DeDupe — Integración SSOT (Single Source of Truth) · Iteración 6

Autor: Víctor (por medio de ChatGPT)
Scope: Integrar el SSOT (TOML/JSON) en CLI/GUI con cancelación, progreso, backend conmutable, y CI.

---

28) Objetivo

Dejar **parches completos** que ya leen/escriben configuración determinista (TOML/JSON), respetan los contratos de progreso/cancelación, y permiten a la GUI operar en **ThreadJob** o **CLI externo** sin tocar el hilo UI.

---

29) Nuevo módulo — `Modules/DeDuPe.Config.psm1`

> SSOT loader: Defaults → (TOML|JSON) → Merge con CLI/GUI → Config efectivo.

```
#requires -Version 7.2
Set-StrictMode -Version Latest
$ErrorActionPreference = 'Stop'

function Get-DeDupeDefaults {
  $reportsDir = Join-Path $env:LOCALAPPDATA 'DeDupe/reports'
  if (-not (Test-Path -LiteralPath $reportsDir)) { New-Item -ItemType Directory
-Path $reportsDir -Force | Out-Null }
  [pscustomobject]@{
    version = @{ schema = 1; app = '1.0.0' }
    paths   = [pscustomobject]@{
      data           = (if (Test-Path 'C:\.CODEX\.DATA'){ 'C:\.CODEX\.DATA' }
else { (Get-Location).Path })
      quarantine     = (Join-Path $env:LOCALAPPDATA 'DeDupe/Quarantine')
      log_jsonl      = (Join-Path $env:LOCALAPPDATA 'DeDupe/logs/actions.jsonl')
      export_summary = $false
      summary_path   = (Join-Path $reportsDir ("DeDupe_Summary_{0}.json" -f
(Get-Date).ToString('yyyyMMdd_HHmmss')))
    }
    run     = [pscustomobject]@{
      mode                  = 'dry-run'   # 'dry-run' | 'run'
      recurse               = $true
      include_hidden        = $false
      allow_zero_byte       = $false
      verify_bytes          = $false
      keep                    = 'Oldest'    # 'Oldest'|'Newest'|'ShortestPath'
      quarantine_layout       = 'BySize'    # 'BySize'|'Flat'
      degree_of_parallelism = 0
      block_size_kb           = 1024
      report_interval_ms      = 500
    }
    backend = [pscustomobject]@{
      type       = 'cli'     # 'threadjob' | 'cli'
      pwsh_path  = 'pwsh'
      cli_script = (Join-Path (Split-Path -Parent $PSScriptRoot) 'DeDupe.ps1')
    }
    progress = [pscustomobject]@{ path = ''; write_snapshots = $true }
    cancel   = [pscustomobject]@{ path = ''; semantics = 'presence' }
    logging  = [pscustomobject]@{ rotate_at_mb = 32; writer = 'AddContent' }
    ui       = [pscustomobject]@{ theme='dark'; color_bg='#202020';
color_panel='#252525'; color_text='#E0E0E0'; color_accent='#3A96DD';
text_view_init='empty' }
    ci       = [pscustomobject]@{ enabled=$true; smoketest='tests/ci-
smoketest.ps1' }
  }
}

function Resolve-DeDupePaths([object]$cfg){
  $p = $cfg.paths
  if ([string]::IsNullOrWhiteSpace($cfg.progress.path)) { $cfg.progress.path =
"$($p.log_jsonl).progress" }
  if ([string]::IsNullOrWhiteSpace($cfg.cancel.path))   { $cfg.cancel.path   =
"$($p.log_jsonl).cancel" }
  $cfg
}

function Read-DeDupeConfig {
  param([Parameter()] [string]$Path)
  $cfg = Get-DeDupeDefaults
  if ([string]::IsNullOrWhiteSpace($Path)) {
    $Path = Join-Path (Split-Path -Parent $PSScriptRoot) 'dedupe.toml'
  }
  if (Test-Path -LiteralPath $Path) {
    $ext = [IO.Path]::GetExtension($Path).ToLowerInvariant()
    try {
      if ($ext -eq '.toml') {
        if (Get-Command -Name ConvertFrom-Toml -ErrorAction SilentlyContinue) {
          $toml = Get-Content -LiteralPath $Path -Raw -Encoding UTF8 |
ConvertFrom-Toml
        } else {
          Write-Warning "ConvertFrom-Toml no disponible; use dedupe.json o
instale PowerShell 7.4+. Usando defaults."
          $toml = $null
        }
        if ($toml) { $cfg = Merge-Objects -Base $cfg -Overlay ($toml) }
      } elseif ($ext -eq '.json') {
        $json = Get-Content -LiteralPath $Path -Raw -Encoding UTF8 |
ConvertFrom-Json -Depth 10
        $cfg  = Merge-Objects -Base $cfg -Overlay $json
      }
    } catch { Write-Warning "Fallo al leer config '$Path': $
($_.Exception.Message)" }
  }
  Resolve-DeDupePaths $cfg
}

function Merge-Objects {
  param([Parameter(Mandatory)]$Base,[Parameter(Mandatory)]$Overlay)
  $result = $Base | ConvertTo-Json -Depth 10 | ConvertFrom-Json
  foreach ($prop in ($Overlay | Get-Member -MemberType NoteProperty | Select-
Object -ExpandProperty Name)){
    $ov = $Overlay.$prop
    if ($null -eq $ov) { continue }
    if ($result.PSObject.Properties.Name -contains $prop) {
      if ($ov -is [System.Management.Automation.PSObject]) {
        $result.$prop = Merge-Objects -Base $result.$prop -Overlay $ov
      } else { $result.$prop = $ov }
    } else { $result | Add-Member -NotePropertyName $prop -NotePropertyValue
$ov }
  }
  $result
}

function Write-DeDupeConfigToml {
  param([Parameter(Mandatory)]$Config,[Parameter(Mandatory)][string]$Path)
  # Writer TOML mínimo (simple; sin arrays anidados complejos)
  $sb = [System.Text.StringBuilder]::new()
  foreach ($top in
'version','paths','run','backend','progress','cancel','logging','ui','ci'){
    $sec = $Config.$top
    if ($null -ne $sec){ [void]$sb.AppendLine("[$top]")
      foreach($p in ($sec | Get-Member -MemberType NoteProperty | Select-Object
-ExpandProperty Name)){
        $v = $sec.$p
        if ($v -is [bool]) { $s = ($v.ToString().ToLower()) }
        elseif ($v -is [int] -or $v -is [long]) { $s = "$v" }
        else { $escaped = ($v -replace '\\','\\\\' -replace '"','\"'); $s = '"'
+ $escaped + '"' }
        [void]$sb.AppendLine("$p = $s")
      }
      [void]$sb.AppendLine()
    }
  }
  $dir = Split-Path -Parent $Path; if (-not (Test-Path -LiteralPath $dir)) {
New-Item -ItemType Directory -Path $dir -Force | Out-Null }
  Set-Content -LiteralPath $Path -Value $sb.ToString() -Encoding UTF8
}

Export-ModuleMember -Function *
```

