

INGEGNERIA DEL SOFTWARE – 2025–26

INTRODUZIONE AL CORSO

LEZIONE 1
29/09/2025
VINCENZO RICCIO



INFORMAZIONI SUL DOCENTE

INFORMAZIONI SUL DOCENTE

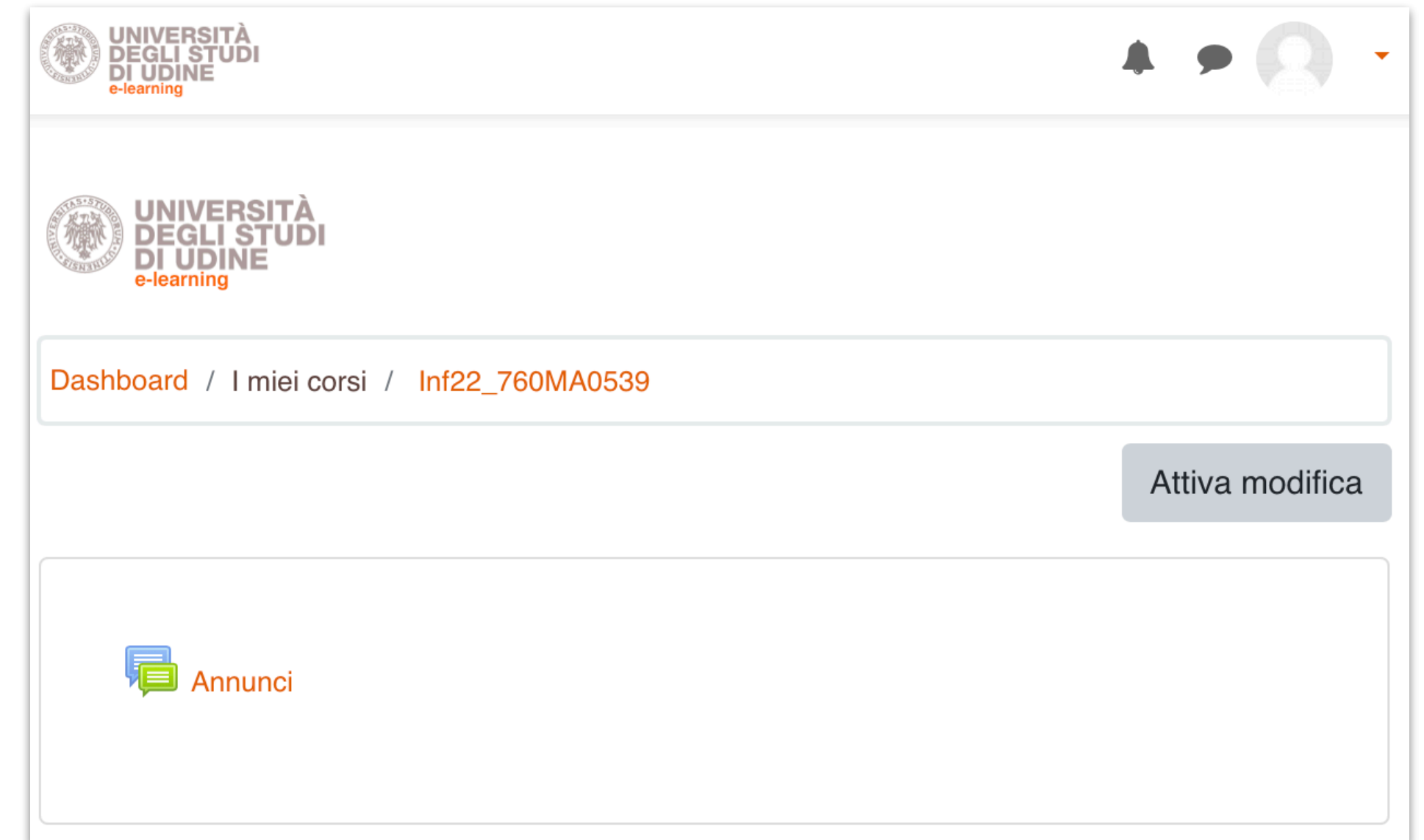
Vincenzo Riccio

- E-mail: vincenzo.riccio@uniud.it
- Per avvisi e materiale didattico:
<https://elearning.uniud.it/>

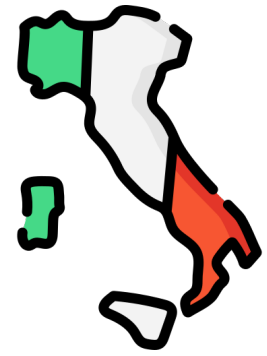
Orario di Ricevimento:

- Lunedì 15:00 - 17:00
- Sede Rizzi, L2-02-BC (a.k.a. A2 129)*

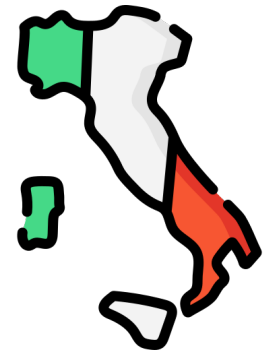
*** PREVIO CONTATTO MAIL**



MY JOURNEY



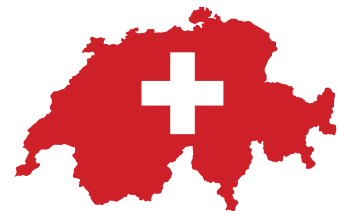
M.SC IN COMPUTER SCIENCE AND ENGINEERING
UNIVERSITÀ DEGLI STUDI DI NAPOLI "FEDERICO II", ITALY
TEST ADEQUACY FOR AUTOMOTIVE EMBEDDED SYSTEMS



