# Cartelia — Despliegue de máquina virtual base (Debian 13 / KVM-QEMU)

**Documentación técnica de proceso** Versión del documento: 1.0 · Producto: Cartelia 1.0 Fecha: 09/07/2026 · Responsable: Iván Matarrubias

## Índice



## 1. Introducción

**Cartelia** es el producto de cartelería digital (digital signage) de la compañía.

El presente documento describe el procedimiento de despliegue de una nueva instancia de Cartelia partiendo de una **máquina virtual base** con las siguientes características:

- Sistema operativo: **Debian 13 (trixie)**.
- Hipervisor: **KVM/QEMU** (gestionado mediante `libvirt` / `virsh` o `virt-manager`).
- Software preinstalado: **Cartelia 1.0**, instalado en `/var/www/html/cartelia`.

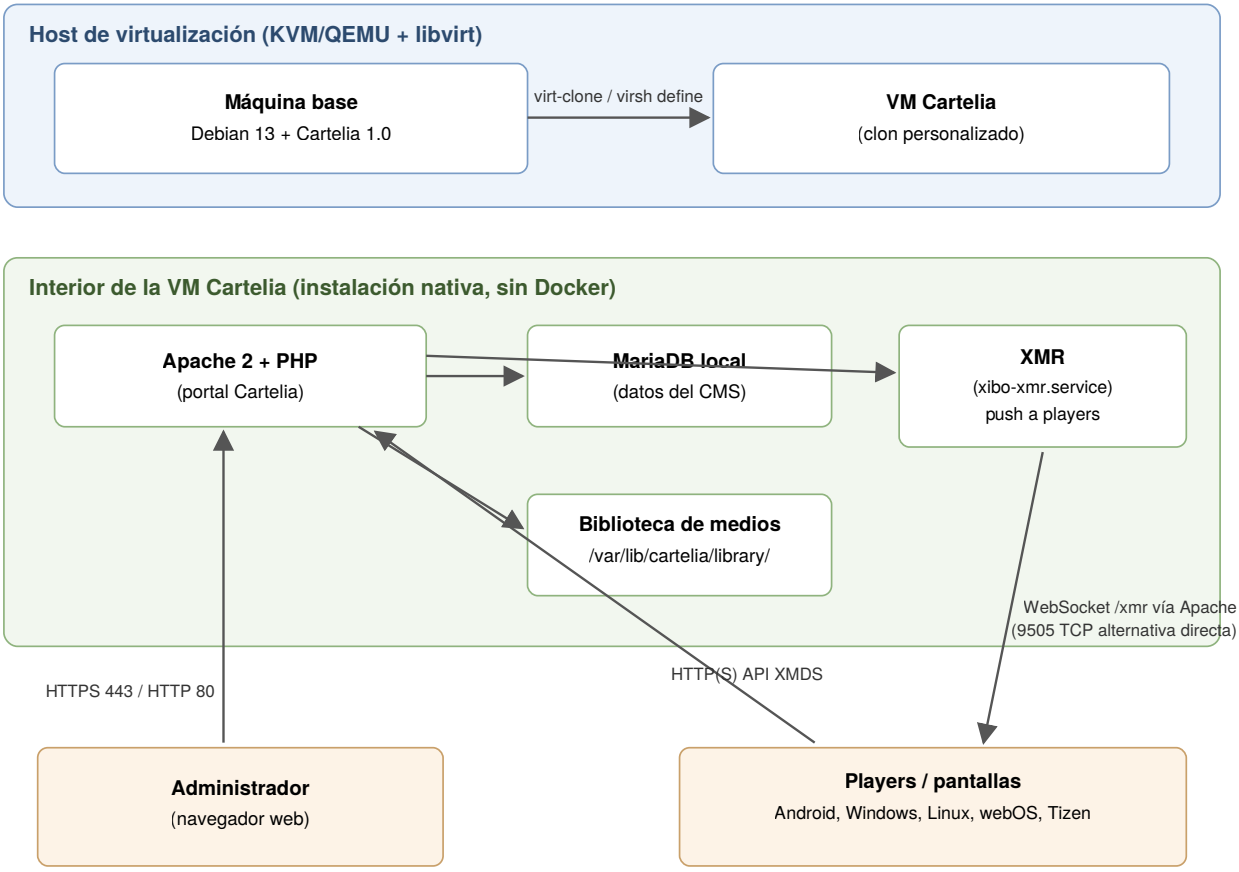
**Punto de partida**: la máquina base se entrega totalmente operativa, con Cartelia instalado de forma nativa sobre Apache + PHP y la base de datos MariaDB local ya creada e inicializada.

