



UNIVERSIDADE D
COIMBRA

José Miguel Rocha Oliveira

**ISABELA – DESENVOLVIMENTO DE UMA
APP PARA UM SISTEMA IOT DE APOIO À
APRENDIZAGEM**

**Dissertação no âmbito da Especialização em Telecomunicações 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 Tecnologia da
Universidade de Coimbra.**

julho de 2022



FCTUC FACULDADE DE CIÊNCIAS
E TECNOLOGIA
UNIVERSIDADE DE COIMBRA

ISABELA – Desenvolvimento de uma app para um sistema IoT de apoio à aprendizagem

José Miguel Rocha Oliveira

Coimbra, julho 2022



ISABELA – Desenvolvimento de uma app para um sistema IoT de apoio à aprendizagem

Orientador:

Professor Doutor Jorge Miguel Sá Silva

Coorientador:

Professor Doutor André Miguel de Almeida Marrão Rodrigues

Júri:

Professor Doutor Marco Alexandre Cravo Gomes (Presidente)

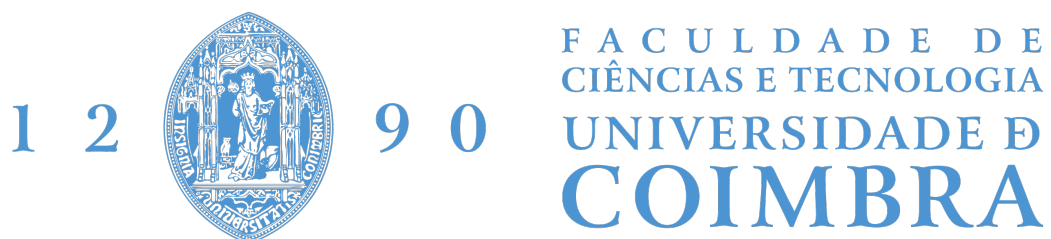
Professor Doutor Jorge Miguel Sá Silva (Orientador)

Professor Doutor Mahmoud Tavakoli (Vogal)

Dissertação apresentada à Faculdade de Ciências e Tecnologia da Universidade de Coimbra como requisito parcial para obtenção do grau de Mestre em Engenharia Eletrotécnica e de Computadores, na especialidade de Telecomunicações.

Coimbra, julho 2022

Este trabalho foi desenvolvido em colaboração com:



Agradecimentos

Começo por agradecer ao Professor Doutor Jorge Sá Silva e Professor Doutor André Rodrigues, respetivamente meu orientador e coorientador do meu trabalho de dissertação, pela oportunidade de integrar e trabalhar neste grupo de investigação. Foi um privilégio poder aprender e investigar junto desta equipa. Agradeço o constante acompanhamento para discutir e tentar solucionar as dificuldades que surgiram ao longo deste percurso.

A todos os meus colegas de laboratório e equipa de trabalho, principalmente ao Gonçalo Dias e Miguel Monteiro, pelo companheirismo e apoio, bem como pelo importante contributo que tiveram para que este trabalho de equipa fosse realizado em sintonia e com o maior dos sucessos. Gostava de agradecer particularmente ao Marcelo Fernandes por todos os ensinamentos técnicos imprescindíveis para a realização deste trabalho e pela enorme disponibilidade que demonstrou ao longo deste ano.

Agradeço à minha família, especialmente aos meus pais, por me proporcionarem a oportunidade de tirar um curso superior e sempre me apoiarem.

Por último, quero agradecer à Diana Adão, que sempre me motivou e ajudou a alcançar os meus objetivos ao longo do meu percurso académico e por ser um exemplo de responsabilidade, dedicação e empenho.

Abstract

Currently, smart devices stand widespread in our lives, with smartphones being, by far, the most used. However, most applications do not consider humans as active system players, providing all users the same service with a unique objective. They disregard human feelings, emotions, intentions, actions, or even social context. In fact, all of them are factors to consider when developing monitoring or lifestyle applications.

Due to the evolution of concepts such as the Internet of Things (IoT), cyber-physical systems (CPS) and their monitoring capabilities, control of physical environment and communication, the interaction of Humans with mobile devices is increasingly facilitated. This interaction can still be improved, in terms of assistance and monitoring, when considering the Human as an active element of the system, using a Human-in-the-Loop Cyber-Physical System (HITLCPS).

This thesis aims to recreate and expand the ISABELA platform, an IoT application for academic monitoring, existing exclusively for the Android operating system. Starting from ISABELA, its recreation is proposed through the development of an innovative platform for educational purposes, developed for smartphones with Android and iOS operating systems and with the aim of helping and guiding university students in order to improve their academic performance. The platform employs new technologies such as smartphones, artificial intelligence, a chatbot and the Internet of Things paradigm. Through these technologies, student data is collected, such as their location, study hours, sleep patterns, physical activity, social context. This information is provided to a HITLCPS, where it is analyzed and later used as a way of helping and monitoring the student in his/her academic path.

Keywords

Internet of Things; Mobile Device Programming; Human-in-the-Loop Cyber-Physical System

Resumo

Atualmente, os dispositivos móveis inteligentes estão bastante difundidos nas nossas vidas, sendo os *smartphones* os mais utilizados. No entanto, a maioria das suas aplicações não considera o Ser Humano como um elemento ativo do sistema, fornecendo a todos os utilizadores o mesmo serviço, com um único objetivo. Estas aplicações desconsideram os sentimentos, emoções, intenções, ações do Ser Humano e ainda o contexto social, que são fatores essenciais quando se pretende desenvolver aplicações de monitorização e auxílio.

Através da evolução de conceitos como a Internet das Coisas, sistemas ciber-físicos e das suas capacidades de monitorização, controlo de fenómenos físicos e de comunicação, a interação do Ser Humano com um dispositivo móvel é cada vez mais facilitada. Esta interação pode ainda ser melhorada, em termos de auxílio e monitorização, quando considerado o Ser Humano como um elemento ativo do sistema, através da utilização do conceito HITLCP.

Esta tese tem como objetivo recriar e expandir a plataforma ISABELA, uma aplicação IoT de monitorização académica, existente exclusivamente para sistema operativo *Android*. Partindo do ISABELA, é proposta a sua recriação, através do desenvolvimento de uma plataforma inovadora, para fins educativos, desenvolvida para *smartphones*, compatível com sistema operativo *Android* e *iOS*, e com o objetivo de ajudar e orientar estudantes universitários, de forma a melhorarem o seu desempenho académico. A plataforma emprega novas tecnologias, como smartphones, inteligência artificial, um *chatbot* e o paradigma da Internet das Coisas. Através destas tecnologias são recolhidos dados do estudante, tais como a sua localização, horas de estudo, padrões de sono, atividade física, contexto social. Estes dados são fornecidos ao sistema HITLCP, onde são analisados e posteriormente utilizados, de forma a auxiliar e monitorizar o estudante no seu percurso académico.

Palavras Chave

Internet das Coisas; Programação de Dispositivos Móveis; *Human-in-the-Loop Cyber-Physical System*

Conteúdo

Agradecimentos	iv
Abstract	v
Resumo	vi
Lista de Acrónimos	xi
Lista de Figuras	xii
Lista de Tabelas	xiv
1 Introdução	1
1.1 Contextualização	2
1.2 Motivação	2
1.3 Objetivos	3
1.4 Metodologia	4
1.5 Estrutura do documento	5
2 Estado da Arte	6
2.1 Conceitos base	7
2.1.1 Internet das Coisas	7
2.1.2 <i>Cyber-Physical System</i>	8
2.1.3 <i>Human-in-the-Loop Cyber-Physical System</i>	9
2.1.4 Importância de dados no século XXI e privacidade	10
2.2 Tecnologias base utilizadas	10
2.2.1 <i>Android</i>	10
2.2.2 <i>iOS</i>	11
2.2.3 <i>.NET</i>	11

2.2.4	<i>Xamarin</i>	11
2.2.5	<i>FIWARE</i>	12
2.3	Plataformas atuais - ISABELA/BATINA	13
2.4	Plataformas semelhantes	15
2.4.1	<i>StudentLife</i>	15
2.4.2	<i>BigBlueButton</i>	16
2.4.3	<i>Aleks</i>	17
2.4.4	<i>Socrative</i>	18
2.5	Análise comparativa	18
3	Aplicação ISABELA	20
3.1	Contextualização	21
3.2	Descrição	21
3.3	Arquitetura Geral	22
3.4	Arquitetura da aplicação móvel	24
3.5	Requisitos	25
3.5.1	Requisitos não funcionais	25
3.5.2	Requisitos funcionais	26
3.6	Desenvolvimento	28
3.6.1	Aplicação em <i>Xamarin</i>	29
3.6.2	Armazenamento	30
3.6.3	Aquisição de dados	32
3.6.4	Comunicação com a <i>FIWARE</i>	33
3.6.5	UI e fluxo da aplicação	34
4	Avaliações e Testes	48
5	Conclusão e Trabalho Futuro	52
5.1	Conclusão	53
5.2	Trabalho Futuro	53
	Bibliografia	55
	Anexos	59

A	Lista de permissões	60
A.1	<i>Android</i>	60
A.2	<i>iOS</i>	61

Lista de Acrónimos

API	<i>Application Programming Interface</i>
CPS	<i>Cyber-Physical System</i>
FCTUC	Faculdade de Ciências e Tecnologia da Universidade de Coimbra
FI-PPP	<i>Future Internet Public-Private Partnership</i>
GE	<i>Generic Enablers</i>
GPS	<i>Global Positioning System</i>
HITLCPS	<i>Human-in-the-Loop Cyber-Physical System</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IA	Inteligência Artificial
IoT	<i>Internet of Things</i>
LMS	<i>Learning Management System</i>
NLP	<i>Natural Language Processing</i>
RF	Requisito Funcional
RFID	<i>Radio-Frequency Identification</i>
RNF	Requisitos Não Funcionais
SO	Sistema Operativo
SDK	<i>Software Development Kit</i>
UI	<i>User Interface</i>
WiFi	<i>Wireless Fidelity</i>

Lista de Figuras

2.1	Previsão do número de dispositivos <i>Internet of Things</i> (IoT) connectados em todo o mundo desde 2019 a 2030 [1].	7
2.2	Arquitetura de um sistema ciber-físico. Adaptado de [11].	8
2.3	Processo de controlo de um sistema <i>Human-in-the-Loop</i> . Adaptado de [12].	9
2.4	Diagrama da arquitetura geral de uma aplicação em <i>Xamarin</i> [21]	12
2.5	Alguns layouts da aplicação ISABELA.	14
2.6	Alguns layouts da aplicação BATINA.	14
2.7	Arquitetura do projeto <i>StudentLife</i> [27].	15
2.8	Página inicial da plataforma <i>BigBlueButton</i> [28].	16
2.9	Relatório de progresso da plataforma <i>Aleks</i> [29].	17
2.10	Menu inicial da plataforma socrative para professores [30].	18
3.1	Arquitetura de aquisição de dados e processamento do ISABELA. Adaptado de [25].	22
3.2	Nuvem ISABELA. Adaptado de [25].	23
3.3	Arquitetura da aplicação móvel.	25
3.4	Exemplo de estruturação do código - <i>ViewModels</i> (esquerda) e <i>Views</i> (direita).	29
3.5	Exemplo de implementação de um botão.	29
3.6	Exemplo de estruturação do código.	30
3.7	Entidades relativos aos formulários.	31
3.8	Entidades relativas aos dados adquiridos pelos sensores do <i>smartphone</i>	32
3.9	Fluxo da aplicação.	34
3.10	Página Introdutória.	35
3.11	Janela <i>pop-up</i> de permissões relativa à localização e atividade física.	35
3.12	Página de <i>Login</i>	36
3.13	Página de configurações iniciais.	36

3.14	Página inicial com os dados ocultos.	37
3.15	Página inicial relativos às últimas 24 horas.	37
3.16	Página inicial relativos aos últimos 4 dias (esquerda) e 8 dias (direita).	37
3.17	Dados recolhidos através dos sensores do <i>smartphone</i>	37
3.18	<i>Shell</i> /Menu do utilizador.	38
3.19	Página de definições.	39
3.20	Página do <i>chatbot</i>	39
3.21	Página de classificações.	40
3.22	Página de estatísticas de aplicações.	40
3.23	Página de personalidade	41
3.24	Janela <i>pop-up</i> com a definição do atributo <i>Abertura e Experiência</i> (esquerda) e <i>Conscienciosidade</i> (direita).	41
3.25	Formulário de sono	42
3.26	<i>DataPicker</i>	42
3.27	Exemplo de perguntas do formulário de personalidade.	43
3.28	Formulário emocional	44
3.29	Formulário emocional com imagens	44
3.30	Formulário de proximidade.	45
3.31	Formulário de transporte.	45
3.32	Formulário finalidade de aplicação.	46
3.33	Exemplo de três notificações.	47
3.34	Formulários por responder	47
4.1	Alguns layouts da aplicação ISABELA executada em <i>iOS</i>	49
4.2	Formulários de proximidade executado em Sistema Operativo (SO) <i>iOS</i> (esquerda) e SO <i>Android</i> (direita).	50
4.3	Energia consumida pela aplicação ISABELA.	51

Lista de Tabelas

3.1	Requisitos não funcionais.	26
3.2	Requisitos funcionais - função e descrição.	27
3.3	Definição das prioridades dos requisitos funcionais.	28
3.4	Requisitos funcionais - pré-requisitos e prioridade.	28

1

Introdução

Conteúdos

1.1	Contextualização	2
1.2	Motivação	2
1.3	Objetivos	3
1.4	Metodologia	4
1.5	Estrutura do documento	5

1.1 Contextualização

Mais de vinte anos após a sua concepção, a IoT é uma tecnologia emergente que tem vindo a ser cada vez mais desenvolvida e aprimorada. Esta área tem uma enorme potencialidade de evolução e é considerada imprescindível por muitas pessoas, razão pela qual a sua utilização tem vindo a aumentar. É estimado que desde 2020 até 2030 o número de dispositivos IoT triplique, passando o seu valor de 9,7 mil milhões para mais de 29 mil milhões [1]. A IoT tem como um dos seus principais objetivos permitir uma interação inteligente e em tempo real entre *coisas*, máquinas e o Ser Humano. Atualmente, este conceito é utilizado em grande parte do nosso quotidiano, em diferentes campos, desde a agricultura até à medicina, de forma a tornar estas áreas de atuação inteligentes e mais rentáveis.

Associado ao crescimento do conceito de IoT, cresceu também a procura e utilização de dispositivos móveis IoT, como *smartphones* [2]. Estes dispositivos tiram partido de tecnologias como robótica e sensores sem fios, áreas que também beneficiaram de uma grande evolução. O crescimento destas vertentes tem vindo a permitir o desenvolvimento de sistemas ciber-físicos ou *Cyber-Physical System* (CPS), os quais possibilitam a monitorização e controlo de fenómenos físicos [3]. Recentemente, com o refinamento de CPS, surgiram sistemas que consideram também fatores humanos, tais como as suas ações, intenções e sentimentos. Estes sistemas, conhecidos como *Human-in-the-Loop Cyber-Physical System* (HITLCPS), permitiram uma considerável evolução em termos de aplicabilidade, eficácia e desempenho dos CPS [4]. Adicionalmente, devido ao desenvolvimento de tecnologias baseadas em Inteligência Artificial (IA), como *chatbots*, é possível, através de um dispositivo móvel, uma interação homem-máquina em diversas tarefas do dia a dia, desde receber indicações de navegação a acompanhamento emocional, melhorando a qualidade de vida do Ser Humano. [5].

1.2 Motivação

Conforme as estatísticas apresentadas pela Direção-Geral de Estatísticas da Educação e Ciência (DGEEC), 29% dos estudantes inscritos num curso superior, desistem do mesmo. Dos restantes alunos, 14% não completa o curso no tempo estipulado [6]. Embora alguns casos de desistência sejam devido a condições financeiras, existem também motivos como *stress*, falta de auto-disciplina e desorganização.

A fim de diminuir estas percentagens, referente a motivos não financeiros, podem ser exploradas várias medidas, sendo o acompanhamento personalizado uma opção eficaz e eficiente. No entanto, embora competente, este método exige uma enorme afetação de

recursos humanos, assim como custos adicionais associados [7].

Atualmente, é muito frequente os estudantes universitários utilizarem o *smartphone*, estimando-se que 96% destes estudantes utilização este dispositivo [8]. Beneficiando deste fator, o acompanhamento personalizado pode ser feito através de uma aplicação para *smartphone*, dotada de novas tecnologias, como técnicas de IA, o paradigma da IoT e ainda utilizando os sensores embutidos nos *smartphones*. Utilizando todas estas ferramentas, é possível recolher e avaliar dados físicos, comportamentais e sociais relativos ao estudante, com o propósito de desenvolver um sistema HITLCPS capaz de monitorizar e auxiliar o estudante. Deste modo, é possível evitar a sobrecarga de horário por parte dos professores e dos profissionais especializados em acompanhamento personalizado, assim como são evitados custos adicionais em remunerações, quer por parte da universidade/instituto superior, quer por parte do estudante.

1.3 Objetivos

Inicialmente, a plataforma ISABELA foi desenvolvida apenas para *smartphones* com o SO *Android*. No entanto, quando se pretende desenvolver uma aplicação para *smartphone*, é necessário ter em conta diferentes sistemas operativos, pois estes apresentam diferentes paradigmas relativamente ao desenvolvimento das suas aplicações (diferentes linguagens de programação e arquiteturas). É estimado que, de todos os sistemas operativos, o *Android* e o *iOS* sejam os mais utilizados, constando em 99% dos *smartphones* [9]. Tendo em conta este fator, a aplicação proposta será desenvolvida através de uma *cross-platform* chamada *Xamarin* que, conciliando a linguagem de programação *C#* com a linguagem de marcação *XAML*, permite o desenvolvimento de aplicações para *smartphones*, tanto para SO *Android*, como para *iOS*, através de um código 90% partilhado.

Esta tese propõe o desenvolvimento e reconstrução de uma plataforma inovadora de apoio à aprendizagem com base em novas tecnologias (como IoT, IA, um *chatbot*, utilização de componentes da *FIWARE*), chamada ISABELA - *IoT Student Advisor and Best Lifestyle Analyzer* - focada na educação e na melhoria do desempenho académico. Para tal, através de *smartphones*, dos sensores neles embutidos (tais como acelerómetro, *Global Positioning System* (GPS), etc.), e de inquéritos lançados na própria aplicação, esta permitirá, através de um sistema HITLCPS, monitorizar e auxiliar o estudante no seu dia a dia, de forma a melhorar o seu desempenho académico.

1.4 Metodologia

Nesta secção será explicado o planeamento para o desenvolvimento do projeto.

A plataforma ISABELA está a ser desenvolvida por uma equipa de um grupo de trabalho orientada pelo Professor Doutor Jorge Sá Silva e pelo Professor Doutor André Rodrigues. O grupo é formado por alunos de mestrado e doutoramento de engenharia eletrotécnica e computadores e de engenharia informática. Este projeto contou também com o grupo de psiquiatria da Faculdade de Medicina da Universidade de Coimbra.

