

# DeDupe — Informe para Codex · Iteración 3

**Autor:** Víctor (por medio de ChatGPT)

**Scope:** Modo `-NonInteractive` en CLI, selector de backend en GUI, y pruebas de estrés para `Logging`

## 16) Objetivo de esta iteración

1. Habilitar **CLI-first opcional**: agregar modo `-NonInteractive` a `DeDupe.ps1` para que acepte parámetros y publique progreso a `<LogPath>.progress` (JSONL).
2. Permitir que la **GUI elija backend**: `ThreadJob` (in-proc) o CLI externo (`pwsh -File DeDupe.ps1 -NonInteractive ...`).
3. Definir **pruebas de estrés** para `DeDupe.Logging` (rotación + concurrencia) y recomendaciones finales.

## 17) Patch completo — `DeDupe.ps1` con `-NonInteractive` + progreso en `.progress`

Este archivo reemplaza al actual **DeDupe.ps1** (mantiene el wizard original; añade rama no interactiva).

El progreso en ambos modos se escribe en `<LogPath>.progress` (una línea JSON por snapshot).

```
#requires -Version 7.2
[CmdletBinding()]
param(
    # --- Modo no interactivo ---
    [switch]$NonInteractive,
    [switch]$Run,
    [switch]$Recurse,
    [string]$Path,
    [string]$QuarantinePath,
    [string]$LogPath,
    [string]$ExportSummaryPath,
    [ValidateSet('Oldest','Newest','ShortestPath')] [string]$Keep = 'Oldest',
    [ValidateSet('BySize','Flat')] [string]$QuarantineLayout = 'BySize',
    [int]$DegreeOfParallelism = 0,
    [int]$BlockSizeKB = 1024,
    [switch]$IncludeHidden,
    [switch]$Verify
)
```

```

Set-StrictMode -Version Latest
$ErrorActionPreference = 'Stop'

# ----- Defaults -----
$defaultData = 'C:\.CODEX\DATA'
if (-not (Test-Path -LiteralPath $defaultData)) { $defaultData = (Get-
Location).Path }
$defaultQuarantine = Join-Path $env:LOCALAPPDATA 'DeDupe\Quarantine'
$defaultLog         = Join-Path $env:LOCALAPPDATA 'DeDupe\logs\actions.jsonl'
$reportsDir         = Join-Path $env:LOCALAPPDATA 'DeDupe\reports'
if (-not (Test-Path -LiteralPath $reportsDir)) { New-Item -ItemType Directory -
Path $reportsDir -Force | Out-Null }
$defaultReport      = Join-Path $reportsDir ("DeDupe_Summary_{0}.json" -f (Get-
Date).ToString('yyyyMMdd_HH:mm:ss'))

# ----- Importar módulos -----
$modDir = Join-Path $PSScriptRoot 'Modules'
$mods = @(
    'DeDupe.Metrics.Ultra',
    'DeDupe.Logging',
    'DeDupe.Hashing.MetricsAdapter',
    'DeDupe.Grouping',
    'DeDupe.DedupeByHash',
    'DeDupe.Quarantine',
    'DeDupe.Pipeline'
)
foreach ($m in $mods) {
    $psm1 = Join-Path $modDir ("{0}.psm1" -f $m)
    if (Test-Path -LiteralPath $psm1) { . $psm1 } else { Import-Module (Join-Path
$modDir ("{0}.psd1" -f $m)) -Force -ErrorAction Stop }
}

function Ensure-Dir([string]$dir){ if (-not [string]::IsNullOrEmpty($dir) -
and -not (Test-Path -LiteralPath $dir)) { New-Item -ItemType Directory -Path
$dir -Force | Out-Null } }

try {
    if ($NonInteractive) {
        # ===== Rama NO INTERACTIVA =====
        $isRun          = [bool]$Run
        $doRecurse       = [bool]$Recurse
        $path            = if ($Path) { $Path } else
{ $defaultData }
        $qPath           = if ($QuarantinePath) { $QuarantinePath } else
{ $defaultQuarantine }
        $logPath         = if ($LogPath) { $LogPath } else
{ $defaultLog }
        $keep            = $Keep
    }
}