**Restricciones conocidas**:

- La plantilla base **no debe arrancarse nunca en producción sin personalizar**: contiene identidad de máquina, claves SSH y secretos del CMS compartidos con cualquier otra copia de la plantilla.
- Los players (pantallas) se autentican contra el CMS mediante la clave del servidor (*CMS Key*) y la URL del CMS; cualquier cambio posterior de IP/FQDN o de claves obliga a volver a autorizar los displays.

# 2. Arquitectura

## 2.1 Esquema general del despliegue




El CMS está instalado de forma nativa sobre el sistema operativo: Apache 2 con PHP como servidor web del portal, MariaDB local como base de datos y el servicio XMR ejecutándose como unidad de systemd.

## 2.2 Flujo del proceso de despliegue

1. El técnico de sistemas clona la plantilla base en el host KVM/QEMU (disco `qcow2` + definición de dominio libvirt).
2. En el primer arranque del clon, se personaliza la identidad del sistema operativo: `machine-id`, hostname, claves SSH de host, red, zona horaria y credenciales locales.
3. Se personaliza el producto Cartelia: URL pública del CMS, secretos y contraseñas de base de datos, clave del servidor (*CMS Key*), dirección pública de XMR y personalización de marca.
4. Se verifican los servicios, la conectividad de los players y el acceso al portal de administración.
5. La máquina queda registrada en inventario/monitorización y se entrega.

## 2.3 Componentes principales

| Componente | Función |
|---|---|
| **Portal Cartelia** | Interfaz de administración de contenidos, layouts, programaciones y pantallas. Servida por Apache 2 + PHP. |

| Componente | Función |
|---|---|
| **MariaDB (local)** | Almacena la configuración, usuarios, programaciones y metadatos de medios del CMS. Servicio nativo del sistema. |
| **XMR (Cartelia Message Relay)** | Canal push que permite al CMS enviar órdenes inmediatas a los players (recarga de contenido, capturas, reinicio). Servicio `xibo-xmr.service`. Los players se conectan por WebSocket a la ruta `/xmr` del vhost de Apache, que reenvía a `ws://127.0.0.1:8080`; el puerto 9505/TCP queda además permitido en el firewall para el modo de publicación directa. |
| **Biblioteca de medios** | `/var/lib/cartelia/library/` : directorio donde el CMS guarda los ficheros multimedia subidos (configurado en `Administración > Configuración > Ubicación de la biblioteca` y alineado con la directiva `XSendFilePath` de los vhosts). |
| **libvirt / KVM-QEMU** | Capa de virtualización que aloja la VM. |

### 2.4 Dependencias e integraciones externas

- **DNS**: registro A/AAAA del FQDN del CMS (imprescindible si los players usan nombre en lugar de IP).
- **NTP**: sincronización horaria; las programaciones de contenido dependen de la hora del sistema.
- **SMTP** (opcional): relay de correo para notificaciones del CMS, si el cliente lo requiere. Se configura en el portal (`Administración > Configuración`, pestaña **Mantenimiento**).
- **Certificados TLS**: siempre certificados válidos, nunca autofirmados. Si el cliente dispone de certificado propio para el FQDN, lo aporta él; en caso contrario se emite con Let's Encrypt.

---

## 3. Personalización del sistema operativo (dentro de la VM)

### 3.1 Personalización de identidad y red

Todos los comandos se ejecutan como `root` desde la consola de `virsh` en el primer arranque.

```
# Paso 1: regenerar las claves SSH de host (las de la plantilla son comunes a todos los
clones)
rm -f /etc/ssh/ssh_host_*
dpkg-reconfigure openssh-server
systemctl restart ssh

# Paso 2: hostname y resolución local
hostnamectl set-hostname <nombre-del-entorno>
sed -i "s/127.0.1.1.*/127.0.1.1\t<nombre-del-entorno> <FQDN-del-CMS>/" /etc/hosts

# Paso 3: configuración de red (ifupdown, /etc/network/interfaces)

# Paso 4: zona horaria y hora (NTP)
vi /etc/systemd/timesyncd.conf.d/local.conf


# Paso 5: actualización de paquetes del SO
apt update && apt upgrade -y
```

