

U.C. de Projeto Integrado de

Telecomunicações

Ano Letivo: **2021/2022**

**Relatório da Fase C**

**Grupo 2**

* Catarina Neves, a93088
* Eduardo Cardoso, a89627
* José Gomes, a93083
* Luís Oliveira, a89380

01/06/2022

Universidade do Minho

Mestrado Integrado em Engenharia de Telecomunicações e Informática

Índice

[1. Introdução 5](#_Toc105017420)

[2. Trabalho Relacionado 6](#_Toc105017421)

[2.1 ESP32 WebSocket: Control ESP32 Pins 6](#_Toc105017422)

[2.2 ESP32 Robot Car Using Websockets 7](#_Toc105017423)

[3. Fases do trabalho desenvolvido 8](#_Toc105017424)

[3.1. Protocolo de comunicação 8](#_Toc105017425)

[3.2 Estrutura da trama 8](#_Toc105017426)

[3.3. Sistema Central 9](#_Toc105017427)

[3.3.1 Mecanismos e políticas de segurança 9](#_Toc105017428)

[3.3.1.1 Enumerados 10](#_Toc105017429)

[3.3.1.2 Configuração do protocolo HTTP 11](#_Toc105017430)

[3.3.2 Tabelas da base de dados 12](#_Toc105017431)

[3.3.2.1 Tabela Users 12](#_Toc105017432)

[3.3.2.2 Tabela Authorities 13](#_Toc105017433)

[3.3.2.3 Tabela Weather 13](#_Toc105017434)

[3.4. Servidor *Web* 14](#_Toc105017435)

[3.4.1 Utilizador normal 14](#_Toc105017436)

[3.4.2 Utilizador administrador 15](#_Toc105017437)

[4. Análise de resultados e testes efetuados 16](#_Toc105017438)

[4.1 Sistemas Simulados 17](#_Toc105017439)

[5. Conclusão 18](#_Toc105017440)

[5.1. Contribuição de cada aluno 18](#_Toc105017441)

[6. Lista de referências 19](#_Toc105017442)

Índice de figuras

Figura 1 - Tarefas propostas pela fase C. 5

Figura 2 – Variação de cor da tira LED. 6

Figura 3 - Serviço web implementado pelo autor. 7

Figura 4 - Lista de componentes necessários à construção do projeto. 7

Figura 5 - Estrutura da trama. 8

Figura 6 - Estrutura da diretoria "security". 9

Figura 7 - Tarefas abordadas no parágrafo anterior. 9

Figura 8 - Habilitações da categoria ADMIN. 10

Figura 9 - Habilitações da categoria USER. 10

Figura 10 - Comportamento do método secure. 11

Figura 11 - Modelo conceptual dos dados. 12

Figura 12 - Valores armazenados na tabela "Users". 12

Figura 13 - Valores armazenados na tabela "Authorities". 13

Figura 14 - Valores armazenados na tabela "Weather". 13

Figura 15 - Página inicial de login. 14

Figura 16 - Página com todas as amostras. 14

Figura 17 - Página de erro por não ter autorização. 15

Figura 18 - Página inicial do servidor web. 15

Figura 19 - Botão para alterar estado de gateway. 15

Figura 20 - Amostras armazenadas na base de dados. 16

Figura 21 - Operação de "START" no gateway 1. 16

Figura 22 - Comportamento dos Sistemas Simulados. 17

Lista de siglas e acrónimos

**SQL** *Structured Query Language*

**LED** *Light Emitting Diode*

**RGB** *Red Green Blue*

**GPIO** *General Purpose Input/Output*

**IDE** *Integrated Development Environment*

**IOT** *Internet Of Things*

**HTTP** *Hypertext Transfer Protocol*

**IPv4** *Internet Protocol version 4*

**TCP** *Transmission Control Protocol*

# Introdução

Serve o presente relatório como síntese do trabalho desenvolvido e implementado no decorrer desta última fase. Este contém a descrição das estratégias e algoritmos adotados pelo grupo, tal como os respetivos testes realizados. Além dos tópicos acima citados, faz-se referência a trabalhos ou projetos similares, que se enquadram na ótica deste projeto.

O relatório abordará cada etapa desta fase de forma minuciosa, ou seja, será apresentada a resposta ou proposta de solução empregue pelo grupo em junção com as ferramentas que foram necessárias para a sua construção.

A figura 1 ilustra o sumário das diferentes interações entre as tarefas propostas nesta fase.

Uma imagem com texto, símbolo

Descrição gerada automaticamente

Figura 1 - Tarefas propostas pela fase C.

O principal objetivo desta fase é a conclusão do Sistema Central, que é caraterizado pelas suas funcionalidades mais avançadas, tais como: monitorização, administração, segurança e aprimoração do protocolo de comunicação entre este sistema e os *gateways* (*Gateway* e Sistemas Simulados) para tornar a aplicação mais eficiente, fiável e robusta.

O Sistema Central recolhe a informação enviada por todos os *gateways*, que por sua vez, será armazenada numa base de dados relacional. Sempre que o serviço *web* queira atualizar os dados apresentados, este efetua um pedido ao gestor de serviço que, de forma encadeada, irá procurar pelos dados dentro da base de dados, e enviará os mesmos para o serviço *web*.

# Trabalho Relacionado

Relativamente aos trabalhos relacionados, enquadram-se os projetos seguintes que serviram de base à construção deste projeto.

## 2.1 ESP32 WebSocket: Control ESP32 Pins

Este projeto [1] demonstra a implementação de um servidor ESP32, do tipo *WebSocket,* que é responsável pelo controlo dos pinos GPIO (*General Purpose Input/Output*) de uma placa ESP32. Para além da implementação do servidor, são apresentados os passos necessários à construção de uma página *web*, que permite efetuar o controlo dos pinos referidos.

Para realizar o efeito pretendido, o autor recorreu a uma fita LED (*Light Emitting Diode)* RGB (*Red Green Blue*), cujas cores são alteradas de forma remota (através do *browser*).

O autor recorreu ao protocolo de comunicação “WebSocket”, que permite a interação entre o cliente (*web browser*)e o servidor *web* (ESP32 WebSocket), isto é, ocorre troca de informação entre a página *web* e o servidor. Desta forma, o servidor manipula os pinos, consoante os dados que são introduzidos na página *web*.

