Saul Martínez

Backend Developer
http://coderoso.io/
https://linkedin.com/in/samacs
https://github.com/samacs

Email: saul.martinez05@gmail.com
Phone: +52 (558) 340 1360
Location: México City, CDT, UTC -5

Hi,
I'm a passionate developer from Mexico City. Over the last few years, I've been helping people grow their business building different applications.

I've been involved in building e-commerce solutions, from the very first line of a POS system to an online store in the cloud.

Here I have selected some of what I consider my most relevant work over the years with a more in-depth explanation about the process.

Ösom Go Backend Developer

May 2017 → March 2019 - *1 year 10 months*
Technologies: Go, Ruby, Elixir, Kubernetes, RabbitMQ, GitLab CI

Ösom is one of the biggest online clothing retailers in México.

Amazon's MWS integration

By 2017 Amazon started offering fulfillment services in México, so the company took advantage of this services and moved all the inventory to Amazon's warehouse. We integrated their MWS (Marketplace Web Service) into our back-office system. This included stock/inventory management, order fulfillment, product management, sales orders creation, etc.

We created a series of services in Go, Ruby and PHP for consuming their API. These services poll information from Amazon periodically to update our back-office system and NetSuite WMS.

I was in charge of the Ruby on Go services development. We started building services with pure Ruby and consuming Amazon’s workflows with background jobs running on Sidekiq. After a couple of months, we decided to move these services to a more performant environment, so we switched to Go. We created a Kubernetes cluster to house all the needed services like PostgreSQL, Redis, RabbitMQ, etc. and automate the deployments. For inter-service communication we used gRPC. After evaluating some microservices frameworks, we decided to go with http://micro.mu. This really sped up the development process and we were sure that there were no semantic problems when communicating with other services. Some services needed to expose some endpoints to other systems, so we put a load balancer and API gateway around these endpoints using Kubernetes and Micro API Gateway.

Saul Martínez

Backend Developer
http://coderoso.io/
https://linkedin.com/in/samacs
https://github.com/samacs

Email: saul.martinez05@gmail.com
Phone: +52 (558) 340 1360
Location: México City, CDT, UTC -5

This was the foundation for the other services we built on top to support multiple sales channels and warehouses.

The goal was that for the commercial department this integration was completely transparent, which we did.

Shopify frontend

Later on, we decided to get rid of our frontend system and use Shopify as our customer facing interface.

Shopify has already a well-structured sharding strategy for supporting millions of requests, so we decided that it was our next move to let Shopify to serve the customers.

For this we created some services between Shopify and our backend systems. Although Shopify became our customer facing system, this wasn't the only channel we offered our products so at the end, all the data needed to converge in a single source of true: NetSuite's WMS.

Having the foundation of services for communicating with Amazon, we started to implement additional services in Go for communicating with Shopify and our Warehouse Management System. This involved creating interfaces for consuming and updating the sales´ channels APis. For this we used the standard library's net/http client and it just worked amazing.

Dub5 Ruby, Go and Objective-C developer

July 2015 → March 2017 - *1 year and 9 months*

Technologies: Go, Ruby, Elixir, Objective-C

Dub5 was a Canadian start up.

At Dub5 we were building two products: Chopeo and the Dub5 POS.

Dub5: An intelligent and gamified POS

Saul Martínez

Backend Developer
http://coderoso.io/
https://linkedin.com/in/samacs
https://github.com/samacs

Email: saul.martinez05@gmail.com
Phone: +52 (558) 340 1360
Location: México City, CDT, UTC -5

I was part of the team that created the initial version of the POS. The application was offline-first so there was no need for internet access for the application to work. Once the app was online, a synchronization was made with the backend API.

The backend database was organized using multitenancy with different schemas per client.

Sooner than later, we needed to process a big number of logs so we implemented a logs watcher in Go that would process the Rails logs and send them to our backend. The payload was structured in such a way that this gave us a clear idea of which client generate the request and the information needed to resolve any critical issue. Our idea was to extend this tool to provide Email and Slack notifications but that didn't happen in the project's lifetime.

We also implemented the protocols for communicating with different printers and barcode scanners.