O grupo de trabalho divide-se em várias equipas, de forma a desenvolver diferentes projetos. Atualmente, podem ser destacadas três aplicações: ISABELA - com o objetivo de monitorizar e melhorar o desempenho académico dos estudantes, MYPOC - com o objetivo de monitorizar doentes com Perturbação Obsessiva Compulsiva (POC) e IFRIEND - com o objetivo de monitorizar o estado de saúde de pessoas idosas. Estas três aplicações baseiam-se essencialmente no paradigma da IoT e de novas tecnologias como forma de monitorizar um conjunto de pessoas. Para além de terem objetivos comuns para diferentes grupos de pessoas, estas aplicações também se aproximam no facto de utilizarem as mesmas bibliotecas de recolha de dados e comunicação com a *FIWARE* (será explicada futuramente, na secção 2.2.5), também desenvolvidas por alunos do grupo de trabalho, e ainda por todas utilizarem a *cross-platform Xamarin*.

O projeto no qual esta tese se insere, a plataforma ISABELA, teve como principal objetivo recriar a aplicação de modo a ser também compatível com SO *iOS*, utilizando a estrutura já definida inicialmente para *Android*, uma vez que a aplicação já estava estruturada para esse sistema operativo. Para tal, embora já tenham colaborado outros membros, atualmente estão destacados para este projeto sete alunos, com diferentes tarefas. Nesta tese, são apresentados os dados relativos à tarefa de desenvolvimento da aplicação, desde a parte do *back-end* até ao *front-end*, utilizando *C#* e *XAML*. O Miguel Monteiro, trabalha na criação de uma biblioteca dinâmica, dedicada à construção da parte visual dos formulários disponibilizados na plataforma. O Afonso Lemos está designado para o desenvolvimento de um *chatbot*. A Sandra Aguilar está encarregue do tratamento e análise dos dados recolhidos pela aplicação e o Jorge Rivadeneira, juntamente com o João Rodrigues, estão reponsáveis pela privacidade e segurança. Por fim, o Marcelo Fernandes é o responsável pelo planeamento da aplicação e ajuda em todas as áreas anteriormente referidas.

Todas as semanas a equipa de trabalho reúne, via *Skype*, com o objetivo de partilhar progressos e possíveis ideias futuras/novas implementações. Para além destas reuniões, são

marcadas também reuniões extraordinárias, quando necessárias, apenas com a equipa de trabalho, de forma a expor e colmatar dúvidas.

1.5 Estrutura do documento

Este documento é agrupado em 5 capítulos: Introdução, Estado da Arte, Aplicação ISABELA, Avaliações e Testes, Conclusões e Trabalho Futuro.

O Capítulo 1 contextualiza o projeto, as suas motivações e objetivos. Neste capítulo estão também presentes metodologias.

No Capítulo 2 é apresentado o estado da arte, onde são abordados conceitos e tecnologias base utilizadas, assim como plataformas semelhantes já existentes. Por fim é feita uma análise comparativa da plataforma ISABELA com as plataformas semelhantes.

No Capítulo 3 o desenvolvimento da aplicação ISABELA é descrito detalhadamente. De seguida, no Capítulo 4 são apresentadas avaliações e testes.

Por fim, são apresentadas conclusões e desafios futuros no Capítulo 5.

2

Estado da Arte

Conteúdos

2.1	Conceitos base	7
2.2	Tecnologias base utilizadas	10
2.3	Plataformas atuais - ISABELA/BATINA	13
2.4	Plataformas semelhantes	15
2.5	Análise comparativa	18

2.1 Conceitos base

Nesta secção, serão abordados conceitos e tecnologias base utilizadas na implementação da plataforma ISABELA. Serão analisadas as plataformas já existentes: ISABELA/BATINA, desenvolvidas para *Android*, assim como plataformas semelhantes, com o objetivo de melhorar o rendimento e desempenho académico. Por fim, será feita uma análise comparativa das plataformas já existentes com a plataforma ISABELA/BATINA.

2.1.1 Internet das Coisas

O conceito de Internet das Coisas foi proposto em 1999, por Kevin Ashton, que definiu a IoT como um conjunto de objetos conectados, com interoperabilidade e com identificação por rádio frequência (*Radio-Frequency Identification (RFID)*). Atualmente, o conceito evoluiu e já não está associado a RFID, sendo agora definida como uma infraestrutura de rede dinâmica, com a capacidade de se auto-configurar, devido à utilização de protocolos de comunicação padronizados e da sua interoperabilidade. “Coisas” físicas e virtuais possuem entidades e atributos de forma a serem capazes de utilizar interfaces inteligentes e serem integradas como uma rede de informação [10].

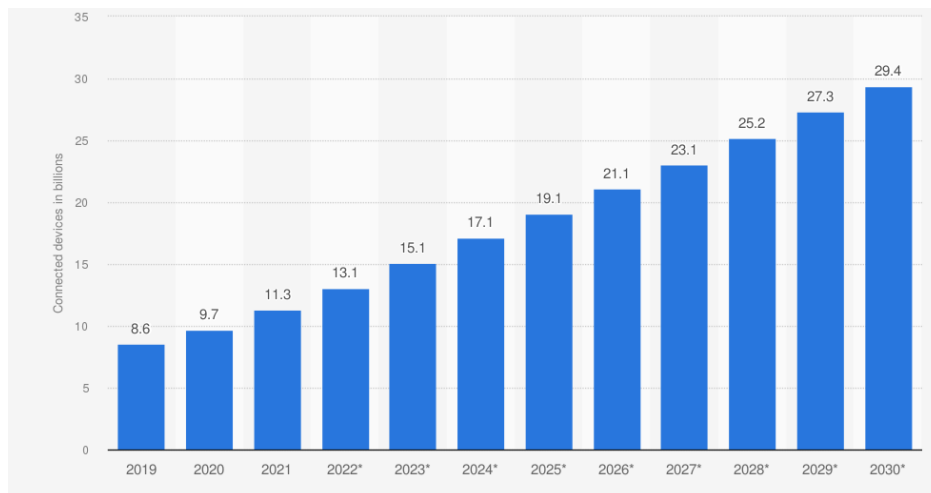


Figura 2.1: Previsão do número de dispositivos IoT conectados em todo o mundo desde 2019 a 2030 [1].

Atualmente existem cerca de 13,1 mil milhões de dispositivos IoT conectados em todo o mundo. Com a evolução da Internet, não só em computadores mas também em *tablets*, *smartphones*, *smart devices*, estima-se que este valor triplique em 2030, atingindo um valor de 29,4 mil milhões [1].

2.1.2 *Cyber-Physical System*

Um CPS consiste num sistema complexo que integra sensores e atuadores, computação, sistemas de controlo e redes de comunicação em processos físicos. Este sistema pode ser diferenciado em dois sub-sistemas: físico e cibernético. A sua arquitetura geral é representada na Figura 2.2. O sistema físico consiste em processos físicos, sensores e atuadores. Já o sistema cibernético engloba redes de comunicação e centros de controlo. Desta forma, é capaz de expandir o mundo virtual para o mundo físico, fornecendo serviços de processamento de dados em tempo real. Mais especificamente, um CPS permite que um sistema físico seja equipado com um sistema virtual como controlador, permitindo que os dados recolhidos do mundo físico sejam analisados no mundo virtual. Deste modo, é possível que decisões possam ser tomadas de maneira a influenciar as ações do mundo físico. [11].

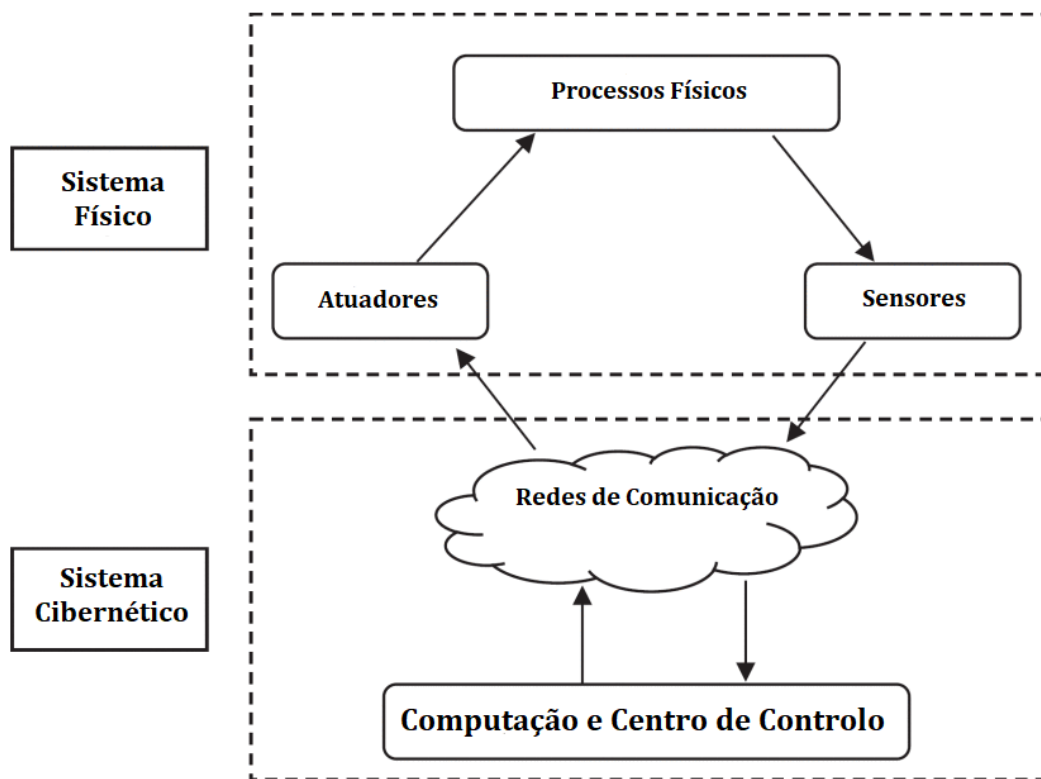


Figura 2.2: Arquitetura de um sistema ciber-físico. Adaptado de [11].

Embora o seu conceito seja semelhante ao de IoT, os CPS estão mais relacionados com a aplicabilidade e integração da parte física, enquanto que o IoT se encarrega mais especificamente da infraestutura de rede e integração de diversos dispositivos.

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

Muitos dos CPS, embora sejam aplicações centradas no Ser Humano, não o consideram como um elemento ativo do sistema, desconsiderando as suas ações, emoções, contexto social.

Com a evolução dos CPS, surgiram os HITLCPs, que têm em conta o fator Humano, sendo muito mais fiáveis e aptos para mecanismos de monitorização e controlo.

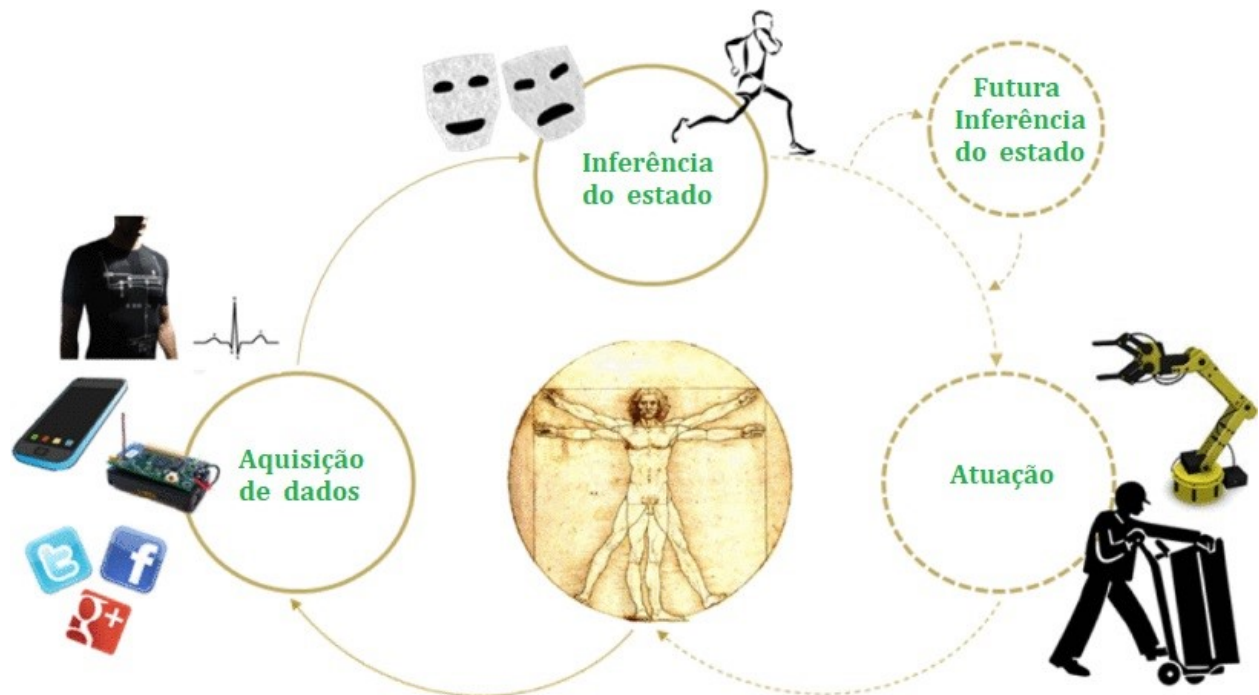


Figura 2.3: Processo de controlo de um sistema *Human-in-the-Loop*. Adaptado de [12].

O processo de controlo de um sistema HITL é caracterizado pela Figura 2.3, representado por quatro principais fases: Aquisição de dados, inferência do estado, futura inferência do estado e atuação [12].

- Aquisição de dados - através dos sensores, presentes nos dispositivos móveis. Esta tarefa atualmente é bastante simplificada, devido ao acesso a um *smartphone*, que cada vez mais apresenta uma maior capacidade sensorial.
- Inferência de estado - depois de recolhidos, os dados são processados de modo a inferir o estado do utilizador.
- Futura inferência do estado - para além da inferência anterior é também calculada uma previsão do estado futuro do utilizador.
- Atuação - por fim, a informação recolhida e processada pelo sistema permite que o sistema determine se alguma ação pode ser efetuada.

Através deste sistema de controlo é possível auxiliar e monitorizar o Ser Humano eficientemente no seu dia a dia.

2.1.4 Importância de dados no século XXI e privacidade

Atualmente, devido ao impacto causado pelo crescimento da IoT, IA e *Big Data*, existem cada vez mais serviços *online*, que fazem parte do dia a dia do Ser Humano. Devido à utilização destes serviços, o volume de dados relativos a cada pessoa tem crescido substancialmente, sendo importante proteger a informação digital de ameaças, através de técnicas de segurança capazes de proteger a informação de possíveis ameaças a bancos de dados [13].

A temática da proteção de dados tem tido extrema relevância no século XXI. Segundo a Carta dos Direitos Fundamentais da União Europeia, todas as pessoas têm direito à proteção dos dados de carácter pessoal a elas associados [14]. Para regulamentar esta proteção, a 25 de maio de 2018 entrou em vigor o Regulamento Geral Sobre Proteção de Dados (RGPD) da União Europeia, que estabelece regras relativas ao tratamento de dados pessoais relativos a pessoas na União Europeia [15].

2.2 Tecnologias base utilizadas

Nesta secção serão apresentadas tecnologias base utilizadas, necessárias para o desenvolvimento desta tese, tais como *Android*, *iOS*, *.NET*, *Xamarin*, *FIWARE*.

2.2.1 *Android*

O *Android* é um SO desenvolvido pela *Google* e baseado em *Linux Kernel* [16]. É um sistema *open-source*, permitindo que fabricantes de telemóveis utilizem e adaptem o SO para os seus próprios dispositivos. O facto de ser *open-source* contribui também para que o sistema possa ser melhorado. As razões anteriormente descritas são o principal motivo pelo qual o SO *Android* é líder de mercado [17]. A principal linguagem de programação utilizada no desenvolvimento das suas aplicações é *Java*, utilizando *Software Development Kit* (SDK)s, com ferramentas que facilitam e permitem o desenvolvimento das aplicações em SO *Android*. Existe também a possibilidade de utilizar outra linguagens de programação, como *C/C++*, quando se pretende desenvolver aplicações de baixo nível. Para esta segunda opção utiliza-se NDKs (Native Development Kits) [16].

2.2.2 *iOS*

O iOS é um SO desenvolvido pela *Apple*. Inicialmente foi apenas desenvolvido para *iPhone* e posteriormente foi expandido, de forma a ser implementado em outros dispositivos da *Apple*.

Este SO é derivado de vários elementos do MAC OS S (SO semelhante ao *Unix*). De forma a desenvolver aplicações compatíveis com *iOS*, os programadores podem utilizar SDKs do *iOS*, que incluem ferramentas e interfaces capazes de desenvolver, instalar, executar e testar aplicações. Aplicações nativas podem ser desenvolvidas utilizando a linguagem de programação *Objective-C*.

Uma das principais vantagens e razões para o *iOS* ser dos mais procurados é o facto de a *Apple* permitir a atualização do SO para dispositivos mais antigos, embora este suporte esteja a diminuir cada vez mais [18].

2.2.3 *.NET*

.NET framework é um conjunto de ferramentas de desenvolvimento de aplicações *cross-platform* e oferecidas no modelo *open-source*. Dotada de várias ferramentas, bibliotecas e *Application Programming Interface* (API)s, com diferentes funcionalidades, permite facilitar o desenvolvimento de diversas aplicações, tais como aplicações móveis, IoT, jogos, e lidar automaticamente com tarefas como alocação de memória e interoperabilidade. Esta plataforma está preparada para lidar com três linguagens de programação: *C#*, *Visual Basic* - orientadas a objetos e *type-safe* (previnem erros de escrita) e *F#*, linguagem útil para escrever código sucinto mas robusto e com boa performance [19].

2.2.4 *Xamarin*

Xamarin estende a plataforma *.NET* com ferramentas e bibliotecas, oferecidas no modelo *open-source*, que permitem o desenvolvimento de aplicações compatíveis com SO *iOS* e *Android*. Esta *cross-platform* permite que os programadores partilhem cerca de 90% do código, sendo possível reutilizar código existente, assim como programar toda a parte lógica do código numa única linguagem (*C#*) [20]. A sua arquitetura geral encontra-se representada na Figura 2.4.

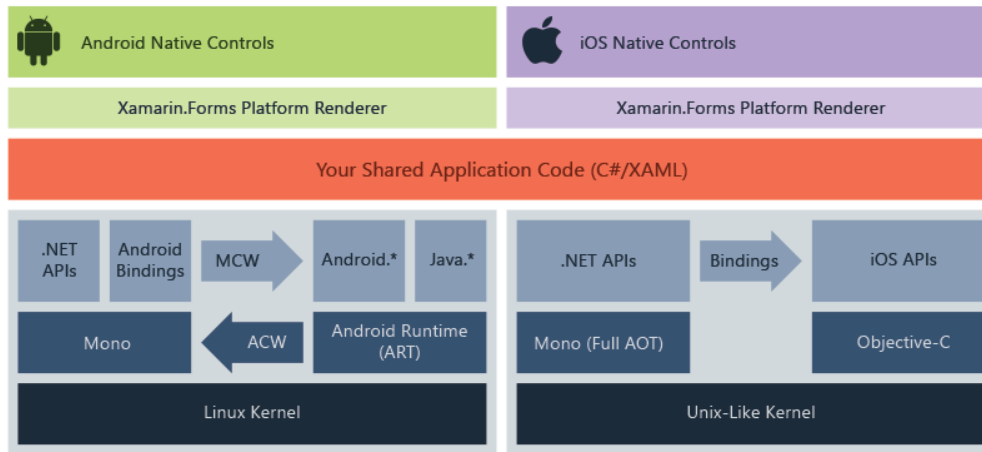


Figura 2.4: Diagrama da arquitetura geral de uma aplicação em *Xamarin* [21]

No entanto, o *Xamarin* nativo não permite a partilha da interface do utilizador. Por este motivo, foi desenvolvida uma *UI framework* - *Xamarin.Forms* que, para além de possibilitar o desenvolvimento da parte lógica de uma aplicação através de um código 90% partilhado, permite ainda que uma aplicação partilhe uma interface de utilizador, utilizando a linguagem de marcação *XAML* [21].

