de transporte . . . . .	39
4.9	Algumas questões do formulário <i>HEXACO</i> . . . . .	40
4.10	Página informativa da personalidade do aluno universitário. . . . .	41
4.11	Formulário de finalidade da aplicação (esquerda). Opções da finalidade da aplicação (direita) . . . . .	42
5.1	Página inicial do MAORI . . . . .	44
5.2	Página de treino de voo enquanto não se efetua um treino (esquerda). Página de treino de voo quando se encontra em treino (direita). . . . .	45
5.3	Formulário <i>POMS</i> da aplicação MAORI. . . . .	46
6.1	Consumo de bateria com os formulários antigos (esquerda). Consumo de bateria com os formulários a usar a biblioteca <i>Socialite-Forms</i> (direita). . . . .	50

# Lista de Tabelas

4.1	Requisitos Não Funcionais da Aplicação ISABELA . . . . .	29
4.2	Requisitos Funcionais da Aplicação ISABELA . . . . .	30
4.3	Questões do formulário de sono e respetiva escala de avaliação . . .	36
4.4	Opções do primeiro <i>picker</i> do formulário de transporte e opções condicionadas do segundo <i>picker</i> . . . . .	39
6.1	Impacto da biblioteca <i>SocialiteForms</i> nas linhas de código utilizadas na criação de formulários da linguagem de marcação <i>XAML</i> . . .	48
6.2	Impacto da biblioteca <i>SocialiteForms</i> nas linhas de código utilizadas na criação de formulários da linguagem de programação <i>C#</i> . .	49

# Lista de Acrónimos

**API** Application Programming Interface

**AI** Artificial Intelligence

**CPS** Cyber-Physical System

**FI-PPP** Future Internet Public-Private Partnership

**GE** Generic Enabler

**HiTL** Human-in-the-loop

**HitLCPS** Human-in-the-Loop Cyber-Physical System

**IoT** Internet of Things

**LMS** Learning Management Systems

**SO** Sistema Operativo

**SDKs** Software Development Kits

**UI** User Interface

# 1

## Introdução

### Conteúdo

---

1.1	Contextualização . . . . .	2
1.2	Motivação . . . . .	2
1.3	Objetivos . . . . .	3
1.4	Metodologia . . . . .	3
1.5	Estrutura da Dissertação . . . . .	4

---

### 1.1 Contextualização

A Internet das Coisas ou Internet of Things (IoT) refere-se à rede abrangente criada por milhares de milhões de dispositivos que têm a capacidade de comunicar entre si através da *Internet*. Estes dispositivos podem variar desde *smartphones* até dispositivos industriais, como máquinas inteligentes, com capacidade de recolher, partilhar e analisar informação. Estima-se que até 2025 existam cerca de 75 mil milhões de dispositivos de IoT conectados em todo o mundo, aproximadamente o triplo dos instalados em 2019 [8].

Nos últimos anos houve um avanço significativo nas áreas de IoT, robótica e sensores sem fio. O crescimento destas áreas permitiu o desenvolvimento de sistemas ciberfísicos inteligentes ou Cyber-Physical System (CPS), com capacidade de monitorizar e controlar fenómenos físicos [9]. O aprimoramento dos CPS possibilitou o aparecimento de sistemas que consideram o Ser Humano como um elemento integrante do sistema. Através do uso de sensores e *smartphones* é possível recolher informações sobre o Ser Humano que podem ser usadas para criar CPSs que integrem o Ser Humano como uma parte do sistema, ou seja um sistema *Human-in-The-Loop Cyber-Physical-System* (HiTLCPS) [10].

### 1.2 Motivação

Um dos fatores que preocupa os estabelecimentos do ensino superior é o abandono escolar de estudantes universitários. Tal, pode ser motivado por diversas razões, entre as quais se destacam as dificuldades económicas, a não entrada no curso pretendido inicialmente, o insucesso académico e o nível de empregabilidade [11]. Em particular, o insucesso académico pode ser determinado pelo *stress*, pela dificuldade na gestão de tempo e falta de auto-disciplina. O abandono escolar associado ao insucesso académico pode ser minimizado através do acompanhamento individual de cada um dos alunos universitários. Porém, para o acompanhamento ser eficaz, requer um elevado número de recursos, sendo por isso um processo dispendioso.

Atualmente, a utilização de *smartphones* por estudantes universitários é de cerca de 91% [12]. Dada a elevada taxa de utilização seria possível efetuar o acompanhamento individual através de um tutor virtual baseado numa aplicação para *smartphones*. A aplicação beneficiaria do avanço de tecnologias, tais como, novas técnicas de Artificial Intelligence (AI), o paradigma de IoT, ainda de sensores embutidos nos *smartphones* e *chatbots*. Estas tecnologias possibilitam a recolha de dados físicos, comportamentais e sociais relativos a cada um dos estudantes universitários, o que permite criar um sistema Human-in-the-Loop Cyber-Physical Sys-

tem (HitLCPS) capaz de monitorizar e aconselhar os estudantes. Este sistema, por sua vez, iria simplificar o acompanhamento individual, sem aumentar os custos associados e sem sobrecarregar professores e profissionais especializados.

### 1.3 Objetivos

Inicialmente a aplicação ISABELA foi desenvolvida apenas para dispositivos com o sistema operativo *Android*. Porém, quando se quer criar uma aplicação para *smartphones* é necessário considerar diferentes tipos de sistemas operativos, pois estes utilizam diferentes tipos de tecnologias na criação de aplicações, uma vez que cerca de 99% dos *smartphones* utilizam ou o Sistema Operativo (SO) *Android* ou o *iOS* [13]. Tendo isto em consideração, será utilizada uma *cross-plataform*, o *Xamarin* que utiliza a linguagem de programação *C#* e a linguagem de marcação *XAML*. O *Xamarin* permite o desenvolvimento de aplicações para o SO *Android* e *iOS*, visto que permite que cerca de 90% do código possa ser partilhado. Na criação de uma aplicação também tem de se ter em consideração a reusabilidade do código, desta forma a criação de bibliotecas e reutilização de código traz benefícios significativos, como o aumento de eficiência, a redução de erros, a padronização e a facilidade de manutenção.

Nesta dissertação propõe-se o desenvolvimento e reconstrução de uma plataforma inovadora baseada em novas tecnologias, tais como IoT, AI, um *chatbot* e componentes da *FIWARE*. A plataforma foi denominada ISABELA, *IoT Student Advisor and Best Lifestyle Analyzer* e tem como foco a educação e melhoria do desempenho académico. Com esse propósito, utilizando *smartphones* e sensores neles embutidos, um *chatbot* e formulários apresentados ao aluno, será possível monitorizar e apoiar o aluno na sua rotina diária, visando aprimorar o seu desempenho académico, através de um sistema HitLCPS. Também foi criada uma nova biblioteca cujo objetivo é simplificar a criação e alteração dos formulários da aplicação ISABELA ou de novos formulários que venham a ser inseridos nesta aplicação ou em outras aplicações. Por fim, também será criada uma nova aplicação, chamada MAORI. Esta aplicação é uma adaptação da aplicação ISABELA e será utilizada pela Força Aérea Portuguesa numa das fases de recrutamento.

### 1.4 Metodologia

A plataforma ISABELA é um dos projetos em desenvolvimento que envolve um grupo de trabalho orientado pelo professor Doutor Jorge Sá Silva e pelo professor Doutor André Rodrigues. O grupo é constituído por alunos de mestrado e doutoramento de engenharia eletrotécnica e de engenharia informática.

## 1. Introdução

---

O grupo de trabalho encontra-se dividido em várias equipas, cada uma envolvida num projeto em particular. Dos projetos em desenvolvimento destacam-se quatro aplicações cujo foco principal é a monitorização de um grupo específico de pessoas. A aplicação ISABELA visa monitorizar e auxiliar estudantes universitários a melhorarem o seu desempenho académico; a aplicação MAORI tem como objetivo monitorizar os candidatos à Força Aérea Portuguesa durante uma das fases de recrutamento; a aplicação MYPOC que monitoriza doentes diagnosticados com perturbação obsessivo-compulsiva e, por fim, a aplicação IFRIEND que monitoriza o estado de saúde da população idosa. Todas as plataformas mencionadas anteriormente utilizam bibliotecas, uma para recolha de dados dos sensores embutidos nos *smartphones* e outra para a comunicação com a *FIWARE* (será explicada na secção 2.1.5). De igual forma, todas as plataformas foram desenvolvidas com recurso à *cross-plataform XAMARIN* (será explicada na secção 2.1.4).

O presente trabalho encontra-se inserido no projeto da aplicação ISABELA e tem como propósito refazer a aplicação previamente desenvolvida para o SO *Android*, para que esta seja compatível com ambos os SOs *Android* e *iOS*. Neste trabalho é ainda apresentada a criação de uma nova biblioteca dinâmica, cujo objetivo é simplificar a construção da parte visual dos formulários que aparecem na plataforma, bem como a adaptação desta plataforma, para uma nova plataforma MAORI.

O Afonso Lemos, aluno de mestrado, está designado pelo desenvolvimento de um *chatbot*. A Filipa Moreira, aluna de mestrado, encontra-se a desenvolver novos formulários para a plataforma. O Jorge Rivadeneira, aluno de doutoramento, é responsável pela segurança e privacidade. Por fim, o Guilherme Borges, aluno de doutoramento, é responsável pelo planeamento da aplicação e ajuda nas tarefas anteriormente referidas.

O projeto foi delineado adotando o modelo *SCRUM*, o qual envolve a realização de uma reunião semanal, onde é definido o *sprint* a realizar por cada elemento da equipa.

### 1.5 Estrutura da Dissertação

Este documento encontra-se estruturado em sete capítulos.

No Capítulo 2 é apresentado o estado de arte, onde é realizada uma breve introdução das tecnologias e conceitos necessários e apresentadas algumas plataformas semelhantes. Posteriormente, é feita uma análise comparativa das plataformas semelhantes com a plataforma ISABELA.

De seguida, no Capítulo 3, é detalhado o trabalho realizado na criação da nova biblioteca *SocialiteForms*.

No Capítulo 4, é feita uma breve contextualização e descrição da aplicação ISABELA. Seguidamente, serão referidos os requisitos necessários e a arquitetura da aplicação ISABELA. Por fim, será feita uma análise aos formulários da aplicação ISABELA quando utilizada a biblioteca *SocialiteForms*.

No Capítulo 5 é detalhado o trabalho realizado na criação de uma nova aplicação denominada de MAORI.

No Capítulo 6 são apresentados os testes e os resultados obtidos.

Por último, no Capítulo 7, são apresentadas as conclusões e identificado o trabalho futuro.

# 2

## Tecnologias Base e Estado da Arte

### Conteúdo

---

2.1	Tecnologias Base . . . . .	7
2.2	Conceitos . . . . .	9
2.3	Plataformas atuais -ISABELA/BATINA e UCFrameworks .	10
2.4	Plataformas Semelhantes . . . . .	13
2.5	Análise Comparativa . . . . .	16

---

## 2.1 Tecnologias Base

Nesta secção serão abordadas resumidamente as tecnologias fundamentais para o desenvolvimento deste projeto de dissertação: *Android*, *iOS*, *.NET*, *Xamarin*, *FIWARE*.

### 2.1.1 Android

O *Android* é um SO *open-source* desenvolvido pela *Google* [14]. O facto de ser *open-source* contribuiu para a melhoria contínua do SO. Tal possibilitou aos fabricantes de telemóveis a adaptação e utilização nos seus dispositivos móveis. Por estas razões o SO *Android* é o mais utilizado a nível mundial [14].

A arquitetura do SO *Android* é uma pilha de *software* baseada em *Linux* constituída por *Kernel* do *Linux*; camada de abstração de *hardware* (*HAL*); *Android Runtime*; bibliotecas *C/C++* nativas; estrutura da *Java API* e aplicações do sistema [15]. Para desenvolver aplicações para o SO *Android* é utilizada geralmente a linguagem de programação *Java* e Software Development Kits (SDKs).

### 2.1.2 iOS

O *iOS* é um SO desenvolvido pela *Apple* e baseado no *Mac OS*. O SO surgiu em 2007 e continua atualmente a ser utilizado em *iPhones*, *iPads* e outros dispositivos *Apple*. O *iOS* é o segundo SO mais utilizado mundialmente [16].

A arquitetura do *iOS* é baseada nas camadas *Core OS*, serviços *Core*, camada *media* e *Cocoa Touch* (camada da aplicação) [17]. As aplicações *iOS* são desenvolvidas com o recurso a SDKs de *iOS*. Estas são ferramentas que permitem desenvolver, instalar, executar e testar aplicações [18].

### 2.1.3 .NET

O *.NET* é uma plataforma de desenvolvimento *open-source* criada pela *Microsoft* de forma a criar diferentes tipos de aplicações. A plataforma suporta diferentes tipos de linguagens de programação e bibliotecas para criar aplicações *web*, móveis, de IoT, entre outras. O *C#* é considerado a principal linguagem de programação de *.NET*, no entanto, existem outras linguagens igualmente suportadas, como o *F#* e *Visual Basic*. A arquitetura *.Net* é baseada nos seguintes componentes: *CoreCLR* e *CoreFX* [19].

### 2.1.4 XAMARIN

O *Xamarin* é uma plataforma *open-source* que utiliza *.NET* para criar aplicações modernas e de alto desempenho para *iOS* e *Android*. O *Xamarin* permite a partilha

## 2. Tecnologias Base e Estado da Arte

---

de cerca de 90% do código entre plataformas, o que possibilita a reutilização do código existente e a criação da parte lógica em apenas uma linguagem. Assim, o desempenho, a aparência e o comportamento nativos são mantidos em cada uma das plataformas [1]. Na Figura 2.1 pode-se observar a arquitetura geral de uma aplicação em *Xamarin*.

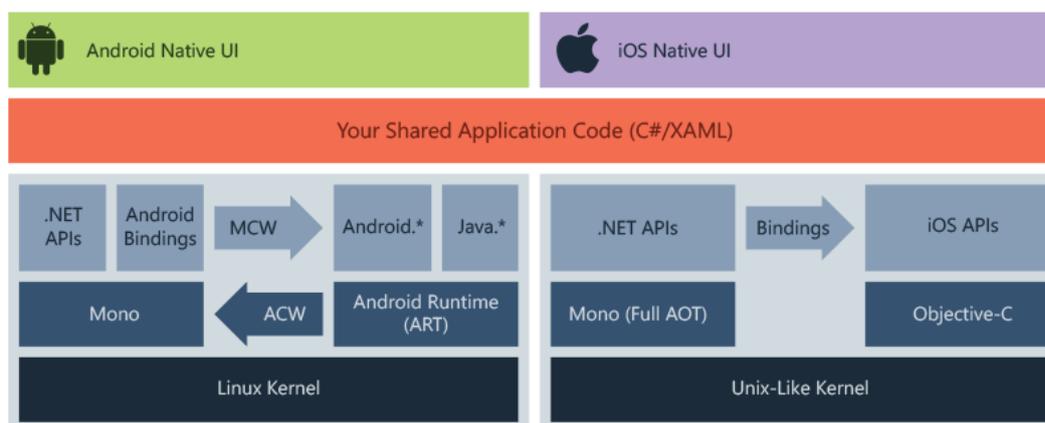


Figura 2.1: Arquitetura de uma aplicação em *Xamarin*. Adaptado de [1]

Uma vez que o *Xamarin* não contempla a partilha da interface do utilizador, surgiu uma User Interface (UI) *framework* - *Xamarin.Forms* que, para além de conter a partilha de código e desenvolvimento da parte lógica, também permite a criação das interfaces de utilizador através de *XAML* e *C#* [20]. Este UI *framework* sofreu uma atualização e foi expandido para as plataformas (*macOS* e *Windows*). O novo UI *framework* foi chamado de *.NET Multi-platform APP UI* (*.NET MAUI*) e veio possibilitar uma maior flexibilidade ao ecossistema *XAMARIN*. Algumas novidades da *cross-platform MAUI* são: a estrutura de projeto única, os controlos nativos para melhorar o desempenho, a multiplataforma Application Programming Interface (API) que permite aceder aos recursos do dispositivo nativo e a *.NET hot reload* que permite observar as alterações efetuadas no código sem que seja necessário fazer *debug* novamente.

