

Manuel Magana

Daly City, CA | manuel.ramon.magana@gmail.com | (562) 360-7054 | [GitHub](#) | [LinkedIn](#) | [Portfolio](#)

EDUCATION

Northeastern University | *M.S. Computer Science*

May 2026

Relevant Courses: Algorithms, Distributed Systems, Computer Networking, Software Engineering, Machine Learning, Deep Learning

University of California, Berkeley | *B.A. Molecular and Cell Biology: Neurobiology*

May 2021

Relevant Courses: Data Structures, Machine Structure, Principles & Techniques of Data Science

TECHNICAL SKILLS

Languages: Python, Java, Go, C, TypeScript, R, SQL, Bash

Frameworks, Libraries, and Tools: FastAPI, SpringBoot, React, Next.js, Three.js, pytest, JUnit, Cucumber, Vitest, AWS, Git, Redis, MongoDB, DuckDB, pandas, PySpark, scikit-learn, PyTorch, TensorFlow/Keras, Docker, Gradle, GitHub Actions, CI/CD

Laboratory: HPLC, LC-MS, GC-MS, QQQ, RapidFire 400, PCR

EXPERIENCE

Genentech

San Francisco, CA

Bioinformatics Scientist

Nov 2022 – Jul 2025

- Designed a data validation library in R, then collaborated with 2 computational scientists to port it to Python with unit tests and PR reviews; wrote scripts consuming input files via the library and containerized them with Docker for an internal workflow system used by 20+ scientists.
- Used PySpark to process a 200GB+ mass spectrometry dataset that exceeded the team's R and pandas tooling; converted to Parquet for faster reads, enabling protein filtering and summarization at scale.
- Implemented features in an R Shiny dashboard with one other scientist, extending support beyond mass spec to additional data types so 4 team scientists could reliably repeat the same analysis across different datasets.
- Built automated benchmarking workflows across 4 proteomics platforms (OLINK, ELLA, ELISA, Mass Spectrometry), accelerating method selection from weeks to days.

Weill Cornell Medicine

New York, NY

Research Technician

Jan 2022 – Jul 2022

- Optimized proteomics data pipelines, increasing peptide identification rates and expanding usable experimental data for downstream analysis.
- Set up and managed compute environments (VMs, WSL, Anaconda) for proteomics software, unblocking postdoctoral researchers from IT bottlenecks.
- Developed a Python tool to extract peptide n-mers from FASTA files, automating a manual sequence analysis step used across multiple experiments.

Endless West

San Francisco, CA

Junior Data Scientist

Mar 2021 – Nov 2021

- Built ML pipelines (Logistic Regression, KNN, XGBoost, Random Forest) for sensory attribute prediction from chromatography data, enabling R&D to prioritize synthesis targets by predicted flavor profile instead of trial and error.
- Developed Python web scrapers using BeautifulSoup to extract and structure chemical property data from public databases, replacing hours of manual lookup per compound with a searchable local dataset.

PROJECTS: SOFTWARE ENGINEERING

PayPath | *Go, Next.js, React, MongoDB, Docker, OpenAI API* | [Live Demo](#) | [GitHub](#)

Jun 2026 – Present

- Built and deployed a full-stack personal finance dashboard: a layered, feature-folder Go REST API (34 endpoints, JWT + bcrypt auth) over MongoDB Atlas, with a Next.js/React frontend tracking income, expenses, debts, and net worth.
- Engineered a reporting service computing federal, state, and FICA taxes, debt payoff timelines, and 90-day cash-flow projections that honor per-occurrence calendar edits (moved bills, one-time purchases, paycheck amount overrides).
- Cut redundant database load with an in-memory TTL cache and singleflight read-collapsing over repository interfaces; added per-page bundle endpoints serving each dashboard view in a single round trip.
- Integrated 5 OpenAI-backed insight reports (financial health score, debt payoff strategy, savings plan, expense audit, income boost) grounded in the user's data; shipped the stack via Docker Compose and Vercel.

DADA-85 | *TypeScript, React, Three.js, GLSL, Vitest* | [Live Demo](#) | [GitHub](#)

Jul 2026

- Built an interactive 3D web app rendering a live DOM calculator inside a WebGL room: dual WebGL/CSS3D scenes share one camera, with invisible depth-writing quads solving mesh-over-DOM occlusion and custom GLSL flame shaders.
- Designed a config-driven engine where basic, scientific, and programmer modes (live HEX/DEC/BIN conversion, bitwise ops) are pure configs over one generic core; verified with 84 Vitest + React Testing Library tests.

Trak | *Java, Swing, DuckDB, Redis, MongoDB, Gradle* | [GitHub](#)

May 2026 – Present

- Built a full-stack sprint planning tool in Java with a desktop GUI (Swing), CLI (22 commands), and REST API (21 endpoints), all sharing a common service layer that switches between local and HTTP backends at startup.

- Implemented 5 interchangeable database backends (DuckDB, Redis, MongoDB, Parquet, JSON) behind a generic DAO interface (25 implementations), enabling runtime storage selection via factory without changing application code.
- Added Google OAuth sign-in, email-based password recovery (Jakarta Mail), a first-run setup wizard, and shipped v1.0.0 as a native macOS installer via jpackage.
- Wrote 254 automated tests: 101 Cucumber BDD scenarios across 12 feature files and 153 JUnit tests covering all 5 database backends, authentication flows, and API endpoints.

