Fernando Romulo da Silva

Address: Rua do Monte Alegre nº 98 left, Zip Code: 4250-269, Cedo Feita, Porto - Portugal / Rua Paula Rodrigues, nº 250, Piratininga, Zip Code: 06233-030 bloco 16 apto 43 Osasco - SP - Brazil
Personal Page: https://fernando-romulo-silva.github.io/personal-page/
Cell Phone: +351 914-090-874 / +55 (11) 97046-6053
E-mail: fernando.romulo.silva@gmail.com

PROFESSIONAL SUMMARY

I am a senior full-stack certified software engineer, highly motivated by new Java platform technologies. Although my primary qualifications are Java back-end and SQL databases, I have also worked with JavaScript front-end and NoSQL databases. I have experience in analysis, design, coding, testing (automated and manual), and delivering applications like REST and SOAP web services, web applications, batch jobs, web crawlers, and desktop applications. During my more than 15 years of experience, I have worked in a few business sectors like ERP, content management systems, employee benefits solutions, insurance provider solutions, credit reporting agencies (credit bureau), and financial solutions.

SKILLS

- **Languages:** I am a native Portuguese speaker and an upper intermediate English speaker.
- **Java:**
  - **Versions:** JDK5 until JDK17, GraalVm, DCEVM and HotSwapAgent.
  - **Libraries:** JUnit, TestNG, AssertJ, Mockito, HtmlUnit, JSoup, ApacheUtils, OpenAPI, OpenFeign, QueryDSL, Guava, etc.
  - **Tools:** Ant, Maven, Spotbugs, PMD, Checkstyle and Gradle.
  - **Frameworks:**
    - Spring: Spring Framework, Spring Boot, Spring Security, Spring Data, Spring Batch and Spring MVC.
    - JEE: JSF, EJB, JTA, JPA, CDI, JAX-WS, JAX-RS, JMS, batch Processing, etc.
    - Others: Swing, Quarkus and Hibernate.
- **Version Control**: Git, Github, Bitbucket and Gitlab.
- **Java Script:**
  - **Version:** JavaScript ES6.
  - **Libraries:** React, Lodash, Axios, Jest and JQuery.
  - **Tools:** JSLint, npm, Webpack.
- **IDES**: Eclipse, Dbeaver and Visual Studio Code.
- **Data Bases:**
  - **SQL:** Oracle, PostgreSQL, SQL Server and MySql.
  - **NoSQL:** MongoDB.
- **Servers:** Wildfly, Glassfish, Tomcat, and Jetty.
- **Project Management:**
  - **Frameworks:** Scrum (daily meeting, task breakdown, grooming, planning)
  - **Software:** Jira, Kanbanize, Microsoft Team Foundation and Redmine.
- **Infrastructure:** Virtual Box, Docker, Docker Compose, and Kubernetes.
- **Messaging:** JMS (ActiveMQ Artemis), Google Pub/Sub and Rabbit MQ.
- **Cloud:** Google Cloud Platform (GCP).
- **Operation Systems:** Windows desktop (all flavors) and Ubuntu Linux.
- **Others:** SOLID, Object Oriented Design, Domain Driven Design, UML, TDD and Design Patterns.

EDUCATION

- **Postgraduate Degree – Software Engineering Specialization,** 03/2018 – 12/2020
  Pontifíca Universidade Católica de São Paulo
  Coursework with emphasis in software development and engineering using new technologies.

- **Technical Course – W1 Desenvolvimento Avançado de Interfaces Web,** 02/2014 to 03/2014
  Global Code Treinamentos – São Paulo, SP – Brazil
  Coursework to learn web technologies like Html5, XPath, JPath, Javascript, and JQuery.

- **Technical Course – Curso Java para Desenvolvimento Web,** 06/2009 to 09/2009
  Caelum Ensino e Inovação – São Paulo, SP – Brazil
  Coursework for learning Java Web technologies, JSP, Servlet, Struts, and JSF.

- **Bachelor of Science – Computer Science,** 03/2004 to 12/2006
  UNOESTE – Universidade do Oeste Paulista – Presidente Prudente – SP Brazil
  Coursework with emphasis on programming, develop computer tools, and software design.

CERTIFICATIONS

- Oracle Java EE 7 Application Developer

- Oracle Java SE 8 Programmer

- Oracle Certified Master, Java SE 6 Developer

- Sun Certified Java Programmer 5

WORK HISTORY

- **Java Software Engineer,** 04/2022
  Dalenys – Porto – Portugal
  I develop and fix cloud-native marketplace applications using Spring Boot, Rabbit MQ, GCP, MongoDB, and MySQL Database Server.

