Olá Voncleyton,

Esta etapa do processo seletivo da **Flex** consiste no desenvolvimento de uma aplicação frontend. Ela se comunicará com uma API pública de dados *fake* (*JSONPlaceholder*) e a nossa API Backend para gravação dos registros, ambas do tipo REST.

**Desafio**

O problema que iremos enfrentar é o gerenciamento de devedores, fazendo a relação das dívidas com as pessoas existentes no *JSONPlaceholder*.

***JSONPlaceholder***

Acesse [https://jsonplaceholder.typicode.com](https://jsonplaceholder.typicode.com/).

Lá você terá uma breve explicação de como a API funciona e quais dados são disponibilizados. Um dos principais *endpoints* é o **/users,** que tem como retorno uma lista de usuários com seus diversos atributos.

**Backend**

Foi preparado um backend para que você possa manipular seus objetos do tipo “dívida”. O banco de dados não é relacional, mas terá uma indicação direta aos “usuários” da API JSONPlaceHolder.

Seguem dados para utilização da API:

url: [https://provadev.xlab.digital/api/v1](https://provadev.xlab.digital/api/v1/divida/5ca56b25c4cc0e390083e71d)

parâmetro de **url** obrigatório = **?uuid=xxx** (com o código enviado ao seu e-mail).

|  |  |  |
| --- | --- | --- |
| **Verbo** | **Path** | **METODO / PARÂMETRO** |
| **GET** | /divida | *getAll* |
| **GET** | /divida/:id | *getDividaById* ( \_id ) |
| **POST** | /divida | *createDivida* (espera body):  {idUsuario: 1, (plaaceholder id)  motivo: “Parcela 3 carro”,  valor: 199.99} |
| **PATCH/PUT** | /divida/:id | *updateDivida* (espera body):  {idUsuario: 2, (plaaceholder id)  motivo: “Parcela 3 casa”,  valor: 299.99} |
| **delete** | /divida/:id | *deleteDivida* ( \_id ) |

**Implementação**

O objetivo desse projeto é o relacionamento de dívidas com nossos clientes (usuários do *JSONPlaceholder*). Cada usuário é uma pessoa com dívidas, na qual iremos fazer o cadastro no nosso sistema.

Para isso você deverá implementar uma aplicação que possibilite realizar as quatro operações básicas:

1. Adicionar nova dívida, informando qual cliente está relacionada a ela;
2. Obter detalhes de uma dívida;
3. Editar uma dívida;
4. Deletar uma dívida.

O foco deste projeto é a experiência do usuário e a usabilidade do frontend desenvolvido. Fique à vontade para trabalhar com o *mockup* proposto ou criar o seu próprio modelo de tela.

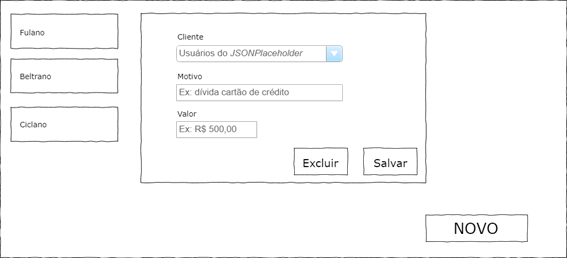
Para a criação de uma dívida, o sistema deverá utilizar o *JSONPlaceholder* para obter a lista de clientes, de modo que o usuário da aplicação possa escolher um cliente na hora da criação da dívida.

Na criação da dívida, o frontend deve apresentar um formulário com todos os clientes disponíveis e também campos relativos às informações da dívida. Ao clicar em salvar a dívida, essas informações devem ser enviadas ao backend, e após salvo, ele deve ser exibido na lista de devedores.

Espera-se que, ao clicar em um devedor, seja apresentada sua lista de dívidas, e ao clicar na dívida, a sua edição seja permitida.

Também deve ser possível deletar um devedor e/ou dívidas associadas ao mesmo. Após a exclusão, o registro deve ser removido da lista também.

Segue um *mockup* da interface como proposta a ser desenvolvida:



**Observações**

* O desenvolvimento poderá ser realizado na linguagem que desejar, utilizando as bibliotecas e frameworks que achar necessário.
* Não é necessário salvar os dados em banco de dados, as operações deverão ser feitas diretamente em nossa API.
* A dívida deverá ter os seguintes atributos:
  + Id do usuário no *JSONPlaceholder*
  + Motivo da dívida
  + Valor
    - * Ao incluir uma dívida, sua data de criação será incluída automaticamente, este dado também deverá ser exibido, no padrão dd/MM/yyyy.
* Dúvidas sobre *JSONPlaceholder* podem ser solucionadas no próprio repositório deles:
  + - [https://github.com/typicode/jsonplaceholder#how-to](https://github.com/typicode/jsonplaceholder" \l "how-to)
* Um design responsivo será um ótimo diferencial
* Frameworks atuais são muito bem vindos (só não esqueça de incluir um tutorial de como subir sua aplicação)

**Sobre a Entrega**

O projeto final deve ser disponibilizado em um repositório público (github, bitbucket, etc..) , e o link do mesmo deve ser enviado para EMAIL,  até dia DATA\_LIMITE ( 7 DIAS APÓS O ENVIO).