### 2.1.5 FIWARE

A *FIWARE* é uma plataforma *open-source* financiada pela Future Internet Public-Private Partnership (FI-PPP) [21], que tem como principal objetivo facilitar a implementação de projetos da IoT. Esta plataforma permite o desenvolvimento de novas aplicações inteligentes de forma rápida, fácil e acessível, com a garantia de modularidade, proteção de dados e máxima interoperabilidade [22].

A *FIWARE* é constituída por uma biblioteca de componentes denominados por Generic Enabler (GE), que facilitam a implementação de APIs. Os GEs permitem a

implementação das seguintes funções: armazenamento numa *cloud*; gestão, análise, processamento e visualização de dados; segurança; interfaces com dispositivos IoT, redes e robôs .

## 2.2 Conceitos

Nesta secção serão abordados os conceitos necessários para o desenvolvimento deste projeto de dissertação: IoT, CPS, HitLCPS.

### 2.2.1 *Internet of Things*

Devido ao rápido avanço nas tecnologias, caminhamos para um mundo em que tudo vai estar conectado e a comunicar entre si. Assim, surgiu a IoT, uma tecnologia que permite a comunicação entre todos e a qualquer momento. O principal objetivo da IoT é criar uma conexão independente e autónoma, que seja segura e permita a troca de informação entre o mundo real, objetos físicos e aplicações [23].

Neste momento existem mais de 13 mil milhões de dispositivos de IoT pelo mundo. Prevê-se que em 2030 este número aumente para 29.4 mil milhões como consequência do aumento do consumo de *Internet* e utilização de dispositivos, tais como computadores, *smartphones* e *smart devices*.

### 2.2.2 *Cyber-Physical System*

O CPS é constituído por vários sistemas interconectados, principalmente sensores e atuadores. Estes podem ser facilmente relacionados com sistemas de IoT, no entanto um CPS foca-se essencialmente na interação entre processos físicos, de computação e de rede [24].

Os CPSs permitem que aplicações inteligentes e serviços operem de forma correta e em tempo-real. Estes são a base para a integração entre sistemas ciberfísicos e físicos, que trocam diversos tipos de dados e informações sensíveis em tempo-real. Assim, os CPSs são capazes de monitorizar, manipular processos e objetos reais, adaptar e controlar sistemas do mundo físico. Isto é, com recurso à computação em tempo real, têm maior capacidade e flexibilidade para alterar o tempo dos processos de um sistema [24].

Embora os sistemas CPSs tenham diversas vantagens, não deixam de estar sujeitos a ameaças e ataques, tanto ciberfísicos como físicos. Os ataques podem provocar efeitos catastróficos levando à necessidade de implementação de medidas robustas de segurança.

### 2.2.3 *Human-in-the-Loop Cyber-Physical System*

Embora na maioria das vezes os CPS sejam aplicações centradas no Ser Humano, estes não o consideram como um elemento ativo do sistema. De forma a criar mecanismos mais fiáveis e aptos para monitorização e controlo, surgiram os HitLCPS, que consideram o fator humano como parte integrante dos CPS.

Na Figura 2.2 está caracterizado um processo de controlo de um sistema Human-in-the-loop (HiTL) que é constituído por quatro fases principais: aquisição de dados, inferência do estado, futura inferência do estado e atuação [2]. A aquisição de dados consiste na recolha dos dados através de sensores. Este processo está cada vez mais simples, visto que, a capacidade sensorial dos *smartphones* têm vindo a aumentar. A inferência do estado consiste no processamento de dados previamente recolhidos, de modo a inferir o estado do utilizador. Para além de uma inferência de estado inicial alguns sistemas também integram uma futura inferência do estado. Esta tenta prever estados de utilizador futuros com recurso a dados históricos e o estado atual do utilizador. Por fim, na fase de atuação, o sistema verifica se existe alguma ação que pode ser efetuada.

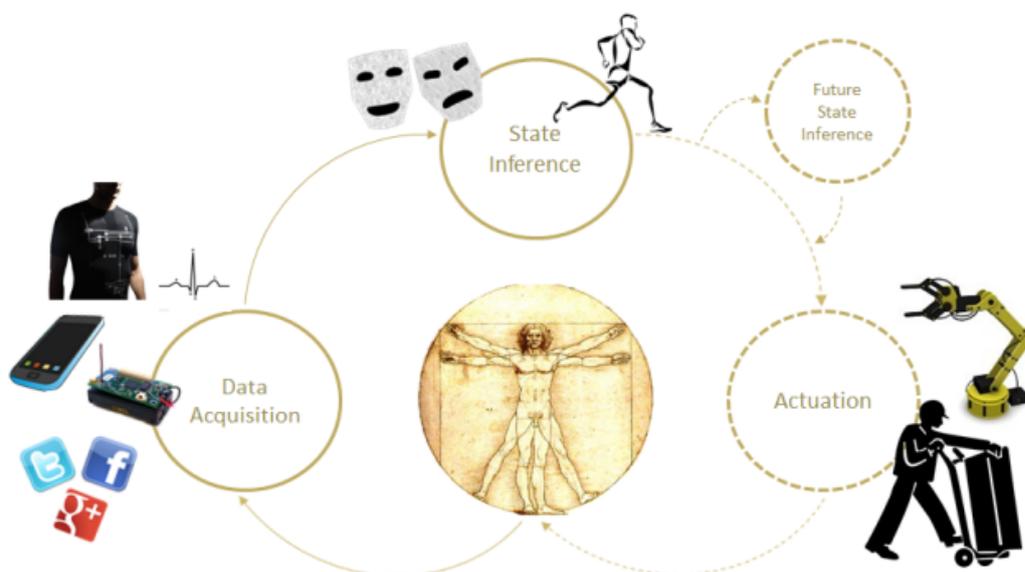


Figura 2.2: Processo de um sistema HiTL. Adaptado de [2]

## 2.3 Plataformas atuais -ISABELA/BATINA e UC-Frameworks

A plataforma ISABELA - *IoT Student Advisor e BEst Lifestyle Analyser* foi desenvolvida por um grupo de investigação da Faculdade de Ciências e Tecnologia da Universidade de Coimbra (FCTUC) em colaboração com outras faculdades. Esta

## 2.3 Plataformas atuais -ISABELA/BATINA e UCFrameworks

plataforma apenas é compatível com o SO Android e tem como principal objetivo melhorar o desempenho acadêmico dos estudantes universitários. A plataforma foi desenvolvida com base num sistema HitLCPS.

Dado que, a plataforma ISABELA é um sistema HitLCPS, esta tem a capacidade de compreender e processar os dados. A recolha de dados é uma etapa fundamental para que a plataforma ISABELA seja capaz de os processar. Esta recolhe diferentes tipos de dados dos utilizadores, alguns de forma mais indireta, isto é, sem a necessidade de ações do utilizador. A recolha desses dados realiza-se através de sensores que se encontram embutidos nos *smartphones*, nomeadamente giroscópio, acelerómetro, *Wifi*, sensor de proximidade, sensor de luz, entre outros. Além disso, também é possível obter dados através de redes sociais como o *Facebook* e o *Twitter*.

Para além da recolha de dados de forma indireta, ainda são disponibilizados formulários na aplicação para recolher dados acerca do contexto físico, da qualidade do estudo, do contexto social e da qualidade do sono. Estes devem ser respondidos diariamente ou quando solicitado pela aplicação, por meio de notificações [25]. A aplicação de forma a auxiliar o utilizador, é constituída também por um *chatbot* que permite recolher dados, através da análise do diálogo e também tem a capacidade de aconselhar o utilizador. Na figura 2.3 estão representados alguns *layouts* da aplicação ISABELA.

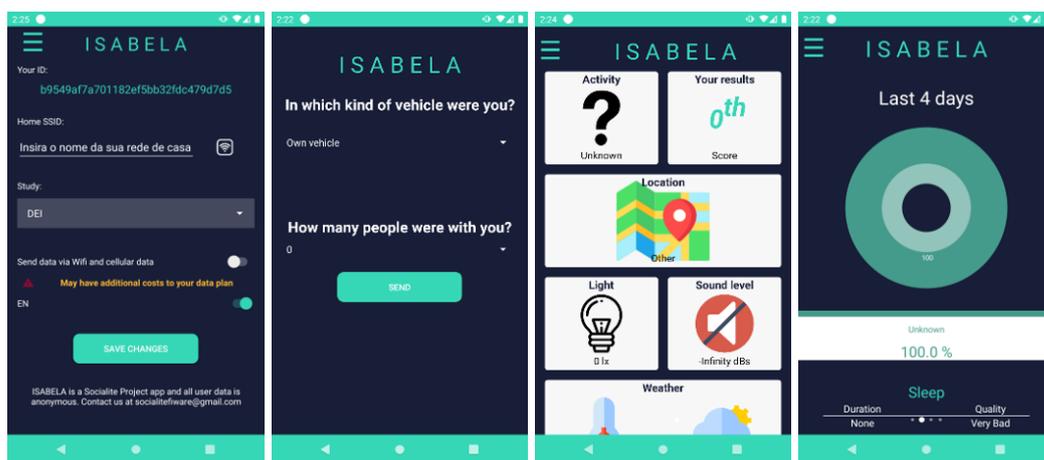


Figura 2.3: Alguns *layouts* da aplicação ISABELA

Posteriormente, desenvolveu-se a aplicação BATINA, uma versão reduzida do ISABELA e centrada apenas na sala de aula. A plataforma BATINA tem como principal objetivo, facilitar a iteração entre o professor e o aluno.

Esta plataforma permite ao professor criar e disponibilizar cursos, aulas e questionários. Os professores conseguem criar ou alterar os cursos a qualquer momento

## 2. Tecnologias Base e Estado da Arte

através da *Dashboard*. Os alunos têm acesso a todos os materiais dos seus cursos. Os professores conseguem com relativa facilidade criar questionários e ainda têm a possibilidade de escolher a data em que serão disponibilizados ao aluno. Os alunos serão notificados que foi disponibilizado um novo formulário.

Para além disso o aluno tem acesso a um *chatbot* e pode tirar dúvidas com o professor sobre um determinado tópico na aplicação. O professor é notificado por *e-mail* acerca de novas dúvidas, às quais pode responder no *dashboard*.

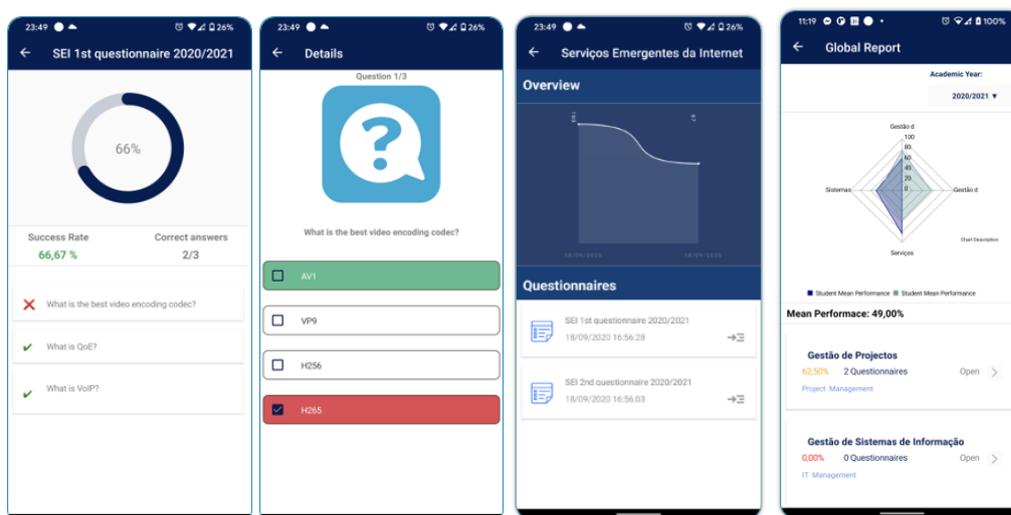


Figura 2.4: Alguns layouts da aplicação BATINA

O projeto BATINA tem sido financiado pela Reitoria como uma plataforma de testes e de novas funcionalidades. Estas funcionalidades poderão ser integradas em tecnologias de produção da *UC Framework*, tais como o *UC TEACHER* e o *UC STUDENT*. Mais tarde, quando a Reitoria se candidatou aos fundos do PRR [26], o projeto BATINA foi inserido no projeto *ON-BOARD*. O projeto *ON-BOARD* tem como principal objetivo combater os fenómenos de insucesso escolar e do abandono do Ensino Superior [27]. Este projeto é constituído por sete atividades, tais como processos de tutoria, recolha de dados para identificar padrões e também tem em conta o bem-estar e a saúde mental de alunos e professores. O nosso grupo de trabalho integra este projeto através do desenvolvimento de um tutor virtual.

A *UC Framework* é o departamento tecnológico da *UC NEXT* [28], uma empresa da Universidade de Coimbra, que tem como objetivo criar soluções para apoiar a estratégia de transformação digital da Universidade. A *UC Framework* desenvolve tecnologias próprias que permitem desenvolver e disponibilizar plataformas cada vez mais completas, seguras e únicas. Algumas das plataformas desenvolvidas por este departamento são: *UC TEACHER*; *UC STUDENT*; *UC MEE-*

*TINGS e UC PAGES* [29].

## 2.4 Plataformas Semelhantes

Nesta secção será feita uma análise de plataformas semelhantes à plataforma ISABELA/BATINA. Assim, será efetuada uma análise de plataformas que têm como propósito a melhoria da qualidade do ensino.

### 2.4.1 *BigBlueButton*

O *BigBlueButton* é um *software open-source* desenvolvido em 2007 pelo Instituto de Empreendedorismo e Comercialização de Tecnologia da Universidade de Carleton em Ottawa, no âmbito do projeto *Technology Innovation Management* (TIM) [3].

