

# Verbale esterno del 2024-03-22

2024-03-22 — v1.0



[overture.unipd@gmail.com](mailto:overture.unipd@gmail.com)

Destinatari	Prof. Tullio Vardanega Prof. Riccardo Cardin <i>Zextras</i> Gruppo <i>Overture</i>
Responsabile	Riccardo Bonavigo
Redattori	Alex Vedovato
Verificatori	Francesco Furno

---

## Registro delle modifiche

Versione	Data	Autori	Verificatori	Dettaglio
1.0	2024-03-22	Alex Vedovato	Francesco Furno	Stesura del verbale

## Indice

1) Contenuti del verbale .....	4
1.1) Informazioni sulla riunione .....	4
1.2) Ordine del giorno .....	4
2) Sintesi dell'incontro .....	4
3) Domande dell'azienda proponente e risposte .....	5
3.1) Potreste fornirci ulteriori dettagli riguardo al funzionamento del codice? .....	5
3.2) Come avete sviluppato i test di unità ed integrazione? Potreste mostrarceli? .....	5
3.3) Qual è stato il vostro approccio allo sviluppo? Come vi siete suddivisi i compiti? .....	5
4) Decisioni prese .....	5

## 1) Contenuti del verbale

### 1.1) Informazioni sulla riunione

- **Luogo:** Chiamata sulla piattaforma dell'azienda Zextras;
- **Ora di inizio:** 10:00;
- **Ora di fine:** 11:00;
- **Partecipanti:** Eleonora Amadori, Michele Bettin, Riccardo Bonavigo, Francesco Costantino Bulychov, Riccardo Fabbian, Francesco Furno, Alex Vedovato;
- **Partecipanti esterni:** Carlo Facci (COO dell'azienda), Fabio Lanciotti (Manager dell'azienda), Federico Rispo.

### 1.2) Ordine del giorno

- Presentazione ed approvazione del prodotto finale come MVP (Minimum Viable Product).

## 2) Sintesi dell'incontro

L'incontro è iniziato con una presentazione condotta da Riccardo Bonavigo sulla struttura generale del nostro prodotto software, illustrando ai partecipanti il diagramma UML delle classi e l'organizzazione del codice. Durante questa presentazione sono stati esaminati i diversi pattern utilizzati, nonché l'architettura logica e di deployment del prodotto, al fine di fornire una panoramica dettagliata delle fondamenta del server. Subito dopo Riccardo Bonavigo ha proceduto a dimostrare le funzionalità implementate, in primis tramite l'utilizzo di Postman e successivamente attraverso il client ltt.rs. Quest'ultimo infatti ci permetteva di dar prova soltanto di alcune delle capacità del nostro prodotto, come la ricezione e l'invio di un'email, ma non implementava le operazioni relative alle cartelle. Queste sono state dimostrate tramite Postman, che ci ha permesso inoltre di dare prova anche della corretta gestione di alcuni errori implementata dal server.

Dopo aver chiarito alcuni dubbi dell'azienda, la quale ha ritenuto soddisfatti tutti i requisiti obbligatori del capitolato e superati i test di accettazione relativi alla funzionalità implementate, abbiamo proceduto con i test di carico implementati utilizzando Locust. Francesco Costantino Bulychov ha quindi illustrato gli scenari da noi proposti, andando poi ad eseguire i test, dimostrando così le performance della soluzione da noi proposta.

La riunione si è conclusa con alcune domande da parte dell'azienda riguardanti alcune parti di codice, i test da noi svolti e il nostro approccio allo sviluppo. Il tutto si è svolto come previsto e l'esito è stato positivo: il prodotto è stato approvato come MVP. Infine, abbiamo comunicato all'azienda che non procederemo con la terza revisione del prodotto, nota come CA (Customer Acceptance), e che il nostro progetto terminerà con la revisione PB.

### 3) Domande dell'azienda proponente e risposte

#### 3.1) Potreste fornirci ulteriori dettagli riguardo al funzionamento del codice?

Per quanto riguarda il funzionamento del codice, abbiamo una classe chiamata BindModule che si occupa di istruire Guice su come gestire le dipendenze. Inizialmente la classe Init crea un Injector, specificando come Modulo di iniezione un'istanza della classe BindModule appena citata. A questo punto, l'iniettore viene utilizzato per creare un'istanza della classe Spark (incaricata per ricevere le richieste http). Quest'ultima viene avviata tramite il metodo start(). Ora le richieste possono entrare nel sistema attraverso le apposite porte in ingresso, essere gestite inizialmente da controller specifici, i quali effettuano una prima validazione, ed infine dalla classi di business che si occuperanno di gestire effettivamente le richieste e di persistere i dati interagendo con le porte in uscita dedicate all'interfacciamento con il database.

#### 3.2) Come avete sviluppato i test di unità ed integrazione? Potreste mostrarceli?

I test di unità ed integrazione sono stati sviluppati utilizzando Junit, Mockito e TestContainers. JUnit ci consente di eseguire test unitari in modo efficiente, Mockito ci aiuta a isolare il codice sotto test dalle sue dipendenze, mentre TestContainers facilita il testing delle componenti esterne. Riccardo Bonavigo ha quindi proceduto a presentare alcuni test implementati.

#### 3.3) Qual è stato il vostro approccio allo sviluppo? Come vi siete suddivisi i compiti?

Conformemente alle linee guida del progetto, i ruoli assegnati ai vari membri del team sono stati alternati in modo che ognuno potesse assumere responsabilità diverse. Abbiamo deciso di adottare periodi di due settimane, dove ogni membro ricopriva ogni volta ruoli diversi in base al lavoro necessario da svolgere. I ruoli in questione sono Responsabile, Amministratore, Verificatore, Analista, Progettista e Programmatore. Il lavoro veniva pianificato all'inizio di ogni periodo, andando a tracciare tutto quanto in un documento apposito denominato Piano di Progetto, all'interno del quale è possibile trovare aspetti chiave come l'Analisi dei Rischi, il modello di sviluppo adottato, la pianificazione delle attività, la suddivisione dei ruoli, nonché stime dei costi e delle risorse necessarie. Per quanto concerne la qualità invece, abbiamo definito delle metriche sia per i processi che per il prodotto, andando a riportare in un documento intitolato Piano di Qualifica i relativi cruscotti, i quali ci hanno fornito una visione completa e dettagliata delle prestazioni e della conformità del nostro lavoro, indicandoci cosa migliorare e cosa invece stavamo già facendo bene.

Tutto il lavoro svolto è stato guidato da un documento chiamato Norme di Progetto, definito da noi con lo scopo di descrivere le regole relative al Way of Working adottato da parte del gruppo per lo svolgimento del progetto didattico.

### 4) Decisioni prese

- Il prodotto è stato approvato come MVP.