2.2.5 FIWARE

A *FIWARE* é um projeto europeu suportado e financiado pela *Future Internet Public-Private Partnership* (FI-PPP) [22], que tem como objetivo principal oferecer um conjunto de elementos *open-source* de forma a ajudar nas implementações e facilitar alguns desafios da IoT. Para tal, esta plataforma IoT garante modularidade, máxima interoperabilidade e ainda proteção de dados [23].

A *FIWARE* é baseada numa biblioteca de componentes, chamados *Generic Enablers* (GE), que implementam APIs, de forma a facilitar o desenvolvimento de aplicações inteligentes IoT.

Os GEs implementam funções como: [24]:

- Gestão, análise, processamento e visualização de dados - facilitando a recolha e processamento de dados, assim como a publicação e monetização dos mesmo em grande escala.
- Interfaces de dispositivos IoT, redes e robôs - permitindo aproveitar ao máximo as capacidades da rede, explorando comunicações eficientes.
- Segurança - garantindo proteção de dados.

- Armazenamento numa *cloud* - com componentes que permitem fornecer computação e armazenamento de dados.

2.3 Plataformas atuais - ISABELA/BATINA

ISABELA - *IoT Student Advisor and BEst Lifestyle Analyzer* é uma aplicação compatível com SO *Android*, desenvolvida por um grupo de investigação da Faculdade de Ciências e Tecnologia da Universidade de Coimbra (FCTUC) em parceria com outras faculdades.

O principal objetivo do projeto foi criar um sistema HITLCPS com o intuito de atuar sobre alunos universitários, proporcionando-lhes auxílio e monitorização, assegurando um aumento do seu desempenho académico.

A plataforma ISABELA, sendo um sistema HITLCPS, tem a capacidade de inferir e processar dados. No entanto, para efetuar esta inferência e processamento, necessita, primeiramente, de os recolher. A recolha de dados é efetuada através de vários sensores embutidos nos *smartphones*, nomeadamente acelerómetro, giroscópio, WiFi, sensor de luz, sensor de proximidade, entre outros sensores essenciais à inferência da informação. São também adquiridos dados de redes sociais (como o *Facebook*).

Para além dos dados recolhidos de forma indireta, isto é, sem ser necessária nenhuma ação do utilizador, são também disponibilizados na aplicação formulários relativos ao contexto social, qualidade de sono, contexto físico e qualidade de estudo, aos quais o estudante deve responder diariamente ou quando pedido pela aplicação, através de notificações [25]. Adicionalmente, o utilizador pode ainda interagir com um *chatbot*, que para além de auxiliar o utilizador em tempo real, contribui também para a inferência de dados, interpretando o diálogo do mesmo.

Na Figura 2.5 podem ser observados alguns *layouts* da aplicação ISABELA.

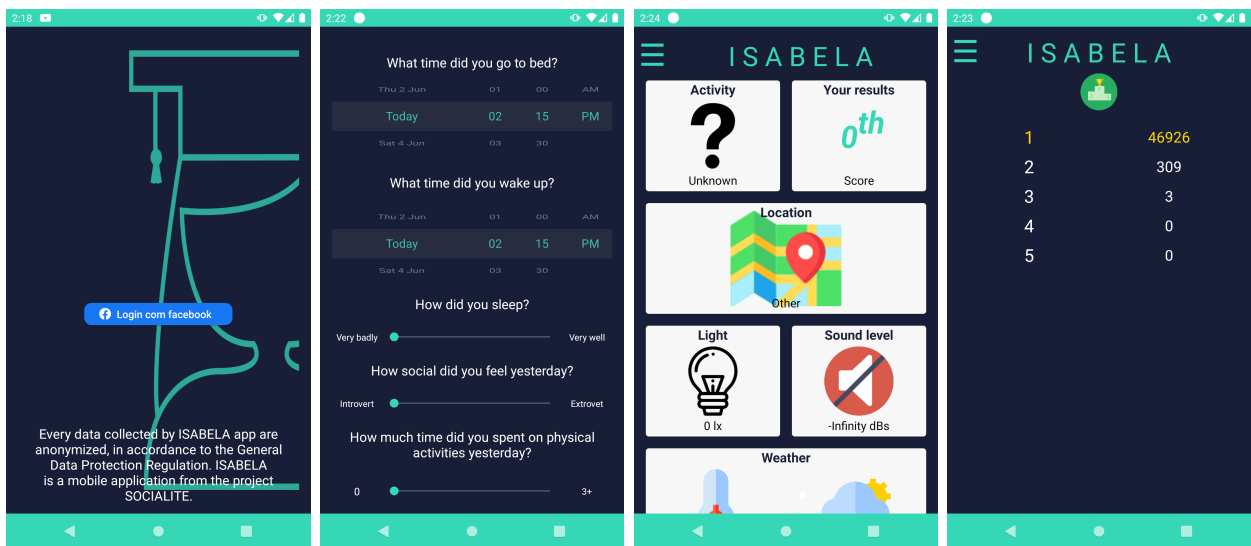


Figura 2.5: Alguns layouts da aplicação ISABELA.

Foi também desenvolvida uma aplicação paralela - BATINA que consiste numa versão reduzida da ISABELA. Esta aplicação visa principalmente facilitar a interação professor-aluno, permitindo que o professor disponibilize questionários na própria aplicação, aos quais o aluno deve responder. O BATINA é também dotado de um *chatbot* e de algumas páginas informativas, relativamente ao desempenho do aluno.

Na Figura 2.6 é possível visualizar alguns *layouts* da aplicação BATINA.

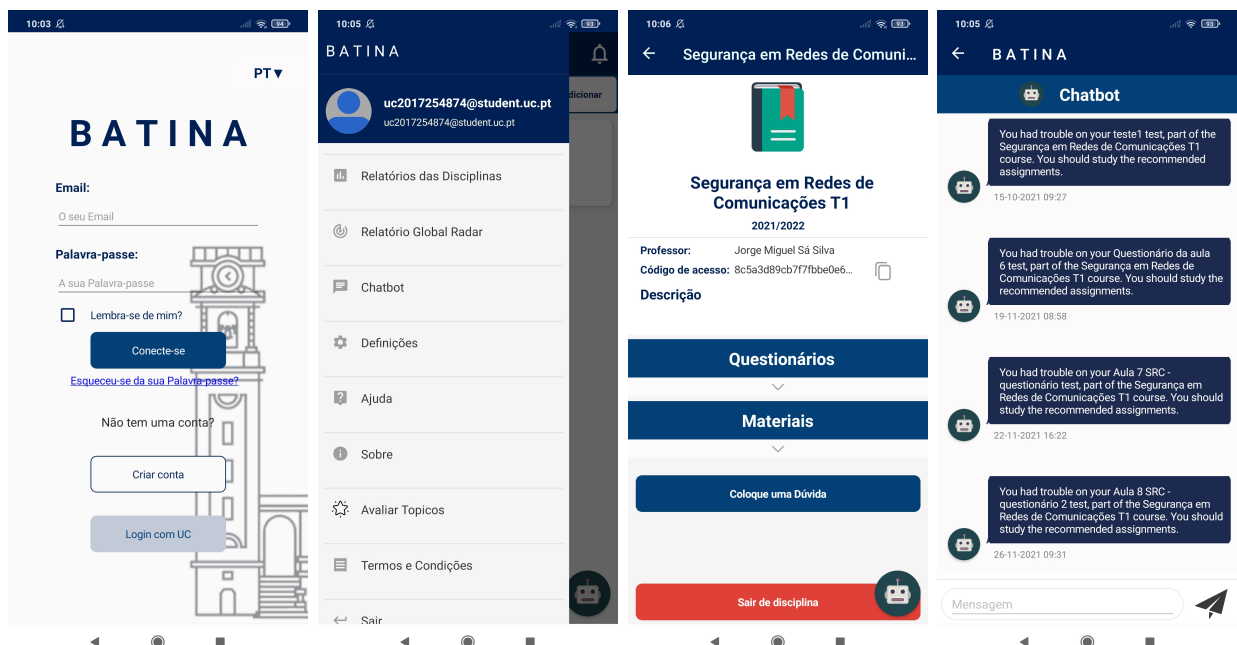


Figura 2.6: Alguns layouts da aplicação BATINA.

2.4 Plataformas semelhantes

Nesta secção será feita uma análise comparativa de plataformas semelhantes à plataforma ISABELA/BATINA. Para tal serão avaliadas plataformas centradas em dispositivos móveis com o principal objetivo de melhorar a qualidade do ensino e desempenho académico.

2.4.1 *StudentLife*

StudentLife foi um estudo realizado por uma equipa de investigação da Universidade de Dartmouth que pretendia avaliar a saúde mental, desempenho académico e tendências comportamentais dos estudantes. Para tal, este estudo contou com a presença de 48 estudantes e foi realizado ao longo de 10 semanas.

Para efetuar este estudo, foi desenvolvida uma aplicação compatível com SO *Android* utilizada para recolher dados dos estudantes de forma passiva (através dos sensores embutidos no *smartphone*), tais como acelerómetro, sensor de luz, entre outros [26].

O processo de aquisição e processamento de dados é caracterizado na Figura 2.7. Através da correlação dos dados recolhidos juntamente com algum *feedback* proporcionado diretamente pelo estudante através do preenchimento de formulários, foi possível inferir resultados relativos ao contexto social, comportamental, emocional e ainda dados relativos ao nível e qualidade de sono [27].

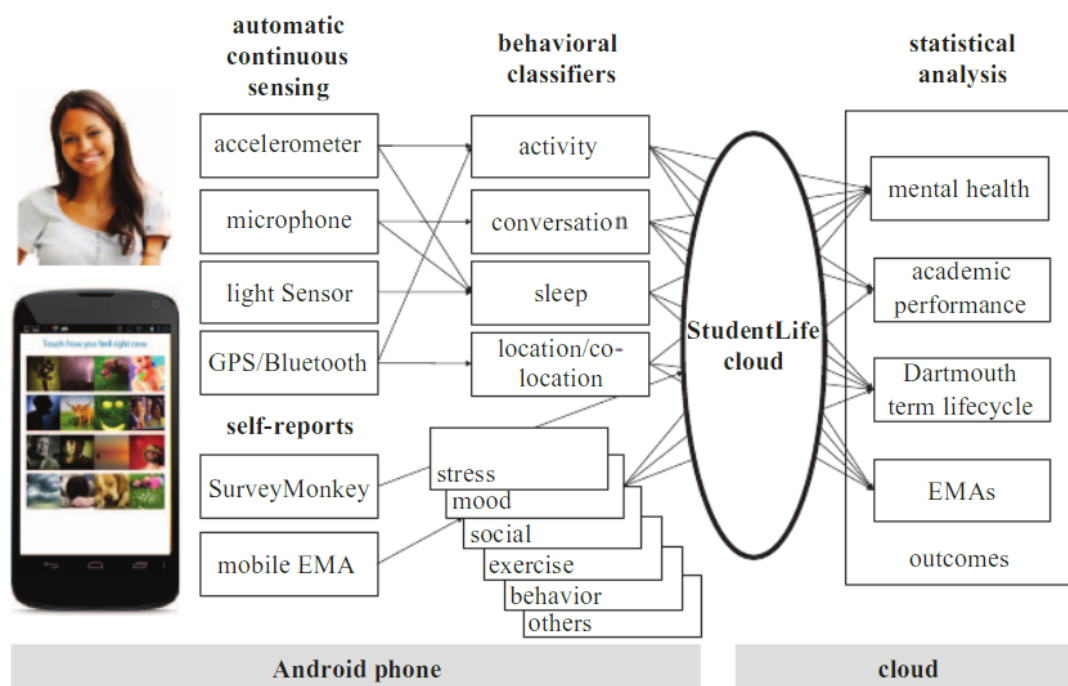


Figura 2.7: Arquitetura do projeto *StudentLife* [27].

2.4.2 *BigBlueButton*

BigBlueButton é uma plataforma *open-source* de apoio à aprendizagem desenvolvida em 2017 pelo Instituto de Empreendedorismo e Comercialização de Tecnologia da Universidade de Carleton em Ottawa, Canadá.

Esta plataforma tem o objetivo de melhorar e apoiar o ensino *online*. Constituída por mecanismos *Learning Management System* (LMS), disponibiliza uma série de recursos, síncronos e assíncronos, facilitando a interação dos professores com os alunos e também entre alunos. Alguns exemplos destes recursos são salas virtuais, *chats*, um quadro virtual, possibilidade de partilhar ecrã, partilhar *emojis* representativos de *feedback* por parte do aluno [28]. A página inicial da aplicação e as suas principais ferramentas podem ser visualizadas na Figura 2.8.

Devido a estas características, o *BigBlueButton* foi profundamente incorporado em bastantes sistemas de gestão de aprendizagem, tendo sido adotado por *Canvas*, *Sakai Jenzabar*, *Moodle Cloud*, *D2L* e *Schoology* como ferramenta padrão de aulas *online* [28].

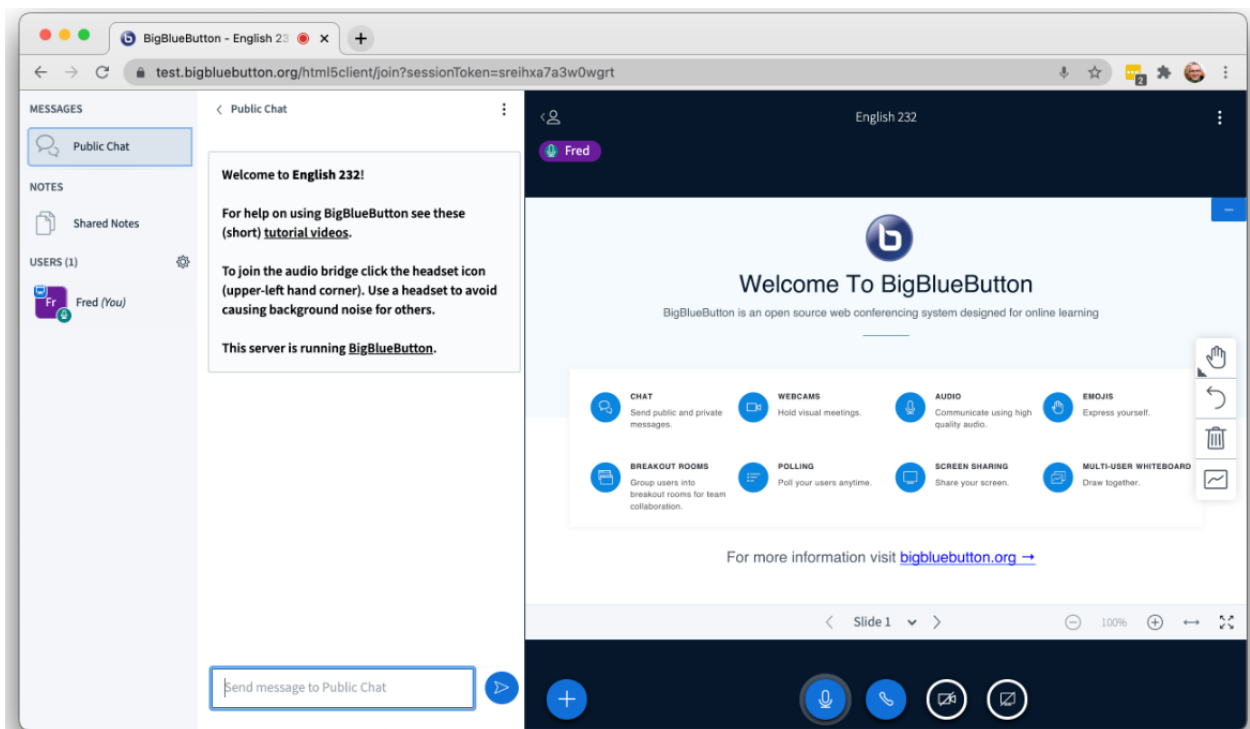


Figura 2.8: Página inicial da plataforma *BigBlueButton* [28].

2.4.3 Aleks

Aleks é uma plataforma virtual constituída por várias módulos de inteligência artificial e *Big Data*, aperfeiçoada por mais de duas décadas, com o principal objetivo de avaliar e ensinar um estudante, tendo em conta os seus conhecimentos prévios duma determinada matéria.

Quando o aluno pretende aprender uma nova matéria, a plataforma começa por avaliar e determinar o seu nível de conhecimento nessa área. Esta avaliação é individual e única de estudante para estudante. Depois dessa análise, a *Aleks* ajuda o estudante a aprender os tópicos que deseja, através de um caminho individual e dinâmico para o sucesso. A progressão do ensino é feita tendo em conta um ciclo de domínio e retenção de conhecimento e *feedback* positivo por parte do utilizador, garantindo que este apenas progride num determinado tema quando as suas bases estão bem consolidadas [29].

A *Aleks* fornece também dados em tempo real e relatórios detalhados relativamente ao progresso de aprendizagem de cada estudante, como se pode visualizar na Figura 2.9.

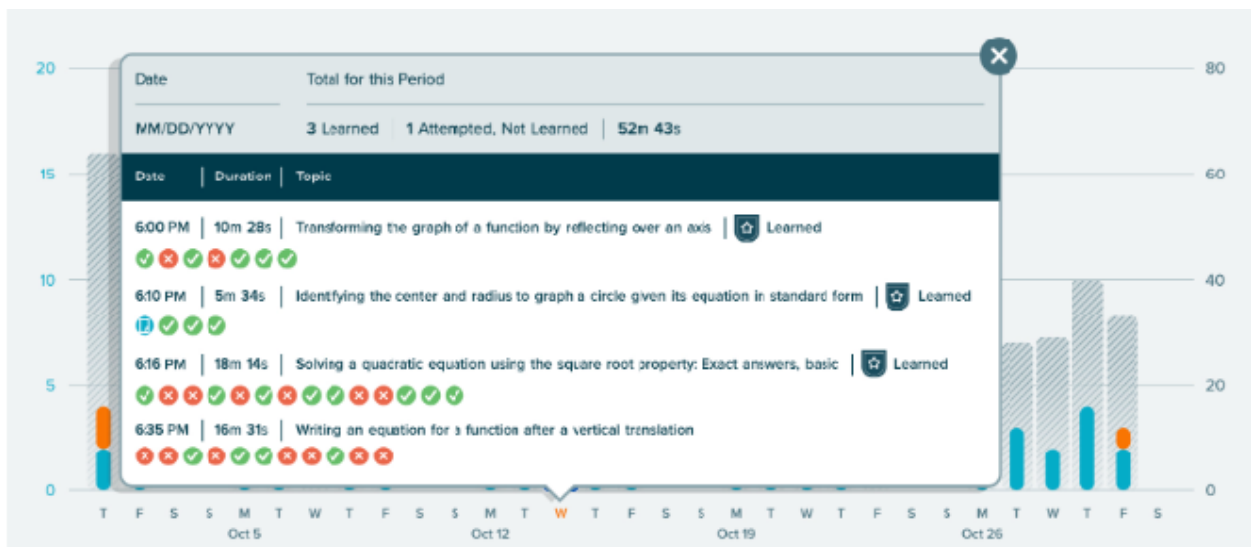


Figura 2.9: Relatório de progresso da plataforma *Aleks* [29].