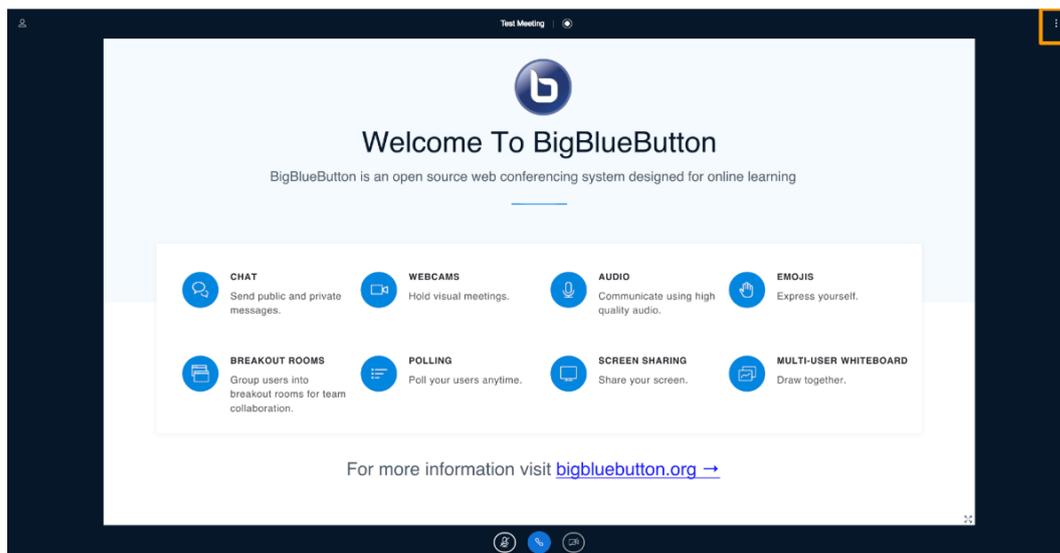


Figura 2.5: Página inicial da plataforma *BigBlueButton* [3]

A plataforma *BigBlueButton* foi criada com o intuito de melhorar o ensino *online*, sendo constituída por mecanismos de Learning Management Systems (LMS). Esta, disponibiliza uma série de recursos síncronos, que permitem uma conexão entre professores e alunos, tais como *chats*; conferências *web* com partilha de áudio e vídeo; partilha de ecrã; partilha de notas; diferentes salas; entre outros [3]. Na Figura 2.5 pode observar-se algumas destas ferramentas na página inicial da aplicação.

Neste momento o *BigBlueButton* encontra-se integrado em vários mecanismos de aprendizagem *online*, tais como *Atutor*; *Canvas*; *Chamilo*; *Docebo*; *Fedena*; entre outras [3].

### 2.4.2 ALEKS

*ALEKS* é uma plataforma constituída por módulos de inteligência artificial, que consegue avaliar o conhecimento de um aluno acerca de um certo tópico relacionado com matemática, química, estatística e contabilidade [4].

Esta plataforma com mais de duas décadas de existência conta com biliões de dados recolhidos que são utilizados para treinar os módulos de *Machine Learning*, de forma a guiar os alunos ao longo da sua aprendizagem. A avaliação é individual e guia o utilizador a aprender os tópicos que deseja, através de estratégias dinâmicas que garantem o sucesso académico. O progresso depende de um questionário feito ao aluno. Consoante o resultado do questionário, a plataforma verifica se os conceitos estão bem consolidados e se o utilizador está apto para avançar [4].

A *ALEKS* fornece relatórios aos alunos sobre o seu progresso, como se pode verificar na Figura 2.6. Os relatórios também podem ser acedidos pelos professores, que podem verificar o grau de progresso dos alunos e fazer assim uma melhor gestão do tempo das aulas.

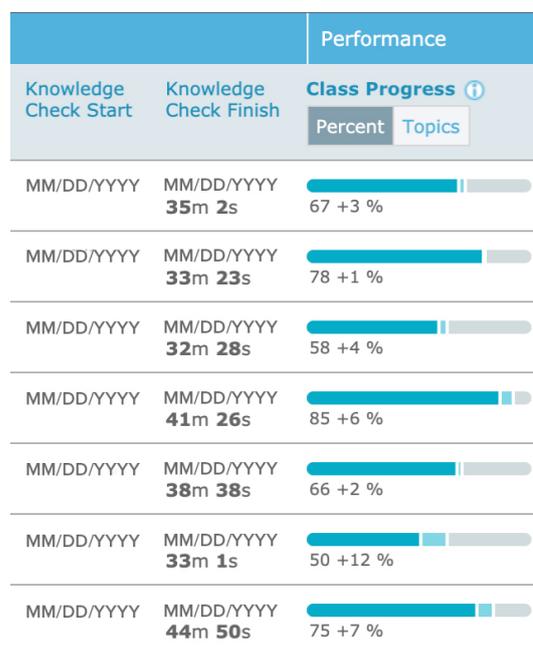


Figura 2.6: Relatório de progressos fornecido aos alunos pela *ALEKS*. Adaptado de [4]

### 2.4.3 Socrative

A *Socrative* é uma aplicação gratuita, fácil de usar e compreender. Esta aplicação encontra-se disponível para quase todas as plataformas e dispositivos como *smartphones*, *tablets*, *laptops* e computadores. A aplicação tem como fi-

nalidade fornecer aos professores ferramentas eficazes para avaliar e acompanhar o progresso dos alunos, das quais se destacam questionários, relatórios, salas virtuais, competições (*Space Race*), resultados e a experiência dos estudantes [5].

Esta plataforma também possibilita que se criem questões rápidas de verdadeiros e falsos; escolhas múltiplas e respostas curtas. A plataforma apresenta dois tipos de *login* que permitem o acesso distinto a alunos e professores. O *login* do professor permite aceder ao menu inicial como se pode observar na Figura 2.7. Por sua vez o *login* do estudante permite que este aceda através de códigos aos diversos tipos de questionários [5].

O *feedback* dos questionários pode ser consultado por alunos e professores em qualquer momento. Tal permite aos professores monitorizar o progresso e perceber as possíveis dificuldades dos alunos. Por outro lado, os alunos poderão perceber em que partes tem maior dificuldade e focar o seu estudo nessas áreas de forma a melhorar o seu desempenho académico.

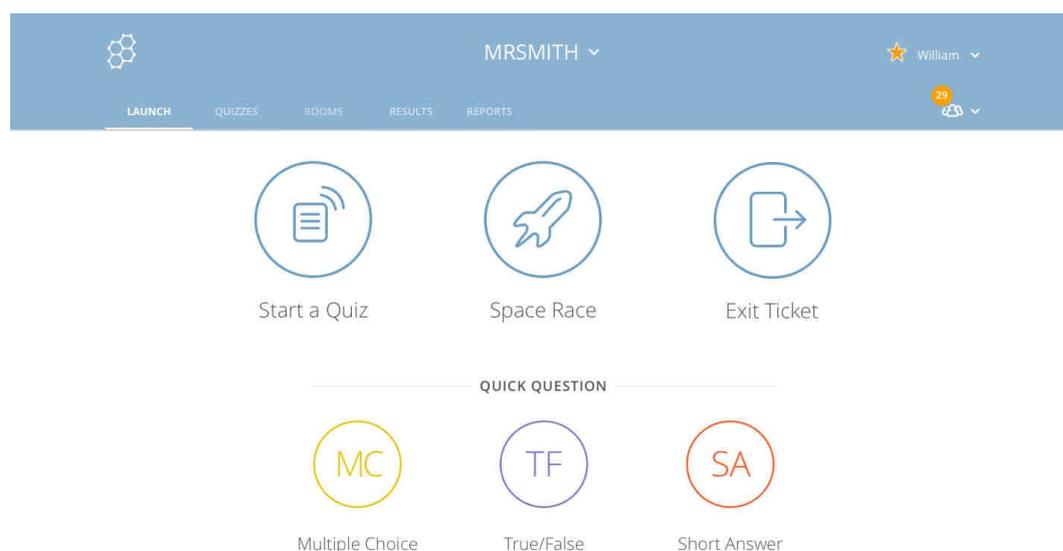


Figura 2.7: Página inicial da aplicação *Socrative* para professores. Adaptado de [5]

### 2.4.4 StudentLife

*StudentLife* foi o primeiro estudo realizado por uma equipa de investigação do *Dartmouth College*, que tinha como propósito avaliar a saúde mental, o desempenho académico e os hábitos de 48 estudantes durante 10 semanas, com o recurso a

## 2. Tecnologias Base e Estado da Arte

dados recolhidos por telemóveis. Desta forma, foi criada uma aplicação para o SO *Android*, que obtinha dados de sensores embutidos nos telemóveis de forma passiva sobre o dia a dia do estudante [30].

Através da arquitetura do *StudentLife*, representada na Figura 2.8, pode verificar-se que para além de serem recolhidos dados através dos sensores, também são obtidos dados do estudante através de formulários na aplicação. Após análise dos dados recolhidos foi possível obter resultados referentes ao contexto comportamental, emocional, social e ainda à qualidade do sono [6].

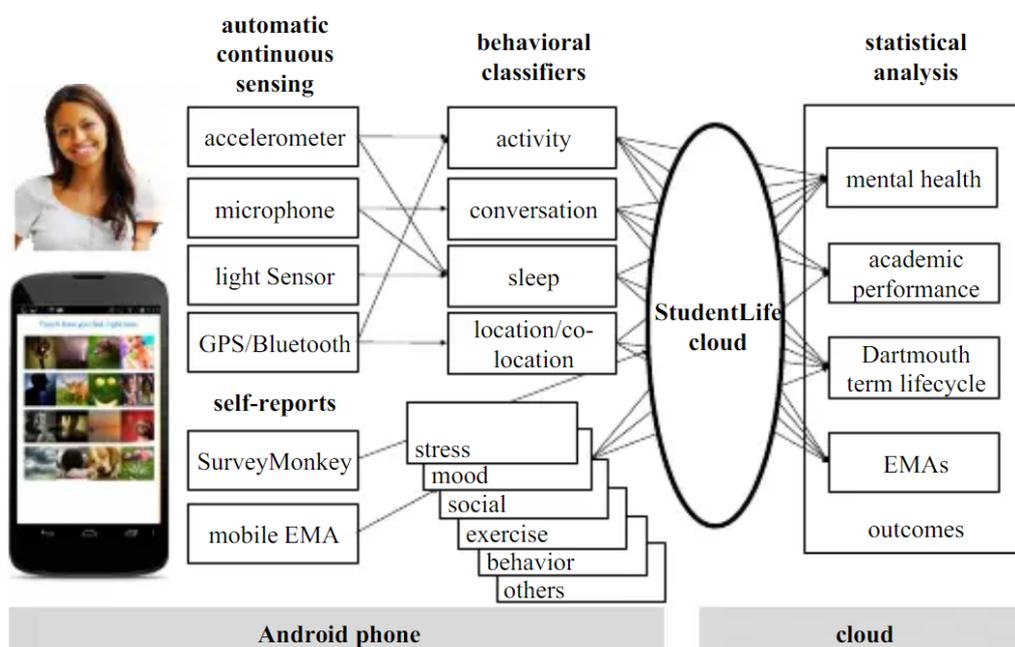


Figura 2.8: Arquitetura da aplicação *StudentLife*. Adaptado de [6]

## 2.5 Análise Comparativa

Ao comparar a plataforma ISABELA/BATINA com as mencionadas anteriormente, pode-se perceber que a mesma foi desenvolvida com base no estudo *StudentLife*. A plataforma ISABELA/BATINA, tal como as outras plataformas, utiliza sensores embutidos em *smartphones* para obter informações sobre o contexto social, emocional e físico dos utilizadores. Porém, esta plataforma tem a capacidade de auxiliar e acompanhar os estudantes, de forma a melhorar seu desempenho académico. Além disso, inclui um *chatbot* que permite aumentar a interação entre o estudante e a plataforma.

A plataforma BATINA permite a disponibilização de formulários e *quizzes*, tal e qual como na plataforma *Socrative*. A plataforma ISABELA, apenas disponibiliza

alguns formulários ao utilizador de forma a recolher dados.

De forma a facilitar o processamento de dados e a monitorização dos estudantes, a plataforma ISABELA/BATINA utiliza mecanismos de *machine learning*, tal como a aplicação *ALeks*. De momento, a plataforma ainda não integra nenhum mecanismo de LMS, como o apresentado na aplicação *BigBlueButton*, no entanto esse mecanismo poderá ser integrado futuramente.

Em suma, a plataforma ISABELA/BATINA integra diversas funcionalidades de aplicações semelhantes, tais como a disponibilização de *quizzes* e formulários, a recolha de dados através de sensores embutidos nos *smartphones*, o processamento dos dados e o uso de mecanismos de AI. A utilização de um sistema HitLCPS e de um *chatbot*, permitem que a aplicação apresente *feedback* ao utilizador, o que torna esta plataforma mais eficaz.

# 3

## Biblioteca SocialiteForms

### Conteúdo

---

3.1	Contextualização . . . . .	19
3.2	Estrutura da biblioteca . . . . .	19
3.3	Características do ficheiro descritivo em <i>JSON</i> . . . . .	20

---

### 3.1 Contextualização

A biblioteca descrita neste relatório tem como objetivo a criação fácil e sistemática de formulários com recurso apenas a um documento descritivo. O objetivo da criação da biblioteca de formulários dinâmicos surgiu pela necessidade de substituir o processo da criação de formulários adicionando as *views* individualmente por um processo mais simples, podendo o formulário ser alterado quando for necessário.

Assim sendo, a biblioteca é constituída por diversos tipos de controlos. Cada controlo representa um tipo de característica que é adicionada aos formulários. A biblioteca apenas necessita de um documento descritivo em *JSON*; de adicionar os recursos necessários à aplicação e de chamar a função de criação de formulários.

### 3.2 Estrutura da biblioteca

A estrutura da biblioteca pode ser representada através de dois fluxogramas, o fluxograma que indica todos os passos sequenciais pelos quais o formulário tem de passar, desde o pedido da criação até à sua conclusão (fluxograma da figura 3.1) e o fluxograma de controlo de questões, que demonstra a forma como a biblioteca adiciona as questões ao *layout* (fluxograma da figura 3.2).

No fluxograma da figura 3.1 consegue observar-se que inicialmente existe uma escolha do tipo de *layout* que será inserido (um *StackLayout* ou um *CarouselView*). Após a escolha, começa a inserir-se o conteúdo no *layout*, no caso de se achar necessário será adicionado um título e uma descrição. Em seguida, serão adicionadas uma a uma, todas as questões que o formulário irá conter. Por fim, antes de retornar o *layout* a biblioteca ainda irá inserir um botão que possibilita o acesso a todas as respostas das questões adicionadas.

No fluxograma da figura 3.2 consegue observar-se que a biblioteca verifica sequencialmente qual o tipo de questão e adiciona o controlo respetivo, seguindo para a próxima questão. Cada um dos controlos terá tipos de parâmetros específicos, que serão explicados no próxima secção. No caso de não encontrar o tipo de questão a ser adicionada, segue para a próxima questão, até não haver mais questões a adicionar.

