

MANUAL TÉCNICO DEL SISTEMA

“ZonaFit – Gestión de Clientes”

1. INTRODUCCIÓN

El presente manual técnico describe la arquitectura, componentes, requisitos, instalación, estructura interna y funcionamiento del sistema ZonaFit, una aplicación desarrollada en Java 17, Spring Boot y con uso de MySQL.

El sistema permite la administración de clientes dentro de un gimnasio, proporcionando operaciones de registro, consulta, actualización y eliminación.

El objetivo de este documento es proporcionar la información técnica necesaria para desarrolladores, administradores del sistema y personal de soporte, facilitando modificaciones, despliegues y mantenimiento del software.

2. REQUISITOS TÉCNICOS

Software necesario

Para compilar, ejecutar o dar mantenimiento al proyecto se requiere:

Java Development Kit (JDK) 17

Cómo descargar e instalar JDK 17

Entrar al sitio oficial de Oracle

Ve a: <https://www.oracle.com/java/technologies/javase/jdk17-downloads.html>

(Es la fuente oficial y segura para descargar Java 17)

Elegir tu sistema operativo

En la página verás opciones como:

- Windows
- Linux
- macOS

Selecciona la que corresponda a tu computadora.

Descargar el instalador

En Windows: selecciona el archivo .exe

En macOS: selecciona el archivo .dmg

En Linux: puedes elegir paquetes .rpm, .deb o archivos tar.gz

Haz clic en Download.

Aceptar la licencia

Oracle te pide marcar un cuadro que dice que aceptas el acuerdo de licencia.

Instalar

En Windows: doble clic en el instalador → "Next" → "Next" → "Install".

En macOS: abrir el .dmg → instalar.

En Linux: instalar con tu gestor de paquetes o extraer la carpeta.

Configurar variables de entorno (solo Windows)

Java normalmente lo hace automáticamente, pero si no:

Ir a Configuración del sistema → Variables de entorno

Agregar a PATH la ruta:

C:\Program Files\Java\jdk-17\bin

Verificar la instalación

Abre la terminal o CMD y escribe:

`java -version`

Debe mostrar algo como:

java version "17.x.x"

Esto requerido debido a las características modernas del lenguaje y compatibilidad con Spring Boot.

Spring Boot

Framework principal para controladores, servicios, inyección de dependencias y servidor embebido.

Spring Boot no es un programa que se instala, sino un conjunto de dependencias que se agregan automáticamente a tu proyecto mediante Maven o Gradle.

Maven automáticamente descarga:

Spring Core

Spring Web

Spring JPA

Starter de la base de datos

Tomcat embebido

Y todo lo necesario al compilar o ejecutar el proyecto.

IDE recomendado

IntelliJ IDEA

Cómo descargar IntelliJ IDEA

1. Entrar a la página oficial

Ve a: <https://www.jetbrains.com/idea/download/>

(Es el sitio oficial y seguro de JetBrains.)

2. Elegir edición

Verás dos versiones:

IntelliJ IDEA Community (GRATIS)

Recomendada para:

- Java
- Spring Boot
- Maven
- Proyectos de escuela

Suficiente para tu proyecto ZonaFit

IntelliJ IDEA Ultimate

Incluye más herramientas empresariales

Es de pago

Para tus necesidades: elige COMMUNITY.

3. Elegir tu sistema operativo

Puedes elegir:

- Windows
- macOS
- Linux

Haz clic en el botón Download de tu sistema.

4. Instalar IntelliJ

En Windows:

Abrir el archivo .exe

Next → Next → Install

(Opcional) Marcar:

"Add to PATH"

"Create Desktop Shortcut"

En macOS:

Abrir .dmg

Arrastrar IntelliJ a Applications

En Linux:

Extraer el .tar.gz

Ejecutar /bin/idea.sh

5. Primer inicio

Al abrir IntelliJ:

Acepta los término

Crea un nuevo proyecto o abre el tuyo

Verificación

Para verificar que todo funciona:

Crea un nuevo proyecto Java

Selecciona Java 17

Compila un programa sencillo

Base de datos recomendable

MySQL

JavaFX (si la aplicación llegará a tener una interfaz gráfica adicional)

Aunque el proyecto actual es orientado a API REST, se deja integración abierta.