---

30) CLI — `DeDupe.ps1` actualizado (lee config, respeta SSOT)

> Añade `-ConfigPath` y mergea con parámetros. Mantiene *wizard* cuando no se usa `-NonInteractive`.

```
#requires -Version 7.2
[CmdletBinding()]
param(
  [switch]$NonInteractive,
  [string]$ConfigPath,
  [switch]$Run,
  [switch]$Recurse,
  [string]$Path,
  [string]$QuarantinePath,
  [string]$LogPath,
  [string]$ExportSummaryPath,
  [ValidateSet('Oldest','Newest','ShortestPath')] [string]$Keep,
  [ValidateSet('BySize','Flat')] [string]$QuarantineLayout,
  [int]$DegreeOfParallelism,
  [int]$BlockSizeKB,
  [switch]$IncludeHidden,
  [switch]$Verify
)

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

$modDir = Join-Path $PSScriptRoot 'Modules'
Import-Module (Join-Path $modDir 'DeDuPe.Config.psm1') -Force
Import-Module (Join-Path $modDir 'DeDuPe.Metrics.Ultra.psd1') -Force
Import-Module (Join-Path $modDir 'DeDuPe.Logging.psd1')       -Force
Import-Module (Join-Path $modDir 'DeDuPe.Hashing.MetricsAdapter.psd1') -Force
Import-Module (Join-Path $modDir 'DeDuPe.Grouping.psd1')      -Force
Import-Module (Join-Path $modDir 'DeDuPe.DedupeByHash.psd1')  -Force
Import-Module (Join-Path $modDir 'DeDuPe.Quarantine.psd1')    -Force
```

```
Import-Module (Join-Path $modDir 'DeDuPe.Pipeline.psd1')      -Force

function Apply-Overrides($cfg){
  if ($PSBoundParameters.ContainsKey('Run'))                { $cfg.run.mode =
(if ($Run) { 'run' } else { 'dry-run' }) }
  if ($PSBoundParameters.ContainsKey('Recurse'))            { $cfg.run.recurse =
[bool]$Recurse }
  if ($PSBoundParameters.ContainsKey('Path'))               { $cfg.paths.data =
$Path }
  if ($PSBoundParameters.ContainsKey('QuarantinePath'))     {
$cfg.paths.quarantine = $QuarantinePath }
  if ($PSBoundParameters.ContainsKey('LogPath'))            {
$cfg.paths.log_jsonl = $LogPath }
  if ($PSBoundParameters.ContainsKey('ExportSummaryPath'))  {
$cfg.paths.export_summary = $true; $cfg.paths.summary_path =
$ExportSummaryPath }
  if ($PSBoundParameters.ContainsKey('Keep'))               { $cfg.run.keep =
$Keep }
  if ($PSBoundParameters.ContainsKey('QuarantineLayout'))   {
$cfg.run.quarantine_layout = $QuarantineLayout }
  if ($PSBoundParameters.ContainsKey('DegreeOfParallelism')){
$cfg.run.degree_of_parallelism = $DegreeOfParallelism }
  if ($PSBoundParameters.ContainsKey('BlockSizeKB'))        {
$cfg.run.block_size_kb = $BlockSizeKB }
  if ($PSBoundParameters.ContainsKey('IncludeHidden'))      {
$cfg.run.include_hidden = [bool]$IncludeHidden }
  if ($PSBoundParameters.ContainsKey('Verify'))             {
$cfg.run.verify_bytes  = [bool]$Verify }
  $cfg = Resolve-DeDupePaths $cfg
  return $cfg
}

try {
  $cfg = Read-DeDupeConfig -Path $ConfigPath
  $cfg = Apply-Overrides $cfg

  if (-not $NonInteractive) {