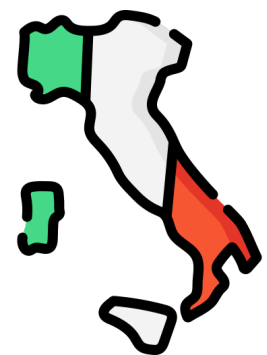
PH.D. IN INFORMATION TECHNOLOGY
UNIVERSITÀ DEGLI STUDI DI NAPOLI "FEDERICO II", ITALY
TEST AUTOMATION FOR ANDROID APPS



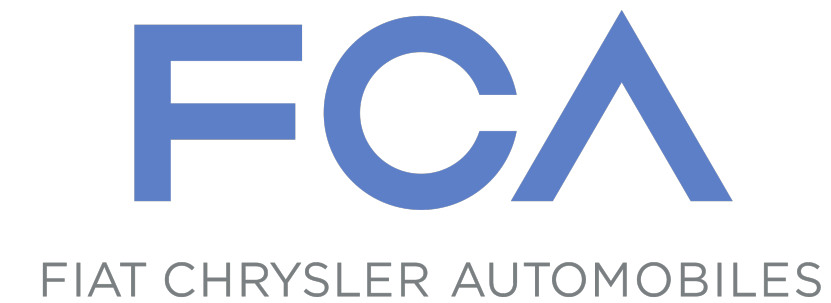
VISITING RESEARCHER
UNIVERSITY OF PASSAU, GERMANY
SEARCH-BASED TESTING FOR ANDROID APPS



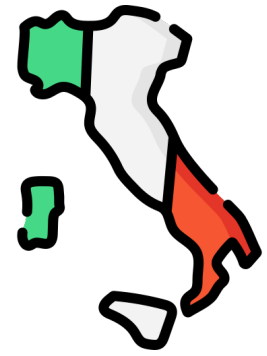
POST-DOC
UNIVERSITÀ DELLA SVIZZERA ITALIANA, SWITZERLAND
TEST INPUT GENERATION FOR MACHINE LEARNING BASED SYSTEMS



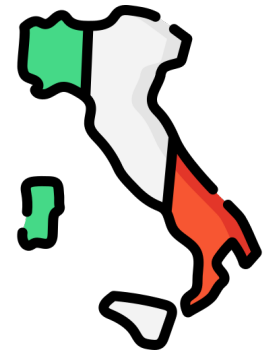
RTDB
UNIVERSITÀ DI UDINE, ITALY
SOFTWARE ENGINEERING AND TESTING



MY JOURNEY



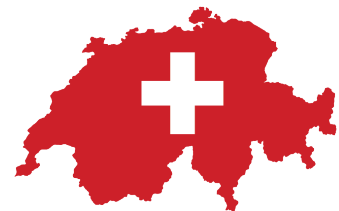
M.SC IN COMPUTER SCIENCE AND ENGINEERING
UNIVERSITÀ DEGLI STUDI DI NAPOLI “FEDERICO II”, ITALY
TEST ADEQUACY FOR AUTOMOTIVE EMBEDDED SYSTEMS



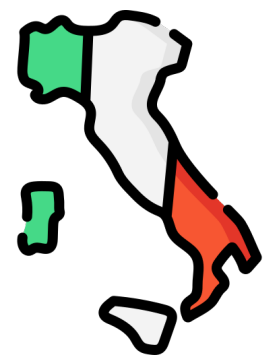
PH.D. IN INFORMATION TECHNOLOGY
UNIVERSITÀ DEGLI STUDI DI NAPOLI “FEDERICO II”, ITALY
TEST AUTOMATION FOR ANDROID APPS



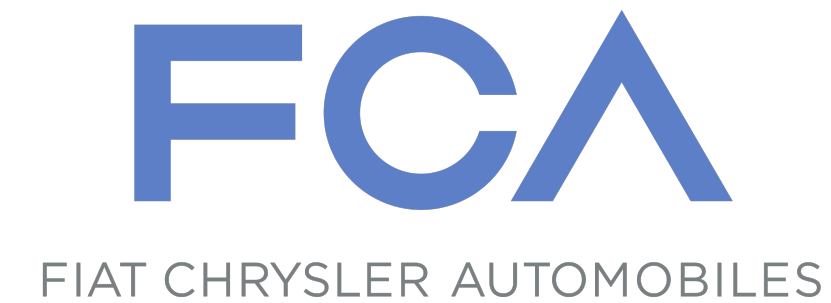
VISITING RESEARCHER
UNIVERSITY OF PASSAU, GERMANY
SEARCH-BASED TESTING FOR ANDROID APPS



