

Sistema de Senhas

Sistema de senhas Totem e Painel.

- [Manual de Instalação — Sistema de Senhas](#)
- [Manual de Atualização — Sistema de Senhas](#)

Manual de Instalação — Sistema de Senhas

Guia completo para instalar o sistema de senhas em um servidor do cliente pela primeira vez.

O sistema é composto por:

- **Backend:** pasta `mvp-senhas` (servidor Node.js com a API e o banco em arquivo JSON).
- **Frontend:** pasta `mvp-senhas-ui` (interface moderna em React, que precisa ser "compilada" antes de ir pro servidor).

Quando o sistema é instalado, o frontend já vem **embutido** dentro da pasta `mvp-senhas/public`. O servidor entrega tudo na mesma porta.

Parte A — Preparar o pacote (na sua máquina)

Você só faz isso uma vez, antes de levar o sistema pro servidor do cliente.

A1. Pré-requisitos

Na sua máquina precisa estar instalado:

- **Node.js LTS** (versão 20 ou superior): <https://nodejs.org/>

Confirme no prompt:

```
node -v  
npm -v
```

A2. Instalar dependências do frontend

Abrir o prompt na pasta `mvp-senhas-ui`:

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

Esse passo só precisa rodar uma vez (ou quando atualizar dependências).

A3. Gerar o build da interface

Ainda na pasta `mvp-senhas-ui`:

```
npm run build
```

Isso cria uma pasta `dist/` com o frontend pronto pra produção (HTML, CSS e JS otimizados).

A4. Substituir o `public` do backend

1. Apagar todo o conteúdo de `Sistema de Senha Legado\mvp-senhas\public\`.
2. Copiar **todo o conteúdo** da pasta `mvp-senhas-ui\dist\` pra dentro de `mvp-senhas\public\`.

Resultado: dentro de `mvp-senhas\public\` deve ter `index.html`, uma pasta `assets/` e demais arquivos do build.

A5. Compactar o pacote

Compactar a pasta `mvp-senhas` inteira em um `.zip`. Esse zip é o que vai pro servidor.

Parte B — Instalar no servidor do cliente

B1. Escolher o servidor

Use um computador que fique sempre ligado:

- servidor Windows da empresa;

- computador dedicado na recepção;
- mini PC ligado na rede;
- servidor virtual interno.

Evite notebook pessoal — se desligar, todos perdem acesso.

B2. Instalar Node.js no servidor

Baixar e instalar Node.js LTS (versão 20 ou superior):

<https://nodejs.org/>

Confirmar no Prompt de Comando do servidor:

```
node -v
```

B3. Copiar a pasta do sistema

Extrair o zip preparado na Parte A para um caminho fixo, por exemplo:

```
C:\Sistemas\mvp-senhas
```

B4. Criar o arquivo `.env`

Dentro de `C:\Sistemas\mvp-senhas`, criar um arquivo chamado `.env` (sem extensão `.txt`) com o conteúdo:

```
PORT=3030
APP_NAME=Atendimento
SESSION_SECRET=troque-por-um-texto-grande-e-aleatorio
COOKIE_SECURE=false
DATA_FILE=data/state.json
TRUST_PROXY=false
```

Atenção:

- Troque o `SESSION_SECRET` por um texto longo e aleatório (qualquer combinação de letras e números, ex: `kj8m2nP9xR4tQ7vW1yU6`).
- `COOKIE_SECURE=false` é o correto pra rede interna sem HTTPS. **Não mude para `true`** a menos que esteja servindo via HTTPS.

B5. Primeiro teste

No Prompt de Comando, dentro da pasta:

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

Se aparecer mensagem indicando que o servidor está rodando na porta `3030`, está correto.

Abrir no navegador do próprio servidor:

```
http://localhost:3030
```

Deve aparecer a tela de login.

B6. Primeiro acesso

Logar com o usuário administrador de suporte:

- **Usuário:** `admin`
- **Senha:** `Adm1nC&bi14863`

⚠ **NÃO trocar a senha do `admin`.**

Esse usuário e senha são **padrão de suporte da CEBI** — é o que garante que a equipe de suporte consiga acessar o sistema em produção a qualquer momento (pra resolver problema, aplicar correção, ajudar o cliente).

Em vez de trocar a senha do `admin`, o procedimento correto é:

1. Entrar **uma vez** com o `admin` (só pra criar usuários iniciais).
2. **Criar os administradores do cliente em Administração → Funcionários → Adicionar** (ver passo B11).
3. Daí em diante, o pessoal do cliente trabalha com os usuários próprios deles. O `admin` fica reservado pra suporte.

