

Curriculum vitae

Personal data

Name: Jefferson S.

Technical Knowledge and Skills

CI/CD, Python, Linux, Javascript, React, CSS



Full stack developer (Python/Django and React) with experience in American companies. I managed to make an impact where I worked by always facilitating and automating processes, increasing revenue, and reducing expenses in these companies.

Professional with more than 13 years of experience in the IT area, I worked as a developer of applications, games, and systems for mobile and web devices. With the experience of working for a software factory, I was exposed to different technologies, such as Python (pandas, TensorFlow, Django, and FastAPI), Javascript (Node.js, React.js, Angular, Express), PHP, SQL, and NoSQL databases services, AWS services, Android, iOS, among others. Self-taught, passionate about innovation, and hardworking.

Education

January 2018 –
December 2018

Barcelona, Spain
Master's degree, Big Data and Business Analysis, ENEB -
European Business School of Barcelona

January 2017 –
September 2018

Barcelona, Spain
Master of Business Administration - MBA, Business
Administration and Management, General, ENEB - European
Business School of Barcelona



January 2011 –
December 2012

Brazil
Postgraduation, Systems Analysis, FUNCEFET

January 2006 –
December 2009

Brazil
Graduate, Information Technology - TI, Educational
Foundation of Duque de Caxias

Work experience

April 2022 -
Present

Software Development, United States
Back End Developer

Stack:

- Django REST Framework,
- TypeScript,
- Bootstrap,
- SASS,
- GitHub,
- Redux.js,
- Software Development,
- English,
- Docker,
- Amazon Web Services,
- Git,
- Banco de dados,
- SQL,
- HTML,
- React.js,
- Django.

Responsibilities:

- Vendor onboarding and management system.
- Delivered quality software, testing and reviewing other colleagues' code, and writing good documentation to help the decision-making process.
- 90% decrease in the number of manual bank verifications;
- 20% decrease in usage of external APIs, decreasing expenditure;
- These improvements enabled the company to get traction not limited by external costs and manual



work.

August 2021 -
April 2022

Investment Management Company, Brazil
Full Stack Python/React Developer

Stack:

- Django REST Framework,
- Amazon Web Services,
- React.js,
- Django,
- Python.

Responsibilities:

- A platform for high ticket courses marketing and selling.
- Being part of a small IT team, working on different parts of the software as frontend, backend, and infrastructure.
- Developed frame detection engine for RTMP, live cam streams, and alert engine to make real-time notifications.
- Improved in 80% the performance of the software;
- Fixed structural problems in the integration with different APIs; Integrated a WYSIWYG editor in the system in order to scale the company.

December 2011 -
August 2021

Software Development Company, Brazil, Rio de Janeiro
Full Stack Engineer (Remote)

Stack:

- Python,
- Pandas,
- Tensorflow,
- Django,
- Angular.js,
- Express.js,
- Node.js,
- jQuery,
- C3.js,
- D3.js,
- PHP,
- MongoDB,

- MySQL.

Responsibilities:

- Responsible for creating and maintaining different Brazilian and international projects.
- Implemented a system for Telespazio to integrate geoinformation and satellite data from different platforms;
- Created a project for Coca-cola Brasil, to gather, clean, organize, and visualize business data from different manufacturers in Brazil;
- Developed a Facebook game that ended up building a school in Haiti.

Portfolio



A Typical Supplier Onboarding Process
Often highly manual, time consuming, and abundant with opportunities for mistakes or fraud.

Collecting supplier info

- Payables Strategy
- Compliance + Security Checks
- Approvals + Audit Trails
- Existing Vendor Updates
- Payment Security

Before PaymentWorks

Piles of paper, unsecured electronic files and time-consuming manual look-ups rife with the potential for human error.

- Manual verification of Tax ID and addresses
- Sanctions lists checked at onboarding and with every PO
- Phone calls to vendors to attempt to verify banking information
- Waiting (and waiting) for missing documents and returned phone calls
- Constant worrying: what did I miss? What is outdated? Why are there so many duplicates?

PaymentWorks/Financing

April 2022 - April 2023

Website: <https://www.paymentworks.com>

Full Stack Python/React Engineer.

What was your role/duties in the project?

Part of a team of 5 engineers focused on the system's data verification. The main challenge of the team was to support operations teams while managing clients onboarding data and related processes. Apart



from Django/React tasks, created a few dashboards to help to understand the status of the system before, during, and after the applied changes.

What kind of goal or task were you supposed to complete?

The most important challenge found was a manual process that was queuing up in the system. An operational team was not able to keep up with the manual verifications and the main challenge for the team was to help to minimize this queue. It was preventing the company to scale up as there were a number of potential clients waiting to sign up for the system but they couldn't.

To improve this process, the most efficient proposed initiative was to recycle previous validations. To do that, it was necessary to unhash a few fields of a huge table and create a new hashed field containing concatenated bank info. It was necessary to run it for each validation in the table. It was performed with celery tasks and Amazon SQS.

What was your (and team) achievements/results in the project? Which problem did you solve?

After automatizing part of the process and searching for older verifications, the manual process was minimized and the 5000 items queue dropped to a 50 items queue. This way, the company was able to receive more clients and keep growing.

What was the technology you were working with? What kind of software did you use?

It was an 80% backend job, with Python/Django tasks, and a few FastApi tasks as well. Apart from that, React was used on the front end and a lot of SQL/NOSQL was used for data dashboard creation.

Describe the complexity, was it something new, was it refactoring something existing?

It was a challenging project as it involved refactoring old code and improving it to make the system ready for scaling up while supporting a reasonable number of users.

Technologies used:
Python, Django, SQL