



Estrutura

Configurando o portal SINPOM em um ambiente de produção.

Estrutura do Projeto

Estrutura do Projeto com Docker

Estrutura do Banco de Dados

Visão Geral

Requisitos

Requisitos do Sistema

Estrutura do Projeto

Estrutura do Projeto com Docker

O projeto do sinpom utiliza o Docker para gerenciamento de containeres e imagens Docker. Ele utiliza o Docker Compose para gerenciamento de containeres e imagens Docker.

```
docker-compose-dev-ubuntu.yml
docker-compose-local.yml
docker-compose.yml
Dockerfile
.env
```

Estrutura do Laravel

O projeto do sinpom utiliza o framework Laravel para desenvolvimento da aplicação. Ele utiliza o Composer para gerenciamento de dependências do projeto.

```
/app
  /Console
  /Exceptions
  /Http
    /Controllers
    /Middleware
  /Library
  /Mail
  /Providers
  /Scopes
/bootstrap
/config
```


Estrutura do Banco de Dados

Visão Geral

O banco de dados do SINPOM é a espinha dorsal do sistema, permitindo armazenamento e a recuperação de dados críticos da plataforma, incluindo contas de usuários, funções, logs de acesso e entidades específicas. Ele usa o MySQL como sgbd, e foi projetado para funcionar perfeitamente com o ORM do Laravel (Eloquent) e é estruturado para se alinhar à arquitetura MVC.

Tabelas Principais

1. **users**: Gerencia detalhes da conta do usuário

Field Name	Data Type	Description
id	INT	Primary key, auto-increment.
name	VARCHAR(255)	User's full name.
email	VARCHAR(255)	Unique email address.
email_verified_at	TIMESTAMP	Email verification timestamp.
password	VARCHAR(255)	Encrypted password.
remember_token	VARCHAR(100)	
google2fa_secret	VARCHAR(255)	
created_at	TIMESTAMP	Record creation timestamp.
updated_at	TIMESTAMP	Record last update timestamp.

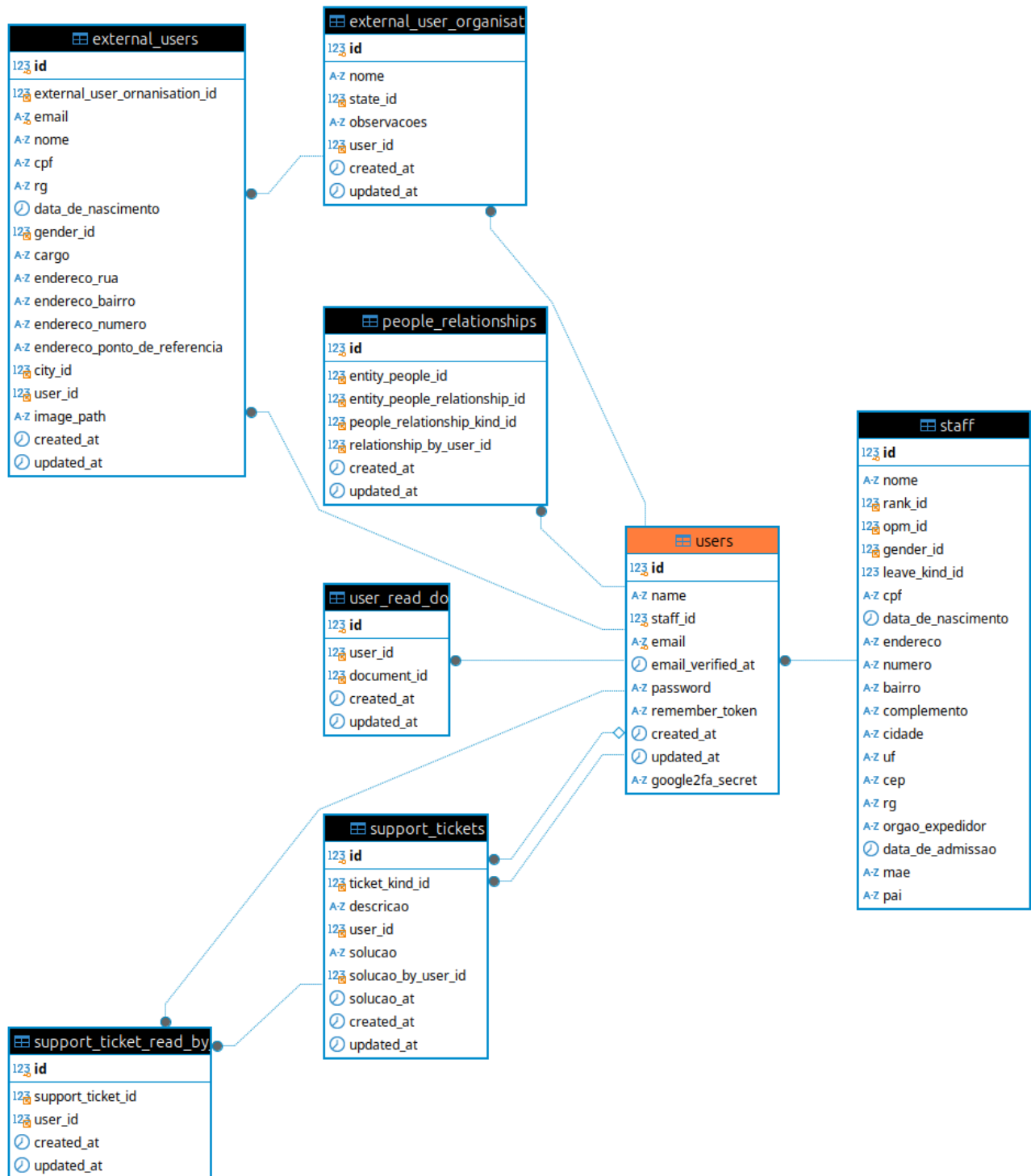
2. **documents**: Tabela de documentos.

