Hector Avalos

Age: 26.

Phone: +52 (333) 733 6504 or +52 (333) 970 9448

Email: hg.avalosc97@gmail.com

Full Stack Software Engineer @ EPAM Systems Mexico S de RL de CV.
July-2023 – Currently.

Working on a project for an EPAM client. In this project I've been using technologies like ReactJS, Storybook, Material UI, Apollo GraphQL, AWS DynamoDB, Docker and docker-compose and tools like Figma, Atlassian Jira and Atlassian Confluence, all this implemented in a serverless architecture. The rendering of the project is dynamic which requires us developers to write "generic" components that depend on the JSON schema provided to the app.

Full Stack Software Engineer @ Gold Media Tech LLC.
July-2020 – Mar-2023.

Working for a Medicare Advisors company. I've written code for both Front and Back End using technologies such as VueJS (NuxtJS/Typescript) and NestJS (Typescript). I have used technologies like Swagger (OpenAPI), Redis, Aptible, Netlify, PostgreSQL, Sentry.io, Airtable, Salesforce (Process Builder, SOQL queries, integration with NodeJS). Initially we started working on a tool for the consumers (elderly american people) and this tool had different modules: The modules for drugs, the module for providers (Doctors), the module for Hospitals, etc. And based on these preferences, the API fetched a medicare plans list from an external API and once that was done, we implemented the Strategy Pattern to determine what sorting algorithm to use (Sort by plan cost, by drugs cost, by hospital fees cost or a combination of them). I was the person in charge of creating the strategy manager and each one of the strategies (sorting algorithms). Once the company grew, the engineering team was divided into two teams: the consumer-team and the agent-team. I was part of the consumer team, that means I used to work a lot to maintain the codebase of the consumer tool described previously, but there were times where there was almost nothing to do in the consumer team, so basically my team lead lent me to the agent team. The agent tool was designed for the agents (people who visit elderly people and enroll them in medicare plans), this tool was quite similar to the consumer one, but this had a few more modules. There was a module designed to allow agents to upload proof of the enrollment (screenshots, PDF files, photos). There was a module for managers to give permissions (create enrollments, delete enrollments, delete data from a customer, delete a customer from the DB, etc...) to each one of the ISRs that reported to them. There were times where I worked with the marketing team too, helping them install tracking scripts in our webpage (using Google Tag Manager). Additionally I developed some cron jobs in the backend to send physical mail (there is a company that provides an API that allows you send physical letters, postcards, etc...). We used this tool to send birthday postcards or very important letters to the customers. I worked a little bit with Salesforce too. We used their API to Create, Read, Update or Delete customers from Salesforce (Opportunities, accounts, addresses, etc...) so I learned a lot about the Salesforce API (Authentication, getting data from objects, etc.), SOQL queries, Salesforce Apex code and the Salesforce Process Builder.

Software Developer @ Vectralis S.A de C.V.

Aug-2017 – July-2020.

R&D Software Developer. I wrote new code for their internal management system, which was initially written in Java, then we had to migrate it to PHP because the company started to grow a lot and the Java code had a lot of tech-debt. After that we decided it was time to upgrade the system so we "translated" again the code to ReactJs for the Front End and Django / Django-REST-Framework for the Back End) as well as deploying new changes to the local server (Using tools like docker/docker-compose, nginx and Gunicorn). Once, one of the automation machines had an issue, if one of the components placed by a robotic arm was inverted, the feeder of these components was very likely to get stuck. To solve this problem the team decided to let me play a little bit with computer vision projects (Python + OpenCV), basically this script detected if the component was inverted, and if so, sent a signal to the PLC via a TCP/IP server (PLC = Server, Python Script = Client), for this purpose I had to apply some masks to the image coming from the webcam to compare with preset values (when the component is placed correctly), adding a certain tolerance to avoid false positives.

Education:

Tecnologo en Informatica y Computacion.

Centro de Enseñanza Técnica Industrial (July 2017)

Languages:

Spanish: Native.

English: Fluent.

Apps/Tools:

VueJS, NuxtJS, NestJS, TypeScript, Git, Github, Docker, docker-compose, Salesforce, Salesforce Apex, Redis, PostgreSQL, Salesforce, SentryIO, Netlify, Aptible, VI/VIM, XAMPP, VSCode, Android Studio, GCC/G++, GraphQL (Apollo and Graphene), ReactJS, Redux, Django, PyQt, OpenCV, OpenAPI, Nginx, Gunicorn.

Soft skills:

Communication, Teamwork, Leadership, Adaptability, Problem solving, Time management, Conflict resolution, Active listening, Critical thinking, Decision making, Collaboration, Self motivation.

Programming languages and frameworks/libraries:

- Python, JavaScript, NodeJS, TypeScript, MariaDB, PostgreSQL (6 years).
- VueJS, NestJS, NuxtJS, Django, Django REST Framework (4 years).
- ReactJS, Express, MongoDB (3 years).
- PHP, C/C++, Java, C#, Flask (2 years).

Courses and certifications:

- **May-2019 – Dec-2019: *Bootcamp Full Stack Python*** course @ BEDU.org GDL (150h).
- **Feb-2020 – Aug-2020: *Bootcamp Full Stack JavaScript*** course @ BEDU.org GDL (150h).

Links:

- https://github.com/havalos97
- https://www.codewars.com/users/havalos97
- https://cv-ha.vercel.app/