O funcionamento deste projeto pode ser ordenado de acordo com as etapas seguintes:

1. Escolher a cor pretendida através de um web *browser.*
2. Envio da informação (cor selecionada) através do protocolo WebSocket.
3. A placa ESP32 (servidor WebSocket) recebe a informação e altera a cor do LED.

É de realçar o cuidado do autor, no tratamento do conjunto de eventos associados ao protocolo aplicado, nomeadamente: conexão e desconexão.

A figura 2 ilustra o resultado pretendido, aquando da finalização o projeto.



Figura 2 – Variação de cor da tira LED.

## 2.2 ESP32 Robot Car Using Websockets

Este projeto [2], da autoria de donsky, consiste num carro robot construído com o auxílio de uma placa ESP32 e de um motor LM298, que por sua vez, é controlado através de um smartphone. Este projeto foi desenvolvido com recurso ao IDE (Integrated Development Environment) PlatformIO, uma vez que este é favorável e adequado a projetos que se enquadram no domínio da IOT (Internet Of Things).

O cerne deste projeto encontra-se na utilização do protocolo de comunicação “*WebSockets”.* O protocolo citado é essencial e preenche todos os requisitos e tópicos, que são exigidos por aplicações do ramo da IOT, devido à pouca latência oferecida pelas *websockets.*

Neste contexto, o carro *robot* desempenha o papel de servidor e o *web browser* desempenha o papel de cliente. O cliente envia um conjunto de mensagens / instruções ao servidor, sendo posteriormente respondidas por este.

A figura 3 ilustra o serviço *web,* implementado pelo autor, que é responsável pelo controlo da direção do respetivo carro.



Figura 3 - Serviço web implementado pelo autor.

A placa ESP32 atua como servidor *websocket* e o *browser* do *smartphone* atua como cliente *websocket.* As mensagens do tipo WebSockets, provenientes do *browser*, são processadas pela placa referida. Concluído o seu processamento, a placa envia mensagens de controlo para o motor LM298, com o intuito de este controlar a rotação das rodas do carro. Para além do controlo da rotação das rodas, também é possível variar a velocidade das mesmas através da modulação por largura de pulso.

Sempre que um utilizador clica nos botões de *Up, Down. Left* ou *Right,* o *smartphone* envia uma mensagem do tipo *websocket* para o robot, que por sua vez será processada pela respetiva placa.

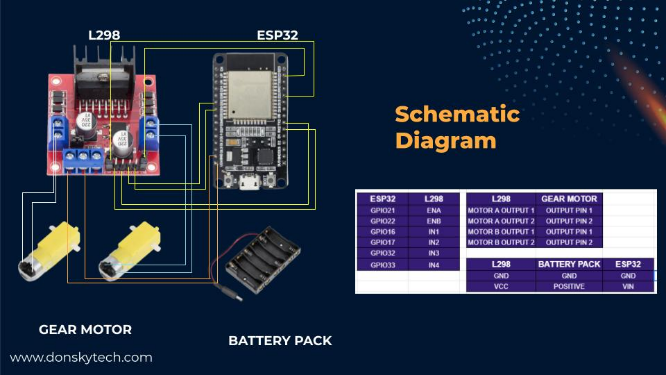
A figura 4 apresenta a lista de componentes que são necessários ao desenvolvimento deste projeto.

Figura 4 - Lista de componentes necessários à construção do projeto.

# Fases do trabalho desenvolvido

Esta secção contém todas as etapas necessárias ao desenvolvimento desta fase, que se encontram descritas pelos mecanismos, ferramentas utilizadas e algoritmos implementados pelo grupo.

## 3.1. Protocolo de comunicação

O protocolo de comunicação aplicacional, que foi desenvolvido pelo grupo possui um pequeno conjunto de mensagens, que são encapsuladas através do protocolo de transporte TCP (*Transmission Control Protocol*). As mensagens estipuladas pelo grupo foram as seguintes:

* START – Início do processo de recolha das amostras dos sensores.
* STOP – Finalização do processo de recolha das amostras dos sensores.
* PAUSE – Interrupção do processo de recolha das amostras dos sensores
* RESTART – Reinicialização do processo de recolha das amostras dos sensores.

## 3.2 Estrutura da trama

A trama desenhada pelo grupo agrupa dois campos: o código identificador do *gateway* (ID) e a mensagem do protocolo de transporte.

A figura 5 ilustra um exemplo da trama referida, cujo destinatário é o *gateway* 1 e cuja mensagem do protocolo associada é a “START”.

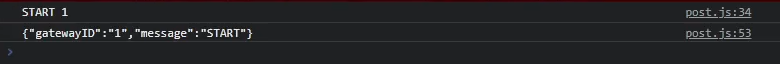


Figura 5 - Estrutura da trama.

É de referir que as tramas são processadas no Sistema Central. O Sistema Central encontra-se munido de mecanismos, que o permitem direcionar a trama para o destino correto, graças à aplicação de algoritmos de concorrência.

## 3.3. Sistema Central

Nas subsecções seguintes serão analisadas e descritas todas as caraterísticas, funcionalidades e matérias inerentes ao Sistema Central, que foram implementadas pelo grupo.

### 3.3.1 Mecanismos e políticas de segurança

Os mecanismos de autenticação, autorização e outros recursos de segurança foram elaborados com o apoio da *framework Spring Security* [3]*.* As políticas e estratégias de segurança adotadas pelo grupo encontram-se disponíveis na diretoria “*security*”.

A diretoria referida é composta por:

* 2 enumerados, um deles é responsável pela conceção do tipo de utilizadores do sistema e o outro define o conjunto de permissões ou habilitações associadas a cada tipo de utilizador.
* 1 classe que é responsável pela encriptação da *password* dos utilizadores. É de salientar que a encriptação das *passwords* é necessária e imposta pela *Spring Security,* para a definição do serviço *web.*
* 1 classe que filtra e configura o protocolo HTTP (*Hypertext Transfer Protocol*), isto é, define quais os *websites* que um determinado tipo de utilizadores possui acesso. Em junção com a qualificação dos *websites,* esta classe é responsável pela extração do conjunto de utilizadores, que se encontram registados na base de dados.

A figura 6 ilustra a estrutura da diretoria “*security*”*.*

Uma imagem com texto

Descrição gerada automaticamente

Figura 6 - Estrutura da diretoria "security".