    # mantener wizard clásico (omitido aquí por brevedad); usa $cfg como
defaults y luego llama a Run-Core con esos valores
    Write-Host 'Ejecutando modo interactivo con defaults de config...'
  }

  $isRun   = ($cfg.run.mode -eq 'run')
  $path    = $cfg.paths.data
  $qPath   = $cfg.paths.quarantine
  $logPath = $cfg.paths.log_jsonl
  $doExport= [bool]$cfg.paths.export_summary
  $outJson = if ($doExport) { $cfg.paths.summary_path } else { $null }
  foreach ($d in @((Split-Path -Parent $qPath),(Split-Path -Parent $logPath),
(if ($outJson){ Split-Path -Parent $outJson }))) {
    if ($d) { if (-not (Test-Path -LiteralPath $d)) { New-Item -ItemType
Directory -Path $d -Force | Out-Null } }
  }

  $progressPath = if ([string]::IsNullOrWhiteSpace($cfg.progress.path)) {
"$logPath.progress" } else { $cfg.progress.path }
  try { Set-Content -LiteralPath $progressPath -Value '' -Encoding UTF8 -Force }
catch {}
  $tick = { param($s) try { $line = $s | ConvertTo-Json -Depth 6; if ($line -
notmatch '"type"') { $line = '{"type":"progress",'+ $line.TrimStart('{') }
($line)+[Environment]::NewLine | Add-Content -LiteralPath $using:progressPath -
Encoding UTF8 } catch {} }

  $summary = Invoke-DeDupePipeline `
    -Path $path `
    -Recurse:$($cfg.run.recurse) `
    -IncludeHidden:$($cfg.run.include_hidden) `
    -AllowZeroByte:$($cfg.run.allow_zero_byte) `
    -Keep $($cfg.run.keep) `
    -QuarantinePath $qPath `
    -LogPath $logPath `
    -QuarantineLayout $($cfg.run.quarantine_layout) `
    -DegreeOfParallelism $($cfg.run.degree_of_parallelism) `
    -BlockSizeKB $($cfg.run.block_size_kb) `
    -ReportIntervalMs $($cfg.run.report_interval_ms) `
    -Verify:$($cfg.run.verify_bytes) `
    -Run:([bool]$isRun) `
    -OnTick $tick `
    -Confirm:$false

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

  # snapshot final de fases (best-effort)
  try {
    $events=@{}; $bytes=@{}
    if (Test-Path -LiteralPath $logPath) { Get-Content -LiteralPath $logPath -
Encoding UTF8 | % { try { $j=$_|ConvertFrom-Json } catch {return};
$evt=$j.event; if ($evt){ $events[$evt]=1+($events[$evt]|%{$_}) }; $b=
if($j.bytes){[int64]$j.bytes}elseif($j.size){[int64]$j.size}else{$null}; if($b -
ne $null){ $k= if($evt){$evt}else{'total'}; $bytes[$k]=($bytes[$k]+$b) } } }
    @{ type='phaseSummary'; ts=(Get-Date).ToString('o'); events=$events;
bytes=$bytes } | ConvertTo-Json -Depth 6 | Add-Content -LiteralPath
$progressPath -Encoding UTF8
  } catch {}
  '==== RESUMEN ===='
  $summary | Format-Table -AutoSize | Out-Host
  "Log: $($summary.log_path_effective)"
  "Quarantine: $($summary.quarantine_path)"
  if ($doExport) { "Reporte JSON: $outJson" }

} catch { Write-Error $_.Exception.Message; exit 1 }
```

---

31) GUI — `DeDuPe.GUI.ps1` (SSOT + cancel + selector backend CLI por defecto)

