# UTF-8 en redes de clúster

Guía general y adaptación específica a DXSpider

## 1. Objetivo

Este documento define una política general de uso de UTF-8 para cualquier tipo de nodo de clúster (DXSpider u otros), y describe:

- Por qué UTF-8 debe considerarse el formato base de interoperabilidad.
- Cómo verificar la configuración real de un sistema.
- Cómo corregir configuraciones erróneas en sistemas habituales.
- Qué cambios son específicos de DXSpider.
- Cómo aplicar esta política en contenedores Docker.

El enfoque es operativo y reproducible, no teórico.

## 2. Contexto general: nodos de clúster heterogéneos

En cualquier red de clúster distribuida (radio, mensajería, sensores, telemetría, etc.) coexisten:

- Sistemas modernos con UTF-8 nativo
- Sistemas legacy configurados en ISO-8859-1 / CP1252
- Software antiguo que asume ASCII
- Software moderno que asume Unicode

Los datos se intercambian como bytes, no como “texto Unicode”.

Sin una política explícita de codificación:

- aparecen caracteres corruptos (mojibake)
- los errores se propagan
- no se puede identificar el nodo origen del problema

## 3. Por qué UTF-8 es el formato base recomendado

### 3.1 UTF-8 es el estándar universal actual

UTF-8 es el formato por defecto en:

- Linux moderno (Debian, Ubuntu, Fedora, etc.)
- Contenedores Docker
- Lenguajes actuales (Perl, Python, Go, Rust, Java)
- Terminales y herramientas de análisis

### 3.2 Compatibilidad con ASCII

Todo ASCII válido es UTF-8 válido.

Esto garantiza que:

- nodos que solo envían ASCII siguen funcionando
- UTF-8 no rompe compatibilidad hacia atrás

### 3.3 Limitaciones de las codificaciones legacy

ISO-8859-1 y CP1252 no son autodescriptivas, reutilizan bytes con significados distintos y no permiten validación fiable. Resultado: los errores no se pueden detectar ni aislar.

## 4. Principios operativos generales (válidos para cualquier nodo)

### Principio 1 — Entrada tolerante

El nodo acepta cualquier byte entrante. No aborta ni descarta datos por codificación.

### Principio 2 — Validación explícita

Los datos se validan como UTF-8. Lo que no es UTF-8 se marca como tal.

### Principio 3 — Normalización para visualización

Es lícito convertir legacy → UTF-8 solo después de registrar el problema.

### Principio 4 — Salida estricta

El nodo emite siempre UTF-8. No se degrada la salida a legacy.

## 5. Procedimiento operativo: comprobar el estado actual

### 5.1 Comprobación del locale activo

Ejecutar en el nodo:

```
locale
locale charmap
echo "LANG=$LANG"
echo "LC_ALL=$LC_ALL"
echo "LC_CTYPE=$LC_CTYPE"
```

Estado correcto:

- locale charmap → UTF-8
- LANG y/o LC_ALL → C.UTF-8 o xx_XX.UTF-8

### 5.2 Prueba funcional rápida

```
printf 'UTF8 test: ñ Ç á é í ó ú €\n'
```

Debe verse correctamente.

## 6. Corrección de configuración por distribución

### 6.1 Debian / Ubuntu / Raspbian

Ver locales disponibles:

```
locale -a
```

Activar UTF-8 (si no existe):

```
sudo dpkg-reconfigure locales
```

Seleccionar al menos:

- C.UTF-8
- o es_ES.UTF-8, en_US.UTF-8, etc.

Fijar configuración recomendada:

```
sudo update-locale LANG=C.UTF-8 LC_ALL=C.UTF-8 LC_CTYPE=C.UTF-8
```

Reiniciar sesión o servicio.