“ Se por qualquer motivo a senha do `admin` for trocada na produção, comunique a equipe de suporte imediatamente — caso contrário, perde-se o acesso de suporte naquele cliente.

B7. Liberar firewall

No Windows Firewall do servidor, liberar entrada **TCP na porta 3030**:

1. Abrir "Windows Defender Firewall com Segurança Avançada".
2. Regras de Entrada → Nova Regra → Porta → TCP → 3030 → Permitir conexão.
3. Aplicar a todos os perfis (Domínio, Particular, Público — ou conforme política do cliente).
4. Nomear a regra (ex: "Sistema de Senhas").

Sem isso, o sistema funciona no próprio servidor mas não abre nos outros computadores da rede.

B8. Acessar de outras máquinas

Em outros computadores da rede:

```
http://IP_DO_SERVIDOR:3030
```

Exemplos:

```
http://192.168.0.50:3030
```

```
http://SERVIDOR-SENHAS:3030
```

Pra descobrir o IP do servidor, no próprio servidor:

```
ipconfig
```

E procurar "IPv4" na seção da placa de rede ativa.

B9. Rodar automaticamente ao ligar o servidor

Em produção, o sistema precisa subir sozinho quando o servidor reinicia.

Opção simples — Agendador de Tarefas do Windows

1. Abrir "Agendador de Tarefas".
2. Criar tarefa (não criar tarefa básica).
3. **Geral**: marcar "Executar com privilégios mais altos".
4. **Disparadores**: novo → "Ao iniciar o computador".

5. **Ações:** nova → Programa: `node` — Argumentos: `server.js` — Iniciar em: `C:\Sistemas\mvp-senhas`.
6. **Configurações:** desmarcar "Parar a tarefa se executar por mais de...".
7. Salvar e testar reiniciando o servidor.

Opção mais robusta — NSSM (recomendada)

Permite rodar como serviço Windows de verdade, com restart automático em caso de queda.

1. Baixar NSSM: <https://nssm.cc/>
2. Abrir prompt como admin e executar `nssm install SistemaSenhas`.
3. Application Path: `C:\Program Files\nodejs\node.exe`
4. Startup directory: `C:\Sistemas\mvp-senhas`
5. Arguments: `server.js`
6. Salvar e iniciar: `nssm start SistemaSenhas`.

B10. Backup obrigatório

O sistema guarda os dados do cliente em **dois lugares distintos**:

O que	Contém
<code>C:\Sistemas\mvp-senhas\data\state.json</code>	Usuários, guichês, salas, configurações, prioridades, motivos, senhas emitidas, histórico, auditoria
<code>C:\Sistemas\mvp-senhas\public\logo.png</code>	Logo do cliente (enviado pelo Admin)
<code>C:\Sistemas\mvp-senhas\env</code>	Configuração local (porta, segredo de sessão)

Configurar backup diário automático **dos três itens acima**. Sugestões:

- Robocopy + Task Scheduler agendado pra rodar diariamente;
- Cópia pra outro disco / drive de rede;
- Backup corporativo (Veeam, etc) incluindo essas pastas/arquivos.

Manter pelo menos 7 cópias (uma por dia da última semana). Recomendado também 1 cópia mensal pra arquivamento.

Exemplo de comando de backup (Robocopy)

```
robocopy C:\Sistemas\mvp-senhas\data C:\Backup-Senhas\%DATE:~6,4%-~3,2%-~0,2%\data /E
copy C:\Sistemas\mvp-senhas\public\logo.png C:\Backup-Senhas\%DATE:~6,4%-~3,2%-
%DATE:~0,2%\logo.png
copy C:\Sistemas\mvp-senhas\env C:\Backup-Senhas\%DATE:~6,4%-~3,2%-~0,2%\env
```

Cria uma pasta com data do dia e copia os três artefatos.

B11. Configuração inicial dentro do sistema

Já logado como admin, configurar:

Funcionários

Administração → Funcionários:

- **Não mexer no usuário `admin`** — ele é reservado pra suporte da CEBI. Não trocar senha nem desativar.
- **Cadastrar pelo menos um administrador do cliente** (com login próprio, ex: nome do responsável pelo sistema na empresa). É esse usuário que o cliente vai usar pra gerenciar o sistema no dia a dia.
- Cadastrar os atendentes (perfil `atendente`).
- Cada usuário recebe login próprio e senha inicial (pelo menos 6 caracteres).
- Vincular cada atendente a um guichê e sala padrão (opcional, mas recomendado).
- Combinar com o cliente que o dia a dia deve ser feito com os usuários dele — o `admin` só é usado por suporte em casos excepcionais.

Guichês

Administração → Guichês:

- Cadastrar os guichês físicos (ex: "Guichê 1", "Guichê 2").
- Ativar/desativar conforme necessário.

