

Manual de Atualização — Sistema de Senhas

Guia para atualizar um sistema **já instalado** num servidor de cliente para uma versão mais nova.

A atualização mais comum é só de **interface (frontend)**. Quando o backend (`src\`, `server.js`) muda, há um passo extra documentado no final.

“ **Antes de começar**, sempre fazer backup. Detalhes no passo B2.

⚠ O que NÃO pode ser perdido

A atualização **mexe na pasta** `public\`, mas existem dados do cliente nessa pasta e fora dela que precisam ser preservados a qualquer custo.

Arquivos que pertencem ao CLIENTE (preservar sempre)

Local	O que contém
<code>C:\Sistemas\mvp-senhas\data\state.json</code>	TUDO que o cliente cadastrou: usuários, guichês, salas, parâmetros, motivos, prioridades, senhas emitidas, histórico completo, auditoria, vínculos atendente↔sala/guichê
<code>C:\Sistemas\mvp-senhas\public\logo.png</code>	Logo do cliente (enviado via Admin → Logo). Cuidado: fica DENTRO da pasta <code>public\</code> , que é a que vamos substituir.
<code>C:\Sistemas\mvp-senhas\.env</code>	Porta, segredo de sessão, caminho do data. Cada cliente tem o seu — nunca sobrescrever.
<code>C:\Sistemas\mvp-senhas\data\</code> (pasta inteira)	Reservada pra dados do cliente. Não tocar nessa pasta em momento algum.

Arquivos que SÃO substituídos na atualização

Local	O que contém
<code>C:\Sistemas\mvp-senhas\public\index.html</code>	Página principal do novo build
<code>C:\Sistemas\mvp-senhas\public\assets\</code>	JS/CSS gerados pelo Vite (nomes mudam a cada build)
<code>C:\Sistemas\mvp-senhas\public\favicon.svg</code> , <code>icons.svg</code>	Ícones do sistema
(opcional) <code>src\</code> , <code>server.js</code> , <code>package.json</code>	Só quando a atualização inclui mudanças no backend

Regra de ouro

“ NUNCA apagar `data\`. NUNCA substituir o `logo.png` do cliente. NUNCA sobrescrever o `.env`.

Se você seguir o procedimento abaixo direitinho, esses três ficam preservados automaticamente.

Parte A — Gerar o pacote da nova versão (na sua máquina)

A1. Garantir que o código está atualizado

Confirmar que você tem a versão nova do código em `Sistema de Senha Novo\mvp-senhas-ui` (e do backend, se for o caso).

A2. Atualizar dependências (se mudaram)

```
cd "Sistema de Senha Novo\mvp-senhas-ui"  
npm install
```

Pular se nenhum pacote foi adicionado/atualizado.

A3. Gerar o novo build

```
npm run build
```

Esse comando gera a pasta `dist/` com o frontend novo.

A4. Empacotar só o novo `public`

1. Criar uma pasta vazia (em qualquer lugar), por exemplo `public-novo`.
2. Copiar **todo o conteúdo de** `mvp-senhas-ui\dist\` pra dentro dessa pasta `public-novo`.
3. Compactar `public-novo` em um `.zip`.

Esse zip é o que vai pro servidor.

“ Se o backend também mudou, ver Parte C (atualização de backend).

Parte B — Aplicar a atualização no servidor

B1. Avisar os usuários

A atualização exige reiniciar o servidor. **Atendimentos em andamento podem ser interrompidos.**

Recomendado fazer:

- fora do horário de atendimento (antes da abertura ou depois do fechamento);
- ou em horário de menor movimento (almoço, etc).

B2. Fazer backup ANTES de mexer (obrigatório)

Este é o passo **mais importante** do procedimento. Se algo der errado, esse backup é o que vai te salvar.

No servidor, copiar **todos os itens abaixo** pra uma pasta de backup datada (ex: `C:\Backup-Senhas\2026-05-28\`):

O que copiar	Por que é crítico
<code>C:\Sistemas\mvp-senhas\data\</code> (pasta inteira)	Contém <code>state.json</code> com TUDO do cliente (usuários, histórico, parâmetros)
<code>C:\Sistemas\mvp-senhas\public\logo.png</code>	Logo do cliente — sem esse arquivo, voltaria o logo padrão
<code>C:\Sistemas\mvp-senhas\.env</code>	Configuração local (porta, segredo)
<code>C:\Sistemas\mvp-senhas\public\</code> (pasta inteira)	Versão atual do frontend, pra rollback caso a nova versão dê problema