Apart from the application itself, we also built an administration interface for clients, where they could manage users (cashiers) and the performance of each teller individually as well as the overall performance.

Hello Code Fullstack developer

June 2014 →April 2015 - *1 year and 9 months*
Technologies: Ruby on Rails, Angular 1, C#

Hello Code was a startup based in Hermosillo, Sonora that was created by a couple of friends who invited me to work in special projects.

The idea behind Hello Code was to build Chopeo. This product was then acquired by *Dub5*.

Special projects

At Hello Code while we all were building Chopeo, I was also in charge of special projects for keeping the company running.

A barbershop

Saul Martínez

Backend Developer
http://coderoso.io/
https://linkedin.com/in/samacs
https://github.com/samacs

Email: saul.martinez05@gmail.com
Phone: +52 (558) 340 1360
Location: México City, CDT, UTC -5

We built a system for managing barbers' schedules. The barber shop manager assigned the free spots for each barber depending the barbers' skill, availability and fee.

The first iteration's goal was to build a clean, simple and usable application optimized for the iPad. Later iterations included a customer facing section to automatically reserve a spot for a particular barber, but this didn't complete.

Toven: Your restaurant. Live.

Hello Code was hired by Toven to build their product.

Toven has been in the POS and restaurant support business for a long time. They wanted to build an application for their clients to have clear view of their business at all times.

The application was built in multiple parts:

- A service living in the POS to retrieve information periodically.
- A company administration area to build and assign different SQL queries for the service to run depending on the POS brand and version.
- A customer administration area for the owners to create branches, manage royalty fees and users.
- A subscription system for the customers to sign up. We used Conekta.
- Finally, a mobile application for the managers and branch owners to see the retrieved information in a meaningful way.

We built the Windows service using C# and took advantage of Microsoft Installation Server in order to automatically update the clients. The service retrieved the SQL queries assigned to the POS version and brand running locally. This payload included the frequency these queries needed to be run at to avoid performance issues during peak hours.

No matter what POS system was installed (Pacific Soft, Soft Restaurant, Micros, etc.), the retrieved information was presented in the mobile app (iOS and Android) uniformly.

Saul Martínez

Backend Developer
http://coderoso.io/
https://linkedin.com/in/samacs
https://github.com/samacs

Email: saul.martinez05@gmail.com
Phone: +52 (558) 340 1360
Location: México City, CDT, UTC -5

This gave the restaurant owners a complete status of the branch's performance at any time: traffic, cancelled tickets, refunds, etc.

CENTEL Systems administrator

April 2013 → June 2014 - *1 year 2 months*

- In-house application development.
- Basic CISCO switching configuration.
- Network performance improvement
- Build a VoIP recorder in C#.

Periódico Expreso PHP developer

June 2010 → September 2012 - *1 year 10 months*

- Implementation of the first newspaper version using Joomla. We switched from a custom developed system that was not intended to cope with the amount of content over the years.
- Creation of the first online printed-edition section using vanilla JavaScript.

CESAVE C# developer

August 2008 → June 2010 - *1 year 10 months*

I worked as a freelancer to build an application for issuing planting permits.

CESAVE (Comité Estatal de Sanidad Vegetal) started building a series of applications to better monitoring sowing fields across the territory. The idea took relevance among other states of México and I was hired to build some of these services at a major scale.

PUS: Permiso Único de Siembra

My project was an offline application that eventually synchronized the local information with different offices across the country. This was so to avoid one producer from issuing multiple

Saul Martínez
Backend Developer
http://coderoso.io/
https://linkedin.com/in/samacs
https://github.com/samacs

Email: saul.martinez05@gmail.com
Phone: +52 (558) 340 1360
Location: México City, CDT, UTC -5

permits for planting under the same sowing field, which is then used to get access to monetary resources from the government.

We first consolidated multiple databases (Visual Fox Pro databases) into a master and a couple of slaves MSSQL databases. Then we distributed a desktop application for the local offices to use the system now with a synchronized database.

Some rural offices had no internet access at all times, but the process of issuing permits could not stop. That's why we used Microsoft Synchronization Services in order to eventually sync all the information from multiple sources.