

TIC SERVICIOS

Central de Licencias

Personalización técnica y arquitectura

Release 1.0.0 · 20 de octubre de 2026

macOS primero. cPanel después.

Fuente autoritativa: archivos Markdown incluidos en docs/. Copie comandos desde Markdown para conservar exactamente las líneas y sustituya los placeholders.

Verificación ejecutada en Linux controlado. Mac, SMTP real, respaldo externo e instalación cPanel requieren validación de destino.

Personalización técnica - Central de Licencias

Base de datos y migraciones

Edite solo .env para credenciales/host/puerto. config/database.php selecciona mariadb, strict mode, utf8mb4 e InnoDB del servidor. Fechas de dominio y auditoría: UTC. No copie bases ni APP_KEY de otra instalación. Inspeccione primero:

```
php artisan migrate:status
```

```
php artisan migrate --pretend
```

Una migración nueva se crea con `php artisan make:migration descripcion`; revise SQL, índices, FK y compatibilidad antes de ejecutar `php artisan migrate --force`. No edite migraciones ya desplegadas. Las migraciones iniciales usan `create/up` y `drop/down`; ejecutar `down` borra datos y queda bloqueado fuera de `_test` para los comandos de `rollback`. Las ampliaciones son aditivas. Pruebe migración e instalación limpia con `php artisan test` en `tic_licencias_test`. No use `migrate:fresh` en `_dev` ni producción.

Los catálogos se definen en `app/Services/Catalog.php`; licencias, revisión firmada y Outbox se cambian exclusivamente mediante `CentralLicensing` dentro de transacciones.

Interfaz e identidad

Tokens visuales y componentes CSS en `resources/css/app.css`; estructura en `resources/views/layouts/`; JS ligero en `resources/js/app.js`. Cambie `--primary` y tokens relacionados preservando contraste, foco y estados. Paleta actual: celeste/azul, blanco, texto oscuro y grises. Recompile con `npm ci` y `npm run build`; distribuya `public/build` y `manifest.json` juntos. No se requiere Node en cPanel. No inserte scripts inline: CSP usa `self`. El menú aplica permisos y el backend vuelve a autorizar.

`APP_NAME` cambia la marca; el panel Configuración guarda organización, soporte y timezone en `settings`. Evite mostrar nombres técnicos de eventos o permisos en flujos de negocio nuevos; el catálogo técnico queda para operadores.

Seguridad y accesos

`UserPolicy` controla el recurso usuario; Los catálogos y transiciones aplican permisos backend, MFA y step-up. Roles propios y permisos directos se asignan por subconjuntos de los permisos del actor. Roles base se sincronizan con `tic:permissions:sync`: no personalice los roles base; cree roles propios. El último propietario activo no se puede desactivar ni degradar. Un permiso registrado sin vínculo no concede comportamiento por sí mismo.

API Sanctum: `read` para lectura, `write` para mutaciones, token con nombre/dispositivo y vencimiento. El token se muestra una vez. Los cambios sensibles requieren `TIC-Confirm-Password` y, para privilegiados, `TIC-MFA-Code` nuevo. Web exige contraseña confirmada en los últimos 5 minutos y MFA configurado. Los tokens nunca reemplazan los permisos del usuario. Consulte OpenAPI para contratos y códigos.

Configuración y llaves

Configuración de Central de Licencias

El inventario completo está en `.env.example`. `.env` es privado, 0600 y diferente de la otra aplicación. No se publica ni incluye en ZIP. `config:cache` copia secretos a `bootstrap/cache`: no transporte esa caché a otra máquina. Tras cambiar `.env`: `php artisan config:clear` y, en cPanel, `php artisan optimize`.

Grupo	Variables	Decisión
Identidad	APP_NAME, APP_URL, APP_ENV, APP_KEY	APP_URL canónica HTTPS para licencias; APP_KEY única, permanente y recuperable
Datos	DB_CONNECTION, DB_HOST, DB_PORT, DB_DATABASE, DB_USERNAME, DB_PASSWORD	mariadb; tic_licencias_dev local, base con prefijo en cPanel; UTC, InnoDB y utf8mb4
Sesión	SESSION_COOKIE, SESSION_DOMAIN, SESSION_SECURE_COOKIE	tic_license_center, null, true en HTTPS; no compartir dominio padre
Procesos	CACHE_STORE, SESSION_DRIVER, QUEUE_CONNECTION	database; nunca array en producción
Correo	MAIL_MAILER, MAIL_SCHEME, MAIL_HOST, MAIL_PORT, MAIL_USERNAME, MAIL_PASSWORD, MAIL_FROM_ADDRESS	smtp real en destino; smtp con STARTTLS 587 o smtps 465 según proveedor
Licencia	TIC_PRODUCT_ID, TIC_TRUST_ENV, TIC_CENTRAL_URL, TIC_NETWORK_ENABLED	producto tic.client-core; ancla dev/staging/prod independiente; false solo para desarrollo aislado
Operación	TIC_CRON_MINUTES, TIC_MONITOR_TOKEN, TIC_AUDIT_DAYS	1 o 5; token largo aleatorio; auditoría 365 días por política propuesta
Recuperación	TIC_BACKUP_DIR, TIC_BACKUP_KEY_FILE, TIC_BACKUP_REMOTE, TIC_BACKUP_SSH_KEY	directorio privado, llave 32 bytes 0600 fuera de app, destino externo usuario@host:/ruta/, SSH dedicada
MariaDB	TIC_MARIADB_DUMP, TIC_MARIADB_BIN	binarios de rama 10.11; nunca mysqldump de MySQL 9.6

TIC_SIGNING_KEY_DIRECTORY permite custodiar las llaves online fuera del proyecto. La raíz privada permanece offline y nunca se configura en el hosting.

