

*Relazione del Progetto relativo l'insegnamento di Sistemi Operativi  
per la sessione Estiva 2020/2021.*

*Relatori:*

Di Fabrizio Giacomo,  
Montanari Matteo Marco

*Docente:*

Prof. Emanuele Lattanzi

*1) Studenti che hanno realizzato il progetto*

Il progetto è stato realizzato in coppia.

Studente: Di Fabrizio Giacomo

Matricola: 301591

Studente: Montanari Matteo Marco

Matricola: 299166

## *2) Descrizione della progettazione*

- Si è scelto di fornire i parametri di input quali il numero di utenti e il numero di sensori da riga di comando all'avvio del programma.
- Si è scelto di fornire i dati di output quali media e deviazione standard, calcolate sul tempo necessario a leggere i dati di temperatura e luminosità da parte degli utenti, su console al termine della simulazione.
- Si è scelto di salvare i valori di temperatura e luminosità letti da un sensore in due file di testo distinti. Ciò è stato fatto per comodità di verifica dei risultati ottenuti e per la realizzazione del grafico che mostra la relazione tra luminosità e temperatura richiesto dal punto 3 della sezione Testing della specifica. Analogamente, per la verifica del calcolo di media e deviazione standard, sono stati salvati i tempi di attesa per la lettura dei dati medi di temperatura e luminosità da parte di tutti gli Utenti nel corso della simulazione.
- Per realizzare la sincronizzazione dei thread che rappresentano i Sensori e il thread Weather Conditioner si è scelto di riferirsi al paradigma dei lettori e scrittori in quanto i Sensori devono leggere i valori di temperatura e luminosità dell'ambiente i quali vengono aggiornati, e dunque scritti, dal Weather Conditioner che risulta essere l'unico scrittore della simulazione, al contrario dei Sensori che rappresentano invece i lettori.
- Per realizzare la sincronizzazione dei thread che rappresentano i Sensori e i thread che rappresentano gli Utenti si è scelto di riferirsi al paradigma del produttore-consumatore a buffer limitato (gestito circolarmente). Infatti i Sensori fungono da produttori di valori di temperatura e luminosità mentre gli Utenti fungono da consumatori, leggendo questi dati a blocchi facendo la media di quattro misurazioni.

### *3) Descrizione dell'implementazione*

- Per la terminazione dei thread Sensori e dell'unico thread WeatherConditioner abbiamo implementato la terminazione deferita in quanto essi debbono essere terminati solamente da parte del gestore della simulazione (main) una volta conclusa l'esecuzione di tutti i thread utente.
- Per la realizzazione dei file di output si è scelto di utilizzare le classi FileWriter e BufferedWriter della libreria standard e di salvare i file nella cartella principale del progetto.
- Per memorizzare i valori di temperatura e luminosità misurati da un particolare sensore (punto 3 sezione Testing) abbiamo selezionato quest'ultimo tramite il costrutto "if" sfruttando il nome del sensore, passato come parametro al costruttore tramite il main.
- Per la mutua esclusione tra il thread WeatherConditioner e i Sensori all'interno della classe Environment abbiamo utilizzato un semaforo con un permesso (costrutto Semaphore libreria standard) al posto di un ReentrantLock poiché quest'ultimo può essere rilasciato solamente dal thread che lo ha bloccato (proprietario).
- Si è implementato il paradigma lettori-scrittori in modo da consentire il massimo parallelismo in lettura, il che comporta la possibile Starvation dello scrittore (Weather Conditioner) in presenza di un elevato numero di lettori (Sensori). La simulazione rimane comunque funzionale ai nostri scopi mantenendo i parametri espressi nella specifica.
- Nella classe Cloud si è scelto di salvare i dati di temperatura e luminosità in due buffer distinti utilizzando due Reentrant Lock per ottenere massimo parallelismo.

#### *4) Contributo degli studenti*

Dato che il progetto è stato svolto in coppia, si è deciso di suddividere in maniera equa il lavoro da svolgere.

Ogni studente dunque si è cimentato nell'implementare un oggetto condiviso a testa e due thread (o un thread e il main).