Script de backup pronto (executar como admin)

Salvar como `backup-pre-update.bat` e rodar antes da atualização:

```
@echo off
set DATA=%DATE:~6,4%-~3,2%-~0,2%-%TIME:~0,2%-~3,2%
set DESTINO=C:\Backup-Senhas\%DATA: =0%
echo Criando backup em %DESTINO%
mkdir "%DESTINO%"
xcopy /E /I /Y C:\Sistemas\mvp-senhas\data "%DESTINO%\data"
xcopy /E /I /Y C:\Sistemas\mvp-senhas\public "%DESTINO%\public"
copy C:\Sistemas\mvp-senhas\.env "%DESTINO%\env"
echo Backup concluído.
pause
```

Confirmar que a pasta `C:\Backup-Senhas\AAAA-MM-DD-HH-MM\` foi criada com `data\`, `public\` e `.env` dentro antes de prosseguir.

“ **Não pule esse passo.** Sem o backup, um erro na cópia ou um arquivo errado de logo significa retrabalho do cliente cadastrando tudo de novo.

B3. Parar o servidor

Se está rodando como serviço (NSSM)

```
nssm stop SistemaSenhas
```

Se está rodando via Agendador de Tarefas

- Abrir Agendador → localizar a tarefa → clicar com botão direito → "Finalizar".

Se está rodando em uma janela de Prompt aberta

- Fechar a janela do prompt, ou pressionar `Ctrl+C` dentro dela.

Confirmar que parou

Abrir o navegador no servidor e acessar `http://localhost:3030`. Deve dar erro de conexão (página não carrega).

B4. Substituir o conteúdo de `public\` (preservando o logo do cliente)

Esse é o passo crítico. O logo do cliente fica DENTRO da pasta `public\` (como `logo.png`), então temos que tomar cuidado pra não sobrescrevê-lo.

Procedimento manual (passo a passo)

1. **Guardar o logo do cliente fora da pasta `public\`:**

```
copy C:\Sistemas\mvp-senhas\public\logo.png C:\Temp\logo-cliente.png
```

(criar `C:\Temp\` se não existir)

2. **Apagar todo o conteúdo de `public\`:**

- Abrir `C:\Sistemas\mvp-senhas\public\` no Explorer;
- Selecionar tudo (`Ctrl+A`) → `Delete`.

3. **Extrair o zip da nova versão** e copiar todo o conteúdo pra dentro de `C:\Sistemas\mvp-senhas\public\`.

4. **Restaurar o logo do cliente por cima:**

```
copy /Y C:\Temp\logo-cliente.png C:\Sistemas\mvp-senhas\public\logo.png
```

O `/Y` confirma a sobrescrita automaticamente (pois o build trouxe um `logo.png` padrão que precisa ser substituído pelo do cliente).

5. **Apagar o arquivo temporário:**

```
del C:\Temp\logo-cliente.png
```

Script pronto (recomendado)

Salvar como `aplicar-update.bat` e rodar como admin **depois de extrair o zip novo numa pasta temporária** (ex: `C:\Temp\public-novo\`):

```
@echo off
set DEST=C:\Sistemas\mvp-senhas\public
set NOVO=C:\Temp\public-novo
set LOGO_TEMP=C:\Temp\logo-cliente-temp.png

if not exist "%NOVO%\index.html" (
    echo ERRO: %NOVO% nao contem index.html. Verificar caminho do build novo.
    pause
    exit /b 1
)

echo Preservando logo do cliente...
copy /Y "%DEST%\logo.png" "%LOGO_TEMP%"

echo Limpando %DEST%...
rmdir /S /Q "%DEST%"
mkdir "%DEST%"

echo Copiando nova versao...
xcopy /E /I /Y "%NOVO%" "%DEST%"

echo Restaurando logo do cliente...
copy /Y "%LOGO_TEMP%" "%DEST%\logo.png"
del "%LOGO_TEMP%"

echo Update aplicado.
pause
```

O que esse procedimento garante

Arquivo	Acontece o que
<code>public\index.html</code>	Substituído pelo novo
<code>public\assets*</code>	Substituído pelo novo (nomes mudam a cada build)
<code>public\favicon.svg</code> , <code>icons.svg</code>	Substituído pelo novo

Arquivo	Acontece o que
public\logo.png	Preservado do cliente (não vai pro logo padrão)
data\state.json (usuários, histórico, etc)	Intocado
.env	Intocado
src\, server.js	Intocado (atualização só de frontend)

“ **NÃO mexer** em `data\`, `src\`, `server.js`, `package.json`, `.env` — só o `public\` é substituído, e mesmo assim o `logo.png` dele é preservado.

