

Progetto di

Ingegneria del Software

A.A. 2024/2025

22/10/2024

SOFTENGUNINA



UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

 <p>UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II</p>  	<p>■ PIANO ATTIVITÀ PIANO OPERATIVO REPORT ATTIVITÀ (INTERMEDIO <input type="checkbox"/> FINALE <input type="checkbox"/>) ALTRO (sostituire il termine “altro” con il nome del documento: Analisi Fattibilità, Specifiche di progetto, Requisiti, Attività, Piani di Formazione,...)</p>
<hr/> <p>ENTE/I: DIETI PROTOCOLLO N.: Uni-01-2025 DATA EMISSIONE: 22/10/2024</p> <p>PAG. 16</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:
ELABORANO: S. Di Martino, L. L. L. Starace APPROVANO: S. Di Martino, L. L. L. Starace	A: Studenti di Ingegneria del Software 2024/2025 P.C.: 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 Requisiti.....	6
3.1 Requisiti del sistema <i>DietiEstates25</i>	6
3.2 Ulteriori requisiti.....	8
4 Input forniti dal Committente	9
5 Output attesi dal Committente	10
6 Formazione dei Gruppi di lavoro	11
6.1 Creazione di gruppi con un singolo componente.....	11
7 Assegnazione delle funzionalità da realizzare	12
7.1 Gruppi con un singolo componente	12
7.2 Gruppi con due componenti	12
7.3 Gruppi con tre componenti.....	12
8 Modalità di consegna e controlli antiplagio	13
8.1 Controlli Antiplagio	14
9 Modalità di presentazione e valutazione	15
10 Validità del progetto	16

1 Revisioni

Data	Versione	Autore/i	Descrizione
21/10/2024	1.0	L. L. L. Starace, S. Di Martino	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 5**.

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 che specifichi le funzionalità del sistema attraverso notazioni UML, descrizioni testuali strutturate (es.: template di A. Cockburn), e prototipazione delle interfacce utente. 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 Requisiti

DietiEstates25 è una piattaforma per la gestione di servizi immobiliari. Il sistema permette a più agenzie di pubblicare inserzioni immobiliari. Gli utenti possono quindi visualizzare le inserzioni, prenotare visite e fare offerte per acquistare/affittare un immobile. 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, rapido e piacevole.

3.1 Requisiti del sistema *DietiEstates25*

Le principali funzionalità offerte da *DietiEstates25* sono indicate di seguito:

1. Ogni installazione di *DietiEstates25* include un'utenza di amministrazione per il gestore dell'agenzia immobiliare, che viene creata con credenziali predefinite. Dopo aver effettuato l'accesso, è possibile modificare la password di amministrazione. L'amministratore può creare altri account di supporto all'amministrazione. I gestori di agenzie immobiliari possono a loro volta creare account per gli agenti immobiliari. Un utente interessato ai servizi immobiliari può registrarsi con un'e-mail e una password. È apprezzata la possibilità di effettuare la registrazione/login tramite credenziali di terze parti (e.g.: accedi con account Google, Facebook o GitHub). Gli utenti possono effettuare il login e accedere all'applicativo. Le credenziali devono essere salvate in modo sicuro.
2. Gli agenti immobiliari possono caricare nuovi immobili con una serie di dettagli: foto, descrizione, prezzo, dimensioni, indirizzo, numero di stanze, piano, presenza di ascensore, classe energetica, ulteriori servizi presenti (portineria, climatizzazione, ...), ecc. Gli immobili devono essere categorizzati in "vendita", "affitto". Forse in futuro sarà introdotto il supporto per affitti brevi e "case vacanze" (ma questa funzionalità al momento non è prevista). Ogni immobile inoltre è associato alla posizione geografica esatta, individuata interagendo con una mappa interattiva (tipo Google Maps).
3. Gli utenti possono eseguire una ricerca avanzata di immobili con parametri multipli: tipologia di inserzione, prezzo minimo e massimo, numero di stanze, classe energetica, posizione, ecc. In particolare, deve essere possibile effettuare filtri per posizione geografica almeno al livello di comune o città. È apprezzata – ma non obbligatoria – la possibilità di effettuare ricerche di immobili che si trovano in un raggio arbitrario da un certo punto specificato dall'utente e selezionato su una mappa interattiva. La ricerca deve essere efficiente e performante. Gli immobili individuati nelle ricerche vengono visualizzati su una mappa interattiva (tipo Google Maps).
4. Gli utenti devono poter modificare i loro dati personali: nome, cognome, e-mail, password. Gli agenti immobiliari devono avere un profilo più dettagliato, con informazioni professionali (e.g.: una breve biografia) e magari una valutazione lasciata dai clienti. È gradita la possibilità di aggiungere anche delle foto per gli agenti immobiliari, per rendere i profili più "umani".