Field Name	Data Type	Description
id	BIGINT unsigned	Primary key, auto-increment

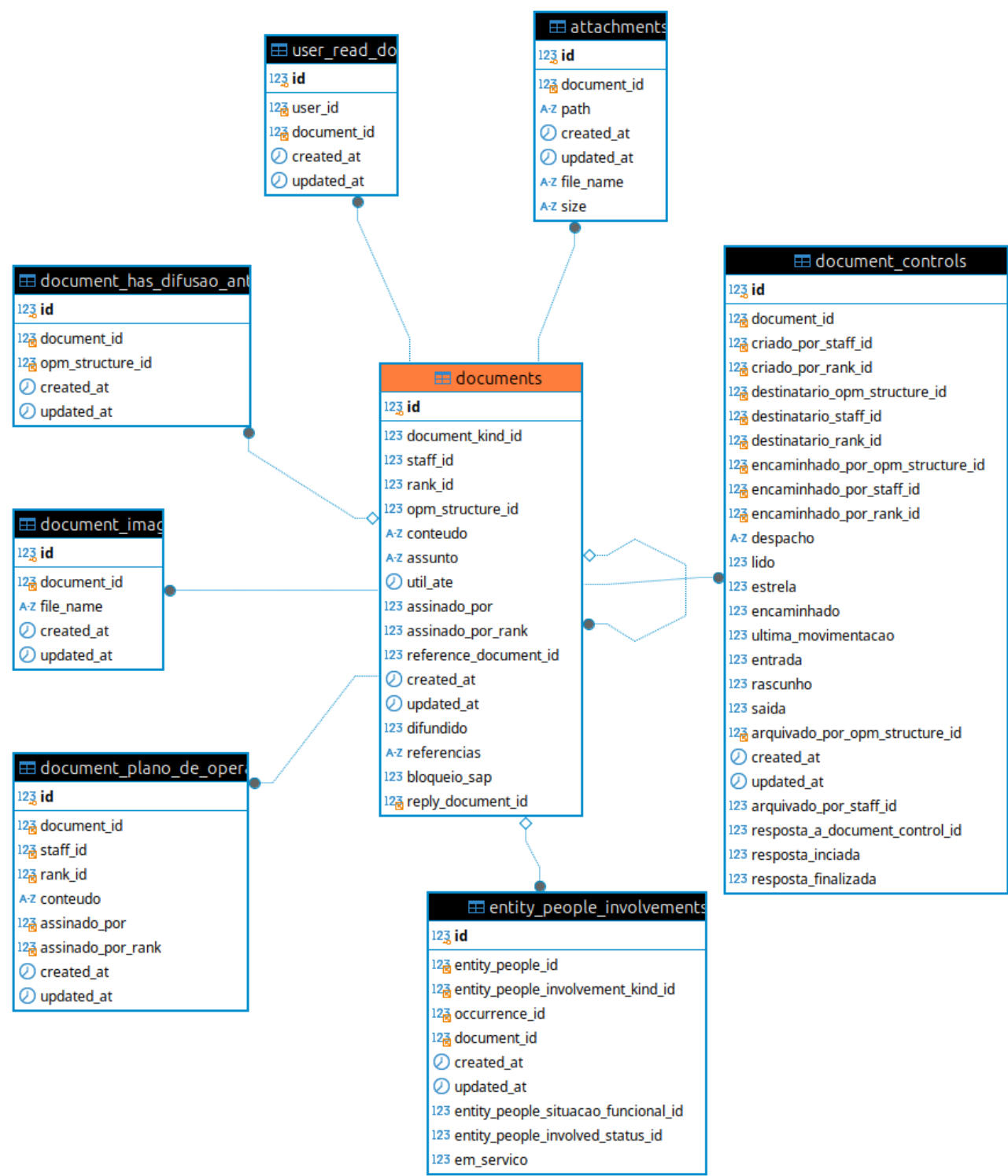
Field Name	Data Type	Description
document_kind_id	BIGINT unsigned	tipo do documento
staff_id	BIGINT unsigned	matricula do criador
rank_id	BIGINT unsigned	posto/graduacao do criador
opm_structure_id	BIGINT unsigned	seção do criador do documento
assinado_por	BIGINT unsigned	assinatura do documento finalizado
assinado_por_rank	BIGINT unsigned	assinatura do documento finalizado
reference_document_id	BIGINT unsigned	documento que deu origem a este
reply_document_id	BIGINT unsigned	
util_ate	DATE	prazo aceitável do documento
conteudo	TEXT	
difundido	TINYINT(1)	
bloqueio_sap	TINYINT(1)	
assunto	VARCHAR(191)	
referencias	VARCHAR(191)	
created_at	TIMESTAMP	Record creation timestamp.
updated_at	TIMESTAMP	Record last update timestamp.

