Sergio Mínguez Berja

Alicante, Spain | +34 635 046 505 | sfm181818@gmail.com
Portfolio: smb96-ua.github.io | LinkedIn: sergio-minguez | GitHub: smb96-ua

Professional Summary

A highly motivated and soon-to-be-graduated Software Engineer with a proven passion for low-level systems programming in C++. Possesses hands-on experience in developing concurrent applications from the ground up, with a strong command of pthreads, mutexes, and semaphores. Fascinated by the core technologies that power the web and deeply driven by the opportunity to contribute to the performance, web compatibility, and standards-correctness of a world-class engine like Gecko.

Education

Bachelor of Science in Computer Engineering
University of Alicante, Spain
Expected Graduation: Spring 2025

Technical Skills

- Core Languages: C++ (Proficient), JavaScript, Java, PHP, Python, SQL
- Concurrency & Systems: pthreads, Multi-threading, Semaphores, Mutexes, POSIX API, Socket Programming
- Web Technologies: HTML5, CSS3, REST APIs, Spring Boot, Angular, Laravel
- Libraries & Tools: SFML, STL, Git, GitHub, Docker, Maven, Linux/Unix Environment
- Key Concepts: Object-Oriented Programming (OOP), Data Structures & Algorithms, Memory Management, Distributed Systems, MVC Architecture
- Languages: Spanish (Native), English (C1 Certified)

Relevant Projects

Concurrent Producer-Consumer System (C++)
A multi-threaded application developed to solve the classic producer-consumer synchronization problem.

- Implemented thread creation and lifecycle management using the low-level pthreads API, demonstrating a practical understanding of systems-level concurrency required for browser engine development.
- Utilized semaphores and mutexes to prevent race conditions and ensure thread-safe access to shared memory buffers, a critical skill for multi-process architectures like Firefox's.
- Technologies: C++, pthreads, Semaphores, Mutexes, Linux.

Billadictos — 2D Billiards Simulator (C++)
A 2D billiards game built from scratch, showcasing the ability to architect a software system with a clear lifecycle.

- Engineered the core application loop, event handling (user interaction), and rendering logic using the SFML multimedia library.
- This project demonstrates experience in managing a complex codebase and integrating external libraries, analogous to working within a large framework like Gecko.
- Technologies: C++, SFML, Object-Oriented Design, Git.

EasyCab — Distributed Taxi System (Java & Microservices)

A sophisticated distributed system simulating a taxi service, built on a microservices architecture.

- While built in Java, this project proves a deep understanding of complex software architecture, inter-process communication (via RabbitMQ), and scalability—concepts directly applicable to the challenges of modern web browsers.
- Technologies: Java, Spring Boot, RabbitMQ, Microservices, Docker.