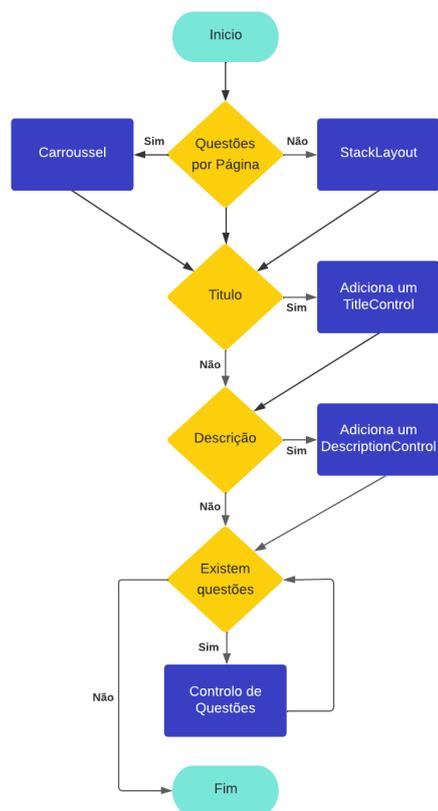


Figura 3.1: Fluxograma que representa os passos necessários desde o pedido de criação de formulário à biblioteca até à sua criação.

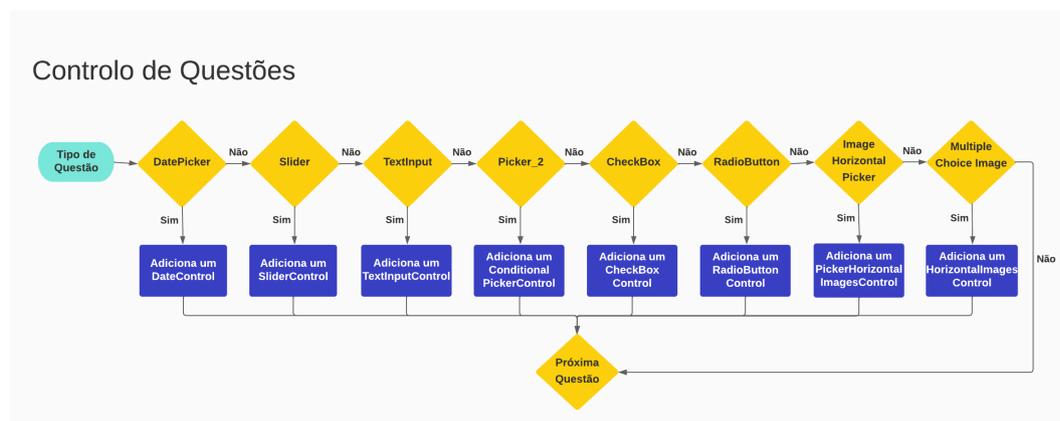


Figura 3.2: Fluxograma do Controlo de Questões.

### 3.3 Características do ficheiro descritivo em *JSON*

Para a biblioteca funcionar de forma correta, uma parte fundamental é a criação do documento descritivo. Por este motivo, para cada controlo presente na biblioteca será apresentada uma imagem da sua aparência na aplicação e o seu respetivo código *JSON*, presente no documento descritivo. O ficheiro *JSON* é constituído por uma

### 3.3 Características do ficheiro descritivo em JSON

---

coleção de pares atributo/valor ou uma lista ordenada de valores.

O *TitleControl* é o controlo representado na Figura 3.3 à direita. O código *JSON* para criar um *TitleControl* está representado na Figura 3.3 à esquerda e tem na sua constituição um atributo denominado *title*, este será um objeto constituído com vários pares atributo/valor, onde os atributos adicionados são as linguagens e o valor será o texto que irá constituir o título.

```
"title": {  
  "pt": "Exemplo de título",  
  "en": "Title Example"  
}
```

A imagem mostra um botão de aplicação com o texto "Exemplo de título" em uma cor verde neon sobre um fundo escuro.

Figura 3.3: *TitleControls* - Código *JSON* (esquerda) e representação na aplicação (direita).

A representação de um *DescriptionControl* encontra-se na Figura 3.4 à direita. O código *JSON* tem a mesma estrutura do título, porém o atributo é alterado para *description* como se pode verificar na Figura 3.4 à esquerda.

```
"description": {  
  "pt": "Exemplo de Descrição",  
  "en": "Description Example"  
}
```

A imagem mostra um botão de aplicação com o texto "Exemplo de Descrição" em uma cor verde neon sobre um fundo escuro.

Figura 3.4: *DescriptionControl* - Código *JSON* (esquerda) e representação na aplicação (direita).

Para adicionar questões ao formulário é necessário adicionar o atributo *n\_questions* ao ficheiro *JSON*. Este atributo é constituído por um valor inteiro, o qual representa o número de questões a adicionar.

```
"n_questions": 5
```

No caso do nosso formulário conter muitas questões e pretender-se criar um *CarouselView* em vez de um *Stacklayout*, é apenas necessário adicionar o atributo *questions\_per\_page* ao ficheiro *JSON*. O valor inteiro adicionado corresponderá ao número de perguntas por página no *Carousel*.

```
"questions_per_page": 6
```

Para adicionar as questões irá ser adicionado um atributo denominado por *questions*. A estrutura do *questions* é um *array* com uma lista de objetos. Para cada objeto será necessário adicionar as características do respetivo controlo da biblioteca.

### 3. Biblioteca SocialiteForms

---

```
questions: [  
  {  
    Type: "Tipo de questão1"  
    ...  
  },  
  {  
    Type: "Tipo de questão2"  
    ...  
  }  
  ...  
]
```

Um *DateControl* é adicionado quando o objeto contém o par atributo/valor *type/DatePicker*. O objeto também pode ter os seguintes atributos: *question*, *min\_date* e *max\_date* como se pode observar na Figura 3.5 à esquerda. O atributo *question* é um objeto que irá ser constituído com a questão para cada linguagem e funcionará da mesma maneira para todos os controlos da biblioteca. O atributo *min\_date* irá ser a data mais antiga que pode ser selecionada no *DatePicker* e o *max\_date* a última data que pode ser selecionada. Na Figura 3.5 à direita está representado um exemplo de um *DateControl*.

```
"type": "DatePicker",  
"question": {  
  "pt": "Questão de um DateControl",  
  "en": "Question of an DateControl"  
},  
"min_date": "01/01/2011",  
"max_date": "Current_Date"
```

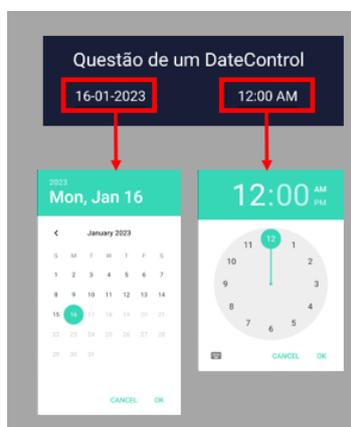


Figura 3.5: *DateControl* - Código *JSON* (esquerda) e representação na aplicação (direita).

Um *SliderControl* é adicionado quando o objeto contém o *type/Slider*. O objeto também pode constituir os seguintes atributos: *question*, *min\_label*, *max\_label*, *min* e *max* como se pode observar na Figura 3.6 à esquerda. Os *min\_label* e *max\_label* são um par atributo/valor simples ou um controlo que representa a legenda anterior e posterior ao *slider*, como está representado na Figura 3.6 à direita. O valor *min* representa o valor mínimo do *slider* e o valor *max* é o valor máximo do *slider*.

### 3.3 Características do ficheiro descritivo em JSON

```
"type": "Slider",
"question": {
  "pt": "Questão de um Slider",
  "en": "Question of an Slider"
},
"min_label": {
  "pt": "label inicial",
  "en": "beginning label"
},
"max_label": {
  "pt": "label final",
  "en": "end label"
},
"max": 4,
"min": 0
```

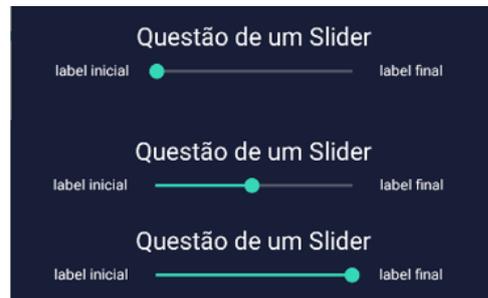


Figura 3.6: *SliderControl* - Código JSON (esquerda) e representação na aplicação (direita).

Um *TextInputControl* é adicionado quando o objeto contém o `type/TextInput`. O objeto também pode constituir os seguintes atributos: *question* e *inputType* como se pode verificar na Figura 3.7 à esquerda. O *inputType* representa o tipo de teclado que irá ser apresentado quando se vai responder. Na Figura 3.7 à direita está representado um exemplo de um *TextInputControl*.

```
"type": "TextInput",
"question": {
  "pt": "Questão de um TitleControl",
  "en": "Question of na TitleControl"
},
"inputType": "numeric"
```



Figura 3.7: *TextInputControl* - Código JSON (esquerda) e representação na aplicação (direita).

Um *ConditionalPickerControl* é adicionado quando o objeto contém o `type/Picker_2`. Este controlo é um caso específico da biblioteca onde se adicionam dois *pickers*, cuja escolha do valor no primeiro *picker* determina quais são as opções que aparecem no segundo. Na Figura 3.8 à esquerda verifica-se que o objeto necessita dos seguintes atributos: *question*, *question\_2* e *item\_source*. Neste caso, este controlo contém duas questões distintas. O *item\_source* é um *array* de controlos constituídos por um par *value* e *value\_2*. O *value* será um controlo constituído por cada valor do primeiro *picker* e a sua respetiva linguagem. O *value\_2* corresponde a um *array* com os valores associados ao *ItemSource* do *picker\_2* e varia consoante o valor escolhido no *picker\_1*, como se observa na Figura 3.8 à direita.

### 3. Biblioteca SocialiteForms

```
"type" : "Picker_2",
"question" :{
  "pt" : "Exemplo da 1ª questão",
  "en" : "Example of 1st questions"
},
"question_2" : {
  "pt": "Exemplo da 2ª questão",
  "en": "Example of 2nd question?"
},
"item_source": [
  {
    "value": {
      "pt": "Valor1",
      "en": "Value1"
    },
    "value_2": [
      "0",
      "1"
    ]
  },
  {
    "value": {
      "pt": "Valor2",
      "en": "Value2"
    },
    "value_2": [
      "- 10",
      "10-19"
    ]
  }
],
...

```

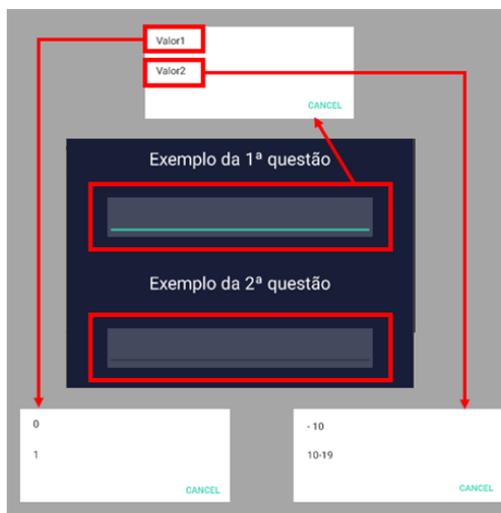


Figura 3.8: *ConditionalPickerControl* - Código *JSON* (esquerda) e representação na aplicação (direita).

Um *CheckBoxControl* é adicionado quando o objeto contém o *type/checkbox*. O objeto do ficheiro *JSON* é constituído pelos seguintes atributos: *question* e *options*, como mostra a Figura 3.9 à esquerda. O *options* é um objeto com uma lista de *arrays*. Cada *array* constitui todas as opções da *checkbox* para a respetiva linguagem. Na Figura 3.9 à direita está representado um exemplo de uma *checkbox*, a qual permite selecionar várias opções.

```
"type": "checkbox",
"question": {
  "pt": "Questão da checkbox",
  "en": "Question of checkbox"
},
"options": {
  "pt": [
    "opção 1",
    "opção 2"
  ],
  "en": [
    "option 1",
    "option 2"
  ]
},
...

```



Figura 3.9: *CheckBoxControl* - Código *JSON* (esquerda) e representação na aplicação (direita).

### 3.3 Características do ficheiro descritivo em JSON

Um *PickerHorizontalImagesControl* é adicionado quando o objeto contém o *type/ImageHorizontalPicker*. O objeto é constituído pelos seguintes atributos: *image\_source*, *label\_source* e *item\_source*, como se pode observar na Figura 3.10 à esquerda. O *image\_source* é constituído por um controlo com a lista de imagens ou o valor da imagem numa lista passada por uma função. O *label\_source* é o valor que irá ser colocado numa *label*. O *item\_source* é um controlo com uma lista de *arrays* com valores para cada uma das linguagens. Na Figura 3.10 à direita está representado um exemplo de um *PickerHorizontalImagesControl*.

```
"type": "ImageHorizontalPicker",
"image_source": 0,
"label_source": "",
"item_source": {
  "pt": [
    "Comunicação",
    "Lazer",
    "Pesquisa",
    "Trabalho"
  ],
  "en": [
    "Communication",
    "Hobby",
    "Search",
    "Work"
  ]
}
```

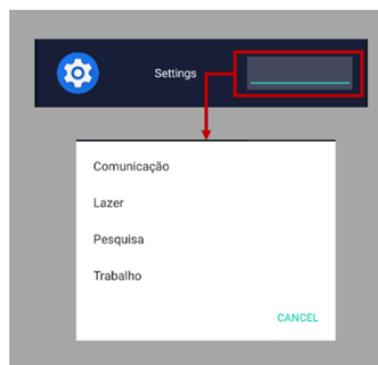


Figura 3.10: *PickerHorizontalImagesControl* - Código JSON (esquerda) e representação na aplicação (direita).

Um *HorizontalImagesControl* é adicionado quando o objeto contém o *type/MultipleChoiceImage*. O objeto é constituído pelos seguintes atributos: *number\_of\_labels*, *labels*, *number\_of\_images*, *list\_image\_source* e *image\_source\_number*, como se pode observar na Figura 3.11 à esquerda. A *labels* será um objeto constituído por uma lista de objetos denominados *label*. Cada *label* é constituída por um *text* que representa o texto da *label* e a *position* que é o local onde vai ser colocada. O *number\_of\_images* é a quantidade de imagens que pertencem à questão. A *list\_image\_source* é um *array* com a lista das imagens e o *image\_source\_number* é necessário caso as imagens sejam passadas por uma função.

### 3. Biblioteca SocialiteForms

```
"type": "MultipleChoiceImage",
"number_of_labels": 1,
"labels": [
  {
    "label": {
      "text": {
        "pt": "Calmo/Relaxado",
        "en": "Calm/Relaxed"
      },
      "position": "initial"
    }
  }
],
"number_of_images": 3,
"image_source_number": 0,
"list_image_source": [
  "sam_vector_relax_1.png",
  "sam_vector_relax_2.png",
  "sam_vector_relax_3.png"
]
```

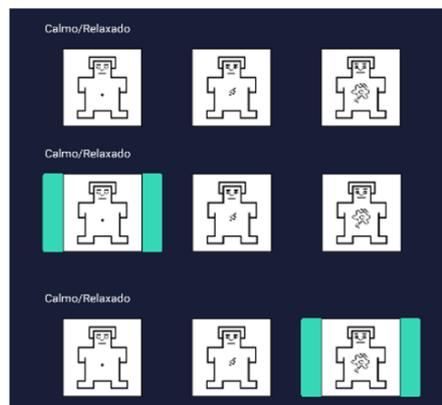


Figura 3.11: *HorizontalImagesControl* - Código *JSON* (esquerda) e representação na aplicação (direita).

