Flujo de Descarga y Procesamiento de Órdenes de Transporte

Descripción General

Este documento describe el flujo completo del sistema de descarga y procesamiento automático de archivos JSON desde el servidor FTP, incluyendo la validación de datos, operaciones en base de datos y gestión de archivos.

□ Esquema del Proceso

```
1□ INICIO DEL PROCESO
   ├─ Verificación de sesión activa
   ├─ Carga de configuración desde settings/{dominio}.json
   └─ Inicialización de variables y contadores

2□ CONEXIÓN A BASE DE DATOS
   ├─ Servidor: 217.154.117.83:3308
   ├─ Base de datos: newproject
   └─ Conexión PDO con preparación de consultas

3□ CONEXIÓN AL SERVIDOR FTP
   ├─ Servidor: 84.127.234.85:21
   ├─ Usuario: ftpEfeuno
   ├─ Modo PASV: Activado
   └─ Listado de archivos *.json disponibles

4□ DESCARGA DE ARCHIVOS (en lotes de 50)
   ├─ Para cada archivo en el servidor FTP:
   │   ├─ Descarga a carpeta temporal: /descargas/
   │   ├─ Validación de tamaño (mínimo 2KB)
   │   ├─ Registro en array de control
   │   ├─ Eliminación del archivo remoto (si descarga exitosa)
   │   └─ Reintento automático (hasta 4 intentos)
   └─ Generación de JSON de control de descarga

5□ PROCESAMIENTO DE ARCHIVOS JSON
   Para cada archivo descargado:

   A) VALIDACIÓN DE DNI
      ├─ Longitud mínima: 4 caracteres
      ├─ Funciones: validarIdentificador(), validarDNIOuNIF(), validarCIF()
      └─ Acción si falla: Mover a /errores_procesados/{YYYYMMDD}/

   B) VALIDACIÓN DE EMAIL
      ├─ Función: validarCorreo()
      ├─ Valor por defecto si es nulo/vacío: "sin-email@transporte.local"
      └─ Formato válido requerido

   C) VALIDACIÓN DE CÓDIGO POSTAL
      ├─ Truncamiento automático: substr(trim($CP), 0, 10)
      └─ Límite: 10 caracteres (VARCHAR(10))

   D) CREACIÓN/ACTUALIZACIÓN DE USUARIO-CONDUCTOR
      ├─ Tabla: tm_usuario
      ├─ Campos: correoUsu, senaUsu, nombreUsu, dniUsu, telefonoUsu, rolUsu (0)
      ├─ Operación: INSERT ON DUPLICATE KEY UPDATE
      └─ Token generado: 30 caracteres hexadecimales

   E) CREACIÓN/ACTUALIZACIÓN DE TRANSPORTISTA
      ├─ Tabla: transportistas-Transporte
      ├─ Campos: nombreTransportista, dniTransportista, correoTransportista, etc.
      ├─ Operación: INSERT ON DUPLICATE KEY UPDATE
      └─ Relación: codUsuarioTrabaja → id_usu del conductor
```

```
   F) CREACIÓN/ACTUALIZACIÓN DE ORDEN
      ├─ Tabla: orden-Transporte
      ├─ Campos: TTE_COD, codCliente_ordenTransporte, fechaOrdenViaje, etc.
      ├─ Operación: INSERT ON DUPLICATE KEY UPDATE
      ├─ Estados: 1=Pendiente, 2=En Progreso, 3=Completado
      └─ Token único: tokenOrden (30 caracteres)

   G) CREACIÓN/ACTUALIZACIÓN DE VIAJES
      ├─ Tabla: viaje-Transporte
      ├─ Array de ubicaciones en el JSON
      ├─ Para cada ubicación:
      │   ├─ INSERT ON DUPLICATE KEY UPDATE
      │   ├─ Campos: codOrdenViaje, codigoViaje, poblacionViaje, etc.
      │   └─ Tipo: C=Carga, D=Descarga
      └─ Contador de viajes procesados

   H) MOVIMIENTO DEL ARCHIVO PROCESADO
      ├─ Si éxito: /descargas_procesados/{YYYYMMDD}/{archivo}.json
      └─ Si error: /errores_procesados/{YYYYMMDD}/{archivo}.json

6️⃣ REGISTRO DEL PROCESO
   ├─ Generación de JSON detallado por archivo:
   │   ├─ Ruta: /descargas_procesados/control_procesados/{YYYYMMDD}/RP_{timestamp}.json
   │   ├─ Contenido: { nombreArchivo, procesado, errores[], detalles[] }
   │   └─ Timestamp: YYYYMMDD_HHMMSS
   │
   └─ Generación de JSON resumen:
      ├─ Ruta: /descargasProcesados/control_procesados/{timestamp}.json
      └─ Contenido: { archivos_procesados, errores, viajes_creados }

7️⃣ RESUMEN FINAL
   ├─ Generación de HTML con estadísticas:
   │   ├─ Total de archivos descargados
   │   ├─ Archivos procesados correctamente
   │   ├─ Archivos con errores
   │   └─ Total de viajes creados/actualizados
   │
   └─ Visualización en ventana emergente con estilos Bootstrap
```

