

# Esercitazione - colloquio per Backend Developer

L'esercitazione prevede di realizzare una **REST-API** che faccia uso di un **Database relazionale** per lo strato di persistenza e che possa interrogare **REST-API esterne** al fine di realizzare la propria logica di business.

## Contesto applicativo

Il contesto applicativo è quello di una REST-API che consente ad ogni **Utente** di consultare, creare ed aggiornare i propri **Itinerari** di viaggio.

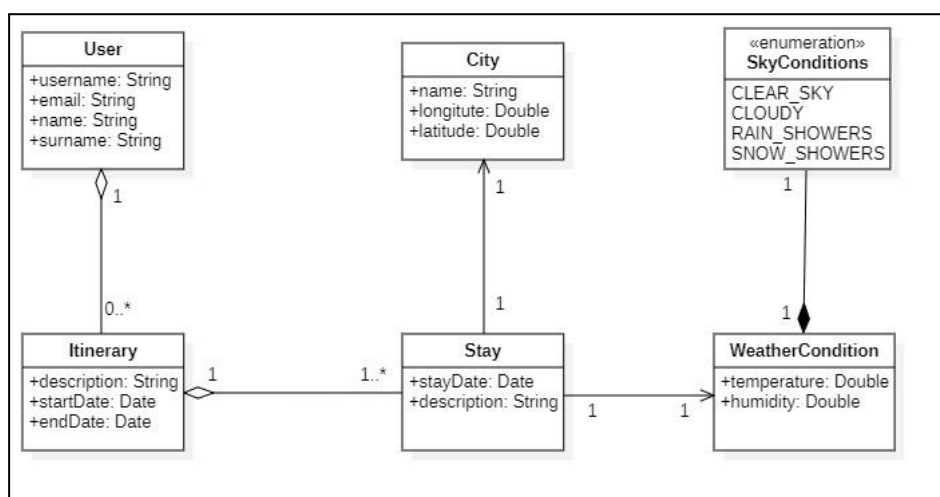
Ogni Itinerario è composto, oltre che da alcuni attributi di base quali la descrizione e l'intervallo di tempo riservato, da almeno due **Tappe**.

Ciascuna Tappa sarà riferita ad una determinata **Città** che dovrà essere visitata in una **Data specifica**.

Ad ogni Tappa saranno inoltre associate le relative **Condizioni Meteo** della Città, prese alla data d'interesse (*quella in cui la Città della Tappa sarà visitata*).

Le Condizioni Meteo di ogni Tappa dovranno includere almeno informazioni su temperatura, umidità e condizione del cielo (*sereno, variabile, piovoso*).

Come **contesto minimo di riferimento** si può considerare quello schematizzato di seguito, aggiunte ed arricchimenti ragionevoli (*anche solo negli attributi delle Entità*) sono ben accetti.



## Traccia di Requisiti funzionali

I seguenti requisiti funzionali sono da intendersi come insieme minimale necessario a realizzare i casi d'uso di creazione e modifica degli Itinerari dell'Utente.

Tramite gli endpoint della REST-API, un Utente deve poter **consultare, creare ed aggiornare la lista di Città** presenti sul Database. Questa sarà poi utilizzata in fase di creazione delle Tappe di un itinerario.

L'endpoint di creazione di una Città deve richiedere di **specificare soltanto il Nome** della città di interesse: le informazioni aggiuntive (*latitudine, longitudine, altezza sul livello del mare*) devono essere **popolate automaticamente**, tramite interfacciamento con una **REST-API di Geocoding** (*vedi Note e riferimenti*). La gestione della cancellazione di Città è lasciata come *opzionale*.

Analogamente al caso delle Città, la REST-API dovrà esporre endpoint per la **consultazione, creazione ed aggiornamento delle Tappe** definite dall'**Utente** (*anche in questo caso la cancellazione è opzionale*). L'endpoint di creazione di una Tappa richiede di specificare - oltre all'Itinerario di cui la Tappa è parte - anche una Città ed una Data, dovrà poi essere la REST-API stessa a corredare la Tappa con le **informazioni meteo adeguate**, ottenute tramite l'interfacciamento con una **REST-API di Weather Forecast** (*vedi Note e riferimenti*). Una procedura analoga dovrà essere eseguita in caso di aggiornamento di una Tappa.

Ogni **Itinerario**, dal momento della creazione in poi, deve avere sempre un **determinato Stato** che rappresenti le seguenti situazioni:

- Itinerario appena creato - *creato ma ancora senza Tappe*
- Itinerario pronto - *creato e con le Tappe, ma ancora da confermare*
- Itinerario confermato - *creato, con le Tappe, e confermato*
- Itinerario concluso - *confermato e concluso (es: la data di conclusione dell'itinerario è precedente a quella attuale)*

La realizzazione di funzionalità ulteriori, anche in termini di quante e quali informazioni sono ottenute da interfacciamento con REST-API, sono opzionali.

## Vincoli tecnologici

- La REST API dovrà essere realizzata con **framework Spring**
- Il layer di persistenza della REST-API dovrà essere realizzato con un DB relazionale e tramite il **framework MyBatis**
- L'interfacciamento con le API esterne dev'essere implementato usando l'**interfaccia WebClient** predisposta dal framework Spring

## Note e riferimenti

- Ai fini dell'elaborato non è necessario implementare alcuna forma di autenticazione. Questo rimane quindi come aspetto *opzionale*.
- Come **REST-API** di riferimento si può utilizzare **Open Meteo**, che è open e non necessita di API-Key.
  - Per il geocoding riferirsi a:  
<https://open-meteo.com/en/docs/geocoding-api>
  - Per il weather forecast riferirsi a:  
<https://open-meteo.com/en/docs>
- L'implementazione degli aspetti opzionali è considerata positivamente solo qualora sia realizzata in maniera corretta.