POST-DOC
UNIVERSITÀ DELLA SVIZZERA ITALIANA, SWITZERLAND
TEST INPUT GENERATION FOR MACHINE LEARNING BASED SYSTEMS



RTDB
UNIVERSITÀ DI UDINE, ITALY
SOFTWARE ENGINEERING AND **TESTING**

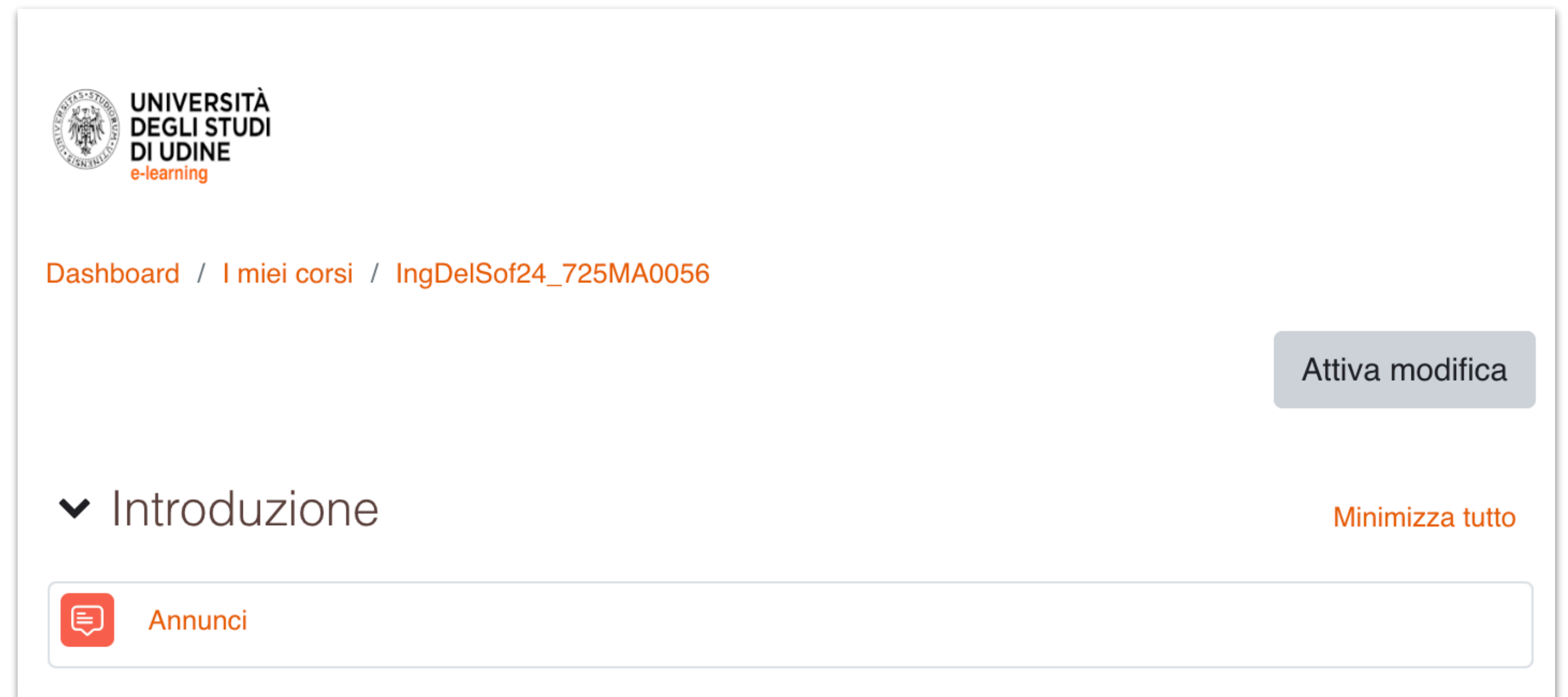




INFORMAZIONI SUL CORSO

INFORMAZIONI SULLE PIATTAFORME ONLINE

- Useremo esclusivamente la piattaforma e-learning per il materiale del corso
- La pagina di riferimento è il corso 725
- Teams sarà utilizzato per eventuale ricevimento online



INFORMAZIONI SUL CORSO

Il materiale sarà caricato sulla piattaforma di e-learning nello spazio riservato al corso

- Aggiornato prima e dopo ogni lezione svolta

La durata del corso è di 48 ore

Le lezioni si svolgeranno:

- **Lunedì 8:30-10:30**, Aula C2 Rizzi
- **Mercoledì 8:30-10:30**, Aula C3 Rizzi

OBIETTIVI DEL CORSO

Gli obiettivi del corso sono:

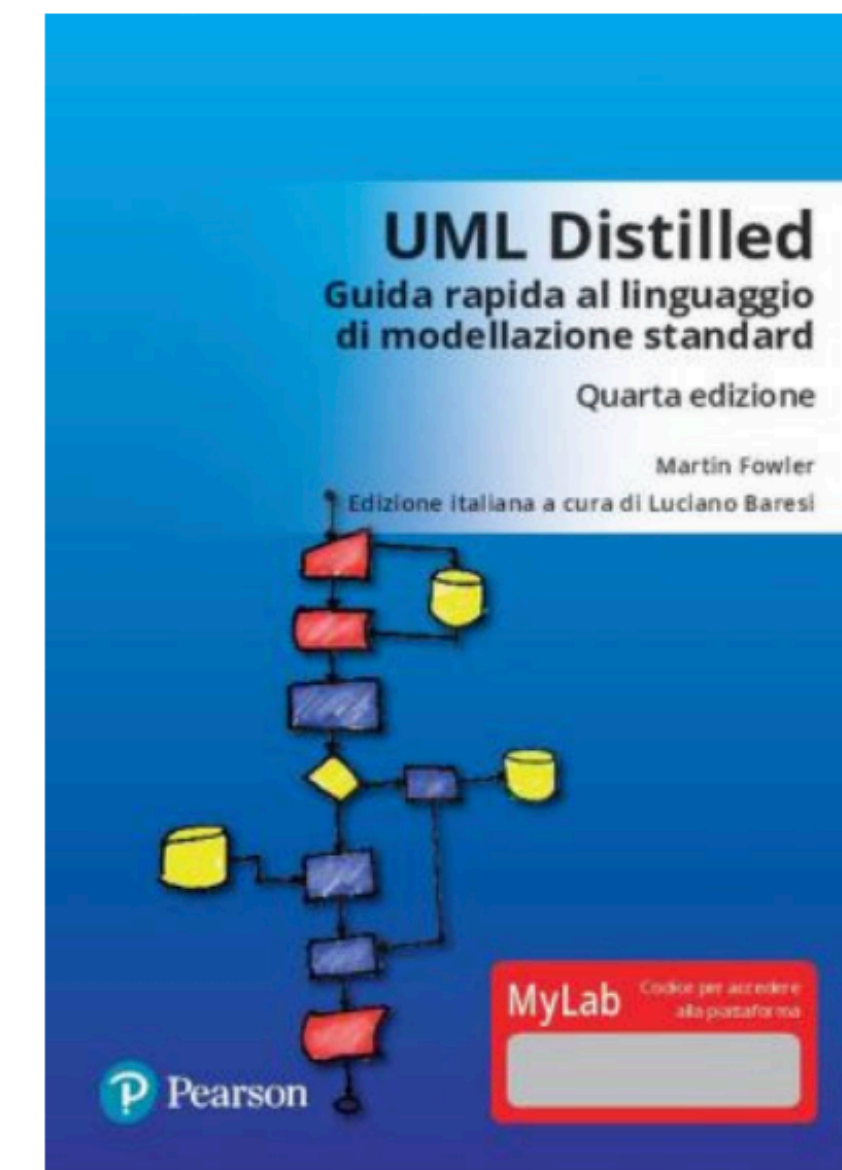
- Conoscere le nozioni di base riguardo l'**ingegneria del software**
- Comprendere i principali problemi di un moderno **processo di sviluppo software**
- Acquisire capacità di **analizzare e progettare** in modo professionale sistemi software di complessità non elementare, e valutarne la **qualità**

MATERIALE DIDATTICO: TESTI DI RIFERIMENTO

Libri di testo:

- “Ingegneria del Software”, I. Sommerville, 10 ed., Pearson 2017
- “UML distilled”, M. Fowler, 3a ed., Pearson, 2015

Copia dei lucidi e del materiale integrativo è fornito dal docente sulla piattaforma di e-learning