Hardware mínimo recomendado:

- Procesador Dual Core 2.0 GHz
- 4 GB de memoria RAM (mínimo)
- 2 GB de espacio en disco para proyecto + dependencias
- Conexión a Internet (para descargar librerías Maven)

3. ARQUITECTURA DEL SISTEMA

La arquitectura implementada corresponde a una arquitectura por capas

1. Capa Modelo (Model)

Contiene las entidades que representan las tablas de la base de datos (Cliente.java).

2. Capa Repositorio

Implementa JPA para acceso a datos (ClienteRepositoryBarbosa.java).

3. Capa Servicio (Service Layer)

Contiene las reglas de negocio y lógica interna (ClienteServicio.java) y (IClienteServicio.java).

4. Capa Controlador (Controller)

Expone las funcionalidades como API REST (GestionClientesFlores.java).

5. Capa de Aplicación (Bootstrapping)

Contiene la clase principal que levanta el servidor Spring Boot (GymPosJEGC7567).

4. DESCRIPCIÓN DE COMPONENTES PRINCIPALES

Modelo → Cliente.java

Representa la entidad cliente:

- id (Integer) – clave primaria
- nombre (String)
- apellido (String)
- membresia (Integer)

Los atributos se mapean con JPA para persistencia.

Repositorio → ClienteRepositoryBarbosa.java

Extiende JpaRepository proporcionando:

- Métodos automáticos de CRUD
- Query derivation
- Integración con Hibernate

Servicio (Interfaz) → IClienteServicio.java

Define las operaciones de negocio:

- listarClientes()
- buscarPorId(id)
- guardarCliente(cliente)
- eliminarCliente(cliente)

Servicio (Implementación) → ClienteServicio.java

Implementa las reglas de negocio, utilizando el repositorio para acceso a BD.

Controlador → GestiónClientesFlores.java

- GET /clientes → lista
- GET /clientes/{id} → buscar
- POST /clientes → guardar
- DELETE /clientes/{id} → eliminar

Retorna objetos

Aplicación Principal → GymPosJEGC7567.java

Clase principal con la anotación:

@SpringBootApplication

Permite iniciar el servidor.

5. ESTRUCTURA DEL PROYECTO (Diagrama UML)

src/main/java/gm/zona_fit/

```
|—— GymPosJEGC7567.java  
|—— modelo/  
|   |—— Cliente.java  
|—— repositorio/  
|   |—— ClienteRepositorioBarbosa.java  
|—— servicio/  
|   |—— IClienteServicio.java  
|   |—— ClienteServicio.java  
└—— controlador/  
    |—— GestiónClientesFlores.java
```

6. DIAGRAMA UML (RESUMEN)

- ClienteServicio → implementa → IClienteServicio
- ClienteRepositoryBarbosa → maneja → Cliente
- GestionClientesFlores → usa → ClienteServicio
- GymPosJEGC7567 → ejecuta la aplicación

7. MANEJO DE ERRORES

El sistema usa:

- ResponseEntity para códigos HTTP
- Mensajes detallados al cliente
- Excepciones controladas en la capa servicio

8. MANTENIMIENTO Y ESCALABILIDAD

El proyecto puede ampliarse fácilmente añadiendo nuevas entidades:

- Productos
- _ Entrenadores
- Pagos
- Membresías detalladas

Spring Boot y JPA permiten modificar o agregar funcionalidades sin afectar partes ya definidas.

9. RECOMENDACIONES PARA FUTURAS MEJORAS

- * Integración con JavaFX para interfaz de escritorio
- * Implementación de JWT para seguridad
- * Manejo de roles (admin/usuario)
- * Documentación automática con Swagger
- * Implementación de pruebas unitarias con JUnit 5

12. CONCLUSIÓN

El sistema ZonaFit es una aplicación moderna basada en Java 17 y Spring Boot, diseñada para ser modular, mantenible y escalable.

Este manual proporciona toda la información necesaria para instalar, ejecutar y extender el proyecto de forma segura y eficiente.