

# Trabalho Prático 1

Grupo 53

---

Licenciatura em Engenharia Informática  
Sistemas Distribuídos

**Autores:**

João Barros – al78806

Miguel Teixeira – al78321

Rui Madureira – al78282

Vila Real, 2025

## Protocolo de Comunicação

O presente relatório tem como objetivo descrever, de forma detalhada, a abordagem adotada na resolução do Trabalho Prático 1.

Numa fase inicial, foi desenvolvido o esboço do protocolo de comunicação subjacente ao sistema, representado na Figura 1. Neste protocolo, o fluxo de comunicação inicia-se com o *Wavy* a enviar um pedido de registo ao Agregador com o qual pretende estabelecer ligação. Após a receção do pedido, o Agregador responde com uma mensagem de confirmação, sinalizando o sucesso do registo. Este mesmo procedimento é replicado na comunicação entre o Agregador e o Servidor, garantindo consistência e fiabilidade em todos os pontos da cadeia de comunicação.

Segundos após a confirmação de registo por parte do Agregador, o *Wavy* envia o seu estado atual, definido por defeito como "**Operacional**", mesmo antes do início da transmissão de dados. Esta escolha visa garantir maior fluidez e simplicidade no processo de envio de dados.

Após a receção dos estados de todos os *Wavys* associados, o Agregador procede à transmissão periódica dos dados. Cada *Wavy* disponibiliza informações organizadas em ficheiros Excel, abrangendo seis categorias distintas: Humidade; Ondulação; Pressão; Temperatura da Água; Temperatura do Ar; Velocidade do Vento.

O envio de dados por cada *Wavy* decorre em ciclos de 10 segundos. O Agregador armazena os dados recebidos em ficheiros Excel próprios, com nomes no formato "**Dados - [Tipo de Dado]**" (por exemplo, "Dados - Humidade").

Após a receção de 12 registos de dados por parte dos *Wavys* associados, o Agregador envia-lhes o comando "**Pausa**", interrompendo temporariamente o envio dos mesmos. Esta pausa permite ao utilizador alterar o estado de cada *Wavy* para uma das seguintes opções:

- **Manutenção** – o *Wavy* permanece *offline* durante 10 minutos;
- **Desativada** – a ligação é encerrada de forma definitiva;
- **Operacional** – o estado mantém-se inalterado.

Simultaneamente ao envio do comando de "Pausa" aos *Wavys*, o Agregador transmite ao Servidor os últimos 12 dados registados, onde, por sua vez, o Servidor armazena estes dados em ficheiros Excel, organizados de forma idêntica à do Agregador, com base na tipologia dos dados.

Por fim, o Agregador gera um ficheiro Excel intitulado "**Wavys Associadas**", no qual são registadas as informações relativas à última ligação de cada dispositivo *Wavy*. Este registo inclui dados como o identificador (ID) de cada *Wavy* associada, o seu estado atual, os tipos de dados transmitidos, a data da última sincronização, bem como o estado dos dados, ou seja, se estes ainda se encontram em processo de agregação ou se já foram enviados para o servidor.

