

## Nome do Software: AGATHA

### 1. Visão Geral

O Objetivo deste manual é fornecer informações e dados técnicos aos envolvidos no processo de instalação do Sistema AGATHA – Sistema de Gestão de Riscos.

### 2. Requisitos de Instalação

#### 2.1. Requisitos de Hardware

- 4 GB de RAM
- 4 CPUs

#### 2.2. Requisitos de Software

Sistema Operacional	Versão do Kernel	
Linux	3.10+	
Ferramenta de container	Versão	
Docker	1.12+	
Docker Compose	1.21.0+	
Ferramenta de build		
Node.js	6.10.1+	
Apache Maven	3.x	
Java JDK	8+	
Banco de Dados	Versão	
PostgreSQL	9.5.4	
Servidor de Aplicação	Versão	
Spring Boot	1.3.8 release	
Navegador Web	Versão	
Google Chrome Internet Explorer Mozilla Firefox Safari	-	
Biblioteca/Componente	Versão	Onde pode ser obtido?
Angular	1.6.9	<a href="https://angular.io/">https://angular.io/</a> <a href="http://hibernate.org/">http://hibernate.org/</a> <a href="https://material.io/">https://material.io/</a> <a href="http://spring.io/">http://spring.io/</a>
Hibernate	4.1.4	
Material Design	-	
Spring	5.0	

## 2.3. Requisitos Gerais

Liberação das credenciais de acesso **client-id** e **client-secret** deverão ser solicitadas ao Departamento de Governo Digital da Secretaria de Tecnologia da Informação e Comunicação do Ministério do Planejamento, por meio do encaminhamento do documento preenchido "plano-configuracao-brasil-cidadao-sistema-agatha.doc" disponível em - <https://softwarepublico.gov.br/social/agatha/manuais-de-usuario/agatha-plano-configuracao.doc> para o email [higo.gomes@planejamento.gov.br](mailto:higo.gomes@planejamento.gov.br)

## 3. Instalação

Para a utilização do sistema Agatha são necessários a prévia instalação do serviço Docker e a aplicação Docker Compose.

### 3.1. Instalação do Docker.

Para a instalação do docker em ambientes Linux como Centos, Ubuntu ou Debian execute os seguintes comandos como **root**:

```
# Instala o serviço docker utilizando o script de conveniência
curl -fsSL https://get.docker.com -o get-docker.sh
sh get-docker.sh
```

```
# Inicia o serviço e o adiciona a inicialização do sistema
systemctl start docker
systemctl enable docker
```

Para validar a instalação com sucesso, testar os comandos abaixo :

```
docker ps
```

deve retornar algo como:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
--------------	-------	---------	---------	--------	-------	-------

### 3.2. Instalação do Docker-compose.

Para a instalação do docker em ambientes Linux como Centos, Ubuntu ou Debian execute os seguintes comandos como **root**:

```
curl -L "https://github.com/docker/compose/releases/download/1.22.0/docker-compose-$(uname -s)-$(uname -m)" -o /usr/local/bin/docker-compose
```

Para testar a instalação como usuário comum, execute o comando

```
docker-compose --version
$ docker-compose version 1.21.2, build a133471 #Atualmente
get http://rpms.famillecollet.com/enterprise/remi-release-7.rpm
```

### 3.3. Instalação do Agatha

Execute o comando abaixo para realizar o clone do código fonte para sua máquina local, pode ser necessário instalar o comando **git** na máquina.

```
git clone http://softwarepublico.gov.br/gitlab/agatha/agatha.git agatha
```

Uma pasta de nome **agatha** deve ser criada no local onde executou o comando.

Acesse a pasta **agatha/docker**,

Perceba que dentro dela existirá um arquivo **docker-compose.yml**, este arquivo é o responsável pela orquestração dos containers de banco, backend e frontend.

As informações de integração do sistema com o BrasilCidadão devem ser configuradas no arquivo **spring/config/application.yaml**, perceba que no arquivo docker-compose, esta pasta spring/config é mapeada como **volume** para facilitar o ajuste destas configurações.

```
ecidadao:

  client-id: SOLICITAR-MP
  client-secret: SOLICITAR-MP
  pre-established-redirect-uri:
http://dns\_definido/gestaoriscos/api/login/openid
  sistema:
    dominio: dns_definido
    url-pattern-protegido: /**
    redireciona-login-sucesso: http://dns\_definido
    redireciona-logout-sucesso: http://dns\_definido

  # URIS dO AMBIENTE DE TESTES
  access-token-uri: SOLICITAR-MP
  user-authorization-uri: SOLICITAR-MP
  use-current-uri: false
  scope: openid
```

Os campos SOLICITAR-MP são as credenciais de acesso client-id e client-secret que deverão ser solicitadas ao Departamento de Governo Digital da Secretaria de Tecnologia da Informação e Comunicação do Ministério do Planejamento conforme o item **2.3 – Requisitos Gerais**.

Os campos **dns\_definido** é necessário informar o DNS definido para a utilização do Sistema.

Importante ressaltar que no parâmetro “pre-established-redirect-uri” precisa-se manter o final da URL “gestaoriscos/api/login/openid”.

