



UNIVERSIDADE D  
COIMBRA

Rui Jorge Magueta Almeida

ESTUDO E DESENVOLVIMENTO DE SOLUÇÕES PARA GESTÃO  
UNIFICADA DE IoT  
DESENVOLVIMENTO DE UM *DASHBOARD*

Relatório de Estágio no âmbito do Mestrado em Engenharia  
Informática, especialização em Comunicações, Serviços e  
Infraestruturas orientada pelo Professor Doutor Fernando Boavida  
e pelo Professor Doutor Jorge Sá Silva e apresentada à Faculdade de  
Ciências e Tecnologia / Departamento de Engenharia Informática.

Setembro de 2020



Faculdade de Ciências e Tecnologia  
Departamento de Engenharia Informática

# ESTUDO E DESENVOLVIMENTO DE SOLUÇÕES PARA GESTÃO UNIFICADA DE IOT

## Desenvolvimento de um *Dashboard*

Rui Jorge Magueta Almeida

Relatório de Estágio no âmbito do Mestrado em Engenharia Informática, especialização em Comunicações, Serviços e Infraestruturas orientada pelo Professor Doutor Fernando Boavida e pelo Professor Doutor Jorge Sá Silva e apresentada à Faculdade de Ciências e Tecnologia / Departamento de Engenharia Informática.

Setembro de 2020



UNIVERSIDADE D  
COIMBRA



## Resumo

Atualmente, a quantidade de dados produzida ao longo de um dia é extremamente elevada. Um dos grandes responsáveis por este crescimento contínuo e acentuado deve-se aos dispositivos da Internet das Coisas (IoT). Estima-se que, em 2025, a quantidade de dados criada globalmente todos os dias será aproximadamente de 463 *exabytes*. Este valor é equivalente a cerca de 98,510,638 DVDs de dados por dia, sendo que cada DVD (de camada única) é capaz de armazenar 4.7 *Gigabytes* [1]. Com esta quantidade enorme de dados é extremamente importante fazer uma análise, para que os sistemas sejam cada vez mais orientados ao utilizador. Com a junção da grande quantidade de dados e a necessidade da sua análise e processamento, surgiu no início dos anos 90 o termo *Big Data*. Este termo serve para apoiar a análise e processamento de dados. [2]

O objetivo desta tese foi especificar, implementar e avaliar uma plataforma do tipo *dashboard* para agregar e apresentar dados IoT, aplicando-a e validando-a em três projetos: ISABELA, 5GOpenclasses e BATINA. Desenvolveu-se, assim, uma plataforma *front-end* capaz de apresentar dados recolhidos a partir de dispositivos IoT. O resultado final incluiu, no geral, e nas três plataformas, funcionalidades de ajuda aos alunos de forma a desenvolverem um estilo de vida mais saudável e melhorarem o seu desempenho académico, como também proporcionar o acesso a uma plataforma de questionários e vídeo-aulas *online* em tempo real.

## Palavras-Chave

IoT, HiTLCPS, Monitorização, Transmissão ao vivo, Questionários Online.



## **Abstract**

Nowadays, the amount of data produced each day is extremely high. One of the major drivers of this continued and sharp growth is IoT devices. By 2025, the amount of data created globally every day is estimated to be around 463 Exabytes. This is equivalent to about 98,510,638 data DVDs per day, taking into account that each single-layer DVD is capable of storing 4.7 *Gigabytes* [1]. With this huge amount of data, it is extremely important to analyze this data to make the systems increasingly user-oriented. With the combination of a large amount of data and the need for this to be analyzed and processed, the term Big Data was presented in the early 90s. This term was created to support data analysis and processing [2].

The purpose of this thesis was to specify, implement and evaluate a dashboard type platform to present IoT data, applying and validating it in three projects: ISABELA, 5GOpenclasses and BATINA. This thesis focused mainly on the development of a front-end platform capable of presenting data collected from IoT devices. In the context of these 3 projects, the result includes functionalities to help students to develop a healthier lifestyle and improve their academic performance, as well as providing access to a questionnaire platform and online video classes in real time.

## **Keywords**

IoT, HiTLCPS, Monitoring, Live Streaming, Online Questionnaires.





## **Agradecimentos**

Chegando ao final desta jornada, resta-me agradecer a todas as pessoas que fizeram parte de todo o meu percurso académico.

Dedico esta tese aos meus pais, irmã e namorada, por toda a força e apoio que me deram. Sem o vosso conforto e amor, tudo teria sido bem mais complicado.

Gostaria de agradecer aos meus orientadores, Professor Doutor Fernando Boavida e Professor Doutor Jorge Sá Silva, pelo suporte constante prestado ao longo desta dissertação e também aos meus colegas de laboratório, nomeadamente ao Marcelo Fernandes e Ngombo Armando, que sempre me ajudaram e apoiaram incondicionalmente. Foi a partilha da vossa experiência, conhecimento e companheirismo que tornou todo este trabalho possível.



# Índice

<b>Capítulo 1</b>	<b>Introdução</b>	<b>1</b>
1.1	Enquadramento e motivação	1
1.2	Objetivos do trabalho	2
1.3	Metodologia	3
1.4	Estrutura do documento	5
<b>Capítulo 2</b>	<b>Tecnologias de suporte e estado de arte</b>	<b>7</b>
2.1	Tecnologias	7
2.1.1	Python	7
2.1.2	FIWARE	7
2.1.3	Prometheus	8
2.1.4	Git	9
2.1.5	Kurento	9
2.2	Estado de arte	10
2.2.1	Conceitos	10
2.2.2	Ferramentas	10
<b>Capítulo 3</b>	<b>Especificação do sistema</b>	<b>17</b>
3.1	Contexto	17
3.1.1	5GOpenclasses	18
3.1.2	ISABELA	19
3.1.3	BATINA	20
3.2	Perfis de utilizadores	20
3.3	Permissões dos utilizadores	21
3.4	Requisitos	21
3.4.1	Requisitos funcionais	23
3.4.2	Requisitos não funcionais	30
3.5	Especificação dos componentes da arquitetura	30
3.6	Protótipos ( <i>Mockups</i> )	32
3.6.1	Administrador	32
3.6.2	Professor	44
<b>Capítulo 4</b>	<b>Implementação</b>	<b>49</b>
4.1	IoTUnifiedManagement	49
4.1.1	Integração com o Prometheus	51
4.1.2	Recolha de dados IoT	56
4.1.3	Funcionalidades 5GOpenclasses	58
4.2	Outras aplicações	65
<b>Capítulo 5</b>	<b>Demonstração e aplicação em projetos</b>	<b>71</b>
5.1	IoTUnifiedManagement	71

5.1.1 Administrador .....	72
5.1.2 Professor .....	81
5.2 Outras aplicações .....	87
<b>Capítulo 6 Conclusões e trabalho futuro .....</b>	<b>95</b>
<b>Referências .....</b>	<b>97</b>
<b>Apêndice A .....</b>	<b>103</b>

## Acrónimos

- API** – *Application Programming Interface* – Interface de programação de aplicações
- FPS** – *Frames per second* – Quadros por Segundo
- HTTP/HTTPS** – *Hypertext Transfer Protocol/ Hyper Text Transfer Protocol Secure*
- IT** – *Information Technology* – Tecnologia de Informação
- IoT** – *Internet of Things* – Internet das coisas
- JSON** – *JavaScript Object Notation*
- KPI** – *Key Performance Infrastructure* – Indicador-chave de performance
- M2M** – *Machine to Machine* – Máquina para máquina
- MEMS** – *Microelectromechanical Systems* – Sistemas micro-eletromecânicos
- NGSI** – *Next Generation Service Interfaces* – Interfaces de serviço de última geração
- OAuth 2.0** – *Open Authentication v2.0* – Autenticação livre v2.0
- OT** – *Operational Technology* – Tecnologia operacional
- QoE** – *Quality of Experience* – Qualidade de experiência
- REST** – *Representational State Transfer* – Transferência representacional de estado
- STUN** – *Session Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs)*
- TURN** – *Traversal Using Relays around NAT*
- WebRTC** – *Web Real-Time Communication* – Comunicações web em tempo real
- eMBB** – *enhanced Mobile Broadband* – Banda larga móvel melhorada
- mMTC** – *massive Machine Type Communications* – Comunicações massivas do tipo máquina
- urLLC** – *Ultra Reliable Low Latency Communications* – Comunicações de baixa latência ultra confiáveis



## Lista de Figuras

Figura 1 – Cronograma primeiro semestre (2019/20). .....	3
Figura 2 – Cronograma segundo semestre (2020). .....	4
Figura 3 – Tabela comparativa entre frameworks / aplicações. ....	15
Figura 4 – Arquitetura IoTUnifiedManagement. ....	17
Figura 5 – Arquitetura 5GOpenclasses [16]. ....	19
Figura 6 – Arquitetura atual ISABELA[5]. ....	19
Figura 7 – Arquitetura atual BATINA [17]. ....	20
Figura 8 – Divisão de perfis na aplicação “IoTUnifiedManagement”. ....	22
Figura 9 – Proposta visual da aplicação (1). ....	33
Figura 10 – Proposta visual da aplicação (2). ....	33
Figura 11 – Proposta visual da aplicação (3). ....	34
Figura 12 – Proposta visual da aplicação (4). ....	35
Figura 13 – Proposta visual da aplicação (5). ....	35
Figura 14 – Proposta visual da aplicação (6). ....	36
Figura 15 – Proposta visual da aplicação (7). ....	36
Figura 16 – Proposta visual da aplicação (8). ....	37
Figura 17 – Proposta visual da aplicação (9). ....	37
Figura 18 – Proposta visual da aplicação (10). ....	38
Figura 19 – Proposta visual da aplicação (11). ....	38
Figura 20 – Proposta visual da aplicação (12). ....	39
Figura 21 – Proposta visual da aplicação (13). ....	39
Figura 22 – Proposta visual da aplicação (14). ....	40
Figura 23 – Proposta visual da aplicação (15). ....	40
Figura 24 – Proposta visual da aplicação (16). ....	41
Figura 25 – Proposta visual da aplicação (17). ....	41
Figura 26 – Proposta visual da aplicação (18). ....	42
Figura 27 – Proposta visual da aplicação (19). ....	43
Figura 28 – Proposta visual da aplicação (20). ....	43
Figura 29 – Proposta visual da aplicação (21). ....	44
Figura 30 – Proposta visual da aplicação (22). ....	45
Figura 31 – Proposta visual da aplicação (23). ....	45
Figura 32 – Proposta visual da aplicação (24). ....	46

Figura 33 – Proposta visual da aplicação (25). .....	47
Figura 34 – Proposta visual da aplicação (26). .....	47
Figura 35 – Proposta visual da aplicação (27). .....	48
Figura 36 – Página de login “IoTUnifiedManagement”. .....	71
Figura 37 – Página de registo “IoTUnifiedManagement”. .....	72
Figura 38 – Página do administrador (1) “IoTUnifiedManagement”. .....	72
Figura 39 – Página do administrador (2) “IoTUnifiedManagement”. .....	73
Figura 40 – Página do administrador (3) “IoTUnifiedManagement”. .....	73
Figura 41 – Página do administrador (4) “IoTUnifiedManagement”. .....	74
Figura 42 – Página do administrador (4) “IoTUnifiedManagement”. .....	74
Figura 43 – Página do administrador (5) “IoTUnifiedManagement”. .....	75
Figura 44 – Página do administrador (6) “IoTUnifiedManagement”. .....	75
Figura 45 – Página do administrador (7) “IoTUnifiedManagement”. .....	76
Figura 46 – Página do administrador (8) “IoTUnifiedManagement”. .....	76
Figura 47 – Página do administrador (9) “IoTUnifiedManagement”. .....	77
Figura 48 – Página do administrador (10) “IoTUnifiedManagement”. .....	77
Figura 49 – Página do administrador (11) “IoTUnifiedManagement”. .....	78
Figura 50 – Página do administrador (12) “IoTUnifiedManagement”. .....	78
Figura 51 – Página do administrador (13) “IoTUnifiedManagement”. .....	79
Figura 52 – Página do administrador (14) “IoTUnifiedManagement”. .....	79
Figura 53 – Página do administrador (15) “IoTUnifiedManagement”. .....	80
Figura 54 – Página do administrador (16) “IoTUnifiedManagement”. .....	80
Figura 55 – Página do administrador (17) “IoTUnifiedManagement”. .....	81
Figura 56 – Página do administrador (18) “IoTUnifiedManagement”. .....	81
Figura 57 – Página do professor (1) “IoTUnifiedManagement”. .....	82
Figura 58 – Página do professor (2) “IoTUnifiedManagement”. .....	82
Figura 59 – Página do professor (3) “IoTUnifiedManagement”. .....	83
Figura 60 – Página do professor (4) “IoTUnifiedManagement”. .....	83
Figura 61 – Página do professor (5) “IoTUnifiedManagement”. .....	84
Figura 62 – Página do aluno (1) 5GOpenclasses. ....	85
Figura 63 – Página do aluno (2) 5GOpenclasses. ....	85
Figura 64 – Página do aluno (3) 5GOpenclasses. ....	86
Figura 65 – Página do aluno (4) 5GOpenclasses. ....	86
Figura 66 – Página de login BATINA. ....	87
Figura 67 – Página de registo BATINA. ....	87



Figura 68 – Página do professor (1) BATINA.....	88
Figura 69 – Página do professor (2) BATINA.....	88
Figura 70 – Página do professor (3) BATINA.....	89
Figura 71 – Página do professor (4) BATINA.....	89
Figura 72 – Página do professor (5) BATINA.....	90
Figura 73 – Página do professor (6) BATINA.....	90
Figura 74 – Página do professor (7) BATINA.....	91
Figura 75 – Página do professor (8) BATINA.....	91
Figura 76 – Página do professor (9) BATINA.....	92
Figura 77 – Página do professor (10) BATINA.....	92
Figura 78 – Página do professor (11) BATINA.....	93
Figura 79 – Página do professor (12) BATINA.....	93
Figura 80 – Página do professor (13) BATINA.....	94



## Lista de Tabelas

Tabela I – Permissões dos perfis de utilizador. ....	21
Tabela II – Contexto de apresentação dos requisitos.....	23
Tabela III – Estrutura de dados “IoTUnifiedManagement”.....	49
Tabela IV – Estrutura de dados BATINA.....	66



# Capítulo 1

## Introdução

### 1.1 Enquadramento e motivação

Atualmente a utilização da Internet das Coisas (IoT) é um termo que tem vindo a ganhar uma maior visibilidade na sociedade. Os dispositivos IoT são dispositivos conectados à Internet, e são capazes de se identificar na rede e de comunicar entre si. Estes dispositivos podem estar ou não em contacto com humanos, e têm como objetivo recolher uma quantidade vasta de informações relativas ao meio que os rodeia, nomeadamente através de sensores [3].

Para além disso, existem os Sistemas Ciber-Físicos, cujas operações são monitorizadas, coordenadas, controladas e integradas com um núcleo de computação e comunicação, e têm uma intensa ligação com o mundo físico envolvente. Combinando os dados que são recolhidos dos sensores da Internet das Coisas com os Sistemas Ciber-Físicos é possível criar um sistema capaz de controlar e monitorizar o mundo físico, atuando em contacto direto com o ser humano. A interação destes sistemas e o ser humano é conhecida pelo domínio do *Human-in-the-Loop Cyber-Physical-Systems* (HiTLCPS). Este é onde a tecnologia aceita o humano como parte do sistema e toma as intenções, ações, emoções e estados físicos dos humanos em consideração e cria um sistema que melhor atende às necessidades dos indivíduos [4], [5].

Esta tese descreve o trabalho que foi elaborado ao longo deste ano no desenvolvimento da aplicação “IoTUnifiedManagement”, sendo que esta irá interagir com diversos dados e sensores das aplicações ISABELA, 5GOpenclasses e BATINA.

A aplicação ISABELA encontra-se inserida no projeto *Social-Oriented Internet of Things Architecture, Solutions and Environment* (SOCIALITE) da Fundação para a Ciência e a Tecnologia (FCT). Este projeto é composto por uma aplicação móvel Android, uma aplicação *wearable* para *smartwatches* e também por dispositivos embebidos (*Raspberry Pi* e *Arduino*), cujo principal objetivo é implementar o conceito de HiTLCPS num sistema capaz de monitorizar os estudantes e de os ajudar a melhorar o seu desempenho académico. Este projeto focou-se principalmente no desenvolvimento da estrutura necessária para a obtenção e armazenamento da informação. Neste projeto, o *smartphone* e o *smartwatch* são utilizados para recolher informações sobre os estudantes e os dispositivos embebidos para recolher dados do ambiente. A partir desta informação são inferidos/classificados a atividade, localização, sociabilidade e padrões de sono. Para além disso, também são recolhidas informações de sensores virtuais/sociais, nomeadamente do Facebook.

A aplicação 5GOpenclasses encontra-se ainda em fase de desenvolvimento e está enquadrada no projeto nacional Mobilizador5G. Este projeto está focado na evolução de redes e serviços do 4G para 5G e aborda um novo conjunto de desafios do estudo e desenvolvimento de produtos e serviços de rede avançados baseados em 5G. Internamente este projeto encontra-se dividido em diferentes PPS's, sendo elas estruturas dedicadas a um determinado produto ou serviço a ser desenvolvido. Atualmente existem 6 PPS's, sendo que

o PPS1 é dedicado aos produtos e serviços para o *edge* da rede, o PPS2 dedicado aos produtos e serviços para o *core* da rede, o PPS3 dedicado aos produtos e serviços para comunicação M2M, o PPS4 dedicado aos produtos e serviços para comunicação humana, o PPS5 à demonstração e integração de resultados, e o PPS6 dedicado à gestão global do projeto. A aplicação 5GOpenclasses encontra-se inserida no PPS4 uma vez que pretende disponibilizar uma solução digital para dar suporte a aulas *online* através de *streaming* de vídeo, tendo sempre em consideração que este producto foi desenhado para explorar, em simultâneo, todos os casos de uso especificados na visão 5G, nomeadamente eMBB, urLLC e mMTC. Atualmente o 5GOpenclasses inclui duas aplicações de utilizadores finais, uma solução móvel Android para servir os estudantes e outra *web-based* para servir os professores. Para além disso, o 5GOpenclasses também é portador de um módulo de processo de qualidade de experiência dos utilizadores, capaz de medir/classificar a qualidade do serviço de *video-streaming* em tempo real. Prevê-se que a qualidade de vídeo e qualidade de experiência dos utilizadores cresça exponencialmente devido à evolução das redes 4G para 5G [6].

A aplicação BATINA (*Best Assisting Tutor and Interactive Advisor*) é uma aplicação financiada pela reitoria da Universidade de Coimbra e tem como objetivo a disponibilização de uma plataforma de questionários *online* em tempo real, que poderá ser utilizada tanto num contexto de uma aula presencial como numa aula remota. Atualmente, esta aplicação é composta por uma aplicação móvel disponível para Android e iOS que pretende servir os alunos, e uma aplicação *web-based* que pretende servir os professores. Numa visão mais detalhada, o BATINA oferece aos professores funcionalidades para gestão das suas disciplinas e seus questionários, gestão de materiais de suporte (associados a questionários e/ou disciplinas), acompanhamento de questionários ao vivo e também funcionalidades de gestão de respostas a questionários. Relativamente ao acompanhamento de questionários ao vivo, os professores têm a possibilidade de observar a totalidade dos estudantes que estão a responder ao questionário, a possibilidade de navegar entre as diferentes questões do questionário e também observar as respostas submetidas em tempo real. Para além disso, no que diz respeito à gestão de respostas a questionários, o BATINA disponibiliza uma vista dedicada à apresentação de respostas a questionários e também a possibilidade de impressão dessas respostas. Ambas as funcionalidades descritas anteriormente podem ser utilizadas apenas para um questionário específico ou para todos os questionários associados a uma determinada disciplina. Num futuro próximo, o BATINA será portador de novas funcionalidades tais como: um ChatBot responsável por efetuar recomendações básicas aos alunos (por exemplo, recomendar que conteúdo lecionado deverá estudar) e também responder a questões básicas feitas pelos alunos. Adicionalmente, também serão implementadas novas funcionalidades tais como: detetar quais os melhores/piiores estudantes, qual o melhor percurso académico para cada estudante, recomendar os melhores alunos finalistas para ofertas de emprego/teses e também detetar se um estudante poderá estar em risco de desistir do curso.

## 1.2 Objetivos do trabalho

O foco deste trabalho, cujo desenvolvimento será apresentado ao longo deste relatório, é a especificação, implementação e avaliação de uma plataforma do tipo *dashboard* que integra as aplicações ISABELA, 5GOpenclasses e BATINA, recolhendo e apresentando os dados IoT. Para além disso, o *dashboard* permite fornecer ferramentas capazes de elaborar vídeo-aulas em tempo real.

Num contexto mais objetivo, esta aplicação apresenta os dados IoT que estão a ser recolhidos nas aplicações ISABELA, 5GOpenclasses e BATINA (e armazenados nas plataformas FIWARE e Prometheus), passíveis de serem analisados, sendo que posteriormente serão efetuadas recomendações aos alunos. Relativamente aos dados recolhidos na aplicação ISABELA e 5GOpenclasses, estes são capturados através das respetivas aplicações móveis Android. Adicionalmente, os sensores considerados como sensores da aplicação “IoTUnifiedManagement”, são sensores presentes nas boxes ISABELA. Visto que essas boxes recolhem dados em salas de aula e essas informações são importantes para ambas as aplicações (ISABELA e 5GOpenclasses), estes serão considerados sensores da plataforma. Por último, a aplicação “IoTUnifiedManagement” é portadora de funcionalidades com capacidade de gravação e transmissão de vídeo em tempo real para a elaboração de vídeo-aulas. Estas funcionalidades são dedicadas à aplicação 5GOpenclasses e são suportadas por um servidor interno que executa o *software open-source* WebRTC Kurento. Estas vídeo-aulas podem ser observadas através da respetiva aplicação móvel Android, aplicação esta que se encontra em fase de desenvolvimento avançado.

### 1.3 Metodologia

Para este trabalho recorreu-se à metodologia SCRUM. Este método consiste na divisão do projeto em *sprints*, sendo que cada *sprint* é considerado como uma meta na divisão do projeto por partes. É habitual no início de cada *sprint* haver uma reunião de planeamento à qual o “proprietário do produto” (neste caso os orientadores) indica quais os requisitos (requisitos que pertencem ao *Product Backlog*) que irão ser implementados. Geralmente, é habitual existirem reuniões diárias chamadas *Daily Scrum*, sendo que o objetivo é apresentar o que foi feito no dia anterior e identificar o que irá ser feito no próprio dia. No final de cada *sprint*, são apresentadas as funcionalidades implementadas (numa *Sprint Review Meeting*) e é feita uma retrospectiva do presente *sprint* (*Sprint Retrospective*). No caso deste trabalho também se adotou esta metodologia (SCRUM), sendo o *Product Backlog* apresentado no subcapítulo 3.4 com cada *sprint* com a duração de uma semana. No início de cada semana foi feita uma reunião com o “proprietário do produto” (ambos os orientadores) para indicar os objetivos da presente semana e apresentar o que foi implementado na semana anterior. No final de cada *sprint* são apresentados os objetivos que foram estipulados e o “cumprimento ou não cumprimento” do mesmo. Para além disso, também era apresentada a evolução do projeto.

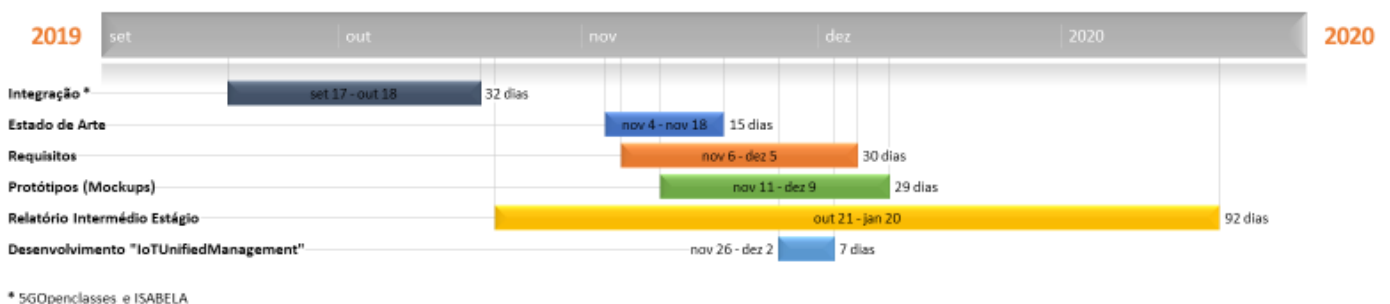


Figura 1 – Cronograma primeiro semestre (2019/20).

Relativamente ao cronograma do primeiro semestre, este encontra-se dividido entre diferentes tarefas: Integração, Estado de Arte, Requisitos, Protótipos (*Mockups*), Relatório Intermédio Estágio e Desenvolvimento “IoTUnifiedManagement”. No que diz respeito à Integração, esta foi composta pelo contacto e contextualização com as aplicações 5GOpenclasses e ISABELA (código fonte e toda a sua arquitetura) e também elaboradas pequenas implementações. Em relação ao Estado de Arte, como o próprio título indica, foi elaborado o estado de arte relativo às ferramentas existentes para o desenvolvimento da aplicação que pretendemos. Dentro do mesmo contexto, também foram elaborados os Requisitos e Protótipos para esta mesma aplicação. Adicionalmente, foi elaborado o Relatório de Avaliação Intermédia do presente Estágio. Por fim, foram elaborados alguns desenvolvimentos na “IoTUnifiedManagement”, tendo sempre em consideração a ferramenta escolhida no Estado de Arte presente no subcapítulo 2.2. Como podemos identificar na Figura 1, inicialmente foi feita uma integração com as aplicações 5GOpenclasses (algumas implementações na aplicação 5GOpenclasses), seguido da elaboração do relatório intermedio. Desde o início do mês de novembro até à metade do mês de dezembro, foi elaborada a análise do estado de arte, em conjunto com os requisitos e protótipos da aplicação. Como podemos verificar, existe um período em que as 3 tarefas estavam a decorrer ao mesmo tempo, sendo que entre os dias 4 de novembro e 18 de novembro, o foco principal era a tarefa Estado de Arte, e a partir do 18 de novembro, o foco ficou sempre dividido entre as tarefas Requisitos e Protótipos. Para além disso, conseguimos identificar que, ao longo de todo o semestre, foram elaborados pequenos desenvolvimentos no Relatório Intermédio, tomando esta maior importância a partir do meio do mês de dezembro.

A Figura 2 ilustra o cronograma relativo ao segundo semestre.

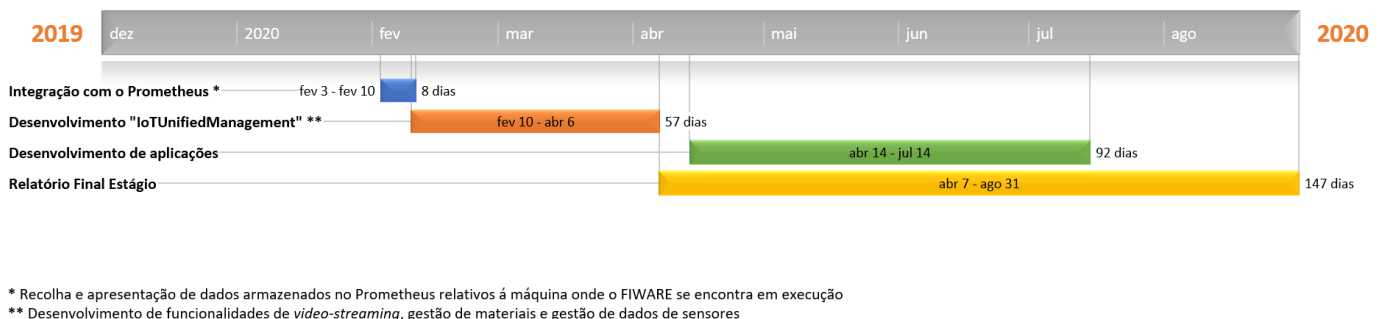


Figura 2 – Cronograma segundo semestre (2020).

Como podemos identificar, a segunda fase do estágio teve início no dia 3 de fevereiro e terminou no final do mês de agosto. Nesta fase, as principais tarefas foram divididas entre desenvolvimento da aplicação “IoTUnifiedManagement”, desenvolvimento da aplicação BATINA *web-based* e a escrita do relatório final do estágio/dissertação.

No desenvolvimento da aplicação, foi feita inicialmente a integração entre a aplicação “IoTUnifiedManagement” e a aplicação Prometheus, tendo sempre em consideração que esta aplicação é responsável por recolher dados de informações da máquina onde executamos os nossos serviços. Após esta etapa, foi elaborado o desenvolvimento de funcionalidades gestão de disciplinas e materiais de suporte, procedidas, não só pela implementação do servidor *open-source* WebRTC Kurento, como também pela recolha e apresentação de dados obtidos pelos sensores móveis (a partir das aplicações moveis



ISABELA e 5GOpenclasses) e fixos (boxes). Relativamente ao desenvolvimento de aplicações, foram elaborados módulos na aplicação *web-based* BATINA. Numa primeira fase foram implementadas as funcionalidades de *back-end* inicialmente desejadas, nomeadamente a definição da estrutura dos objetos de dados a utilizar com *tokens* de acesso ao serviço de armazenamento de dados (acesso via HTTPS) e foram elaboradas as funcionalidades de registo e autenticação dos utilizadores via OAuth2.0. Numa segunda fase de desenvolvimento da aplicação BATINA, foram implementadas funcionalidades de gestão de disciplinas e dos seus materiais de suporte, funcionalidades de visualização de questionários a decorrer ao vivo (atualizados em tempo real), funcionalidades de apresentação dos resultados dos alunos para um questionário numa vista dedicada (incluindo informações de médias de resultados, resultados por questão, resultados por aluno, etc.) e também funcionalidades de impressão de resultados (impressão apenas para um questionário ou para todos os questionários de uma disciplina).

Por último, no decorrer do segundo semestre, a escrita do relatório final do estágio/dissertação foi elaborada a partir do dia 7 de abril, tomando maior importância a partir do início do mês de julho.

### 1.4 Estrutura do documento

O documento encontra-se dividido em seis capítulos: introdução, tecnologias de suporte e estado de arte, especificação do sistema, implementação, demonstração e aplicação em projetos, conclusões e trabalho futuro.

No primeiro capítulo é apresentado o enquadramento e a motivação do presente documento, quais os objetivos principais desta tese e qual a metodologia que utilizámos ao longo da mesma.

No segundo capítulo foram descritas as informações relativas às tecnologias de suporte complementares da aplicação “IoTUnifiedManagement”, incluindo um resumo explicativo breve para cada uma. Foi ainda abordado o tema relativo ao estado de arte, onde foram apresentados os conceitos e ferramentas que fizeram parte do projeto. No final foi feita a seleção de uma ferramenta que ficou responsável pelo desenvolvimento da aplicação “IoTUnifiedManagement”.

No que diz respeito ao terceiro capítulo, este irá abordar informações relativas à especificação da aplicação “IoTUnifiedManagement”. Neste capítulo é apresentado um contexto da aplicação, os seus perfis de utilizador, as permissões para cada perfil, os requisitos funcionais e não funcionais, a especificação dos componentes da arquitetura e também uma proposta visual para a aplicação “IoTUnifiedManagement”.

No quarto capítulo serão explicados os detalhes de toda a implementação da aplicação “IoTUnifiedManagement” e da aplicação BATINA, incluindo informações técnicas (ex: Kurento WebRTC, MediaServerHelper, etc.) de todos os componentes de cada aplicação.

O quinto capítulo será responsável por apresentar uma demonstração de ambas as aplicações e de todas as funcionalidades nelas implementadas.

No sexto e último capítulo é feita a apresentação do resumo do trabalho elaborado ao longo do ano, assim como, informações relativas ao trabalho futuro para a aplicação.



# Capítulo 2

## Tecnologias de suporte e estado de arte

Este capítulo tem como objetivo a apresentação das tecnologias de suporte utilizadas ao longo de todo o projeto. Como o principal componente a utilizar neste projeto é uma aplicação em Python, a tecnologia mais utilizada neste projeto será o IntelliJ IDEA. Para além disso, são apresentados os conceitos e o estado de arte envolvidos neste projeto. No final deste capítulo, será ainda apresentada a escolha da *framework* selecionada para o desenvolvimento da aplicação “IoTUnifiedManagement”.

### 2.1 Tecnologias

#### 2.1.1 Python

Python é uma linguagem de programação de alto nível, interpretada, orientada a objetos e com semântica dinâmica. Esta é uma linguagem muito atrativa para “aplicações de desenvolvimento rápido” e também é muito usada como linguagem de *scripting* que conecta componentes já existentes. Para além disso, o Python é uma linguagem que suporta módulos e pacotes, o que incentiva a modularidade e reutilização do código [7].

Esta linguagem de programação, pertencente à organização sem fins lucrativos Python Software Foundation, é atualmente uma das linguagens de programação mais utilizada. Esta é conhecida pela alta produtividade que proporciona aos programadores. Cerca de 46% dos programadores em Python usam esta linguagem para programação web, e cerca de 41% usam a biblioteca/*framework* Flask para desenvolvimento web [8].

Para o desenvolvimento deste trabalho utilizou-se a linguagem de programação Python em conjunto com a *framework* Dash (escrita por cima de Flask), e iremos utilizar o IntelliJ IDEA com uma versão de SDK Python v3.7.6.

#### 2.1.2 FIWARE

O FIWARE é uma *framework* de componentes de plataforma *open-source* que facilita e acelera o desenvolvimento de soluções inteligentes. A plataforma FIWARE fornece um conjunto simples e poderoso de *APIs* que facilitam o desenvolvimento de aplicações inteligentes. As especificações dessas *APIs* são públicas e isentas de *royalties*.

Atualmente o FIWARE conta com seis principais características:

- *Open-source - software* de código aberto pronto para o mercado que permite a conexão de componentes IoT com serviços de gestão de informações de contexto e serviços de *BigData* na *cloud*;

- Uso inteligente de dados - disponibiliza *APIs* padrão para gestão e troca de dados, bem como modelos de dados "harmonizados";
- Soluções e serviços inteligentes - automação de processos ao longo de toda a sua cadeia de valor, fácil integração *plug & play* com outros serviços e soluções;
- Fundação FIWARE - promove ativamente a adoção do FIWARE, apoia a comunidade, fornece recursos partilhados e valida tecnologias FIWARE;
- Cimeiras FIWARE - um local de encontro para programadores, empreendedores, políticos, líderes de opinião, executivos e investidores;
- Ecossistema FIWARE - mais de 100 cidades, 13 *iHubs*, vários programas de aceleração e parcerias estratégicas com: GSMA, TM Fórum, CEF, ETSI, entre outros.

A comunidade FIWARE é uma comunidade independente e aberta, cujos membros estão comprometidos em realizar a missão do FIWARE – “construir um ecossistema aberto e sustentável em torno de padrões de plataforma de *software* públicos, isentos de *royalties* e orientados à implementação, que facilitarão o desenvolvimento de novas aplicações inteligentes em múltiplos setores” [9], [10].

### 2.1.3 Prometheus

O Prometheus é um sistema de monitorização e alerta *open-source* originalmente construída no SoundCloud. Desde 2012 muitas empresas e organizações adotaram o Prometheus, sendo que este projeto possui uma comunidade de programadores e utilizadores muito ativa. O Prometheus ingressou na Cloud Native Computing Foundation em 2016. Este sistema armazena dados numa linha temporal, denominada internamente como “dados de series temporais” [11].

Algumas das principais características são:

- Modelo de dados multidimensional com dados de series temporais pelo nome e par *key/value*;
- Usa uma linguagem de consulta PromQL;
- A recolha de dados de series temporais ocorre sobre HTTP (através de um *gateway* intermediário);
- Vários suportes a gráficos e painéis.

Para além destas características o Prometheus conta com um ecossistema com múltiplos componentes, como os seguintes:

- Um servidor Prometheus principal onde se armazenam os dados de séries temporais;
- Bibliotecas clientes para executar códigos de aplicações (bibliotecas para trabalhar como cliente);
- Um *gateway* de envio para suportar trabalhos de curta duração;
- Exportadores de dados como HAProxy, StatsD, Graphite, etc.;
- Um gestor para controlar os alertas.

### 2.1.4 Git

O Git é um sistema *open-source* de controlo de versões distribuído, desenhado para controlar versões de *software* ao longo do seu desenvolvimento. Este pode ser utilizado como armazenamento de dados caso algo ocorra ao nosso computador.

Algumas das principais características são:

- Troca de contexto sem problemas - criar um novo ramo (*branch*) para testar uma ideia, une esse ramo com outro, caso pretenda, etc.;
- Código baseado em regras - ter um ramo com o código principal e outro onde se evolui e testa novas funcionalidades;
- Fluxo de trabalho baseado em funcionalidades - criar novos ramos para novas funcionalidades, para que seja possível alternar entre eles;
- Experiência descartável - criar um novo ramo para experimentar, e no caso de este não funcionar, exclua-o.

Para além destas funcionalidades, este sistema fornece-nos a possibilidade de trabalhar num projeto com um determinado número de pessoas, indica-nos que alterações foram feitas, quem fez essas alterações, que linhas do código foram afetadas, entre outras. Isto ajuda-nos a controlar e a gerir a evolução de um projeto [12], [13].

No nosso caso real utilizamos uma instância local que pode ser acedida por outros membros. Para além disso, este sistema também ajuda a controlar quais as tarefas que se encontram feitas, tendo em consideração os diferentes requisitos do sistema.

### 2.1.5 Kurento

O Kurento é servidor multimédia WebRTC e um conjunto de APIs clientes que simplificam o desenvolvimento de aplicações de vídeo avançadas para plataformas web e *smartphones*. Este sistema *open-source*, foi lançado sobre os termos da Apache License v2.0 e o seu código está disponível no Git.

Algumas das principais características são:

- Comunicações de grupo;
- *Transcoding* e *broadcasting*;
- Gravação do conteúdo transmitido;
- Encaminhamento de fluxos audiovisuais;
- Processamento flexível;
- Realidade aumentada;
- Recursos avançados de processamento de *média*.

Neste trabalho foi utilizado o servidor multimédia WebRTC Kurento para difundir vídeo em tempo real a partir do professor para os alunos, utilizando uma arquitetura de transmissão de 1 para **muitos** (*One to Many video call*). Este servidor WebRTC encontra-se suportado por outros dois servidores (STUN/TURN e Media-Server-Helper), um responsável por estabelecer ligações seguras entre dispositivos e outro como auxiliar para clientes web.

## 2.2 Estado de arte

### 2.2.1 Conceitos

Neste projeto é importante salientar os seguintes conceitos: IoT e *Human-in-the-Loop Cyber-Physical-Systems* (HiTLCPS).

O IoT refere-se aos dispositivos conectados à Internet, capazes de se identificarem na rede e de comunicarem entre si. Estes dispositivos podem estar ou não em contacto com seres humanos, mas têm o objetivo de recolher uma quantidade vasta de informações relativas ao meio que os rodeia, nomeadamente através de sensores. Atualmente existe muito estudo e desenvolvimento nas tecnologias do IoT, nomeadamente na utilização controlada das baterias destes dispositivos. Este conceito foi inicialmente apresentado pelo cofundador do Auto-ID Center do MIT, Kevin Ashton em 1999, para a empresa Procter & Gamble. O IoT evoluiu da convergência de tecnologias sem fio, sistemas micro eletromecânicos (MEMS), micro serviços e da Internet. Esta convergência permitiu derrubar a parede entre a tecnologia operacional (OT) e a tecnologia da informação (TI), permitindo que dados não estruturados gerados por máquina possam ser analisados. Com o rápido desenvolvimento e utilização dos IoT, conta-se que em 2020 estejam cerca de 38.5 bilhões de dispositivos conectados à internet [14].

Relativamente ao *Human-in-the-Loop Cyber-Physical-Systems* (HiTLCPS), esta tecnologia aceita o ser humano como parte do sistema e toma as intenções, ações, emoções e estados físicos dos humanos em consideração e cria um sistema que melhor atenda às necessidades dos seres humanos. O HiTLCPS permite que a Internet transmita emoções, estados psicológicos, ações e impulsione os seres humanos como parte de sistemas de larga escala. Assim, as abordagens de deteção e atuação vão além dos dispositivos físicos eletrónicos. Por outras palavras, estes sistemas contam com a recolha de dados de ambiente através de sensores que estão ou não com o ser humano, e que seja efetuada uma análise desses dados e possa ser possível atuar no meio ambiente [15].

### 2.2.2 Ferramentas

Em relação às ferramentas (*frameworks*/aplicações) existentes para o desenvolvimento da aplicação “IoTUnifiedManagement”, foram avaliadas as *frameworks* Dash by Plotly, Cube.js, Dashing, e as aplicações Knowage e SingApp (template do Bootstrap). Para uma escolha mais decisiva e assertiva, escolhemos as seguintes métricas para análise:

- Linguagem de programação;
- Flexibilidade (novos gráficos/conteúdos ou mesmo integração com outras *frameworks*);
- Design responsivo (utilização coerente em qualquer dispositivo);
- Comunidade de suporte;
- Documentação;
- Integração direta com FIWARE;
- Atuação com sensores (funcionalidades que contactem diretamente com sensores em Android ou *Raspberry Pi*);
- Ferramentas *open-source*;
- Recomendação direta pelo FIWARE.

## Dash by Plotly

O Dash by Plotly é uma *framework open-source* dedicada para construção de aplicações web analíticas de dados, desenvolvida sobre Flask, Plotly.js e React.js para ser trabalhada em Python ou R. Esta é ideal para desenhar aplicações web de visualização de dados com interfaces dedicadas ao utilizador altamente personalizáveis, sendo esta interface a ideal para quem pretende trabalhar com dados em Python. Esta *framework* encontra-se coberta por uma licença do MIT.

