

ESPECIFICACIÓN TÉCNICA OFICIAL

Winfac Office — Registro de Eventos Veri*Factu (Ledger JSONL)

ESQUEMA 1.0 (versión estable)

1. Introducción

El sistema de registro de eventos Veri*Factu de **Winfac Office** garantiza la integridad, trazabilidad e inalterabilidad de todos los eventos relacionados con la generación y gestión de registros de facturación exigidos por el **Real Decreto 1007/2023**, relativo a los Sistemas Informáticos de Facturación (SIF).

Cada evento se almacena en formato **JSON Lines (JSONL)**, un estándar ampliamente utilizado para auditoría y análisis masivo.

Cada línea representa un evento completo, independiente y encadenado criptográficamente con el anterior mediante **hashes SHA-256**.

Este documento define la **estructura del esquema 1.0**, obligatorio para todas las instalaciones de Winfac Office que operen en modo Veri*Factu.

2. Filosofía del diseño

El ledger cumple los principios esenciales del RD 1007/2023:

- **Inalterabilidad:**
Cada evento incluye un `entry_hash` que se calcula en función de su contenido y del hash del evento anterior (`prev_hash`).
- **Trazabilidad:**
Se registran todos los eventos relevantes del ciclo de vida de una factura (generación, cierre, modificación, anulación, transmisión, errores, etc.).
- **Integridad técnica:**
El fichero anual contiene una cadena hash completa. Modificar o eliminar eventos rompe la cadena.
- **Lectura estructurada:**
JSONL permite lectura secuencial y análisis eficiente con herramientas estándar.
- **Portabilidad:**
El fichero puede entregarse directamente a la AEAT sin conversión adicional.

3. Ubicación y estructura de archivos

Carpeta base de logs

<carpeta_programa>\verifactu_logs\

Estructura por anualidad

```
verifactu_logs/
├── 2025/
│   ├── verifactu_2025.ledger.jsonl
│   └── <otros_subficheros_futuros>
```

- Cada año tiene su propio ledger.
- No se mezclan anualidades en un único fichero.

Archivo principal

verifactu_YYYY.ledger.jsonl

- Contiene **un evento por línea**, en JSON válido.
- El orden es estrictamente cronológico.
- Cada evento incluye `entry_hash` y `prev_hash`.

4. Estructura de cada evento (Esquema 1.0)

Cada línea contiene un objeto JSON con esta estructura:

```
{
  "ts": "2025-11-07T18:26:48.001",
  "year": 2025,
  "user": "UsuarioWindows",
  "host": "NombreEquipo",
  "app": "Winfac",
  "app_ver": "v2025.11",
  "serie": "A",
  "num": "8",
  "accion": "MODIFICA",
  "motivo": "NUEVOS DATOS",
  "outcome": "N/A",
  "cadena_id": "A-8|2025-10-23|326.00|B25211540",
  "hash_actual": "9608AD47BA4...",
  "hash_anterior": "65D905BF1B5...",
  "csv": "true",
  "resultado_aeat": "OK",
  "http_status": 200,
  "exit_code": 0,
  "prev_hash": "E035D714DE0D...",
  "log_schema_ver": "1.0",
  "extra": {},
}
```

```
"entry_hash": "D249486EC4B9FA5F..."
}
```

5. Definición de campos

5.1 Metadatos del sistema

Campo	Tipo	Descripción
ts	string	Fecha/hora local en ISO-8601 con milisegundos.
year	integer	Año natural correspondiente al fichero.
user	string	Usuario Windows que efectuó la acción.
host	string	Nombre del equipo donde se ejecutó Winfac.
app	string	Nombre de la aplicación (“Winfac”).
app_ver	string	Versión instalada que generó el evento.

