

Fernando Martínez

Chile • rfernandomartinez@gmail.com • 9 9360 0749 • fernandomartinez.dev
github.com/rfernandomartinez • linkedin.com/in/fernando-martinez

Profile

Software engineer specialized in backend systems, databases, and distributed systems. Passionate about scalability and systems that combine consistency and performance. Recognized for autonomy, clear communication, and technical rigor within multidisciplinary teams.

Experience

Fullstack Developer

Chile

Universidad Autónoma de Chile

June 2024 – present

- Implemented Java/Spring Boot microservices and React/Next.js front-ends integrated with Oracle, supporting hundreds of daily users across academic, budgeting, and analytics processes.
- Optimized SQL queries (stored procedures, views, OLAP analytics) and improved performance of reporting pipelines and critical services.
- Designed and implemented a real-time notification subsystem based on Apache Kafka (topics, partitions, consumer groups, delivery semantics) for high concurrency with idempotent event processing, WebSocket gateway, and Next.js/React client integration.
- Designed the relational model for notifications (storage, per-recipient state, delivery lifecycle) and history endpoints.
- Implemented end-to-end observability using OpenTelemetry (tracing, metrics, logs), Grafana dashboards, and Prometheus.
- Introduced modern DevOps practices (GitHub Actions, IaC, Terraform) to accelerate deployments and standardize environments.

Mobile Developer

Chile

Gravat Chile

July 2023 – June 2024

- Lead developer for a mobile application for license plate management, recognition, administration, and engraving.
- Implemented persistent local storage using Isar/Hive and SharedPreferences.
- Built advanced networking using Dio and HTTP.
- Managed state with Riverpod (StreamProvider, NotifierProvider) and dependency injection patterns.
- Implemented background processing and concurrency with isolates.
- Delivered native integrations via Method Channels (camera and device capabilities).
- Developed custom UI using CustomPainter.
- Implemented feature flags (feature toggles) for controlled rollout.
- Built distribution pipeline using GitHub Actions and Firebase App Distribution.

Teaching Assistant – Data Engineering

Chile

Universidad de La Frontera

Aug 2022 – Dec 2023

- Data analysis with R and implementation of ML/DL and data mining models using Python, scikit-learn, and TensorFlow.

Education

Universidad de La Frontera

Chile

Bachelor in Computer Engineering

Mar 2019 – Mar 2024

Projects

Metadata Store over FoundationDB + MinIO

2025

ACID metadata layer (KV-based) for object storage in Java using FoundationDB as a transactional control plane and MinIO as the blob backend.

- Modeled hierarchical metadata (buckets, objects, versions) in FoundationDB's ordered key-space to enable efficient range reads and listings.
- Integrated MinIO for blob storage while keeping strongly consistent catalog metadata in FoundationDB.
- Used FoundationDB transactions for atomic updates and maintenance of secondary indexes.
- Architecture inspired by the “composable blob store” pattern; documented in a technical article.
- Github [Metadata Store over FoundationDB + MinIO](#)
- Blog [FDB + MinIO Metadata \(Java\)](#)

KV Store Log-Structured (WAL + MemTable)

2025

Transactional write-ahead log and crash-safe recovery in C++ for an LSM-style key-value store.

- Implemented a custom binary WAL format with CRC32 integrity checks and configurable durability policies.
- Built WAL with POSIX I/O (open, write, fsync) including magic header, op type, lengths, and serialized payload.
- Supported durability policies FSYNC ALWAYS, FSYNC EVERY N, and FSYNC NEVER.
- Prepared foundational architecture for SSTable flush and compaction (WAL + mutable MemTable).
- Github [Metadata Store over FoundationDB + MinIO](#)

Blogs

Exploration of FoundationDB Internal Architecture (DD, SS, TLogs)

2025

Research series analyzing FoundationDB internals focused on data distribution, scheduling, replication pipeline, bulk load path, and deterministic actor-based transitions.

- Reverse-engineered Data Distributor workflow (team formation, shard assignment, rebalance heuristics, failure reactions, Cluster Controller interactions).
- Analyzed Storage Server internals (shard acquisition, Redwood/RocksDB behaviors, mutation application, MVCC maintenance, durability and I/O patterns).
- Studied Bulk Load mechanics across Proxy/TLog/SS and recovery-time implications.
- Broke down the deterministic simulation framework (fault injection, time control, message reordering, randomized state exploration).
- Produced diagrams/models of actor interactions (Proxies, TLogs, DD, SS) to explain transaction lifecycle and consistency.
- Link [Inside FoundationDB Bulk Load From System Keys to SST Ingestion in Storage Servers](#)