### Pontos Fortes:

- A *framework* DASH é considerada como declarativa e reativa, o que facilita a implementação de diferentes elementos estéticos (gráficos) complexos que podem interagir entre si, originando diversos "outputs" dinâmicos para o utilizador;
- Os elementos estéticos são configuráveis, levando à possibilidade de alteração do tamanho, posição, cor, etc... dos diferentes elementos gráficos que compõem a interface web;
- O DASH já fornece integração dos elementos HTML, CSS, etc... quando *renderizado* na web;
- Livre arbítrio para a construção da própria interface do utilizador para análise de dados;
- Quando um "input" é alterado, os elementos estéticos (gráficos) são atualizados com o novo valor desse "input";
- As comunicações das aplicações DASH são feitas utilizando pacotes JSON complementados por pedidos HTTP/HTTPS, o que é favorável para a nossa parte, pois os pedidos/respostas que pretendemos utilizar para integrar a nossa aplicação com o FIWARE serão efetuados nesse formato (JSON) através de serviços web RESTful (*GET, PUT, POST e DELETE*);
- Possibilidade de integração direta com componentes de design do Bootstrap através da biblioteca *Dash Bootstrap Components*;
- Possibilidade de integração direta com a biblioteca de gráficos *open-source* Plotly.

### Pontos Fracos:

- Possível necessidade de implementar novos gráficos em React.js e JavaScript;
- Possibilidade de alguns problemas que não consigamos resolver sem conhecer como o DASH funciona internamente;
- Até ao momento, ainda não foi implementada nenhuma integração direta com o FIWARE;
- Esta *framework* não é recomendada diretamente pelo FIWARE;
- Não é portadora de funcionalidades de atuação em sensores.

## Cube.js

O Cube.js é uma *framework open-source* para desenvolvimento de aplicações web analíticas de dados, que foi desenhada para trabalhar com conjuntos de dados de grande escala. Esta *framework* foi desenhada para trabalhar com mecanismos de consulta sem servidor (*Serverless Query Engines*) e também fornece módulos para executar “transformações” e modelação de dados em *data warehouses*. Esta *framework* encontra-se coberta por uma licença do MIT.

### Pontos Fortes:

- O Cube.js disponibiliza uma estrutura que fornece uma *API* para consultas (*queries*) à base de dados, oferece segurança na troca destes dados, oferece cache dos resultados de consultas (*queries*) e execução "orquestrada" de consultas (*queries*) à base de dados (*query results caching and execution orchestration*). Adicionalmente, esta estrutura oferece pré-agregação de dados e produção/geração/formação de SQL analítico;
- Possibilidade de integração direta com componentes de design do Bootstrap através da biblioteca *reactstrap*;
- Possibilidade de integração direta com a biblioteca de gráficos *open-source* Recharts e BizCharts.

### Pontos Fracos:

- Aprendizagem profunda da linguagem JavaScript;
- Até ao momento, ainda não foi implementada nenhuma integração direta com o FIWARE;
- Esta *framework* não é recomendada diretamente pelo FIWARE;
- Não é portadora de funcionalidades de atuação em sensores.

## Dashing

O Dashing é uma *framework open-source* dedicada à construção de interfaces do utilizador (*dashboards*) que poderá ser integrada na representação de elementos estéticos (gráficos) que facilitam a análise crítica de dados. Esta usa *widgets* (elementos estéticos) pré-desenvolvidos pelos programadores da ferramenta para apresentar dados que são previamente recolhidos a partir de uma *API* da nossa base de dados. Esta *framework* encontra-se coberta por uma licença do MIT.

### Pontos Fortes:

- A interface do utilizador (*dashboard*) é portadora de uma funcionalidade que promove a edição dinâmica do posicionamento dos elementos estéticos (gráficos) apresentados nesta interface (*dashboard*);
- Os elementos estéticos (gráficos) são atualizados frequentemente quando existem alterações nos "*inputs*" a partir da recolha de dados direta na *API*;
- Possibilidade de integração direta com novos gráficos *open-source* propostos pelo Dashing e da autoria de inúmeros desenvolvedores.



### Pontos Fracos:

- Caso a nossa base de dados não for portadora de uma *API*, apenas é disponibilizado um simples conector desenvolvido em ruby DSL para obter os dados da mesma, sendo que estes (conector e base de dados) podem não ser compatíveis;
- Para implementar novos *widgets* (elementos estéticos), será necessário utilizar as linguagens SCSS, HTML e CoffeeScript;
- Interface do utilizador não atrativa;
- Até ao momento, ainda não foi implementada nenhuma integração direta com o FIWARE;
- Esta *framework* não é recomendada diretamente pelo FIWARE;
- Não é portadora de funcionalidades de atuação em sensores.

## Knowage

O Knowage é uma aplicação *open-source* dedicada à análise de negócios (*Business Analytics*) que oferece uma interface do utilizador (*dashboard*) muito completa. Esta aplicação fornece autonomia ao utilizador, para que este seja capaz de construir a sua própria análise dos dados fornecidos pela base de dados, organizar e explorar esses mesmos dados. Esta aplicação também fornece vários módulos, sendo que cada módulo foi concebido para um domínio analítico específico. Esses domínios são os seguintes: *Big Data*, onde é possível analisar dados em grande quantidade; *Smart Intelligence*, onde é possível expressar a inteligência “comercial” de uma forma mais explicativa graficamente; *Enterprise Reporting*, onde é possível produzir relatórios estáticos explicativos; *Location Intelligence*, onde é possível relacionar dados comerciais com informações geográficas; *Performance Management*, onde é possível gerir *PKIs* (Indicador-chave de performance) e também *Scorecards* (são indicadores de desempenho) e *Predictive Analysis* onde é possível efetuar análises mais avançadas. Esta aplicação encontra-se coberta por uma licença Affero GPL v3.

### Pontos Fortes:

- É uma ferramenta poderosa, pois é composta por muitos elementos importantes como: elementos estéticos (gráficos) interativos e reativos com outros elementos, gestão de *PKIs* (Indicador-chave de performance) e *scorecards* (são indicadores de desempenho), relatórios explicativos, entre outros;
- Disponibilização componentes de operação de serviços web RESTful (*GET*, *PUT*, *POST* e *DELETE*) integradas na aplicação;
- Tem integração direta com o FIWARE e também é aconselhada pelo mesmo.

### Pontos Fracos:

- Ferramenta fechada, sem possibilidade de edição do código fonte;
- Para integrar novos gráficos, será necessário criar uma nova contribuição no GitHub do Knowage e desenvolver esses mesmos gráficos (com ajuda profissional);
- Não é portadora de funcionalidades de atuação em sensores.

## SingApp

A SingApp é uma aplicação que fornece um *template open-source* de uma interface de utilizador (*dashboard*) muito completa, onde é possível editar o código fonte para modificar/integrar preferências estéticas e também novas funcionalidades para solucionar as nossas necessidades. Esta aplicação de *templates* é disponibilizada para trabalhar em diferentes linguagens de programação tais como: ReactJS, Angular, AngularJS, HTML5+CSS3 e também Vue JS.

### Pontos Fortes:

- Liberdade de implementação e escolha de linguagem de desenvolvimento;
- Possibilidade de integração de novos componentes estéticos (gráficos), funcionalidades (ex: *drag-and-drop*), etc;
- Possibilidade de integração com outras linguagens de programação;
- Possibilidade de integração direta com a biblioteca de gráficos *open-source* Chart.js e canvasJS;
- Possibilidade de integração direta com componentes de design *open-source* com Bootstrap 4.

### Pontos Fracos:

- Necessidade de implementação de todo o funcionamento por detrás da interface;
- Até ao momento, ainda não foi implementada nenhuma integração direta com o FIWARE;
- Esta aplicação não é recomendada diretamente pelo FIWARE;
- Não é portadora de funcionalidades de atuação em sensores.

## Capítulo 2

Em suma, o resumo comparativo das diferentes *frameworks*/aplicações é o seguinte:

	Linguagem de programação	Flexibilidade (novos gráficos)	Responsivo (apresentação web)	Comunidade de Suporte	Documentação	Integração direta com FIWARE	Atuação com sensores	"Livre"	Pontos Fortes	Pontos Fracos	Recomendação direta pelo FIWARE
<b>DASH</b>	Python ou R	✓	✓	✓	✓	✗	Necessário implementar	✓	<ul style="list-style-type: none"> <li>• Atualizações dinâmicas de conteúdo quando alterado o "input"</li> <li>• Integração direta com bibliotecas de gráficos e componentes de design</li> <li>• Renderização automática de elementos HTML, CSS, etc...</li> </ul>	<ul style="list-style-type: none"> <li>• O não conhecimento do funcionamento interno do DASH pode não solucionar alguns problemas internos</li> </ul>	✗
<b>CUBE</b>	JavaScript	✓	✓	✓	✓	✗	Necessário implementar	✓	<ul style="list-style-type: none"> <li>• Disponibilização de esquema de infraestrutura para análise de dados de grande escala</li> <li>• Integração direta com bibliotecas de gráficos e componentes de design</li> </ul>	<ul style="list-style-type: none"> <li>• Necessidade de aprendizagem profunda em JavaScript</li> </ul>	✗
<b>DASHING</b>	Ruby e Coffeescript	✓	✓	✓	✓	✗	Necessário implementar	✓	<ul style="list-style-type: none"> <li>• Edição dinâmica do posicionamento dos gráficos</li> <li>• Atualizações dinâmicas de conteúdo quando alterado o "input"</li> <li>• Integração direta com bibliotecas de gráficos</li> </ul>	<ul style="list-style-type: none"> <li>• Novos gráficos em SCSS, HTML e CoffeeScript</li> <li>• Apenas disponibilizado um simples conector desenvolvido em ruby DSL para conexão com a base de dados</li> <li>• Interface do utilizador não atrativa</li> </ul>	✗
<b>KnowAge</b>	Java	Necessário implementar	✓	✓	✓	✓	Necessário implementar	✓	<ul style="list-style-type: none"> <li>• Ferramenta poderosa</li> <li>• Recomendada diretamente pelo FIWARE</li> <li>• Disponibilização serviços web RESTful integrados (GET, PUT, POST e DELETE)</li> </ul>	<ul style="list-style-type: none"> <li>• Ferramenta "fechada" sem possibilidade de edição</li> <li>• Dificuldade/Impossibilidade na integração de novos gráficos</li> </ul>	✓
<b>SingApp</b>	HTML, CSS, Javascript, etc...	✓	✓	✓	✓	✗	Necessário implementar	✓	<ul style="list-style-type: none"> <li>• Flexibilidade na implementação</li> <li>• Integração direta com bibliotecas de gráficos e componentes de design</li> </ul>	<ul style="list-style-type: none"> <li>• Implementação de raiz</li> </ul>	✗

Figura 3 – Tabela comparativa entre frameworks / aplicações.

Relativamente à *framework*/aplicação escolhida para efetuar o desenvolvimento da aplicação "IoTUnifiedManagement", tendo em consideração todas as métricas indicadas acima e os seus pontos fortes/fracos, foi a *framework* Dash by Plotly.

Tal como foi descrito anteriormente, esta *framework open-source* é dedicada para construção de aplicações web analíticas de dados altamente personalizáveis, sendo a sua interface ideal para quem pretende trabalhar com dados em Python. Escolhemos esta *framework* como opção principal para o desenvolvimento da aplicação "IoTUnifiedManagement", pois esta pode ser integrada diretamente com elementos gráficos

desenvolvidos pela Plotly em Python. Para além disso, pode ser integrada diretamente com elementos de design do Bootstrap através da biblioteca Dash Bootstrap Components. Por outro lado, é do meu interesse melhorar e aperfeiçoar a minha aprendizagem na linguagem Python. Relativamente à integração com o FIWARE, esta pode ser elaborada diretamente através de operações de serviços web RESTful (*GET*, *PUT*, *POST* e *DELETE*).

Comparativamente com as outras *frameworks*/aplicações, conseguimos perceber que existem algumas “limitações” nas próprias *frameworks*/aplicações que poderão afetar o desenvolvimento da nossa aplicação. Por exemplo, a *framework* Cube.js é uma *framework* para desenvolvimento em JavaScript com uma arquitetura analítica de *back-end* um pouco peculiar. Por outro lado, a *framework* Dashing é uma *framework* composta por um baixo número de elementos gráficos que impossibilitam uma apresentação de dados satisfatória, sendo que a elaboração de novos elementos gráficos obrigaria a aprendizagem e implementação através das linguagens SCSS, HTML e CoffeeScript. Relativamente ao Knowage, esta é claramente uma aplicação muito completa, mas o seu código fonte é fechado e não é possível modelar essa aplicação com novos elementos que poderão ser necessários na aplicação “IoTUnifiedManagement”. Por fim, a aplicação SingApp é apenas um *template* do Bootstrap 4 que nos oferece livre arbítrio para desenvolver a aplicação a nosso gosto. Contudo, seria necessário implementar toda uma aplicação de raiz.

# Capítulo 3

## Especificação do sistema

Este capítulo tem como objetivo a apresentação do contexto da aplicação “IoTUnifiedManagement” e das aplicações que a suportam (ISABELA, 5GOpenclasses e BATINA). Adicionalmente, será feita a explicação dos perfis de utilizadores e as suas permissões na aplicação, apresentação dos seus requisitos funcionais e não funcionais, especificação dos componentes da sua arquitetura, assim como uma apresentação de protótipos visuais da mesma.

### 3.1 Contexto

Para um melhor enquadramento, esta dissertação tem como objetivo a especificação, implementação e avaliação de uma plataforma do tipo *dashboard* para agregar e apresentar dados IoT, aplicando-a e validando-a em três projetos: ISABELA, 5GOpenclasses e BATINA. Para que uma plataforma do tipo *dashboard* consiga trabalhar com dados IoT, esta precisa de aplicações que suportem o fornecimento desses dados. Para tal, uma parte do tempo do segundo semestre desta tese foi dedicado à implementação de uma nova aplicação denominada BATINA.

A aplicação “IoTUnifiedManagement” conta com a seguinte arquitetura:

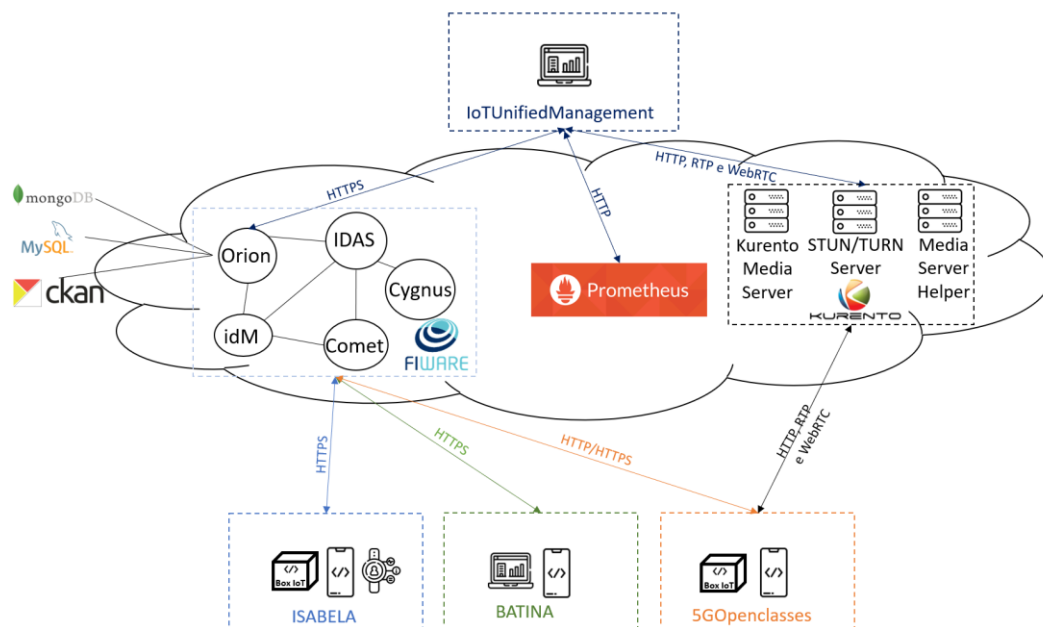


Figura 4 – Arquitetura IoTUnifiedManagement.

Ao proceder à análise da Figura 4, é possível identificar as diferentes aplicações que atualmente suportam a aplicação “IoTUnifiedManagement”: ISABELA, 5GOpenclasses e

BATINA. Para além disso, é também perceptível que atualmente todas as comunicações com o *middleware* FIWARE foram atualizadas para utilização do protocolo HTTPS na troca de dados. O *middleware* FIWARE é responsável por armazenar/fornecer todos os dados relativos à aplicação “IoTUnifiedManagement” como também das aplicações de suporte (ISABELA, 5GOpenclasses e BATINA). Em relação ao Prometheus, este é um sistema de monitorização e de armazenamento de dados numa resolução temporal. Neste projeto em específico, este sistema está a ser utilizado para armazenar informações dos estados dos serviços do servidor, informações de percentagem de utilização dos componentes do servidor, informações dos componentes do servidor e informações do tempo de resposta, percentagem de utilização do CPU e percentagem de utilização da memória RAM numa resolução temporal de 24 horas. O servidor multimédia WebRTC Kurento é responsável pela transmissão de vídeo-aulas em tempo real entre os diferentes utilizadores da aplicação 5GOpenclasses. Nesta arquitetura, tanto *middleware* FIWARE (e seus módulos internos) como o Prometheus e o Kurento (e seus servidores internos) encontram-se atualmente alojados num servidor na *cloud*.

### 3.1.1 5GOpenclasses

No que diz respeito à aplicação 5GOpenclasses, esta inclui atualmente duas aplicações de utilizadores finais: uma solução móvel Android para servir os estudantes e outra *web-based* para servir os professores.

Através da análise da Figura 5 identificamos os *IoT Boxset* como sensores da plataforma “IoTUnifiedManagement”, sensores esses responsáveis por monitorizar temperatura, luz e ruído de uma sala de aula física. Tal como foi descrito anteriormente, estas boxes foram inicialmente projetadas para serem usadas na aplicação ISABELA. No entanto, como os dados recolhidos são favoráveis a ambas as aplicações (ISABELA e 5GOpenclasses), estas boxes são consideradas atualmente como sensores da plataforma “IoTUnifiedManagement” e são apresentadas em ambas as arquiteturas das aplicações ISABELA e 5GOpenclasses. Estas são compostas por um *gateway Raspberry Pi*, uma *board Arduino* e sensores complementares (som, luz e temperatura). O UE é o dispositivo móvel Android responsável por executar a aplicação 5GOpenclasses dedicada aos alunos, possibilitando que estes usufruam de funcionalidades de transmissão de vídeo-aulas em tempo real (incluindo chat direto com o professor) e o download de materiais de suporte (incluindo vídeos e/ou ficheiros). Esta aplicação Android também é responsável por recolher e disponibilizar dados de sensores como: giroscópio, acelerómetro, proximidade, som e luz.

No centro da Figura 5 encontra-se o *middleware* FIWARE, responsável por armazenar e fornecer todos os dados IoT recolhidos, assim como alguns serviços externos *web-based*. Estes serviços externos são representados por um módulo de autenticação privada disponibilizado pela empresa IT-Aveiro no âmbito do projeto Mobilizador5G, com o objetivo do fornecimento de um processo de autenticação de preservação de identidade mais seguro. Para além disso, relativamente ao *middleware* FIWARE, consideramos que o servidor que executa os serviços do FIWARE é também o servidor responsável por fornecer as funcionalidades de transmissão de vídeo em tempo real via servidores multimédia Kurento WebRTC e seus componentes.

No que diz respeito aos componentes de fontes de fluxos de vídeo (canto superior direito da Figura 5) os alunos têm a possibilidade de visualizar fluxos de vídeo de ambientes internos (câmaras fixas nas salas de aula) ou até em ambientes externos (câmaras de vídeo UAV ou *bodykit*), tudo isto a partir da aplicação móvel Android 5GOpenclasses. Mais uma

vez, tanto os fluxos de vídeo das câmaras de vídeo *UAV* como das câmaras do *bodykit*, são fluxos provenientes de parceiros pertencentes ao projeto Mobilizador5G.

Por último, o *Teaching center* surge como uma pequena parte da aplicação “*ToTUnifiedManagement*”, parte essa que é dedicada à aplicação 5GOpenclasses e ao perfil de utilizador professor.

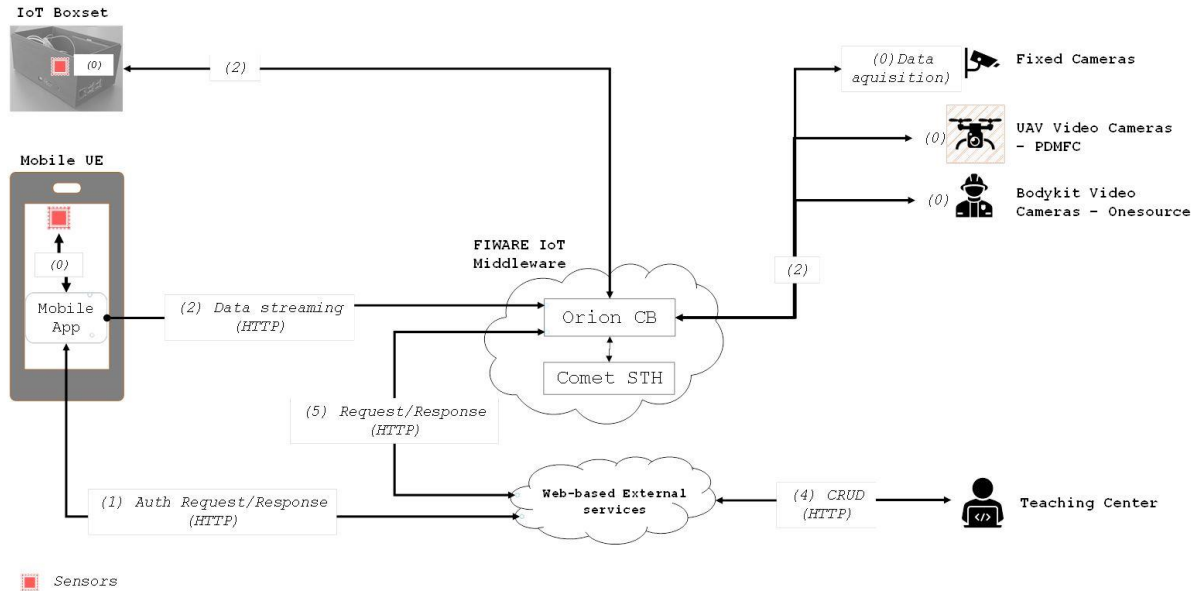


Figura 5 – Arquitetura 5GOpenclasses [16].

### 3.1.2 ISABELA

Em relação à aplicação ISABELA, esta conta com a arquitetura central de armazenamento de dados idêntica, visto que está a ser utilizado o *middleware* FIWARE. Tal como foi descrito anteriormente, este *middleware* FIWARE é responsável por armazenar/fornecer todos os dados IoT recolhidos, podendo armazenar dados tanto em MySQL, MongoDB ou CKAN. Para além disso, nesta arquitetura é visível a utilização de sistemas de monitorização diversos (como Grafana, Prometheus e Slack), a utilização de boxes ISABELA em vários locais (universidade e quarto de estudantes), a utilização de serviços de localização e *chatting*, a utilização de dados de redes sociais (Facebook e Twitter), a utilização de serviços desta plataforma (análise de sentimentos e sono) e a utilização das aplicações móveis para telemóveis e *smartwatch*.

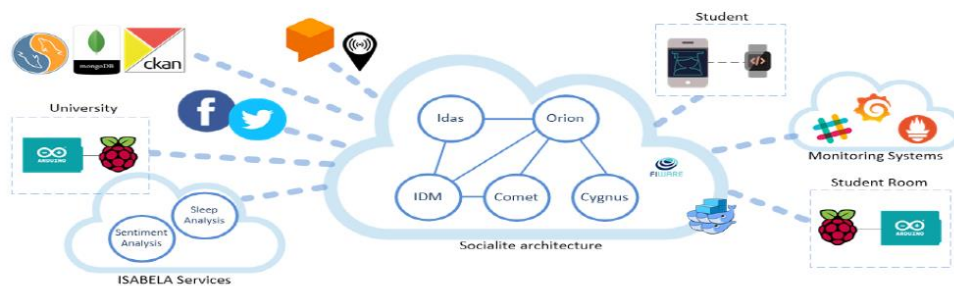


Figura 6 – Arquitetura atual ISABELA[5].

### 3.1.3 BATINA

A aplicação BATINA utiliza uma arquitetura idêntica usando o *middleware* FIWARE como responsável por armazenar/fornecer dados. Atualmente, a arquitetura da fase 1 do BATINA ainda é composta principalmente pela troca de dados entre o FIWARE e a aplicação *web-based*/móvel do BATINA. Neste caso, todas as comunicações com o FIWARE são efetuadas através do protocolo HTTPS, incluindo a utilização de *tokens* de acesso geridas por um módulo complementar ao FIWARE denominado por KEYROCK. Esta módulo é responsável por identificar se um *token* de acesso é válido, qual o utilizador associado a esse *token*, quais os seus grupos associados e quais os privilégios desse utilizador.

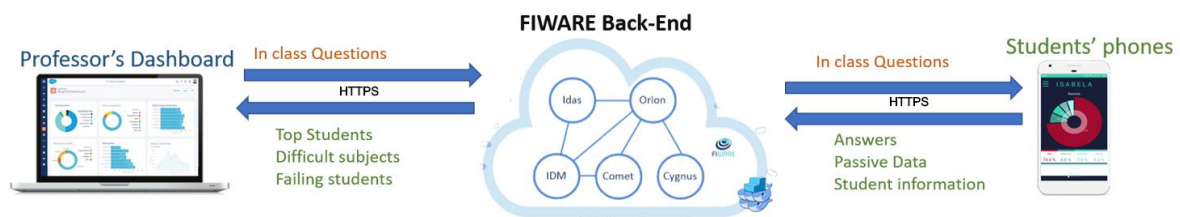


Figura 7 – Arquitetura atual BATINA [17].

## 3.2 Perfis de utilizadores

Relativamente aos perfis de utilizador da aplicação “IoTUnifiedManagement”, esta encontra-se dividida em dois diferentes perfis: **administrador** e **professor**. Como o seu objetivo é fornecer um *dashboard* para gestão unificada de dados IoT, esta gestão será totalmente administrada pelo perfil de utilizador **administrador**. Para além disso, o *dashboard* também será responsável por fornecer funcionalidades de gravação e transmissão de vídeo em tempo real, sendo esta componente dedicada exclusivamente à aplicação 5GOpenclasses e ao perfil de utilizador **professor**.

No que diz respeito ao perfil **administrador**, este é responsável por administrar todos os componentes base da aplicação, nomeadamente as componentes de gestão de periféricos e de sensores da plataforma, assim como a gestão de sensores das aplicações ISABELA e 5GOpenclasses. Mais detalhadamente, o administrador tem permissões para visualizar e gerir informações provenientes dos serviços e componentes do servidor que suportam a aplicação (FIWARE e sensores IoT), como visualizar e gerir dados de sensores IoT provenientes das aplicações ISABELA e 5GOpenclasses. Para além disso, o administrador também visualiza e gere os dados provenientes da aplicação 5GOpenclasses, dados esses relativos a todas as disciplinas da aplicação, aulas, materiais de suporte e *feedback* de QoE da aplicação.

Finalmente, o perfil **professor** dedica-se exclusivamente à aplicação 5GOpenclasses, sendo por isso responsável por visualizar e gerir as suas aulas (conforme as disciplinas atribuídas pelo administrador), visualizar e gerir materiais de suporte e usufruir das funcionalidades de vídeo-aulas em tempo real disponibilizadas.



### 3.3 Permissões dos utilizadores

De uma forma mais detalhada, os perfis utilizadores têm as seguintes permissões:

Tabela I – Permissões dos perfis de utilizador.

	Administrador	Professor	Detalhes
Visualizar infos do FIWARE	✓	✗	
Gerir dados IoT da aplicação *	✓	✗	* Sensores das boxes ISABELA
Gerir dados IoT *	✓	✗	* Sensores IoT ISABELA e 5GOpenclasses
Visualizar dados ISABELA *	✓	✗	* Gráficos de médias
Enviar notificações	✓	✗	
Gerir bloqueio de sensores IoT *	✓*	✗	* Gerir alunos bloqueados na recolha de dados
Visualizar dados 5GOpenclasses	✓	✓ <sup>1</sup>	<sup>1</sup> não visualiza dados de <i>feedback</i>
Gerir dados 5GOpenclasses	✓ <sup>1</sup>	✓ <sup>2</sup>	<sup>1</sup> apenas pode adicionar novas disciplinas <sup>2</sup> apenas não pode gerir suas disciplinas
Usufruir de funcionalidades de vídeo-aulas (via Kurento)	✗	✓	

### 3.4 Requisitos

O perfil de utilizador **administrador** encontra-se dividido em 3 principais secções denominadas por “Gestão”, “ISABELA” e “5GOpenclasses”. Na secção “Gestão”, o administrador terá a possibilidade de:

- Visualizar informações de serviços do *middleware* FIWARE;
- Visualizar informações dos componentes do servidor;
- Gerir a informação dos sensores que complementam as salas de aula (sensores da aplicação ISABELA, mas pertencentes à plataforma);
- Gerir os sensores IoT onde são recolhidos dados (ISABELA e 5GOpenclasses);
- Enviar notificações na aplicação móvel ISABELA.

Na secção “ISABELA”, o administrador terá a possibilidade de:

- Visualizar e tratar os dados recolhidos na aplicação móvel Android ISABELA (gráficos com médias de dados).

Por fim, na secção “5GOpenclasses” o administrador terá a possibilidade de:

- Visualizar e gerir parte dos dados relativos à aplicação 5GOpenclasses.

Por outro lado, o perfil de **professor** é dedicado exclusivamente à aplicação 5GOpenclasses terá apenas uma secção denominada “5GOpenclasses” onde tem a possibilidade de visualizar e gerir dados relativos a esta aplicação.

Deste modo, a aplicação encontra-se dividida da seguinte forma:

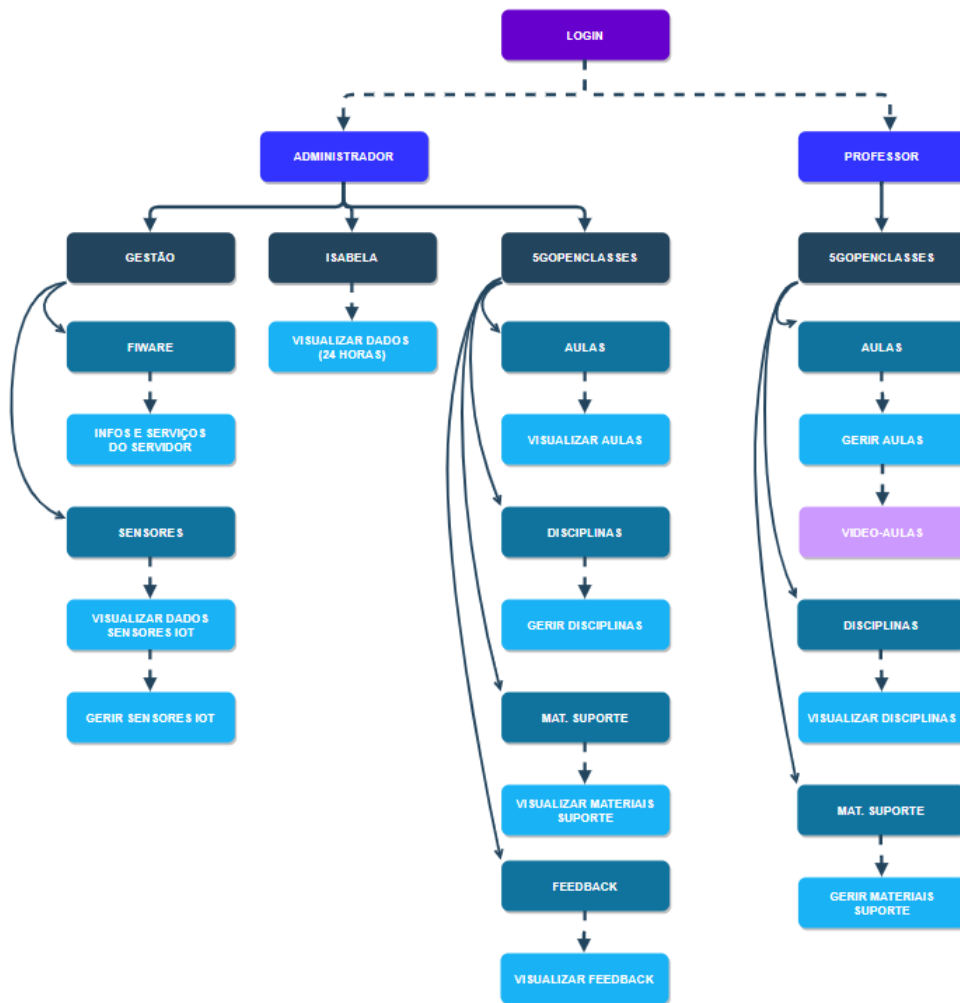


Figura 8 – Divisão de perfis na aplicação “IoTUnifiedManagement”.

### 3.4.1 Requisitos funcionais

A Tabela II apresenta os requisitos funcionais da aplicação, apresentados e descritos no seguinte contexto:

Tabela II – Contexto de apresentação dos requisitos.

Parâmetros	Descrição dos parâmetros
Descrição	Descrição do requisito.
Secção	Descrição da secção pertencente do requisito (Gestão, ISABELA ou 5GOpenclasses).
Ator(es)	Tipo de utilizador que poderá atuar no requisito (administrador e professor).
Prioridade	<p><u>Obrigatório</u>: requisito essencial para o projeto.</p> <p><u>Possível</u>: requisito que deve ser implementado, mas a sua ausência não impede o normal funcionamento da aplicação.</p> <p><u>Prescindível</u>: requisito não essencial e com baixa prioridade.</p>
Resultado esperado	Após acionado o requisito, o que é esperado que aconteça.

- O administrador já deve estar registado por defeito na aplicação.
  - **Descrição**: as credenciais do administrador já se encontram registadas na aplicação;
  - **Secção**: Não aplicável;
  - **Ator**: Administrador;
  - **Prioridade**: Obrigatório;
  - **Resultado esperado**: ao iniciar sessão, o utilizador é redirecionado para a página inicial da aplicação.
- Os professores podem-se registar na aplicação.
  - **Descrição**: os professores podem-se registar na aplicação, usando a página dedicada ao registo;
  - **Secção**: Não aplicável;
  - **Ator**: Professor;
  - **Prioridade**: Obrigatório;
  - **Resultado esperado**: ao adicionar novos professores à aplicação, estes passam a poder iniciar sessão na aplicação.

3. Login dos utilizadores com e-mail e palavra-passe.
  - **Descrição:** introdução do email e password;
  - **Secção:** Não aplicável;
  - **Atores:** Administrador e professor;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** com a introdução correta das credenciais de autenticação, o utilizador será redirecionado para a página inicial da aplicação.
  
4. Visualização de uma linha de localização espacial na utilização da aplicação.
  - **Descrição:** apresentação de um mecanismo que será responsável por apresentar uma localização temporal na utilização da nossa aplicação (ex: ISABELA>Activity Info>Special Activity Info);
  - **Secção:** Não aplicável;
  - **Atores:** Administrador e professor;
  - **Prioridade:** Prescindível;
  - **Resultado esperado:** se as credenciais de autenticação estiverem corretas, o utilizador será redirecionado para a página inicial da aplicação.
  
5. Visualizar informações dos estados dos serviços.
  - **Descrição:** apresentação de componentes informativos que relatam o estado (ativo ou não ativo) dos serviços do FIWARE;
  - **Secção:** Gestão – FIWARE;
  - **Ator:** Administrador;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correta apresentação do estado dos serviços do FIWARE.
  
6. Visualizar informações de utilização dos componentes do servidor.
  - **Descrição:** apresentação de componentes informativos que relatam os componentes do servidor onde se encontra a executar o FIWARE (utilização de CPU, utilização de memória RAM, utilização de memória SWAP e carga do CPU numa média de 5 minutos);
  - **Secção:** Gestão – FIWARE;
  - **Ator:** Administrador;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correta apresentação da informação de utilização dos componentes da máquina.
  
7. Visualizar informações acerca das especificações do servidor.
  - **Descrição:** apresentação de componentes informativos que relatam as informações de especificações do servidor onde se encontra a executar o FIWARE (memória RAM total, total de memória SWAP, carga do sistema numa média de 1 minuto, o tempo de atividade e o espaço de armazenamento livre);

- **Secção:** Gestão – FIWARE;
  - **Ator:** Administrador;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correta apresentação das especificações da máquina.
8. Visualizar informações do servidor num espaço de tempo de 24 horas.
- **Descrição:** apresentação de componentes informações que relatam a utilização do sistema por parte do FIWARE num espaço temporal de 24 horas (tempo de resposta registado, utilização de CPU e utilização de memória RAM);
  - **Secção:** Gestão – FIWARE;
  - **Ator:** Administrador;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correta apresentação das informações de utilização do sistema num espaço de tempo de 24 horas.
9. Visualização de informações de sensores das salas de aula (boxes).
- **Descrição:** apresentação de dados que relatam informações relativas aos sensores que se encontram dentro das salas de aulas (data da última recolha de dados, nome da “Box”, localização, tipo de sensor, valor da última recolha, frequência de recolha e estado);
  - **Secção:** Gestão Sensores;
  - **Ator:** Administrador;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correta apresentação das informações dos sensores das salas de aula.
10. Gestão de informações de sensores de salas de aula.
- **Descrição:** edição das informações dos sensores das salas de aula (frequência da recolha de dados);
  - **Secção:** Gestão Sensores;
  - **Ator:** Administrador;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correta gestão da informação dos sensores de dados de salas de aula.
11. Visualização do número total de sensores das aplicações ISABELA e 5GOpenclasses.
- **Descrição:** apresentação de componentes de informação que relatam o número total de sensores onde estão a ser recolhidos dados dos alunos nas aplicações móveis ISABELA e 5GOpenclasses;
  - **Secção:** Gestão Sensores;
  - **Ator:** Administrador;
  - **Prioridade:** Obrigatório;

- **Resultado esperado:** correta apresentação das informações relativas ao número total de sensores nas aplicações móveis ISABELA e 5GOpenclasses.
12. Gestão dos sensores das aplicações ISABELA e 5GOpenclasses.
- **Descrição:** apresentação e edição das informações relativas aos sensores onde estão a ser recolhidos dados nas aplicações móveis ISABELA e 5GOpenclasses (data da última recolha de dados, tipo, ID estudante, valor da última recolha, frequência de recolha de dados e estado, sendo apenas possível editar a frequência de recolha de dados);
  - **Secção:** Gestão Sensores;
  - **Ator:** Administrador;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correta apresentação e gestão das informações relativas aos sensores das aplicações móveis ISABELA e 5GOpenclasses.
13. Visualização de informações de alunos bloqueados das aplicações móveis ISABELA e 5GOpenclasses.
- **Descrição:** apresentação de componentes informações que relatam os alunos que se encontram bloqueados na recolha de dados das aplicações móveis ISABELA e 5GOpenclasses;
  - **Secção:** Gestão Sensores;
  - **Ator:** Administrador;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correta apresentação das informações relativas aos alunos bloqueados nas aplicações móveis ISABELA e 5GOpenclasses.
14. Gestão de informações de alunos bloqueados das aplicações móveis ISABELA e 5GOpenclasses.
- **Descrição:** Bloqueio/Desbloqueio da recolha de dados dos alunos nas aplicações móveis ISABELA e 5GOpenclasses;
  - **Secção:** Gestão Sensores;
  - **Ator:** Administrador;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correta gestão das informações dos alunos bloqueados nas aplicações móveis ISABELA e 5GOpenclasses.
15. Envio de notificações.
- **Descrição:** mecanismo de envio de notificações para os alunos registados na aplicação móvel ISABELA;
  - **Secção:** Gestão Sensores;
  - **Ator:** Administrador;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correto envio de notificações dos alunos registados na aplicação móvel ISABELA.

16. Visualização informações gerais relativas aos componentes: média de atividade dos alunos, média de tempo de descanso dos alunos, localização média dos alunos, total de questionários respondidos pelos alunos, média de sociabilidade dos alunos, aplicações mais utilizadas pelos alunos e top 3 do ranking da aplicação.
  - **Descrição:** apresentação de componentes de informação que relatam as médias dos dados que são coletados pela aplicação móvel ISABELA;
  - **Secção:** ISABELA;
  - **Ator:** Administrador;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correta apresentação das informações relativas aos dados recolhidos na aplicação móvel ISABELA das últimas 24 horas.
  
17. Todos os gráficos do requisito anterior, devem ter um botão para ver visualizar mais informações.
  - **Descrição:** disponibilização de um botão para cada gráfico que possibilite o acesso à informação mais detalhada;
  - **Secção:** ISABELA;
  - **Ator:** Administrador;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correta apresentação dos botões nos diferentes gráficos.
  
18. Visualização de informações mais detalhadas acerca de cada componente descrito no requisito 16.
  - **Descrição:** apresentação de componentes de informação de relatam informações mais detalhadas acerca de cada componente descrito no requisito 16. Este requisito vai ser composto por uma Tabela onde serão apresentados os diferentes alunos presentes na aplicação móvel ISABELA e, para cada aluno, deverá ser possível consultar informações de forma mais específica (apresentando dados através de uma linha temporal de resultados);
  - **Secção:** ISABELA;
  - **Ator:** Administrador;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correta apresentação das informações mais detalhadas para cada componente descrito no requisito 15.
  
19. Mecanismo de *drag-and-drop* dos componentes de informação descritos no requisito 16.
  - **Descrição:** mecanismo que possibilita o *drag-and-drop* dos componentes de informação que são apresentados pelo requisito 16;
  - **Secção:** ISABELA;
  - **Ator:** Administrador;
  - **Prioridade:** Prescindível;
  - **Resultado esperado:** utilização sem problemas deste mecanismo de *drag-and-drop* dos componentes de informação.