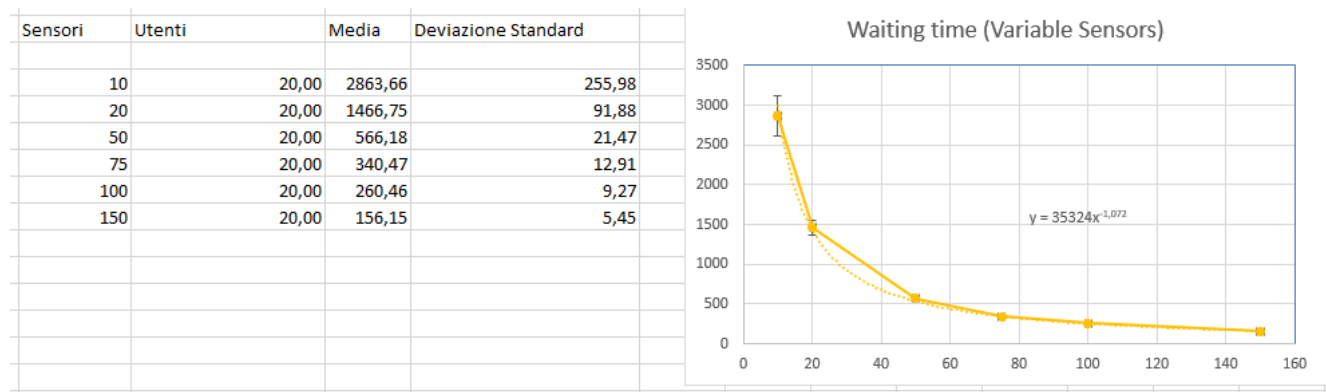
In particolare lo studente Giacomo Di Fabrizio si è occupato delle classi Environment, Sensor e WeatherConditioner, mentre lo studente Matteo Marco Montanari si è occupato delle classi Cloud, Utente e della classe InternetOfThings contenente il metodo main.

Per quanto riguarda la documentazione si è deciso di ripartire il lavoro nel seguente modo. Lo studente Matteo Marco Montanari si è occupato della stesura riguardante la descrizione della progettazione, mentre lo studente Giacomo Di Fabrizio si è occupato della descrizione dell'implementazione e del testing (con relativi grafici e analisi critica).

## 5) Testing e commenti ai risultati ottenuti

Si riportano i test eseguiti con grafici e analisi critiche.

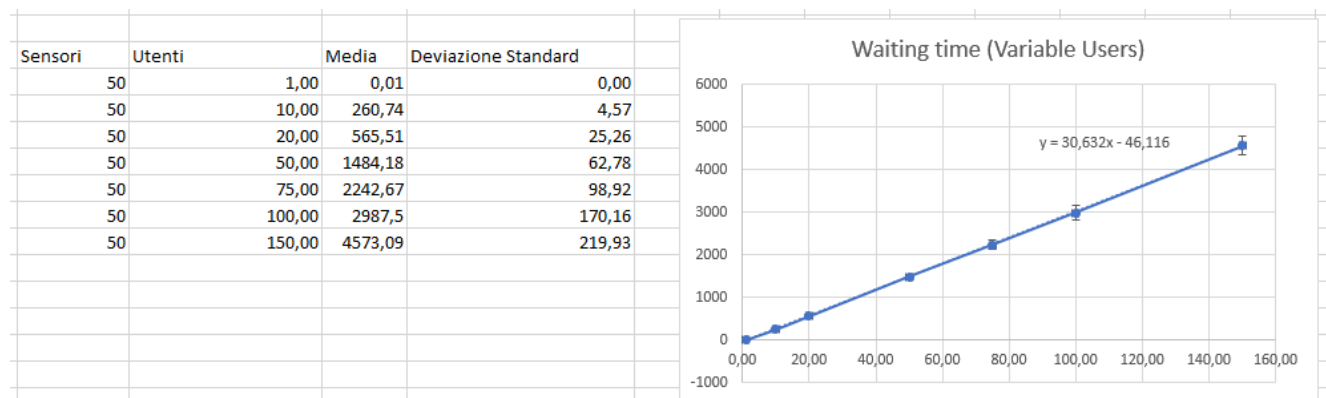
### 1. Tempo medio per utente per ottenere le letture al variare del numero di sensori



Come si può notare dal grafico, nel quale gli utenti rimangono fissi e i sensori variano, il waiting time decresce con l'aumentare del numero dei sensori (stessa cosa vale per la deviazione standard).

La linea di tendenza mostra che il grafico si comporta come un ramo di iperbole.

