

PrintControl - Documentacao Tecnica

Data: 2025-12-30

Responsavel: Sidertec / TI

1. Visao geral

PrintControl é um sistema leve para controle de impressões em ambiente Windows sem servidor de impressão. Ele coleta eventos locais de impressão nas estações e envia para um host central que armazena em SQLite e exibe um dashboard web com filtros, métricas e exportação CSV.

Princípios do projeto:

- Leve: baixo consumo de CPU/RAM no agente.
- Local: sem broadcast; o agente apenas envia para o host.
- Simples: manutenção fácil, poucos componentes.
- Windows-first: baseado no log do PrintService e serviços Windows.

2. Tecnologias usadas

Host:

- .NET 8 (ASP.NET Core Minimal API)
- SQLite (Microsoft.Data.Sqlite)
- HTML/CSS/JavaScript (dashboard)

Agente:

- .NET 8
- Windows Service (Microsoft.Extensions.Hosting.WindowsServices)
- Windows Event Log (EventLogWatcher)

Infra e ferramentas:

- WiX Toolset (geração de MSI)
- sc.exe (gerenciamento de serviços Windows)
- msieexec (instalação MSI)
- wevtutil (log de eventos)
- gpresult/gpupdate (diagnóstico de GPO)
- PowerShell (automação e testes)

3. Arquitetura e fluxo de dados

Componentes principais:

- 1) Agente (serviço Windows)
- 2) Host (API + dashboard web, serviço Windows)

3) SQLite (banco local do host)

Fluxo resumido:

- 1) Usuario imprime em uma estacao.
- 2) O Windows registra o evento 307 no log Microsoft-Windows-PrintService/Operational.
- 3) O Agent monitora esse log com EventLogWatcher.
- 4) O Agent extrai dados (usuario, maquina, impressora, paginas, bytes, data/hora).
- 5) O Agent envia um POST para o Host em /api/print-jobs.
- 6) O Host grava no SQLite e o dashboard exibe os dados.

Seguranca:

- Dashboard protegido por Basic Auth (DashboardUser e DashboardPassword).
- API de ingest pode usar X-Api-Key (ApiKey) se configurado.
- Recomendado: manter o host em rede interna.

4. Como o Host funciona (interno)

Tecnologias:

- ASP.NET Core (Minimal API)
- SQLite via Microsoft.Data.Sqlite
- Front-end simples (HTML/CSS/JS)

Pipeline do Host:

- 1) Le appsettings.json (PrintControlOptions).
- 2) Configura URLs de escuta (ListenUrls).
- 3) Inicializa banco de dados (PrintJobRepository.InitializeAsync).
- 4) Serve arquivos estaticos (wwwroot).
- 5) Exponde APIs REST para consultas e exportacoes.
- 6) Aplica Basic Auth no dashboard (nao bloqueia POST /api/print-jobs).

Endpoints principais:

- POST /api/print-jobs
- Entrada: PrintJobIngest
- Saida: { id }
- GET /api/print-jobs
- Filtros: from, to, user, machine, printer, minPages, maxPages, limit, offset
- GET /api/print-jobs/export
- Retorna CSV
- GET /api/stats/users
- GET /api/stats/printers
- GET /api/stats/summary

Banco de dados:

Tabela print_jobs:

- id (INTEGER PK AUTOINCREMENT)
- printed_at (TEXT ISO 8601 UTC)
- user_name (TEXT)
- machine_name (TEXT)
- printer_name (TEXT)
- pages (INTEGER)
- bytes (INTEGER)
- job_id (TEXT, opcional)
- document_name (TEXT, opcional)

Indices:

- printed_at, user_name, printer_name

5. Como o Agent funciona (interno)

Tecnologias:

- .NET 8
- Servico Windows (Microsoft.Extensions.Hosting.WindowsServices)
- EventLogWatcher (PrintService/Operational)

Pipeline do Agent:

- 1) Inicia como servico Windows (PrintControl.Agent).
- 2) Habilita o log PrintService/Operational (se possivel).
- 3) Escuta EventID 307.
- 4) Faz parse do evento (PrintEventParser):
 - Tenta XML (DocumentPrinted / EventData / UserData)
 - Usa fallback por index quando o evento nao tem nomes
- 5) Normaliza:
 - Maquina: remove prefixo \\ se existir
 - Impressora: sempre prefere IP quando encontrado (campo de endereco ou nome)
- 6) Ignora impressoras virtuais de PDF (qualquer nome contendo "PDF")
- 7) Envia para o host. Se falhar, grava em fila local (DiskQueue).
- 8) Periodicamente tenta reenviar a fila.

