Guía Técnica para la Implementación de Facturación Electrónica en Go (Normativa SRI Ecuador)

1.0 Introducción: El Contexto Técnico y Regulatorio

La implementación de la facturación electrónica obligatoria en Ecuador, bajo la estricta supervisión del Servicio de Rentas Internas (SRI), representa un hito fundamental en la transformación digital del sistema tributario del país. Una correcta implementación no es solo un requisito de cumplimiento, sino una decisión estratégica que impacta la eficiencia operativa y la resiliencia fiscal de cualquier empresa. Esta guía cobra especial relevancia ante la inminente normativa que, a partir del **1 de enero de 2026**, exigirá la transmisión de todos los comprobantes electrónicos en tiempo real, eliminando los plazos de gracia actuales y demandando una arquitectura de software robusta y de alta disponibilidad.

Para afrontar este desafío, el lenguaje de programación **Go (Golang)** se posiciona como una opción tecnológica excepcionalmente adecuada. Sus ventajas en rendimiento, su modelo de concurrencia nativo para gestionar múltiples solicitudes simultáneas y sus robustas capacidades criptográficas lo convierten en una herramienta ideal para desarrollar sistemas que deben firmar, transmitir y autorizar transacciones de manera inmediata y segura, cumpliendo con las exigencias del SRI sin degradar el rendimiento de las aplicaciones de negocio.

Ante este panorama, todo equipo de desarrollo enfrenta una decisión fundamental que definirá su arquitectura y hoja de ruta: construir una solución nativa en Go desde cero o integrar una API de facturación electrónica de un proveedor tercero.

2.0 Decisión Estrategica: ¿Desarrollo Nativo en Go o Integración vía API de Terceros?

La elección arquitectónica entre desarrollar un módulo de facturación electrónica propio o consumir los servicios de un proveedor de API es una de las decisiones más críticas del proyecto. Esta elección tiene un impacto directo en el costo inicial de implementación, el esfuerzo de mantenimiento a largo plazo, el control sobre el flujo de datos y la gestión de la seguridad de los activos criptográficos.

Desarrollo Nativo en Go

Optar por un desarrollo nativo en Go implica construir cada componente del proceso de facturación electrónica internamente.

- **Ventajas:** Ofrece control total sobre el proceso, desde la generación del XML hasta la comunicación con el SRI. La firma electrónica, uno de los activos más críticos, reside en la infraestructura local del cliente, lo que maximiza la seguridad y el control de acceso.
- **Desventajas:** El costo inicial es significativamente alto, requiriendo meses de desarrollo especializado. Además, la responsabilidad del mantenimiento es total; cualquier cambio en las resoluciones o esquemas técnicos del SRI exige actualizaciones rápidas y constantes del código base para mantener el cumplimiento.

Integración vía API REST

Esta opción consiste en delegar la complejidad del cumplimiento a un proveedor de servicios especializado, comunicándose con su plataforma a través de una API REST.

- **Ventajas:** El costo inicial es bajo y el tiempo de integración se reduce a días. El mantenimiento, las actualizaciones normativas y la gestión de la comunicación con los Web Services del SRI son responsabilidad del proveedor, liberando al equipo de desarrollo de esta carga.
- **Desventajas:** Genera una dependencia de la disponibilidad y el rendimiento de un tercero. El control sobre el proceso es menor y, dependiendo del proveedor, la firma electrónica podría residir en la nube del tercero, lo que requiere un análisis de seguridad adicional. En Ecuador, existen proveedores de API como los siguientes:
    - **Azur:** Utiliza autenticación por `API-KEY` y recibe un payload `JSON` a través de una solicitud `POST`. Un detalle técnico importante es que su servicio permite un máximo de 5 decimales para los campos `cantidad` y `precioUnitario`, a diferencia del estándar general del SRI.
    - **FacturaSoft:** Opera con autenticación `Token Bearer` en los encabezados HTTP y se comunica mediante `JSON`. Su URL base es `https://services.abitmedia.cloud/facturasoft-v1-billing-documents`.

Tabla Comparativa de Factores Clave

Factor	Desarrollo Nativo en Go	Integración vía API REST
Costo Inicial	Alto (Meses de desarrollo)	Bajo (Días de integración)
Mantenimiento	Requiere actualización constante ante cambios del SRI	Delegado al proveedor de la API

Control	Total sobre el proceso y los datos	Dependiente de la disponibilidad del tercero
Seguridad de Firma	La firma reside en el cliente local	La firma puede residir en la nube del proveedor
Infraestructura	Solo el cliente de escritorio y su conexión	Requiere conexión constante a la API del proveedor

A continuación, esta guía se centrará en los detalles técnicos del desarrollo nativo en Go, el enfoque que proporciona el mayor nivel de control y personalización para cumplir con la normativa del SRI.