Principais Relacionamentos

Relacionamentos de users



Relationamentos de documents

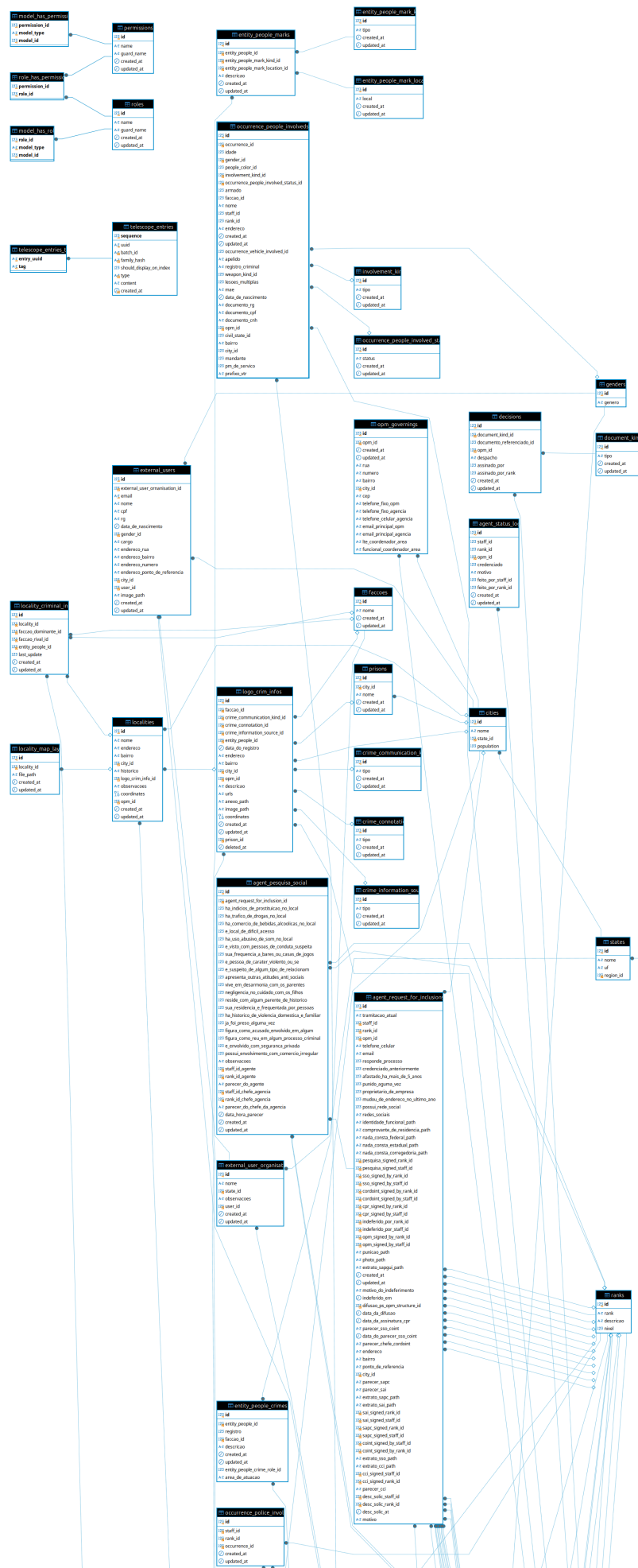


Exemplo de Consulta SQL

Lista todos os agentes de uma OPM ou de um CPR

```
$user = Auth::user();
$opm = Opm::find($request->opm_id);
$isCpr = $opm ? $opm->grande_comando : null;
$agentsQuery = User::join('staff', 'users.staff_id', '=', 'staff.id')
    ->leftJoin('agent_request_for_inclusions',
'agent_request_for_inclusions.staff_id', '=', 'staff.id')
    ->join('opms', 'opms.id', '=', 'staff.opm_id')
    ->join('model_has_roles', 'users.id', '=', 'model_has_roles.model_id')
    ->join('ranks', 'staff.rank_id', '=', 'ranks.id')
    ->join('roles', 'roles.id', '=', 'model_has_roles.role_id')
    ->whereNotIn('roles.name', ['Comandante - OPM', 'Subcomandante - OPM', 'Comandante
- CPR', 'Subcomandante - CPR'])
    ->when(($request->opm_id == $user->staff->opm_id) and ($isCpr), function($q) use
($request){
        $q->where('opms.cpr_id', $request->opm_id);
    })
    ->when($request->opm_id, function($q) use ($request){
        $q->where('staff.opm_id', $request->opm_id);
    })
    ->when(!$request->opm_id and $user->hasAnyPermission(['Consultar agentes OPM']),
function($q) use ($user){
        $q->where('staff.opm_id', $user->staff->opm_id);
    })
    ->when(($request->staff_id) and $user->hasAnyPermission(['Consultar agentes
OPM']), function($q) use ($request, $user){
        $q->where('staff.opm_id', $user->staff->opm_id)
        ->where('users.id', Crypt::decrypt($request->staff_id));
    })
    ->when(!$request->opm_id and $user->hasAnyPermission(['Consultar agentes CPR']),
function($q) use ($user){
        $q->where('opms.cpr_id', $user->staff->opm->cpr_id);
    })
    ->when(($request->staff_id) and $user->hasAnyPermission(['Consultar agentes
CPR']), function($q) use ($request, $user){
        $q->where('opms.cpr_id', $user->staff->opm->cpr_id)
        ->where('users.id', Crypt::decrypt($request->staff_id));
    })
    ->when(($request->staff_id) and $user->hasAnyPermission(['Ver todos agentes']),
function($q) use ($request, $user){
        $q->where('users.id', Crypt::decrypt($request->staff_id));
    })->groupBy('staff.id');
$agentsCountByRank =
$agentsQuery->select('users.*')->orderBy('ranks.nivel')->get()->countBy('staff.rank.rank');
$agents =
```


