

## Manual de Desenvolvimento de Módulos para o SUPER.GOV.BR

Este Manual pode ser acessado pelo link: [http://bit.ly/SEI\\_Manual\\_Padrees\\_Desenvolvimento\\_Modulos](http://bit.ly/SEI_Manual_Padrees_Desenvolvimento_Modulos)

- [1. Observações Gerais](#)
- [2. Ambiente de Desenvolvimento Padrão Localhost](#)
  - [2.1. Introdução](#)
  - [2.2. Softwares Necessários](#)
  - [2.3. Configuração do Ambiente](#)
    - [2.3.1. Proxy para Acesso à Internet](#)
    - [2.3.2. BIOS do Computador deve Aceitar VM](#)
    - [2.3.3. Instalação da VM Localhost](#)
    - [2.3.4. Comandos Básicos do Vagrant](#)
  - [2.4. Snapshots](#)
  - [2.5. Acesso ao Banco de Dados](#)
  - [2.6. Comutação entre Tipos de Bancos de Dados](#)
  - [2.7. Execução de Scripts por Linha de Comando](#)
  - [2.8. Execução do Agendamento do SEI no crontab em localhost](#)
  - [2.9. Acesso ao MailCatcher](#)
  - [2.10. Acesso ao Solr](#)
  - [2.11. Acesso ao JOD Converter](#)
- [3. Padrão de Modelagem de Dados](#)
  - [3.1. Padrão Específico para Módulos](#)
  - [3.2. Padrão Geral do SEI](#)
- [4. Informações importantes sobre desenvolvimento dos módulos](#)
- [5. Padrão de Codificação PHP](#)
- [6. Padrão de script PHP de Instalação/Atualização de Módulos](#)
- [7. Debug de Ambiente](#)

### 1. Observações Gerais

- a. Este Manual tem por objetivo orientar o desenvolvimento de Módulos para o SEI, em aspectos práticos e padrões a serem seguidos.
- b. O [Manual Técnico de Desenvolvimento de Módulos do TRF4 para o SEI 4.0](#) deve ser seguido em todo o contexto apresentado no presente Manual, pois este apenas complementa-o, enquanto que o mencionado Manual Técnico do TRF4 descreve toda a API de integração de Módulos do SEI em aspectos técnicos.

## 2. Ambiente de Desenvolvimento Padrão Localhost

### 2.1. Introdução

- a. Neste tópico descreveremos os procedimentos para instalação do ambiente de desenvolvimento padrão do SEI em Localhost, de forma rápida e padronizada através do **Vagrant**.
  - i. O software Vagrant permite que seja provisionado de forma automatizada todo um ambiente funcional na máquina de trabalho do desenvolvedor, sem que seja necessária a instalação manual dos servidores de aplicação e de banco de dados.
  - ii. Outra vantagem é que esse ambiente é completamente provisionado utilizando **máquinas virtuais padronizadas**, necessitando apenas que o desenvolvedor configure sua IDE ou editor de texto preferido para iniciar o desenvolvimento.
- b. Gostaríamos de reforçar a necessidade que todo o desenvolvimento esteja alinhado com as diretrizes e padrões de interface, codificação PHP e modelagem de banco de dados utilizados pelo SEI, indicados no [Manual Técnico de Desenvolvimento de Módulos do TRF4 para o SEI 4.0](#) e no presente Manual.
- c. **Atenção:** Além dos requisitos de softwares necessários que serão abaixo indicados, o maior de todos os requisitos para o desenvolvimento de Módulos para o SEI, independente de utilização do Ambiente de Desenvolvimento Padrão Localhost, é o órgão possuir os códigos-fonte do SEI, que ocorre por meio de termo de cooperação com o Tribunal Regional Federal da 4ª Regional (TRF4), que é o desenvolvedor e proprietário do SEI. Os códigos-fonte do SEI **não são disponibilizados em conjunto** com o Ambiente de Desenvolvimento Padrão Localhost.

### 2.2. Softwares Necessários

- a. A configuração do Ambiente de Desenvolvimento Padrão Localhost exige a instalação dos softwares abaixo:
  - i. **Vagrant:** <https://www.vagrantup.com/>
  - ii. **VirtualBox e VirtualBox Extensions:** <https://www.virtualbox.org/>
    - 1. O VirtualBox Extensions é instalado dentro do VirtualBox, acessando o menu *File* > *Preferences* > *Extensions*.
  - iii. **Git:** <https://git-scm.com/>
    - 1. Sugerimos a instalação do Git com a funcionalidade *GitBash* ativado. Essa ativação é feita pelo *Wizard* de instalação do software e, caso necessário, ele permitirá o acesso direto ao prompt de comando das Máquinas Virtuais Linux.
    - 2. Para o desenvolvedor que preferir utilizar o Git por meio de software que interface gráfica, aconselhamos o uso do [TortoiseGit](#).
- b. Caso tenha interesse de obter conhecimento mais detalhado sobre o funcionamento do Vagrant e demais softwares acima para o provisionamento automatizado de ambiente padrão de desenvolvimento, sugerimos uma breve leitura de suas documentações ou dos materiais abaixo:
  - i. [Começando com o Vagrant](#)
  - ii. [Usando o Vagrant como Ambiente de Desenvolvimento](#)
  - iii. [Vagrant, como usar para desenvolvimento PHP](#)
  - iv. [Desenvolvimento PHP com Vagrant](#)
- c. Em resumo, o Vagrant é uma ferramenta que permite a construção de Máquinas Virtuais (VMs) para desenvolvimento sem que seja necessária a instalação de todos os componentes da infraestrutura (banco de dados, bibliotecas, servidores web, etc) na máquina do desenvolvedor. Entre as vantagens em sua utilização estão:
  - i. Rápido início do projeto de desenvolvimento ou testes por parte do desenvolvedor.
  - ii. Padronização do ambiente de desenvolvimento (versões de bibliotecas, banco de dados, servidores de aplicação, etc).
  - iii. Foco nas tarefas de desenvolvimento e não em questões relacionadas à configuração de ambiente.
- d. O Vagrant trabalha com o conceito de Box, basicamente uma "imagem/iso", para criar a instância da VM, onde é definido o sistema operacional e os demais componentes da solução.
  - i. Para o projeto do SEI, foi disponibilizado livremente uma Box contendo todo o ambiente funcional do SEI e seus componentes (SEI, SIP, JOD Converter, Apache Solr e Memcached), tirando a necessidade do desenvolvedor realizar toda a configuração do sistema, podendo focar apenas no desenvolvimento.

