



Processo Seletivo: Teste Técnico

Candidato: Eduardo Atene Silva

Número : +55 (31) 9 9424-8936

GitHub : <https://github.com/EduardoAtene>

Linkedin : <https://www.linkedin.com/in/eduardo-atene/>

1. Instalação

Primeiramente, de o git clone do seguinte repositório: <https://github.com/EduardoAtene/testeTecnicoJedis>. Em seguida, abra o terminal ubuntu (caso estiver em windows, siga o passo a passo da instalação/ativação do WLS2: <https://laravel.com/docs/9.x#getting-started-on-windows>) e vá ao diretório do repositório. Logo após ir ao diretório, será necessário abrir um servidor para a execução da aplicação. Para abrir, basta inserir o seguinte código.

```
php artisan serve --port 8000
```

Observação: A --port é opcional. Entretanto, já que vamos utilizar para visualizar a aplicação em um site, será necessário abrir uma outra porta para as API's. Por padrão, defini a porta como da API como 8002.

Após subir o serve, é necessário realizar o seguinte comando para controle de versão do banco de dados:

```
php artisan migrate
```

Para a aplicação front-end, onde utilizará a API do próprio sistema, precisará criar um outro servidor com porta diferente. Basta digitar o seguinte código

```
php artisan serve --port 8002
```

2. Rotas (endpoint)

Aqui se encontra todas as rotas do sistema criado. Dentre as rotas, temos as rotas web e de API (contendo seus parâmetros para requisição).

2.1. Web

- /
- /consulta/{cpf}
- /create
- /store
- /disponibilidade/index
- /disponibilidade/store

2.2. API

- /api/v1/simulacao/credito
 - cpf
- /api/v1/simulacao/oferta
 - cpf
 - instituicao_id
 - codModalidade
- /api/v1/sumulacao/disponibilidade
 - cpf
 - valorSolicitado
 - qntParcelas

2.3. Todas Rotas

- /
- /consulta/{cpf}
- /create

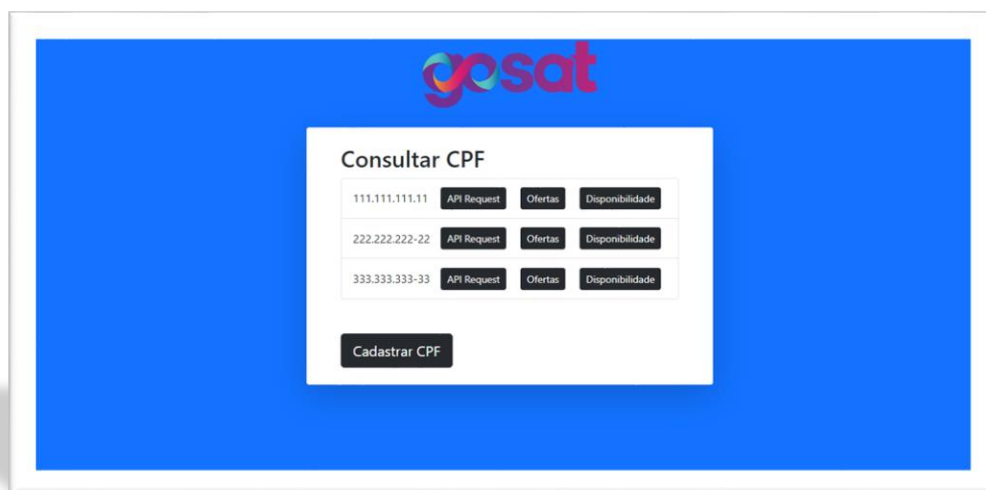
- /store
- /disponibilidade/index
- /disponibilidade/store
- /api/v1/simulacao/credito
- /api/v1/simulacao/oferta
- /api/v1/sumulacao/disponibilidade

3. Telas

3.1. Tela principal

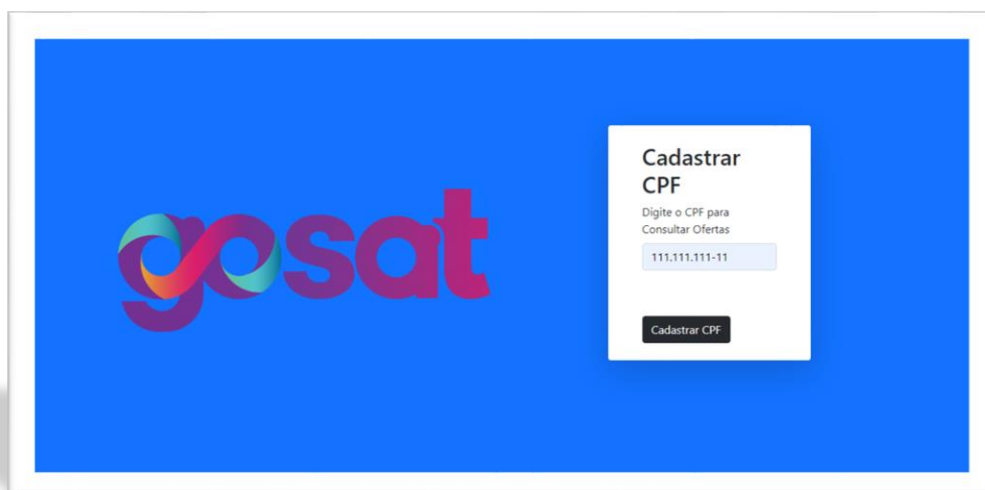
3.2.