2.4.4 Socrative

Socrative é uma aplicação gratuita, simples e intuitiva utilizada principalmente em sala de aula. Permite a elaboração de perguntas de resposta curta, múltipla ou verdadeiro/falso (por parte de um professor) e respostas (por parte dos alunos). Para haver distinção entre professor e alunos, a plataforma disponibiliza duas formas de *login*: *login de professor* - onde é disponibilizado o menu inicial representado na Figura 2.10 (com um código de sala definido) e *login de aluno* - que através da inserção do código, o aluno tem acesso às perguntas efetuadas pelo professor [30].

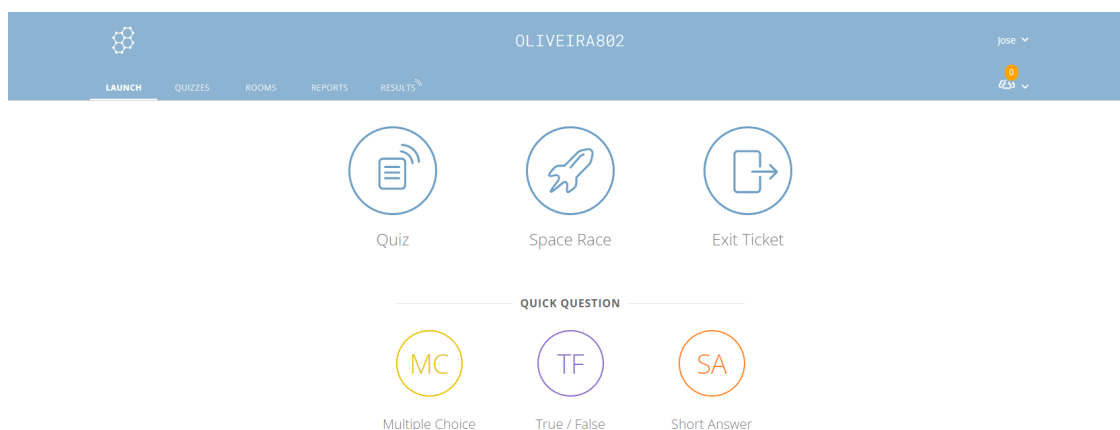


Figura 2.10: Menu inicial da plataforma socrative para professores [30].

Através destas perguntas, variando entre formulários de preparação para avaliações, *quizzes* ou até mesmo avaliações, é possível que um professor receba *feedback* em tempo real da aprendizagem/desempenho de um aluno. Em simultâneo, o aluno também tem acesso ao seu desempenho, tendo assim a oportunidade de imediatamente saber os tópicos com mais dificuldade, favorecendo o seu estudo e consequentemente melhorando o seu desempenho académico.

2.5 Análise comparativa

Comparando as aplicações anteriormente abordadas com a plataforma ISABELA/BATINA, é possível concluir que esta foi desenvolvida tendo por base o estudo *StudentLife*, uma vez que este utiliza os sensores embutidos nos *smartphones* como forma de inferir resultados relativamente a contexto social, físico e emocional. No entanto, a plataforma ISABELA/BATINA

pretende dar *feedback* ao utilizador, tendo em conta os dados analisados, oferecendo auxílio e acompanhamento ao estudante, de modo a melhorar o seu desempenho académico. Outra diferença é a utilização de um *chatbot*, melhorando a interação entre o aluno e a plataforma.

Juntamente à utilização de sensores, a aplicação ISABELA disponibiliza formulários, aos quais o aluno deve responder na própria aplicação. Já o BATINA explora a elaboração de formulários e *quizzes* (de teste ou de avaliação), tal como foi explorado na aplicação *Socratic*.

Relativamente às aplicações *Alek*, a utilização de inteligência artificial para processar e inferir dados, assim como mecanismos de LMS robustos, podem contribuir para uma plataforma de ensino mais completa, com maior facilidade de inferência de dados e monitorização de estudantes. A plataforma ISABELA/BATINA pode ainda ser expandida de forma a obter mecanismos de LMS, como apresenta a aplicação *BigBlueButton*.

Em suma, a plataforma ISABELA/BATINA é composta por várias funcionalidades das aplicações anteriormente exploradas, tais como processamento e inferência de dados através de inteligência artificial e dos sensores embutidos nos *smartphones* e disponibilização de diversos formulários. No entanto, a diferença mais relevante é a utilização de um sistema HITLCPS onde é apresentado *feedback* ao utilizador e ainda a interação com um *chatbot*, permitindo um sistema conselheiro muito mais eficiente.

3

Aplicação ISABELA

Conteúdos

3.1	Contextualização	21
3.2	Descrição	21
3.3	Arquitetura Geral	22
3.4	Arquitetura da aplicação móvel	24
3.5	Requisitos	25
3.6	Desenvolvimento	28

3.1 Contextualização

Como dito anteriormente, a aplicação ISABELA tem como principal objetivo monitorizar, auxiliar e melhorar o desempenho acadêmico dos estudantes. Com esta aplicação é também possível compreender melhor o dia a dia dos alunos e tentar prevenir comportamentos que possam levar a resultados acadêmicos negativos ou até mesmo a desistência acadêmica.

Atualmente, umas das principais causas de mau desempenho acadêmico e abandono escolar dos estudantes é a depressão [31]. Com a pressão e exigência associadas ao ensino superior, são vários os fatores que podem contribuir para a depressão de estudantes universitários, tais como fatores físicos, não fazer exercício físico, isolamento social, baixa qualidade/quantidade de sono ou até falta de organização e de método de estudo. Através do controlo destes fatores, monitorizando-os de forma a evitá-los, seria possível alcançar melhores resultados académicos e ainda facilitar a aprendizagem do aluno [31].

3.2 Descrição

A aplicação ISABELA é baseada num sistema HITLCPS com o intuito de monitorizar e auxiliar estudantes ao longo do seu dia a dia. Através dos sensores incutidos nos *smartphones* e com a permissão prévia do utilizador (apenas pedida na primeira vez que a aplicação é utilizada), esta aplicação recolhe dados (tais como nível de temperatura, humidade, ruído, valores de acelerómetro e barómetro, etc.) de forma a inferir o comportamento, contexto social, físico e emocional do utilizador. A aplicação disponibiliza também questionários aos estudantes, de modo a facilitar a interpretação dos dados. Esta inferência é conseguida devido ao processamento de dados que é efetuado através de componentes da *FIWARE*. Uma vez que é necessário comunicar com a *FIWARE*, estaria implícita a necessidade de estabelecer uma conexão à *Internet*. No entanto, a aplicação não necessita de realizar esta conexão imediatamente, guardando os dados recolhidos localmente, no *smartphone*, enquanto aguarda que ocorra a ligação com a *FIWARE*. Desta forma, é garantido que não ocorre perda de dados recolhidos. Posteriormente, depois de analisados os dados, estes são mostrados ao utilizador.

A aplicação garante também a possibilidade de comunicação com um *chatbot*, que, para além de responder a perguntas efetuadas pelo estudante, pode ter também um comportamento de antecipação, escrevendo uma mensagem ao utilizador quando é detetado um comportamento que possa implicar um mau desempenho académico.

3.3 Arquitetura Geral

A ISABELA é uma aplicação móvel desenvolvida com o objetivo de auxiliar e monitorizar os estudantes de modo a melhorar o seu desempenho académico. A sua arquitetura geral encontra-se representada na Figura 3.1, caracterizada por vários mecanismos que permitem implementar um sistema HITLCPS, tais como: aquisição e processamento de dados, de modo a que seja feita uma inferência, uma inferência futura e atuação.

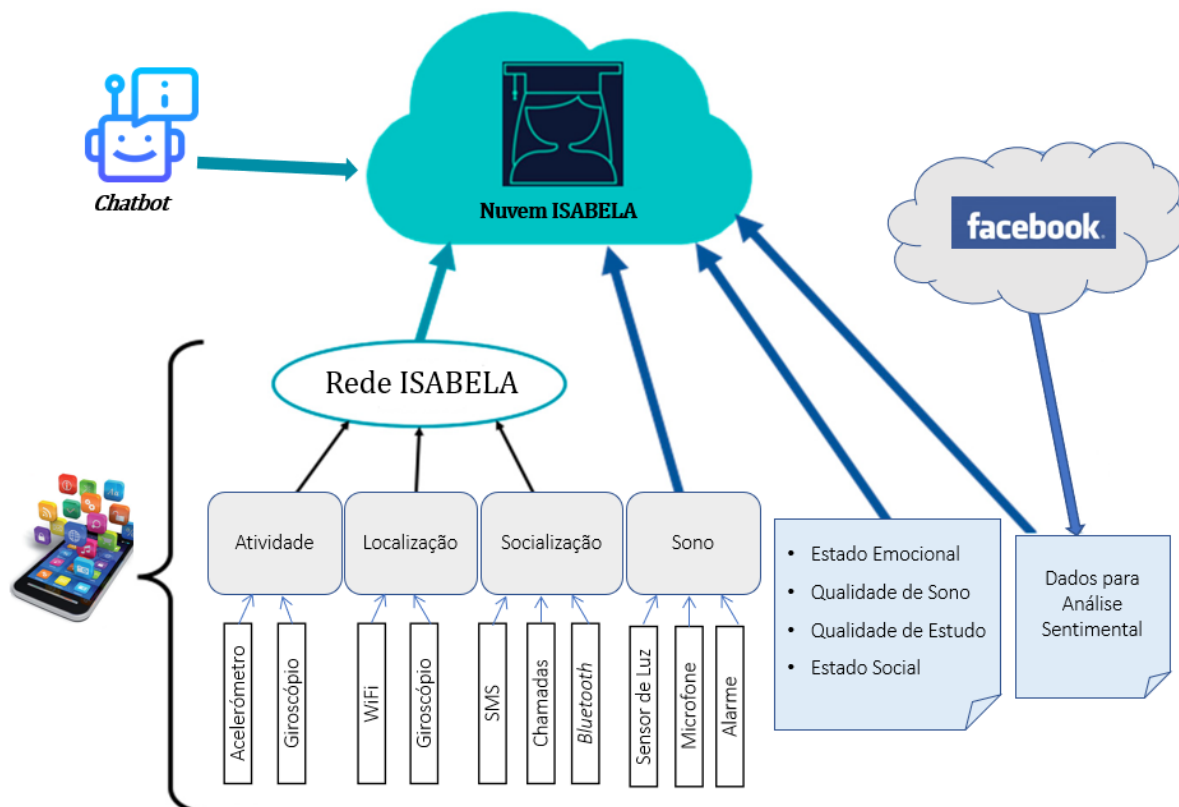


Figura 3.1: Arquitetura de aquisição de dados e processamento do ISABELA. Adaptado de [25].

Um sistema HITLCPS necessita de ser escalável, adaptável e dinâmico. Como forma de alcançar estas necessidades, a plataforma ISABELA utiliza componentes da *FIWARE*, *Generic Enablers* (GEs), que adotam o modelo de informação NGSI9/10. O modelo é baseado em entidades e atributos. Cada entidade possui seu próprio tipo e é representada por atributos, utilizando o formato *JSON*. A utilização da *FIWARE* oferece também um padrão aplicável às plataformas IoT na Europa. A arquitetura da Nuvem ISABELA, que utiliza componentes da *FIWARE*, é caracterizada pela Figura 3.2.

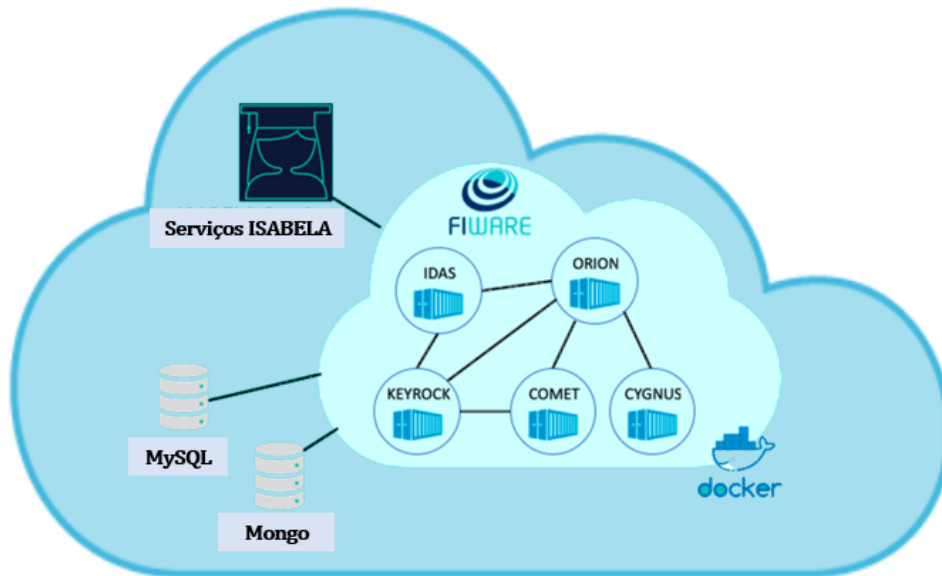


Figura 3.2: Nuvem ISABELA. Adaptado de [25].

Nesta implementação são utilizados cinco GEs: *ORION*, *CYGNUS*, *STH-COMET*, *IDAS* e o *KEYROCK*, com diferentes funções [32]:

- *IDAS* - utilizado para gestão de todos os tipos de dispositivos IoT. Este GE suporta recursos de gestão e interoperabilidade entre dispositivos IoT e a plataforma *FIWARE*.
- *ORION* - gere todas as entidades da aplicação. Através desta API é possível inserir/excluir/recuperar entidades, assim como atualizar as existentes. Todos os dados enviados por dispositivos IoT são associados às entidades armazenadas no *ORION*. No entanto, este GE não garante o armazenamento de dados históricos.
- *CYGNUS* - este GE é responsável por gerir o armazenamento dos dados, permitindo criar entidades para um sistema de armazenamento de terceiros específico, como *MongoDB* ou *MySQL*, possibilitando o armazenamento histórico dos dados.
- *STH-COMET* - Uma vez que as bases de dados, por norma, não fornecem APIs para troca de dados entre elas e as aplicações, é necessário utilizar o GE *STH-COMET*, ou *Short-Term History* que fornece uma *RESTful* API com a capacidade de guardar dados históricos e agregação de métodos. Sempre que a aplicação ISABELA necessita de aceder a dados históricos, este GE permite a conexão entre os dados guardados nas bases de dados e a aplicação.
- *KEYROCK* - este GE permite adicionar autenticação e segurança aos dispositivos IoT, às suas aplicações e ainda aos utilizadores. Para além disso, de forma a atender a todos

os requisitos das leis de privacidade da UE e do Regulamento Geral de Proteção de Dados, todos os dados tratados pela aplicação são anonimizados [33].

Para além da nuvem composta por componentes da *FIWARE*, a aplicação conta ainda com dois serviços: análise do sono - que é responsável por monitorizar o nível/qualidade de sono do utilizador, com base nos dados coletados por meio dos sensores do *smartphone* - e *Natural Language Processing* (NLP) que utiliza os dados provenientes das redes sociais (como o *Facebook*) para detetar sentimentos e emoções.

3.4 Arquitetura da aplicação móvel

A arquitetura da aplicação móvel segue os padrões comuns de aplicações móveis, sendo definida por *threads*. Tal como caracterizado na Figura 3.3, existe uma *thread* que trata da parte do *User Interface* (UI) (*UI Thread*) e outro conjunto de *threads*, executados em segundo plano, que permitem lidar em simultâneo com outras operações, de modo a que a aplicação funcione corretamente, com fluidez e rapidez.

O *UI thread* é a *thread* principal de execução da aplicação. Esta é responsável por toda a interação visual do utilizador com a aplicação, como por exemplo alterar as configurações, mostrar uma janela *pop-up* ou executar eventos quando um botão é primido, etc. É nesta *thread* que a maior parte do código é executado, sendo também responsável pela inicialização de *threads* em segundo plano.

As *threads* em segundo plano são responsáveis por executar serviços mais complexos, como por exemplo comunicação *Hypertext Transfer Protocol* (HTTP), aquisição, armazenamento e recuperação de dados. Na arquitetura ISABELA existem dois serviços principais, tal como representados na Figura 3.3. O *Serviço principal* é responsável pela aquisição de dados através dos sensores do *smartphone* e o *Serviço de comunicação com a FIWARE* é responsável pelo envio de informação para a *FIWARE*.

O *chatbot* é utilizado para processar mensagens enviadas pelo utilizador, enviá-las para o *Dialogflow*, onde é feita a comunicação com a *FIWARE*, e são retirados os dados que o utilizador solicitar.

A comunicação com a *FIWARE* necessita de ligação à *Internet* para que seja bem sucedida. Desta forma, é utilizada uma base de dados, onde todos os dados adquiridos são guardados caso não exista conexão à *Internet* no momento da sua aquisição, para que não haja perda de dados. Posteriormente, quando ocorrer ligação à *Internet* os dados são enviados para a *FIWARE*.

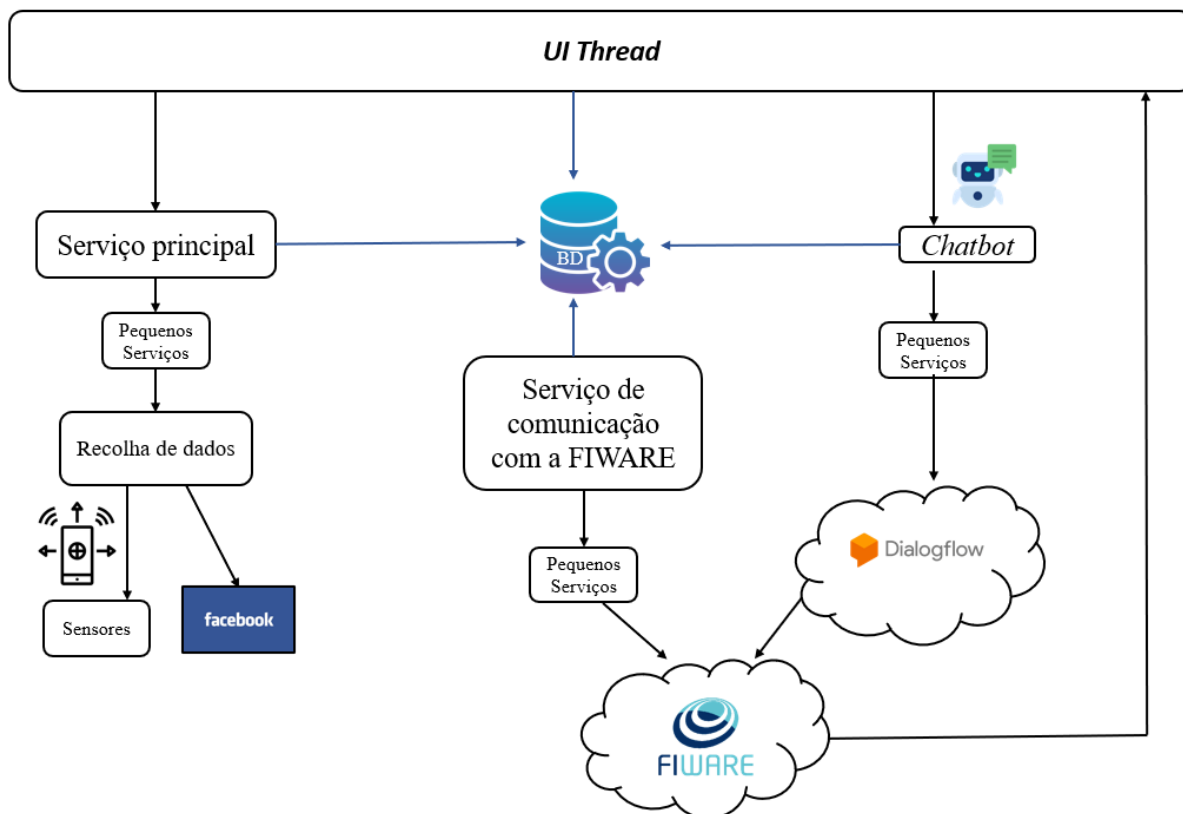


Figura 3.3: Arquitetura da aplicação móvel.

