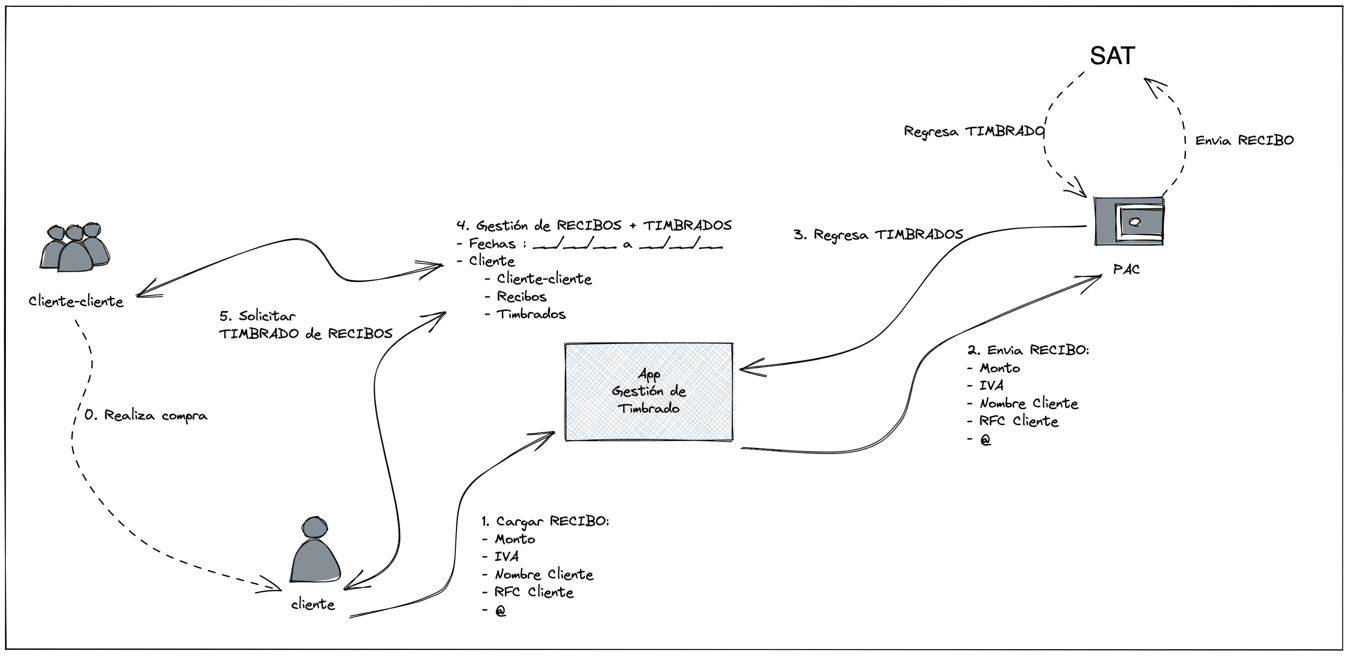
**Modelo conceptual**



**Conceptos:**

* Recibos: Comprobante físico/digital del registro de la compra
* Facturas: Timbrado (PDF y xml)
* Cliente : Nuestro cliente
* Cliente-cliente: Cliente del cliente
* Timbrado : xml y pdf timbrados.
* L2BC: somos nosotros.

**1. Cargar recibo**

Cargar recibos en la plataforma para su posible timbrado

**Cliente**

Como cliente quiero cargar recibos en la plataforma para obtener timbrado

*Datos entrada*:

* Nombre
* RFC
* Fecha
* Monto

*Datos salida*:

* Timbrado (PDF + xml)

**2. Enviar recibo**

Enviar recibos en la plataforma para su posible timbrado.

**L2BC**

Como L2BC quiero enviar recibos desde la plataforma para obtener timbrados de mis clientes y clientes-clientes

*Datos entrada*:

* Nombre
* RFC
* Fecha
* Monto

*Datos salida*:

* Timbrado (PDF + xml)

**3. Regresa timbrado**

Recibir timbrados en la plataforma para su gestión.

**L2BC**

Como L2BC quiero recibir facturas en la plataforma para gestionar timbrados de mis clientes y clientes-clientes.

*Datos entrada*:

* Nombre
* RFC
* Fecha
* Monto

*Datos salida*:

* Timbrado (PDF + xml)

**4. Gestión de RECIBOS y TIMBRADOS**

Recibir timbrados en la plataforma para su posible timbrado

**Cliente**

Como cliente quiero consultar mis recibos y timbrados para una correcta gestión con el SAT

*Datos entrada*:

* Login
* Búsqueda por fechas \_\_/\_\_/\_\_ \_\_/\_\_/\_\_
  + Nombre.
  + Cliente-cliente.