### 6.2 Fedora / RHEL / Rocky / Alma

Fedora usa UTF-8 por defecto. Verificar:

```
localectl status
```

Forzar:

```
sudo localectl set-locale LANG=C.UTF-8
```

## 7. Adaptación específica para nodos DXSpider

### 7.1 Consideraciones DXSpider

DXSpider está escrito en Perl, recibe texto como bytes de red y procesa logs y mensajes históricos. Por tanto, la entrada debe tratarse como raw y la salida debe ser UTF-8.

### 7.2 Entorno de ejecución DXSpider

Aplicar al usuario del servicio (sysop u otro):

```
export LANG=C.UTF-8
export LC_ALL=C.UTF-8
export LC_CTYPE=C.UTF-8
```

Preferible en script de arranque o servicio systemd.

### 7.3 Scripts Perl auxiliares

Buenas prácticas:

```
use strict;
use warnings;
use utf8;  # para literales del código

binmode(STDIN,  ':raw');
binmode(STDOUT, ':encoding(UTF-8)');
binmode(STDERR, ':encoding(UTF-8)');
```

## 8. Contenedores Docker

### 8.1 Principio

El contenedor no tiene terminal: solo emite bytes.

Si el contenedor emite UTF-8 y el host está en UTF-8, la visualización es correcta.

### 8.2 Recomendación mínima (Dockerfile)

Añadir siempre:

```
ENV LANG=C.UTF-8
ENV LC_ALL=C.UTF-8
ENV LC_CTYPE=C.UTF-8
```

### 8.3 Nota sobre Alpine

Alpine no genera locales regionales. C.UTF-8 es la opción correcta y suficiente.

## 9. Política de red recomendada (resumen)

Fase / Política:

- Recepción: Tolerante
- Validación: UTF-8 estricta
- Normalización: Permitida
- Emisión: UTF-8 obligatoria

## 10. Herramienta opcional: script de normalización (HOST)

Para facilitar la normalización de nodos que no estén correctamente en UTF-8, se pone a disposición un script opcional para hosts Linux (no contenedores) que:

- Verifica la codificación efectiva con `locale charmap`.
- Si no es UTF-8, intenta habilitar el locale `*.UTF-8` equivalente al idioma configurado.
- Asegura que `C.UTF-8` esté disponible (en Debian puede aparecer como `C.utf8`).

- Mantiene el idioma existente siempre que sea posible; su objetivo es garantizar UTF-8, no cambiar el idioma.

### 10.1 Ubicación

https://github.com/EA3CV/dxspider_info/blob/main/ensure_utf8_locale.pl

### 10.2 Descarga

Ejemplo con curl:

curl -fsSL -o ensure_utf8_locale.pl \

https://raw.githubusercontent.com/EA3CV/dxspider_info/main/ensure_utf8_locale.pl

Ejemplo con wget:

wget -O ensure_utf8_locale.pl \

https://raw.githubusercontent.com/EA3CV/dxspider_info/main/ensure_utf8_locale.pl

### 10.3 Permisos de ejecución

chmod +x ensure_utf8_locale.pl

### 10.4 Cómo ejecutarlo y con qué usuario

Verificación (no cambia nada): se puede ejecutar como usuario normal.

./ensure_utf8_locale.pl --check

Aplicar cambios (si fuese necesario): debe ejecutarse como root (o con sudo).

sudo ./ensure_utf8_locale.pl --apply

Notas operativas:
- Si el sistema ya está correctamente en UTF-8, el script termina informando del estado (sin cambios).
- Si `C.UTF-8` no aparece en `locale -a`, en Debian puede mostrarse como `C.utf8`.
- Este script está pensado para hosts y no para contenedores.

## 11. Conclusión

El uso sistemático de UTF-8 no rompe compatibilidad, permite diagnóstico fiable, evita propagación de errores y prepara la red para el futuro.

UTF-8 debe considerarse el formato interno y de salida de cualquier nodo moderno, con tolerancia controlada en la entrada para mantener interoperabilidad con nodos legacy.

DXSpider encaja perfectamente en este modelo cuando se configura correctamente.

Kin EA3CV

20251227