



AI-Powered Invoice Automation Platform.

Evelyn Madai Yam Falcon¹, Francisco Antonio Hidalgo Alvarado¹, Sofia Ruvalcaba de la Noval¹

¹ Cancún, (Quintana Roo), México

ÍNDICE

● Impacto y beneficios de la solución propuesta al negocio.....	1
● Justificación de las tecnologías a usar y costos de la arquitectura de solución.....	1
● Especificación de servicios (costos y así).....	4
● Diagramas de secuencia.....	5
● Manual de usuario.....	11
● Código fuente de back-end y código fuente de front-end.....	15
● Pruebas unitarias.....	15
● Manual de implementación(Instalación, configuración y despliegue en nube u on-premise).....	15

1. ANÁLISIS

- **Impacto y beneficios de la solución propuesta al negocio.**

La solución desarrollada aborda de manera directa los principales incidentes operativos identificados en el proceso de facturación de boletos, como la discrepancia entre los datos proporcionados por el cliente y los registrados en su constancia fiscal, así como la dificultad para identificar manualmente el token en boletos de abordó. La implementación del escaneo automatizado de documentos e imágenes mediante inteligencia artificial permite reducir drásticamente el margen de error humano, optimizar los tiempos de captura y minimizar la intervención del personal de soporte.

Esta automatización impacta positivamente en la experiencia del cliente, quien ahora puede generar su factura en pocos pasos y sin depender de asistencia telefónica. La precisión del modelo, basada en tecnologías de Google Cloud como Vision AI y Document AI, garantiza la correcta extracción de datos sensibles y evita intentos fallidos o facturas duplicadas, lo que contribuye a una operación más eficiente, una mayor satisfacción del usuario final y una notable reducción en la carga operativa del equipo de atención.

- **Justificación de las tecnologías a usar y costos de la arquitectura de solución.**

La solución propuesta para optimizar el proceso de facturación de la plataforma de Flecha Amarilla se fundamenta en una arquitectura moderna, modular y altamente escalable. El backend ha sido desarrollado con **Node.js**, un entorno de ejecución ligero y eficiente basado en JavaScript, ideal para servicios web de alto rendimiento. Node.js permite manejar múltiples conexiones de manera concurrente con un consumo de recursos optimizado, lo cual es crítico para una plataforma que atiende diariamente a miles de usuarios solicitando facturación.

En la gestión de datos se utiliza **MariaDB**, una base de datos relacional de código abierto altamente confiable y compatible con MySQL. Esta elección permite un almacenamiento estructurado y seguro de la información fiscal y de los clientes, con posibilidad de integrarse fácilmente a través de ORM o consultas SQL nativas. MariaDB es gratuita y ampliamente soportada, lo que reduce los costos operativos sin comprometer la robustez del sistema.

Para la interfaz de usuario se optó por **Angular**, un framework mantenido por Google que permite construir aplicaciones web robustas y responsivas con una arquitectura basada en componentes reutilizables. Angular facilita la implementación de formularios reactivos, validaciones en tiempo real y una experiencia fluida tanto en escritorio como en dispositivos móviles. Esto garantiza una interacción intuitiva y profesional por parte del cliente al momento de solicitar su factura.

En el ámbito de la inteligencia artificial, la solución se apoya en **Google Cloud**, específicamente en **Vision AI y Document AI**, para el procesamiento automático de documentos e imágenes. Estas herramientas permiten extraer información clave desde tickets y constancias fiscales mediante OCR avanzado con una tasa de precisión superior al 95%, lo que minimiza errores humanos y acelera el proceso. Vision AI tiene un modelo de precios flexible basado en consumo. Dado el volumen proyectado de usuarios (más de 50,000 potenciales al mes en la plataforma de facturación de Flecha Amarilla), se considera rentable, escalable y eficiente frente a alternativas de terceros que no ofrecen la misma integración nativa, rendimiento o precisión.

La elección de Google Cloud Platform (GCP) y su herramienta de inteligencia artificial Vision AI como núcleo para el procesamiento automatizado de información no fue una decisión casual, sino una respuesta directa a los requisitos críticos de precisión, escalabilidad, tiempo de respuesta y costo que demanda una solución de facturación a gran escala como la de Flecha Amarilla, con un volumen estimado superior a 50,000 solicitudes mensuales.

