Tu Mentor

Manual de administración · Operación de la plataforma

Este manual es para quien administra Tu Mentor a nivel servidor: instalación, configuración, gestión de usuarios y roles, biblioteca de contenidos, troubleshooting. Asume conocimientos básicos de Linux, nginx, PHP y PostgreSQL.

Audiencia: Simón Berton (dev) y eventualmente cualquier persona técnica que asuma la operación de la plataforma. Si sos un padre o maestra, el manual que querés leer es el de **usuario**, no este.

1. Visión general

Tu Mentor es una aplicación web hecha en Symfony 8 (PHP 8.4), con Postgres 16 como base, Redis para cache, Mercure para mensajería en tiempo real (SSE), y Claude (Anthropic) como motor de IA conversacional. El frontend es vanilla JS sin build step, servido por nginx.

Stack desplegado

- **Web server**: nginx 1.24 (Ubuntu 24.04). Misma instalación que sirve los WordPress.
- **PHP**: 8.4-fpm (Ondrej Sury PPA). Coexiste con 8.3 (que usan los WP).
- **Postgres**: 16 nativo de Ubuntu.
- **Redis**: 7 nativo.
- **Mercure**: binario standalone basado en Caddy, en 127.0.0.1:3000.
- **Worker**: systemd unit corriendo bin/console messenger:consume async failed.
- **HTTPS**: Let's Encrypt vía certbot (renovación automática).

Server actual

- DigitalOcean Droplet: 1 vCPU AMD, 2 GB RAM, 50 GB NVMe, $14/mes.
- IP pública: `24.199.96.252`.
- Dominio: `tumentor.com.uy` (DNS en DO, A record + CNAME www).
- + Swap 2 GB configurado en `/swapfile`.
- Coexiste con 3 WordPress (agritech.uy, aiu.com.uy, codicsi.com, mundialyerbamateuruguay.com).

2. Estructura de archivos en el server

```
/mnt/www/guia-parental/ ← código (clone de git) ■■■ public/ ← document root para nginx ■ ■■■ app.html ← PWA single-page ■ ■■■ index.php ← front controller Symfony ■ ■■■ tumentor.png ← logo ■ ■■■ manuales/ ← estos PDFs ■■■ src/ ← código PHP ■■■ config/ ← configuración Symfony ■ ■■■ jwt/ ← claves JWT (NO en git) ■ ■■■ secrets/prod/ ← secrets encriptados ■■■ var/ ← cache, logs (writable por www-data) ■■■ vendor/ ← composer install ■■■ .env.local ← DB, mailer (NO en git)
```

```
/etc/nginx/sites-available/tumentor.com.uy /etc/mercure/Caddyfile
/etc/systemd/system/tumentor-mercure.service
/etc/systemd/system/tumentor-worker.service /usr/local/bin/tumentor-deploy ← script
de deploy /root/.tumentor_db_password ← password Postgres (solo root)
/root/.tumentor_jwt_passphrase ← passphrase JWT (solo root)
```

3. Roles y permisos

Symfony tiene jerarquía de roles definida en config/packages/security.yaml:

Roles globales (en User.roles)

- **ROLE_USER** — todo usuario logueado.
- **ROLE_FAMILY_ADMIN** — padre/madre que administra una familia (puede aprobar miembros, agregar hijos, etc.).
- **ROLE_TEACHER** — maestra. Habilita el botón "Biblioteca" en la UI, da acceso a la gestión de contenidos curados.
- **ROLE_ADMIN** — admin completo. Hereda ROLE_TEACHER, ROLE_FAMILY_ADMIN, etc.

Roles contextuales (en FamilyMembership.role)

- **parent** — padre/madre dentro de una familia específica.
- **relative** — familiar (abuelo, tío, primo).
- **child** — chico/a.
- **ai_assistant** — el usuario virtual "Guía", uno por familia.

Un usuario puede tener roles distintos en familias distintas. Por ejemplo, una mujer puede ser **parent** en la familia de sus hijos, y **relative** en la familia de sus sobrinos.

4. Promover usuarios

El registro vía UI siempre crea usuarios con ROLE_USER. Para promover, se hace por SQL directo en Postgres.

