

Plan de Implementación: Facturación Electrónica ARCA (ex AFIP) Multi-Inquilino

Tecnología base: TypeScript / Node.js **Estrategia de Seguridad:** Opción A (Almacenamiento encriptado de claves privadas)

🔑 Fase 0: Instructivo para el Usuario (Generación de Certificados)

Tus usuarios deben seguir estos pasos obligatorios dentro de su portal de ARCA (AFIP) con Clave Fiscal Nivel 3 para poder utilizar el sistema:

1. Generar los archivos locales:

- El usuario debe generar una Clave Privada (archivo **.key**) y un Pedido de Certificado (archivo **.csr**) mediante OpenSSL (el sistema puede proveer un script o comando automatizado para facilitarles esto).

2. Obtener el Certificado en ARCA:

- Ingresar a la web de AFIP y adherir el servicio **"Administración de Certificados Digitales"**.
- Crear un nuevo alias (ej. **SistemaFacturacion**) y subir el archivo **.csr** generado en el paso anterior.
- El sistema de ARCA validará el pedido y permitirá descargar el Certificado Digital (archivo **.crt**).

3. Asociar el Certificado al Web Service:

- Regresar al menú principal de AFIP y entrar a **"Administrador de Relaciones de Clave Fiscal"**.
 - Hacer clic en **"Nueva Relación"** -> **"Buscar"** -> **"ARCA"** -> **"Web Services"**.
 - Seleccionar **"Facturación Electrónica"** (**wsfe**).
 - En el campo **"Representante"**, buscar y asociar el alias creado previamente.
 - Confirmar la operación. A partir de este momento, el usuario ya puede subir su **.crt** y **.key** a la plataforma.
-

🏠 1. Modelo de Datos y Gestión de Secretos

Cada inquilino del sistema requiere un registro aislado en la base de datos con los siguientes campos fiscales:

- **Identificadores Públicos:** CUIT, Razón Social, Condición de IVA y Punto de Venta (habilitado exclusivamente para Web Services).
- **Certificado Público:** El contenido de texto plano del archivo **.crt**.
- **Secreto Criptográfico:** El contenido del archivo **.key encriptado en reposo** mediante el algoritmo **AES-256-GCM**.

- **Caché del Ticket de Acceso (TA):** Campos `token_acceso`, `sign_acceso` y `fecha_expiracion`. Permiten mitigar peticiones innecesarias al Web Service de Autenticación durante sus 12 horas de validez.

2. Flujo de Procesos Core del Backend

Flujo A: Carga de Credenciales (Onboarding)

1. El usuario sube los archivos `.crt` y `.key` desde su panel de configuración.
2. **Procesamiento en Memoria:** El backend intercepta los archivos estrictamente en la memoria RAM (utilizando objetos `Buffer`). **Queda prohibida la escritura de archivos temporales en el disco.**
3. **Cifrado Inmediato:** Se aplica el cifrado `AES-256-GCM` sobre la clave privada utilizando una llave maestra de 32 bytes inyectada únicamente por variables de entorno de producción (`process.env.MASTER_KEY`).
4. **Persistencia:** Se almacena el registro en la base de datos y se fuerza el vaciado de las variables en memoria para evitar fugas de datos.

Flujo B: Autenticación Eficiente (WSAA)

Antes de interactuar con el Web Service de facturación:

1. El backend consulta los datos de caché de sesión (`fecha_expiracion`) del inquilino actual.
2. **Token Vigente:** Si faltan más de 10 minutos para su expiración, se reutiliza el `token` y `sign` almacenados de forma directa.
3. **Token Expirado o Inexistente:** - Se recupera la clave `.key` encriptada de la base de datos.
 - Se descripta temporalmente en memoria RAM.
 - Se firma digitalmente el pedido de acceso (CMS) y se solicita un nuevo Ticket de Acceso a ARCA.
 - Se actualiza la caché del usuario en la base de datos para las próximas 12 horas.

Flujo C: Emisión Sincrónica y Asincrónica Resiliente (WSFE)

Este flujo mitiga la duplicación de comprobantes ante pérdidas de conexión o caídas parciales de ARCA:

1. El usuario envía la orden de facturación desde la interfaz.
2. El sistema consulta a ARCA el **último número de comprobante autorizado** para ese Punto de Venta y tipo de factura (A, B, C).
3. Se calcula el número correlativo siguiente (`Último + 1`) y se despacha la estructura de la factura a ARCA.
4. **Gestión de Respuestas:**
 - **Éxito Inmediato (Sincrónica):** ARCA devuelve el número de **CAE** y su fecha de vencimiento. La factura se marca localmente como **Aprobada**.
 - **Falla de Red / Timeout (Asincrónica):** Si el servidor de ARCA no responde en un lapso de 15 segundos o arroja un error HTTP 502/504, el sistema **no cancela la operación**. La factura se resguarda localmente en estado **Pendiente de Verificación** y se notifica al usuario que su comprobante está en procesamiento diferido.
5. **Worker de Recuperación en Segundo Plano:** Un proceso cron/worker automatizado evalúa las facturas en estado **Pendiente** cada 5 minutos. Realiza una consulta directa a ARCA

(FECompConsultar). Si ARCA confirma que el comprobante posee un CAE asignado, el sistema recupera los datos y la aprueba; de lo contrario, se declara como **Fallida** para que el usuario pueda reintentar de manera segura.

Flujo D: Generación y Entrega de PDF

Una vez que el comprobante cuenta con su correspondiente CAE:

1. El backend recupera la información de la transacción y calcula un **Código QR Obligatorio** bajo las especificaciones y URLs oficiales dispuestas por ARCA.
2. Los datos y el código QR se inyectan en una plantilla HTML parametrizada con el diseño reglamentario de una factura comercial.
3. Un motor de renderizado convierte el HTML en un archivo binario PDF en milisegundos.
4. **Almacenamiento Seguro:** El PDF se almacena en un repositorio de objetos privado y aislado (ej. AWS S3 o una terna de almacenamiento segura del servidor). La base de datos conserva únicamente la referencia de acceso.

3. Matriz Estricta de Seguridad y Control

- **Aislamiento Multi-Inquilino (Prevención de IDOR):** Cada endpoint del backend (facturación, lectura de credenciales o descarga de PDFs) debe validar rigurosamente mediante un middleware que el **usuarioId** de la sesión autenticada coincida de forma unívoca con el dueño del recurso fiscal o del comprobante solicitado.
- **Descargas Mediante URLs Firmadas:** Los archivos PDF almacenados en la nube no poseen acceso público. El sistema debe generar URLs firmadas con expiración corta (ej. 15 minutos) únicamente cuando el cliente legítimo requiera la descarga.
- **Control de Concurrencia (Bloqueo por CUIT):** Para evitar colisiones numéricas provocadas por operarios facturando en el mismo milisegundo bajo un mismo CUIT, el backend implementará un mecanismo de bloqueos distribuidos (Mutex). Las solicitudes dirigidas a un mismo identificador fiscal se procesarán estrictamente en cola.
- **Seguridad de Red Perimetral (VPC):** La base de datos que aloja los certificados cifrados debe carecer de direccionamiento IP público. El tráfico estará restringido mediante firewalls únicamente para peticiones internas originadas desde la IP privada del servidor backend TypeScript.