3.0 El Proceso Técnico de Emisión: Flujo de Trabajo de Principio a Fin

Antes de profundizar en la implementación de código, es fundamental comprender el flujo de trabajo completo para la generación, firma y autorización de un comprobante electrónico, de acuerdo con las directrices técnicas del SRI. Este proceso es secuencial y cada paso depende del éxito del anterior.

1. **Generación del Comprobante en formato XML.** Se crea la estructura del documento (factura, nota de crédito, etc.) en un archivo XML que cumple estrictamente con los esquemas XSD definidos por el SRI.
2. **Cálculo y Ensamblaje de la Clave de Acceso Única.** Se genera un identificador numérico único de 49 dígitos que sirve como número de autorización y clave de búsqueda del comprobante.
3. **Aplicación de la Firma Electrónica (Estándar XAdES-BES).** El archivo XML se firma digitalmente utilizando el certificado electrónico del emisor. Este paso garantiza la autenticidad, integridad y no repudio del documento.
4. **Comunicación y Envío al Web Service del SRI.** El XML firmado se envía a través de una solicitud SOAP al servicio web de "Recepción" del SRI para una validación inicial.
5. **Recepción de la Respuesta de Autorización.** Tras una recepción exitosa, se consulta el servicio web de "Autorización" del SRI para obtener el estado final del comprobante: `AUTORIZADO` o `NO AUTORIZADO`.
6. **Generación de la Representación Impresa (RIDE).** Una vez autorizado el XML, se genera una versión legible en formato PDF (RIDE) para su entrega al receptor.
7. **Almacenamiento y Conservación del Comprobante.** Tanto el emisor como el receptor deben almacenar el archivo XML autorizado por un período legalmente establecido.

Ahora, analicemos en detalle el primer paso técnico: la correcta estructuración del archivo XML.

4.0 Paso 1: Generación del Archivo XML en Go

El archivo XML es la base legal y fiscal de la transacción. Su correcta generación es el paso más crítico del proceso, ya que cualquier error estructural o de formato provocará el rechazo inmediato por parte de los servicios del SRI. Es un requisito indispensable que el archivo sea generado con codificación **`UTF-8 sin BOM`**, porque la presencia del Byte Order Mark, aunque invisible en muchos editores, es interpretada por los parsers del SRI como un carácter inválido al inicio del documento, lo que lleva a un rechazo inmediato.

La estructura del XML se divide en dos secciones principales: una sección común a todos los comprobantes, denominada `infoTributaria`, que contiene datos del emisor, el ambiente y el tipo de documento; y una sección específica del comprobante, como `infoFactura` para las facturas, que detalla la información de la transacción.

En Go, el método más eficiente y seguro para generar este XML es mediante la definición de `structs` que representen la estructura del comprobante. Utilizando las etiquetas de mapeo del paquete `encoding/xml`, se puede serializar un objeto de negocio directamente a un XML válido.

```
import (
	"encoding/xml"
	"fmt"
)

// Ejemplo de estructura para infoTributaria
type InfoTributaria struct {
	XMLName     xml.Name `xml:"infoTributaria"`
	Ambiente    string   `xml:"ambiente"`
	TipoEmision string   `xml:"tipoEmision"`
	RazonSocial string   `xml:"razonSocial"`
	Ruc         string   `xml:"ruc"`
	ClaveAcceso string   `xml:"claveAcceso"`
	CodDoc      string   `xml:"codDoc"`
	Estab       string   `xml:"estab"`
	PtoEmi      string   `xml:"ptoEmi"`
	Secuencial  string   `xml:"secuencial"`
	DirMatriz   string   `xml:"dirMatriz"`
}

// Función de ejemplo para serializar la estructura a XML
func GenerarXML(infoTributaria InfoTributaria) ([]byte, error) {
	// Serializar la estructura a XML con indentación para legibilidad
	xmlBytes, err := xml.MarshalIndent(infoTributaria, "", "  ")
	if err != nil {
```

```
            return nil, fmt.Errorf("error al serializar a XML: %w", err)
        }

        // Preceder el XML con el encabezado requerido por el SRI
        finalXML := []byte(xml.Header + string(xmlBytes))
        return finalXML, nil
}
```

Es crucial adherirse a las **reglas de precisión numérica** especificadas por el SRI en su ficha técnica:

- Se permite un máximo de **6 decimales** para los campos de `cantidad` y `precioUnitario`.
- Para el resto de los valores monetarios, se permite un máximo de **2 decimales**.

