

# Processo de Release

As *branches* liberadas para uma nova versão do sistema devem ser consultadas nas tarefas do [Redmine](#) com a situação "*Liberação (Release)*" relacionadas a um projeto que tenha o campo **Grupo** com o valor *SIG-UFLA* ou que tenham o campo **Sistema** igual a *SIG-UFLA* (estes campos são campos personalizados criados no Redmine, este último está relacionado a tarefa e só precisa ser consultado se o projeto for de um grupo diferente do SIG-UFLA). Por padrão, o nome da *branch* corresponde ao número da tarefa, entretanto, uma *branch* pode abranger várias tarefas, portanto, é necessário verificar o valor do campo **Branch** das tarefas.

Se a *branch* tiver relacionada a alguma tarefa que ainda não foi liberada para *release*, ela não poderá ser incluída na próxima versão do sistema. Também é necessário verificar se a tarefa liberada é subtarefa de outra ainda não liberada, ou se possui tarefas filhas ainda não liberadas. *Atenção*: este último passo não é verificado automaticamente pela ferramenta de *release*.

Para facilitar a seleção e o acompanhamento do processo de montagem de *release*, foi desenvolvido uma página no módulo de desenvolvimento do SIG chamada "[Preparar Release](#)". Esta página mostra uma tabela com a lista das tarefas disponíveis para *release* e suas respectivas *branches*. Também é informado a situação do merge de cada *branch* das tarefas com a *branch next*.

Para utilizar esta ferramenta é necessário:

- executar o script "*scripts/git\_daemon.php*" com um usuário que possua permissão de acesso ao repositório do SIG-UFLA no GIT. Este script irá se comportar como um servidor e processar as requisições da página "Preparar Release", portanto deverá permanecer em execução enquanto esta ferramenta estiver sendo utilizada.
- configurar um usuário de acesso ao banco de dados do Redmine no arquivo `sig_local.php` (com o identificador `sig::ID_BD_REDMINE`).

## Passos para Montagem da Nova Versão

1. Selecione a *branch next* e atualize o repositório do SIG-UFLA:

```
1 xxxxxxxxxxxx
```

```
1  
1 git checkout next
```

```
2  
1 git pull
```

2. Para cada *branch* liberada, realize o *merge*:

```
1 xxxxxxxxxxxx
```

```
1  
1 git merge --no-commit --no-ff <número da tarefa>
```

## O uso das opções `--no-commit --no-ff` é opcional mas recomendado, pois permite a conferência das modificações antes da conclusão do `commit` e facilita a visualização do histórico das branches **next** e **master**. Durante a conferência, se for necessário cancelar o `merge` basta usar a opção `git reset --hard`

**Se houver conflitos**, verifique se é possível resolvê-los sem prejudicar as funcionalidades existentes e as novas que porventura estejam sendo introduzidas na nova versão. Se não for viável a resolução do conflito, retorne a tarefa para o desenvolvedor responsável e solicite que sejam realizadas as alterações necessárias, ponderando, inclusive, a necessidade de realização de novos testes.

Se houver modificações em arquivos de instalação de uso geral do sistema (grupos, tipos de usuários, etc.), confira se os valores das constantes estão corretos e sem valores coincidentes.

3. Atualize o número da versão do SIG-UFLA:

Altere no arquivo `classes/definicoes/sig.class.php` o valor da constante `VERSAO`, exemplo:

```
1 xxxxxxxxxxxx
```