Tela principal é mostrado todos os CPF cadastrados no Sistema. Além disso, é possível realizar a chamada de cada uma delas.



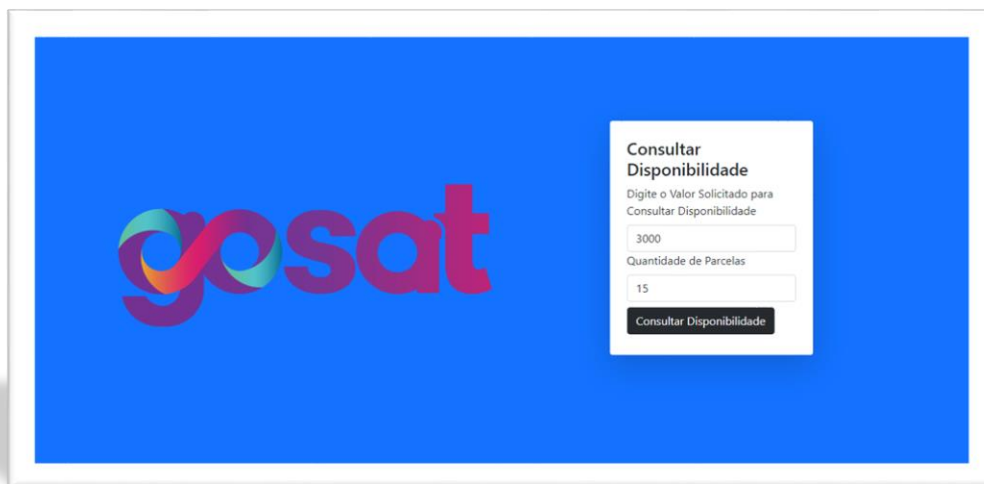
3.3. Tela Cadastrar CPF

Tela para cadastrar os CPF no Sistema para realizar os request na API do sistema.



3.4. Tela Disponibilidade

Tela de disponibilidade, que é para consultar a disponibilidade do CPF escolhido no Sistema. Ao pesquisar por disponibilidade, irá ser redirecionado para a Tela de Disponibilidade de Crédito do CPF.



3.5. Tela de Disponibiliidade de Crédito do CPF

Tela de disponibilidade de crédito do CPF, irá ilustrar todas as disponibilidades de ordem da mais vantajosa, para a menos vantajosa. Ela é responsável por realizar uma chamada para a API criada no Sistema.



Instituição Financeira	modalidade de Crédito	valor a Pagar	valor Solicitado	taxa/juros	qntParcelas
Banco PingApp	crédito consignado	3,035.40	3000	35.40	15
Financeira Assert	crédito pessoal	3,109.50	3000	109.50	15
Banco PingApp	crédito pessoal	3,148.50	3000	148.50	15

3.6. Tela de Ofertas do CPF

Tela de ofertas do CPF, irá ilustrar todas as ofertas de crédito de um determinado cpf. Ela é responsável por realizar uma chamada para a API criada no Sistema.



