

Trabajo Fin de Grado

Grado en Ingeniería de las Tecnologías de
Telecomunicación

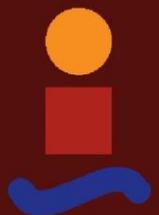
Plataforma web de prescripción de ejercicios
usando Spring

Autor: Pablo Carmona Rebollo

Tutor: María Teresa Ariza Gómez

**Departamento de Ingeniería Telemática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla**

Sevilla, 2018



Trabajo Fin de Grado
Grado en Ingeniería de las Tecnologías de Telecomunicación

Plataforma web de prescripción de ejercicios usando Spring

Autor:

Pablo Carmona Rebollo

Tutor:

María Teresa Ariza Gómez

Profesor titular

Departamento de Ingeniería Telemática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2018

Trabajo Fin de Grado: Plataforma web de prescripción de ejercicios usando Spring

Autor: Pablo Carmona Rebollo

Tutor: María Teresa Ariza Gómez

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2018

El Secretario del Tribunal

A mi familia

A mis maestros

A mis amigos

Resumen

Este proyecto consiste en una aplicación Web de creación de rutinas de ejercicios de entrenamiento personalizado llamada MyTraining.

Esta aplicación da la oportunidad al usuario de seguir una rutina de ejercicios dada por un especialista sin tener que desplazarse a un centro especializado en ello con las molestias que eso conlleva. Es una forma de favorecer un estilo de vida saludable de una forma práctica y sencilla sin tener que moverse de casa.

El usuario dispondrá de toda la información necesaria para realizar dichas rutinas, así como de videos explicativos alojados en Youtube y visibles desde la propia aplicación acerca de la realización de los ejercicios.

Resumen	9
Índice	11
Índice de Figuras	13
1 Introducción	17
1.1 <i>Motivación y Objetivos</i>	17
1.2 <i>Antecedentes</i>	17
1.3 <i>Descripción de la Solución</i>	21
1.3.1 <i>Objetivos específicos</i>	21
1.3.2 <i>Arquitectura</i>	21
2 Tecnologías utilizadas	23
2.1 <i>Servicio Web REST</i>	23
2.1.1 <i>Arquitectura REST</i>	23
2.2 <i>Formato para el intercambio de datos (JSON)</i>	23
2.2.1 <i>Introducción</i>	23
2.2.2 <i>Otros formatos (XML)</i>	24
2.3 <i>Tecnologías en el Servidor: Java Spring Framework</i>	24
2.3.1 <i>Introducción: Spring</i>	24
2.3.2 <i>Spring Boot</i>	25
2.3.3 <i>Spring Security</i>	25
2.3.4 <i>Spring Data</i>	26
2.3.4.1 <i>Patrón DAO (Data Access Object)</i>	26
2.3.4.2 <i>JDBC (Java Database Connectivity)</i>	26
2.3.5 <i>Spring Beans</i>	27
2.3.6 <i>Construcción y Gestión de Software (Maven)</i>	27
2.4 <i>Persistencia: Base de datos</i>	28
2.4.1 <i>Introducción: Modelo Relacional</i>	28
2.4.2 <i>Gestor MySQL y uso en nuestro proyecto</i>	28
2.5 <i>Tecnologías para la interfaz web de usuario</i>	29
2.5.1 <i>Patrón Modelo-Vista-Controlador (MVC)</i>	29
2.5.2 <i>Bootstrap (HTML5 + CSS3 + JavaScript) / Youtube</i>	30
2.5.3 <i>Controlador de cliente: JQuery/Ajax</i>	32
3 Herramientas Utilizadas	35
3.1 <i>Spring Tool Suite (STS)</i>	35
3.2 <i>XAMPP</i>	36
3.3 <i>Google Chrome</i>	37
3.4 <i>EditThisCookie</i>	38
3.5 <i>Wireshark</i>	39
3.6 <i>Winpcap</i>	39

3.7	Mobile browser emulator	40
3.8	Account kit de Facebook	41
3.9	Logo de Youtube	41
4	Servicio web	43
4.1	<i>Clases</i>	43
4.2	<i>Diagramas de clases</i>	44
4.3	<i>Interacción entre clases</i>	46
4.4	<i>URIs</i>	50
	<i>A.1 Recurso Usuario</i>	50
	<i>A.2 Recurso Rutina</i>	50
	<i>A.3 Recurso Ejercicio</i>	52
	<i>A.4 Recurso Vídeo</i>	54
4.5	<i>Youtube</i>	54
4.6	<i>Spring Security</i>	56
4.7	<i>Diagrama ER de la BD</i>	60
	4.7.1 Detalle de los elementos del diagrama EERR	60
5	Interfaz de usuario y funcionalidad	65
5.1	<i>Introducción</i>	65
5.2	<i>Autenticación o Inicio de Sesión</i>	65
5.3	<i>Vistas y funciones de la web</i>	68
6	Conclusiones	87
	Referencias	89
	Anexo A: Manual de Instalacion y Despliegue de la aplicacion	91
	<i>A.1 Instalación de Java JDK 8</i>	91
	<i>A.2 MAVEN</i>	92
	<i>A.3 Creación de variables de entorno</i>	92
	<i>A.4 Comprobación en cmd e instalación</i>	93
	<i>A.5 Instalación de Spring Tools Suite (STS)</i>	93
	<i>A.6 Instalación de XAMPP</i>	94
	<i>A.7 Despliegue de la aplicación</i>	96
	<i>A.8 Account kit de Facebook</i>	97

ÍNDICE DE FIGURAS

Figura 1.1: Aplicación Móvil Rutina App	18
Figura 1.2: Aplicación Web de RutinaApp.	18
Figura 1.3: Aplicación José	19
Figura 1.4: Aplicación Antonio	20
Figura 1.5: Aplicación Juan	20
Figura 1.6: Arquitectura de la aplicación Web	22
Figura 2.1: Logo HTTP REST.	23
Figura 2.2: Logo de JSON.	24
Figura 2.3: Logo de XML	24
Figura 2.4: Logo de Spring Framework	24
Figura 2.5: Módulos de Spring Usados en este proyecto.	25
Figura 2.6: Logo de Spring Boot.	25
Figura 2.7: Logo de Spring Security.	25
Figura 2.8: Logo de Spring Data	26
Figura 2.9: Logo de JDBC	26
Figura 2.10: Logo de Java Beans.	27
Figura 2.11: Logo de Maven.	27
Figura 2.11.1: pom.xml	28
Figura 2.12: Logo de MySQL.	29
Figura 2.13: Esquema del Modelo MVC.	30
Figura 2.14: Logo de BootStrap.	30
Figura 2.15: Logo de HTML5.	31
Figura 2.16: Logo de CSS3	31
Figura 2.17: Logo de JavaScript.	31
Figura 2.18: Logo de DOM.	32
Figura 2.19: Logo de JQuery.	32
Figura 2.20: Logo de AJAX.	33
Figura 2.21: Logo de JQuery AJAX.	33
Figura 2.22: Ejemplo de sintaxis de una consulta simple en AJAX.	33
Figura 2.23: Logo de JQuery Plugin Validation.	34
Figura 3.1: Logo de Spring Tool Suite.	35
Figura 3.2: Interfaz Gráfica de STS.	36

Figura 3.3: Logo de XAMPP.	37
Figura 3.4: Logo de phpMyAdmin.	37
Figura 3.5: Logo de Google Chrome.	37
Figura 3.6: Logo de EditThisCookie.	38
Figura 3.7: Interfaz Gráfica de EditThisCookie.	38
Figura 3.8: Wireshark.	39
Figura 3.9: Winpcap.	39
Figura 3.10: Mobile browser emulator	40
Figura 3.11: MyTraining en móvil	40
Figura 3.12: Account kit Facebook	41
Figura 3.13: Logo de Youtube	41
Figura 4.1: Diagrama de usuario	44
Figura 4.2: Diagrama rutina	44
Figura 4.3: Diagrama ejercicio	45
Figura 4.4: Diagrama video	45
Figura 4.5: SqlConstants.java	46
Figura 4.6: UriConstants.java	46
Figura 4.7: Rutina.java	47
Figura 4.8: RutinaDao.java	47
Figura 4.9: RutinaDaoImpl.java	48
Figura 4.10: RutinaController.java	48
Figura 4.11: RutinaRowMapper.java	49
Figura 4.12: Petición AJAX	49
Figura 4.13: Código de Youtube	54
Figura 4.14: Código de Youtube 2	55
Figura 4.15: Diagrama de acción de Spring Security.	56
Figura 4.16: SQL Authentication	57
Figura 4.17: Authentication Method	57
Figura 4.18: Filter Method	58
Figura 4.19: RESTAuthenticattionSuccessHandler	58
Figura 4.20: RESTAuthenticattionFailureHandler	59
Figura 4.21: RESTAuthenticattionEntryPoint	59
Figura 4.22: Diagrama ER	60
Figura 5.1: Página de Inicio de Sesión	65
Figura 5.2: Página SMS.	66
Figura 5.3: Verificación teléfono.	66

Figura 5.4: Código SMS	67
Figura 5.5: Mensaje código	67
Figura 5.6: Mensaje de Aviso de Credenciales Introducidas Correctamente	67
Figura 5.7: Página principal	68
Figura 5.8: Página principal con menú lateral desplegado de Usuario Especialista.	69
Figura 5.9: Pestaña ‘Crear usuario’.	70
Figura 5.10: Página ‘Crear usuario’	70
Figura 5.11: Usuario creado	70
Figura 5.12: Pestaña ‘Rutinas Disponibles’	71
Figura 5.13: Página ‘Rutinas Creadas’	71
Figura 5.14: Página ‘Rutinas Creadas’ con búsqueda	72
Figura 5.15: Página ‘Añadir rutina’	72
Figura 5.15.1: Lista usuarios	73
Figura 5.16: Rutina añadida	73
Figura 5.17: Rutinas de otros usuarios	73
Figura 5.18: Página Rutinas de otros usuarios.	74
Figura 5.19: Pestaña Modificar Rutina	74
Figura 5.20: Pestaña Ejercicios Disponibles	75
Figura 5.21: Página Ejercicios Creados	75
Figura 5.22: Página Añadir Ejercicio	76
Figura 5.23: Ejercicios de otros usuarios	76
Figura 5.24: Página Ejercicios de otros usuarios	77
Figura 5.25: Página Modificar Ejercicio	77
Figura 5.26: Eliminar Ejercicio	78
Figura 5.27: Página Añadir video	78
Figura 5.28: Página Ver video	78
Figura 5.29: Eliminar video	79
Figura 5.30: Página Añadir Ejercicio a Rutina	79
Figura 5.31: Ejercicio asociado	79
Figura 5.32: Ejercicio sin Video.	80
Figura 5.33: Página Ver Ejercicios de Rutina	80
Figura 5.34: Eliminar rutina 1	80
Figura 5.35: Eliminar rutina 2	81
Figura 5.36: Descargar rutina	81

Figura 5.37: JSON rutina	81
Figura 5.38: Pestaña Ajustes	81
Figura 5.39: Pestaña Ajustes 2.	81
Figura 5.40: Página Datos de Usuario	82
Figura 5.41: Página Darse de Baja	82
Figura 5.42: Perfil borrado	82
Figura 5.43: Página Ayuda	83
Figura 5.44: Correo.	83
Figura 5.45: Cerrar sesión	83
Figura 5.46: Menú usuario	84
Figura 5.47: Rutinas de usuario	84
Figura 5.48: Ejercicios de rutinas de usuario	85
Figura 5.49: Ejercicios de usuario	85
Figura A.1: JDK	91
Figura A.2: MAVEN	92
Figura A.3: Variables de entorno	92
Figura A.4: versión Maven/java	93
Figura A.5: Descarga STS	93
Figura A.6: Inicio STS	93
Figura A.7: Workspace	94
Figura A.8: Descarga de XAMPP	94
Figura A.9: Interfaz XAMPP	95
Figura A.10: phpMyAdmin XAMPP	95
Figura A.11: Importar BD	96
Figura A.12: Importar proyecto	96
Figura A.13: Root Directory	97
Figura A.14: Proyecto importado	97
Figura A.15: Account kit	98
Figura A.16: Account kit 2	98
Figura A.17: Account kit 3	99

1 INTRODUCCIÓN

“Cualquier tecnología suficientemente avanzada es equivalente a la magia.”

Arthur C. Clarke

Actualmente, el desarrollo software se encuentra en auge. Los servicios Web son la revolución informática de la nueva generación de aplicaciones que trabajan colaborativamente y en las cuales el software está distribuido en diferentes servidores. Los principales objetivos de estos son la disponibilidad, usabilidad y funcionalidad de cualquier aplicación a través de internet.

Entre otros aspectos de los servicios web, destaca la interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen, fomenta los estándares y protocolos basados en texto y permite proveer servicios integrados en cualquier parte del mundo.

1.1 Motivación y Objetivos

El principal objetivo de este proyecto es crear una aplicación Web REST llamada MyTraining empleando para ello el framework Spring. Dicha aplicación pretende proporcionar una forma cómoda de favorecer un estilo de vida saludable desde casa mediante rutinas de ejercicios de entrenamiento personalizadas dadas por un especialista.

MyTraining permitirá tener dos tipos de usuarios, el usuario ‘paciente’ y el usuario ‘especialista’. Los ‘especialistas’ serán los encargados de la gestión de rutinas, ejercicios y ‘pacientes’, mientras que el resto serán atendidos por ellos recibiendo rutinas de entrenamiento personalizadas con videos explicativos y asesoramiento si lo requieren a través de un correo electrónico.

Mi principal motivación a la hora de realizar este proyecto fue la oportunidad de aprender acerca de diversas tecnologías y herramientas que desconocía hasta entonces y ampliar mis conocimientos.

1.2 Antecedentes

Como antecedentes principales a nuestro proyecto, debemos destacar principalmente la siguiente línea de trabajo:

Aplicación Android de rutinas de entrenamiento adaptadas al usuario usando SQLite y JSON

Este es un proyecto desarrollado en la Escuela Técnica Superior de Ingeniería de Sevilla por la alumna Mirian Franco Maireles en el año 2015.

El objetivo principal de este proyecto es el desarrollo de una aplicación móvil en Android que permita al usuario tener sus propias rutinas de entrenamiento en el lugar y el momento que lo desee, con la mayor calidad y exactitud posible, y con la ausencia de un profesional especializado, tales como un entrenador personal o fisioterapeuta.

El funcionamiento de dicha aplicación está totalmente desvinculado a la existencia de una red de datos, lo cual no permite la gestión de los recursos que maneja dicha aplicación a través de Internet mediante alguna plataforma o servicio Web, dejándose esta como línea de desarrollo.

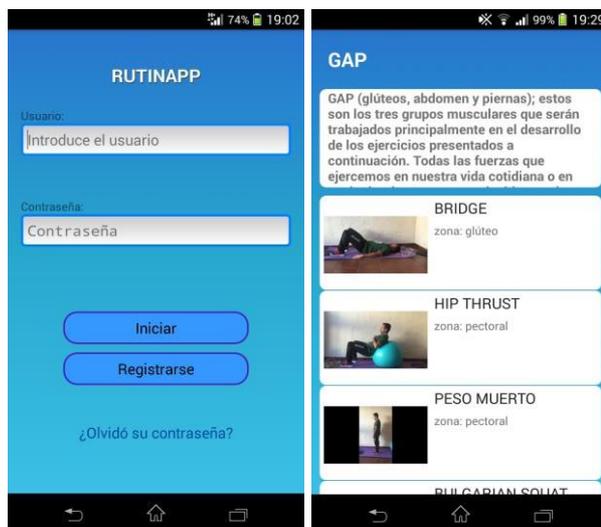


Figura 1.1: Aplicación Móvil Rutina App

Plataforma Web para la creación de rutinas de entrenamiento para la App RutinaApp

Este es un proyecto desarrollado en la Escuela Técnica Superior de Ingeniería de Sevilla por el alumno Francisco José Díaz Romero en el año 2017.