## 2.3. Configuração do Ambiente

### 2.3.1. Proxy para Acesso à Internet

- a. Se o computador (*desktop ou notebook*) onde vai ser instalado o Ambiente de Desenvolvimento Padrão Localhost depender de um proxy para acesso à Internet, será necessário configurar o proxy como variável de ambiente:

i. No Windows:

1. Acesse o prompt de comando do Windows e insira os seguintes comandos, trocando os trechos em vermelho para os dados corretos:

```
set http_proxy=http://<endereço_do_proxy>:8080
set https_proxy=https://<endereço_do_proxy>:8080
```

2. Se o proxy exigir autenticação será necessário incluir o nome e senha do usuário, conforme exemplo abaixo:

```
set http_proxy=http://<usuario>:<senha>@<endereço_do_proxy>:8080
set https_proxy=http://<usuario>:<senha>@<endereço_do_proxy>:8080
```

**Observação:** Esta configuração também pode ser realizada visualmente através da interface gráfica do Windows para gerenciamento de variáveis de ambiente. Ela fica localizada em Windows >> Meu Computador >> Propriedades >> Configurações Avançadas >> Variáveis de Ambiente.

ii. No Linux/Mac

1. Acesse o prompt de comando do Linux ou MAC e insira os seguintes comandos, trocando os trechos em vermelho para os dados corretos:

```
export http_proxy=http://<endereço_do_proxy>:8080
export https_proxy=https://<endereço_do_proxy>:8080
```

2. Se o proxy exigir autenticação será necessário incluir o nome e senha do usuário, conforme exemplo abaixo:

```
export http_proxy=http://<usuario>:<senha>@<endereço_do_proxy>:8080
export https_proxy=http://<usuario>:<senha>@<endereço_do_proxy>:8080
```

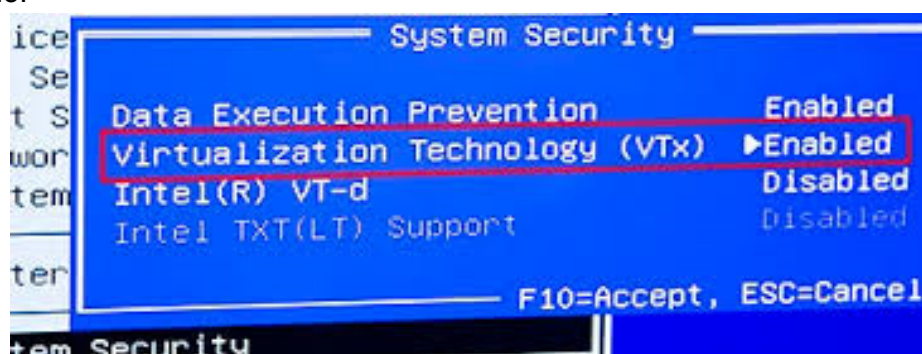
PS: Para persistir tal configuração, as variáveis de ambiente podem ser armazenadas no arquivo de perfil do usuário através do comando dos seguintes comandos:

```
echo "export http_proxy=http://<endereço_do_proxy>:8080" >> $HOME/.bashrc
echo "export https_proxy=https://<endereço_do_proxy>:8080" >> $HOME/.bashrc
```

### 2.3.2. BIOS do Computador deve Aceitar VM

- a. A BIOS do computador (*desktop ou notebook*) onde vai ser instalado o Ambiente de Desenvolvimento Padrão Localhost deve ter a opção de virtualização habilitada.

- b. Vide exemplo:



### 2.3.3. Instalação da VM Localhost

- a. Via prompt de comandos, navegue até o diretório onde estão os códigos fonte do SEI. O diretório é o mesmo disponibilizado para instalação e dentro dele deve conter as seguintes pastas de código:

```
/infra  
/sei  
/sip
```

- b. Esses arquivos serão compartilhados para dentro da VM criada pelo Vagrant para ativação do ambiente de desenvolvimento.
- c. As alterações feitas diretamente nos arquivos PHP durante o desenvolvimento refletirão de forma automática no ambiente que estará disponível em <http://localhost:8000/sei>
- d. Para baixar o arquivo de configuração inicial da Box a ser instalada pelo Vagrant, no diretório raiz dos códigos fonte do SEI acima citado, execute o comando:

```
vagrant init processoeletronico/sei-4.0
```

- i. O comando acima criará um arquivo de configuração inicial para o vagrant chamado "Vagrantfile", sem extensão, no diretório raiz dos códigos fonte do SEI acima citado, contendo a referência para a Box do SEI completamente configurado, que será baixado somente no próximo passo.
- e. Para iniciar o download de fato da Box do Ambiente de Desenvolvimento Padrão Localhost e iniciar a instalação da VM, no diretório raiz dos códigos fonte do SEI execute o comando:

```
vagrant up
```

