

# **Definición de API externa para conexión con software SOLVI usando Api Rest**

**Proyecto:** Routemod (Módulo de rutas SOLVI simplificado para encargados de entrega en PYMES)

**Tecnologías principales:** Android Studio (Kotlin) + API REST en PHP + MySQL

## **1. Introducción**

Este documento especifica las librerías, servicios externos y tecnologías que van a ser utilizadas en el desarrollo del proyecto final de esta clase: Una aplicación destinada a repartidores, la cual permitirá consultar rutas asignadas, ver detalle de entregas y registrar la información de llegada, salida y finalización de cada entrega diaria.

Para comunicar la app implementada con la base de datos ya existente y la lógica empresarial del sistema Solvi (Software web ya existente creado en PHP y MySQL), se creará una API REST sencilla y funcional desde cero. Este documento justifica las decisiones tomadas y confirma que todas las herramientas elegidas son gratuitas, accesibles y adecuadas para el proyecto.

## **2. Objetivo del API**

Existen varias razones por las cuales se decidió usar una API REST:

- Validación de credenciales (login) con usuario ya existente al ser afiliado al SAAS Solvi.
- Descarga de ruta asignada al chofer diariamente.
- Consulta del detalle de cada entrega.

- Registro de hora de salida, llegada y entrega completada, así como las coordenadas geográficas.
- Consulta y actualización del inventario de objetos a entregar en cada destino.

El API actuará como intermediario entre la aplicación Android y la base de datos MySQL del sistema Solvi. Por eso es indispensable.

### **3. Tecnologías Seleccionadas (Se solicitó a la IA una amplia variedad de opciones, con pros y contras, y se terminaron seleccionando estas, sobre todo por su accesibilidad, tanto en experiencia necesaria para usarlas como en precio)**

Componente	Tecnología	Justificación
Lenguaje Backend	PHP	Lenguaje usado actualmente en Solvi
Framework Backend	Slim Framework (PHP)	Micro-framework sencillo, rápido y recomendado para APIs ligeras
Base de Datos	MySQL	Ya utilizada por el sistema empresarial Solvi
Auth	JWT (JSON Web Token – librería Firebase PHP JWT)	Autenticación simple, sin sesiones, estándar moderno
CORS	Middleware Slim	Necesario para que Android pueda consumir la API
Servidor	Apache (XAMPP o hosting PHP)	Gratis, fácil de configurar
Cliente móvil	Android Studio (Kotlin)	
Consumo API en Android	Retrofit + Gson	Simplifica uso de APIs REST y parseo JSON

### **4. Librerías Específicas**

#### **Backend (PHP/Slim)**

Librería	Descripción	Modo
Slim Framework	Manejo de rutas y respuestas	PHP Composer (gratuito)
Firebase JWT PHP	Generación y verificación de tokens JWT	PHP Composer
Dotenv (vlucas/phpdotenv)	Manejo seguro de variables del entorno (BD, claves)	PHP Composer

Estas librerías permiten crear un API con pocas líneas de código, buena organización y seguridad básica sin complejidad innecesaria.

## Frontend móvil (Android / Kotlin)

Librería	Función
Retrofit	Comunicación con la API
Gson Converter	Conversión JSON ↔ objetos Kotlin
Kotlin Coroutines	Manejo simple de llamadas asíncronas
OkHttp (incluido con Retrofit)	Manejo de solicitudes HTTP

Estas son librerías estándar para consumo de APIs Android, fáciles de integrar y ampliamente documentadas.

## 5. Arquitectura de la Solución

Android App (Kotlin) → Retrofit → API REST (Slim PHP) → MySQL (Solvi DB)

El flujo será el siguiente:

1. El usuario ingresa sus credenciales en la app.
2. La app envía los datos al endpoint `/login`.
3. Si las credenciales son correctas, la API responde con un token JWT.
4. La app almacena el token y lo usa para cada solicitud.
5. La app consulta la ruta con `/ruta/{idChofer}`.
6. El conductor marca eventos (salida, llegada, entrega finalizada, registrar punto geográfico).

7. La API actualiza MySQL y devuelve confirmación.

---

## 6. Endpoints Planeados (Esquema)

Endpoint	Método	Descripción
/login	POST	Autentica usuario y devuelve token JWT
/ruta/{choferId}	GET	Obtiene lista de destinos asignados
/entrega/{idEntrega}	GET	Obtiene detalle de una entrega
/entrega/{idEntrega}/salida	POST	Registra hora de salida
/entrega/{idEntrega}/llegada	POST	Registra hora de llegada
/entrega/{idEntrega}/finalizar	POST	Marca entrega finalizada
/inventario/{idEntrega}	GET	Muestra ítems a entregar
/entrega/{idEntrega}/geo	POST	Registra las coordenadas del lugar

## 7. Costo y Licencias

Todas las herramientas seleccionadas son gratuitas y open-source.

## 8. Cierre

La arquitectura propuesta nos parece viable, funcional y adecuada para las expectativas del proyecto. Utiliza herramientas modernas pero sencillas, garantiza compatibilidad con el sistema actual basado en PHP y MySQL, y permite extender el proyecto en el futuro.