# David Ferrari

Software Engineer | Computer systems focus

## CONTACT

**Phone**: +39 3283169485
**Github**: github.com/ferraridavid-hub
**Address**: Bologna (Italy)
**Email**: ferraridavid.hub@gmail.com
**Linkedin:** https://www.linkedin.com/in/dav%C3%ACd-ferrari-239131269/

## PROFESSIONAL EXPERIENCE

**Software Engineer | Primeur Srl, Milan (Italy) | 06/2023 - Present**

- **End-to-end ML API** for pattern recognition in large-scale event timing datasets, implementing clustering algorithms to identify recurring patterns and developing supervised learning pipeline for anomaly detection
- **Developed automated migration module** for transition from legacy C application to new Quarkus system, automatically exporting configurations and mapping all functionalities
- **Secured enterprise Java distributed application** against OWASP Top Ten vulnerabilities
- **Built and maintained CI/CD pipelines** using Jenkins and Docker
- **Presented technical demos** of new applications and features during company training sessions

**IT Analyst | Cams Srl, Bologna (Italy) | 01/2022 - 04/2023**

- **Automated several IT workflows** improving operational efficiency
- **Designed and developed company website** and customer care systems

## PERSONAL PROJECTS

**Orthogonal Polynomial Library (C++):** A physics computing library developed during University studies. Focus on high-level abstraction and numerical computation
**Terminal-based text editor (C++, ongoing):** a vim-inspired text editor. Currently featuring raw mode terminal I/O, keyboard event handling and file operations. Focus on UNIX system programming
**Matrix multiplication performance analysis (C, ongoing)**: analyzing different techniques to improve matrix multiplication performance. Focus on high-performance computing
**Diophantine Equation solver (Java):** focus on coding and number theory

## EDUCATION

**Bachelor of Physics, with Honors | University of Bologna | 2023**

## TECHNICAL SKILLS

**AI/ML:** Supervised learning, Unsupervised learning (clustering), Anomaly detection, Neural network architectures, Deep learning fundamentals, Pattern recognition
**Systems & Performance:** Memory optimization, Vectorization (SIMD), Parallelization, Performance benchmarking, Low-level programming
**DevOps & Administration:** Linux administration, End-to-end CI/CD workflows, Multi-node containerized environments (Docker Compose), Legacy build systems management

## TOOLS

**Programming Languages:** C++, C, Python, Java (8-21)
**System performance**: valgrind, perf
**AI/ML & Data Science:** PyTorch, Scikit-learn, Numpy, Pandas, Jupyter Notebook
**Development & DevOps:** Linux, Windows, Bash, PowerShell, Docker, Docker Compose, Make, Poetry, Maven, Jenkins, Git/Github/Bitbucket
**Databases & Servers:** PostgreSQL, H2, MongoDB, OpenLiberty, TomEE
**Frameworks**: FastAPI, Flask, Quarkus, Spring