Manual de Desenvolvimento de Módulos para o SEI

Este Manual pode ser acessado pelo link: http://bit.ly/SEI_Manual_Padroes_Desenvolvimento_Modulos

1. Observações Gerais

a. Este Manual tem por objetivo orientar o desenvolvimento de Módulos para o SEI, em aspectos práticos e por meio de padrões a serem seguidos, além do procedimento para desenvolvimento colaborativo junto a projetos de Módulos mantidos por outros órgãos no Gitlab do Portal do SPB.
b. O Manual Técnico de Desenvolvimento de Módulos para o SEI 3.0 deve ser seguido em todo o contexto apresentado no presente Manual, pois este apenas complementa-o, enquanto que o mencionado Manual Técnico 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 diretivas e padrões de interface, codificação PHP e modelagem de banco de dados utilizados pelo SEI, indicados no Manual Técnico de Desenvolvimento de Módulos para o SEI 3.0 e no presente Manual.

c. É importante destacar que qualquer desenvolvimento a ser depois incorporado nos Módulos mantidos pelos outros órgãos deve ser coordenado. Assim, deve-se conhecer a Lista de Demandas do Projeto do PEN-SEI ou do projeto do Módulo específico. Somente demandas **previamente aprovadas** pelo Comitê Gestor do PEN-SEI ou do órgão mantenedor de determinado Módulo podem ser desenvolvidas.

d. **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 possui 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 a ambiente funcional do SEI e seus componentes (SEI, SIP, JOD Converter, Apache Solr e Mencached), 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

   PS: Esta configuração também podem 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 está contido 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/sei
d. Para baixar o arquivo de configuração inicial da Box a ser instalado pelo Vagrant, no diretório raiz dos códigos-fonte do SEI acima citado, execute o seguinte comando:
   vagrant init processoeletronico/sei-3.1

e. 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.
   i. Outra alternativa para configuração do Ambiente de Desenvolvimento Padrão Localhost e execução de comandos específicos do Vagrant é por meio de arquivos BAT utilitários, conforme destacado em tópico mais à frente
f. 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*, devido a utilização da porta 80.
   ii. É normal que a primeira execução desse comando demore vários minutos para ser concluído, 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 Vagrant já armazenou a Box em seu *cache*.
   v. O comando acima é utilizado no primeiro provisionamento (que demora mais, por envolver o download da Box, conforme acima explicado) e sempre que precisar iniciar a VM novamente, seja após parar a execução da VM ou quando precisar destruí-la, conforme comandos do Vagrant indicados em tópico mais à frente.
   vi. Ao final da inicialização do ambiente, sempre deve ser apresentada a seguinte mensagem abaixo, que indica que todos os serviços do SEI estão operando na VM:

      ...
      default: Starting jod
      default: Starting mysql
      default: Starting solr
      default: Starting memcached
      default: Starting smtp
      default: Starting httpd

g. 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/sei
      1. Utilizar o login "**teste**" e a senha "**teste**"

ii. Para acesso SIP, em um browser, acesse o endereço: http://localhost/sip
    1. Utilizar o login "**teste**" e a senha "**teste**"

iii. **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-3.1***" 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 vagrant devem sempre ser executados por meio do prompt de comandos do sistema operacional e 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, com cerca de 4 Gb) e inicia a VM sempre que precisar retomar o trabalho de desenvolvimento ou recria a VM após ser destruída:
    vagrant up