### 3.4. Instalação do Ambiente JDK Java, Maven e NodeJs

A instalação do Kit de desenvolvimento Java (JDK) e Maven (Gerenciador de dependências Java utilizado no Agatha), pode variar de acordo com o sistema operacional utilizado, por favor pesquisar como proceder com a instalação de acordo com o sistema operacional utilizado.

Para validar a instalação com sucesso, testar os comandos abaixo:

```
javac -version
```

deve retornar algo como:

```
javac 1.8.0_171
```

e o maven:

```
mvn -version
```

deve retornar algo como:

```
Apache Maven 3.5.2
```

```
Maven home: /usr/share/maven
```

```
Java version: 1.8.0_171, vendor: Oracle Corporation
```

```
Java home: /usr/lib/jvm/java-8-oracle/jre
```

A Instalação do NodeJs e Npm(necessário a compilação do frontend), também pode variar de sistema para sistema, por favor pesquisar como proceder com a instalação de acordo com o sistema operacional utilizado.

Para validar a instalação com sucesso, testar os comandos abaixo:

```
node --version
```

deve retornar algo como:

```
v8.10.0
```

e para o npm:

```
npm -version
```

deve retornar algo como:

```
3.5.2
```

- Build do Frontend

Execute o comando abaixo dentro do diretório **docker** para executar o script de build do frontend.

**./build-frontend.sh**

Em caso de sucesso, um novo pacote **dist.tar.gz**, será gerado dentro de **nginx/**

- Build do Backend

Para compilar o código e criar o arquivo **app.jar** execute o código abaixo na pasta **docker**.

**./build-backend.sh**

Em caso de sucesso, um novo pacote **app.jar**, será gerado dentro de **spring/**

### 3.5. Executando a aplicação pela primeira vez

Após a execução destes passos já será possível iniciar a aplicação a primeira vez.

A versão padrão esta utilizando a árvore de órgãos e departamentos do Ministério do Planejamento.

Após um primeiro teste, caso se deseje alterar este apontamento para a árvore do seu órgão, será necessário executar os passos explicados mais a frente no item 4 – Alteração de Estrutura Organizacional.

Para iniciar a aplicação acesse a pasta **docker** pelo terminal e execute o comando como **root**:

**docker-compose up -d**

O parâmetro **-d** envia a execução e os log's dos containers para segundo plano (background), para visualizar os logs e identificar possíveis problemas execute:

**docker-compose logs -f**

Se quiser parar a aplicação execute:

**docker-compose stop**

O container de backend é o que demora mais para subir, um indicativo de subida com sucesso, pode ser visto como abaixo:

```
mpog-gestaoriscos-backend | 2018-09-24 22:00:05.798 INFO 1 --- [      main] b.g.m.gestaoriscos.SistemaA
pplication : Started SistemaApplication in 22.44 seconds (JVM running for 23.197)
```

executando o comando **docker ps**, os 3 containers de frontend, backend e banco devem estar em execução:

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
889ab9b1468b	mpog-gestaoriscos-frontend	"sh -c '/etc/nginx/e..."	22 seconds ago	Up 21 seconds
0.0.0.0:80->80/tcp	mpog-gestaoriscos-frontend			
0e383122015b	mpog-gestaoriscos-backend	"sh -c 'java \$JAVA_O..."	24 seconds ago	Up 23 seconds
8080/tcp	mpog-gestaoriscos-backend			
fc7d13be95cb	docker_mpog-gestaoriscos-bd	"/docker-entrypoint..."	25 seconds ago	Up 24 seconds
5432/tcp	mpog-gestaoriscos-bd			

Desta forma, já deve ser possível acessar a aplicação pelo navegador, acessando <http://localhost/> ou <http://0.0.0.0/>

### 3.6. Criando um usuário administrador

Utilize o passo a passo abaixo para criação de um usuário “administrador”. O perfil administrador é o “Núcleo” de ID=3, somente ele possui acesso a criação de novos usuários e a todos os menus.

No terminal, estando na pasta **docker**, execute os comandos abaixo para acessar o container de banco de dados:

**docker-compose exec mpog-gestaoriscos-bd bash**

```
root@fc7d13be95cb:/# su postgres
root@fc7d13be95cb:/# psql
root@fc7d13be95cb:/# \c gestaoriscos
```

- Insere o usuário “JOSE...” na tabela de usuário utilizando o próximo valor da sequence ‘gestaoriscos.sk\_usuario’.

```
insert into gestaoriscos.tb_usuario values (nextval('gestaoriscos.sk_usuario'),'JOSÉ GONÇALVES', '00002037343',null, 'alterar@planejamento.gov.br');
```

Após esse comando, será inserido um novo registro na tabela ‘tb\_usuario’ e gerado um ‘id’ para o usuário José.