En este proyecto se desarrolló una aplicación Web que permite gestionar rutinas de auto-entrenamiento, cuya finalidad es la creación de dichas rutinas y su personalización por el mismo usuario que las crea, de la forma que crea oportuna, para su posterior uso y reproducción en una aplicación Móvil desarrollada en Android, llamada Rutina App.

Dicho proyecto y las tecnologías utilizadas en él han sido utilizados como base para alcanzar el objetivo deseado.

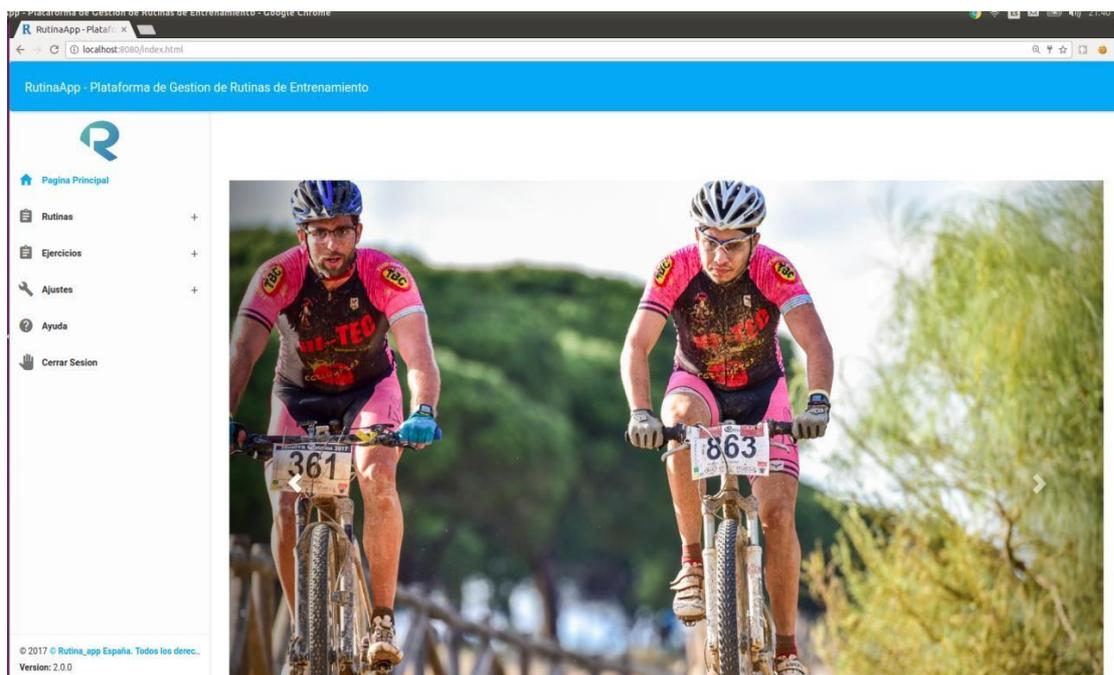


Figura 1.2: Aplicación Web RutinaApp

Desarrollo de Servicio Web RESTful para el Acceso a Base de Datos de Entrenamiento desde Android

Este es un proyecto desarrollado en la Escuela Técnica Superior de Ingeniería de Sevilla por el alumno José María Valverde Baena en el año 2015.

Esta App facilita a los usuarios la posibilidad de realizar un entrenamiento remoto en el momento y lugar que deseen de manera autónoma, sin necesidad de tener la asistencia de un profesional en el momento de uso, debido a que los ejercicios han sido previamente asignados por un médico o fisioterapeuta.



Figura 1.3: Aplicación José

Aplicación Android para la rehabilitación física de usuario usando MySQL, PHP y codificación JSON

Este es un proyecto desarrollado en la Escuela Técnica Superior de Ingeniería de Sevilla por el alumno Antonio José Díaz Lora en el año 2014.

Dicho proyecto, orientado a la rehabilitación física de los usuarios a través de ejercicios de entrenamiento, realizaba una aplicación Android de carácter deportivo que se ponía en contacto con un servidor web donde había alojados unos scripts PHP. Éstos, dotaban de funcionalidad al sistema, pues eran los que realizaban las labores de comunicación con la base de datos.



Figura 1.4: Aplicación Antonio

Personal Trainer, entrenador personal para Android

Este es un proyecto desarrollado en la Escuela Técnica Superior de Ingeniería de Sevilla por el alumno Juan García Piosa en el año 2014.

La aplicación Personal Trainer facilita la práctica de ejercicio físico controlado y personalizado por un entrenador. En dicho proyecto se ha utilizado como uno de los lenguajes principales PHP. Lo único que se requiere es un terminal móvil con sistema operativo android y una conexión a internet.

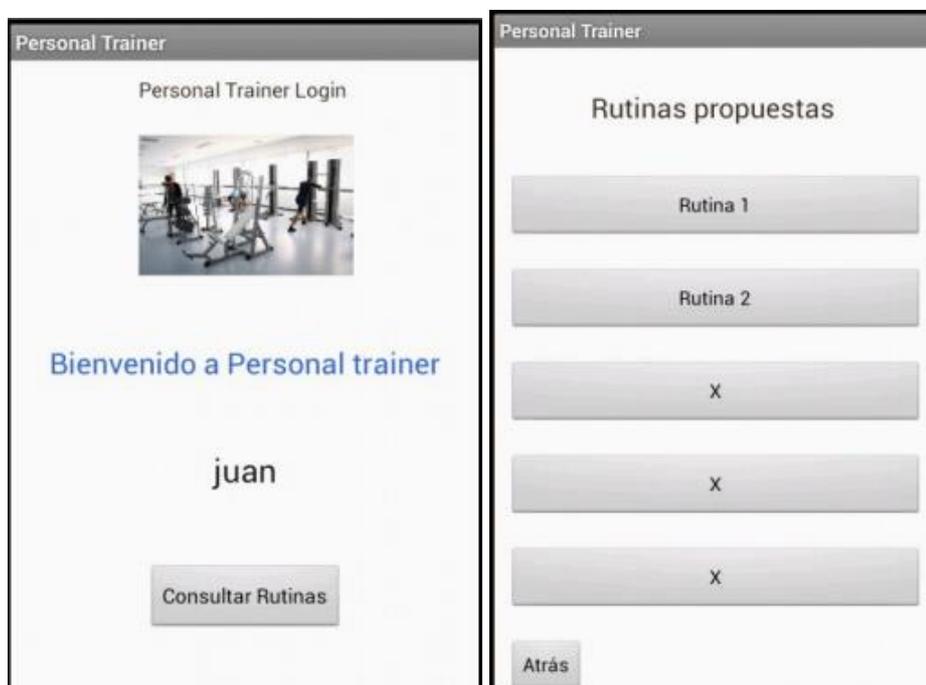


Figura 1.5: Aplicación Juan

1.3 Descripción de la Solución

1.3.1 Objetivos específicos

- Interfaz de usuario amigable e intuitiva.
- Diferenciar dos tipos de usuarios: Especialistas y “paciente”.
- Implementar autenticación Email/contraseña y por SMS.
- Los Especialistas se encargarán de dar de alta en la aplicación a los “pacientes”.
- Los Especialistas serán los encargados de crear, modificar, asociar y eliminar las rutinas, ejercicios y videos relacionados con el entrenamiento individual de cada “paciente”.
- Buscar solución al almacenamiento físico de videos en el servidor.
- Cada “paciente” tendrá visibles solo aquellas rutinas y ejercicios que el Especialista le haya asignado.
- Los Especialistas tendrán dos vistas a la hora de las rutinas y ejercicios:
 - Las propias
 - Las del resto de Especialistas
- Los Especialistas podrán añadir ejercicios a cualquier rutina, sea o no el creador.
- Las rutinas podrán descargarse en formato JSON para un posible uso en el futuro en la app móvil homóloga a esta aplicación web.

1.3.2 Arquitectura

Nuestra aplicación web basada en un servicio web REST tiene la siguiente arquitectura:

- ❖ **Navegador Web:** Esta parte de la aplicación es la denominada como *Front-end*, que es la parte del software que interactúa con los usuarios. Consta de:
 - **Modelo Web:** Interfaz de la aplicación con la que interactúan los usuarios. Está diseñada para ser intuitiva y amigable. Las tecnologías utilizadas para su implementación han sido *HTML5* y *CSS3*.
 - **Controlador Web:** Controla las interacciones de los usuarios con la interfaz a través de Javascript/Jquery.
- ❖ **Servicio Web REST:** Esta es la parte del servidor de la aplicación denominada como *Back-end*, que procesa la entrada desde el *Front-end*. Dicho servidor está implementado mediante el framework *Spring*, empleando como lenguaje principal *Java*.
- ❖ **Base de Datos:** Encargada de la permanencia de la información referente a la aplicación web e implementada mediante *MySql*

El componente ‘Navegador Web’ se comunica con el servidor a través de llamadas a las URIs en las que se encuentran alojados los recursos de este. Dichas llamadas las realiza a través de *JQUERY* mediante *AJAX*.

El componente ‘Servicio Web REST’ se encarga de capturar las llamadas del ‘Navegador Web’ mediante *Java* y de realizar las peticiones correspondientes a través de *SQL* y *HTTP*, empleando los métodos CRUD (**create, read, update, and delete**), a la Base de Datos con ayuda de *Java* y *JDBC*.

El intercambio de información entre el ‘Navegador Web’ y el servidor se realiza en formato *JSON*, y la Base de Datos devuelve la información en un objeto *Java*.

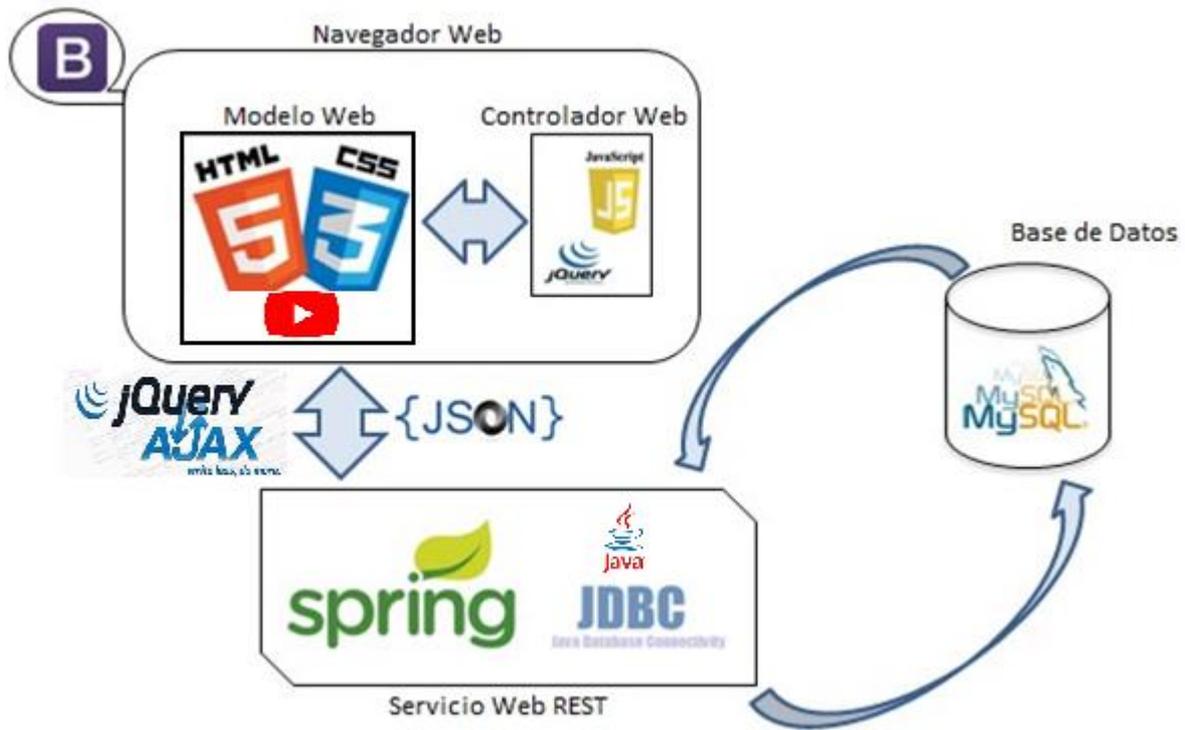


Figura 1.6: Arquitectura de la aplicación Web.

2 TECNOLOGÍAS UTILIZADAS

2.1 Servicio Web REST

2.1.1 Arquitectura REST

La arquitectura REST o Transferencia de Estado Representacional es un estilo de arquitectura introducido y definido en el año 2000 por Roy Fielding en su tesis doctoral. Fielding es uno de los principales autores del protocolo de transferencia de hipertexto (HTTP).

Esta arquitectura hace énfasis en que las interacciones entre los clientes y los servicios se mejoran al tener un número limitado de operaciones (verbos). La flexibilidad se obtiene asignando recursos a sus propios identificadores de recursos universales únicos (URI). Debido a que cada verbo tiene un significado específico (GET, POST, PUT y DELETE), REST evita la ambigüedad:

- ✓ GET: Obtener un recurso.
- ✓ POST: Crear un recurso en el servidor.
- ✓ PUT: Cambiar el estado de un recurso o actualizarlo.
- ✓ DELETE: Eliminar un recurso.

El uso de REST es preferible a la arquitectura SOAP (Simple Object Access Protocol) que en comparación es más pesado, porque REST no aprovecha tanto ancho de banda, lo que hace que sea un mejor ajuste para su uso a través de Internet.



Figura 2.1: Logo HTTP REST.

2.2 Formato para el intercambio de datos (JSON)

2.2.1 Introducción

Cuando intercambiamos datos entre el navegador Web del cliente y el servidor necesitamos gestionar de forma sencilla y eficaz mucha información. JSON es un estándar para el intercambio de datos siguiendo una sintaxis específica y definida de forma que podemos intercambiar información entre diferentes aplicaciones independientemente de la plataforma o lenguaje en el que hayan sido desarrolladas. JSON nació como una alternativa a XML, debido a su gran facilidad de uso con JavaScript.

Se destaca principalmente porque este formato puede ser leído por cualquier lenguaje de programación, lo cual nos proporciona mayor integridad y flexibilidad.

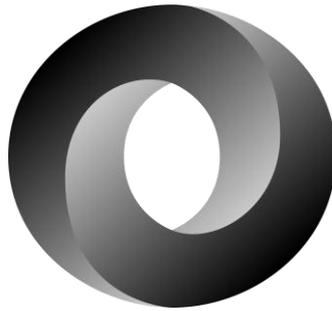


Figura 2.2: Logo de JSON.

2.2.2 Otros formatos (XML)

XML ,usado para algunos ficheros de configuración de Spring, es un lenguaje de marcado que define un conjunto de reglas para codificar información de forma que sea legible por un ser humano o por un ordenador. Gracias a la flexibilidad que proporciona, carece de las limitaciones que tiene HTML y puede ser usada para representar cualquier estructura de datos. Como ventaja principal de este lenguaje, cabe destacar que proporciona soporte Unicode, lo cual permite escribir la información en cualquier idioma del mundo.



