



Universidade do Minho
Escola de Engenharia

Laboratórios de Telecomunicações e Informática II

ENGENHARIA DE TELECOMUNICAÇÕES E INFORMÁTICA
2020/2021

(Docentes: Bruno Daniel Mestre Viana Ribeiro, Vadym Serhiyovych Hapanchak,
José Augusto Afonso, Sérgio Adriano Fernandes Lopes)

15 de junho de 2021

Relatório

FASE C

Rui Filipe Ribeiro Freitas – a84121@alunos.uminho.pt

Sandro Teixeira Ribeiro – a85316@alunos.uminho.pt

Tiago João Pereira Ferreira – a85392@alunos.uminho.pt

Índice

Índice de figuras	3
Índice de tabelas.....	4
Lista de abreviaturas	5
Introdução	6
1. Planeamento do projeto	7
1.1. Planeamento temporal	7
1.2. Tecnologias/Ferramentas necessárias.....	8
1.2.1. Ao nível do <i>hardware</i>	8
1.2.2. Ao nível do <i>software</i>	9
2. Fundamentos.....	10
2.1. Base de dados	10
2.2. Gestor de serviços.....	11
2.3. Sistema central	11
2.4. MQTT	12
3. Implementação.....	13
3.1. Front-end	13
3.2. Back-end	16
3.3. Base de dados relacional	18
Fase Extra.....	19
Conclusão.....	20
Referências	21
Autoavaliação	22

Índice de figuras

Figura 1 - Planeamento temporal - tabela.	7
Figura 2 - Planeamento temporal - gráfico.	7
Figura 3 - Diagrama de Entidades e Relacionamentos.	10
Figura 4 - Protocolo MQTT.	12
Figura 5 - Página de login.	13
Figura 6 - Página de registo.	13
Figura 7 - Página do dashboard do utilizador.	14
Figura 8 - Página do sistema sensor.	14
Figura 9 - Página do administrador.	15
Figura 10 - Página para a adição de um concentrador.	15
Figura 11 - Excerto de código com as rotas.	16
Figura 12 - Função do login.	17
Figura 13 - Função para a visualização dos sistemas sensores.	17
Figura 14 - Tabelas da base de dados.	18
Figura 15 - Conteúdo da tabela AmostraLux.	18

Índice de tabelas

Tabela 1 - Tecnologias ao nível do hardware.	8
Tabela 2 - Tecnologias ao nível do software.	9

Lista de abreviaturas

LTI II – Laboratórios de Telecomunicações e Informática II

MQTT – *Message Queuing Telemetry Transport*

Introdução

No âmbito da Unidade Curricular de LTI II (Laboratórios de Telecomunicações e Informática II) foi-nos proposto o desenvolvimento de um sistema de iluminação inteligente que permita monitorizar a ocupação e a luminosidade em diferentes áreas de uma residência, bem como controlar a luminosidade desses ambientes através do acionamento de lâmpadas.

Este relatório diz respeito à terceira fase de 3 do desenvolvimento deste projeto em que nesta última fase os objetivos passam pela elaboração de uma base de dados relacional para o nosso projeto, implementação de uma interface web para o sistema central e detalhes das funcionalidades a implementar no terminal de utilizadores. Para além disto este relatório contém também a implementação da fase extra com os detalhes das funcionalidades a implementar na aplicação MQTT (*Message Queuing Telemetry Transport*) assim como detalhes sobre as bibliotecas MQTT a serem utilizadas nos vários componentes: Concentrador, Aplicação MQTT e Sistemas Sensores.

De modo a sermos capazes de cumprir com os objetivos deste projeto semestral devemos colocar em prática conhecimentos adquiridos noutras unidades curriculares, nomeadamente Redes de Computadores, Sistemas Operativos, Sistemas Distribuídos, Eletrónica e Laboratórios de Telecomunicações e Informática I.

1. Planeamento do projeto

1.1. Planeamento temporal

De modo que o grupo se mantenha focado no trabalho e com um compromisso para cumprir horários, resolvemos planear as tarefas a fazer nesta fase. Na figura 1 observamos em forma de tabela os vários assuntos a ser tratados nesta fase com um período dado por nós para cumprir. Em relação à figura 2 demonstramos em forma de gráfico o tempo despendido nas várias tarefas. Ambas as figuras foram retiradas do programa *Gantt*.

		
Nome	Data de início	Data de fim
☐ • Fase C	18-05-2021	14-06-2021
• Planeamento Temporal	18-05-2021	20-05-2021
• Definição da Arquitetura do Sistema	19-05-2021	19-05-2021
• Início da elaboração do relatório	20-05-2021	20-05-2021
• Desenvolvimento código Da Base de Dados	24-05-2021	26-05-2021
• Desenvolvimento código Sistema Central	27-05-2021	09-06-2021
• Realização de Testes	10-06-2021	10-06-2021
• Discussão dos resultados obtidos	11-06-2021	11-06-2021
• Conclusão e revisão do relatório	14-06-2021	14-06-2021

Figura 1 - Planeamento temporal - tabela.