5.2 Identificación de factura

Campo	Tipo	Descripción
serie	string	Serie documental de la factura.
num	string	Número de factura.
cadena_id	string	Cadena base usada para el hash Veri*Factu (`SERIE-NUM
hash_actual	string	Hash actual de la factura según RD 1007/2023.
hash_anterior	string	Hash anterior encadenado de factura.

5.3 Evento funcional

Campo	Tipo	Descripción
accion	string	Acción realizada (CREAR, MODIFICA, CERRAR, ANULA, GENERA_XML, TRANSMITE, ERROR, etc.)
motivo	string	Mensaje explicativo generado por Winfac.
outcome	string	Resultado genérico de la acción.

5.4 Interacción con AEAT

Campo	Tipo	Descripción
csv	string/bool	Indica si se generó CSV o si hubo error al hacerlo.
resultado_aeat	string	Respuesta lógica (“OK”, “ERROR”, vacío).
http_status	int	Código HTTP de la respuesta AEAT (si aplica).
exit_code	int	Código interno de proceso/subproceso.

5.5 Encadenado criptográfico

Campo	Tipo	Descripción
prev_hash	string	entry_hash del evento anterior en el ledger.
entry_hash	string	SHA-256 calculado como: <code>SHA256(prev_hash + core_json)</code> .

5.6 Extensión del esquema

Campo	Tipo	Descripción
log_schema_ver	string	Versión del formato del ledger (actual: "1.0").
extra	objeto JSON	Campo libre para datos adicionales.

6. Encadenado criptográfico (mecanismo)

Cada evento se genera así:

1. Se lee el `entry_hash` del evento anterior (`prev_hash`).
2. Se construye el **objeto JSON sin `entry_hash`**.
3. Se calcula:

```
entry_hash = SHA256(prev_hash + core_json)
```

4. Se añade el campo `entry_hash`.
5. Se escribe la línea completa en el fichero JSONL.

Esto produce una cadena ininterrumpida de hashes.
Cualquier modificación retroactiva invalida todos los hashes siguientes.

7. Ventajas del formato

- ✓ Cumple el RD 1007/2023 (integridad y trazabilidad)
- ✓ No requiere base de datos adicional
- ✓ Exportable tal cual a la AEAT
- ✓ Fácilmente auditable
- ✓ Resistente a manipulaciones
- ✓ Portable entre equipos

- ✓ Versión explícita de esquema (mantiene compatibilidad futura)
-

8. Ejemplo real de encadenado simplificado

Evento 1

- `prev_hash = ""`
- `entry_hash = ABC123...`

Evento 2

- `prev_hash = "ABC123..."`
- `entry_hash = F67A09...`

Evento 3

- `prev_hash = "F67A09..."`
- `entry_hash = D991AA...`

Si alguien:

- borra el evento 2 → los hashes 3, 4, 5... dejan de cuadrar
 - modifica el importe de una línea → mismo resultado
-

9. Cumplimiento normativo

El diseño del ledger satisface las áreas críticas de la normativa:

- Registro de eventos del SIF
- Seguridad e integridad
- Trazabilidad
- Garantía de no alteración
- Registro cronológico
- Capacidad de entrega de ficheros

Al ser un formato estructurado y legible por máquina, está alineado con los principios recogidos en el RD 1007/2023.

10. Evolución del esquema

La versión actual es:

```
log_schema_ver = "1.0"
```

Futuras versiones podrán añadir:

- Campos específicos de transmisión AEAT
- Identificación de terminal o puesto
- Tipos de evento extendidos
- Firma digital del ledger completo
- Segmentación mensual, si se desea

Los leds antiguos se mantienen plenamente compatibles.

11. Conclusión

El formato **Winfac Office Ledger JSONL** — **Esquema 1.0** ofrece:

- robustez criptográfica
- claridad técnica
- cumplimiento con VERI*FACTU
- portabilidad
- longevidad documental
- y una base sólida para futuras mejoras

Se trata de un sistema profesional, auditable y diseñado para integrarse directamente en los procedimientos de control tributario actuales y futuros.