

Progetto di

# Ingegneria del Software

A.A. 2025/2026

---

30/9/2025

---

SWEUNINA



UNIVERSITÀ DEGLI STUDI DI NAPOLI  
**FEDERICO II**

   <p><b>ENTE/I: DIETI</b></p> <p>PROTOCOLLO N.: Uni-01-2026</p> <p>DATA EMISSIONE: 25/09/2025</p> <p><b>PAG. 13</b></p>	<p>■ <b>PIANO ATTIVITÀ</b>  <b>PIANO OPERATIVO</b>  <b>REPORT ATTIVITÀ ( INTERMEDIO <input type="checkbox"/> FINALE <input type="checkbox"/> )</b>  <b>ALTRO</b> (sostituire il termine “altro” con il nome del documento:  Analisi Fattibilità, Specifiche di progetto, Requisiti,  Attività, Piani di Formazione,...)</p>
---	---

**OGGETTO:**

**Richiesta Attività Fornitori Esterni**  
**Specifica, progettazione, implementazione e validazione del**  
**progetto di Ingegneria del Software**

**SINTESI DEI CONTENUTI:**

Il documento si riferisce alle attività di progetto da effettuare nell'ambito del corso di Ingegneria del Software, che comprende le seguenti linee di attività:

- A. Analisi e specifica dei requisiti, con definizione di un piano di testing a livello di sistema
- B. Definizione dell'architettura e progettazione del sistema
- C. Implementazione del sistema
- D. Definizione di test automatici di unità, e analisi dell'usabilità sul campo

EMITTENTE: (FIRMA)	DESTINATARI:
<b>ELABORANO:</b> P. Tramontana, B.Breve <b>APPROVANO:</b> P. Tramontana, B.Breve	<b>A:</b> Studenti di Ingegneria del Software 2025/2026, canale 2 <b>P.C.:</b> n.a.

# Indice

Indice .....	3
1 Revisioni .....	4
2 Obiettivi .....	5
2.1 Analisi e specifica dei requisiti .....	5
2.2 Progettazione del sistema .....	5
2.3 Implementazione del sistema .....	5
2.4 Testing e valutazione dell'usabilità.....	5
3 Funzionalità del sistema.....	6
3.1 Funzionalità del sistema <i>BugBoard26</i> .....	6
3.2 Ulteriori vincoli .....	7
4 Modalità di svolgimento del progetto .....	8
5 Formazione dei Gruppi di lavoro .....	9
5.1 Creazione di gruppi con un singolo componente.....	9
6 Assegnazione delle funzionalità da realizzare .....	9
7 Input forniti dal Committente .....	9
8 Output attesi dal Committente .....	10
9 Modalità di consegna e controlli antiplagio .....	11
9.1 Modalità di consegna con svolgimento in itinere .....	11
9.2 Modalità di consegna con svolgimento tradizionale.....	11
9.3 Controlli Antiplagio .....	12
10 Modalità di presentazione e valutazione .....	12
11 Validità del progetto .....	13

# 1 Revisioni

Data	Versione	Autore/i	Descrizione
26/09/2025	1.0	P. Tramontana, B. Breve	Prima stesura.

## 2 Obiettivi

La società SoftEngUniNA ha l'obiettivo di commercializzare diversi sistemi informatici.

Per la realizzazione del suddetto obiettivo, SoftEngUniNA richiede la progettazione, implementazione e parziale verifica dei moduli software necessari al corretto funzionamento dei sistemi informatici. Le linee di attività richieste sono descritte di seguito, mentre un elenco dettagliato degli artefatti da realizzare e consegnare per ciascuna attività è specificato in **Sezione 8**.

### 2.1 Analisi e specifica dei requisiti

Obiettivo di questa linea di attività è la fornitura di un documento che contenga la definizione, attraverso opportuni strumenti di formalizzazione, delle funzionalità del sistema. In particolare, si richiede la fornitura di un documento di specifica dei requisiti, possibilmente conforme ad uno degli standard studiati. In tale documento (oppure come appendici) dovranno essere specificate le funzionalità del sistema attraverso notazioni UML, fornite descrizioni testuali strutturate (es.: template di A. Cockburn), e prototipazione delle interfacce utente tramite mock-up. Si richiede inoltre una definizione degli utenti target. Se necessario, altri modelli di dominio di analisi, possono essere impiegati per migliorare la formalizzazione del problema.

