Cambios principales (funcionales)

1) Doble hash: estricto vs “tolerante a whitespace”

- **Antes:** el `read` devolvía un único SHA256 calculado sobre los **bytes originales** del archivo.
- **Ahora:** se generan y exponen dos hashes:
    - `Sha256` **(strict):** hash de bytes originales (igual que antes).
    - `Sha256NormalizedWhitespace` **(loose):** hash de un texto **normalizado**:
        - newlines a LF,
        - colapsa runs de espacios/tabs (y ahora también **NBSP U+00A0**) a un único espacio,
        - elimina whitespace al final de línea.
- **Impacto:** `apply-patch` acepta como `expectedHash` **cualquiera de los dos**. Esto reduce falsos “archivo modificado externamente” cuando Claude cambia indentación/espaciado.

Archivos relevantes:

- `FileSnapshot.cs` (nuevo campo `Sha256NormalizedWhitespace`)
- `FileGateway.cs` (cálculo y comparación de hashes)
- `Utils/WhitespaceNormalizeUtil.cs` (normalización)

---

2) Refuerzo fuerte del validador semántico del diff

Se mejoró el “semantic validation” para que sea **más robusto** y **más explicativo**:

- **Comparación de líneas más tolerante**:
    - ignora trailing whitespace,

    - colapsa runs de whitespace (tabs/espacios/NBSP) para evitar mismatch por indentación.

- **Errores más accionables**:
    - incluye número de línea,
    - muestra “Archivo vs Diff” haciendo visibles `\t` y `\r` y truncando para no explotar stdout.

- **Reubicación de hunks (estilo patch) si el `@@ -start,count` está corrido**:
    - busca una **coincidencia única** por contexto dentro de una ventana,
    - usa “fuzz” controlado (recorta hasta 2 líneas de contexto al inicio/fin),
    - si hay **múltiples matches**, falla (no elige al azar) → evita aplicar cambios en lugar equivocado.

Archivos relevantes:

- `Diff/UnifiedDiffSemanticValidator.cs` (gran refactor: búsqueda, fuzz, matching, mensajes)
- `Diff/UnifiedDiffValidator.cs` (evita división por cero si el archivo tiene 0 líneas)

---

3) Parser de unified diff más compatible

- Soporta headers tipo git donde el `,1` se omite:
    - `@@ -5 +5 @@` (interpreta count=1)
- Normaliza también `\r` suelto además de `\r\n` (CRLF) → `\n`

Archivos relevantes:

- `Diff/UnifiedDiffParser.cs`

---

4) Guardrails de Unicode: detección de U+FFFD / U+FEFF con “corte temprano”

Se incorporó una política explícita contra caracteres “problemáticos”:

- Si el **archivo de entrada** ya contiene `U+FFFD` (replacement) o `U+FEFF` (BOM/ZWNBSP), **`apply-patch` aborta temprano** con un error instructivo.
- Si el **resultado del patch** contiene esos caracteres, el applier lo rechaza con detalle de ubicaciones (línea/columna).

Archivos relevantes:

- `Utils/UnicodeIssueUtil.cs` (nuevo)
- `Diff/UnifiedDiffApplier.cs` (usa el util y mejora el error)
- `FileGateway.cs` y `Program.cs` (warning/fail early)

Cambios en CLI / DX (Developer Experience)

5) `read` ahora imprime 2 hashes + warning a stderr

- `read <path>`:
  - **stdout:** contenido del archivo + `-----HASH-----` (whitespace-normalized) + `-----HASH-STRICT-----` (bytes)
  - **stderr:** warning si detecta unicode inválido (no contamina el contenido)

6) Nuevo comando `read-range`

- `read-range <path> <startLine> <endLine>`
  - imprime rango con numeración “1-based” (`123 | ...`)
  - vuelve a imprimir hashes (loose + strict)
  - también emite warning a stderr si hay unicode inválido
    Esto ayuda mucho para **pedirle a Claude el bloque exacto** sin volcar el

archivo completo.

7) Normalización de paths al invocar el EXE desde WSL (Windows)

- Soporta:
  - `/mnt/d/Algo/...` → `D:\Algo\...`
  - `D:\mnt\d\Algo\...` → `D:\Algo\...`
- Se corrigieron escapes de `\\` que habían quedado mal en una iteración previa.

Archivo relevante:

- `Program.cs` (`NormalizePathArg`)

Limpieza / mantenimiento

- Se eliminó `FileSnapshotLoader.cs` (quedó redundant con `FileGateway.Read`).
- `NewLineUtil` mejoró detección:
  - maneja texto vacío,
  - detecta `\r` (old Mac) además de `\n`/`\r\n`.

Notas para el equipo (integración y operación)

- **Integración recomendada:** tomar como “hash principal” el que sale bajo `-----HASH-----` (whitespace-normalized) para bajar fricción con diffs. Mantener el strict como diagnóstico/auditoría.
- **Seguridad del patch:** aunque se tolera whitespace para comparación y hash, el validador ahora:
  - exige coincidencia única si reubica,

    - evita aplicar cuando el contexto es ambiguo,
    - mejora el diagnóstico cuando hay tabs/espacios o líneas “wrappeadas”.
- **Unicode inválido:** si aparece el error/warning, no es tema de “reintentar”; hay que **corregir el archivo** (quitar U+FFFD/U+FEFF) y recién ahí volver a aplicar.

Si querés, te lo formateo como `CHANGELOG.md` listo para commitear, con secciones “Breaking/Behavior change / Added / Fixed”.