

**Università Del Salento**

**Facoltà di Ingegneria**

Corso di Laurea in Ingegneria dell’Informazione

Documentazione di Progetto

per il corso

*Internet of Things*

**Smart Drive Panel**

Docente

*Prof. Luigi Patrono*

Tutor

*Ing. Teodoro Montanaro*

**Indice**

[*Premessa (da eliminare)* 1](#_heading=h.gjdgxs)

[Capitolo 1 – Introduzione 2](#_heading=h.30j0zll)

[Capitolo 2 – Analisi dei requisiti 3](#_heading=h.1fob9te)

[Capitolo 3 – Stato dell’Arte 4](#_heading=h.3znysh7)

[Capitolo 4 – Tecnologie utilizzate 5](#_heading=h.2et92p0)

[2.1 Tecnologia A 5](#_heading=h.tyjcwt)

[2.2 Tecnologia B 5](#_heading=h.3dy6vkm)

[Capitolo 5 – Soluzione proposta 6](#_heading=h.1t3h5sf)

[3.1 Paragrafo A 6](#_heading=h.4d34og8)

[3.2 Paragrafo B 6](#_heading=h.2s8eyo1)

[3.3 Altri paragrafi 6](#_heading=h.17dp8vu)

[Capitolo 6 – Validazione funzionale e/o prestazionale 7](#_heading=h.3rdcrjn)

[Capitolo 7 – Conclusioni e Sviluppi futuri 8](#_heading=h.26in1rg)

***Premessa (da eliminare)***

*Questo documento fornisce il formato e le principali linee guida per la realizzazione di una documentazione di buona qualità, che faciliti lo studente nella scrittura e il professore e il tutor nella lettura. Se ci sono delle variazioni importanti di questa struttura, discutere con il proprio tutor eventuali modifiche*

***Indicazioni Generali****:*

1. *Nella scrittura del documento utilizzare sempre il modo impersonale (es. “Per implementare l’applicazione è stato utilizzato Android..” e non “Per implementare l’applicazione ho utilizzato Android…”)*
2. *Curare la forma del testo, evitando frasi troppo lunghe e complesse, utilizzando opportunamente la punteggiatura e rispettando le regole dell’italiano scritto*
3. *Ad ogni immagine deve essere associata una didascalia esplicativa del contenuto, numerata secondo la convenzione “Figura Num\_capitolo.numero” (es. Figura 5.1 per la prima figura del capitolo 5)*
4. *Le immagini devono essere referenziate nel testo*
5. *I paragrafi vanno numerati secondo la numerazione del capitolo, evitando di lasciare titoli di sezioni senza numerazione*
6. *La documentazione deve avere una lunghezza media dalle 25 alle 40 pagine, in relazione all’entità del progetto*
7. *La documentazione finale in formato cartaceo va consegnata al professore una settimana prima della data dell’appello*
8. *È consigliabile inviare la documentazione in formato elettronico al proprio tutor qualche giorno prima della consegna ufficiale al professore, per una pre-validazione dei contenuti*

**Capitolo 1 – Introduzione**

**1.1 – Contesto ed Obiettivi**

La sicurezza stradale e l'efficienza nella guida dei veicoli sono temi di crescente importanza nel contesto delle moderne tecnologie di mobilità. Con l'incremento dell'adozione di sistemi di monitoraggio e telemetria, è diventato possibile raccogliere una quantità significativa di dati riguardanti il comportamento del veicolo e dello stile di guida.

Questo progetto si pone l'obiettivo di sviluppare un sistema integrato per l'acquisizione, l'analisi e la visualizzazione dei dati di guida, al fine di stimare lo stile di guida del conducente e migliorare la sicurezza e l'efficienza.

**1.2 – Criticità e Soluzioni**

La problematica principale risiede nella necessità di un sistema che possa raccogliere dati in tempo reale dai sensori degli smartphone e dei veicoli, analizzare tali dati per determinare lo stile di guida e presentare i risultati in modo chiaro e comprensibile. Attualmente, esistono diverse soluzioni commerciali e accademiche per il monitoraggio dello stile di guida, ma molte di esse presentano limitazioni in termini di precisione, tempestività e capacità di integrare dati da diverse fonti.

La nostra soluzione prevede l'uso di un'applicazione mobile (disponibile sia per Android che per iOS), *Sensor Logger*, per raccogliere dati dai sensori degli smartphone, tra cui accelerometro, giroscopio e GPS. Questi dati vengono inviati ogni secondo ad un sistema back-end in Python, che li memorizza in un database MongoDB. ~~Utilizzando un dataset fornito da una nota compagnia di assicurazioni auto, che contiene dati di 100 veicoli,~~ svilupperemo modelli di machine learning per riconoscere e stimare lo stile di guida. Il front-end sarà realizzato con una Single Page Application (SPA) in Vue.js per visualizzare i dati e i risultati dell'analisi in tempo reale.