### 2.2 Progettazione del sistema

Obiettivo di questa linea di attività è la fornitura di un documento che descriva in maniera dettagliata la progettazione del sistema per le funzionalità richieste, attraverso modelli e notazioni UML. In particolare, i Gruppi di lavoro contraenti devono motivare e dettagliare le proprie scelte sia per quanto riguarda aspetti architetturali e tecnologici, sia per quanto riguarda la progettazione delle interfacce utente.

### 2.3 Implementazione del sistema

Obiettivo di questa linea di attività è la fornitura di codice sorgente e altri artefatti necessari alla compilazione ed al deployment del sistema che implementa le funzionalità richieste. L'implementazione va realizzata utilizzando un **linguaggio Object-Oriented a scelta del Contraente**, e deve soddisfare gli ulteriori requisiti dettagliati in **Sezione 3**.

### 2.4 Testing e valutazione dell'usabilità

Obiettivo di questa linea di attività è fornitura di un documento e di codice sorgente per la parziale verifica del sistema. In particolare, si richiede, la realizzazione di una batteria di test automatici per alcuni metodi a scelta del Contraente, sviluppata utilizzando framework della famiglia xUnit. Si richiede inoltre una valutazione dell'usabilità, realizzata sia tramite ispezioni o review sistematiche basate su checklist, che tramite il coinvolgimento di utenti reali o potenziali per condurre test di usabilità sul prodotto finito, osservando direttamente le interazioni degli utenti con il sistema e proponendo un survey di valutazione.

## 3 Funzionalità del sistema

*BugBoard26* è una piattaforma per la gestione collaborativa di issue in progetti software. Il sistema consente a team di sviluppo di segnalare problemi relativi a un progetto, monitorarne lo stato, assegnarli a membri del team e tenere traccia delle attività di risoluzione. Il sistema deve consistere in un'applicazione (mobile, desktop o web-based) performante e affidabile, attraverso cui gli utenti possono fruire delle funzionalità in modo intuitivo e rapido.

### 3.1 Funzionalità del sistema *BugBoard26*

1. Deve essere implementato un sistema di autenticazione semplice e sicuro, basato su email e password. Le informazioni gestite dall'applicazione sono critiche per l'azienda, ed è fondamentale preservarne l'integrità e la segretezza. Il sistema viene fornito con un account da amministratore già attivo, con credenziali di default. Un amministratore può creare ulteriori utenze, specificando una email, una password, e indicando se quell'utenza sarà "normale" oppure "di amministrazione".
2. Tutti gli utenti autenticati possono segnalare una issue indicando almeno un titolo e una descrizione. Alcuni utenti potrebbero voler specificare anche una priorità e sarebbe gradita la possibilità di allegare un'immagine. Le issue possono essere di diverso tipo: question (per richieste di chiarimenti), bug (per segnalare malfunzionamenti), documentation (per segnalare problemi relativi alla documentazione), e feature (per indicare la richiesta o il suggerimento di nuove funzionalità). Le issue create sono inizialmente nello stato "todo".
3. Il sistema deve offrire una vista riepilogativa delle issue, con la possibilità di filtrare o ordinare i risultati in base a criteri come tipologia, stato, priorità o altri parametri rilevanti.
4. Ogni bug dovrebbe poter essere assegnato (da un amministratore) a un membro del team. Quando viene assegnato un bug a un utente, quest'ultimo riceve una notifica. Gli utenti possono visualizzare i bug loro assegnati.
5. Deve essere prevista una sezione per commenti associata a ciascun bug, in cui tutti gli utenti possano lasciare messaggi, aggiornamenti o richieste di chiarimento.
6. L'utente cui un bug viene assegnato può successivamente modificarne lo stato. Quando un bug viene risolto, il sistema dovrebbe notificare l'utente che lo ha segnalato.
7. È richiesta una dashboard per gli amministratori che mostri informazioni aggregate sui bug, come il numero di bug aperti, quelli assegnati per utente, il tempo medio di risoluzione (aggregato e per utente) e altri dati utili.
8. Deve essere possibile esportare l'elenco dei bug in un formato condivisibile, come CSV, Excel o PDF.