Figura 2.3: Logo de XML

2.3 Tecnologías en el Servidor: Java Spring Framework

2.3.1 Introducción: Spring

Este es un framework que se usa para el desarrollo de Servicios y aplicaciones, y como contenedor de inversión de control de código abierto para la plataforma Java.

Spring fue escrito para la plataforma J2EE de Java, plataforma orientada al desarrollo de aplicaciones web, y ha ido evolucionando rápidamente hasta el día de hoy, donde podemos encontrar diferentes ramas de desarrollo de la mano de SpringSource y todo su equipo de desarrolladores.



Figura 2.4: Logo de Spring Framework

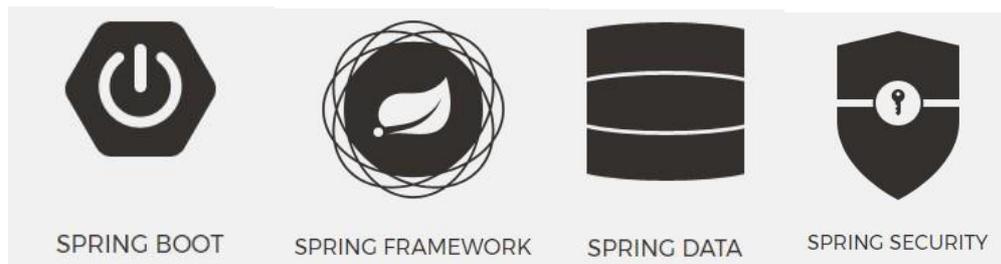


Figura 2.5: Módulos de Spring Usados en este proyecto.

2.3.2 Spring Boot

Spring Boot es un módulo de Spring que nos permite crear diferentes tipos de aplicaciones autónomas de una manera rápida y sencilla, donde simplemente hay que ejecutarlas, simplificando la creación de dependencias y el despliegue en un servidor de aplicaciones.

Principalmente se trata de un mecanismo centrado en una clase principal, la cual tiene una configuración por defecto y que podemos personalizar. Este es compatible con Maven y Gradle, y su formato por defecto para el despliegue es “.JAR”, aunque también soporta “.WAR”.



Figura 2.6: Logo de Spring Boot.

2.3.3 Spring Security

Spring Security es un módulo que permitirá gestionar todo lo relativo a la seguridad de nuestra aplicación web, desde el protocolo de seguridad, hasta los roles que necesitan los usuarios para acceder a los diferentes recursos de la aplicación. Desde el punto de vista de la autenticación, nos permite comprobar que un usuario exista y permitir el acceso a los recursos de la aplicación, dando autorización a unos recursos u otros según sus credenciales y sus privilegios.

Este módulo consta de una serie de elementos, como clases y parámetros en Java, los cuales hay que personalizar para que se adapte de la mejor manera posible al comportamiento y finalidad que queremos llegar a conseguir en nuestra aplicación Web.



Figura 2.7: Logo de Spring Security.

2.3.4 Spring Data

Spring Framework ya proporcionaba soporte para el acceso a bases de datos como JDBC, Hibernate o JPA, simplificando la implementación de la capa de acceso a datos, unificando la configuración y creando una jerarquía de excepciones común para todas ellas.

Spring Data permite cubrir el soporte necesario para todas las tecnologías de persistencia y además, integra las tecnologías de acceso tradicionales, simplificando el trabajo a la hora de crear las implementaciones concretas.

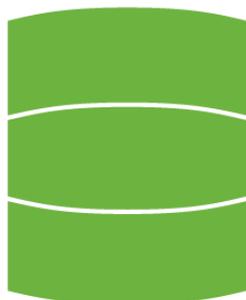


Figura 2.8: Logo de Spring Data

2.3.4.1 Patrón DAO (Data Access Object)

Este patrón de diseño software pertenece al catálogo de Core J2EE Patterns de Java, por lo que no es un patrón exclusivo de Spring, aunque es el utilizado para realizar aplicaciones con este framework que hagan uso de la persistencia.

El patrón DAO muestra una forma de envolver el conocimiento sobre el acceso a datos, mediante el uso de objetos de acceso a datos para conseguir abstraer y encapsular el acceso a dichos datos.

Un objeto DAO permite obtener y guardar datos, manejando la conexión con el sistema de persistencia que se haya implementado. Todo ello lo realiza mediante operaciones atómicas (Creación, actualización, borrado, obtención de registros) en la base de datos, por lo que nunca son necesarias las transacciones. Normalmente, se crea un objeto DAO por cada Objeto que tengamos en nuestra aplicación.

2.3.4.2 JDBC (Java Database Connectivity)

JDBC es una API que describe o define una librería estándar para el acceso a Fuentes de datos desde el lenguaje de programación Java, independientemente del Sistema operativo donde se ejecute o de la base de datos a la que se acceda, principalmente orientado a bases de datos relacionales.



Figura 2.9: Logo de JDBC

Esta API consiste en un conjunto de interfaces y clases escritas en Java. Con estas interfaces y clases estándar, los programadores pueden escribir aplicaciones que conecten con la base de datos, enviar consultas escritas en el lenguaje de consulta estructurada “SQL” y posteriormente procesar los resultados obtenidos.

2.3.5 Spring Beans

Un bean se define como un componente de software que tiene la particularidad de ser reutilizable. Estos deben cumplir ciertos criterios en java:

- Implementación serializable.
- Poseer todos sus atributos privados (private).
- Poseer métodos “set” y “get” públicos de los atributos privados que nos interese.
- Poseer un constructor público por defecto.

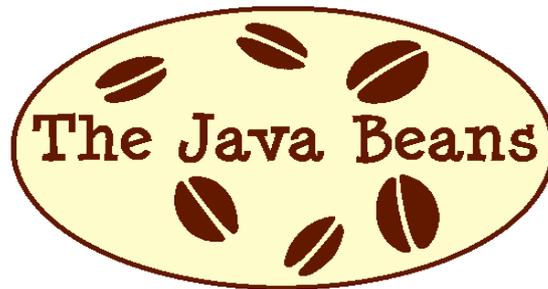


Figura 2.10: Logo de Java Beans.

A diferencia de los beans convencionales que representan una clase, la particularidad de los beans en Spring es que son objetos creados y manejados por el contenedor de Spring.

2.3.6 Construcción y Gestión de Software (Maven)

Maven utiliza un Project Object Model (POM) para describir el proyecto de software a construir, sus dependencias de otros módulos y componentes externos, y el orden de construcción de los elementos. Viene con objetivos predefinidos para realizar ciertas tareas claramente definidas, como la compilación del código y su empaquetado.

Una característica clave de Maven es que está listo para usar en red. El motor incluido en su núcleo puede dinámicamente descargar plugins de un repositorio, el mismo repositorio que provee acceso a muchas versiones de diferentes proyectos Open Source en Java, de Apache y otras organizaciones y desarrolladores. Maven provee soporte no solo para obtener archivos de su repositorio, sino también para subir artefactos al repositorio al final de la construcción de la aplicación, dejándola al acceso de todos los usuarios.

Maven está construido usando una arquitectura basada en plugins que permite que utilice cualquier aplicación controlable a través de la entrada estándar. En teoría, esto podría permitir a cualquiera escribir plugins para su interfaz con herramientas como compiladores, herramientas de pruebas unitarias, etcétera, para cualquier otro lenguaje.



Figura 2.11: Logo de Maven.

Todo ello se hará de forma automática sin que el usuario tenga que hacer nada más que definir las dependencias en el formato que mostramos a continuación:

```

<?xml version="1.0" encoding="UTF-8"?>
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>

  <groupId>tfg</groupId>
  <artifactId>TFG</artifactId>
  <version>0.0.1-SNAPSHOT</version>

  <name>TFG</name>
  <description>tfg project for Spring Boot</description>

  <parent>
    <groupId>org.springframework.boot</groupId>
    <artifactId>spring-boot-starter-parent</artifactId>
    <version>1.5.10.RELEASE</version>
    <relativePath/> <!-- lookup parent from repository -->
  </parent>

  <properties>
    <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
    <project.reporting.outputEncoding>UTF-8</project.reporting.outputEncoding>
    <java.version>1.8</java.version>
  </properties>

  <dependencies>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-web-services</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-logging</artifactId>
    </dependency>
    <dependency>
      <groupId>org.springframework.boot</groupId>
      <artifactId>spring-boot-starter-tomcat</artifactId>
      <scope>provided</scope>
    </dependency>
  </dependencies>

```

Figura 2.11.1: pom.xml

2.4 Persistencia: Base de datos

2.4.1 Introducción: Modelo Relacional

El modelo relacional es una alternativa para la organización y representación de la información que se pretende almacenar en una base de datos. Se trata de un modelo teórico matemático que, además de proporcionarnos los elementos básicos de modelado (las relaciones), incluye un conjunto de operadores (definidos en forma de un álgebra relacional) para su manipulación, sin ambigüedad posible.

Este modelo, hace relativamente sencilla su representación y gestión por medio de herramientas informáticas, siendo elegido como referencia para la gran mayoría de los sistemas de Gestión de Bases de Datos disponibles en el mercado. Además, también es seleccionado como referencia para la elaboración del esquema lógico de una base de datos en su diseño, junto con herramientas como el álgebra relacional.

2.4.2 Gestor MySQL

Es un sistema de gestión de bases de datos relacional desarrollado por Oracle Corporation, considerado como el más popular del mundo, además de ser desarrollado y distribuido libremente (Open Source).

MySQL es multiplataforma, lo cual da soporte a una amplia lista de sistemas operativos y trae soporte para aproximadamente diez motores de almacenamiento, en los cuales se encuentran InnoDB que es el motor predeterminado, que soporta transacciones y bloqueo de registros. Esto hace de este gestor que posea un alto rendimiento comparado con sistemas similares.



Figura 2.12: Logo de MySQL.

Por último, este sistema de gestión de bases de datos provee de varias herramientas de gestión y de diseño de bases de datos de forma interactiva y gráfica, como son XAMPP y MySQLWorkbench. Estas herramientas nos han facilitado el diseño y montaje de la base de datos, además de su gestión.

2.5 Tecnologías para la interfaz web de usuario

2.5.1 Patrón Modelo-Vista-Controlador (MVC)

El objetivo principal de este patrón es el de realizar una separación de los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello, este patrón define tres componentes o módulos básicos:

- **Modelo:** el modelo representa la parte de la aplicación que implementa la lógica de negocio. Esto significa que es responsable de la recuperación de datos convirtiéndolos en conceptos significativos para la aplicación, así como su procesamiento, validación, asociación y cualquier otra tarea relativa a la manipulación de dichos datos.
- **Vista:** es una presentación de los datos del modelo al usuario, estando separada de los objetos del modelo. Es responsable del uso de la información de la cual se dispone para producir cualquier interfaz de presentación de cualquier petición que se presente.
- **Controlador:** gestiona las peticiones de los usuarios. Es responsable de responder a la información solicitada con la ayuda tanto del modelo como de la vista.

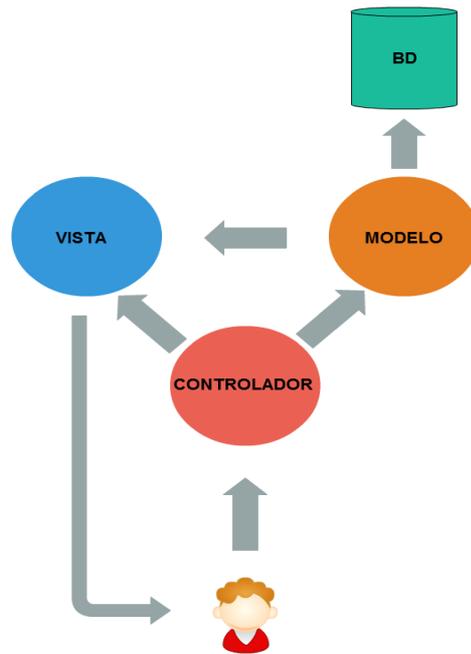


Figura 2.13: Esquema del Modelo MVC.

2.5.2 Bootstrap (HTML5 + CSS3 + JavaScript)

Este es un framework o conjunto de herramientas de código abierto para el diseño de sitios y aplicaciones Web. Contiene plantillas de diseño de tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basados en HTML y CSS, así como extensiones de JavaScript adicionales. Además es compatible con la mayoría de los navegadores web.



Figura 2.14: Logo de Bootstrap.

2.5.2.1 HTML5

HTML es un lenguaje de marcado utilizado para definir la estructura y contenido de una página o documento Web. La idea es utilizar un lenguaje para hacer referencia a otros documentos, como archivos, imágenes, vídeos, audio, etc.

Es por tanto que HTML5 se trata de una nueva versión de HTML en la cual se especifican nuevos elementos, atributos y comportamientos. Además, contiene un conjunto más amplio de tecnologías que permite a los sitios Web y a las aplicaciones ser más diversas y de gran alcance.



Figura 2.15: Logo de HTML5.

2.5.2.2 CSS3

CSS es un lenguaje usado para definir el estilo o apariencia de las páginas Web, escritas con HTML o de los documentos XML. Este lenguaje se creó para separar el contenido de la forma, a la vez que permite a los diseñadores mantener un control mucho más preciso sobre la apariencia de las páginas.



Figura 2.17: Logo de CSS3.

2.5.2.3 JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Este estándar es también un lenguaje de programación interpretado, definido como orientado a objetos y basado en prototipos, imperativo y dinámico. Se utiliza principalmente en su forma del lado cliente, implementado como una parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.



Figura 2.18: Logo de JavaScript.

Para interactuar con una página Web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

El modelo DOM es esencialmente una interfaz de plataforma que proporciona un conjunto estándar de objetos para representar documentos HTML, XHTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos. A través del DOM, los programas pueden acceder y modificar el contenido, estructura y estilo de los documentos HTML y XML, que es para lo que se diseñó principalmente.



Figura 2.19: Logo de DOM.

Tradicionalmente, JavaScript se venía utilizando en páginas Web HTML para realizar operaciones y únicamente en el marco de la aplicación del cliente, sin acceso a funciones del servidor. Actualmente es ampliamente utilizado para enviar y recibir información del servidor junto con la ayuda de otras tecnologías como AJAX. JavaScript se interpreta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

2.5.3 Controlador de cliente: JQuery/Ajax

2.5.3.1 JQuery

JQuery es la biblioteca ECMAScript más extendida y usada en el mundo. Esta biblioteca de software libre y de código abierto permite simplificar la forma de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción mediante AJAX a páginas Web.



Figura 2.20: Logo de JQuery.

2.5.3.2 AJAX, Asynchronous JavaScript And XML

AJAX, acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas. Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios, mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas completamente, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.



Figura 2.21: Logo de AJAX.

2.5.3.3 JQuery AJAX

JQuery facilita el trabajo con AJAX mediante métodos que simplifican todo el proceso anterior. La más importante es “.ajax()”, que realiza una petición asíncrona y tiene multitud de opciones.



Figura 2.22: Logo de JQuery AJAX.