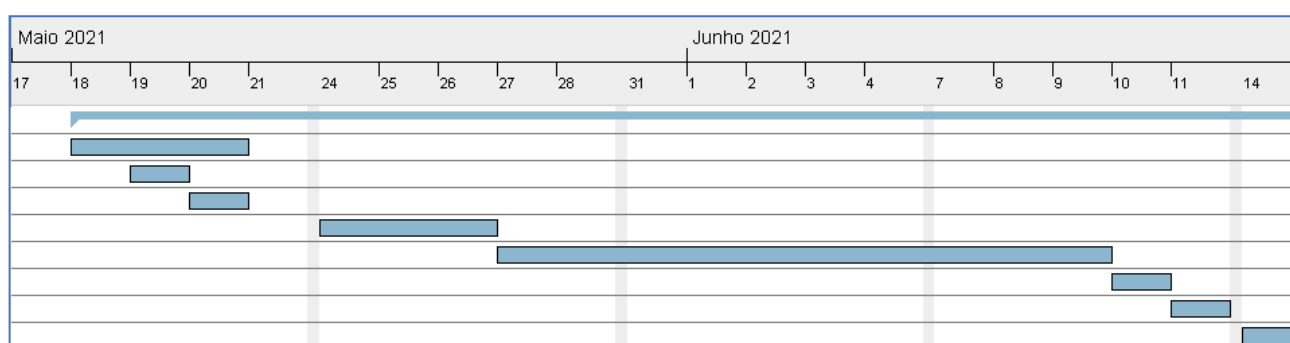


Figura 2 - Planeamento temporal - gráfico.

1.2. Tecnologias/Ferramentas necessárias

Em qualquer projeto são necessárias certas tecnologias/ferramentas que nos facilitem o trabalho e ao mesmo tempo que aumentem a nossa produtividade. Para este trabalho em específico teremos de recorrer a tecnologias específicas tanto ao nível do *hardware* como ao nível do *software*, de modo a cumprir com os objetivos propostos.

1.2.1. Ao nível do *hardware*

No que toca à parte do *hardware* deste nosso trabalho as tecnologias por nós utilizadas estão presentes na tabela 1.

FERRAMENTA	QUANTIDADE	UTILIZAÇÃO
Computador Portátil	1	Desenvolvimento de código, elaboração do relatório, pesquisa e testes
Placa ESP32	1	Comunicação com os sensores e os LEDs
Sensor de movimento (PIR HC-SR501)	1	Deteção de movimento
Sensor de luminosidade (LDR 5...10kΩ)	1	Medição da luminosidade
LED	3	Simulação de lâmpadas
Cabo Micro USB	1	Ligações físicas entre o PC e a placa ESP32
Fios de ligação	Vários	Efetuar as ligações entre as placas ESP32 e os sensores e LEDs
Resistência	Várias	Limitar a corrente drenada pelos LEDs

Tabela 1 - Tecnologias ao nível do *hardware*.

1.2.2. Ao nível do *software*

Em relação ao *software* que irá ser utilizado na execução do nosso projeto este passará maioritariamente pelas aplicações apresentadas na tabela 2. Nesta realçamos as mais importantes e fundamentais, sabendo que, no entanto, poderão sofrer alterações no futuro dado ainda nos encontrarmos numa fase mais introdutória do trabalho.

LOGO	DESCRIÇÃO	UTILIZAÇÃO
	Microsoft Word	Realização e sincronização do relatório do projeto
	GanttProject	Planeamento do projeto através de diagramas de Gantt
	Arduino IDE	Software para a conexão com a placa ESP32 e comunicação com o mesmo
	Visual Studio Code	Execução e compilação de código
	GitHub	Sincronização do código da aplicação
	Visual Paradigm	Implementação de fluxogramas para a elaboração da aplicação e do projeto
	Google Drive	Sincronização de ficheiros essenciais à realização do trabalho
	Discord	Comunicação entre o grupo
	Google Chrome	Browser da interface Web
	MySQL Workbench	Criação e gestão da base de dados

Tabela 2 - Tecnologias ao nível do *software*.

2. Fundamentos

2.1. Base de dados

De modo a armazenar todas as informações enviadas pelos vários concentradores, iremos elaborar uma base de dados para guardar todos esses dados numa forma mais eficiente e mais prática aos olhos do utilizador. O grupo decidiu implementar a base de dados em linguagem MySQL e o planeamento inicial da mesma num diagrama de entidades e relacionamentos é apresentado na figura seguinte.

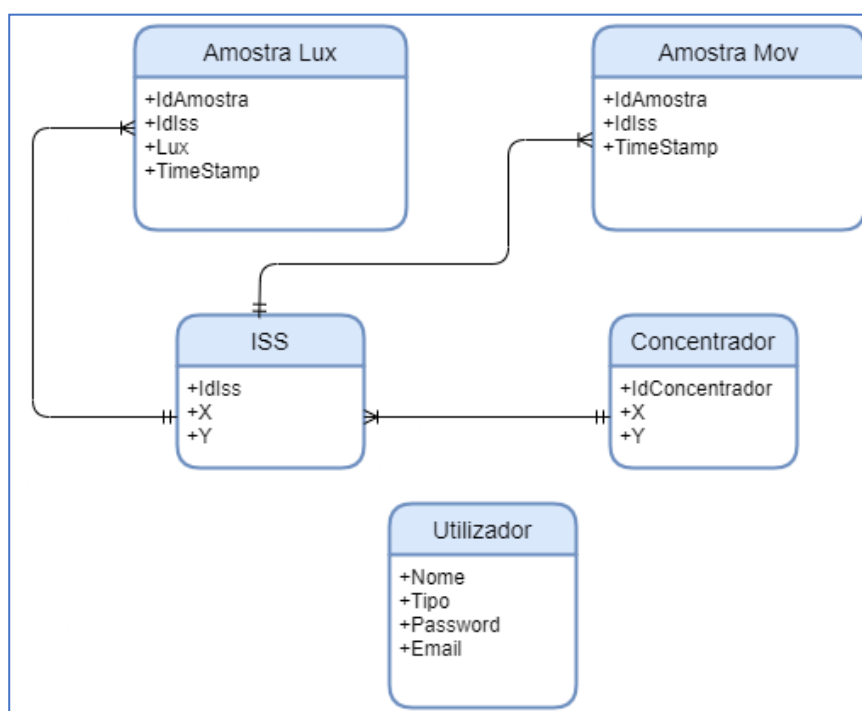


Figura 3 - Diagrama de Entidades e Relacionamentos.

