

Guía de Uso - Colección Postman TPI Logística

Descripción General

Esta colección de Postman proporciona una suite completa de pruebas para la arquitectura de microservicios de logística, incluyendo:

- **API Gateway** (`ms-gateway`): `http://localhost:8080`
 - **Microservicios:**
 - `ms-solicitudes`: Gestión de solicitudes de transporte y clientes
 - `ms-rutas`: Gestión de rutas, tramos, depósitos y tarifas
 - `ms-camiones`: Gestión de camiones y asignaciones
 - **Seguridad:** Keycloak (`http://localhost:8180`) con autenticación OAuth2
-

Instalación

Paso 1: Descargar los archivos

Descarga estos dos archivos en la carpeta raíz del proyecto:

- `TPI_Logistica.postman_collection.json`
- `TPI_Logistica.postman_environment.json`

Paso 2: Importar en Postman

1. Abre Postman
2. **Importar Colección:**
 - Haz clic en **Import** (parte superior izquierda)
 - Selecciona `TPI_Logistica.postman_collection.json`
3. **Importar Entorno:**
 - Haz clic en el ícono de **Ambientes** (parte superior derecha)
 - Haz clic en **Import**
 - Selecciona `TPI_Logistica.postman_environment.json`

Paso 3: Seleccionar el Entorno

En Postman, en la parte superior derecha, asegúrate de seleccionar el entorno "**TPI Logística - Entorno**".

Configuración de Autenticación

Variables de Entorno (Credenciales)

Antes de ejecutar las peticiones, actualiza las credenciales en el entorno según tus usuarios de Keycloak:

Variable	Descripción	Valor por Defecto
----------	-------------	-------------------

Variable	Descripción	Valor por Defecto
user_cliente	Email del usuario CLIENTE	cliente@mail.com
pass_cliente	Contraseña del usuario CLIENTE	1234
user_operador	Email del usuario OPERADOR	operador@mail.com
pass_operador	Contraseña del usuario OPERADOR	1234
user_transportista	Email del usuario TRANSPORTISTA	transportista@mail.com
pass_transportista	Contraseña del usuario TRANSPORTISTA	1234

Para actualizar las credenciales:

1. Haz clic en el icono de Ambientes (parte superior derecha)
2. Selecciona **"TPI Logística - Entorno"**
3. Haz clic en **Editar**
4. Modifica los valores de los usuarios y contraseñas
5. Haz clic en **Guardar**

Flujo de Ejecución Recomendado

Fase 1: Obtener Tokens (Autenticación)

Primero, **ejecuta los 3 requests de autenticación en esta orden:**

1. **1. Autenticación (Keycloak) > Obtener Token CLIENTE**
2. **1. Autenticación (Keycloak) > Obtener Token OPERADOR**
3. **1. Autenticación (Keycloak) > Obtener Token TRANSPORTISTA**

Estos requests ejecutarán un script de test que guardará automáticamente los tokens JWT en las variables de entorno.

¿Cómo sé si funcionó?

- Si ves una respuesta con **access_token** y el test pasa (verde), los tokens se han guardado correctamente.
- Si ves un error 401 o 403, verifica que los usuarios y contraseñas sean correctos.

Fase 2: Ejecutar el Flujo Principal (Demo)

Una vez obtenidos los tokens, ejecuta el flujo completo en esta orden:

1. **(CLIENTE) Crear Solicitud:** Crea una nueva solicitud de transporte
2. **(OPERADOR) Crear Ruta:** Crea una ruta para esa solicitud (usa la ID guardada automáticamente)
3. **(OPERADOR) Asignar Camión a Tramo:** Asigna un camión al primer tramo
4. **(TRANSPORTISTA) Iniciar Tramo:** El transportista inicia el tramo
5. **(TRANSPORTISTA) Finalizar Tramo:** El transportista finaliza el tramo

6. **(OPERADOR) Finalizar Solicitud:** El operador finaliza la solicitud completa

Variables de Encadenamiento (Chaining)

Durante el flujo, se capturan automáticamente:

- `id_solicitud_creada`: ID de la solicitud (usado en pasos posteriores)
 - `id_ruta_creada`: ID de la ruta creada
 - `id_tramo_1`: ID del primer tramo
-

Fase 3: Pruebas Adicionales (CRUD)

Después del flujo principal, puedes ejecutar pruebas individuales de cada microservicio:

3. CRUD - ms-solicitudes

- Crear, listar y obtener solicitudes
- Crear y obtener clientes
- Obtener estado de solicitudes

4. CRUD - ms-rutas

- Crear rutas y tramos
- Listar rutas con paginación
- Obtener costos y tarifas vigentes
- Asignar camiones a tramos

5. CRUD - ms-camiones

- Crear y listar camiones
 - Buscar camiones aptos por peso y volumen
 - Obtener tramos asignados a un transportista
-

Estructura de la Colección

```
TPI Logística (Colección)
├── 1. Autenticación (Keycloak)
│   ├── Obtener Token CLIENTE
│   ├── Obtener Token OPERADOR
│   └── Obtener Token TRANSPORTISTA
├── 2. Flujo Principal (Demo)
│   ├── (CLIENTE) Crear Solicitud
│   ├── (OPERADOR) Crear Ruta
│   ├── (OPERADOR) Asignar Camión a Tramo
│   ├── (TRANSPORTISTA) Iniciar Tramo
│   ├── (TRANSPORTISTA) Finalizar Tramo
│   └── (OPERADOR) Finalizar Solicitud
```