Una vez que el archivo XML ha sido correctamente generado, el siguiente paso lógico es construir su identificador único e irrepetible: la Clave de Acceso.

5.0 Paso 2: Construcción y Validación de la Clave de Acceso

La Clave de Acceso es el identificador único de **49 dígitos** que debe portar cada comprobante electrónico. Cumple una doble función: es el número de autorización del documento en el esquema de comunicación del SRI y actúa como la clave principal para su consulta y verificación en el portal público de la entidad. Su estructura está rigurosamente definida y debe ser generada con precisión.

Desglose de la Estructura de la Clave de Acceso

Componente	Descripción	Formato	Longitud
1	Fecha de Emisión	`ddmmaaaa`	8
2	Tipo de Comprobante	Código SRI (Tabla 3)	2
3	Número de RUC	Emisor	13

4	Tipo de Ambiente	1: Pruebas, 2: Producción	1
5	Serie	Establecimiento (3) + Punto Emisión (3)	6
6	Secuencial	Número del comprobante	9
7	Código Numérico	Aleatorio generado por el sistema	8
8	Tipo de Emisión	1: Normal	1
9	Dígito Verificador	Calculado con Módulo 11	1

Cálculo del Dígito Verificador (Algoritmo Módulo 11)

El último dígito de la clave se calcula aplicando el algoritmo Módulo 11 a los 48 dígitos anteriores. El proceso es el siguiente:

1. Cada uno de los 48 dígitos, leídos de derecha a izquierda, es multiplicado por una secuencia repetitiva de factores: 2, 3, 4, 5, 6, 7. El primer dígito a la derecha se multiplica por 2, el segundo por 3, y así sucesivamente. El séptimo dígito se multiplica por 7, y el octavo dígito reinicia la secuencia, siendo multiplicado por 2.
2. Se suman todos los resultados de las multiplicaciones.
3. Se calcula el residuo de la suma total dividida para 11 (`suma % 11`).
4. El dígito verificador se obtiene restando el residuo de 11 (`11 - residuo`).

Se deben aplicar dos reglas especiales al resultado final:

- Si el resultado es **11**, el dígito verificador es **0**.
- Si el resultado es **10**, el dígito verificador es **1**.

Una vez generado el XML y su correspondiente Clave de Acceso, el siguiente paso crucial es asegurar su autenticidad e integridad a través de la firma digital.

6.0 Paso 3: Firma Electrónica XAdES-BES en Go (El Reto Criptográfico)

La firma electrónica es el componente criptográfico que garantiza la **integridad** (el documento no ha sido alterado), **autenticidad** (la identidad del emisor es verificable) y el **no repudio** (el emisor no puede negar haber firmado el documento). El estándar exigido por el SRI es **XAdES-BES** (XML Advanced Electronic Signatures - Basic Electronic Signature) en su variante `"enveloped"`, lo que significa que el nodo de la firma se incrusta dentro del propio archivo XML que se está firmando.

Implementar el estándar XAdES-BES desde cero en Go es una tarea de alta complejidad. Requiere un manejo preciso de la canonicalización del XML (C14N), un proceso de normalización que asegura que el hash del documento sea consistente, así como la construcción de una estructura de nodos muy específica que incluye digests, propiedades firmadas y la información del certificado. Aunque la documentación más antigua del SRI puede hacer referencia a SHA-1, las mejores prácticas criptográficas actuales y los requisitos técnicos exigen el uso de **SHA-256** para todos los cálculos de digest a fin de garantizar la seguridad.

Dada esta complejidad, la solución más pragmática y segura es utilizar una biblioteca de Go especializada. Se recomienda explícitamente el uso de una librería como `goxades_sri`, ya que está específicamente diseñada y ajustada para cumplir con las particularidades de la normativa del SRI de Ecuador, simplificando enormemente el proceso de firmado.

El proceso de firmado en Go utilizando una librería de este tipo se puede resumir en los siguientes pasos:

1. **Carga del Certificado:** Se utiliza el paquete estándar `crypto/pkcs12` de Go para decodificar el archivo de firma electrónica (`.p12`) proporcionado por una entidad certificadora. De este archivo se extraen la clave privada (generalmente RSA) y el certificado X.509.
2. **Preparación del XML:** Se genera el documento XML completo, con todos sus campos y la clave de acceso, pero aún sin el nodo de la firma.
3. **Canonicalización (C14N):** La librería aplica un proceso de normalización al XML para estandarizar espacios, saltos de línea y otros caracteres, garantizando que el hash calculado sea siempre el mismo para el mismo contenido.
4. **Generación de Digests:** Se calculan los resúmenes criptográficos (hashes SHA-256) de los nodos clave requeridos por el estándar XAdES-BES.
5. **Firma RSA:** El digest principal resultante se cifra utilizando la clave privada del emisor, generando así la firma digital.
6. **Ensamblaje del Nodo `ds:Signature`:** La librería construye la estructura completa del nodo de la firma, incluyendo los digests, la firma cifrada y la información del certificado público, y la inserta en el lugar correcto dentro del archivo XML.

