David Omar

Front-end Engineer

✉ david@davidomar.com   🏠 davidomar.com/   /davidomarf   /davidomarfch

Strongly focused on Development Experience, Web Performance, and Technical Writing.

Work Experience

GBM Grupo Bursátil Mexicano

Dec 2020 - Present

Front-end Engineer (Fulltime)

React · **AngularJS** · **Angular 2+** · **TypeScript** · **RxJS** · **WebSockets** · **Unit Testing**

- I maintain and develop two web applications for the largest stock broker in Mexico (>3.8M investment accounts). I work on new features, but I'm mostly involved in maintenance and improvement of the existing codebase.
- Led the migration of an 8-year-old AngularJS project from Gulp to Webpack.
  - Upgraded our tech stack (Vanilla JS, and CSS ⟶ ES2020, Typescript, and Sass).
  - Reduced non-cold start re-compilation time for development builds **from 20s to <1s**.
  - Simplified the build process by reducing the number of commands that had to be ”manually” run.
- Led the continuous upgrade of a core application (>120k weekly active users) from Angular v4 to v9.
  - Reduced initial bundle size, loading times, and crash rate, which **improved Apdex score from .6 to .85**.
  - Documented the core architecture of the application using Sphinx.
  - Created a project-specific roadmap to keep up with the upgrades up to Angular 14.
- Improved client-side performance on multiple front-end projects across the whole company by adding compression
  - Configured CachePolicies on AWS CloudFront to accept Brotli and Gzip encoding
  - Updated Webpack configurations to run compression on the generated bundles
  - Reduced the initial JS and CSS bundle size across all projects to **15% of the original size** (avg.)
- Participated in the candidate selection process, by directing technical interviews, reviewing take-home assignments, and mentoring new interviewers.

Umvel

Feb 2020 - Dec 2020

Front-end Developer

Angular · **Next.js (React)** · **TypeScript** · **Socket.IO** · **RxJS** · **Unit testing**

- Developed an internal library for utilities that were being re-implemented constantly in different projects, making code more maintainable, robust, and better documented.
- Wrote scripts to seed content and users, with different information in the database, making testing and development faster, preventing the repetition of steps that were up to 10 minutes long.

Relevant Projects

Ginpar

CLI static content generator for generative artists

Python · **Jinja2** · **Click**

- Convert P5.js scripts into interactive pages that let you control the script parameters in a GUI.
- Templating engine to generate the GUI using a user-defined list of parameters.
- Generate buttons for value randomization, sketch regeneration, and image download with seeding information.
- CLI commands to initialize projects and sketches; build projects, and start a live reloading server.

PyPi: pypi.org/project/ginpar · **Repository**: davidomarf/ginpar · **Docs**: ginpar.readthedocs.io ·

Attractor Seeder

Web tool to assist generative artists interested in rendering attractors

HTML · **CSS** · **Vanilla JS** · **P5.js**

- Mass-produce attractors to efficiently choose attractor building values
- Create multiple canvas elements that depend on the size of the screen and the URL parameters
- Each canvas can be regenerated without affecting the others

Homepage: attractors.davidomar.com · **Repository**: davidomarf/attractor-seeder

Other

Generative Art: A quick introduction to start producing algorithmic visual art *TECH TALK*

PRESENTED AT **EVENTLOOP: JAVASCRIPT MEETUP FROM MEXICO CITY** *04 Sep. 2019*

👤 150 attendees · 🕘 40 min talk + 20 min Q&A

Slides: davidomarf.github.io/talks/eventloop-19-08

Generative Mistakes *WRITING*

COLLECTION OF TEXTS AND PROJECTS TO EXPLORE GENERATIVE ART AND PROCEDURAL GENERATION *Mar. 2018 - Present*

Interactive sketches: play.generativemistakes.art · **Writings**: generativemistakes.art

- Algorithms consist mostly on the production of data that is later visualized in creative ways
- Made me get familiar with computational geometry concepts like triangulations, meshes, Voronoi diagrams, hulls, and interpolations
- Visualizations created with **P5.js** and **D3.js**