No se aplica BILLING_*, Redis, S3, Flutter ni credenciales de servicios no implementados. La organización y timezone de visualización se cambian en Configuración; los registros de auditoría muestran UTC explícitamente.

Actualización y recuperación

Siga ACTUALIZAR.md y RESPALDO_RECUPERACION.md. Preserve .env, APP_KEY, storage y config/tic_trust.php; además la ruta externa de llaves online cuando corresponda. Nunca copie una raíz privada al hosting. No ejecute composer update en cPanel ni cambie composer.lock manualmente. Una variante requiere nuevo número de versión, pruebas y ZIP inmutable, con CHANGELOG y SHA-256.

Puntos de extensión

Los servicios en app/Services se comparten entre Blade y API. Las vistas no llaman por HTTP a la propia API. Nuevas operaciones deben definir permisos, validación, política, transacción, idempotencia, auditoría segura y pruebas. No copie tablas de negocio a la central. La biblioteca packages/license-protocol es la única implementación criptográfica. Cambios al protocolo requieren una versión y conformidad nueva, nunca renombrar campos silenciosamente.

Arquitectura - Central de Licencias

Monolito Laravel con Blade y API REST. Controllers comparten servicios, validación y transacciones; Blade no consume la propia API. MariaDB por instalación. Fortify guarda sesiones y MFA; Sanctum tokens; Spatie permisos. Base de datos, APP_KEY, cookies, llaves y ciclo de vida independientes entre las dos apps.

CentralLicensing emite documentos y eventos Ed25519 con transacción de revisión, Outbox y auditoría. SigningKeys usa anclas públicas por entorno y llaves license/event separadas. SafeOriginProof y WebhookTransport resuelven todas las IP, rechazan redes internas, fijan DNS en cURL, validan TLS y no siguen redirects. La central no tiene conexión ni credenciales para la BD de negocio del cliente.

Modelo e índices

Común: users, roles, permissions, pivotes Spatie, audits por actor/evento/fecha/request_id, settings.key único, personal_access_tokens, sessions, jobs/failed_jobs/cache/cache_locks. Idempotency_records.scope_hash único, expiración indexada 30 días.

customers.uuid único, products.code único, plans FK producto, installations.uuid y (origen,entorno) únicos, licenses UUID e índices estado, issued_documents unique(license_id,revision), Outbox.event_id único, códigos SHA-256 únicos, idempotencia unique(fingerprint,request_id), nonces hash únicos por ventana 600 s. current_license_id enlaza el registro activo de la instalación; los anteriores se conservan.

La política de datos es una organización por instalación. Módulos verticales, integración de pagos y Flutter no se implementan en 1.0.0. Contratos públicos en OpenAPI y, en cliente, EXTENDER_MODULOS.md. No hay servidor Redis ni worker permanente obligatorio.

Seguridad y datos - Central de Licencias

Rol/recurso	Acceso base
propietario	Permisos vinculados del producto; delegación, último propietario protegido
administrador	Gestión operativa, sin roles.manage ni keys.manage
consulta	Dashboard y lectura demo propia; sin administración
comercial	Clientes, planes, instalaciones y licencias; MFA obligatorio
tecnico	Productos, instalaciones, diagnóstico y auditoría; MFA obligatorio
custodio	Metadatos de llaves y auditoría; firma de raíz offline

La lista exacta está en `app/Services/Permissions.php`. Los permisos directos y por rol se unen, y solo se delegan subconjuntos del actor. UserPolicy y middleware de permisos controla catálogos, no basta ocultar el menú. Propietarios y administradores requieren MFA; todos los operadores central también. Tokens read/write no elevan permisos. Reautenticación sensible: contraseña reciente web o contraseña+TOTP API. Login tiene límite IP/cuenta y espera progresiva 1/5/15 minutos desde 5/10/15 errores; no bloquea indefinidamente.

Controles: CSRF web, escape Blade, Eloquent/SQL parametrizado, CSP self, nosniff, frame deny, no-store, HTTPS/HSTS en destino, ficheros privados MIME+size y descarga autorizada, BOLA por propiedad y permisos, CSV con prefijo protector de fórmulas, payload protocol 64 KiB y firmas/nonces. Las APIs nunca devuelven hashes de contraseña ni MFA.