DiskQueue:

- Arquivo local (QueuePath) com JSON de jobs pendentes.
- Usa fallback em %LOCALAPPDATA% se o caminho principal falhar.

6. Configuracoes (appsettings.json)

Host:

- ApiKey: chave para validar ingest (X-Api-Key). Vazio = sem validacao.
- DatabasePath: caminho do SQLite.
- ListenUrls: URLs de escuta (ex: http://0.0.0.0:5080).
- DashboardUser / DashboardPassword: Basic Auth do dashboard.

Agent:

- HostUrl: URL do host.
- ApiKey: igual ao Host (se ativado).
- QueuePath: caminho do JSON de fila.
- FlushIntervalSeconds: intervalo de envio da fila.
- IncludeDocumentName: se true, envia nome do documento (padrao false).

7. Instalacao e deploy

Host (servidor):

1) Publicar (self-contained):

```
dotnet publish src\Host\PrintControl.Host.csproj -c Release -r win-x64 --self-contained true
-p:PublishSingleFile=true
```

O que faz: compila e gera um executavel que nao depende do runtime do .NET instalado na maquina.

2) Copiar a pasta publicada para C:\Controle Impressao\Host

O que faz: define um caminho fixo para o servico do host.

3) Editar appsettings.json

O que faz: configura URL, banco de dados e credenciais do dashboard.

4) Criar servico:

```
sc.exe create PrintControl.Host binPath= "C:\Controle Impressao\Host\PrintControl.Host.exe" start= auto
```

O que faz: cria um servico Windows com inicio automatico.

```
sc.exe start PrintControl.Host
```

O que faz: inicia o servico imediatamente.

5) Liberar firewall na porta configurada (ex: 5080)

O que faz: permite acesso HTTP ao dashboard na rede.

Agent (manual):

1) Publicar (self-contained):

```
dotnet publish src\Agent\PrintControl.Agent.csproj -c Release -r win-x64 --self-contained true
-p:PublishSingleFile=true
```

O que faz: gera o agent em formato standalone (sem exigir .NET runtime).

2) Copiar pasta para C:\Program Files (x86)\PrintControl\Agent

O que faz: padroniza o caminho do binario do servico.

3) Editar appsettings.json

O que faz: aponta o agent para o host e define API key (se houver).

4) Criar servico:

```
sc.exe create PrintControl.Agent binPath= "C:\Program Files  
(x86)\PrintControl\Agent\PrintControl.Agent.exe" start= auto
```

O que faz: cria o serviço do agente.

```
sc.exe start PrintControl.Agent
```

O que faz: inicia o serviço.

Agente via MSI (GPO):

- 1) Gerar MSI com build-agent-msi.ps1 (WiX).

O que faz: publica o agente self-contained e empacota no MSI.

- 2) Copiar MSI para share.

O que faz: permite que as estações acessem o instalador.

- 3) Aplicar GPO de Software Installation ou script de startup.

O que faz: distribui o agente para todas as máquinas alvo.

- 4) Verificar se o serviço PrintControl.Agent existe e está em execução.

O que faz: confirma instalação e operação do agente.

8. Operação e manutenção

Serviços:

- Host: PrintControl.Host
- Agente: PrintControl.Agent

Comandos úteis:

- sc.exe query PrintControl.Host

O que faz: mostra status do serviço do host.

- sc.exe query PrintControl.Agent

O que faz: mostra status do serviço do agente.

- sc.exe stop PrintControl.Agent

O que faz: para o serviço do agente.

- sc.exe start PrintControl.Agent

O que faz: inicia o serviço do agente.

Banco:

- Caminho default: C:\ProgramData\PrintControl\Host\prints.db
- Backup: copiar o arquivo com o serviço parado.
- Reset total: parar o serviço do host, apagar prints.db, iniciar o serviço.

Atualizações:

- Rebuild Host e Agente.
- Atualizar MSI no share.
- Reaplicar GPO (ou reinstalar manualmente).

9. Troubleshooting

Problemas comuns:

1) Erro 1920 no MSI:

- O servico nao inicia. Normalmente causado por binario incompleto.
- Confirme se o MSI contem um exe self-contained (~70MB) e nao apenas o stub (150KB).

2) Nao aparece impressao:

- Verifique se o log PrintService/Operational esta habilitado.
- Confirme conectividade com o host (porta 5080).

3) Paginas = 0:

- Alguns drivers nao informam paginas no EventID 307.

4) Dashboard pede senha:

- Verifique DashboardUser/DashboardPassword no appsettings do host.