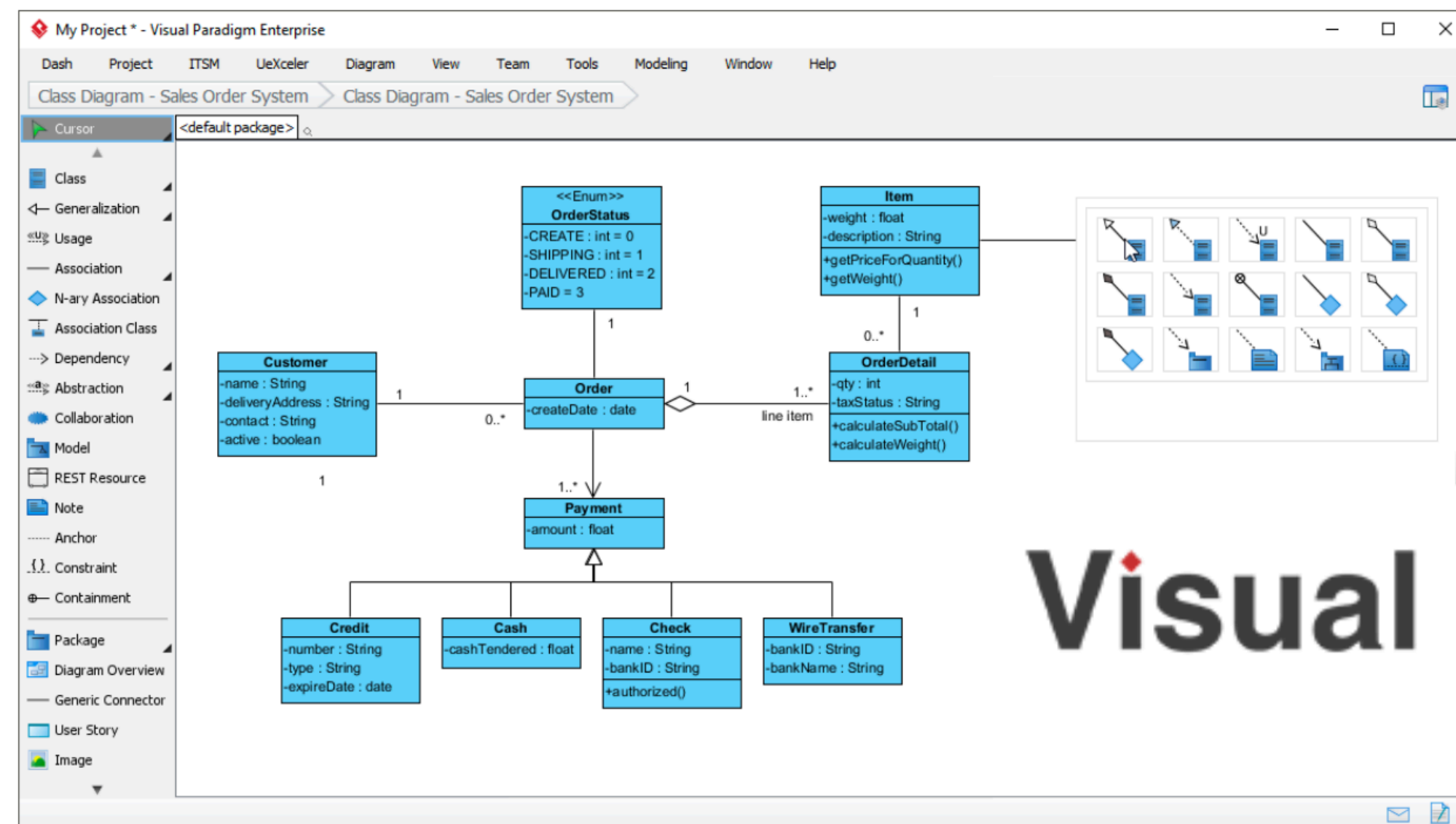


MATERIALE DIDATTICO: SOFTWARE DI RIFERIMENTO

Visual Paradigm Community Edition Disponibile al link:

<https://www.visual-paradigm.com/download/community.jsp>

FREE for non commercial use only





ESAME FINALE

PROVA D'ESAME

- La prova d'esame consiste in un compito scritto (carta e penna), della durata massima indicativa di 120 minuti
- La prova d'esame è divisa in tre parti:
 - ⦿ Esercizi di analisi e modellizzazione
 - ⦿ Esercizi su testing
 - ⦿ Domande di teoria
- Per il superamento della prova è necessario che **tutte queste tre** le parti siano svolte in maniera adeguata (voto almeno sufficiente)

REGOLAMENTO

- Durante l'esame NON è consentito l'uso degli appunti.
- Non è consentito l'uso di altri dispositivi: in particolare, non devono essere presenti portatili, cellulari, smartwatch, auricolari, etc. Essi saranno posti sulla scrivania della postazione all'inizio della prova.
- Ogni forma di comunicazione tra gli studenti è vietata durante l'esame.

