

Inicialização

```
alias sail='[ -f sail ] && sh sail || sh vendor/bin/sail'
sail up -d
```

Descomentar linhas a em App/Core/Providers/AuthServiceProvider

```
exec -it crud_host_araguaia_mysql_1 bash
mysql -uroot -ppassword
use mysql;
grant all privileges on *.* to 'bd_user'@'%';
exit

exit
```

```
sail migrate
sail artisan db:seeder -class=InitSeeder
```

```
sail npm run dev
```

acessar localhost:8001

login: admin
senha: 123456

MySQL

Triggers

- tg_frentistas_insert_before
- tg_frentistas_update_before
- tg_motoristas_insert_before
- tg_motoristas_update_before
- tg_postos_insert_before
- tg_postos_update_before

Functions

- fn_calcula_idade
- fn_limpa_telefone

Permissões

As permissão são controladas utilizando a técnica de Gates, registradas em (App/Core/Providers/AuthServiceProvider) e podem ser manipuladas através do banco de dados pelas tabelas gates e nivel_gates (pivot).

Rotas Web (routes/web.php)

As rotas são dinâmicas. Para adicionar um novo esquema de rotas, basta inserir um novo valor em \$rotas (linha 27).

Rotas Api (routes/api.php)

As rotas são dinâmicas. Para adicionar um novo esquema de rotas, basta inserir um novo valor em \$rotas (linha 21).

Listas: (GET) {URL}/api/{NOME DA ROTA}

Cadastrar: (POST) {URL}/api/{NOME DA ROTA}/store

Listas: (DELETE) {URL}/api/{NOME DA ROTA}/destroy/{ID}

Consultar: (GET) {URL}/api/{NOME DA ROTA}/show/{ID}

Blade components

Menu Superior: <x-crud.veiculos.menu
:rota="\$rotaBase">

Botões de Ações: <x-crud.listas.acoes
:rotaBase="\$rotaBase"
:item="\$item">

Input Text: <x-crud.forms.input-text
:label="'CNPJ'"
:name="'cnj'"
:value="\$item"
:required="true"
:mask="'00.000.000/0000-00'">

Hidden: <x-crud.forms.hidden
:name="'user_id'"
:value="\$item">

Select: <x-crud.forms.select
:label="'Nível'"
:name="'nivel_id'"
:value="\$item"
:options="\$opcoesNiveis"
:required="true">

Custom Requests

Cada Model contém suas regras de requests específicas. São encontradas em App/Web/{NOME_DO_MODEL}/Requests.

Actions & DTOs

Utilizei Data Transfer Objects para garantir a tipagem das variáveis e enviá-las para as Actions. São encontradas em Domain/{NOME_DO_MODEL}/DTO.

Para ações únicas no sistema, utilizei Actions, ex.: (StoreVeiculoAction, UpdateFrentistaAction, etc...). São encontradas em Domain/{NOME_DO_MODEL}/Actions.

Para facilitar, criei comandos para gerar novas Actions e DTOs, conforme visto abaixo.

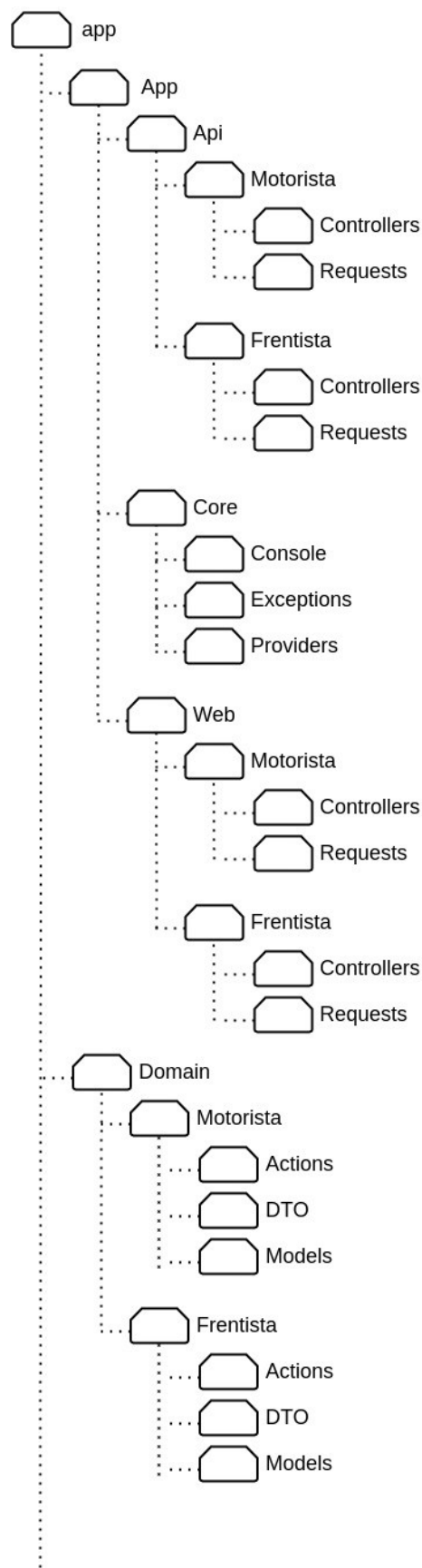
Commands

```
make:Actions: php artisan make:action Caminho --model=User  
make:dto: php artisan make:dto Caminho --model=Motorista
```

