**Instrucciones de ejecución**

Github: <https://github.com/EGarcia30/testMomoto>

1. Requisitos previos:  
   tener instalado Git, Java JDK 8 o posterior, y un IDE o editor de codigo  
   como Intellij o VSCode.
2. Clonar el repositorio  
   Abre la terminal (Git Bash en Windows)  
   Utiliza el comando git clone seguido de la URL del repositorio de GitHub( git clone https://github.com/<usuario>/<nombre-del-repositorio>.git)
3. Navegar al directorio del proyecto  
   Utiliza el comando cd para cambiar al directorio del proyecto descargado:  
   cd <nombre-del-repositorio>
4. Compilar el proyecto con Maven  
   Ejecuta el siguiente comando para compilar el proyecto y crear el archivo JAR:  
   mvn clean install
5. Ejecutar aplicación  
   Una vez que la compilación sea exitosa, se creará un archivo JAR en el directorio target. Ejecuta la aplicación con el siguiente comando:  
   java -jar target/<nombre-del-archivo>.jar  
     
   Reemplaza <nombre-del-archivo> con el nombre del archivo JAR generado en el directorio target.
6. Acceder a la aplicación  
   La aplicación se ejecutará en el puerto 8080 por defecto. Abre un navegador web y visita http://localhost:8080.

**Informe**

Las tecnologías y librerias a utilizar:

* Html
* CSS
* Java
* Springboot
* Springboot web
* Thymeleaf
* Bootstrap 5.3
* Maven
* Jquery

Tome la decision de ocupar estas tecnologías porque cumplían con eficiencia   
lo que requeria la prueba tecnica ocupando un orden de carpetas independientes tanto como para la API Rest y la interfaz web.  
  
API Rest se ocupo la arquitectura MVC dejando una responsabilidad única a cada objeto, actuando como intermediario de POKEAPI y el consumo de esta en mi proyecto la carpeta de servicios.  
  
Interfaz Web se ocupo la arquitectura MVC para dejar una responsabilidad única de objetos lógicos pero interconectados para el funcionamiento web.

**Mejoras**

El proyecto puede recibir mejoras como integrar una base de datos en memoria y almacenar los datos de entrenador y equipo, así mismo poder realizar el CRUD en la API Rest y mostrarlo en la interfaz web y estas realizarlas a través de JQuery con Ajax/Fetch API.