Advertencia Crítica: Una vez que el documento ha sido firmado, no debe ser modificado de ninguna manera. Cualquier cambio, por mínimo que sea (como un salto de línea adicional o un cambio de codificación), invalidará la firma y provocará el rechazo inmediato por parte de los servicios del SRI con un error de "Firma Inválida" (código 39).

7.0 Paso 4: Comunicación con los Web Services del SRI

La comunicación entre el sistema del contribuyente y la administración tributaria se realiza a través de Web Services que utilizan el protocolo **SOAP 1.1 y 1.2**. La aplicación desarrollada en Go deberá actuar como un cliente SOAP, capaz de construir y enviar las solicitudes requeridas a los *endpoints* del SRI y procesar sus respuestas.

Ambientes Operativos y Endpoints WSDL

El SRI dispone de dos ambientes operativos, cada uno con sus propias URLs para los servicios de recepción y autorización. Es mandatorio completar una fase exitosa de validación en el ambiente de pruebas antes de poder operar en producción.

- **Ambiente de Pruebas:**
  - Recepción: `https://celcer.sri.gob.ec/comprobantes-electronicos-ws/RecepcionComprobantesOffline?wsdl`
  - Autorización: `https://celcer.sri.gob.ec/comprobantes-electronicos-ws/AutorizacionComprobantesOffline?wsdl`
- **Ambiente de Producción:**
  - Recepción: `https://cel.sri.gob.ec/comprobantes-electronicos-ws/RecepcionComprobantesOffline?wsdl`
  - Autorización: `https://cel.sri.gob.ec/comprobantes-electronicos-ws/AutorizacionComprobantesOffline?wsdl`

Proceso de Comunicación en Dos Fases

a. Envío al Servicio de Recepción

El primer paso es enviar el comprobante para una validación inicial. Esto se realiza mediante una solicitud POST HTTPS al servicio de `RecepcionComprobantesOffline`. El cuerpo de la solicitud SOAP debe contener el archivo XML firmado y codificado en **Base64**. Este servicio realiza una validación estructural y de firma, devolviendo uno de dos posibles estados:

- `RECIBIDA`: El comprobante pasó las validaciones iniciales y ha sido encolado para su autorización final.
- `DEVUELTA`: El comprobante fue rechazado. La respuesta incluirá un código de error, un `mensaje` descriptivo y, crucialmente, un campo `informacionAdicional` con detalles técnicos del error que deben ser registrados para depuración.

b. Consulta al Servicio de Autorización

Tras obtener una respuesta `RECIBIDA`, el sistema debe consultar el estado final del comprobante en el servicio `AutorizacionComprobantesOffline`. La consulta se realiza enviando la `claveAcceso` del comprobante. La respuesta de este servicio indicará uno de los dos estados finales:

- `AUTORIZADO`: El comprobante es legalmente válido. La respuesta incluirá el XML autorizado completo.
- `NO AUTORIZADO`: El comprobante fue rechazado por una regla de negocio. La respuesta contendrá uno o más mensajes de error, como el código `39` (Firma inválida) o `45` (Secuencial ya registrado).

Advertencia Estratégica: Es un error común en las implementaciones no registrar y analizar adecuadamente el campo `informacionAdicional` en las respuestas de error. Este campo contiene la información de depuración más valiosa proporcionada por el SRI y es esencial para diagnosticar y resolver problemas de validación de manera eficiente.

Recomendación Técnica: Una buena práctica es consumir los Web Services de manera asíncrona. Después de recibir una respuesta `RECIBIDA`, el sistema debe esperar un tiempo prudencial y configurable (ej. unos segundos) antes de consultar el servicio de autorización. Esto le da tiempo al SRI para procesar el documento en su sistema interno.

Una vez obtenido el estado `AUTORIZADO`, el proceso no termina. El emisor debe cumplir con las obligaciones posteriores a la autorización.