WhatToDo | *Python, FastAPI, Claude API, Docker, AWS* | [Live Demo](#) | [GitHub](#)

Jan 2026 – May 2026

- Drafted and owned backend design for a 4-person team: a 3-model SQLAlchemy schema on AWS RDS with JWT auth and bcrypt password hashing.
- Built a dual-provider AI engine using an abstract base class for provider-agnostic discovery with automatic failover between LLM providers.
- Established CI/CD pipeline with GitHub Actions and deployed containerized API to AWS (App Runner, ECR).

Physiological Signal Analytics Platform | *Python* | [API](#) | [Library](#)

Sept 2025 – Jan 2026

- Architected a two-layer system: a Python ETL/analysis library for batch processing and a production REST API for external and visualization-driven access.
- Built a FastAPI service over DuckDB exposing raw signals and derived features with offset pagination, composable query filters, and Pydantic validation.
- Implemented analysis endpoints for PCA, FFT, and spectral density supporting interactive exploration of physiological signals.

Rate Limiter Service | *Go, Redis, Docker* | [GitHub](#)

Jan 2026

- Built a high-performance Go reverse proxy implementing fixed window and token bucket rate limiting with configurable per-user quotas.
- Used Redis transactions for atomic token consumption to prevent race conditions; validated under 1,000+ concurrent requests.
- Built CSV-driven load testing harness validating throttle accuracy across burst, steady-state, and multi-user traffic patterns.

Paxos Key-Value Store | *Java, RPC* | [GitHub](#)

Apr 2025

- Implemented a fault-tolerant distributed KV store using Paxos consensus with Proposer, Acceptor, and Learner roles over RPC.
- Designed RPC-based node discovery and membership protocol enabling dynamic cluster joins with reliable state propagation.
- Validated under chaos testing: 10% acceptor failure rate, leader crashes, and network partitions while maintaining consensus via majority quorum.

CPU Scheduler Simulator | *C* | [GitHub](#)

Nov 2025

- Built process scheduler implementing MLFQ, Round Robin, and SJF with preemption, priority boosting, and I/O wait handling via custom min-heap queues.
- Validated with Valgrind (zero leaks); implemented verbose tracing for job state transitions and simulation statistics.

Staircase Concurrency Simulator | *C* | [GitHub](#)

Oct 2025

- Built multi-threaded simulator with semaphore-based synchronization; enforced direction constraints to prevent deadlock and starvation.
- Implemented RR and FIFO schedulers; validated correctness with unit tests and concurrency stress tests.

PROJECTS: MACHINE LEARNING

Cell Type Classification from scRNA-seq | *Python, PyTorch, NumPy, Pandas* | [GitHub](#)

Jan 2026 – Present

- Benchmarked 3 deep learning architectures (1D CNN, TOSICA-inspired Transformer, GraphSAGE GNN) for cell-type annotation on 76K-cell Allen Brain Atlas scRNA-seq data across 17 classes, achieving 92.65% balanced accuracy and 0.867 macro F1.
- Engineered a sparse masked embedding layer (98.7% sparsity) that projects 8,822 genes into 300 Reactome pathway tokens, enabling biologically interpretable attention scores that rank pathway importance per cell type.
- Designed a leakage-aware preprocessing pipeline with stratified splitting before normalization, architecture-specific feature selection (200 HVGs vs. 8,822 pathway-mapped genes), and inverse-frequency class weighting with label smoothing.
- Built a cross-platform transfer pipeline to SMART-seq data (47K cells) with confidence-based unknown cell detection; identified and diagnosed a critical class-collapse failure mode driven by pathological loss weighting at extreme imbalance ratios.

Deep Learning Coursework (CS7150) | *Python, PyTorch, TensorFlow*

Mar 2026 – Apr 2026

- Built U-Net from scratch for brain tumor segmentation on BraTS 2020 multi-modal MRI data (T1, T1ce, T2, FLAIR); evaluated with Dice coefficient and IoU across binary and multi-class segmentation tasks.
- Applied transfer learning (VGG16, InceptionV3) to image classification and trained 1D CNNs on DNA sequences for biological motif detection.
- Implemented generative models from scratch: GAN, DDPM diffusion, Flow Matching (RK4 ODE solver), RealNVP normalizing flows, and convolutional diffusion on MNIST.
- Built RNN, multi-layer RNN, Transformer (self-attention, causal masking, positional encoding), and VAE (reparameterization trick, KL divergence) architectures for text generation and latent space modeling.

Emotion Recognition from Physiological Signals | *Python, TensorFlow/Keras, scikit-learn*

Sept 2025 – Dec 2025

- Designed a Conv1D-BiLSTM hybrid classifier in TensorFlow/Keras for binary emotion recognition (Neutral vs. Threat) from 6 biosignals (ECG, EDA, blood pressure, respiration, temperature).
- Engineered a signal preprocessing pipeline with 5,000-sample sliding-window segmentation, Gaussian smoothing, and per-channel standardization; integrated raw data across two POPANE studies (25+ GB, 7 biosignal modalities) into a unified training corpus.
- Diagnosed a class-collapse failure mode under 85/15 Neutral/Threat imbalance (Threat recall dropped to 17% despite 87% overall accuracy); applied inverse-frequency class weighting and architecture search (LSTM units 64–750) to improve minority-class F1 from 0.04 to 0.39.
- Implemented INT8 quantization of input signals in a PyTorch LSTM baseline, reducing model input precision from float64 to int8 while preserving 91% classification accuracy, targeting TinyML deployment on wearable microcontrollers.

PUBLICATIONS

MSstats+ enhances differential analysis in proteomic experiments | *bioRxiv* 2025

- Co-author. [doi:10.1101/2025.09.11.675573](https://doi.org/10.1101/2025.09.11.675573)