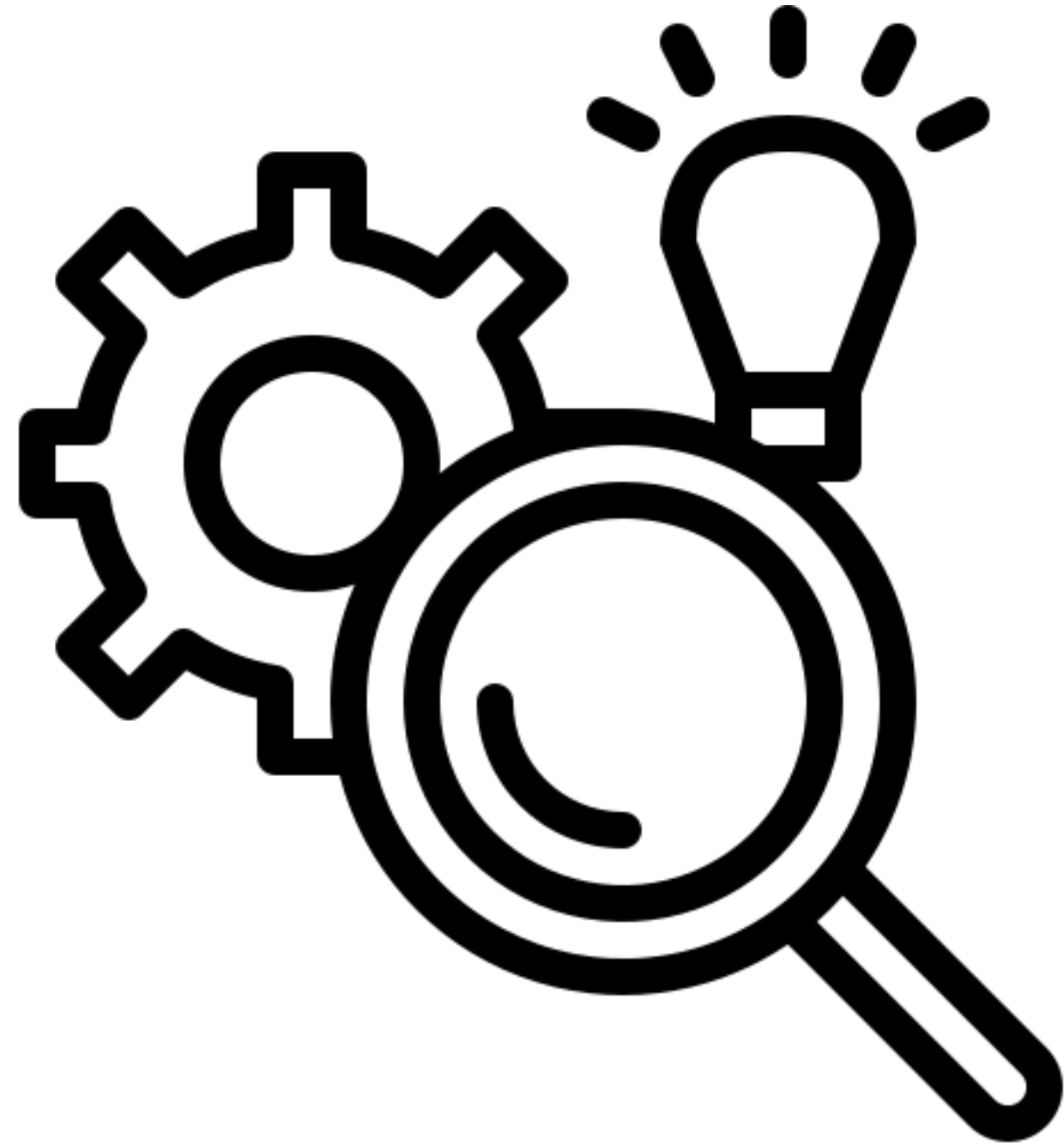
TEMPLATE DELLA TRACCIA

- **Traccia contenente i requisiti**
- **Prima parte: Analisi e Modellazione**
 - ◎ 1.2: Esercizio sui casi d'uso, ~2 punti, rappresentare diagramma senza descrizione degli scenari (salvo diverse indicazioni)
 - ◎ 1.3: Esercizio su Class Diagram, ~10 punti, rappresentare system domain model di alto livello (salvo indicazioni specifiche)
 - ◎ 1.4: Esercizi su diagrammi comportamentali, ~4-6 punti, rappresentare i casi d'uso indicati attraverso uno o due sequence/activity/statechart diagrams
- **Seconda parte: Testing**
 - ◎ 2: Esercizi su testing, ~4-6 punti, uno o due esercizi
- **Terza parte: Teoria**
 - ◎ 3: Domande di teoria, 6-8 punti, una o due domande che possono spaziare su tutto il programma