8.0 Gestión Post-Autorización y Obligaciones del Emisor

La autorización del SRI confirma la validez fiscal del comprobante, pero no es el final del proceso. El emisor tiene obligaciones clave relacionadas con la entrega del documento al receptor y su conservación a largo plazo para fines de auditoría y control.

Generación del RIDE (Representación Impresa del Documento Electrónico)

El RIDE es la versión legible en formato PDF del archivo XML autorizado. Aunque el documento con plena validez legal es el XML, el RIDE es el formato estándar que se entrega al cliente, ya sea de forma física o, más comúnmente, adjunto en un correo electrónico. Este documento debe contener la información clave del comprobante y, de forma visible, la Clave de Acceso y el número de autorización para que el receptor pueda verificar su validez en el portal del SRI.

Obligación de Almacenamiento

La normativa ecuatoriana exige que tanto emisores como receptores conserven los comprobantes electrónicos (específicamente, los archivos XML autorizados) por un período de **7 años**. Para un sistema de software, almacenar miles de archivos XML individuales en

el sistema de ficheros puede ser ineficiente y dificultar la búsqueda y gestión. Como buena práctica, se recomienda almacenar el contenido del XML y su estado de autorización en una base de datos local (como SQLite o PostgreSQL), lo que facilita las consultas, la generación de informes y la exportación masiva en caso de una auditoría.

El cumplimiento de estas obligaciones es tan importante como la correcta emisión técnica del comprobante, y el sistema debe estar preparado para las importantes adaptaciones que requiere la nueva normativa.

9.0 Adaptación a la Normativa 2026: La Transición a Tiempo Real

El cambio regulatorio más significativo en el ecosistema de facturación electrónica de Ecuador, establecido en la **Resolución NAC-DGERCGC25-00000017**, entrará en vigencia el **1 de enero de 2026**. A partir de esta fecha, será obligatoria la transmisión de todos los comprobantes electrónicos al SRI en **tiempo real**, es decir, en el mismo momento en que se genera la operación.

Este cambio elimina el período de gracia anterior para la transmisión (que era de hasta 72 horas después de la emisión, según la Resolución NAC-DGERCGC18-00000233) y obliga a que la fecha de emisión del comprobante corresponda a la fecha corriente de la operación. Los sistemas ya no podrán emitir con fechas retroactivas ni utilizar procesos de envío por lotes de forma diferida.

Recomendación Arquitectónica Clave: El flujo de facturación en el sistema Go debe ser rediseñado para que la transmisión al SRI sea una parte integral e inmediata del proceso de guardado de la transacción. Se desaconseja explícitamente el uso de colas de envío diferidas o procesos *batch* nocturnos, ya que estos modelos inducirán al incumplimiento de la nueva norma.

Advertencia de Continuidad de Negocio: Una arquitectura síncrona en tiempo real introduce un punto de fallo crítico: ¿qué sucede si los servicios del SRI no están disponibles? La operación de venta no puede detenerse. Por lo tanto, es imperativo diseñar mecanismos de resiliencia, como **políticas de reintento inteligentes con *exponential backoff*** y la implementación de patrones como el ***circuit breaker*** para evitar sobrecargar los servicios del SRI durante una interrupción y gestionar las transacciones pendientes de forma ordenada una vez que se restablezca el servicio.

10.0 Bibliotecas y Enlaces de Utilidad

Para facilitar el desarrollo y la integración, a continuación se presenta una lista de recursos técnicos y enlaces de utilidad:

- **Biblioteca de Firma XAdES-BES para Go:**
  - `goxades_sri`: Una biblioteca recomendada, ajustada a las especificaciones del SRI de Ecuador.
  - *Enlace:* https://github.com/example/goxades_sri

- **Paquete Criptográfico Estándar de Go:**
  - `crypto/pkcs12`: Utilizado para la decodificación de archivos de firma electrónica (`.p12`).
  - *Enlace a la documentación oficial:* https://pkg.go.dev/crypto/pkcs12
- **API de Terceros (Ejemplos):**
  - **FacturaSoft API:**
    - **Autenticación:** `Token Bearer` en los encabezados.
    - **Formato:** `JSON`.
    - **Base URL:** `https://services.abitmedia.cloud/facturasoft-v1-billing-documents`
  - **Azur API:**
    - **Autenticación:** `API-KEY`.
    - **Formato:** `JSON` enviado vía método `POST`.
- **Documentación Oficial del SRI:**
  - La "Ficha Técnica de Comprobantes Electrónicos" es el documento de referencia principal que contiene todos los esquemas, tablas de códigos y reglas de validación.
  - *Enlace al portal:* https://www.sri.gob.ec (La ficha técnica se encuentra en la sección de Facturación Electrónica).