- i. Em computadores com sistema operacional Linux ou Mac é necessário executar o comando acima como usuário *root*.
- ii. É normal que a primeira execução desse comando demore vários minutos para ser concluída, pois a Box, com cerca de 4 GB, será baixada para o computador.
- iii. Com o fim da transferência, o Ambiente de Desenvolvimento Padrão Localhost estará disponível em minutos.
- iv. Após o primeiro provisionamento, o ambiente poderá ser destruído e recriado rapidamente, já que o Vagrant já baixou e armazenou a Box localmente.
- v. O comando acima é utilizado no primeiro provisionamento (que demora mais, por envolver o download da Box) e sempre que precisar iniciar a VM novamente, seja após parar a execução da VM ou quando precisar destruí-la.
- vi. Ao final da inicialização do ambiente, sempre deve ser apresentada a seguinte mensagem, que indica que todos os serviços do SEI estão operando na VM:

```
..
==> default: =====
==> default: INICIALIZAÇÃO DO AMBIENTE DE DESENVOLVIMENTO FINALIZADA COM SUCESSO!
==> default: =====
==> default:
==> default: = Endereços de Acesso à Aplicação =
==> default: SEI ..... http://localhost:8000/sei
==> default: SIP ..... http://localhost:8000/sip
==> default:
==> default: Outros endereços úteis:
==> default: Acesso de Usuário Externo ....
http://localhost:8000/sei/controlador_externo.php?acao=usuario_externo_logar&id_orgao_acesso_externo=0
==> default: Autenticidade de Documentos ...
http://localhost:8000/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0
==> default: Publicações Eletrônicas .....
http://localhost:8000/sei/publicacoes/controlador_publicacoes.php?acao=publicacao_pesquisar&id_orgao_publicacao=0
==> default: WSDL de integração do SEI ..... http://localhost:8000/sei/ws/SeiWS.php
==> default:
==> default: = Outros Serviços =
==> default: Solr ..... http://localhost:8083/solr
==> default: MailCatcher ..... http://localhost:1080
==> default: Mysql ..... localhost:3306
==> default: Oracle ..... localhost:1521
==> default: SQLServer ..... localhost:1433
==> default:
==> default: = Comandos Úteis =
==> default: vagrant up ..... Inicializar ambiente do SEI
==> default: vagrant halt ..... Desligar ambiente
```

```

==> default: vagrant destroy ----- Destruir ambiente e base de testes
==> default: vagrant ssh ----- Acessar máquina virtual
==> default: vagrant status ----- Verificar situação atual do ambiente
==> default:
==> default: Utilize o parâmetro ' provision-with ' para alterar o banco de dados padrão:
==> default:
==> default: vagrant up --provision-with [mysql|oracle|sqlserver]
==> default: --ou--
==> default: vagrant provision --provision-with [mysql|oracle|sqlserver]
==> default:
==> default: - Debug -----
==> default: PHP xDebug 3
==> default: Porta: 9003
...

```

f. Com a inicialização com sucesso da VM, conforme acima, agora já é possível acessar o SEI e o SIP:

- i. Para acesso SEI, em um browser, acesse o endereço: <http://localhost:8000/sei>
  1. Utilizar o login “teste” e a senha “teste”
- ii. Para acesso SIP, em um browser, acesse o endereço: <http://localhost:8000/sip>
  1. Utilizar o login “teste” e a senha “teste”
- iii. Também é possível acessar o **PHP Info** para visualizar as configurações do ambiente SEI Localhost: <http://localhost:8000/info.php>
- iv. **Atenção:** O SEI e SIP rodando nos dois endereços locais acima são os correspondentes ao diretório raiz dos códigos fonte do SEI, onde foi executado o comando **vagrant up**. As alterações feitas nos mencionados códigos fonte são replicadas instantaneamente na aplicação, através de um simples *refresh* no browser.

#### 2.3.4. Comandos Básicos do Vagrant

- a. Além do comando “**vagrant init processoeletronico/sei-4.0**” acima indicado para a criação do “**Vagrantfile**”, outros comandos básicos do Vagrant são necessários para a inicialização, pausa, destruição e atualização do Ambiente de Desenvolvimento Padrão Localhost.
  - i. Os comandos devem sempre ser executados por meio do prompt de comandos no diretório raiz dos códigos fonte do SEI, onde está localizado o arquivo “**Vagrantfile**”.
- b. Segue abaixo a descrição de cada comando necessário à operação básica da VM de desenvolvimento:
  - i. Iniciar a VM de desenvolvimento, que, conforme explicado mais acima, faz o provisionamento inicial caso seja a primeira execução (envolve o download completo da Box) e inicia a VM sempre que precisar retomar o trabalho de desenvolvimento ou recria a VM após ser destruída.
  - ii. Para subir a box com o banco MySQL:

**vagrant up --provision-with mysql**
  - iii. Para subir a box com o banco Oracle

**vagrant up --provision-with oracle**
  - iv. Para subir a box com o banco SQLServer

**vagrant up --provision-with sqlserver**
  - v. Desligar a VM de desenvolvimento:

**vagrant halt**
  - vi. Reiniciar a VM de desenvolvimento:

**vagrant reload**
  - vii. Destruir a VM de desenvolvimento, apagando todas as informações persistidas no banco de dados e Snapshots que tenham sido criados no VirtualBox. Após destruir a VM, em seguida é possível reconstruir novamente com o comando “**vagrant up**”, conforme acima indicado:

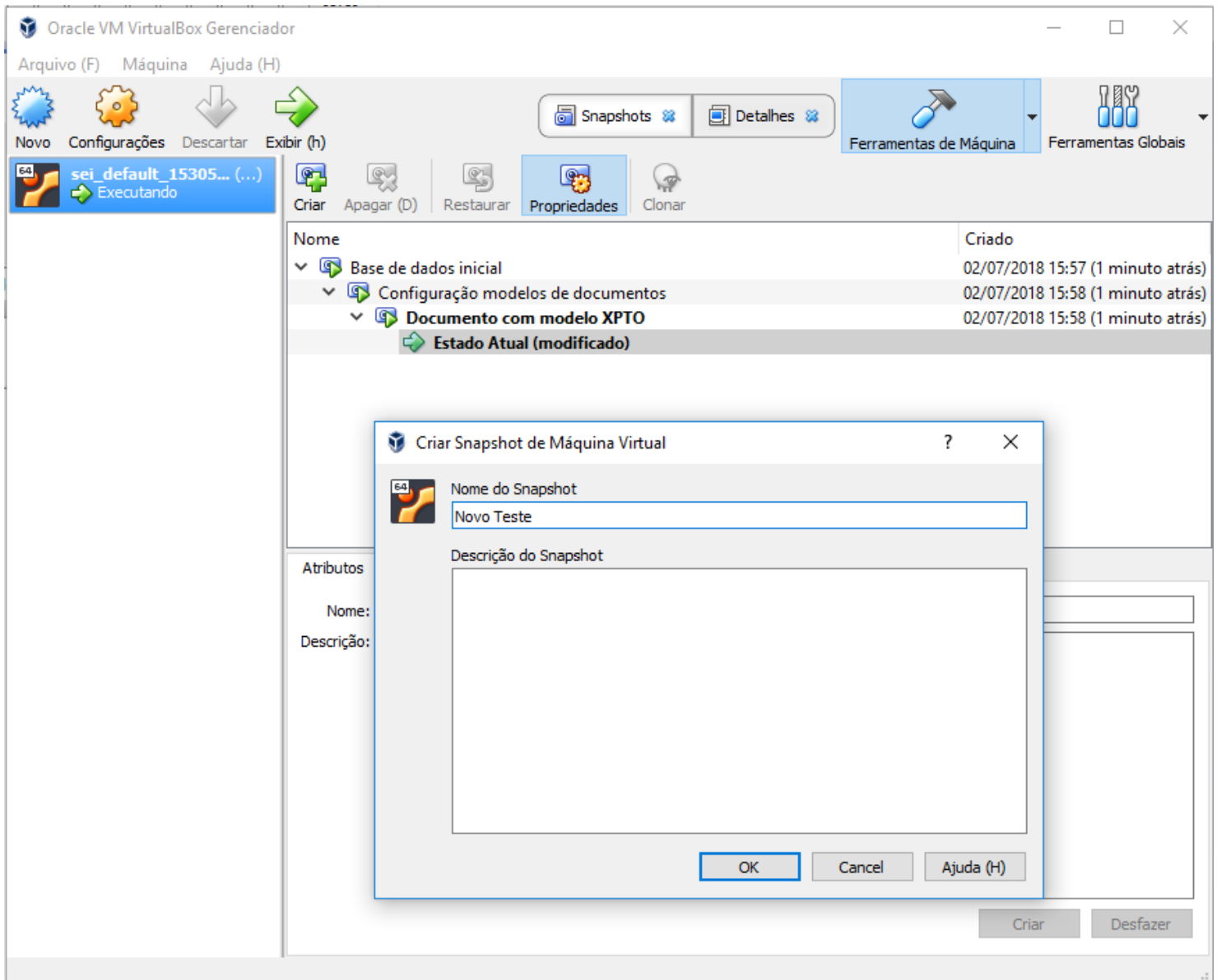
**vagrant destroy**
  - viii. Atualizar a VM de desenvolvimento para versão mais recente:

**vagrant box update**

#### 2.4. Snapshots

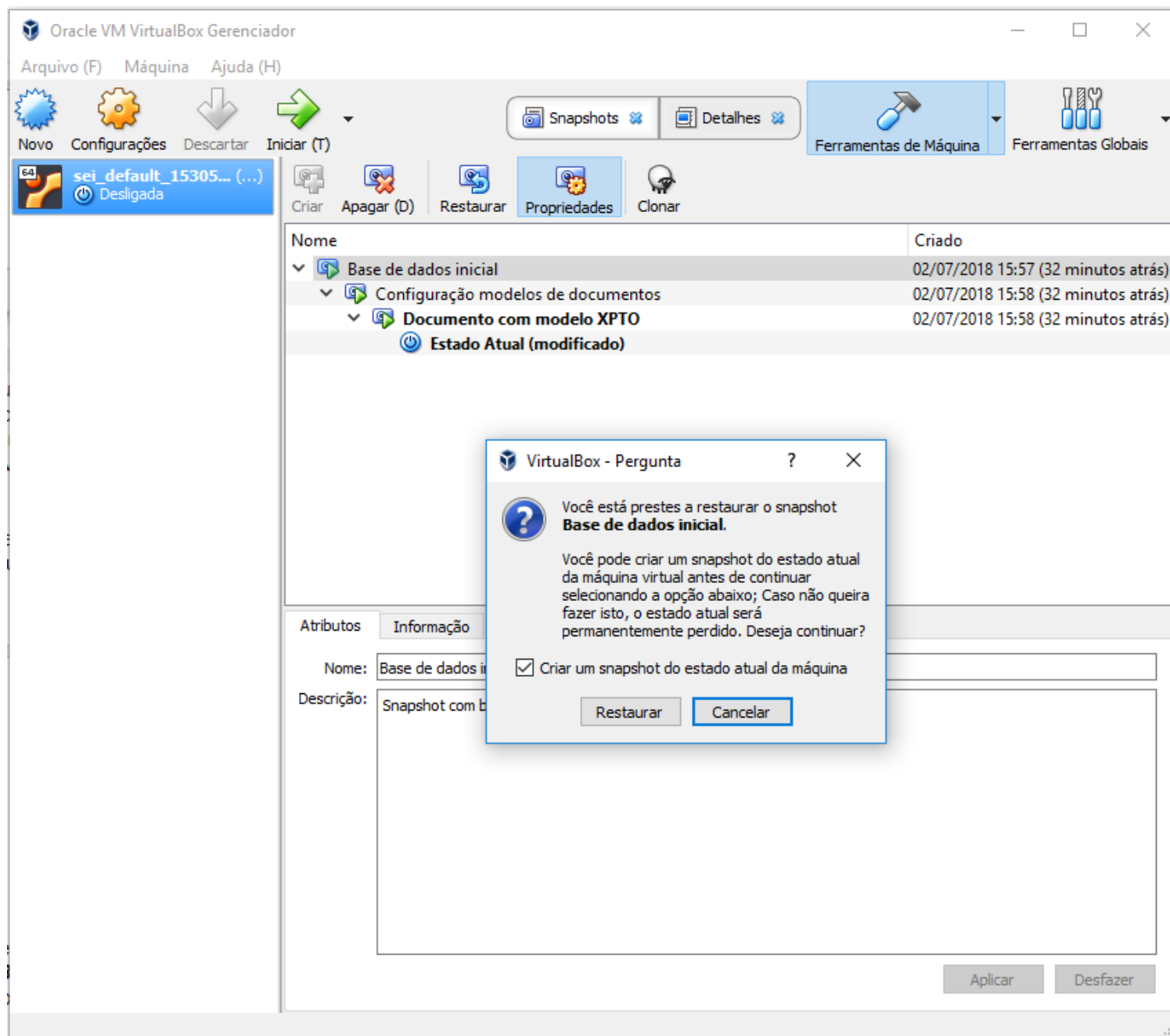
- a. Snapshot é uma funcionalidade do **VirtualBox**, que permite ao usuário criar versões do seu ambiente de desenvolvimento para posteriormente restaurar determinada versão mais rapidamente.
  - i. No contexto do desenvolvimento do SEI, que implica em poucas alterações nas configurações dos serviços que rodam na VM, é mais utilizado para voltar às versões anteriores do banco de dados, com massa de dados específica para determinado teste.
- b. Para criar um Snapshot no Virtual Box, abra o mencionado software e siga os seguintes passos:
  - i. Na coluna da esquerda, selecione a instância da máquina virtual em uso no desenvolvimento;