1

```
1 const VERSAO = '1.64.5';
```

4. Envie as mudanças para o repositório remoto:

```
1 xxxxxxxxxxxx
```

1

```
1 git push origin next
```

5. Realize o merge da *branch next* com a *branch master*:

```
1 xxxxxxxxxxxx
```

1

```
1 git checkout master
```

2

```
1 git pull
```

3

```
1 git merge --no-commit --no-ff next
```

6. Envie as mudanças para o repositório remoto:

```
1 xxxxxxxxxxxx
```

1

```
1 git add .
```

2

```
1 git commit -m "Release x.y.z"
```

3

```
1 git push origin master
```

7. Se houve conflitos durante a montagem da *release*, faça um teste local da nova versão do sistema:

1. Restaure o banco de dados local (se necessário);
2. Execute os scripts das tarefas da *release* encontrados no diretório **atualizacoes** (se houver);
3. Execute o scripts de atualização de permissões;
4. Acesse como administrador e verifique a página de atualização do sistema;
5. Execute um caso de teste de cada tarefa liberada.

8. Se for necessário bloquear o acesso de usuários autenticados no sistema, é preciso adicionar um aviso de manutenção no SIG-UFLA e, pelo menos um dia antes, solicitar a inserção de uma janela de manutenção do sistema no Zabbix. Esta solicitação deve ser feita através de um e-mail encaminhado ao Coordenador da Coordenadoria de Operação e Segurança da Informação com cópia para o Diretor da DGTI e o Coordenador da CSI. Após o cadastro da janela de manutenção, o cadastro do aviso no SIG-UFLA pode ser feito através da página de configurações no menu *Administradores*.

## Quando Bloquear o SIG-UFLA?

Dependendo das alterações realizadas nas tarefas liberadas para *release* é necessário bloquear o acesso ao sistema. Este bloqueio pode ser apenas para usuários autenticados (via configuração na área administrativa) ou para qualquer acesso às páginas do SIG (via alteração do arquivo *classes/definicoes/sig\_local.php*).

Para decidir quando realizar o bloqueio considere as seguintes diretivas:

- Existem scripts a serem executados? Se não, *não* é necessário bloquear o sistema.

Se existir scripts, é *aconselhável* bloquear o sistema caso exista pelo menos um comando para:

- Instalar ou alterar entidades.
- Alterar a estrutura do banco de dados (instruções SQL DDL, por exemplo, ALTER, CREATE, DROP, etc.).
- Alterar registros identificados por atributos que não são únicos (nestes casos existe a possibilidade de surgirem novos registros que atendam as condições da consulta durante a execução do script).

## Número de Versão do SIG

O número de versão do SIG-UFLA é constituído de três dígitos no formato "**x.y.z**". Após cada *release* é incrementado o número *y* e se houver uma mudança significativa no *framework*, caso o contrário é incrementado apenas o número *z*. Apesar de não haver um consenso em relação à alteração do número *x*, está previsto uma alteração neste valor após a implantação de um novo *layout*.

# Atualização do SIG-UFLA

A seguir são apresentados os passos para realizar a atualização do sistema utilizando o script de *release*. Antes de iniciar estes passos, é necessário ter um usuário de acesso aos servidores que hospedam os serviços Web. Para isto, com o auxílio de um administrador do servidor, siga os passos descritos no documento [Criação de usuários para gerentes de configuração do SIG-UFLA](#). Outras informações importantes podem ser consultadas em [Configurações do servidor do SIG-UFLA](#). **Obs.:** O usuário responsável pela release precisa ter permissão de *push* na *branch master* do repositório [sigufla](#).

## 1. Acessar servidor

O acesso ao servidor é realizado utilizando o protocolo SSH, exemplo:

```
1 xxxxxxxxxxxx
```

```
1 ssh login_do_servidor@200.131.250.223
```

Caso o SIG-UFLA esteja rodando em mais de uma máquina, é necessário verificar o IP do servidor principal (normalmente é o 223). As atividades de *release*, utilizando o script de *release*, são executadas em apenas uma máquina. Se for necessário bloquear totalmente o acesso ao sistema, será necessário acessar todas as instâncias e alterar o arquivo *sig\_local.php* antes de executar o script de *release*.

## 2. Bloquear o sistema para manutenção (se necessário)

O objetivo desta etapa é evitar a utilização do sistema enquanto não for concluída a execução dos scripts da *release*.