> Archivo completo listo para reemplazo. **Default backend = CLI externo** siguiendo el SSOT. Lee `dedupe.toml` en arranque y precarga controles.

```
#requires -Version 7.2
using namespace System
using namespace System.IO
using namespace System.Windows
using namespace System.Windows.Controls
using namespace System.Windows.Threading
using namespace System.Windows.Media

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

$base  = Split-Path -Parent $PSCommandPath
$modDir= Join-Path $base 'Modules'
foreach ($m in
'DeDuPe.Config','DeDuPe.Metrics.Ultra','DeDuPe.Logging','DeDuPe.Grouping','DeDuPe.DedupeByHash','
{
  $psd1 = Join-Path $modDir ("{0}.psd1" -f $m)
  if (Test-Path -LiteralPath $psd1) { Import-Module $psd1 -Force } else {
Import-Module (Join-Path $modDir ("{0}.psm1" -f $m)) -Force }
}
Import-Module ThreadJob -ErrorAction SilentlyContinue | Out-Null

Add-Type -AssemblyName PresentationCore, PresentationFramework, WindowsBase
Add-Type -AssemblyName System.Windows.Forms

# ---- UI ----
$win = New-Object Window; $win.Title = 'DeDupe — GUI'; $win.Width = 920;
$win.Height = 720
$bc = New-Object BrushConverter
```

```
$bg=[SolidColorBrush]$bc.ConvertFrom('#FF202020'); $fg=[SolidColorBrush]
$bc.ConvertFrom('#FFE0E0E0'); $bg2=[SolidColorBrush]
$bc.ConvertFrom('#FF252525'); $acc=[SolidColorBrush]$bc.ConvertFrom('#FF3A96DD')
$win.Background=$bg; $win.Foreground=$fg

$grid = New-Object Grid; $grid.Margin='12'
0..9 | % { [void]$grid.RowDefinitions.Add((New-Object RowDefinition)) }
$grid.RowDefinitions[9].Height='*'

function New-Label([string]$t){ $tb=New-Object TextBlock; $tb.Text=$t;
$tb.Margin='0,0,6,0'; $tb.VerticalAlignment='Center'; $tb.Foreground=$fg; return
$tb }
function New-Combo([string[]]$items,[int]$idx){ $cb=New-Object ComboBox; $items
| % { [void]$cb.Items.Add($_) }; $cb.SelectedIndex=$idx; $cb.Width=180;
$cb.Margin='0,0,16,0'; return $cb }
function New-Text([string]$text,[int]$w){ $t=New-Object TextBox; $t.Text=$text;
$t.Width=$w; $t.Margin='0,0,16,0'; $t.Background=$bg2; $t.Foreground=$fg; return
$t }

function Row([int]$i,[UIElement]$e){ [Grid]::SetRow($e,$i);
$null=$grid.Children.Add($e) }

# Cargar config SSOT
$cfg = Read-DeDupeConfig
$cfg = Resolve-DeDupePaths $cfg

# Filas de paths
$sp1=New-Object StackPanel -Property @{Orientation='Horizontal'}
$sp1.Children.Add((New-Label 'Ruta origen:'));     $tbPath = New-Text
$cfg.paths.data 520; $sp1.Children.Add($tbPath)
$btnBrowse1=New-Object Button -Property @{Content='Browse…';Margin='6,0,0,0'};
$sp1.Children.Add($btnBrowse1)
Row 0 $sp1

$sp2=New-Object StackPanel -Property @{Orientation='Horizontal'}
$sp2.Children.Add((New-Label 'Cuarentena:'));      $tbQ = New-Text
$cfg.paths.quarantine 520; $sp2.Children.Add($tbQ)
$btnBrowse2=New-Object Button -Property @{Content='Browse…';Margin='6,0,0,0'};
$sp2.Children.Add($btnBrowse2)
Row 1 $sp2

$sp3=New-Object StackPanel -Property @{Orientation='Horizontal'}
$sp3.Children.Add((New-Label 'Log JSONL:'));       $tbLog = New-Text
$cfg.paths.log_jsonl 520; $sp3.Children.Add($tbLog)
$btnBrowse3=New-Object Button -Property @{Content='Browse…';Margin='6,0,0,0'};
$sp3.Children.Add($btnBrowse3)
Row 2 $sp3
```

```
# Opciones
$sp4=New-Object StackPanel -Property @{Orientation='Horizontal'}
$chkRec=New-Object CheckBox -Property
@{Content='Recursivo';IsChecked=$cfg.run.recurse;Margin='0,0,16,0';Foreground=$fg}
$chkVer=New-Object CheckBox -Property @{Content='Verificar (byte a
byte)';IsChecked=$cfg.run.verify_bytes;Margin='0,0,16,0';Foreground=$fg}
$chkHid=New-Object CheckBox -Property @{Content='Incluir