5. Gli immobili possono essere modificati e cancellati, ma solo dall'agente che li ha inseriti o dall'amministratore.
6. I clienti devono poter prenotare una visita (nelle successive due settimane) per un immobile direttamente dal sistema. Gli agenti immobiliari ricevono notifiche (e-mail o interne al sistema, o possibilmente entrambe) quando una visita è prenotata per uno dei loro immobili. Le visite possono essere confermate o rifiutate dagli agenti. Se un appuntamento viene rifiutato, il cliente deve poter selezionare un altro orario. Sarebbe apprezzata la possibilità di integrare un calendario per gestire visivamente gli appuntamenti di un agente immobiliare.
7. Quando prenotano una visita, gli utenti visualizzano anche le previsioni meteo per i diversi giorni in cui è possibile prenotare, in modo da poter pianificare meglio la visita secondo le proprie preferenze e necessità. Le previsioni meteo vanno ottenute tramite API di servizi esterni come [Open Meteo](#).
8. Gli utenti possono fare un'offerta su un immobile, specificando un prezzo eventualmente inferiore a quello riportato nell'inserzione. Gli agenti immobiliari possono accettare o rifiutare l'offerta, o fare una controproposta. Deve esserci un tracking delle offerte fatte e ricevute, magari uno storico visibile sia agli agenti sia ai clienti. Gli agenti hanno la possibilità di inserire manualmente offerte ricevute al di fuori del sistema.
9. Quando viene creata un'inserzione, il sistema verifica automaticamente, tramite API di servizi esterni come [Geoapify](#), la presenza di scuole, parchi pubblici, o fermate del trasporto pubblico nei pressi della posizione dell'immobile. Nel caso di riscontro affermativo, un apposito indicatore viene associato all'inserzione ("Vicino a scuole", "Vicino a parchi", "Vicina a trasporto pubblico").
10. Il sistema deve tenere traccia delle ricerche effettuate da un utente in precedenza, in modo tale che un utente possa facilmente ri-eseguire interrogazioni già eseguite in passato per vedere se sono presenti nuove inserzioni che soddisfano le caratteristiche richieste.
11. Il sistema deve inviare notifiche agli utenti per vari eventi: nuove proprietà in linea con le loro ricerche precedenti, conferma o rifiuto delle visite, messaggi promozionali. Gli utenti devono poter attivare/disattivare singole categorie di notifiche.
12. Gli agenti immobiliari devono avere accesso a una dashboard con statistiche sugli immobili: numero di visualizzazioni dell'immobile, numero di visite prenotate, immobili venduti/affittati nel tempo, offerte ricevute, ecc. Deve essere possibile esportare report in formato PDF o Excel/CSV per la gestione interna delle agenzie.
13. Gli utenti normali possono visualizzare un riepilogo delle attività svolte sul sito (immobili visualizzati, visite prenotate, offerte fatte).

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.

3.2 Ulteriori requisiti

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. 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 Input forniti dal Committente

Il Committente fornisce i seguenti input.

- Descrizione dei requisiti 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.

5 Output attesi dal Committente

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

1. Documento 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, tramite Personas.
- d. Descrizione dei requisiti non-funzionali e di dominio.
- e. Formalizzazione di almeno quattro **(due, se si consegna prima del 24/03/2025)** 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, attraverso strumenti di *rapid prototyping*.

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.
- c. Descrizione dello schema per la persistenza dati.
- d. Descrizione e motivazione delle scelte di design dell'interfaccia utente adottate.
- e. Diagramma delle classi di design.
- f. Diagrammi di sequenza di design per i casi d'uso identificati al punto **1.e**.

3. Artefatti Software e documentazione del processo di sviluppo.

- a. Codice Sorgente sviluppato, comprensivo di eventuale Dockerfile.
- b. File di build automatica
- c. Evidenza dell'uso di strumenti di versioning (e.g.: descrizione del sistema di versioning utilizzato, report con statistiche come numero di commit e di contributors, frequenza dei commit, etc..).
- d. Report di qualità del codice, generati da SonarQube o simili (soltanto per il back-end).

