

Università degli Studi di Napoli Federico II
Esame di Advanced Computer Programming
Proff. De Simone, Della Corte

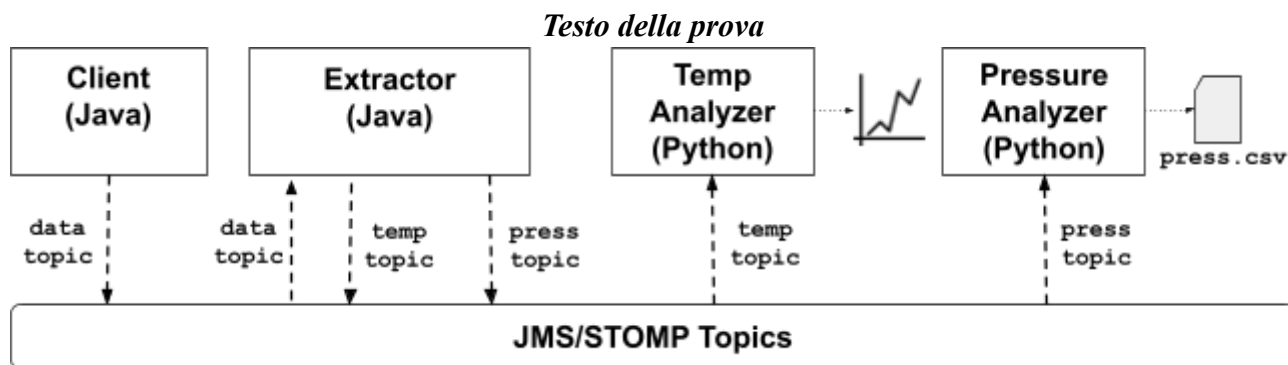
Proff. De Simone, Della Corte

Prova pratica del giorno 08/09/2023

Durata della prova: 120 minuti

*Lo studente legga attentamente il testo e produca il programma ed i casi di test necessari per dimostrarne il funzionamento. La mancata compilazione dell'elaborato, la compilazione con errori o l'esecuzione errata daranno luogo alla valutazione come **prova non superata**.*

Al termine della prova lo studente dovrà far verificare il funzionamento del programma ad un membro della Commissione.



Il candidato realizzi un sistema di analisi dati meteo basato su **JMS** e **STOMP**. Il sistema include i seguenti componenti:

- **Client.** Componente Java che genera richieste di tipo *temperature* o *pressure*, sotto forma di messaggi JMS. Client attende 2 secondi tra le richieste, e genera un totale di 20 richieste. Per ogni richiesta, Client genera un MapMessage composto da due campi: **type**, che conterrà il tipo di richiesta (cioè *temperature* o *pressure*), e **value**, che conterrà un *intero* scelto a caso tra 0 e 100 (per richieste *temperature*) e tra 1000 e 1050 (per richieste di *pressure*). Il MapMessage è poi inviato sul topic *data*. Il tipo di richiesta è fornito da terminale. Ad esempio: **java nomepackage.Client temperature**
- **Extractor.** Componente Java che implementa la ricezione asincrona JMS sul topic *data*. Alla ricezione di ciascun MapMessage, il *listener* JMS estrae i due campi (*type* e *value*), e prepara un TextMessage nel quale inserisce il valore presente nel campo *value*. Il TextMessage generato viene inviato sul topic *temp*, nel caso di richiesta di tipo *temperature*, o su quello *press*, in caso di richiesta *pressure*.
- **TempAnalyzer.** Componente Python che implementa la ricezione asincrona STOMP sul topic *temp*. Alla ricezione di ciascun TextMessage, il componente preleva la stringa ricevuta, trasforma la stringa in valore numerico, e lo scrive in una Queue. Dopo la ricezione di 20 TextMessage, il componente genera un *lineplot* con i valori di temperatura prelevati dalla Queue, e lo mostra a video.
- **PressAnalyzer.** Componente Python che implementa la ricezione asincrona STOMP sul topic *press*. Alla ricezione di ciascun TextMessage, il componente preleva la stringa ricevuta, trasforma la stringa in valore numerico, e lo scrive in un file nel formato CSV (*press.csv*) insieme a un valore numerico che viene incrementato a ogni ricezione.

Lo studente dovrà sviluppare le applicazioni **Client**, **Extractor**, **TempAnalyzer**, e **PressAnalyzer**. Il sistema sarà testato da terminale con 2 Client (uno per *temperature* e uno per *pressure*), 1 Extractor, 1 TempAnalyzer e 1 PressAnalyzer.