**Capitolo 2 – Analisi dei requisiti**

**2.1 – Requisiti funzionali**

*Acquisizione dati in tempo reale*: Il sistema deve essere in grado di raccogliere dati dai sensori dello smartphone ogni secondo e inviarli al back-end. Questo richiede un'alta frequenza di acquisizione ed una connessione di rete stabile.

*Memorizzazione dei dati*: I dati acquisiti devono essere memorizzati in un database documentale noSQL (MongoDB), garantendo che tutte le informazioni rilevanti siano accessibili per analisi future.

*Analisi dello stile di guida*: Il sistema deve utilizzare tecniche di machine learning per analizzare i dati raccolti e stimare lo stile di guida del conducente, identificando comportamenti potenzialmente pericolosi come accelerazioni brusche, frenate improvvise e cambiamenti di direzione rapidi.

*Visualizzazione in tempo reale*: Il front-end deve presentare i dati raccolti e i risultati dell'analisi in tempo reale, utilizzando una SPA in Vue.js. Questo include la visualizzazione di grafici e metriche chiave che rappresentano lo stile di guida.

**2.2 – Requisiti non funzionali**

*Scalabilità*: Il sistema deve poter gestire un elevato numero di connessioni simultanee e grandi volumi di dati, assicurando che la performance rimanga costante anche sotto carichi elevati.

*Affidabilità*: Il sistema deve essere resiliente agli errori e garantire la continuità del servizio, anche in caso di problemi di rete o guasti hardware.

*Sicurezza*: I dati raccolti devono essere protetti contro accessi non autorizzati, con meccanismi di crittografia sia durante la trasmissione che nello storage.

*Performance*: Il sistema deve rispondere rapidamente alle richieste del front-end, mantenendo una latenza minima per assicurare che i dati siano visualizzati in tempo reale.

**Capitolo 3 – Stato dell’Arte**

**3.1 – Soluzioni attualmente disponibili**

**3.1.1 - Telemetria veicolare tradizionale**

Le soluzioni tradizionali di telemetria veicolare utilizzano dispositivi installati nei veicoli per raccogliere dati su velocità, posizione, e comportamento del conducente. Questi sistemi sono spesso costosi e complessi da installare e mantenere. Inoltre, richiedono hardware dedicato che può non essere facilmente aggiornabile. La nostra soluzione mira a superare queste limitazioni utilizzando smartphone, che sono già ampiamente diffusi e dotati di sensori avanzati. Questo approccio riduce i costi e semplifica l'implementazione, permettendo di beneficiare degli aggiornamenti software e hardware che gli smartphone ricevono regolarmente.

**3.1.2 - App mobile per il monitoraggio della guida**

Alcune app mobili esistenti offrono funzionalità di monitoraggio dello stile di guida utilizzando sensori dello smartphone. Tuttavia, molte di queste app mancano di precisione e non integrano dati da altre fonti, come i sensori del veicolo. La nostra soluzione integra dati da più fonti per migliorare l'accuratezza delle analisi. Inoltre, molte app esistenti non forniscono feedback in tempo reale, limitando la loro utilità nell'aiutare i conducenti a migliorare il proprio stile di guida.

**3.1.3 - Sistemi basati su GPS**

I sistemi di monitoraggio basati su GPS possono rilevare la posizione e la velocità del veicolo, ma non forniscono informazioni dettagliate su accelerazioni, frenate e altre manovre del conducente. La nostra soluzione utilizza anche dati di accelerometro e giroscopio per una visione più completa del comportamento di guida. Questo approccio consente di identificare comportamenti specifici, come brusche accelerazioni o frenate, che possono essere indicativi di uno stile di guida aggressivo o pericoloso.

**3.1.4 - Sistemi di monitoraggio avanzati**

Alcune soluzioni avanzate utilizzano hardware specifico per raccogliere dati dai sensori del veicolo. Questi sistemi possono essere precisi, ma sono costosi e non facilmente accessibili a tutti gli utenti. La nostra soluzione sfrutta la tecnologia disponibile sugli smartphone per offrire una soluzione più accessibile e versatile. Inoltre, l'uso di machine learning consente di adattare e migliorare continuamente i modelli di analisi, rendendo il sistema più efficace nel tempo.