20. Visualização de informações relativas às disciplinas existentes na plataforma 5GOpenclasses.
- **Descrição:** apresentação de componentes de informação que apresentam as disciplinas que se encontram presentes na plataforma 5GOpenclasses;
  - **Secção:** 5GOpenclasses;
  - **Atores:** Administrador e professor;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correta apresentação das disciplinas que se encontram na plataforma 5GOpenclasses.
21. Gestão de informações relativas às disciplinas existentes na plataforma 5GOpenclasses.
- **Descrição:** gestão de disciplinas que se encontram presentes na plataforma 5GOpenclasses;
  - **Secção:** 5GOpenclasses;
  - **Ator:** Administrador;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correta gestão das disciplinas que se encontram na plataforma 5GOpenclasses.
22. Visualização de informações relativas aos materiais de suporte existentes na plataforma 5GOpenclasses.
- **Descrição:** apresentação de componentes de informação que apresentam os materiais de suporte que se encontram presentes na plataforma 5GOpenclasses
  - **Secção:** 5GOpenclasses;
  - **Atores:** Administrador e professor;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correta apresentação dos materiais de suporte que se encontram na plataforma 5GOpenclasses.
23. Gestão de informações relativas aos materiais de suporte existentes na plataforma 5GOpenclasses.
- **Descrição:** gestão dos arquivos que se encontram presentes na plataforma 5GOpenclasses;
  - **Secção:** 5GOpenclasses;
  - **Ator:** Professor;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correta gestão dos materiais de suporte que se encontram na plataforma 5GOpenclasses.
24. Visualização de informações relativas às vídeo-aulas existentes na plataforma 5GOpenclasses.
- **Descrição:** apresentação de componentes de informação que apresentam os vídeo-aulas que se encontram presentes na plataforma 5GOpenclasses;



- **Secção:** 5GOpenclasses;
  - **Atores:** Administrador e professor;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correta apresentação das vídeo-aulas que se encontram na plataforma 5GOpenclasses.
25. Gestão de informações relativas às vídeo-aulas existentes na plataforma 5GOpenclasses.
- **Descrição:** gestão das vídeo-aulas que se encontram presentes na plataforma 5GOpenclasses;
  - **Secção:** 5GOpenclasses;
  - **Ator:** Professor;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correta apresentação dos materiais de suporte que se encontram na plataforma 5GOpenclasses.
26. Mecanismo para iniciar uma vídeo-aula.
- **Descrição:** mecanismo que possibilita a iniciação de uma vídeo-aula com a possibilidade de captura do próprio ecrã;
  - **Secção:** 5GOpenclasses;
  - **Ator:** Professor;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correta iniciação da vídeo-aula.
27. Mecanismo para encerrar uma vídeo-aula.
- **Descrição:** mecanismo que possibilita o encerramento de uma vídeo-aula;
  - **Secção:** 5GOpenclasses;
  - **Ator:** Professor;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correto encerramento da vídeo-aula.
28. Mecanismo para adicionar novos materiais de suporte a uma disciplina.
- **Descrição:** mecanismo que possibilita o *upload* de novos arquivos que completam uma disciplina;
  - **Secção:** 5GOpenclasses;
  - **Ator:** Professor;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correto *upload* dos arquivos necessários.
29. Visualização de questões elaboradas pelos alunos durante uma vídeo-aula.
- **Descrição:** apresentação das questões elaboradas pelos alunos durante o decorrer de uma vídeo-aula;
  - **Secção:** 5GOpenclasses;
  - **Ator:** Professor;

- **Prioridade:** Obrigatório;
  - **Resultado esperado:** correta apresentação das questões dos alunos na vídeo-aula.
30. Visualização de informações relativas às classificações da qualidade de experiência dos alunos.
- **Descrição:** apresentação das informações de classificações de qualidade de experiência dos alunos na utilização da aplicação móvel 5GOpenclasses;
  - **Secção:** 5GOpenclasses;
  - **Ator:** Administrador;
  - **Prioridade:** Obrigatório;
  - **Resultado esperado:** correta apresentação das informações de qualidade de experiência dos alunos na aplicação 5GOpenclasses.

### 3.4.2 Requisitos não funcionais

1. Compatibilidade - a plataforma deve ser compatível com os *browsers* atualmente mais utilizados;
2. Escalabilidade - a plataforma deve permitir o escalonamento dos recursos necessários para servir um crescente número de utilizadores;
3. Usabilidade - a aplicação deve adaptar-se aos diferentes tipos de ecrãs de dispositivos móveis;
4. Privacidade - todos os dados devem ser privados e seguros;
5. Segurança - deve ser assegurado o nível de segurança alto, tanto a nível da plataforma como da comunicação.

Relativamente à compatibilidade, a plataforma pode ser acedida pelos *browsers* Google Chrome [18], Safari [19], Mozilla Firefox [20] ou Microsoft Edge [21].

Os recursos da plataforma devem ser escaláveis conforme o número de utilizadores. Estes podem ser alterados nas definições dos *containers* Docker.

Em relação à usabilidade, a plataforma adapta-se conforme os acessos dos diferentes dispositivos móveis.

A privacidade restringe os utilizadores a visualizar apenas os seus próprios dados.

A segurança da plataforma oferece encriptação de ponto a ponto via SSL e uma troca de dados com o FIWARE através de HTTPS.

## 3.5 Especificação dos componentes da arquitetura

Na arquitetura apresentada no início deste capítulo, é possível identificar quais os componentes principais da nossa aplicação “IoTUnifiedManagement”: FIWARE, Prometheus e Kurento.

No que diz respeito ao *middleware* FIWARE, este é responsável por armazenar/fornecer todos os dados recolhidos nas bases de dados (MongoDB, MySQL e CKAN) e composto pelos seguintes módulos: IDAS, Orion, idM, Comet e Cygnus.

O módulo IDAS é a parte do sistema que lida com a comunicação com os todos os tipos de dispositivos IoT. Este módulo suporta recursos de gestão e interoperabilidade entre os dispositivos (comunicam usando diversos protocolos) e o FIWARE. Nesta caso, o módulo

proporciona o suporte para todos os sensores IoT da aplicação “IoTUnifiedManagement” [5].

O módulo Orion é a parte do sistema que permite gerir todo o ciclo de vida das informações de contexto como atualizações, consultas, registos e subscrições. Este modelo é baseado em entidades e atributos que representam entidades do mundo real, sendo que cada um deles podem ser representados através do formato JSON. Para aumentar a distribuição e escalabilidade, este módulo atua como sendo um agente/intermediário que permite aos sistemas externos (consumidores) de fazer subscrições de dados com regras específicas para as entidades e atributos. O Orion envia notificações para esses sistemas quando as regras de subscrição são disparadas. Embora este módulo permita criar novas entidades e guardar os dados dos seus atributos, o Orion apenas armazena os dados relativos à última recolha. No nosso caso em particular, este módulo proporciona suporte à gestão de entidades e de dados IoT recolhidos pelos diferentes sensores (giroscópio, acelerômetro, proximidade, som, luz e temperatura) [5], [22].

O módulo Cygnus é responsável por coordenar o armazenamento dos dados nas diferentes bases de dados. Ao contrário do Orion, o Cygnus é responsável por armazenar os dados de subscrições de entidades em sistemas de armazenamento de dados (como MySQL, MongoDB e CKAN), criando desta maneira uma visão histórico-temporal desses dados. No nosso caso, este módulo é responsável por comunicar com bases de dados do tipo MySQL e mongoDB [5].

O módulo Comet é responsável por gerir (armazenar/fornecer) informações históricas da evolução de um determinado contexto registado numa instância do Orion. Este módulo implementa APIs REST para comunicar com o ORION e com terceiros (ex: aplicação “IoTUnifiedManagement”) [5], [23].

O módulo idM é responsável por gerir entidades dentro da aplicação FIWARE. O uso do idM permite adicionar segurança na autenticação e autorização OAuth2.0. Este módulo tem a possibilidade de restringir o acesso de determinadas informações e funcionalidades a determinadas contas do FIWARE e todos os seus módulos. No âmbito do nosso projeto, este módulo não está a ser utilizado pela aplicação “IoTUnifiedManagement” [5], [24].

O FIND (considerado no perfil **administrador** para visualização dos estados dos serviços do FIWARE) é uma aplicação fornecedora da localização interna de um dispositivo num ambiente doméstico ou empresarial. Este utiliza a intensidade do sinal *Bluetooth* ou *wifi* para identificar a posição de um *smartphone* ou computador num ambiente pré-configurado.

Relativamente às atuações nos sensores, o *middleware* FIWARE possui dois agentes que trabalham como “ponte” para comunicar com dispositivos IoT, sendo eles o Lightweight M2M (trabalha sobre CoaP) [25] e o Ultralight 2.0 (trabalha sobre HTTP/MQTT) [26]. Estes dois agentes são destinados para trabalhar com comunicações e dispositivos restritos, onde a largura de banda e memória do dispositivo podem ser recursos limitados [25][26]. Ao usarem um agente IoT, os dispositivos finais podem tornar-se uma entidade de dados no FIWARE, realizar consultas e obter atualizações, seguir as suas alterações e atualizar esses valores nas suas entidades (como por exemplo, atualizar da frequência de recolha de dados). Tudo isto pode ser feito através da utilização do agente IoT IDAS, visto que este foi projetado pelo FIWARE para esta finalidade.

No que diz respeito ao Prometheus, este é um sistema dedicado à monitorização e alerta e possui funcionalidades de armazenamento de dados numa linha temporal. No nosso

caso, este sistema está a ser utilizado pela aplicação “IoTUnifiedManagement” como um sistema de armazenamento de dados de informações tais como:

- Informações dos estados dos serviços “Orion”, “Idas”, “Comet” e “Find” do servidor;
- Informações de percentagem de utilização dos componentes do servidor, tais como: CPU, memória RAM, memória SWAP e a carga do CPU (numa média de 5 minutos);
- Informações dos componentes do servidor, tais como: memória RAM total, memória SWAP total, carga do sistema (numa média de 1 minuto), o tempo de atividade do servidor e o espaço de armazenamento livre;
- Informações numa resolução temporal de 24 horas: tempo de resposta, percentagem de utilização do CPU e percentagem de utilização da memória RAM.

Todos os dados apresentados acima têm informações armazenadas numa resolução temporal, mas apenas no último tópico estão a ser utilizadas estas informações numa resolução temporal de 24 horas.

Por último, o Kurento é o servidor de multimédia WebRTC que irá fornecer funcionalidades aos utilizadores (professores e alunos) de difusão de fluxos de vídeo em tempo real entre diferentes aplicações. Neste projeto em específico, foi utilizada uma plataforma web para suportar os professores e uma aplicação móvel *android* para suportar os alunos. Este servidor multimédia Kurento é suportado por outros dois servidores (STUN/TURN e Media-Server-Helper), responsáveis por estabelecer ligações seguras entre dispositivos e auxiliar clientes web a utilizar o Kurento. A necessidade de utilização deste servidor STUN/TURN advém da segurança proporcionada pelas *firewalls* e NATs.

## 3.6 Protótipos (*Mockups*)

Tal como foi descrito anteriormente, a aplicação “IoTUnifiedManagement” encontra-se dividida em 2 diferentes perfis de utilizador: **administrador** e **professor**. O perfil de administrador será composto por 3 principais secções denominadas “Gestão”, “ISABELA” e “5GOpenclasses” e o perfil de professor será composto apenas por uma secção dedicada ao “5GOpenclasses”. Neste subcapítulo vai ser apresentadas as propostas visuais detalhadas de toda a aplicação “IoTUnifiedManagement”, incluindo os diferentes perfis de utilizador que foram desenhados para a nossa aplicação.

### 3.6.1 Administrador

Na secção “Gestão” será possível visualizar informações de serviços do *middleware* FIWARE, visualizar informações dos componentes do servidor e gerir a informação dos sensores que complementam as salas de aula (sendo estes sensores pertencentes às boxes da plataforma). Para além disso, é possível gerir os sensores IoT onde são recolhidos dados (ISABELA e 5GOpenclasses) e enviar notificações na aplicação móvel ISABELA. Na secção “ISABELA”, o administrador terá a possibilidade de visualizar e tratar os dados IoT recolhidos na aplicação móvel Android ISABELA. No que diz respeito à secção “5GOpenclasses”, o administrador terá a possibilidade de visualizar dados relativos à aplicação 5GOpenclasses e gerir uma pequena parte desses dados.

Os protótipos para este perfil de utilizador são os seguintes:

**Gestão:**



Figura 9 – Proposta visual da aplicação (1).

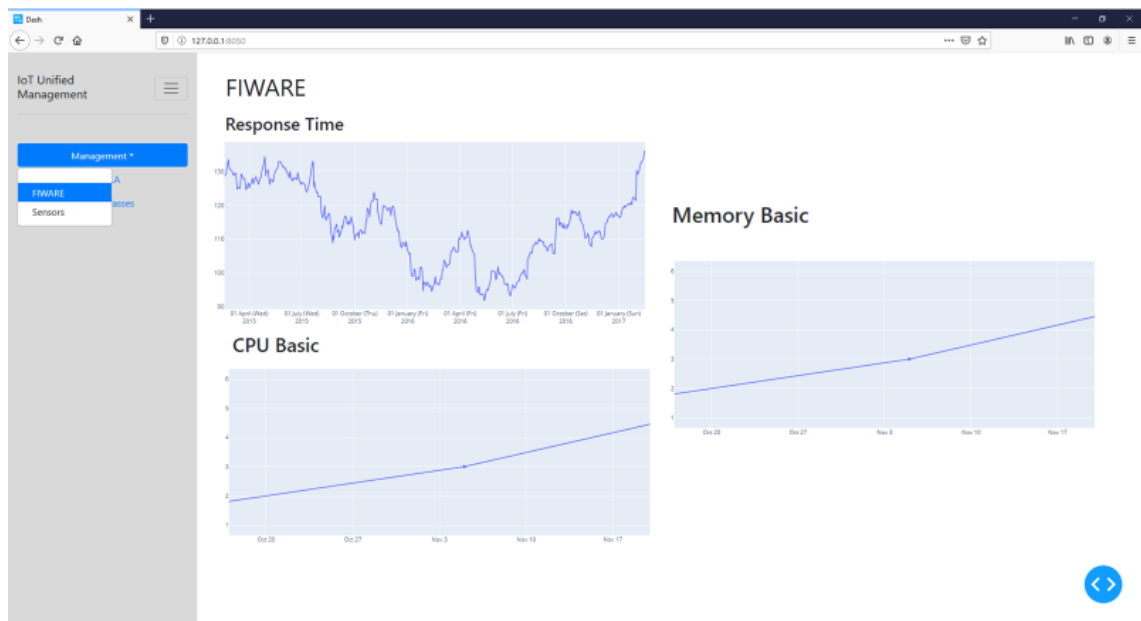


Figura 10 – Proposta visual da aplicação (2).

As Figuras 9 e 10 representam as propostas visuais relativas à primeira e segunda parte da secção de gestão. Relativamente à gestão do FIWARE, esta proposta inclui a representação dos estados dos serviços “Orion”, “Idas”, “Comet” e “Find” e inclui informações do servidor onde se encontram a executar os serviços da plataforma FIWARE

(ex: percentagem de utilização do CPU, percentagem de utilização da memória RAM, percentagem de utilização da memória SWAP e a carga do CPU numa média de 5 minutos). Adicionalmente, é possível representar informações relativas à memória RAM total, memória SWAP total, carga do sistema numa média de 1 minuto, o tempo de atividade e o espaço de armazenamento livre.

Na segunda parte de gestão do FIWARE, esta proposta inclui a representação do tempo de resposta registado aquando de um pedido (*request*) ao FIWARE, utilização de CPU, utilização de memória RAM e tempo de resposta dos serviços “Orion”, “Idas”, “Comet” e “Find”. Estes gráficos irão ter como objetivo a análise ao longo da linha temporal das últimas 24 horas.

The screenshot displays the 'Sensors' section of the IoT Unified Management interface. It features a table of Platform Sensors and two notification forms.

TimeStamp	Name	Location	Sensor Type	Sensor Value	Acquisition Frequency	State	Edit
2019-01-01T14:00:00.00Z	Box #1	G.6.1	Sound Sensor	10.1573561431822208768	3600 s	Green	>>
2019-01-01T14:00:00.00Z	Box #1	G.6.1	Temperature Sensor	1073561431822208768	1800 s	Green	>>
2019-01-01T14:00:00.00Z	Box #2	G.6.2	Sound Sensor	10.1573561431822208768	7200 s	Red	>>

Below the table, the 'ISABELA/5GOpenclasses' section shows two summary cards: 'Total Sensors Number' (137) and 'Total Blocked Members' (12), each with an '+ Info' button.

Two 'Notifications' forms are visible, each with a text input field, a radio button for 'All students' or 'Specific one', and a 'Send' button. The second form has a dropdown menu set to 'ID 123'.

Figura 11 – Proposta visual da aplicação (3).

A Figura 11 representa a proposta visual relativa à parte de gestão dos sensores, quer sejam eles os sensores da plataforma (boxes) ou mesmo até dos sensores IoT das aplicações móveis Android ISABELA e 5GOpenclasses. Na secção relativa aos sensores da plataforma, é possível listar todos os sensores pertencentes às boxes. Já na secção dos sensores IoT das aplicações móveis Android ISABELA e 5GOpenclasses, são apenas apresentados dados relativos ao número total de sensores IoT onde estão a ser recolhidos e também o número total de membros bloqueados na recolha dos dados IoT das aplicações. Para além disso, é possível enviar notificações diretamente aos alunos de ambas as aplicações, tanto de uma forma genérica como para um aluno específico.

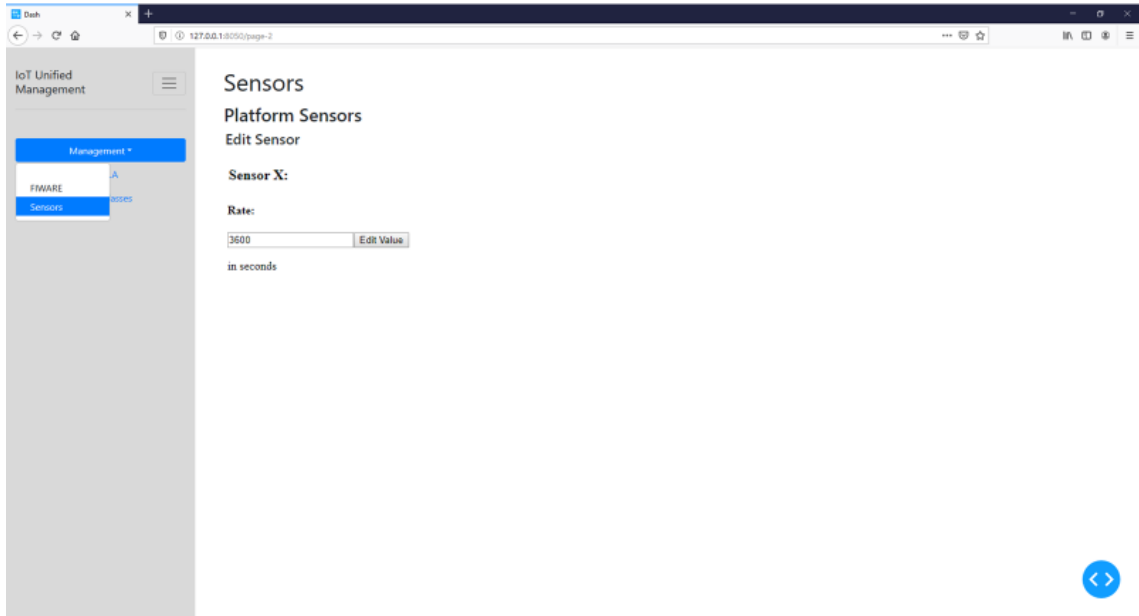


Figura 12 – Proposta visual da aplicação (4).

A proposta visual relativa à parte de edição da frequência de recolha de dados IoT num sensor específico dos sensores da plataforma (boxes) encontra-se representada na Figura 12.

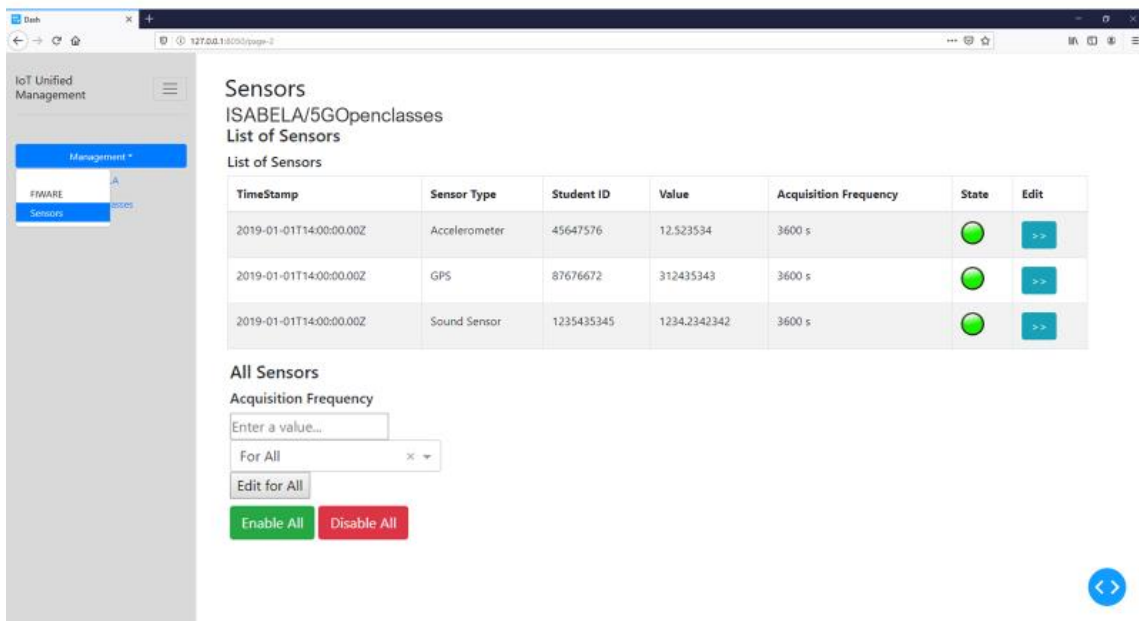


Figura 13 – Proposta visual da aplicação (5).

A Figura 13 representa a proposta visual da secção de apresentação de informações relativas à lista de sensores IoT onde são recolhidos dados nas aplicações móveis Android ISABELA e 5GOpenclasses, assim como uma secção dedicada à edição de frequência de recolha de dados IoT. Esta secção de edição de frequência de recolha de dados possibilita a edição para todo o tipo de sensores ou apenas para um tipo de sensor específico.

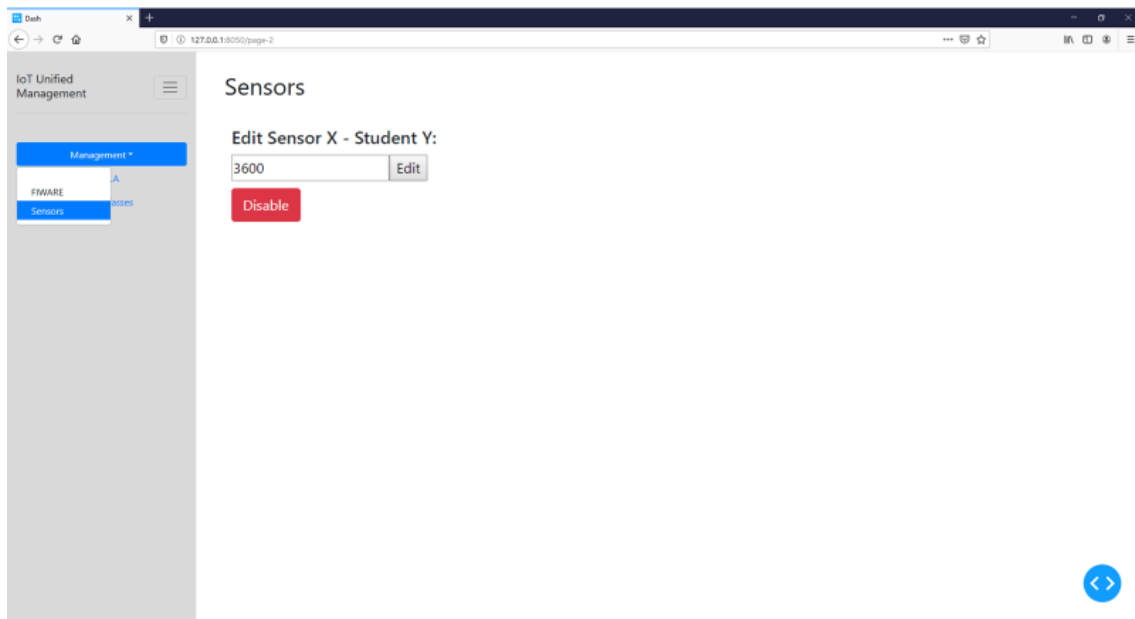


Figura 14 – Proposta visual da aplicação (6).

Na Figura 14 está representada a proposta visual relativa à edição de um determinado tipo de sensor para um estudante específico. Esta oferece a possibilidade de edição de frequência de recolha de dados de um sensor ou a desativação da recolha de dados do mesmo.

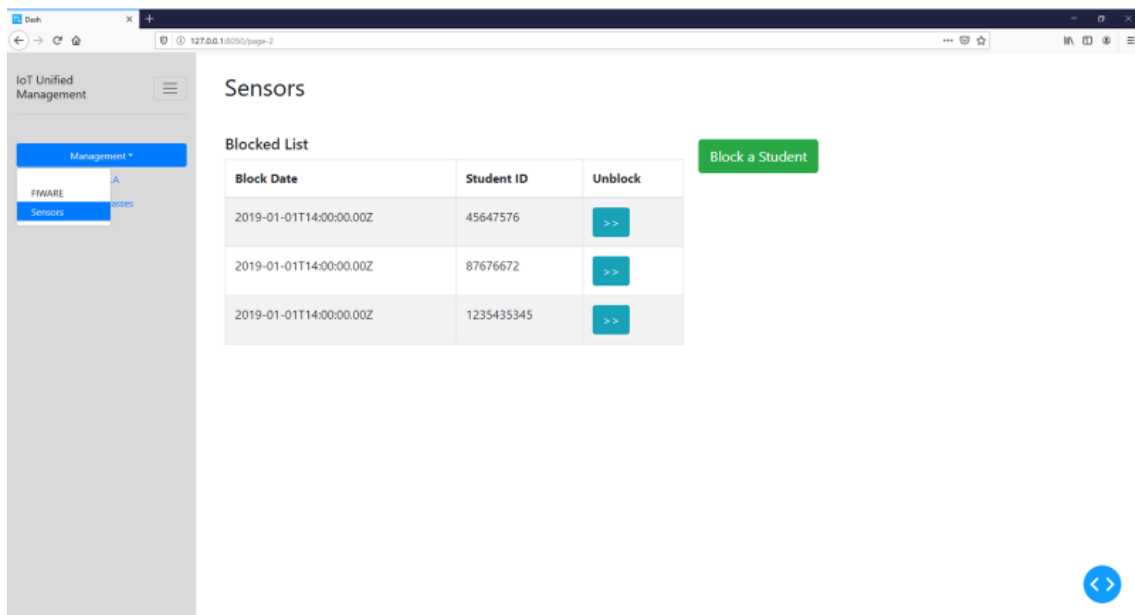


Figura 15 – Proposta visual da aplicação (7).



### Capítulo 3

A proposta visual referente à visualização da lista de utilizadores que se encontra bloqueada na recolha de dados IoT via aplicações móveis Android ISABELA e 5GOpenclasses encontra-se representada na Figura 15.

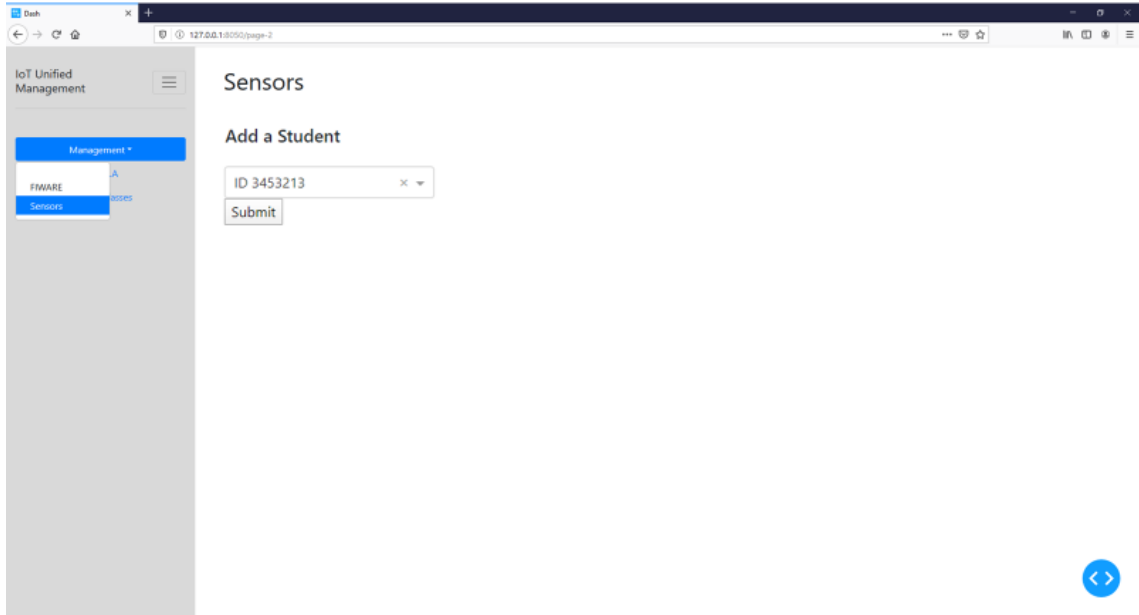


Figura 16 – Proposta visual da aplicação (8).

Por último, a Figura 16 representa a proposta visual da adição de um novo membro à lista de utilizadores bloqueados na recolha de dados nos sensores IoT via aplicações móveis Android ISABELA e 5GOpenclasses.

### ISABELA:

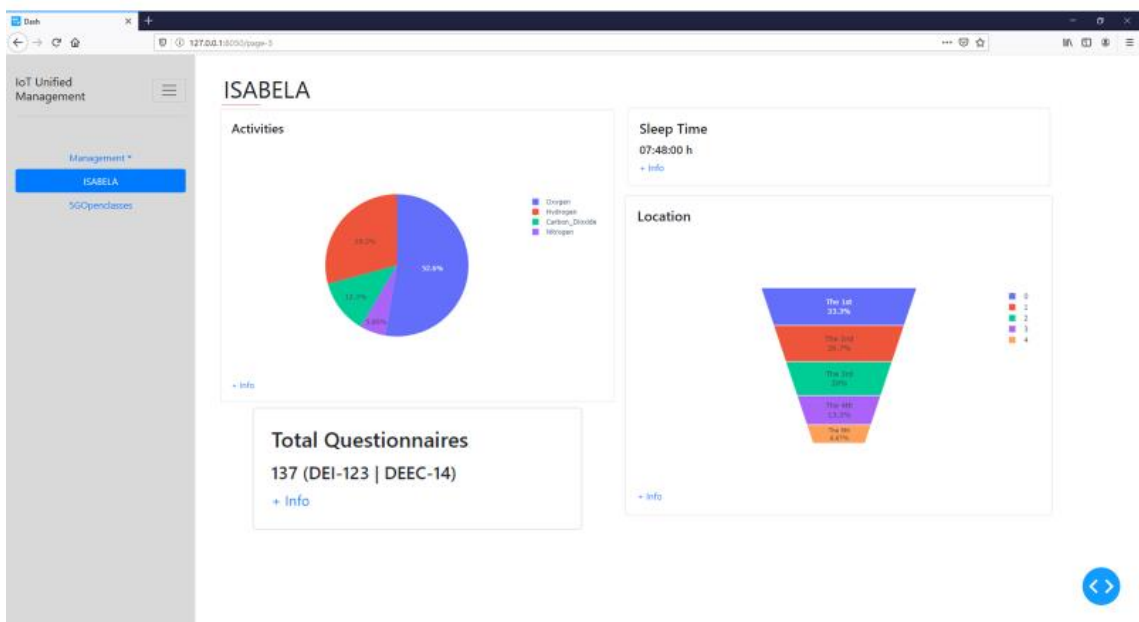


Figura 17 – Proposta visual da aplicação (9).

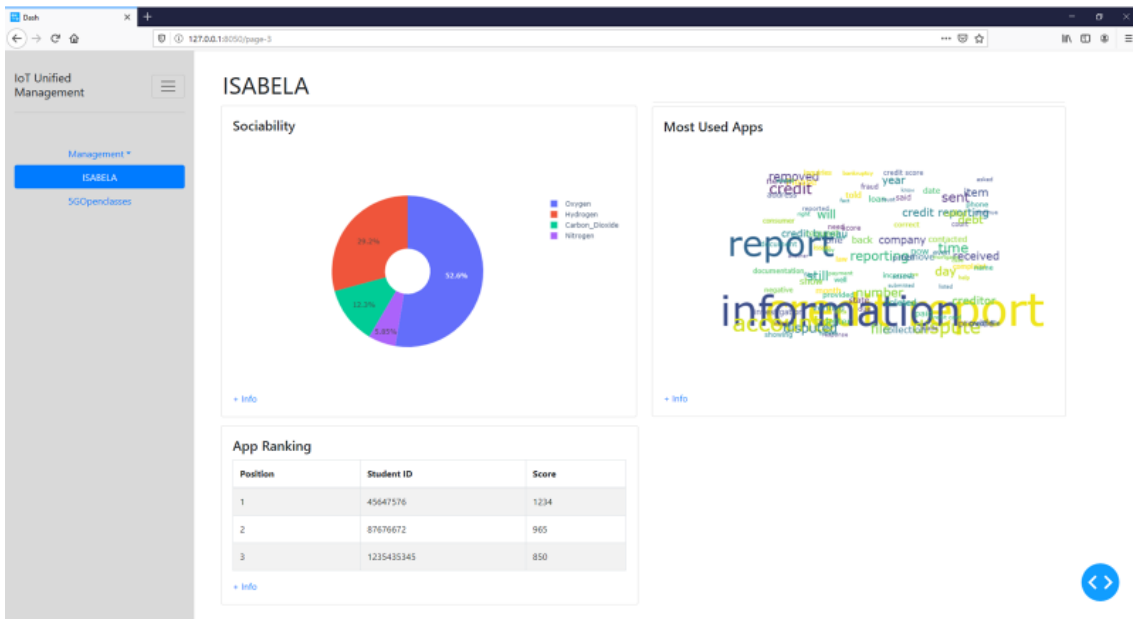


Figura 18 – Proposta visual da aplicação (10).

As Figuras 17 e 18 representam a proposta visual relativa à secção inicial do ISABELA onde são visualizadas informações relativas a: média de atividades dos alunos, média de tempo de descanso dos alunos, localização média dos alunos, total de questionários respondidos pelos alunos, média de sociabilidade dos alunos, aplicações mais utilizadas pelos alunos e top 3 do ranking da aplicação. Todas as informações apresentadas são correspondentes a dados recolhidos nas últimas 24 horas. Cada componente é completo por um botão “+ Info”, onde nos possibilita observar informações mais detalhadas.

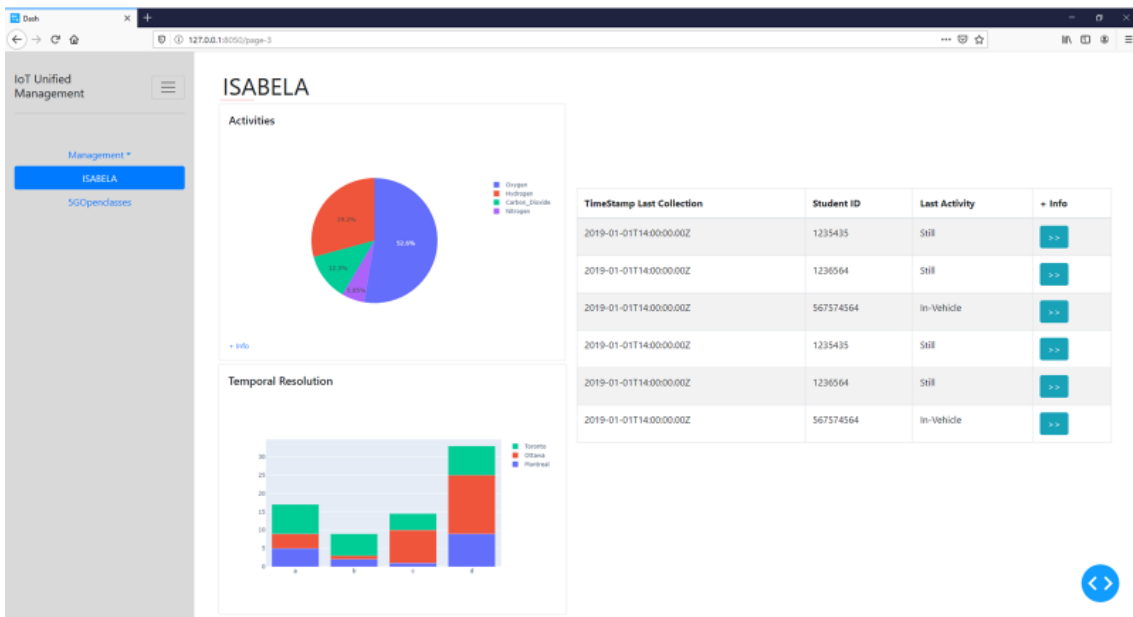


Figura 19 – Proposta visual da aplicação (11).

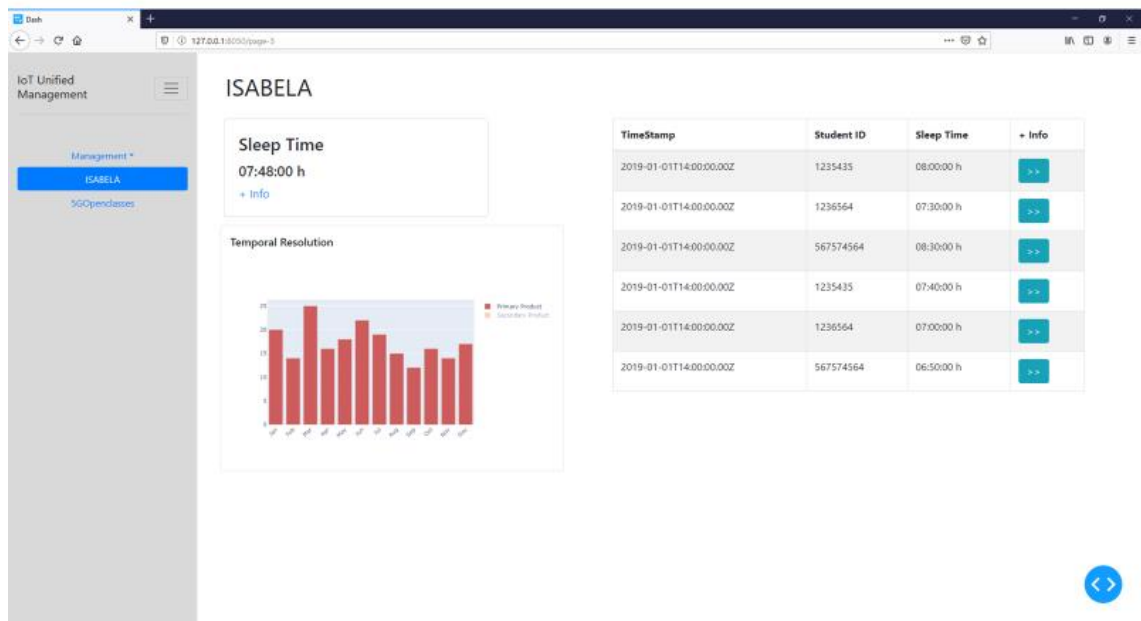


Figura 20 – Proposta visual da aplicação (12).

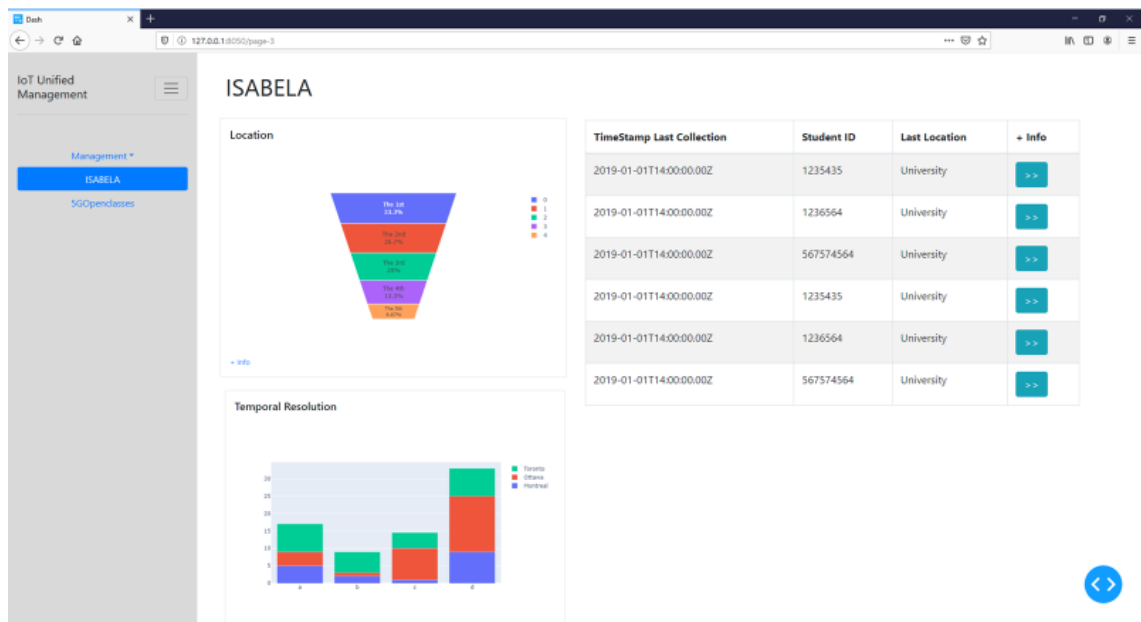


Figura 21 – Proposta visual da aplicação (13).