3.5 Requisitos

Nesta secção serão abordados os requisitos não funcionais e requisitos funcionais da aplicação.

3.5.1 Requisitos não funcionais

Os Requisitos Não Funcionais (RNF) são bastante importantes na maioria das aplicações de *smartphones*. Embora os utilizadores não contactem com eles diretamente, muitas vezes são estes os principais requisitos do sucesso de um aplicação. Os RNF da ISABELA encontram-se caracterizados na Tabela 3.1.

Requisitos Não Funcionais		
N.º	Função	Descrição
RNF01	Desempenho	A aplicação de não deve afetar muito o tempo de vida da bateria.
RNF02	Portabilidade	A aplicação deve ser compatível com SO Android e iOS.
RNF03	Usabilidade	A aplicação deve ter um UI intuitivo e objetivo.
RNF04	Reusabilidade	A aplicação deve utilizar bibliotecas e APIs que facilitem o seu desenvolvimento.
RNF05	Acessibilidade	A aplicação deve ter a capacidade de satisfazer um largo número de utilizadores
RNF06	Privacidade	O utilizador tem acesso exclusivamente ao seus dados.
		A autenticação da aplicação é feita através do Facebook.
		Os dados recolhidos são eliminados quando enviados para a FIWARE ou passados 30 minutos.
		Cada utilizador é associado a um ID, permanecendo anónima a sua real identidade.

Tabela 3.1: Requisitos não funcionais.

Atualmente, a Privacidade é um dos RNF mais importante a ter em conta no desenvolvimento de aplicações de recolha de dados e monitorização. Por este motivo, o grupo de trabalho do ISABELA tem dois elementos destacados para lidar com toda a questão de privacidade e segurança, assim como alguns problemas que possam surgir relativamente a este assunto.

3.5.2 Requisitos funcionais

Cada Requisito Funcional (RF) é avaliado tendo em conta a sua função e ainda consoante uma certa prioridade. Na Tabela 3.2 são descritas as funções de cada requisito funcional, assim como uma breve descrição de cada um. De seguida é possível observar, na Tabela 3.4, os pré-requisitos necessários para cada requisito, assim como um grau de prioridade, variando entre 1,2 e 3 consoante padronizado na Tabela 3.3.

Requisitos Funcionais		
N.º	Função	Descrição
RF01	Introdução	A aplicação deve apresentar uma página de introdução, referindo o nome do projeto e a Universidade de Coimbra.
RF02	Janela <i>pop-up</i> de permissões	A aplicação, quando é iniciada pela primeira vez, deve pedir ao utilizador que conceda permissões para utilização dos vários sensores instalados no <i>smartphone</i> .
RF03	Login com o <i>Facebook</i>	A aplicação deve permitir que o utilizador efetue Login através da sua conta de <i>Facebook</i> .
RF04	Registo de utilizador	A aplicação deve permitir que uma nova conta seja registada.
RF05	<i>Logout</i>	A aplicação deve permitir que o utilizador desligue a sua sessão.
RF06	<i>ChatBot</i>	A aplicação deve ter um <i>ChatBot</i> que interage com o utilizador e o auxilia, fornecendo-lhe informação.
RF07	Fluxo da aplicação	A aplicação deve permitir utilizar opções de retroceder e retornar à página inicial.
RF08	Primeira configuração	A aplicação, quando é iniciada pela primeira vez, deve abrir uma página de configuração inicial, permitindo que o utilizador escolha as opções desejadas.
RF09	Página de configurações	A aplicação deve permitir que o utilizador altere as configurações inicialmente escolhidas.
RF10	Gráficos com informação de atividade	A aplicação deve apresentar sob a forma de um gráfico, informação relativa ao nível de atividade do utilizador
RF11	Recolher Informação	A aplicação deve recolher informação através dos sensores do telemóvel de modo a inferir dados relativos à sua atividade, localização e sociabilidade.
RF12	Guardar informação	A aplicação deve guardar informação do utilizador mesmo sem acesso à <i>Internet</i> .
RF13	Fornecer formulários	A aplicação deve fornecer regularmente formulários ao utilizador.
RF14	Notificações	A aplicação deve enviar notificações relativamente a formulários disponibilizados para o utilizador responder.
RF15	Enviar informação para a <i>FIWARE</i>	A aplicação deve comunicar e enviar toda a informação recolhida dos sensores do <i>smartphone</i> para a <i>FIWARE</i> .
RF16	Detetar comportamentos	A aplicação deve detetar comportamentos que levem a um mau desempenho académico.
RF17	Informação sobre utilizador	A aplicação deve ter inferido conhecimento do utilizador, quer através dos formulários, quer através do <i>ChatBot</i> .
RF18	Informação sobre disciplinas	A aplicação deve conhecer informação relativamente às disciplinas que o aluno frequenta.
RF19	Formulários de estudo	A aplicação, conhecendo as disciplinas e matérias que o estudante necessita, deve disponibilizar formulários que ajudem o aluno a estudar

Tabela 3.2: Requisitos funcionais - função e descrição.

Prioridade	Descrição
1	Requisito não implementado mas considerado para implementação futura.
2	Requisito importante para a aplicação, mas a sua utilização é possível sem este.
3	Requisito essencial na utilização da aplicação.

Tabela 3.3: Definição das prioridades dos requisitos funcionais.

Requisitos Funcionais		
N.º	Pré-requisito	Prioridade
RF01	Nenhum	2
RF02	Nenhum	3
RF03	Conexão à <i>Internet</i> Ter conta de <i>Facebook</i>	3
RF04	Conexão à <i>Internet</i>	3
RF05	O utilizador tem que ter sessão iniciada	3
RF06	Conexão à <i>Internet</i>	2
RF07	Nenhum	3
RF08	Conexão à <i>Internet</i>	3
RF09	Conexão à <i>Internet</i>	3
RF10	Conexão à <i>Internet</i>	2
RF11	Garantir permissões de acesso aos sensores do smartphone	3
RF12	Nenhum	3
RF13	Nenhum	2
RF14	Permitir o envio de notificações	2
RF15	Conexão à <i>Internet</i>	3
RF16	Conexão à <i>Internet</i>	2
RF17	Conexão à <i>Internet</i>	1
RF18	Conexão à <i>Internet</i>	1
RF19	Conhecer informação sobre disciplinas	1

Tabela 3.4: Requisitos funcionais - pré-requisitos e prioridade.

3.6 Desenvolvimento

Nesta secção será explicado todo o processo de desenvolvimento da aplicação ISABELA em *Xamarin*, assim como todas as suas funcionalidades.

3.6.1 Aplicação em *Xamarin*

A aplicação ISABELA teve por base a aplicação já desenvolvida para *Android*, em *Java*. No entanto, toda a programação, desde o *back-end* até *front-end* foi feita de raiz, não havendo reutilização de código já existente.

Deste modo, através da *cross-platform Xamarin*, foram criadas todas as páginas presentes no ISABELA, com a estrutura da Figura 3.4. Nesta estrutura o código é dividido em *Views* e *ViewModels*.

Cada *View* representa uma página diferente da aplicação. É aqui que é programada toda a parte visual, de UI. Por exemplo, é possível visualizar na Figura 3.4, pasta *Forms*, que, uma vez que a aplicação conta com sete formulários, existem sete ficheiros *.xaml*, onde em cada um é definida a parte visual de cada formulário.

De forma a conectar a parte visual à parte lógica, existem os *ViewModels*. Cada *View* apresenta um *ViewModel* associado, onde é definida toda a programação lógica (em *C#*) relativa a cada *View*.

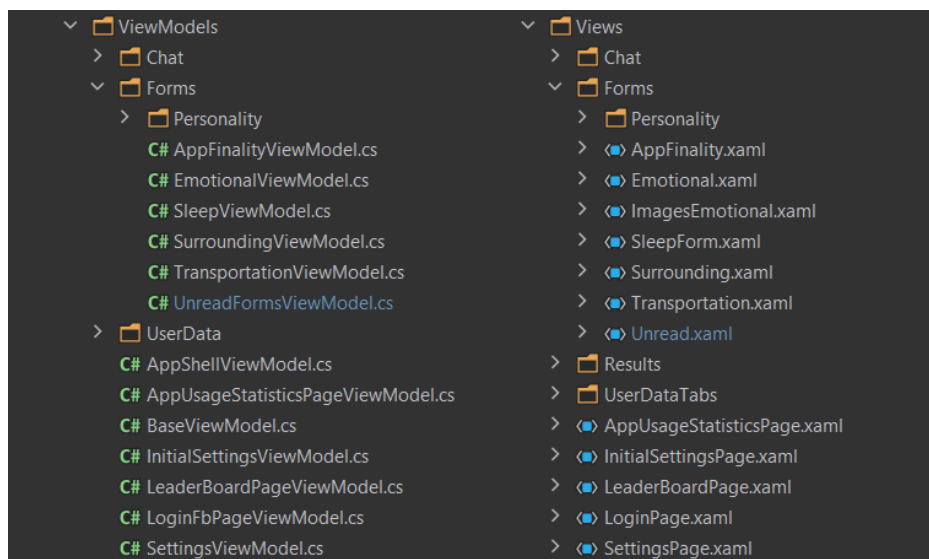


Figura 3.4: Exemplo de estruturação do código - *ViewModels* (esquerda) e *Views* (direita).

Como exemplo, na Figura 3.5, é apresentada a implementação de um botão (*Button*).

```

306
307
308
309
310
311
312
313
314

```

```

<Button
  BackgroundColor="{StaticResource ResourceKey=Green_ISABELA}"
  HorizontalOptions="Center" WidthRequest="200"
  Text="{DynamicResource ResourceKey=TfButton}"
  Margin="{t} 0, [t] 30, [r] 0, [b] 0"
  CornerRadius="10"
  TextColor="{StaticResource ResourceKey=White_ISABELA}"
  Clicked="SendClick"/>

```

Figura 3.5: Exemplo de implementação de um botão.

Este *Button* é implementado numa *View*, onde são definidas (em *XAML*) algumas características como a cor, alinhamento vertical e horizontal, etc. Este *Button* é ainda caracterizado com um método, chamado *Clicked*, que permite que o programador defina a reação do *Button* quando este é premido. Esta reação é caracterizada pela função *SendClick*, implementada no *ViewModel* associado à *View* onde está definido o *Button*.

Para adicionar outras funcionalidades, ou outros serviços necessários, que não necessitem de uma página visual na aplicação, isto é, uma *View*, a programação lógica (em *C#*) é efetuada separadamente, em outras pastas (Figura 3.6). Por exemplo, na pasta *Languages* foram definidos os dois idiomas possíveis da aplicação: Português e Inglês, e na pasta *Services* e *Utils* foram definidos alguns serviços como por exemplo de armazenamento de dados e comunicação com a *FIWARE*.

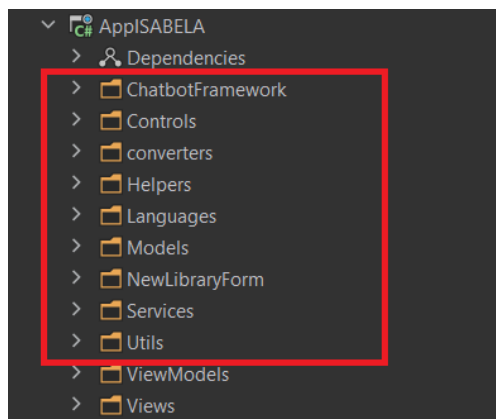


Figura 3.6: Exemplo de estruturação do código.

Na programação da parte visual (*Views*), em cada ficheiro *.xaml* foram utilizadas em média 300 linhas de código. Considerando todas as páginas visíveis na Figura 3.4 e ainda as páginas ocultas presentes nas pastas *Chat*, *Personality*, *Results* e *UserDataTabs* (Figura 3.4), foram utilizadas, no total, para a implementação da parte visual, cerca de 7000 linhas de código.

Relativamente à parte lógica, em *C#*, foram utilizadas cerca de 5000 linhas de código.

3.6.2 Armazenamento

Para efetuar o armazenamento, é criada uma base de dados no *smartphone*, com a única intenção de armazenar dados caso não exista conexão à *Internet*, de forma a evitar a sua perda. Assim que for possível comunicar com a *FIWARE*, isto é, quando o *smartphone* estabelece ligação com a *Internet*, os dados são enviados para a *FIWARE* onde são armazenados e eliminados da base de dados do *smartphone*.

Para criar e aceder aos dados na base de dados do *smartphone* foi utilizada a biblioteca *sqlite-net-pcl*.

As entidades utilizadas para guardar informação relativa aos formulários estão representadas na Figura 3.7. Todos os dados adquiridos através dos sensores físicos do telemóvel apresentam as entidades presentes na Figura 3.8. Para a criação desta entidades, primeiramente foram criadas classes representativas de cada entidade e só depois foi criada a base de dados com essas classes.

Personality	
id (auto incremento)	int
conscientiousness	string
extraversion	string
honesty	string
neuroticism	string
openness	string
agreeableness	string
answers	JsonObject

AppFinality	
id (auto incremento)	int
app1	string
app2	string
app3	string
app4	string
app5	string
timestamp	string

Sleep_Form	
id (auto incremento)	int
wakeup_time	string
sleep_time	string
sleep	string
sociability	string
exercise	string
study	string
timestamp	string

Emotional	
id (auto incremento)	int
nervous	int
scared	int
timestamp	string

Psychology_Form	
id (auto incremento)	int
angry	string
animated	string
anxious	string
desperate	string
despondent	string
fulloflife	string
happy	string
pessimistic	string
relaxed	string
sad	string
unnerved	string
upset	string
timestamp	string

TransportationForm	
id (auto incremento)	int
typeOfTransportation	string
numberOfPeople	string
timestamp	string

UnreadForm	
id (auto incremento)	string
Form	bool
FormName	string
timestamp	string

SurroundingForm	
id (auto incremento)	int
SurroundingPeople	string
timestamp	string

Figura 3.7: Entidades relativos aos formulários.

Como é possível observar na Figura 3.7 e na Figura 3.8, todas as entidades apresentam um *id* que auto incrementa, um *timestamp* e atributos necessários para a análise e processamento da informação recolhida através dos sensores do *smartphone* e dos formulários.

SmartphoneData			
id (auto incremento)	int	gyroscope	string
operating_system	string	gps	string
activity	string	wifi	string
location	string	bledevices	string
distanceHome	float	connectivity	string
step_count	int	proximitymax	float
gravity	string	proximitymin	float
altitude	string	proximity	string
phone_lock	string	soundmax	float
foregroundApp	string	soundmin	float
lightmax	float	soundavg	float
lightmin	float	battery	string
lightavg	float	compass	string
accelerometer	string	barometer	string

SensorConfig	
id (auto incremento)	int
samplesTime	int
maxArraySize	int
dutyCycle	int
permissionGranted	int

SleepSegment	
id (auto incremento)	int
status	string
start	string
end	string
description	string
duration	int
timestamp	string

LocationConfig	
id (auto incremento)	int
Interval	int
FastestInterval	int
MaxWaitTime	int
SmallestDisplacement	int
Priority	int

SleepClassify	
id (auto incremento)	int
confidence	int
light	int
motion	int
timestamp	string

Figura 3.8: Entidades relativas aos dados adquiridos pelos sensores do *smartphone*.

3.6.3 Aquisição de dados

A aquisição de dados é efetuada diretamente através do *smartphone*, através dos sensores nele incutidos e também de respostas a formulários disponibilizados pela aplicação.

3.6.3.1 Sensores físicos

A aquisição de dados através dos sensores presentes no *smartphone* é efetuada através da utilização de uma biblioteca chamada *SocialiteSensors*, que permite estas recolhas de dados. Todos os dados recolhidos estão representados na Figura 3.8, através das entidades *SmartphoneData*, *SensorConfig*, *LocationConfig*, *SleepSegment* e *SleepClassify*.

Toda esta aquisição de dados é possível pois a biblioteca utiliza APIs disponibilizadas por cada sistema operativo de forma a recolher os dados através dos sensores do *smartphone*.

Para o caso da entidade *SmartphoneData*, nem todos os dados são possíveis de recolher através dos sensores do *smartphone* devido às permissões impostas por cada SO. Para o caso do *Android* os valores *altitude* e *gravity* não são recolhidos, assim como não é possível recolher os valores de *phone_lock*, *foregroundApp*, *connectivity*, *compass* e *barometer* para *iOS*. Quando a recolha não é efetuada os valores são apresentados como *null*.

3.6.3.2 Formulários

Para além da aquisição de dados sensoriais, são também recolhidos dados influenciados diretamente pelo utilizador, através da sua resposta a formulários disponibilizados pela aplicação. Estes formulários serão todos aprofundados em 3.6.5.8.

3.6.4 Comunicação com a *FIWARE*

Para a comunicação com a *FIWARE* foi desenvolvida uma biblioteca - *RestFiwareAPI*, cujo principal objetivo é enviar os dados adquiridos, quer dos sensores do *smartphone*, quer dos formulários respondidos pelo utilizador, diretamente para a *FIWARE*. Em caso de não haver conexão à Internet esta biblioteca envia os dados da base de dados do *smartphone* para a *FIWARE*.

A primeira interação que a aplicação faz com a *FIWARE* é através do módulo *ORION*, quando o utilizador pretende efetuar *login*. Para efetuar *login* é necessário um *Token* de acesso. No entanto, quando a aplicação é iniciada pela primeira vez, não existe ainda um *Token*. Caso não exista *Token* de acesso é feito um pedido *POST* para o *IDM*. Este pedido *POST* permite criar um novo *Token* de acesso para um novo utilizador.

Depois de executado *login* com sucesso, a *FIWARE* retorna o *Token* de acesso que é utilizado em todos os pedidos efetuados aos restantes módulos da *FIWARE*. Esses módulos validam o acesso através do *Token* de acesso com o *IDM* antes de retornar ao utilizador qualquer pedido.