    ii. Desligar a VM de desenvolvimento:
    vagrant halt

    iii. Reiniciar a VM de desenvolvimento:
    vagrant reload

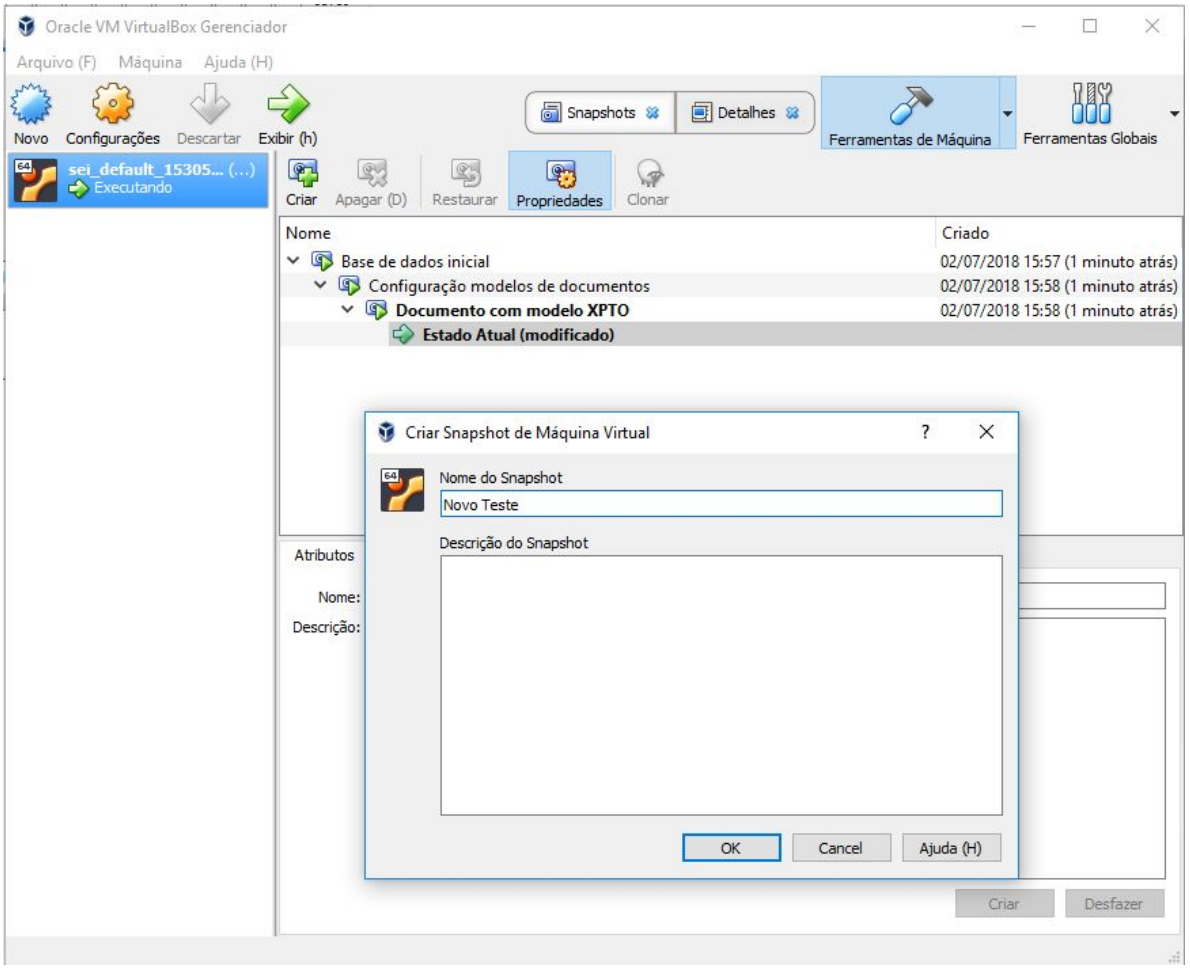
    iv. Destruir a VM de desenvolvimento, apagando todas as informações persistidas no banco de dados e Snapshots que tenham sido criados pelo Virtual Box. Após destruir a VM, em seguida é possível reconstruir novamente com o comando "***vagrant up***", conforme acima indicado.
    vagrant destroy

    v. Atualizar a VM de desenvolvimento para versão mais recente:
    vagrant box update