- ii. ~~Acesse a aba Snapshots localizada no canto superior direito;~~
- iii. ~~Clique no botão “Criar” e informe um nome descritivo para identificar a nova versão gerada;~~
- iv. ~~Com isto, agora estará salvo uma nova versão do estado atual da máquina virtual, que poderá ser restaurada a qualquer momento.~~



- e. ~~Para restaurar um Snapshot criado anteriormente, siga os seguintes passos:~~
  - i. ~~Na coluna da esquerda, selecione a instância da máquina virtual em uso no desenvolvimento;~~
  - ii. ~~Desligue a instância da máquina virtual, clicando com o botão direito do mouse e selecionando Fechar > Desligar. A restauração somente é possível com a VM desligada;~~
  - iii. ~~Acesse a aba Snapshots localizada no canto superior direito do VirtualBox;~~
  - iv. ~~Selecione a versão da máquina que se deseja restaurar na árvore de versões;~~
  - v. ~~Clique no botão “Restaurar” e confirme a operação;~~
  - vi. ~~Agora a instância da VM estará restaurada à versão selecionada, para novos testes.~~





## 2.5. Acesso ao Banco de Dados

### MySQL

- a. O componente **mysql**, iniciado pela VM, se refere ao serviço de banco de dados do MySQL, que estará acessível na máquina local por meio da porta **3306**:
  - i. Este componente pode ser acessado utilizando a SSH do bach, com o Login "**root**" e Senha "**root**".
  - ii. Exemplo: **mysql -h 127.0.0.1 -u root -p sei**
- b. A instância de banco pode ser acessada por qualquer ferramenta de conexão à banco de dados MySQL, como o **MySQL Workbench**, com os dados abaixo:
  - i. Hostname: **127.0.0.1 ou localhost**
  - ii. Port: **3306**
  - iii. Root Username: **root**
  - iv. Root Password: **P@ssword**
- c. Os usuários de banco do SEI e SIP são os indicado abaixo:
  - i. Banco do SEI: Login "**sei\_user**" e Senha "**sei\_user**"
  - ii. Banco do SIP: Login "**sip\_user**" e Senha "**sip\_user**"

### SQL Server

- d. A instância de banco pode ser acessada por qualquer ferramenta de conexão à banco de dados SQL Server, como o **SQL Server Management Studio (SSMS)**, com os dados abaixo:
  - i. Hostname: **127.0.0.1 ou localhost**

- ii. — Port: **1433**
- iii. — SA Username: **sa**
- iv. — SA Password: **P@ssword**
- e. — Os usuários de banco do SEI e SIP são os indicado abaixo:
  - i. — Banco do SEI: Login “**sei\_user**” e Senha “**sei\_user**”
  - ii. — Banco do SIP: Login “**sip\_user**” e Senha “**sip\_user**”

#### **Oracle**

- f. — A instância de banco pode ser acessada por qualquer ferramenta de conexão à banco de dados Oracle, como o **Data Azure Studio** com os dados abaixo:
  - i. — Hostname: **127.0.0.1 ou localhost**
  - ii. — Port: **1521**
  - iii. — SID: **xe**
  - iv. — Sys Username: **sys as sysdba**
  - v. — Sys Password: **oracle**
- g. — Os usuários de banco do SEI e SIP são os indicado abaixo:
  - i. — Banco do SEI: Login “**sei**” e Senha “**sei\_user**”
  - ii. — Banco do SIP: Login “**sip**” e Senha “**sip\_user**”

## 2.6. Comutação entre Tipos de Bancos de Dados

- a. — A partir do SEI Localhost 4.0 a escolha do tipo de banco usado ocorre junto com o comando “**vagrant up**”, conforme os comandos abaixo:
  - a. — Para subir a box com o banco MySQL:
   
**vagrant up —provision-with-mysql**
  - b. — Para subir a box com o banco Oracle
   
**vagrant up —provision-with-oracle**
  - c. — Para subir a box com o banco SQLServer
   
**vagrant up —provision-with-sqlserver**

## 2.7. Execução de Scripts por Linha de Comando

Para execução de scripts do SEI e SIP via linha de comando no SEI Localhost deve seguir o passo a passo abaixo:

- a. — Subir a máquina com o comando:
   
**vagrant up**
- b. — Na pasta principal do Localhost acessar o bash e depois digitar o comando:
   
**vagrant ssh**
  - i. — Se solicitar login e senha, informar: **vagrant**
  - ii. — Caso apresente a mensagem “**vagrant@127.0.0.1: Permission denied (publickey,gssapi-keyex,gssapi-with-mic)**”, usar um dos dois comandos a seguir:
    - 1. — **set VAGRANT\_PREFER\_SYSTEM\_BIN=0** ou
    - 2. — **\$env:VAGRANT\_PREFER\_SYSTEM\_BIN="0"**
- c. — Após entrar na máquina, digitar o comando a seguir para entrar no container principal do SEI e SIP:
   