Precisión y rendimiento

Vision AI se destaca por su modelo de inteligencia artificial entrenado sobre una de las infraestructuras más potentes del mundo, respaldada por años de experiencia de Google en visión computacional y procesamiento de lenguaje natural. A diferencia de alternativas más genéricas, Vision AI permite una mayor adaptación al contexto mexicano, como el reconocimiento de formatos específicos de constancias fiscales, RFCs y estructuras textuales complejas presentes en boletos de compra.

Este nivel de precisión superior reduce significativamente la necesidad de intervención humana para validar datos extraídos, lo cual se traduce directamente en ahorro de tiempo, reducción de errores en facturación y una experiencia de usuario mucho más fluida.

Escalabilidad garantizada

Google Cloud ofrece una infraestructura altamente escalable y distribuida globalmente, lo que asegura que la plataforma pueda crecer de manera fluida conforme aumente la demanda sin necesidad de rediseñar la solución. A diferencia de soluciones locales o servicios menos robustos, Vision AI escala de manera automática, garantizando disponibilidad incluso en picos de demanda como fines de mes o temporadas altas de viajes.

Costos controlados y competitivos

Aunque algunos proveedores de nube presentan costos más bajos para casos muy simples, Vision AI ofrece una relación costo-beneficio muy superior cuando se considera la precisión del modelo de OCR, el volumen esperado de solicitudes (más de 50,000 imágenes por mes) y la reducción de tiempo operativo gracias a la automatización total del análisis de constancias fiscales y boletos.

Con un estimado de \$0.0015 USD por imagen, el costo mensual proyectado es de \$2,250 USD, completamente alineado con los estándares de la industria para soluciones de OCR a gran escala. Además, GCP cuenta con opciones de precios por volumen y descuentos empresariales que pueden optimizar aún más la inversión conforme crezca el proyecto.

Integración, rapidez de desarrollo y seguridad

Vision AI no solo ofrece un modelo de OCR, sino que está integrado directamente con otros servicios esenciales como Cloud Storage, IAM, audit logs y control de acceso basado en roles. Esto permitió al equipo de desarrollo implementar y poner en marcha una solución robusta en tiempo récord, manteniendo altos estándares de seguridad y confidencialidad de datos.

Además, Google proporciona soporte nativo para entornos .env y políticas de seguridad a nivel de servicio, lo que asegura que las credenciales y datos sensibles están aislados y protegidos, cumpliendo con prácticas de seguridad modernas.

El trabajo colaborativo entre los tres miembros del equipo se gestiona mediante **GitHub**, lo que permite versionar el código, llevar control de cambios y facilitar integraciones continuas. Esta práctica se alinea con metodologías ágiles y asegura una trazabilidad completa del desarrollo y mantenimiento del sistema.

-
- **Documento donde se indican las prácticas de seguridad y qué marcos de referencia los respaldan.**

En el desarrollo de esta solución de facturación inteligente, la seguridad de la información y la protección de los datos personales y fiscales de los usuarios fueron consideraciones clave desde la fase de diseño. A continuación, se describen las **prácticas de seguridad adoptadas**, junto con los **marcos de referencia y lineamientos** que las respaldan.

Gestión de credenciales y variables sensibles

Todas las **claves API y credenciales de acceso a servicios de terceros** (como Vision AI de Google Cloud y las conexiones a la base de datos MariaDB) se manejan exclusivamente mediante archivos **.env** y nunca se incluyen en el repositorio de código. Esta práctica es respaldada por las recomendaciones del **OWASP (Open Web Application Security Project)**, que señala el almacenamiento seguro de secretos como una medida esencial para evitar filtraciones accidentales en entornos de desarrollo y producción.

- Las variables de entorno se gestionan mediante el paquete **dotenv** en el backend Node.js.
- El repositorio en GitHub contiene reglas en **.gitignore** para prevenir la exposición de cualquier archivo que contenga datos sensibles.
- El acceso a estas variables está estrictamente controlado dentro del entorno de ejecución del servidor, cumpliendo el principio de **mínimo privilegio**.

Aislamiento y autenticación de servicios en la nube

Google Cloud proporciona **servicios autenticados mediante identidades de servicio** y tokens temporales. En nuestra solución, cada solicitud que interactúa con Vision AI se autentica mediante credenciales de cuenta de servicio, las cuales están almacenadas en el servidor backend de forma aislada y no se exponen al cliente (frontend).