En nuestro caso, usaremos AJAX a través del método \$.ajax() que es la implementación de dicha tecnología a través de JavaScript, concretamente de la biblioteca JQuery, que es configurado a través de un objeto, el cual contiene todas las instrucciones que necesita JQuery para completar una petición de este tipo. Dicho método es muy útil debido a que ofrece la posibilidad de especificar acciones en caso de que la petición haya fallado o no. Además, al estar configurado a través de un objeto, es posible definir sus propiedades de forma separada, haciendo que sea más fácil la reutilización del código.

```
$.ajax({  
    url: '/ruta/hasta/pagina.php',  
    type: 'POST',  
    async: true, // Valor por defecto, no necesario  
    data: 'parametro1=valor1&parametro2=valor2',  
        // Datos que se envían al servidor  
    success: procesaRespuesta,  
    // función que procesa los datos devueltos por el servidor  
    error: muestraError // función invocada si la petición falla  
});
```

Figura 2.23: Ejemplo de sintaxis de una consulta simple en AJAX.

2.5.3.4 JQuery Validation Plugin

Este es un Plugin de JQuery cuya finalidad es realizar la validación de formularios HTML en el lado del cliente, ofreciendo una gran cantidad de opciones personalizables.



Figura 2.24: Logo de JQuery Plugin Validation.

3 HERRAMIENTAS UTILIZADAS

3.1 Spring Tool Suite (STS)

Spring Tool Suite proporciona un entorno *ready-to-use* para implementar, depurar, ejecutar y desplegar las aplicaciones Spring, incluyendo integraciones para Pivotal tc Server, Pivotal Cloud Foundry, Git, Maven, AspectJ, y viene encima de las últimas versiones de Eclipse.

Este entorno de desarrollo incluye la edición para desarrolladores de Pivotal tc Server, una versión de Apache Tomcat optimizado para Spring. Con su consola Spring Insight, tc Server Developer Edition ofrece una visión en tiempo real gráfica de los parámetros de rendimiento de aplicaciones que permite a los desarrolladores identificar y diagnosticar los problemas desde sus escritorios.

Además, soporta el despliegue de aplicaciones tanto en servidores locales, virtuales y en la nube. Es de libre acceso para el desarrollo y uso en operaciones internas sin límite de tiempo, completamente de código abierto y licenciada bajo los términos de la Licencia Pública Eclipse.



Figura 3.1: Logo de Spring Tool Suite.

Algunas de las características que podemos destacar son las siguientes:

- ✓ **Validaciones para la configuración de Spring:** ofrece un amplio conjunto de validaciones que se están aplicando de forma automática. Esas validaciones indican errores en las configuraciones directamente en el IDE, mucho antes de que sea ejecutada la aplicación. Encontrar problemas y errores de configuración es mucho más fácil.
- ✓ **Soporte Refactoring para su aplicación Spring:** el término refactorización en ingeniería del Software es usado para describir la modificación del código fuente sin alterar el comportamiento del mismo. Es una de las partes más importantes de la ingeniería de software de hoy. El IDE agrega nuevas refactorizaciones para los elementos de Spring (como el cambio de nombre de los beans de Spring, por ejemplo).
- ✓ **Code Assistant:** proporciona contenido de ayuda significativo en todos lados, junto con soluciones rápidas para los errores y problemas comunes.
- ✓ **Soporte AOP (Programación Orientada a Aspectos):** la Spring Tool Suite se integra con las herramientas de Eclipse y proporciona el soporte más completo para AOP disponible hoy en día.

- ✓ **Integrado con Cloud Foundry y Pivotal tc Server:** permite el despliegue de las aplicaciones directamente en Cloud Foundry o una instancia tc Server (incluyendo el soporte para la depuración, creación de la instancia, Spring Insight, servicios y más).

En nuestro proyecto hemos utilizado este framework para construir nuestra aplicación web.

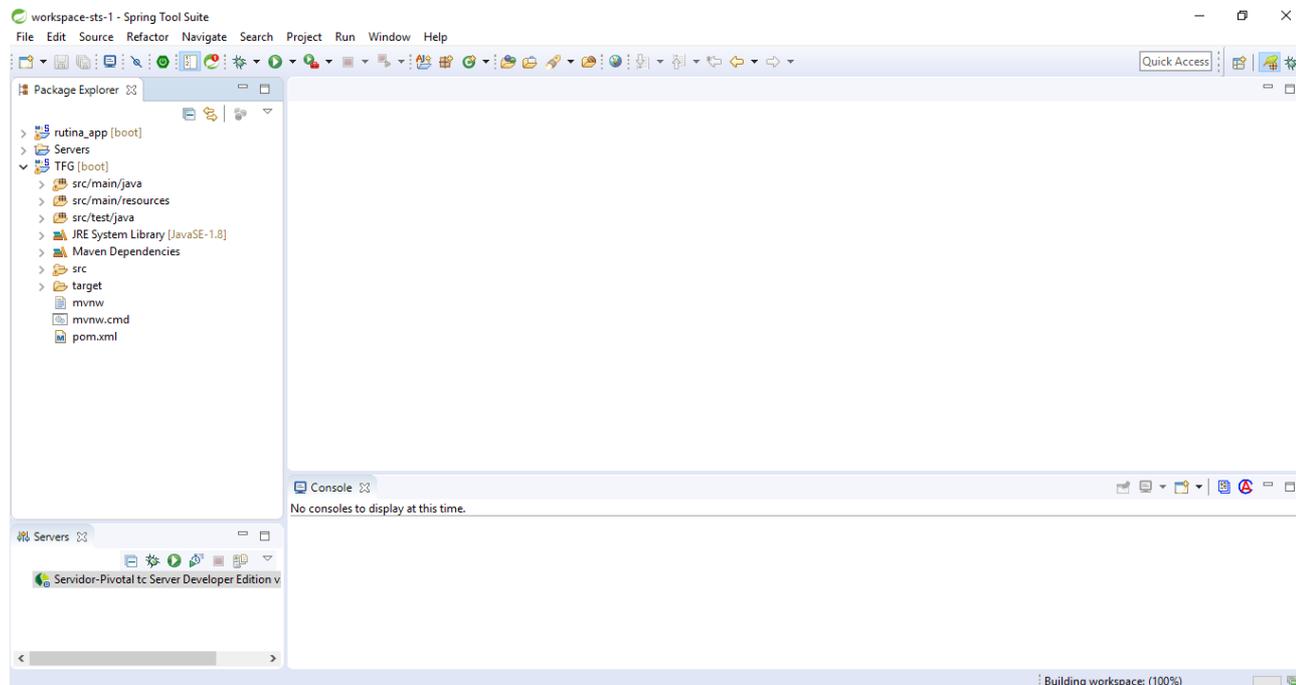


Figura 3.2: Interfaz Gráfica de STS.

3.2 XAMPP

XAMPP es un paquete de instalación independiente de plataforma, software libre, que consiste en el sistema de gestión de bases de datos MySQL, el servidor web Apache y los intérpretes para lenguajes de script: PHP y Perl.

El nombre proviene del acrónimo de X (Multiplataforma), Apache, MariaDB, PHP, Perl. Desde la versión "5.6.15", XAMPP cambió la base de datos de MySQL a MaríaDB. El cual es un fork de MySQL con licencia GPL.

El programa se distribuye bajo la licencia GNU y actúa como un servidor web libre, fácil de usar y capaz de interpretar páginas dinámicas. Actualmente XAMPP está disponible para Microsoft Windows, GNU/Linux, Solaris y Mac OS X.

XAMPP se diseñó para su uso como una herramienta de desarrollo, para permitir a los diseñadores de sitios webs y programadores testear su trabajo en sus propios ordenadores cuando no tienen ningún acceso a Internet. En la práctica, sin embargo, XAMPP se utiliza actualmente como servidor de sitios web, ya que, con algunas modificaciones, es generalmente lo suficientemente seguro para serlo. Con el paquete se incluye una herramienta especial para proteger fácilmente las partes más importantes en una página.



Figura 3.3: Logo de XAMPP.

phpMyAdmin es una herramienta escrita en PHP con la intención de manejar la administración de MySQL a través de páginas web, utilizando Internet. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia SQL, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 72 idiomas. Se encuentra disponible bajo la licencia GPL Versión 2.



Figura 3.4: Logo de phpMyAdmin.

3.3 Google Chrome

Google Chrome es un navegador web de software privativo o código cerrado desarrollado por Google, aunque derivado de proyectos de código abierto (como el motor de renderizado Blink). El nombre del navegador deriva del término en inglés usado para el marco de la interfaz gráfica de usuario («chrome»).

Los lanzamientos de Chrome son obtenidos a partir de Chromium, el proyecto de software libre que también sirve de base para el sistema operativo Chrome OS. En esencia, los aportes hechos por el proyecto libre Chromium fundamentan el código fuente del navegador base sobre el que está construido Chrome y por tanto tendrá sus mismas características, a las cuales Google adiciona otras que no son software libre. También se cambia el nombre y logotipo por otros ligeramente diferentes para proteger la marca comercial de Google.

Actualmente el navegador está disponible para Windows, OS X, Linux, Android y iOS.



Figura 3.5: Logo de Google Chrome.

Chromium es el proyecto de software libre con el que se ha desarrollado Google Chrome y es de participación comunitaria (bajo el ámbito de *Google Code*) para fundamentar las bases del diseño y desarrollo del navegador Chrome (junto con la extensión Chrome Frame), además del sistema operativo Google Chrome OS.

Se ha hecho uso de la herramienta de depuración de código del navegador (a la que se accede pulsando F12) para la inspección y comprobación del buen funcionamiento y la detección de errores del código del cliente.

3.4 EditThisCookie

Edit this cookie es una extensión para Google Chrome con la que podremos ejercer mayor control sobre las *cookies* que genera nuestra navegación.

Con esta extensión podremos **gestionar las cookies** que se generan durante nuestra navegación usando Google Chrome y, por tanto, podremos **añadir, eliminar, borrar o modificar cookies** además de **realizar búsquedas entre las que tengamos, bloquearlas** (evitar que se generen) **o ponerlas en modo “solo lectura”**. Todas estas funciones estarán accesibles desde la esquina superior derecha del interfaz de Chrome, es decir, donde se concentran todos los iconos de las extensiones (justo al lado de la barra de búsquedas, el *Omnibox*).



Figura 3.6: Logo de EditThisCookie.

Esta extensión se ha utilizado para gestionar las cookies de sesión de los usuarios.

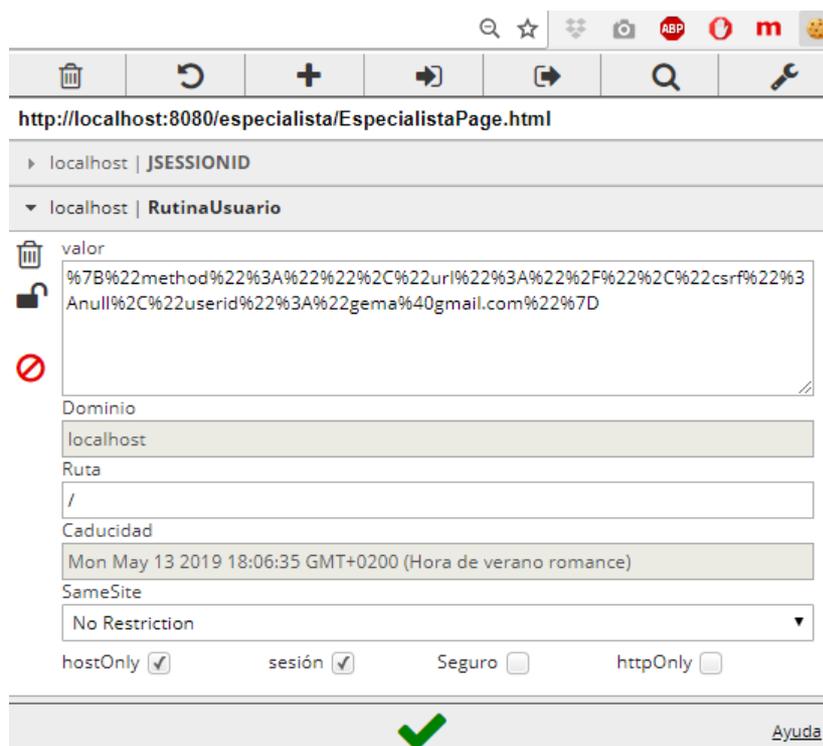


Figura 3.7: Interfaz Gráfica de EditThisCookie.

3.5 Wireshark

Wireshark, antes conocido como Ethereal, es un analizador de protocolos utilizado para realizar análisis y solucionar problemas en redes de comunicaciones.

Añade una interfaz gráfica y muchas opciones de organización y filtrado de información. Así, permite ver todo el tráfico que pasa a través de una red (usualmente una red Ethernet, aunque es compatible con algunas otras) estableciendo la configuración en modo promiscuo.

Permite examinar datos de una red viva o de un archivo de captura salvado en disco. Se puede analizar la información capturada, a través de los detalles y sumarios por cada paquete. Wireshark incluye un completo lenguaje para filtrar lo que queremos ver y la habilidad de mostrar el flujo reconstruido de una sesión de TCP.

Wireshark es software libre, y se ejecuta sobre la mayoría de sistemas operativos Unix y compatibles, incluyendo Linux, Solaris, FreeBSD, NetBSD, OpenBSD, Android, y Mac OS X, así como en Microsoft Windows.

En este proyecto se ha empleado para capturar el tráfico de la interfaz 'loopback' para observar que las peticiones a la Base de Datos mediante las URIs de recursos se realizaban correctamente.



Figura 3.8: Wireshark

3.6 Winpcap

WinPcap es la herramienta estándar de la industria para acceder a la conexión entre capas de red en entornos Windows. Permite a las aplicaciones capturar y transmitir los paquetes de red puenteando la pila de protocolos; y tiene útiles características adicionales que incluyen el filtrado de paquetes a nivel del núcleo, un motor de generación de estadísticas de red y soporte para captura de paquetes. WinPcap consiste en un controlador, que extiende el sistema operativo para proveer acceso de red a bajo nivel, y una biblioteca que se usa para acceder fácilmente a las capas de red de bajo nivel. Esta biblioteca también contiene la versión de Windows de la bien conocida API de Unix, libpcap.

Gracias a este conjunto de características, WinPcap es el motor de captura de paquetes y filtrado de muchas de las herramientas de red comerciales y de código abierto, incluyendo analizadores de protocolos, monitores de red, sistemas de detección de intrusos de red, sniffers, generadores de tráfico y *network testers*.

Esta biblioteca es indispensable para el correcto funcionamiento de programas como Wireshark (antes conocido como Ethereal).



Figura 3.9: Winpcap

3.7 mobile browser emulator

Es una extensión de Google Chrome para poder visualizar una página web en diferentes resoluciones de pantalla móvil y tablet.

