

# Fernando Romulo da Silva

---

| Software Engineer | Full-stack | Java | Java & Jakarta EE | Spring Frameworks | SQL | NoSQL | Docker |  
| Kubernetes | JavaScript | TypeScript | React |

**Address:** Cedo Feita, Porto – Portugal, **Personal Page:** <https://fernando-romulo-silva.github.io/personal-page>

**Cell Phone:** +351 914-090-874, **E-mail:** [fernando.romulo.silva@gmail.com](mailto:fernando.romulo.silva@gmail.com)

**Relocation:** Canada, EUA, Ireland, Australia and UK

## PROFILE SUMMARY

---

I am a senior full-stack certified software engineer, highly motivated by Java platform and JavaScript technologies. Although my primary qualifications are Java back-end, front-end server-side, 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, console applications, libraries, 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.

## EXPERIENCE

---

### ● Senior Full-stack Java and JavaScript Software Engineer, 04/2022 to 12/2023

**Natixis** – Porto – Portugal (*contract through Aubay Portugal*)

I develop and maintain cloud-native financial applications (marketplace and data lake applications) using Java, Spring Boot, RabbitMQ, GCP, JavaScript, Typescript, React, MongoDB, and MySQL. I have been troubleshooting, resolving complex technical problems, and collaborating with my team to achieve its objectives.

#### Activities:

- Conduct and participate actively in Scrum meetings.
- Presenting demonstrations for users and other developers.
- Fix and update front-end applications.
- Fix and develop new features back-end (Rest API and Messaging) and docs OpenAPI (Swagger).
- Create automated tests and test other developers' tasks.
- Code review on merge request.
- Deploy applications (CI/CD).

### ● Senior Back-end Java Software Engineer, 12/2019 to 03/2022

**Boa Vista SCPC** – Barueri, SP – Brazil (*first as a contract through Mooven, later as a permanent employee*)

I developed cloud-native credit bureau applications (debt checking, renegotiating debts, and credit score provider) using Spring Boot, Docker, Kubernetes, GCP, Google Pub/Sub, Redis, PostgreSQL, and MySQL Database Server. I improved Java applications by leveraging innovative Docker and Spring technologies to enhance performance and reliability.

#### Activities:

- Participate actively in Kanban meetings.
- Create and improve Docker and Kubernetes applications on the GCP cloud.
- Fix and create new features back-end (Rest API, Messaging, and batch) and docs OpenAPI (Swagger).
- Create automated tests.
- Adapt and enhance code to Sonar measurements.
- Create application documentation.

### ● Senior Back-end Java Software Engineer, 04/2015 to 12/2019

**Fleetcor VB Serviços** – São Paulo, SP – Brazil

I developed web crawlers and employee benefits solutions using Java, JSoup, Apache HttpClient, HtmlUnit, Spring, JAX-WS, JAX-RS, Wildfly, Jetty, Tomcat, and Oracle Database Server. To increase performance on web navigation, I utilized multi-thread processing. Additionally, I designed and implemented several other features, including authentication, session management, and error handling.

#### Activities:

- Create and update Web crawler applications with multi-thread processing.
- Fix and develop new features back-end (Rest API, SOAP, Messaging, and batch).
- Training junior developers.
- Adapt JEE applications to cloud computing.

- **Mid Level Back-end Java Developer**, 06/2014 to 04/2015

**BNP Paribas Cardif** – São Paulo, SP – Brazil (*contract through PSM Company*)

I developed and maintained Java applications for insurance providers, utilizing Java, Spring, Web Services SOAP using Apache CXF, Tomcat, Jetty, and SQL Server. Additionally, I have revamped some applications developed on Java web (JSP, Servlet, Taglib, etc.), leveraging the Spring framework for improved efficiency and functionality.

**Activities:**

- Create and update SQL Server queries and procedures.
- Fix and develop new features back-end (SOAP WS).
- Training users and product owners.
- Integrate software components and third-party programs with applications.

- **Junior/Mid Level Full-stack Java Developer**, 09/2009 to 12/2013

**VB Serviços** – São Paulo, SP – Brazil

As a Java Developer, I worked extensively with employee benefits software solutions utilizing Java Enterprise Edition (JSF, EJB, CDI, JSP, Servlet, JMS, JAX-WS, and JAX-RS), Oracle Database Server, and Wildfly servers. I developed comprehensive back-end software, and front-end pages server-side.

**Activities:**

- Create and update JSF pages.
- Fix and develop new features back-end (EJB, SOAP WS, and JMS).
- Create scripts for deploying applications.
- Recommend and execute improvements.

- **Junior Full-stack Java and JS Developer**, 04/2008 to 08/2009

**Vetor Web** – São Paulo, SP – Brazil

I developed innovative interfaces for content management systems with advanced features using JavaScript, Java, JQuery, JSP, JSF, Spring, Tomcat, and MySQL.

**Activities:**

- Improving pages JSP with JavaScript and JQuery framework.
- Maintain up-to-date documentation about business and technical guides.
- Deploy applications on web containers.

- **Junior Full-stack Delphi and PL/SQL Developer**, 04/2007 to 04/2008

**DB1** – Maringá, Paraná – Brazil

I was a Delphi and Oracle PL/SQL developer working on various ERP projects and had extensive experience creating and maintaining applications for medium to large wholesale companies.

**Activities:**

- Create and update Oracle DB procedures, functions, and triggers.
- Testing and evaluating new Delphi forms.
- Configure applications on computer clients.

## SKILLS

---

- **Languages:** I am a native Portuguese speaker and an advanced English speaker.

