

Technical Test

This test is to be used when evaluating the skills of a prospective Frontend Developer. We designed it to be short (estimated less than 8 hours' worth of work) while making it complex enough that the candidate will have to muster all his capabilities.

Glossary

The candidate will encounter several keywords in this document that each have a precise meaning:

Must means that the condition is mandatory and will result in a failing test if not applied.

Can means that the condition is a nice to have and will not result in a failing test if not applied.

Should means that the condition should be done in a certain way for the candidates own interests (code reuse, code simplicity).

Our Expectations

Our top priority is to see the best of what the candidate can produce in terms of elegant and well architected code as well as his insights into the Angular framework. Considerations like speed of delivery and UI/design skills are of course always considered but are distant second and third.

Requirements

The test **must** be done using angular 16. Below is the list of dependencies that the candidate **can** use:

RxJS subscriptions: <https://github.com/wardbell/subsink>

UI design:

<https://github.com/ng-bootstrap/ng-bootstrap>

Helper functions: <https://github.com/lodash/lodash>

If the candidate uses any other dependencies **must** explain why he makes the choice.

The candidate **must** use either less or scss for the project stylesheets. The candidate **can** use css only to import from a third-party library.

The test **must** be sent back as a zip file or on a public repository and **must** compile, after downloading and installing dependencies (the candidate **must** not send the node_modules folder within the test).

Please don't share the content of this document with anybody outside of Reditus.

Description of the task

The task is here to create 2 pages:

- To upload new images.
- To display the recently uploaded images.

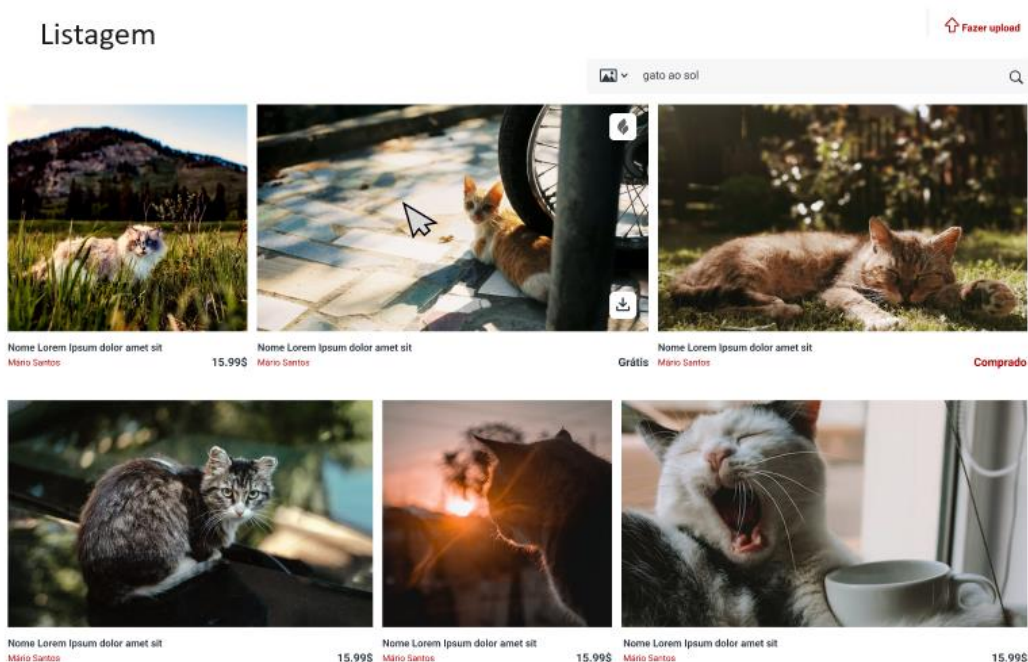
These 2 pages **must** be:

- navigated to using angular routing.
- 2 separate lazy loaded modules.

To display uploaded images

LOGOTIPO

Listagem



Approximative expected result:

This page should display a list of images and they should adjust itself to fill the available space.

This list **must** be read from local storage and display all images stored.