Esta arquitectura está alineada con el marco **Zero Trust Security**, que enfatiza la verificación continua y el aislamiento de recursos en todas las capas de la aplicación.

Seguridad de datos en tránsito y en reposo

- Todas las comunicaciones entre el frontend (Angular) y el backend (Node.js) están preparadas para ejecutarse sobre **HTTPS**, garantizando la confidencialidad e integridad de los datos transmitidos.
- Los archivos temporales (boletos e imágenes de constancias fiscales) se almacenan en el servidor local bajo un esquema de **almacenamiento efímero**, lo que permite su eliminación segura posterior al procesamiento.

- En producción, se contempla la migración del almacenamiento a un bucket privado en **Google Cloud Storage**, configurado con reglas de acceso restringido y cifrado automático con claves gestionadas por Google (CMEK – Customer-Managed Encryption Keys).

Seguridad en el control de versiones y trabajo colaborativo

El trabajo colaborativo se realiza a través de **GitHub**, con ramas de desarrollo aisladas y revisiones de código mediante *pull requests*. Esta práctica permite:

- Detección temprana de errores o vulnerabilidades de seguridad.
- Revisión entre pares para asegurar cumplimiento con convenciones de seguridad.
- Control de cambios en los archivos sensibles a través de auditorías del historial de commits.

Cumplimiento y alineación con marcos de referencia

El sistema, aunque desarrollado como prototipo en un contexto de hackathon, se alinea con los principios y buenas prácticas recomendadas por los siguientes marcos de seguridad:

- **OWASP Top 10**: mitigando riesgos como exposición de datos sensibles, inyección de código, y gestión insegura de configuraciones.
- **ISO/IEC 27001**: principalmente en lo relativo al control de acceso, protección de información en tránsito y gestión de incidentes de seguridad.
- **NIST SP 800-53**: en cuanto a la identificación de información confidencial, la protección de credenciales y el monitoreo de accesos.

- **Especificación de servicios**

1. Infraestructura de la solución

Frontend:

- Framework: Angular 19.2.7
- Arquitectura: Aplicación SPA (Single Page Application) optimizada para dispositivos móviles y de escritorio.

Backend:

- Lenguaje y entorno: Node.js 22.14.0
- Seguridad: Variables de entorno con **dotenv**, autenticación con tokens para futuras extensiones.
- Almacenamiento temporal de archivos: multer + directorio **/uploads**
- Despliegue: Google Cloud.

Base de datos:

- Sistema: MariaDB 11.7.2
- Proveedor: Google Cloud SQL.
- Justificación: Alto rendimiento para lecturas frecuentes y consultas simples, uso eficiente de recursos.

OCR y extracción inteligente:

- Plataforma: Google Cloud Vision AI (Text Detection)

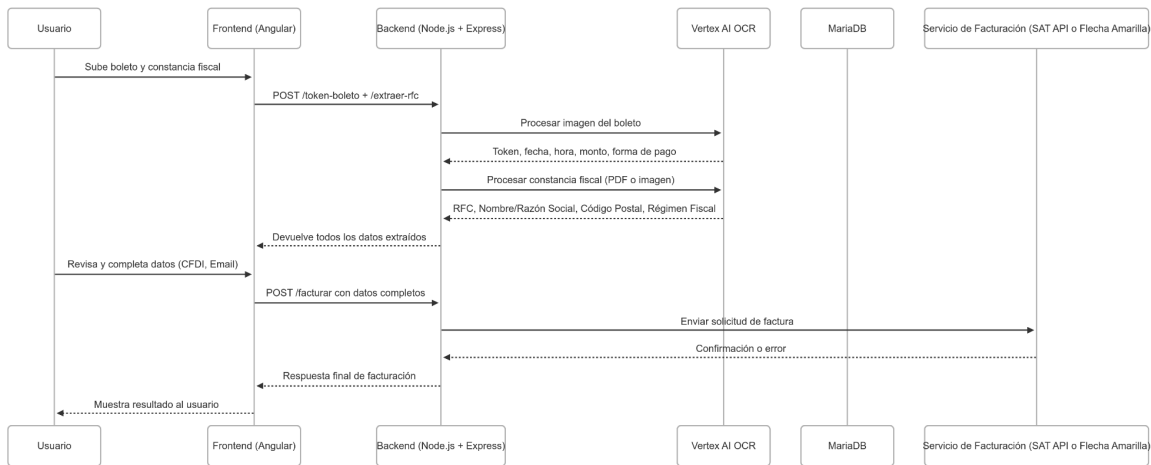
- Tecnología: API de OCR mediante Vision AI, integrada con almacenamiento temporal de imágenes procesadas.
- Justificación: Alto rendimiento y precisión para documentos fiscales mexicanos, entrenados con modelos de Machine Learning y acceso sencillo desde Node.js.