Logs e diagnostico:

- diagnose-agent.cmd gera log completo em C:\Windows\Temp\PrintControl_diagnose.log
- PrintControl_install.log mostra logs do MSI

Comandos de diagnostico (com comentario):

- gpresult /r /scope computer

O que faz: mostra quais GPOs estao aplicados na maquina.

- gpupdate /force

O que faz: forza a atualizacao imediata das politicas de grupo.

- wevtutil gl Microsoft-Windows-PrintService/Operational

O que faz: mostra se o log de impressao esta habilitado.

- wevtutil qe Microsoft-Windows-PrintService/Operational /q:"*[System[(EventID=307)]]" /f:xml /c:1

O que faz: mostra o ultimo evento 307 (impressao).

- Test-NetConnection -ComputerName 192.168.22.5 -Port 5080

O que faz: testa conectividade com o host (porta 5080).

- msixexec /i "\\\ti-fabiano\SHARE\PrintControl.Agent.msi" /qn /l*v C:\Windows\Temp\PrintControl_install.log

O que faz: instala o MSI em modo silencioso e gera log detalhado.

10. Referencia de arquivos

Raiz:

- .gitignore: lista de pastas/arquivos ignorados (dist, bin, obj, .vs, exe/pdb).
- README.md: resumo do sistema e funcionalidades.
- diagnose-agent.cmd: utilitario para diagnostico rapido (copia de installer\Agent\diagnose-agent.cmd).
- PrintControl_diagnose.log: log gerado pelo diagnostico (pode ser apagado).
- docs\DEPLOYMENT.md: guia rapido de publicacao/installacao.
- docs\PrintControl_Documentacao.md: esta documentacao (fonte).

src\Host:

- Program.cs: inicializa o host, configura Basic Auth, rotas e repositorio.
- ApiKeyValidator.cs: valida X-Api-Key quando ApiKey e configurada.
- appsettings.json: configuracao do host.
- Data\PrintJobRepository.cs: acesso ao SQLite, cria tabela e executa queries.
- Data\CsvWriter.cs: exporta dados para CSV.
- Models\PrintControlOptions.cs: opcoes de configuracao do host.
- Models\PrintJobIngest.cs: modelo de ingestao (POST).
- Models\PrintJobRecord.cs: modelo persistido no banco.
- Models\PrintJobFilter.cs: filtros de consulta.
- Models\PrintJobFilterParser.cs: parser de querystring para filtros.
- Models\AggregateRow.cs: linhas agregadas para graficos.
- Models\SummaryRow.cs: totais resumidos.
- wwwroot\index.html: estrutura do dashboard.
- wwwroot\app.js: logica de UI, filtros, chamadas ao backend.
- wwwroot\styles.css: estilo do dashboard.
- PrintControl.Host.csproj: projeto do host e dependencias.

src\Agent:

- Program.cs: inicializa o servico, HttpClient e PrintWatcherService.
- appsettings.json: configuracao do agent.
- Models\PrintControlOptions.cs: opcoes do agent.
- Models\PrintJobPayload.cs: payload enviado ao host.
- Services\PrintWatcherService.cs: monitora eventos e envia jobs.
- Services\PrintEventParser.cs: extrai dados do EventID 307.
- Services\DiskQueue.cs: fila local em disco para resiliencia.
- PrintControl.Agent.csproj: projeto do agent e dependencias.

installer\Agent:

- PrintControl.Agent.wxs: definicao WiX do MSI (arquivos e servico).
- build-agent-msi.ps1: script que publica o agent e gera o MSI.
- install-printcontrol-agent.cmd: script de instalacao (GPO).
- diagnose-agent.cmd: diagnostico do agent em ambiente Windows.

dist (gerado):

- dist\Host: publicacao do host (self-contained).
- dist\Agent: publicacao do agent (self-contained).
- dist\Installer\PrintControl.Agent.msi: pacote MSI do agent.
- dist\Installer\PrintControl.Agent.wixpdb: metadata do MSI.

11. Checklist de continuidade

Antes de mexer:

- Verifique appsettings.json do host e agent.
- Verifique onde o MSI esta publicado no share.
- Verifique se o host esta acessivel na porta configurada.

Ao atualizar:

- Rebuild do host e agent.
- Gerar MSI novo (build-agent-msi.ps1).
- Substituir o MSI no share.
- Forcar update via GPO.

Ao dar suporte:

- Rodar diagnose-agent.cmd.
- Checar Event Viewer (PrintService/Operational).
- Validar que o servico PrintControl.Agent esta rodando.