Esta base de dados será constituída por 5 tabelas, uma para as amostras de luminosidade, outra para as amostras de movimento, para o sistema sensor, concentrador e por último a tabela do utilizador que conterà todos os utilizadores e o administrador da aplicação.

2.2. Gestor de serviços

O gestor de serviços é responsável pela recolha dos dados provenientes dos vários concentradores e guardá-los numa base de dados. Na fase passada utilizamos ficheiros CSV para guardar estes dados, no entanto visto ser necessária uma base de dados relacional decidimos implementar esta em MySQL. Nesta fase temos de implementar a escrita na base de dados dos vários dados retirados dos concentradores para que o sistema central possa aceder a estes mesmos dados de modo a mostrá-los ao utilizador na interface web.

2.3. Sistema central

O foco desta fase C para além da base de dados relacional passa pela implementação do sistema central, que é o responsável pela gestão e comunicação entre a base de dados e o utilizador. Este sistema será acedido pelos vários utilizadores através de um *browser*. É apresentado ao utilizador uma interface de login e/ou registo em que após autenticar a sua conta será fornecida informação acerca dos sistemas sensores pertencentes. Poderão ainda ser adicionadas funcionalidades extra que o grupo ache relevantes.

2.4. MQTT

Na fase extra deste projeto é sugerido implementar um protocolo de comunicação com baixos requisitos tanto ao nível de largura de banda como ao nível de hardware, o que faz com que este seja simples e leve ou seja, perfeito para internet das coisas (IoT).

Este protocolo é baseado num protocolo da camada de transporte, o TCP/IP, que não nos é desconhecida a sua implementação. O MQTT é composto por um MQTT Broker, um *publisher* e um *subscriber*. O Broker é responsável por receber e transmitir mensagens recebidas dos publishers para os *subscribers*.

Na figura seguinte é apresentado um esquema que explica mais detalhadamente como funciona este protocolo, com um *publisher* a enviar mensagens para o Broker que as transmite para os *subscribers* [1].

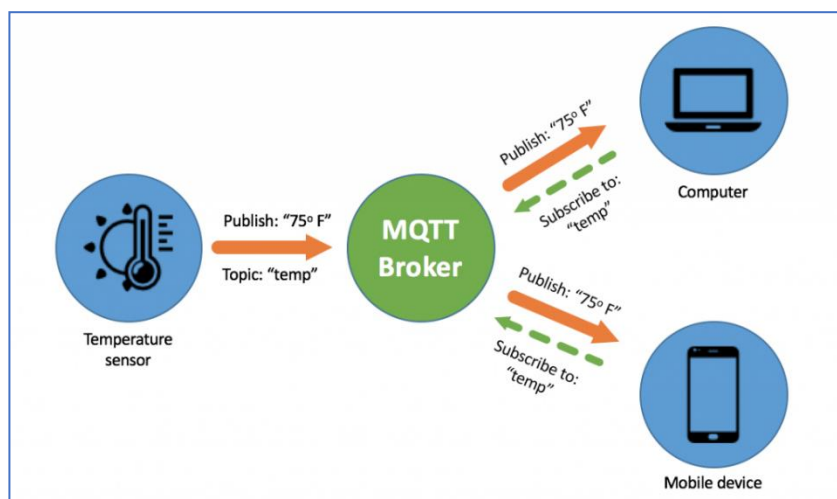


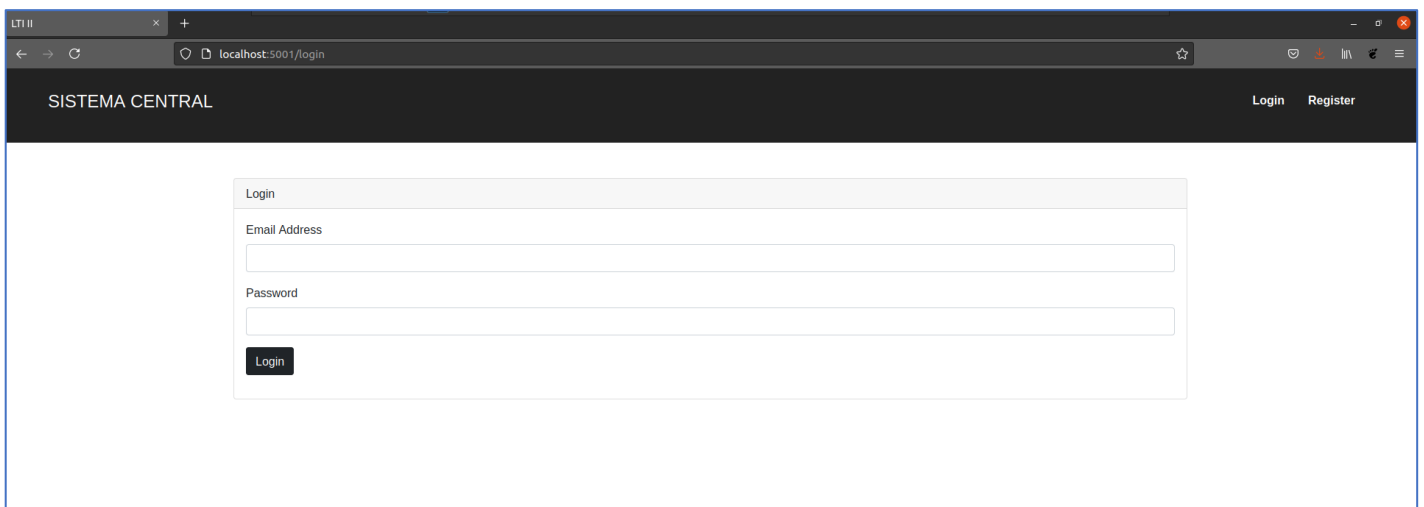
Figura 4 - Protocolo MQTT.

