

UNIVERSITÀ DEGLI STUDI DI PADOVA



SCUOLA DI SCIENZE

CORSO DI LAUREA IN INFORMATICA

Piano di lavoro

Studente:

Gabriele Isacco MAGNELLI - 2075542

Azienda:

CWBI

5 settembre 2025

Contatti

Studente: Gabriele Isacco Magnelli, gabrieleisacco.magnelli@studenti.unipd.it, + 39 347 317 0439

Tutor aziendale: Roberto Martina, ingmcrm@gmail.com, + 39 389 788 7744

Azienda: CWBI, Via Venezia 92/B, Padova (PD), <https://www.cwbi.eu/it/>

Informazioni sull'azienda

CWBI è una software house specializzata nello sviluppo di soluzioni innovative per il settore bancario e finanziario. La nostra missione è unire tecnologia e ricerca applicata per migliorare i processi e la qualità del software.

Scopo dello stage

Il progetto di stage riguarda la creazione di un modello di intelligenza artificiale (LLM) in grado di supportare lo sviluppo e la qualità del codice prodotto dal team. Le fasi chiave del progetto includono:

- Analisi del codice sorgente prodotto dagli sviluppatori.
- Rilevamento delle non conformità rispetto a standard e best practice aziendali.
- Segnalazione automatica degli errori individuati.
- Proposta ed eventuale esecuzione di correzioni automatiche.

Il risultato atteso è un modello LLM in grado di:

- Analizzare in modo accurato il codice prodotto dai nostri sviluppatori, identificando punti critici o potenziali errori.
- Individuare deviazioni dagli standard e dalle best practice, migliorando la qualità complessiva del software.
- Fornire segnalazioni chiare e comprensibili sugli errori rilevati, facilitando l'intervento dei programmatori.
- Proporre e, quando possibile, applicare correzioni automatiche per ridurre i tempi di debugging e aumentare l'efficienza del processo di sviluppo.

Interazione tra studente e tutor aziendale

Regolarmente ci saranno incontri diretti con il tutor aziendale Roberto Martina per verificare lo stato di avanzamento, chiarire eventualmente gli obiettivi, affinare la ricerca e aggiornare il piano stesso di lavoro.

Pianificazione del lavoro

Pianificazione settimanale

• Prima Settimana (XX ore)

- Incontro con persone coinvolte nel progetto per discutere i requisiti e le richieste relativamente al sistema da sviluppare;
- Verifica credenziali e strumenti di lavoro assegnati;



- Presa visione dell'infrastruttura esistente;
- Formazione sulle tecnologie adottate;

• **Seconda Settimana - Sottotitolo (XX ore)**

- ;

• **Terza Settimana - Sottotitolo (XX ore)**

- ;

• **Quarta Settimana - Sottotitolo (XX ore)**

- ;

• **Quinta Settimana - Sottotitolo (XX ore)**

- ;

• **Sesta Settimana - Sottotitolo (XX ore)**

- ;

• **Settima Settimana - Sottotitolo (XX ore)**

- ;

• **Ottava Settimana - Conclusione (XX ore)**

- ;



Ripartizione ore

La pianificazione, in termini di quantità di ore di lavoro, sarà così distribuita:

Durata in ore	Descrizione dell'attività
38	Formazione sulle tecnologie
38	Definizione architettura di riferimento e relativa documentazione
12	<i>Analisi del problema e del dominio applicativo</i>
22	<i>Progettazione della piattaforma e relativi test</i>
4	<i>Stesura documentazione relativa ad analisi e progettazione</i>
38	Collaudo Finale
30	<i>Collaudo</i>
5	<i>Stesura documentazione finale</i>
1	<i>Incontro di presentazione della piattaforma con gli stakeholders</i>
2	<i>Live demo di tutto il lavoro di stage</i>
Totale ore	



Obiettivi

Notazione

Si farà riferimento ai requisiti secondo le seguenti notazioni:

- *O* per i requisiti obbligatori, vincolanti in quanto obiettivo primario richiesto dal committente;
- *D* per i requisiti desiderabili, non vincolanti o strettamente necessari, ma dal riconoscibile valore aggiunto;
- *F* per i requisiti facoltativi, rappresentanti valore aggiunto non strettamente competitivo.

Le sigle precedentemente indicate saranno seguite da una coppia sequenziale di numeri, identificativo del requisito.

Obiettivi fissati

Si prevede lo svolgimento dei seguenti obiettivi:

Requisiti (O,D,F)	Attività	Descrizione
Requisiti Obbligatori		
O1	Analisi del codice	Il modello deve essere in grado di leggere e interpretare codice sorgente prodotto dagli sviluppatori
O2	Rilevamento delle non conformità	Il modello deve individuare errori o deviazioni dagli standard e best practice aziendali
O3	Segnalazione automatica degli errori	Ogni non conformità deve essere riportata chiaramente agli sviluppatori
O4	Addestramento con dataset etichettato	Il progetto deve prevedere la raccolta e l'annotazione di codice conforme/non conforme
O5	Integrazione nella pipeline CI/CD	Il modello deve essere utilizzabile in contesti di sviluppo reale senza bloccare i processi
O6	Test e validazione	Le correzioni proposte devono essere testate per verificare efficacia e sicurezza
O7	Documentazione	Fornire una documentazione completa per l'uso e la manutenzione del sistema
Requisiti Desiderabili		
D1	Correzione automatica degli errori	Il modello dovrebbe proporre e, se possibile, applicare correzioni al codice
D2	Espandibilità e aggiornabilità del dataset	Il sistema deve permettere di aggiornare facilmente il modello con nuovi snippet o standard
D3	Interpretabilità delle decisioni	Il modello dovrebbe fornire spiegazioni sul perché segnala un errore o propone una correzione
Requisiti Facoltativi		
F1	Ottimizzazione delle prestazioni	Il modello dovrebbe essere in grado di gestire grandi volumi di codice senza degradare le prestazioni
F2	Analisi di sicurezza	Il modello dovrebbe identificare potenziali vulnerabilità nel codice