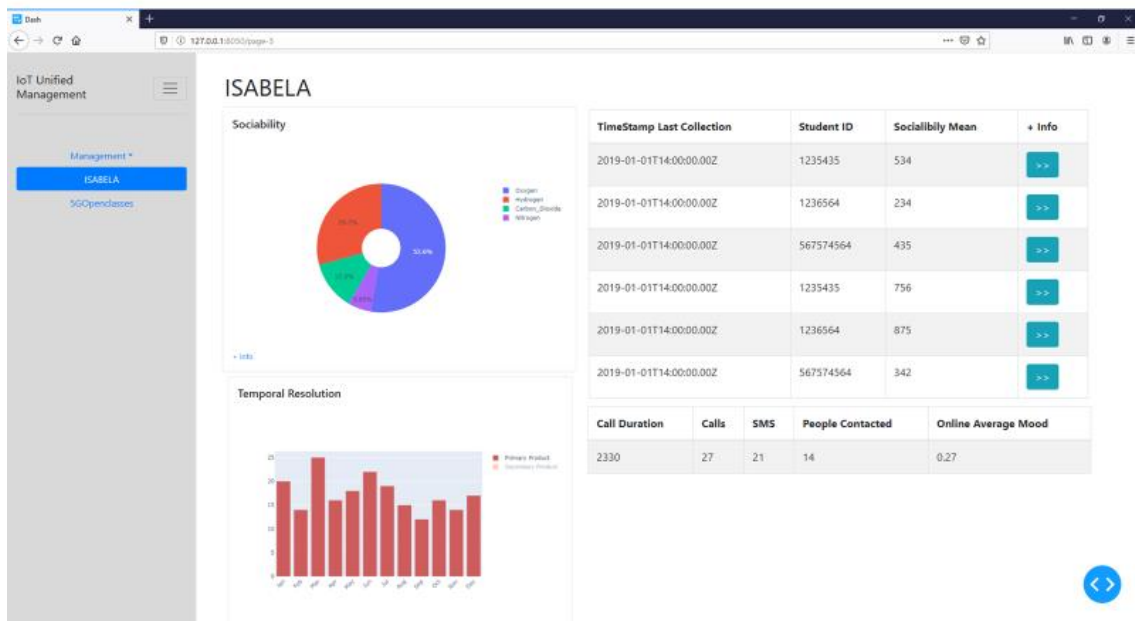


Figura 22 – Proposta visual da aplicação (14).

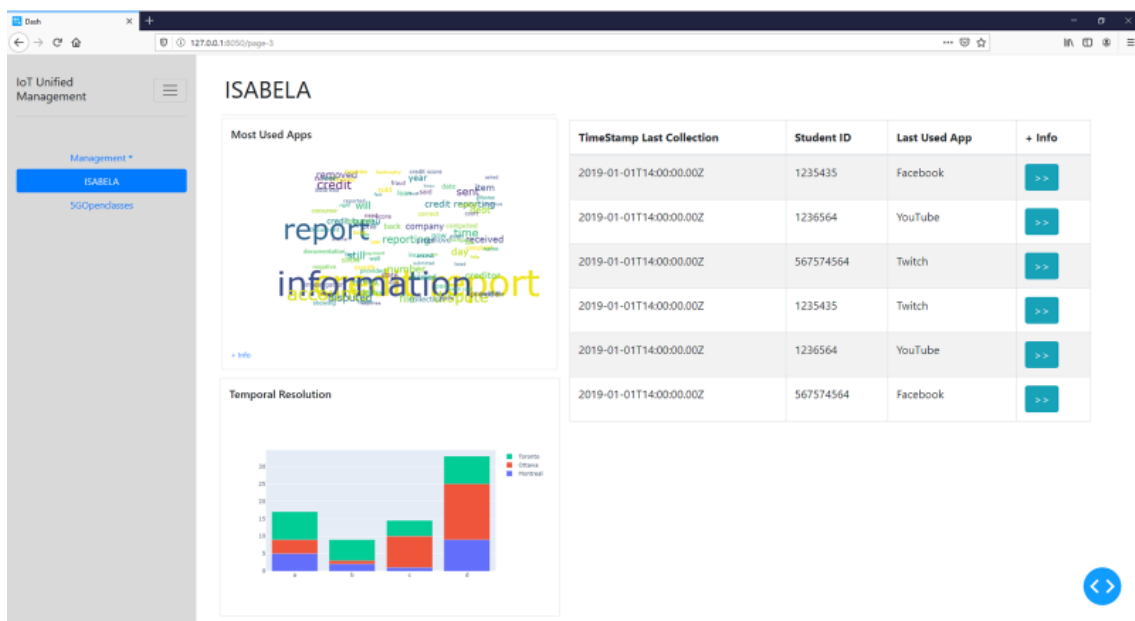


Figura 23 – Proposta visual da aplicação (15).

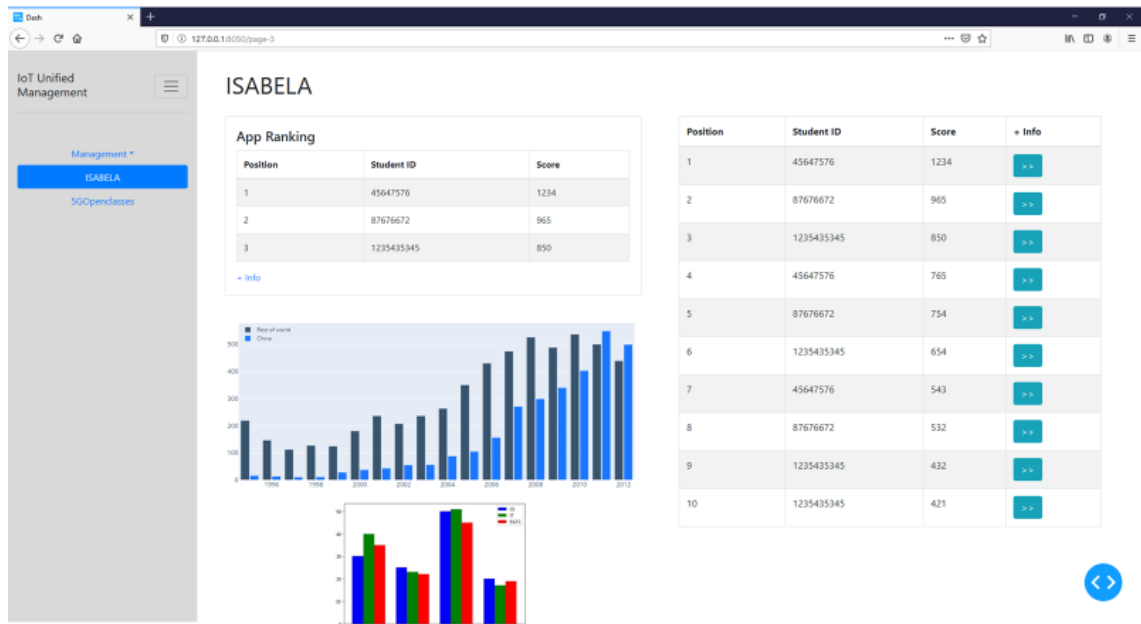


Figura 24 – Proposta visual da aplicação (16).

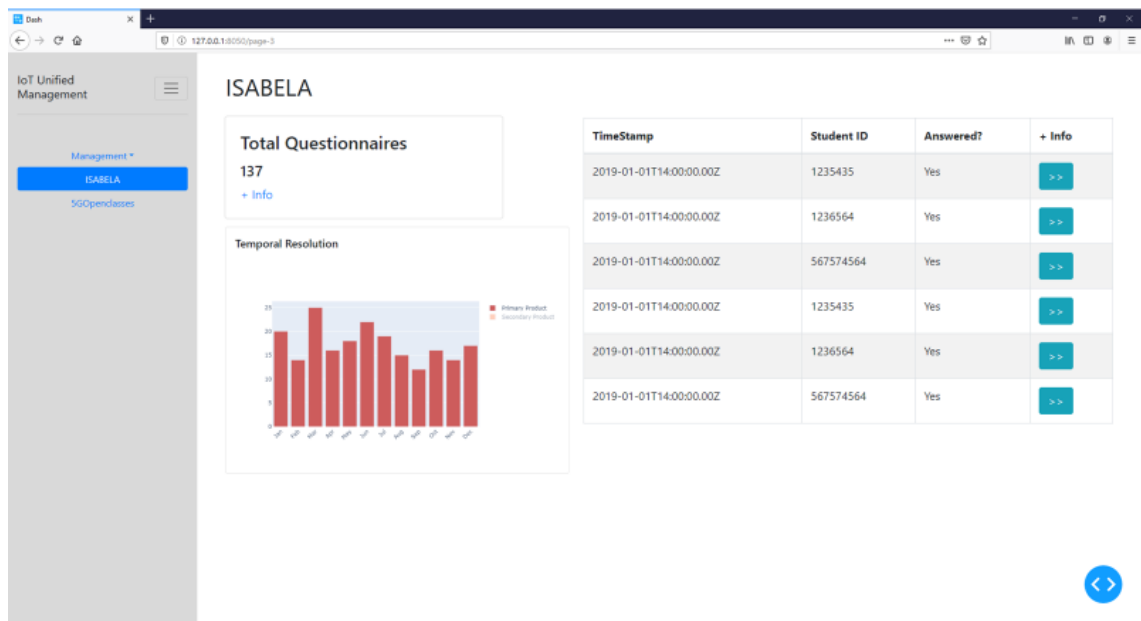


Figura 25 – Proposta visual da aplicação (17).

As Figuras 19 a 25 são representativas da proposta visual da secção de informações mais detalhadas referentes a cada componente de recolha de dados na aplicação móvel Android ISABELA. Nesta proposta a página encontra-se dividida verticalmente. Relativamente à parte direita da página, será apresentada uma lista dos alunos que pertencem à aplicação móvel ISABELA à qual estão a ser recolhidos dados. Todos os valores

de dados apresentados nesta parte direita da página são relativos à última recolha para cada um dos alunos.

No que diz respeito à parte esquerda da página, inicialmente serão apresentados dados relativos a informações das últimas 24 horas para todos os estudantes (gráficos de informações iguais aos presentes nas Figuras 17 e 18). Após selecionado um aluno da Tabela apresentada na parte direita da página (clicando no botão “+ Info”), tanto o gráfico superior como o inferior esquerdo serão atualizados para dados desse aluno selecionado. O gráfico superior apresentará informações relativas aos dados desse aluno nas últimas 24 horas e o gráfico inferior apresentará, numa perspetiva temporal, dados desse aluno dos últimos 5 dias.

### 5GOpenclasses:

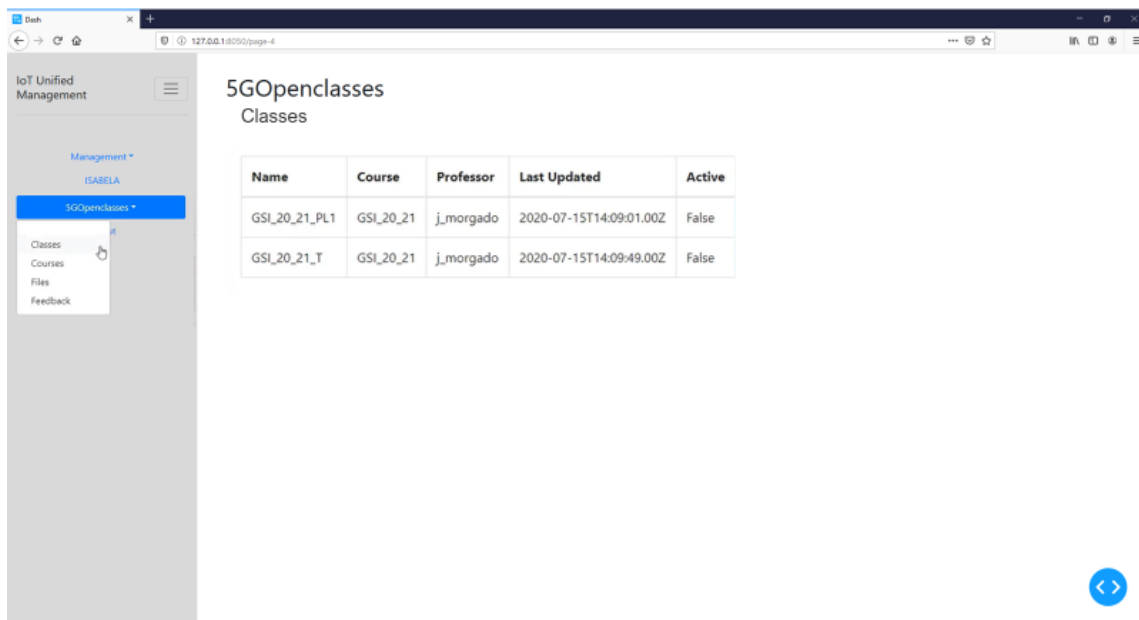


Figura 26 – Proposta visual da aplicação (18).

A Figura 26 representa a proposta visual relativa à listagem das aulas que se encontram na aplicação 5GOpenclasses. Estas aulas estão diferenciadas pelos diferentes professores e disciplinas.

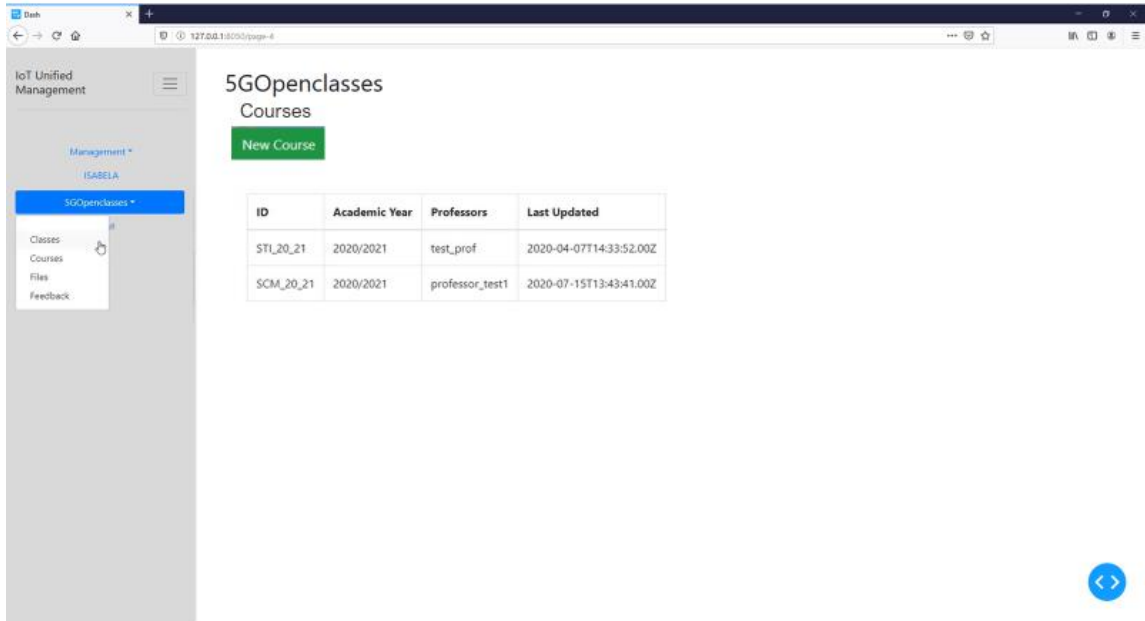


Figura 27 – Proposta visual da aplicação (19).

Já na Figura 27 encontra-se representada a proposta visual relativa à listagem das disciplinas que se encontram na aplicação 5GOpenclasses. Para além disso, é apresentado um botão responsável por redirecionar o utilizador para a página de adição de uma nova disciplina à aplicação. Esta disciplina é composta por atributos como: nome, ano académico e lista de professores associados a ela.

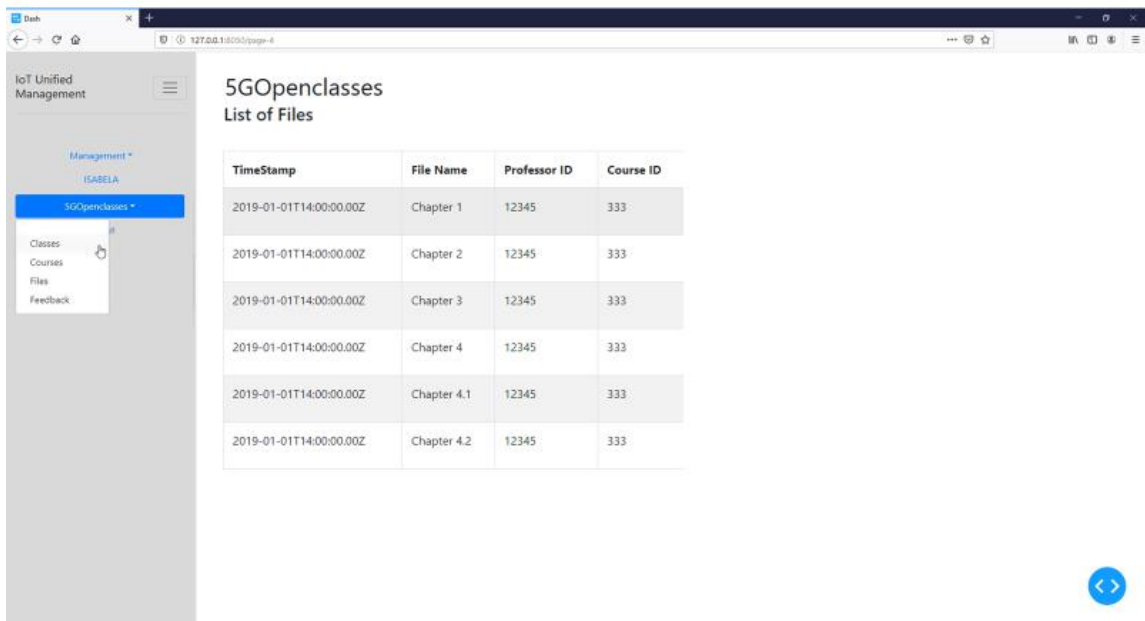


Figura 28 – Proposta visual da aplicação (20).

A Figura 28 representa a proposta visual relativa à listagem de materiais de suporte encontram na aplicação 5GOpenclasses.

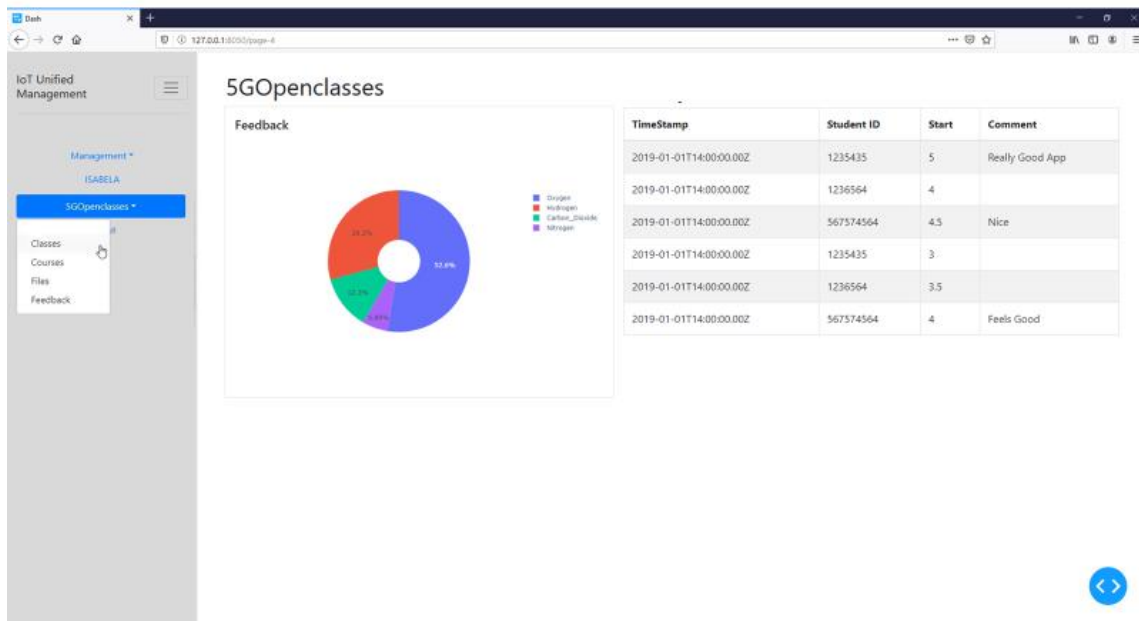


Figura 29 – Proposta visual da aplicação (21).

Por fim, a Figura 29 representa a proposta visual relativa à listagem dos resultados de classificação da qualidade de experiência (QoE) durante a utilização da aplicação móvel Android 5GOpenclasses (apresentada à direita na página). Para além disso é apresentado um gráfico *donut* que é responsável pela demonstração da soma de cada resultado da qualidade de experiência da utilização da aplicação móvel 5GOpenclasses nas últimas 24 horas.

### 3.6.2 Professor

Tal como foi descrito anteriormente, este perfil de utilizador é dedicado à aplicação “5GOpenclasses”. O professor terá apenas uma secção denominada “5GOpenclasses”, sendo por isso responsável por visualizar e gerir as suas aulas (conforme as disciplinas atribuídas pelo administrador), visualizar e gerir materiais de suporte e usufruir das funcionalidades de vídeo-aulas em tempo real disponibilizadas.



Os protótipos para este perfil de utilizador são os seguintes:

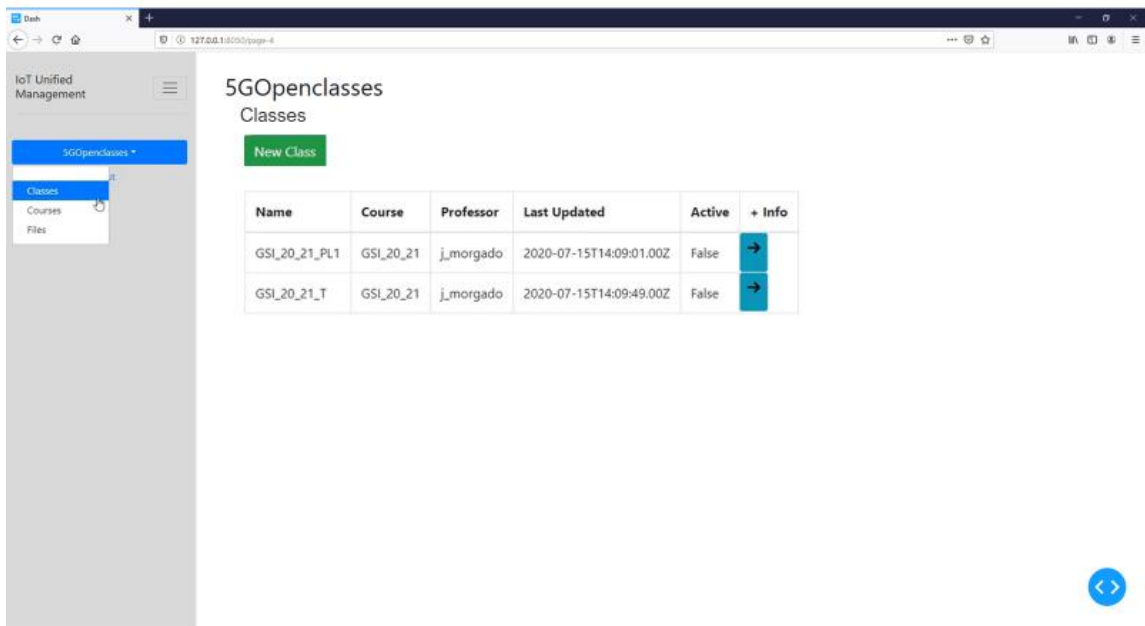


Figura 30 – Proposta visual da aplicação (22).

A Figura 30 representa a proposta visual relativa à listagem das aulas associadas a um professor que se encontram na aplicação 5GOpenclasses. Nesta listagem, o professor tem a possibilidade de adicionar uma nova aula à sua listagem clicando no botão “New Class” e ser redirecionado para uma aula já existente selecionando o botão da coluna “+ Info”.

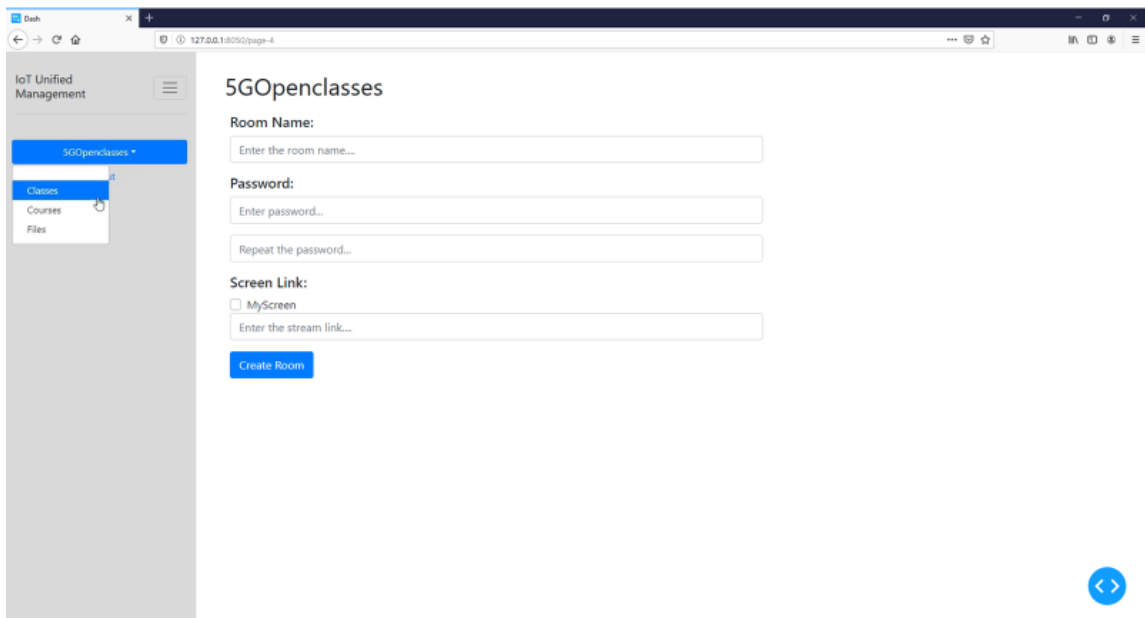


Figura 31 – Proposta visual da aplicação (23).

Na Figura 31 encontra-se a proposta visual relativa à adição de uma nova aula à aplicação 5GOpenclasses, associada ao utilizador em questão. Ao preencher todos os campos, o utilizador será redirecionado para a sala de aula em questão.

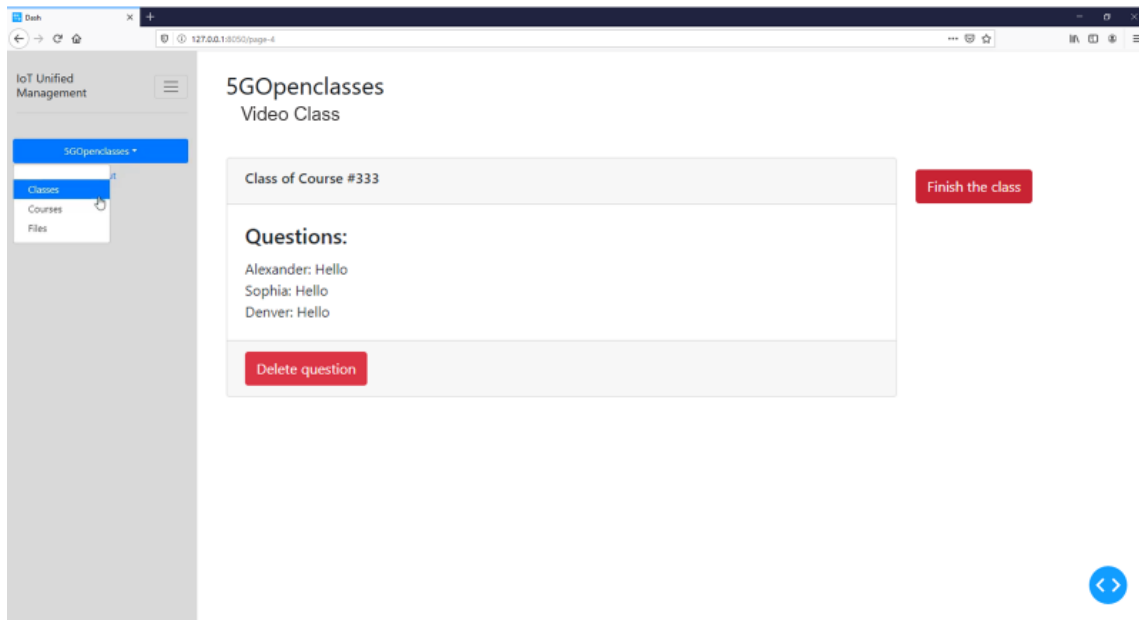


Figura 32 – Proposta visual da aplicação (24).

A Figura 32 representa a proposta visual relativa à vista do utilizador durante uma vídeo-aula, onde é possível visualizar e eliminar as perguntas dos alunos em tempo real. Para além disso, é possível efetuar o carregamento de arquivos para a plataforma e encerrar a aula a decorrer. Ao ser redirecionado para uma aula já existente, o utilizador indicará que essa aula está disponível para os alunos participarem e poderá, assim, inicial a transmissão de vídeo em tempo real, utilizando as ferramentas de transmissão WebRTC proporcionadas.

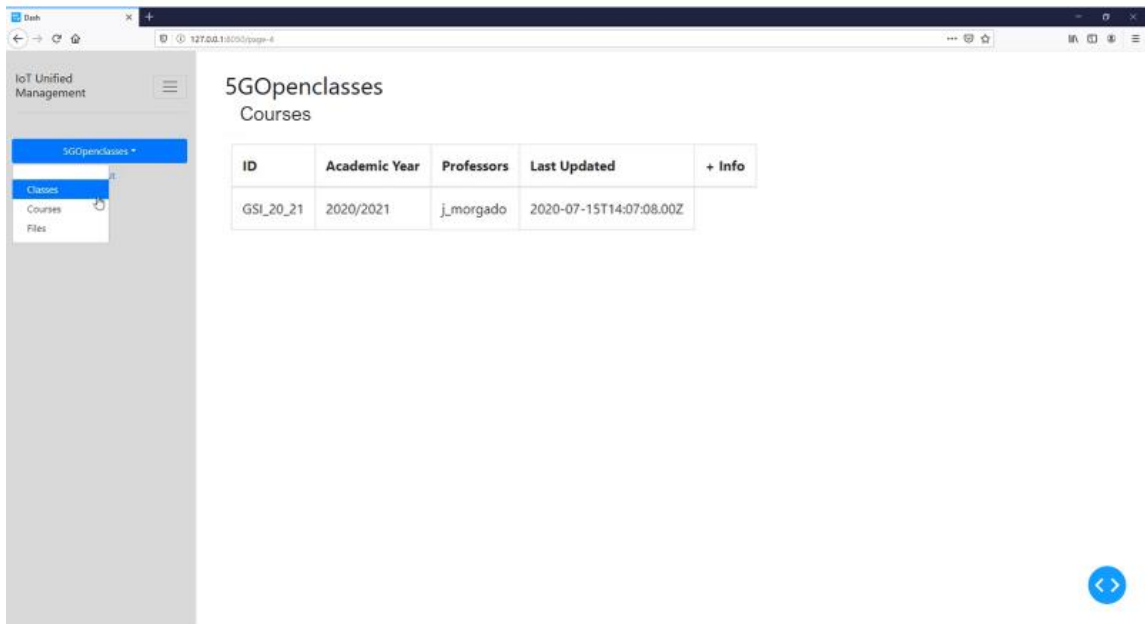


Figura 33 – Proposta visual da aplicação (25).

A proposta visual relativa à listagem das disciplinas associadas ao utilizador e que se encontram na aplicação 5GOpenclasses encontra-se apresentadas na Figura 33. Estas disciplinas são previamente associadas pelo perfil de utilizador administrador.

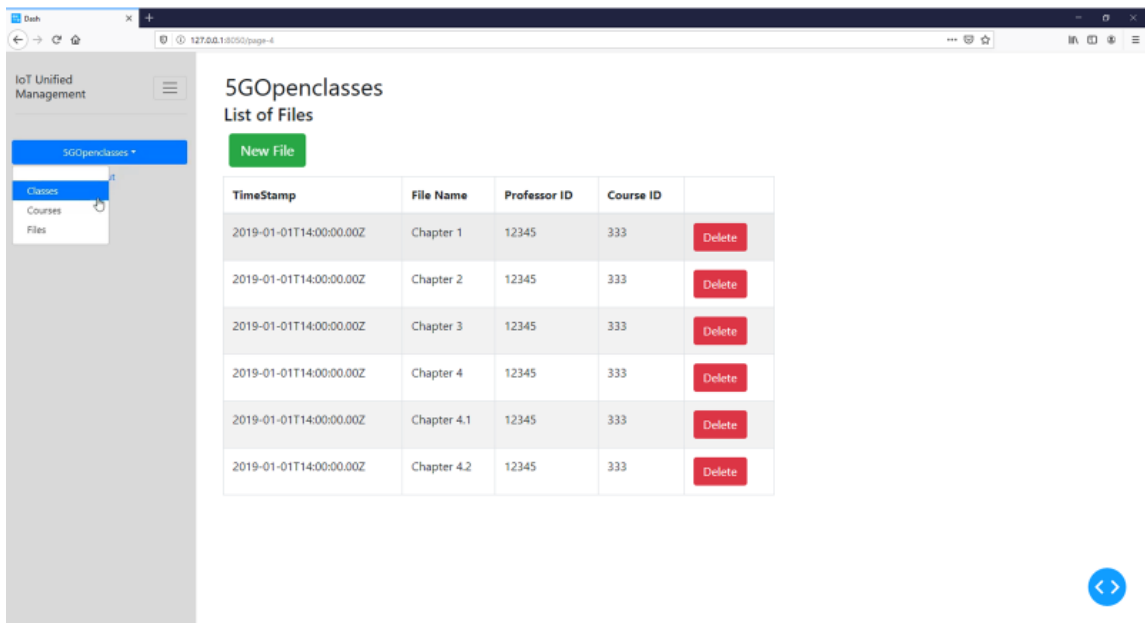


Figura 34 – Proposta visual da aplicação (26).

A Figura 34 representa a proposta visual relativa à listagem dos materiais de suporte que se encontram associados às disciplinas do utilizador e disponibilizadas na aplicação 5GOpenclasses.

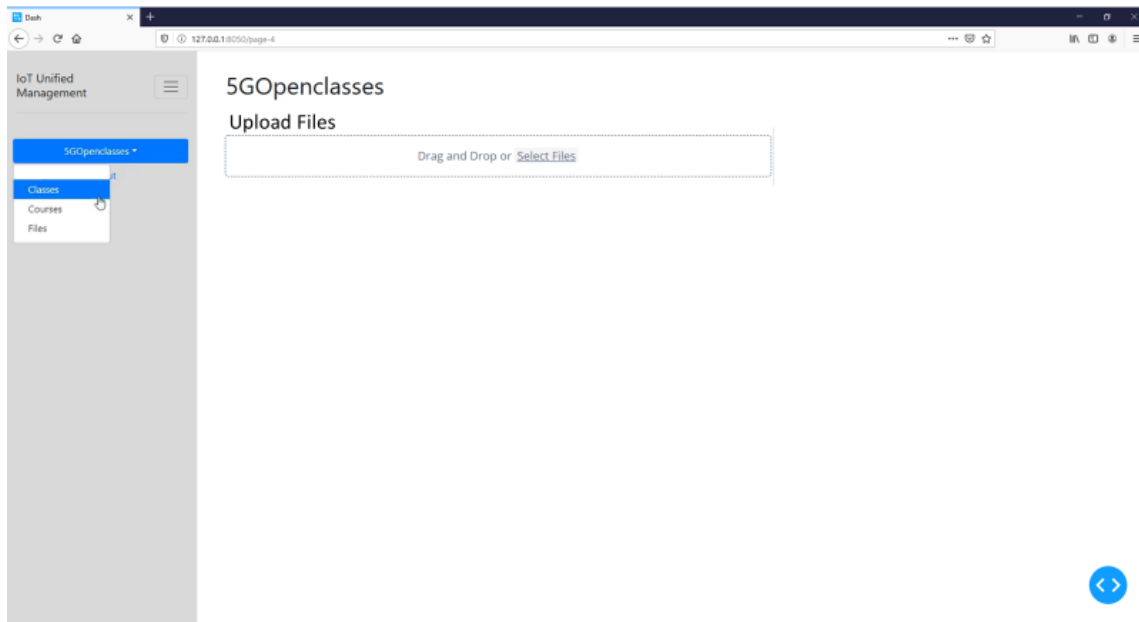


Figura 35 – Proposta visual da aplicação (27).

Finalmente, na Figura 35 encontra-se a proposta visual relativa à vista de carregamento de novos materiais de suporte associados às disciplinas do utilizador para a aplicação 5GOpenclasses.

# Capítulo 4

## Implementação

Este capítulo tem como objetivo a apresentação e explicação detalhada de todas as implementações elaboradas ao longo desta dissertação de mestrado. Este está dividido em dois subcapítulos distintos, um dedicado exclusivamente às implementações elaboradas na nossa aplicação “IoTUnifiedManagement” e outro dedicado exclusivamente às implementações elaboradas noutras aplicações, nomeadamente no BATINA. É de salientar que as implementações dedicadas à aplicação 5GOpenclasses, expostas na secção 5GOpenclasses do perfil **professor** na aplicação “IoTUnifiedManagement”, serão consideradas implementações desta própria aplicação.

### 4.1 IoTUnifiedManagement

As implementações elaboradas na aplicação podem ser descritas maioritariamente pelas seguintes categorias: integração com Prometheus, recolha de dados IoT e funcionalidades 5GOpenclasses. Como esta aplicação foi desenhada com o objetivo de apresentar dados IoT e fornecer funcionalidades de transmissão de vídeo-aulas em tempo real, a estrutura de dados de suporte utilizada no FIWARE é a seguinte:

Tabela III – Estrutura de dados “IoTUnifiedManagement”.

Entidades	Atributos
Administrador (Admin)	<i>Id, type, password, create_time e update_time</i>
Utilizador (User)	<b>5GOpenclasses:</b> <i>Id, type, name, email, password, is_online, in_class, create_time, update_time, TimeStamp, gyroscope, accelerometer, proximity, sound, light</i> <b>ISABELA:</b> <i>Id, type, Activity, location, calls_received, calls_made, calls_missed, calls_duration, sms_received, sms_sent, gps, TimeStamp, accelerometer, proximity, sound, light</i>
Professor (Teacher)	<i>Id, type, name, email, password, create_time, update_time</i>
Disciplina (Course)	<i>Id, type, academic_year, profs, create_time, update_time</i>

Sala (Room)	<i>Id, type, password, name, description, is_online, recording, prof_id, course_id, create_time, update_time, location</i>
Questões (Questions)	<i>Id, type, asking, room_id, user_id, create_time, update_time</i>
Salas de grupo (Group_Room)	<i>Id, type, creator_id, password, create_time, update_time</i>
Classificações (Rating)	<i>Id, type, comment, rate, recommend, room_id, create_time, update_time</i>
Materiais de suporte (Media_File)	<i>Id, type, download_path, file_name, room_id, create_time, update_time</i>
Ambientes (Environment)	<i>Id, type, TimeStamp, light, noise, temperature</i>

Todas as entidades têm um *id* único e são diferenciadas pelo seu tipo. Quase todas têm atributos que indicam quando foram criadas e qual foi a última atualização dessa entidade (através dos atributos *create\_time* e *update\_time*). Sempre que cada objeto é criado ou atualizado, os valores dos atributos *create\_time* e/ou *update\_time* são preenchidos pela data atual no formato ISO 8601 (ex: 2020-02-19T13:58:26.00Z). Para além disso, todas as passwords são encriptadas no formato de encriptação SHA-256.

Relativamente às entidades do tipo *user*, estas estão associadas aos alunos das aplicações móveis Android e são completas pelos seus diferentes atributos de dados. Na aplicação 5GOpenclasses, os atributos responsáveis por identificarem os valores de dados IoT recolhidos são: *TimeInstant, gyroscope, accelerometer, proximity, sound* e *light*. Já na aplicação ISABELA, os atributos com esse mesmo objetivo são: *activity, location, calls\_received, calls\_made, calls\_missed, calls\_duration, sms\_received, sms\_sent, gps, TimeStamp, accelerometer, proximity, sound* e *light*. Os objetos de dados IoT das aplicações 5GOpenclasses e ISABELA são diferentes, mas o seu objetivo para a aplicação “IoTUnifiedManagement” é o mesmo. Por outro lado, os atributos *is\_online* e *in\_class* da aplicação 5GOpenclasses são responsáveis por controlar se os alunos estão *online* ou numa aula, respetivamente.

Quanto às entidades do tipo *course*, estas são compostas por atributos como *academic\_year* e *profs* sendo estes atributos responsáveis por identificar o ano académico da disciplina em questão (ex:2019/2020) e qual a lista de professores, respetivamente. Por outro lado, as entidades do tipo *room* têm atributos como *location, is\_online, recording, prof\_id* e *course\_id*, sendo estes atributos responsáveis por identificar a localização de uma sala, identificar se uma aula se encontra a ser lecionada, se essa aula é para ser gravada para um ficheiro passível de ser vista mais tarde e qual o professor e o curso associado a ela, respetivamente.