Desta forma, cada utilizador possui os atributos seguintes: um *username*, uma *password*, um tipo e um conjunto de habilidades associadas a esse tipo. Tal como foi citado anteriormente, o conjunto de utilizadores aptos a interagir com o serviço *web*, encontram-se armazenados numa tabela, dentro da base de dados (tópico que será explorado na secção seguinte).

Uma imagem com texto

Descrição gerada automaticamenteA figura 7 ilustra as questões que foram abordadas no parágrafo anterior.

Figura 7 - Tarefas abordadas no parágrafo anterior.

#### 3.3.1.1 Enumerados

O grupo categorizou e diferenciou os clientes/utilizadores em duas categorias distintas: administradores ou utilizadores normais.

Cada categoria possui um conjunto de permissões ou habilitações associadas, que por consequência, terão impacto nas funcionalidades disponibilizadas pelo serviço *web*.

A figura 8 descreve a categoria “ADMIN” (administrador) e o conjunto de habilitações associadas.

Uma imagem com texto, eletrónica, disco compacto

Descrição gerada automaticamente

Figura 8 - Habilitações da categoria ADMIN.

Tal como é evidenciado na figura 8, qualquer utilizador do tipo “ADMIN” pode efetuar operações de leitura e escrita sobre as amostras, por exemplo:

* Ler as amostras que se encontram armazenadas na base de dados.
* Pesquisar por uma determinada amostra através do seu código identificador (que compõe a chave primária da tabela das amostras).
* Introdução de uma amostra na base de dados.
* Remoção de uma amostra da base de dados.

Todos os tópicos acima referidos, foram realizados com recurso a anotações, que são caraterísticas da *framework Spring Security,* que no fundo podem ser interpretadas como filtros. Nessas anotações é passado como parâmetro, a categoria de utilizador pretendida. As anotações citadas foram aplicadas sobre um conjunto de métodos intrínsecos ao protocolo HTTP (*POST, GET, PUT* e *DELETE*.), que por sua vez encontram-se disponíveis na classe *WeatherSampleController*.

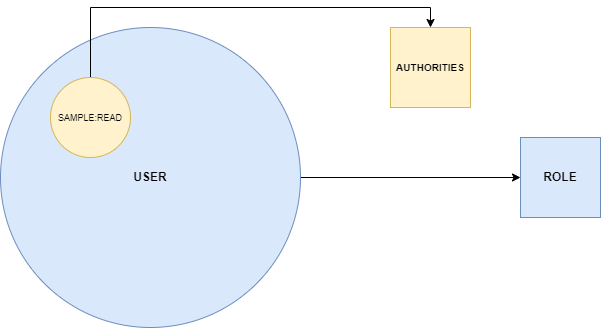
A figura 9 descreve a categoria USER (administrador) e o conjunto de habilitações correspondentes.

Figura 9 - Habilitações da categoria USER.

Em contrapartida, qualquer utilizador do tipo “USER”, apenas pode efetuar operações de leitura sobre as amostras, como é o caso da leitura da série de amostras, que se encontra armazenada na base de dados.

#### 3.3.1.2 Configuração do protocolo HTTP

A configuração do protocolo HTTP e da sua segurança, é efetuada dentro do método *configure* da classe *ApplicationSecurityConfig*. Dentro do método referido, são definidos o conjunto de *websites* disponíveis às categorias de utilizadores, que foram descritas na subsecção anterior. Em junção com o tópico anterior*,* também são referenciados ou direcionados, os ficheiros html associados às páginas de *login* e *logout*.

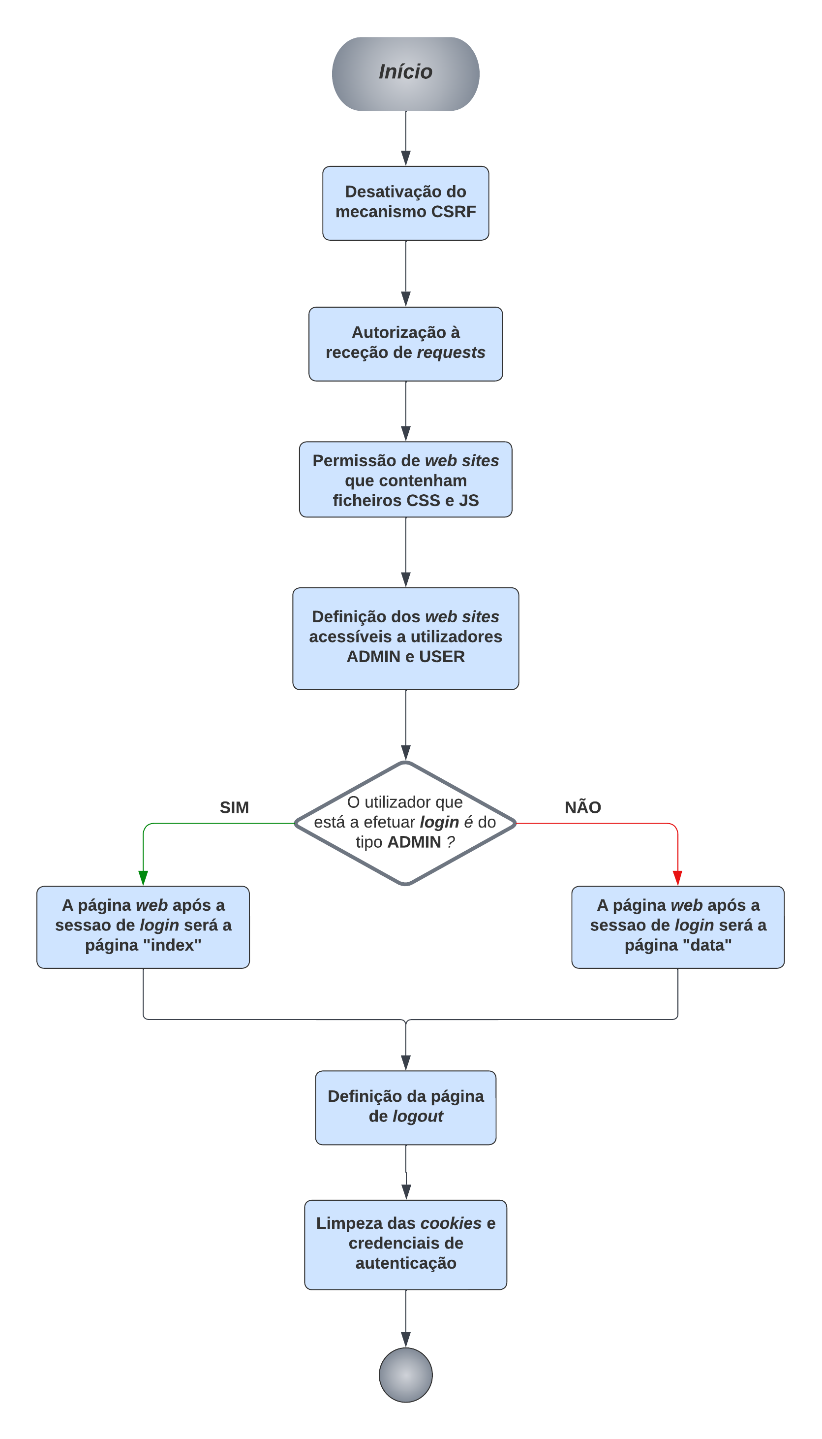
O algoritmo, que descreve o método citado, encontra-se traduzido de acordo com o fluxograma da figura 10.