Depois de todo o processo de registo e autenticação, sempre que a biblioteca pretende fazer pedido à *FIWARE* é necessária uma solicitação *GET* (solicitação REST para recuperar informações). Este serviço executa a sua própria *thread* e é executado em segundo plano com três funções principais: ler os dados armazenados na base de dados do *smartphone*, retornar esses valores e iniciar os serviços de solicitação *PATCH* (solicitação que atualiza os valores das entidades na *FIWARE*). Por fim, para garantir que nenhuma informação é perdida, o serviço *PATCH* recebe um código de confirmação da *FIWARE* em como os dados foram recebidos. Este serviço envia todos os dados de todas as entidades representadas na Figura 3.7 e na Figura 3.8.

A biblioteca permite também a comunicação com o módulo *COMET*, que funciona de igual forma à comunicação com o módulo *ORION*. No entanto, o módulo *COMET* é utilizado quando se pretende recuperar dados históricos ou agregar dados, como por exemplo quando se pretende mostrar ao utilizador o seu *feedback* relativo à sua atividade dos últimos oito dias.

Esta biblioteca permite que o utilizador defina se pretende que as comunicações com a *FIWARE* sejam feitas através de redes móveis e *Wireless Fidelity* (WiFi) ou apenas com rede WiFi. Permite ainda escolher a frequência com que os dados são enviados para a *FIWARE*, pois os dados enviados e recebidos possuem *duty-cycle*.

3.6.5 UI e fluxo da aplicação

Nesta subsecção será explorada a forma como a aplicação interage com o utilizador e as escolhas de *design*. A aplicação foi desenvolvida à base das cores azul escuro e verde claro e o fluxo da aplicação é caracterizado pela Figura 3.9.

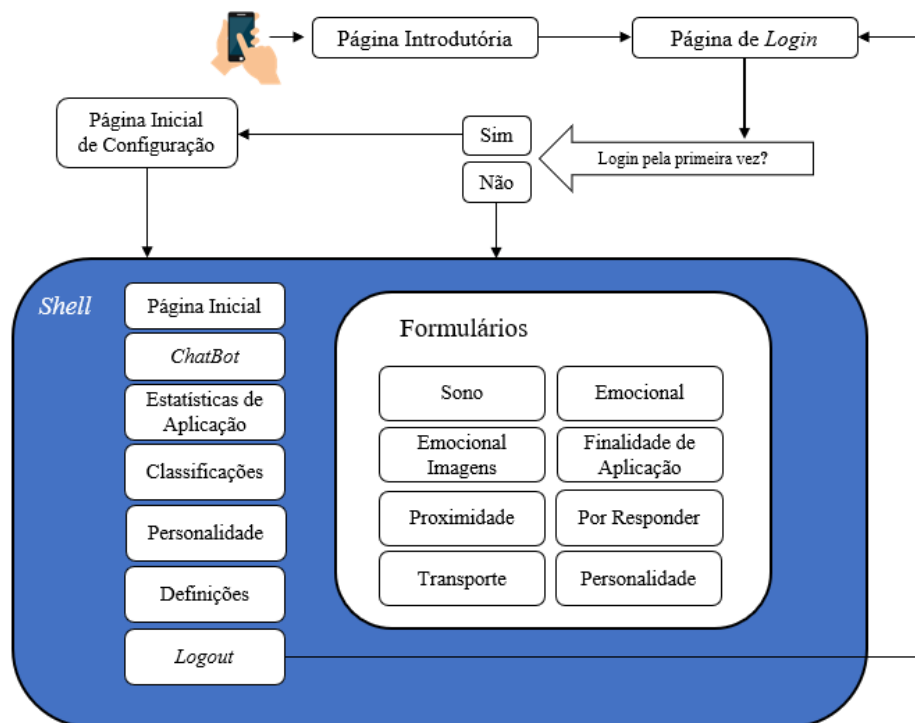


Figura 3.9: Fluxo da aplicação.

3.6.5.1 Página Introdutória e de *Login*

A aplicação é iniciada através do clique no ícone da mesma, presente no menu do *smartphone*. Quando o utilizador inicia a aplicação esta apresenta uma página introdutória que tem uma duração de 5 segundos, onde é referido o nome do projeto e a Universidade de Coimbra, Figura 3.10. De seguida, caso seja a primeira vez que a aplicação é iniciada no *smartphone*, são disponibilizadas janelas *pop-up*, como os exemplos da Figura 3.11, onde é pedido ao utilizador que conceda as permissões necessárias para o funcionamento da aplicação. Estas permissões serão abordadas em pormenor em A.



Figura 3.10: Página Introdutória.

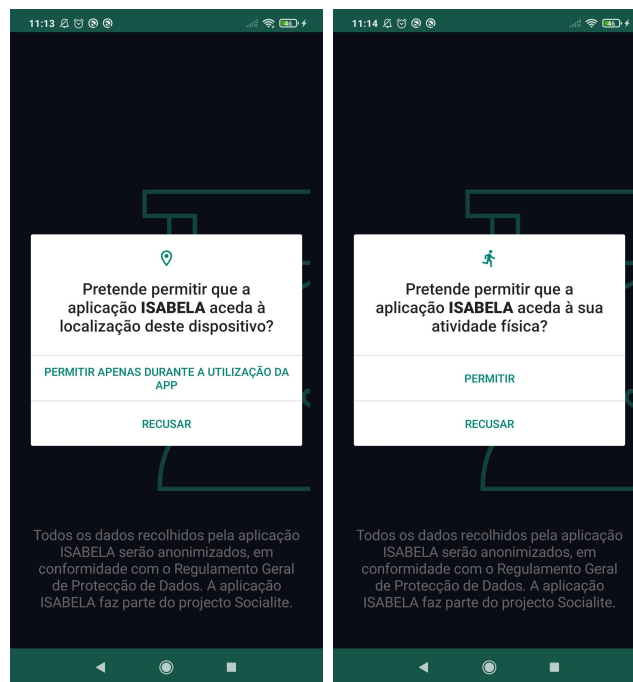


Figura 3.11: Janela *pop-up* de permissões relativa à localização e atividade física.

Depois de concedidas as permissões necessárias, é apresentada uma página de *Login*, Figura 3.12, onde o utilizador, ao clicar num botão consegue entrar na aplicação através da sua conta de *facebook*. Para implementar este serviço foi necessário criar uma aplicação na plataforma *Meta for developers* e associá-la à aplicação ISABELA. Após isto, através de uma API, foi possível implementar métodos e serviços de *login*, facilitando o desenvolvimento da aplicação.

Durante o *login* a aplicação faz uma verificação: caso seja a primeira vez que aquela conta de *facebook* efetua *login* na aplicação, o mesmo é registado (sendo-lhe atribuído um ID) e é-lhe concedido acesso a uma página de configurações iniciais, Figura 3.13. Nesta página o utilizador deve primir o botão com o ícone do WiFi de modo a fazer *scan* das redes *WiFi* alcançadas e deve escolher a rede de sua casa. Esta escolha é importante, pois é a partir do SSID do WiFi da rede de casa do utilizador que a aplicação deteta que o mesmo se encontra em casa. Ainda nesta página, o utilizador deve escolher o estudo a que pertence, de modo a que os dados enviados para a *FIWARE* sejam agregados de acordo com que se deseja realizar. Por último, o utilizador tem a opção de escolher o idioma (entre português e inglês) e pode ainda escolher se pretende que a aplicação envie informação apenas com rede *WiFi* ou se também com dados móveis. Quando concluída a configuração inicial o utilizador deve primir o botão de *concluir*.



Figura 3.12: Página de *Login*.

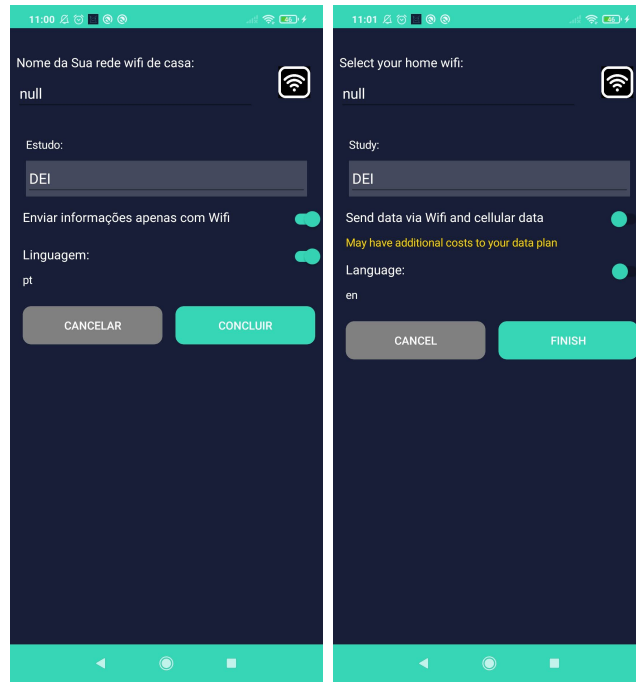


Figura 3.13: Página de configurações iniciais.

3.6.5.2 Página inicial e *Shell*

Depois de efetuado o *Login*, a aplicação navega diretamente para uma página inicial.

Como a plataforma ISABELA é um sistema HITLCPS, por norma são apresentados dados informativos ao utilizador. No entanto, dependendo do estudo que se pretenda fazer, esta página inicial pode ser uma simples página inicial, Figura 3.14, onde não são apresentados dados, ou pode ser um conjunto de páginas, Figura 3.15, Figura 3.16 e Figura 3.17, onde são apresentados dados ao utilizador. Esta decisão ocorre na *FIWARE* e é comunicada à aplicação. Desta forma, é possível o utilizador ter acesso aos seus dados, por exemplo, durante uma semana e noutra semana já não ter.

Quando se pretende mostrar os dados ao utilizador, estes estão distribuídos por 4 páginas, que podem ser acedidas através do deslize do ecrã para o lado. Nas 3 primeiras páginas, Figura 3.15 e Figura 3.16, são apresentados dados relativos à atividade (variando entre *parado*, *desconhecido*, *num veículo* e *em exercício*), ao sono (qualidade e duração), estado emocional (sob a forma de imagens e também quantitativamente, numa escala de 0 a 4). Para além destes dados, são ainda apresentados o nível de risco do concelho e a proximidade social, como forma de prevenir a COVID-19.



Figura 3.14: Página inicial com os dados ocultos.



Figura 3.15: Página inicial relativos às últimas 24 horas.



Figura 3.16: Página inicial relativos aos últimos 4 dias (esquerda) e 8 dias (direita).

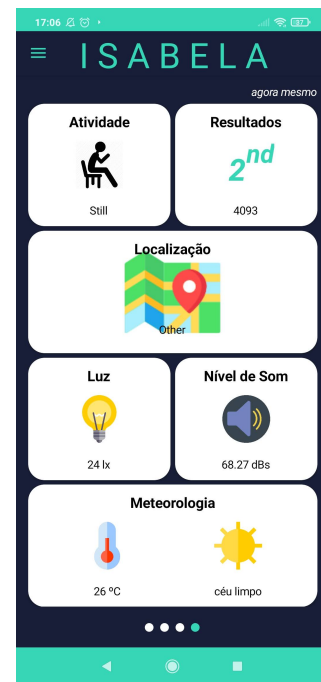


Figura 3.17: Dados recolhidos através dos sensores do smartphone.

Na quarta página, Figura 3.17, são mostrados ao utilizador dados, em tempo real, que foram obtidos através dos sensores do telemóvel, tais como atividade, localização, nível de luz, som e meteorologia. Esta página conta ainda com os *resultados* proveninetes na página de classificações, explicada em 3.6.5.5.

Para além desta página inicial, depois de efetuado o *login*, o utilizador tem acesso à *Shell* - uma ferramenta disponibilizada pelo *Xamarin.Forms* que permite criar um esquema de navegação baseado em *Uniform Resource Identifier*. Esta *Shell*, Figura 3.18, funciona como um menu para o utilizador, permitindo uma navegação facilitada entre páginas. É composta por várias páginas, que serão abordadas uma a uma.

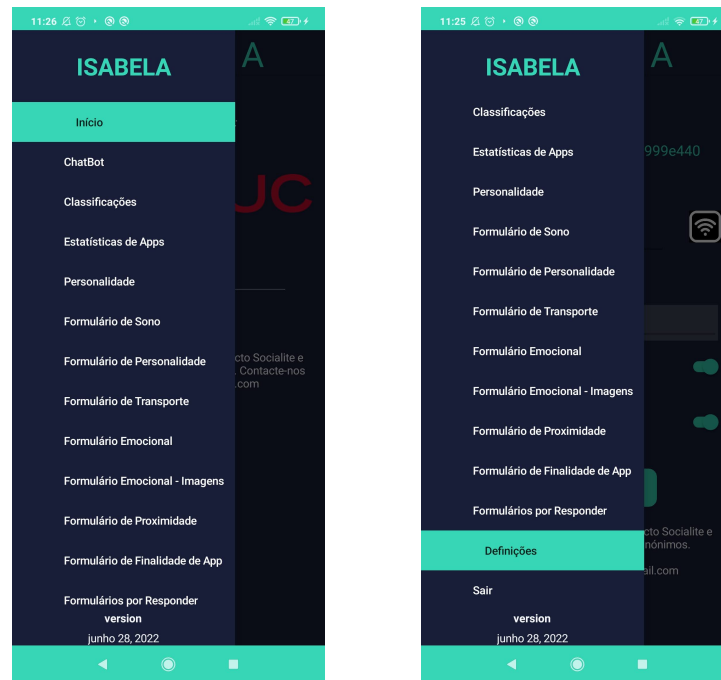


Figura 3.18: *Shell*/Menu do utilizador.

3.6.5.3 Página de definições

A página de definições, (Figura 3.19), permite que o utilizador mude a sua configuração da aplicação ISABELA sempre que quiser. Assim como a página de configurações inicial, o utilizador tem a opção de escolher entre dois idiomas - português e inglês, enviar informação via dados móveis e WiFi ou apenas WiFi, escolher a sua rede WiFi de casa e ainda escolher o estudo a que pertence.

3.6.5.4 *Chatbot*

A aplicação apresenta um *chatbot*, com o qual o utilizador pode interagir. O utilizador tem a possibilidade de colocar questões ao *chatbot* (como por exemplo qual o menu da cantina do Pólo 2 da Universidade de Coimbra para certo dia), pode pedir para o *chatbot* lhe definir um alarme, entre outras opções. Para além disso, o *chatbot* pode comunicar com o utilizador caso verifique que o mesmo não apresenta um comportamento adequado para um bom desempenho académico (como por exemplo incentivar o utilizador a fazer mais exercício, dormir mais

tempo, estudar mais, etc.). O *layout* do *chatbot* encontra-se representado na Figura 3.20.

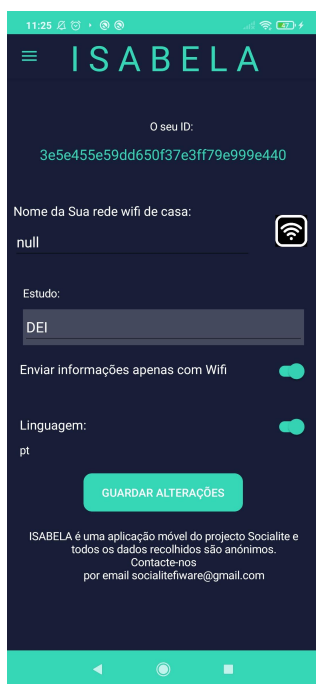


Figura 3.19: Página de definições.

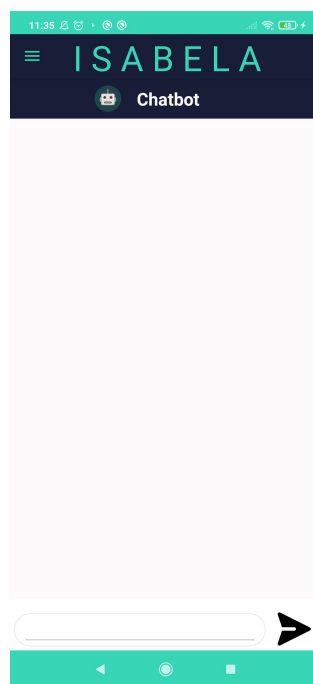


Figura 3.20: Página do *chatbot*.

3.6.5.5 Classificações

A aplicação conta com uma página de classificações, Figura 3.21. Esta página é apresentada ao utilizador como forma de o incentivar a utilizar a aplicação e melhorar o seu *rank*. Os pontos são atribuídos tendo em conta a recolha de dados feito por parte dos sensores do telemóvel. Enquanto os dados estiverem a ser adquiridos e enviados para a *FIWARE*, os pontos vão aumentando. Quanto mais tempo o utilizador deixar a aplicação ligada (mesmo que em plano em segundo plano), mais os seus pontos aumentam.

A classificação correspondente ao utilizador é apontada por uma seta, permitindo ao mesmo saber em que lugar se encontra na tabela classificativa. Por questões de privacidade, embora cada utilizador consiga saber os pontos dos outros utilizadores, assim como o seu *rank* associado, este não tem acesso a mais nenhuma informação identificativa relativamente aos outros utilizadores, mantendo o anonimato entre utilizadores.

3.6.5.6 Estatísticas de aplicações

Existe também uma página que apresenta estatísticas relativamente à utilização das aplicações presentes no *smartphone*, Figura 3.22. Esta página mostra ao utilizador as aplicações que mais utilizou e a sua duração de utilização nas últimas vinte e quatro horas.

Esta funcionalidade apenas foi implementada através de uma API em SO *Android*, uma vez que o SO *iOS* não permite que as aplicações tenham acesso a informação relativa a outras aplicações, por questões de privacidade.

Com esta página o utilizador pode ter uma noção da quantidade de tempo que gasta em certas aplicações e com essa informação gerir melhor a utilização das mesmas.

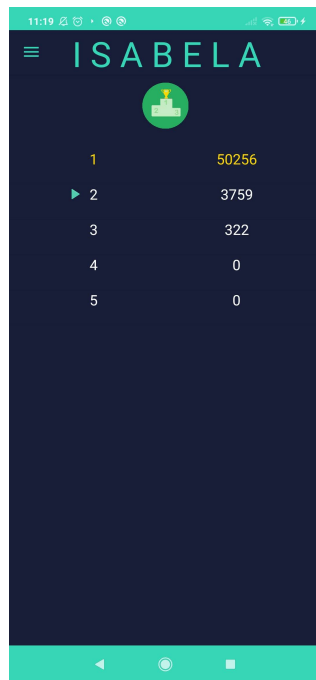


Figura 3.21: Página de classificações.

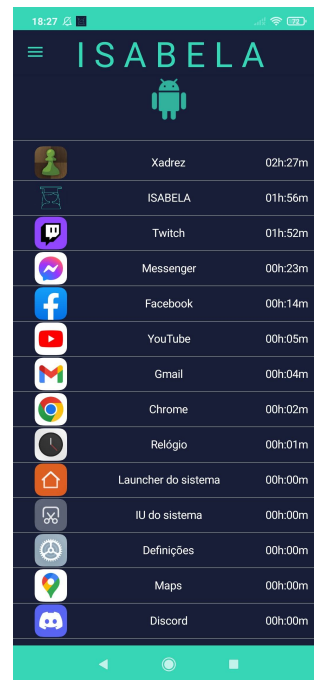


Figura 3.22: Página de estatísticas de aplicações.