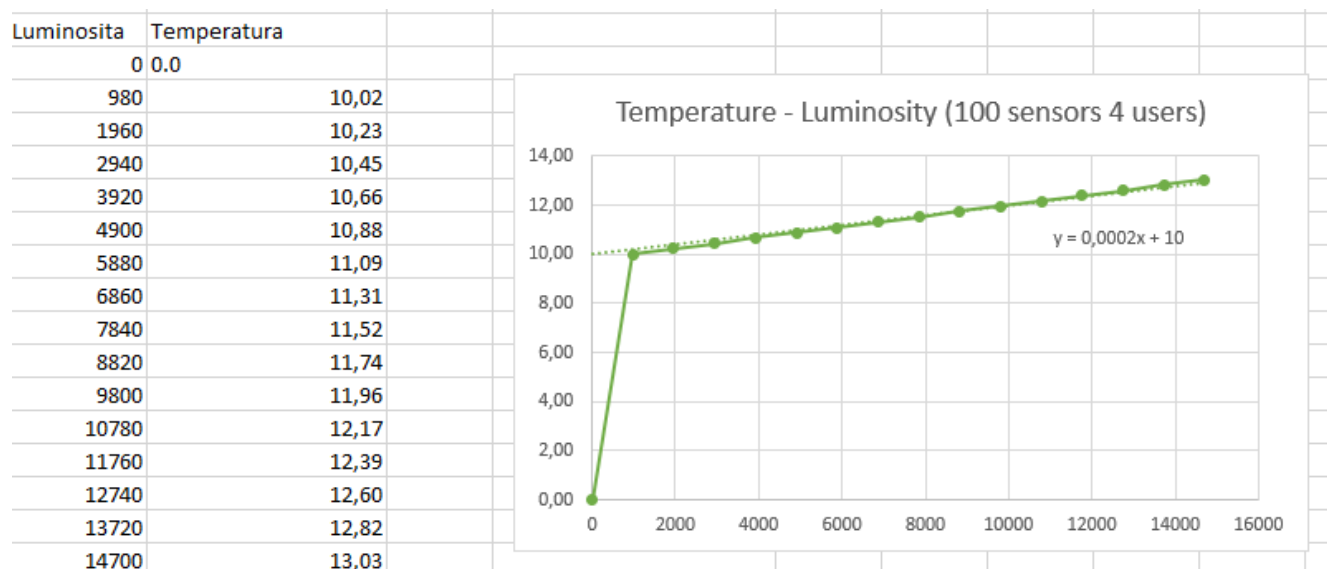
### 2. Tempo medio per utente per ottenere le letture al variare del numero di utenti



Come si può notare dal grafico, tenendo fisso il numero di sensori e variando il numero di utenti, il waiting time aumenta tendenzialmente in maniera lineare.

Anche la deviazione standard aumenta all'aumentare del numero di utenti.

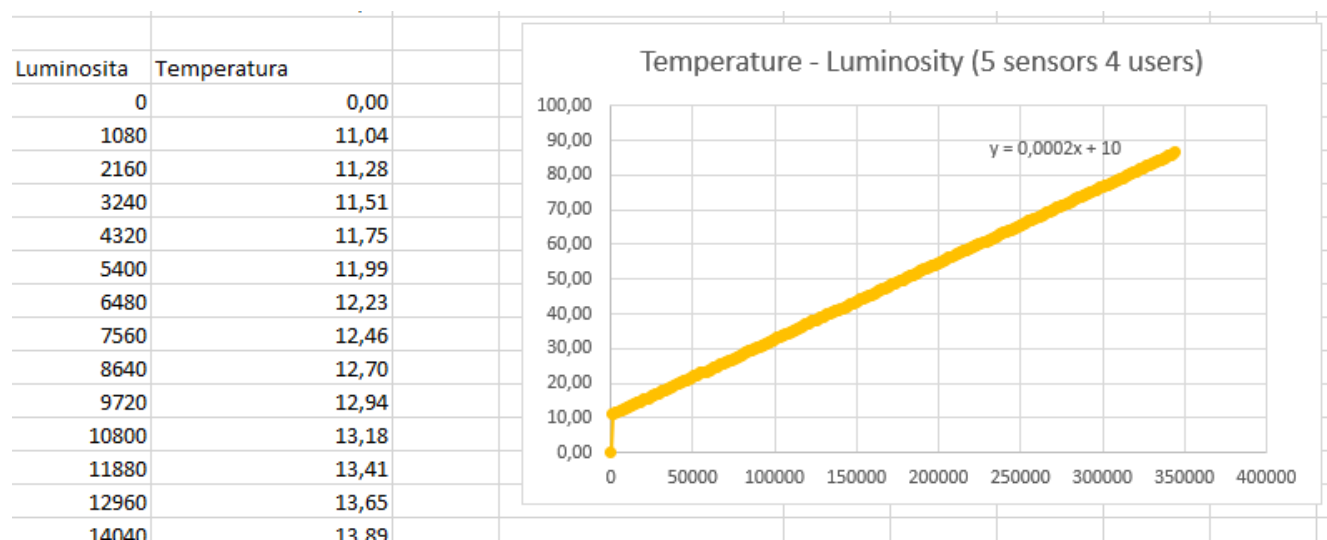
### 3. Temperatura al variare della luminosità misurate da un sensore (100 sensori, 4 utenti)



Come si può notare dal grafico la temperatura a partire dalla seconda misurazione in poi cresce linearmente secondo l'equazione  $y = 0.0002x + 10$  che è la stessa equazione che si può dedurre leggendo la specifica.