No que diz respeito às entidades do tipo *questions*, estas são compostas por atributos como *asking, room\_id* e *user\_id*, sendo estes atributos responsáveis por identificar a questão colocada pelo aluno, qual a sala destinada a questão e que aluno fez a questão, respetivamente.

As entidades do tipo *rating* são compostas por atributos como *comment, rate, recommend* e *room\_id*, sendo estes responsáveis por identificar o comentário feito ao questionário de QoE da aplicação móvel Android 5GOpenclasses, qual a classificação (entre

0 e 5 estrelas), quais as possibilidades de recomendar a aplicação a outra pessoa e qual a sala de aula que precedeu esse aluno.

As entidades do tipo *média\_file* são compostas por atributos como *download\_path*, *file\_name* e *room\_id*, responsáveis por identificar qual o *path* onde podemos fazer o download do ficheiro, qual o seu nome e qual a disciplina que está associado esse ficheiro.

Por último, as entidades do tipo *environment* são compostas por atributos como *TimeStamp*, *light*, *noise* e *temperature*, atributos esses responsáveis por identificar o valor do atributo de luz, som e temperatura, para uma determinada data no formato ISO 8601 (ex: 2020-02-19T13:58:26.00Z).

Todas as comunicações entre a aplicação e o *middleware* FIWARE são neste momento efetuadas através do protocolo HTTPS. O padrão de troca de dados com o FIWARE é feito através do formato JSON sendo que por sua vez, esses dados podem ser armazenados numa base de dados MySQL, MongoDB ou CKAN.

### 4.1.1 Integração com o Prometheus

Numa primeira fase de desenvolvimento, a integração entre a aplicação “ToTUnifiedManagement” e o Prometheus foi implementada com sucesso num espaço de 8 dias. Como o Prometheus é um sistema de monitorização e de armazenamento de dados numa resolução temporal, torna-se deste modo possível a obtenção de informações dos diferentes serviços e componentes do servidor através da utilização de *queries*. Estas *queries* são responsáveis por questionar os diferentes *endpoints* e recolher as suas informações. Neste projeto em específico, este sistema está a ser utilizado para recolher e armazenar informações tais como:

- Informações dos estados dos serviços “Orion”, “Idas”, “Comet” e “Find” do servidor;
- Informações de percentagem de utilização dos componentes do servidor, tais como: CPU, memória RAM, memória SWAP e a carga do CPU (numa média de 5 minutos);
- Informações dos componentes do servidor, tais como: memória RAM total, memória SWAP total, carga do sistema (numa média de 1 minuto), o tempo de atividade do servidor e o espaço de armazenamento livre;
- Informações numa resolução temporal de 24 horas: tempo de resposta, percentagem de utilização do CPU e percentagem de utilização da memória RAM.

Detalhando, as *queries* a executar pelo Prometheus, responsáveis pela obtenção de todas as informações indicadas anteriormente são as seguintes:

- Estado do serviço “Orion”:

```
probe_success{instance="socialiteorion2.dei.uc.pt:9014/v2/entities",job="blackbox"}
```

- Estado do serviço “Idas”:

```
probe_success{instance="socialiteorion2.dei.uc.pt:9002/iot/devices",job="blackbox"}
```

- Estado do serviço “Comet”:

```
probe_success{instance="socialiteorion2.dei.uc.pt:9005/STH/v1/contextEntities/type/Environment/id/University1/attributes/temperature?lastN=100",job="blackbox"}
```

- Estado do serviço “Find”:

```
probe_success{instance="socialiteorion2.dei.uc.pt:9012/status",job="blackbox2"}
```

- Percentagem de utilização do CPU:

```
((count(count(node_cpu_seconds_total{instance="nodeexporter:9100",job="node-exporter"}) by (cpu))) - avg(sum by mode)(irate(node_cpu_seconds_total{mode="idle",instance="nodeexporter:9100",job="node-exporter"}[5m]))) * 100) / count(count(node_cpu_seconds_total{instance="nodeexporter:9100",job="node-exporter"}) by (cpu))
```

- Percentagem de utilização da memória RAM:

```
100-((node_memory_MemAvailable_bytes{instance="nodeexporter:9100",job="node-exporter"} * 100)/node_memory_MemTotal_bytes{instance="nodeexporter:9100",job="node-exporter"})
```

- Percentagem de utilização da memória SWAP:

```
((node_memory_SwapTotal_bytes{instance="nodeexporter:9100",job="node-exporter"} - node_memory_SwapFree_bytes{instance="nodeexporter:9100",job="node-exporter"}) / (node_memory_SwapTotal_bytes{instance="nodeexporter:9100",job="node-exporter"})) * 100
```

- Percentagem de utilização da carga do CPU (numa média de 5 minutos):

```
avg(node_load5{instance="nodeexporter:9100",job="node-exporter"})/count(count(node_cpu_seconds_total{instance="nodeexporter:9100",job="node-exporter"}) by (cpu)) * 100
```

- Memória RAM total:

```
node_memory_MemTotal_bytes{instance="nodeexporter:9100",job="node-exporter"}
```

- Memória SWAP total:

```
node_memory_SwapTotal_bytes{instance="nodeexporter:9100",job="node-exporter"}
```

- Carga do sistema (numa média de 1 minuto):

```
node_load1{instance="nodeexporter:9100",job="node-exporter"}
```

- Tempo de atividade do servidor:

```
node_time_seconds{instance="nodeexporter:9100",job="node-exporter"} - node_boot_time_seconds{instance="nodeexporter:9100",job="node-exporter"}
```

- Armazenamento livre:

```
node_filesystem_avail_bytes{device="/dev/vda1",fstype="ext4",instance="nodeexporter:9100",job="node-exporter",mountpoint="/etc/hosts"}
```



- Tempo de resposta do servidor numa resolução temporal de 24 horas:

Tempo de resposta para “Orion”:

```
probe_duration_seconds{instance="socialiteorion2.dei.uc.pt:9014/v2/entities",job="blackbox"} +&start=' + date_start +&end=' + date_end +&step=' + step + 's'
```

Tempo de resposta para “Idas”:

```
probe_duration_seconds{instance="socialiteorion2.dei.uc.pt:9002/iot/devices",job="blackbox"} +&start=' + date_start +&end=' + date_end +&step=' + step + 's'
```

Tempo de resposta para “Comet”:

```
probe_duration_seconds{instance="socialiteorion2.dei.uc.pt:9005/STH/v1/contextEntities/type/Environment/id/University1/attributes/temperature?lastN=100",job="blackbox"} +&start=' + date_start +&end=' + date_end +&step=' + step + 's'
```

Tempo de resposta para “Find”:

```
probe_duration_seconds{instance="socialiteorion2.dei.uc.pt:9012/status",job="blackbox2"} +&start=' + date_start +&end=' + date_end +&step=' + step + 's'
```

- Percentagem de utilização do CPU do servidor numa resolução temporal de 24 horas:

**Busy System:**

```
sum by
(instance)(rate(node_cpu_seconds_total{mode="system",instance="nodeexporter:9100",job="node-exporter"}[5m]))*100 + '&start=' + date_start + '&end=' + date_end + '&step=' + step + 's'
```

**Busy User:**

```
sum by (instance)(rate(node_cpu_seconds_total{mode="user",instance="nodeexporter:9100",job="node-exporter"}[5m]))*100 + '&start=' + date_start + '&end=' + date_end + '&step=' + step + 's'
```

**Busy Iowait:**

```
sum by
(instance)(rate(node_cpu_seconds_total{mode="iowait",instance="nodeexporter:9100",job="node-exporter"}[5m]))*100 + '&start=' + date_start + '&end=' + date_end + '&step=' + step + 's'
```

**Busy Irqs:**

```
sum by (instance)(rate(node_cpu_seconds_total{mode=~".*irq",instance="nodeexporter:9100",job="node-exporter"}[5m]))*100 + '&start=' + date_start + '&end=' + date_end + '&step=' + step + 's'
```

**Busy Idle:**

```
sum by (instance)(rate(node_cpu_seconds_total{mode="idle",instance="nodeexporter:9100",job="node-exporter"}[5m]))*100 + '&start=' + date_start + '&end=' + date_end + '&step=' + step + 's'
```

**Busy Other:**

```
sum
(rate(node_cpu_seconds_total{mode!="idle",mode!="user",mode!="system",mode!="iowait",mode!="irq",mode!="softirq",instance="nodeexporter:9100",job="node-exporter"}[5m]))*100 + '&start=' + date_start + '&end=' + date_end + '&step=' + step + 's'
```

- Percentagem de utilização de memória RAM numa resolução temporal de 24 horas:

**RAM Total:**

```
node_memory_MemTotal_bytes{instance="nodeexporter:9100",job="node-exporter"} +&start=' +  
date_start +&end=' + date_end +&step=' + step +&s'
```

**Free RAM:**

```
node_memory_MemFree_bytes{instance="nodeexporter:9100",job="node-exporter"} +&start=' +  
date_start +&end=' + date_end +&step=' + step +&s'
```

**Used RAM:**

```
node_memory_MemTotal_bytes{instance="nodeexporter:9100",job="node-exporter"}-  
node_memory_MemFree_bytes{instance="nodeexporter:9100",job="node-exporter"}-  
node_memory_Cached_bytes{instance="nodeexporter:9100",job="node-exporter"}-  
node_memory_Buffers_bytes{instance="nodeexporter:9100",job="node-exporter"} +&start=' + date_start  
+&end=' + date_end +&step=' + step +&s'
```

**Cache Buffer:**

```
node_memory_Cached_bytes{instance="nodeexporter:9100",job="node-  
exporter"}+node_memory_Buffers_bytes{instance="nodeexporter:9100",job="node-exporter"} +&start=' +  
date_start +&end=' + date_end +&step=' + step +&s'
```

**Swap Used:**

```
node_memory_SwapTotal_bytes{instance="nodeexporter:9100",job="node-exporter"}-  
node_memory_SwapFree_bytes{instance="nodeexporter:9100",job="node-exporter"} +&start=' +  
date_start +&end=' + date_end +&step=' + step +&s'
```

Desde a iniciação do serviço no servidor, o Prometheus já se encontrava a recolher informações dos estados dos serviços "Orion", "Idas", "Comet" e "Find". Na fase de definição de requisitos do *dashboard* a recolha e apresentação destas informações não foram consideradas sequer como obrigatórias mas, visto que seria apenas necessário adicionar um novo gráfico à interface, decidimos contemplar um gráfico dedicado à apresentação destes conteúdos. Para que essas informações sejam recolhidas, o Prometheus executa as seguintes *queries*:

```

Status Orion:
probe_success[instance="socialiteorion2.dei.uc.pt:9014/v2/entities",job="blackbox"] + '&start=' +
date_start + '&end=' + date_end + '&step=' + step + 's'

Status Idas:
probe_success[instance="socialiteorion2.dei.uc.pt:9002/iot/devices",job="blackbox"] + '&start=' +
date_start + '&end=' + date_end + '&step=' + step + 's'

Status Comet:
probe_success[instance="socialiteorion2.dei.uc.pt:9005/STH/v1/contextEntities/type/Environment/id/Un
iversity1/attributes/temperature?lastN=100",job="blackbox"] + '&start=' + date_start + '&end=' + date_end
+ '&step=' + step + 's'

Status Find:
probe_success[instance="socialiteorion2.dei.uc.pt:9012/status",job="blackbox2"] + '&start=' + date_start
+ '&end=' + date_end + '&step=' + step + 's'

```

Para toda a resolução temporal de 24 horas, o intervalo de amostragem para a obtenção de resultados será sempre recolhido com um valor de 300 segundos, ou seja, 5 minutos. Este valor pode ser explicado por: quanto menor for o número de intervalo de amostragem para a obtenção dos dados, maior vai ser a quantidade de dados para processar e maior o tempo de espera do utilizador até que estes lhes sejam apresentados.

#### 4.1.2 Recolha de dados IoT

Relativamente à recolha de dados dos diferentes sensores IoT, estes são capturados através das aplicações móveis Android ISABELA e 5GOpenclasses, assim como pelas *boxes* da plataforma (boxes ISABELA). Mais detalhadamente, a aplicação móvel *android* ISABELA é responsável por recolher dados IoT como *gps*, *accelerometer*, *proximity*, *sound* e *light*. Já a aplicação móvel Android 5GOpenclasses é responsável por recolher dados IoT como *gyroscope*, *accelerometer*, *proximity*, *sound* e *light*, e as boxes da plataforma são responsáveis pela recolha de dados IoT como *light*, *noise* e *temperature*.

Passando aos dados IoT capturados pela aplicação móvel Android ISABELA, estes são recolhidos por uma aquisição direta dos valores dos sensores implantados no *hardware* dos *smartphones*. Para capturar os valores do sensor do *smartphone*, o sistema operacional Android fornece uma API denominada **android.hardware** onde poderão ser recolhidas essas informações [27]. Esta API possui classes dedicadas à gestão da aquisição de sensores, nomeadamente a classe *SensorManager* [28]. Com esta classe é possível recuperar valores de grande parte dos sensores de um *smartphone*. As etapas para esta aquisição são as seguintes:

1. Obter a instância de gestão do sensor a partir do *context* da aplicação;
2. Obter o sensor do `SensorManager`;
3. Criar um *listener* que implementa o código que precisamos de executar quando um valor de um sensor altera;
4. Registrar o *listener* desse sensor na instância de gestão do sensor.

Estas etapas necessitam de ser replicadas para todos os diferentes sensores que desejamos recolher dados, utilizando sempre o mesmo *listener*. Para distinguir qual dos sensores foi acionado, precisamos apenas de implementar uma condição que verifique qual sensor que disparou o evento e indicar o código a executar para esse evento. Um evento pode ser acionado aquando da mudança de valor de um sensor e este evento retornará um objeto do tipo `SensorEvent` [29].

Os objetos do tipo `SensorEvent` têm 4 atributos que podem ser recuperados: a precisão, o tipo de sensor, a data e hora de criação desse objeto, e os valores do respetivo sensor. Dentro destes atributos, o mais importante é o atributo *valor* pois, este é o responsável por retornar os valores que pretendemos. Na aplicação móvel ISABELA foi implementada uma interface `SensorEventListener` num serviço. Este serviço é responsável pela aquisição e armazenamento local dos dados dos diferentes sensores e trabalha com um temporizador. Sempre que esse temporizador é acionado, o valor da aquisição dos diferentes sensores é armazenado. Este armazenamento local é suportado pela biblioteca `Android greenDAO` e foi elaborado para evitar a perda de dados quando não houvesse uma conexão disponível com o *middleware* FIWARE [30]. Esta aquisição é dedicada diretamente aos sensores do tipo *proximity*, *light* e *accelerometer*.

Em relação aos sensores do tipo *sound*, os valores de aquisição podem ser coletados a partir da API `Android MediaRecorder` [31]. Essa API regista a amplitude de resposta do microfone e armazena os seus valores. O fluxo de armazenamento destes dados segue a mesma terminologia indicada anteriormente, onde é armazenado o valor obtido pela recolha de dados deste sensor, sempre que o temporizador for acionado.

Relativamente ao sensor do tipo GPS, os dados de aquisição deste sensor são recolhidos com a ajuda `API Fused Location Provider`, proveniente da Google [32]. Esta API é responsável por recolher e armazena informações de localização a partir dos dados fornecidos pelo sensor. O armazenamento acontece de acordo com a terminologia anteriormente discutida.

Relativamente à implementação na aplicação móvel `Android`, esta encontra-se implementada exatamente com a mesma configuração, mas desta vez para os sensores *gyroscope*, *accelerometer*, *proximity*, *sound* e *light*. Os dados recolhidos pelo sensor *gyroscope* são recolhidos pela classe `SensorEvent`, tal como acontece com a recolha de dados dos sensores *proximity*, *light* e *accelerometer*.

Por último, os sensores das boxes da plataforma recolhem dados dos sensores *light*, *noise* e *temperature*. Tal como foi descrito anteriormente, estas boxes são compostas por um *gateway* `Raspberry Pi`, uma *board* `Arduino` e pelos seus diversos sensores complementares. A *board* `Arduino` tem como objetivo o processamento os dados dos sensores e por isso estes sensores estão conectados a ela. Como esta não tem a capacidade de conexão à Internet, foi necessária a utilização de um *gateway* `Raspberry Pi`.

Relativamente às comunicações com o *middleware* FIWARE, atualmente todo o processo de conexão e partilha de dados IoT com este *middleware* acontece usando pedidos `HTTPS/REST`. Estas comunicações entre dispositivos fornecedores de dados IoT e *middleware* FIWARE poderiam ser melhoradas. No plano inicial de trabalhos, esta melhoria encontrava-

se integrada. No entanto, apenas será realizada num futuro próximo. Esta melhoria incluí a utilização de diferentes protocolos de comunicação que sejam dedicados a dispositivos IoT. Para isso, poderíamos utilizar os agentes Lightweight M2M e/ou o Ultralight 2.0 [25], [26], uma vez que estes agentes IoT são destinados para trabalhar com comunicações e dispositivos restritos, onde a largura de banda e memória do dispositivo podem ser recursos limitados. Numa visão mais detalhada e relativamente ao Ultralight, este agente utiliza um *payload* para troca de dados constituída por pares de chave-valor separados por uma barra vertical. Por outras palavras, para comunicar um valor com o *middleware* FIWARE, a *payload* de dados será constituída por algo semelhante a: “sensorx|123”. A informação “sensorx” é relativa a um determinado tipo de sensor e a informação “123” é relativa aos valores de dados IoT recolhidos para este sensor. Por outro lado, o LwM2M foi desenhado com o objetivo de gerir aplicações e dispositivos IoT. Com esta oferta de controlo e gestão de dispositivos IoT, um dispositivo pode usar o mesmo protocolo tanto para atividades de atuação e deteção, como para atividades de gestão. O LwM2M é um agente IoT que aceita pedidos CoAP e os redireciona para consumidores NGSI, como por exemplo o FIWARE Orion. Por outras palavras, o LwM2M é um agente que fornece uma ponte de conexão direta com o FIWARE e enquadra-se melhor com as nossas necessidades. Uma vez escolhido um agente a utilizar, para ser possível atuar nos diferentes sensores, podemos utilizar um *smartphone* e uma aplicação para recolher dados, tal como acontece com as aplicações Android ISABELA e 5GOpenclasses. Como a aplicação “IoTUnifiedManagement” tem por objetivo obter dados e atuar nos diferentes tipos de sensores IoT, seria a partir desta integração que executaríamos operações CRUDN (*Create, Read, Update, Delete, Notify*) nas entidades monitorizadas. As diferentes aplicações móveis Android necessitariam de incorporar um cliente LwM2M e teriam de seguir os seguintes passos para possibilitar o FIWARE Orion de gerir as suas entidades:

1. Fornecimento da porta administrativa do agente IoT nas aplicações móveis Android. Aqui o FIWARE fica encarregado de criar uma estrutura de dados para cada dispositivo monitorizado;
2. Conexão entre o cliente LwM2M (nas aplicações móveis Android) e o agente IoT no FIWARE;
3. Exposição das informações dos clientes ao FIWARE e habilitação das respetivas interações.

A partir deste momento, novas leituras fornecidas pelos sensores IoT (com valores diferentes do anterior) irão desencadear uma nova atualização no FIWARE. Por último, de maneira a possibilitar a edição de dados no FIWARE (como por exemplo, a frequência de recolha), seria necessário enviar dados formatados em JSON para o FIWARE com o valor dos atributos atualizado, utilizando o protocolo HTTPS. Sempre que acionada uma atualização, o FIWARE consultará o agente IoT dentro do domínio NGSI (LwM2M) para que este agente comunique com o respetivo cliente LwM2M através do protocolo de ponte NGSI-LwM2M. Esta mesma ideia poderá também ser utilizada em sensores como os pertencentes às boxes da plataforma.

### 4.1.3 Funcionalidades 5GOpenclasses

Relativamente às funcionalidades implementadas na aplicação “IoTUnifiedManagement” dedicadas à aplicação 5GOpenclasses, estas são maioritariamente compostas por: gestão disciplinas, gestão de vídeo-aulas, gestão de materiais de suporte e gestão do *feedback*. Todas estas funcionalidades de gestão são suportadas pelo *middleware*

FIWARE, pois este é responsável por armazenar estes objetos de dados nas suas diferentes bases de dados (MySQL, MongoDB, CKAN).

Outra importante funcionalidade implementada na aplicação “IoTUnifiedManagement” dedicada à aplicação 5GOpenclasses é a transmissão de vídeo em tempo real, através do servidor multimédia WebRTC Kurento. O servidor multimédia WebRTC Kurento tem uma arquitetura de transmissão de fluxos de vídeo em tempo real de **1 para muitos** (*One to Many video call*). Este é suportado por outros dois servidores (STUN/TURN), (STUN/TURN e Media-Server-Helper), um responsável pelo estabelecimento de ligações seguras entre dispositivos e outro como auxiliar para clientes web, respetivamente. Com estas particularidades, o perfil de utilizador professor tem a possibilidade de transmitir os seus fluxos de vídeo para os alunos que utilizam a aplicação móvel Android 5GOpenclasses. Esta aplicação oferece aos utilizadores (alunos) a possibilidade de se registarem, de aceder à aplicação, de aceder aos materiais de suporte que se encontram disponíveis para download, conversar num chat de grupo e de aceder a vídeo-aulas que estão a acontecer em tempo real.

Já as funcionalidades de gestão são suportadas pelo *middleware* FIWARE e são as seguintes: gestão de disciplinas, gestão de vídeo-aulas, gestão de materiais de suporte e gestão de *feedback*. De acordo com estas funcionalidades, apenas a gestão de *feedback* é dedicada exclusivamente ao perfil administrador. Esta encontra-se dividida em apresentação visual de um resumo dos resultados de *feedback* das últimas 24 horas e uma lista de todos os *feedbacks* submetidos até ao momento. A primeira é composta por um gráfico do tipo *donut*, onde é apresentado o somatório dos resultados do questionário QoE da aplicação 5GOpenclasses. A classificação do QoE varia entre valores de 0 a 5 (com intervalos de 0,5). Se o conjunto de resultados for de uma classificação de 4 e outra de 4,5, a apresentação do gráfico *donut* será de 50% para ambos os valores, pois apenas existem duas classificações nas últimas 24 horas e para cada um dos valores existe apenas uma submissão. Se existissem duas submissões e ambas tivessem um valor de 4, seria apresentado o gráfico *donut* com 100% para esse valor.

Relativamente à gestão de disciplinas, o perfil administrador é responsável por adicionar novas disciplinas (introduzindo os dados necessários para os atributos da disciplina) aos diversos professores existentes na plataforma, os quais apenas podem visualizar uma lista das disciplinas que lhes são atribuídas. No que diz respeito à gestão de vídeo-aulas, o perfil de administrador só tem a possibilidade de visualizar uma lista das vídeo-aulas existentes na plataforma. Ao passo que, o perfil professor tem a possibilidade de adicionar novas vídeo-aulas às suas disciplinas, introduzindo os dados necessários nos atributos da sua vídeo-aula.

Quanto à gestão de materiais de suporte, o perfil de administrador apenas tem a possibilidade de visualizar uma lista de todos os materiais de suporte na plataforma, enquanto que, o perfil professor tem a possibilidade de adicionar, editar e eliminar estes materiais. Estes materiais de suporte podem ser compostos pelos diferentes tipos de materiais (como PowerPoints, PDF's, vídeos de gravações de vídeo-aula, etc.) e na sua edição apenas é possível alterar o atributo nome do ficheiro.

As funcionalidades de transmissão de vídeo em tempo real são suportadas por três servidores distintos: servidor multimédia Kurento WebRTC, servidor STUN/TURN e servidor Media-Server-Helper. A escolha de utilização deste servidor multimédia Kurento WebRTC advém da recomendação que o catálogo de aplicações do FIWARE proporciona aos consumidores finais. Neste caso, o FIWARE inclui um *generic enabler* dedicado ao

Kurento WebRTC, que permite o processamento de fluxos de mídia em tempo real, facilitando assim a sua integração.

Devido à segurança proporcionada pelas *firewalls* e NATs, surgiu a necessidade de utilização de um servidor STUN/TURN para ultrapassar tal barreira. A implementação deste servidor STUN/TURN advém da preferência de construção de uma solução local sem custos. Para que seja possível comunicar em tempo real, todos os dispositivos que pretendem oferecer ou receber o conteúdo de dados, têm de estar sincronizados temporalmente. Este processo de sincronização simultânea intitula-se de *signaling* e serve para trocar informações de controlo de sessão, mensagens de erro, dados do canal de transmissão (nomeadamente codificadores e definições de codificadores, largura de banda e tipos de mídia), dados chave (para estabelecer ligações seguras) e dados da rede (endereço IP e porto). Para que seja possível ultrapassar a segurança das *firewalls* e NATs, as aplicações WebRTC podem usar a *framework* ICE para superar a complexidade das redes atuais. O ICE tenta encontrar o melhor “caminho” para conectar dispositivos e, para isso, ele tem de testar a seguinte ordem de opções:

1. Tentativa de conexão direta entre dispositivos (impossível devido aos NATs);
2. Utilização de um endereço externo de um servidor STUN;
3. Se falhar, o tráfego é dirigido através de um servidor de retransmissão TURN.

Os servidores STUN são responsáveis por obter endereços IP/porto externos de uma rede, enquanto os TURN são responsáveis por retransmitir o tráfego se a conexão direta entre dispositivos falhar.

No que diz respeito ao servidor STUN, as NATs oferecem a possibilidade de comunicação de um endereço IP interno de uma rede local com o exterior, através da sua “tradução”. No cenário de comunicação direta entre dispositivos, não existe a possibilidade de um endereço IP de uma rede privada trabalhar como endereço IP público. Para isso, o WebRTC recorre aos servidores STUN. Estes servidores STUN podem encontrar-se alojados na Internet pública e as aplicações WebRTC utilizam este tipo de servidor para descobrir o “seu” endereço IP/porto público. Através da descoberta deste endereço IP/porto público, as aplicações WebRTC têm agora a possibilidade de transmitir esse endereço público ao “colega” da transmissão. A troca destes dados entre transmissor e recetor do conteúdo multimédia é feita através do mecanismo de *signaling* e o resultado dela é o estabelecimento de uma ligação direta entre ambos. Para complementar a transmissão deste conteúdo multimédia entre dispositivos, é necessário a utilização de servidores TURN para retransmitir o tráfego entre dispositivos. O servidor TURN é considerado um protocolo de retransmissão de tráfego de rede que auxilia a travessia de conteúdo de dados através de NATs e *firewalls*. Por outras palavras, com estes dois tipos de servidores, as aplicações WebRTC são capazes de estabelecer comunicações e transmitir fluxos de vídeo em tempo real entre si, sem quaisquer restrições [33]. Na configuração do servidor STUN/TURN, apenas foi alterado o endereço IP externo, o porto que se encontra à escuta para o servidor STUN, quais os portos mínimos e máximos para retransmissão do servidor TURN e qual a conta de utilizador da máquina a usar nesta configuração. Esta configuração pode ser feita com base no seguinte:



```
# This server's external/public address, if Coturn is behind a NAT. It must be an IP address, not a domain name.  
external-ip=<CoturnIp>  
  
# STUN listener port for UDP and TCP. Default: 3478.  
listening-port=<CoturnPort>  
  
# TURN lower and upper bounds of the UDP relay ports. Default: 49152, 65535.  
min-port=49152  
max-port=65535  
  
# Uncomment to run server in 'normal' 'moderate' verbose mode. Default: verbose mode OFF.  
verbose  
  
# 'Static' user accounts for long-term credentials mechanism.  
user=<TurnUser>:<TurnPassword>
```

O servidor multimédia WebRTC Kurento é construído sobre a livraria GStreamer e é responsável pela transmissão, processamento e reprodução de conteúdo multimédia. Este servidor WebRTC fornece recursos como: protocolos de *streaming* (incluindo HTTP, RTP e WebRTC), comunicações de grupo, gravação de conteúdo nos formatos WebM e MP4 e codificação automática nos formatos VP8, H.264, H.263, AMR, OPUS, Speex, G.711, e outros. No caso da aplicação “IoTUnifiedManagement”, o servidor multimédia Kurento foi executado a partir de uma solução baseada numa imagem Docker disponibilizada pela plataforma Kurento. Esta imagem contém algumas alterações na configuração, nomeadamente na indicação dos endereços IP/porto de configuração do servidor STUN/TURN [34]. Estas configurações de endereço IP e porto do servidor STUN/TURN podem ser fornecidas pelas variáveis `KMS_STUN_IP`, `KMS_STUN_PORT` e `KMS_TURN_URL` na configuração do servidor multimédia Kurento.

Por último, a arquitetura de transmissão de fluxos de vídeo em tempo real de **1 para muitos** (*One to Many video call*) é suportada pela aplicação web baseada em Node.js. Esta aplicação web está atualmente em execução no servidor Media-Server-Helper e consiste numa aplicação para *broadcast* de vídeo que trabalha para auxiliar clientes web (como a nossa aplicação “IoTUnifiedManagement”) a difundir o seu conteúdo multimédia em conjunto com o servidor multimédia WebRTC Kurento. Deste modo, para que seja possível comunicar e transmitir vídeo em tempo real, o servidor Media-Server-Helper auxilia os clientes web com as seguintes funcionalidades: gestão de mensagens WebSocket, gestão da comunicação com o servidor multimédia Kurento (criando uma conexão com o servidor *WebRtcEndpoint* Kurento) e envio de conteúdo para o servidor multimédia Kurento. Uma vez acionados os mecanismos corretos para a transmissão de vídeo, este servidor é responsável por difundir o vídeo em tempo real até aos seus clientes [35]. Esta configuração pode ser feita com base no seguinte:

```
var ws = require('ws');
var wss = new ws.Server({
  server : server,
  path : '/one2many'
});
var candidatesQueue = {};
var argv = minimist(process.argv.slice(2), {
  default: {
    as_uri: 'https://localhost:8443/',
    ws_uri: 'ws://localhost:8888/kurento'
  }
});
/*
 * Management of WebSocket messages
 */
wss.on('connection', function(ws) {

  var sessionId = nextUniqueId();
  console.log('Connection received with sessionId ' + sessionId);

  ws.on('error', function(error) {
    [...]
  });

  ws.on('close', function() {
    [...]
  });

  ws.on('message', function(_message) {
    var message = JSON.parse(_message);
    console.log('Connection ' + sessionId + ' received message ', message);

    switch (message.id) {
      case 'presenter':
        [...]
        break;

      case 'viewer':
        [...]
        break;

      case 'stop':
        [...]
        break;

      case 'onIceCandidate':
        [...]
        break;

      default:
        [...]
        break;
    }
  });
});
```

```
/*
 * Management of client connections
 */

function getKurentoClient(callback) {
  if (kurentoClient !== null) {
    return callback(null, kurentoClient);
  }

  kurento(argv.ws_uri, function(error, _kurentoClient) {
    if (error) {
      [...]
    }

    kurentoClient = _kurentoClient;
    callback(null, kurentoClient);
  });
}

function onIceCandidate(sessionId, _candidate) {
  var candidate = kurento.getComplexType('IceCandidate')(_candidate);

  if (presenter && presenter.id === sessionId && presenter.webRtcEndpoint) {
    [...]
  }
  else if (viewers[sessionId] && viewers[sessionId].webRtcEndpoint) {
    [...]
  }
  else {
    [...]
  }
}

function clearCandidatesQueue(sessionId) {
  if (candidatesQueue[sessionId]) {
    [...]
  }
}
```

```

/*
 * Send media content
 */
function startPresenter(sessionId, ws, sdpOffer, callback) {
  clearCandidatesQueue(sessionId);

  if (presenter !== null) {
    stop(sessionId);
    return callback("Another user is currently acting as presenter. Try again later ...");
  }

  presenter = {
    id : sessionId,
    pipeline : null,
    webRtcEndpoint : null
  }

  getKurentoClient(function(error, kurentoClient) {
    [...]
  });
});
}

```

Para além destas funcionalidades base, foi ainda adicionada a funcionalidade de gravação do conteúdo multimédia a ser partilhada pelo perfil professor. Esta é adicionada ao servidor Media-Server-Helper para que este seja responsável por “indicar” ao servidor multimédia Kurento para retornar o conteúdo de vídeo fornecido para os clientes finais e efetuar a gravação desse conteúdo de vídeo, ao mesmo tempo. Esta gravação de conteúdo só acontece quando o perfil professor indica que uma aula será gravada e esta é iniciada e/ou terminada enquanto uma se encontra a decorrer. Como o perfil professor é o único fornecedor de conteúdo multimédia, apenas esse conteúdo será gravado incluindo a gravação do áudio e vídeo fornecido. No final de cada conteúdo gravado, este servidor Media-Server-Helper é responsável por armazenar todo o conteúdo gravado para um ficheiro e, por conseguinte, submeter um novo objeto de dados do tipo “material de suporte” (contendo todos os valores dos atributos necessários) no *middleware* FIWARE. Assim, esta é vivenciada pelo seu nome, uma vez que este é composto por uma compilação que contém o nome da aula que foi lecionada e a data de gravação do conteúdo. Esta funcionalidade foi adicionada para que os utilizadores (alunos) da aplicação 5GOpenclasses pudessem visualizar o conteúdo de uma vídeo-aula depois de esta ser lecionada [36]. Esta configuração foi feita com base no seguinte:

```

function startRecording() {
  var options = {
    #record only video from presenter
    localVideo: videoInput,
  };

  webRtcPeer = kurentoUtils.WebRtcPeer.WebRtcPeerSendrecv(options, function(error) {
    if(error) return onError(error)
    this.generateOffer(onOffer)
  });
  [...]
}

```

Quanto à aplicação móvel Android 5GOpenclasses, todas as funcionalidades indicadas anteriormente são suportadas pelo *middleware* FIWARE e pela livraria de suporte WebRTC **nc-kurento-android** [37]. Relativamente às funcionalidades de registo e login disponibilizadas pela aplicação Android 5GOpenclasses, estas têm uma vista dedicada para cada uma delas, possibilitando aos utilizadores de estes introduzirem os dados necessários e usufruírem da aplicação. Após os utilizadores se encontrarem registados e dentro da aplicação, estes têm acesso a uma determinada lista de vídeo-aulas disponíveis naquele momento para visualização de conteúdo multimédia em tempo real. Adicionalmente, os alunos também têm uma vista dedicada para a visualização de todo um catálogo de materiais de suporte disponibilizado, assim como uma vista dedicada para a conversação entre os vários utilizadores (alunos) da aplicação móvel Android. Por último, e mais importante ainda, esta aplicação Android disponibiliza uma vista dedicada para a visualização de conteúdo multimédia em tempo real, suportado pela livraria **nc-kurento-android**. Livraria esta que possibilita, não só ligação e comunicação direta com o nosso servidor multimédia WebRTC Kurento (incluindo a ultrapassagem das adversidades das NATs e *firewalls* com o servidor STUN/TURN), como também a transmissão do conteúdo multimédia em tempo real proveniente do nosso servidor multimédia WebRTC Kurento. Tal como já foi descrito anteriormente, esta aplicação Android apenas será responsável por apresentar o conteúdo de vídeo em tempo real proveniente do perfil professor, utilizando uma SurfaceView para reproduzir o vídeo. Para além desta vista dedicada aos fluxos de vídeo em tempo real provenientes do nosso servidor WebRTC, dedicados exclusivamente à aplicação 5GOpenclasses, esta aplicação Android é também portadora de outras funcionalidades para apresentação de vídeo em tempo real utilizando um servidor WebRTC Janus. Estas funcionalidades dedicadas à apresentação de fluxos de vídeo provenientes de um servidor WebRTC foram implementadas para o projeto à qual a aplicação pertence (projeto Mobilizador 5G) e estes têm como objetivo reproduzir fluxos de vídeo provenientes de câmaras de drones e de um *bodykit* de uma corporação de bombeiros. Inicialmente o nosso objetivo era também utilizar uma SurfaceView para reproduzir o vídeo em tempo real proveniente do servidor WebRTC Janus. No entanto houve alguns contratemplos que levaram à utilização de uma WebView para reprodução destes fluxos de vídeo. Esta WebView permite que sejam exibidos conteúdos tal como se fossem uma página web, sendo que, ao utilizarmos este método, não tivemos qualquer problema na reprodução dos fluxos de vídeo provenientes do servidor WebRTC Janus.

## 4.2 Outras aplicações

As implementações elaboradas noutras aplicações, nomeadamente na aplicação BATINA, podem ser descritas maioritariamente pelas seguintes categorias: funcionalidades de *back-end* iniciais, funcionalidades de registo e autenticação de utilizadores via OAuth2.0, funcionalidades de gestão de disciplinas e dos seus materiais de suporte, funcionalidades de visualização de questionários a decorrer em tempo real, funcionalidades de apresentação de relatórios de resultados dos alunos e funcionalidades de impressão desses relatórios de resultados. Todas elas foram desenvolvidas em prol da aplicação *web-based*, tendo em consideração que esta foi desenvolvida sobre a *framework* Dash by Plotly, e apenas o perfil professor desta aplicação poderá utilizar esta versão *web-based*.

No que diz respeito às funcionalidades de *back-end* iniciais, estas foram compostas pela definição da estrutura dos objetos de dados a ser utilizada e armazenada pelo *middleware* FIWARE. A estrutura de dados que atualmente suporta esta aplicação é a seguinte:

Tabela IV – Estrutura de dados BATINA.

Entidades	Atributos
Professor	<i>Id, type, courses, email, picture, username, create_time e update_time</i>
Estudante	<i>Id, type, courses, email, picture, username, questionnaire_live, create_time e update_time</i>
Disciplina	<i>Id, type, name, description, professor_id, academic_year, questionnaires, n_questionnaires, support_materials, picture, create_time e update_time</i>
Ficheiro de suporte	<i>Id, type, parent_dir, file_name, download_path, create_time e update_time</i>
Questionário	<i>Id, type, academic_year, description, professor_id, duration, release_time, supp_materials, course, picture, create_time e update_time</i>
Questões	<i>Id, type, question, duration, supp_materials, options, answer, picture, create_time e update_time</i>
Respostas	<i>Id, type, answer, course_id, question_id, questionnair_id, student_id, is_correct, create_time e update_time</i>
Relatório de respostas	<i>Id, type, answers, correct_answers, course_id, questionnaire_id, student_id, user_notified, create_time e update_time</i>

A implementação de funcionalidades de registo e autenticação de utilizadores via OAuth2.0 encontra-se suportada pelo gestor de identidades KEYROCK [24]. Este gestor de entidades é um componente do FIWARE e encontra-se diretamente conectado com este. Para além deste ser um gestor de entidades, o KEYROCK também é responsável por fornecer uma autenticação OAuth2.0 à nossa aplicação BATINA (tanto para a versão *web-based* como também para ambas as aplicações móveis Android e iOS). No KEYROCK é possível gerir organizações e aplicações. Cada organização poderá ter um determinado número de utilizadores autorizados a geri-la, assim como poderá ter uma ou mais aplicações associadas a ela. Dentro de cada aplicação é possível agrupar um determinado número de utilizadores. Este grupo de utilizadores irá partilhar regras e permissões nesta aplicação, ou seja, se adicionarmos *n* utilizadores a um determinado grupo, todas as regras e permissões destinadas a esse grupo serão automaticamente aplicadas a todos os utilizadores. O acesso de dados está atualmente implementado através da utilização de *tokens* no *middleware* FIWARE, sendo estes do tipo *Bearer tokens*. Sempre que um utilizador efetua o login, quer na aplicação *web-based*, quer nas aplicações móveis, é atribuído um *Bearer token* a esse utilizador, com um tempo de vida de uma hora para o *access token* e um tempo de vida de duas semanas para o *refresh token*. Tanto o *access token* como o *refresh token* estão associados a um utilizador e sempre que necessário, este utilizador pode aceder aos dados armazenados no FIWARE utilizando o *access token*. Se esse *access token* ainda for válido (ou seja, se ainda tiver tempo de vida) e se as regras e permissões destinadas a ele o permitirem, este utilizador poderá efetuar pedidos e receber informações do FIWARE sem qualquer problema. Sempre

que necessário, o *access token* destinado a este utilizador poderá ser atualizado através do *refresh token*.

Em relação às funcionalidades de gestão de disciplinas, o professor tem a possibilidade de adicionar novas disciplinas, visualizar, editar e eliminar disciplinas existentes, como listar todas as suas disciplinas. Neste último aspeto, o professor tem a possibilidade de ordenar essa listagem por nome, ano académico, número de questionários, data de criação e data de atualização pela forma crescente ou decrescente. Para além disso, esta listagem encontra-se inserida numa tabela com paginação e o professor tem a possibilidade de editar o número de disciplinas apresentadas nessa tabela, como por exemplo, 10, 20, 30, 40, 50 ou 100 disciplinas por página. Relativamente à adição de uma nova disciplina, o professor tem a possibilidade de introduzir dados num formulário e esse formulário tem campos como: nome da disciplina, ano académico da disciplina, descrição da disciplina, fotografia da disciplina. Neste formulário, o professor terá a possibilidade de adicionar novos materiais de suporte à disciplina, embora esta não seja uma atividade obrigatória. Por outro lado, o professor também terá a possibilidade de visualizar, editar e eliminar informações de uma determinada disciplina, incluindo todas as suas informações básicas e todos os materiais de suporte. Na edição destes materiais de suporte, este tem a possibilidade de eliminar ou fazer download dos materiais de suporte, selecionando um ou mais itens da lista de materiais. No caso de o professor eliminar uma disciplina, todos os materiais de suporte, questionários e relatórios de respostas a questionários serão automaticamente eliminados.

