GamiBot: atualização pelo sistema de ficheiros

O pacote `gamibot-atualizacao-ficheiros.zip` destina-se a um Moodle onde **GamiBot Manager** e **GamiBot Chat** já estão instalados. Contém os dois plugins completos; a atualização coloca o código no servidor e executa as migrações pela linha de comandos, sem carregar ZIPs no instalador web do Moodle.

Plugin	Versão deste pacote	Diretório no pacote	Destino na árvore Moodle
GamiBot Manager (`local_gamibot_manager`)	`2026091802`	`local/gamibot_manager/`	`local/gamibot_manager/`
GamiBot Chat (`block_gamibot`)	`2026091801`	`blocks/gamibot/`	`blocks/gamibot/`

O Chat depende do Manager. Os conectores `gbtool_langflow` e `gbtool_ollama` e as finalidades `gbpurpose_clarify`, `gbpurpose_summarize` e `gbpurpose_quiz` já estão dentro do diretório do Manager; não os mova separadamente. Ambos os plugins requerem Moodle **4.4 ou posterior**.

1. Confirmar versões e caminhos

Confirme que os dois plugins estão instalados e compare as versões registadas no site com a tabela acima. **Não aplique este pacote se algum plugin instalado tiver uma versão superior**; uma atualização do Moodle não suporta regressar à versão anterior de um plugin. Se a versão já for igual, `upgrade.php` pode não ter migrações para executar.

Defina `MOODLE_CODE` como o diretório que contém `admin/cli/upgrade.php`, `local/` e `blocks/`. Em Moodle 4.4 a 5.0 é a raiz da instalação (por exemplo, `/srv/moodle`). Em Moodle 5.1 e posteriores é o subdiretório `public/` (por exemplo, `/srv/moodle/public`). Se estiver também a mudar de Moodle 5.0 para 5.1, use os destinos dentro de `public/`.

2. Preparar a janela e as cópias de segurança

Teste a atualização numa cópia do site antes de a aplicar em produção. Planeie uma janela de manutenção, evite novas execuções do cron durante a troca do código e espere que as execuções já iniciadas terminem. Execute os comandos CLI com o utilizador e a versão de PHP usados normalmente por este Moodle.

```
set -e
MOODLE_CODE=/srv/moodle/public
cd "$MOODLE_CODE"
php admin/cli/maintenance.php --enable
```

O caminho acima é apenas um exemplo; para Moodle 4.4 a 5.0 pode ser `/srv/moodle`. Continue na mesma sessão de terminal ou volte a definir `MOODLE_CODE`. Antes de trocar os diretórios, faça cópias de segurança **da base de dados, de `moodledata` e dos diretórios atuais dos dois plugins** através dos procedimentos do seu servidor. Guarde as cópias fora da árvore Moodle e do diretório servido pela web.

3. Substituir os dois diretórios completos

Descomprima o ZIP do pacote numa área temporária fora da árvore Moodle. Escolha um diretório de backup **novo e vazio**, também fora da árvore Moodle. Substitua os caminhos de exemplo antes de executar qualquer comando. **Se algum comando falhar, pare e mantenha o site em manutenção.**

```
set -eu
PACOTE=/srv/staging/gamibot-atualizacao-ficheiros
BACKUP_DIR=/srv/backups/gamibot-antes-atualizacao
: "${MOODLE_CODE:?Defina MOODLE_CODE}"
test -f "$PACOTE/local/gamibot_manager/version.php"
test -f "$PACOTE/blocks/gamibot/version.php"
mkdir "$BACKUP_DIR"
mv "$MOODLE_CODE/local/gamibot_manager" "$BACKUP_DIR/gamibot_manager"
mv "$MOODLE_CODE/blocks/gamibot" "$BACKUP_DIR/gamibot"
cp -R "$PACOTE/local/gamibot_manager" "$MOODLE_CODE/local/"
cp -R "$PACOTE/blocks/gamibot" "$MOODLE_CODE/blocks/"
```

Não copie sobre as pastas antigas: podem ficar ficheiros obsoletos. Não deixe pastas de backup dentro de `local/` ou `blocks/`, pois o Moodle analisa esses diretórios como plugins. Confirme a estrutura e os ficheiros compilados:

```
test -f "$MOODLE_CODE/local/gamibot_manager/version.php"
test -f "$MOODLE_CODE/blocks/gamibot/version.php"
test -f "$MOODLE_CODE/blocks/gamibot/amd/build/dialog.min.js"
```

Mantenha o proprietário e as permissões adequados ao seu servidor. O processo web deve conseguir ler os novos ficheiros.

4. Executar a atualização Moodle

Depois de **ambos** os diretórios estarem no destino, execute a atualização da base de dados e limpe as caches a partir de `MOODLE_CODE`. Se a atualização falhar, não avance para a reabertura do site.

```
set -eu
cd "$MOODLE_CODE"
php admin/cli/upgrade.php --non-interactive
php admin/cli/purge_caches.php
```

Não desinstale os plugins para os atualizar. O pacote não contém credenciais; confirme as definições existentes, sobretudo endpoints e conexões Langflow personalizadas, após a atualização.

5. Validar e reabrir

Confirme que o comando de atualização terminou sem erros e que o Moodle reconhece as versões do pacote. Verifique as definições do Manager, uma disciplina com o Chat ativo e as tarefas agendadas do GamiBot. Reative o cron e, depois da validação, desative a manutenção:

```
set -eu
cd "$MOODLE_CODE"
php admin/cli/maintenance.php --disable
```

Se for necessário reverter

Após uma migração da base de dados, **não reverta apenas os ficheiros de código**. Reponha em conjunto a base de dados, `moodledata` e os dois diretórios de plugins das cópias de segurança consistentes efetuadas antes da atualização, seguindo o procedimento de recuperação do seu site. Mantenha a manutenção ativa até a reposição estar validada.

Referências Moodle: atualização e cópias de segurança, instalação manual e nomes dos diretórios e árvore `public/` a partir do Moodle 5.1.