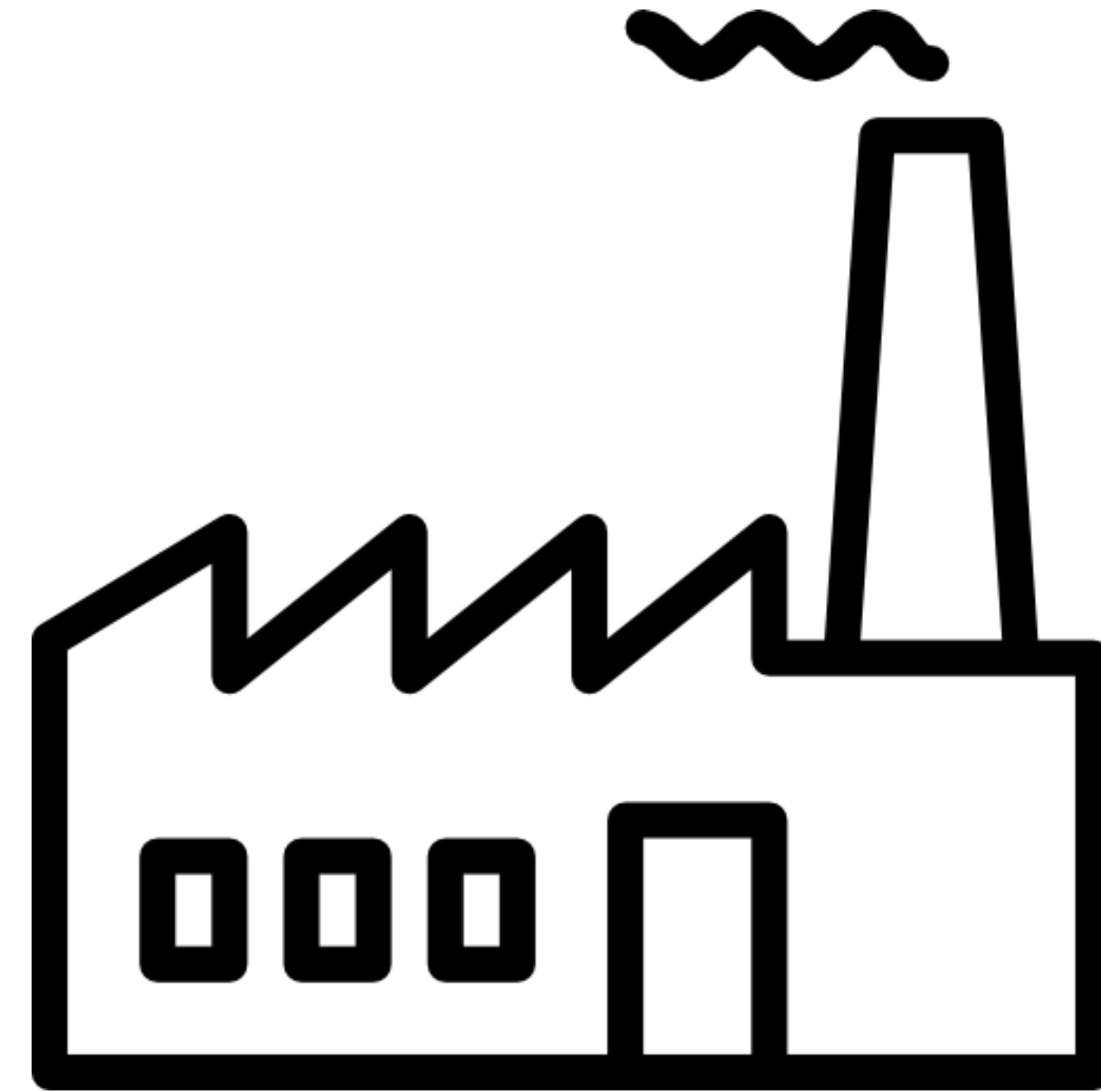


TESI

TIPOLOGIA DI SUPERVISIONE



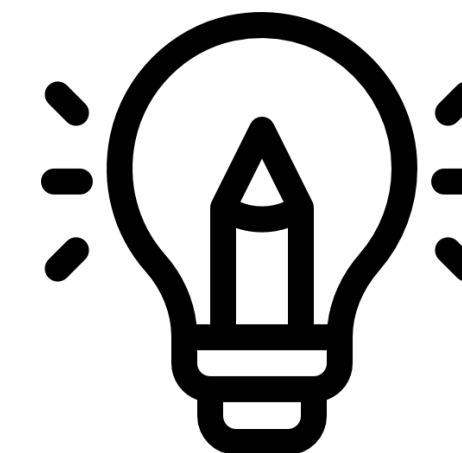
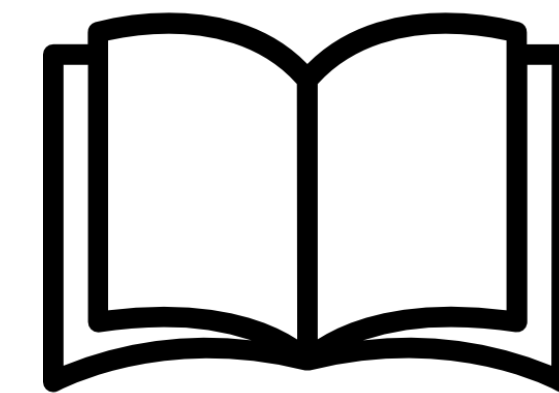
Tesi intra-moenia



Tesi extra-moenia

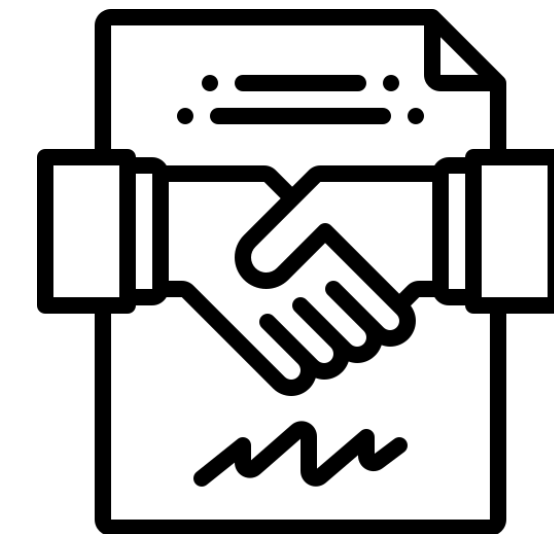
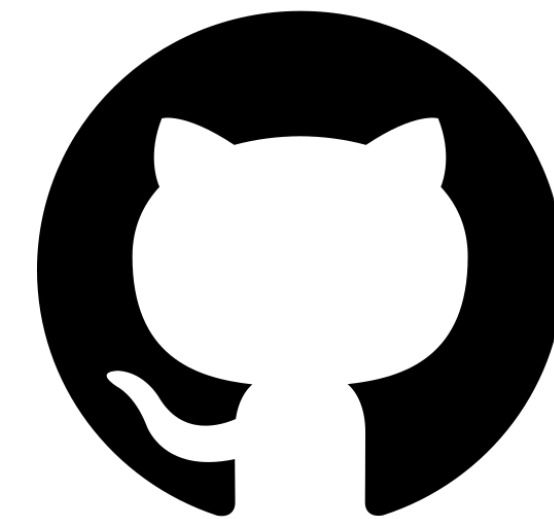
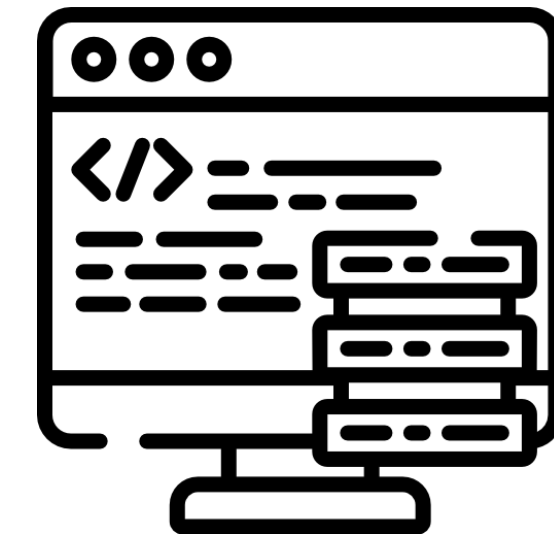
STILE DI SUPERVISIONE (COSA ASPETTARSI)