No que diz respeito à gestão de materiais de suporte, o professor terá a possibilidade de adicionar novos materiais de suporte à aplicação e associá-los qualquer das suas disciplinas. Na listagem de materiais de suporte, este consegue visualizar todos os ficheiros associados às suas disciplinas, podendo ordená-los por nome do ficheiro, data de criação e data de atualização pela forma crescente ou decrescente. Nesta listagem é também possível editar o número de materiais de suporte a apresentar na tabela, podendo estes mais uma vez serem apresentados por 10, 20, 30, 40, 50 ou 100 materiais por página. Nesta gestão de materiais de suporte, o professor tem ainda a possibilidade de efetuar *download* ou eliminar múltiplos ficheiros de suporte associados às suas disciplinas.

Quanto aos questionários e, sendo este um dos pontos fulcrais da aplicação, o professor tem a possibilidade de adicionar um novo questionário, visualizar informações de questionários, visualizar o acontecimento dos questionários ao vivo e visualizar os relatórios de respostas dos questionários. Ao adicionar um novo questionário, este tem a obrigatoriedade de preencher um formulário com campos obrigatórios como: nome do questionário, curso destinado ao questionário, descrição desse questionário, data de lançamento do questionário (incluindo a hora), indicar os materiais de suporte fulcrais para o questionário (incluindo as páginas fulcrais desse material), o tempo total que os estudantes terão para responder ao questionário e as respetivas questões. Nas questões, o professor tem a possibilidade de adicionar uma imagem a cada delas, indicar a respetiva questão, as opções de resposta dessa questão e qual das opções é a resposta à questão (pode selecionar mais do que uma opção). Na visualização dos questionários, este terá a possibilidade de observar todas as informações já anteriormente introduzidas incluindo questões e respostas a questões. Os questionários ao vivo servem para o professor analisar as respostas provenientes dos alunos às questões de um questionário ao vivo. Estes questionários são dedicados apenas a um questionário e às respostas das suas questões. Dentro desta vista é possível visualizar o número total de respostas para cada opção de uma questão, é possível

visualizar quais das opções de resposta são as corretas e também é possível visualizar a percentagem de sucesso para essa determinada questão.

Em relação aos relatórios de resultados dos questionários, estes podem englobar respostas para apenas um questionário ou para os vários questionários de uma disciplina. Ou seja, podemos visualizar o relatório de respostas de um questionário específico ou visualizar todas as respostas de todos os questionários de uma disciplina. Dentro dos relatórios de resultados de um questionário é possível observar informações em gráficos como a percentagem de respostas corretas submetidas pelos alunos, um somatório de classificação dos estudantes entre as diferentes percentagens de sucesso dos estudantes (percentagens entre 0%-20%, 20%-40%, 40%-60%, 60%-80% e 80%-100%) e uma percentagem respostas submetidas ao respetivo questionário (divisão entre o número total de respostas submetidas sobre o número total de respostas esperadas). Para além das informações apresentadas nestes gráficos, também são apresentados componentes de informação que indicam o número total de notas positivas, o número total de notas negativas, o número total de respostas submetidas sobre o número total de respostas esperadas e o número total de estudantes que responderam ao questionário e o número total de estudantes que era esperado responderem ao questionário. Por último, nestes relatórios de respostas a um questionário também são apresentadas duas listas de dados, uma correspondente à classificação para cada questão do questionário e outra correspondente às classificações dos alunos neste questionário. Na listagem da classificação para cada questão do questionário é apresentada a questão em si, qual o número total de respostas corretas a essa questão, qual o número total de respostas e qual a taxa de sucesso para cada questão. Na listagem das classificações dos alunos para o questionário, são apresentadas informações relacionadas ao email do aluno, assim como a sua percentagem de sucesso do aluno nesse questionário.

No que diz respeito aos relatórios de resultados para uma disciplina, este irá incluir todas as respostas de todos os questionários pertencentes a uma determinada disciplina, caso existam questionários e respostas. Dentro destes relatórios serão apresentadas informações em gráficos como a percentagem da média da taxa de sucesso de cada estudante pertencente à disciplina, o somatório da média da taxa de sucesso para cada estudante da disciplina (percentagens entre 0%-20%, 20%-40%, 40%-60%, 60%-80% e 80%-100%). Também a percentagem de respostas submetidas aos diferentes questionários da disciplina (divisão entre o número total de respostas submetidas sobre o número total de respostas esperadas) são apresentadas. Para além disso, os componentes de informação que indicam o número total de notas positivas e negativas, o número total de respostas e o número total esperado de respostas para todos os questionários são apresentados. Em conformidade com o anterior, o relatório de respostas a questionários de uma disciplina também é composto por uma listagem de sucesso dos diferentes questionários desta disciplina e por uma listagem de sucesso dos diferentes alunos pertencentes a esta disciplina. Na listagem de sucesso dos questionários é apresentada a informação do nome dos diferentes questionários, qual o número total de respostas certas para esse questionário, qual o total de respostas submetidas, qual o total de questões desse questionário e qual a taxa de sucesso nesse questionário. Na listagem de sucesso dos alunos nesta disciplina são apresentadas informações relativas aos diferentes alunos que submeteram respostas a questionários, qual a média de sucesso obtida para cada um nos vários questionários respondidos e qual o número total de questionários que um aluno já respondeu.

Por fim, relativamente às funcionalidades de impressão dos relatórios de resultados, esta ainda é uma funcionalidade recente e não muito evoluída. Deste modo, encontra-se disponível para ser utilizada tanto nos relatórios de resultados para um questionário, como



## Capítulo 4

nos relatórios de resultados para uma disciplina e pode ser acionada apenas com um clique, sendo que este funciona como se o utilizador estivesse a fazer impressão da página web do *browser* (Ctrl+P).



# Capítulo 5

## Demonstração e aplicação em projetos

Quanto à demonstração do trabalho elaborado ao longo desta dissertação de mestrado, este será dividido entre as implementações do *dashboard* “IoTUnifiedManagement” e outras aplicações, nomeadamente a aplicação BATINA.

### 5.1 IoTUnifiedManagement

O *dashboard* “IoTUnifiedManagement” encontra-se atualmente dividido entre dois diferentes perfis de utilizadores: **administrador** e **professor**. Para estes diferentes tipos de utilizadores, serão apresentadas as diferentes páginas web dedicadas a cada um deles.

Todas as implementações elaboradas na aplicação “IoTUnifiedManagement” encontram-se explícitas no seguinte vídeo: <https://bit.ly/393FR5c>. Este foi elaborado para apresentação das funcionalidades dedicadas ao 5GOpenclasses. No entanto, incorpora a apresentação de todas as funcionalidades que a aplicação “IoTUnifiedManagement” disponibiliza aos seus utilizadores. Alguns componentes apresentados neste vídeo encontram-se atualizados e, atualmente, a aplicação “IoTUnifiedManagement” encontra-se exatamente como é apresentado nas figuras seguintes.

As páginas dedicadas ao login e registo são as seguintes:

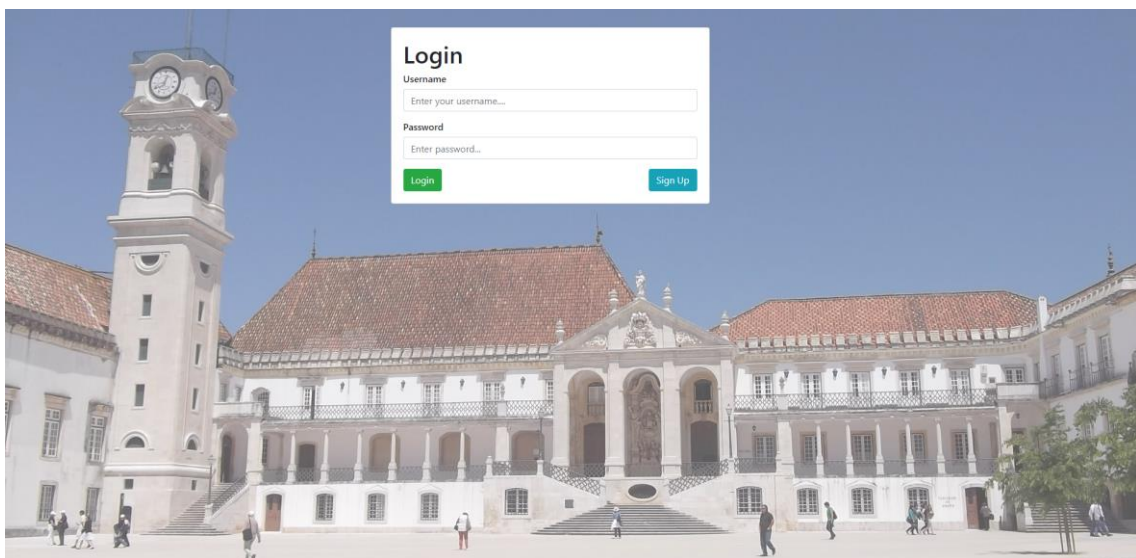


Figura 36 – Página de login “IoTUnifiedManagement”.

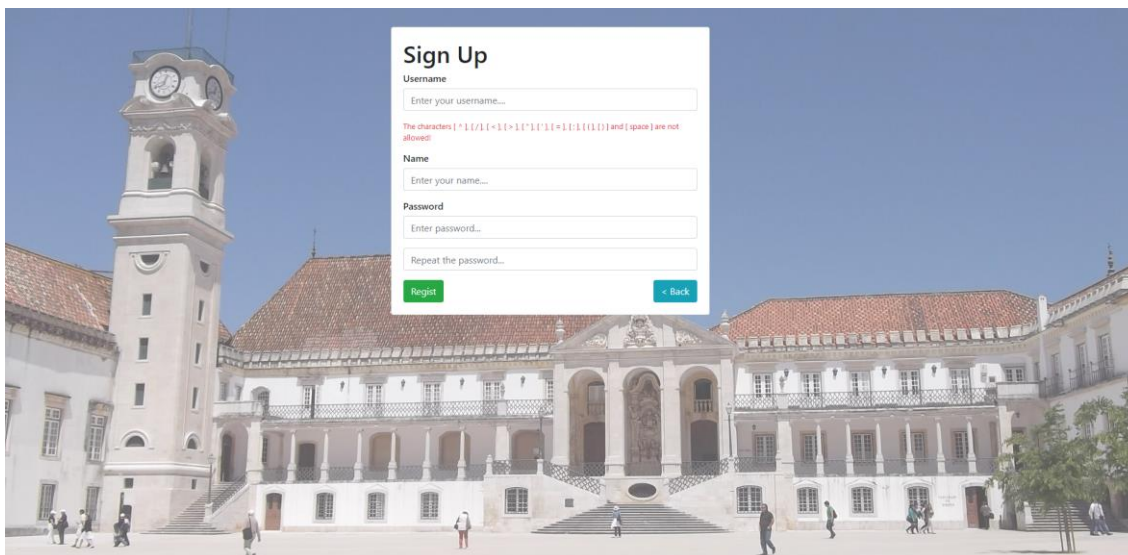


Figura 37 – Página de registo “IoTUnifiedManagement”.

### 5.1.1 Administrador

Dentro do perfil de administrador, existem várias secções onde são apresentadas informações. Nas Figuras 38, 39 e 40 estão apresentadas as informações relacionadas com a secção de “Gestão”. Nestas figuras é possível visualizar informações dos serviços do *middleware* FIWARE e visualizar informações dos componentes do servidor. Na Figura 38 não foram recuperados com sucesso os dados sobre o tempo de resposta dos serviços e os estados dos serviços do FIWARE numa resolução temporal de 24 horas.



Figura 38 – Página do administrador (1) “IoTUnifiedManagement”.



Figura 39 – Página do administrador (2) “IoTUnifiedManagement”.



Figura 40 – Página do administrador (3) “IoTUnifiedManagement”.

Já nas Figuras 41 a 50 é possível observar as páginas web dedicadas à gestão dos sensores IoT. Nas Figuras 41 e 42 são apresentados os dados recolhidos dos sensores IoT das boxes da plataforma e as informações relativas aos dados dos sensores IoT recolhidos pela aplicação móvel 5GOpenclasses e ISABELA. Nas restantes figuras, é possível visualizar os valores dos respetivos dados IoT numa resolução de 24 horas (com início na data da última recolha de dados) e a listagem dos resultados apresentados no gráfico acima.

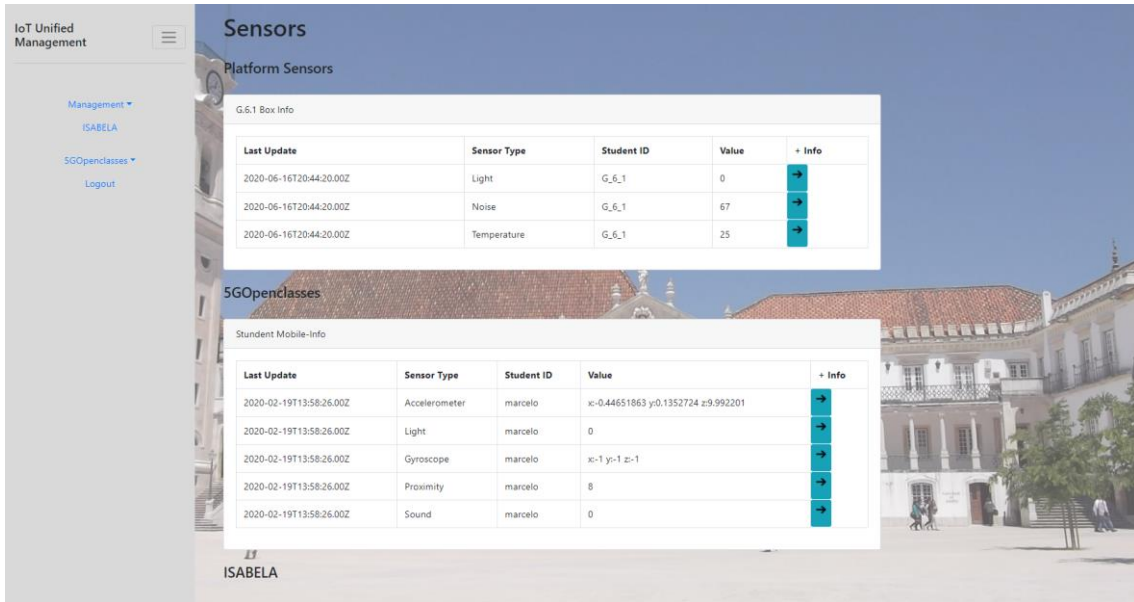


Figura 41 – Página do administrador (4) “IoTUnifiedManagement”.

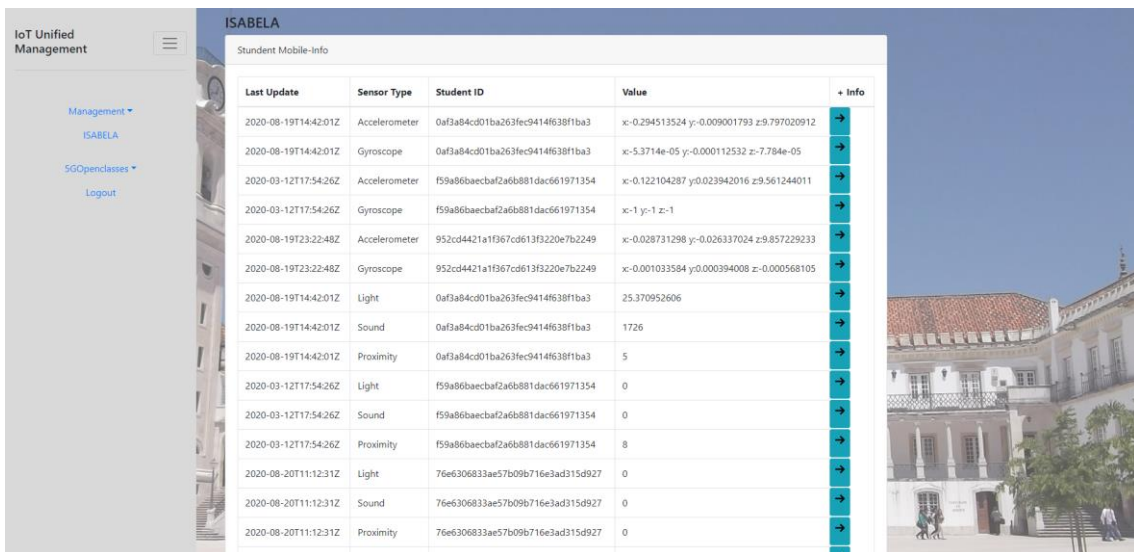


Figura 42 – Página do administrador (4) “IoTUnifiedManagement”.



Figura 43 – Página do administrador (5) “IoTUnifiedManagement”.



Figura 44 – Página do administrador (6) “IoTUnifiedManagement”.





Figura 45 – Página do administrador (7) “IoTUnifiedManagement”.



Figura 46 – Página do administrador (8) “IoTUnifiedManagement”.





Figura 47 – Página do administrador (9) “IoTUnifiedManagement”.

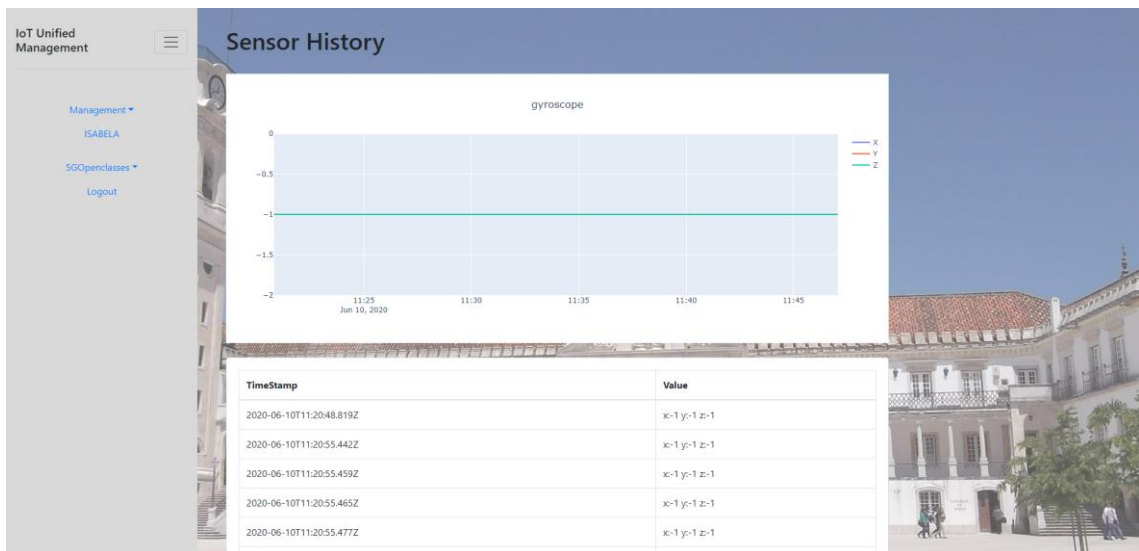


Figura 48 – Página do administrador (10) “IoTUnifiedManagement”.

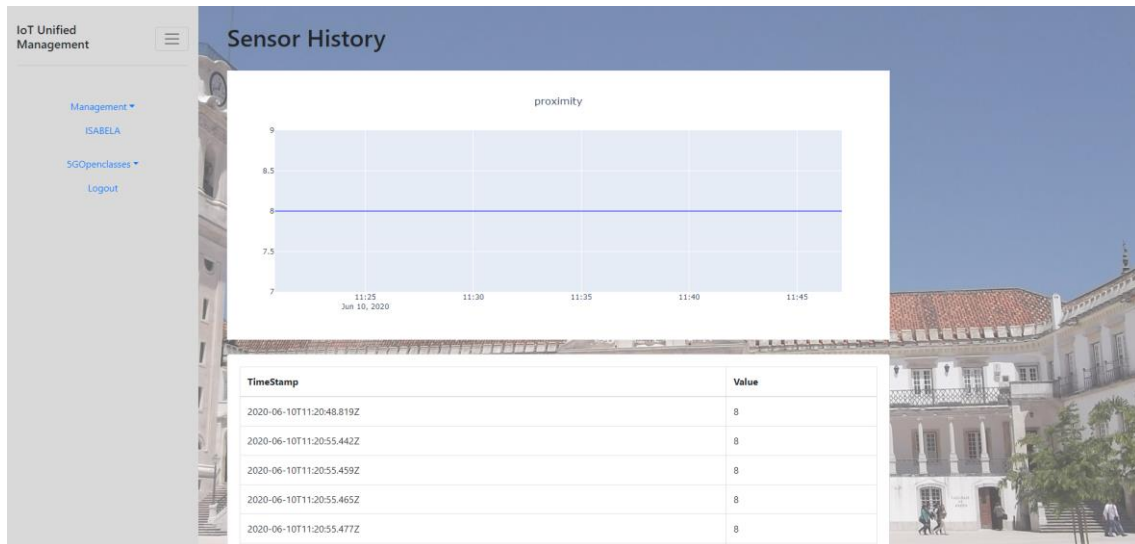


Figura 49 – Página do administrador (11) “IoTUnifiedManagement”.



Figura 50 – Página do administrador (12) “IoTUnifiedManagement”.

Na secção “ISABELA” não foram implementadas quaisquer funcionalidades e o resultado encontra-se apresentado na Figura 51.



Figura 51 – Página do administrador (13) “IoTUnifiedManagement”.

Na secção do “5GOpenclasses”, o resultado das implementações nesta secção é apresentado nas Figuras 52 e 56, onde é possível visualizar informações acerca das salas de aulas que se encontram disponíveis neste momento, as disciplinas existentes, os materiais de suporte existentes e o feedback da aplicação. Tal como foi descrito anteriormente, o perfil administrador é responsável por adicionar novas disciplinas aos perfis professor existentes na aplicação e essa funcionalidade é apresentada na Figura 53.

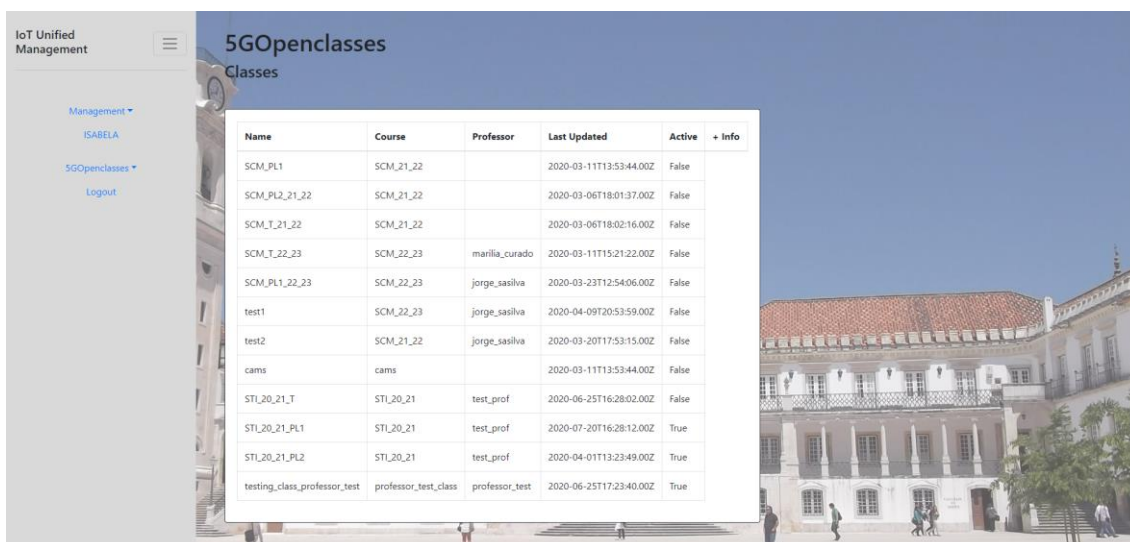


Figura 52 – Página do administrador (14) “IoTUnifiedManagement”.

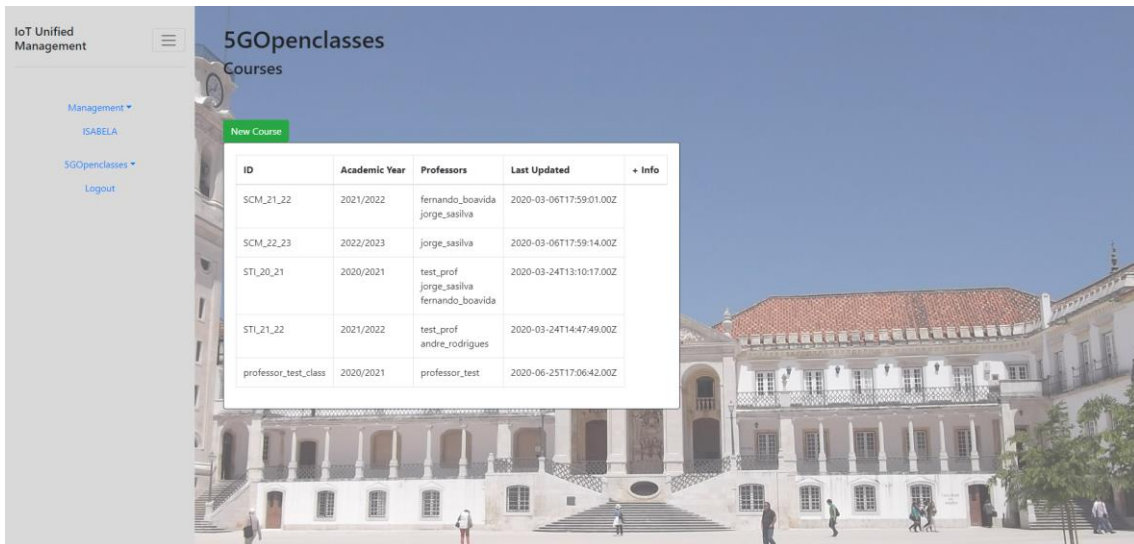


Figura 53 – Página do administrador (15) “IoTUnifiedManagement”.

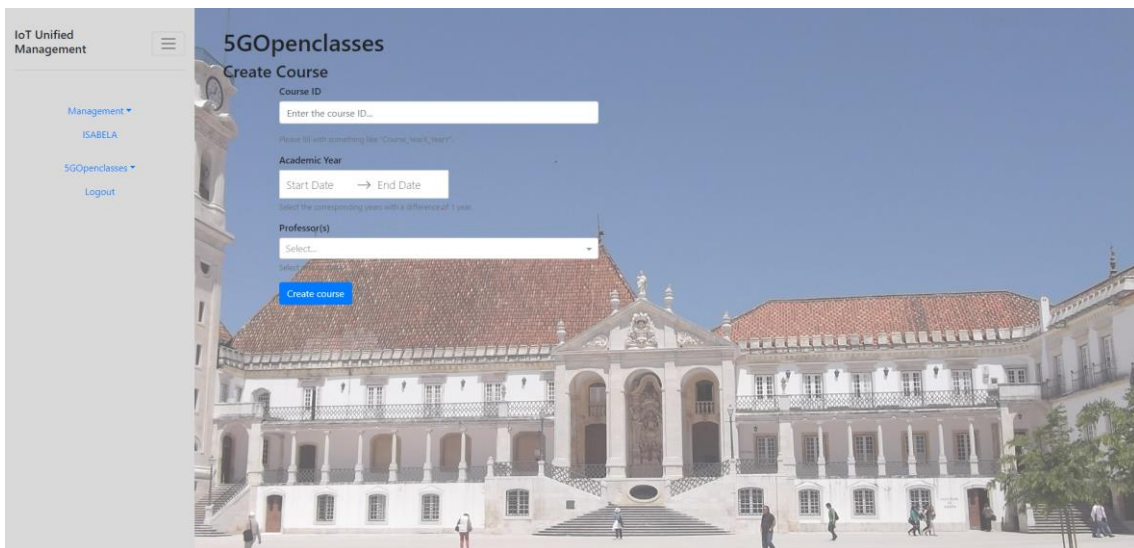


Figura 54 – Página do administrador (16) “IoTUnifiedManagement”.

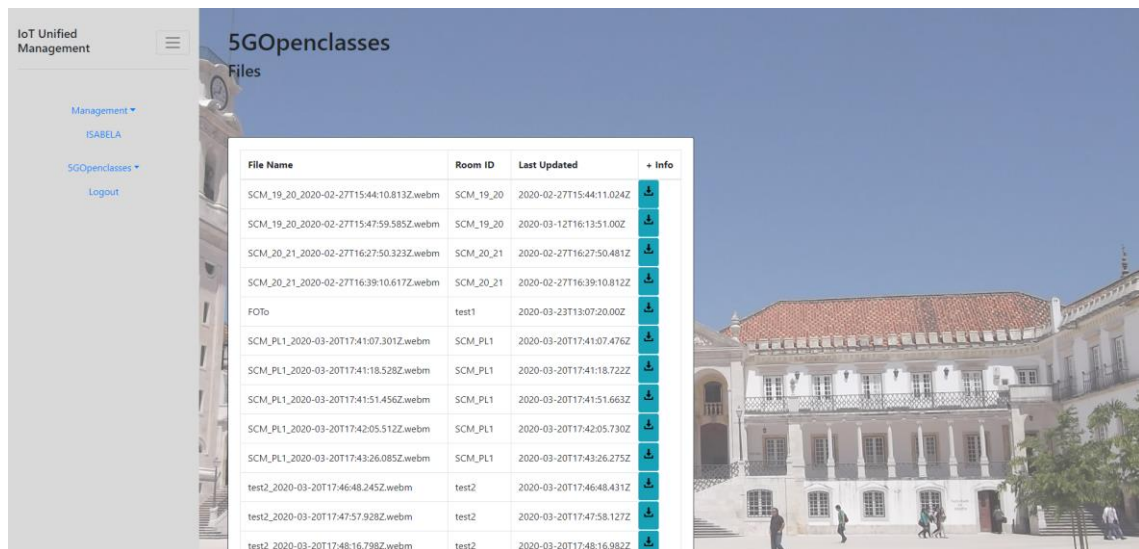


Figura 55 – Página do administrador (17) “IoTUnifiedManagement”.

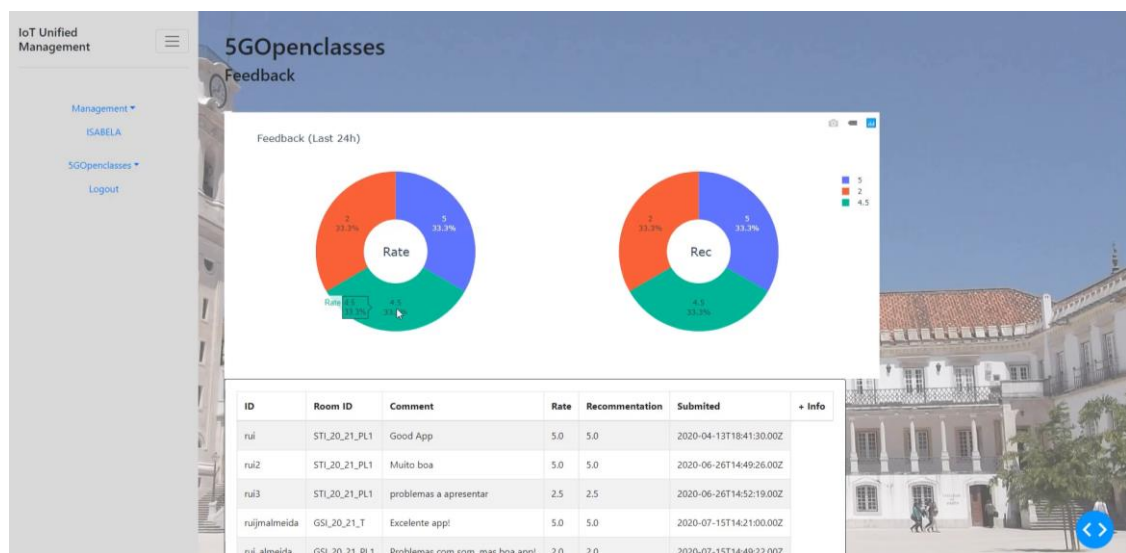


Figura 56 – Página do administrador (18) “IoTUnifiedManagement”.

### 5.1.2 Professor

O perfil professor da aplicação “IoTUnifiedManagement” dedica-se à aplicação 5GOpenclasses. Neste campo, este utilizador terá a possibilidade de visualizar e gerir as suas aulas, visualizar e gerir os materiais de suporte das suas disciplinas e usufruir das funcionalidades de vídeo-aulas em tempo real. As funcionalidades de vídeo de aulas são apresentadas na Figura 58, onde o professor é capaz de iniciar uma transmissão de vídeo selecionando transmissão da *webcam* do seu computador ou mesmo a gravação do seu ecrã. Finalmente, este terá apenas de desligar a transmissão e encerrar a sua aula. Para além disso, nesta página web também é possível indicar se a transmissão da aula será gravada ou não para um ficheiro do tipo *webm*. A Figura 61 representa a página web dedicada à edição de um material de suporte.



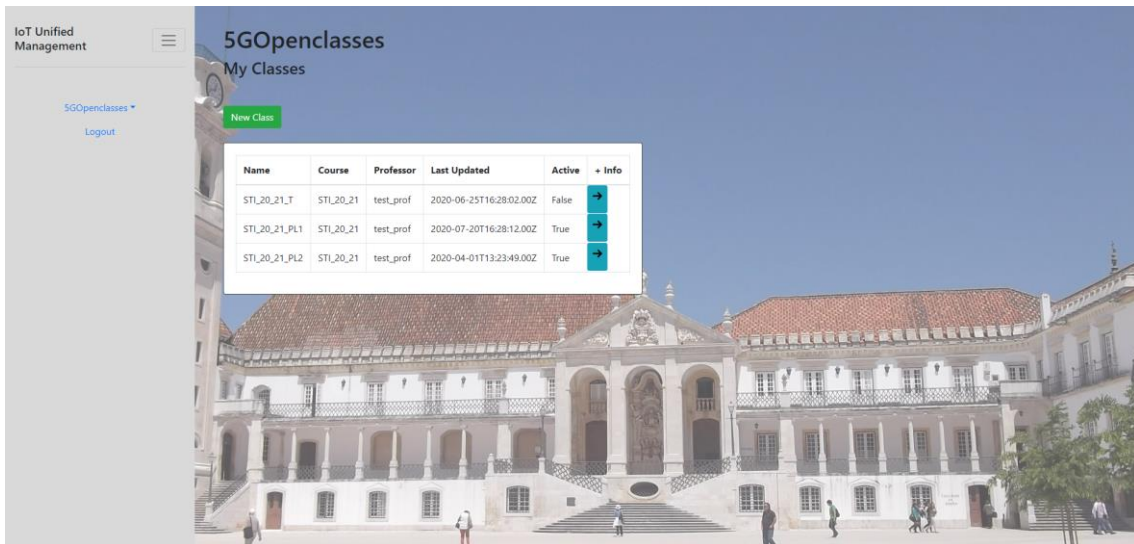


Figura 57 – Página do professor (1) “IoTUnifiedManagement”.

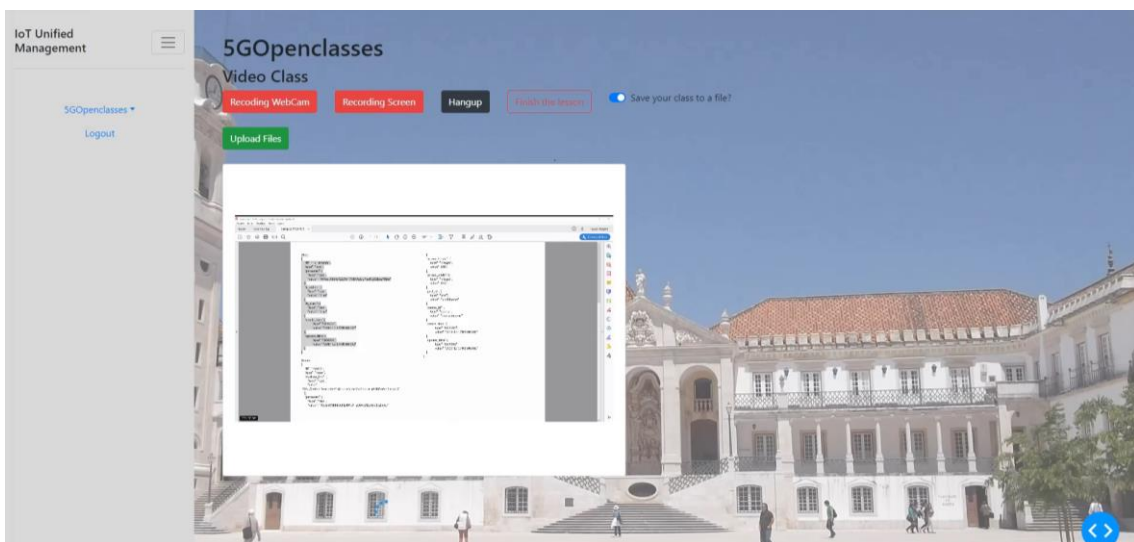


Figura 58 – Página do professor (2) “IoTUnifiedManagement”.

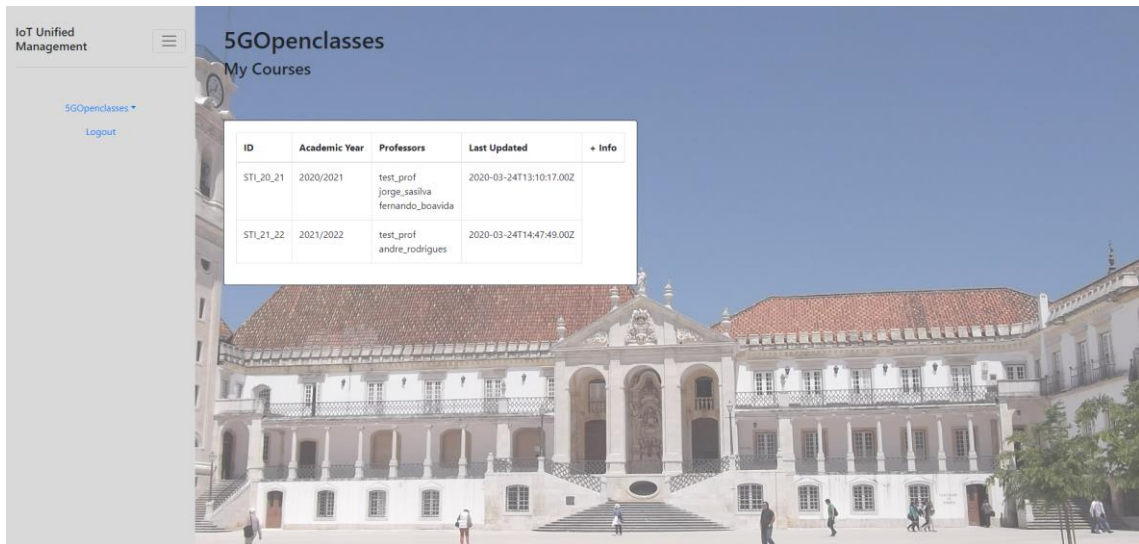


Figura 59 – Página do professor (3) “IoTUnifiedManagement”.

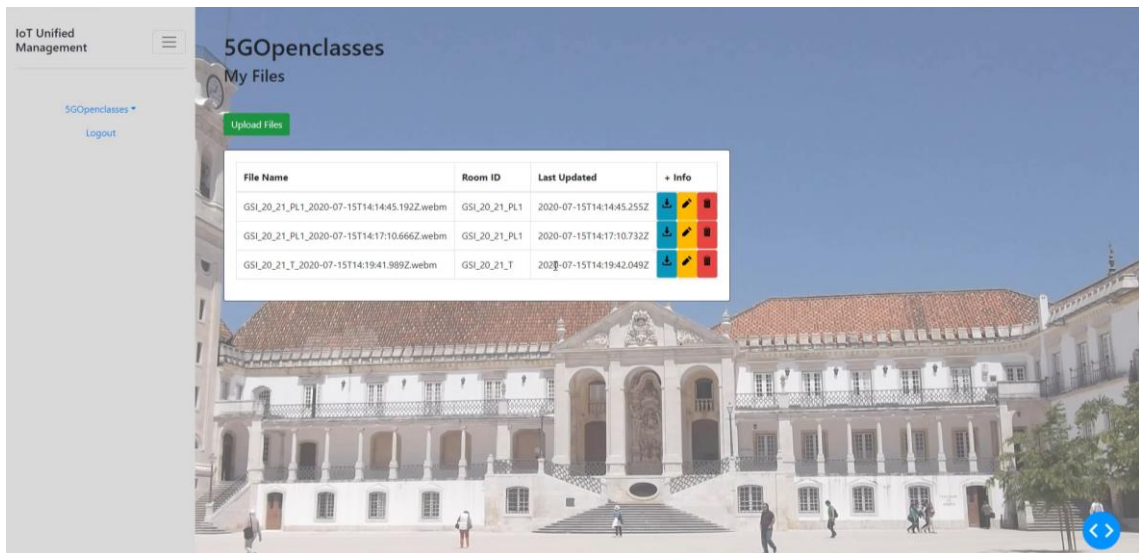


Figura 60 – Página do professor (4) “IoTUnifiedManagement”.



Figura 61 – Página do professor (5) “IoTUnifiedManagement”.

Tendo em conta que a aplicação “IoTUnifiedManagement” suporta estas funcionalidades dedicadas ao perfil de utilizador **professor**, é também necessário existir uma aplicação que suporte as funcionalidades de visualização de conteúdos multimédia em tempo real. Essas funcionalidades são apresentadas na aplicação móvel Android 5GOpenclasses, sendo esta dedicada ao aluno. Este poderá usufruir das funcionalidades de visualização de conteúdos de vídeo em tempo real, visualização dos materiais de suporte disponibilizados e contribuir com um *feedback* de QoE das vídeo-aulas.

A Figura 62 dedica-se à apresentação da listagem de vídeo-aulas disponíveis para os alunos naquele momento. A vista dedicada a esta visualização de conteúdo multimédia em tempo real é apresentada na Figura 63. As Figuras 64 e 65 são representativas das funcionalidades de resposta a um questionário QoE do utilizador e listagem do repositório de materiais de suporte, respetivamente.