Figura 10 - Comportamento do método secure.

### 3.3.2 Tabelas da base de dados

Dentro do sistema de gerenciamento de base de dados [4], o grupo definiu 3 tabelas, que são cruciais ao correto funcionamento deste projeto. É de salientar que cada tabela representa uma determinada entidade e que existem ligações entre as mesmas.

As tabelas definidas pelo grupo foram as seguintes:

* A tabela “users”, que armazena a lista de utilizadores autorizados a interagir com o serviço *web.*
* A tabela “authorities”, que contém o conjunto de competências associadas a cada categoria de utilizador.
* A tabela “weather”, que conserva todas as amostras recebidas pelo Sistema Central.

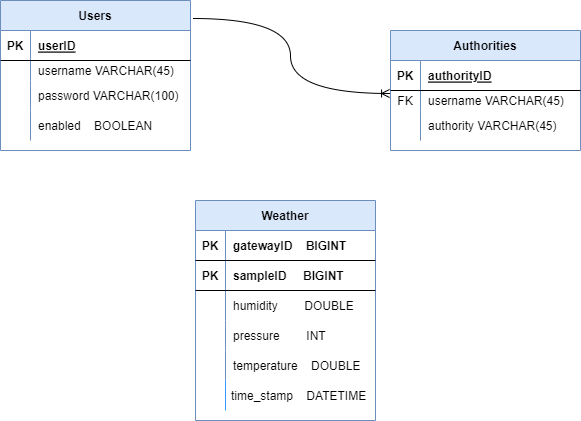
A figura 11 representa o modelo conceptual de dados, relativamente às tabelas apresentas anteriormente.

Figura 11 - Modelo conceptual dos dados.

#### 3.3.2.1 Tabela Users

De forma a garantir e a reforçar a segurança dos utilizadores, aliada ao cumprimento dos requisitos das *frameworks* [5][3] utilizadas, foi necessário encriptar a *password* de cada utilizador. É de sublinhar, que as *passwords* (“grupo2”) são iguais para todos os utilizadores e que a encriptação das mesmas foi realizada com recurso a um *website* [6] apropriado.

A figura 12 apresenta o conjunto de valores armazenados na tabela “Users”.

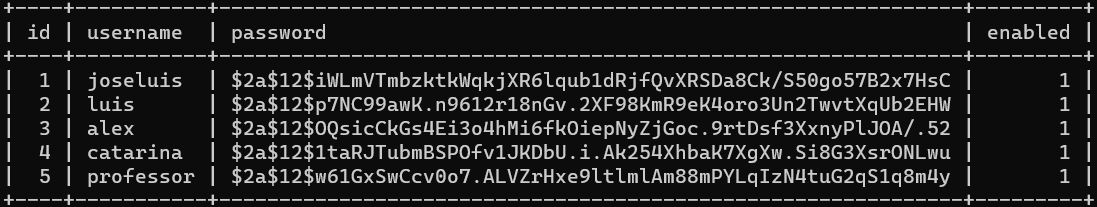


Figura 12 - Valores armazenados na tabela "Users".

#### 3.3.2.2 Tabela Authorities

Em relação à tabela “Authorities”, esta define a categoria de cada utilizador, que se encontra registado na base de dados. Tal como foi referido em algumas secções anteriores, cada categoria possui um conjunto de habilitações ou aptidões associadas.

A figura 13 apresenta o conjunto de valores armazenados na tabela “Authorities”.

Uma imagem com texto

Descrição gerada automaticamente

Figura 13 - Valores armazenados na tabela "Authorities".

#### 3.3.2.3 Tabela Weather

No que toca à tabela “Weather”, esta define a estrutura, isto é, define os atributos de cada amostra e regista o *gateway* responsável pela sua recolha.

A figura 14 apresenta o conjunto de alguns valores armazenados na tabela “Weather”.

Uma imagem com texto, quadro de resultados, captura de ecrã

Descrição gerada automaticamente

Figura 14 - Valores armazenados na tabela "Weather".

## 3.4. Servidor *Web*

Para a presente etapa, o Servidor *Web* que o grupo irá apresentar é uma versão melhorada e atualizada da que foi apresentada na fase B.

Uma modificação visual importante foi a substituição do submenu *Feeds* porum novo, intitulado de “*Historical Data”*, cuja função é a adição de uma dimensão gráfica à apresentação dos dados obtidos, através de gráficos que refletem as variações de temperatura, humidade e pressão atmosférica ao longo do tempo.

Outra modificação visual crucial foi a adição do controlo, por parte de um utilizador com capacidades de administração, podendo, desta forma, qualquer utilizador que seja administrador alterar o estado de cada *gateway.*

Tal como foi apresentado anteriormente, o grupo também adicionou ao *Servidor Web* um protocolo de *login*, sendo essa a primeira página onde cada utilizador irá parar, como mostra a figura 15.

Uma imagem com texto

Descrição gerada automaticamente

Figura 15 - Página inicial de login.

### 3.4.1 Utilizador normal

Caso o utilizador não seja administrador, este apenas terá acesso a uma página com uma tabela, que apresenta em tempo real, o ID da amostra, o ID do *gateway* associado a essa amostra, tal como a temperatura, humidade e pressão atmosférica recolhida pelos sensores, juntamente com a *timestamp*, tal como é mostrado na figura 16.