Salas

Administração → Salas:

- Cadastrar as salas de atendimento.
- Vincular os atendentes que podem trabalhar em cada sala.

Configurações gerais

Administração → Configurações:

- Nome da empresa/órgão (aparece no painel e no topo da interface).
- Lógica de prioridade (FIFO / alternância configurável).
- Reset automático do dia (se quiser que zere automaticamente após X horas sem movimento).
- Outras opções (som, tempo de chamada, etc).

Logo do cliente

Administração → Configurações → Logo:

1. Clicar em "Carregar logo" (ou área equivalente).
2. Selecionar o arquivo de imagem do cliente (PNG recomendado, com fundo transparente).
3. Confirmar o upload.

O logo aparece automaticamente em:

- Topo da interface (todas as telas);
- Tela do Painel (TV);
- Tela do Totem/Recepção.

Importante saber pra atualizações futuras:

- O logo do cliente é gravado diretamente como `C:\Sistemas\mvp-senhas\public\logo.png` no servidor.
- Quando você for atualizar o sistema (substituir a pasta `public\`), **o logo do cliente precisa ser preservado** — caso contrário ele será sobrescrito pelo logo padrão que vem no pacote. Ver o **Manual de Atualização** pra o procedimento correto.
- Backup recomendado: incluir `public\logo.png` na rotina de backup junto com o `data\state.json`.

B12. Telas do sistema

Cada terminal abre a tela conforme a função:

Tela	URL/Caminho	Uso
Recepção	<code>http://IP:3030</code> → Recepção	Emitir senhas (totem ou recepcionista)
Atendimento	<code>http://IP:3030</code> → Atendimento	Atendente chama/finaliza senhas
Painel (TV)	<code>http://IP:3030</code> → Painel	TV grande mostrando senhas chamadas (com som)
Histórico	<code>http://IP:3030</code> → Histórico	Consulta histórico do dia
Administração	<code>http://IP:3030</code> → Administração	Só pra usuários admin

Painel na TV

1. Abrir o navegador na TV.
2. Acessar `http://IP_DO_SERVIDOR:3030`.
3. Fazer login (pode criar um usuário específico tipo "painel" com permissão apenas de visualização).

4. Entrar em **Painel**.
5. Clicar uma vez na tela pra ativar o som (browsers bloqueiam áudio sem interação do usuário).
6. Pressionar **F11** pra tela cheia.

B13. Validação final

Antes de entregar pro cliente, testar:

- Emitir uma senha comum na Recepção.
- Chamar a senha em Atendimento de outro computador.
- Ver a senha sendo anunciada no Painel (com som).
- Finalizar a senha.
- Conferir o registro em Histórico.
- Reiniciar o servidor e confirmar que o sistema sobe sozinho.
- Confirmar backup do `data\state.json`.

Resumo dos caminhos importantes

Arquivo/pasta	Para que serve
<code>C:\Sistemas\mvp-senhas\server.js</code>	Inicia o servidor
<code>C:\Sistemas\mvp-senhas\.env</code>	Configuração de porta, sessão, etc
<code>C:\Sistemas\mvp-senhas\data\state.json</code>	Banco de dados (fazer backup diário)
<code>C:\Sistemas\mvp-senhas\public\</code>	Interface (HTML/JS/CSS gerados do React)
<code>C:\Sistemas\mvp-senhas\src\</code>	Código do backend Node.js

Problemas comuns

"Funciona no servidor mas não abre nos outros computadores"

- Firewall não liberado (passo B7).
- IP do servidor mudou (servidor sem IP fixo). Considere reservar IP no DHCP.

"Login não funciona depois de algumas horas"

- O sistema mantém sessão por **8 horas** absolutas. Após esse tempo o usuário precisa fazer login de novo.
- Se o servidor reinicia, todas as sessões caem (comportamento esperado).

"O servidor não sobe ao reiniciar"

- Verificar se a tarefa do Agendador / serviço NSSM está configurada (passo B9).
- Conferir se o `node.exe` está no PATH ou se o caminho completo foi usado na configuração do serviço.

"Som do painel não funciona"

- Clicar uma vez na tela depois de abrir o Painel — é uma restrição dos navegadores.
- Conferir o volume do sistema operacional.

Limitações desta versão

Esta versão é adequada para **um cliente/unidade em rede interna**.

Não é ideal para:

- vários clientes diferentes no mesmo portal (multi-tenancy);
- exposição pública na internet sem HTTPS/proxy reverso;
- alto volume (centenas de atendentes simultâneos);
- integração com banco corporativo (Oracle/SQL Server);
- controle avançado de infraestrutura.

Para esses casos, planejar evolução pra arquitetura corporativa.

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).