Existem duas maneiras de fechar o sistema:

- Acessar o SIG-UFLA em produção com um usuário do grupo Administradores, ir em Configurações e na aba "Autenticação" alterar a opção Fechado na caixa Manutenção para *Sim* e adicionar a mensagem de quanto tempo o sistema ficará indisponível. Isto apenas impede que os usuários realizem o *log in* no sistema, mas as funcionalidades abertas (Acesso de Candidatos, Consultas de Diplomas, Horário de Disciplinas, Matrizes Curriculares, etc.) permanecem funcionando.
- Para retirar o sistema totalmente do ar é necessário editar no arquivo *classes/definicoes/sig\_local.php* a variável `simp:: $manutencao` com a mensagem de aviso sobre a atualização do sistema, ex.:

```
1 xxxxxxxxxxxx
```

```
1 simp::$manutencao = 'Atualização do sistema. Previsão de abertura para 9h00 de 21/12/2020.';
```

## ATENÇÃO:

- Esta alteração deve ser feita nos servidores do SIG-UFLA, pois o arquivo *sig\_local.php* não é versionado. Após a *release*, a alteração deve ser desfeita.
- Independente da opção escolhida, os scripts no *crontab* continuam funcionando, portanto, é necessário verificar se precisa parar ou não este serviço.

## 3. Executar o [script de release](#)

```
1 xxxxxxxxxxxx
```

1

```
1 cd /var/www/sistema/sigufla/scripts
```

2

```
1 php release.php -l
```

Leia *atentamente* as mensagens exibidas. O script `release.php` realiza os seguintes passos mediante confirmação do usuário:

- Cria uma tag da versão atual do sig
- Atualiza o repositório do SIG-UFLA nos servidores.
- Para cada script das tarefas liberadas:
- Executa o script e grava a saída na pasta de histórico (se o retorno do script for igual a `0`)
- Move o script para a pasta de histórico
- Se houver outros arquivos relacionados à tarefa, move eles para a pasta de histórico
- Atualiza o repositório de histórico remoto.
- Executa o script de atualização de permissões
- Limpa o cache de memória
- Limpa o cache do opcode
- Atualiza o redmine
- Cadastra a nova versão do sistema
- Atualiza as versões das documentações do sistema

5. Se houver alguma modificação no arquivo `composer.lock`

Execute o comando abaixo em **cada uma** das instância de servidores do SIG-UFLA:

```
1 xxxxxxxxxxxx
```

1

```
1 cd /var/www/sistema/sigufla/scripts
```

2

```
1 php release.php -c
```

6. Se houver a necessidade de atualizar/alterar o **crontab**:

Alguns scripts do SIG-UFLA precisam ser executados a cada intervalo de tempo. Isto é feito utilizando o `crontab`. Normalmente estes scripts são executados apenas no servidor 200.131.250.223, mas em algumas situações específicas pode ser necessário que um determinado script seja executado em todos os servidores. O seguinte comando pode ser utilizado para editar o `crontab`:

```
1 xxxxxxxxxxxx
```

1

```
1 sudo nano /etc/crontab
```

## 7. Verificar a nova versão e liberar o sistema

Desfaça o bloqueio do sistema (caso tenha sido realizado). Acesse o sistema em produção e confira o número de versão mostrado ao passar o *mouse* em cima do logo **SIG**.

**Obs.:** Caso tenha bloqueado o sistema pela área administrativa, é necessário limpar o cache de memória de todos os servidores para que o desbloqueio tenha efeito. Isto pode ser feito via script (`php webservice/servidor.php -o limpar_apcu`), via área administrativa através da página "Cache em Memória", ou reiniciando o serviço do php (`php-XXX.fpm`) pelo terminal dos servidores.