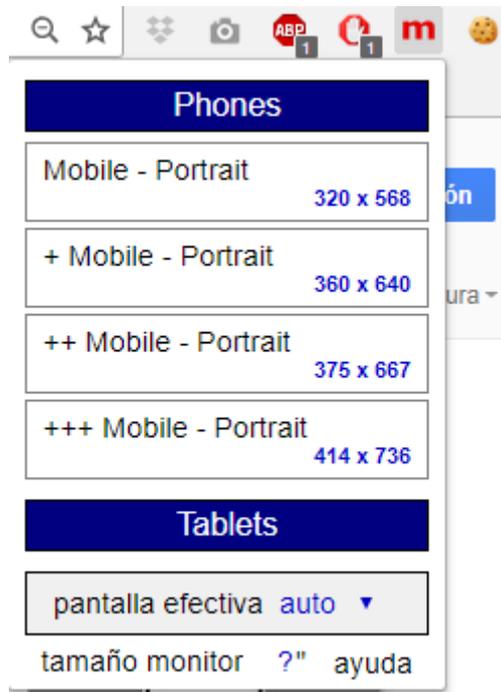


Figura 3.10: mobile browser emulator

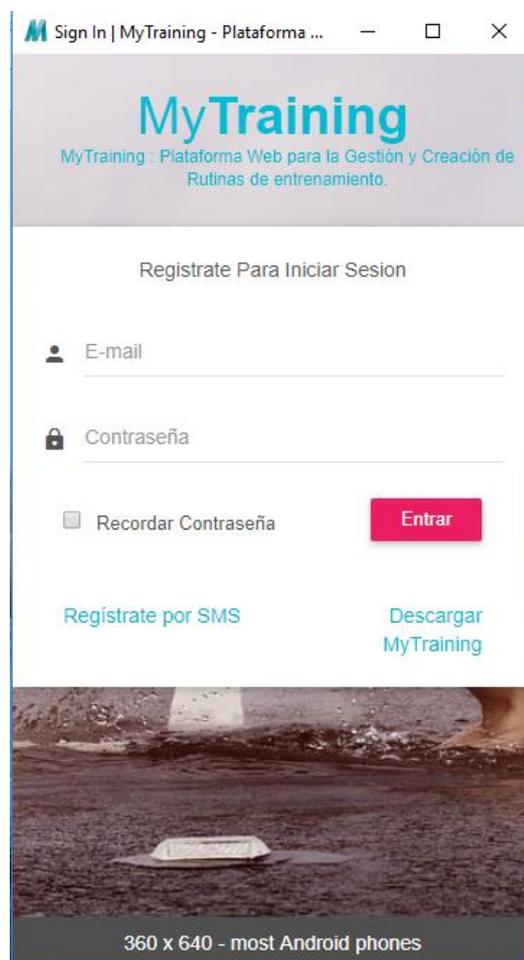


Figura 3.11: MyTraining en móvil

3.8 Account kit de Facebook

Account Kit es una herramienta proporcionada por Facebook que te permite iniciar sesión en las aplicaciones de una manera rápida y sencilla. Para ello solo te hace falta tu correo electrónico o tu número de teléfono (no es necesario introducir contraseña). Se requiere crearse una cuenta de desarrollador en Facebook. Incluso en el caso de que la tengas, con Account Kit no tienes que compartir información directamente desde tu perfil de Facebook para iniciar sesión en las aplicaciones.

Con esta herramienta es con la que realizamos en el proyecto la autenticación por SMS.



Figura 3.12: Account kit Facebook

3.9 Youtube

YouTube (pronunciación [AFI](#) ['ju:tju:b]) es un sitio web dedicado a compartir vídeos. Presenta una variedad de clips de películas, programas de televisión y vídeos musicales, así como contenidos amateur como videoblogs y YouTube Gaming. A pesar de las reglas de YouTube contra subir vídeos con todos los derechos reservados, este material existe en abundancia.

YouTube usa un reproductor en línea basado en HTML5, que incorporó poco después de que la W3C lo presentara y que es soportado por los navegadores web más difundidos. Antiguamente su reproductor funcionaba con Adobe Flash, pero esta herramienta fue desechada en 2016. Los enlaces a vídeos de YouTube pueden ser también insertados en blogs y sitios electrónicos personales usando API o incrustando cierto código HTML.



Figura 3.13: Logo de Youtube

4 Servicio web

4.1 Clases

A continuación, se explican brevemente las distintas clases java que hacen que la aplicación web funcione:

- **UriConstants:** En esta clase definimos como variables estáticas las URIs que se emplean para solicitar al servidor a través de AJAX los recursos necesarios.
- **SQLConstants:** En esta clase definimos como variables estáticas las sentencias SQL que ejecuta el servidor a la Base de Datos a través de la implementación del modelo DAO.
- **UsuarioController, RutinaController, EjercicioController y VideoController:** Estas son las clases “Controlador”. Dichas clases poseen una serie de funciones que son llamadas a través de las peticiones realizadas a las URIs y del método utilizado (GET, POST, PUT, DELETE).
- **UsuarioDaoImpl, RutinaDaoImpl, EjercicioDaoImpl y VideoDaoImpl:** Estas clases poseen funciones que son llamadas por la clase “Controlador” correspondiente. Dichas funciones se encargan de realizar las sentencias SQL asociadas para obtener los datos deseados y mapearlos en un objeto java el cual es devuelto al controlador. Estas clases son implementaciones de las siguientes interfaces: **UsuarioDao, RutinaDao, EjercicioDao y VideoDao.**
- **Usuario, UsuarioLogin, UsuarioRol, Rutina, Ejercicio y Video:** Estas clases definen los objetos que maneja el servicio Web, y en ellas se define el constructor, los atributos y métodos ‘get’ y ‘set’ correspondientes a cada objeto.
- **UsuarioRowMapper, UsuarioRowLoginMapper, UsuarioRolRowMapper, RutinaRowMapper, EjercicioRowMapper y VideoRowMapper:** Estas clases permiten la realización de un mapeo entre los datos obtenidos de la BBDD a través de un objeto DAO a un objeto Java determinado.
- **SecurityConfiguration, RESTAuthenticattionSuccessHandler, RESTAuthenticationFailureHandler, RESTAuthenticationEmtryPoint:** Estas clases son las que configuran el módulo de Spring Security.
- **MyTrainingApplication:** Clase principal del servicio web.

4.2 Diagramas de clases

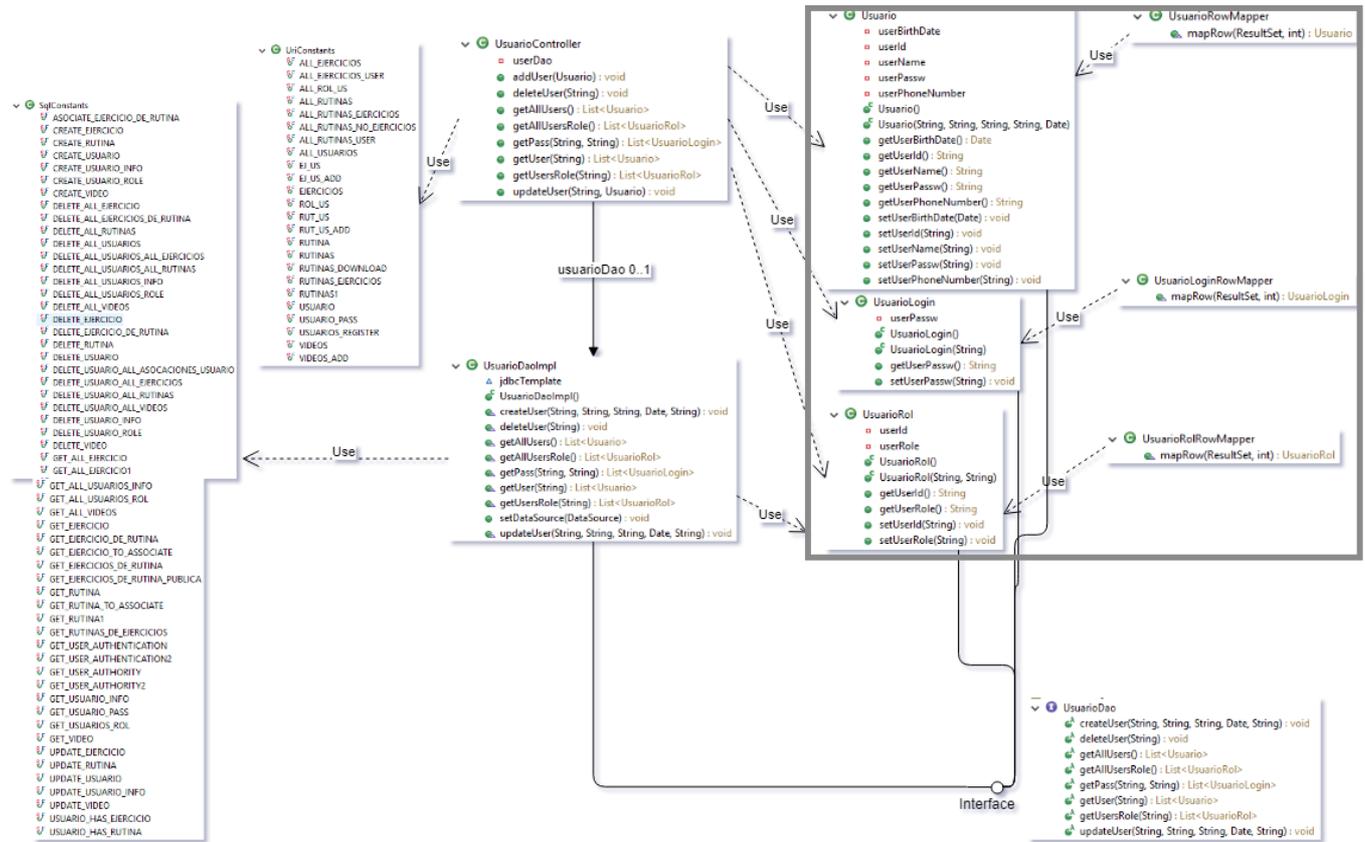


Figura 4.1: Diagrama usuario

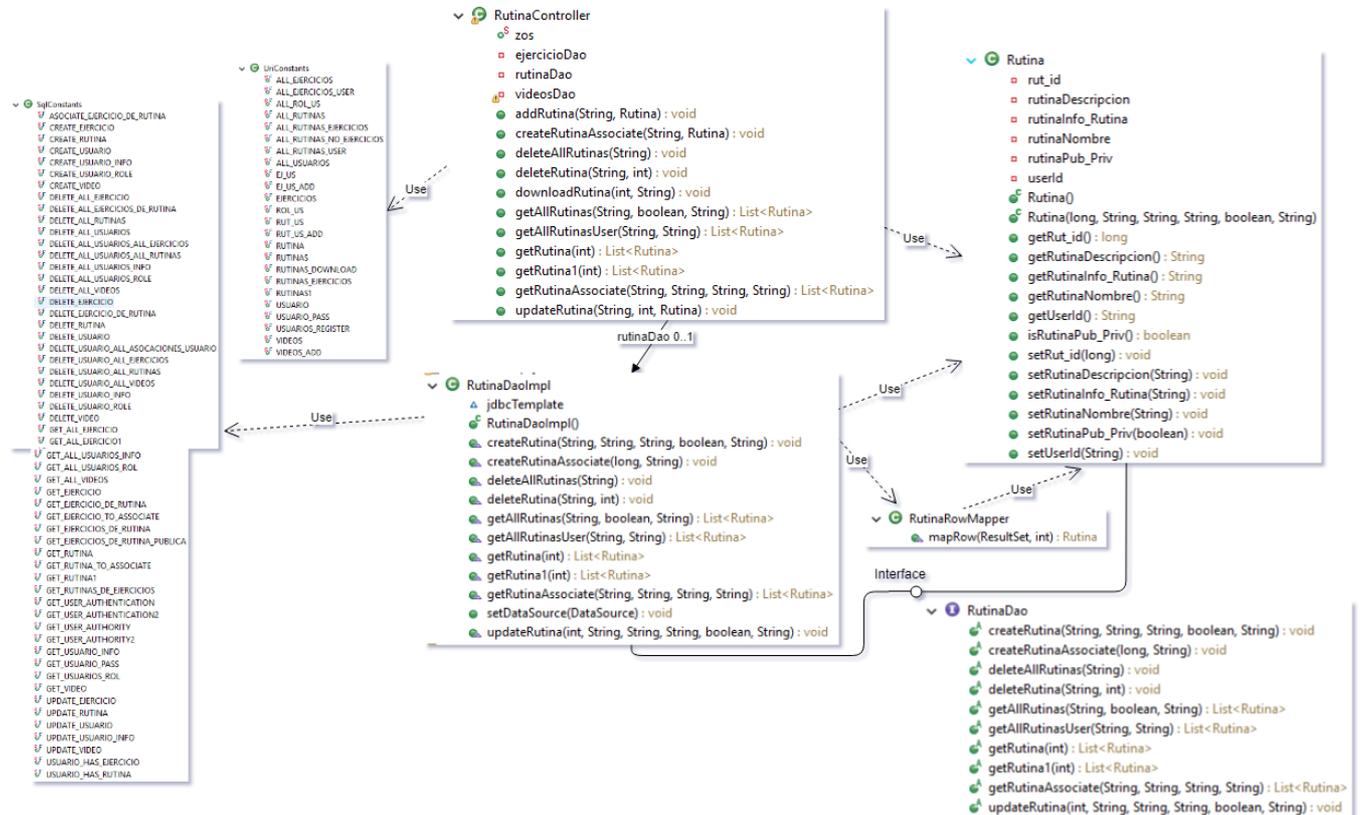


Figura 4.2: Diagrama rutina.

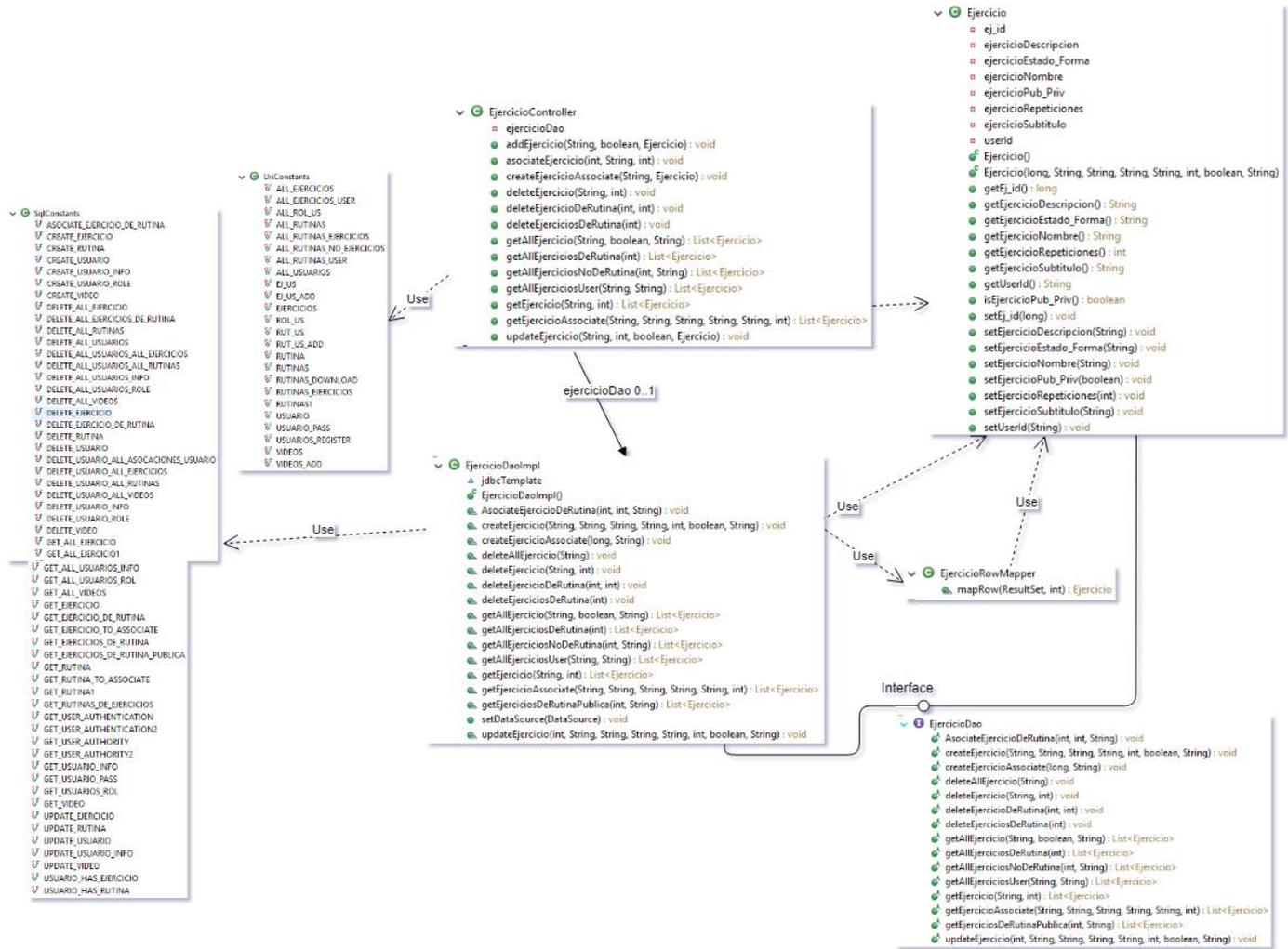


Figura 4.3: Diagrama ejercicio.