Promover a admin

```
PGPASSWORD=$(sudo cat /root/.tumentor_db_password) \ psql -h 127.0.0.1 -U guia -d
guia <<EOF UPDATE "user" SET roles = '["ROLE_USER","ROLE_ADMIN"]'::json WHERE email
= 'simonberton@gmail.com'; EOF
```

Promover a teacher (maestra)

```
UPDATE "user" SET roles = '["ROLE_USER","ROLE_TEACHER"]'::json WHERE email =
'patricia@example.com';
```

Después del UPDATE, el usuario tiene que **hacer logout y login** para que el JWT levante el rol nuevo.

5. API keys (Anthropic y OpenAI)

Tu Mentor usa el sistema de **Symfony Secrets** para guardar las API keys encriptadas en disco. La clave de desencriptación (prod.decrypt.private.php) vive en el server y **nunca se commitea**. El backup de esa clave es responsabilidad del admin (guardarla en password manager).

Setear o rotar una API key

```
cd /mnt/www/guia-parental sudo -u www-data php8.4 bin/console secrets:set
ANTHROPIC_API_KEY --env=prod # (te pide el valor por stdin, no aparece en bash
history) sudo -u www-data php8.4 bin/console secrets:list --env=prod sudo -u
www-data php8.4 bin/console cache:clear --env=prod sudo systemctl restart
tumentor-worker
```

Keys actuales en uso

- **ANTHROPIC_API_KEY**: obligatorio. La usa el chat principal (Opus) y el extractor de intereses (Haiku).
- **OPENAI_API_KEY**: opcional. La usa ModerationService para chequear cada mensaje. Si está vacía, hace fail-open con warning. Para piloto con menores: completarla.

Modelos de Claude configurados

```
# en .env.local ANTHROPIC_MODEL=claude-opus-4-7 ← chat principal (más caro y
potente) ANTHROPIC_HAIKU_MODEL=claude-haiku-4-5-20251001 ← análisis (intereses,
desafíos)
```

6. La biblioteca curada

Es la única fuente de donde la IA puede recomendar contenido. La gestiona la maestra (ROLE_TEACHER) o un admin desde el botón ■ **Biblioteca** en el header de la app.

Cómo poblarla

1. **Importar URLs sueltas**: botón "+ Importar URLs". Pegás URLs de YouTube (una por línea), el backend hace oEmbed y crea borradores con título + autor + thumbnail.
2. **Importar canal completo**: botón "+ Importar canal completo". Pegás la URL de un canal (ej. https://youtube.com/@derivando), el backend baja los últimos ~15 videos vía RSS público de YouTube (sin API key) y los crea como borradores.
3. **Agregar manual**: para artículos, ejercicios o contenidos que no son YouTube.

Aprobar items

Los borradores **no son recomendados por la IA**. Para que la IA pueda usar un item:

1. Hacer click en "Editar" en la card del item.
2. Completar: **edad mínima/máxima**, **materias** (matemática, ciencia, programación, etc.), **tags** (libres), **idioma** (es/en/pt).
3. Guardar.
4. Hacer click en "✓ Aprobar". A partir de ahí queda visible para el system prompt.

La IA solo recomienda **items de tipo VIDEO**, aprobados, activos, filtrados por edad del chico Y por la intersección de materias con el perfil + temas del aula. Los canales se mantienen para curación pero no se recomiendan automáticamente (no se pueden embeber inline en el chat).

7. Gestión de magic links

Cuando un padre crea un chico, el sistema genera un **magic link** reusable. La URL es del tipo https://tumentor.com.uy/?t=<token>.

Endpoints relevantes

- POST /api/families/{id}/children — crea chico + genera magic link.
- GET /api/families/{id}/children/{childId}/magic-link — devuelve el link actual (lazy-creates si falta).
- POST /api/families/{id}/children/{childId}/rotate-magic-link — invalida el token previo y emite uno nuevo (caso de filtración).
- POST /api/auth/login-with-token — endpoint público que intercambia el token por un JWT.

La rotación es responsabilidad del adulto admin de la familia. No hay UI todavía para "rotar" (solo "obtener"); si hace falta rotar, se llama el endpoint con curl.

8. Deploys

Hay un script /usr/local/bin/tumentor-deploy que hace toda la secuencia de deploy:

```
sudo tumentor-deploy
```

Pasos que ejecuta (todos como www-data para mantener permisos consistentes):

1. git pull (vía deploy key SSH guardada en /var/www/.ssh/).
2. composer install --no-dev --optimize-autoloader --no-interaction.
3. Limpia var/cache/prod y reapropia var/ a www-data.
4. doctrine:migrations:migrate -n (no-op si no hay nuevas).
5. cache:warmup.
6. systemctl reload php8.4-fpm.
7. systemctl restart tumentor-worker.