- **Java:**

- ◆ **Versions:** JDK 5 until JDK 17, GraalVm, DCEVM and HotSwapAgent.
- ◆ **Libraries:** JUnit, TestNG, AssertJ, Mockito, HtmlUnit, JSoup, ApacheUtils, OpenAPI, OpenFeign, QueryDSL, Guava, etc.
- ◆ **Tools:** Ant, Maven, Gradle, Spotbugs, PMD, and Checkstyle.
- ◆ **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: Quarkus, Hibernate and Swing.

- **JavaScript:**

- ◆ **Version:** JavaScript ES6 and Typescript.
- ◆ **Libraries:** React, React Router, Redux, Tanstack Query, Lodash, Axios, Jest and JQuery.
- ◆ **Tools:** JSLint, Npm, Yarn, and Webpack.

- **Messaging:** RabbitMQ, JMS (JEE provider and ActiveMQ Artemis), and Google Pub/Sub.

- **Data Bases:**
  - ◆ **SQL:** Oracle, PostgreSQL, SQL Server and MySQL.
  - ◆ **PL/SQL:** Oracle and SQL Server.
  - ◆ **NoSQL:** MongoDB and Redis.
- **Infrastructure:** Virtual Box, Docker, Docker Compose, and Kubernetes.
- **Cloud:** Google Cloud Platform (GCP), Rancher (private cloud).
- **Version Control:** Git, Github, Bitbucket and Gitlab.
- **IDES:** Eclipse, Visual Studio Code, Dbeaver, and MongoDB Compass.
- **Servers:** Wildfly, Glassfish, Tomcat, Undertow, and Jetty.
- **Operation Systems:** Windows desktop (all flavors) and Ubuntu Linux.
- **Project Management:**
  - ◆ **Frameworks:** Scrum (daily, task breakdown, grooming, planning, and retro), Kanban (daily, replenishment).
  - ◆ **Software:** Jira, Kanbanize, Microsoft Team Foundation and Redmine.
- **Architecture:** Event-driven architecture, Hexagonal architecture, Microservices, Monolith, and CQRS.
- **Others:** SOLID, Design Patterns, Object Oriented Design, Domain Driven Design, UML, and TDD.

## EDUCATION

---

- **Postgraduate Degree – Software Engineering Specialization**, 03/2018 – 12/2020  
Pontifícia Universidade Católica de São Paulo  
Specialization course with emphasis on software development and engineering using new technologies.
- **Technical Course – W1 Advanced Web Interface Development**, 02/2014 to 03/2014  
Global Code – São Paulo, SP – Brazil  
Coursework to learn web technologies like HTML5, XPath, JPath, Javascript, and JQuery.
- **Technical Course – Java Web Development**, 06/2009 to 09/2009  
Caelum – São Paulo, SP – Brazil  
Coursework for learning Java Web technologies, JSP, Servlet, Struts, and JSF.
- **Bachelor of Science – Computer Science**, 03/2002 to 12/2006  
UNOESTE – Presidente Prudente – SP Brazil  
Computer science major with emphasis on math, computer system knowledge, programming, developing 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

## RECENT PROJECTS

---

**Company:** Natixis

**Project :** Frontend Data Lake (New features)

**Stack:** TypeScript, React 18, Axios, Jest, Yarn, ESLint

**Role:** Senior Software Engineer

**Description:** This a front-end application for the next project described. On this application, I fixed and updated React JS pages.

**Steps:**

- Updated functions and layouts
- Update/create configs files

---

**Company:** Natixis

**Project :** API for Data Lake (New features)

**Stack:** Java 17, Spring Boot 3.0, JUnit, AssertJ, MongoDB, RabbitMQ, Docker, Kubernetes and GCP

**Role:** Senior Software Engineer

**Description:** This API create scheduled reports that read data from a big data structure expose its results on line. I intensively used automated tests to guarantee the quality of its development.

- Steps:**
- Create domains classes and tests
  - Update/create services classes and tests
  - Update/create endpoint classes and tests
  - Update/create OpenAPI (Swagger) documentation

---

**Company:** Boa Vista SCPC

**Project :** 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 to be 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 documentation files (readme.md)
  - Create Dockerfile
  - Create Kubernetes yml files (service, configMap, deployment, kustomization)

---

**Company:** Boa Vista SCPC

**Project:** Batch Processing For Debts (New Project)

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

**Role:** Senior Software Engineer

**Description:** This batch application reads specific files, usually 10GB, treats them, and stores the data (credit score) on a Redis Hashes. 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 documentation files (readme.md)
  - Create Dockerfile
  - Create Kubernetes yml files (service, cron job, kustomization)

---

**Company:** Fleetcor VB Serviços

**Project:** Benefits buyer (Maintenance and Refactor)

**Stack:** Java 11, Wildfly, JEE, Spring Boot, OracleSQL, Docker

**Role:** Senior Software Engineer

**Description:** Refactoring was done by splitting some specific modules (navigation) and transforming a monolithic architecture application towards micro-service architecture.

- Steps:**
- Update/create domains classes
  - Update/create services classes
  - Create/create documetation classes

---

**Company:** Fleetcor VB Serviços

**Project:** Job Scheduler (New Project)

**Stack:** Java 11, JMS, Spring Boot, MySQL, Docker

**Role:** Senior Software Engineer

**Description:** This application is a job executor for other applications, listening requests from Rest API, SOAP, and JMS. But some JSF pages was developed to create executions, too.

- Steps:**
- Create domains classes
  - Create services classes
  - Create endpoint classes
  - Create JMS client
  - Create Dockerfile