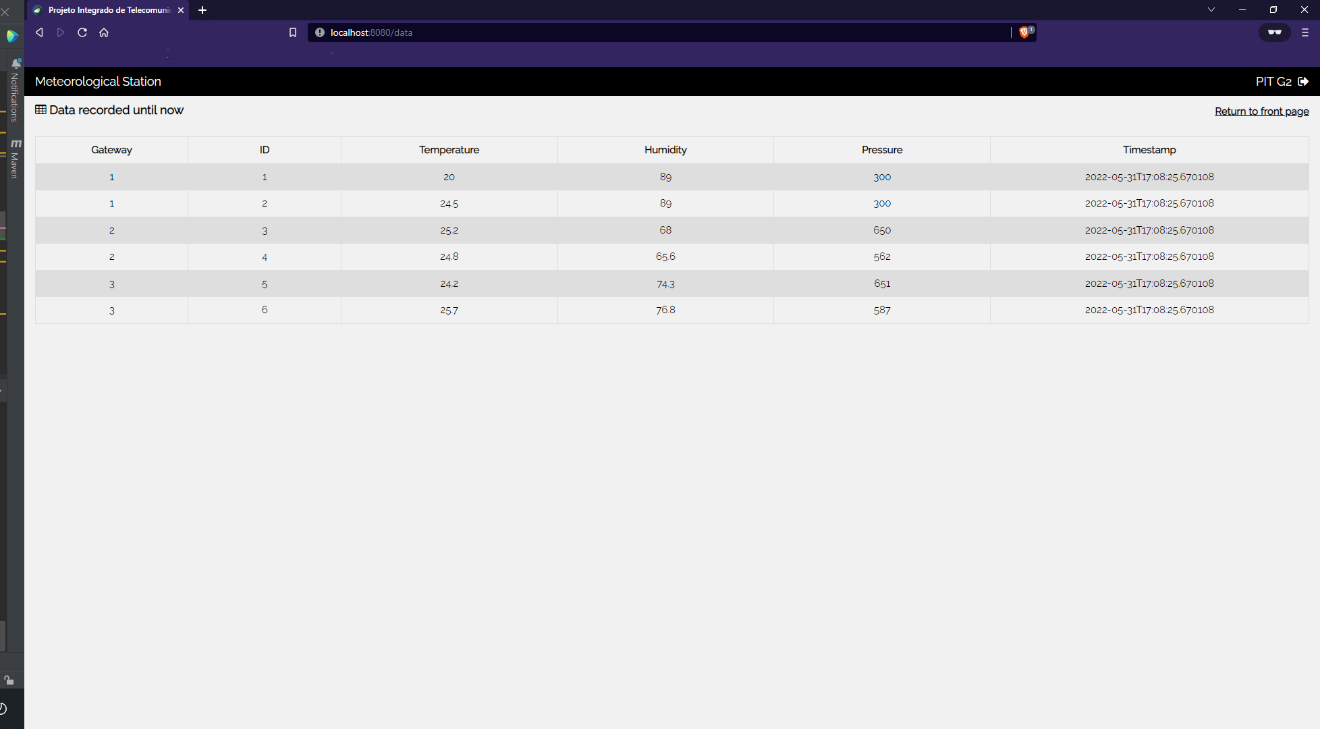


Figura 16 - Página com todas as amostras.

Se por alguma razão, um utilizador do tipo “USER” tentasse visualizar a página com capacidades de administração, o Servidor *Web* apresentar-lhe-ia uma página em branco de erro, que apesar de sugerir um erro, a mesma foi planeada e definida no código.

A figura 17 ilustra a página de erro referida.

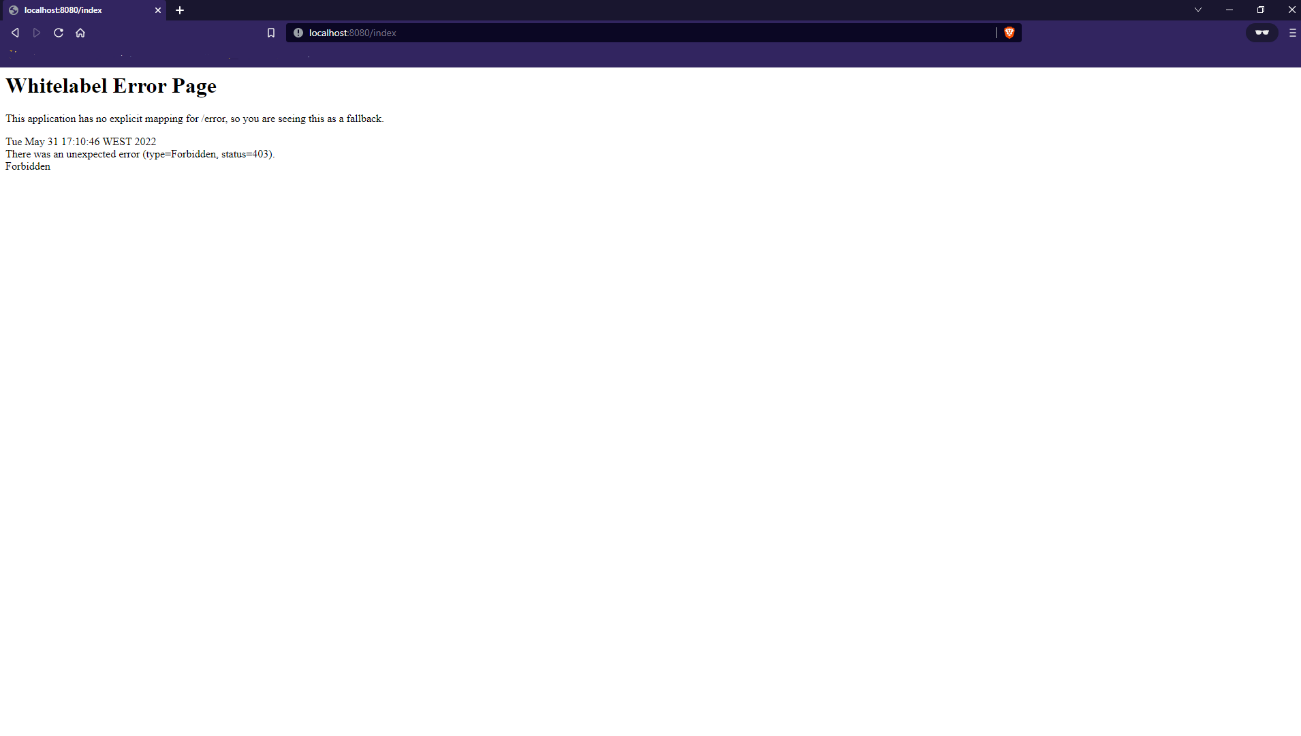


Figura 17 - Página de erro por não ter autorização.