3.6.5.7 Personalidade

A página de personalidade, Figura 3.23, representa e mostra ao utilizador cinco atributos que caracterizam a sua personalidade. Esta página apenas é apresentada depois de o utilizador responder ao formulário de personalidade (abordado em 3.6.5.8) e de os dados serem processados. Os cinco atributos são mostrados ao utilizador sob a forma de uma barra de preenchimento. Quanto mais a barra estiver preenchida, maior o nível de cada atributo. Estas barras podem também apresentar três cores diferentes: verde menos carregado, quando o nível é inferior à média (valor fornecido pelo grupo de psiquiatria da Faculdade de Medicina da Universidade de Coimbra), um verde intermédio quando o valor está dentro da média e ainda um verde mais carregado quando o nível é superior à média.

Cada uma destas definições é apresentada numa janela *pop-up* sempre que o utilizador carregar no ícone com um ponto de interrogação em cada diferente atributo (representados dois exemplos na Figura 3.24).

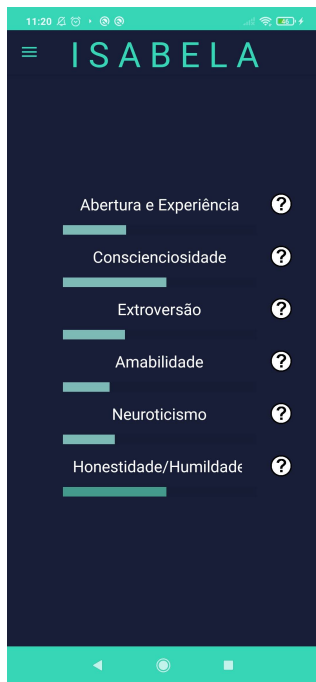


Figura 3.23: Página de personalidade



Figura 3.24: Janela *pop-up* com a definição do atributo *Abertura e Experiência* (esquerda) e *Conscienciosidade* (direita).

3.6.5.8 Formulários

As páginas abordadas anteriormente, são apenas informativas. Agora nesta secção serão analisados os vários formulários disponibilizados pela aplicação, aos quais o utilizador deve responder, pelo menos uma vez por dia.

Todos os formulários apresentam um botão *Enviar*. Quando este é primido são feitas verificações em cada formulário. Caso algum preenchimento esteja vazio ou incorreto, a aplicação apresenta uma janela *pop-up* e pede ao utilizador para verificar o preenchimento dos formulários. Caso o utilizador pressione o botão *Enviar* e não ocorram erros, aparece uma janela *pop-up* informativa de que o formulário foi enviado com sucesso (para a *FIWARE*). Caso o utilizador tenha conexão à *Internet* estabelecida os dados são diretamente enviados para a *FIWARE*, caso contrário, cada formulário é guardado localmente no *smartphone* e aguarda que seja estabelecida conexão à *Internet* para executar o envio. Depois de enviados, todos os dados são processados na *FIWARE*, para posteriormente serem mostrados ao utilizador.

3.6.5.8.1 Formulário de sono

Como é possível observar na Figura 3.25, este formulário questiona o estudante relativamente à que horas se deitou e acordou, através de um *DataPicker*, Figura 3.26. Depois, o estudante deve responder a cada *slider* com as seguintes questões:

- *Como dormiu a noite passada?* - com valores qualitativos entre *Muito mal* e *Muito bem*.
- *Como se sentiu socialmente ontem?* - com valores qualitativos entre *Introvertido* e *Extrovertido*
- *Quantas horas de exercício físico praticou ontem?* - com as opções 0, 1, 2 e 2+.
- *Quantas horas estudou ontem?* - com as opções 0, 2, 4 e 8+.

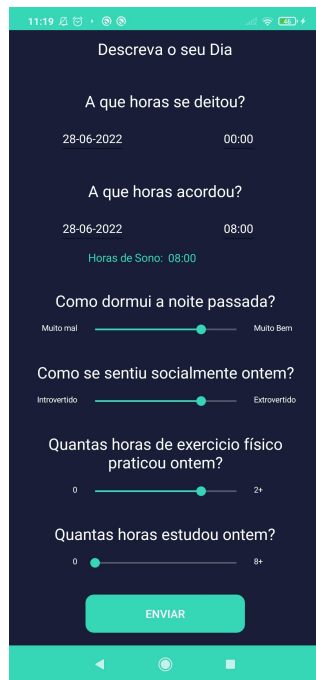


Figura 3.25: Formulário de sono

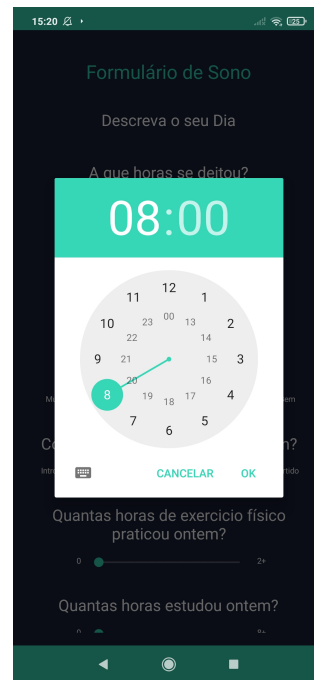


Figura 3.26: *DataPicker*

3.6.5.8.2 Formulário de personalidade

O formulário de personalidade (representadas algumas perguntas na Figura 3.27), foi desenvolvido tendo em conta um conjunto de questões propostas pela Faculdade de Psicologia da Universidade de Coimbra, com o objetivo de traçar a personalidade do estudante, tendo em conta a sua resposta às diversas questões. Este formulário pode apenas ser respondido uma vez, pois conclui-se que a personalidade tende a não mudar num espaço de 6 meses (tempo considerado necessário para realizar estudos e testes de utilização da aplicação).

As questões colocadas no formulário de personalidade são respondidas através da escolha de uma resposta de um *slider*. Cada *slider* apresenta quatro níveis, sendo o primeiro nível *Discordo Fortemente* e o último nível *Concordo Fortemente*. Uma vez que este formulário conta com sessenta questões, estas encontram-se distribuídas ao longo de seis páginas, em blocos de dez questões. A primeira página apresenta um botão que permite a navegação para a próxima página. A sexta página tem dois botões: um que permite a navegação para a página anterior e outro que permite finalizar o formulário e enviar os dados. As restantes

páginas apresentam dois botões: um que permite que o utilizador retroceda para a página anterior e outro que permite que avance para a próxima página

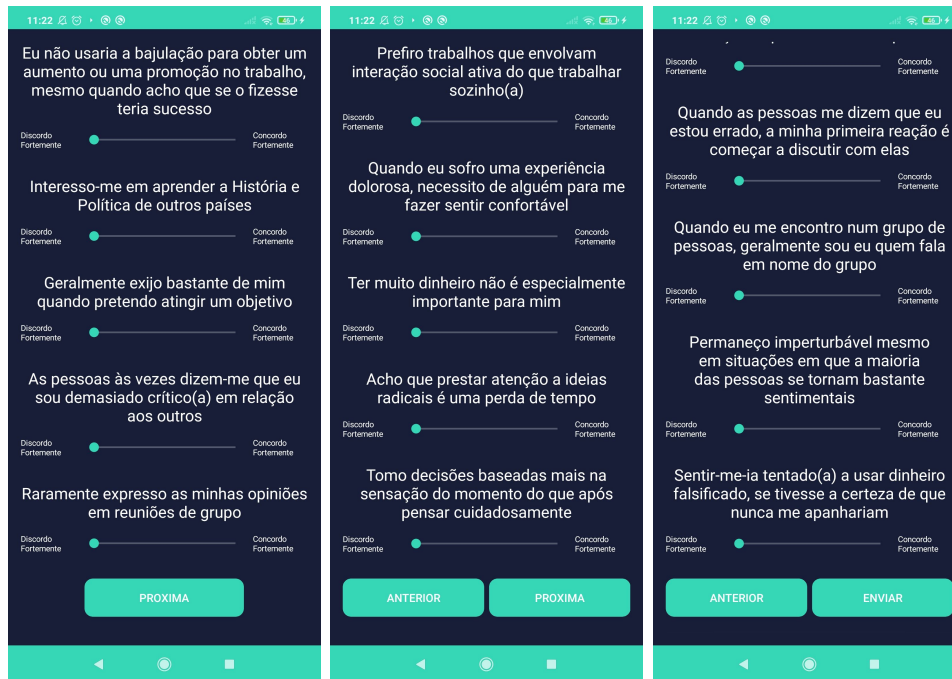


Figura 3.27: Exemplo de perguntas do formulário de personalidade.

3.6.5.8.3 Formulário emocional e emocional com imagens

São submetidos ao utilizador dois formulários com o objetivo de analisar o seu estado emocional. O primeiro conta com doze adjetivos, como por exemplo *irritado*, *triste*, Figura 3.28, ao qual o utilizador deve responder utilizando um *slider* de quatro níveis em que o primeiro representa *Nada* e o último *Muitíssimo*. No segundo formulário o estudante deve responder a dois conjuntos de cinco imagens, Figura 3.29, de acordo com o seu nível de agitação (cinco imagens superior) e prazer (cinco imagens inferiores). Para o nível de agitação pode optar entre cinco imagens, em que a primeira representa *Calmo/Relaxado* e a última *Agitado/Alerta*. Para o nível de prazer pode optar por cinco imagens, em que a primeira representa *Muito Prazer/Agradável* e a última *Muito Desprazer/Desagradável*.

Figura 3.28: Formulário emocional

Figura 3.29: Formulário emocional com imagens

3.6.5.8.4 Formulário de proximidade

A aplicação consegue, através do sensor do *bluetooth*, estimar quantas pessoas se encontram próximas do utilizador, de modo a prevenir contágio pela COVID-19. Para verificar as medições efetuadas, sempre que a aplicação deteta proximidade através do sensor, pede ao utilizador, através de uma notificação, que responda a um formulário de proximidade (Figura 3.30) para confirmar quantas pessoas se encontram a menos de dois metros de si. O utilizador apenas consegue responder numericamente, garantindo uma resposta rápida e eficaz.

3.6.5.8.5 Formulário de transporte

O formulário de transporte é constituído por duas questões: *Em que veículo esteve?* e *Quantas pessoas estavam consigo?* (Figura 3.31). Sempre que é detetada, através dos sensores do *smartphone*, uma velocidade não atingível pelo Ser Humano, é lançada uma notificação onde é pedido ao utilizador que responda a este formulário.

A resposta à primeira questão é efetuada através da seleção de um *picker*, com cinco opções: *Veículo de amigos/colegas*, *Autocarro*, *Barco*, *Taxi/TVDE* e *Metro/Comboio/Elétrico*. A resposta à segunda questão é também efetuada através de um *picker* com quatro opções, e é feita tendo em conta a primeira resposta. Como é possível visualizar na Figura 3.31, caso a primeira resposta escolhida seja *Barco*, na segunda resposta são disponibilizadas as opções *<10*, *10-29*, *30-50* e *50+*. Caso a primeira resposta seja *Veículo próprio* as opções

disponibilizadas são 0, 1, 2 ou 2+.

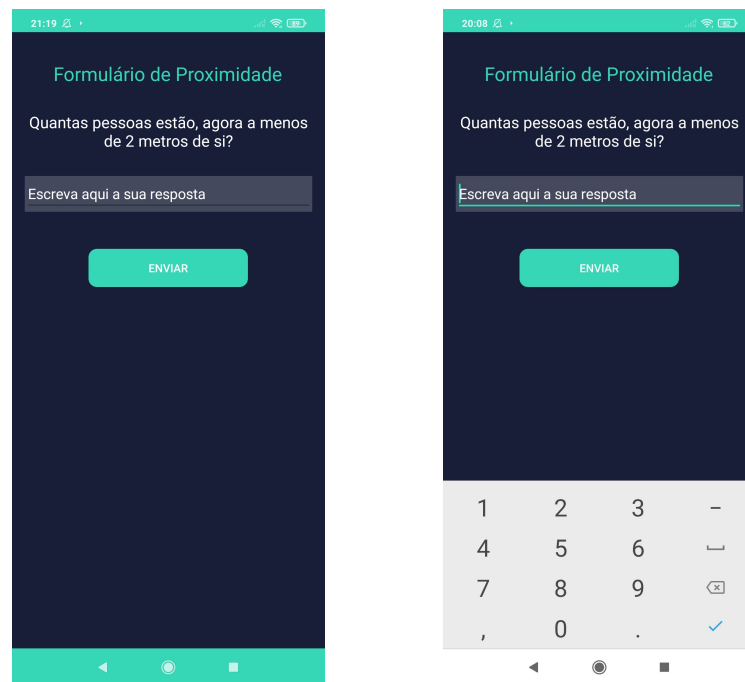


Figura 3.30: Formulário de proximidade.

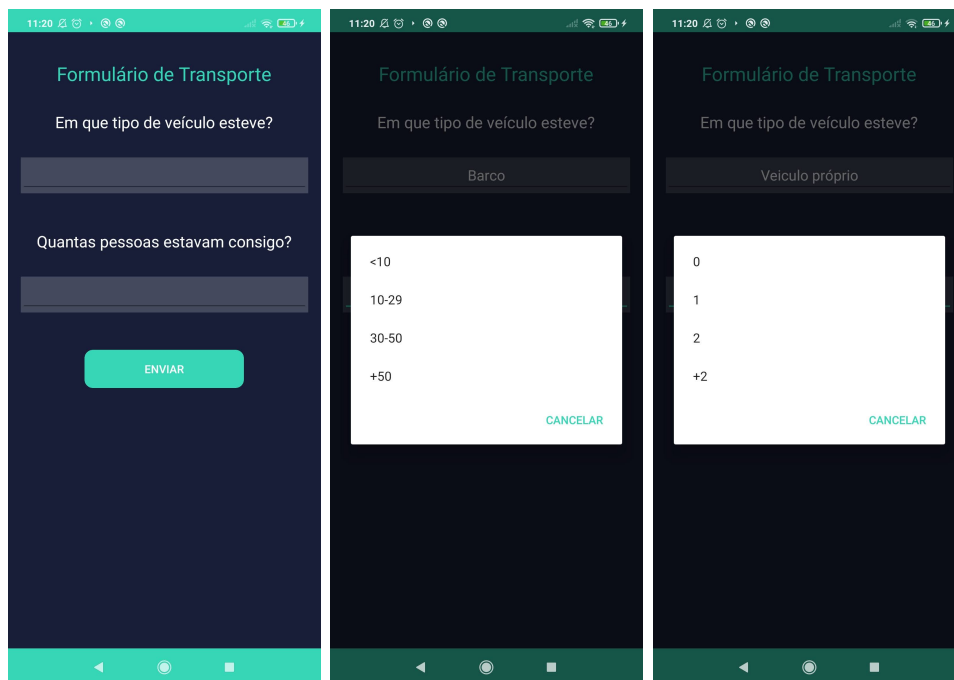


Figura 3.31: Formulário de transporte.

3.6.5.8.6 Formulário de finalidade de aplicação

A aplicação disponibiliza um formulário de finalidade de aplicação. Como já foi visto anteriormente, a ISABELA apresenta um página que mostra ao utilizador as aplicações do

seu *smartphone* que mais utilizou nas últimas vinte e quatro horas. De modo a perceber e interpretar a finalidade de cada utilização é disponibilizado este formulário, onde estão presentes as cinco aplicações mais utilizadas pelo utilizador, ao qual o mesmo deve responder para justificar a razão da sua utilização. A resposta é efetuada através da escolha de uma de quatro opções disponibilizadas num *picker*, podendo ser: *Comunicação*, *Lazer*, *Pesquisa* ou *Trabalho*. Cada resposta é independente, como representado na Figura 3.32.

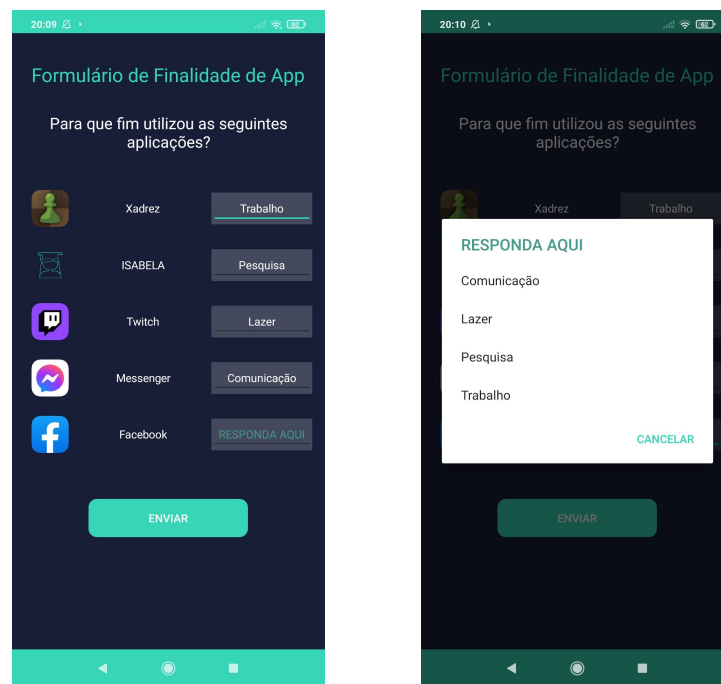


Figura 3.32: Formulário finalidade de aplicação.

3.6.5.8.7 Formulários por responder e notificações

A aplicação envia notificações como forma de avisar o utilizador dos formulários que necessita responder. Pelo menos uma vez por dia é enviada uma notificação a informar o utilizador de algum formulário que possa ter por responder. Esta notificação apenas é enviada caso se verifique que o utilizador não respondeu a um certo formulário num determinado dia. Os formulários de transporte e proximidade podem ser disponibilizados várias vezes por dia, ou seja, ao longo do dia, sempre que é detetado contacto social próximo, ou que o utilizador andou num transporte. Nestes casos, a aplicação envia uma notificação a avisar que foram disponibilizados formulários para responder. Caso o utilizador não responda ao formulário quando é enviada a notificação, cada formulário relativo a cada notificação fica guardado na página de formulários por responder, durante vinte e quatro horas.

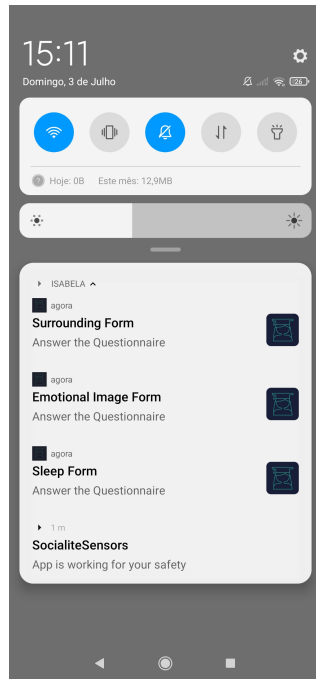


Figura 3.33: Exemplo de três notificações.