3.Implementação

3.1. Front-end

Inicialmente começamos pelo desenvolvimento da interface gráfica do nosso sistema central onde demostramos nas seguintes figuras a sua implementação.

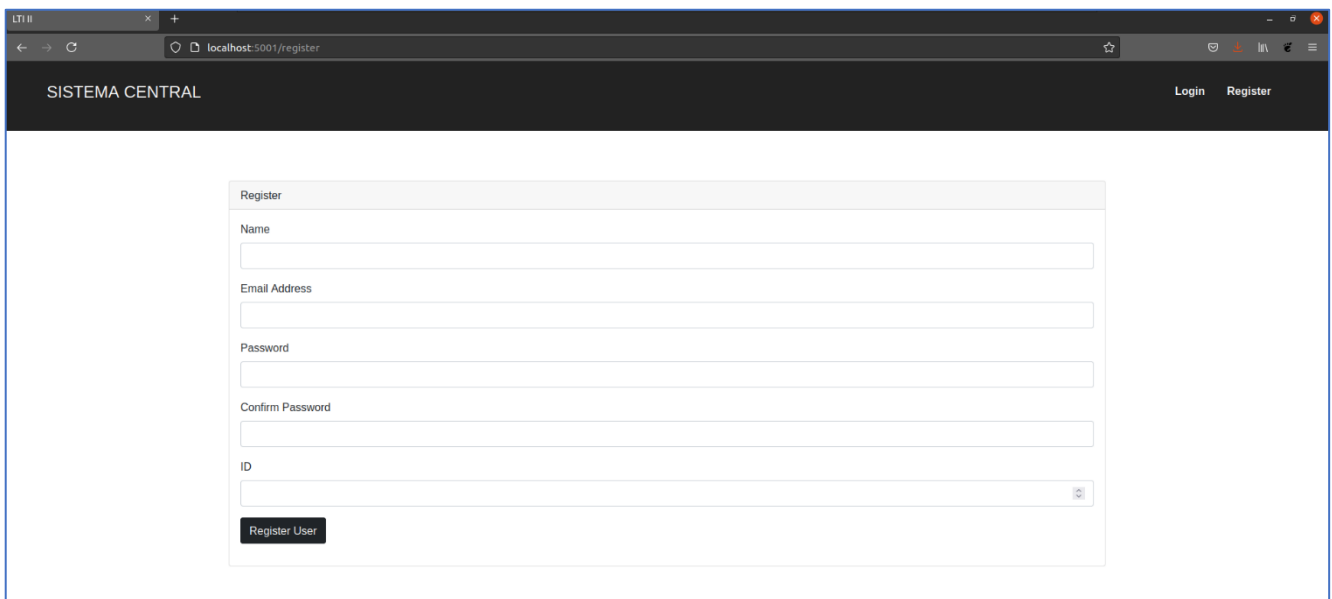
Na figura 5 é possível observar a página de login de um utilizador onde este acede à sua conta se tiver uma conta já registada e é realizado o acesso às várias funcionalidades do sistema central.



The screenshot shows a web browser window with the address bar displaying 'localhost:5001/login'. The page has a dark header with the text 'SISTEMA CENTRAL' on the left and 'Login Register' on the right. The main content area features a light gray box titled 'Login'. Inside this box, there are two input fields: 'Email Address' and 'Password'. Below the 'Password' field is a dark gray button with the text 'Login' in white.

Figura 5 - Página de login.