### 3.4.2 Utilizador administrador

Caso o utilizador seja administrador, a primeira página que terá acesso será a página *index*, onde esta apresenta o número de amostras recebidas, tal como a temperatura, humidade e pressão atmosférica da última amostra enviada pelo *gateway*, seguida pela localização do mesmo. Os gráficos ilustram a variação dos parâmetros meteorológicos, tal como é evidenciado na figura 18.

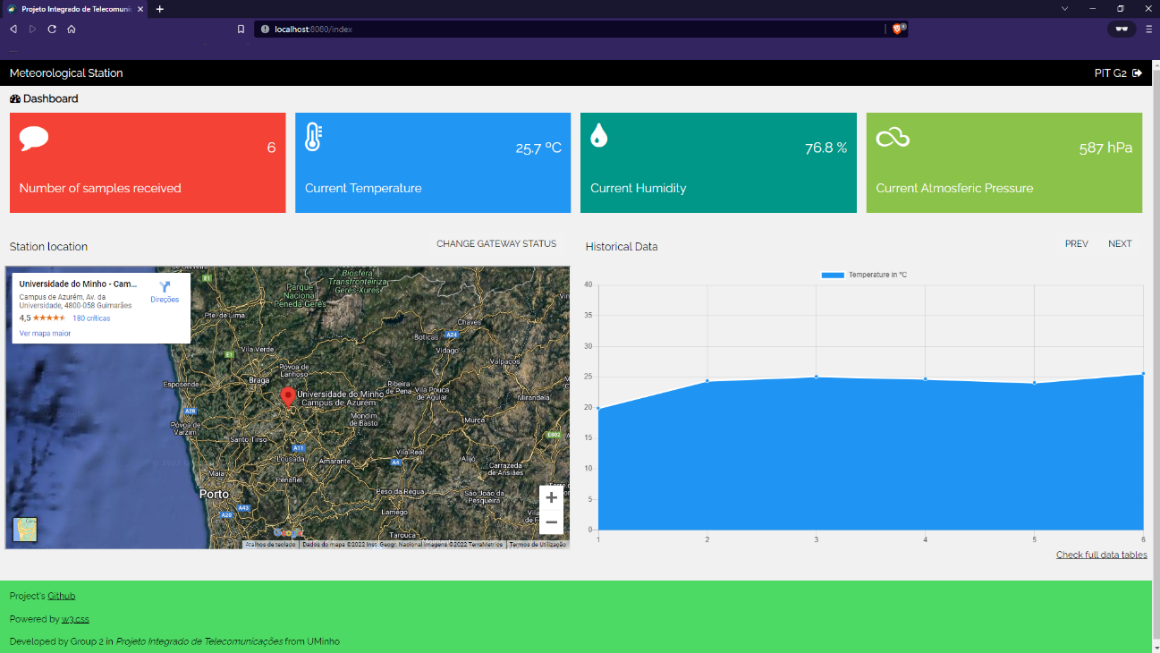


Figura 18 - Página inicial do servidor web.

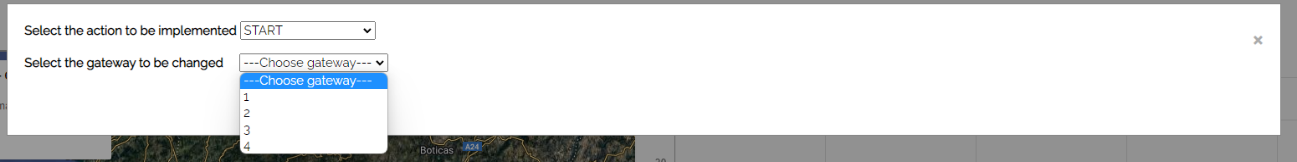
O utilizador também terá acesso a um botão no centro do ecrã, do qual consegue alterar o estado de cada um dos gateways, como mostra a figura 19.

Figura 19 - Botão para alterar estado de gateway.

# 4. Análise de resultados e testes efetuados

De forma a retificar o comportamento do Sistema Central, o funcionamento do protocolo de comunicação, a interação com o serviço *web* e a cinética dos mecanismos e políticas de segurança, o grupo testou as funcionalidades oferecidas pelo serviço web, com todos os utilizadores, que se encontram registados na base de dados.

A figura 20 ilustra a página sucessora da página *login*,aquando da sua finalização (login efetuado por um utilizador do tipo administrador). Tal como se encontra evidenciado no código, qualquer utilizador que seja do tipo referido, logo que efetue a operação de *login*, terá uma nova página *web*, que contêm um conjunto de componentes gráficas que refletem informação, relativamente às amostras recolhidas pelo sistema central.