**Capitolo 4 – Tecnologie utilizzate**

Tecnologia A: **Sensor Logger**

Sensor Logger è un'app mobile che raccoglie dati dai sensori dello smartphone, tra cui quelli d’interesse per il progetto quali: accelerometro, giroscopio e GPS. Questi dati vengono inviati al back-end in formato JSON via HTTP POST. Sensor Logger permette di configurare la frequenza di acquisizione dei dati e l'indirizzo del server destinatario, rendendola una scelta ideale per il nostro sistema di acquisizione dati. La flessibilità di Sensor Logger consente di adattare facilmente il sistema a diverse configurazioni di rete e requisiti di acquisizione.

Tecnologia B: **Python e MongoDB**

Il back-end del nostro sistema è sviluppato in Python, utilizzando framework come Flask per gestire le richieste HTTP. I dati raccolti vengono memorizzati in MongoDB, un database NoSQL che offre scalabilità e flessibilità nella gestione di grandi volumi di dati. MongoDB supporta query avanzate e aggregazioni, facilitando l'analisi dei dati raccolti. L'integrazione con Python permette di sfruttare numerose librerie per l'elaborazione dei dati e l'implementazione di algoritmi di machine learning.

Tecnologia C: **Strumenti di Machine Learning**

Per l'analisi dei dati e la stima dello stile di guida, utilizziamo librerie di machine learning come *scikit-learn* e *TensorFlow*. Queste librerie offrono strumenti potenti per la pre-elaborazione dei dati, l'addestramento di modelli e la valutazione delle prestazioni. L'uso di tecniche di machine learning ci consente di sviluppare modelli predittivi accurati basati sui dati raccolti dai sensori. Scikit-learn offre una vasta gamma di algoritmi di classificazione e regressione, mentre TensorFlow permette di costruire reti neurali complesse per modelli più sofisticati.

Tecnologia D: **Vue.js**

Il front-end del nostro sistema è sviluppato con Vue.js, un framework JavaScript per la creazione di Single Page Applications (SPA). Vue.js offre un'architettura reattiva e componenti modulari, permettendo di costruire interfacce utente dinamiche e responsive. Utilizziamo Vue.js per visualizzare i dati raccolti ed i risultati dell'analisi in tempo reale, offrendo agli utenti una visualizzazione chiara ed intuitiva del loro stile di guida. Vue.js supporta inoltre l'integrazione con librerie grafiche come *Chart.js* e *D3.js*, che permettono di creare visualizzazioni interattive e informative dei dati raccolti.

**Capitolo 5 – Soluzione proposta**

**5.1 – Implementazione del Machine Learning**

Per implementare un sistema di riconoscimento dello stile di guida, è necessario seguire i seguenti step:

*Pre-elaborazione dei dati*: I dati raccolti dai sensori devono essere pre-elaborati per rimuovere rumori e outlier. Questo include la normalizzazione dei valori dei sensori e l'applicazione di filtri per eliminare picchi anomali.

*Feature Engineering*: Creazione di caratteristiche rilevanti dai dati grezzi. Ad esempio, possiamo calcolare l'accelerazione media, la variazione dell'angolo di sterzata, la velocità media e la frequenza delle frenate brusche.

*Divisione del dataset*: Il dataset deve essere diviso in set di addestramento e di test per valutare le prestazioni dei modelli. Possiamo utilizzare tecniche di cross-validation per garantire che i modelli siano robusti.

*Addestramento del modello*: Utilizzando librerie di machine learning, possiamo addestrare diversi modelli (es. Random Forest, SVM, Reti Neurali) sui dati pre-elaborati.

*Valutazione del modello*: Misurare le prestazioni del modello utilizzando metriche come accuratezza, precisione, richiamo e F1-score. Questo ci aiuta a selezionare il modello più appropriato per il riconoscimento dello stile di guida.

*Implementazione del modello*: Una volta selezionato il modello migliore, possiamo integrarlo nel sistema backend per analizzare i dati in tempo reale e fornire feedback istantanei agli utenti.

Questo processo ci permette di creare un sistema robusto e affidabile per il riconoscimento dello stile di guida, utilizzando tecniche di machine learning avanzate e sfruttando i dati raccolti dai sensori degli smartphone e dei veicoli.

**Capitolo 6 – Validazione funzionale e/o prestazionale**

Spiegare come è stata validata, dal punto di vista funzionale e/o prestazionale, l’applicazione realizzata. Per fare questo, si può descrivere un caso d’uso reale, mostrare l’eventuale hardware utilizzato e descrivere come viene utilizzato in tale scenario, corredando la descrizione con opportuni screenshot. Numerare e fornire didascalia a tutte le immagini inserite.

**Capitolo 7 – Conclusioni e Sviluppi futuri**

Riassumere l’obiettivo, evidenziando come questo sia stato effettivamente raggiunto, e accennare ad eventuali sviluppi che possono migliorare il presente lavoro.