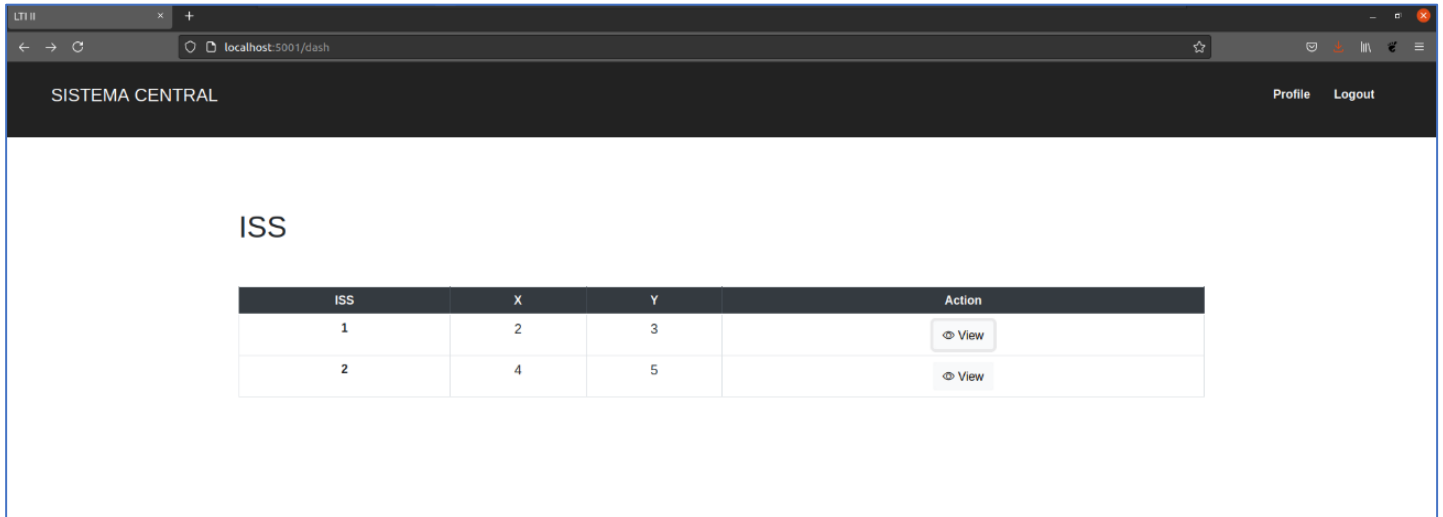
De modo a realizar o registo de um utilizador este tem de indicar qual o seu nome, email, password e ID do concentrador que lhe diz respeito. Estas informações são fornecidas na página de registo apresentada na figura seguinte.



The screenshot shows a web browser window with the address bar displaying 'localhost:5001/register'. The page has a dark header with the text 'SISTEMA CENTRAL' on the left and 'Login Register' on the right. The main content area features a light gray box titled 'Register'. Inside this box, there are five input fields: 'Name', 'Email Address', 'Password', 'Confirm Password', and 'ID'. Below the 'ID' field is a dark gray button with the text 'Register User' in white.

Figura 6 - Página de registo.

Após ser realizada a autenticação do utilizador é apresentada uma página com a informação acerca dos sistemas sensores que lhe dizem respeito. Na figura 7 é apresentado ao utilizador a tabela de sensores onde este pode escolher qual ISS quer observar mais informações assim como as posições geográficas deste.



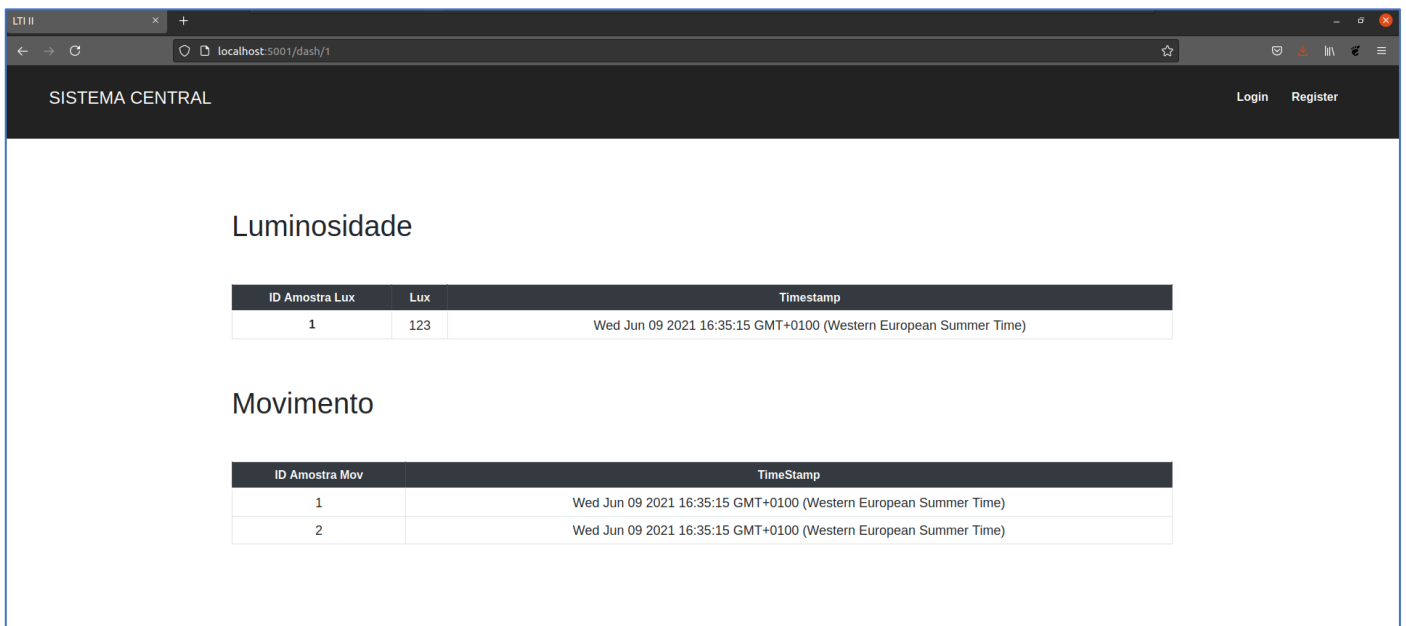
SISTEMA CENTRAL Profile Logout

ISS

ISS	X	Y	Action
1	2	3	View
2	4	5	View

Figura 7 - Página do dashboard do utilizador.