Um *RadioButtonControl* é adicionado quando o objeto contém o *type/radiobutton*. O objeto é constituído pelos seguintes atributos: *question* e *options*, como se pode observar na Figura 3.12 à esquerda. O *options* é um objeto com uma lista de *arrays*. Cada *array* constitui a lista de *radiobuttons* para a respetiva linguagem. Na Figura 3.12 à direita está representado um exemplo de um *RadioButtonControl*.

```
"type": "radiobutton",
"question": {
  "pt": "Questão do radiobutton",
  "en": "Question of a radiobutton"
},
"options": {
  "pt": [
    "opção 1",
    "opção 2"
  ],
  "en": [
    "option 1",
    "option 2"
  ]
}
```

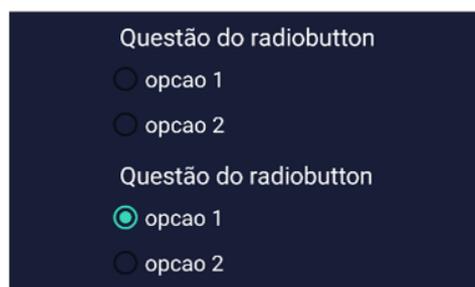


Figura 3.12: *RadioButtonControl* - Código *JSON* (esquerda) e representação na aplicação (direita).

# 4

## Aplicação ISABELA

### Conteúdo

---

4.1	Contextualização . . . . .	28
4.2	Descrição . . . . .	28
4.3	Requisitos . . . . .	29
4.4	Arquitetura . . . . .	30
4.5	Formulários da aplicação ISABELA . . . . .	34

---

### 4.1 Contextualização

O principal objetivo da aplicação ISABELA é melhorar o desempenho acadêmico dos estudantes universitários e irá também acompanhar os estudantes diariamente de modo a prevenir determinados comportamentos que poderão culminar em hábitos que prejudiquem o desempenho acadêmico ou levem ao abandono acadêmico.

Existem vários fatores que podem contribuir para um mau desempenho acadêmico e até mesmo desistência acadêmica. Um dos fatores mais preocupantes é a depressão [31], que pode advir da falta de exercício físico, do isolamento social, de perturbações de sono e da ausência de metodologias de estudo. A monitorização destes fatores irá permitir uma melhoria no desempenho acadêmico.

Como foi dito anteriormente a aplicação ISABELA foi apenas desenvolvida para o SO *Android* e neste momento encontra-se a ser recriada utilizando a *cross-platform XAMARIN*. Esta *cross-plataform* permite que a aplicação seja desenvolvida em simultâneo para os SOs *Android* e *iOS*.

### 4.2 Descrição

A aplicação ISABELA é um sistema HitLCPS com o objetivo de monitorizar e auxiliar estudantes diariamente. Esta aplicação tem a capacidade de deduzir dados acerca do comportamento, contexto social, físico e emocional do estudante, recorrendo apenas a informações recolhidas pelos *smartphones* dos utilizadores. Também recolhe dados através de formulários disponibilizados aos utilizadores, que ajudam na interpretação dos dados recolhidos através dos sensores.

A dedução é conseguida através de componentes da *FIWARE* que têm a capacidade de processar os dados. A comunicação com a *FIWARE* requer conexão à *Internet*. Na ausência dessa conexão, a comunicação não ocorre podendo levar à perda de dados. Desta forma criou-se um armazenamento local que permite guardar todos os dados quando o dispositivo não tem acesso à *Internet*. Assim que a conexão é reestabelecida, os dados são imediatamente enviados para a *FIWARE* e eliminados do armazenamento local.

A aplicação também é constituída por um *chatbot* cuja finalidade é auxiliar o utilizador. Este tem a capacidade de responder a questões efetuadas pelo estudante ou alertá-lo de algumas atitudes que o possam levar a um mau desempenho acadêmico.

## 4.3 Requisitos

Nas seguintes subsecções serão apresentados os requisitos da aplicação ISABELA. Estes podem ser divididos em requisitos funcionais e não funcionais.

### 4.3.1 Requisitos não funcionais

Na Tabela 4.1 podem ser observados os Requisitos Não Funcionais (RNF) da aplicação ISABELA. Para cada um dos requisitos é apresentada a sua função e a sua descrição. Estes requisitos embora não se encontrem em contacto direto com o utilizador, permitem que este confie e utilize a aplicação. Por este motivo os RNF são tratados com um grau de prioridade elevada.

Para aplicações de monitorização e recolha de dados, a privacidade é um dos RNF mais importante. Como tal, o grupo de trabalho ISABELA tem elementos dedicados exclusivamente à segurança e privacidade. Estes elementos são responsáveis por gerir todos os problemas que possam surgir nesse âmbito.

Requisitos Não Funcionais		
N.º	Função	Descrição
RNF1	Desempenho	A aplicação não deve ter grande impacto no tempo de vida da bateria.
RNF2	Portabilidade	A aplicação deve ser desenvolvida de forma a funcionar em dispositivos <i>Android</i> e <i>iOS</i> .
RNF3	Usabilidade	A aplicação deve ter um UI claro e objetivo.
RNF4	Reusabilidade	Devem ser usadas bibliotecas e APIs para facilitar o desenvolvimento da aplicação.
RNF5	Acessibilidade	A aplicação não deve ser condicionada pelo elevado número de utilizadores.
RNF06	Privacidade	O utilizador não tem acesso a dados de outro utilizador. A autenticação da aplicação é efetuada através do <i>Facebook</i> . Os dados são eliminados dos <i>smartphones</i> após 30 minutos ou após o envio para a <i>FIWARE</i> no caso de haver acesso à <i>Internet</i> .

Tabela 4.1: Requisitos Não Funcionais da Aplicação ISABELA

### 4.3.2 Requisitos Funcionais

Na Tabela 4.2 podem ser observados os Requisitos Funcionais (RF) da aplicação. Para cada um dos requisitos é apresentada a função, a prioridade e uma breve descrição do mesmo. O grau de prioridade de cada requisito varia de 1 a 3. O grau de prioridade 1 consiste nos requisitos que ainda não estão implementados na aplicação. Estes requisitos estão planeados para uma futura implementação. O grau de prioridade 2 é constituído por requisitos de elevada importância para a aplicação,

## 4. Aplicação ISABELA

que não afetam o seu funcionamento. Por fim, o grau de prioridade 3 é constituído por requisitos que são essenciais para que a aplicação funcione.

Requisitos Funcionais			
N.º	Função	Prioridade	Descrição
RF01	Introdução	2	A aplicação deve apresentar uma página de introdução, com o nome do projeto e a Universidade de Coimbra.
RF02	Janela de permissões	3	A aplicação quando é iniciada pela primeira vez, deve pedir permissão ao utilizador para aceder aos sensores, embutidos no <i>smartphone</i> .
RF03	Login com o Facebook	3	A plataforma tem que permitir fazer <i>login</i> através da conta do Facebook.
RF04	Logout	3	A aplicação deve permitir que o utilizador termine a sua sessão.
RF05	Registo do utilizador	3	A aplicação deve permitir o registo de novas contas.
RF06	Chatbot	2	Um <i>chatbot</i> deve estar inserido na aplicação para auxiliar o utilizador.
RF07	Configuração inicial	3	Após aberta a aplicação pela primeira vez esta deve abrir uma página de configuração inicial, permitindo ao utilizador escolher as configurações iniciais.
RF08	Página de Configurações	3	A aplicação deve conter uma página de configurações que permita alterar as configurações iniciais.
RF09	Fluxo da aplicação	3	A aplicação deve permitir que o utilizador retroceda e volte para a página inicial.
RF10	Recolher Informação	3	A aplicação deve ter a capacidade de recolher informação dos sensores de modo a deduzir dados relativamente à sua localização, sociabilidade e atividade.
RF11	Guardar Informação	3	A aplicação deve recolher e guardar dados mesmo sem acesso à Internet.
RF12	Formulários	2	A aplicação fornece regularmente formulários ao utilizador.
RF13	Notificações	2	A aplicação deve avisar o utilizador de formulários por responder através de notificações.
RF14	Enviar os dados para a FIWARE	3	A aplicação deve enviar toda a informação recolhida para a FIWARE mal tenha acesso à Internet.
RF15	Gráficos com informação da atividade	2	A aplicação deve apresentar através de gráficos, o nível de atividade do utilizador.
RF16	Detetar hábitos	2	A aplicação precisa reconhecer hábitos que possam afetar negativamente o desempenho académico.
RF17	Informação sobre o utilizador	1	A aplicação deve recolher informação sobre o utilizador através de formulários e através do <i>chatbot</i> .
RF18	Informação sobre as unidades curriculares	1	A aplicação deve ter conhecimento das unidades curriculares que o aluno frequenta.
RF19	Formulário de estudo	1	A aplicação, após conhecer as unidades curriculares que o utilizador frequenta, deve disponibilizar formulários que ajudem o aluno a estudar.

Tabela 4.2: Requisitos Funcionais da Aplicação ISABELA

Alguns RF necessitam de pré-requisitos para a sua implementação, tais como a permissão de acesso aos sensores do *smartphone*, a conexão à Internet, a permissão para o envio de notificações, o acesso a informação acerca das unidades curriculares, o utilizador ter uma conta no Facebook e informação sobre as unidades curriculares.

### 4.4 Arquitetura

A arquitetura do sistema ISABELA irá ser apresentada em duas secções, a arquitetura geral e a arquitetura da aplicação móvel. Na secção da arquitetura geral será

feita uma breve explicação da arquitetura de um sistema HitLCPS seguida de uma breve explicação dos componentes da *FIWARE* que foram utilizados. Na secção da arquitetura da aplicação móvel será explicada a arquitetura da aplicação, o que não difere muito de aplicações móveis comuns.

#### 4.4.1 Arquitetura geral

A aplicação ISABELA é baseada num sistema HitLCPS cujo objetivo é auxiliar os estudantes universitários a melhorarem os seus desempenhos académicos. Na figura 4.1 encontra-se representada a arquitetura geral da aplicação ISABELA. Esta é constituída por diversos tipos de mecanismos que ajudam à aquisição e processamento de dados. Os dados processados permitem fazer uma inferência, uma inferência futura e atuação, tal como é feito em sistemas HitLCPS

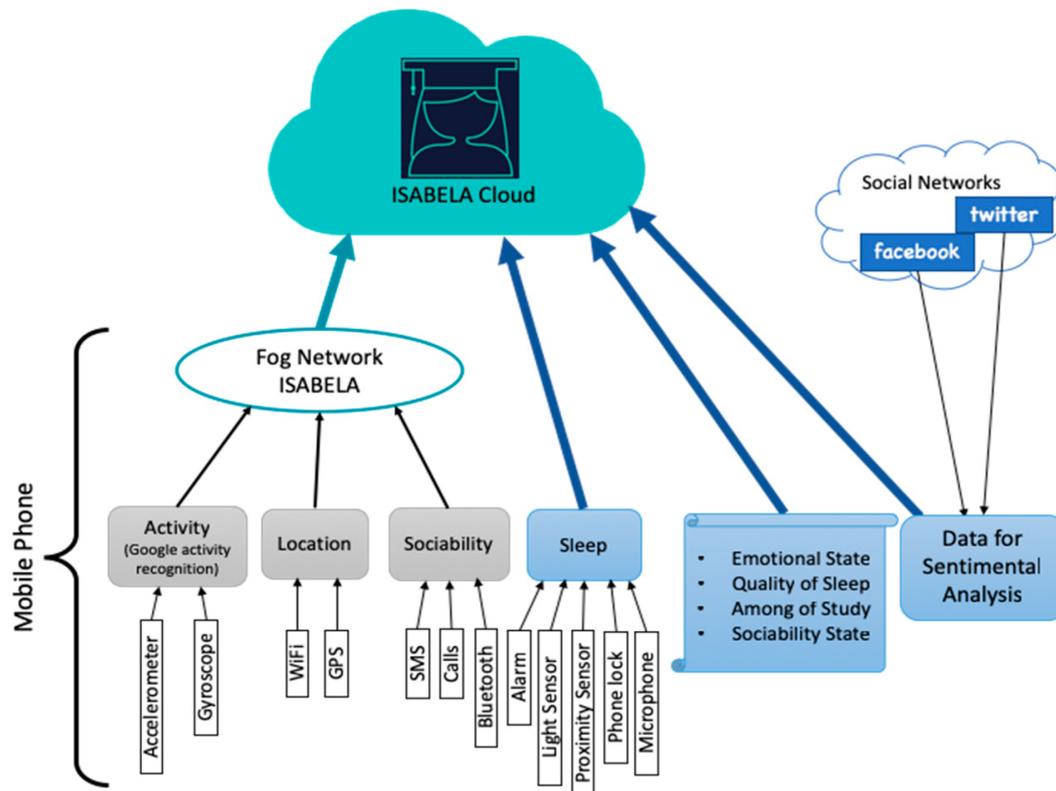


Figura 4.1: Arquitetura da aplicação ISABELA para recolha e processamento de dados. Adaptado de [7]

A plataforma ISABELA utiliza componentes da *FIWARE*, GEs, para cumprir as necessidades de um sistema HitLCPS. Os GEs são baseados no modelo NGSI9/10, que é baseado em atributos e entidades. As entidades deste modelo constituem características próprias e são representadas por atributos, através do formato *JSON*. Na figura 4.2 pode observar-se a nuvem do sistema ISABELA, a qual utiliza com-

## 4. Aplicação ISABELA

---

ponentes da *FIWARE*.