```

```

$layout      = $QuarantineLayout
$dop         = $DegreeOfParallelism
$blkKB       = $BlockSizeKB
$verify      = [bool]$Verify
$includeHidden = [bool]$IncludeHidden

Ensure-Dir (Split-Path -Parent $qPath)
Ensure-Dir (Split-Path -Parent $logPath)

$doExport = -not [string]::IsNullOrWhiteSpace($ExportSummaryPath)
$outJson  = if ($doExport) { $ExportSummaryPath } else { $null }
if ($doExport) { Ensure-Dir (Split-Path -Parent $outJson) }

# Progreso a <LogPath>.progress
$progressPath = "$logPath.progress"
try { Set-Content -LiteralPath $progressPath -Value '' -Encoding UTF8 -
Force } catch {}
$tick = { param($s) try { ($s|ConvertTo-Json -Depth 6)+
[Environment]::NewLine | Add-Content -LiteralPath $using:progressPath -Encoding
UTF8 } catch {} }

$summary = Invoke-DeDupePipeline `
    -Path $path `
    -Recurse:$doRecurse `
    -IncludeHidden:$includeHidden `
    -AllowZeroByte:$false `
    -Keep $keep `
    -QuarantinePath $qPath `
    -LogPath $logPath `
    -QuarantineLayout $layout `
    -DegreeOfParallelism $dop `
    -BlockSizeKB $blkKB `
    -ReportIntervalMs 750 `
    -Verify:$verify `
    -Run:([bool]$isRun) `
    -OnTick $tick `
    -Confirm:$false

if ($doExport) {
    ($summary | ConvertTo-Json -Depth 8) | Set-Content -LiteralPath $outJson -
Encoding UTF8
}

# Salida mínima a consola
"==== RESUMEN ====="
$summary | Format-Table -AutoSize | Out-Host
"Log: $($summary.log_path_effective)"
"Quarantine: $($summary.quarantine_path)"

```

```

        if ($doExport) { "Reporte JSON: $outJson" }
        "Listo."
    }
    else {
        # ===== Wizard INTERACTIVO (sin cambios visibles para el usuario) =====
        function New-Choice {
            param([Parameter(Mandatory)][string]$Title,[Parameter(Mandatory)][string]
$Message,[Parameter(Mandatory)][string[]]$Options,[int]$Default = 0)
            $list = New-Object
System.Collections.ObjectModel.Collection[System.Management.Automation.Host.ChoiceDescription]
            foreach ($opt in $Options)
{ $list.Add([System.Management.Automation.Host.ChoiceDescription]::new($opt)) }
            return $Host.UI.PromptForChoice($Title, $Message, $list, $Default)
        }
        function Prompt-Path {
            param([Parameter(Mandatory)][string]$Prompt,[Parameter(Mandatory)][string]
$Default,[switch]$MustExist,[switch]$EnsureDirectory)
            while ($true) {
                $val = Read-Host "$Prompt [`$ENTER = default]`n Default: $Default"
                if ([string]::IsNullOrEmpty($val)) { $val = $Default }
                try {
                    $expanded = [Environment]::ExpandEnvironmentVariables($val)
                    if ($MustExist) {
                        if (Test-Path -LiteralPath $expanded) { return $expanded }
                        Write-Warning "No existe: $expanded"
                    } elseif ($EnsureDirectory) {
                        $dir = $expanded; Ensure-Dir $dir; return $dir
                    } else { return $expanded }
                } catch { Write-Warning $_.Exception.Message }
            }
        }

        $modeIdx = New-Choice -Title "Modo de ejecución" -Message "Elige acción:" -
Options @("&Dry-run (no borra)", "&Run (aplica cambios)") -Default 0
        $isRun = ($modeIdx -eq 1)
        $recIdx = New-Choice -Title "Recorrer subcarpetas" -Message "¿Procesar
recursivamente?" -Options @("&Sí", "&No") -Default 0
        $doRecurse = ($recIdx -eq 0)

        $path = Prompt-Path -Prompt "Ruta a limpiar (Path)" -Default $defaultData
-MustExist
        $qPath = Prompt-Path -Prompt "Carpeta de cuarentena (QuarantinePath)" -
Default $defaultQuarantine -EnsureDirectory
        $logPath= Prompt-Path -Prompt "Ruta del log JSONL (LogPath)" -Default
$defaultLog
        $expIdx = New-Choice -Title "Exportar resumen JSON" -Message "¿Guardar un
JSON con el resumen?" -Options @("&Sí", "&No") -Default 1
        $doExport = ($expIdx -eq 0)
    }
}