B5. Religar o servidor

Se está como serviço (NSSM)

```
nssm start SistemaSenhas
```

Se está no Agendador

- Botão direito na tarefa → "Executar".

Se está manual

```
cd C:\Sistemas\mvp-senhas  
node server.js
```

B6. Validação obrigatória (confirma que nada foi perdido)

Abrir o navegador no servidor e acessar `http://localhost:3030`. Passar pela checklist completa antes de liberar:

Aparência geral

- A tela de login aparece e tem a nova aparência esperada.
- O logo do cliente aparece** (não é o logo padrão / "exemplo"). Se estiver errado, parar e refazer o passo B4.

- O nome do cliente/órgão aparece corretamente no topo.

Cadastros (não perder!)

Fazer login com um administrador existente e verificar:

- Funcionários:** todos os atendentes/admins existentes estão lá (Administração → Funcionários).
- Guichês:** a lista de guichês cadastrados está completa (Administração → Guichês).
- Salas:** a lista de salas está completa, com os atendentes vinculados (Administração → Salas).
- Vínculos atendente↔guichê e atendente↔sala** continuam corretos (ao abrir um atendente, o guichê/sala padrão dele permanece).

Parâmetros e configurações

- Configurações gerais:** nome da empresa, lógica de prioridade, reset automático, etc — todos com os valores que o cliente havia configurado.
- Motivos de prioridade** (se o cliente cadastrou): presentes em Administração → Configurações.

Histórico

- Histórico do dia anterior** (ou último dia de movimento) aparece em Histórico → filtros "7 dias" / "30 dias" / "Tudo".
- Auditoria** (Administração → Auditoria) mostra ações anteriores à atualização.

Funcionamento

- Emitir uma senha de teste na Recepção.
- Chamar a senha em Atendimento.
- Confirmar no Painel que a senha aparece e o som toca.
- Finalizar a senha.
- Confirmar registro em Histórico.

Caso algum dos itens acima falhe

PARAR a liberação imediatamente e:

1. Verificar se o `data/state.json` ainda existe e tem tamanho razoável (não 0 bytes):

```
dir C:\Sistemas\mvp-senhas\data\state.json
```

2. Se o arquivo sumiu ou está corrompido, executar o **rollback completo** (seção Plano B abaixo).
3. Se só o logo está errado, refazer só o passo B4.

Se tudo OK, avisar os usuários que podem voltar a trabalhar.

B7. Limpar cache do navegador dos usuários

Os navegadores costumam guardar a versão antiga em cache. Se o usuário disser que "continua igual" após a atualização:

- Pedir pra apertar `Ctrl+F5` (Windows) ou `Cmd+Shift+R` (Mac).
- Se persistir, limpar cache do navegador.

Geralmente o sistema já faz cache-busting automático (arquivos novos têm nome diferente do antigo), então isso é raro.

Parte C — Atualização que inclui mudanças no backend

Quando a atualização também muda código do servidor Node.js (arquivos `server.js`, `src/database.js`, `src/services.js`, etc).

“ **Verifique sempre o que mudou junto comigo antes de aplicar.** Mudanças no backend exigem mais cuidado e podem requerer migração de dados.

C1. Empacotar o backend novo

Na sua máquina, dentro de `Sistema de Senha Legado\mvp-senhas`:

Copiar pro pacote os arquivos:

- `server.js`
- pasta `src\` inteira
- `package.json` e `package-lock.json` (se mudaram dependências)

NÃO incluir no pacote:

- `data\` (essa é dos dados do cliente, não pode sobrescrever)
- `.env` (cada cliente tem o seu)
- `node_modules\` (será gerado no servidor)

Compactar esses arquivos junto com o `public-novo\`.

C2. Aplicar no servidor

Repetir B1-B3 (avisar, backup, parar).

Adicionalmente:

- Fazer cópia de segurança da pasta `src\` atual:

```
xcopy /E /I C:\Sistemas\mvp-senhas\src C:\Backup-Senhas\src-AAAA-MM-DD\
```

- Copiar a pasta `src\` nova por cima da antiga.
- Substituir `server.js` se mudou.
- Se `package.json` mudou, dentro de `C:\Sistemas\mvp-senhas` rodar:

```
npm install
```

Aplicar o `public\` novo conforme B4.

