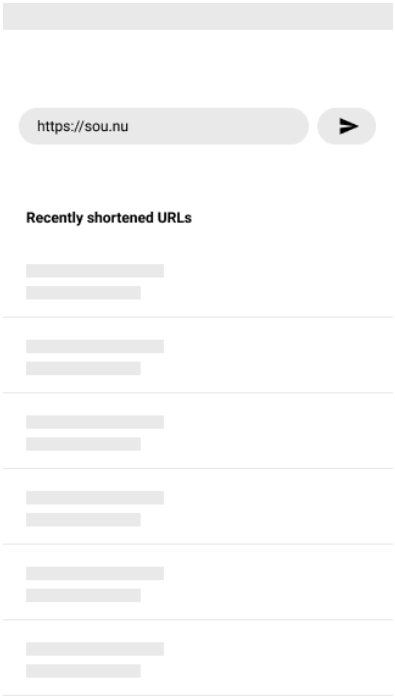
Nubank Mobile Take-Home Test

Description

This exercise aims at creating a small application that allows you to shorten links and displays a history of the recently shortened links to your favorite websites.

In order to do this, you'll use an already implemented service that will handle all the backend logic, which is to accept the links and return an alias for them.

This application is composed of only one screen:



Example

It has:

- One text input in which the user can type the website URL to shorten;
- One button which will trigger the action of sending this link to the service;
- A list with the recently shortened links/aliases.

API

Base Url: https://url-shortener-nu.herokuapp.com

Shorten url

A method for creating aliases with the following body:

Request:

- Path: /api/alias
- Method: POST
    - Body (json): { "url": "<the url>" }

Response:

- Status: 201 on success with the following body
    - Body:

```
{
  "alias" : "<url alias>",
  "_links" : {
    "self" : "<original url>",
    "short": "<short url>"
  }
}
```

Read shortened url

A method for retrieving a link through an alias.

Request:

- Path: /api/alias/:id
- Method: GET

Response:

- Status: 200 on success with the following body
    - Body: { "url": "<the original url>" }

- Status: 404 when not found

What will be evaluated?

We're going to evaluate a set of programming skills, including:

- Architecture chosen with a good separation of concepts
- Unit tests
- UI testing
- State management
- Code organization without code smell (e.g.: lint warnings...)

What will not be evaluated?

- CI/CD Mechanism
- Customized Code Static Analysis

FAQ

Must the UI be equal to the one presented here?
No. The UI is not being evaluated, only the code. Feel free to adapt the UI to the one you'll feel more comfortable with.

Must data be stored?
No. Just keep it in memory.

Must all API endpoints be used?
No. Use only what you feel it's required for solving the problem.