Diagrama ER



Requisitos

Requisitos do Sistema

- **PHP >= 8.0:** Para o servidor
- **MySQL >= 8.0:** Para o banco de dados
- **Composer >= 2.4:** Para gerenciamento de dependências do Composer
- **Docker >= 20.10:** Para gerenciar os ambientes da aplicação.
- **Docker Compose >= 1.29:** Para gerenciamento de containeres e imagens Docker.
- **Traefik >= 2.9:** Para gerenciamento de rotas e certificados SSL.
- **Node.js >= 16:**

Configuração

Configurando o portal SINPOM para desenvolvimento local.



Configuração do Ambiente

Clonando o Repositório



Instalar dependências

Executar `composer install` para instalar as dependências do sinpom.



Migrations e seeds

Executar as migrações e popular o banco de dados.



Proxy/Proxy Reverso

O Traefik está configurado para gerenciar o roteamento de tráfego com base em domínios. Certifique-se de que o DNS ou o arquivo hosts aponta para os domínios configurados.



Acessando o projeto

Acessando o portal sinpom em ambiente de desenvolvimento

Configuração do Ambiente

Clonando o Repositório

```
git clone https://github.com/sinpom/sinpom.git
cd sinpom
```

Configuração do Docker

1. Copie o arquivo `.env.example` para `.env`

```
cp .env.example .env
```

2. Configure as variáveis de ambiente no arquivo `.env`

Você pode usar o arquivo `.env.local` para configuração local do sinpom.

3. Inicie os containeres com o Docker Compose

Para ambiente de desenvolvimento em Ubuntu:

Criar containeres

```
docker compose -f docker-compose-dev-ubuntu.yml up -d
```

Remover containeres

```
docker compose -f docker-compose-dev-ubuntu.yml down
```

Para ambiente local:

Criar containeres

```
docker compose -f docker-compose-local.yml up -d
```

Remover containeres

```
docker compose -f docker-compose-local.yml down
```

WARNING

Use o arquivo `docker-compose-local.yml` para configuração local do sinpom ou use `docker-compose-dev-ubuntu.yml` para ambiente de desenvolvimento em Ubuntu(#).

Para ambiente em produção:

Criar containeres

```
docker compose -f docker-compose-producao.yml up -d
```


Instalar dependências

Executar **composer install** para instalar as dependências do sinpom.

! INFO

O comando **composer install** deve ser executado dentro do container do portal sinpom

Execute o comando abaixo para instalar as dependências do projeto.

```
docker exec -it sinpom-web bash -c "composer install"
```

Migrations e seeds

Executar as migrações e popular o banco de dados.

Migrações

Criando as tabelas no banco de dados.

! INFO

O comando **php artisan migrate** deve ser executado dentro do container do portal sinpom

Execute o comando abaixo para executar a migrações do banco de dados.

```
docker exec -it sinpom-web bash -c "php artisan migrate"
```

Você pode verificar se o sistema está funcionando corretamente, acesse o portal sinpom em:

- <https://sinpom.docker.localhost>

Populando o banco de dados

! INFO

O comando **php artisan db:seed** deve ser executado dentro do container do portal sinpom

Execute o comando abaixo para popular o banco de dados.

```
docker exec -it sinpom-web bash -c "php artisan db:seed"
```

Proxy/Proxy Reverso

O Traefik está configurado para gerenciar o roteamento de tráfego com base em domínios. Certifique-se de que o DNS ou o arquivo hosts aponta para os domínios configurados.

Usando o Traefik com Docker

WARNING

O Traefik precisa ser executado em um container separado, fora do diretório do sinpom.

Você pode criar um arquivo `docker-compose.yml` para configuração do Traefik, fora do diretório do sinpom e então gerencia-lo com o Docker Compose.

```
# Saindo do diretório do sinpom
cd ../
# Criando diretório para o traefik
mkdir traefik
cd traefik
touch docker-compose.yml
touch dynamic.yml
```

Exemplo de arquivo `docker-compose.yml` para configuração do Traefik:

```
networks:
  proxy:
    external: true
```

Use um arquivo `dynamic.yml` para configuração do Traefik. Exemplo:

```
tls:
  certificates:
    - certFile: /certs/cert.cert
      keyFile: /certs/cert.key
```

Configurando o hosts

No arquivo de hosts do seu sistema operacional, adicione a seguinte linha:

```
127.0.0.1 sinpom.docker.localhost
```

Exemplo de edição do Hosts no Ubuntu

O arquivo de **hosts** no Ubuntu é usado para mapear endereços IP a nomes de domínio de forma local, sem a necessidade de configurar um servidor DNS.