### 3.2 Arranque del producto Cartelia

El CMS ya viene instalado y operativo en la máquina base; en esta fase únicamente debe verificarse el arranque de los servicios nativos tras la personalización de red:

```
systemctl status apache2 mariadb
systemctl status xibo-xmr.service

# Si algún servicio no arranca tras el cambio de red/hostname:
systemctl restart apache2
journalctl -u apache2 -n 50 --no-pager
```

### 3.3 Verificación de la instalación

```
# El portal responde en local (la máquina base publica en HTTP; ver 4.3 para HTTPS)
curl -Is http://localhost/ | head -1          # esperado: HTTP/1.1 200 OK (o 302 a /
login)

# Puertos de servicio a la escucha
ss -tlnp | grep -E ':(80|9505)\b'

# Resolución del FQDN desde un puesto cliente
getent hosts <FQDN-del-CMS>
```

Acceso final: abrir `http://<FQDN-del-CMS>/` en el navegador y validar la pantalla de login de Cartelia (o `https://` si se ha habilitado el vhost TLS, ver 4.3).

---

## 4. Configuración

### 4.1 Ficheros de configuración del producto

La instalación es nativa (Apache + PHP + MariaDB local); estos son los ficheros que el SAT debe revisar/personalizar en cada clon:

**`settings.php` del CMS** (`/var/www/html/cartelia/web/settings.php`):

| Parámetro | Descripción | Acción en el despliegue |
|---|---|---|
| `$dbhost` | Host de MariaDB (normalmente `localhost`) | No cambiar |
| `$dbname`, `$dbuser` | Base de datos y usuario del CMS | No cambiar salvo política del cliente |
| `$dbpass` | Contraseña del usuario del CMS en MariaDB | No cambiar |
| `SECRET_KEY` | Clave secreta de la instalación (cifrado interno) | Actualmente no se regenera: se mantiene el valor de la máquina base en todos los despliegues |

**XMR** (`xibo-xmr.service`, unidad en `/etc/systemd/system/xibo-xmr.service`): ejecuta `/var/www/html/cartelia/vendor/bin/xmr.phar` con PHP como usuario `www-data`, con reinicio automático

(`Restart=always`). Sockets por defecto: WebSocket para players en `127.0.0.1:8080`, expuesto a través del proxy `/xmr` del vhost de Apache, y publicación interna del CMS en `tcp://127.0.0.1:50001`.

Tras cualquier cambio: `systemctl restart xibo-xmr.service`.

**Apache**: la máquina base incluye dos vhosts en `/etc/apache2/sites-available/`:

- `cartelia_http.conf` (habilitado por defecto): sirve el portal en el puerto 80. En la máquina base el `ServerName` es la IP de la propia máquina; en cada cliente debe actualizarse al FQDN definitivo. Incluye el proxy WebSocket de XMR (`ProxyPass /xmr ws://127.0.0.1:8080`), `XSendFile` para la entrega de medios y una exclusión de mod_security (`SecRuleRemoveById 920350`).
- `cartelia_https.conf` (deshabilitado por defecto): vhost TLS preparado para clientes con FQDN y certificado. Su activación se describe en la sección 4.3.

Validar siempre con `apache2ctl configtest` antes de `systemctl reload apache2`.

## 4.2 Configuración desde el portal de administración de Cartelia

Acceder con el usuario **`admin`** y la contraseña estándar de la máquina base (conocida por el SAT; no se documenta aquí) y cambiar dicha contraseña de inmediato, ya que es común a todos los clones. A continuación:

