

## Titolo

SmartTravel: Progettazione e Sviluppo di un E-Commerce di Pacchetti Viaggio

## Tutor

Fabio Pasqua

## Tema, Obiettivi e Piano di Attività dello stage

L'obiettivo dello stage è di progettare e sviluppare un'applicazione e-commerce dedicata alla vendita di pacchetti viaggio (voli, hotel, tour organizzati, ecc). L'applicazione dovrà gestire la consultazione dei prodotti, il processo di acquisto, la gestione degli ordini e l'integrazione con terze parti (ad esempio, servizi di localizzazione su mappa e i pagamenti).

L'applicazione dovrà rispettare gli standard e le best practice di programmazione attuali, utilizzando le diverse tecnologie per lo sviluppo dei vari componenti: **Backend** con microservizi Spring Boot (Java) e Quarkus (Java), **Code di messaggi o Topic** con RabbitMQ o similari, **Frontend** con Angular o React e **Database** con MongoDB (non relazionale).

## Descrizione dell'Applicazione

### *Area Clienti (Frontend)*

- Gli utenti potranno cercare, visualizzare e selezionare pacchetti viaggio.
- Sarà possibile registrarsi, effettuare il login, gestire il proprio profilo e visualizzare lo storico ordini.
- Verranno utilizzate tecnologie Angular o React per lo sviluppo di un'interfaccia reattiva, modulare e facilmente manutenibile.

### *Backend a Microservizi*

- **Microservizio di Gestione Utenti** (Spring Boot): si occuperà di gestire l'autenticazione, l'autorizzazione, la creazione del profilo utente e la sua gestione.
- **Microservizio di Catalogo Viaggi** (Spring Boot o Quarkus): offrirà l'accesso e la gestione delle informazioni sui pacchetti viaggio, incluse descrizioni, foto, disponibilità e prezzi.
- **Microservizio di Ordini e Pagamenti** (Quarkus): gestirà la logica di creazione e monitoraggio degli ordini, incluse le richieste di pagamento verso servizi esterni (es. PayPal, Stripe, ecc.).
- **Microservizio di Notifiche** (opzionale, Spring Boot o Quarkus): invierà notifiche email o push ai clienti, ad esempio per confermare l'acquisto o segnalare modifiche e aggiornamenti.

### *Comunicazione Asincrona*

- Verrà utilizzata **RabbitMQ** (o un sistema di messaggistica equivalente) per la comunicazione asincrona tra i microservizi.
- Ad esempio, all'acquisto di un pacchetto viaggio, il Microservizio di Ordini pubblicherà un messaggio di "Ordine Creato" su un topic/queue.
- Al ricevimento del messaggio, il Microservizio di Notifiche invierà un'e-mail di conferma al cliente.

## Database

- **MongoDB** gestirà i dati relativi a utenti, pacchetti, ordini e pagamenti in un formato non relazionale.
- L'adozione di MongoDB permetterà di memorizzare in maniera flessibile i dati dei pacchetti (incluse informazioni variabili come descrizioni multilingua, foto aggiuntive, dettagli specifici di una destinazione, ecc.).

## Sicurezza

- Ogni microservizio esporrà API REST protette da meccanismi di autenticazione (JWT o OAuth2).
- Verranno implementati pattern di sicurezza come l'utilizzo di token sicuri, HTTPS per la comunicazione, gestione delle eccezioni e logging centralizzato.

Il candidato, dopo aver seguito un breve periodo di formazione, sarà inserito nel ciclo di sviluppo del progetto seguendo una metodologia Agile: raccolta dei requisiti, fasi di progettazione, fase di sviluppo, fase di test, fase di integrazione, produzione della documentazione tecnica necessaria.

Al termine del periodo di stage potrà essere valutato il suo inserimento in azienda.

## Prerequisiti richiesti allo studente/competenze tecniche attese in ingresso

- Conoscenza base della programmazione OOP in Java (o altro linguaggio)
- Conoscenza base dei linguaggi HTML, Javascript e della tecnologia CSS
- Familiarità con il linguaggio SQL

## Obiettivi Formativi dello Stage

- Approfondire la conoscenza di **Java** e dei framework **Spring Boot** e **Quarkus** in ambito microservizi.
- Sperimentare i benefici della comunicazione **asincrona** tramite **RabbitMQ** (o altra piattaforma di messaggistica).
- Acquisire competenze nello sviluppo di **frontend** web con **Angular** o **React**, seguendo pattern di progettazione modulari (SPA, component-based).
- Imparare a progettare e gestire un database **NoSQL** come **MongoDB**, approfondendo i concetti di schema design flessibile, sharding, e indexing.
- Applicare strategie di **Continuous Integration / Continuous Delivery** per rilasciare software in modo frequente e controllato.
- Sviluppare buone pratiche di **collaborazione in team**, versionamento del codice (Git), code review e documentazione.

## Dotazioni

PC aziendale

## Agevolazioni

Ticket Restaurant