9. Gli utenti devono poter modificare solo i bug a loro assegnati, mentre gli amministratori devono avere accesso completo.
10. Il sistema deve permettere di associare un numero variabile di etichette personalizzabili ai bug (es. “frontend”, “urgente”, “sicurezza”).
11. È necessaria una funzione di ricerca che consenta di trovare bug in base a parole chiave.
12. Ogni bug dovrebbe avere una cronologia delle modifiche (inclusa la creazione), che mostri chi ha effettuato cambiamenti e quando.
13. Deve essere possibile archiviare un bug, in modo che non appaia più nelle liste principali ma resti comunque accessibile da una pagina dedicata. Soltanto gli amministratori possono archiviare un bug.
14. Il sistema dovrebbe suggerire automaticamente a chi assegnare un bug, basandosi sul carico di lavoro corrente degli utenti.
15. Il sistema deve offrire una modalità “readonly” per utenti esterni (es. stakeholder), che consenta di visualizzare i bug e i commenti senza possibilità di modifica. Gli account per questi utenti vengono creati dagli amministratori.
16. Un amministratore può contrassegnare un bug come duplicato di un altro. Questo comporta la chiusura immediata del bug duplicato.
17. Gli amministratori devono poter visualizzare report mensili sull’attività del team, contenenti metriche (numero di bug aperti/gestiti, tempo di risoluzione medio, etc) aggregate e dettagliate per singolo utente.
18. Gli amministratori possono impostare scadenze opzionali per la risoluzione di una certa issue.

Dato questo insieme di funzionalità, il Committente assegnerà un sottoinsieme **non negoziabile** di caratteristiche da modellare, implementare e verificare, tenendo in dovuta considerazione il numero di partecipanti al Gruppo di Lavoro contraente. I contraenti sono tenuti a utilizzare la propria **conoscenza del dominio** per interpretare e completare i requisiti non completamente specificati nella traccia, colmando eventuali lacune **in modo coerente**.

## 3.2 Ulteriori vincoli

Tutte le funzionalità assegnate devono essere realizzate implementando **un sistema distribuito** che preveda due macro-componenti indipendenti:

- **Back-end:** Almeno un componente che gestisce la logica di business e i dati, esponendo interfacce di programmazione accessibili via rete (ad esempio, tramite API REST). Il back-end deve essere completamente indipendente dalla parte front-end, ovvero, deve poter funzionare ed essere eseguito in maniera autonoma, senza alcuna conoscenza o dipendenza diretta da come il front-end verrà implementato.



- **Front-end:** Un'interfaccia utente che si appoggia ai servizi offerti dal back-end esclusivamente attraverso la rete. Il front-end deve comunicare con il back-end solo tramite le API esposte, e deve poter essere sostituito o modificato senza la necessità di modifica alcuna sul back-end. Analogamente, il front-end deve poter essere eseguito separatamente dal back-end.

Il back-end è responsabile della gestione centralizzata dello stato del sistema, garantendo che tutte le informazioni e i dati persistenti siano conservati e aggiornati unicamente su di esso. È altresì auspicabile (ma non obbligatorio) che il back-end sia distribuito utilizzando tecnologie di containerizzazione come Docker e che, al momento della discussione del progetto con demo del prodotto, sia messo in opera utilizzando tecnologie allo stato dell'arte quali ad esempio servizi di public Cloud Computing come Azure o AWS, e sia quindi accessibile attraverso la rete Internet.