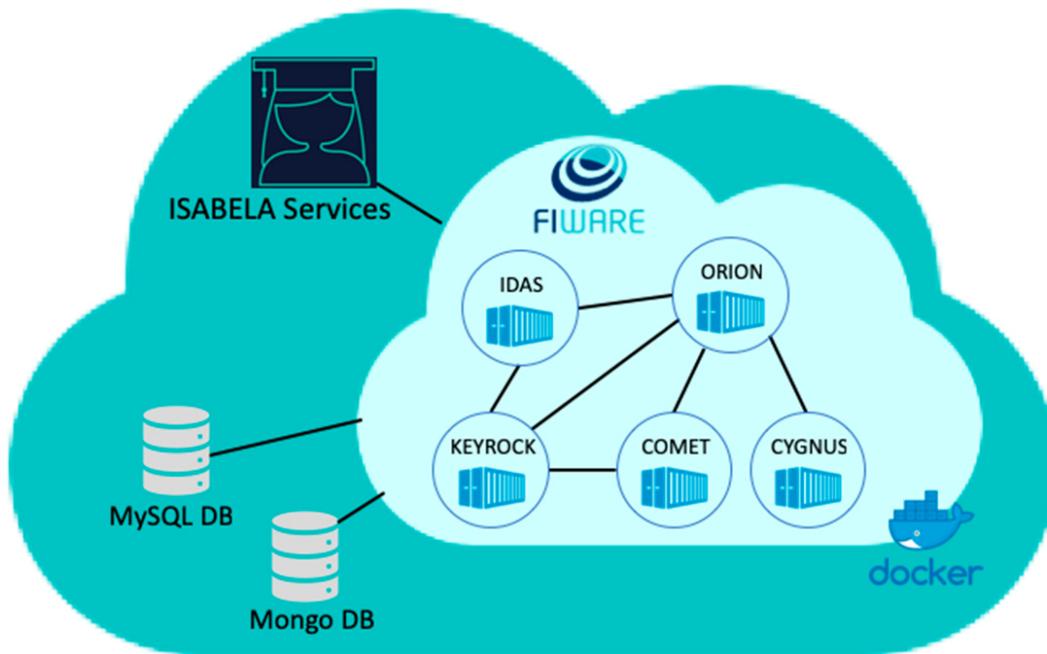


Figura 4.2: Nuvem ISABELA. Adaptado de [7]

A aplicação ISABELA é constituída por cinco GEs, e cada um deles tem diferentes características e funções [32]:

- *IDAS* - fornece uma forma padronizada e unificada de conectar e gerir dispositivos IoT. Atua como uma ponte entre os dispositivos IoT e a *FIWARE*, facilitando a integração e interoperabilidade.
- *ORION* - permite criar, apagar e obter entidades, bem como atualizar as existentes. As entidades armazenadas por este GE possibilitam o armazenamento dos dados recolhidos de dispositivos IoT. Contudo, este GE não é adequado para armazenar dados históricos.
- *CYGNUS* - é responsável por armazenar certas fontes de dados em diferentes sistemas de armazenamento de terceiros, como *MySQL* e *MongoDB*. Quando uma nova entidade chega a este GE, ela é colocada num canal específico e enviada para o sistema de armazenamento escolhido.
- *STH-COMET* - é responsável por recuperar os dados armazenados diretamente para a aplicação. Fornece uma API *RESTful* para consultas históricas e métodos agregados. Quando a aplicação ISABELA precisa de aceder a dados históricos, este GE conecta-se à base de dados e procura os dados solicitados.

- *KEYROCK* - este GE adiciona autenticação e segurança à aplicação, além de políticas de autorização. Também garante a privacidade dos dados, seguindo as leis de privacidade da União Europeia, como o Regulamento Geral de Proteção de Dados da União Europeia [33]. Todos os dados recolhidos e tratados pela aplicação são anonimizados.

Além dos componentes da *FIWARE* na nuvem, a aplicação oferece dois serviços adicionais:

- Análise do sono - monitoriza o sono do utilizador com base nos dados dos sensores do *smartphone*.
- *Natural Language Processing* (NLP) - utiliza dados das redes sociais, tais como o *Facebook*, para detetar sentimentos e emoções.

### 4.4.2 Arquitetura da aplicação móvel

Na figura 4.3 é possível observar que existe uma UI essencial (*UI Thread*) e um conjunto de pequenas *threads* que atuam em segundo plano, resultando numa aplicação rápida e fluída nas iterações com o utilizador. Nesta figura também é possível observar que a arquitetura ISABELA se divide em três serviços principais: a *UI Thread*, o serviço principal e o serviço de comunicação com a *FIWARE*. Estes serviços estão responsáveis por:

- *UI Thread* - esta *thread* da aplicação lida com tarefas da interface do utilizador. Ela garante uma experiência de utilização suave, gerindo elementos da interface, respondendo a ações do utilizador e atualizando o ecrã. Para manter a interface responsiva, tarefas longas ou intensivas devem ser executadas em *threads* ou processos separados.
- Serviço Principal - este serviço é responsável pela aquisição de dados através de sensores nos telemóveis e de redes sociais como o *Facebook*.
- Serviço de Comunicação com a *FIWARE* - este serviço é responsável por enviar os dados recolhidos para a *FIWARE* e também pela comunicação entre o utilizador e o *chatbot*.

A comunicação com a *FIWARE* necessita de uma conexão à *Internet* estável. Deste modo, foi criada uma base de dados local, responsável pelo armazenamento de dados na ausência de conexão à *Internet*. No momento em que a ligação seja

## 4. Aplicação ISABELA

reestabelecida os dados serão enviados para a *FIWARE* e eliminados da base de dados local. Deste modo não há perda de dados.

O *chatbot* é responsável por responder a questões feitas pelo utilizador. Este envia as mensagens para o *Dialogflow* [34], que posteriormente comunica com a *FIWARE* para obter os dados que o utilizador solicitou.

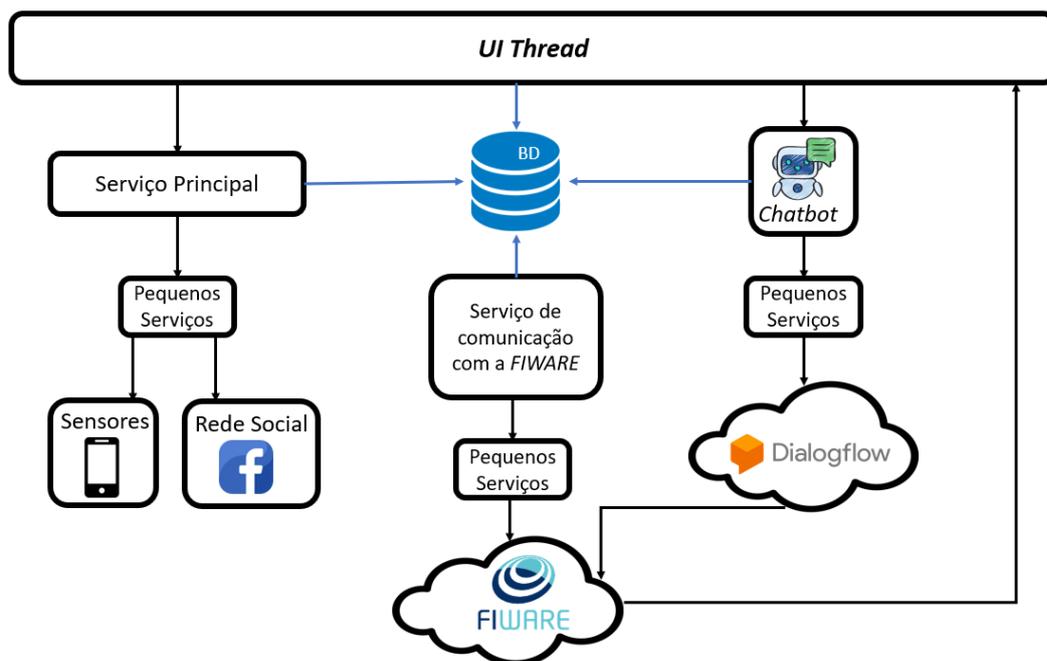


Figura 4.3: Arquitetura da aplicação móvel ISABELA.

## 4.5 Formulários da aplicação ISABELA

Nesta secção conseguiu-se testar o desempenho da biblioteca *SocialiteForms*. Através da utilização desta biblioteca foi feita uma reformulação dos formulários que integram a aplicação ISABELA. Esta aplicação é constituída por sete formulários diferentes, que têm como principal objetivo obter informação acerca do utilizador de forma direta.

Os formulários são constituídos por um botão *Enviar*, que permite validar o seu preenchimento. No caso de ocorrer algum erro será apresentada uma janela *pop-up* a pedir ao utilizador que verifique o formulário. Se o utilizador não tiver acesso à *Internet*, estes formulários serão guardados localmente no *smartphone* até que seja estabelecida uma conexão. Ao ser reestabelecida, esses formulários serão enviados para a *FIWARE* e eliminados do *smartphone*. Após o envio, todos os dados serão processados na *FIWARE* de modo a que sejam apresentados ao utilizador mais tarde.

### 4.5.1 Formulário de proximidade

O formulário de proximidade permitiu que a aplicação ISABELA fosse adaptada para o projeto VITORIA - *Amiga Virtual para Assistência Inteligente*. Este formulário foi criado com o aparecimento da pandemia da doença do coronavírus (COVID-19) e tem como objetivo saber quantas pessoas se encontram na proximidade do utilizador. Através da utilização de *Bluetooth* a aplicação consegue estimar se o utilizador se encontra na proximidade de outras pessoas. A aplicação notifica o utilizador que tem um formulário de proximidade por responder sempre que se detetar alguém na proximidade. Na Figura 4.4 à esquerda pode verificar-se que o formulário apenas é constituído por um *TextInputControl*. Como a resposta esperada é numérica, o teclado é alterado para numérico como está representado na Figura 4.4 à direita.

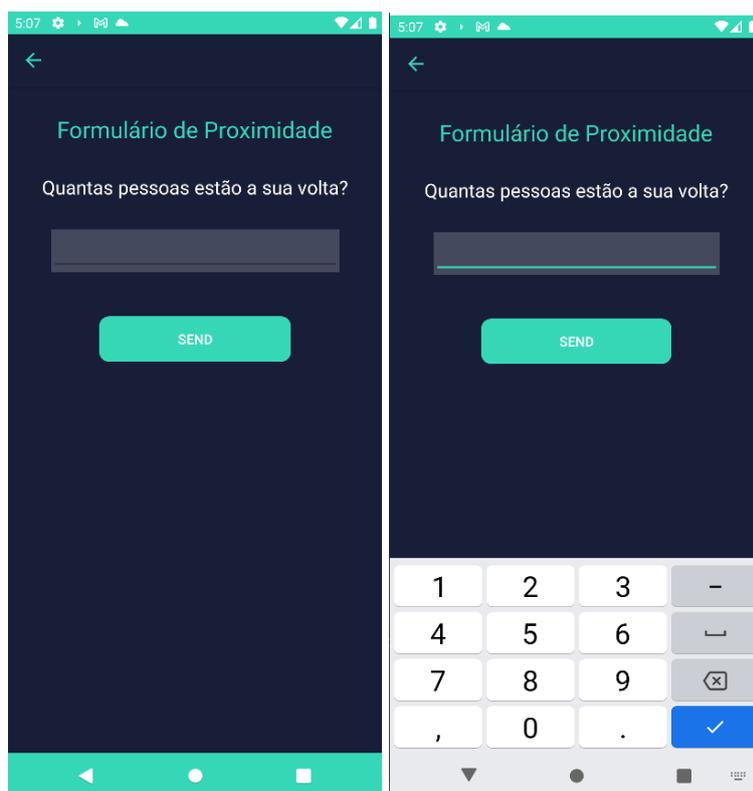


Figura 4.4: Formulário de proximidade (esquerda). Tipo de teclado utilizado no formulário de proximidade (direita).

## 4. Aplicação ISABELA

### 4.5.2 Formulário de Sono

Na Figura 4.5 pode verificar-se que o formulário de sono é constituído por seis questões. As duas primeiras questões estão interligadas e têm como objetivo saber quantas horas o aluno dormiu, recorrendo a dois *DateControl*. As restantes questões são constituídas por *SliderControls*, onde cada questão têm uma escala de avaliação distinta como se pode observar na Tabela 4.3.

Figura 4.5: Formulário de Sono

Questão	Escala
Como dormiu a noite passada?	Escala qualitativa entre Muito Mal e Muito Bem
Como se sentiu socialmente ontem?	Escala qualitativa entre Introverso e Extroverso
Quantas horas de exercício físico praticou ontem?	Escala constituída por 0, 1, 2, 2+
Quantas horas estudou ontem?	Escala constituída por 0, 2, 4, 8+

Tabela 4.3: Questões do formulário de sono e respetiva escala de avaliação

### 4.5.3 Formulário *POMS*

O formulário *POMS* é um formulário que tem como finalidade obter informação acerca do estado emocional do utilizador. Na figura 4.6 à esquerda, consegue observar-se parte dos doze adjetivos que constituem o formulário emocional. O formulário emocional é constituído por doze *SliderControls*. Cada *SliderControl* é constituído por uma escala de avaliação com cinco níveis distintos (nada; um pouco; moderadamente; muito e muitíssimo) que são explicados ao utilizador através de um *DescriptionControl*.



Figura 4.6: Formulário *POMS* da aplicação ISABELA.

### 4.5.4 Formulário emocional com imagens

O formulário emocional com imagens também tem como finalidade obter informação sobre o estado emocional do utilizador. Este formulário é constituído por dois *HorizontalImagesControls*. Na figura 4.7 à direita consegue observar-se que cada *HorizontalImagesControl* é constituído por cinco imagens. As cinco imagens do primeiro controlo permitem ao utilizador avaliar o seu nível de agitação, onde a escala varia entre *Calmo/Relaxado* (primeira imagem) e *Agitado/Alerta* (última imagem). As cinco imagens do último controlo permitem ao utilizador avaliar o seu nível de prazer, onde a escala varia entre *Muito Prazer/Agradável* (primeira imagem) e *Muito Desprazer/Desagradável* (última imagem).

## 4. Aplicação ISABELA

---

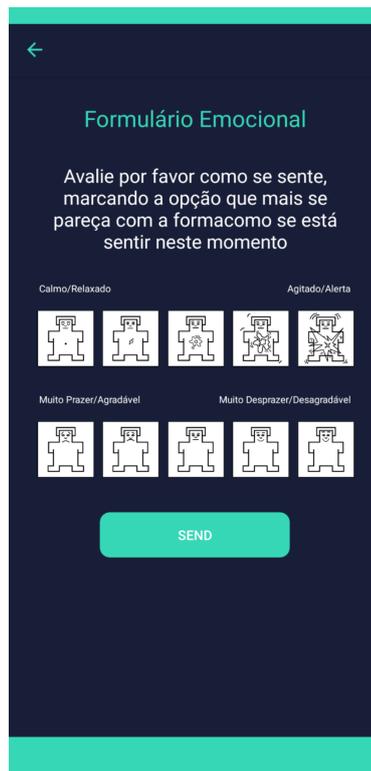


Figura 4.7: Formulário emocional com imagens.

### 4.5.5 Formulário de transporte

O formulário de transporte aparece para ser respondido quando a API da *Google - Activity Recognition API* [35], deteta que o utilizador se encontra num veículo. O utilizador recebe uma notificação que existe um novo formulário de transporte para responder.

O formulário de transporte é constituído por um *ConditionalPickerControl*. Como se pode observar na figura 4.8, o controlo é constituído por duas questões que estão relacionadas entre si, sendo estas *Em que veículo esteve?* e *Quantas pessoas estavam consigo?*. Na tabela 4.4 podem observar-se os meios de transporte do *picker* inicial e as respetivas opções que aparecem no segundo *picker*.

