ComparaJá.pt

Tech Challenge

Full Stack Software Engineer

Things to keep in mind:

- This challenge is for the **Full Stack Software Engineer** position at ComparaJá
- We'll give you **5 days** to complete this challenge - please let us know if you need more time
- You should **do this challenge by yourself** (but feel free to google anything you need 😀)
- When answering the questions, please make **clear any assumptions** that you have made
- We prefer **quality over quantity** - avoid answering questions for the sake of solving them all
- To submit your answers, **create a GitHub repository** and share it with **gaspar.costa@comparaja.pt**

Have fun and good luck! 💪🚀

Context

ComparaJá is one of Portugal's largest price comparison platforms. We allow customers to compare prices from different providers across multiple verticals, including Home Loans, Personal Loans, Credit Cards, Broadband and more.

To provide the best possible experience, we personalize comparisons by asking customers for specific details and showing them the results that best fit their profile.

For this challenge we'll be looking at Broadband (a.k.a. "TV Net Voz").

The Case

We have been asked to develop a new results page (see current one here) for the Broadband vertical and, as a part of our Product & Tech team, you'll be involved in this initiative. After some discussion, the team has decided that you'll be responsible for a couple of things:

- Designing and implementing the API that returns the list of products to be displayed
- Implementing the list that displays the products returned by the API

The rules

When answering the questions below, please consider that the team's tech stack includes: **Java, Groovy, Python, JavaScript, SQL and AWS.**

Questions

We provide an estimation of how much time you should spend on each question. This is supposed to give you a sense of complexity - the less time, the simpler it is. Don't overthink it. Overall, this challenge should take **about 2h30 - 3h30 to finish.**

There are no wrong answers or thoughts, we appreciate owned minded people. If you can't finish, no worries, send it back to us and we will evaluate.

Spend some time exploring the files that were sent with this challenge - knowing them well will help you answer questions quicker. You can also take a look at our current results page and understand the dynamics of it (providers, filters, etc) to have a bit more context.

API Design & Implementation

1. *(5-10min)* Since we'll only be looking at Broadband, we need to start by filtering all products related to other verticals. Looking at the CSV files provided to you and assuming that each file represents a separate table, **write the SQL query that returns a list of all products for Broadband.** *Hint: Bear in mind that some providers might not be active at this point.*

2. *(20-30min)* Now, that we have all the active products for Broadband, let's start filtering them out. Using the query below as an example, **write a function that returns a list of products based on specific filters.** *Hint: If needed, assume you have the initial list of objects in memory.*

   Example query:
   a. Internet's download speed needs to be higher than 100Mb/s
   b. Have, at least, one mobile phone
   c. Mobile phones need to have more than 2Gb of mobile data
   d. Product costs, at most, 60€

3. *(45min-1h)* We now know how to fetch the relevant products, let's make sure your fellow engineers at ComparaJá can access this data.

   In this exercise, we need you to **design an API that returns a list of products relevant to a particular user.** We should be able to pass in some filters (similar to the ones in the previous question). *Hint: You can answer using a plain text file or a tool like Mocky.io*.

   Your answer should:
   a. Outline your endpoint(s). *Hint: assume the base URL is `api.comparaja.pt`*
   b. Clearly define your inputs and outputs
   c. If applicable, the type of security you would apply to it
   d. (At least) 2 default responses status from your API

Build Front-end for Results Page

If you managed to deploy your API, feel free to use it in the following questions. If not, simply hardcode some sample results.

We're now ready to build our brand new results page! Let's break it down into two parts:

1. *(1h-1h30)* Create a UI (ignore formatting for now), where you display a list of cards - one for each product returned from your API. Your project should include the following:

   a. A request to get the products from your API
      Hint: If you hardcoded the results, comment out this section
   b. A README file, explaining how to run it locally

2. *(20min-∞)* Add distinctive formatting (go nuts!) to any product that ***`is_sponsored`***.
   Hint: This is a good time to show off your front-end skills.

Bonus (you don't need to answer these!)

As a bonus question (like a million dollar one), we want you to create a service (data storage + api development) and deploy it (if you can't, no worries, just send us the code).

With that in mind, please do the following:

1. *Create a local database where you can store your data (Postgres is a plus).*
2. *Create your service API and request/response classes (AWS and Groovy are a plus)*