Per lo svolgimento delle attività di progettazione, è obbligatorio l'utilizzo di tool di CASE (ad es. Visual Paradigm). Inoltre, si richiede tassativamente di astrarre il design per favorire il riutilizzo del codice e la futura implementazione di altre funzionalità, **esplicitando nella documentazione** (si veda **Sezione 5**) **le scelte intraprese per favorire tale astrazione**.

Per quanto riguarda le tecnologie da utilizzare, è data **piena libertà** di scelta al Gruppo di Lavoro contraente, **con l'unico vincolo dell'utilizzo obbligatorio di linguaggi di programmazione Object-Oriented**. Per esempio, la parte front-end può essere realizzata come applicazione desktop (e.g.: con Java + Swing/JavaFx), come applicazione web (utilizzando framework per Single Page Applications, come Angular o React), oppure come applicazione mobile (per esempio per dispositivi Android o iOS). Il Gruppo di Lavoro dovrà motivare le proprie scelte tecnologiche in fase di discussione del prodotto (vedi **Sezione 9**).

Si sottolinea che le fasi di analisi e progettazione object-oriented del sistema sono fondamentali per una valutazione positiva del Progetto. L'utilizzo di servizi **MBaaS** (come Firebase, Supabase, e simili) non sostituisce in alcun modo queste fasi. Pertanto, **non saranno accettati progetti in cui la logica applicativa e la persistenza dei dati sono gestite esclusivamente tramite servizi esterni**, poiché ciò renderebbe il sistema troppo dipendente da un singolo provider esterno.

## 4 Modalità di svolgimento del progetto

Per l'a.a. 2025/2026, il progetto di Ingegneria del Software potrà essere svolto secondo due diverse modalità:

- **Modalità in itinere:** pensata per gli studenti che seguono in corso e vogliono sostenere l'esame completo nella sessione invernale (gennaio-marzo 2026). Questa modalità prevede un ridotto insieme di funzionalità da realizzare e una ridotta quantità di documentazione da produrre, pensata per essere compatibile con gli impegni degli studenti durante il semestre. Saranno presenti scadenze intermedie, che saranno comunicate con congruo anticipo durante il corso e nel Team dell'Insegnamento.



- **Modalità tradizionale:** insieme standard di funzionalità e documentazione, con consegna effettuabile liberamente entro marzo 2027, senza scadenze intermedie.

## 5 Formazione dei Gruppi di lavoro

**I gruppi di lavoro sono generalmente formati da 2 o da 3 componenti.** Per richiedere la formazione di un gruppo di lavoro, è necessario che **uno (e soltanto uno)** dei componenti del gruppo compili un modulo online, indicando i propri dati (nome, cognome, matricola, **e-mail istituzionale**) e quelli degli altri membri del gruppo.

Il link per accedere al modulo è il seguente: <https://forms.office.com/e/1i4Yiew3Gi>

Un recap della composizione dei gruppi verrà pubblicato nel Team dell'insegnamento per confronto e per consultare la propria assegnazione.

### 5.1 Creazione di gruppi con un singolo componente

La creazione di **gruppi con un singolo componente** è consentita ma non è consigliata, a causa della conseguente impossibilità di fare pratica con i problemi di lavoro condiviso e collaborativo.

## 6 Assegnazione delle funzionalità da realizzare

Le funzionalità assegnate a ciascun gruppo sono determinate in base alla numerosità dei gruppi (più funzionalità per i gruppi da 3 persone) e al periodo nel quale il progetto viene svolto (meno funzionalità per chi conclude all'interno della sessione invernale, entro marzo 2026).

L'assegnazione delle funzionalità avverrà nella prima settimana di ottobre a seguito della formazione dei gruppi.

## 7 Input forniti dal Committente

Il Committente fornisce i seguenti input.

- Descrizione delle funzionalità del sistema (**Sezione 3** di questo documento).
- Elenco di riferimenti a modelli e notazioni formali, sul sito del corso e sul gruppo MS Teams dell'insegnamento di Ingegneria del Software.

- Appunti, slide e materiale vario, sul sito del corso e/o sul gruppo MS Teams dell'insegnamento di Ingegneria del Software.

