



Departamento de Ciência da Computação  
Universidade de Brasília  
Disciplina: Teleinformática e Redes 2  
Segundo Semestre de 2025

## **Relatório - Entrega 2**

**Alunos:** Felipe Lopes Gibin Duarte (231025207)  
Artur Kohara Guerra (231025181)

### **1) Objetivos da segunda entrega**

Esta segunda entrega tinha como foco a transição do protótipo apenas simulado para a integração real com o hardware, usando 2 Arduinos Uno acoplados a módulos LoRa. O objetivo final era conseguir uma comunicação ponta a ponta desde a geração dos dados simulados pelo arduino cliente/sensor até sua visualização no dashboard, em que todas as etapas foram validadas em um contexto de comunicação física real.

### **2) Integração dos módulos e ajustes no sistema**

A arquitetura do sistema foi majoritariamente mantida, embora agora tenha sido implementada sobre componentes reais. O sistema ainda é composto por nós sensores, gateway LoRa, servidor backend e dashboard. Contudo, quando comparado com a entrega um, em que todos os componentes eram virtuais, algumas alterações foram realizadas a fim de atingir o objetivo proposto de integração do sistema. Dessa forma, a entrega dois se aproxima mais da arquitetura completa proposta anteriormente.

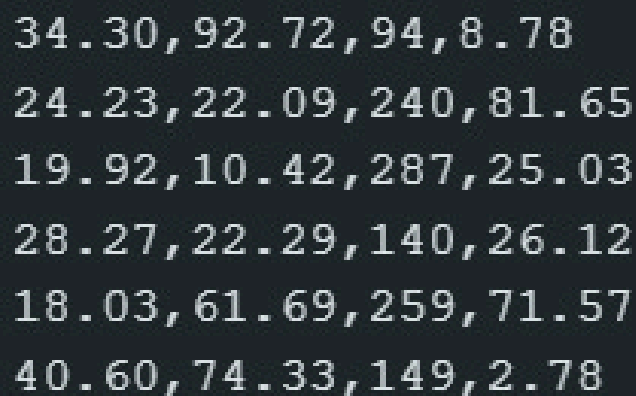
O nó sensor é representado por um Arduino Uno cliente, que simula um payload de sensoriamento (dado a ausência de sensores físicos reais no momento) e envia essas informações em formato de String, em que cada leitura é representada por um número decimal e separada por vírgula. Esse payload é enviado via LoRa ao segundo Arduino Uno, que representa o gateway. O gateway recebe essa String

também via LoRa e a encaminha pela porta serial ao computador, onde um script Python realiza o parsing da String recebida e cria um pacote de dados no formato JSON. Esse pacote JSON é então enviado ao servidor por meio de uma requisição HTTP POST, e o resto do fluxo segue normalmente conforme especificado na entrega um.

Assim, a entrega evidencia a substituição de componentes puramente lógicos e virtuais previamente utilizados, como o arquivo de geração de dados python, pelos elementos físicos representados pelos Arduinos, que se comunicam pelo protocolo LoRa.

### **3) Demonstração da comunicação real**

#### **3.1) Arduino cliente/sensor:**



```
34.30,92.72,94,8.78  
24.23,22.09,240,81.65  
19.92,10.42,287,25.03  
28.27,22.29,140,26.12  
18.03,61.69,259,71.57  
40.60,74.33,149,2.78
```

A primeira imagem se refere às mensagens enviadas pelo Arduino cliente/sensor, que atua como o nó sensor do sistema, ao segundo Arduino, que atua como o gateway. Note que foi escolhido um formato de mensagem em String, com vírgulas atuando como separadores. A estrutura da mensagem segue a lógica: "temperatura,umidade,poeira,bateria", tornando a interpretação dos dados bem direta. Esse formato é propício pois não adiciona um overhead significativo, e não precisa de bibliotecas extras para ser codificado. Isso é importante ao considerar que em sistemas embarcados, como é o caso do Arduino, há uma grande restrição de poder computacional, memória e energia. Dessa forma, faz-se

necessário o uso de estratégias que minimizem o gasto desses recursos já escassos.

### 3.2) Gateway:

```
Recebido: 34.30,92.72,94,8.78
Dados enviados com sucesso
Recebido: 24.23,22.09,240,81.65
Dados enviados com sucesso
Recebido: 19.92,10.42,287,25.03
Dados enviados com sucesso
Recebido: 28.27,22.29,140,26.12
Dados enviados com sucesso
Recebido: 18.03,61.69,259,71.57
Dados enviados com sucesso
Recebido: 40.60,74.33,149,2.78
Dados enviados com sucesso
```

Essa segunda imagem mostra o gateway implementado em um dos arduinos recebendo os dados via LoRa do outro arduino (cliente/sensor). O Gateway, então, pega os dados da string, lê eles e os imprime, enviando-os por meio da porta serial (USB). Isso permite que o script python do gateway obtenha os dados e os transforme para o formato JSON, permitindo que sejam lidos adequadamente pelo servidor posteriormente.

### 3.3) Dashboard

#### Dashboard de Monitoramento Ambiental

ID do Nó	Timestamp	Temperatura (°C)	Umidade (%)	Poeira (a definir unidade)	Bateria (%)
sala_B_02	2025-11-06T15:28:44.189351+00:00	40.6	74.33	149.0	2.78
sala_B_02	2025-11-06T15:28:14.160737+00:00	18.03	61.69	259.0	71.57
sala_B_02	2025-11-06T15:27:44.051277+00:00	28.27	22.29	140.0	26.12
sala_B_02	2025-11-06T15:27:13.941596+00:00	19.92	10.42	287.0	25.03
sala_B_02	2025-11-06T15:26:43.827761+00:00	24.23	22.09	240.0	81.65
sala_B_02	2025-11-06T15:26:13.719675+00:00	34.3	92.72	94.0	8.78

Após o envio do pacote JSON ao servidor, este adiciona as informações coletadas ao banco de dados relacional, em forma de tabela, que é exibida no dashboard, conforme especificado na entrega um. Note que o conteúdo dos dados apresentados são condizentes com a troca de dados via LoRa, o que valida a atuação do sistema no seu

contexto de aplicação. Também note que é adicionado um timestamp que contextualiza a leitura dos sensores em uma sequência temporal, a fim de prover mais informações ao usuário que irá monitorar o sistema.

#### **4) Problemas e soluções**

Alguns possíveis problemas foram identificados e solucionados, principalmente no que cabe à configuração da conexão dos arduinos via LoRa. Por exemplo, foi preciso garantir que ambos os arduinos operassem sobre a mesma frequência de sinal LoRa (nesse caso, 915 MHz), para que o gateway e o script python que trata os dados enviados por ele se comuniquem adequadamente. Além disso, foi preciso identificar a porta serial correta (a porta pode ter nomes diferentes dependendo do sistema operacional da máquina). Por fim, o baud rate de transmissão entre o gateway e o script python também precisaram ser iguais para garantir que os dados não acabassem corrompidos ou incompletos.