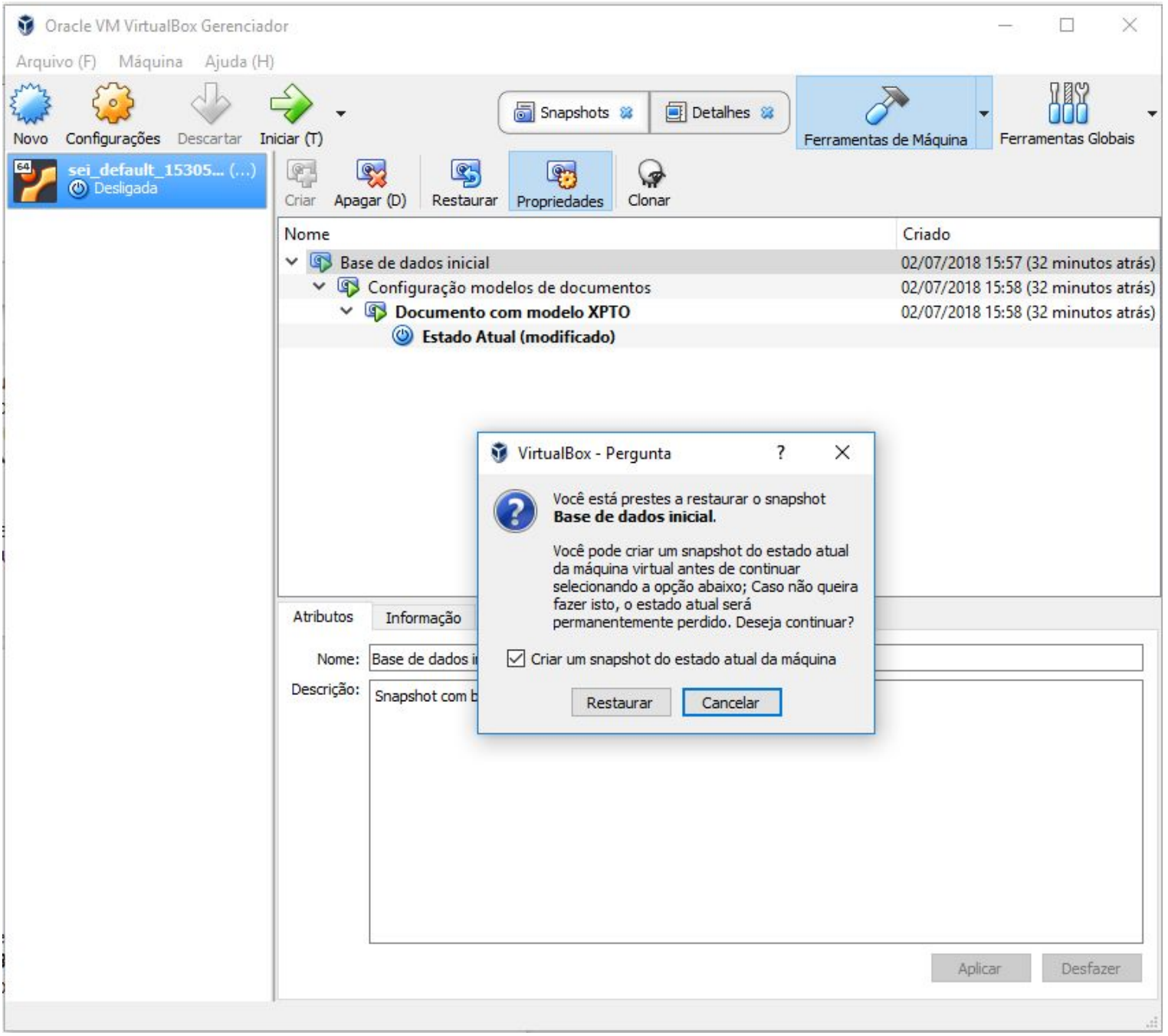
2.4. Snapshots

a. Snapshot é uma funcionalidade disponibilizada pela máquina virtual criada pelo software **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 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.



c. 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

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 portal ***3306***.
   i. Este componente pode ser acessado utilizando a SSH do bach utilizando o Login "***root***" e Senha "***root***".

Ex: **mysql -h 127.0.0.1 -u root -p sei**

b. O banco de dados poderá ser acesso pelo ***MySQL Workbench*** ou qualquer outra ferramenta de conexão à banco de dados.
c. O componente contém o banco de dados do SEI e do SIP, e a instância de banco pode ser acessada com os dados abaixo:
   i. Hostname: 127.0.0.1
   ii. Port: 3306
   iii. Username: root
   iv. Password: root
d. Conforme consta nos arquivos ConfiguracaoSEI.php e ConfiguracaoSip.php, 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***"

2.6. Comutação entre Tipos de Bancos de Dados

a. Subir a Máquina Virtual (VM) do SEI localhost com o comando: vagrant up
b. Utilizando o Git Bash, execute o comando a seguir para entrar na Máquina Virtual (VM): vagrant ssh
    i. Se solicitar login e senha, informar:
        1. Login: vagrant
        2. Senha: vagrant
c. Após entrar na VM, execute o comando a seguir para entrar no modo de edição do arquivo de configuração: sudo vi /docker-compose.yml
    i. Antes, para editar, clicar no teclado em "a" ou INSERT.
    ii. **como nosso exemplo será comutar para o banco do tipo Oracle**, descomentar as linhas da seção "oracle" para ativar este banco.
    iii. comentar as linhas da seção "mysql" para desativar este banco.
    iv. na seção "http" > "links":
        1. descomentar a linha "oracle:oracle" para ativar este banco.
        2. comentar a linha "mysql:mysql" para desativar este banco.
    v. Para sair do editor salvando as edições acima, acionar no teclado: Esc e depois :wq
d. Nas pastas "config" dos códigos-fontes do SEI e SIP no localhost, editar o arquivo:
    i. ConfiguracaoSEI.php
        1. descomentar as linhas da seção "BancoSEI" relacionadas ao banco "oracle" para ativar este banco.
        2. comentar as linhas da seção "BancoSEI" relacionadas ao banco "mysql" para desativar este banco.
    ii. ConfiguracaoSip.php
        1. descomentar as linhas da seção "BancoSip" relacionadas ao banco "oracle" para ativar este banco.
        2. comentar as linhas da seção "BancoSip" relacionadas ao banco "mysql" para desativar este banco.
e. Execute o comando a seguir para sair da VM: exit
f. Na pasta raiz dos códigos-fontes do SEI e SIP no localhost, execute o comando: vagrant reload
g. Todos os passos acima são similares para comutar o banco para **SQL Server**.
h. Para o banco em Oracle, se ocorrer o erro "ORA-28002: the password will expire within 7 days" ou "ORA-01017: invalid username/password; logon denied":
    i. execute novamente o passo "b" acima para entrar na VM.
    ii. execute o comando: docker exec -it oracle bash
    iii. execute o comando: sqlplus
    iv. e em seguida informe o usuário e senha de banco constantes nos arquivos ConfiguracaoSEI.php e o ConfiguracaoSip.php
        1. seguir os passos para atualizar a senha.
        2. verificar se aceita repetir a senha, para não ter que atualizar a senha de banco também nos arquivos ConfiguracaoSEI.php e o ConfiguracaoSip.php.