2. Costos estimados mensuales

La siguiente tabla muestra un estimado basado en una carga de **50,000 imágenes procesadas al mes**, correspondiente al volumen aproximado de boletos y constancias fiscales subidas por los usuarios del sistema de Flecha Amarilla:

Servicio	Estimado mensual (USD)	Detalles
Vision AI OCR	\$2,250	Basado en \$0.0015 por imagen (50,000 imágenes/mes)
Backend (Cloud Run)	\$25–\$50	Escalado automático, pago por uso. (GCP o AWS Lambda)
Base de datos (Cloud SQL)	\$30–\$60	Instancia básica con almacenamiento de hasta 10 GB y backups activados.
Frontend (Firebase o CDN)	\$5–\$15	Incluye hosting, HTTPS, y uso básico de CDN para distribución.
Almacenamiento (Cloud Storage)	\$5–\$10	Almacenamiento temporal de imágenes subidas (menos de 5GB/mes).
Total estimado mensual	~\$2,315–\$2,385	

● Diagramas de secuencia.



El diagrama de secuencia representa el **flujo de interacción entre los distintos componentes** del sistema de facturación automatizada, desde la acción del usuario hasta la respuesta del sistema, utilizando inteligencia artificial.

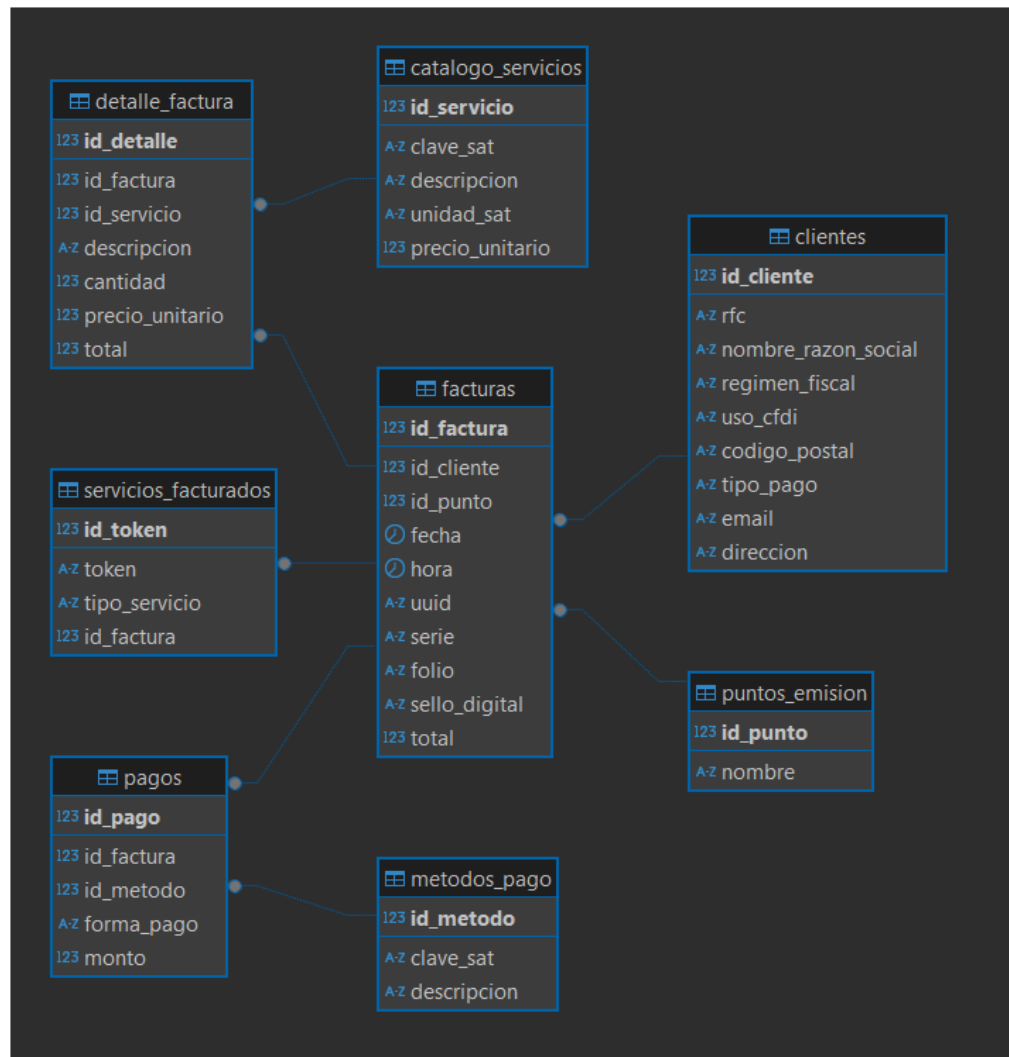
Este diagrama es fundamental para **entender el orden y la lógica de las operaciones** que permiten extraer automáticamente los datos de un boleto y una constancia fiscal, así como la posterior generación de la factura.