**sudo docker exec -it httpd /bin/bash**
- d. — Após entrar no container principal do SEI e SIP, podem ser executados os scripts PHP de interesse. Exemplos:

```
/usr/bin/php -c /etc/php.ini /opt/sei/scripts/atualizar_sequencias.php 2>&1 >
atualizacao_sequencias_sei.log
```

```
/usr/bin/php -c /etc/php.ini /opt/sip/scripts/atualizar_sequencias.php 2>&1 >
atualizacao_sequencias_sip.log
```

```
/usr/bin/php -c /etc/php.ini /opt/sei/scripts/sei_atualizar-versao_modulo_peticaoamento.php 2>&1 >
atualizacao_modulo_peticaoamento_sei.log
```

```
/usr/bin/php -c /etc/php.ini /opt/sip/scripts/sip_atualizar-versao_modulo_peticaoamento.php 2>&1 >
atualizacao_modulo_peticaoamento_sip.log
```

- e. — Para verificar a saída da execução durante a execução, digitar: **tail -f nomedoarquivodelog.log**



i. Exemplos:

```
tail -f atualizacao_sequencias_sei.log  
tail -f atualizacao_sequencias_sip.log
```

```
tail -f atualizacao_modulo_peticaoamento_sei.log  
tail -f atualizacao_modulo_peticaoamento_sip.log
```

f. Para verificar a saída da execução finalizada com a definição da quantidade de linhas que deseja visualizar, digitar: **tail -n [número qtd linhas] nome do arquivo de log.log**

i. Exemplos:

```
tail -n 500 atualizacao_sequencias_sei.log  
tail -n 1000 atualizacao_sequencias_sip.log
```

```
tail -n 500 atualizacao_modulo_peticaoamento_sei.log  
tail -n 1000 atualizacao_modulo_peticaoamento_sip.log
```

g. Para sair do arquivo de log, digitar:

**Ctrl + e**

## 2.8. Execução do Agendamento do SEI no crontab em localhost

Para execução do agendamento central do SEI e SIP via linha de comando no SEI Localhost deve seguir o passo a passo abaixo:

a. Subir a máquina com o comando:

```
vagrant up
```

b. Na pasta principal do Localhost acessar o bash e depois digitar o comando:

```
vagrant ssh
```

i. Se solicitar login e senha, informar: **vagrant**

ii. Caso apresente a mensagem **"vagrant@127.0.0.1: Permission denied (publickey,gssapi-keyex,gssapi-with-mic)"**, usar um dos dois comandos a seguir:

1. **set VAGRANT\_PREFER\_SYSTEM\_BIN=0** ou
2. **\$env:VAGRANT\_PREFER\_SYSTEM\_BIN="0"**

c. Após entrar na máquina, digitar o comando a seguir para entrar no container principal do SEI e SIP:

```
sudo docker exec -it httpd /bin/bash
```

d. Após entrar no container principal do SEI e SIP, executar o comando crontab e ao abrir o editor incluir as seguintes linhas:

```
/usr/bin/php -c /etc/php.ini /opt/sei/scripts/AgendamentoTarefaSEI.php 2>&1 >>  
infra_agendamento_sei.log
```

```
/usr/bin/php -c /etc/php.ini /opt/sip/scripts/AgendamentoTarefaSIP.php 2>&1 >>  
infra_agendamento_sip.log
```

## 2.9. Acesso ao MailCatcher

a. O SEI Localhost possui um serviço interno de e-mail, o **MailCatcher**, para viabilizar testes de disparo de e-mail e conferência do e-mail enviado, que fica contido na VM e não é enviado de fato para a Internet.

b. O MailCatcher pode ser acessado diretamente pelo endereço: <http://localhost:1080>

## 2.10. Acesso ao Solr

a. O Solr instalado na VM pode ser acessado diretamente pelo endereço: <http://localhost:8983/solr>

## 2.11. Acesso ao JOD Converter

a. O JOD Converter instalado na VM pode ser acessado diretamente pelo endereço: <http://localhost:8080>

### 3. Padrão de Modelagem de Dados

#### 3.1. Padrão Específico para Módulos

- a. Este subtópico define padrões específicos que todo Módulo desenvolvimento para o SEI deve seguir para sua adequada modelagem de dados, especialmente para não existir confusão entre as tabelas do *core* do SEI com as tabelas de Módulos.
  - i. Ressalvados os padrões específicos aqui definidos, a modelagem de dados de Módulo também deve seguir o padrão de modelagem geral do SEI documentado no subtópico [Padrão Geral do SEI](#), mais abaixo.
  - ii. Importante destacar que para banco MySQL sempre deve utilizar a engine “InnoDB”.
- b. Os nomes de tabelas de Módulos sempre devem ser iniciados com os **sufixos** abaixo, antes do nome funcional da tabela em questão. Vide abaixo:

**md\_siglamd\_funcaotabela**

ou

**md\_siglamd\_adm\_funcaotabela**

ou

**md\_siglamd\_rel\_funcaotabela**

Onde:

**md** = sufixo que indica se tratar de uma tabela de Módulo;

**siglamd** = sufixo com abreviação que identifique o Módulo correspondente, devendo ser limitado a **3 caracteres** (Exemplos: “**md\_pet\_**” identifica as tabelas do Módulo de Peticionamento e Intimação Eletrônicos; “**md\_ri\_**” identifica as tabelas do Módulo de Relacionamento Institucional);

**adm** = sufixo que indica se tratar de uma tabela relativa a funcionalidades de Administração do Módulo (utilizar somente nestes casos);

**rel** = sufixo que indica se tratar de uma tabela de relacionamento;

**funcaotabela** = nome funcional da tabela em si, que identifica sua finalidade no contexto do Módulo, sempre respeitando o limite total de 30 caracteres.