Flujo desde Win

1. Hacer cambios en D:\Proyectos\Guia Parental.
2. git add -A && git commit -m "..." && git push.
3. En el server (vía SSH): sudo tumentor-deploy.

9. Servicios systemd

Tres unidades systemd corren la app:

- **nginx** — recibe HTTPS, sirve estáticos, reverse-proxea a PHP-FPM y a Mercure.
- **php8.4-fpm** — ejecuta el código Symfony.
- **postgresql** — la base.
- **redis-server** — cache.
- **tumentor-mercure** — hub SSE en 127.0.0.1:3000.
- **tumentor-worker** — worker de Messenger (procesa MaybeAskAi, ExtractChildInterests).

Para controlar:

```
sudo systemctl status tumentor-worker sudo systemctl restart tumentor-mercure sudo journalctl -u tumentor-worker -f ← logs en vivo del worker sudo journalctl -u tumentor-mercure --since "10 min ago"
```

10. Logs y troubleshooting

Ubicaciones

- **Symfony**: /mnt/www/guia-parental/var/log/prod-YYYY-MM-DD.log
- **Nginx**: /var/log/nginx/tumentor.access.log y tumentor.error.log.
- **PHP-FPM**: /var/log/php8.4-fpm.log.

- **Postgres**: /var/log/postgresql/postgresql-16-main.log.
- **Worker y Mercure**: journalctl -u tumentor-worker / tumentor-mercure.

Problemas comunes

- **"Cache file is not writable"**: permisos en var/. Fix: sudo chown -R www-data:www-data /mnt/www/guia-parental/var.
- **Worker en loop de FAILURE**: revisar journalctl -u tumentor-worker -n 50. Casi siempre es messenger_messages no creada (corre messenger:setup-transports) o API key faltante.
- **IA no responde**: verificar que el worker esté active, que ANTHROPIC_API_KEY esté seteada como secret, y mirar logs por errores "Error llamando a Claude".
- **WP roto después de cambios**: hay sockets PHP separados (8.3 para WP, 8.4 para Tu Mentor). Si algún site WP empieza a usar 8.4 por error, revisar /etc/nginx/sites-enabled/* y forzar php8.3-fpm.sock explícito.

11. Backup

Crítico: hacer backup periódico de Postgres y de las claves.

Backup de Postgres

```
# Manual PGPASSWORD=$(sudo cat /root/.tumentor_db_password) \ pg_dump -h 127.0.0.1
-U guia guia > /root/backups/tumentor-$(date +%Y%m%d).sql # Cron diario (sugerido) 0
3 * * * PGPASSWORD=... pg_dump ... > /root/backups/tumentor-$(date +\%Y\%m\%d).sql
```

Archivos críticos a respaldar fuera del server

- /mnt/www/guia-parental/config/secrets/prod/prod.decrypt.private.php — sin esto, los secrets son irrecuperables.
- /mnt/www/guia-parental/config/jwt/private.pem y public.pem — sin esto, todos los JWT emitidos pierden validez.
- /root/.tumentor_db_password y .tumentor_jwt_passphrase.
- /mnt/www/guia-parental/.env.local.
- Dumps de Postgres recientes.

12. Consideraciones legales

Tu Mentor maneja datos de menores. En Uruguay aplica la **Ley 18.331** de Protección de Datos Personales, que es estricta con menores:

- Base legal explícita para el tratamiento (consentimiento del padre/madre).
- Notificación de brechas a la URCDP en un plazo definido.
- Derecho de acceso, rectificación y supresión del titular (el chico, vía sus padres).

Para el piloto en Durazno: tener firmado **consentimiento informado** con las 8 familias **antes** de que el primer chico mande un mensaje. Documento físico o digital con firma, archivado fuera del sistema.

Adicionalmente, los datos de las conversaciones se procesan en USA (Anthropic). Eso constituye una **transferencia internacional**. La base legal recomendada: el consentimiento informado del padre debe mencionar esto explícitamente.

Versión del manual: 1.0 · Mayo 2026

Contacto: simonberton@gmail.com