|  |
| --- |
| **DOCUMENTAÇÃO** |

# **INTRODUÇÃO**

Para o presente teste foi escolhido o framework Laravel v8.x, tanto backend quanto frontend. O banco de dados, como foi proposto, é um arquivo SQLITE localizado dentro da aplicação. Para seu desenvolvimento, foram utilizadas algumas bibliotecas que serão listadas a seguir:

## **Backend**

* Dingo API
  + Biblioteca base para criação e manuseio da API.
* JWT Auth
  + Biblioteca responsável por gerir a autenticação da API utilizando token para acesso.
* Spatie/Laravel Permission
  + Ferramenta utilizada para gerar as permissões dos usuários, bem como a estrutura no banco de dados.
* Spatie/Laravel Activitylog
  + Biblioteca responsável por realizar logs de acesso dos usuários.

## **Frontend**

* Bootstrap 4.5;
* AdminLTE 3;
* jQuery 3.5;

Todos os *assets* do frontend estão sendo adicionados remotamente (CDN).

*Obs: Nenhum compilador CSS foi utilizado, visto que o template utilizado no painel já possui sua versão “minificada”.*

# **Principais Arquivos e Classes**

* 1. **Controllers**
     1. Namespace
        1. App\Http\Controllers
* **UserController**

Arquivo que controla todas as ações frontend do CRUD de User.

* + - 1. App\Http\Controllers\API
* **UserAPIController**

Arquivo que controla todas as ações backend do CRUD da classe User.

* **JWTAuthController**

Arquivo responsável pela autenticação do frontend com a API (backend) e demais ações referentes à sessão do usuário.

* 1. **Middlewares**
* **JWTMiddleware**

Arquivo responsável por verificar o token de acesso à API.

* **UserRolePermission**

Arquivo responsável por verificar a permissão de acesso do usuário para cada página do sistema.

* 1. **Routes**
* **API**

Arquivo que possui todos os *endpoints* da API.

* **WEB**

Arquivo que possui todas as rotas de acesso do frontend.

* 1. **Helper**

Arquivo que auxilia no manuseio das respostas das requisições da API.

* 1. **Trait**
* **ConsumesExternalApi**

Arquivo responsável por realizar as requisições originadas do frontend e receber as informações processadas no backend.