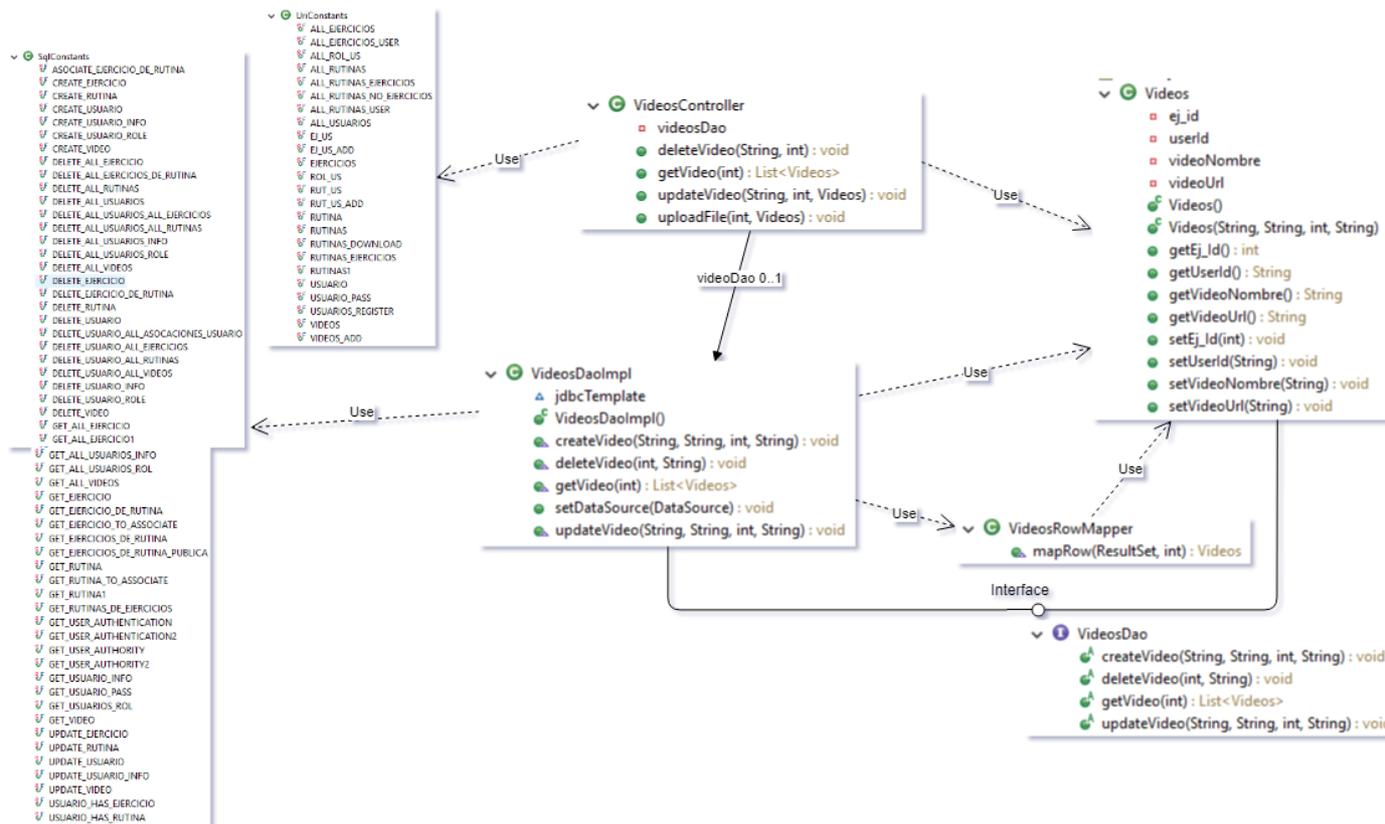


Figura 4.4: Diagrama video.

4.3 Interacción entre clases

A continuación, vamos a explicar la estructura del código y cómo interaccionan entre sí las clases relacionadas tomando de ejemplo “rutina”. La estructura del resto de clases es similar, por ello nos centramos en una.

Sentencias SQL

En esta clase se definen las sentencias SQL para obtener los datos de la BD. Los interrogantes (“?”) indican los parámetros que habrá que pasarle a cada una para que la operación sea efectiva.



```

SqlConstants.java
/* RUTINAS */
public static final String CREATE_RUTINA =
    "INSERT INTO RUTINA (Nombre,Descripcion,Info_Rutina,Pub_priv,USUARIOS_Email) VALUES (?,?,,?,?)";

public static final String GET_RUTINA_TO_ASSOCIATE =
    "SELECT rut_id,Nombre,Descripcion,Info_Rutina,Pub_priv FROM RUTINA WHERE USUARIOS_Email=? AND Nombre=? AND Descripcion=? AND Info_Ruti

public static final String USUARIO_HAS_RUTINA =
    "INSERT INTO USUARIO_HAS_RUTINA (usuario_email,rutina_id) VALUES (?,?)";

public static final String UPDATE_RUTINA =
    "UPDATE RUTINA SET Nombre=?,Descripcion=?,Info_Rutina=?,Pub_priv=? WHERE rut_id=? AND USUARIOS_Email=? ";

//Obtener rutinas de un usuario
public static final String GET_RUTINA =
    "SELECT rut_id,Nombre,Descripcion,Info_Rutina,Pub_priv FROM RUTINA WHERE rut_id=?";

//Obtener rutinas de un usuario
public static final String GET_RUTINA1 =
    "SELECT rut_id,Nombre,Descripcion,Info_Rutina,Pub_priv FROM RUTINA WHERE rut_id=?";

//Los dos siguientes estan implementados directamente en RutinaDaoImpl!

//Eliminar rutinas de un usuario
public static final String DELETE_RUTINA =
    "DELETE FROM RUTINA WHERE USUARIOS_Email=? AND rut_id=?";
public static final String DELETE_ALL_RUTINAS =
    "DELETE FROM RUTINA WHERE USUARIOS_Email=?";

/* EJERCICIOS */
public static final String CREATE_EJERCICIO =
    "INSERT INTO EJERCICIO (Nombre,Subtitulo,Descripcion,Estado_forma,Repeticiones,Pub_priv,RUTINA_USUARIOS_Email) VALUES (?,,?,,?,,?,?)";

```

Figura 4.5: SqlConstants.java

URIs

En esta clase se definen las URIs en las que se alojan los recursos. Los elementos entre corchetes son parámetros utilizados por una de las clases que explicaremos a continuación.



```

UriConstants.java
public class UriConstants {

    /* REGISTER Rutinas */
    public static final String USUARIOS_REGISTER = "/RutinaRegister/";

    /* OWNERS */
    public static final String USUARIO = "/Rutina_app/{user_id:}";
    public static final String USUARIO_PASS = "/SMS/{user_id:}/{phoneNumber:}";
    public static final String ALL_USUARIOS = "/Rutina_app/";
    public static final String ROL_US = "/Rutina_app/rol/{user_id:}";
    public static final String ALL_ROL_US = "/Rutina_app/rol/";

    /* RUTINAS*/
    public static final String RUTINA = "/Rutina_app/rutinas/{rut_id:}";
    public static final String RUTINAS = "/Rutina_app/rutinas/{user_id:}/{rut_id:}";
    public static final String RUTINAS1= "/Rutina_app/rutinas/publicas/{user_id:}";
    public static final String RUTINAS_DOWNLOAD = "/Rutina_app/downloads/{rut_id:}";
    public static final String ALL_RUTINAS = "/Rutina_app/rutinas/{user_id:}";
    public static final String ALL_RUTINAS_USER = "/Rutina_app/rutinas_user/{user_id:}";
    public static final String RUT_US = "/Rutina_app/rut/{user_id:}";
    public static final String RUT_US_ADD = "/Rutina_app/rut/add/{user_id:}";

    /* EJERCICIOS */
    public static final String EJERCICIOS = "/Rutina_app/ejercicios/{user_id:}/{ej_id:}";
    public static final String ALL_EJERCICIOS = "/Rutina_app/ejercicios/{user_id:}";
    public static final String EJ_US = "/Rutina_app/ej/{user_id:}";
    public static final String EJ_US_ADD = "/Rutina_app/ej/add/{user_id:}";
    public static final String ALL_EJERCICIOS_USER = "/Rutina_app/ejercicios_user/{user_id:}";

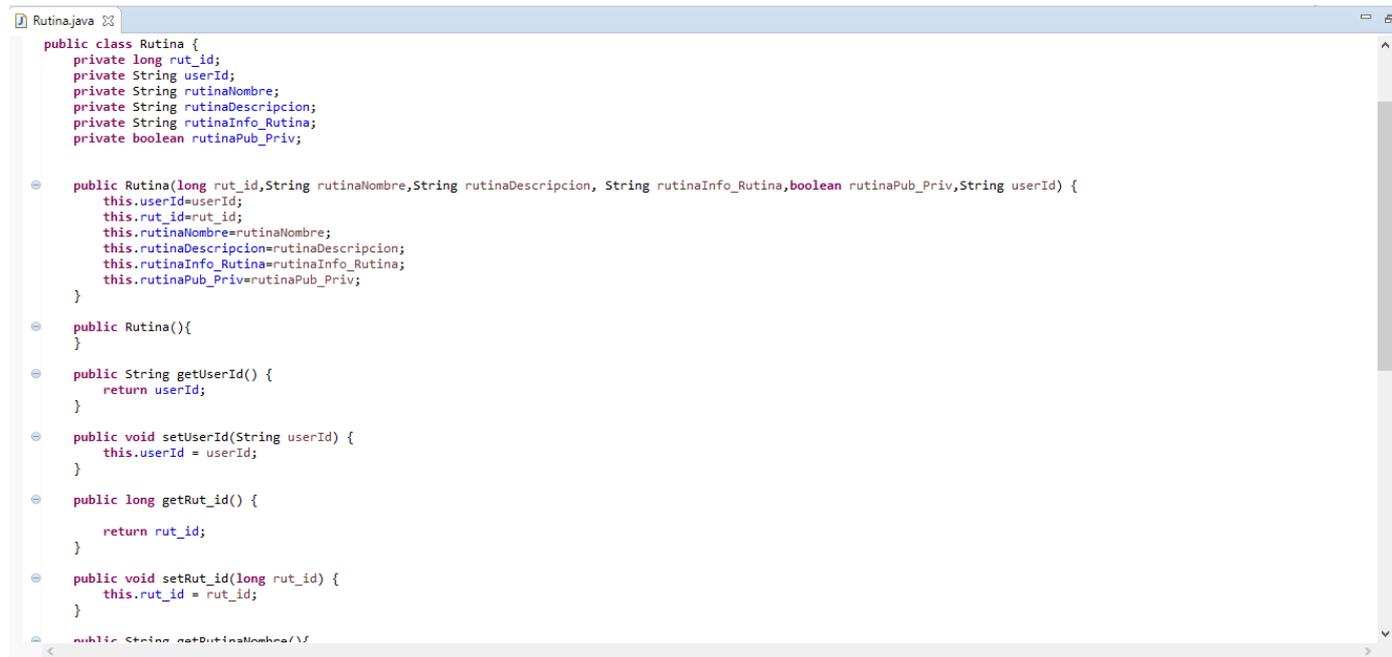
    /* VIDEOS */
    //public static final String ALL_VIDEOS = "/Rutina_app/videos/{owner_id:}";
}

```

Figura 4.6: UriConstants.java

Objeto 'Rutina'

En esta imagen podemos ver los atributos propios de dicha clase seguido de su constructor y sus métodos correspondientes.



```

public class Rutina {
    private long rut_id;
    private String userId;
    private String rutinaNombre;
    private String rutinaDescripcion;
    private String rutinaInfo_Rutina;
    private boolean rutinaPub_Priv;

    public Rutina(long rut_id,String rutinaNombre,String rutinaDescripcion, String rutinaInfo_Rutina,boolean rutinaPub_Priv,String userId) {
        this.userId=userId;
        this.rut_id=rut_id;
        this.rutinaNombre=rutinaNombre;
        this.rutinaDescripcion=rutinaDescripcion;
        this.rutinaInfo_Rutina=rutinaInfo_Rutina;
        this.rutinaPub_Priv=rutinaPub_Priv;
    }

    public Rutina(){
    }

    public String getUserId() {
        return userId;
    }

    public void setUserId(String userId) {
        this.userId = userId;
    }

    public long getRut_id() {
        return rut_id;
    }

    public void setRut_id(long rut_id) {
        this.rut_id = rut_id;
    }

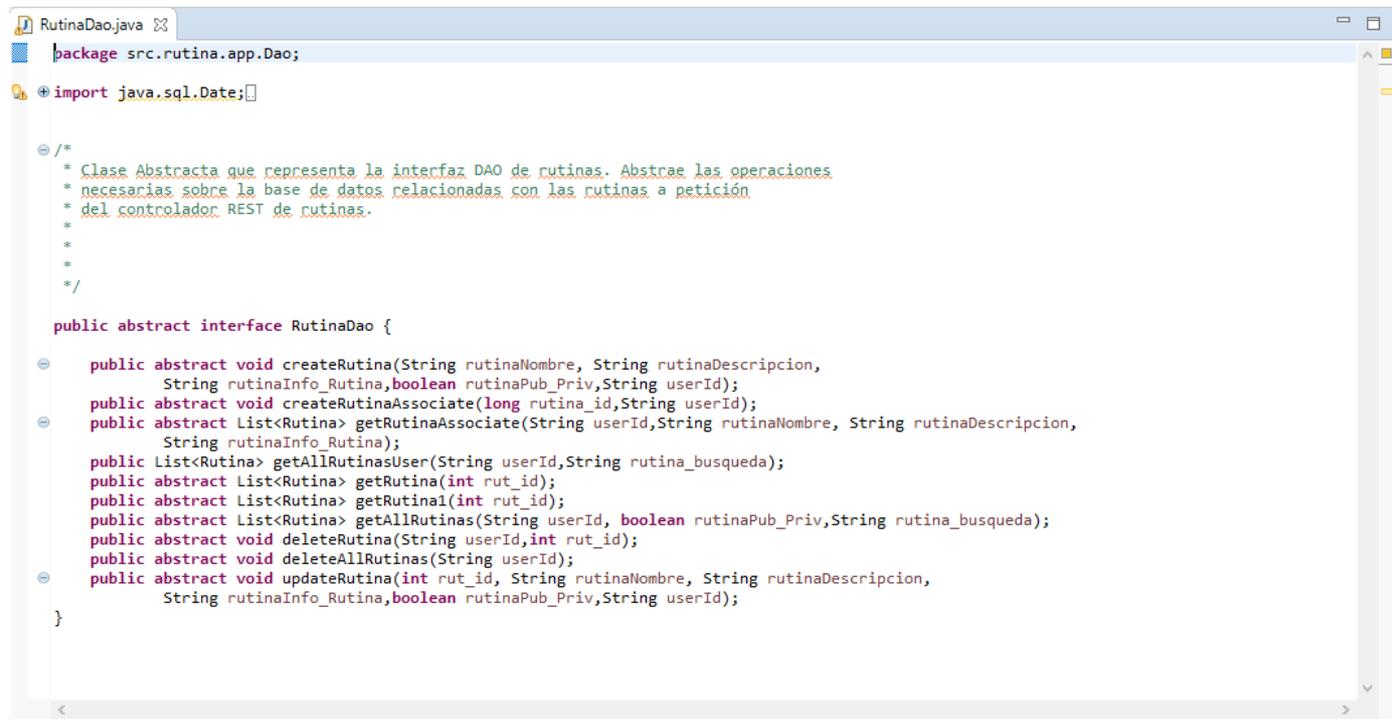
    public String getRutinaNombre() {