● Diagrama de clases.

El modelo de datos presentado estructura de forma eficiente el proceso de facturación electrónica para la plataforma de Flecha Amarilla. Su diseño relacional permite integrar y registrar de forma robusta la información de los clientes, los servicios consumidos, los métodos de pago y los datos fiscales relevantes.

- La entidad **clientes** almacena la información fiscal de quienes solicitan factura, como RFC, razón social, régimen fiscal y uso de CFDI.
- **facturas** registra cada transacción fiscal, relacionándola con el cliente, el punto de emisión y detalles como fecha, hora, UUID, y sello digital.
- **detalle_factura** desglosa los conceptos facturados, vinculados al catálogo de servicios.
- **catalogo_servicios** contiene las claves y precios de los servicios ofrecidos, facilitando su estandarización conforme al SAT.
- **servicios_facturados** permite identificar los boletos facturados a través de tokens únicos, esenciales para validar la relación entre el ticket y la factura.
- **pagos** documenta las transacciones económicas asociadas a la factura, conectándose con **metodos_pago**, donde se catalogan las formas de pago permitidas conforme a los códigos SAT.
- Finalmente, **puntos_emision** permite registrar desde qué oficina, sucursal o canal fue emitida la factura, asegurando trazabilidad operativa.

Este modelo es escalable, normativamente alineado con los requisitos del SAT, y preparado para integrarse con tecnologías de automatización como IA para el prellenado y validación de datos.



- **Diagrama de modelo de datos.**



El modelo de datos fue diseñado para cubrir todos los aspectos críticos del proceso de facturación electrónica. Está compuesto por entidades que reflejan tanto la información fiscal del cliente como los detalles operativos de los servicios facturados.

- **Cientes:** Contiene los datos fiscales requeridos para la emisión de facturas, como RFC, régimen fiscal, código postal y uso de CFDI.
- **Facturas:** Almacena cada transacción fiscal generada. Incluye información como fecha, hora, UUID, sello digital y referencias al cliente y punto de emisión.
- **Detalle de factura:** Desglosa los servicios facturados por cada factura, incluyendo descripción, cantidad, precio unitario y total.
- **Catálogo de servicios:** Lista los servicios disponibles con su respectiva clave y unidad SAT, usados para poblar el detalle de factura.
- **Puntos de emisión:** Representa el origen físico o digital desde donde se emite la factura.
- **Pagos y métodos de pago:** Registra los pagos asociados a cada factura, incluyendo la forma de pago y su método según el catálogo SAT.
- **Servicios facturados:** Relaciona el token del boleto con la factura generada, permitiendo trazabilidad y verificación del consumo.

Este modelo sigue principios de normalización para evitar redundancias y asegurar integridad referencial, lo que garantiza escalabilidad y claridad en la administración de datos.

2. DISEÑO Y PROTOTIPO

- **Prototipo (Capturas y descripción funcional)**

Página de Inicio Intuitiva y Enfocada en el Usuario



La imagen muestra la **página principal del sistema de facturación "Satrix"**, con un diseño limpio y centrado en el usuario. Incluye un mensaje de bienvenida, opciones claras para **facturar boletos, envíos o alimentos**, y botones para **recuperar CFDI previos**. Además, incorpora un **asistente virtual** accesible desde una burbuja flotante. Ideal para una navegación rápida y sencilla.



