

# Progetto Programmazione di Reti

*Relazione progetto traccia 1*  
A.A 2020/2021

**A cura di:**

- Francesco Magnani (Matricola: 0000916594)
- Capelli Thomas (Matricola: 0000925150)
- Stricescu Ciprian (Matricola: 0000915547 )

# Indice

<b>1</b>	<b>Analisi</b>	<b>2</b>
1.1	Scelte di progetto . . . . .	2
1.2	Strutture dati . . . . .	3

# Capitolo 1

## Analisi

Il progetto, realizzato per il corso di Programmazione di Reti, consiste nello svolgimento della traccia 1.

### 1.1 Scelte di progetto

Il codice è composto principalmente da tre parti:

- **4 Device:** Per facilitare la correzione e il debug abbiamo deciso di mantenere, consapevoli delle ripetizioni, sia il codice di 4 dispositivi differenti, ognuno con il proprio indirizzo di classe C prestabilito, sia quello di un unico dispositivo che è da avviare più volte e richiede in input un indirizzo IP.  
Principalmente ogni device ha la funzione di campionare i dati necessari salvandoli su file secondo la corretta formattazione richiesta dalla consegna; inoltre, utilizza il protocollo UDP per comunicare con il gateway. In particolare la connessione viene simulata utilizzando la porta 10000 e localhost.  
La generazione dei dati avviene in maniera dinamica attraverso l'utilizzo di:
  - funzioni gaussiane per ottenere valori pseudo randomici (Temperatura e umidità).
  - utilizzo della libreria Time per ottenere la data relativa al campionamento.
- **4 Device (script riutilizzabile):** Simile al singolo device ma presenta alcune differenze per poter riutilizzare lo stesso script per i 4 device.

Per la creazione del file con i dati raccolti abbiamo deciso di non usare un file solo ma crearne uno in base all'IP del device. Per evitare problemi con l'estensione .txt abbiamo usato gli underscore per il nome del file.

La funzione checkIP prende da input l'IP dell'host e controlla che sia giusto (indirizzo di classe C) attraverso l'uso di split e confronti con lunghezze. Se l'indirizzo è corretto viene mandato un messaggio al gateway.

- **1 Gateway:** La gestione dei dati inviati dai device avviene attraverso una connessione UDP al gateway. I dati vengono manipolati e adattati per poter generare un header e mandarli al cloud attraverso una connessione TCP.

Per il corretto invio dei dati al cloud abbiamo ritenuto opportuno usare una metodo per "catalogare" i vari device attraverso il loro indirizzo IP. La connessione TCP avviene soltanto quando tutti e 4 i device inviano i loro dati, e in caso di più dati inviati dallo stesso device si limita soltanto ad aggiornare i dati nel gateway.

Nel caso di utilizzo dello script riutilizzabile per i device il gateway controlla il messaggio ricevuto attraverso processData. I controlli principali si basano sulla presenza di una keyword nel messaggio (Dev) e la presenza o meno dell'indirizzo IP tra quelli già occupati. Il gateway in questo modo è allo stesso momento sia un server UDP che un client TCP (utilizzando due porte differenti).

- **1 Cloud:** Per la realizzazione del cloud abbiamo fatto ricorso ad un classico server TCP, con le opportune modifiche, viste le necessità progettuali. La funzione del cloud è quella di ricevere i dati dal gateway attraverso una connessione TCP e formattarli in modo da presentarli all'utente.

## 1.2 Strutture dati

In questo progetto abbiamo utilizzato:

- **File:** Vengono utilizzati dai device per salvare le informazioni campionate che successivamente vengono lette per creare il messaggio da inviare.
- **Dictionary:** Usato nel gateway per gestire i dati inviati dai device. L'uso di associazione chiave-valore permette di poter semplicemente aggiornare i dati in caso di riinvio prematuro (indirizzo IP come chiave).

- **List:** Usate nel gateway per la gestione dei dati nella versione un solo client per i 4 device.