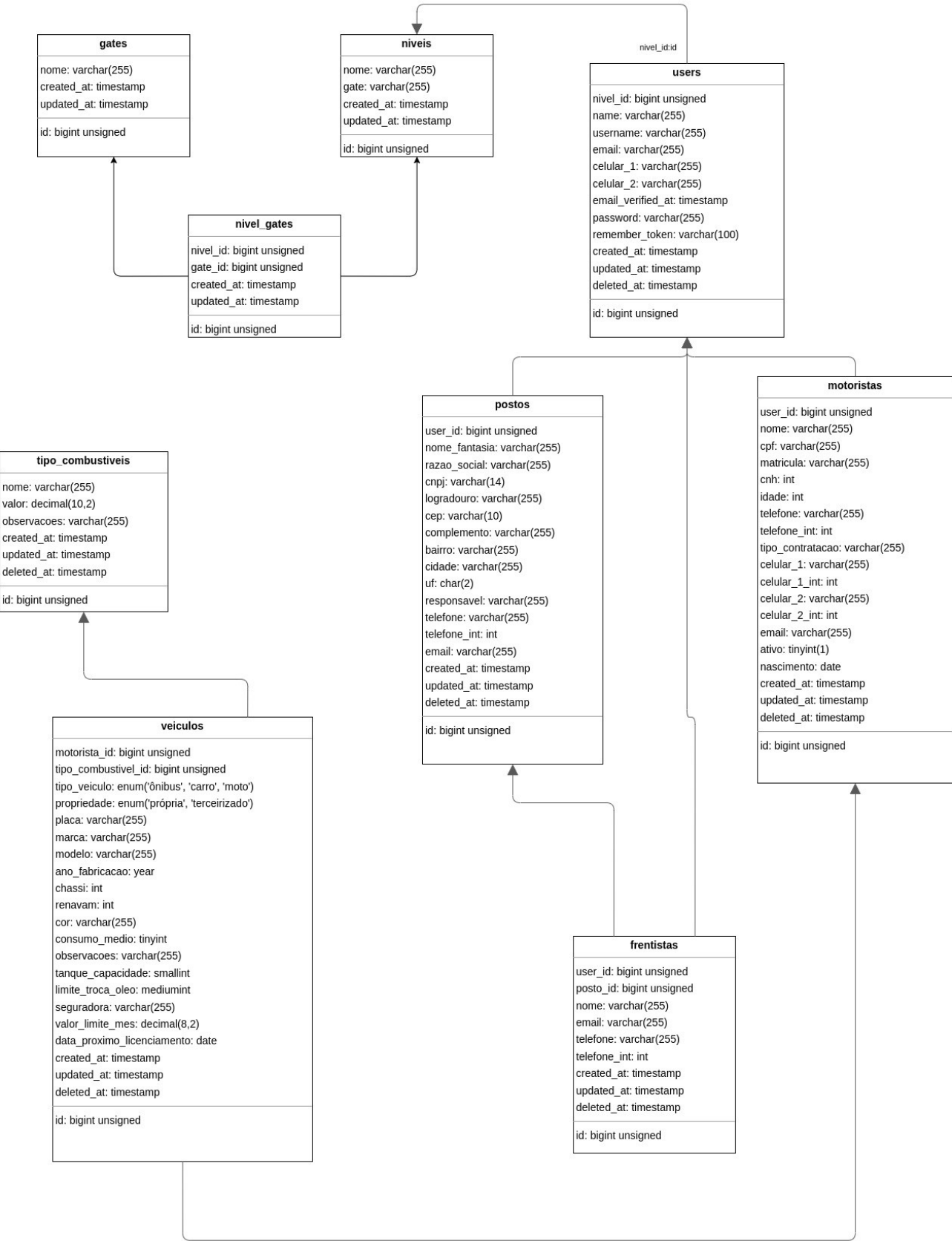
Pacotes Utilizados

```
spatie/data-transfer-object  
barryvdh/laravel-dompdf  
laravel/breeze  
realrashid/sweet-alert  
php-open-source-saver/jwt-auth  
livewire/livewire
```

Arquitetura: DDD



DER



DIAGRAMA