- Em seguida, você deve:
- substituir o trecho ‘<ID\_USUARIO>’ da query abaixo

pelo do usuário gerado no passo anterior (faça um select na tabela para visualizar);

**select \* from gestaoriscos.tb\_usuario;**

- substitua o trecho ‘<ID\_PERFIL>’ da query abaixo pelo perfil desejado (utilize o ID\_PERFIL=3 para inserir um Administrador se for o primeiro usuário).

```
insert into gestaoriscos.tb_permissao values
(nextval('gestaoriscos.sk_permissao'),<ID_USUARIO>,<ID_PERFIL>, false, current_timestamp);
```

**Após esse comando, será vinculado o perfil desejado ao usuário informado e já será possível efetuar uma primeira tentativa de login no sistema, assim como efetuar a degustação das funcionalidades de acordo com o explicado no Manual de Usuário!**

Lista de possíveis perfis (Consultar o Manual de Usuário):

ID      Descrição

1. Comitê
2. Subcomitê
3. Núcleo(Administrador)
4. Unidade
5. Gestor doProcesso
6. Analista de Risco

#### **4. Alteração da Estrutura Organizacional (Opcional)**

Para customizar a estrutura organizacional utilizada pelo sistema para a realidade do seu Órgão é necessário alterar os arquivos conforme indicado abaixo e posteriormente compilar o Frontend e Backend.

- Alteração deArquivos

Alteração 01:

Nome do arquivo: gerenciar-permissoes.controller.js

Diretório onde fica o arquivo:

codigo-fonte/cliente/src/scripts/pages/gerenciar-permissoes/controller/

Alterar linha 19

de: scope.idOrgaoPai = 2981;

para: scope.idOrgaoPai = código do Órgão no SIORG;

Alteração 02:

Nome do arquivo: gerenciar-permissoes.service.js

Diretório onde fica o arquivo:

codigo-fonte/cliente/src/scripts/pages/gerenciar-permissoes/services/

Alterar linha 26

de: return \$http.get(orgaoUrl + "/orgao/2981");

para: return \$http.get(orgaoUrl + "/orgao/código do Órgão no SIORG");

Alteração 03:

Nome do arquivo: home.controller.js

Diretório onde fica o arquivo:

codigo-fonte/cliente/src/scripts/pages/home/controller/

Alterar linha 11

de: scope.idOrgaoPai = 2981;

para: scope.idOrgaoPai = **código do Órgão no SIORG**;

Alteração 04 (Backend):

Nos arquivos abaixo:

- codigo-fonte/servico/src/main/java/br/gov/mpog/gestaoriscos/servico/impl/  
TaxonomiaServiceImpl.java

- codigo-fonte/servico/src/main/java/br/gov/mpog/gestaoriscos/servico/impl/  
PermissaoServiceImpl.java

- codigo-fonte/servico/src/main/java/br/gov/mpog/gestaoriscos/repositorio/impl/  
OrgaoCustomRepositorioImpl.java

Alterar o valor **2981** (ID do Ministério do Planejamento) pelo ID de seu órgão.

Para ter conhecimento de qual é o **código do Órgão** em que o sistema esta sendo instalado, basta acessar o sítio do SIORG, no menu dedicado as Estruturas Organizacionais ([https://siorg.planejamento.gov.br/siorg-cidadao-webapp/pages/listar\\_orgaos\\_estruturas/listar\\_orgaos\\_estruturas.jsf](https://siorg.planejamento.gov.br/siorg-cidadao-webapp/pages/listar_orgaos_estruturas/listar_orgaos_estruturas.jsf))

- Novo Build do Frontend

Execute o comando abaixo dentro do diretório **docker** para executar o script de build do frontend.

**./build-frontend.sh**

Em caso de sucesso, um novo pacote **dist.tar.gz**, será gerado dentro de **nginx/**

- Novo Build do Backend

Para compilar o código e criar o arquivo **app.jar** execute o código abaixo na pasta **docker**.

**./build-backend.sh**

Em caso de sucesso, um novo pacote **app.jar**, será gerado dentro de **spring/**

- Removendo o cache das imagens docker e reiniciando

Após gerar os novos pacotes, é necessário remover os containers e as imagens docker criadas anteriormente, do contrário elas serão utilizadas como cache para tornar o processo mais rápido.

Dentro da pasta **docker**, execute:

**docker-compose stop**

**docker-compose rm** (e digite y para confirmar a remoção dos containers)

**docker images** (vai listar as imagens docker em cache)



REPOSITORY	TAG	IMAGE ID
docker_mpog-gestaoriscos-bd	latest	32be5fd74de9
mpog-gestaoriscos-frontend	latest	61fceb76d8c0
<none>	<none>	d98eb590861d
mpog-gestaoriscos-backend	latest	eb43db5de40e

# Apaga o cache das imagens para forçar a recriação com os novos binários  
**docker rmi** <id da imagem do frontend> <id da imagem do backend>

Agora basta executar novamente a aplicação e a árvore do órgão já deve estar apontando para a de seu órgão:

**docker-compose up -d**

## 5. Integração com o BrasilCidadão (Apenas informativo)

Para maiores informações sobre como é realizada a integração com Brasil Cidadão, acesse os links abaixo:

<http://www.planejamento.gov.br/antigocidadaniadigital/brasil-cidadao>

<https://www.governodigital.gov.br/documentos-e-arquivos/sdk-Brasil-cidadao.pdf>