ocultos';IsChecked=$cfg.run.include_hidden;Margin='0,0,16,0';Foreground=$fg}
$sp4.Children.Add($chkRec);$sp4.Children.Add($chkVer);
$sp4.Children.Add($chkHid); Row 3 $sp4

$sp5=New-Object StackPanel -Property @{Orientation='Horizontal'}
$sp5.Children.Add((New-Label 'Conservar:')); $cbKeep=New-Combo
@('Oldest','Newest','ShortestPath')
(@('Oldest','Newest','ShortestPath').IndexOf($cfg.run.keep));
$sp5.Children.Add($cbKeep)
$sp5.Children.Add((New-Label 'Diseño:'));    $cbLay=New-Combo @('BySize','Flat')
(@('BySize','Flat').IndexOf($cfg.run.quarantine_layout));
$sp5.Children.Add($cbLay)
$sp5.Children.Add((New-Label 'DOP:'));       $tbDop=New-Text ([string]
$cfg.run.degree_of_parallelism) 60; $sp5.Children.Add($tbDop)
$sp5.Children.Add((New-Label 'Bloque KB:')); $tbBlk=New-Text ([string]
$cfg.run.block_size_kb) 80; $sp5.Children.Add($tbBlk)
$sp5.Children.Add((New-Label 'Backend:'));   $cbBk =New-Combo @('ThreadJob (in-
proc)','CLI externo') (if ($cfg.backend.type -eq 'cli'){1}else{0});
$sp5.Children.Add($cbBk)
Row 4 $sp5

# Progreso
$sp6=New-Object StackPanel -Property @{Orientation='Vertical'}
$bar = New-Object ProgressBar -Property
@{Minimum=0;Maximum=100;Height=18;Foreground=$acc}
$lbl = New-Object TextBlock  -Property @{Text='ETA
—';Margin='0,6,0,0';Foreground=$fg}
$sp6.Children.Add($bar); $sp6.Children.Add($lbl); Row 5 $sp6

# Botones
$sp7=New-Object StackPanel -Property @{Orientation='Horizontal'}
$btnDry=New-Object Button -Property
@{Content='Simulación';Width=140;Background=$bg2;Foreground=$fg}
$btnRun=New-Object Button -Property
@{Content='Ejecutar';Width=140;Margin='8,0,0,0';Background=$bg2;Foreground=$fg}
$chkExp=New-Object CheckBox -Property @{Content='Exportar resumen
JSON';IsChecked=$cfg.paths.export_summary;Margin='16,6,0,0';Foreground=$fg}
$tbOut =New-Object TextBox -Property
@{Text=$cfg.paths.summary_path;Width=320;Margin='8,0,0,0';Background=$bg2;Foreground=$fg}
$btnOut=New-Object Button -Property @{Content='…';Width=28;Margin='6,0,0,0'}
$btnCfgSave=New-Object Button -Property @{Content='Guardar
```

```
config';Margin='16,0,0,0';Width=140;Background=$bg2;Foreground=$fg}
$btnCancel = New-Object Button -Property
@{Content='Cancelar';Margin='8,0,0,0';Width=120;Background=$bg2;Foreground=$fg;IsEnabled=$false}
$sp7.Children.Add($btnDry);$sp7.Children.Add($btnRun);
$sp7.Children.Add($chkExp);$sp7.Children.Add($tbOut);$sp7.Children.Add($btnOut);
$sp7.Children.Add($btnCfgSave);$sp7.Children.Add($btnCancel)
Row 6 $sp7

# Visor de logs
$txt = New-Object TextBox -Property
@{Margin='0,10,0,0';VerticalScrollBarVisibility='Auto';TextWrapping='NoWrap';AcceptsReturn=$true;
Row 9 $txt

$win.Content=$grid

# Navegación
function Browse-Folder { $dlg=New-Object
System.Windows.Forms.FolderBrowserDialog; if ($dlg.ShowDialog() -eq 'OK'){
$dlg.SelectedPath } }
function Browse-FileSave{ param([string]$Filter='JSON files|*.json|All files|
*.*') $dlg=New-Object System.Windows.Forms.SaveFileDialog; $dlg.Filter=$Filter;
if ($dlg.ShowDialog() -eq 'OK'){ $dlg.FileName } }
$btnBrowse1.Add_Click({ $p=Browse-Folder; if($p){ $tbPath.Text=$p } })
$btnBrowse2.Add_Click({ $p=Browse-Folder; if($p){ $tbQ.Text=$p } })
$btnBrowse3.Add_Click({ $p=Browse-Folder; if($p){ $tbLog.Text=(Join-Path $p
'actions.jsonl') } })
$btnOut.Add_Click   ({ $p=Browse-FileSave; if($p){ $tbOut.Text=$p } })

# Helpers progreso/log
function Update-UIProgress($s){ if ($null -eq $s) { return }; $pct=
if($s.percent){ [int][math]::Floor($s.percent)} else {0};
$bar.Value=[Math]::Max(0,[Math]::Min(100,$pct)); $eta= if($s.eta_sec){"$
($s.eta_sec)s"} else {'—'}; $mbs= if($s.mbps){"$($s.mbps) MB/s"} else {'0 MB/