```

```

    $outJson = if ($doExport) { Prompt-Path -Prompt "Ruta del reporte JSON" -
Default $defaultReport } else { $null }
    if ($outJson) { Ensure-Dir (Split-Path -Parent $outJson) }

    # Valores por defecto "pro"
    $keep = 'Oldest'; $layout = 'BySize'; $dop = 0; $blkKB = 1024; $verify =
    $false; $includeHidden = $false; $allowZero = $false; $reportIntervalMs = 750

    # Progreso a <LogPath>.progress
    $progressPath = "$logPath.progress"
    try { Set-Content -LiteralPath $progressPath -Value '' -Encoding UTF8 -
Force } catch {}
    $tick = { param($s) try { ($s|ConvertTo-Json -Depth 6)+
[Environment]::NewLine | Add-Content -LiteralPath $using:progressPath -Encoding
UTF8 } catch {} }

    $summary = Invoke-DeDupePipeline `
        -Path $path `
        -Recurse:$doRecurse `
        -IncludeHidden:$includeHidden `
        -AllowZeroByte:$allowZero `
        -Keep $keep `
        -QuarantinePath $qPath `
        -LogPath $logPath `
        -QuarantineLayout $layout `
        -DegreeOfParallelism $dop `
        -BlockSizeKB $blkKB `
        -ReportIntervalMs $reportIntervalMs `
        -Verify:$verify `
        -Run:([bool]$isRun) `
        -OnTick $tick `
        -Confirm:$false

    "`n==== RESUMEN ====="
    $summary | Format-Table -AutoSize | Out-Host
    "`nLog: $($summary.log_path_effective)"
    "`nQuarantine: $($summary.quarantine_path)"
    if ($doExport) {
        ($summary | ConvertTo-Json -Depth 8 -Compress) | Set-Content -LiteralPath
$outJson -Encoding UTF8
        "`nReporte JSON: $outJson"
    }
    "`nListo.`n"
}
}
catch { Write-Error $_.Exception.Message; exit 1 }

```

### Uso (ejemplos):

```
# Dry-run no interactivo (auto DOP, layout BySize, keep Oldest)
pwsh -NoProfile -File .\DeDupe.ps1 -NonInteractive -Recurse -Path C:\Data -
QuarantinePath "$env:LOCALAPPDATA\DeDupe\Quarantine" -LogPath
"$env:LOCALAPPDATA\DeDupe\logs\actions.jsonl"

# Run real + export de resumen
pwsh -NoProfile -File .\DeDupe.ps1 -NonInteractive -Run -Recurse -Path C:\Data -
QuarantinePath "$env:LOCALAPPDATA\DeDupe\Quarantine" -LogPath
"$env:LOCALAPPDATA\DeDupe\logs\actions.jsonl" -ExportSummaryPath
"$env:LOCALAPPDATA\DeDupe\reports\run.json"
```

## 18) Cambios en la GUI — Selector de backend (ThreadJob / CLI externo)

Añade un **ComboBox** para seleccionar backend. Si es **CLI externo**, la GUI construye los parámetros y ejecuta `DeDupe.ps1 -NonInteractive ...` dentro de un **ThreadJob** (para no bloquear WPF). Los *tailers* siguen leyendo `<LogPath>.progress` y el JSONL.

### 18.1 Control en UI (junto a DOP/Bloque):

```
# En la sección de opciones avanzadas:
$opts2.Children.Add((New-Label 'Backend:'))
$cmbBackend = New-Combo @('ThreadJob (in-proc)', 'CLI externo') 0
$opts2.Children.Add($cmbBackend)
```

### 18.2 Rama en **Run-Job**:

```
$useCli = ($cmbBackend.SelectedIndex -eq 1)
$progressFile = "$log.progress"; $lastProgSize = 0
try { Set-Content -LiteralPath $progressFile -Value '' -Encoding UTF8 -Force }
catch {}
if (-not $progTimer.IsEnabled) { $progTimer.Start() }

$btnDry.IsEnabled = $false; $btnRun.IsEnabled = $false

if (-not $useCli) {
    # == Backend ThreadJob (in-proc), igual que en Iteración 2 ==
    # ... (mismo bloque Start-ThreadJob que ya dejamos)
}
else {
```

```
# === Backend CLI externo ===
$args = @(
    '-NoProfile','-File', (Join-Path $base 'DeDupe.ps1'), '-NonInteractive',
    "-Run:$apply",
    "-Recurse:$recurse",
    "-IncludeHidden:$hidden",
    "-Verify:$verify",
    '-Path', $path,
    '-QuarantinePath', $q,
    '-LogPath', $log,
    '-Keep', $keep,
    '-QuarantineLayout', $layout,
    '-DegreeOfParallelism', $dop,
    '-BlockSizeKB', $blk
)
if ($btnExportChk.IsChecked) { $args += @('-ExportSummaryPath',
$btnExportPath.Text) }

$job = Start-ThreadJob -Name 'DeDupe.CLI' -ArgumentList (@{ Args=$args }) -
ScriptBlock {
    param($a) Start-Process pwsh -WindowStyle Hidden -ArgumentList $a.Args -Wait
}

Register-ObjectEvent -InputObject $job -EventName StateChanged -Action {
    try { if ($eventArgs.JobStateInfo.State -in
@('Completed','Failed','Stopped')) { $using:progTimer.Stop();
$using:btnDry.IsEnabled=$true; $using:btnRun.IsEnabled=$true; Append-LogLine
("CLI state: {0}" -f $eventArgs.JobStateInfo.State) } } catch {}
} | Out-Null
}
```