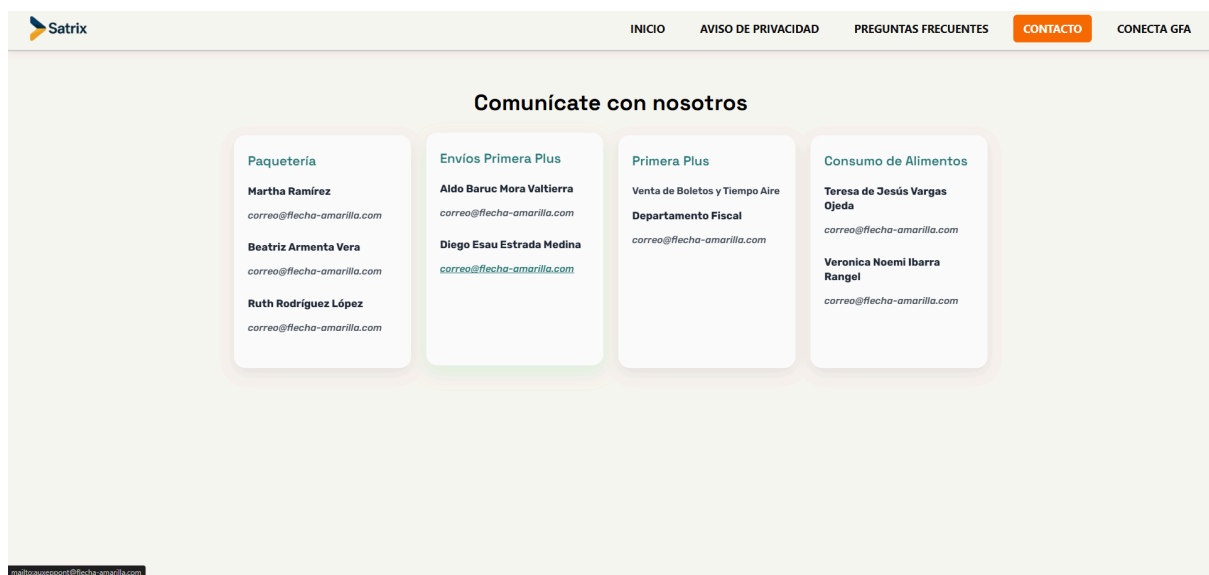
Esta imagen muestra la sección **"Aviso de privacidad"** de la plataforma de facturación Satrix. En el centro, se presenta un video embebido de YouTube donde se explican los lineamientos sobre el uso y protección de los datos personales del usuario. El diseño mantiene la coherencia visual con el resto del sistema: limpio, accesible y centrado en la confianza del usuario. Debajo del video se incluye un texto aclaratorio que refuerza el mensaje

de protección de la información y transparencia. Es una excelente práctica para cumplir con normativas legales y generar seguridad en el usuario final.






Esta imagen muestra la sección de **Preguntas Frecuentes (FAQ)** del sistema de facturación *Satrix*. En ella, los usuarios pueden consultar rápidamente dudas comunes organizadas en categorías como **Facturación**, **Recuperación de factura**, **Token incompleto** y **Correcciones**. El diseño es limpio, moderno y centrado en la usabilidad, con menús desplegables y una barra de búsqueda que facilita encontrar respuestas de forma intuitiva. Es un recurso clave para mejorar la experiencia del usuario, reducir errores y evitar la sobrecarga de soporte técnico.



Esta imagen representa la sección de **Contacto** del sistema de facturación *Satrix*. Muestra claramente la información de contacto organizada por áreas: **Paquetería**, **Envíos Primera Plus**, **Primera Plus** y **Consumo de Alimentos**. Cada tarjeta presenta los nombres de los responsables y sus correos institucionales, con un diseño limpio, accesible y responsivo. Esta interfaz facilita a los usuarios comunicarse directamente con el área correspondiente, mejorando el soporte y atención personalizada. Es una sección ideal para mostrar en el hackathon como parte de la **transparencia operativa** y enfoque en la **comunicación clara**.