s'}; $lbl.Text = "$mbs | ETA $eta | $($s.items_done) done" }
function Append-Log($line){ if (-not [string]::IsNullOrWhiteSpace($line)) {
$txt.AppendText($line+[Environment]::NewLine); $txt.ScrollToEnd() } }

$tailLogTimer = New-Object DispatcherTimer -Property @{
Interval=[TimeSpan]::FromMilliseconds(750) }
$tailProgTimer= New-Object DispatcherTimer -Property @{
Interval=[TimeSpan]::FromMilliseconds(250) }
$lastLog=0L; $lastProg=0L; $progressPath=$cfg.progress.path;
$cancelPath=$cfg.cancel.path

$tailLogTimer.Add_Tick({ try { $p=$tbLog.Text;
if([string]::IsNullOrWhiteSpace($p) -or -not (Test-Path -LiteralPath $p)){
return }; $fi=[IO.FileInfo]::new($p); if($fi.Length -le $lastLog){ return };
$fs=[IO.File]::Open($p,'Open','Read','ReadWrite'); try{ $fs.Position=$lastLog;
$sr=New-Object IO.StreamReader($fs); while(-not $sr.EndOfStream){ Append-Log
($sr.ReadLine()) } } finally { $lastLog=$fi.Length; $fs.Dispose() } } catch
{} })
$tailProgTimer.Add_Tick({ try { $pp=$progressPath;
if([string]::IsNullOrWhiteSpace($pp) -or -not (Test-Path -LiteralPath $pp)){
return }; $fi=[IO.FileInfo]::new($pp); if($fi.Length -le $lastProg){ return };
$fs=[IO.File]::Open($pp,'Open','Read','ReadWrite'); try{ $fs.Position=$lastProg;
$sr=New-Object IO.StreamReader($fs); while(-not $sr.EndOfStream){
$line=$sr.ReadLine(); if(-not [string]::IsNullOrWhiteSpace($line)){ try{
$snap=$line|ConvertFrom-Json } catch { continue }; if($snap.type -eq 'progress')
{ Update-UIProgress $snap } else { Append-Log $line } } } } finally {
$lastProg=$fi.Length; $fs.Dispose() } } catch {} })

# Cancelación
$btnCancel.Add_Click({ try { if ($cancelPath) { Set-Content -LiteralPath
$cancelPath -Value '1' -Encoding ASCII -Force; Append-Log 'Cancel requested…';
$btnCancel.IsEnabled=$false } } catch {} })

# Guardar config (TOML)
$btnCfgSave.Add_Click({ try { $cfg.paths.data=$tbPath.Text;
$cfg.paths.quarantine=$tbQ.Text; $cfg.paths.log_jsonl=$tbLog.Text;
$cfg.paths.export_summary=[bool]$chkExp.IsChecked;
$cfg.paths.summary_path=$tbOut.Text; $cfg.run.recurse=[bool]$chkRec.IsChecked;
$cfg.run.verify_bytes=[bool]$chkVer.IsChecked; $cfg.run.include_hidden=[bool]
$chkHid.IsChecked; $cfg.run.keep=@('Oldest','Newest','ShortestPath')
[$cbKeep.SelectedIndex]; $cfg.run.quarantine_layout=@('BySize','Flat')
[$cbLay.SelectedIndex]; $cfg.run.degree_of_parallelism=[int]$tbDop.Text;
$cfg.run.block_size_kb=[int]$tbBlk.Text; $cfg.backend.type=
(if($cbBk.SelectedIndex -eq 1){'cli'} else {'threadjob'}); $cfg = Resolve-
DeDupePaths $cfg; Write-DeDupeConfigToml -Config $cfg -Path (Join-Path $base
'dedupe.toml'); Append-Log 'Config guardada (dedupe.toml)' } catch {
[System.Windows.MessageBox]::Show($_.Exception.ToString(),'Error','OK','Error')
| Out-Null } })

# Run core
function Run-Core([bool]$apply){
  try {
    $cfg.paths.data=$tbPath.Text; $cfg.paths.quarantine=$tbQ.Text;
$cfg.paths.log_jsonl=$tbLog.Text; $cfg.paths.export_summary=[bool]
$chkExp.IsChecked; $cfg.paths.summary_path=$tbOut.Text; $cfg.run.recurse=[bool]
$chkRec.IsChecked; $cfg.run.verify_bytes=[bool]$chkVer.IsChecked;
$cfg.run.include_hidden=[bool]$chkHid.IsChecked;
$cfg.run.keep=@('Oldest','Newest','ShortestPath')[$cbKeep.SelectedIndex];
$cfg.run.quarantine_layout=@('BySize','Flat')[$cbLay.SelectedIndex];
$cfg.run.degree_of_parallelism=[int]$tbDop.Text; $cfg.run.block_size_kb=[int]
$tbBlk.Text; $cfg.run.mode= (if ($apply) {'run'} else {'dry-run'});
$cfg.backend.type=(if($cbBk.SelectedIndex -eq 1){'cli'} else {'threadjob'});
$cfg=Resolve-DeDupePaths $cfg
```

```
    $txt.Clear(); $lastLog=0; $lastProg=0; $progressPath=$cfg.progress.path;