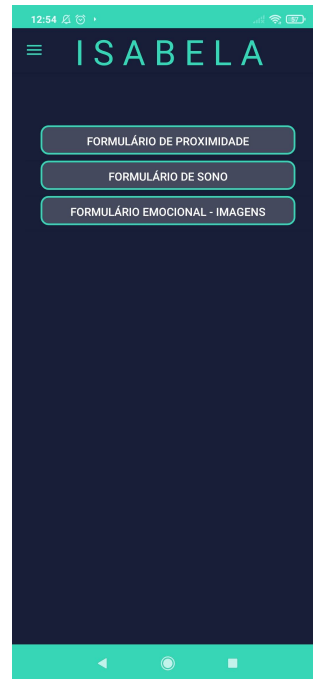


Figura 3.34: Formulários por responder

O utilizador deve responder pelo menos uma vez por dia a cada formulário. Enquanto este não responder, os formulários por responder são mostrados ao utilizador na página de *Formulários por responder* (Figura 3.34). Nesta página os formulários por responder estão representados através de botões, que uma vez primidos fazem a navegação para o respetivo formulário. Quando o utilizador responde, o formulário respondido é retirado da lista de formulários por responder. Esta lista é reiniciada a cada dia.

4

Avaliações e Testes

Uma vez que o principal objetivo desta dissertação foi desenvolver um sistema de apoio à aprendizagem compatível com SO *Android* e *iOS*, uma das primeiras avaliações efetuadas foi compilar e executar o código de forma a comprovar a possibilidade do desenvolvimento de aplicações em *Xamarin*, através de um código 90% partilhado. Ao longo do Capítulo 3, foram demonstrados alguns *layouts* da aplicação executada em SO *Android*. São agora apresentados alguns *layouts* da aplicação executada em SO *iOS*, Figura 4.1.



Figura 4.1: Alguns layouts da aplicação ISABELA executada em *iOS*.

O desenvolvimento da aplicação ISABELA em *Xamarin* foi iniciado numa época de pandemia devido à doença do coronavírus (Covid-19). Deste modo, no desenvolvimento da aplicação foi implementado um novo formulário - *Formulário de proximidade* (Figura 4.2), possibilitando a monitorização e controlo do nível de risco e proximidade do utilizador relativamente a terceiros. Este formulário não existia na aplicação ISABELA *Android*, tendo sido implementado pela primeira vez em *Xamarin*.

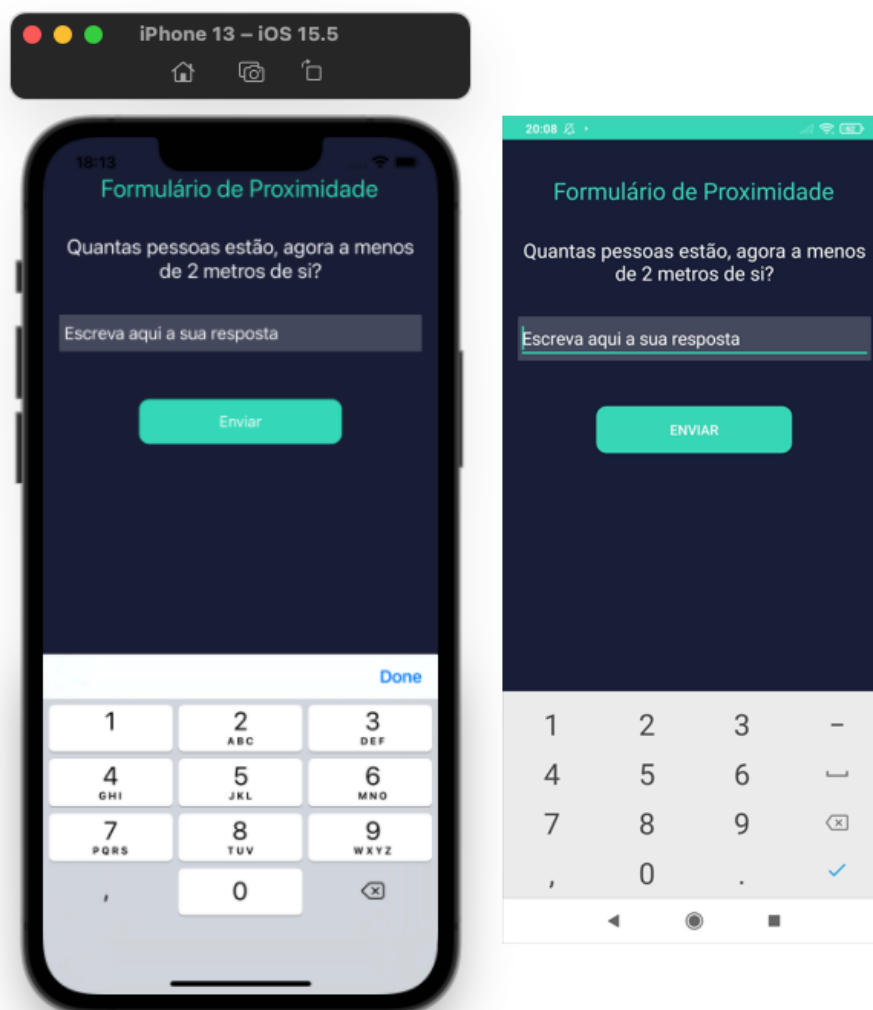


Figura 4.2: Formulários de proximidade executado em SO *iOS* (esquerda) e SO *Android* (direita).

Uma vez que este objetivo foi cumprido, e a aplicação se apresenta funcional em SO *Android* e *iOS*, o próximo passo seria testar a aplicação num grupo amostral de alunos, de forma a obter resultados relativos à eficiência da aplicação. O tempo ideal para realizar estes testes seria de um semestre. Devido ao período de duração da dissertação, não foi possível realizar testes e retirar conclusões.

No entanto, foi realizado um teste relativo à bateria consumida pela aplicação.

Atualmente, embora grande parte dos *smartphones* tenham uma boa capacidade de bateria, o consumo desta é um dos maiores requisitos em aplicações móveis. Deste modo, a aplicação ISABELA foi desenvolvida de forma a rentabilizar a utilização e consumo da bateria do *smartphone* de modo a não afetar o seu desempenho.

Na Figura 4.3, na aplicação ISABELA (delineada a vermelho) é possível visualizar que o seu consumo de bateria médio é de 17,30 mAh (miliAmperes hora). Tendo em conta que os *smartphones* atuais apresentam em média uma capacidade de bateria entre 4000 e 4500 mAh [34], a aplicação ISABELA consome em média 415mAh de bateria por dia, representando um consumo de aproximadamente 1% da bateria total do *smartphone* por dia.

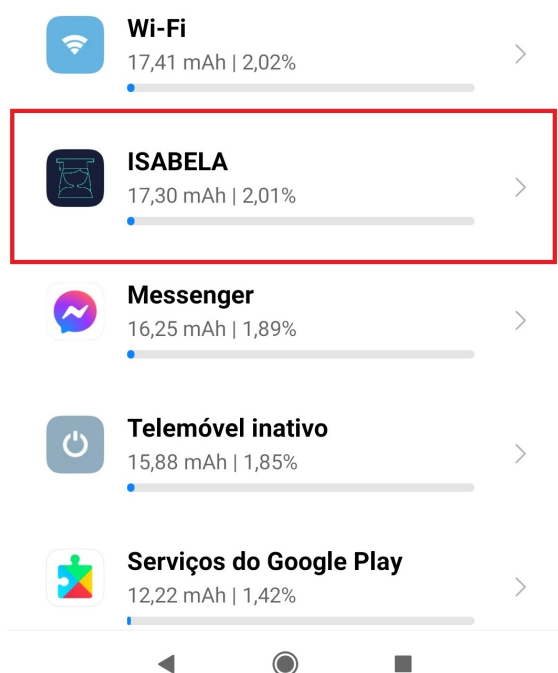


Figura 4.3: Energia consumida pela aplicação ISABELA.

5

Conclusão e Trabalho Futuro

Conteúdos

5.1	Conclusão	53
5.2	Trabalho Futuro	53

5.1 Conclusão

Esta dissertação teve como objetivo criar uma aplicação que pudesse monitorizar e melhorar o desempenho académico dos alunos e que fosse compatível com SO *Android* e *iOS*, abrangendo cerca de 98% dos utilizadores de *smartphone*. Para tal, foi utilizada como ponto de partida a aplicação ISABELA (compatível apenas com SO *Android*). Esta aplicação foi recriada através da utilização de uma *cross-platform-Xamarin*, que conciliando a linguagem de programação C# com a linguagem de marcação XAML, permite desenvolver aplicações compatíveis com SO *Android* e *iOS* através de um código único 90% partilhado.

A grande diferença da aplicação desenvolvida em relação a aplicações com finalidades semelhantes é o facto de ser utilizado um sistema HITLCPs, que considera o fator Humano na inferência dos dados, como meio de apresentar *feedback* ao utilizar e desta forma o auxiliar e monitorizar, de forma a melhorar o seu desempenho académico. Este sistema é dividido em quatro etapas: inicialmente são adquiridos dados, depois é feita uma primeira inferência e uma inferência futura tendo em conta os dados recolhidos e, por último, é efetuada uma atuação, onde o sistema determina se alguma ação pode ser efetuada. Para além do conceito de sistemas HITLCPs, esta aplicação conta também com novas tecnologias como IA, um *chatbot* e ainda o paradigma da IoT. Ao implementar todos estes conceitos, foi possível desenvolver uma plataforma IoT de apoio à aprendizagem, utilizando um *smartphone* e componentes da FIWARE, que permitem facilitar alguns desafios da IoT e questões de privacidade da monitorização móvel.

5.2 Trabalho Futuro

A aplicação ISABELA é um projeto ambicioso, que foi primeiramente desenvolvido apenas para SO *Android* e com pequenas funcionalidades. Neste momento já foi expandido sendo compatível com SO *Android* e *iOS*. A aplicação foi também desenvolvida sem ter em conta a sua reusabilidade. No entanto, neste momento já foram criadas duas bibliotecas: uma de recolha de dados através dos sensores do *smartphone* - *SocialiteSensors* e outra que permite que o *smartphone* comunique com a FIWARE - *RestFiwareAPI*. Estas bibliotecas, para além de facilitarem certas implementações, garantem ainda reusabilidade, podendo ser utilizadas noutros projetos. Adicionalmente, está a ser criada uma biblioteca dinâmica dedicada à construção da parte visual dos formulários disponibilizados na plataforma. Através desta biblioteca será possível reutilizar os *layouts* dos formulários noutras aplicações.

Futuramente, é necessário testar a aplicação num grupo amostral de alunos de forma a concluir resultados relativos à utilização da aplicação e da eficiência de um sistema HITLCPS.

Para além disso, podem ainda ser criadas mais funcionalidades na aplicação, como a possibilidade de a aplicação inferir conhecimentos do utilizador, quer através dos formulários, quer através do *chatbot* e ainda a possibilidade de aplicação conhecer informação relativa às disciplinas que o aluno frequenta. Conhecendo esta informação, a aplicação pode ainda ter a possibilidade de disponibilizar formulários que ajudem o aluno a estudar.

Bibliografia

- [1] L. S. Vailshery, “Iot connected devices worldwide 2019-2030,” <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/>, visitado a 10-06-2022.
- [2] S. O’Dea and J. 19, “Mobile os market share 2021,” <https://www.statista.com/statistics/272698/global-market-share-held-by-mobile-operating-systems-since-2009/>, visitado a 10-06-2022.
- [3] D. Nunes, J. S. Silva, C. Herrera, and F. Boavida, “Human-in-the-loop Connectivity Management in Smartphones,” in *14th International Conference on Wired/Wireless Internet Communication (WWIC)*, ser. Wired/Wireless Internet Communications, Thessaloniki, Greece, May 2016, pp. 159–170.
- [4] P. Carmona, D. Nunes, D. Raposo, D. Silva, J. S. Silva, and C. Herrera, “Happy Hour - Improving Mood With An Emotionally Aware Application,” in *2015 15TH INTERNATIONAL CONFERENCE ON INNOVATIONS FOR COMMUNITY SERVICES (I4CS)*, Roth, J and Eichler, G and Erfurth, C, Ed., 2015, 2015 15th International Conference on Innovations for Community Services (I4CS), Nuremberg, GERMANY, JUL 08-10, 2015.
- [5] B. R. Ranoliya, N. Raghuwanshi, and S. Singh, “Chatbot for university related faqs,” in *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, 2017, pp. 1525–1530.
- [6] DGEEC, <https://www.dgeec.mec.pt/np4/292>, visitado a 28-06-2022.
- [7] O. Tavares, C. Sin, D. Dias, and C. Rego, “Drop-out and completion among portuguese students,” in *EDULEARN18 Proceedings*, ser. 10th International Conference on Education and New Learning Technologies, <https://dx.doi.org/10.21125/edulearn.2018.0545>, visitado a 02-07-2022.
- [8] “How many phones are in the world,” <https://www.bankmycell.com/blog/how-many-phones-are-in-the-world>, visitado a 02-07-2022.

- [9] O. C. Novac, M. Novac, C. Gordan, T. Berczes, and G. Bujdosó, “Comparative study of google android, apple ios and microsoft windows phone mobile operating systems,” in *2017 14th International Conference on Engineering of Modern Electric Systems (EMES)*, 2017, pp. 154–159.
- [10] S. Li, L. D. Xu, and S. Zhao, “The internet of things: A survey - information systems frontiers,” *SpringerLink*, Apr 2014, <https://link.springer.com/article/10.1007/s10796-014-9492-7>.
- [11] W. Duo, M. Zhou, and A. Abusorrah, “A survey of cyber attacks on cyber physical systems: Recent advances and challenges,” *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 5, pp. 784–800, 2022.
- [12] D. S. Nunes, P. Zhang, and J. Sá 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, 2015.
- [13] E.-M. Schomakers, C. Lidynia, and M. Ziefle, “All of me? users’ preferences for privacy-preserving data markets and the importance of anonymity,” *Electronic Markets*, vol. 30, no. 3, pp. 649–665, 2020.
- [14] U. C. N. de Proteção de Dados, <https://www.cnpd.pt/>, visitado a 09-07-2022.
- [15] “Regulamento geral de proteção de dados,” <https://www.sg.pcm.gov.pt/sobre-nos/regulamento-geral-de-prote%C3%A7%C3%A3o-de-dados.aspx>, visitado a 09-07-2022.
- [16] “Android,” https://www.android.com/intl/pt_pt/, visitado a 16-06-2022.
- [17] “Android os market share of smartphone sales to end users from 2009 to 2021,” <https://www.statista.com/statistics/216420/global-market-share-forecast-of-smartphone-operating-systems/>, visitado a 07-06-2022.
- [18] “<https://www.techopedia.com/definition/25206/ios>.”
- [19] “.NET,” <https://dotnet.microsoft.com/en-us/learn/dotnet/what-is-dotnet>, visitado a 24-06-2022.
- [20] “Xamarin,” <https://docs.microsoft.com/en-us/xamarin/get-started/what-is-xamarin>, visitado a 24-06-2022.

- [21] “Xamarin forms,” <https://docs.microsoft.com/en-us/xamarin/get-started/what-is-xamarin-forms>, visitado a 24-06-2022.
- [22] “The future internet platform fiware,” <https://wayback.archive-it.org/12090/20161223145325/https://ec.europa.eu/digital-single-market/en/future-internet-public-private-partnership>, visitado a 23-06-2022.
- [23] “Fiware about-us,” <https://www.fiware.org/about-us>, visitado a 23-06-2022.
- [24] “Fiware catalogue,” <https://www.fiware.org/catalogue>, visitado a 23-06-2022.
- [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, <https://www.mdpi.com/2673-4001/1/1/3>.
- [26] “Studentlife,” <https://studentlife.cs.dartmouth.edu>, visitado a 23-05-2022.
- [27] R. Wang, F. Chen, Z. Chen, T. Li, G. Harari, S. Tignor, X. Zhou, D. Ben-Zeev, and A. T. Campbell, “StudentLife: Assessing Mental Health, Academic Performance and Behavioral Trends of College Students using Smartphones,” in *UBICOMP’14: PROCEEDINGS OF THE 2014 ACM INTERNATIONAL JOINT CONFERENCE ON PERVASIVE AND UBIQUITOUS COMPUTING*.
- [28] “Open source virtual classroom software,” Aug 2021, <https://bigbluebutton.org/>, visitado a 25-05-2022.
- [29] “Aleks,” <https://www.aleks.com/>, visitado a 25-05-2022.
- [30] “Socrative,” <https://www.socrative.com/>, visitado a 25-05-2022.
- [31] “Depression and college students,” <https://www.healthline.com/health/depression/college-students>, visitado a 26-05-2022.
- [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, <https://www.sciencedirect.com/science/article/pii/S246869642030001X>.
- [33] “A proteção de dados na ue,” https://ec.europa.eu/info/law/law-topic/data-protection/data-protection-eu_pt, visitado a 09-07-2022.

- [34] “How big is your smartphone’s battery?” <https://www.androidauthority.com/smartphone-battery-size-poll-results-1221015>, visitado a 04-07-2022.
- [35] “Manifest.permission,” <https://developer.android.com/reference/android/Manifest.permission>, visitado a 06-07-2022.
- [36] “ios security and privacy features,” <https://docs.microsoft.com/en-us/xamarin/ios/app-fundamentals/security-privacy?tabs=windows>, visitado a 06-07-2022.

Anexos

A Lista de permissões

Nesta secção em anexo é apresentada a lista de todas as permissões [35] e [36] utilizadas pela aplicação bem como a explicação da utilização de cada uma delas.

A.1 *Android*

- **PACKAGE_USAGE_STATS**: Permite ter acesso a vários parâmetros das aplicações utilizadas no *smartphone*.
- **ACCESS_NETWORK_STATE**: Permite verificar se o *smartphone* se encontra ligado à *internet*.
- **ACCESS_WIFI_STATE** e **CHANGE_WIFI_STATE**: Permite obter uma lista de todas as rede WiFi alcançadas e desligar ou ligar o WiFi.
- **ACCESS_FINE_LOCATION**, **ACCESS_COARSE_LOCATION**, **ACCESS_BACKGROUND_LOCATION** e **LOCATION_HARDWARE**: Permite ter acesso à localização do *smartphone*.
- **RECORD_AUDIO**: Garante acesso ao microfone do *smartphone*.
- **BLUETOOTH**, **BLUETOOTH_PRIVILEGED**, **BLUETOOTH_ADMIN**: Permite acesso ao *bluetooth* do *smartphone*.
- **BATTERY_STATS**: Permite verificar o estado da bateria.
- **WRITE_EXTERNAL_SOURCE**: Permite criar uma base de dados localmente no *smartphone*.
- **ACTIVITY_RECOGNITION**: Garante acesso à API *Google Activity Recognition*.
- **FOREGROUND_SERVICE** Permite que a aplicação seja iniciada em segundo plano.

A.2 *iOS*

- **NSLocationAlwaysUsageDescription:** Permite acesso à localização do *smartphone*.
- **NSMicrophoneUsageDescription:** Permite acesso ao microfone do *smartphone*.
- **NSBluetoothPeripheralUsageDescription:** Permite acesso ao *bluetooth* do *smartphone*.
- **NSHealthShareUsageDescription:** Permite acesso a dados de saúde do utilizador.
- **NSMotionUsageDescription:** Permite acesso ao sensor de acelerómetro.