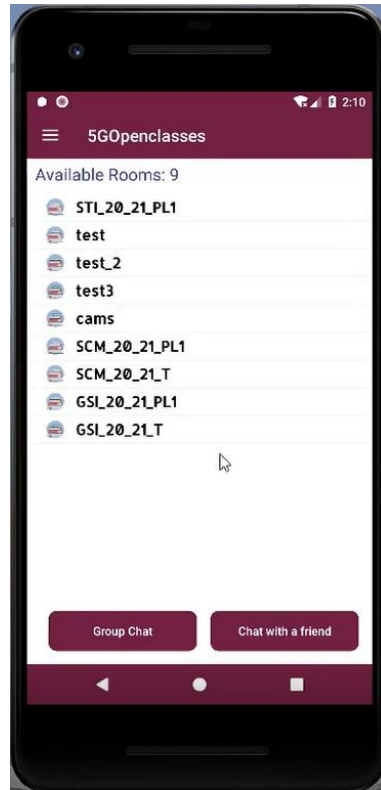


Figura 62 – Página do aluno (1) 5GOpenclasses.

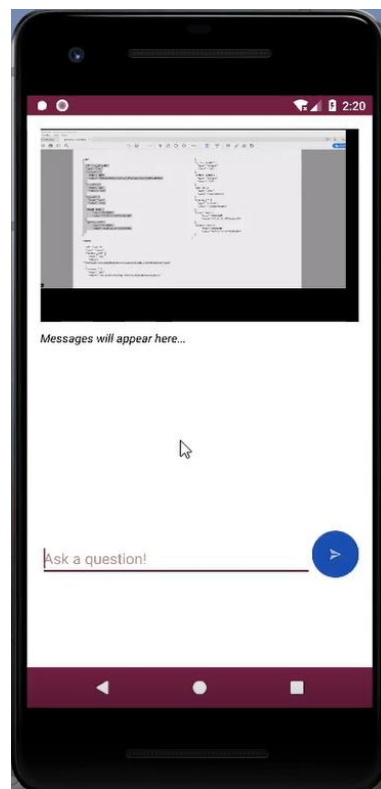


Figura 63 – Página do aluno (2) 5GOpenclasses.

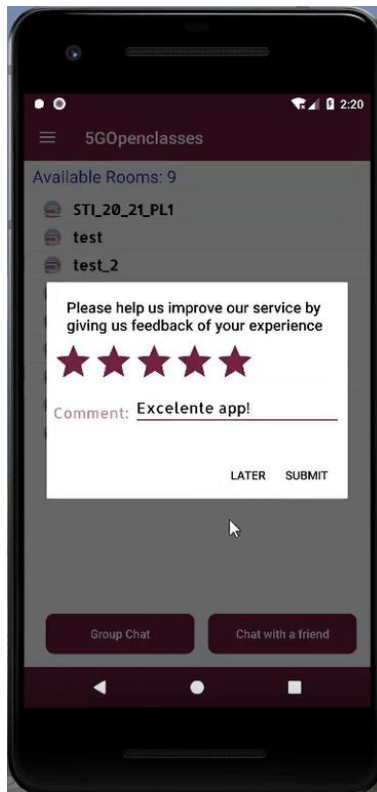


Figura 64 – Página do aluno (3) 5GOpenclasses.

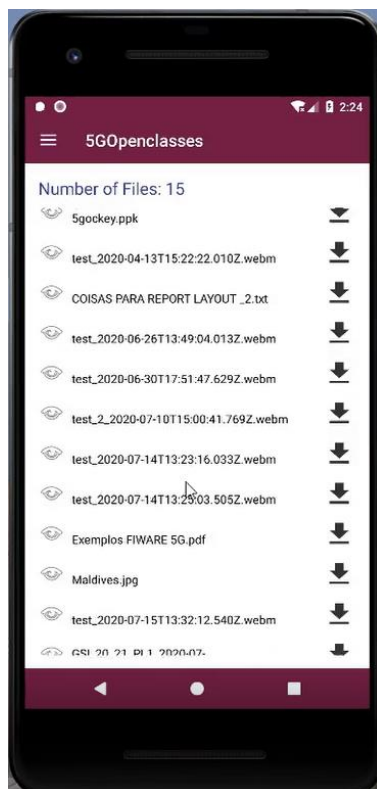


Figura 65 – Página do aluno (4) 5GOpenclasses.

## 5.2 Outras aplicações

As implementações noutras aplicações que não o “IoTUnifiedManagement” foram todas dedicadas à aplicação BATINA. Esta aplicação tem como objetivo a disponibilização de uma plataforma de questionários *online* em tempo real, que poderá ser utilizada tanto num contexto de uma aula presencial como numa aula remota. Relativamente às páginas web dedicadas ao login e registo, estas encontram-se apresentadas nas Figuras 66 e 67.

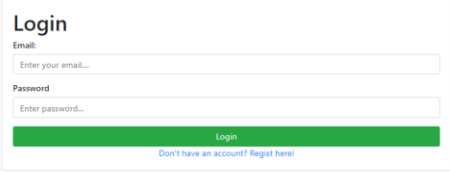


Figura 66 – Página de login BATINA.

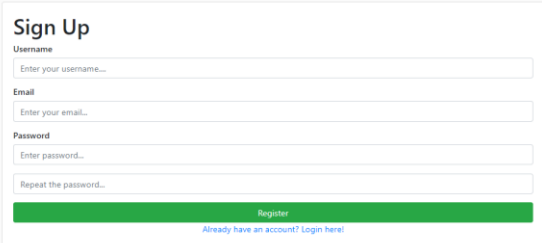


Figura 67 – Página de registo BATINA.

Após o registo e login, os utilizadores (professores) serão diretamente redirecionados para a sua lista de disciplinas. Num primeiro login, esta vista não terá nenhuma disciplina apresentadas, mas este terá a possibilidade de adicionar uma nova logo que possível. Na parte superior da listagem apresentada na Figura 68, é possível observar um botão dedicado á adição de uma nova disciplina, assim como botões dedicados ao ordenamento e tamanho

de apresentação dos dados na listagem. Na Figura 69 são apresentados todos os campos necessários à criação de uma nova disciplina incluindo também a adição de materiais de suporte.

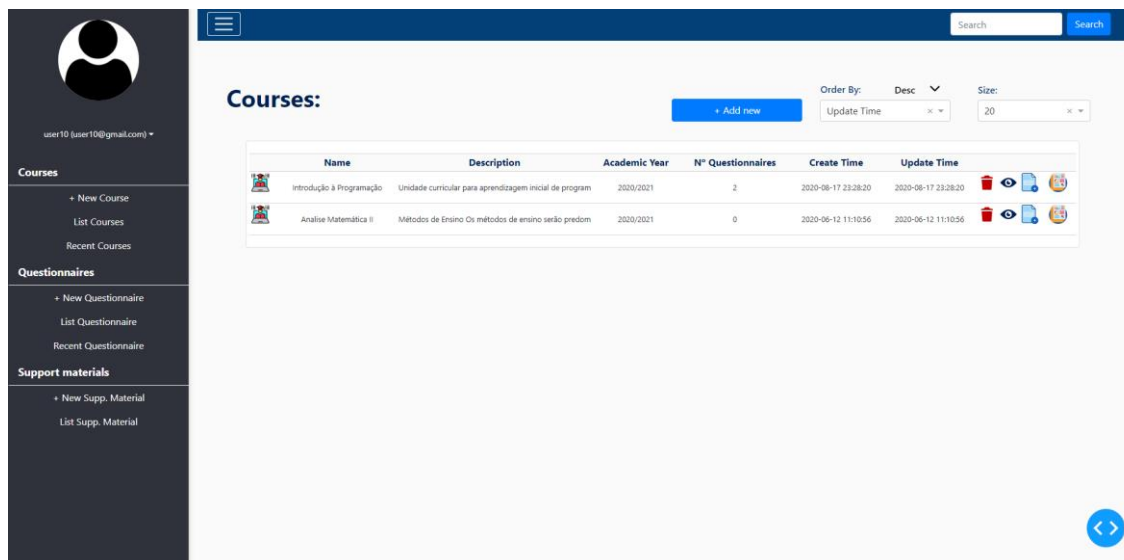


Figura 68 – Página do professor (1) BATINA.

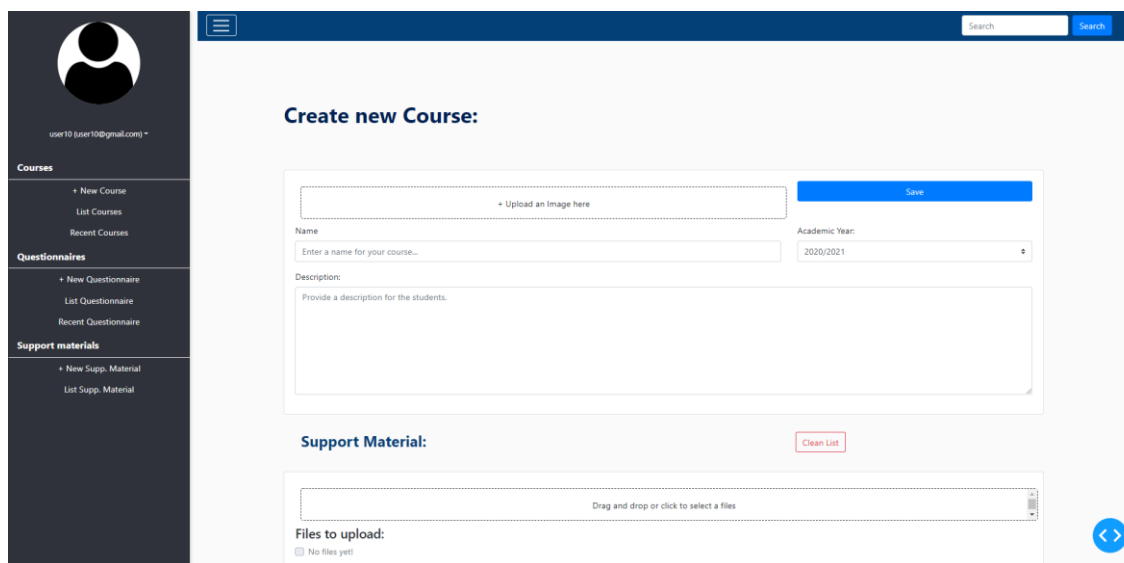


Figura 69 – Página do professor (2) BATINA.

Na Figura 70, é possível visualizar os dados relativos a uma determinada disciplina. Estes dados podem ser editados através da opção que se encontra no canto superior direito.

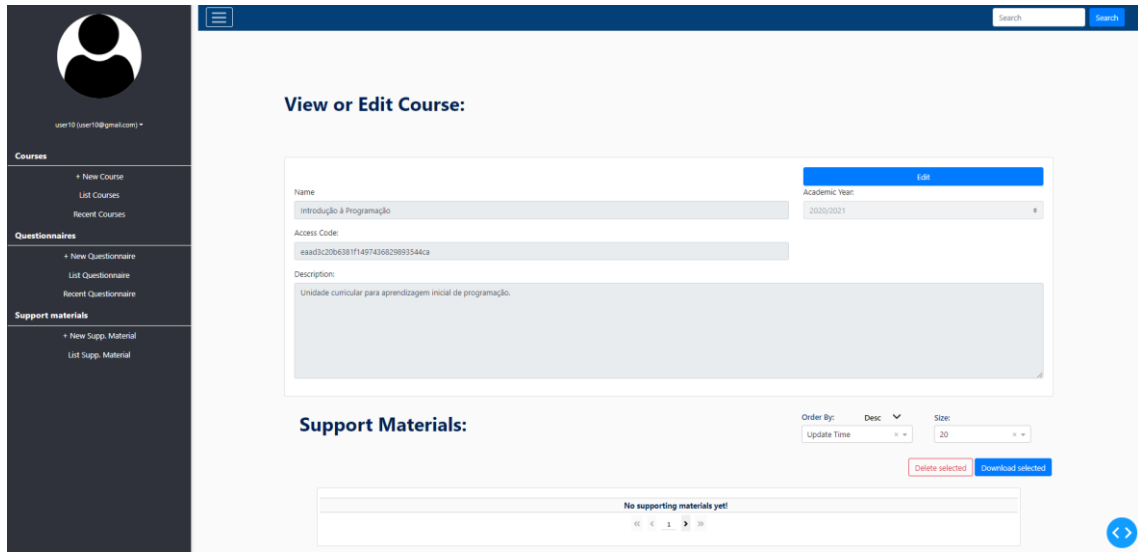


Figura 70 – Página do professor (3) BATINA.

Na Figura 71 é possível visualizar a listagem dos questionários existentes na plataforma dedicados a este professor. No canto superior direito são apresentados os botões dedicados ao ordenamento e tamanho de apresentação dos dados na listagem.

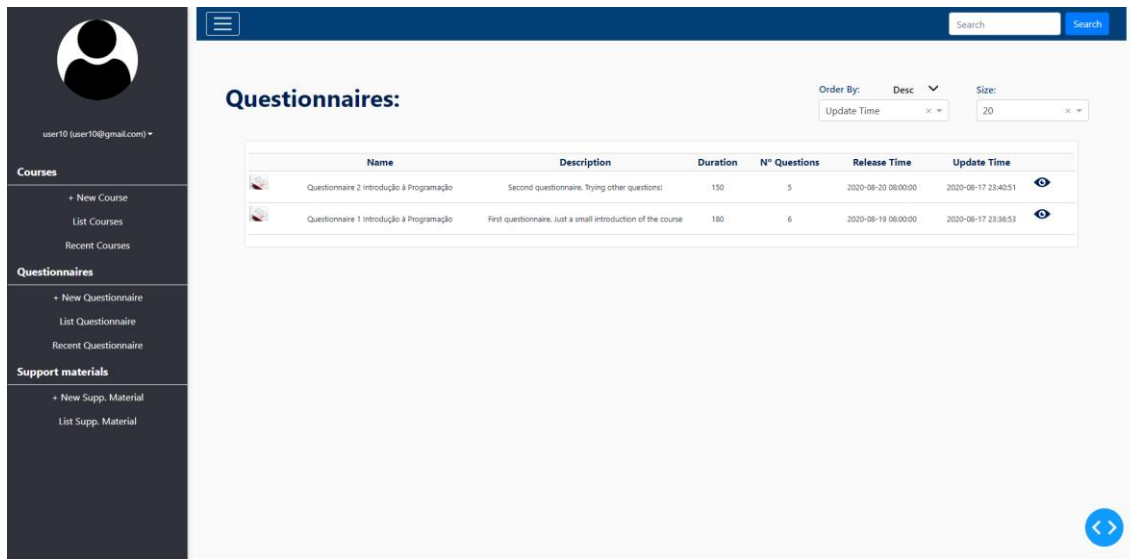


Figura 71 – Página do professor (4) BATINA.

Na Figura 72 é apresentada a página web dedicada à criação de um novo questionário. Neste campo o número de questões e respostas não é limitado, pelo que o questionário tem de ter mais do que dois minutos de duração e a data de lançamento do questionário terá de ser ter no mínimo um dia de diferença. O resultado deste questionário poderá ser observado e este é apresentado na Figura 73.

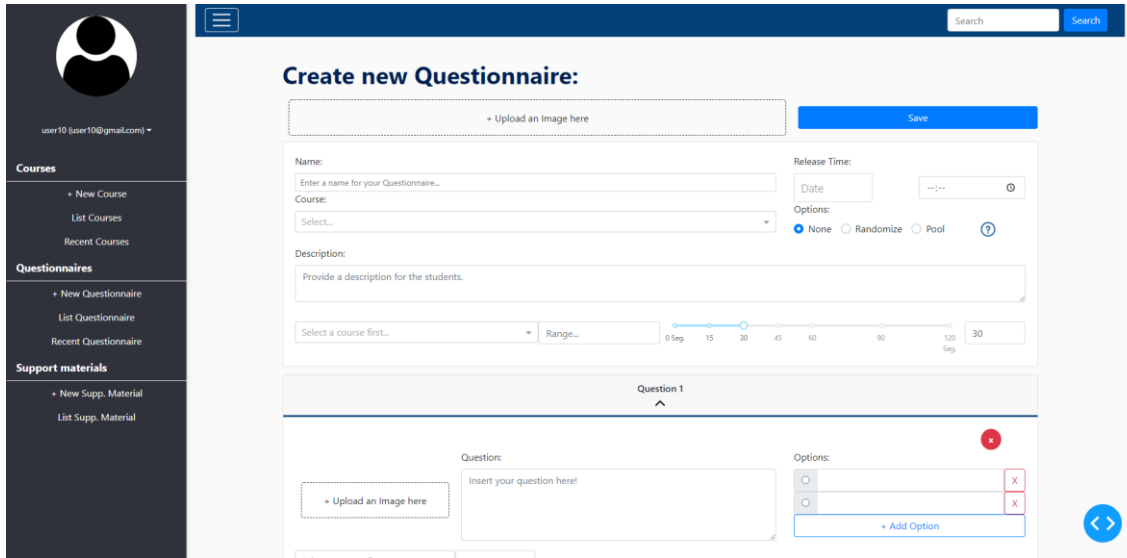


Figura 72 – Página do professor (5) BATINA.

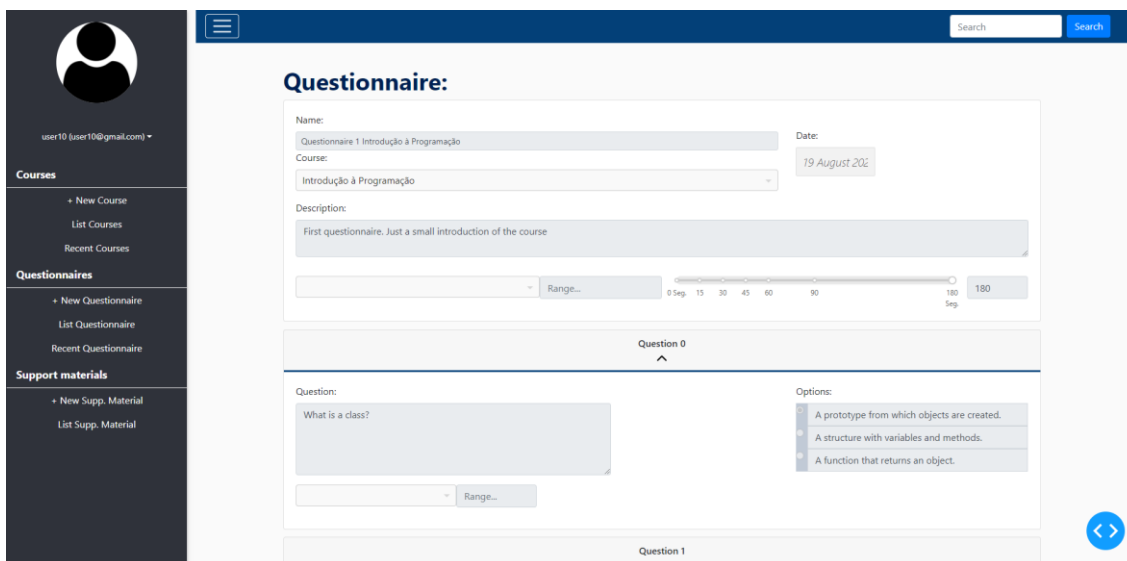


Figura 73 – Página do professor (6) BATINA.

## Capítulo 5

Na Figura 74 é apresentada a listagem de todos os materiais de suporte relativos às disciplinas do professor. No canto superior direito são apresentados os botões dedicados ao ordenamento e tamanho de apresentação dos dados na listagem. A Figura 75 representa a página web dedicada à adição de novos materiais de suporte, indicando a disciplina que esses materiais irão ser adicionados.

The screenshot shows the 'Support Materials' page. On the left is a dark sidebar with navigation options: 'Courses' (New Course, List Courses, Recent Courses), 'Questionnaires' (New Questionnaire, List Questionnaire, Recent Questionnaire), and 'Support materials' (New Supp. Material, List Supp. Material). The main content area has a search bar at the top right. Below it, the title 'Support Materials:' is followed by sorting options: 'Order By: Desc' and 'Size: 20'. There are 'Delete selected' and 'Download selected' buttons. A table lists four files with columns for 'File Name', 'Create Time', and 'Update Time'. Each row includes a checkbox, a file name, create and update timestamps, a delete icon, and a download icon.

	File Name	Create Time	Update Time		
<input type="checkbox"/>	filewith forbidden =characters.png	2020-06-12 13:27:46.854000	2020-06-12 13:27:46.854000		
<input type="checkbox"/>	batina_icon_blue.jpg	2020-06-12 13:27:46.648000	2020-06-12 13:27:46.648000		
<input type="checkbox"/>	batina_background_black.png	2020-06-10 11:17:50.677000	2020-06-10 11:17:50.677000		
<input type="checkbox"/>	xparent-functions.png.pagespeed.ic.P2v4M-PBYL_analise_mat.png	2020-06-10 09:23:19.199000	2020-06-10 09:23:19.199000		

Figura 74 – Página do professor (7) BATINA.

The screenshot shows the 'New Support Material' page. The sidebar is identical to Figure 74. The main content area has a search bar at the top right. Below it, the title 'New Support Material:' is followed by 'Cancel' and 'Upload' buttons. A dashed box contains the text 'Drag and drop or click to select a files'. Below this, there is a 'Files to upload:' section with a 'No files yet!' message and a 'Select the course to upload the file(s):' dropdown menu.

Figura 75 – Página do professor (8) BATINA.

Na Figura 76 é apresentada a página web dedicada à visualização de respostas a um questionário ao vivo. Nesta página o professor tem a possibilidade de navegar entre as várias questões (e suas respostas) do questionário. Ao lado de cada opção de resposta é apresentado o seu total de respostas, onde no fundo da página é apresentada a taxa de sucesso da respetiva questão.

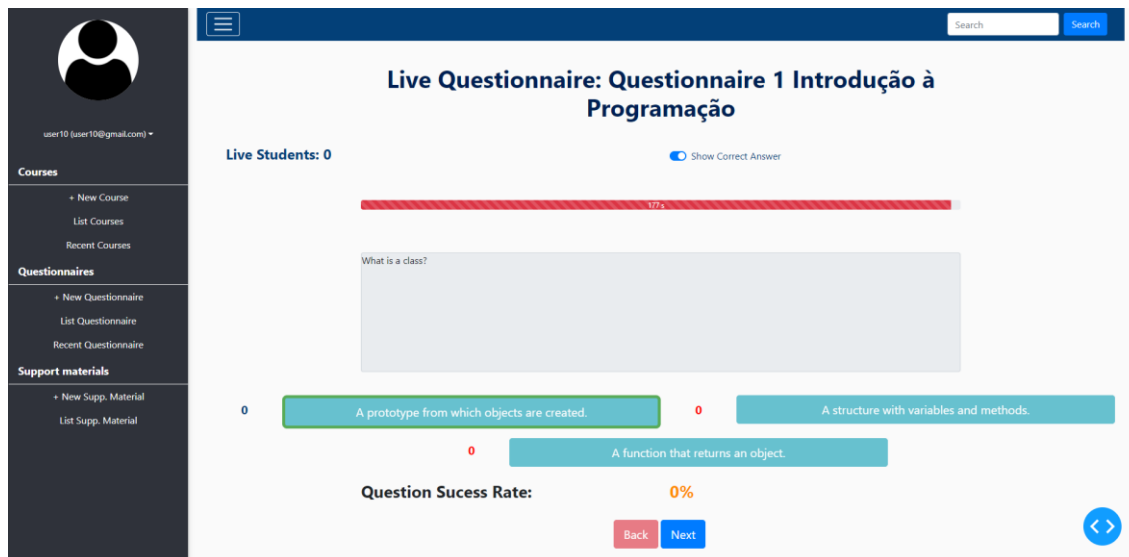


Figura 76 – Página do professor (9) BATINA.

As Figuras 77 a 80 representam a página web dedicada aos relatórios de resultados. As Figuras 77 e 78 representam os relatórios de resultados dedicados exclusivamente a um questionário e as Figuras 79 e 80 representam os relatórios de resultados dedicados a todos os questionários de uma disciplina. Nestas páginas web, são apresentados sempre os resultados em gráficos, em componentes de informação e também em listagens.



Figura 77 – Página do professor (10) BATINA.



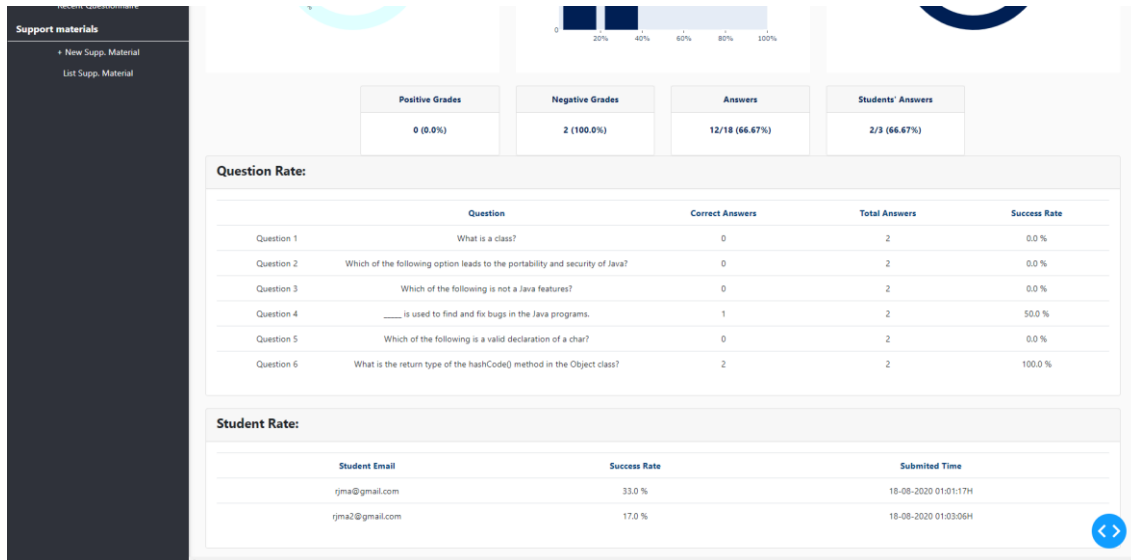


Figura 78 – Página do professor (11) BATINA.



Figura 79 – Página do professor (12) BATINA.

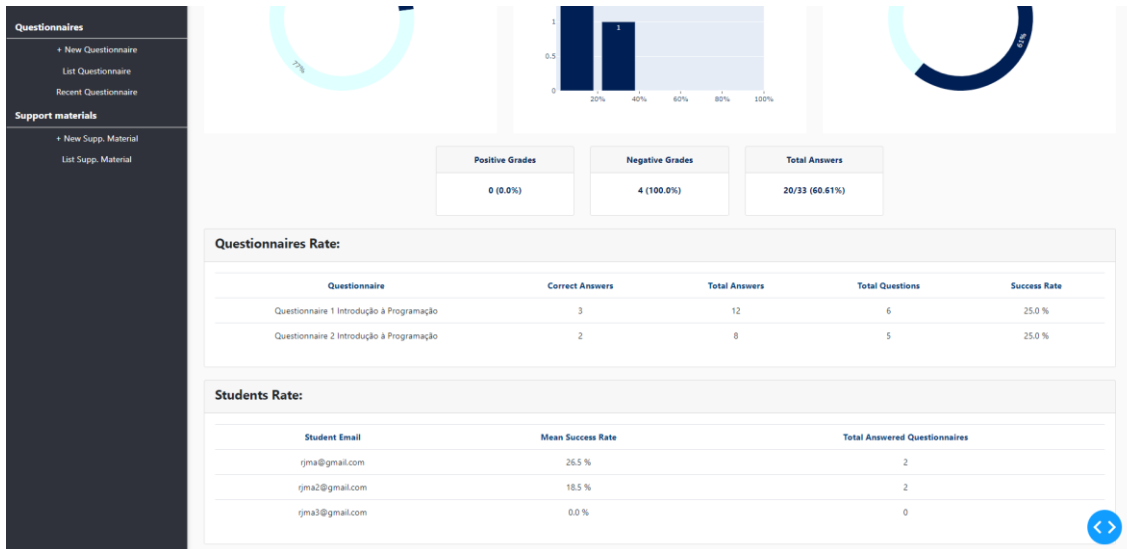


Figura 80 – Página do professor (13) BATINA.

# Capítulo 6

## Conclusões e trabalho futuro

Tal como foi explicado ao longo deste relatório, o objetivo deste trabalho era especificar, implementar e avaliar uma plataforma do tipo *dashboard* para integrar as aplicações ISABELA, 5GOpenclasses e BATINA, recolhendo e apresentando dados IoT. A especificação deste *dashboard* “IoTUnifiedManagement” foi apresentada ao longo de todo o capítulo 3 e a sua implementação apresentada nos capítulos 4 e 5. É de salientar que o plano de trabalhos definido para o segundo semestre sofreu algumas alterações, uma vez que foram também elaboradas implementações na aplicação BATINA.

Relativamente às funcionalidades de transmissão de conteúdo multimédia via WebRTC Kurento, considero que estas apresentam uma boa qualidade de implementação, uma vez que são suportadas por protocolos de transmissão de vídeo recentes em constante evolução.

O BATINA conta atualmente com uma versão *web-based* e com duas versões móveis: uma para Android e uma para iOS. Esta aplicação ainda se encontra em fase de desenvolvimento. Contudo, num futuro próximo poderá já estar disponível para uso académico.

Numa perspetiva futura, todas as especificações identificadas nos capítulos 3 e 4 relativas à gestão unificada de sensores IoT, utilizando os respetivos agentes, poderão ser integradas e implementadas no *dashboard* “IoTUnifiedManagement”. As integrações abordadas nestes capítulos incluem os agentes IoT Lightweight M2M e Ultralight 2.0, sendo estes agentes dedicados a trabalhar com comunicações e dispositivos restritos, onde a largura de banda e memória do dispositivo podem ser recursos limitados. Após estas integrações, a perspetiva de crescimento de utilização da aplicação “IoTUnifiedManagement” aumenta fortemente, uma vez que a gestão destes sensores IoT poderá ser implementada em diferentes aplicações móveis para além do ISABELA, 5GOpenclasses ou mesmo o BATINA.



# Referências

- [1] J. Desjardins, "How much data is generated each day?," 2019. [Online]. Available: <https://www.weforum.org/agenda/2019/04/how-much-data-is-generated-each-day-cf4bddf29f/>. [Accessed: 14-Jan-2020].
- [2] B. D. Framework, "Where does 'Big Data' come from?," 2019. [Online]. Available: <https://www.bigdataframework.org/short-history-of-big-data/>. [Accessed: 14-Jan-2020].
- [3] C. N. de Cibersegurança, "A Internet das Coisas." [Online]. Available: <https://www.cncs.gov.pt/a-internet-das-coisas-iot-internet-of-things/>. [Accessed: 14-Jan-2020].
- [4] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," *Proc. - Des. Autom. Conf.*, pp. 731–736, 2010.
- [5] J. Fernandes *et al.*, "A human-in-the-loop cyber-physical approach for students performance assessment," *Soc. 2019 - Proc. 2019 4th Int. Work. Soc. Sens.*, pp. 36–42, 2019.
- [6] "5G\_Anexo\_Tecnico\_2016-SI-10 ALB v1," Aveiro, Portugal.
- [7] Python Software Foundation, "What is Python?" [Online]. Available: <https://www.python.org/doc/essays/blurb/>. [Accessed: 14-Jan-2020].
- [8] J. Brains, "Python - The Python questions were only shown to the developers who chose Python as one of their three primary programming languages," 2019. [Online]. Available: <https://www.jetbrains.com/lp/devecosystem-2019/python/>. [Accessed: 14-Jan-2020].
- [9] F. Organization, "WHAT IS FIWARE?" [Online]. Available: <https://www.fiware.org/about-us/>. [Accessed: 14-Jan-2020].
- [10] F. Organization, "WHAT IS FIWARE?," 2011. [Online]. Available: <https://www.fiware.org/2011/05/17/what-is-fiware/>. [Accessed: 14-Jan-2020].
- [11] Prometheus, "What is Prometheus?" [Online]. Available: <https://prometheus.io/docs/introduction/overview/>. [Accessed: 14-Jan-2020].
- [12] J. M. Fernandes, J. Sá Silva, and F. Boavida, "IoT Student Advisor and BEst Lifestyle Analyzer," 2017.
- [13] Git, "About Git." [Online]. Available: <https://git-scm.com/about/>. [Accessed: 14-Jan-2020].
- [14] I. Agenda, "What is Internet of Things (IoT)?," 2019. [Online]. Available: <https://internetofthingsagenda.techtarget.com/definition/Internet-of-Things-IoT>. [Accessed: 14-Jan-2020].
- [15] J. Research, "'INTERNET OF THINGS' CONNECTED DEVICES TO ALMOST TRIPLE TO OVER 38 BILLION UNITS BY 2020." [Online]. Available: <https://www.juniperresearch.com/press/press-releases/iot-connected-devices-to-triple-to-38-bn-by-2020>. [Accessed: 14-Jan-2020].
- [16] N. Armando,, R. Almeida,, J. M. Fernandes,, J. S. Silva,, and F. Boavida,, "End-to-end Experimentation of a 5G Vertical within the Scope of Blended Learning," *IEEE Trans. Netw. Serv. Manag.* [Submetido em 18/08/2020]
- [17] J. M. Fernandes; *et al.*, "BATINA," Coimbra, Portugal, 2020.

- [18] Google, "Google Chrome," 2020. [Online]. Available: <https://www.google.com/intl/pt-PT/chrome/>. [Accessed: 22-Aug-2020].
- [19] Apple, "Safari," 2020. [Online]. Available: <https://www.apple.com/safari/>. [Accessed: 22-Aug-2020].
- [20] "Mozilla Firefox," 2020. [Online]. Available: <https://www.mozilla.org/pt-PT/firefox/new/>. [Accessed: 22-Aug-2020].
- [21] Microsoft, "Microsoft Edge," 2020. [Online]. Available: <https://www.microsoft.com/pt-pt/edge>. [Accessed: 22-Aug-2020].
- [22] F. Organization, "WELCOME TO ORION CONTEXT BROKER." [Online]. Available: <https://fiware-orion.readthedocs.io/en/master/index.html>. [Accessed: 14-Jan-2020].
- [23] F. Organization, "WELCOME TO THE FIWARE SHORT TIME HISTORIC (STH)." [Online]. Available: <https://fiware-sth-comet.readthedocs.io/en/latest/>. [Accessed: 14-Jan-2020].
- [24] F. Organization, "IDENTITY MANAGER - KEYROCK." [Online]. Available: <https://fiware-idm.readthedocs.io/en/latest/>. [Accessed: 14-Jan-2020].
- [25] FIWARE, "Fiware LWM2M," 2020. [Online]. Available: <https://fiware-iotagent-lwm2m.readthedocs.io/en/latest/>. [Accessed: 22-Aug-2020].
- [26] FIWARE, "Fiware ULTRALIGHT," 2020. [Online]. Available: <https://fiware-iotagent-ul.readthedocs.io/en/latest/>. [Accessed: 22-Aug-2020].
- [27] A. Developers, "Android.hardware," 2020. [Online]. Available: <https://developer.android.com/reference/android/hardware/package-summary>. [Accessed: 22-Aug-2020].
- [28] A. Developers, "SensorManager," 2020. [Online]. Available: <https://developer.android.com/reference/android/hardware/SensorManager>. [Accessed: 22-Aug-2020].
- [29] A. Developers, "SensorEvent," 2020. [Online]. Available: <https://developer.android.com/reference/android/hardware/SensorEvent>. [Accessed: 22-Aug-2020].
- [30] greenrobot, "greenDAO," 2020. [Online]. Available: <https://greenrobot.org/greendao/>. [Accessed: 22-Aug-2020].
- [31] A. Developers, "MediaRecorder," 2020. [Online]. Available: <https://developer.android.com/reference/android/media/MediaRecorder>. [Accessed: 22-Aug-2020].
- [32] G. Developers, "Fused Location Provider API," 2020. [Online]. Available: <https://developers.google.com/location-context/fused-location-provider>. [Accessed: 22-Aug-2020].
- [33] Kurento, "How to install Coturn?," 2020. [Online]. Available: <https://doc-kurento.readthedocs.io/en/6.14.0/user/faq.html#how-to-install-coturn>. [Accessed: 22-Aug-2020].
- [34] Kurento, "Kurento Media Server Docker," 2020. [Online]. Available: <https://hub.docker.com/r/kurento/kurento-media-server>. [Accessed: 22-Aug-2020].
- [35] Kurento, "One to many video call," 2020. [Online]. Available: <https://doc-kurento.readthedocs.io/en/6.14.0/tutorials/node/tutorial-one2many.html>. [Accessed: 22-Aug-2020].

- [36] Kurento, "JavaScript Recorder," 2020. [Online]. Available: <https://doc-kurento.readthedocs.io/en/6.14.0/tutorials/js/tutorial-recorder.html#javascript-logic>. [Accessed: 22-Aug-2020].
- [37] N. Cao, "nc-kurento-android," 2020. [Online]. Available: <https://github.com/nhancv/nc-kurento-android>. [Accessed: 22-Aug-2020].





# Apêndices



# Apêndice A

Neste apêndice é apresentado o artigo “End-to-end Experimentation of a 5G Vertical within the Scope of Blended Learning”, escrito durante o segundo semestre e submetido ao *journal* IEEE Transactions on Network and Service Management [16].





### End-to-end Experimentation of a 5G Vertical within the Scope of Blended Learning

Journal:	<i>IEEE Transactions on Network and Service Management</i>
Manuscript ID	TNSM-2020-03361
Manuscript Type:	Original Article
Date Submitted by the Author:	18-Aug-2020
Complete List of Authors:	Armando, Ngombo; Universidade de Coimbra, DEI Magueta Almeida, Rui Jorge; University of Coimbra Faculty of Sciences and Technology, Fernandes, José Marcelo; Universidade de Coimbra, Computer Science Sa Silva, Jorge; University of Coimbra, Department of Electrical and Computer Engineering Boavida, Fernando; University of Coimbra, Informatics Engineering
Keywords:	Multimedia services < Service Management, Information technology services < Service Management, Middleware < Technologies, Prototype implementation and testbed experimentation < Methods

# End-to-end Experimentation of a 5G Vertical within the Scope of Blended Learning

Ngombo Armando, *DEI, University of Coimbra, CISUC*, Rui Almeida, *DEI, University of Coimbra, CISUC*, José Marcelo Fernandes, *DEI, University of Coimbra, CISUC*, Jorge Sá Silva, *DEEC, University of Coimbra, CISUC*, and Fernando Boavida, *DEI, University of Coimbra, CISUC*

**Abstract**—The next generation of mobile networks, widely known as 5G, was designed to respond to the communication challenges for the next ten years. 5G will be an essential structure on which all types of electronic communications will rely. Built on the emerging 5G technology, we developed a vertical solution for blended learning environments within the scope of a Portuguese national project. The project aims to propose an integrated demonstration of a set of products, capable of being part and providing services in the framework of the future 5G networks. To this end, we adopted a bottom-up approach to design and develop a product named *5GOpenclasses*, where we leveraged FIWARE middleware for the management of all entities. In this paper, we present the architecture, technological platform, associated data structures, and end-user applications of *5GOpenclasses*. We also present the design of innovative location-based service for blended learning environments. This paper is the first step of the proposed product towards its quantitative evaluation when running Over-the-Top on both 4G to 5G networks. Although successful unit tests were carried out in what concerns the functional outcome, the integration tests for quantitative results depend on the availability of other components of the project.

**Index Terms**—5G, Blended Learning, FIWARE.

## I. INTRODUCTION

THE defined use cases for 5G are high-level expectations that drive the development of all components for the communications infrastructure in the year 2020 and beyond. Such use-cases are known as enhanced Mobile Broadband (eMBB), massive Machine Type Communications (mMTC) and ultra-reliable and Low-Latency Communications (urLLC) [1]. In this context, we designed and developed a product named *5GOpenclasses* to support blended learning activities.

*5GOpenclasses* includes a mobile application that allows students to view high definition (HD) live streams of the class sessions, access materials from a multimedia repository and use a real-time chat. During the class sessions, remote students can interact with teachers. From a web application interfaces, a teacher can create and delete classrooms, and upload, edit or

Submitted for review at 18-August-2020. This paper was carried out within the scope of the 5G Project (POCI/01/0247/FEDER/024539), co-financed by COMPETE 2020, Portugal 2020 - Operational Program for Competitiveness and Internationalization (POCI I), European Union's ERDF (European Regional Development Fund) and the Portuguese Foundation for Science and Technology (FCT). J.M. Fernandes also wishes to acknowledge the FCT for supporting his research under the grant number SFRH/BD/147371/2019.

delete files from the backend application server. An array of live video streams serving several students is an ideal scenario to leverage the eMBB network capability, while the collection of both mobile and stationary sensor data enables to assess the network's mMTC capability. The reactivity on providing the tailored resources to the students, depending on their location, is a relevant test case for the urLLC capability.

In this paper, we present the approach that we adopted towards the implementation of a 5G-ready product. We highlight the design methodology that is based on a bottom-up approach and good practices from the literature for successful development of blended learning environments in 5G. We also provide a demonstration video clip with the functional results of the product and present the design for the quantitative assessment of the test cases, when running Over-the-Top (OTT), on both 4G and 5G 3GPP networks.

The remainder of this paper is organised as follows. In Section II, we present the related work and the context of the project. In Section III, we describe the architecture and the components of the proposed solution. Section IV focuses on end-user applications of *5GOpenclasses*. In Section V, we present the unit tests carried out in what concerns the functional outcome, and we address the scenario designed for the integrated demonstration. Finally, we provide concluding remarks in Section VI.