$cancelPath=$cfg.cancel.path
    try { if (Test-Path -LiteralPath $progressPath) { Remove-Item -LiteralPath
$progressPath -Force } } catch {}
    try { if (Test-Path -LiteralPath $cancelPath)   { Remove-Item -LiteralPath
$cancelPath   -Force } } catch {}
    Set-Content -LiteralPath $progressPath -Value '' -Encoding UTF8 -Force
    if (-not $tailLogTimer.IsEnabled) { $tailLogTimer.Start() }
    if (-not $tailProgTimer.IsEnabled){ $tailProgTimer.Start() }

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

    if ($cfg.backend.type -eq 'cli') {
      $args = @('-NoProfile','-File', $cfg.backend.cli_script, '-
NonInteractive', '-ConfigPath', (Join-Path $base 'dedupe.toml'))
      $job = Start-ThreadJob -Name 'DeDupe.CLI' -ArgumentList (@{ Args=$args;
Pwsh=$cfg.backend.pwsh_path }) -ScriptBlock { param($a) Start-Process $a.Pwsh -
WindowStyle Hidden -ArgumentList $a.Args -Wait }
    } else {
      $a = @{
        Path=$cfg.paths.data; Recurse=$cfg.run.recurse;
Hidden=$cfg.run.include_hidden; Keep=$cfg.run.keep; QPath=$cfg.paths.quarantine;
LPath=$cfg.paths.log_jsonl; Layout=$cfg.run.quarantine_layout;
Dop=$cfg.run.degree_of_parallelism; BlkKB=$cfg.run.block_size_kb;
Verify=$cfg.run.verify_bytes; Apply=($cfg.run.mode -eq 'run'); ModDir=$modDir;
PFile=$cfg.progress.path; Export=$cfg.paths.export_summary;
OutPath=$cfg.paths.summary_path; CFile=$cfg.cancel.path
      }
      $job = Start-ThreadJob -Name 'DeDupe.Run' -ArgumentList ($a) -ScriptBlock
{
        param($a)
        foreach($m in
'DeDuPe.Metrics.Ultra','DeDuPe.Logging','DeDuPe.Grouping','DeDuPe.DedupeByHash','DeDuPe.Quarantin
{ Import-Module (Join-Path $a.ModDir ("{0}.psd1" -f $m)) -Force }
        $tick = { param($s) if (Test-Path -LiteralPath $using:a.CFile) { throw
[System.OperationCanceledException]::new('Canceled by user') } ; ($s|ConvertTo-
Json -Depth 6)+[Environment]::NewLine | Add-Content -LiteralPath $using:a.PFile
-Encoding UTF8 }
        $sum = Invoke-DeDupePipeline -Path $a.Path -Recurse:$a.Recurse -
IncludeHidden:$a.Hidden -AllowZeroByte:$false -Keep $a.Keep -QuarantinePath
$a.QPath -LogPath $a.LPath -QuarantineLayout $a.Layout -DegreeOfParallelism
$a.Dop -BlockSizeKB $a.BlkKB -ReportIntervalMs 500 -Verify:$a.Verify -Run:
$a.Apply -OnTick $tick -Confirm:$false
        if ($a.Export) { try { $dir=Split-Path -Parent $a.OutPath; if (-not
(Test-Path -LiteralPath $dir)){ New-Item -ItemType Directory -Path $dir -Force |
Out-Null }; ($sum|ConvertTo-Json -Depth 8)|Set-Content -LiteralPath $a.OutPath -
Encoding UTF8 } catch {} }
        # snapshot de fases
        try { $events=@{};$bytes=@{}; if (Test-Path -LiteralPath $a.LPath){ Get-
Content -LiteralPath $a.LPath -Encoding UTF8 | % { try{$j=$_|ConvertFrom-Json}
catch{return}; $evt=$j.event; if($evt){$events[$evt]=1+($events[$evt]|%{$_})};
$b= if($j.bytes){[int64]$j.bytes}elseif($j.size){[int64]$j.size}else{$null};
if($b -ne $null){ $k= if($evt){$evt}else{'total'}; $bytes[$k]=($bytes[$k]+
$b) } } }; @{type='phaseSummary'; ts=(Get-Date).ToString('o'); events=$events;
bytes=$bytes } | ConvertTo-Json -Depth 6 | Add-Content -LiteralPath $a.PFile -
Encoding UTF8 } catch {}
        [pscustomobject]@{ Ok=$true; When=(Get-Date).ToString('u') }
      }
    }

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

  } catch {
[System.Windows.MessageBox]::Show($_.Exception.ToString(),'Error','OK','Error')
| Out-Null; $btnDry.IsEnabled=$true; $btnRun.IsEnabled=$true;
$btnCancel.IsEnabled=$false }
}

$btnDry.Add_Click({ Run-Core $false })
$btnRun.Add_Click({ Run-Core $true })

[void]$win.ShowDialog()
```

