

- Scrivere immediatamente, su ogni foglio che vi è stato consegnato, cognome, nome, numero di matricola.
- Non è consentito consultare appunti, libri, colleghi, né qualunque dispositivo elettronico, pena l'immediato annullamento della prova.
- **L'Esercizio 1, gli esercizi Modulo A e gli Esercizi Modulo B devono essere svolti su fogli differenti.**
- Tempo a disposizione: 3 ore.

Vi è commissionata la realizzazione di un sistema per gestire i tirocini curriculari (sia interni che esterni) di studenti del Corso di Laurea in Informatica. Il sistema permette agli studenti di inserire una richiesta di tirocinio, indicando la tipologia di tirocinio (interno o esterno), una breve descrizione testuale della tematica trattata, una data prevista di inizio e di fine. Per tirocini interni, inoltre, lo studente deve indicare un docente di riferimento tra i Professori del Corso di Laurea in Informatica. Facoltativamente, lo studente può indicare anche al più due altri co-tutor, da selezionare tra i Professori e/o i Ricercatori del Corso di Laurea in Informatica. Per i tirocini esterni, invece, lo studente deve selezionare un'azienda tra quelle convenzionate con l'Università, e inserire nome, cognome ed e-mail del referente aziendale che seguirà il tirocinio. I membri della commissione tirocini del Corso di Laurea possono visualizzare le domande inviate dagli studenti e decidere se approvarle o rigettarle. In caso di approvazione, il sistema tiene traccia della data in cui la richiesta di tirocinio è stata approvata. Per i soli tirocini esterni, la commissione, al momento dell'approvazione di una richiesta, deve selezionare un docente di riferimento tra i Professori del Corso di Laurea in Informatica. In caso di rigetto, la commissione tirocini deve indicare una breve descrizione testuale con le motivazioni del rigetto. Sia in caso di accettazione che in caso di rifiuto, il sistema invia una e-mail di notifica – sfruttando le API del servizio esterno "EasyMail" – a tutti gli interessati, ovvero allo studente, ai Professori e/o Ricercatori di riferimento, e all'eventuale referente aziendale.

### Esercizio 1

- Si modellino tutti i requisiti del sistema descritto sopra utilizzando un Use Case Diagram;
- Realizzare i mock-up dell'applicazione descritta, relativamente alla funzionalità di accettazione di una richiesta di tirocinio.
- Dettagliare il caso d'uso relativo alla funzionalità di accettazione di una richiesta di tirocinio, per mezzo descrizioni testuali strutturate secondo il formalismo di Cockburn. Usare la propria conoscenza del dominio per derivare dettagli non definiti nei requisiti.
- A partire dai mock-up definiti al punto (b), realizzare uno statechart per modellare il funzionamento dell'interfaccia grafica.

### Modulo A - Esercizio 2A

Il metodo `getQuote` della classe `Utils` viene utilizzato per calcolare il prezzo mensile da offrire ad un cliente per accedere a un SaaS. Il metodo prende in input i seguenti parametri:

- `int maxUsers`: indica il numero massimo di utenti singoli che possono accedere al servizio;
- `String priority`: indica il livello di priorità ("STANDARD" oppure "HIGH") assegnato al cliente;
- `boolean support`: indica se è incluso nel pacchetto l'accesso al supporto telefonico dedicato.

Se i parametri non sono validi, il metodo solleva una `IllegalArgumentException`. In caso contrario, ritorna il prezzo mensile da proporre al cliente. Il prezzo mensile di base è calcolato moltiplicando il numero di utenti per il costo mensile per utente determinato in base al livello di priorità richiesto, come da tabella seguente. Inoltre, se è richiesto anche l'accesso al supporto telefonico, la somma di 25.00 € va aggiunta al totale.

	STANDARD	HIGH
Costo mensile per utente	0.50 €	0.75 €

- Indicare, per ciascuno dei parametri del metodo `getQuote`, le classi di equivalenza individuate.
- Scrivere quattro test JUnit con strategia Black Box per il metodo `getQuote`, indicando per ciascuno di essi quali classi di equivalenza copre. Si richiede inoltre che un test corrisponda a scenari in cui i parametri non sono validi, e che i restanti tre corrispondano a scenari in cui i parametri sono validi.
- Quanti test sono necessari per testare il metodo con strategia WECT? Motivare la risposta.

### **Modulo A – Esercizio 3A**

Il sindaco di una piccola città vi ha assunto come consulenti nell'ambito di un progetto di ammodernamento delle infrastrutture informatiche finanziato con fondi PNRR. La città al momento offre diversi servizi informatici grazie ad un piccolo datacenter (un server DELL PowerEdge 1800 acquistato nel 2004) presente *on-premises* (in soffitta). I servizi offerti includono il sito web istituzionale (con bandi e avvisi), un applicativo per l'accesso a bonus e servizi sociali, e diversi siti web per eventi organizzati sul territorio (sagre, concerti, etc.). Il sito web istituzionale e l'applicativo per la gestione dei servizi sociali necessitano anche di una base di dati relazionale, che al momento è installata sul server *on-premises* di cui sopra.

Il sindaco vi chiede di realizzare una relazione di al più una pagina, da condividere con l'ufficio tecnico, in cui valutate la fattibilità di migrare il datacenter comunale verso servizi public cloud, descrivendo brevemente un piano di migrazione (i.e.: quali tipologie servizi cloud utilizzare e con quale scopo), e fornendo un'analisi dei vantaggi (e degli eventuali svantaggi) che la migrazione comporterebbe, con particolare enfasi su costi, affidabilità, prestazioni, e sicurezza. Il sindaco ci tiene a sottolineare che, per mancanza di fondi, sono assolutamente inaccettabili proposte che necessitano di realizzare nuovi software o reingegnerizzare gli applicativi esistenti, che devono rimanere immutati.

### **Modulo B - Esercizio 3B**

*Definire il concetto di usabilità elencando e descrivendo le tre caratteristiche principali che la compongono.*