- └─ 3. CRUD - ms-solicitudes
 - └─ Crear Solicitud
 - └─ Listar Solicitudes
 - └─ Obtener Solicitud por ID
 - └─ Obtener Estado de Solicitud
 - └─ Actualizar Estado Contenedor
 - └─ Crear Cliente
 - └─ Obtener Cliente por ID
- └─ 4. CRUD - ms-rutas
 - └─ Crear Ruta
 - └─ Listar Rutas (Paginado)
 - └─ Obtener Rutas por Solicitud
 - └─ Obtener Ruta por ID
 - └─ Asignar Ruta
 - └─ Obtener Costo de Traslado Real
 - └─ Obtener Tarifas Vigentes
 - └─ Asignar Camión a Tramo
 - └─ Iniciar Tramo
 - └─ Finalizar Tramo
 - └─ Listar Tramos por Patente
 - └─ Obtener Tramos por Patente (públicos)
- └─ 5. CRUD - ms-camiones
 - └─ Crear Camión
 - └─ Listar Camiones
 - └─ Obtener Camiones Aptos
 - └─ Obtener Tramos del Transportista

Endpoints por Rol

CLIENTE (client@mail.com)

- **POST /solicitudes** - Crear solicitud de transporte
- **GET /solicitudes/{id}** - Ver su solicitud
- **GET /solicitudes/{id}/estado** - Ver estado de entrega

OPERADOR (operador@mail.com)

- **GET /solicitudes** - Listar todas las solicitudes
- **POST /rutas** - Crear rutas
- **PUT /tramos/{id}/asignar-camion** - Asignar camión a tramo
- **GET /camiones/buscar-apto** - Buscar camiones disponibles
- **PUT /solicitudes/{id}/finalizar** - Finalizar solicitud

TRANSPORTISTA (transportista@mail.com)

- **POST /tramos/{id}/iniciar** - Iniciar tramo
- **POST /tramos/{id}/finalizar** - Finalizar tramo
- **GET /camiones/transportistas/me/tramos** - Ver sus tramos asignados

Uso de Tests

Cada request de autenticación tiene un **Test Script** que:

1. Extrae el **access_token** de la respuesta
2. Lo guarda en la variable de entorno correspondiente
3. Lo valida automáticamente

Otros requests en el flujo principal capturan IDs de respuestas anteriores y las guardan en **variables de la colección**, permitiendo encadenamiento automático.

Ver logs de tests:

Después de ejecutar un request, abre la pestaña **Test Results** para ver qué variables se guardaron.

Troubleshooting

Error 401 Unauthorized

- **Causa:** Token expirado o no válido
- **Solución:** Vuelve a ejecutar los requests de autenticación (Fase 1)

Error 403 Forbidden

- **Causa:** El usuario no tiene el rol necesario
- **Solución:** Verifica que el usuario tenga el rol correcto en Keycloak (CLIENTE, OPERADOR, TRANSPORTISTA)

Error 404 Not Found

- **Causa:** La ID de recurso no existe
- **Solución:** Asegúrate de ejecutar los requests en el orden recomendado para capturar las IDs correctas

Error 500 Internal Server Error

- **Causa:** Error en el servidor
- **Solución:** Verifica los logs del microservicio correspondiente

Las variables no se llenan automáticamente

- **Causa:** El script de test no se ejecutó
 - **Solución:**
 1. Ejecuta el request nuevamente
 2. Abre la pestaña **Test Results** para ver si hubo errores
 3. Verifica que estés usando el entorno correcto
-

Consejos Útiles

Ejecutar toda la colección

1. Haz clic derecho en la carpeta (ej. "2. Flujo Principal")
2. Selecciona **Run Folder**
3. Postman ejecutará todos los requests en orden

Inspeccionar respuestas

- Abre la pestaña **Body** para ver la respuesta JSON
- Usa la pestaña **Tests** para ver qué variables se guardaron

Variables de Entorno vs Colección

- **Variables de Entorno:** Compartidas entre requests y persistentes (credenciales, URLs)
- **Variables de Colección:** Específicas de la colección y usadas para encadenamiento (IDs de recursos)

Contacto y Soporte

Si encuentras problemas:

1. Verifica que todos los microservicios estén corriendo
2. Comprueba que Keycloak tenga los usuarios configurados correctamente
3. Revisa los logs de los servicios para errores específicos
4. Asegúrate de que el entorno de Postman esté seleccionado correctamente

DTOs Usados

SolicitudRequestDTO

```
{
  "clienteDni": "12345678",
  "pesoContenedor": 500,
  "volumenContenedor": 1000,
  "origenLatitud": -31.4201,
  "origenLongitud": -64.1888,
  "destinoLatitud": -31.5350,
  "destinoLongitud": -64.2637
}
```

CrearRutaRequest

```
{
  "idSolicitud": 1,
  "tramos": [
    {
      "idDepositoOrigen": 1,
```

```
    "idDepositoDestino": 2,  
    "idTipoTramo": 1  
  }  
]  
}
```

AsignarCamionRequest

```
{  
  "patenteCamion": "ABC123",  
  "pesoContenedor": 500,  
  "volumenContenedor": 1000  
}
```

Versión: 1.0

Última Actualización: 17/11/2025

Autor: QA Senior - Sistema TPI Logística