**Progetto: Sistema di rilevamento temperatura e umidità.**

**Descrizione e analisi in dettaglio**

Partendo da un’attenta analisi del testo del problema e in base alle richieste fatte, abbiamo deciso di scomporre il problema il problema in 4 parti in modo tale che ogni membro del gruppo si potesse dedicare appieno ad un’attività e finirla nel minor tempo possibile. La suddivisione è stata la seguente:

* **Parte Arduino**: abbiamo analizzato degli schemi logici in modo tale da verificare il corretto collegamento dei componenti sulla breadboard e sull’Arduino. Dopodiché abbiamo scaricato la libreria per far funzionare il componente DHT11 e le abbiamo implementate all’interno delle librerie di Arduino. Successivamente ci siamo dedicati alla scrittura di un programma che leggesse i valori di temperatura e di umidità e le visualizzasse sul monitor seriale (Della Torre);
* **Parte Client**: la scrittura del programma client è stata suddivisa a sua volta in due in modo tale da scomporre il più possibile il problema:
  + **Interfacciamento del programma Java con Arduino**: abbiamo creato una seriale, all’interno del programma Java in modo tale che gli potessimo collegare il programma scritto in Arduino (Marzorati);
  + **Inserimento dei dati rilevati all’interno di un ArrayList ed invio tramite socket al Server**: abbiamo deciso di salvare le varie rilevazioni all’interno di un ArrayList<Rilevazioni> in modo tale di avere maggior semplicità nel gestire i dati e effettuare delle modifiche, sfruttando i metodi già presenti con la classe ArrayList. Per inserire i dati, abbiamo creato una classe a parte, con i relativi metodi get e set, in modo tale da non avere dei problemi con i dati. Ci siamo preoccupati di sovrascrivere i dati una volta arrivati al 100, inserendo un contatore, e ripartendo da inizio del vettore per scrivere i dati (Fornaro).
* **Parte Server**: abbiamo realizzato un programma che rimanesse in continua attesa dei dati invitati dai vari client (il server può elaborare i dati da diversi client). Una volta che il server riceve tramite socket l’ArrayList, chiude temporaneamente la socket, apre un file e fa un’append in modo tale che i dati vengano scritti all’interno di esso e che siano scritti sempre nella posizione esatta (dopo l’ultima riga) (Fornaro);
* **Parte diagramma ER e protocolli**: ci siamo dedicati a fare una struttura di un diagramma ER tenendo conto di una possibile espansione, in futuro, inserendo la possibilità di scrivere i dati all’interno di un database posto in rete, invece che un file posto in locale. Successivamente ci siamo occupati di disegnare i vari protocolli di comunicazione che ci sono fra:
  + **Arduino e Client**;
  + **Client e Server**;

(Della Torre & Zocca)

**Commenti:** Infine ci siamo dedicati ad una parte di commento dei vari programmi in moda tale da rendere il tutto più semplice da leggere, nel caso in cui il programma deve essere analizzato da degli esterni oppure se deve essere ripreso dopo diverso tempo. (Marzorati)

**Componenti utilizzati:**

* Arduino;
* Sensore DHT11;

**Software utilizzati:**

* Arduino IDE;
* NetBeans IDE;
* Librerie Arduino DHT11;

**Competenze software sviluppate:**

* Utilizzazione ArrayList;
* Implementazione Socket;
* Scrittura su file;
* Utilizzo librerie Arduino;