

JOEL TEODORO GÓMEZ

Backend Software Engineer · Go

Barcelona, Spain · joel.teodoro.software@gmail.com · (+34) 640 31 80 80
linkedin.com/in/joel-teodoro-gomez · github.com/JoelTeoGom · runtimerants.dev

Backend engineer with over 2 years building production Go services. I own two booking platforms end to end (Andorra.com and CityXerpa): multi provider integrations, payments, and concurrent reservation flows on Kubernetes. I build small systems in my own time to understand specific failure modes, and I write about it at runtimerants.dev.

SKILLS

Go & Backend	Go (Golang), gRPC, GraphQL, REST, Protobuf, WebSockets, concurrency (goroutines, channels, sync primitives, worker pools)
Data	MySQL, PostgreSQL, Redis (cache, Pub/Sub, distributed locks, Lua scripting), Elasticsearch, Cassandra
Messaging	Kafka, RabbitMQ, Redis Pub/Sub
Infrastructure	Docker, Kubernetes (GKE), GCP (Cloud SQL, Memorystore, Cloud Build, Cloud Logging), GitLab CI/CD, Git, Linux
Architecture	Microservices, Hexagonal Architecture, Domain Driven Design, Event Driven, System Design, TDD, Clean Code
Other languages	Java (Spring Boot), Python, C, JavaScript

EXPERIENCE

Dribba — Backend Software Engineer (Go)

Oct 2024 – Present · Barcelona, Spain · Clients: Andorra.com, CityXerpa

- Own two production microservices end to end, from API research and architecture decisions to database schema design, implementation and deployment. Work directly under the CTO with full ownership from architecture to production.
- Built the complete hotel booking domain for Andorra.com from scratch: multi provider API integration, data normalization across heterogeneous providers, availability search, reservations, payments through Redsys, cancellations, refunds and invoicing.
- Built the bus reservation system for CityXerpa on top of the Movelia API: route search, seat allocation, fare calculation and ticket management, exposed through GraphQL resolvers.
- Designed and built a federated multi vertical search with Elasticsearch, covering index modelling, shard sizing, analyzers and query design, so a single text query returns results across every product vertical.
- Designed the internal event bus for CityXerpa: unified fan out over a single WebSocket subscription, with rate limiting so high frequency clients such as the operations dashboard are not flooded.
- Resolved a critical database bottleneck that was forcing an oversized instance to absorb one recurring query spike: analyzed execution plans, denormalized tables, added composite indexes on real access patterns and segmented heavy queries with CTEs. Also ran a zero downtime column migration on a 1.6M row table with batched backfill to avoid locking under load.
- Applied resilience and concurrency patterns in production: singleflight to prevent cache stampedes, circuit breakers around an unstable third party API, Redis based locking for concurrent bookings, worker pools, graceful shutdowns and idempotent operations. Deployed on Kubernetes (GKE) with Docker, GitLab CI/CD and GCP.
- Drove the migration from monolithic GraphQL resolvers to a modular architecture split by domain, with hexagonal boundaries and provider integrations extracted into adapters behind interfaces, migrated incrementally with the system live.

PROJECTS

realtime-odds-platform — Real time odds fan out · Go, WebSockets, RabbitMQ, Redis, Hexagonal Architecture

A system design problem from a sportsbook interview, built out. Provider updates are partitioned across queues by hash(eventId) so one event stays ordered while different events process in parallel, then fanned out over WebSocket through a hub sharded by the same key to split lock contention. One write pump per connection and a non blocking send that drops slow consumers rather than stalling the broadcast. github.com/JoelTeoGom/realtime-odds-platform

CrispLite — Real time 1:1 chat · Go, WebSockets, Redis Pub/Sub, PostgreSQL, Docker

Chat system built from scratch to find out what breaks in real time messaging. Redis Pub/Sub for cross instance delivery, per user mailbox pattern, a global batcher goroutine doing bulk inserts with a ticker plus buffer size flush, presence via Redis keys with TTL renewed by heartbeats, and hexagonal architecture throughout. github.com/JoelTeoGom/CrispLite

DDD-Ecommerce — Domain Driven Design in Go · Go, PostgreSQL, pgx, Chi, Docker

E commerce backend applying DDD tactical patterns: bounded contexts (Order, Catalog, Shared Kernel), aggregate roots enforcing invariants inside the aggregate instead of in a service, value objects (Money, Address, OrderStatus), and domain events over a decoupled bus. Zero infrastructure imports in the domain layer. github.com/JoelTeoGom/DDD-Ecommerce

Also on GitHub: HTTP server written from scratch over raw TCP, distributed rate limiter with a Redis atomic Lua script, priority scheduler with min heap and sync.Cond, plus runnable Go concurrency patterns (errgroup, circuit breaker, singleflight, Snowflake IDs).

EDUCATION

Software Engineering Degree — Universitat Rovira i Virgili (URV), Tarragona

Sep 2020 – Jun 2024

Strong foundations in computer architecture, operating systems, networking and distributed systems. Built a compiler from scratch (lexing, parsing, AST, code generation), an OS process scheduler in assembly handling context switching and ready queues, CPU components at the gate level, and an IoT platform for high volume real time sensor data.

LANGUAGES

Spanish (Native) · Catalan (Native) · English (Professional, B2)