CPF Crédito

Finança Assert	Banco ATR SA
crédito pessoal	crédito pessoal
Saque FGTS	

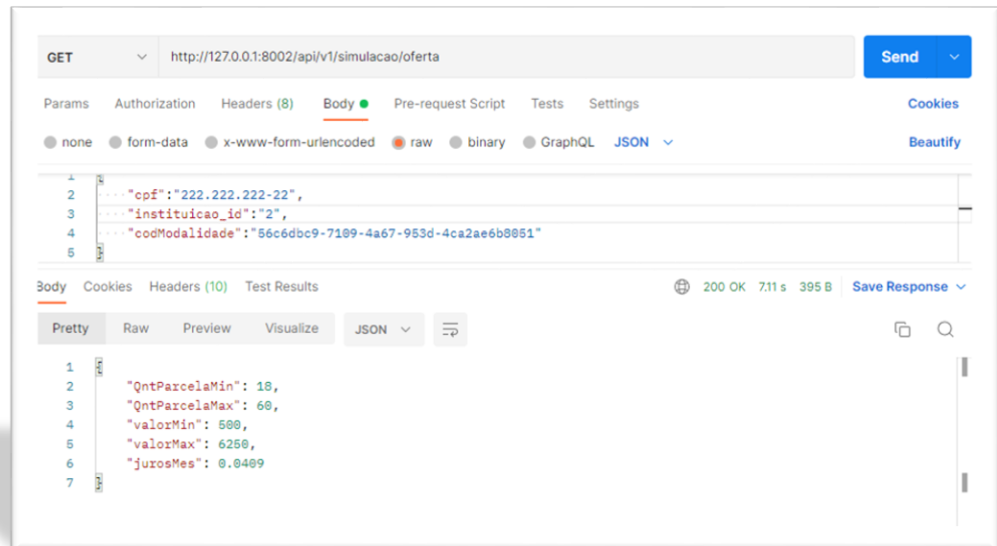
Voltar

3.7. Tela Puro JSON do Retorno da API de Crédito

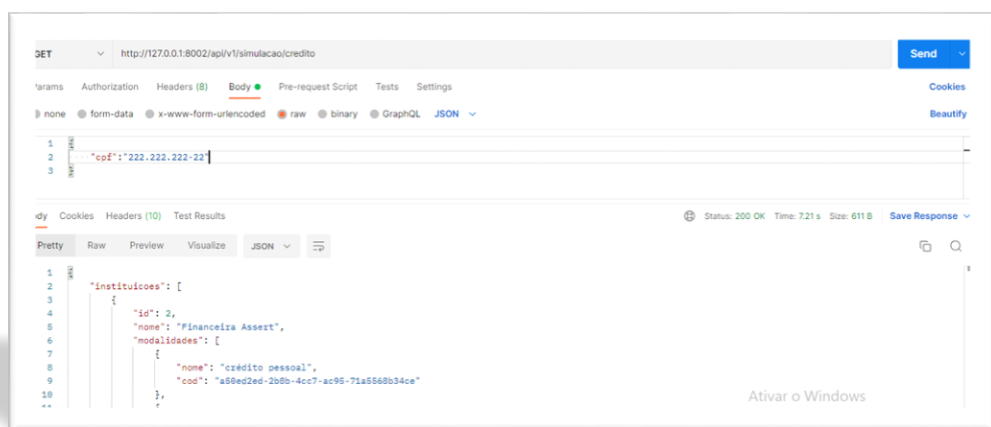
```
// http://localhost:8001/api/v1/simulacao/credito?cpf=222.222.222-22
{
  "instituicoes": [
    {
      "id": 2,
      "nome": "Finança Assert",
      "modalidades": [
        {
          "nome": "crédito pessoal",
          "cod": "a50ed2ed-2b8b-4cc7-ac95-71a5568b34ce"
        },
        {
          "nome": "Saque FGTS",
          "cod": "56c6dbc9-7109-4a67-953d-4ca2ae6b8051"
        }
      ]
    },
    {
      "id": 3,
      "nome": "Banco ATR SA",
      "modalidades": [
        {
          "nome": "crédito pessoal",
          "cod": "12"
        }
      ]
    }
  ]
}
```

4. Retornos API's

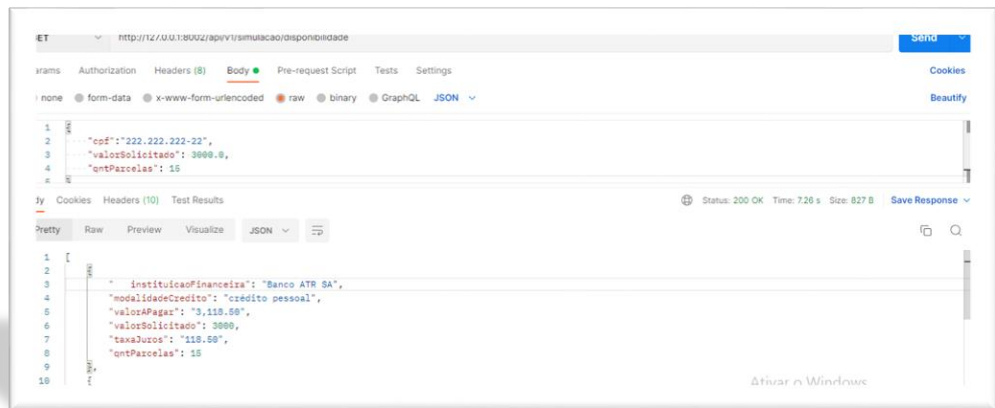
4.1. Crédito



4.2. Oferta



4.3. Disponibilidade



5. Ferramentas e Conceitos Utilizadas

- Laravel v9.x;
- Bootstrap v5.5;
- SQLite;
- Arquitetura MVC;
- Padrão de Armazenamento: Active Record;
- Postman (Testagem das API's);
- API Rest;