```

Figura 4.7: Rutina.java

Interfaz del patrón de diseño DAO

En esta clase se declaran los métodos que serán utilizados por la clase de implementación del patrón de diseño DAO.



```

package src.rutina.app.Dao;

import java.sql.Date;

/**
 * Clase Abstracta que representa la interfaz DAO de rutinas. Abstrae las operaciones
 * necesarias sobre la base de datos relacionadas con las rutinas a petición
 * del controlador REST de rutinas.
 */

public abstract interface RutinaDao {

    public abstract void createRutina(String rutinaNombre, String rutinaDescripcion,
        String rutinaInfo_Rutina,boolean rutinaPub_Priv,String userId);
    public abstract void createRutinaAssociate(long rutina_id,String userId);
    public abstract List<Rutina> getRutinaAssociate(String userId,String rutinaNombre, String rutinaDescripcion,
        String rutinaInfo_Rutina);
    public List<Rutina> getAllRutinasUser(String userId,String rutina_busqueda);
    public abstract List<Rutina> getRutina(int rut_id);
    public abstract List<Rutina> getRutina1(int rut_id);
    public abstract List<Rutina> getAllRutinas(String userId, boolean rutinaPub_Priv,String rutina_busqueda);
    public abstract void deleteRutina(String userId,int rut_id);
    public abstract void deleteAllRutinas(String userId);
    public abstract void updateRutina(int rut_id, String rutinaNombre, String rutinaDescripcion,
        String rutinaInfo_Rutina,boolean rutinaPub_Priv,String userId);
}

```

Figura 4.8: RutinaDao.java

Implementación del patrón de diseño DAO

Esta clase implementa los métodos declarados en la interfaz anteriormente descrita. Utiliza un objeto 'JdbcTemplate' para realizar las operaciones con la base de datos empleando para ello las sentencias SQL definidas en la clase 'SqlConstants.java' y un objeto con los parámetros a pasarle a dicha sentencia.



```

public class RutinaDaoImpl implements RutinaDao {
    JdbcTemplate jdbcTemplate;

    @Override
    public void createRutina(String rutinaNombre, String rutinaDescripcion, String rutinaInfo_Rutina, boolean rutinaPub_Priv, String userId) {
        jdbcTemplate.update(SqlConstants.CREATE_RUTINA,
            new Object[] { rutinaNombre, rutinaDescripcion, rutinaInfo_Rutina, rutinaPub_Priv, userId });
    }

    @Override
    public void updateRutina(int rut_id, String rutinaNombre, String rutinaDescripcion,
        String rutinaInfo_Rutina, boolean rutinaPub_Priv, String userId) {

        System.out.println("DaoImplUpdate:"+ rut_id+rutinaNombre+rutinaDescripcion+rutinaInfo_Rutina+rutinaPub_Priv+userId);
        jdbcTemplate.update(SqlConstants.UPDATE_RUTINA,
            new Object[] { rutinaNombre, rutinaDescripcion, rutinaInfo_Rutina, rutinaPub_Priv, rut_id, userId});
    }

    @Override
    public List<Rutina> getRutina( int rut_id) {
        return jdbcTemplate.query(SqlConstants.GET_RUTINA,
            new Object[] {rut_id },
            new RutinaRowMapper());
    }

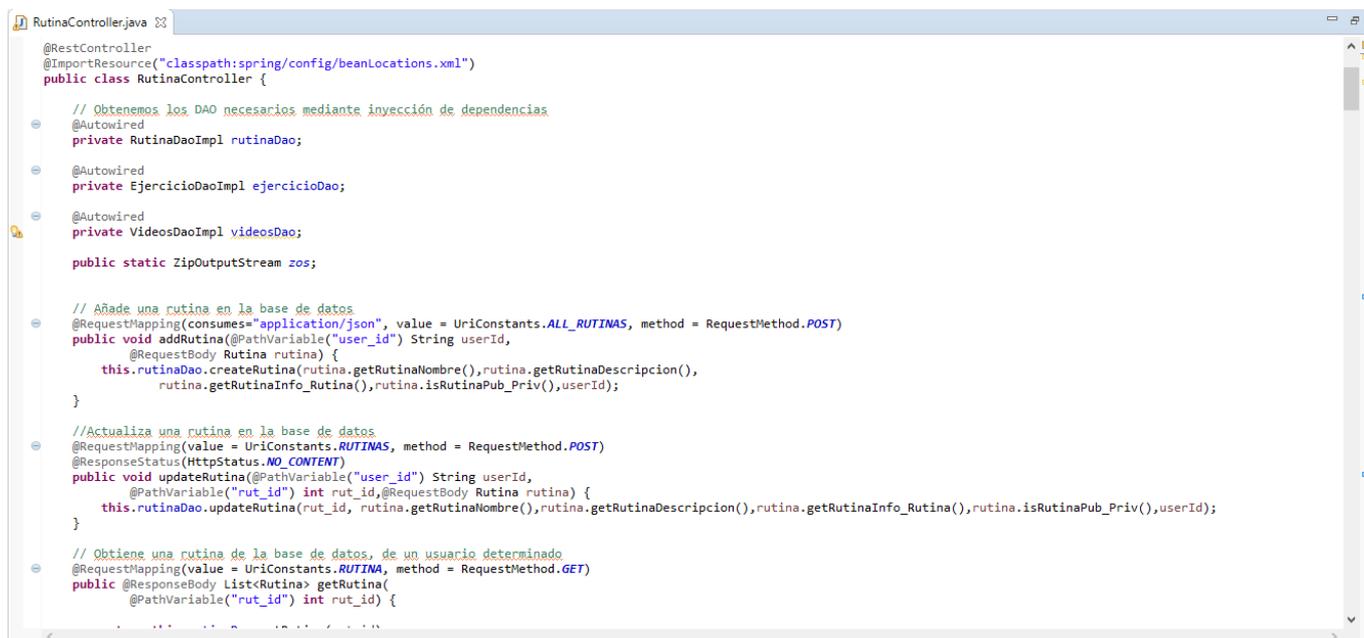
    @Override
    public void createRutinaAssociate(long rutina_id, String userId) {
        jdbcTemplate.update(SqlConstants.USUARIO_HAS_RUTINA,
            new Object[] { userId, rutina_id });
    }
}

```

Figura 4.9: RutinaDaoImpl.java

Controlador

Esta clase se encarga de, mediante anotaciones de Spring, asociar las peticiones HTTP a la ejecución de un método concreto. En la anotación se debe indicar la URI (presente en UriConstants.java) asociada a ese método y el tipo de petición (GET, POST, PUT o DELETE). Dicho método contiene una llamada de ejecución a uno de los métodos declarado y definido en las dos clases anteriores del patrón DAO. Mediante anotaciones se obtienen también los parámetros enviados en la propia URI para utilizarlos en la llamada.



```

@RestController
@ImportResource("classpath:spring/config/beanLocations.xml")
public class RutinaController {

    // Obtenemos los DAO necesarios mediante inyección de dependencias
    @Autowired
    private RutinaDaoImpl rutinaDao;

    @Autowired
    private EjercicioDaoImpl ejercicioDao;

    @Autowired
    private VideosDaoImpl videosDao;

    public static ZipOutputStream zos;

    // Añade una rutina en la base de datos
    @RequestMapping(consumes="application/json", value = UriConstants.ALL_RUTINAS, method = RequestMethod.POST)
    public void addRutina(@PathVariable("user_id") String userId,
        @RequestBody Rutina rutina) {
        this.rutinaDao.createRutina(rutina.getRutinaNombre(),rutina.getRutinaDescripcion(),
            rutina.getRutinaInfo_Rutina(),rutina.isRutinaPub_Priv(),userId);
    }

    //Actualiza una rutina en la base de datos
    @RequestMapping(value = UriConstants.RUTINAS, method = RequestMethod.POST)
    @ResponseStatus(HttpStatus.NO_CONTENT)
    public void updateRutina(@PathVariable("user_id") String userId,
        @PathVariable("rut_id") int rut_id,@RequestBody Rutina rutina) {
        this.rutinaDao.updateRutina(rut_id, rutina.getRutinaNombre(),rutina.getRutinaDescripcion(),rutina.getRutinaInfo_Rutina(),rutina.isRutinaPub_Priv(),userId);
    }

    // Obtiene una rutina de la base de datos, de un usuario determinado
    @RequestMapping(value = UriConstants.RUTINA, method = RequestMethod.GET)
    public @ResponseBody List<Rutina> getRutina(
        @PathVariable("rut_id") int rut_id) {

```

Figura 4.10: RutinaController.java

Mapeador

Esta clase es utilizada por la clase que implementa el patrón DAO. Es utilizada en aquellos métodos cuya sentencia SQL devuelve datos (sentencias ‘select’) para mapearlos, en este caso, en un objeto rutina y enviarlo como respuesta a la llamada de dicho método realizada por el controlador.

```

package src.rutina.app.RowMappers;

import java.sql.ResultSet;

/**
 * Clase que representa un mapeador de Rutinas. Obtiene y crea una Rutina
 * a partir de los datos obtenidos de la base de datos.
 */

public class RutinaRowMapper implements RowMapper<Rutina> {

    public Rutina mapRow(ResultSet resultSet, int arg1) throws SQLException {
        Rutina rutina = new Rutina(resultSet.getInt(1), resultSet.getString(2),
            resultSet.getString(3),resultSet.getString(4),resultSet.getBoolean(5), "");

        return rutina;
    }
}

```

Figura 4.11: RutinaRowMapper.java

Ejemplo de petición

Las peticiones realizadas a través de HTTP para la realización de operaciones con los recursos se ejecutan mediante AJAX, especificando la URI junto con sus parámetros si los tuviera, el tipo de petición y datos en formato json si fuera necesario. AJAX es asíncrono, por lo que las peticiones realizadas serán asíncronas. Esto implica que si realizamos dos peticiones consecutivas, no tiene por qué finalizar primero la petición que se ejecutó en primer lugar. Esto puede acarrear problemas en el caso de que la segunda petición dependa de la correcta finalización de la primera. Para ello utilizamos el parámetro de AJAX ‘async’ que gobierna la asincronía y que, si ponemos a ‘false’, no seguirá ejecutando código hasta que esa petición se haya realizado. Por defecto, tiene valor ‘true’.

```

// Añadimos la rutina a la base de datos
$.ajax({
    url : "/Rutina_app/rutinas/" + cookie.userid + "/",
    headers: {'X-CSRF-TOKEN': cookie.csrf},
    type : "POST",
    data : JSON.stringify(rutina_json),
    contentType : "application/json",
    async:false, // Hey browser! first complete this request,
                // then go for other codes
    // En caso de éxito: informamos y redirigimos
}).done(function (data, textStatus, jqXHR) {
    alert("Rutina añadida con éxito al Especialista");

// Avisamos al usuario de que ha surgido un error
}).fail(function (jqXHR, textStatus, errorThrown) {
    alert("Se ha producido un error.");
});

```

Figura 4.12: Petición AJAX

4.4 URIs

En este apartado vamos a proceder a explicar con detalle las URIs en las que se alojan los recursos que empleamos en la aplicación.

A.1 Recurso Usuario

Método	URI	Parámetros	Acción
POST	"/RutinaRegister/"	Atributos de un Usuario	Crear un nuevo usuario con sus datos personales
POST	"/Rutina_app/{user_id:.+}"	<i>user_id</i> : Identificador único del usuario. Atributos de un Usuario.	Modificar los datos personales propios del usuario
GET	"/Rutina_app/{user_id:.+}"	<i>user_id</i> : Identificador único de un usuario.	Obtener datos de un usuario
GET	"/SMS/{user_id:.+}/{phoneNumber:.+}"	<i>user_id</i> : Identificador único de un usuario. <i>phoneNumber</i> : Teléfono móvil del usuario.	Obtener la contraseña de un usuario
GET	"/Rutina_app/"	(Sin parámetros)	Obtener datos de todos los usuarios
GET	"/Rutina_app/rol/{user_id:.+}"	<i>user_id</i> : Identificador único de un usuario.	Obtener el rol de un usuario
GET	"/Rutina_app/rol/"	(Sin parámetros)	Obtener los roles de todos los usuarios
DELETE	"/Rutina_app/{user_id:.+}"	<i>user_id</i> : Identificador único de un usuario.	Eliminar un usuario junto con sus datos

A.2 Recurso Rutina

Método	URI	Parámetros	Descripción
POST	"/Rutina_app/rutinas/{user_id:.+}/"	<i>user_id</i> : Identificador único del usuario. Atributos de una Rutina	Crear una nueva rutina con sus datos

POST	"/Rutina_app/rutinas/{user_id:.+}/{rut_id:.+}"	<p><i>user_id:</i> Identificador único de un usuario.</p> <p>Atributos de una Rutina.</p> <p><i>rut_id:</i> Identificador único de una rutina.</p>	Modificar una rutina.
GET	"/Rutina_app/rutinas/{rut_id:.+}"	<p><i>rut_id:</i> Identificador único de una rutina.</p>	Obtener los datos de una rutina.
GET	"/Rutina_app/rut/{user_id:.+}/"	<p><i>user_id:</i> Identificador único de un usuario.</p> <p>Atributos de una Rutina.</p>	Obtener el identificador de una rutina.
POST	"/Rutina_app/rut/add/{user_id:.+}/"	<p><i>user_id:</i> Identificador único de un usuario.</p>	Asociar una rutina a un usuario.
GET	"/Rutina_app/rutinas_user/{user_id:.+}/"	<p><i>user_id:</i> Identificador único de un usuario.</p> <p>Elemento de búsqueda (opcional)</p>	Obtener las rutinas asociadas a un usuario.
GET	"/Rutina_app/rutinas/{user_id:.+}/"	<p><i>user_id:</i> Identificador único de un usuario.</p> <p>Elemento de búsqueda (opcional)</p> <p>Bandera de rutinas: propias/Del resto</p>	Obtener las rutinas relacionadas.
DELETE	"/Rutina_app/rutinas/{user_id:.+}/{rut_id:.+}"	<p><i>user_id:</i> Identificador único de un usuario.</p>	Eliminar una rutina junto a sus datos.
POST	"/Rutina_app/downloads/{rut_id:.+}"	<p><i>rut_id:</i> Identificador único de una rutina.</p>	Descargar la rutina en un json