**Importante:** el `.progress` lo genera ahora **la CLI** gracias al patch de `DeDupe.ps1`. La GUI únicamente lo lee.

## 19) Pruebas de estrés — DeDuPe.Logging

### 19.1 Rotación por tamaño (QPS alto, proceso único)

```
Import-Module (Join-Path $PSScriptRoot 'Modules/DeDuPe.Logging.psd1') -Force
$log = Join-Path $env:LOCALAPPDATA 'DeDupe\logs\stress.jsonl'
$logger = New-JsonlLogger -Path $log -RotateAtMB 5
$sw = [Diagnostics.Stopwatch]::StartNew()
1..200000 | ForEach-Object {
    Write-Jsonl -Logger $logger -Object @{ ts=(Get-Date).ToString('o'); i=$_;
```

```

msg='stress' }
}
$sw.Stop(); "Wrote 200k in $($sw.Elapsed)"; ls (Split-Path $log)

```

**Esperado:** archivos `stress.jsonl`, `stress.jsonl.bak`, etc. sin errores y con tiempos razonables.

## 19.2 Concurrencia multi-hilo (mismo proceso)

```

Import-Module (Join-Path $PSScriptRoot 'Modules/DeDuPe.Logging.psd1') -Force
$log = Join-Path $env:LOCALAPPDATA 'DeDuPe\logs\multi.jsonl'
$logger = New-JsonlLogger -Path $log -RotateAtMB 5
1..4 | ForEach-Object { Start-ThreadJob -ArgumentList $logger,{ param($lg)
    1..50000 | ForEach-Object { Write-Jsonl -Logger $lg -Object @{ ts=(Get-
Date).ToString('o'); t=$PID; i=$_ } }
} }
Get-Job | Wait-Job | Receive-Job; ls (Split-Path $log)

```

**Esperado:** sin excepciones; rotación coherente; tamaño final  $\approx 4 \times 50k$  líneas.

## 19.3 Concurrencia multi-proceso (riesgo)

```

$log = Join-Path $env:LOCALAPPDATA 'DeDuPe\logs\multi-proc.jsonl'
$code = @"
Import-Module `"$($PSScriptRoot)\Modules\DeDuPe.Logging.psd1`" -Force
$lg = New-JsonlLogger -Path `"$log`" -RotateAtMB 5
1..40000 | % { Write-Jsonl -Logger $lg -Object @{ ts=(Get-Date).ToString('o');
pid=$PID; i=$_ } }
"@
1..3 | % { Start-Process pwsh -NoProfile -WindowStyle Hidden -ArgumentList @('-
Command', $code) }

```

**Observación:** si aparecen fallos ocasionales de sharing, considerar **mutex con nombre** (ver recomendaciones).

## 20) Recomendaciones (Logging)

- **Hoy (suficiente):** diseño actual con `Add-Content` y rotación por tamaño es estable para 1 proceso.
- **Si hay múltiples procesos:**
- Añadir `[Mutex]` **con nombre** (por ejemplo, `Global/DeDuPe.Log.<hash(ruta)>`) antes de escribir/rotar; liberar al final.



- O migrar a `StreamWriter` **persistente** por proceso, abierto con `FileShare.ReadWrite`, y reabrir tras rotación.
  - **AutoFlush:** habilitar `AutoFlush = $true` si se usa `StreamWriter`.
- 

## 21) Criterios de aceptación

- CLI con `-NonInteractive` produce salida correcta y **genera** `<LogPath>.progress`.
  - GUI puede alternar entre **ThreadJob** y **CLI externo** sin congelarse.
  - `Logging` rota sin errores en 200k eventos y se mantiene consistente en 4×50k hilos.
  - Documentación y ejemplos incluidos en este Canvas.
- 

## 22) Siguientes pasos

- (Opcional) Botón **Cancelar:** `Stop-ThreadJob` + señal de cancelación en pipeline.
- (Opcional) Métrica avanzada: registrar *MB/s* y *EPS* por fase (hashing, compare, quarantine) en `.progress`.