🗂️ Estructura de Directorios

```
/view/Ordenes/
├── descargas/                          # Descarga temporal desde FTP
│   └── control_descargas/              # JSONs de control de descarga
│       └── control_descarga_{timestamp}.json
│
├── descargas_procesados/               # Archivos procesados exitosamente
│   ├── {YYYYMMDD}/                     # Carpetas por fecha
│   │   └── {archivo}.json
│   └── control_procesados/             # Logs detallados por archivo
│       └── {YYYYMMDD}/
│           └── RP_{timestamp}.json
│
├── errores_procesados/                 # Archivos con errores
│   └── {YYYYMMDD}/                     # Carpetas por fecha
│       └── {archivo}.json
│
└── descargasProcesados/
    └── control_procesados/             # Resúmenes globales
        └── {timestamp}.json
```

🗄️ Tablas de Base de Datos Afectadas

1. `tm_usuario`

Propósito: Almacenar usuarios-conductores

Campos principales:

- `id_usu` (INT, AUTO_INCREMENT, PK)
- `correoUsu` (VARCHAR, UNIQUE)
- `senaUsu` (VARCHAR, MD5 hash)
- `nombreUsu` (VARCHAR)
- `dniUsu` (VARCHAR)
- `telefonoUsu` (VARCHAR)
- `rolUsu` (INT: 0=Usuario, 1=Admin, 999=SuperAdmin)
- `tokenUsu` (VARCHAR, 30 caracteres)

Operación: `INSERT ... ON DUPLICATE KEY UPDATE correoUsu`

---

2. `transportistas-Transporte`

Propósito: Almacenar empresas transportistas

Campos principales:

- `id_Transportista` (INT, AUTO_INCREMENT, PK)
- `nombreTransportista` (VARCHAR)
- `dniTransportista` (VARCHAR, UNIQUE)
- `correoTransportista` (VARCHAR)
- `telefonoTransportista` (VARCHAR)
- `cpTransportista` (VARCHAR(10))
- `codUsuarioTrabaja` (INT, FK → tm_usuario.id_usu)

Operación: `INSERT ... ON DUPLICATE KEY UPDATE dniTransportista`

---

3. `orden-Transporte`

Propósito: Almacenar órdenes de transporte

Campos principales:

- `id_ordenTransporte` (INT, AUTO_INCREMENT, PK)
- `TTE_COD` (VARCHAR(10), código único de orden)
- `codCliente_ordenTransporte` (VARCHAR)
- `codConductor_ordenTransporte` (INT, FK → tm_usuario.id_usu)
- `fechaOrdenViaje` (DATE)
- `ordenRecogida` (VARCHAR)
- `tipoCarga_ordenTransporte` (VARCHAR)
- `estadoOrdenTransporte` (INT: 1=Pendiente, 2=En Progreso, 3=Completado)
- `tokenOrden` (VARCHAR, 30 caracteres)
- `nombreTransportista_ordenTransporte` (VARCHAR)

