



**FUNDACIÓN
KINAL**

Centro Educativo Técnico Laboral Kinal

Materia: 5to Perito Informática

Profesor: Eduardo Hor

Sección: IN5CM

Manual de Usuario

Nombre: Josué Daniel Jolón Motta

Carné: 2022205

Fecha de Entrega:

13/02/2026



Manual de Usuario – Repuestos Automotricez

Descripción General del Sistema

Este sistema es una API REST desarrollada con Spring Boot que permite gestionar proveedores, repuestos y ventas dentro de un negocio automotriz. El sistema permite crear, consultar, actualizar y eliminar registros desde Postman. También incluye validaciones, manejo de errores y conexión a base de datos MySQL usando una arquitectura por capas para mantener el orden del proyecto.

Requisitos para usar la aplicación

- Computadora con Windows, Linux o macOS.
- Java JDK 17 o superior.
- IntelliJ IDEA o IDE similar.
- Maven.
- MySQL configurado.
- Postman para pruebas.

Inicio de la aplicación

1. Abrir el proyecto en IntelliJ IDEA.
2. Verificar que MySQL esté activo.
3. Revisar application.properties.
4. Ejecutar la clase principal.
5. Esperar que arranque en <http://localhost:8080>.

Guía rápida para un usuario general

Si nunca has usado una API REST, aquí te va una explicación. Primero se inicia el sistema y luego se utiliza Postman para enviar solicitudes. Cada solicitud tiene un método (GET para ver datos, POST para crear, PUT para actualizar y DELETE para eliminar). El usuario solo necesita copiar el endpoint, elegir el método correcto y enviar el JSON cuando sea necesario. Siempre es recomendable probar primero los GET para verificar que los datos antes de que actualicemos y para no tener inconvenientes.



FUNDACIÓN KINAL

También es importante revisar los errores que devuelve el sistema. Si sale 404 significa que el ID no existe, si sale 400 probablemente algún campo está vacío o mal escrito. La idea es que el usuario pueda probar todo paso a paso sin necesidad de modificar el código del Proyecto y que vea la funcionalidad y control de errores que hay en el programa.

Endpoints – Empleados

GET /api/empleados

GET /api/empleados/{id}

POST /api/empleados

JSON:

```
{  
  "nombre_empleado": "Juan",  
  "apellido_empleado": "Perez",  
  "puesto_empleado": "Vendedor",  
  "email_empleado": "juanperez@gmail.com"  
}
```

PUT /api/empleados/{id}

JSON:

```
{  
  "nombre_empleado" : "juan",  
  "apellido_empleado" :"perez",  
  "puesto_empleado" : "vendedor",  
  "email_empleado" : "juanlperez@gmail.com"  
}
```

DELETE /api/empleados/{id}

Endpoints – Proveedores

GET /api/proveedores

GET /api/proveedores/{id}

POST /api/proveedores



FUNDACIÓN KINAL

JSON:

```
{  
  "nombre_proveedor": "AutoPartes GT",  
  "telefono_proveedor": "55554444",  
  "email_proveedor": "autopartesgt@gmail.com"  
}
```

PUT /api/proveedores/{id}

JSON:

```
{  
  "nombre_proveedor": "Proveedor 4",  
  "telefono_proveedor": 555334,  
  "direccion": "Calle falsa 123",  
  "email_proveedor": "proveedor4@example.com",  
}  
DELETE /api/proveedores/{id}
```

Endpoints – Repuestos

GET /api/repuestos

GET /api/repuestos/{id}

POST /api/repuestos

JSON:

```
{  
  "nombre_repuesto": "Filtro de aceite",  
  "precio": 75.50,  
  "stock": 20,  
  "id_proveedor": 1  
}
```

PUT /api/repuestos/{id}

JSON:

```
{  
  "nombre_repuesto": "repuesto2",  
  "categoria_repuesto": "categoria2",  
  "precio_compra": 123,
```



FUNDACIÓN KINAL

```
    "precio_venta" : 124,  
    "id_proveedor" : 2  
}  
DELETE /api/repuestos/{id}
```

Endpoints – Ventas

GET /api/ventas

GET /api/ventas/{id}

POST /api/ventas

JSON:

```
{  
  "fecha_venta": "2026-02-10",  
  "cantidad": 2,  
  "total": 150.00,  
  "id_empleado": 1,  
  "id_reuesto": 1  
}
```

PUT /api/ventas/{id}

JSON:

```
{  
  "fecha_venta": "2024-04-01",  
  "cantidad": 22,  
  "total": 10,  
  "id_empleado": 2,  
  "id_reuesto": 2  
}
```

DELETE /api/ventas/{id}

Manejo de Errores

400 → Datos inválidos o campos vacíos.

404 → ID no encontrado.

Errores FK → IDs relacionados inexistentes.

Errores de validación → correos inválidos o formatos incorrectos.