---

32) Muestras SSOT (colocar en raíz)

`dedupe.toml`

```
[version]
schema = 1
app = "1.0.0"

[paths]
data = ""
quarantine = ""
```

```toml
log_jsonl = ""
export_summary = true
summary_path = ""

[run]
mode = "dry-run"
recurse = true
include_hidden = false
allow_zero_byte = false
verify_bytes = false
keep = "Oldest"
quarantine_layout = "BySize"
degree_of_parallelism = 0
block_size_kb = 1024
report_interval_ms = 500

[backend]
type = "cli"
pwsh_path = "pwsh"
cli_script = "./DeDupe.ps1"

[progress]
path = ""
write_snapshots = true

[cancel]
path = ""
semantics = "presence"

[logging]
rotate_at_mb = 32
writer = "AddContent"

[ui]
theme = "dark"
color_bg = "#202020"
color_panel = "#252525"
color_text = "#E0E0E0"
color_accent = "#3A96DD"
text_view_init = "empty"

[ci]
enabled = true
smoketest = "tests/ci-smoketest.ps1"
```

`dedupe.json` **(fallback)**

```
{
  "version": { "schema": 1, "app": "1.0.0" },
  "paths": { "data": "", "quarantine": "", "log_jsonl": "", "export_summary":
true, "summary_path": "" },
  "run": { "mode": "dry-run", "recurse": true, "include_hidden": false,
"allow_zero_byte": false, "verify_bytes": false, "keep": "Oldest",
"quarantine_layout": "BySize", "degree_of_parallelism": 0, "block_size_kb":
1024, "report_interval_ms": 500 },
  "backend": { "type": "cli", "pwsh_path": "pwsh", "cli_script": "./
DeDupe.ps1" },
  "progress": { "path": "", "write_snapshots": true },
  "cancel": { "path": "", "semantics": "presence" },
  "logging": { "rotate_at_mb": 32, "writer": "AddContent" },
  "ui": { "theme": "dark", "color_bg": "#202020", "color_panel": "#252525",
"color_text": "#E0E0E0", "color_accent": "#3A96DD", "text_view_init": "empty" },
  "ci": { "enabled": true, "smoketest": "tests/ci-smoketest.ps1" }
}
```

---

33) CI de humo (actualizado)

> Igual que en Iteración 4, pero pasa `-ConfigPath dedupe.json` o `dedupe.toml` si la acción corre con PS 7.4+.

```
name: dedupe-ci
on: { push: { branches: [ main ] }, pull_request: { branches: [ main ] } }
jobs:
  smoke:
    runs-on: windows-latest
    steps:
      - uses: actions/checkout@v4
      - name: PowerShell version
        run: pwsh -NoProfile -Command "$PSVersionTable.PSVersion"
      - name: Create sample config
        run: |
          Set-Content -LiteralPath dedupe.json -Value '{"paths":
{"export_summary":true}}'
      - name: Smoke test (NonInteractive + config)
        run: |
          pwsh -NoProfile -File .\DeDupe.ps1 -NonInteractive -ConfigPath .
\dedupe.json -Recurse -Path $env:USERPROFILE\dedupe-ci -QuarantinePath
"$env:LOCALAPPDATA\DeDupe\Quarantine" -LogPath
"$env:LOCALAPPDATA\DeDupe\logs\actions.jsonl"
```

```
          if (-not (Test-Path "$env:LOCALAPPDATA\DeDupe\logs\actions.jsonl"))
{ throw 'No log' }
          if (-not (Test-Path
"$env:LOCALAPPDATA\DeDupe\logs\actions.jsonl.progress")) { throw 'No progress' }
```

---

34) Criterios de aceptación

1) CLI con `-ConfigPath` respeta TOML (si `ConvertFrom-Toml` disponible) y siempre admite JSON como fallback.
2) GUI arranca leyendo `dedupe.toml` (o defaults), precarga controles y **no bloquea** al ejecutar.
3) `.progress` recibe `type=progress` durante ejecución y `type=phaseSummary` al finalizar.
4) Botón **Cancelar** funciona y limpia estado.

¿Necesitas que deje los archivos como **.ps1/.psm1** listos para copiar/pegar en el repo? Puedo adjuntarlos en la siguiente iteración.