The action button above each image **must** be able to delete that image from the list. You **must** persist in the change in local storage so upon page reload the image is not shown anymore.

An input filter to display only images with a specific “Palavra chave”. The list **must** be filtered each time a new value is entered by the user. A new value is defined as follows:

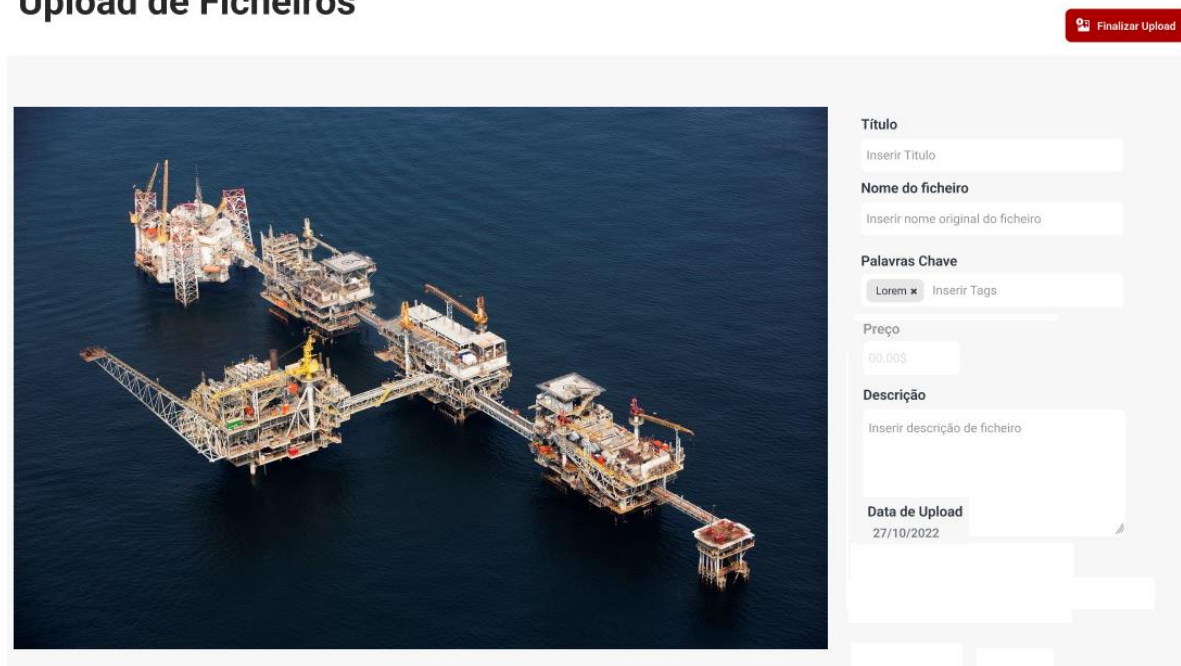
1. the user inputs at least one key stroke.
2. A debounceTime of 300 milliseconds is applied, waiting for the user's next key stroke.
3. If no other key stroke was added the list **must** be filtered
4. If another key stroke was added, go back to step 2.

The candidate **must** use a reactive form approach and RxJs operators to achieve the desired result.

To submit new images

Approximative expected result:

Upload de Ficheiros



The form **must** be an Angular Reactive form, the candidate **cannot** use the template driven Form or 2-way data binding.

All fields **must** be validated. The validation rules are the following:

1. All fields are mandatory.
2. The "Preço" **must** be a number.
3. "Título" and "Descrição" fields **must** be of type string.
4. "Nome do ficheiro" must be the name of the file and readonly. There should be a button to choose the file in the file system.
5. At least one "Palavras Chave" **must** be filled.
6. "Criado em" must present actual date and shall open a DatePicker.

On page load the "Finalizar Upload" button **must** be disabled. It will become enabled only when the form is valid.

When clicking "Finalizar Upload" the content of the images listing grid **must** be serialized and saved in the local storage with a specific ID. The content of the images listing must be updated.

The candidate **can** display the validation errors as soon as the form starts being filled in.