Juan Ibiapina

juanibiapina@gmail.com
Berlin, Germany

About me

I have a **Computer Science** background and **14 years of experience** developing software professionally. I have worked in many different layers of software development, including **backend**, **frontend**, mobile and infrastructure, with focus on backend. I advocate **code quality** and **continuous delivery**. **Tests** are part of my development practices, along with **TDD** and refactoring. I'm confident working with **agile** practices that enable **fast delivery** in small increments. I also have experience designing distributed services, microservices and **REST APIs**. I'm a big fan of **tooling** and **observability**.

I like environments that support **growth**, working along with **smart**, **respectful** and **open-minded** people who care about their systems, tools and code.

I love to **code**! I'm a **NixOS**, **terminal** and **neovim** user. I often learn new programming languages, frameworks and technologies. I have worked professionally with Java, Javascript, **Typescript**, **Ruby** and **Go**, but have also experimented with SML, Python, Haskell, Scala, Rust, Racket, Bash, Clojure, IO, and many others. I designed my own toy language, a package manager for Bash and an organizer for command line scripts that I use as part of my 12 years old dotfiles. I try to keep at least one active coding side project. You can find more details about personal projects, passions and open source on my Github Profile and my blog.

I'm also a **game**, **music** and **climbing** enthusiast.

Experience

Principal Software Engineer at Babbel

August 2017 - present

As a Principal at Babbel, I work in the Content Platform team, developing the learning content CMS and related content distribution **APIs**. We design **cross-team architecture**, striving for **small deliverables** that bring value to users and allow teams to achieve their **goals** and work **independently**. This is mostly **remote work**, but we like to meet once per week.

Accomplishments:

- **Content Pipeline**: Designed an integration between Babbel's **CMS** and **Contentful**, enabling fast creation of content for new learning experiences. This system now holds content for more than 20 learning experiences, fulfilling the company goal of experimenting with new forms of learning. Teams can **autonomously** create new types through Pull Requests. All content is **Git** versioned and **immutable**.
- **Content Deployment**: Led a multi-year project to enable content editors to deploy content. Through a series of **modeling** changes, API improvements, **monitoring**, data migrations, **test** improvements, education and even a Slack bot, Babbel went from one painful content deployment per quarter to several uneventful content deployments per week.
- **Content Modeling**: Designed an extensible new data model for current and future Babbel content, inspired by **NixOS** and served through a **GraphQL** API. The new model is backwards compatible but also enables new use cases involving personalization and AI integration.
- **Content APIs**: Designed **RESTful** content APIs for current and future company initiatives, focusing on **performance** and **cacheability**.
- **Workshops**: Organized internal **presentations** and **workshops** about Babbel's content domain and architecture, **empowering** other teams and content creators.
- **API Gateway tooling**: Improved build times (**from 15m to 5m**) and tooling for Babbel's main API Gateway, shared between teams. Engineers can simply run `make` and all checks run locally using **Docker** with zero setup required. The same setup runs on **CI**.
- **Engineer-friendly Documentation**: Started a documentation initiative to create engineer-friendly documentation in repositories. Many teams have adopted this practice and maintain Git versioned documentation.

- **User vocabulary migration**: Migrated all user vocabulary from **MySql** to **DynamoDB**. It went from 7 unmaintainable, untested joins, to a performant and scalable single table design.
- **Service extraction**: Actively drove or participated in many service extractions from the **Rails** monolith, including user vocabulary, content, authorization, b2b, accounts and user progress **services**.
- **API Tests**: Wrote a suite of **API Integration** tests (mostly for myself, at the beginning). It has grown and is now a valuable tool used by many teams to find regressions.
- **Mentoring**: Mentored **junior**, **professional** and **senior** engineers, directly supporting their **growth** to higher roles, including several promotions to **Principal**.
- **Advice and support**: Teams contact me on a weekly basis to discuss system design. Babbel already includes **observability** and **automated infrastructure** by default, so I often advise about sound **domain entities** with clear nomenclature, well defined **system boundaries** and responsibilities, **small deliverables**, and then small deliverables a few more times because the **project only starts once it's in production**.
- **Principal Role**: I was the first engineer to be promoted to Principal. I helped define the role and its responsibilities drawing from my own experience temporarily moving between teams.