## II. RELATED WORK AND CONTEXT

### A. Related Work

The effective design of blended learning environments should consider four aspects, namely, incorporate flexibility, stimulate interaction, facilitate student learning processes, and foster a productive learning environment [2]. The use of Information and Communication Technologies as a support tool for better education is not new and is not meant to replace face-to-face activities. Amid pandemic situations such as COVID-19, online teaching facilities and blended learning environments are essential to avoid the lockdown of training programs. The

The authors are with the Centre of Informatics and Systems of the University of Coimbra (CISUC), 3000-214 Coimbra, Portugal (e-mails: narmando@dei.uc.pt, rjma@student.dei.uc.pt, jmfernandes@dei.uc.pt, sasilva@deec.uc.pt, boavida@dei.uc.pt).

N. A is also with Escola Superior Politécnica do Uíge, Universidade Kimpa Vita, Nkondo Mbenza, Uíge, Angola.

1 exercise of designing a blended learning environment must be a  
2 joint work by both researchers and practitioners on education.

3 According to *Boelens et al.* [2], such design must incorporate  
4 solutions that target the following question: which amount of  
5 flexibility is desirable? As for the stimulating interaction, it is  
6 interesting to mention that many learners do not want to lose the  
7 social interaction and human touch they are used to in a face-to-  
8 face environment while benefiting from the flexibility offered by  
9 the blended learning method. There exist different categories of  
10 effective strategies to foster a productive learning environment.  
11 The strategies include motivating, concentrating and exerting  
12 efforts, in addition to attributing and judging oneself, appraising,  
13 and finally, dealing with emotions. Overall, the comprehensive  
14 literature review by *Boelens et al.* shows that the implementation  
15 of a blended learning environment must rely on a strategy from  
16 the institution that wants to adopt it. The institution should also  
17 choose the kind of activities that can be blended and introduce a  
18 new culture for all the stakeholders.

19 Recent studies have been presented with approaches to  
20 leverage both 5G and the Internet of Things (IoT) into blended  
21 learning environments. In [3], *Ever and Rajan* investigated and  
22 identified how these emerging technologies would be integrated  
23 into education sciences like medicine. Their findings include the  
24 fact that 5G will especially foster mobile learning by providing  
25 educators with efficient learning scenarios that satisfy high  
26 priority learner requirements. To *Ever and Rajan*, security and  
27 privacy are also the focus in the development of future solutions  
28 on blended learning environments over 5G.

29 In [4], *Zhongmei et al.* measured the effectiveness of learning  
30 through long-distance digital support. They submitted a Quality  
31 of Experience (QoE) questionnaire to 180 college students  
32 through a Mobile Communication course, comprising both  
33 theoretical and practical operation exercises. The authors  
34 recommended that in the future, schools should try to integrate  
35 both 5G and IoT technologies to their hybrid teaching. To them,  
36 5G will shorten the download delay time and expand the capacity  
37 of mixed reality content and video of the class sessions. An  
38 “automatic login when entering the classroom” is an example of  
39 leveraging IoT to enhance both learning and teaching experience.  
40 Likewise, the possibility of gathering sensor data that would be  
41 exploited to analyse any relevant information in real-time to help  
42 teachers in successfully delivering lectures.

43 Similarly, *Baratè et al.* [5] presented an example of the  
44 adoption of 5G technologies in an educational context. More  
45 specifically, they presented an application to laboratory activities  
46 of music education based on both Augmented Reality (AR) and  
47 Virtual Reality (VR). The authors state that teaching music to  
48 remote students is difficult because since it implies the exchange  
49 of high-quality audio streams and symbolic information such as  
50 score, metadata, and lyrics. The complexity in remote teaching of  
51 music increases when combined with AR/VR format classes. For  
52 this reason, the authors believe that available communication  
53 technologies are not suitable for experiencing this kind of  
54 didactic activities. Hence, the capabilities envisioned in 5G  
55 would be relevant to the needs of remote live music classes.

56 The studies presented above give essential guidance for  
57 designing an effective blended learning environment [2], [3],  
58  
59  
60

over 5G [3]–[5] and IoT technologies [3], [4]. The implemented  
solutions in [3]–[5] are described as based on a top-down  
approach. Accordingly, we believe that their proposed designs  
are more generic to the supported communication network and  
thus; they risk not to produce fully optimised products for the 5G  
network as envisioned in [1].

Based on the guidelines from [2]–[5], we designed and  
developed a product named *5GOpenclasses*. Our design relies on  
a bottom-up approach; that is, we started by studying the  
capabilities offered by the future 5G network, and then set a  
solution that makes the most of it. The main advantage of a  
bottom-up approach is that a proposed artefact is designed based  
on the reality of the network enablers. The first novelty of the  
current work is the exploitation, in a single product, of all use-  
cases described in the 5G vision [1]. Second, we propose a  
process to provide a differentiated service on-the-fly, according  
to the location of the couple UE.

Next, we will present the context of our study and the proposed  
solution.

### B. Context of the Project

We developed a use case study within the scope of a national  
project named *Mobilizador 5G* (<https://5go.pt/en/home/>). For  
Portugal, this is one of the main projects to provide 5G Processes,  
Products and Services. From a technological viewpoint, the  
project aimed to research, develop, validate, integrate, and  
demonstrate both mechanisms and solutions to provide services  
in the framework of future 5G networks. The project also aims at  
joining and harmonising efforts of different telecommunication  
operators, intending to create innovative solutions for markets in  
both Business-to-Business and Business-to-Customer models.  
The targeted products should cover all the functional domains in  
5G networks, namely access network, core network, and vertical  
sectors organised into machine-to-machine communication and  
human communication.

*Mobilizador 5G* involved fourteen companies, research, and  
innovation centres, under the leadership of Altice Labs - Portugal.  
The set of Processes, Products and Services to be developed by  
the partners of the project are expected to be of Technology  
Readiness Level (TRL) 7/8 [6] and will be assessed in an  
integrated demonstration infrastructure.

In this context, we proposed a product for human  
communication that would stress the supported communication  
network in what concerns its capabilities for latency, broadband  
and machine-type communications. To this end, we designed,  
developed, and tested a product to support blended learning  
activities named *5GOpenclasses*.

## III. PROPOSED SOLUTION

### A. Architecture

The architecture of the *5GOpenclasses* is displayed in Fig. 1,  
which we will explain the different components. The IoT boxset  
is a 3D printed component that we developed in the scope of a  
previous project [7], to monitor the temperature, light and noise  
of the surroundings, in this case, the physical classroom. The IoT  
boxset contains a Raspberry Pi gateway, an Arduino board, and  
sensors for communication, preprocessing, and sensing purposes,

1 respectively. The mobile User Equipment (UE) is an Android OS  
 2 handheld device that runs the *5GOpenclasses* mobile application  
 3 serving the students to view live streams of the class sessions,  
 4 replay and download videos from a repository, and chat with both  
 5 teachers and classmates. More details about the mobile  
 6 application will be given in Section IV.A.

7 During the class sessions, we can collect two categories of  
 8 sensor data, namely, mobile and stationary. The former  
 9 comprises gyroscope, accelerometer, proximity, sound, and light.  
 10 The latter comprises the ones from the IoT boxset. Overall, the  
 11 collected data from sensors feed several analytical services so  
 12 that the teachers can be able to monitor the influence of the sensed  
 13 values in the students' performances as we have also exploited in  
 14 previous works [7], [8].

15 The core functions of the *5GOpenclasses* system are  
 16 represented in the middle of Fig.1. The application server is based  
 17 on FIWARE IoT middleware that will be described in Section  
 18 III.B. The application server handles the provision of session  
 19 control and media server. The application server also hosts the  
 20 storage for the class sessions materials.

21 The Class Management component is provided by a web  
 22 application that serves both the teachers and the administrator of  
 23 the *5GOpenclasses* system. Via this application, a teacher can  
 24 create, manage and close class sessions. In turn, the administrator  
 25 can, for instance, add new subjects to teachers or view the current  
 26 list of classes per subject. More details about the web application  
 27 will be given in Section IV.B.

28 At the top-right side of Fig. 1, we can see different sources of  
 29 videos streams. During the class session, students can consume  
 30 an array of live video both from indoors (fixed cameras) and  
 31 outdoors (attached to Unmanned Aerial Vehicles (UAV) 360°  
 32 cameras and cameras attached to firefighters suits labelled  
 33 *Bodykit*). Two partners of the project provide the outdoors  
 34

streams from the two type of cameras. Likewise, the optimised  
 video streaming will rely on an eMBB slice to be configured by  
 the partner of the project in charge of the 3GPP mobile networks.

In *5GOpenclasses*, we propose that the students should not  
 be able to chat, ask questions, and watch the live streams if they  
 were detected as being inside the physical classroom facility.  
 Moreover, the mobile sensors data are to be collected only for  
 students detected as being inside the physical classroom  
 facility. For the current testbed of the *5GOpenclasses*, this  
 detection is time-based. Hence, every 30 seconds, we toggle the  
 value of a Boolean named *in\_class* (see Table I), which is an  
 attribute of the *Student* entity. However, in an advanced version  
 of the *5GOpenclasses*, the *in\_class* values are to be managed  
 automatically by the process that is depicted in Fig. 2.

The automatic location-based service was not developed in  
 the initial version of *5GOpenclasses* because it involves  
 interfaces from different stakeholders of the consortium that  
 must be carefully articulated. Furthermore, the privacy issue  
 related to the UE tracking that must be carefully considered.  
 The sequence diagram for the location-based service in Fig. 2  
 works as follows.

Firstly, the teacher initiates the class session and  
 automatically informs the location of the classroom session to  
 the application server (*location* Attribute from *Rooms* entity in  
 Table I). The location can be based on Global Navigation  
 Satellite System (GNSS), plus the elevation coordinates or any  
 other format compliant with the mobile network location data  
 type. The location may include the coordinates of a single point  
 of the classroom or an array of coordinates that enables the  
 representation of the entire classroom object in the space. When  
 a student logs in and joins a class session, the mobile application  
 will send the users credentials and its UE identification to the  
 application server.

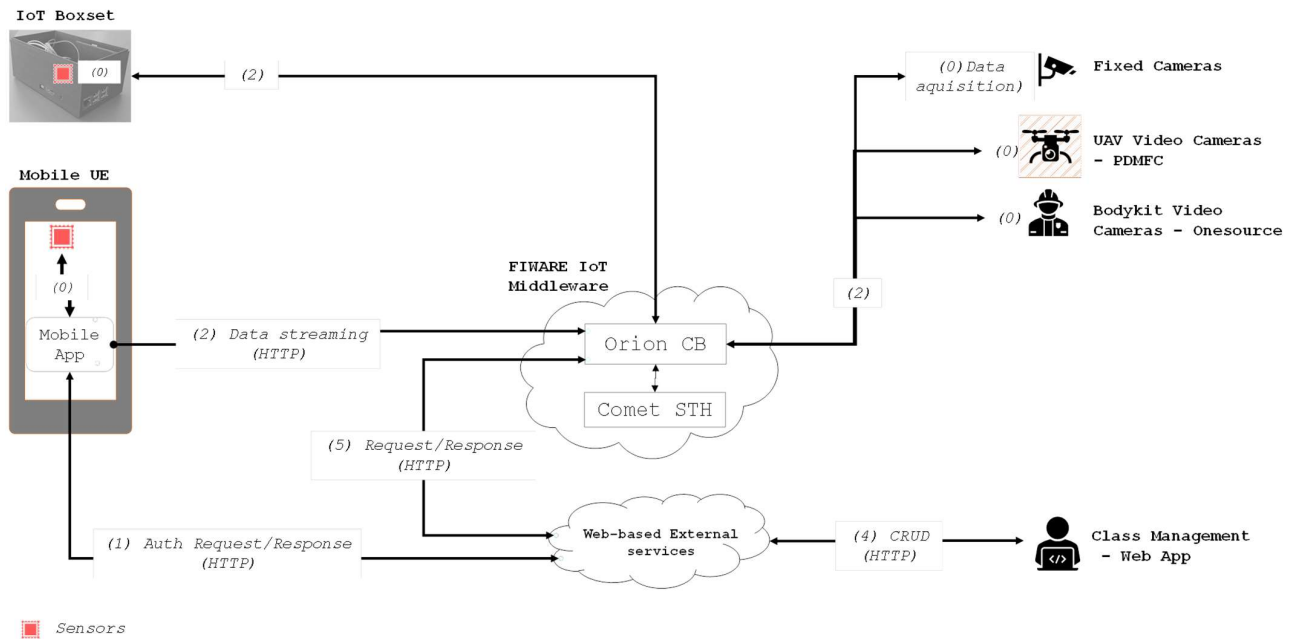


Fig. 1. 5GOpenclasses - Technologies and data flows



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

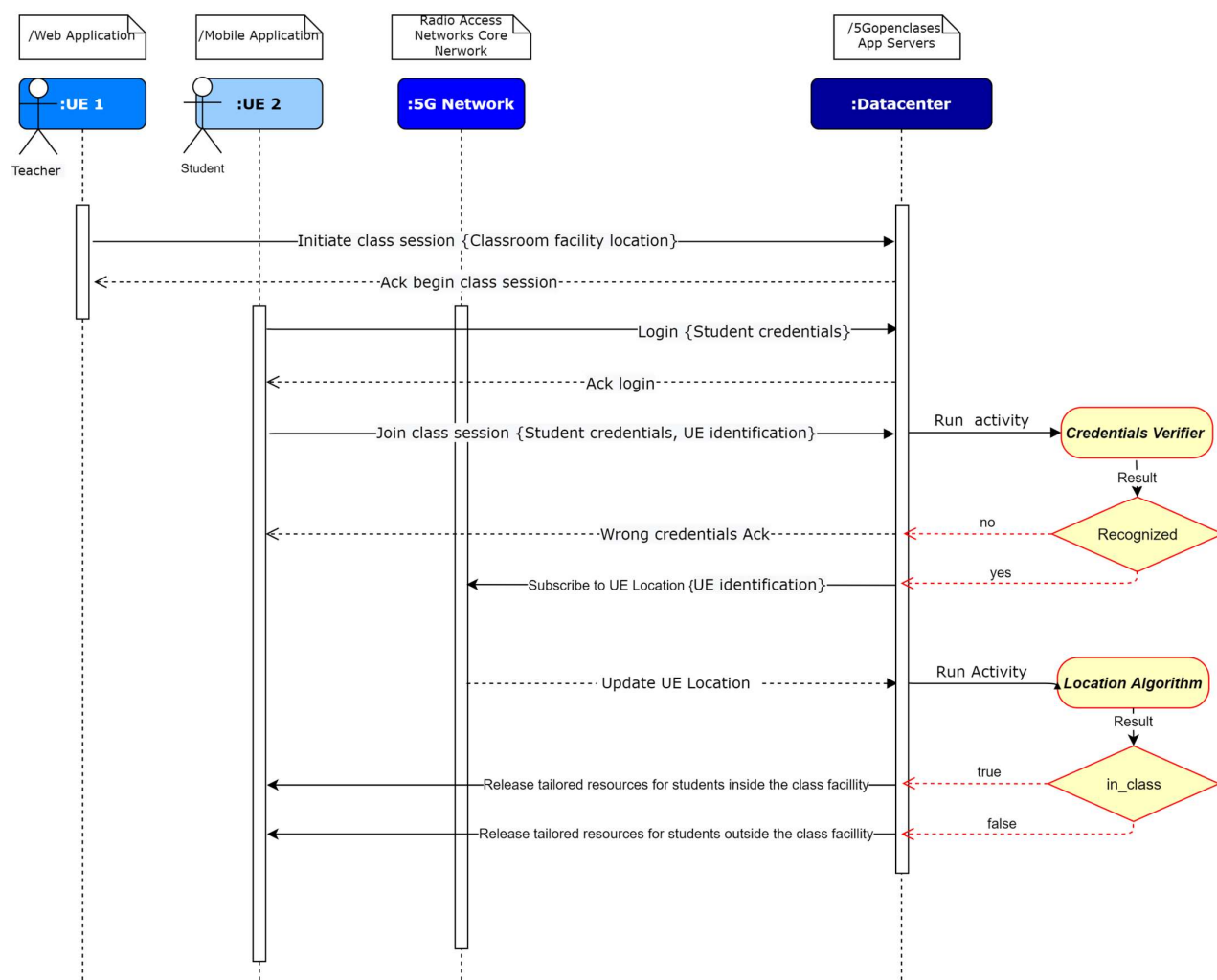


Fig. 2. Location-based service sequence diagram

The UE identification can be any unique number that enables the mobile network to track its location throughout the cells. E.g. the International Mobile Equipment Identity (IMEI), the Integrated Circuit Card Identifier (ICCID), and the International Mobile Subscriber Identity (IMSI). The user credentials serve to authenticate the users in the 5GOpenclasses application server. If the credentials are recognised, the application server will send the associated UE identification to the mobile network entities in charge of tracking the UEs. The mobile network must provide the application server with the initial UE location and the updated values. Such communication is to be made via a subscription so that whenever UE moves, the mobile network publishes the new location coordinates to the application server.

Both the initial and updated locations coordinates of the UE feed a computational process tagged as “Location Algorithm” in Fig. 2. This algorithm will compare the pre-registered location of the Room entity with the current location of the UE. If the point UE is located inside the object Room, then the *in\_class* value will be “true”. Else, the *in\_class* value will be “false”. Finally, the

mobile application will release tailored resources to the end-users, depending on its binary location. As we proposed for the current version of the product, if *in\_class = true*, the mobile application will deactivate the chat, ask questions functionality, and live streaming. With *in\_class = false*, the mobile sensors data will not be collected.

B. Core Servers

FIWARE Platform

FIWARE is a European set of specifications to enable the smooth development of smart applications in multiple vertical sectors. FIWARE is open, public and royalty-free. The FIWARE project started in 2011 within the Seventh Framework Programme (FP7) of the European Commission as part of the Future Internet Public-Private Partnership Programme (FI-PPP). The primary goal of the FI-PPP was to advance a process for harmonised European technology platforms and their implementation. Moreover, the FI-PPP aim was to provide integration and harmonisation of relevant policies frameworks.

The process to establish FIWARE in the market was divided into three phases. The first phase aimed at creating the technological core, while the second mainly aimed at the implementation of the FIWARE data centers known as “nodes”. The last phase aimed primarily at creating a sustainable ecosystem for Small Medium Enterprises, through the selection of sixteen business accelerators.

FIWARE is based on a library of components called Generic Enablers (GEs) that offer reusable and commonly shared functions as a Service exposed by RESTful APIs. Fig. 3 portrays the typical components of the FIWARE platform.

The FIWARE project adopts the Next-Generation Service Interface (NGSI) 9/10 data model, previously developed by the Open-Mobile Alliance (OMA) [9]. The model is based on entities and attributes. Functions in *5GOpenclasses* need to be scalable, and dynamic models are needed to represent the data of such a system. Each entity has its type and is represented by attributes, using JSON format.

All the entities in *5GOpenclasses* are managed by the ORION GE. Thus, it is in ORION where entities can be created, deleted, retrieved, and updated. This module acts as a broker, by allowing external systems (consumers) to make data subscriptions with specific rules to the entities and attributes, which improves the architectural distribution and scalability. ORION sends notifications to these consumers when subscription rules are observed. The last values of information sent by the entities are stored in the attached ORION databases. However, ORION is not fit for storing historic data. Instead, the Cygnus module is the GE in charge of persisting specific sources of data in several storages, using the Apache Flume technology and by subscribing to the ORION entities. Accordingly, when a new entity arrives at the Cygnus GE, the listener will put it in a specific channel and forward it to a third-party data storage (e.g., MySQL, MongoDB and CKAN channel).

Databases usually do not provide APIs to retrieve the data to applications. Therefore, another module of the FIWARE platform handles this issue, namely the STH -Comet GE. STH provides a RESTFUL API with historic-queries capabilities, and

aggregated methods so that each time the end applications need to access historic data, the Comet will connect to the MongoDB and retrieve it.

As for the FIWARE implementation, the *5GOpenclasses* application server runs a cloud-based Virtual Machine (VM) with a docker environment. The docker environment serves to manage a suite of containers that implement the GEs.

#### WebRTC Video Streaming

We are using a Kurento Web Real-Time Communication (WebRTC) server to work as a receiver and transmitter for real-time video streams. Kurento is also a GE of FIWARE. While standard WeRTC media servers' tasks include transcoding, Multipoint Control Unit (MCU) and recording, Kurento server adds tasks such as flexible processing, augmented reality, blending, mixing, and analysing. We recall that the *5GOpenclasses* web application provides the features for recording and transmitting video to students. These video streams are redirected to a WebRTC multimedia server. When the students want to view these contents, they can do so through the mobile application. In this scenario, we leverage a One-to-Many video call real-time transmission architecture.

Due to security reasons, we had to deploy two independent servers, namely a STUN Server (for Session Traversal of User Datagram Protocol [UDP] Through Network Address Translators [NATs]) and a TURN Server (for Traversal Using Relays around NAT). In a real context, most devices are behind firewalls, where NAT rules filter the communications. STUN servers are used to obtain an IP address and an external network port, while TURN servers are used to relay traffic if the direct connection (device to device) fails. In other words, the overall purpose of STUN servers is to obtain external IP/port addresses, and TURN servers are used to relay traffic if the peer-to-peer connection fails.

All devices that want to offer/receive real-time data content must be synchronised temporally to be able to communicate in real-time. This simultaneous synchronisation process is called *signalling* and is used to exchange information of:

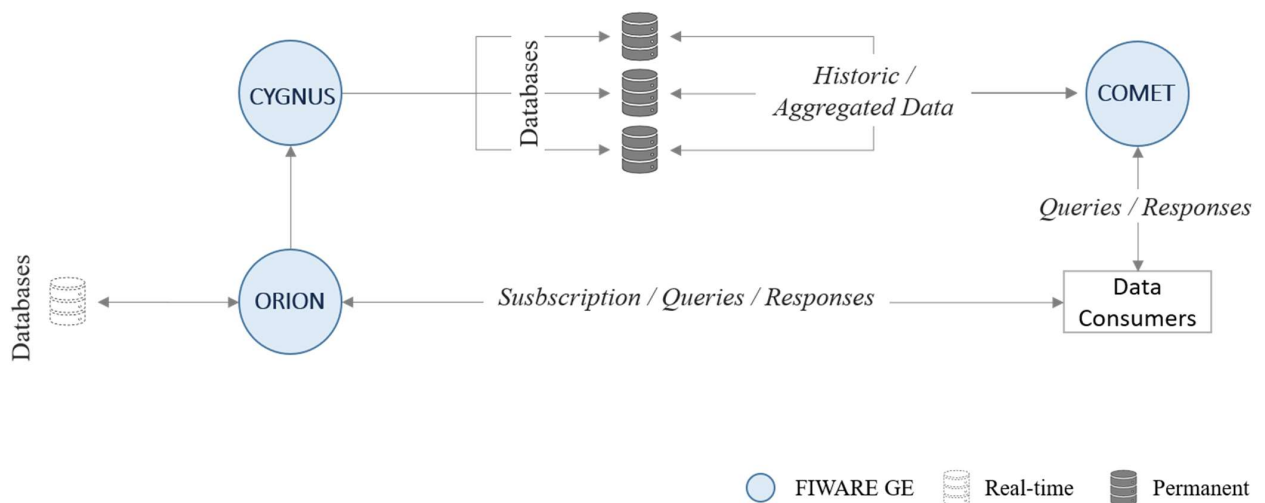


Fig. 3. Typical Components of a FIWARE Instance

- session control,
- error messages,
- transmission channel data
- critical data for establishing secure connections and
- network data (IP address and port).

Even if the devices that should receive content in real-time are synchronised, they should not yet be able to do so because of NATs and firewalls filtering rules. WebRTC applications can use the Impact, Confidence, and Ease (ICE) framework to overcome the complexity of today's networks. An ICE framework is meant to find the best way to connect devices. To achieve such a task, the best path to connect the devices is tested in the following order of options:

- A direct connection between devices (impossible due to NATs)
- The utilisation of an external address of a STUN server
- If it fails, the traffic is directed through a TURN relay server.

NATs offer the possibility of communicating an internal IP address of a local network with the outside, through its translation. In a peer-to-peer scenario, there is no possibility for an IP address, on a private network, to be used as a public address. To overcome this issue, WebRTC uses STUN servers that are hosted on the public Internet. Besides, WebRTC applications use STUN servers to discover both the end-device public IP address and port, thus enabling the transmission of that public address to the peer of direct transmission (peer-to-peer transmission). The exchange of both the IP and public port between the transmitter and receiver of multimedia content is done through the signalling mechanism. The result of this communication with the STUN servers is the establishment of a direct connection between IP addresses/external ports.

For most WebRTC applications to function normally, it is necessary to use servers to relay traffic between the provider and receiver of the multimedia content. Indeed, it is often impossible to create a direct connection between them. The most common way to solve this problem is to use a TURN server since it is a network traffic relay protocol that assists the bypass of NATs and firewalls. In other words, a TURN server can be used to relay audio/video/data streams between peer devices.

### C. Data Structure

The structure of the data object storage in *5GOpenclasses* system is as presented in Table I. All passwords are encrypted in SHA-256 format. The *update\_time* attribute values are updated by the host machine's current date, in ISO 8601 format (e.g., 2020-02-19T13:58:26.00Z). The media files in the *5GOpenclasses* repository contain the path to their downloading and are associated with a past class session. These multimedia files can be stored in different formats, and the paths to download them are indicated in the *download\_path* attribute of the *Media\_Files* entity.

## IV. END-USER APPLICATIONS

### A. Mobile Application

We are currently using *webrtcpeer-Android* library to support WebRTC communications. As described in Section III-B, the

mobile application only receives real-time video content from the teacher since the communication architecture with the Kurento multimedia server is a One-to-Many video call. We can, therefore, identify that the stream comes only from one teacher, while there can be several students.

The communication of this video stream is of a one-way type, i.e. the stream goes from the source of the video (web application) to the student's video destination (mobile application). Besides the video streams of the indoors cameras, *5GOpenclasses* provides video stream contents from both UAV and *Bodykit* outdoors cameras, via a Janus WebRTC server. Janus is a server that implements extra functionalities that go beyond the features that we initially designed for WebRTC real-time communications. In the case of the mobile application, we implemented a classroom dedicated to the cameras where the video streams are received and presented.

For the purposes of the QoE assessment, students are invited to answer an electronic form that appears in the mobile application before they leave a class session. The form comprises two mandatory rating questions and a free-comment space. The data collected from the user's feedback are sent to the application server, more precisely, to the *Ratings* Entity in ORION. Both rates and comments are to feed an external service that performs Mean Opinion Scores (MOS) [10] and Topic modelling [11].

The registration/authentication process leverage an external service implemented by a partner from the project (IT Aveiro). This service is based on the Privacy-Preserving User Authentication Protocol (PPUAP) described in [12]. This process corresponds to the "Credentials Verifier" in Fig. 2.

### B. Web Application

We developed the web application in Python using the Dash by *Plotly* framework. The Dash by *Plotly* framework is an open-source framework dedicated to building data analytics web applications based on Flask, Plotly.js, and React.js.

TABLE I  
5GOPENCLASSES -DATA STRUCTURE

Entities	Attributes*
Users	id, type, name, email, password, is_online, in_class, TimeInstant, gyroscope, accelerometer, proximity, sound, light, create_time, update_time
Teachers	id, type, name, email, password, create_time, update_time
Rooms	id, type, password, name, description, is_online, recording, prof_id, course_id, create_time, update_time, location
Courses	id, type, academic_year, profs, create_time, update_time
Questions	id, type, asking, room_id, user_id, create_time, update_time
Group_Rooms	id, type, creator_id, password, create_time, update_time
Ratings	id, type, comment, rate, recommend, room_id, create_time, update_time
Media_Files	id, type, download_path, file_name, room_id, create_time, update_time
Environment	id, type, TimeInstant, light, noise, temperature

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

The construction of this end-user application relies on the use of a WebRTC Kurento server, supported by STUN and TURN servers. Consequently, we can stream video content in real-time leveraging the One-to-Many video calls architecture. We chose this server architecture because our target scenario is the provision of tools that enable the teacher to transmit their classes online and in real-time to their students.

For real-time video streaming purposes, a teacher can broadcast the contents of its computer screen such as slide presentations and his webcam. Finally, a teacher can manage its class sessions, share complementary materials, view the list of subjects assigned to him and answer the questions asked by students during class sessions. The data generated for these activities are stored on-the-fly, on the central databases.

#### V. TESTED FUNCTIONALITIES AND ASSESSMENT DESIGN

We have successfully tested the main functionalities of the *5GOpenclasses* listed in Table II, within a LAN environment. The multimedia results namely a video clip and the pertained screenshots, are available online at <https://bit.ly/393FR5c>.

The final assessment of the product is to be carried out in an integrated demonstration scenario displayed in Fig 4, that involves all the partners in the national project consortium. It is important to remind that the components of Fig 4 are those already described in Sections III and IV.

Table III shows the designed test plan for the integrated scenario where the main goal will be to compare the behaviour of the product when running the test cases OTT on both 4G to 5G networks simultaneously. We will split the tests into three types, namely:

- Integration with other partners of the project;
- Validation of the *5GOpenclasses* functionalities in a production-like scenario;
- Evaluation of the product from QoS viewpoint.

We plan to run the series of tests cases in Table III, with at least three iterations of 60 minutes each. The final evaluation will be carried out on fully-deployed mobile networks, we will leverage their Operation Support Systems (OSS) for tracking the selected QoS indicators [13].

Referring to the results from the OSS is the most reliable way for the selected indicators because these systems are developed to provide vital information about the activities in the networks. With the values from the OSS, we will be able to draw comparative charts and visually conclude the performances of the product on both networks, i.e., 4G and 5G.

#### VI. CONCLUSION

In this paper, we presented the design features of a 5G vertical product for blended learning environments, developed in the scope of a national 5G project. The main objective of the paper was to research, develop, validate, integrate and demonstrate both mechanisms and solutions that explore the functionality of future 5G networks for supporting human and IoT communication, targeting all the use cases envisioned for 5G. We have also proposed the design of innovative location-based service for blended learning environments.

In this scope we presented the main product features and the choice of implementation tools for the proposed product. We adopted a bottom-up approach to design and develop the proposed product, to assess its behavior when running OTT on both 4G and 5G networks that are being set up by the project consortium. The next step will be to carry out the final integrated product validation, which will involve all the partners from the project.

The other future works consist of implementing the location-based service, to assess the ability of the network to deal with the stringent real-time requirements.

TABLE II  
TESTED FUNCTIONALITIES

#	Tested functionalities	Comments
1	Secure Signup and Sign in for the mobile application	Via PPUAP from IT Aveiro.
2	Room creation	As soon as the class session is created, students may securely log to it.
3	Starting a class session	This functionality is the one that opens the full panel for a teacher to manage a class session, e.g. see the asked questions or its own stream feed.
4	Student registration and authenticated login	This is the only way students can access the functionalities of the mobile application
5	Authenticated access to classes and video streams reception	The student can select different views from the array of available streams of a class session
6	Interventions to classes	All students attending the class session can visualize the intervention of the colleagues.
7	Management of the repository by the teacher (feeding, renaming, suppression)	The repository supports different digital formats (e.g. .doc, .ppt., .avi, .mp3)
8	Resource consumption du repository	The available materials can be downloaded or played directly from the repository.
9	QoE questionnaire triggering and answering	The questionnaire is triggered at the end of each class session.
10	Questionnaire data collection by the server	The results of the questionnaires are sent to the backend App server (FIWARE – Orion).
11	Integrated Video sources	Video and audio from Drones and Bodykit. Ready for the integrated demonstration
12	Collect and display mobile sensor data	The UE must be provided with the required sensors hardware
13	Collect and display stationary sensor data	IoT Boxset developed by the University of Coimbra

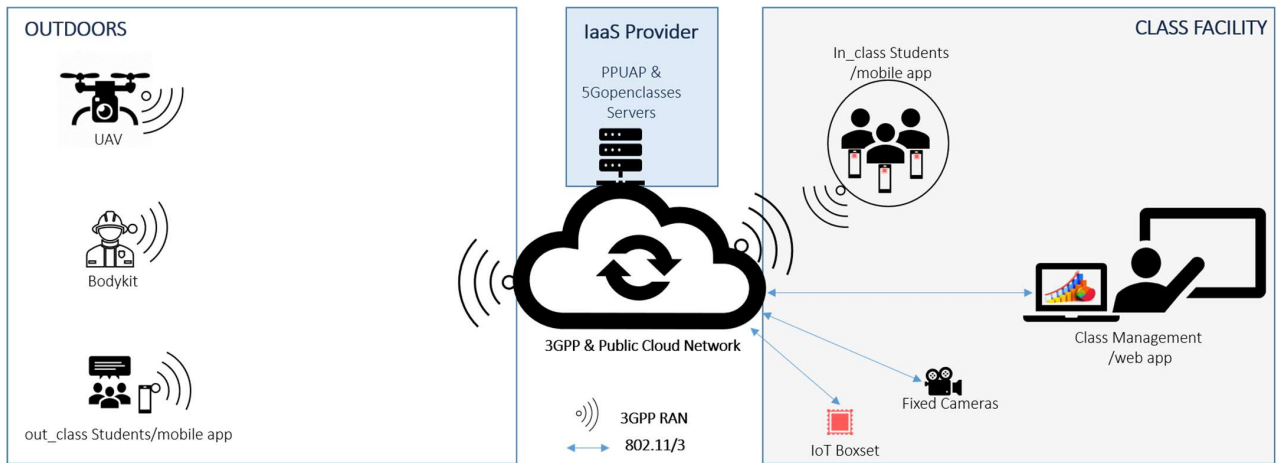


Fig. 4. Integrated demonstration scenario

TABLE III  
 RUN TESTS AND INDICATORS

#	Test Cases	Test Type*	QoS Indicators
1	From the web application: Creation of a room with 3 video streams (Fixed Camera, UAV and Bodykit)	i, ii	Not applicable (Fully LAN-based communication)
2	From the mobile application: Registration via PPUAP	i, ii, iii	Latency
3	From the mobile application: Access to a room and consumption of video streaming	i, ii, iii	Latency, Jitter, Loss, Bit Rate & Play Rate, filling in The Buffer, Post-Buffer Lag Length, Play Length of Video, Lag Ratio of Video
4	From the web application: Upload an HD video to the repository	ii, iii	Not applicable (Fully LAN-based communication)
5	From the mobile application: Direct consumption of the video from the repository	ii, iii	idem #4
6	From the mobile application: Download the video from the repository	ii, iii	Latency
7	From the UE: Upload sensor data	ii, iii	Latency, Loss, Bit Rate
8	IoT Boxset: Upload sensor data	ii, iii	Not applicable (Fully LAN-based communication)
9	From the mobile application and middleware: Monitor the <i>in_class</i> toggle and its effect on the received mobile sensors values	ii, iii	Latency

\* i) integration, (ii) validation and (iii) evaluation

## VII. REFERENCES

- [1] ITU-T Study Group 5 (SG 5), "M.2083 : IMT Vision - 'Framework and overall objectives of the future development of IMT for 2020 and beyond,'" 2015. [Online]. Available: <https://www.itu.int/rec/R-REC-M.2083>. [Accessed: 10-Jul-2020].
- [2] R. Boelens, B. De Wever, and M. Voet, "Four key challenges to the design of blended learning: A systematic literature review," *Educ. Res. Rev.*, vol. 22, pp. 1–18, Nov. 2017.
- [3] Y. K. Ever and A. V. Rajan, "The Role of 5G Networks in the Field of Medical Sciences Education," in *IEEE 43rd Conference on Local Computer Networks Workshops (LCN Workshops)*, 2018, pp. 59–63.
- [4] L. Zhongmei, H. Yu-Che, and C. Bangjun, "A Study on the Effects of Distance Learning and the Application of 5G Technolog.," in *IEEE 11th International Conference on Advanced Infocomm Technology (ICAIT)*, 2019, pp. 218–222.
- [5] A. Baratè, G. Haus, L. A. Ludovico, E. Pagani, and N. Scarabottolo, "5G Technology for Augmented and Virtual Reality in Education," in *Education and New Developments International Conference on Education and New Developments*, 2019, pp. 512–516.
- [6] M. Héder, "From NASA to EU: The evolution of the TRL scale in Public Sector Innovation," *Innov. J. Public Sect. Innov. J.*, vol. 22, no. 2, pp. 1–23, 2017.
- [7] J. Fernandes, D. Raposo, N. Armando, S. Sinche, J. Sá Silva, A. Rodrigues, V. Pereira, H. Gonçalo Oliveira, Luís Macedo, F. Boavida, "ISABELA – A Socially-Aware Human-in-the-Loop Advisor System," *Online Soc. Networks Media*, vol. 16, p. 100060, Mar. 2020.
- [8] S. Sinche, P. Hidalgo, J. Fernandes, D. Raposo, J. Sá Silva, A. Rodrigues, N. Armando, F. Boavida, "Analysis of Student Academic Performance Using Human-in-the-Loop Cyber-Physical Systems," *Telecom*, vol. 1, no. 1, pp. 18–31, Mar. 2020.
- [9] S. Krco, B. Pokric, and F. Carrez, "Designing IoT architecture(s): A European perspective," in *2014 IEEE World Forum on Internet of Things, WF-IoT 2014*, 2014, pp. 79–84.
- [10] L. Skorin-Kapov, M. Varela, T. Hößfeld, and K.-T. Chen, "A Survey of Emerging Concepts and Challenges for QoE Management of Multimedia

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

Services,” *ACM Trans. Multimed. Comput. Commun. Appl.*, vol. 14, no. 2s, pp. 1–29, May 2018.

[11] S. Arora *et al.*, “Learning topic models -- provably and efficiently,” *Commun. ACM*, vol. 61, no. 4, pp. 85–93, Mar. 2018.

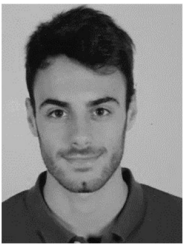
[12] V. Sucasas, G. Mantas, A. Radwan, and J. Rodriguez, “A Lightweight Privacy-Preserving OAuth2-Based Protocol for Smart City Mobile Apps,” in *IEEE Globecom Workshops (GC Wkshps)*, 2016, pp. 1–6.

[13] D. Pal and V. Vanijja, “Effect of network QoS on user QoE for a mobile video streaming service using H.265/VP9 codec,” *Procedia Comput. Sci.*, vol. 111, pp. 214–222, 2017.



**NGOMBO ARMANDO** received the M.Eng. degree in information and communication technologies from IMT Lille-Douai, France, in 2010. He is currently a PhD candidate in information science and technology at the University of Coimbra, Department of Informatics Engineering (DEI), Coimbra, Portugal. He is a Member of IEEE. From 2010 to 2012, he was a telecoms consultant for Altran-T&M, Paris, France. Since 2013, he is a senior lecturer at Universidade Kimpa Vita, Escola Superior Politécnica do Uíge (ESPU), Uíge, Angola. At ESPU, he also heads the

project management service in charge of both IT management and support, statistics and working good practices. Since 2018 he is a research fellow for 5G products at the Centre of Informatics Engineering of University of Coimbra (CISUC), Coimbra, Portugal. His research interests include IoT, Devices Management, 5G products and services. (<https://www.cienciavita.pt/en/5916-4A11-E165>)



**RUI ALMEIDA** is a master’s degree candidate in Informatics Engineering at Science and Technology at the University of Coimbra, DEI, Coimbra, Portugal. During his final project stage, he joined the Laboratory of Communication and Telematics (LCT) of CISUC, Coimbra, Portugal, as a research fellow. At LCT, he has been developing on enhanced dashboards Human-in-The-Loop Cyber-Physical System (HiLCPS). His research topics include IoT, device management, and web-based dashboards for IoT systems.



**JOSÉ MARCELO FERNANDES** received his master’s degree in Biomedical Engineering, with specialisation in Bioinformatics and Clinical Informatics, in 2017 by the University of Coimbra, Portugal. He has since enrolled in the Doctoral Program in Information Science and Technology, in the same University. He simultaneously works as a Researcher at the LCT of the CISUC, Portugal. His main research interests are HiLCPS, IoT, Non-obtrusive sensing, Mobile sensing, Bioinformatics and Informatic Systems.

(<https://www.cienciavita.pt/en/F216-7C1A-36E3>)



**JORGE SÁ SILVA** received his PhD in Informatics Engineering in 2001 from the University of Coimbra, where is Associate Professor with Habilitation at the Department of Electrical and Computer Engineering of Sciences and Technology of the University of Coimbra and a Senior Researcher of LCT, Portugal. His main research interests are IoT, Network Protocols, M2M, and WSNs. He has been serving as a reviewer and publishing in top conferences and journals in his expertise areas. His publications include 2 books, 5 book chapters and over 170 papers

in refereed national and international conferences and magazines. He participated in European initiatives and projects such as FP5 E-NET, FP6 NoE E-NEXT, FP6 IP EuQoS, FP6 IP WEIRD and FP7 Ginseng (as Portuguese Leader). He actively participated in the organisation of several international conferences and workshops, (e.g. he was the Workshop Chair of IFIP

Networking2006, Publicity Chair of EWSN2009, General Co-Chair of EWSN2010, General Co-Chair of Mobiquitous2015, General Vice-Chair of WoWMoM2016) and he was also involved in program committees of national and international conferences. He is a senior a Member of IEEE, and he is a licensed Professional Engineer. (<http://www.dei.uc.pt/sasilva>)



**FERNANDO BOAVIDA** received his PhD in Informatics Engineering in 1990, and he currently is Full Professor at the DEI of the Faculty of Sciences and Technology of the University of Coimbra. His main research interests are people-centric Internet of Things, wireless sensor networks, mobility, and quality of service. He is author/co-author of more than 170 international publications (books, book chapters, refereed journals and conference proceedings) and 50 national publications. He was the chairman of the Program Committee of QoS’2001, IDMS-PROMS’2002, NETWORKING 2006, WWIC 2007, FMN 2008, EWSN 2010, FMN 2012, IWQoS 2012, ACM SIGCOMM FhMN 2013, Mobiquitous 2015, and WoWMoM 2016 international conferences/ workshops. He is a Senior Member of the IEEE and a licensed Professional Engineer. He is a member of the Editorial Advisory Board of the Computer Communications journal. (<http://www.uc.pt/go/boavida>)