

# Pré-requisitos

- JAVA: versão 11.
- Netuno
- NodeJS: versão 18.15.0 ou maior.
- NPM: versão 9.5.0 ou maior.
- Acesso a um banco de dados PostgreSQL

## Bibliotecas Backend

- @netuno/service-client: versão 1.2.0

## Bibliotecas Frontend

- antd: versão 5.1.2,
- @animated-burgers/burger-slip: versão 1.1.2,
- antd-password-input-strength: versão 2.0.1,
- @testing-library/jest-dom: versão 5.16.5
- @testing-library/react: versão 13.4.0
- @testing-library/user-event: versão 14.4.3
- classnames: versão 2.3.1,
- craco-less: versão 2.0.0,
- less: versão 4.1.3,
- less-loader: versão 11.1.0,
- react: versão 18.2.0,
- react-dom: versão 18.2.0,
- react-ga: versão 3.3.1
- react-icons: versão 4.7.1
- react-map-gl: versão 7.0.20
- react-scroll-parallax: versão 3.3.2
- sal.js: versão 0.8.5
- react-router-dom: versão 6.2.1,
- react-scripts: versão 5.0.1,
- sprintf-js: versão 1.1.2,
- web-vitals: versão 3.1.0

# Configuração do Projeto

## Configuração do Netuno

Para configurar o Netuno será necessário criar o arquivo de configuração **config/\_development.json** neste arquivo é onde ficará informações como as credenciais do banco de dados, configuração do Cron Job etc. Para criar o arquivo de configuração utilize o arquivo **config/sample.json** de exemplo.

### Informações Gerais Sobre o Netuno

- O servidor do netuno roda por padrão na porta 9000 (ex: <http://localhost:9000>)
- Os serviços do netuno são acessíveis a partir da rota **/services** (ex: <http://localhost:9000/services>)

### Configuração do Nome da Aplicação

Para configurar o nome da aplicação será necessário editar a sessão de configuração na chave name do arquivo **config/\_development.json**.

```
"name": "netuno_website",
```

Substitua os valores das seguintes chaves com as credenciais corretas do seu banco de dados:

- **name**: O nome desejado para a aplicação.

## Configuração do Frontend

### Informações Gerais Sobre o Frontend

- O frontend da aplicação roda por padrão na porta 3000

### Configuração da Conexão Com os Serviços do Netuno

Para configurar a conexão com os serviços do Netuno será necessário editar a seção de configuração com a chave api do arquivo **website/src/config/config.json**.

```
"services": {"api": "http://localhost:9000/services/"},
```

Substitua os valores das seguintes chaves com os valores corretos:

- **services**
  - **api**: URL base para os serviços do netuno.

## Rodando o Projeto

Para rodar o projeto basta entrar na pasta onde o Netuno foi instalado e rodar o comando `./netuno server app=${NOME_DA_PASTA_DO_PROJETO}`.

## Estrutura do código do website

O código está organizado da seguinte forma:

- **website/src/base** Contém os componentes de cabeçalho (**header** e **menu**) e o rodapé (**footer**), e ainda o alerta de privacidade de cookies.
- **website/src/common** Código essencial e útil de modo geral, e ainda o motor de processamento das páginas (**Builder**).
- **website/src/components** Componentes para ser utilizados para construir os conteúdos das páginas.
- **website/src/components/functionality** Contém os componentes feitos à medida para ser integrados nos conteúdos das páginas.
- **website/src/config** Parametrização da configuração essencial utilizada em código.
- **website/src/styles** Aqui contém o CSS principal e global, com a gestão das variáveis (**variables**) de configurações de layout e design.
- **website/src/pages** Páginas desenvolvidas à medida sem serem processadas pelo motor do CLUAR CMS.
- **website/src/App.js** Definição das rotas de navegação (**react-router-dom**) e tem a estrutura global do website.
- **website/src/index.js** É onde tudo começa, também é onde é carregado o conteúdo base do CLUAR CMS e onde define o endereço da API REST de serviços.
- **website/craco.config.js** Customização das variáveis de layout e design do Ant.Design.

## Estrutura do código da aplicação netuno

Sobre a estrutura da aplicação netuno:

- **config** Configuração da aplicação Netuno como base de dados, comandos executados na inicialização, CORS, SMTP, e muito mais.
- **db** Onde fica os ficheiros de base de dados (H2Database) e onde pode ser guardado scripts para outros tipos de bases de dados.
- **public** Ficheiros públicos da aplicação Netuno.
- **server** Código de servidor da aplicação Netuno.
- **server/sevices** Código a programação dos serviços da API REST.
- **storage** Ficheiros gerais geridos pelo servidor da aplicação Netuno.
- **ui** Programação do dashboard que fica no backoffice da aplicação Netuno.

# Customização de Layout & Design

Para customizar a aparência é utilizado o LESS, então basta editar todos os arquivos **.less** que estão dentro da pasta **website/src** e nos respectivos subdiretórios.

O código LESS central e global fica em **website/src/styles**, dentro desta pasta ficheiro **variables.less** tem as principais parametrizações de estilos globais.

## Estilização do Ant.Design

A estilização do Ant.Design é feita no ficheiro **website/craco.config.js**, dentro das configurações do less é configurar as variáveis do Ant.Design em `modifyVars`, verifique as variáveis do Ant.Design que podem ser redefinidas:

- `Themes/default.less`

## Os principais componentes ReactJS

### Cabeçalho (Header) e Menu

- **website/src/base/Header** Contem os ficheiros responsáveis pelo header e menu.

### Rodapé (Footer)

- **website/src/base/Footer** Contém os ficheiros responsáveis pelo footer.

### Rotas com React Router

- **website/src/app.js** Ficheiro responsável por definir as rotas.