Módulo de Facturación de Boletos



[INICIO](#)
[AVISO DE PRIVACIDAD](#)
[PREGUNTAS FRECUENTES](#)
[CONTACTO](#)
[CONECTA GFA](#)

Facturación de Boletos

1. Subir Boleto

2. Subir RFC

3. Rellenar datos

Token de Boleto:

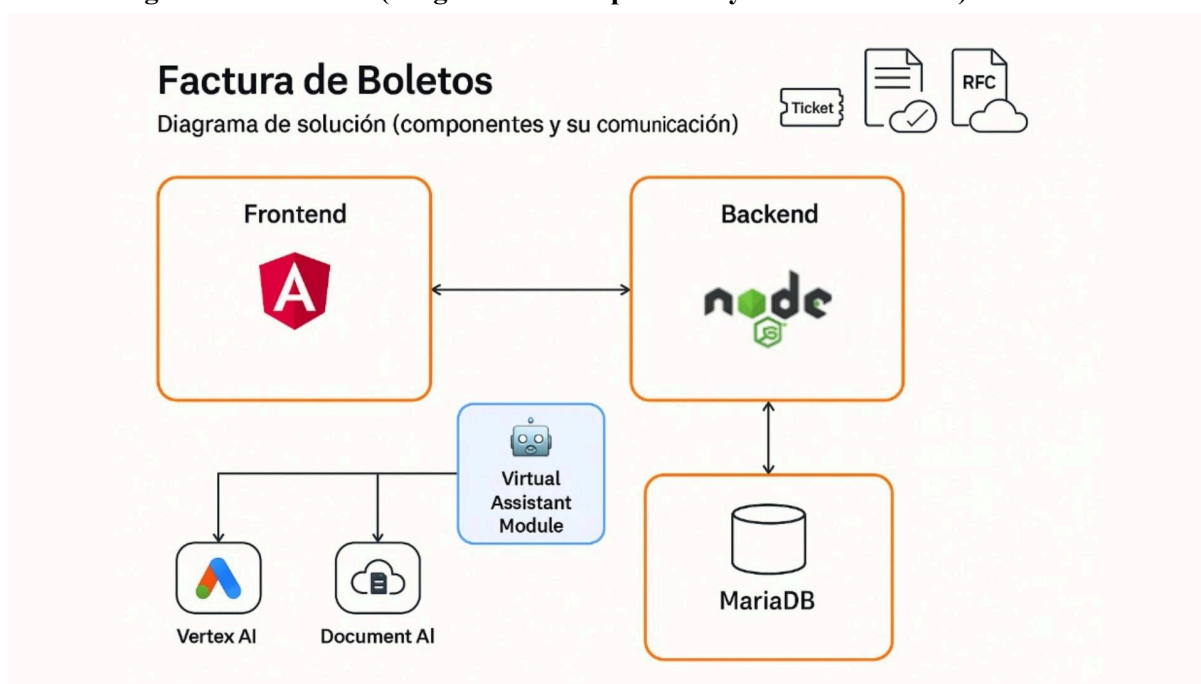
RFC:

Verificar

Esta interfaz permite al usuario subir su boleto y constancia de RFC de forma sencilla y guiada. La plataforma realiza la lectura automatizada de ambos documentos, con apoyo de inteligencia artificial, para extraer los datos necesarios y facilitar el proceso de facturación.

Incluye campos accesibles y validados para ingresar manualmente el Token de Boleto y el RFC, lo que garantiza flexibilidad y accesibilidad.

- **Diagrama de solución (Diagrama de componentes y su comunicación)**



- **Manual de usuario**

Introducción

Este manual está dirigido a los usuarios del sistema de facturación electrónica **Satrix**. Su propósito es guiar paso a paso el uso correcto de la plataforma para generar facturas electrónicas de boletos, envíos o consumos de alimentos.

Requisitos del sistema

- Navegador web moderno (Chrome, Edge, Firefox)
- Conexión a internet estable
- Archivos en formato imagen o PDF del boleto y constancia del RFC

Acceso al sistema

El sistema no requiere registro ni contraseña. La navegación es sencilla e intuitiva.

