

## Teste Atêlie de Propagandas

Clone o projeto no Github, siga as instruções do README e vamos rodar esse projeto.

Neste projeto eu usei o Docker, e as imagens PHP 7.4, Redis e MariaDB, além de usar o framework Laravel para a construção da Api rest.

Depois de seguir os passos do readme, verifique se a estrutura de pasta do seu projeto está assim:

```
adolpho.nascimento@CTNLDLL2CK3S63:~/Documentos/estudos/Laravel/atelie$ ls
docker-compose.yml  dockerfile  README.md  src
adolpho.nascimento@CTNLDLL2CK3S63:~/Documentos/estudos/Laravel/atelie$
```

Dentro da pasta src é que está a aplicação Laravel e sua api.

Com o docker-compose up já rodando verifique se no seu terminal está assim:

```
dev_1 |
dev_1 |      Laravel Mix v6.0.18
dev_1 |
dev_1 |      ✓ Compiled Successfully in 569ms
dev_1 |
dev_1 |      |-----|
dev_1 |      |      File      | Size |
dev_1 |      |-----|
dev_1 |      | /js/app.js      | 597 KiB |
dev_1 |      | css/app.css      | 1 bytes |
dev_1 |      |-----|
dev_1 |
dev_1 |      ✓ Mix: Compiled successfully in 588.07ms
dev_1 |      webpack compiled successfully
dev_1 |
dev_1 |      |-----|
dev_1 |      | npm update check failed |
dev_1 |      | Try running with sudo or get access |
dev_1 |      | to the local update config store via |
dev_1 |      | sudo chown -R $USER:${id -gn $USER} /.config |
dev_1 |      |-----|
dev_1 |
dev_1 |      Done in 2.28s.
dev_1 |      Starting Laravel development server: http://0.0.0.0:8000
dev_1 |      [Sun May 9 08:15:02 2021] PHP 7.4.14 Development Server (http://0.0.0.0:8000) started
dev_1 |
```

Se estiver é sinal que tudo está OK.

Abra uma nova aba do terminal e se você ainda não tiver rodado as migrations e seeders a hora é essa. Hehehe

Digite:

```
~/Documentos/estudos/Laravel/atelie$ docker-compose exec dev bash
```

para acessar os comandos artisan do laravel.

A imagem abaixo mostra o comando acima, e já dentro do bash o comando para gerar as migrations e seeders.

```
adolpho.nascimento@CTNLDL2CK3S63:~/Documentos/estudos/Laravel/atelie$ docker-compose exec dev bash
I have no name!@3b0f3f8c7f8a:/app$ php artisan migrate:fresh --seed
Dropped all tables successfully.
Migration table created successfully.
Migrating: 2014_10_11_203331_create_companies_table
Migrated: 2014_10_11_203331_create_companies_table (12.49ms)
Migrating: 2014_10_12_000000_create_users_table
Migrated: 2014_10_12_000000_create_users_table (11.29ms)
Migrating: 2014_10_12_100000_create_password_resets_table
Migrated: 2014_10_12_100000_create_password_resets_table (14.73ms)
Migrating: 2019_08_19_000000_create_failed_jobs_table
Migrated: 2019_08_19_000000_create_failed_jobs_table (11.29ms)
Migrating: 2019_12_14_000001_create_personal_access_tokens_table
Migrated: 2019_12_14_000001_create_personal_access_tokens_table (15.89ms)
Migrating: 2021_05_08_105434_create_participants_table
Migrated: 2021_05_08_105434_create_participants_table (36.78ms)
Seeding: Database\Seeders\CompaniesTableSeeder
Seeded: Database\Seeders\CompaniesTableSeeder (8.80ms)
Seeding: Database\Seeders\UsersTableSeeder
Seeded: Database\Seeders\UsersTableSeeder (57.33ms)
Seeding: Database\Seeders\ParticipantsTableSeeder
Seeded: Database\Seeders\ParticipantsTableSeeder (3.59ms)
Database seeding completed successfully.
I have no name!@3b0f3f8c7f8a:/app$
```

## Regra de Negócio:

A regra de negócio aplicada neste projeto foi a seguinte:

- Como tinha que ser feito um cadastro de Empresas e Participantes para uma campanha, essa regra foi atendida e com um relacionamento entre essas entidades, que no projeto receberam o nome de Company e Participant para padronizar a criação de arquivos e pastas no Laravel.
- Mas, como a Api desse CRUD não pode ser pública na minha visão, implementei a autenticação com utilizando o Sanctum do Laravel e nesse caso um usuário ADMIN é quem faz o CRUD de Empresas e Participantes, esse ADMIN tem que está devidamente logado na Api e após a sua autenticação é gerado um Bearer Token que tem que ser utilizado nos endpoints de CRUD.

Os endpoints ficaram assim:

```
Route::post('/login', [UserController::class, 'login']);

Route::group([
    'prefix' => 'dashboard',
    'middleware' => 'auth:sanctum',
], function () {
    Route::get('/users', [UserController::class, 'index']);

    Route::get('/companies', [CompanyController::class, 'index']);
    Route::post('/companies', [CompanyController::class, 'store']);
    Route::get('/company/{id}', [CompanyController::class, 'show']);
    Route::put('/company/{id}', [CompanyController::class, 'update']);
    Route::delete('/company/{id}', [CompanyController::class, 'destroy']);

    Route::get('/participants', [ParticipantController::class, 'index']);
    Route::post('/participants', [ParticipantController::class, 'store']);
    Route::get('/participant/{id}', [ParticipantController::class, 'show']);
    Route::put('/participant/{id}', [ParticipantController::class, 'update']);
    Route::delete('/participant/{id}', [ParticipantController::class, 'destroy']);
});
```