Technologies:

- Ruby, Typescript, Go
- Rails, React
- Terraform
- AWS Lambda, DynamoDB, Cloudwatch and many other services
- Datadog, Rollbar

Senior Full-Stack Software Engineer at Movinga

Feb 2017 - August 2017

I joined a team of full-stack engineers at Movinga and immediately started working on a microservice and its three client applications. I was also responsible for starting a couple of frontend applications from scratch. Later we extracted a pricing microservice from the main legacy code in order to improve deployment times, frequency (from once every two weeks to 10 times a day) and provide better insights about our prices.

Technologies:

- Ruby, Javascript

- Grape, React, Redux and other supporting libraries for both frontend and backend
- AWS, Docker

Senior Software Engineer at Globo.com

May 2015 - Jan 2017

I joined Globo.com to work on the web media platform team. Globo is the biggest television channel in Brazil. As part of its technology team, we were responsible for ingesting raw television content videos and encoding them for online streaming. We worked on a system that provided both an UI and an API for uploading videos and metadata. This system was distributed in more than 50 locations across Brazil, gathering and processing videos from all partners. We processed more than 2.5TB of videos on a weekly basis, accounting for more than a thousand hours of content.

Accomplishments:

- Designed a content protection microservice that integrates with 3 third party APIs
- Participated on hiring interviews for most people hired in the Porto Alegre office
- Started an observability initiative to gather video encoding metrics and better understand our system
- Helped shape the office culture
- Mentored less experienced engineers

Technologies:

- Ruby, Javascript and Coffeescript
- Rails and Backbone
- Ffmpeg and x264
- Mongo and Redis

Senior Software Engineer at Bearch, Inc.

November 2014 - April 2015 *(6 months)*

We developed an anonymous social network. The whole team worked with all parts of the development, so I had the opportunity to work with a Go backend hosted on Google App Engine, an Android and an iOS application. All of these technologies were new to me.

Accomplishments:

- Shipped a huge amount of features in very little time due to our experimental nature
- Created a pipeline for building and deploying android and iOS apps to their respective stores, improving the testing and release cycles
- Created a camera for Android that worked on the new screen sizes being released at the time.
- Worked daily with unknown technologies and new challenges

Technologies:

- Go on Google App Engine
- Javascript with Angular
- Android with Java
- iOS with Objective C

Software Engineer at e-Core

March 2013 - November 2014 *(1 year 9 months)*

We developed a single page application for a remote client in New York. I was hired not only because I knew some of the technologies involved but also because I was willing to learn the ones I didn’t know. This was a project that could change very fast.

Accomplishments:

- Automated the deployment process and reduced the total time from 2 hours down to 160 seconds.
- Helped migrate a legacy application from Grails to Rails
- Experimented with virtual machines, docker and several deployment strategies

Technologies:

- Ruby, Javascript, Coffeescript
- Rails, Knockout.js
- Groovy with Grails
- Puppet, Chef
- Mongo, Postgres, Redis, Solr
- VirtualBox, Docker
- Nginx

Software Engineer at Codeminer42

July 2012 - February 2013 *(8 months)*

We developed systems for startups. It was my first experience with Rails, but also included frontend development. Since each engineer was responsible for managing a whole project, our responsibilities involved coding, creating interfaces, thinking about user experience, negotiating with clients, managing infrastructure and deployment, etc.

I was one of the first people to be hired on our branch of the company, so I had the opportunity to help build the culture we wanted. I learned a lot from the experience and it helped shape some of my future career goals.

Accomplishments:

- Migrated an application from rails 2 to rails 4
- Helped shape the office culture
- Developed a project from conception to delivery

Technologies:

- Ruby, Javascript
- Postgres

Development Consultant at ThoughtWorks

January 2011 - June 2012 *(1 year 6 months)*

Consisted mainly of consulting and web development for the retail industry, with daily client facing situations. I worked mostly on maintaining and developing new features for a large legacy Java code base. Teams were distributed across Brazil, India and the US.

This was my first contact with agile practices. I have learned much from it, including the importance of pair programming, TDD and Continuous Integration. I had a mentor that helped me build most of my ideas about development.

Accomplishments:

- Started an initiative to improve development and feedback cycles
- Introduced Javascript unit tests into the project
- Started a culture of looking into build status
- Extended the development tools, greatly improving development speed and team motivation

- Acquired the trust of some of the clients, making our work much easier
- Participated on ThoughtWorks University, which changed my life
- Started learning about interviewing candidates

Technologies:

- Java and Spring
- Javascript
- Ruby for integration tests

Education

Universidade Federal do Piauí (UFPI)

Bachelor of Science (BS), Computer Science, 2004 - 2007

Languages

- Portuguese (Native)
- English (Fluent)