4. Testing e valutazione dell'usabilità.

- a. Codice xUnit per *unit testing* di quattro **(due, se si consegna prima del 24/03/2025)** metodi non banali, **che abbiano almeno due parametri**. In aggiunta al codice di test, una apposita sezione della documentazione deve descrivere le strategie adottate per la progettazione dei test dei (e.g.: classi di equivalenza individuate e coperte, criteri di copertura strutturale adottati, etc.).
- b. Descrizione delle tecniche di valutazione dell'usabilità applicate, che devono includere almeno:
 - i. *Expert reviews / inspections*: definizione di una checklist per review sistematica dell'usabilità del prodotto finito, e applicazione della stessa al prodotto.
 - ii. Progettazione e conduzione di un esperimento con utenti reali o potenziali per la valutazione dell'usabilità del sistema realizzato.
 1. Descrizione dei soggetti reclutati, della procedura sperimentale e delle metriche considerate.
 2. Definizione di un survey da sottoporre agli utenti dopo l'esperimento.
 3. Presentazione e discussione dei risultati dell'esperimento e del survey.

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

6 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 form è il seguente: <https://forms.office.com/e/S8LZAShLWt>.

In seguito alla compilazione del form, i membri del gruppo riceveranno una e-mail generata automaticamente riportante il codice identificativo del gruppo creato.

Attenzione: A causa della configurazione delle e-mail istituzionali, l'e-mail di conferma della formazione del gruppo potrebbe essere recapitata nella casella di posta istituzionale (accessibile via web da <https://posta.unina.it/>) oppure nella casella di posta Exchange (accessibile via web da <https://outlook.com>, dopo aver effettuato l'accesso con le credenziali istituzionali). **Le due caselle di posta di cui sopra non sono sempre allineate.** Gli studenti sono pertanto pregati di controllare **entrambe** le caselle di posta.

In caso non si riceva risposta entro 24 ore dall'invio del modulo **in nessuna delle due caselle di posta di cui sopra**, uno dei membri del gruppo è invitato a contattare via e-mail il docente del Mod. B all'indirizzo luigiliberolucio.starace@unina.it, inserendo in Copia Conoscenza tutti gli altri membri del gruppo.

6.1 Creazione di gruppi con un singolo componente

La creazione di **gruppi con un singolo componente** è consentita soltanto in presenza di giustificati e documentati motivi. Per richiedere la creazione di un gruppo con un singolo partecipante, è necessario fissare un incontro con uno dei docenti in orario di ricevimento, secondo le modalità indicate sul sito web istituzionale www.docenti.unina.it.

7 Assegnazione delle funzionalità da realizzare

Le funzionalità assegnate a ciascun gruppo sono determinate in base alla numerosità dei gruppi e alle matricole dei componenti, come descritto di seguito.

Sia x il numero ottenuto sommando l'ultima cifra della matricola di tutti i membri del gruppo. Per esempio, se le matricole dei tre membri di un gruppo sono N86001234, N86002345, e N86003456, allora $x = 4 + 5 + 6 = 15$.

7.1 Gruppi con un singolo componente

Se x è pari, le funzionalità da realizzare sono: 1, 2, 3, 4.

Se x è dispari, le funzionalità da realizzare sono: 1, 2, 3, 5.

7.2 Gruppi con due componenti

Se $x \bmod 4 = 0$, le funzionalità da realizzare sono: 1, 2, 3, 6, 7.

Se $x \bmod 4 = 1$, le funzionalità da realizzare sono: 1, 2, 3, 8, 9.

Se $x \bmod 4 = 2$, le funzionalità da realizzare sono: 1, 2, 3, 10, 11.

Se $x \bmod 4 = 3$, le funzionalità da realizzare sono: 1, 2, 3, 9, 12.

7.3 Gruppi con tre componenti

Se $x \bmod 4 = 0$, le funzionalità da realizzare sono: 1, 2, 3, 6, 7, 8, 9, 13.

Se $x \bmod 4 = 1$, le funzionalità da realizzare sono: 1, 2, 3, 8, 9, 10, 11, 5.

Se $x \bmod 4 = 2$, le funzionalità da realizzare sono: 1, 2, 3, 10, 11, 6, 7, 4.

Se $x \bmod 4 = 3$, le funzionalità da realizzare sono: 1, 2, 3, 9, 12, 8, 6, 13.

8 Modalità di consegna e controlli antiplagio

Tutti gli output previsti vanno consegnati entro il **31 marzo 2026** secondo le modalità descritte di seguito. La consegna caricando gli artefatti tramite il servizio [GigaMail Unina](#), e inviando una e-mail di consegna ai docenti.