Religar o servidor (B5) e validar (B6).

C3. Migração de dados

Se a atualização exigir **mudança de estrutura no `state.json`**, isso será sempre comunicado por escrito e com instruções específicas. Não atualizar às cegas — em caso de dúvida, perguntar antes.

Plano B — Rollback (voltar pra versão anterior)

Se a nova versão deu problema, voltar pra anterior:

Rollback do frontend (volta a versão anterior do `public\`)

1. Parar o servidor (B3).
2. Apagar tudo de `C:\Sistemas\mvp-senhas\public\`.
3. Copiar tudo de `C:\Backup-Senhas\AAAA-MM-DD\public\` (do backup feito em B2) de volta pra `C:\Sistemas\mvp-senhas\public\`.
4. Religar o servidor (B5).

Resultado: volta tudo (inclusive `logo.png`) ao estado anterior à atualização.

Rollback só do logo (se o resto está OK)

Se a atualização funcionou mas o logo virou o padrão por engano:

```
copy /Y C:\Backup-Senhas\AAAA-MM-DD\public\logo.png C:\Sistemas\mvp-senhas\public\logo.png
```

Não precisa reiniciar o servidor — o navegador pode precisar de `Ctrl+F5` pra recarregar a imagem.

Rollback do backend

1. Parar o servidor.
2. Substituir `src\` pelo backup.
3. Substituir `server.js` pelo backup (se foi alterado).
4. Se o `package.json` foi alterado, restaurar o anterior e rodar `npm install`.
5. Religar.

Rollback dos dados (último recurso)

Se o `state.json` ficou corrompido por alguma operação:

1. Parar o servidor.
2. Renomear `data\state.json` atual pra `state-corrompido.json` (preservar pra análise).
3. Copiar `C:\Backup-Senhas\state-AAAA-MM-DD.json` pra `data\state.json`.
4. Religar.

“ **Atenção:** rollback de dados desfaz qualquer atendimento feito após o backup.
Usar só em casos críticos.

Checklist resumido da atualização

Antes (preparação)

- Avisar os usuários do horário da atualização.
- Backup completo:** pasta `data\`, arquivo `public\logo.png`, arquivo `.env`, pasta `public\` inteira (script `backup-pre-update.bat`).
- Confirmar que o backup foi criado e tem os arquivos esperados.

Durante (execução)

- Parar o servidor.
- Preservar o logo do cliente** (copiar `public\logo.png` pra local temporário).
- Apagar conteúdo de `public\`.
- Copiar conteúdo novo pra `public\`.
- Restaurar o logo do cliente** (copiar de volta sobrescrevendo o padrão).
- (Se aplicável) Atualizar `src\` e `server.js` e rodar `npm install`.
- Religar o servidor.

Depois (validação)

- Login funciona.
- Logo do cliente aparece** (não é o padrão).
- Nome da empresa/órgão correto no topo.

- Funcionários cadastrados ainda existem.
 - Guichês cadastrados ainda existem.
 - Salas cadastradas ainda existem, com atendentes vinculados.
 - Configurações gerais (prioridade, reset, etc) preservadas.
 - Histórico antigo aparece em "7 dias" / "30 dias" / "Tudo".
 - Auditoria mostra ações anteriores.
 - Emitir → chamar → finalizar uma senha de teste funcionou.
 - Painel mostra a senha com som.
 - Avisar os usuários que podem voltar.
-

Frequência recomendada de atualizações

- **Correções urgentes:** aplicar assim que disponíveis.
 - **Melhorias visuais / novas funcionalidades:** agrupar em janelas mensais, sempre fora do horário de atendimento.
 - **Mudanças no banco/backend:** planejar com pelo menos 1 semana de antecedência e testar em ambiente paralelo, se possível.
-

Em caso de problema

Se algo der errado:

1. Fazer rollback (procedimento acima).
 2. Anotar mensagens de erro, comportamento observado e horário.
 3. Reportar pra equipe de desenvolvimento junto com:
 - Versão anterior (que funcionava) e versão nova (que falhou).
 - Cópia do `state.json` (pra reprodução em ambiente isolado).
 - Logs do servidor (saída do `node server.js` ou logs do serviço NSSM).
-

Revisão #1

Criado 28 maio 2026 14:50:58 por Stephanie Correa

Atualizado 28 maio 2026 14:51:50 por Stephanie Correa