Passo 1: Abrir o Arquivo de Hosts com Permissões de Superusuário

O arquivo de hosts está localizado em `/etc/hosts`. Para editá-lo, você precisa de permissões de administrador. Use o editor de texto de sua preferência (neste exemplo, usamos `nano`).

Digite o seguinte comando no terminal:

```
sudo nano /etc/hosts
```

Passo 2: Editar o arquivo

O conteúdo do arquivo `/etc/hosts` será exibido no editor. Ele geralmente terá esta aparência inicial:

```
127.0.0.1 localhost
127.0.0.1 seu-host-local
```

Adicione a entrada necessária

```
127.0.0.1 sinpom.docker.localhost
```

Passo 3: Salvar e Fechar o Arquivo

```
Ctrl + O
Enter
Ctrl + X
```

Passo Opcional: Limpar o Cache DNS (se necessário)

Se você tiver problemas para resolver o domínio, limpe o cache de DNS. Para sistemas que usam o `systemd-resolved`, execute:

```
sudo systemctl restart systemd-resolved
```


Acessando o projeto

Acessando o portal sinpom em **ambiente de desenvolvimento**

Desenvolvimento local

Após as etapas passadas você já pode acessar o portal sinpom em:

- <https://sinpom.docker.localhost>

Use um dos usuários do sistema para acessar o portal sinpom.

Usuário nível **Agente**:

- Usuário: **agente@sinpom.com**
- Senha: **123**

Desenvolvimento

Configurando o portal SINPOM para desenvolvimento local.



Controle de Versão

Práticas de Versionamento



Casos de Uso

Em breve



Detalhamento de Funcionalidades

Em breve



Plano de Testes

Em breve

Controle de Versão

Práticas de Versionamento

- A branch `master` é a versão estável do sistema
- Desenvolva novas funcionalidades em uma branch diferente da `master` ex: 'feature/funcionalidade-nova'

Fluxo de Trabalho

1. Crie um branch para a nova funcionalidade

```
git checkout -b 'feature/funcionalidade-nova'
```

2. Depois de implementar as alterações, faça o commit e envie a branch para o GitHub:

```
git add .  
git commit -m 'Adicionando nova funcionalidade'  
git push origin 'feature/funcionalidade-nova'
```



Estabelecer rotina de desenvolvimento/versionamento

Casos de Uso

 INFO

Em breve

Detalhamento de Funcionalidades

! INFO

Em breve

Plano de Testes

! INFO

Em breve

Implantação

Configurando o portal SINPOM em um ambiente de produção.



Estratégia de Deploy

Verificar com o desenvolvedor a estratégia de deploy



Backup e Recuperação

Definir se vai ser adicionada a documentação online ou mantida em arquivo separado.

Estratégia de Deploy

WARNING

Verificar com o desenvolvedor a estratégia de deploy

Backup e Recuperação

WARNING

Definir se vai ser adicionada a documentação online ou mantida em arquivo separado. Verificar com o desenvolvedor a estratégia de backup e recuperação

Referências



Changelog

Adicionar historico de alterações no sistema

Changelog

INFO

Adicionar historico de alterações no sistema

Contribuição

- Siga o padrão de commits convencional
- Siga os padrões de código do Laravel e as melhores práticas de desenvolvimento do PHP

Autor

- Cap PM **Vailson** - github.com/vailson

Documentos Externos

WARNING

Documentos que devem ser mantidos fora da documentação técnica online.

- Manual do usuário
- Plano de Gerenciamento de Riscos
- Matriz de Responsabilidades (RACI)
- Política de Backup e Recuperação (Avaliar se será incluída ou não na seção monitoramento e manutenção)