

Manual de Configuração do Ambiente de Desenvolvimento

Equipe Nano
Aldísio Medeiros
Átila Camurça
Leandro Souza
Lucas Silva
Manuel Paulo
Péricles Henrique
Murilo Barata

25 de fevereiro de 2015

1 Sistema recomendado

Ubuntu 14.04 LTS

Para outros sistemas procure pelos pacotes com o nome igual ou semelhante.

1.1 Dependências

Execute o *script* abaixo para instalar as dependências:

```
sudo add-apt-repository ppa:webupd8team/java
sudo apt-get update
sudo apt-get install oracle-java7-installer
sudo apt-get install php5 apache2 php5-pgsql php5-mcrypt php5-intl
```

1.2 IDE

Escolha a IDE de sua preferência.

Recomendado: [NetBeans IDE](#).

Outras:

- [Atom](#)
- [Eclipse IDE](#)

1.3 PostgreSQL

<http://www.postgresql.org/download/linux/ubuntu/>

Crie o arquivo `/etc/apt/sources.list.d/pgdg.list`, adicione a linha para o repositório:

```
deb http://apt.postgresql.org/pub/repos/apt/ trusty-pgdg main
```

Importe a chave assinada do repositório, e atualize a lista de pacotes

```
wget --quiet -O - https://www.postgresql.org/media/keys/ACCC4CF8.asc | \
  sudo apt-key add -
sudo apt-get update
apt-get install postgresql-9.4
```

2 Configurar projeto local

Faça um *clone* do projeto usando ssh:

```
cd NetBeansProjects
git clone git@github.com:ifce-gp-20151/saa.git
```

Obs.: Você pode utilizar outro diretório para seu projeto, não é necessário ser NetBeansProjects.

Obs.: Se você ainda não configurou sua chave ssh acesse este [link](#).

Entre no diretório do projeto:

```
cd saa
```

Instale as dependências do projeto:

```
php composer.phar install
```

Crie o arquivo `local.php` em `config/autoload` com o conteúdo:

```
<?php

return array(
    'db' => array(
        'username' => 'postgres',
        'password' => '**secret**',
    )
);
```

Crie o arquivo `ocra-service-manager.local.php` em `config/autoload` com o conteúdo:

```
<?php

return array(
    'ocra_service_manager' => array(
        // Turn this on to disable dependencies view in Zend Developer Tools
        'logged_service_manager' => true,
    ),
);
```

Copie o arquivo `zenddevelopertools.local.php.dist`:

```
cp ./vendor/zendframework/zend-developer-tools/config/zenddevelopertools.local.p
./config/autoload/zenddevelopertools.local.php
```

Execute os scripts abaixo usando PgAdmin.

- ./docs/db/ddl.sql
- ./docs/db/dml.acl.sql

2.0.1 Já tinha o banco mas está diferente

Utilize os arquivos dentro de /docs/db/migrations/*.sql, execute um a um na ordem.

2.0.2 Configuração Apache 2 (2.2.22-1ubuntu1.7)

Crie o arquivo saa em /etc/apache2/sites-enabled com o conteúdo:

```
<VirtualHost *:80>
    ServerName saa.local
    DocumentRoot /path/to/saa/public

    <Directory /path/to/saa/public>
        DirectoryIndex index.php
        AllowOverride All
        Order allow,deny
        Allow from all
    </Directory>
</VirtualHost>
```

2.0.3 Apache 2 (2.4.7-1ubuntu4.1)

Crie o arquivo saa.conf em /etc/apache2/sites-enabled com o conteúdo:

```
<VirtualHost *:80>
    ServerName saa.local

    DocumentRoot /path/to/saa/public

    <Directory /path/to/saa/public/>
        Options Indexes FollowSymLinks
        AllowOverride All
        Require all granted
```

```

</Directory>

    ErrorLog /var/log/apache2/error.log
    CustomLog /var/log/apache2/access.log combined
</VirtualHost>

```

Habilite o `rewrite` (se já não estiver):

```
sudo a2enmod rewrite
```

Reinicie o `apache2`:

```
sudo service apache2 restart
```

Crie um virtual host, edite `/etc/hosts`, e adicione esta linha ao final:

```
127.0.1.1      saa.local
```

Ok, você deve estar pronto agora. Acesse o endereço <http://saa.local>

2.1 Armadilhas

2.1.1 Erro de sessão

<http://stackoverflow.com/questions/26377753/zend-authentication-storage-session->

O `ZendDeveloperTools` possui uma função que quebra o sistema. O link acima descreve o problema.

Solução Comente a linha 64 do arquivo:

```

./vendor/zendframework/zend-developer-tools/src/ZendDeveloperTools/Listener/E
public function attachShared(SharedEventManagerInterface $events)
{
    //$events->attach($this->identifiers, '*', array($this, 'onCollectEvent'), P
}

```

2.2 Criação de entidades (Entities) a partir do banco de dados

Obs.: esta parte já foi feita, é apenas para documentar.

```

./vendor/bin/doctrine-module orm:convert-mapping --filter="Usuario" \
--from-database annotation --namespace="Application\\Entity\\" \
module/Application/src

```

onde `Usuario` é o nome da tabela a ser gerada.

2.3 Criação dos *getters* e *setters*

Obs.: esta parte já foi feita, é apenas para documentar.

```
./vendor/bin/doctrine-module orm:generate-entities \  
--filter="Usuario" module/Application/src/
```

2.4 Links úteis

- <http://ocramius.github.io/presentations/doctrine2-zf2-introduction/#/1>