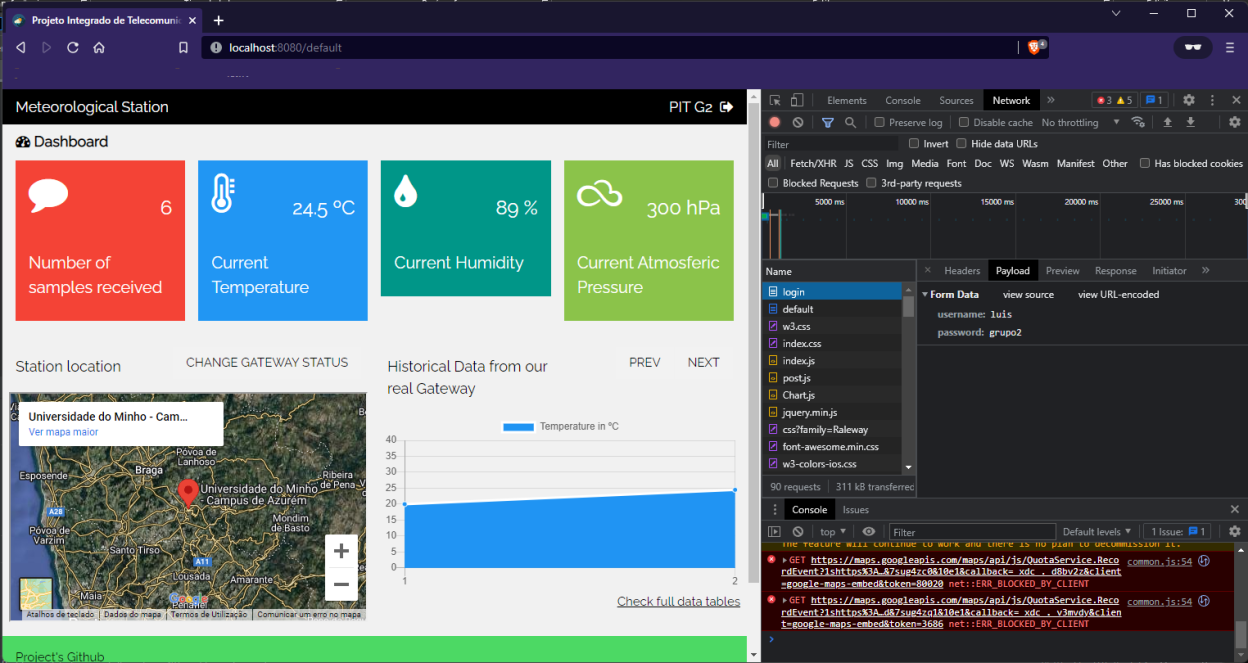
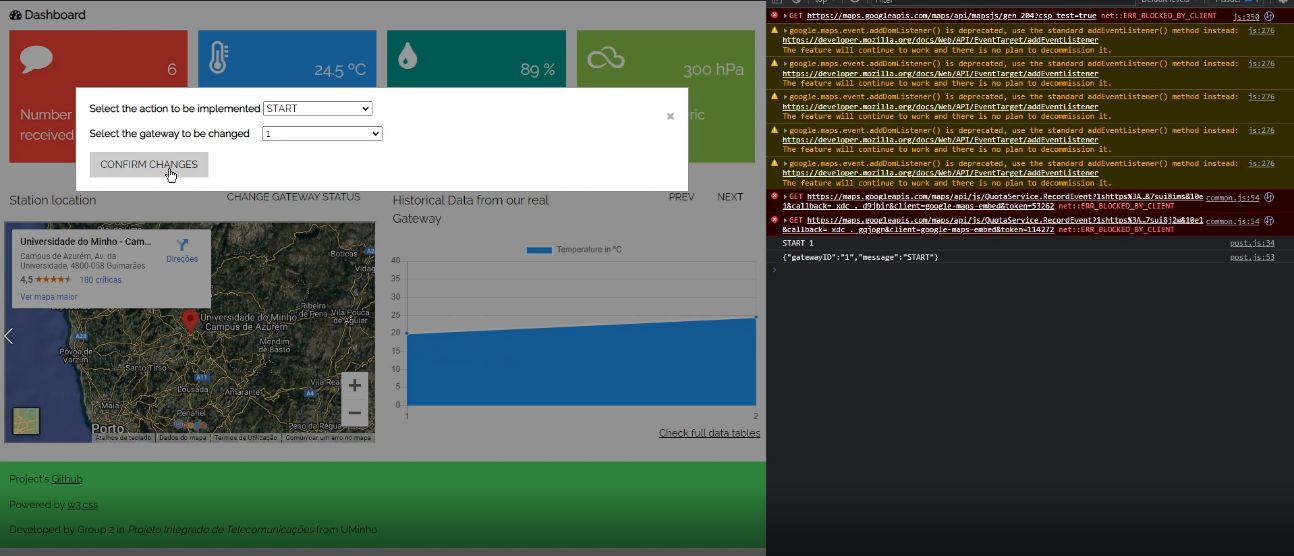


Figura 20 - Amostras armazenadas na base de dados.

A figura 21 mostra o utilizador a selecionar a operação de “START” no *gateway* 1, é de destacar que do lado direito da figura, é possível observar a trama que é enviada para o Sistema Central. Por sua vez, a trama é processada e remetida para o *gateway* respetivo.



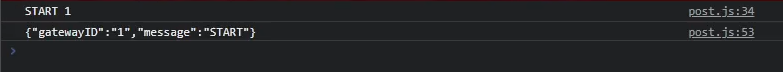


Figura 21 - Operação de "START" no gateway 1.

## 4.1 Sistemas Simulados

O grupo definiu uma classe Java, que simula os efeitos pretendidos com os sistemas simulados. Nessa classe, foi definida uma *socket* de ligação no endereço IPv4 (*Internet Protocol version 4*), numa determinada porta TCP (*Transmission Control Protocol*). Para além da utilização da *socket* de ligação, foi necessário recorrer a um ciclo infinito, com o intuito de enviar amostras de forma sucessiva, corrente para o Sistema Central.

O algoritmo, que descreve o comportamento da classe, encontra-se traduzido de acordo com o fluxograma da figura 22.