A figura 8 corresponde à visualização da página que é apresentada ao utilizador quando este clica no botão *View* onde são apresentadas 2 tabelas, uma relativa ao sensor de luminosidade com os valores do id da amostra, dos luxs calculados e do timestamp da amostra e outra tabela que diz respeito ao sensor de movimento com informações sobre o id da amostra e o timestamp da amostra.



SISTEMA CENTRAL Login Register

Luminosidade

ID Amostra Lux	Lux	Timestamp
1	123	Wed Jun 09 2021 16:35:15 GMT+0100 (Western European Summer Time)

Movimento

ID Amostra Mov	TimeStamp
1	Wed Jun 09 2021 16:35:15 GMT+0100 (Western European Summer Time)
2	Wed Jun 09 2021 16:35:15 GMT+0100 (Western European Summer Time)

Figura 8 - Página do sistema sensor.

Relativamente ao administrador do sistema central, este ao realizar o seu login é redirecionado para uma página diferente das vistas anteriormente onde é possível editar e remover utilizadores assim como adicionar, editar e remover concentradores como demonstrado na figura seguinte. Para além disso dispõe de uma barra de procura de modo a encontrar mais facilmente um utilizador ou um concentrador.

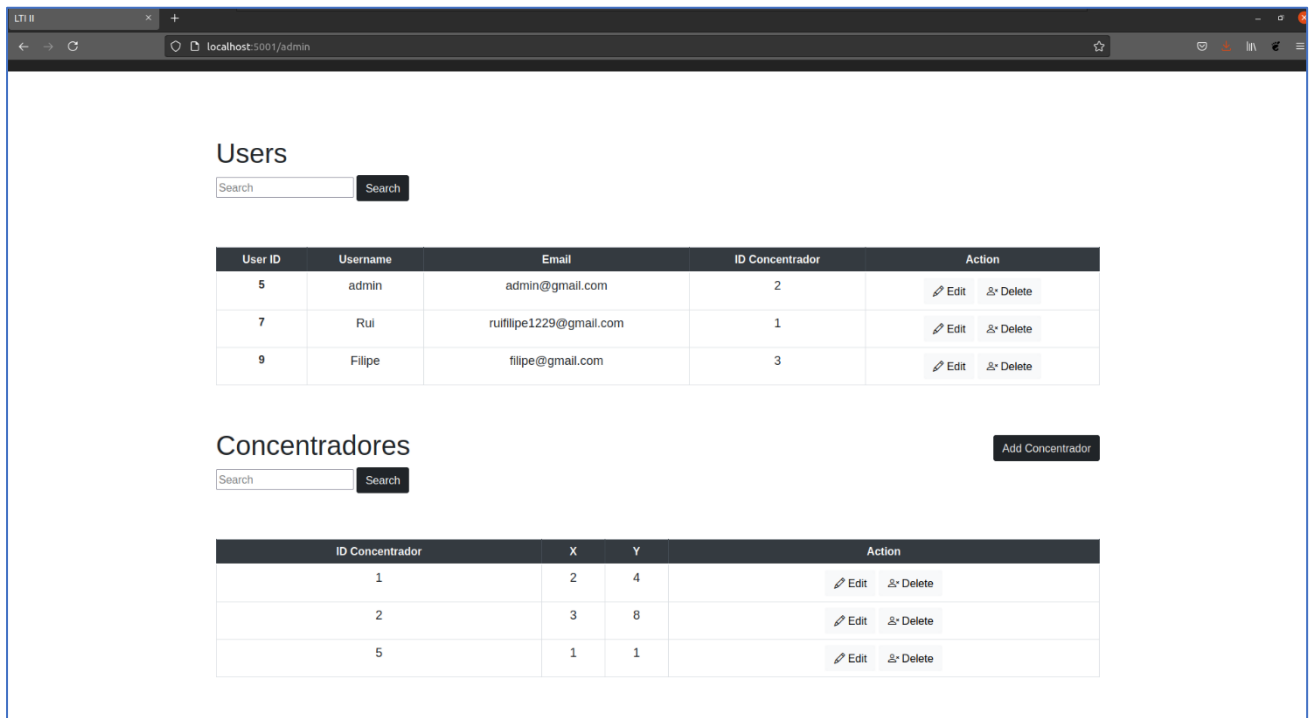


Figura 9 - Página do administrador.

Para adicionar um concentrador é apresentado ao administrador um formulário como ilustrado na figura 10.