- c. Em razão de especificidades do “infra\_php” do SEI, as tabelas que controlam o sequencial de auto incremento de outras tabelas recebem automaticamente a adição do sufixo “**seq\_**”, repetindo o nome da tabela principal correspondente no restante do nome.
  - i. Exemplo: “md\_pet\_adm\_hipotese\_legal” e “seq\_md\_pet\_adm\_hipotese\_legal” - “**md**” indica que é uma tabela de Módulo; “**pet**” identifica que é do Módulo de Peticionamento e Intimação Eletrônicos; “**adm**” indica que é uma tabela relativa a funcionalidade de Administração; “**hipotese\_legal**” identifica a função desta tabela; e “**seq\_**” identifica que é uma tabela de controle de sequencial de auto incremento correspondente.
- d. A identificação do Módulo no nome das tabelas (**siglamd**) deve ser o menor possível (no máximo 3 caracteres), tendo em vista o limite geral de até 30 caracteres para nome de qualquer objeto de banco, em razão de limitação do banco de dados Oracle no qual o SEI também funciona.
  - i. Em razão da mencionada limitação e do padrão de identificação de tabelas de controle de sequenciais de auto incremento, **aconselha-se sempre a limitar o nome total das tabelas e colunas a 26 caracteres**.
- e. Todas as colunas deverão possuir uma descrição comercial em seu “comment”, informando sua finalidade funcional e, caso existam estados distintos, também as suas opções e finalidades.
  - i. Esta prática visa a dicionarização do banco de dados do SEI.

#### 3.2. Padrão Geral do SEI

- a. A primeira regra na hora de pensar em nome de tabelas e de campos, é usar nomes que façam sentido para o que irão armazenar e sempre lembrar que o máximo de caracteres para o banco de dados Oracle é de 30 caracteres, sendo assim, sempre usar no máximo 25 caracteres por causa dos prefixos usados **pk\_**, **fk\_**, **ixx\_** e também a tabela de seq que é criada que sempre usa **seq\_** com isso, não terá nenhum problema na hora das execuções dos scripts.
- b. Regras de nomenclatura aplicáveis a todos os elementos (tabelas, colunas, índices, etc) do modelo de dados:
  - i. usar somente letras, números e o caractere sublinhado;
  - ii. utilizar apenas letras minúsculas;

- iii. palavras internas devem ser separadas pelo caractere sublinhado;
- iv. suprimir as preposições;
- v. utilizar apenas palavras no singular.
- c. Tabelas
  - i. Não utilizar verbos para designar nomes de tabelas, priorizar o uso de substantivos:
    - 1. Exemplo: md\_ri\_resposta\_reiteracao
  - ii. Para tabelas que implementam relacionamentos (n x n) utilizar o prefixo rel seguido dos nomes das tabelas envolvidas separados por sublinhado:
    - 1. Exemplo: md\_ri\_rel\_processo\_parte
- d. Se o relacionamento expressar um conceito forte do sistema então o nome deste conceito pode ser utilizado:
  - i. Exemplo: md\_ri\_administrador\_sistema (e não rel\_usuario\_sistema)
- e. Tabelas que representam conjuntos de valores relacionados a uma determinada entidade, devem conter um prefixo indicativo do tipo do conjunto seguido do nome da entidade:
  - i. Exemplo: md\_ri\_tipo\_dependente
- f. Colunas:
  - i. Para chaves primárias sequenciais utilizar o prefixo id seguido do nome da tabela:
    - 1. Exemplo: id\_pedido
  - ii. Para chaves primárias naturais (simples ou compostas) utilizar o prefixo id, o nome da tabela e o nome da coluna:
    - 1. Exemplo:
      - a. Simples: id\_servidor\_cpf
      - b. Composta: id\_pedido\_codigo id\_pedido\_data
  - iii. Para chaves primárias que fazem uso de chaves estrangeiras, manter a nomenclatura da chave estrangeira igual a da chave primária de origem. Aconselha-se que as chaves estrangeiras sejam posicionadas antes dos demais campos que compõem a chave primária, isto permite uma melhor identificação visual das chaves estrangeiras que compõem uma chave primária. Chave primária da tabela dependente:
    - 1. Exemplo: id\_servidor id\_dependente
  - iv. Chave primária da tabela tarefa (dependente de processo e atividade):
    - 1. Exemplo: id\_processo id\_atividade id\_tarefa
  - v. Chaves primárias cujo conteúdo é numérico devem possuir um tipo de dado numérico e sem decimais.
  - vi. Os nomes das chaves estrangeiras devem ser iguais ao nome da chave primária de origem.
    - 1. i. Prefixos recomendados:
      - a. **sin\_** campo sinalizador que aceita apenas os valores S ou N
      - b. **sta\_** status multi-valorado. Ex.: S=Servidor, M=Magistrado, T=todos
      - c. **dta\_** data
      - d. **dth\_** data/hora
      - e. **din\_** dinheiro
  - vii. Para exclusão lógica utilizar o nome de campo:
    - 1. Exemplo: sin\_ativo
- g. Tipo de Dado:
  - i. Quanto ao tipo de dado usado na representação, aconselha-se, para maior portabilidade, a escolha de um dos tipos principais definidos pelo padrão SQL-99 (ou SQL3).
- h. Restrições de Integridade
  - i. Chave Primária: Utilizar o prefixo pk seguido do nome da entidade.
    - 1. Exemplo: pk\_servidor
  - ii. Chave Alternativa: Utilizar o prefixo ak (1-9) seguido do nome da entidade.
    - 1. Exemplo: ak1\_servidor
  - iii. Chave Estrangeira: Utilizar o prefixo fk (1-9) seguido do nome da entidade.
    - 1. Exemplo: fk1\_servidor
- i. Índice: Para índices utilizar o prefixo i(01-99) seguido do nome da entidade.
  - i. Exemplo:
    - i01\_processo
    - i02\_processo
    - .
    - .
    - i99\_processo
- j. Sequência: Utilizar o prefixo seq seguido do nome da Entidade ao qual a sequência atende.