- `Administración > Configuración > Red`: establecer la dirección del CMS (`CMS Address`) con el FQDN definitivo. Los players usan este valor para conectarse.
- `Administración > Configuración`, pestaña **Configuración**, campo **"Clave secreta de CMS"**: es la clave que debe introducirse en cada player para autenticarlo contra el CMS. En un clon es idéntica a la de la máquina base (y, por tanto, conocida por cualquier otro despliegue), así que debe sustituirse por una clave nueva y única en cada cliente.
- `Administración > Configuración`, pestaña **Configuración**, campo **"Ubicación de la biblioteca"**: verificar que mantiene el valor `/var/lib/cartelia/library/`; no cambiarlo, ya que debe coincidir con la directiva `XSendFilePath` de los vhosts de Apache.
- `Administración > Configuración`, pestaña **Red** (sección XMR): establecer la dirección pública de XMR como `ws://<FQDN-del-CMS>/xmr` (o `wss://<FQDN-del-CMS>/xmr` si está activo el vhost HTTPS). Los players se conectan por WebSocket a través del proxy de Apache; el puerto 9505 permanece abierto en el firewall únicamente como vía alternativa de publicación directa.

Los cambios de claves y direcciones obligan a volver a autorizar los displays que estuvieran dados de alta en la máquina base (en un despliegue nuevo no debería haber ninguno; si los hubiera, eliminarlos).

## 4.3 Configuración a nivel de sistema operativo

| Elemento | Fichero / mecanismo | Acción |
|---|---|---|
| Firewall | `firewall.service` → ejecuta `/etc/firewall/firewall.sh`, parametrizado por `/etc/firewall/vars.sh` | Editar `vars.sh` en cada despliegue (ver 4.4): la IP local está fijada a la de la máquina base y debe cambiarse junto con la red del cliente |
| fail2ban | `/etc/fail2ban/jail.d/` | Verificar jail de `sshd` activo tras el cambio de IP |

| Elemento | Fichero / mecanismo | Acción |
|---|---|---|
| Certificado TLS / HTTPS | Vhost `cartelia_https.conf` incluido de serie (deshabilitado) | Certificado siempre válido, nunca autofirmado: lo aporta el cliente si dispone de él o se emite con Let's Encrypt ( `certbot certonly` para el FQDN; requiere el puerto 80 alcanzable desde internet y renovación automática vía timer de certbot). Instalar certificado y clave en `/etc/ssl/cartelia/` (o apuntar las directivas `SSL*` a `/etc/letsencrypt/live/<FQDN>/` ), ajustar `ServerName` , `a2enmod ssl proxy_wstunnel headers` , `a2ensite cartelia_https` , `apache2ctl configtest` y `systemctl reload apache2` . Activar después la redirección HTTP→HTTPS en `cartelia_http.conf` y actualizar en el portal la CMS Address ( `https://<FQDN>` ) y la dirección de XMR ( `wss://<FQDN>/xmr` ) |
| Rotación de logs | `logrotate` | Verificar |

### 4.4 Firewall ( `/etc/firewall/` )

El firewall se implementa con nftables mediante dos ficheros, ejecutados en el arranque por `firewall.service` :

| Fichero | Función |
|---|---|
| `/etc/firewall/firewall.sh` | Reglas nftables: acepta 80, 443 y 9505 (XMR) hacia la IP local/flotante, 22 solo desde `RED_GESTION` , checks pasivos de Zabbix (10050) desde el servidor Zabbix, ICMP y tráfico establecido; el resto de entrada se descarta ( `drop` ). |
| `/etc/firewall/vars.sh` | Variables de entorno que parametrizan las reglas. |

Variables de `vars.sh` a ajustar en cada despliegue:

| Variable | Valor en la máquina base | Acción en el despliegue |
|---|---|---|
| `ETH` | `ens3` | Cambiar solo si la interfaz del clon tiene otro nombre ( `ip a` ) |
| `IP_LOC` | IP de la máquina base | Cambiar siempre a la IP del cliente (debe coincidir con la configurada en la sección 3.1, paso 3) |
| `IP_FLO` | Igual a `IP_LOC` (sin IP flotante) | Mantener igual a `IP_LOC` salvo despliegue con alta disponibilidad |
| `RED_DATOS` | Red corporativa de la máquina base | Ajustar al direccionamiento del cliente |
| `RED_GESTION` | Red corporativa de la máquina base | Ajustar a la red desde la que se accede por SSH; es la única red con acceso al puerto 22 |

Aplicación y verificación de cambios:

```
# Aplicar (corta y recrea todas las reglas; las conexiones establecidas se mantienen)
/etc/init.d/firewall start

# Verificar reglas cargadas y contadores
/etc/init.d/firewall status

# Verificación remota mínima tras el cambio (desde RED_GESTION)
ssh <usuario>@<IP-del-servidor>          # el 22 debe seguir accesible
curl -Is http://<IP-del-servidor>/ | head -1
```

> **Riesgo operativo**: editar `IP_LOC` o `RED_GESTION` con valores erróneos y ejecutar el script **puede dejar la máquina inaccesible por SSH** (la política final es `drop`). Realizar siempre el cambio con una consola de `virsh console` abierta como vía de recuperación.