## 8 Output attesi dal Committente

Per le attività sopracitate, il Committente richiede i seguenti output:

### 1. Documento di Specifica dei Requisiti Software

- a. Glossario.
- b. Modellazione di **tutti** i casi d'uso richiesti tramite Use Case Diagrams.
- c. Individuazione e caratterizzazione del target degli utenti.
- d. Descrizione dei requisiti non-funzionali e di sistema.
- e. Formalizzazione di almeno due (**uno, se si svolge il progetto con la modalità in itinere**) casi d'uso significativi (**esclusi quelli relativi a registrazione/autenticazione utenti**) a scelta dei contraenti tra quelli richiesti. Per ciascuno di questi casi d'uso, è richiesta:
  - i. Descrizione testuale strutturata, realizzata utilizzando un formalismo tabellare, come quello di A. Cockburn.
  - ii. Prototipazione visuale via Mock-up delle relative interfacce utente.

### 2. Documento di Design del sistema.

- a. Descrizione dell'architettura proposta, con esplicita definizione dei criteri di design adottati e delle motivazioni dietro tale scelta.
- b. Descrizione e motivazione delle scelte tecnologiche adottate.

### 3. Documento di Design del Software, documentazione del processo di sviluppo, e artefatti software.

- a. Descrizione dello schema per la persistenza dati.
- b. Diagramma delle classi di design.
- c. Descrizione e motivazione delle scelte di software design adottate.
- d. Evidenza dell'uso di strumenti di versioning (link al repository, ad es. Github, con eventuali report con statistiche come numero di commit e di contributors, frequenza dei commit, etc.).
- e. Report di qualità del codice, generati da SonarQube o simili (soltanto per il back-end).
- f. Codice Sorgente sviluppato, comprensivo di eventuale Dockerfile e file di build automatica (anche accessibili da repository Github).

### 4. Attività di Testing

- a. Test plan per una funzionalità a scelta dei contraenti.
- b. Codice per test di unità automatici per tre (**due, se si svolge il progetto con la modalità in itinere**) metodi non banali, **che abbiano almeno due parametri**.
- c. Documentazione che descrive strategie adottate per la progettazione dei test di cui al punto precedente (e.g.: classi di equivalenza individuate e coperte, criteri di copertura strutturale adottati, etc.).

Si noti che i Punti 1, 2 e 4 vanno realizzati come capitoli distinti in un unico documento, di seguito indicato come “**documentazione**”.

## 9 Modalità di consegna e controlli antiplagio

### 9.1 Modalità di consegna con svolgimento in itinere

Per i gruppi che svolgono il progetto con modalità in itinere, sono previste **tre scadenze intermedie**:

- Una prima scadenza per la consegna del documento di specifica dei requisiti (Punto 1 in **Sezione 8 - Output attesi dal Committente**).
- Una seconda scadenza per la consegna del documento di design del sistema (Punti 2 e 3 in **Sezione 8 - Output attesi dal Committente**).
- Una ultima scadenza per la consegna degli artefatti software, dei test di unità, e documentazione collegata (Punto 4 in **Sezione 8 - Output attesi dal Committente**).

Le date esatte delle scadenze di cui sopra e le modalità di consegna saranno comunicate con congruo anticipo nel Team dell'insegnamento. **Qualora dovessero emergere esigenze di modifica o integrazione rispetto ai documenti già consegnati nelle scadenze precedenti, sarà possibile includere tali aggiornamenti nelle consegne successive senza penalizzazioni nella valutazione, purché siano chiaramente indicati e motivati.**

### 9.2 Modalità di consegna con svolgimento tradizionale

Tutti gli output previsti vanno consegnati entro il **30 settembre 2026**.

**Le consegne avvengono tutte tramite attività Teams nel canale dell'insegnamento.**

Ogni consegna dovrebbe consistere:

- in un link al repository (preferibilmente Github) contenente tutto il materiale realizzato (codice, eseguibili, report, documentazioni);
- Un link diretto ad una release creata sullo stesso repository al momento della consegna;
- Eventuali link ad altro materiale che per qualche motivo non può essere caricato nell'ambito del repository. Per l'invio di materiale particolarmente pesante, l'utilizzo del servizio Unina Gigamail (<https://gigamail.unina.it/filesender/>) è richiesto.