Operación: `INSERT ... ON DUPLICATE KEY UPDATE TTE_COD`

---

4. `viaje-Transporte`

Propósito: Almacenar ubicaciones de carga/descarga por orden

Campos principales:

- `id_viajeTransporte` (INT, AUTO_INCREMENT, PK)
- `codOrdenViaje` (INT, FK → orden-Transporte.id_ordenTransporte)
- `codigoViaje` (VARCHAR, identificador único)
- `poblacionViaje` (VARCHAR)
- `cpViaje` (VARCHAR(10))
- `direccionViaje` (VARCHAR)
- `clienteViaje` (VARCHAR)
- `tipoViaje` (CHAR(1): 'C'=Carga, 'D'=Descarga)

Operación: `INSERT ... ON DUPLICATE KEY UPDATE codigoViaje`

📄 Estructura de los JSONs de Control

JSON de Control de Descarga

Ubicación: `/descargas/control_descargas/control_descarga_{timestamp}.json`

```
{
  "fecha_proceso": "2025-01-23 14:30:45",
  "timestamp": "20250123_143045",
  "archivos": [
    {
      "nombre": "orden_12345.json",
      "descargado": true,
      "fecha_hora_descarga": "2025-01-23 14:30:47",
      "tamano": "5.2 KB"
    },
    {
      "nombre": "orden_12346.json",
      "descargado": false,
      "fecha_hora_descarga": "2025-01-23 14:30:48",
      "error": "Archivo corrupto"
    }
  ]
}
```

JSON de Control de Procesamiento (Resumen)

Ubicación: `/descargasProcesados/control_procesados/{timestamp}.json`

```
{
  "fecha_proceso": "2025-01-23 14:30:45",
  "timestamp": "20250123_143045",
  "archivos_procesados": 48,
  "errores": 2,
  "viajes_creados": 156
}
```

JSON de Registro Detallado por Archivo

Ubicación: `/descargas_procesados/control_procesados/{YYYYMMDD}/RP_{timestamp}.json`

```
{
  "nombreArchivo": "orden_12345.json",
  "procesado": true,
  "errores": [],
  "detalles": [
    "Usuario creado/actualizado: conductor@example.com",
    "Transportista creado/actualizado: DNI 12345678X",
    "Orden creada/actualizada: TTE_COD 34051401",
    "Viajes procesados: 3"
  ]
}
```

O en caso de error:

```
{
  "nombreArchivo": "orden_12346.json",
  "procesado": false,
  "errores": [
    "DNI inválido: longitud menor a 4 caracteres",
    "Email con formato incorrecto"
  ],
  "detalles": []
}
```

Reglas de Validación

DNI / NIF / CIF

- **Longitud mínima:** 4 caracteres
- **Validaciones:** `validarDNIOuNIF()`, `validarCIF()`, `validarIdentificador()`
- **Acción si falla:** Archivo movido a `/errores_procesados/`

Email

- **Función:** `validarCorreo()`
- **Valor por defecto:** `sin-email@transporte.local`
- **Formato:** Validación con `filter_var($email, FILTER_VALIDATE_EMAIL)`

Código Postal (CP)

- **Límite:** 10 caracteres máximo
- **Operación:** `substr(trim($CP), 0, 10)`
- **Tipo de campo:** `VARCHAR(10)` en todas las tablas

TTE_COD (Código de Orden)

- **Tipo:** `VARCHAR(10)`
- **Transformación:** Función `transformarNumero()`
  - Si termina en "00": eliminar últimos 2 dígitos
  - Ejemplo: `34051400` → `340514`
  - Si no: añadir "/" antes de últimos 2 dígitos
  - Ejemplo: `34051401` → `340514/01`

