JULIAN PADRON

julian.padronnunez03@gmail.com • linkedin.com/in/julian-padron-72669227a • juliannp253.github.io/portafolio-web

PROFESSIONAL SUMMARY

Computer Science Engineer specialized in backend development (Java 17 / Spring Boot 3). Hands-on experience building decoupled architectures, agile proofs of concept (PoC), and automating cloud deployments (AWS/Docker). Strong focus on solving business problems by integrating Artificial Intelligence agents (Python/LangChain), mitigating hallucinations through hybrid algorithms. I rigorously apply SOLID principles, Clean Code and security approaches in each project.

EDUCATION

The University of Texas Rio Grande Valley (UTRGV) — B.S. Computer Science | GPA: 3.40/4.0 • Dean's List 2023–2025
January 2022 – December 2025 | Relevant Coursework: Software Engineering · Algorithms & Data Structures · Databases · AI-Powered Apps

TECHNICAL SKILLS

Languages: Java (17/21), Python, C#, SQL (PostgreSQL/MySQL), JavaScript, HTML/CSS
Frameworks & Architecture: Spring Boot 3, Spring Security 6, Spring Data JPA/Hibernate, ASP.NET Core, React 19, REST APIs, OpenAPI/Swagger, SOLID, Clean Code, POO
Cloud & DevOps: AWS (EC2, IAM), Docker, RabbitMQ, GitHub Actions (CI/CD), Azure DevOps, Git, Linux
Databases: PostgreSQL, MySQL, Redis, MongoDB, Supabase
Testing & Agile: JUnit 5, Mockito, TDD, BDD (Cucumber) • Scrum, XP, Pair Programming
AI & Agents: LLM Orchestration, Tool Calling, OpenAI API, Google Gemini API
Languages (Spoken): Spanish (Native) • English (Professional Proficiency)

TECHNICAL PROJECTS

Distributed Banking Core | *Java 21 · Spring Boot · Redis · PostgreSQL · Docker | (github.com/juliannp253/money-flow)*

- **Microservices Architecture**: Engineered a distributed financial engine behind an API Gateway, implementing a Pub/Sub pattern with RabbitMQ for asynchronous communication and high domain decoupling.
- **Performance & Scalability**: Implemented a Cache-Aside pattern with Redis and polymorphic serialization, mitigating database bottlenecks (PostgreSQL) and ensuring eventual consistency through event-driven flows.

WatchWise - AI-Powered Movie Recommendation Engine | *Java 17 · Spring Boot 3 · Python · OpenAI API · MongoDB · AWS EC2 · Docker | (github.com/juliannp253/Movies-Recommender)*

- **Hybrid Algorithm**: I designed an engine that mitigates AI hallucinations by combining Collaborative Filtering algorithms (Python) with a curating LLM (OpenAI) for data precision..
- **Batch Processing**: I automated data ingestion using Cron Jobs in Spring Boot that execute Python scripts on a scheduled and recurring basis.
- **Efficient Deployment**: Containerized the complete solution in Docker (Multi-stage) and automated the CI/CD pipeline to AWS EC2 using GitHub Actions.

EduLink — Educational Gamification PoC | *Java 17 · Spring Boot 3 · Spring Security 6 · PostgreSQL · Swagger*

- **Backend Architecture**: Built from scratch prioritizing security using Spring Security 6 (centralizing validation on the server) and interactive documentation with OpenAPI/Swagger.
- **Domain Modeling**: Application of OOP (inheritance and polymorphism) to support multiple types of challenges and business logic.

Nova — Local AI Agent Assistant | *Python · LangChain · Google Gemini API · Tool Calling*

- **Secure Agent (HITL)**: Implemented a Human-in-the-Loop flow that schedules actions, requiring strict manual authorization before executing destructive commands in PowerShell.
- **Software Design**: I structured the system in modular layers (Brain, Executor, Memory, Tools) applying SOLID principles to decouple the LLM reasoning from local tools.
- **API Integration**: Orchestrated Tool Calling using the Google Gemini API and LangChain, persisting the conversational context locally in JSON

CERTIFICATIONS

SOLID Principles & Clean Code — Fernando Herrera / DevTalles | April 2026 | 6.5 hrs
Azure DevOps — Alberto Picazo | April 2026 | 10 hrs