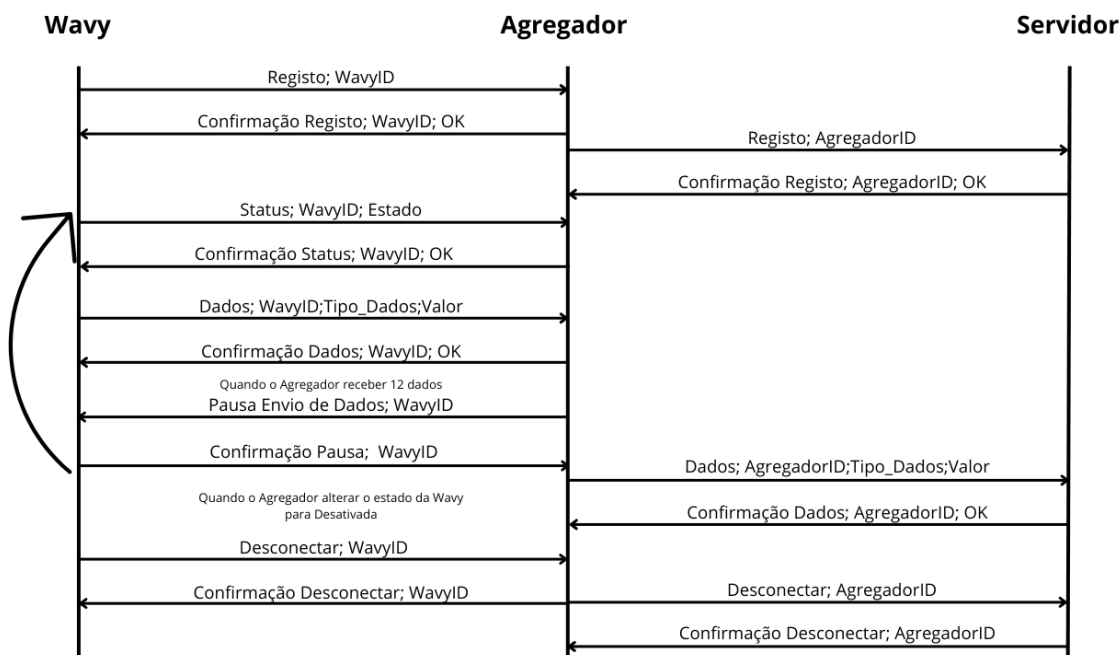


Figura 1 - Protocolo de comunicação cliente/servidor

## Implementação

Antes de iniciar a implementação do código, criamos um repositório na plataforma **GitHub** para facilitar a gestão de versões ao longo do desenvolvimento do projeto. Neste repositório, criamos três *branches* distintas, cada uma correspondente a um elemento da equipa, permitindo assim um desenvolvimento colaborativo organizado e simultâneo, sem comprometer a integridade do código na *branch* principal. Adicionalmente, foi incorporada a biblioteca **ClosedXML.Excel** para trabalhar e manipular ficheiros Excel de forma eficiente e estruturada.

### I. Atendimento dos Wavys

O Agregador implementa um servidor TCP na porta 4000 (porta 4500 no Agregador456), configurado para aguardar e aceitar conexões provenientes dos dispositivos *Wavy*. Para cada dispositivo que estabelece uma conexão, o sistema cria uma *Thread* dedicada através do método '**HandleClient()**', garantindo assim a gestão individualizada e eficiente de cada ligação.

Em paralelo, o sistema mantém um registo atualizado do estado operacional de todos os dispositivos *Wavy* conectados, informação esta armazenada no dicionário '**estadosWavys**'. Uma vez que estávamos com dificuldades no processo do Agregador alterar estado e guardar dados, decidimos implementar um mecanismo de **pausa automática**, o qual é acionado após a receção de cada lote de **12 registos de dados**. Assim, é proporcionado ao utilizador uma janela de oportunidade para modificar o estado dos dispositivos sem interromper o fluxo global de operações do sistema.

## II. Comunicação com cada Wavy

A comunicação entre o Agregador e os dispositivos Wavy é realizada através de um protocolo de mensagens baseado em texto, seguindo um formato estruturado que inclui comandos específicos como: **'Registo;WavyID'** e **'STATUS;WavyID;Estado'**. Para cada conexão estabelecida, o sistema utiliza **streams** de rede dedicadas, organizadas no **dicionário 'streamsWavys'**, assegurando assim a separação adequada dos canais de comunicação. Cada mensagem processada pelo sistema requer uma confirmação explícita de receção, implementando um mecanismo de comunicação fiável e resistente a falhas.

Adicionalmente, comandos como **'PAUSA'** e **'NOVO\_STATUS'** permitem a gestão remota dos dispositivos, possibilitando transições entre os diversos estados: **Operacional**, **Manutenção** e **Desativada**.

## III. Garantia de processamento dos ficheiros

Para salvaguardar a integridade e consistência dos dados no Agregador durante o processamento dos ficheiros, o sistema recorre a mecanismos de bloqueio, geridos através do **dicionário 'fileLocks'**, prevenindo conflitos decorrentes de acessos simultâneos aos ficheiros Excel. Foi igualmente implementado um sistema de tratamento de exceções, o qual permitiu lidar adequadamente com falhas ocorridas durante a escrita dos dados, possibilitando a identificação e correção de eventuais erros. Os **dados são organizados em lotes de 12 registos**, facilitando a visualização e sincronização com o servidor central. Esta abordagem garante que nenhuma informação seja perdida, mesmo em situações de falhas temporárias de comunicação ou sistema.

## IV. Atendimento simultâneo de múltiplos Agregadores

Para permitir o atendimento simultâneo de múltiplos Agregadores, foi implementado **multithreading**, criando uma **Thread** por conexão. Para evitar conflitos de escrita nos ficheiros Excel, usamos **Mutex** individuais por ficheiro. Além disso, o **Mutex global 'agregadorEscritaMutex'** garante que, mesmo que vários lotes de dados **'BATCH\_DADOS'** do **mesmo Agregador** cheguem ao mesmo tempo por diferentes Threads, **apenas um é processado de cada vez**, evitando conflitos ou sobreposição na escrita dos dados.

O servidor também previne duplicações através de um conjunto de registos já processados. Cada novo registo é incluído num lote, determinado com base na quantidade total de entradas. A confirmação de receção dos dados só é enviada após o processamento completo, garantindo integridade e fiabilidade nesta operação.