## 4.5 Formulários da aplicação ISABELA

Figura 4.8: Formulário de transporte

<b>Opções do primeiro <i>picker</i></b>	<b>Opções do segundo <i>picker</i> condicionadas</b>
Veículo próprio	0; 1; 2; 2+
Veículo de amigos/colegas	0; 1; 2; 2+
Autocarro	-10; 10-19; 20-30; +30
Barco	-10; 10-29; 30-50; 50+
Táxi/TVDE	0; 1; 2; 2+
Metro/Comboio/Elétrico	-10; 10-19; 20-30; +30

Tabela 4.4: Opções do primeiro *picker* do formulário de transporte e opções condicionadas do segundo *picker*.

## 4. Aplicação ISABELA

### 4.5.6 Formulário *HEXACO*

O grupo de Psicologia Aplicada da Faculdade de Medicina formulou as questões que fazem parte do formulário *HEXACO*. Este formulário é apresentado apenas uma vez a cada utilizador, pois estima-se que a personalidade não varie em menos de 6 meses (tempo estimado para testar a aplicação ISABELA).

O formulário *HEXACO* tem como objetivo traçar a personalidade de um aluno universitário, com o recurso a sessenta questões. Este, é apresentado em seis páginas distintas de forma a não ficar muito extenso em apenas uma página. Para tal, utilizou-se um *CarouselView*, onde cada página do *carousel* é constituída por dez *SliderControls*. Cada *SliderControl* é constituído por uma escala de avaliação com cinco níveis distintos (discordo fortemente; discordo; neutro; concordo e concordo fortemente) que são explicados ao utilizador através de um *DescriptionControl*. Na Figura 4.9 encontram-se apresentadas várias questões do formulário de personalidade.

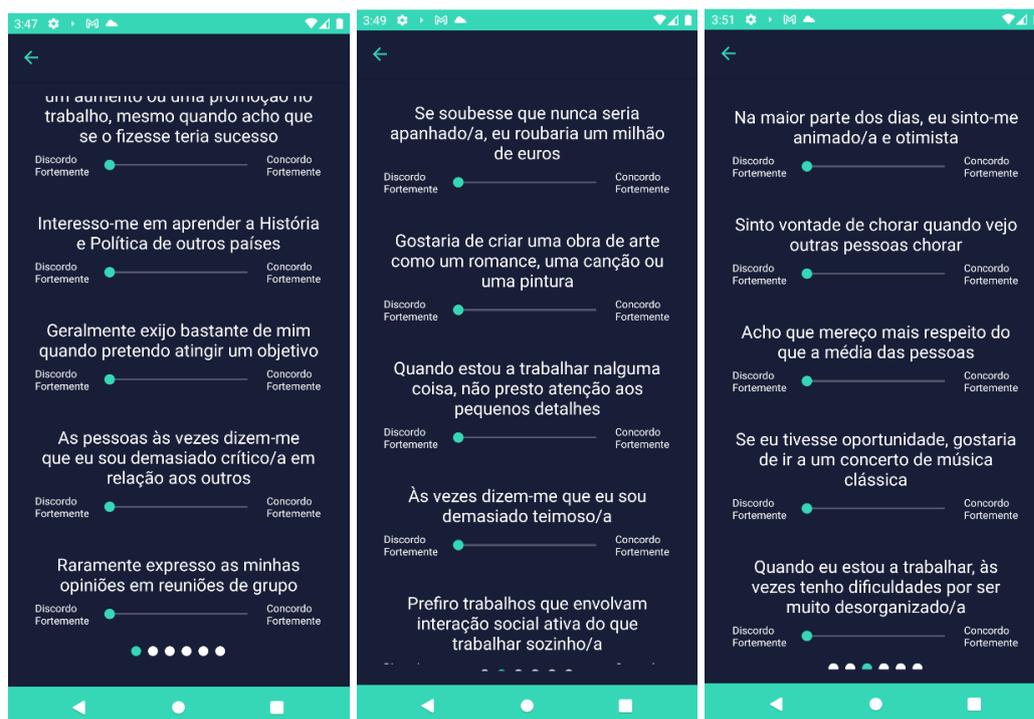


Figura 4.9: Algumas questões do formulário *HEXACO*

Após o envio dos dados deste formulário para a *Firebase*, este é substituído por uma página informativa acerca da personalidade. Na Figura 4.10 está representada a página informativa onde constam os vários elementos da personalidade do aluno: abertura e experiência, conscienciosidade, extroversão, amabilidade, neuroticismo e honestidade/humildade.



Figura 4.10: Página informativa da personalidade do aluno universitário.

### 4.5.7 Formulário de finalidade da aplicação

A aplicação disponibiliza um formulário de finalidade da aplicação, que tem como objetivo saber qual a finalidade da utilização das cinco aplicações mais usadas pelo utilizador. Na figura 4.11 à esquerda pode observar-se que o formulário de finalidade da aplicação é constituído por cinco *PickerHorizontalImageControls*. Cada *picker* é constituído por quatro opções, sendo estas: *Comunicação*, *Lazer*, *Pesquisa* e *Trabalho*, como se pode observar na figura 4.11 à direita.

## 4. Aplicação ISABELA

---

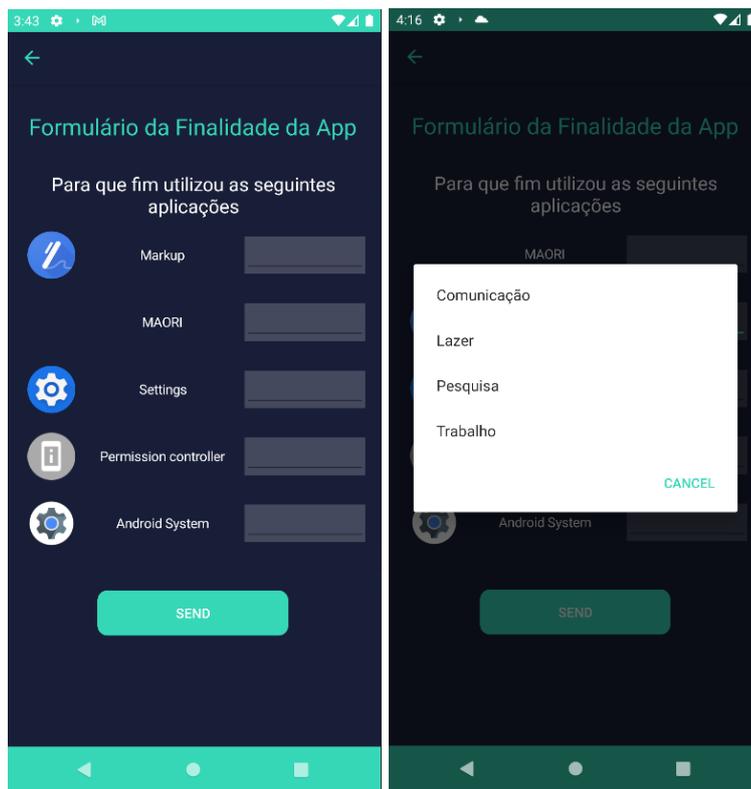


Figura 4.11: Formulário de finalidade da aplicação (esquerda). Opções da finalidade da aplicação (direita)

# 5

## MAORI

### Conteúdo

---

5.1	Contextualização . . . . .	44
5.2	Desenvolvimento . . . . .	44

---

### 5.1 Contextualização

A aplicação MAORI é uma adaptação da aplicação ISABELA para a Força Aérea Portuguesa. Esta aplicação será utilizada numa das fases de recrutamento, no estágio de seleção de voo. Este estágio, se não ocorrerem imprevistos, terá a duração de nove dias nos quais os candidatos terão dois dias de formação teóricos seguidos de sete dias nos quais terão voos diários.

O objetivo da aplicação será recolher o máximo de dados acerca dos candidatos através dos sensores dos *smartphones* e formulários durante a semana de estágio de voo. Posteriormente, serão utilizadas técnicas de *machine learning* de forma a criar perfis de futuros candidatos com o recurso ao seu desempenho no voo e aos dados recolhidos.

### 5.2 Desenvolvimento

A aplicação MAORI tem a maioria das funcionalidades da aplicação ISABELA, pois mantém a recolha de dados dos sensores embutidos no *smartphone*, algumas páginas informativas e três formulários. Na figura 5.1 estão representados os três formulários da aplicação MAORI e a página referente ao tempo do treino de voo.

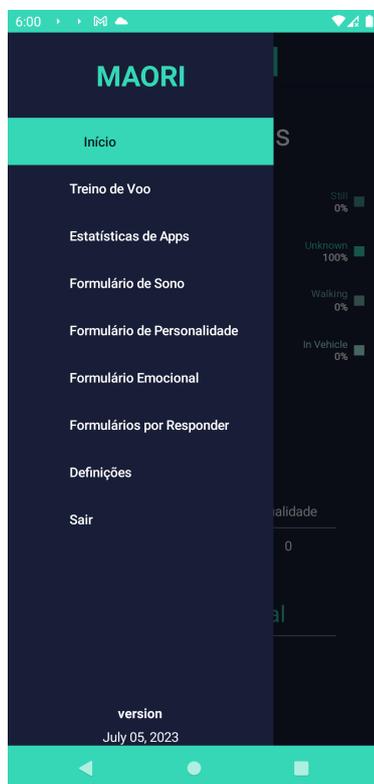


Figura 5.1: Página inicial do MAORI

### 5.2.1 Página do treino de Voo

Nas figura 5.2 à esquerda encontra-se representada a página do treino de voo que é constituída por uma *label* e um *imagebutton* que se encontra a vermelho. A cor vermelha indica que o candidato não se encontra a efetuar um treino de voo. Na figura 5.2 à direita está representada a mesma página, porém com a cor verde indicando que o candidato se encontra a efetuar o treino de voo. O objetivo será adicionar a duração do treino de voo aos dados recolhidos.

A aplicação deverá funcionar sem acesso à *Internet*, dado que durante os treinos de voo os candidatos devem desligar todas as conexões nos seus *smartphones*. Desta forma, as horas de início e término de voo serão armazenadas localmente e enviadas para o servidor assim que for estabelecida uma conexão à *Internet*.

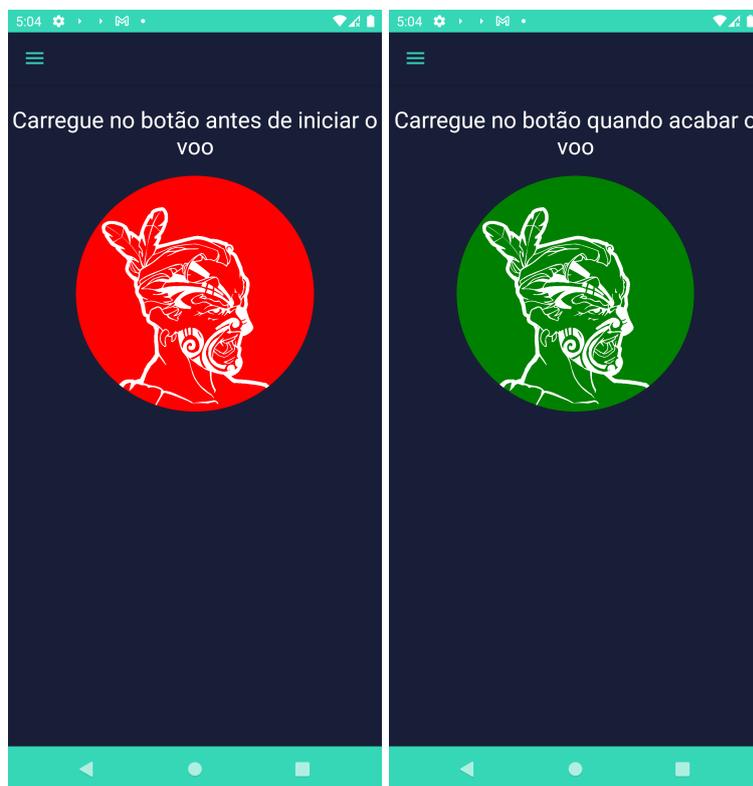


Figura 5.2: Página de treino de voo enquanto não se efetua um treino (esquerda). Página de treino de voo quando se encontra em treino (direita).

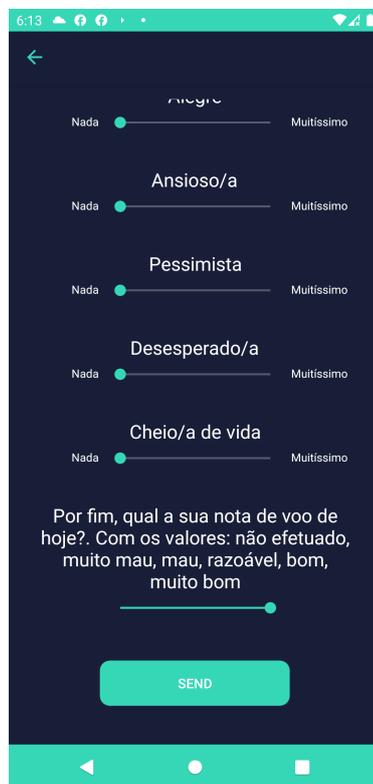
### 5.2.2 Formulário *POMS*

O formulário *POMS* da aplicação MAORI foi ligeiramente alterado em relação ao formulário emocional usado na aplicação ISABELA. Ao formulário original foi adicionada uma questão que será feita ao candidato. Na figura 5.3 pode observar-se a nova questão (*SliderControl*): *Por fim, qual a sua nota de voo de hoje?. Com os*

## 5. MAORI

---

valores: *não efetuado, muito mau, mau, razoável, bom, muito bom*. Esta questão permitirá obter os dados relacionados com o desempenho do candidato durante o estágio de voo.



The image shows a mobile application interface for a POMS questionnaire. The screen has a dark blue background with white text. At the top, there is a status bar with the time 6:13 and various icons. Below the status bar is a back arrow icon. The main content consists of five horizontal sliders, each with a red dot indicating a selected value. The sliders are labeled as follows: 1. 'Nada' on the left and 'Muitíssimo' on the right. 2. 'Ansioso/a' in the center. 3. 'Pessimista' in the center. 4. 'Desesperado/a' in the center. 5. 'Cheio/a de vida' in the center. Below the sliders is a text prompt: 'Por fim, qual a sua nota de voo de hoje?. Com os valores: não efetuado, muito mau, mau, razoável, bom, muito bom'. Underneath the text is another horizontal slider with a red dot. At the bottom of the form is a red button labeled 'SEND'. The bottom of the screen shows the standard Android navigation bar with back, home, and recent apps icons.

Figura 5.3: Formulário *POMS* da aplicação MAORI.

# 6

## **Avaliações e Testes**

## 6. Avaliações e Testes

---

A biblioteca *SocialiteForms* permitiu que a criação de formulários se tornasse num processo mais simples. Como se pode observar na tabela 6.1, antes da biblioteca o *XAML* de cada formulário era variável e com muitas linhas de código. Após a criação da biblioteca o mesmo passou a conter apenas treze linhas de código, tornando-se o documento descritivo (ficheiro *JSON*) responsável pela parte estrutural e pelas funcionalidades das questões dos formulários. Na tabela 6.1 também se verifica que existe uma redução de linhas de código em praticamente todos os formulários.