Per questioni di tracciabilità, ogni membro del gruppo deve effettuare la sua consegna (ovviamente membri dello stesso gruppo possono effettuare consegne identiche).

Al momento della consegna del progetto in forma definitiva, è opportuno inviare mail o messaggio teams ai docenti per notificare l'invio e iniziare a prenotare la successiva presentazione.

### 9.3 Controlli Antiplagio

Si rende noto che tutti gli artefatti consegnati vengono processati con software antiplagio Turnitin ©, e raffrontati automaticamente con progetti degli anni scorsi e di quest'anno. In presenza di presunto plagio, ad insindacabile giudizio dei docenti, il lavoro viene annullato nella sua interezza a tutti i gruppi coinvolti, che riceveranno un nuovo progetto.

## 10 Modalità di presentazione e valutazione

Per ciascuno degli appelli di esame previsti sarà organizzata una seduta di presentazione e discussione dei progetti. Le date in cui si svolgeranno tali sedute e le modalità di prenotazione saranno pubblicizzate con congruo anticipo tramite i canali ufficiali (piattaforma [www.docenti.unina.it](http://www.docenti.unina.it) e canale Teams dell'Insegnamento). Potranno sostenere la discussione del progetto tutti i gruppi che abbiano consegnato il progetto entro il termine che sarà reso noto per ciascuna seduta, e che si siano prenotati.

La discussione consisterà in una **presentazione tecnica** della durata di al più **10 minuti**, seguita da una breve dimostrazione pratica del funzionamento del prodotto, con domande dei docenti (anche sul codice sorgente). **La presentazione deve essere realizzata utilizzando MS PowerPoint o strumenti analoghi.**

Si sottolinea che la presentazione tecnica costituisce il cuore della valutazione per l'esame di Ingegneria del Software, i cui destinatari sono i docenti di Ingegneria del Software (**non più da considerare i contraenti**). Pertanto, la presentazione tecnica **non deve riepilogare le funzionalità del prodotto, né essere finalizzata al puro marketing**. Al contrario, la presentazione deve mettere in evidenza le scelte effettuate in fase di progettazione e implementazione, e convincere i docenti della correttezza delle scelte e delle soluzioni individuate, e della propria padronanza dell'Ingegneria del Software. **Per qualsiasi dubbio in merito, gli studenti sono invitati a partecipare al ricevimento studenti dei docenti.**

La valutazione dell'elaborato progettuale sarà determinata dalla qualità del prodotto software sviluppato e dalla discussione dello stesso, tenendo in considerazione la capacità di esposizione con terminologia appropriata, la capacità di organizzazione autonoma dell'esposizione e la capacità di motivare ed argomentare opportunamente le scelte di progetto.

Si sottolinea infine che, in sede di discussione del progetto, tutti i membri del gruppo di lavoro contraente devono conoscere **ogni aspetto** del sistema presentato. La mancata conoscenza di parti del prodotto da parte di uno dei membri, anche se dovuta alla suddivisione interna dei compiti, risulterà in una penalizzazione per tutti i membri del gruppo.

## 11 Validità del progetto

**È possibile consegnare il progetto 2025/2026 con modalità in itinere soltanto rispettando tutte le scadenze previste, che verranno esplicitate in ognuna delle attività Teams che verranno proposte.** Se non si rispettano le scadenze, sarà necessario svolgere il progetto con modalità tradizionale.

**È possibile consegnare il progetto 2025/2026 svolto in modalità tradizionale fino al 30 settembre 2026.** Dopo tale data, sarà necessario svolgere il progetto dell'anno accademico 2026/2027.

**Il progetto ha inoltre validità di un anno dalla data della consegna.** Ciò vuol dire che gli studenti hanno, dopo aver consegnato il progetto, un anno di tempo per discuterlo, superare la prova scritta (se non già superata in precedenza) e verbalizzare l'esame.