Gli artefatti consegnati devono rispettare i seguenti vincoli:

- La documentazione per il progetto deve essere in formato PDF, ed avere **tassativamente** come nome file “IdGruppo.pdf” (e.g.: “INGSW2425_001.pdf”);
- Il codice sorgente sviluppato deve essere caricato come un unico archivio in formato **ZIP** (non RAR, TAR, o altro). Il file deve avere **tassativamente** come nome “IdGruppo.zip” (e.g.: “INGSW2425_001.zip”);
- L’archivio zip contenente il codice sorgente non deve contenere altri archivi, ma soltanto directory e file sorgente. Inoltre, l’archivio **NON** deve contenere file compilati e/o dipendenze sviluppate da terze parti.

Per caricare gli artefatti e ottenere un link pubblico, uno dei membri del gruppo deve effettuare l’accesso al servizio GigaMail Unina tramite le proprie credenziali istituzionali. Dopodichè, si procede al caricamento della documentazione e dell’archivio contenente i file sorgente. In questa fase, è necessario selezionare l’opzione “*Get a link instead of sending to recipients*”, come mostrato in **Figura 1**. Si raccomanda, inoltre, di selezionare una data di scadenza più lontana possibile nel futuro (un mese dalla data di caricamento). Dopo aver effettuato il caricamento, il sistema GigaMail fornisce un link pubblico per il download degli artefatti. Tale link deve essere trasmesso ai docenti all’interno del corpo della e-mail di consegna.

Nella **Tabella 1** sono dettagliati i destinatari a cui deve essere inviata la e-mail di consegna e l’oggetto che quest’ultima deve **tassativamente** avere. Si noti che è necessario sostituire a “<ID>” l’identificativo del proprio gruppo.

Tabella 1 - Dettagli email consegna prodotto

Oggetto	Destinatari	Copia Conoscenza
[INGSW2425] Consegna <ID>	sergio.dimartino@unina.it luigiliberolucio.starace@unina.it	Indirizzi e-mail di tutti i membri del gruppo.

I committenti, a valle della ricezione della e-mail di consegna, verificano che tutti gli artefatti necessari siano stati caricati rispettando i requisiti dati, e tengono traccia della data di consegna.

Le mail di consegna non conformi a quanto specificato in questa sezione saranno ignorate e fonte di penalizzazione in fase di valutazione. Le mail ben formate riceveranno, entro sette giorni lavorativi, conferma dell’avvenuta consegna.

The screenshot shows the GigaMail.unina interface. At the top, there are logos for GigaMail.unina, the University of Naples Federico II, and CSI. Below the logos is a navigation bar with links: Upload, Guests, My Transfers, My profile, Help, About, Privacy, and Log-off.

The main area displays two uploaded files:

- INGSW2425_001.pdf : 11.7 kB
- INGSW2425_001.zip : 280.2 kB

 Below the files is a large dashed box with the text "drag & drop your files here". At the bottom of this section are buttons for "Clear all" and "Select files". To the right, it shows "Number of files : 2/30" and "Size : 291.9 kB/200 GB".

Below the upload section, there is a form for sending the files. On the left, it says "From : mario.rossi@unina.it" and has a checkbox for "File Encryption". A red arrow points to this checkbox with the text: "Spuntare l'opzione per ottenere un link invece di inviare una e-mail".

On the right, there is a section for "Expiry date" set to "29/10/2024". A red arrow points to this date with the text: "Selezionare la data più lontana possibile nel futuro (un mese dalla data di caricamento)". Below the expiry date are three checkboxes:

- ☐ Notify me when expired
- ☐ Send me daily statistics
- ☒ Get a link instead of sending to recipients

 Below these is a section for "Advanced settings" with two checkboxes:

- ☐ Notify me when upload is done
- ☐ Notify me upon downloads

 There is also a "TeraSender worker count" set to "10" and a checkbox for "Disable parallel upload (Tick if you are on a slow connection)".

At the bottom of the form is a large "Send" button with an upload icon. Below the button is a line graph showing "Encryption in transit & rest" (green) and "Encryption in transit" (orange) over time. The x-axis shows dates from 2024-09-19 to 2024-10-19. The y-axis shows values from 0 to 30.

Figura 1 Caricamento degli artefatti tramite GigaMail Unina

8.1 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.

9 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 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.

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.

10 Validità del progetto

È possibile consegnare il progetto 2024/2025 fino al 31 marzo 2026. Dopo tale data, sarà necessario svolgere il progetto dell'anno accademico 2025/2026. **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.