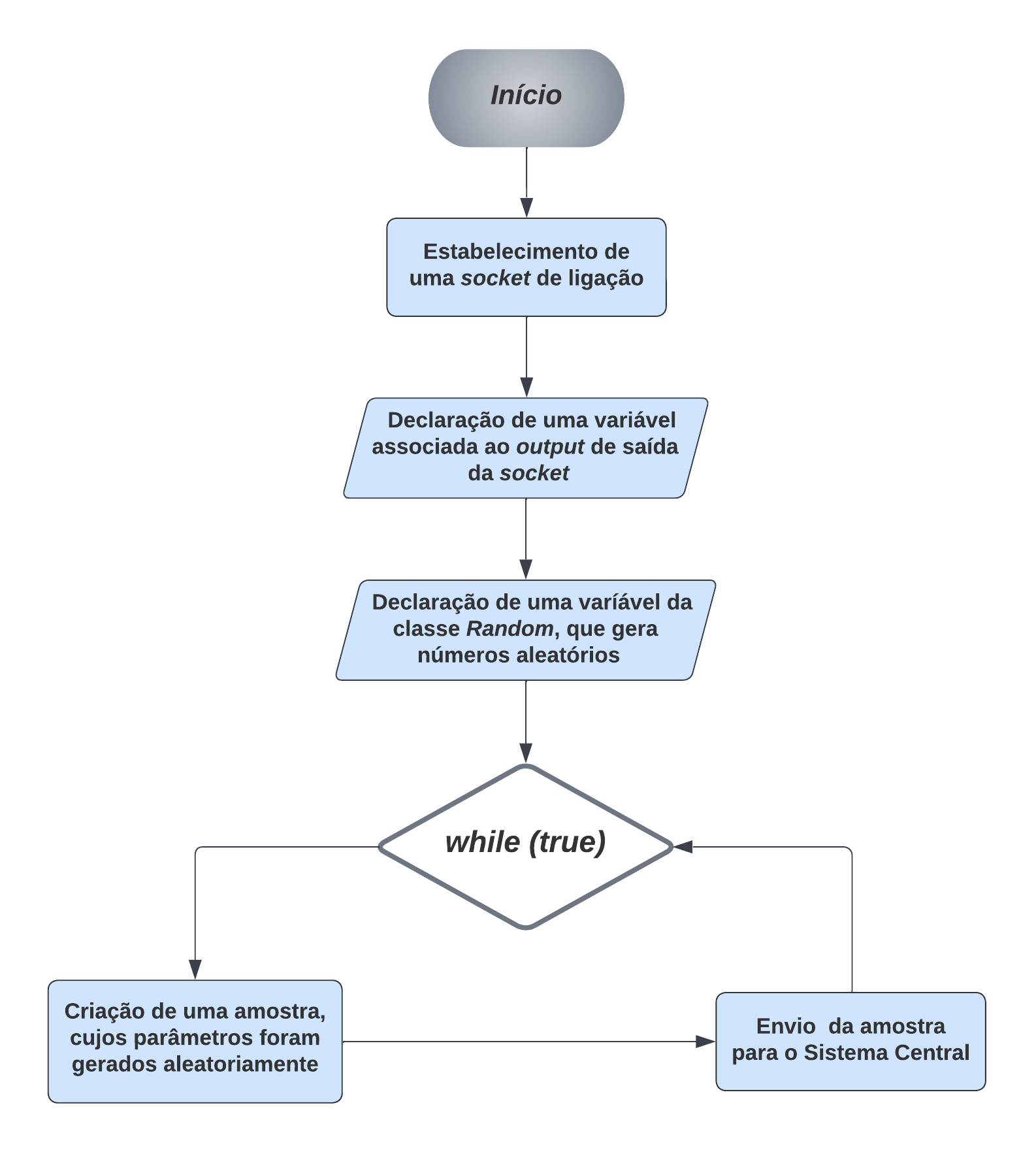


Figura 22 - Comportamento dos Sistemas Simulados.

# 5. Conclusão

Concluída a fase final deste projeto, na opinião do grupo conseguir-se atingir todos os objetivos propostos. Esta fase aumentou o leque de conhecimentos relativamente ao conceito e comportamento do protocolo HTTP, nomeadamente na implementação das *Sockets* de ligação, na comunicação sobre TCP/IP e sobre o *design* de Servidores *Web*.

É de salientar que foi efetuada uma ligação *full-duplex* entre o Sistema Central e o Servidor Web, da qual se tirou partidodo protocolo HTTP, aliada com o armazenamento das credenciais de autenticação de cada utilizador na base de dados relacional.

O grupo pretende que, com o trabalho desenvolvido nesta fase final, ter concluído com sucesso todos os tópicos que se comprometeu a realizar e a implementar, onde se inserem a correção de erros e a conclusão de algumas etapas da fase anterior.

## 5.1. Contribuição de cada aluno

Catarina Neves

* Conceção dos algoritmos;
* Tradução dos algoritmos em diagramas;
* Tratamento de dados, tabelas e imagens a estes relacionados;
* Definição e implementação do protocolo de comunicação;
* Programação e implementação da aplicação dos Sistema Sensor e *gateway*;
* Desenvolvimento do relatório;
* Desenvolvimento da apresentação *PowerPoint*;

Eduardo Cardoso

* Conceção dos algoritmos;
* Tradução dos algoritmos em diagramas;
* Definição e implementação do protocolo de comunicação;
* Programação e implementação da aplicação dos Sistema Sensor e *gateway*;
* Desenvolvimento e implementação das *sockets* e comunicação TCP/IP;
* Desenvolvimento do relatório;
* Desenvolvimento da apresentação *PowerPoint*;

José Gomes

* Tradução dos algoritmos em fluxogramas;
* Implementação e configuração do protocolo HTTP;
* Implementação e definição dos mecanismos e políticas de segurança;
* Desenvolvimento e implementação das *sockets* e comunicação TCP/IP;
* Desenvolvimento e construção da base de dados relacional;
* Desenvolvimento do relatório;
* Desenvolvimento da apresentação *PowerPoint*;

Luís Oliveira

* Conceção dos algoritmos;
* Tradução dos algoritmos em fluxogramas;
* Desenvolvimento do *Frontend* e *Backend* do Servidor *Web*;
* Implementação da comunicação para o Sistema Central;
* Desenho de esquemas e gráficos;
* Desenvolvimento do relatório;
* Desenvolvimento da apresentação *PowerPoint*;

# Lista de referências

|  |  |
| --- | --- |
| [1] | SWA, “ESP32 Websocket: Control ESP32 Pins [PlatformIO],” 4 2021. [Online]. Available: https://www.survivingwithandroid.com/esp32-websocket-control-esp32-pins-platformio/. |
| [2] | donsky, “ESP32 Robot Car Using Websockets,” julho 2021. [Online]. Available: https://www.donskytech.com/esp32-robot-car-using-websockets/. |
| [3] | VMware, “Spring Security,” 2022. [Online]. Available: https://spring.io/projects/spring-security?msclkid=03ed647cc53c11ec824b0986c6465ca1. |
| [4] | MySQL, “MYSQL,” 2022. [Online]. Available: https://www.mysql.com/. |
| [5] | W3Schools, “W3Schools W3.CSS,” [Online]. Available: https://www.w3schools.com/w3css/. |
| [6] | “Bcrypt-Generator.com - Online Bcrypt Hash Generator & Checker,” 2022. [Online]. Available: https://bcrypt-generator.com/. |
| [7] | R. Santos, “Random Nerd Tutorials,” 10 2020. [Online]. Available: https://randomnerdtutorials.com/esp32-websocket-server-arduino/. |
| [8] | Spring, “Spring Boot,” 2022. [Online]. Available: https://spring.io/projects/spring-boot. |