

Requerimientos de Web Services (API) - Módulo Móvil de Gastos y Viáticos

Objetivo: Permitir la captura, validación fiscal y registro de gastos de viaje desde una App Móvil hacia el ERP.

1. Catálogos (Endpoints GET)

Necesarios para las listas desplegables en la interfaz del usuario.

Endpoint Sugerido	Descripción Funcional	Datos Mínimos de Respuesta (JSON)
GET /api/catalogs/expense-categories	Listado de conceptos de gasto (ej. Alimentos, Hospedaje, Vuelos).	id, description, icon_name (opcional), default_gl_account (interno).
GET /api/catalogs/projects	Centros de costo o proyectos a los que se imputará el gasto.	id, name, customer_name, status (solo activos).
GET /api/catalogs/payment-methods	Formas de pago disponibles para el empleado.	id, description, type (Reembolsable vs. Corporativa).
GET /api/catalogs/currencies	Monedas soportadas y tipo de cambio actual.	currency_code (MXN, USD), exchange_rate (del día).

2. Procesamiento de Facturas/XML (Endpoints POST)

Este flujo consta de dos pasos para la integridad fiscal y operativa.

Paso A: Validación y Lectura de XML (Pre-llenado)

Endpoint: POST /api/expenses/validate-xml

- **Propósito:** La App envía el archivo XML crudo. El servidor lo valida ante el SAT, verifica que no esté duplicado en el ERP y extrae la información para evitar captura manual.
- **Input (Request):**
 - Archivo .xml (Multipart/Form-data).

- **Output (Response):**
-

JSON

```
{  
    "isValid": true,           // Si es falso, mostrar mensaje de error (ej. Cancelado)  
    "validationMessage": "OK",  
    "data": {  
        "uuid": "A1B2C3D4-...", // Folio Fiscal  
        "rfcInfo": {  
            "rfc": "XAXX010101000",  
            "name": "Empresa Patito S.A."  
        },  
        "amount": 1500.00,  
        "currency": "MXN",  
        "date": "2023-10-27T12:00:00"  
    }  
}
```

Paso B: Registro del Gasto ([Guardado Final](#))

Endpoint: POST /api/expenses/submit

- **Propósito:** Guardar el gasto definitivo uniendo los datos del usuario + los datos validados del XML + Archivos adjuntos.
- **Input (Request Payload):**

JSON

```
{  
    "employeeId": "12345",
```

```

"expenseDate": "2023-10-27",
"categoryId": 10,           // ID seleccionado del catálogo (ej. Alimentos)
"projectId": 505,          // ID seleccionado del catálogo
"paymentMethodId": 1,      // ID seleccionado (ej. Tarjeta Personal)
"currencyCode": "MXN",
"exchangeRate": 1.0,
"totalAmount": 1500.00,
"notes": "Comida con cliente",
"fiscalData": {             // Puede ir null si es gasto sin factura (propina)
    "uuid": "A1B2C3D4-...",
    "xmlFileName": "factura.xml"
}
}

```

- **Adjuntos:** Archivo PDF (representación gráfica) y XML.

3. Lógica Importante (Para el Backend)

Hay que considerar las siguientes reglas:

1. **Proveedores Nuevos:** Si el RFC del XML no existe en el catálogo de proveedores (Vendors) del ERP, el servicio submit debe ser capaz de:
 - Crearlo automáticamente usando los datos del XML, **O**
 - Asignar el gasto a un "Proveedor Genérico de Viáticos" (Hay que definir la estrategia).
2. **Gastos sin XML:** El endpoint submit debe aceptar peticiones donde el campo fiscalData o uuid venga vacío (para casos como propinas, taxis de sitio o gastos en el extranjero).
3. **Duplicidad:** El sistema debe bloquear el intento de subir un XML con un UUID que ya fue registrado anteriormente por cualquier usuario.