2. First approach

Back to home | 3 - Case diagrams

Our metodology

We develop software solutions with agile metodology. First, we need to define scope and requirements of the project and then we can proceed to development planning.

Usually, we develop in six weeks cicles and, then, we take two weeks to analyze the job done and refine the next development cycle.

For us, is fundamental have a MVP (minimum viable product) delivered even from the first development cycle. For you, that means a first version of your product totally functional with his fundamental features ready to use (IN SIX WEEKS).

Working in this way, after each delivery, we have two weeks to analyze the product with you and your users and planning posible pivots. In our experience, it's very frecuent that users don't use some very complex features and, in the other hand, are loving other thing that we considered a "secondary thing" at first. Fortunately, we don't need wait a year or six months to take other path or maybe give more protagonism to that secondary thing that was awesome for your users. *We could do that in the next six weeks development cicle*.

Tech stack

For this specific problem, we propose to use the following technology stack. It ensures scaling posibilities and fast developments cycles. Plus, we love using modern and super powerful technologies.

Frontend

Next JS (used by companies like Netflix, Uber, Starbucks, or Twitch) with Tailwind styles.

SRE

Debian O.S., Nginx, Passenger, Docker and Kubernetes.

Backend

Ruby on Rails (used by companies like Airbnb, Shopify, Github, Scribd, Slideshare, Soundcloud or Twitch) and Sidekiq for background jobs.

Languages

Mainly, Typescript and Ruby

Databases

Mainly, PostgreSQL and Redis for caching and supporting in-memory operations.

Third Party services

AI capabilities provided by Elastic Search and ML models programmed by this specific application.

New Relic for infraestructure monitoring, logging and error tracking.

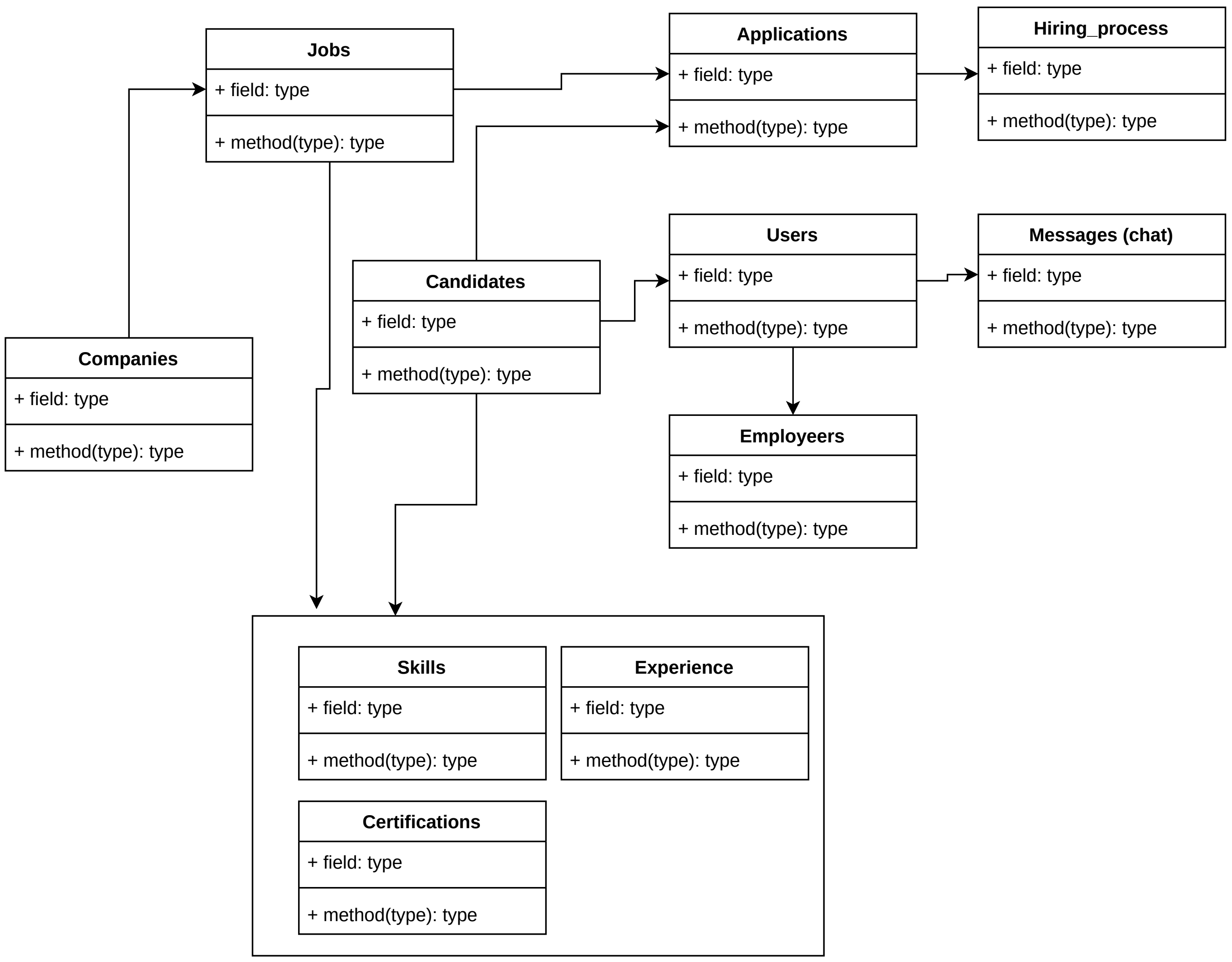
Platforms

At the moment, web application with mobile-first design.

Extensible to desktop and mobile apps.

Preliminar models

This is a simplified version of full class diagram. Methods, attributes and encapsulated objects are not shown here.



Relations

Click here

Preliminar views

A first fast estimation is that is necessary build a CRUD for 12 resources. So, we need, at least, 48 views.

Team

We suggest a team composed by:

- Project manager (part-time).
- UI designer.
- SRE specialist.
- Sr. Frontend Developer.
- Sr. Backend Developer.
- Sr. QA/QC Engineer (part-time).

Time estimation for deploy a MVP

This preliminary estimations could vary a lot after we hold the first meeting with technical team, but give an idea of time, complexity and budget.

- First six weeks cicle: BASIC and functional MVP.
- Two weeks cool down.
- Second six weeks cicle: COMPLETE MVP, fix bugs.
- Two weeks cool down.
- Third six weeks cicle: Adding features (like IA, login with social media and other requirements arisen in cool down cicles).
- Two weeks cool down: Review next steps (scalability, new features, general balance).

Total duration: 24 weeks (6 months) with a total functional first product since second month.

- *Our work*: 2100 hours of team work. This have an approximate cost of US$ 52500.
- *Other expenses not contemplated*:
  - Legal (privacy, terms and conditions contracts).
  - Third-party licences (ElasticSearch and New Relic).
  - Domains, SSL.
  - Server infraestructure leasing (API, databases, web frontend, notifications).