- i. Exemplo: seq\_dependente
  - 1. Observação: O uso deste recurso, bem como de campos do tipo auto-incremento, não é recomendado pois afeta negativamente a portabilidade do sistema.
- k. Aconselha-se a criação de uma tabela auxiliar para geração de sequências, com estrutura igual a descrita abaixo:
  - i. Exemplos:
    - 1. No caso do MySql
 

```
CREATE TABLE seq_dependente (
  id int(11) NOT NULL AUTO_INCREMENT,
  campo char(1) DEFAULT NULL,
  PRIMARY KEY (id)
);
```
    - 2. No caso do SqlServer
 

```
CREATE TABLE seq_dependente (
  id bigint identity(1,1),
  campo char(1) null)

```
    - 3. No caso do ORACLE
 

```
CREATE SEQUENCE seq_dependente;
```

## 4. Informações importantes sobre desenvolvimento dos módulos

O TRF4 passou algumas informações importantes a serem observadas no desenvolvimento dos módulos do SEI.

- a. Quando for executar uma transação de inserção de informações no banco de dados que utiliza a API de GerarProcedimento. Deve-se criar ou renomear a **FuncaoExistenteControlado** para **novaFuncaoInterna**. E então reorganizar conforme exemplo abaixo. Duas coisas são importantes para se observar, que nessa refatoração deve se retirar a função que envia os e-mail de dentro desta transação e deve gerar um acúmulo de indexação do Solr para ser disparado após a conclusão da transação com o banco de dados evitando assim gargalos e lentidão no código. Usaremos o exemplo abaixo uma função chamada **GerarProcedimentoControlado** no módulo do Peticionamento:

- i. Código antes:

```
protected function gerarProcedimentoControlado( $arrParametros ){
..... Tudo era feito aqui dentro, gerarProcedimento, envio de e-mails e indexação do Solar
}
```

- ii. Código depois:

```
protected function gerarProcedimentoControlado( $arrParametros )
{
    //Coloca o Solr para acumular a indexação
    FeedSEIProtocolos::getInstance()->setBolAcumularFeeds(true);

    //Chama a função que irão fazer todo o procedimento de inclusão das informações em banco
    $retorno = $this->gerarProcedimentoInterno($arrParametros);

    //Desativa a função de acumular a indexação e solicita a execução
    FeedSEIProtocolos::getInstance()->setBolAcumularFeeds(false);
    FeedSEIProtocolos::getInstance()->indexarFeeds();

    try {
        //Chama a função que irá disparar os e-mails pós inclusão de tudo em banco
        $emailMdPetEmailNotificacaoRN = new MdPetEmailNotificacaoRN();
        $emailMdPetEmailNotificacaoRN->notificacaoPeticionamentoExterno( $retorno['parametrosEmail'] );
    } catch( Exception $exEmail ){

    }

    return $retorno['parametrosRecibo'];
}
```

```
protected function gerarProcedimentoInterno( $arrParametros ){  
    .....  
}
```

- b. Outra coisa importante para os autocompletes de Contato, é ao invés de usar o campo “nome”, “e-mail”, “cpf” e “cnpj” para utilizar o LIKE do banco de dados nessas colunas em específico na tabela de contato ou de usuário, melhor é utilizar o campo **idx\_contato com a função do core do SEI a seguir**, que tira todos os caracteres indevidos para a pesquisa nesse campo de forma mais assertiva e eficaz
- ```
InfraString::prepararIndexacaoInfraString::prepararIndexacao($post['txtUsuario'],true);
```

## 5. Padrão de Codificação PHP

Este tópico será inicialmente composto pelo conteúdo a seguir:

<https://softwarepublico.gov.br/social/sei/manuais/padrao-de-codificacao-php>

A codificação a ser seguida deve ser a padrão para o PHP 7, conforme link a seguir: <https://www.php-fig.org/psr/>

Além do padrão técnico de codificação PHP acima, devem ser observados os padrões adicionais abaixo:

- a. ddd
- b. ffff

## 6. Especificidades no Desenvolvimento de módulos

### 6.1. Componente de Modal no SEI 4.0

- a. O componente ...
- b. dd ....

## 6. Padrão de script PHP de Instalação/Atualização de Módulos

Faltando escrever:

Faltando escrever:

- a. Importante também conhecer cada função constante no **InfraBD.php** do SEI.

## 7. Debug de Ambiente

Para diagnosticar problemas de desempenho no ambiente, tanto em produção como em homologação, pode ser utilizado as ferramentas do xdebug profile para gerar dados para análise de lentidão e gargalos no sistema.

Para gerar os dados mencionados anteriormente, deverão ser realizados os seguintes passos:

1. Selecionar um servidor de aplicação entre aqueles configurados no balanceador de carga. Este servidor será o responsável por coletar os logs de profile.
2. Remover do balanceador o servidor selecionado anteriormente para fazer a configuração da ferramenta de profile.
3. Instalar o componente XDebug do PHP e ativar as configurações de profile.

**Utilizando o gerenciador de pacotes yum:**

```
> sudo yum install php-pecl-xdebug.x86_64
```

**Utilizando o gerenciador de pacotes dnf:**

```
> sudo dnf install php-xdebug
```

**Utilizando o gerenciador de pacotes APT**

```
> sudo apt install php-xdebug
```

4. Modificar as configurações padrões do xdebug no arquivo xdebug.ini (ou php.ini dependendo a instalação utilizada), modificando as seguintes chaves:

ATENÇÃO: O NOME DOS PARÂMETROS DO XDEBUG MUDOU NA VERSÃO 3.0.

```
; Configurações do XDEBUG, versão 3.X
xdebug.mode=profile
xdebug.output_dir = /tmp
xdebug.profiler_aggregate = 0
xdebug.profiler_append = 1
```

5. Voltar com servidor para o balanceador de carga para que ele possa coletar logs de desempenho de toda a aplicação. Sugerimos que esta máquina fique coletando dados por cerca de 1 hora ou que o diretório de logs (/tmp) não ultrapasse 5 GB. Também é recomendado que a coleta seja feita nos momentos de pico quando o sistema apresenta lentidão.
6. Os dados gerados da coleta podem ser avaliados em ferramentas como o KCacheGrind ou QCacheGrind para análise detalhada sobre quais os pontos do sistema que estão gerando problemas de desempenho.

