

Matteo Scala

Java Backend Developer con ~9 anni di esperienza e una strana passione per self-hostare le cose che gli interessano

I testi in verde sono collegamenti



Info Personali

Nato nel 1998

Residente a Faenza

Con Patente B

Contatti

scala.matteo.98@gmail.com

Link

[GitHub](#)

[LinkedIn](#)

Lingue Straniere

Inglese (B2)

Altro

Corsi sicurezza sul lavoro

Competenze

Linguaggi

Python

Golang

Javascript / Typescript

Bash

Database

SQL / SQLite

MongoDB

Redis

Altri Strumenti

Git / SVN / RTC

Docker

NginX / Caddy

Virtuozzo

Esperienze Lavorative

VM Sistemi - SMEUP BSA / Apr 2023 — presente

Ho lavorato con **Java EE**, Jboss e Wildfly, sviluppando integrazioni con fornitori del mondo del materiale elettrico riguardanti importazioni automatiche di dati e interfacciamento con basi dati esterne. Ho imparato a

- seguire lunghi progetti, documentare e specificare, testare (e leggere un po' di COBOL), quando mi è stato chiesto di riscrivere il servizio che gestisce l'anagrafica dell'ERP proprietario per renderlo accessibile tramite HTTP,
- gestire 6 o 7 diverse combinazioni di versioni di Java, web server, librerie, di sistema operativo, piattaforme di accesso (per accontentare tutti i clienti)
- imbastire macchine virtuali con hypervisor e assistere nell'adozione di Docker
- leggere e adattare flussi di CI/CD (vecchi e) personalizzati

GFT Italia / Apr 2022 — Apr 2023

Ho approfondito la mia conoscenza di **Java** e altri linguaggi come JSP e HQL, acquisendo esperienza con architetture complesse, software proprietario e tecnologie legacy durante consulenze in ambito bancario. Ho imparato a coordinare il lavoro con il mio team (di 5 persone, completamente remoto, anche con **Jira**) utilizzando **SVN** e a navigare log complessi. In azienda mi hanno dato la possibilità di partecipare a gruppi di approfondimento sull'**AI** dove ho portato una **proposta** per una tecnica per applicare un *watermark* a un qualsiasi LLM

Halnet Srl / Nov 2017 — Apr 2022

Qui ho lavorato mentre portavo avanti gli studi universitari. Ho sviluppato servizi di monitoraggio per operazioni critiche di clienti e personalizzato interfacce web di gestionali utilizzando **Vaadin**, oltre a sviluppare alcuni strumenti interni. Ho maturato le mie prime esperienze lavorative come programmatore **Java** e come gestore di infrastruttura: ho impostato il repository interno di **Maven** e di **GitLab**. Qui ho anche sviluppato e scritto la tesi.

Istruzione e Formazione

UniBo / Nov 2017 — Lug 2021

Corso di **Ingegneria e Scienze Informatiche**. La **tesi di laurea** consisteva in un elaborato di *Data Intensive Application* nel quale ho scritto un sistema di classificazione di stenosi coronariche, tramite la combinazione di metodi di **apprendimento automatico**, sviluppati con **Colab**

Progetti Personali

- **Deep Learning for ECG Segmentation**: Ho realizzato un'implementazione open-source di uno studio sul **riconoscimento automatico** di diversi tipi di onde in elettrocardiogrammi. Poiché il codice originale non era stato rilasciato, l'ho scritto da zero, accettando alcuni compromessi e semplificazioni (nel paper si parla di resampling delle frequenze che io non ho fatto e di un dataset più ampio che io non ho trovato), ottenendo comunque un modello in grado di identificare i tipi di onde.
- **NoNoisePlease**: È un sito (+ estensione per browser) che permette di salvare pagine web, le rende cercabili attraverso un indice sulle parole nel testo (gestito con **Meilisearch**) e accessibili durante le ricerche su Google. Con questo sito sto consolidando le mie conoscenze di Golang (con il quale è scritto tutto il backend) e di AlpineJS, inoltre ho sviluppato il mio personale ciclo di **CI** e **CD**, che include one-click-rollback, test e backup automatici, rilasci in 10 secondi di downtime, indipendente da GitHub e completamente basato su Docker ed SSH
- **Titolarly**: App Android per aiutare nella scelta della formazione del fantacalcio. Sviluppata con **Flutter**, con backend in **Typescript** per gestire sia le API che il web scraping. I dati erano memorizzati in **MongoDB** (che al tempo volevo imparare) e l'intera applicazione era ospitata su un server **Linux** su **AWS**. Creare quest'app mi ha dato l'occasione di mettere le mani su alcuni servizi gestiti di Firebase e AdMob per analytics e pubblicità.
- **Folkorama**: Sito web per gestire **loreserver** per team che sviluppano videogiochi. Folkorama è ospitato in una VM, scritto in **Golang**, ed è in grado di gestire più VM (in caso di saturazione) comunicando con esse tramite **gRPC** per istanziare container **Docker** all'interno dei quali girano i loreserver. Fornisce tutto il layer di autenticazione e autorizzazione.