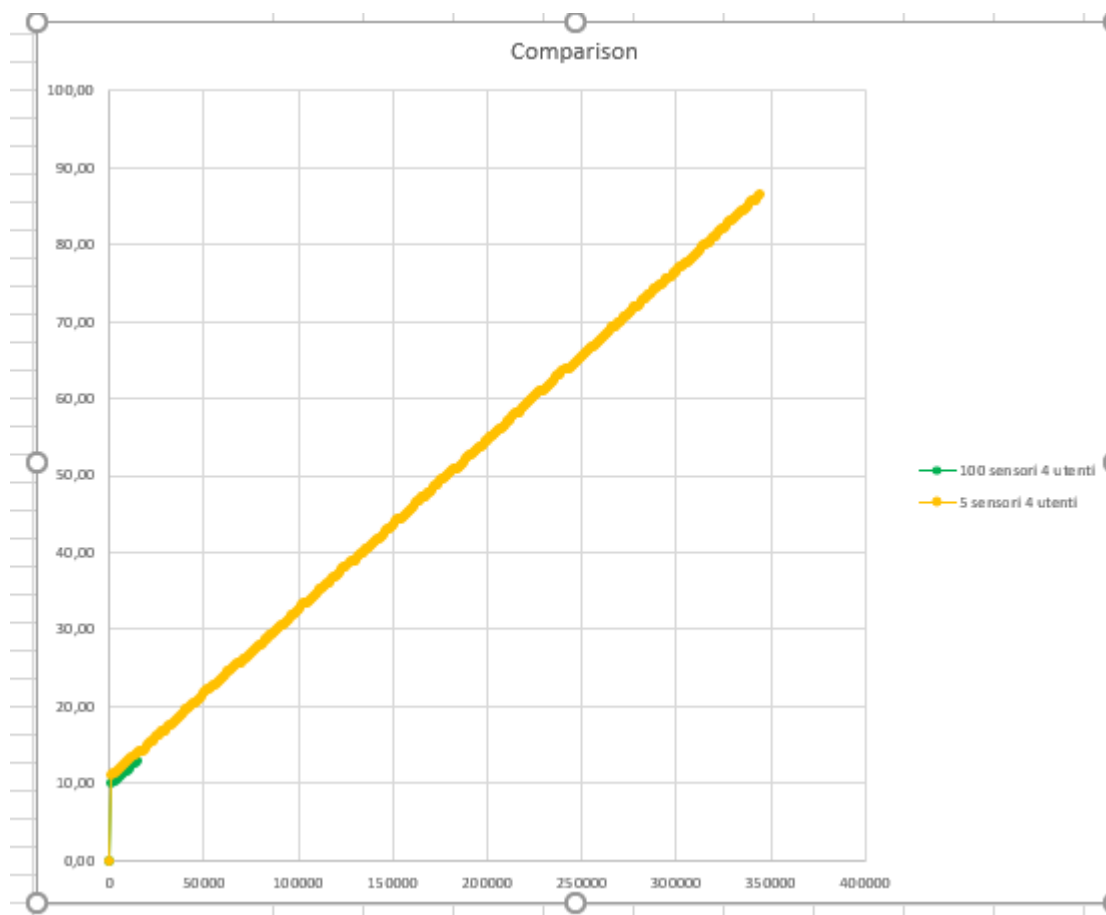
La crescita non è perfettamente lineare poiché i sensori non misurano in maniera corretta la temperatura e la luminosità presente nell'ambiente, ma applicano un errore di misurazione pseudo-casuale.

### 4. Temperatura al variare della luminosità misurata da un sensore (5 sensori, 4 utenti)



Anche in questo caso l'equazione della retta viene opportunamente rispettata, anche se il grafico risulta più marcato in quanto, dati gli input in ingresso, il sensore specifico è in grado di effettuare più misurazioni rispetto al test del grafico precedente. La cosa importante da notare è che l'equazione della retta viene mantenuta anche se ci si spinge a valori di temperatura e luminosità più elevati.

##### 5. Confronto tra i due grafici precedenti.



Ecco il confronto diretto che fa notare che l'intercetta e la pendenza delle due rette risulti circa la medesima.

*Termine della relazione*