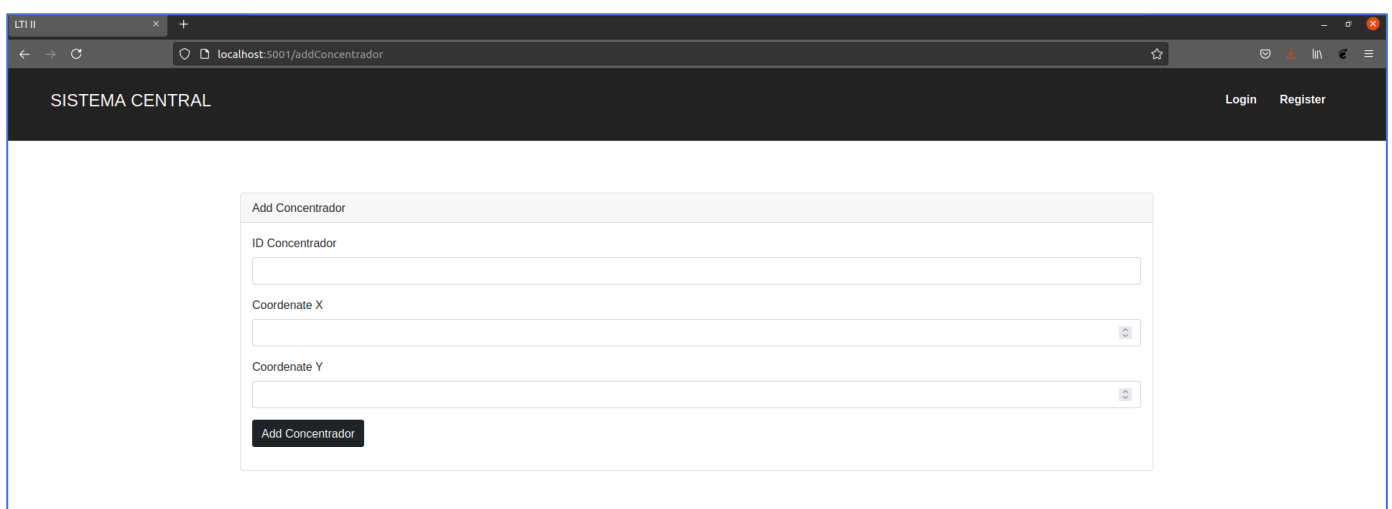


Figura 10 - Página para a adição de um concentrador.

3.2. Back-end

Depois de explicada a implementação ao nível do *front-end* vamos agora detalhar a parte que gere a receção de dados. O *back-end* implementado pelo grupo é composto por um serviço web com recurso a node JS [2] e através da utilização da framework Express [3].

Os principais ficheiros que dizem respeito ao *back-end* da nossa aplicação são a classe *app.js* onde é iniciado o servidor na porta 5001 e dentro das pastas *routes* onde é realizada a distribuição das várias rotas através dos ficheiros *auth.js* e *pages.js*.

De seguida é apresentada um excerto do código com as rotas realizadas na nossa aplicação onde foram utilizados os métodos GET e POST.

```
router.get("/", authController.isLoggedIn, (req, res) => {
  res.render("index", {
    user: req.user,
  });
});

router.get("/register", (req, res) => {
  res.render("register");
});

router.get("/login", (req, res) => {
  res.render("login");
});

router.get("/profile", authController.isLoggedIn, (req, res) => {
  console.log(req.user);
  if (req.user) {
    res.render("profile", {
      user: req.user,
    });
  } else {
    res.redirect("/login");
  }
});

router.get('/admin',authController.view);
router.get('/dash', authController.viewISS);
router.post('/admin',authController.find);
router.get('/edit/:id',authController.edit);
router.post('/edit/:id',authController.update);
router.get('/:dash/:id', authController.viewIdISS);
router.get('/addConcentrador', authController.form);
router.post('/addConcentrador',authController.addConcentrador);
router.get('/editConcentradores/:id',authController.editConcentradores);
router.post('/editConcentradores/:id',authController.updateConcentradores);
router.get('/:id',authController.delete);
router.get('/remove/:id',authController.deleteConcentradores);

module.exports = router;
```

Figura 11 - Excerto de código com as rotas.

A classe responsável por realizar as várias funcionalidades das rotas está presente na pasta *controllers* no ficheiro *auth.js*. Na figura seguinte apresentamos um exemplo ilustrativa de uma das funções onde é realizado o *login* depois de um acesso à base de dados e na figura 13 a função para a visualização dos vários sistemas sensores.

```
exports.login = async (req, res) => {
  try {
    const { email, password } = req.body;
    if (!email || !password) {
      return res.status(400).render("login", {
        message: "Please provide an email and password",
      });
    }
    db.query(
      "SELECT * FROM accounts WHERE email = ?",
      [email],
      async (error, resul) => {
        console.log(resul);
        if (!resul || !(await bcrypt.compare(password, resul[0].password))) {
          return res.status(400).render("login", {
            message: "Email or Password is incorrect",
          });
        } else {
          const id = resul[0].id;
          const token = jwt.sign({ id }, process.env.JWT_SECRET, {
            expiresIn: process.env.JWT_EXPIRES_IN,
          });
          console.log("The token is: " + token);
          const cookieOptions = {
            expires: new Date(
              Date.now() + process.env.JWT_COOKIE_EXPIRES * 24 * 60 * 60 * 1000
            ),
            httpOnly: true,
          };
          res.cookie("jwt", token, cookieOptions);
          if (email === "admin@gmail.com" && await bcrypt.compare(password, resul[0].password)) {
            res.status(200).redirect("/admin");
          } else {
            res.status(200).redirect("/dash");
          }
        }
      }
    );
  } catch (error) {
    console.log(error);
  }
};
```

Figura 12 - Função do login.

```
exports.viewISS = async (req, res, next) => {
  // console.log(req.cookies);
  let idUser = 0;
  if (req.cookies.jwt) {
    try {
      //1) verify the token
      const decoded = await promisify(jwt.verify)(
        req.cookies.jwt,
        process.env.JWT_SECRET
      );
      idUser = decoded.id;
      console.log(decoded);
      //2) Check if the user still exists
      db.query(
        "SELECT * FROM accounts WHERE id = ?",
        [decoded.id],
        (error, result) => {
          console.log(result);
          req.user = result[0];
          console.log("user is");
          console.log(req.user);
        }
      );
    } catch (error) {
      console.log(error);
    }
  }

  db.query('SELECT * FROM Iss WHERE idConcentrador = (SELECT idConcentrador FROM accounts WHERE id = ?)', [idUser], (err, results) => {
    if (!err) {
      console.log("-----");
      console.log(results);
      console.log("-----");
      res.render("dash", { user: req.user, results });
    } else {
      console.log(err);
    }
  });
};
```

Figura 13 - Função para a visualização dos sistemas sensores.