---

## 5. Diagnósticos

### 5.1 Comandos de diagnóstico

```
# Estado general de la VM desde el host de virtualización
virsh dominfo cartelia-<nombre-del-entorno>
virsh domifaddr cartelia-<nombre-del-entorno>

# Dentro de la VM — estado de servicios
systemctl status apache2 mariadb
systemctl status xibo-xmr.service
journalctl -u apache2 -n 100 --no-pager
journalctl -u xibo-xmr.service -n 100 --no-pager

# Conexión a la base de datos del CMS
mysql -u <usuario-cms> -p -e "SELECT COUNT(*) FROM display;" <bbdd-cms>

# Firewall: reglas activas y contadores por regla
/etc/init.d/firewall status

# Conectividad de los puertos de producto
ss -tlnp | grep -E ':(80|443|9505)\b'

# Prueba del API de players (XMDS): debe responder el WSDL
curl -s "http://localhost/xmds.php?wsdl" | head -5

# Comprobar identidad correcta del clon (no debe coincidir con la plantilla)
cat /etc/machine-id
ssh-keygen -lf /etc/ssh/ssh_host_ed25519_key.pub
```

### 5.2 Ficheros de log

| Fichero / origen | Contenido | Rotación |
|---|---|---|
| Portal `Administración > Avanzado > Log` | Log de aplicación del CMS (registro de players, errores XMDS) | Gestionada por la tarea de mantenimiento del CMS |

| Fichero / origen | Contenido | Rotación |
|---|---|---|
| `/var/log/apache2/cartelia_access.log`, `/var/log/apache2/cartelia_error.log` (y `cartelia_ssl_*.log` con el vhost HTTPS activo) | Accesos y errores del servidor web (portal y API XMDS de players) | logrotate estándar de apache2 |
| `journalctl -u xibo-xmr.service` | Log del servicio XMR | journald |
| `journalctl -u ssh`, `/var/log/fail2ban.log` | Seguridad de acceso al SO | Estándar journald / logrotate |

## 5.3 Errores comunes y soluciones

| Síntoma / mensaje | Causa probable | Solución |
|---|---|---|
| El portal no carga tras el clonado | La red de la VM no se personalizó o la interfaz cambió de nombre | Revisar `ip a`, `/etc/network/interfaces.d/`, y el bridge asignado en libvirt |
| Los players no consiguen registrarse | `CMS Address` o `CMS Secret Key` incorrectos, o 80/443 filtrados | Verificar sección 4.2 y `vars.sh` del firewall (4.4); probar `curl` al XMDS desde la red de players |
| Los players se registran pero no reciben órdenes push | Dirección pública de XMR errónea, proxy `/xmr` no operativo o `xibo-xmr` caído | Verificar `ws(s)://<FQDN>/xmr` en el portal, `systemctl status xibo-xmr.service` y el módulo `proxy_wstunnel` de Apache; reiniciar XMR |
| Aviso de certificado en el navegador | Certificado de la plantilla aún instalado | Sustituir por el certificado del FQDN del entorno |
| Dos clones aparecen como la misma máquina en DHCP/monitorización | `machine-id` no regenerado | Ejecutar `rm -f /etc/machine-id && systemd-machine-id-setup` y reiniciar |
| El CMS no conecta con la base de datos tras cambiar contraseñas | Contraseña cambiada en `settings.php` pero no en MariaDB (o viceversa) | Sincronizar ambas (ver sección 4.1) |
| Sin acceso SSH tras personalizar el firewall | `RED_GESTION` o `IP_LOC` erróneas en `vars.sh` con política final `drop` | Entrar por `virsh console`, corregir `vars.sh` y re-ejecutar `/etc/firewall/firewall.sh` |
| La hora de las programaciones no coincide | Zona horaria o NTP sin configurar | `timedatectl set-timezone Europe/Madrid` y verificar `timedatectl status` |