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. Caso peça o password digitar: **vagrant**
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, copiar e colar o script a ser executado. Exemplos:

    /usr/bin/php -c /etc/php.ini /opt/sip/scripts/atualizar_versao_sei.php 2>&1 > atualizacao_sip.log &

    /usr/bin/php -c /etc/php.ini /opt/sei/scripts/atualizar_versao.php 2>&1 > atualizacao_sei.log &

e. Quando o comando for executado com sucesso sem erro, aparecerá a seguinte linha no prompt:

    [root@11e3c5f99dcc /]#

f. Para verificar a saída da execução finalizada ou durante a execução, digitar: **tail -f nomedoarquivodelog.log**
   Exemplos:
   tail -f atualizacao_sip_sei.log
   tail -f atualizacao_sei.log
g. Para sair do arquivo digitar: **Ctrl + c**

2.8. Acesso ao MailCatcher

a. texto

2.9. Acesso ao Solr

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

2.10. 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 a **26 caracteres**.
e. Todas as colunas deverão possuir uma descrição negocial em seu "comment", informando sua finalidade funcional e, caso existam estados distintos, suas opções e finalidades correspondentes.
    i. Esta prática visa a dicionarização do banco de dados do SEI, podendo alimentar catálogo de dados ou ferramenta de Dicionário de Dados que a instituição possua.

3.2. Padrão Geral do SEI

Este tópico será composto pelo conteúdo a seguir (**documento elaborado pelo TRF4**): https://softwarepublico.gov.br/social/sei/manuais/padrao-de-modelagem-de-dados/sumario

Até a consolidação do texto completo do documento do TRF4 do link acima, segue um texto de **resumo**:

a. 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.
b. 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
c. 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)
d. 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
e. Colunas:
    i. Para chaves primárias seqüenciais 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

f. 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).

g. 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

h. Índice: Para índices utilizar o prefixo i(01-99) seguido do nome da entidade.

i. Exemplo:

i01_processo
i02_processo
.
.
i99_processo

i. Sequência: Utilizar o prefixo seq seguido do nome da Entidade ao qual a seqüência 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.

j. Aconselha-se a criação de uma tabela auxiliar para geração de seqüências, 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. Padrão de Codificação PHP de Módulos

Este tópico será inicialmente composto pelo conteúdo a seguir: https://softwarepublico.gov.br/social/sei/manuais/padrao-de-codificacao-php/sumario

Somado com o tópico "**Padrão de Codificação PHP de Módulos para o SEI**" do artigo a seguir: https://softwarepublico.gov.br/social/sei/manuais/desenvolvimento-colaborativo-e-de-modulos

a. texto
b. texto

5. Padrão de script PHP de Instalação/Atualização de Módulos

Este tópico será inicialmente composto tópico "**Padrão de script PHP de Instalação/Atualização de módulos para o SEI**" constante no artigo a seguir: https://softwarepublico.gov.br/social/sei/manuais/desenvolvimento-colaborativo-e-de-modulos

a. texto
b. texto

6. Desenvolvimento Colaborativo pelo Gitlab do Portal do SPB

6.1. Metodologia de Desenvolvimento Colaborativo no Gitlab

Este tópico será inicialmente composto pelo conteúdo de boa parte do constante no artigo a seguir: https://softwarepublico.gov.br/social/sei/manuais/desenvolvimento-colaborativo-e-de-modulos

a. texto
b. texto

6.2. Configuração de Projetos de Módulos para o SEI no Gitlab

Este tópico será inicialmente composto pelo conteúdo de boa parte do constante no artigo a seguir: https://softwarepublico.gov.br/social/sei/manuais/desenvolvimento-colaborativo-e-de-modulos

a. texto
b. texto

6.3. Configuração de Labels e Milestones para Gestão das Issues no Gitlab

Este tópico será inicialmente composto pelo conteúdo de boa parte do constante no artigo a seguir: https://softwarepublico.gov.br/social/sei/manuais/desenvolvimento-colaborativo-e-de-modulos

a. texto
b. texto