*Datos salida*:

* Listado en pantalla.
* Por fechas \_\_/\_\_/\_\_ \_\_/\_\_/\_\_
  + Nombre.
  + Cliente-cliente.
  + Timbrado (PDF + xml)

**Cliente-cliente**

Como cliente-cliente quiero consultar mis recibos y timbrados para una correcta gestión con el SAT

*Datos entrada*:

* Login
* Búsqueda por fechas \_\_/\_\_/\_\_ \_\_/\_\_/\_\_
  + Nombre.

*Datos salida*:

* Listado en pantalla.
* Por fechas \_\_/\_\_/\_\_ \_\_/\_\_/\_\_
  + Nombre.
  + Timbrado (PDF + xml)

**L2BC**

Como L2BC quiero clasificar los timbrados de mis clientes y clientes-clientes.

*Datos entrada*:

* Recibos ← cliente, cliente-cliente

*Datos salida*:

* Listato de timbrados por:
  + Fecha
  + Cliente-cliente

**5. Solicitar timbrado de recibos**

Posterior a la consulta, poder solicitar el timbrado de recibos, obviamente no es posible si ya esta timbrado.

**Cliente**

Como cliente quiero solicitar timbrar mis recibos.

*Datos entrada*:

* Login
* Búsqueda
* Solicita timbrar recibos
  + Correo electrónico para el envío automático

*Datos salida*:

* Recibir timbrado (PDF + xml) en plataforma y por correo electrónico.

**Cliente-cliente**

Como cliente-cliente quiero solicitar timbrar mis recibos.

*Datos entrada*:

* Login
* Búsqueda
* Solicita timbrar recibos
  + Correo electrónico para el envío automático.

*Datos salida*:

* Recibir timbrado (PDF + xml) en plataforma y por correo electrónico.

**Requerimiento Técnicos API**

A continuación se presentan los conceptos claves que se deben considerar al diseñar las API REST.

* Autenticación/Autorización
* Validación de entrada
* Definir tipos de contenido
* Codificación de la salida
* Seguridad para datos en tránsito y almacenamiento
* Responder con los códigos de estado apropiados

**Autenticación/Autorización**

Aunque a veces estos términos se consideran como si fueran sinónimos en realidad no lo son.

* Autenticación: La autenticación es el proceso de identificar si las credenciales pasadas juntos con la solicitud son válidas o no, aquí las credenciales se pueden pasar como ID de usuario y contraseña o un token asignado para la sesión de usuario.
* Autorización: La autorización es el proceso de identificar o validar si la solicitud recibida puede acceder al punto final o método solicitado.

Nota: La Autorización debe procesarse después de la Autenticación y solo si esta última es correcta.

**Validación de entrada**

La validación de entrada se debe aplicar a los niveles sintáctico y semántico .

* Sintáctico: Debe aplicar la sintaxis correcta de los campos estructurados (por ejemplo: RFC, CURP, fechas, etc.).
* Semántico: Debe de validar si los datos de entrada cumplen las reglas de negocio (por ejemplo: cuando un campo es requerido o no, si debe existir o no, sumas de importes correctos, valores dentro de catálogos, etc.)

**Definir tipos de contenido**

Se debe de definir que tipo de contenido serán aceptados, JSON, XML, Texto Plano, etc. En caso de recibir un contenido fuera de los aceptados se deberá informar además de un mensaje descriptivo también se deberá devolver el código HTTP correcto: 406 Unacceptable o 415 Unsupported Media Type.

**Codificación de la salida**

Se deberá identificar correctamente el tipo de contenido de salida a fin de que sea aceptado e interpretado correctamente JSON, XML, Texto Plano, etc.

**Seguridad para datos en tránsito y almacenamiento**

Solo se debe de aceptar datos y peticiones a nuestra API utilizando el protocolo HTTPS en caso de que el cliente inicie la comunicación mediane HTPP se debe actualizarlo a HTTPS y manejar la solicitud.  
Se recomienda que todo dato almacenado sea cifrado así como las copias de seguridad.  
Responder con los códigos de estado apropiados para evitar la ambigüedad  
A fin evitar la ambigüedad, deberá responder además de un mensaje descriptivo adicionar el código HTTP estándar correcto:  
• 201 – Creado  
• 200 – OK  
• 202 – Aceptado y en cola para procesamiento  
• 204 – Sin contenido  
• 304 – No modificado  
• 400 – Petición Incorrecta  
• 401 – No autorizado  
• 403 – Prohibido  
• 404 – No encontrado  
• 405 – Método no permitido  
• 406 – No aceptable (utilizado con tipos de contenido)  
• 415 – Tipo de medio no compatible  
• 429 – Demasiadas solicitudes