UNIVERSITÀ DEGLI STUDI DI PADOVA

DIPARTIMENTO MATEMATICA

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

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 - Fondamenti e preparazione (40 ore)**
  - Studio di ML, NLP e LLM applicati al codice;
  - Installazione ambienti, librerie Python e IDE;
  - Familiarizzazione con repository e pipeline CI/CD;

  - Documentazione dell'ambiente e primi esperimenti;
  - Formazione sulle tecnologie adottate;

- **Seconda Settimana - Analisi del codice e raccolta dati (38 ore)**
  - Raccolta di codice conforme/non conforme;
  - Annotazione manuale degli errori e non conformità;
  - Creazione dataset iniziale pronto per ML;

- **Terza Settimana - Preprocessing e rappresentazione del codice (38 ore)**
  - Tokenizzazione e pulizia del dataset;
  - Studio e implementazione di embedding per il codice;
  - Automatizzazione preprocessing e validazione dataset;

- **Quarta Settimana - Addestramento iniziale del modello (38 ore)**
  - Configurazione modello LLM e parametri di training;
  - Addestramento su subset del dataset;
  - Monitoraggio delle metriche e prima valutazione;

- **Quinta Settimana - Fine-tuning e ottimizzazione (38 ore)**
  - Fine-tuning su dataset specifico aziendale;
  - Analisi dei casi di errore e miglioramento iterativo;
  - Tecniche di data augmentation;
  - Studio di interpretabilità del modello;

- **Sesta Settimana - Integrazione in pipeline CI/CD (38 ore)**
  - Studio e configurazione della pipeline CI/CD;
  - Integrazione del modello e scripting per test automatici;
  - Logging, gestione feedback e validazione preliminare;

- **Settima Settimana - Validazione e test delle correzioni (36 ore)**
  - Test su nuovo codice non incluso nel training;
  - Valutazione metriche di accuratezza e recall;
  - Iterazioni per miglioramento del comportamento del modello;

- **Ottava Settimana - Documentazione e presentazione finale (32 ore)**
  - Redazione documentazione tecnica completa;
  - Creazione esempi pratici e guide d'uso;
  - Preparazione presentazione finale al team;
  - Presentazione e discussione dei risultati;

Ripartizione ore

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

Durata in ore	Descrizione attività
50	**Analisi delle esigenze e preparazione** • Studio di ML, NLP e LLM applicati al codice • Familiarizzazione con repository, pipeline CI/CD e ambienti di sviluppo • Definizione dei requisiti per il modello e raccolta informazioni preliminari
130	**Progettazione e implementazione del modello** • Creazione del dataset iniziale conforme/non conforme • Preprocessing, tokenizzazione e embedding del codice • Configurazione e addestramento iniziale del modello LLM • Fine-tuning e ottimizzazione su dataset specifico aziendale
50	**Testing e ottimizzazione** • Monitoraggio delle metriche e valutazione del comportamento del modello • Analisi dei casi di errore e miglioramento iterativo • Applicazione di tecniche di interpretabilità e data augmentation
54	**Integrazione e validazione in pipeline CI/CD** • Studio e configurazione della pipeline CI/CD • Integrazione del modello con test automatici • Logging, gestione feedback e validazione preliminare • Test su nuovo codice non incluso nel training
16	**Documentazione e presentazione finale** • Redazione documentazione tecnica completa • Creazione esempi pratici, guide d'uso e presentazione al team
Totale ore	**300**

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