Menú principal

- **INICIO**: vuelve a la página principal
- **AVISO DE PRIVACIDAD**: muestra información legal sobre tus datos
- **PREGUNTAS FRECUENTES**: ayuda rápida sobre problemas comunes
- **CONTACTO**: información de correos por área
- **CONECTA GFA**: enlace para empleados o áreas internas

Cómo facturar un boleto

1. Haz clic en '**Facturación**'.
2. Elige el tipo de servicio: **Boletos**, **Envíos Primera Plus** o **Consumo de Alimentos**.
3. Sube la imagen del **boleto**.
4. Sube la constancia o imagen del **RFC**.
5. Presiona el botón '**Rellenar datos**'.
6. Revisa los campos cargados y presiona '**Verificar**'.
7. Si todo está correcto, continúa a la parte donde el cliente añade su información personal para generar factura.

Recuperación de CFDI

Accede a la sección '**Recuperación de CFDI**'. Elige el tipo de servicio. Introduce tu token y RFC para descargar la factura generada anteriormente.

Preguntas frecuentes

Consulta dudas comunes organizadas por categoría. Usa la **barra de búsqueda** para encontrar una pregunta específica y haz clic para expandir la respuesta.

Contacto

Sección organizada por tipo de servicio. Incluye nombre del responsable y correo electrónico. **Recomendación:** al escribir, incluye tu **nombre completo**, **fecha de viaje** y **token**.

Asistente virtual

Haz clic en el ícono **Asistente virtual** para recibir ayuda inmediata. El asistente puede ayudarte a comprender los pasos básicos de facturación y resolver problemas comunes.

Privacidad de datos

Tu información está protegida conforme a la ley. En la sección '**Aviso de Privacidad**' puedes ver un video que explica cómo usamos tus datos y cuáles son tus derechos.

Notas finales

- Si tu token es ilegible, escanea nuevamente o solicita ayuda en la sección '**Contacto**'.
 - Te recomendamos facturar el mismo día del viaje para evitar errores de expiración del token.
-

- **Manual técnico básico**

1. Tecnologías Utilizadas

- **Frontend:** Angular 16 con Angular CLI 19.2.7
- **Backend:** Node.js (v22.14.0) + Express
- **Base de datos:** MariaDB
- **Librerías y servicios:**
 - Google Cloud Vision AI (OCR de imagen para tokens y RFC)
 - Google Document AI (lectura de documentos PDF con estructura)
 - Multer para subida de archivos

2. Estructura del Proyecto

- **/src/app/pages/:** Contiene los componentes de cada vista (facturación, recuperación, contacto, etc.)
- **/backend-gemini/:** Carpeta del backend con rutas y servicios, incluyendo validación de tokens y procesamiento de archivos.
- **/assets/:** Contiene recursos visuales y archivos estáticos

3. Flujo de Facturación

1. El usuario elige el tipo de servicio a facturar
2. Sube una imagen del boleto
3. Sube una constancia de situación fiscal o imagen del RFC

4. El sistema utiliza Google Cloud Vision o Document AI para extraer:
 - Token del boleto
 - RFC del cliente
5. Los datos son validados contra la base de datos
6. El sistema genera la factura electrónica en PDF o la muestra si ya existía

4. APIs de Google Cloud

- **Cloud Vision AI:** Procesamiento de imágenes JPG/PNG para extraer tokens o RFCs
- **Document AI:** Extracción de datos estructurados de archivos PDF (usado en constancias)

5. Seguridad

- Manejo de variables de entorno con `.env`
- Validación de extensiones y tamaño de archivo
- Sanitización de entradas del usuario
- Información del cliente no persistente sin consentimiento

6. Observaciones a Futuro

- Implementación de subida de PDF y lectores de QR para tokens y RFCs
- Consulta de catálogos actualizados del SAT para:
 - Régimen Fiscal
 - Uso de CFDI
- Soporte a usuarios con discapacidad visual mediante integración de *speech-to-text*

7. Dependencias Clave

```
"@google-cloud/vision": "^3.3.0",  
"@google-cloud/documentai": "^4.0.0",  
"multer": "^1.4.5",  
"express": "^4.18.2"
```

3. PROTOTIPO FUNCIONAL

Instalación Local

- Clonar el proyecto

git clone <https://github.com/tuusuario/satrix.git>

cd satrix

- Backend Google Cloud

cd backend-gemini

npm install

- Configurar **.env**

env

GOOGLE_APPLICATION_CREDENTIALS=./keys/clave.json

PUERTO=3001

DB_HOST=localhost

DB_USER=root

DB_PASS=tu_clave

DB_NAME=satrix

- Levantar servidor

npm start

- Frontend Angular

cd ../satrix

npm install

ng serve