3.3. Base de dados relacional

A base de dados implementada pelo grupo é constituída por 5 tabelas, a tabela accounts que diz respeito à base de dados dos utilizadores registados, a tabela AmostraLux onde estão contidas as amostras do sensor de luminosidade, a tabela AmostraMov com os valores do sensor de movimento, a tabela Concentrador com a informação da posição geográfica e dos utilizadores com acesso a certo concentrador e por fim a tabela lss com a posição geográfica e o id do concentrador. Para a realização disto foi utilizada a linguagem MySQL com o auxílio do MySQL Workbench onde na figura 14 podemos observar as tabelas já implementadas nesta aplicação.

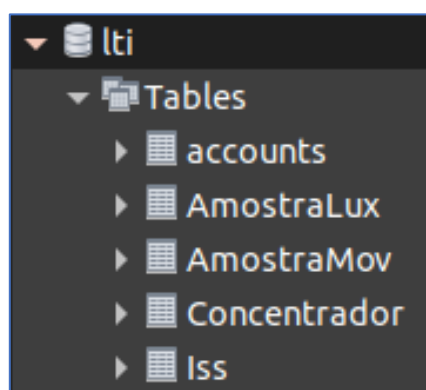


Figura 14 - Tabelas da base de dados.

De seguida apresentamos o conteúdo da tabela AmostraLux com o objetivo de exemplificar o resultado da inserção de dados através do MySQL Workbench.

#	idAmostraLux	idlss	idConcentrador	lux	timeStamp
1	1	1	1	123	2021-06-09 16:35:15
2	2	5	1	323	2021-06-09 16:35:15
3	3	2	1	234	2021-06-09 18:35:15
*	NULL	NULL	NULL	NULL	NULL

Figura 15 - Conteúdo da tabela AmostraLux.

Fase Extra

Relativamente à fase extra, infelizmente esta não foi implementada pois deparamo-nos com obstáculos na realização desta fase principalmente na utilização de programação web pois era nos desconhecido este tipo de programação e tivemos de aprender os vários aspetos desta programação num curto período de tempo.

Devido a estas dificuldades encontradas não conseguimos realizar a fase extra, no entanto achamos que se tivéssemos resolvidos certos problemas mais rápido teríamos conseguido realizar a sua implementação com sucesso.

De modo a realizar a implementação desta fase era necessária a utilização de uma biblioteca no Arduino IDE, neste caso a biblioteca MQTT de modo que fosse possível a placa ESP32 aceder à internet e receber mensagens de publishers de modo a ligar/desligar um LED por exemplo. Como esta funcionalidade já foi implementada na fase B também achamos não haver uma necessidade de ser implementada novamente. Após realizada a conexão da biblioteca e feito o código correspondente era necessário conectar à interface web através de código em C.

Uma vez mais achamos que podíamos ter implementado esta fase extra, no entanto o tempo tornou-se escasso devido a obstáculos encontrados no decorrer da realização desta fase.

Conclusão

Após a conclusão desta última fase do projeto podemos afirmar que nos sentimos satisfeitos com o resultado e com o nosso desempenho uma vez que cumprimos com os objetivos propostos. O facto de termos elaborado um planeamento antes da realização de cada fase do projeto ajudou na coordenação e organização do que cada elemento do grupo tinha de fazer.

Relativamente às dificuldades que foram surgindo ao longo da realização desta fase estas foram ultrapassadas com sucesso apesar do desconhecimento inicial de alguns recursos e tecnologias como por exemplo no que toca a programação web. Para além disso não foi implementada a fase extra por motivos de tempo graças a problemas externos à realização desta fase.

Para esta fase a parte mais crucial da sua execução passou pela implementação da interface web e pensamos ter sido realizada com sucesso, no entanto achamos que também não se encontra o melhor que podíamos ter feito. Apesar disso acabamos este semestre com um projeto ao qual nos sentimos orgulhosos visto ter sido implementado grande parte do que era pedido e assim termos adquiridos conhecimentos para o nosso futuro.

Referências

- [1] <https://pplware.sapo.pt/tutoriais/networking/mqtt-protocolo-de-comunicacao/>
- [2] <https://nodejs.org/en/docs/guides/>
- [3] <https://expressjs.com/en/guide/routing.html>

Autoavaliação

Rui Filipe Ribeiro Freitas:

Nesta terceira e última fase do projeto ajudei no desenvolvimento do código da base de dados e da interface web assim como na pesquisa e elaboração do relatório.

Sandro Teixeira Ribeiro:

Nesta fase ajudei na pesquisa teórica, escrita e revisão do relatório.

Tiago João Pereira Ferreira:

Nesta fase ajudei no desenvolvimento do código do gestor de serviço e interface web. Ajudei também na elaboração do relatório.