Flujo de Movimiento de Archivos

Configuración de Conexiones

Base de Datos

Archivo: `config/settings/{dominio}.json`

```
{
  "database": {
    "host": "217.154.117.83",
    "port": "3308",
    "dbname": "newproject",
    "username": "...",
    "password": "..."
  }
}
```

Servidor FTP

Archivo: `config/settings/{dominio}.json`

```
{
  "ftp": {
    "host": "84.127.234.85",
    "port": 21,
    "username": "ftpEfeuno",
    "password": "fTp2o24efeUn0",
    "passive_mode": true
  }
}
```

Visualización de Logs

Pantalla: control_descargas.php

Ubicación: `/view/Logs/control_descargas.php`

Funcionalidad:

- Lista JSONs de control de descarga
- DataTable con 6 columnas horizontales:
    1. Fecha
    2. Hora
    3. Timestamp
    4. Archivos FTP
    5. Procesados OK
    6. Con Errores

Origen de datos: `/descargas/control_descargas/*.json`

Pantalla: control_procesados.php

Ubicación: `/view/Logs/control_procesados.php`

Funcionalidad:

- Selección en cascada: Carpeta → Archivo
- DataTable con 4 columnas:
    1. Nombre Archivo
    2. Procesado (badge Sí/No)
    3. Errores (lista HTML)
    4. Detalles (lista HTML)

Origen de datos: `/descargas_procesados/control_procesados/{carpeta}/*.json`

Funciones Clave

`validarCorreo($correo)`

- Valida formato de email con `filter_var()`
- Retorna email válido o `sin-email@transporte.local`

`validarIdentificador($identificador)`

- Valida DNI, NIE, NIF o CIF
- Llama a `validarDNIOuNIF()` o `validarCIF()`
- Retorna `true` / `false`

`generarToken($longitud = 30)`

- Genera token hexadecimal aleatorio
- Usado para `tokenUsu` y `tokenOrden`

`transformarNumero($numero)`

- Formatea código de orden (TTE_COD)
- Elimina "00" finales o añade "/"

Ejecución del Proceso

URL: `http://192.168.31.19/leader/logistica/view/Ordenes/subirOrdenes.php`

Botón: "Descargar Órdenes del Servidor FTP"

Trigger: `onclick="descargarFicheros()"`

Función JavaScript:

```
function descargarFicheros() {
    window.open(
        '../Ordenes/descargarficheros.php',
        '_blank',
        'width=800,height=600,scrollbars=yes'
    );
}
```

Salida: Ventana emergente con log HTML en tiempo real

Notas Importantes

1. **Procesamiento por lotes:** 50 archivos por iteración para optimizar memoria
2. **Reintentos FTP:** Hasta 4 intentos por archivo
3. **Validación de tamaño:** Mínimo 2KB para considerar archivo válido
4. **Modo PASV:** Obligatorio para conexión FTP
5. **Eliminación remota:** Los archivos se eliminan del FTP tras descarga exitosa
6. **Auditoría completa:** Cada operación genera registro en JSON
7. **Seguridad:** Sesión verificada antes de cada operación
8. **Codificación de contraseñas:** MD5 para `senaUsu` en `tm_usuario`
9. **Tokens únicos:** 30 caracteres hexadecimales para usuarios y órdenes
10. **Carpetas por fecha:** `YYYYMMDD` para organización temporal

Mantenimiento

Limpieza de archivos procesados

Los archivos en `/descargas_procesados/` y `/errores_procesados/` se organizan por fecha. Se recomienda implementar un script de limpieza periódica para archivos antiguos (ejemplo: > 90 días).

Monitorización de errores

Revisar regularmente la carpeta `/errores_procesados/{fecha}/` para identificar patrones de fallos recurrentes.

Backup de control JSONs

Los archivos en `/control_descargas/` y `/control_procesados/` sirven como auditoría. Considerar backup periódico antes de eliminación.

Documento generado: 23/11/2025
Sistema: Leader Transport - Gestión Logística
Versión PHP: 8.5.0
Base de datos: MySQL (newproject)