- **Java Software Engineer,** 12/2019 to 03/2022
  Boa Vista SCPC – Barueri, SP – Brazil
  I developed cloud-native credit bureau applications using Spring Boot, Docker, Kubernetes, GCP, Google Pub/Sub, Redis, PostgreSQL, and MySQL Database Server. I improved Java applications using advanced docker and Spring resources.

- **Java Software Engineer,** 04/2015 to 12/2019
  Fleetcor VB Serviços – São Paulo, SP – Brazil
  I developed browser automation projects and employee benefits solutions using Java, JSoup, HtmlUnit, Spring, JAX-WS, JAX-RS, JBoss, Wildfly, Jetty, Tomcat, and Oracle Database Server. I used multi-thread processing (Java Executor Framework) to gain performance on web navigation.

- **Java Developer,** 06/2014 to 04/2015
  BNP Paribas Cardif – São Paulo, SP – Brazil
  I developed and maintained Java projects for insurance providers using Java, Spring, Web Services SOAP, Tomcat, Jetty, and SQL Server. I revamp various applications developed on Java core using Spring framework.

- **Java Developer,** 09/2009 to 12/2013
  VB Serviços – São Paulo, SP – Brazil
  I worked with employee benefits software solutions in JEE (JSF, EJB, CDI, JSP, Servlet, JMS, JAX-WS, and JAX-RS), Oracle Database Server, and Wildfly servers. I developed back-end software, batch applications, and sometimes, front-end pages server-side for most of my time.

- **Java Developer,** 04/2008 to 08/2009
  Vetor Web – Negócios na Internet - São Paulo, SP – Brazil
  I worked with content management systems done in JavaScript, JQuery, Java, JSP, JSF, Spring, Tomcat, and MySQL. I developed, for its time, advanced interfaces features using JavaScript and Java.

- **Delphi and SQL Developer,** 04/2007 to 04/2008
  DB1 Informática Software e Consultoria – Maringá, Paraná – Brazil
  I was a Delphi and Oracle SQL developer in ERP projects. I develop and fixing applications mainly for medium/large wholesale companies.

RECENT PROJECTS

Company: Boa Vista SCPC

Project 01: API for Layouts (New Project)

Stack: Java 11, Spring Boot 2.5, Spring Security, JUnit, AssertJ, MySQL, Docker, Kubernetes and GCP

Role: Senior Software Engineer

Description: This API provides file layouts on JSON and XML formats. These file layouts are used on batch processing on GCP. Due to layouts having to follow strict rules, the application has to validate a lot of situations; I used automated tests like unit (TDD), integration, and acceptance tests (Spring Mock MVC) to solve this problem.

Steps:
- Create domains classes and tests
- Create services classes and tests
- Create endpoint classes and tests
- Create OpenAPI (Swagger) documentation
- Create the readme.md file
- Create Dockerfile with security and perfomance
- Create Kubernets yml files (service, deployement, kustomization)

Company: Boa Vista SCPC

Project 02: Batch Processing For Debts (New Project)

Stack: Java 11, Spring Boot 2.5, Spring Batch, JUnit, AssertJ, PostgreSQL, Docker, Kubernetes and GCP

Role: Senior Software Engineer

Description: This is a batch application that reads some specific files, usually 3GB, treats and stores the data (credit score) on a database. The business rules were simple, but some analysis had to be done.

Steps:
- Create domains classes and tests
- Create services classes and tests
- Create batch classes and tests
- Create the readme.md file
- Create Dockerfile with security and perfomance
- Create Kubernets yml files (service, cron job, kustomization)

Company: Boa Vista SCPC

Project 03: Credit Score Web (Maintenance)

Stack: Java 8 and 11, WebSphere JEE, Spring Security, Spring Boot, PostgreSQL, Docker, Kubernetes and GCP

Role: Senior Software Engineer

Description: An upgrade for a web application, a credit score consulting. This project was a transforming monolithic architecture towards micro service architecture. Only modules that made sense.

Steps:
- Update domains classes
- Update services classes
- Create documetation
- Create automated tests

Company: Boa Vista SCPC

Project 04: Communication Service (New Project)

Stack: PubSub, Spring Boot, MongoDB, Docker, Kubernetes and GCP

Role: Senior Software Engineer

Description: This application is a communication broker for other applications. Its responsibility is to send emails, SMS, etc, and manage the messages. It works as messaging consumer and API.

Steps:
- Create domains classes and tests
- Create services classes and tests
- Create endpoint classes and tests
- Create OpenAPI (Swagger) documentation
- Create the readme.md file
- Create Dockerfile with security and perfomance
- Create Kubernets yml files (service, deployement, kustomization)