Formulários	<i>XAML</i> antes da biblioteca	<i>XAML</i> depois da biblioteca	Ficheiro <i>JSON</i>	Redução em % de linhas de código
Formulário do sono	202	13	111	38.61
Formulário <i>POMS</i>	391	13	229	38.11
Formulário <i>HEXACO</i>	1954	13	1042	46.11
Formulário de proximidade	49	13	33	6.12
Formulário do transporte	64	13	111	-93.4
Formulário finalidade da aplicação	220	13	90	53.18
Formulário emocional com imagens	391	13	229	38.11

Tabela 6.1: Impacto da biblioteca *SocialiteForms* nas linhas de código utilizadas na criação de formulários da linguagem de marcação *XAML*.

Na tabela 6.2 pode observar-se que, no geral, existe uma diminuição das linhas de código da linguagem de programação *C#*. Tal deve-se à adição de algumas das funcionalidades dos formulários à biblioteca. A biblioteca adiciona um botão que pode ser acedido no *C#*, permitindo que quem a utiliza aceda aos valores das questões. Assim, conclui-se que a biblioteca facilita o processo da criação dos formulários.

<b>Formulários</b>	<b>C# antes da biblioteca</b>	<b>C# depois da biblioteca</b>	<b>Redução em % de linhas de código</b>
Formulário do sono	70	46	34.29
Formulário emocional	128	62	51.56
Formulário personalidade	146	150	-2.74
Formulário de proximidade	42	42	0
Formulário do transporte	150	44	70.67
Formulário finalidade da aplicação	302	125	58.61
Formulário emocional com imagens	128	82	35.94

Tabela 6.2: Impacto da biblioteca *SocialiteForms* nas linhas de código utilizadas na criação de formulários da linguagem de programação *C#*

Ao criar novas aplicações móveis é necessário ter em atenção o consumo de bateria. A criação de novas bibliotecas requer também algum cuidado para que não haja alterações no consumo de bateria da aplicação. Assim realizou-se um teste antes e depois da adição da biblioteca *SocialiteForms*, para analisar o seu impacto no consumo de bateria.

Na figura 6.1 à direita pode observar-se que o consumo de bateria é 15,06 mAh, isto é 15,06 miliamperes por hora, indicando um consumo diário de 361.44 mA. Visto que um *smartphone* tem normalmente entre 3500 mA e 5000 mA [36], pode-se concluir que o impacto da aplicação na bateria é inferior a 10%. Comparando o consumo com a figura 6.1 à esquerda pode observar-se que o consumo antigo é quase idêntico ao consumo com a utilização da biblioteca representado na 6.1 à direita.

## 6. Avaliações e Testes

---

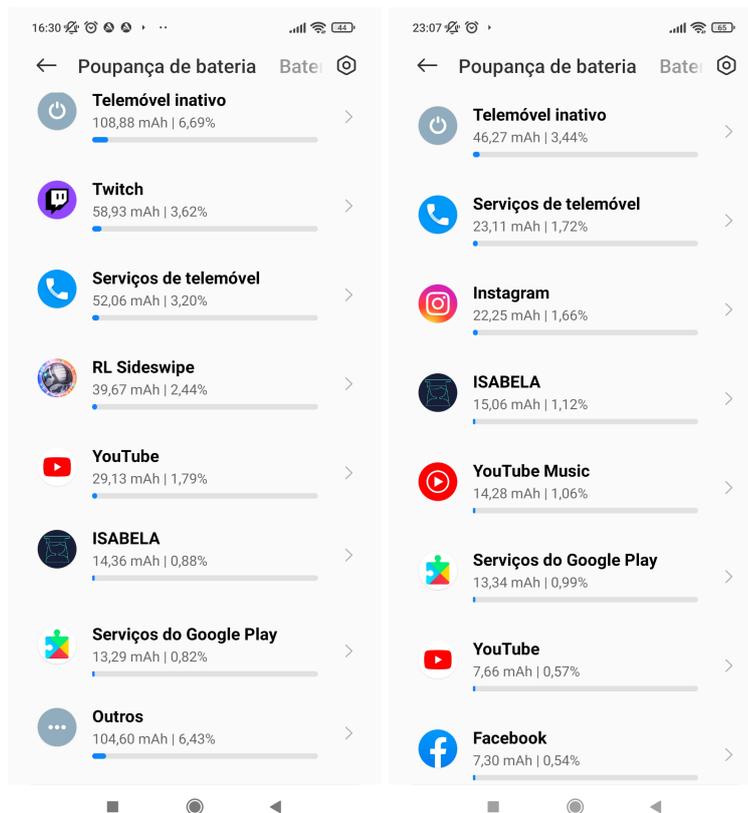


Figura 6.1: Consumo de bateria com os formulários antigos (esquerda). Consumo de bateria com os formulários a usar a biblioteca *SocialiteForms* (direita).

# 7

## **Conclusão e Trabalho Futuro**

### **Conteúdo**

---

<b>7.1</b>	<b>Conclusão . . . . .</b>	<b>52</b>
<b>7.2</b>	<b>Trabalho Futuro . . . . .</b>	<b>52</b>

---

### 7.1 Conclusão

Esta dissertação teve como objetivo a recriação da aplicação ISABELA, que inicialmente se encontrava desenvolvida apenas para o SO *Android*. De forma a criar uma aplicação que fosse capaz de alcançar 98% de utilizadores dos *smartphones*, foi utilizada a *cross-platform XAMARIN*, que permite desenvolver simultaneamente aplicações para os SOs *Android* e *iOS*. A plataforma ISABELA difere de aplicações com funcionalidades semelhantes, visto que considera o fator Humano como um elemento ativo do sistema (HitLCPS). Estes sistemas são divididos em quatro etapas: aquisição de dados inicial, inferência inicial, seguida de uma inferência futura e, por fim, atuação. Para além deste sistema, a plataforma de apoio ao ensino necessitou de *smartphones*, componentes da *FIWARE* e das seguintes tecnologias: AI, *chatbot* e o paradigma de IoT. Tal possibilitou a criação de uma aplicação com o propósito de monitorizar e aconselhar estudantes universitários a melhorarem o desempenho académico.

Também foi desenvolvida uma biblioteca para facilitar a criação de formulários em aplicações móveis desenvolvidas em *XAMARIN*. A biblioteca *SocialiteForms* permite a alteração dos formulários das plataformas MAORI e ISABELA com relativa facilidade e a diminuição do trabalho necessário na criação de novos formulários. Para tal, esta biblioteca apenas necessita de um ficheiro descritivo *JSON* e poucas linhas de código na linguagem de programação *C#* e linguagem de marcação *XAML*, para criar a parte visual dos formulários. A biblioteca *SocialiteForms* insere automaticamente um botão no final dos formulários com o objetivo de fornecer ao utilizador uma forma de aceder às respostas das questões.

Por fim, desenvolveu-se também a aplicação MAORI, que é uma adaptação da aplicação ISABELA. A aplicação MAORI tem como objetivo recolher dados acerca de candidatos à Força Aérea Portuguesa, durante uma das fases de recrutamento. De momento esta aplicação não tem a capacidade de aconselhar o candidato das atitudes que o podem ajudar ou prejudicar durante o recrutamento, dado que ainda não foram recolhidos dados acerca dos candidatos.

### 7.2 Trabalho Futuro

Futuramente serão adicionados novos formulários tanto na aplicação ISABELA como noutras aplicações do projeto usando a biblioteca *SocialiteForms*. Ainda que a biblioteca *SocialiteForms* consiga cumprir com todas as necessidades da criação da parte visual dos formulários, poderá ser necessário criar novos controlos para questões mais complicadas.

No futuro, será crucial realizar testes da aplicação ISABELA num conjunto

de alunos universitários para que possam obter informações relacionadas com a utilização da aplicação e a eficácia do sistema HitLCPS. Relativamente à aplicação MAORI, a mesma será sujeita a testes de desempenho, para que esteja operacional e pronta a ser utilizada pelos candidatos à Força Aérea Portuguesa na fase de recrutamento, especificamente durante o estágio de seleção de voo. Através da medição de indicadores e mecanismos de *machine learning* será possível obter os perfis dos candidatos e dessa forma aconselhá-los individualmente a adotar ou modificar determinadas atitudes e comportamentos. As recomendações serão fornecidas através de um *chatbot* ou notificações.

Por fim, a aplicação ISABELA poderá ser melhorada através da adição de novas funcionalidades, como por exemplo: a adição de formulários semanais ou mensais relacionados com as unidades curriculares que o aluno frequenta. Isto permitirá verificar qual o nível de conhecimento de determinada unidade curricular, sendo apenas necessário que o aluno faculte informação relativamente à inscrição nessa mesma unidade curricular.

# Bibliografia

- [1] “Xamarin,” <https://learn.microsoft.com/en-us/xamarin/get-started/what-is-xamarin>, accessed:2023-01-23.
- [2] P. Z. D. S. Nunes and J. S. Silva, “A survey on human-in-the-loop applications towards an internet of all,” *IEEE Communications Surveys Tutorials*, vol. 17, no. 2, pp. 944–965.
- [3] “Bigbluebutton,” <https://bigbluebutton.org/>, accessed: 2023-05-01.
- [4] “Aleks,” <https://www.aleks.com/>, accessed: 2023-05-01.
- [5] “Socrative,” <https://www.socrative.com/>, accessed: 2023-05-01.
- [6] R. Wang, F. Chen, Z. Chen, T. Li, G. Harari, S. Tignor, X. Zhou, D. Ben-Zeev, and A. T. Campbell, “StudentLife,” in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. New York, NY, USA: ACM, Sep. 2014.
- [7] S. Sinche, P. Hidalgo, J. Fernandes, D. Raposo, J. Silva, A. Rodrigues, N. Armando, and F. Boavida, “Analysis of student academic performance using human-in-the-loop cyber-physical systems,” *Telecom*, vol. 1, no. 1, p. 18–31, Mar 2020. [Online]. Available: <http://dx.doi.org/10.3390/telecom1010003>
- [8] “Internet of things - number of connected devices worldwide 2015-2025,” <https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/>, accessed: 2023-06-25.
- [9] D. Nunes, J. S. Silva, C. Herrera, and F. Boavida, “Human-in-the-loop connectivity management in smartphones,” in *Wired/Wireless Internet Communications*, L. Mamatas, I. Matta, P. Papadimitriou, and Y. Koucheryavy, Eds. Cham: Springer International Publishing, 2016, pp. 159–170.

- [10] D. Nunes, J. S. Silva, and F. Boavida, *A Practical Introduction to Human-in-the-Loop Cyber-Physical Systems*, 1st ed. Wiley-IEEE Press, 2018.
- [11] M. I. L. d. S. Lopes, A. P. S. d. S. Pereira, and P. M. F. Vaz, “Abandono escolar no ensino superior e fatores concorrentes,” *EduSer*, vol. 15, no. 1, Mai. 2023.
- [12] “How many phones are in the world,” <https://www.bankmycell.com/blog/how-many-phones-are-in-the-world>, accessed: 2023-07-02.
- [13] “How many devices have ios or android in the world,” <https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/>, accessed: 2023-06-07.
- [14] “The history of android: The evolution of the biggest mobile os in the world,” <https://www.androidauthority.com/history-android-os-name-789433/>, accessed: 2023-01-23.
- [15] “Platform architecture,” <https://developer.android.com/guide/platform?hl=en>, accessed: 2023-01-23.
- [16] “Apple ios operating system: History, origin, and more,” <https://history-computer.com/apple-ios-operating-system-guide/>, accessed: 2023-01-23.
- [17] “Architecture of ios operating system,” <https://www.geeksforgeeks.org/architecture-of-ios-operating-system/>, accessed: 2023-01-23.
- [18] “ios sdk,” <https://www.techopedia.com/definition/3819/ios-sdk>, accessed: 2023-01-23.
- [19] “What is .net? an overview of the platform,” <https://auth0.com/blog/what-is-dotnet-platform-overview/>, accessed: 2023-01-23.
- [20] “Xamarin forms,” <https://learn.microsoft.com/en-us/xamarin/get-started/what-is-xamarin-forms>, accessed: 2023-04-25.
- [21] “Future internet public-private partnership,” <https://wayback.archive-it.org/12090/20161223145325/https://ec.europa.eu/digital-single-market/en/future-internet-public-private-partnership>, accessed: 2023-04-25.
- [22] “Fiware about us,” <https://www.fiware.org/about-us/#community>, accessed: 2023-04-25.
-

## Bibliografia

---

- [23] K. Chopra, K. Gupta, and A. Lambora, “Future internet: The internet of things-a literature review,” in *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 2019, pp. 135–139.
- [24] J.-P. A. Yaacoub, O. Salman, H. N. Noura, N. Kaaniche, A. Chehab, and M. Malli, “Cyber-physical systems security: Limitations, issues and future trends,” *Microprocessors and Microsystems*, vol. 77, p. 103201, 2020.
- [25] S. Sinche, P. Hidalgo, J. M. Fernandes, D. Raposo, J. S. Silva, A. Rodrigues, N. Armando, and F. Boavida, “Analysis of student academic performance using human-in-the-loop cyber-physical systems,” *Telecom*, vol. 1, no. 1, pp. 18–31, 2020. [Online]. Available: <https://www.mdpi.com/2673-4001/1/1/3>
- [26] “Plano de recuperação e resiliência,” <https://recuperarportugal.gov.pt/>, accessed: 2023-07-11.
- [27] “On-board: na luta contra o abandono escolar,” <https://noticias.uc.pt/artigos/on-board-na-luta-contr-o-abandono-escolar/>, accessed: 2023-07-11.
- [28] “Uc next,” <https://www.ucnext.pt/>, accessed: 2023-05-02.
- [29] “Uc framework,” <https://ucpages.uc.pt/ucframework/>, accessed: 2023-05-02.
- [30] “Studentlife,” <https://studentlife.cs.dartmouth.edu/>, accessed: 2023-05-01.
- [31] S. Khurshid, Q. Parveen, M. Yousuf, and D. A. Chaudhry, “Effects of depression on students’ academic performance,” *Science International 1013-5316*, vol. 27, pp. 1619–1624, 10 2015.
- [32] J. Fernandes, D. Raposo, N. Armando, S. Sinche, J. S. Silva, A. Rodrigues, V. Pereira, H. G. Oliveira, L. Macedo, and F. Boavida, “Isabela – a socially-aware human-in-the-loop advisor system,” *Online Social Networks and Media*, vol. 16, p. 100060, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S246869642030001X>
- [33] “A proteção de dados na ue,” [https://commission.europa.eu/law/law-topic/data-protection/data-protection-eu\\_pt](https://commission.europa.eu/law/law-topic/data-protection/data-protection-eu_pt), accessed: 2023-07-06.
- [34] “Documentação dialogflow,” <https://cloud.google.com/dialogflow/docs?hl=pt-br>, accessed: 2023-07-06.

- [35] “Activity recognition api,” <https://developers.google.com/location-context/activity-recognition?hl=pt-br>, accessed: 2023-07-12.
- [36] “Capacity of battery in smartphones,” <https://www.androidauthority.com/smartphone-battery-size-poll-results-1221015/>, accessed: 2023-07-07.
- [37] “.net maui,” <https://learn.microsoft.com/en-us/dotnet/maui/what-is-maui>, note = Accessed: 2023-07-11.