# Comandos para realizar a liberação manual de novas versões

Se por algum motivo não for possível executar o script de *release*, os passos realizados por ele podem ser feitos manualmente. A seguir são apresentados alguns comandos para isto (estes comandos também pode ser conferidos no próprio script de *release*).

- Criação uma nova *tag*:

```
1 xxxxxxxxxxx
```

1

```
1 cd /var/www/sistema/sigufila
```

2

```
1 git checkout master
```

3

```
1 git tag -a sig-x.y.z -m "Release x.y.z"
```

4

```
1 git push origin sig-x.y.z
```

- Atualização do repositório:

```
1 xxxxxxxxxxx
```

1

```
1 git checkout master
```

2

```
1 git pull
```

3

```
1 git submodule update --init --recursive
```

4

```
1 .
```

5

```
1 # Para atualizacao do repositorio de outros servidores:
```

6

```
1 ssh login_do_servidor@200.131.250.22X 'hostname; cd /var/www/sisistema/sigufla; git pull origin master; e
```

- Para executar um script de *release*:

```
1 xxxxxxxxxx
```

1

```
1 cd /var/www/sistema/sigufla/atualizacoes
```

2

```
1 php redmineXXXXX.php > historico/saida_redmineXXXXX.txt
```

3

```
1 # Se a execução ocorrer sem problemas:
```

4

```
1 mv redmineXXXXX.php historico/redmineXXXXX.php
```

5

```
1 cd historico
```

6

```
1 git add .
```

7

```
1 git commit -m "Release da tarefa XXXXX"
```

8

```
1 git push origin master
```

Após a execução do(s) scrip(s) e atualização do repositório *historico*, é necessário atualizar o repositório principal do SIG-UFLA:

```
1 xxxxxxxxxxxx
```

1

```
1 cd /var/www/sistema/sigufla
```

2

```
1 git add .
```

3

```
1 git commit -m "Scrip(s) da release X.Y.Z"
```

4

```
1 git push origin master
```

- Para executar o script de atualização de permissões:

```
1 xxxxxxxxxxxx
```

1

```
1 cd /var/www/sistema/sigufla/scripts
```

2

```
1 php permissoes.php
```

- Para limpar o cache do php (OPcache):

```
1 xxxxxxxxxxxx
```

1

```
1 cd /var/www/sistema/sigufla
```

2

```
1 php webservice/servidor.php -o limpar_opcache
```

- Para limpar o cache de memória (APCU):

```
1 xxxxxxxxxxxx
```

1

```
1 cd /var/www/sistema/sigufila
```

2

```
1 php webservice/servidor.php -o limpar_apcu
```

- Para atualizar o [composer](#) (se necessário):

```
1 xxxxxxxxxxxx
```

1

```
1 cd /var/www/sistema/sigufila
```

2

```
1 composer install --optimize-autoloader
```

3

```
1 sudo chown -R dgti:www-data vendor/mpdf/mpdf/tmp
```

4

```
1 sudo chmod -R 775 vendor/mpdf/mpdf/tmp
```

O script de *release* também realiza as seguintes atualizações no Redmine:

- Para cada tarefa liberada:
- Muda a situação para "Concluída"
- Altera o campo **Versão do sistema** com o nome da versão que foi liberada (ex.: SIG-UFLA 1.64.5)
- Atribui para o analista que especificou a tarefa para que ele possa entrar em contato com o cliente

## Atividades finais

Após a release, os seguintes passos devem ser executados:

1. Acesse o SIG-UFLA e cadastre a nova versão do sistema (Caminho: "Página Principal » Sistemas » Versões do Sistema » Cadastrar Versão de Sistema"). Este passo já é executado pelo script de release.
2. Envie uma mensagem no grupo do *Gtalk* "Notas de Versão" informando sobre a nova *release* e solicitando o cadastro das notas de versão.

Obs.: Após o cadastro das notas de versão, é necessário realizar a limpeza do cache de memória para que a página de [notas de versão do SIG-UFLA](#) seja atualizada.