Inventario mínimo y retención propuesta

Categoría	Finalidad	Acceso	Retención / salida
Nombre/correo/contraseña hash	Autenticación y gestión	Usuario propio/administrador o autorizado	Cuenta activa y plazo acordado; rectificación por administración
Roles, sesiones, tokens	Autorización y seguridad	Usuario propio y propietario	Sesiones 120 min; tokens 1-90 días, revocables
Auditoría IDs/metadatos seguros	Evidencia de cambios	audit.view	365 días propuestos; exportación propia autenticada
Logs	Diagnóstico	Operador hosting	14 días, sin secretos; dev mail log requiere custodia
Contacto de organización y licencias	Gestión contractual	Comercial autorizado	Plazo contractual y obligaciones por aprobar
Copias cifradas	Recuperación	Custodio designado	7 diarias/4 semanales/12 mensuales propuestos

El usuario puede descargar sus datos en Seguridad > Mis datos aun con licencia restringida. Rectificación por usuarios.manage. No hay borrado masivo automático de cuentas con FK ni borrado de auditoría a solicitud sin revisión del responsable. La política propuesta necesita aprobación de conservación y protección de datos antes de producción; no es una opinión legal. Datos RR.HH./salud/remuneraciones requieren evaluación propia y no se copian a la central.

No se registra el código TIC1, tokens, contraseñas, secretos MFA, llaves privadas, cuerpos de canje o .env en auditoría. request_id permite correlacionar. Revisar logs de proxy/hosting y desactivar captura de cuerpos sensibles fuera de Laravel.

Versiones del release license-center 1.0.0

Componente	Construcción Linux real	Perfil Mac/cPanel del consolidado
PHP	8.4.26	8.4.26
Laravel	13.37.0	13.x
MariaDB	10.11.14	10.11.19 (patch de destino pendiente de prueba)
Composer	2.10.3	2.10.3 local; sin Composer cPanel
Node / npm	24.19.0 / 11.9.0	24.22.0 / 11.19.1 observados en consolidado
Vite / plugin	8.3.4 / 3.x (lock exacto)	8 / 3.x
Tailwind	>=4.2.2, versión exacta package-lock.json	>=4.2.2
Fortify / Sanctum / Spatie	1.41.0 / 4.3.3 / 8.3.2	Composer lock
PHPUnit / Larastan	12.5.40 / 3.13.0	Solo desarrollo
Protocolo	1.0.0	Mismo contrato

El inventario completo está en composer.lock y package-lock.json. config.platform.php=8.4.26, platform-check activo. Sin --ignore-platform-reqs. Paquetes productivos creados en Linux x86_64 con vendor no-dev y assets compilados; no llevan Node ni binarios nativos de frontend. Se debe comprobar el patch MariaDB real del hosting antes de homologar.

UI/UX - Central de Licencias

Diseño común corporativo: fondo gris claro, tarjetas blancas, texto oscuro, acento celeste/azul #0369a1, sidebar de navegación y formularios de una o dos columnas. Sin imágenes decorativas ni framework JS pesado. Recursos locales sin CDN.

Estados vacíos, errores y confirmaciones se muestran en contexto. Foco visible, labels asociados, mensajes de validación, reducción de animación por preferencia y navegación de menú móvil por botón/overlay/Escape. Tablas tienen scroll interno; no se debe desplazar horizontalmente toda la página. El código conserva permisos backend y responsive a 360,390,768,1024,1440 px. Zoom/reflujo y capturas reales se describen en el informe de QA. Los estados de derecho y comunicación están separados y ningún timeout de central vence una PERPETUAL.

Pantallas: login, recuperación, contraseña, MFA, seguridad/sesiones/tokens, dashboard, usuarios, roles, configuración, auditoría y diagnóstico; organizaciones, productos/módulos, planes, instalaciones, licencias y llaves. Consulte [resources/views](#) y [resources/css/app.css](#) para personalización.

QA - Central de Licencias

Estado de evidencias: PROBADO_LINUX en PHP 8.4.26 y MariaDB 10.11.14. No equivale a PROBADO_MAC ni PROBADO_STAGING_CPANEL. Ejecute php artisan test con .env.testing propio en tic_licencias_test; el TestCase valida entorno y nombre antes de borrar tablas. Cada app supera los 36 vectores normativos más casos de enteros, keysets, revocación y secretos de prueba.

Pruebas funcionales: autenticación y MFA, primer propietario CLI, roles/directos, último propietario, delegación, revocación de tokens, exportación propia, headers, rate limit y guarda destructiva. Código hash/72 h/5 intentos, canje, respuesta perdida, firma de solicitud y replay, origen fallido, REKEY, estados terminales, ACK, Outbox, rollback transaccional y restauración atrasada.

El informe consolidado y qa/informes conservan resultados reales, validación visual, análisis estático y verificación de paquetes. Las pruebas de protocolo HTTPS usan prueba de origen sustituida controladamente: el canje real con DNS/TLS público se valida en H5. Se prueba defensa SSRF por código/vectores y entradas inválidas; no se presenta como certificado de pentest.

Pendientes de destino: PHP web/CLI/cron, SMTP/TLS y recepción, respaldo hacia cuenta externa real, restauración desde esa cuenta, inodos/LVE, aislamiento, Document Root y todos los alias. WCAG AA es objetivo revisado con inspección y contrastes, no una certificación formal.