- I miei progetti cercano di includere le seguenti attività caratteristiche:
 - Investigare lo stato dell'Arte tra le tecniche di Ingegneria del Software
 - Sviluppare nuove idee partendo da soluzioni esistenti
 - Valutare l'implementazione delle proprie idee, confrontandole con le soluzioni esistenti



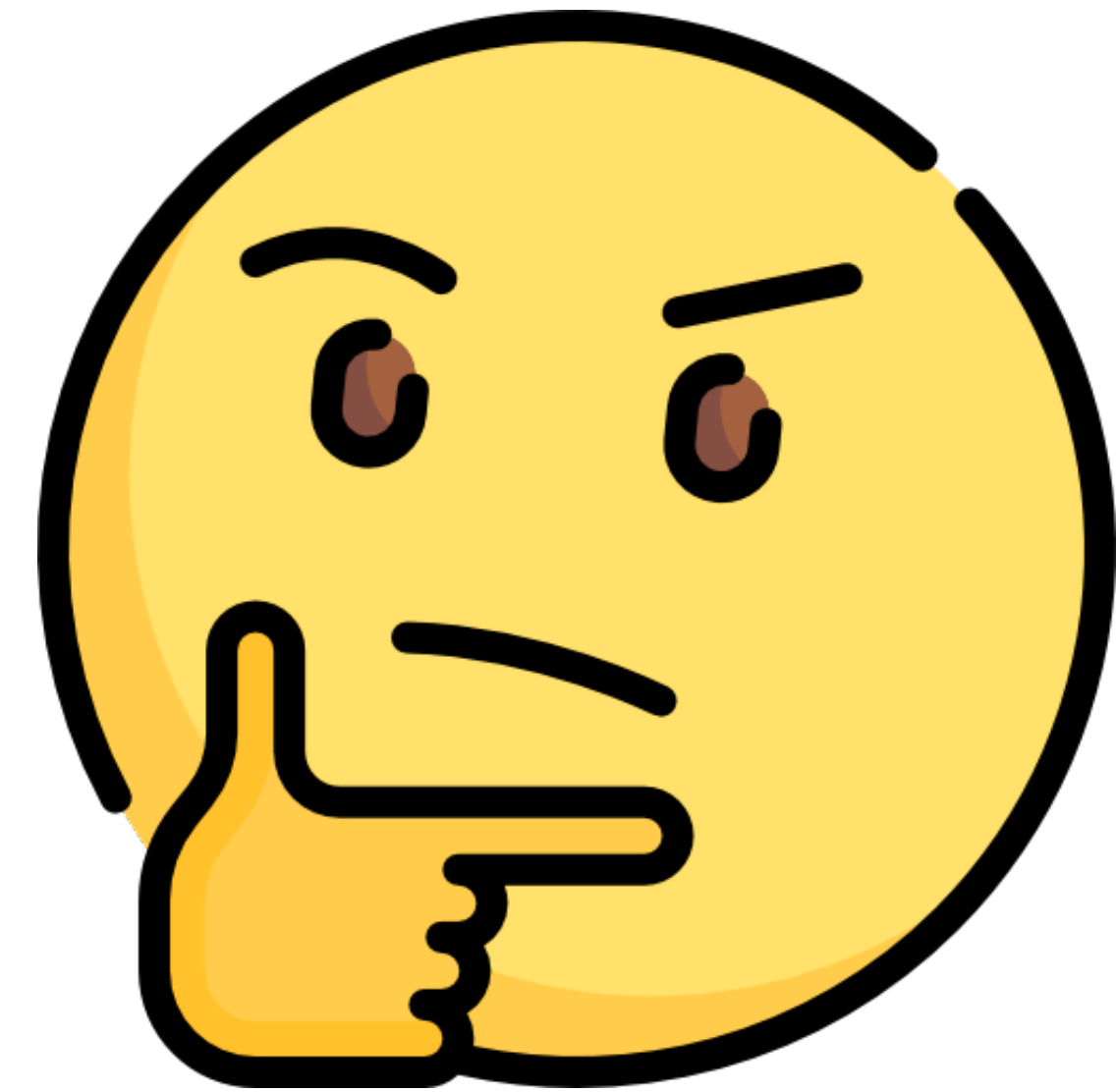
SCRIVERE CODICE?

- La maggior parte dei progetti coinvolge la scrittura di programmi o script (ad es. in Python o Java a seconda del progetto)
- Il codice deve essere open-source e condiviso su repository come GitHub (non in listati contenuti nella tesi)
- Bisogna discutere eventuali NDA con le aziende



INCORAGGIARE LE IDEE DEGLI STUDENTI

- Sono interessato a discutere nuove idee per affrontare sfide interessanti in IS (invece di imporre le mie idee)
- Gli studenti dovranno imparare a sviluppare il pensiero critico:
 - ◎ Rispondendo a domande (invece di chiedere soluzioni immediate)
 - ◎ Proponendo problemi+soluzioni invece di problemi



CARRIERA FUTURA

- I miei progetti riguardano solitamente lo stato dell'Arte nell'ingegneria del software e, in particolare, il **testing**
- Sono interessato a tecniche per valutare la qualità di software

