

Universidad Mariano Gálvez de Guatemala

Boca del monte

Facultad de ingeniería en sistemas de la información

Curso: Programación II

Sección: A



Fidian Bianchi Morales Pastran 7690-22-22593

Guatemala, 11 de noviembre de 2025

## 1. Descripción general

El sistema **SMGIAAN API** (Sistema de Monitoreo y Gestión Integral de Atletas de Alto Nivel) es una API REST desarrollada en **Java 17 + Spring Boot + MySQL**, diseñada para registrar, consultar y gestionar información de atletas, entrenamientos, estadísticas y pagos.

## 2. Arquitectura del sistema

- **Backend:** Java con Spring Boot
- **Base de datos:** MySQL
- **Modelo MVC:**
  - modelo/ → Clases de entidades (Atleta, Entrenamiento, Pago, etc.)
  - persistencia/ → DAO y conexión con la base de datos
  - servicio/ → Lógica de negocio
  - controlador/ → Endpoints REST (APIs públicas)

### Estructura del paquete:

```
com.smgiaan
└── modelo/
└── persistencia/
└── servicio/
└── controlador/
└── Application.java
```

## 3. Tecnologías

Tecnología	Versión	Uso
Java	17	Lenguaje principal
Spring Boot	3.2+	Framework backend
Maven	3.9+	Gestión de dependencias
MySQL	8+	Base de datos relacional
Swagger / OpenAPI 3		Documentación interactiva
Postman	—	Pruebas de endpoints

## 4. Endpoints principales

### Atletas

- GET /api/atletas — Lista todos los atletas

- POST /api/atletas — Registra un atleta

## Estadísticas

- GET /api/estadisticas/promedio/{atletald} — Promedio de rendimiento
- GET /api/estadisticas/mejorMarca/{atletald} — Mejor marca
- GET /api/estadisticas/resumen/{atletald} — Resumen general

## Entrenamientos

- GET /api/entrenamientos/atleta/{atletald} — Listar por atleta
- POST /api/entrenamientos — Registrar entrenamiento

## Pagos

- GET /api/pagos — Listar todos los pagos
- POST /api/pagos — Registrar manualmente
- POST /api/pagos/calcular — Calcular pago automático

## 5. Configuración de conexión

Archivo: ConexionBD.java

```
private static final String URL = "jdbc:mysql://localhost:3306/atletas";
private static final String USER = "fidiantest";
private static final String PASSWORD = "9n9jCHb9";
```

## 6. Ejecución

1. Asegurarse de tener MySQL levantado.
2. Abrir el proyecto en **NetBeans / IntelliJ**.
3. Ejecutar con el comando Maven:
4. mvn spring-boot:run
5. Acceder a:
  - **API Base:** <http://localhost:8080/api>
  - **Swagger UI:** <http://localhost:8080/swagger-ui/index.html>

## 7. Base de datos

Tablas principales:

- atletas
- entrenamientos
- pagos

## 8. Seguridad y manejo de errores

- Validaciones básicas en controladores.
- Manejo de excepciones con ResponseEntity.
- Mensajes en formato JSON.

## 9. Exportaciones

- Los pagos pueden exportarse a CSV mediante:
- GET /api/pagos/exportar?rutaArchivo=C:/export/pagos.csv

### Manual de Usuario — “SMGIAAN API”

#### 1. Ingreso

- Ejecutar la aplicación y abrir el navegador en:

👉 <http://localhost:8080/swagger-ui/index.html>

The screenshot shows the Swagger UI interface for the 'entrenamiento-controlador' endpoint. The URL is `PUT /api/entrenamientos/{id}`. The 'Parameters' section shows a required parameter `id` of type `integer($int32)`. The 'Request body' section shows a schema example for the JSON payload:

```
{
  "atletaId": 0,
  "fecha": "2025-11-08",
  "tipoEntrenamiento": "string",
  "valorRendimiento": 0,
  "tip": "string",
  "valor": 0,
  "normal": true,
  "pais": "string",
  "id": 0,
  "ubicacion": "string"
}
```

The interface includes a 'Try it out' button, a sidebar with icons for Home, Camera, and Manage, and a status bar indicating the URL `http://localhost:8080/swagger-ui/index.html#/entrenamiento-controlador`.

## 2. Registro de datos

1. Ir a **POST /api/atletas**
2. Presionar "Try it out"
3. Ingresar un JSON como:
4. {
5.   "nombre": "Juan Pérez",
6.   "edad": 24,
7.   "pais": "México",
8.   "disciplina": "Atletismo"
9. }
10. Presionar **Execute** → se registrará el atleta.

## 3. Registrar entrenamiento

1. Abrir POST /api/entrenamientos
2. Ingresar:
3. {
4.   "atletald": 1,
5.   "fecha": "2025-11-05",
6.   "tipoEntrenamiento": "Velocidad",
7.   "valorRendimiento": 92.5,
8.   "ubicacion": "CDMX",
9.   "pais": "México"
- 10.}

## 4. Ver estadísticas

Ejemplo:

 Promedio de rendimiento →  
GET /api/estadisticas/promedio/1

 Resumen general →  
GET /api/estadisticas/resumen/1

## 5. Calcular pago automático

Ir a:

POST /api/pagos/calcular?atletald=1&mes=11&periodo=2025

Esto genera y guarda el pago correspondiente.

## **6. Exportar datos**

En Swagger:

GET /api/pagos/exportar?rutaArchivo=C:/export/pagos.csv

## **7. Consultar pagos**

GET /api/pagos → lista todos los pagos

GET /api/pagos/atleta/1 → pagos de un atleta

## **8. Cierre del sistema**

Solo detener el servidor en NetBeans o terminal.

Los datos quedan almacenados en MySQL.