A.3 Recurso Ejercicio

Método	URI	Parámetros	Descripción
GET	"/Rutina_app/ejercicios/{user_id:.+}/{ej_id:.+}"	<i>user_id</i> : Identificador único de un usuario. <i>ej_id</i> : Identificador único de un ejercicio.	Obtener los datos de un ejercicio
GET	"/Rutina_app/ejercicios/{user_id:.+}/"	<i>user_id</i> : Identificador único de un usuario. Elemento de búsqueda (opcional) Bandera de ejercicios: propias/Del resto	Obtener los ejercicios relacionados
GET	"/Rutina_app/ej/{user_id:.+}/"	Atributos de un Ejercicio. <i>user_id</i> : Identificador único de un usuario.	Obtener el identificador del ejercicio
POST	"/Rutina_app/ej/add/{user_id:.+}/"	<i>user_id</i> : Identificador único de un usuario. Atributos de un Ejercicio	Asociar un ejercicio a un usuario
GET	"/Rutina_app/ejercicios_user/{user_id:.+}/"	<i>user_id</i> : Identificador único de un usuario. Elemento de búsqueda (opcional)	Obtiene los ejercicios asociados a un usuario
DELETE	"/Rutina_app/ejercicios/{user_id:.+}/{ej_id:.+}"	<i>user_id</i> : Identificador único de un usuario. <i>ej_id</i> : Identificador único de un ejercicio	Eliminar un ejercicio

DELETE	"/Rutina_app/rutinas/asociaciones/{rut_id:.+}/{ej_id:.+}"	<i>ej_id</i> :Identificador único de un ejercicio <i>rut_id</i> :Identificador único de una rutina	Eliminar asociación de 'y' ejercicio asociado 'x' a rutina
POST	"/Rutina_app/ejercicios/{user_id:.+}/"	<i>user_id</i> : Identificador único de un usuario. Atributos de un Ejercicio	Crear un ejercicio
POST	"/Rutina_app/ejercicios/{user_id:.+}/{ej_id:.+}"	<i>user_id</i> : Identificador único de un usuario. <i>ej_id</i> :Identificador único de un ejercicio Atributos de un Ejercicio	Modificar un ejercicio
DELETE	"/Rutina_app/rutinas/asociaciones/{rut_id:.+}/"	<i>rut_id</i> :Identificador único de una rutina	Eliminar las asociaciones de todos los ejercicios asociados a una rutina 'x'
GET	"/Rutina_app/rutinas/noasociaciones/{rut_id:.+}/"	<i>rut_id</i> :Identificador único de una rutina Elemento de búsqueda (opcional)	Obtener los ejercicios que no están asociados a 'x' rutina
POST	"/Rutina_app/rutinas/asociaciones/{rut_id:.+}/{ej_id:.+}"	<i>user_id</i> : Identificador único de un usuario. <i>ej_id</i> :Identificador único de un ejercicio	Asociar ejercicios a rutinas
GET	"/Rutina_app/rutinas/asociaciones/{rut_id:.+}/"	<i>rut_id</i> :Identificador único de una rutina	Obtener los ejercicios asociados a una rutina

A.4 Recurso Vídeo

Método	URI	Parámetros	Descripción
GET	"/Rutina_app/videos/{user_id:.+}/{ej_id:.+}"	<i>user_id</i> : Identificador único de un usuario. <i>ej_id</i> : Identificador único de un ejercicio	Obtener video
POST	"/Rutina_app/videos/{ej_id:.+}"	<i>ej_id</i> : Identificador único de un ejercicio Atributos de un Video	Añadir video
DELETE	"/Rutina_app/videos/{user_id:.+}/{ej_id:.+}"	<i>user_id</i> : Identificador único de un usuario. <i>ej_id</i> : Identificador único de un ejercicio	Eliminar video

4.5 Youtube

A través de HTML5 hemos conseguido alojar videos de Youtube en nuestra aplicación web sin necesidad de subirlos y alojarlos en el servidor físicamente, lo que hubiera supuesto una gran necesidad de recursos para ello. El propio Youtube te proporciona el código para ello.

El código se obtiene pulsando en la opción de  **COMPARTIR** en cualquier video de Youtube que quieras y escogiendo **INSERTAR**.



Figura 4.13: Código de Youtube

Este código lo hemos implementado en nuestra aplicación de manera que a la hora de añadir un video a un ejercicio, el especialista solo tenga que rellenar un formulario en el que debe copiar y pegar la URL de la parte superior de la página para que pueda visualizarse el video. Esto se ha logrado mediante AJAX, modificando el código base que proporciona Youtube para incrustar la URL de cualquier video que los especialistas hayan añadido.

```
/* Funcion que permite la visualizacion de un video de un ejercicio */
function repvideo() {
  // Obtenemos el contenedor donde imprimiremos los ejercicios
  var container = $("#videCont")[0];

  var userid = getUrlParameter('user_id');
  var ej_id = getUrlParameter('ej_id');
  var json = getVideoData(ej_id);
  console.log("JSON video URL: " + json.videoUrl);
  var url = json[0].videoUrl;
  var url1 = url.split("v=");
  var url2 = url1[1].split("&");
  console.log("URL: " + url1 + "URL2: " + url2);

  console.log(ej_id);
  var summedSeeVideo = "<h3 class='text-center'>" + json[0].videoNombre + "</h3>"
    + "<iframe class='youtube' "
    + "src='https://www.youtube.com/embed/"
    + url2[0] + "?rel=0'"
    + "frameborder='0' allow='autoplay; encrypted-media' allowfullscreen></iframe>"

  container.innerHTML += summedSeeVideo;
}
```

Figura 4.14: Código de Youtube 2

4.6 Spring Security

En nuestro proyecto hemos utilizado este módulo de Spring como encargado del ámbito de la seguridad en torno a la autenticación de usuario, el control de acceso a las diversas páginas de la aplicación basándose en el rol del usuario y la obtención de recursos de la aplicación.

En la siguiente imagen se explica de manera visual como actúa Spring Security a la hora de enfrentarse al proceso de autenticación de credenciales antes de otorgar los recursos que la petición realizada por el usuario pide.

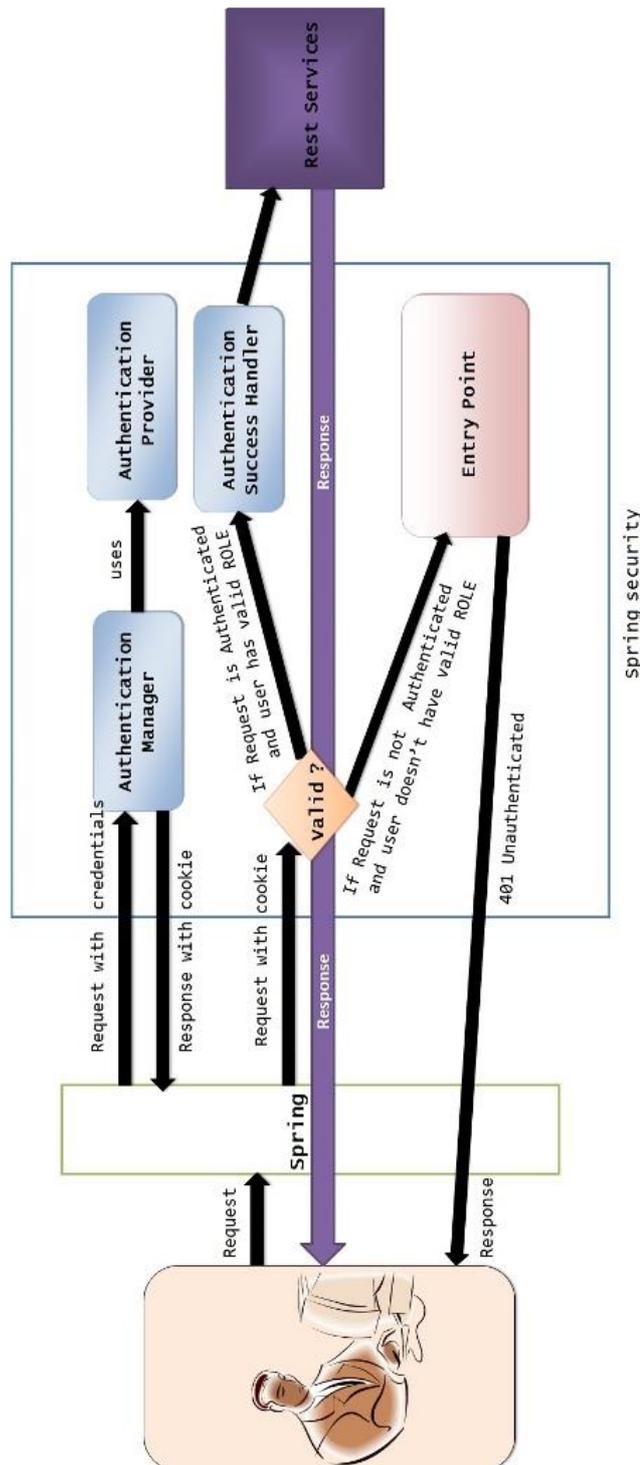


Figura 4.15: Diagrama de acción de Spring Security.

En el diagrama puede verse como a la hora de comprobar credenciales sucede lo siguiente:

1. El usuario envía una petición con sus credenciales como dato y Spring la captura.
2. La petición es manejada por el “Authentication Manager” o “Gestor de autenticación” que decide cuando un usuario es válido. Este gestor está relacionado con el “Authentication Provider” que es quién define la forma en la que un usuario se ha de validar. Puede ser contra una base de datos, puede ser contra un Ldap, puede ser contra un fichero o puede personalizarse.
3. El “Authentication Manager” devuelve una cookie respuesta a Spring.
4. Si la Cookie es válida, significa que la autenticación de las credenciales ha resultado con éxito por lo que el “Authentication Success Handler” maneja la petición y, finalmente, el servicio REST otorga el recurso. Si la Cookie no es válida devolverá un error de autenticación.

Spring Security tiene muchas funcionalidades y configuraciones posibles. A continuación, se procede a mostrar los métodos de la clase principal de configuración de Spring Security:

SecurityConfiguration.java

En este método se obtiene, mediante las sentencias SQL “GET_USER_AUTHENTICATION” y “GET_USER_AUTHORITY”, las credenciales del usuario de la base de datos para que Spring Security las compare con las que el usuario ha introducido y su rol para establecer los permisos de acceso.

Las sentencias tienen un único parámetro que en nuestro caso es el Email del usuario. Esto se debe a que Spring espera que los métodos “users-by-username-query” y “authorities-by-username-query” retengan como único parámetro el ‘username’. Esto puede modificarse, pero no es tarea fácil.

```

/* AUTHENTICATION */
public static final String GET_USER_AUTHENTICATION =
    "SELECT Email,Password,Enabled FROM USUARIOS WHERE Email=?";

public static final String GET_USER_AUTHORITY =
    "SELECT USUARIOS_Email,Role FROM USUARIOS_ROLES WHERE USUARIOS_email=?";

```

Figura 4.16: SQL Authentication

```

@Override
protected void configure(AuthenticationManagerBuilder builder) throws Exception {

    builder.jdbcAuthentication()
        .dataSource(dataSource)
        .usersByUsernameQuery(SqlConstants.GET_USER_AUTHENTICATION)
        .authoritiesByUsernameQuery(SqlConstants.GET_USER_AUTHORITY);

    /*
    builder.jdbcAuthentication()
        .dataSource(dataSource)
        .usersByUsernameQuery(SqlConstants.GET_USER_AUTHENTICATION2)
        .authoritiesByUsernameQuery(SqlConstants.GET_USER_AUTHORITY2);*/
}

```

Figura 4.17: Authentication Method

En este método se configuran los filtros a las distintas páginas de la aplicación y las acciones en caso de acceso permitido o no.

Los dos tipos de roles para usuario son “ROLE_USER” o “USER” y “ROLE_ESPECIALISTA” o “ESPECIALISTA”.

Puede establecer los filtros para que solo accedan a ‘x’ URL un tipo específico de rol, o todos los autenticados, o todos en general, etc.

```
@Override
protected void configure(HttpSecurity http) throws Exception {
    // Las reglas van de más específicas a menos específicas
    // antMatchers() indica el patrón sobre el que se aplicará la regla
    http.authorizeRequests().antMatchers("/").permitAll();
    http.authorizeRequests().antMatchers("/RutinaRegister/**").authenticated();
    http.authorizeRequests().antMatchers("/login.html", "/SMSLogin.html").permitAll();
    http.authorizeRequests().antMatchers("/WelcomeRutinaApp.html").authenticated();
    http.authorizeRequests().antMatchers("/usuario/**").hasRole("USER");
    http.authorizeRequests().antMatchers("/especialista/**").hasRole("ESPECIALISTA");
    http.authorizeRequests().antMatchers("/*.html").authenticated();

    //AÑADIR PÁGINAS PARA "Acceso denegado"
    //Probar a ver si Spring redirige a las páginas de acceso denegado y not found
    http.exceptionHandling().accessDeniedPage("/error.html");

    http.authorizeRequests().antMatchers("/Rutina_app/**").authenticated();

    http.exceptionHandling().authenticationEntryPoint(authenticationEntryPoint);

    http.formLogin().successHandler(authenticationSuccessHandler);
    http.formLogin().failureHandler(authenticationFailureHandler);
    http.logout().logoutSuccessUrl("/login.html");

    http.csrf().disable();
    // CSRF tokens handling
    http.addFilterAfter(new CsrfTokenResponseHeaderBindingFilter(), CsrfFilter.class);
}
```

Figura 4.18: Filter Method

Y finalmente, mostramos los métodos de las clases RESTAuthenticationSuccessHandler, RESTAuthenticationFailureHandler, RESTAuthenticationEntryPoint:

```
@Component
public class RESTAuthenticationSuccessHandler extends SimpleUrlAuthenticationSuccessHandler {

    @Override
    public void onAuthenticationSuccess(HttpServletRequest request, HttpServletResponse response,
        Authentication authentication) throws IOException, ServletException {

        clearAuthenticationAttributes(request);
    }
}
```

Figura 4.19: RESTAuthenticationSuccessHandler

```
@Component
public class RESTAuthenticationFailureHandler extends
    SimpleUrlAuthenticationFailureHandler {

    @Override
    public void onAuthenticationFailure(HttpServletRequest request,
        HttpServletResponse response,
        AuthenticationException exception) throws IOException,
        ServletException {
        super.onAuthenticationFailure(request, response, exception);
    }
}
```

Figura 4.20: RESTAuthenticationFailureHandler

```
@Component
public class RESTAuthenticationEntryPoint implements AuthenticationEntryPoint {

    @Override
    public void commence(HttpServletRequest request,
        HttpServletResponse response,
        AuthenticationException authException)
        throws IOException, ServletException {
        response.sendError(HttpServletResponse.SC_UNAUTHORIZED,
            "No tiene autorización para acceder a este contenido");
    }
}
```

Figura 4.21: RESTAuthenticationEntryPoint

4.7 Diagrama ER de la BD

Hemos tomado de base la BD del trabajo de Francisco José Díaz Romero, a la que le hemos realizado varias modificaciones.

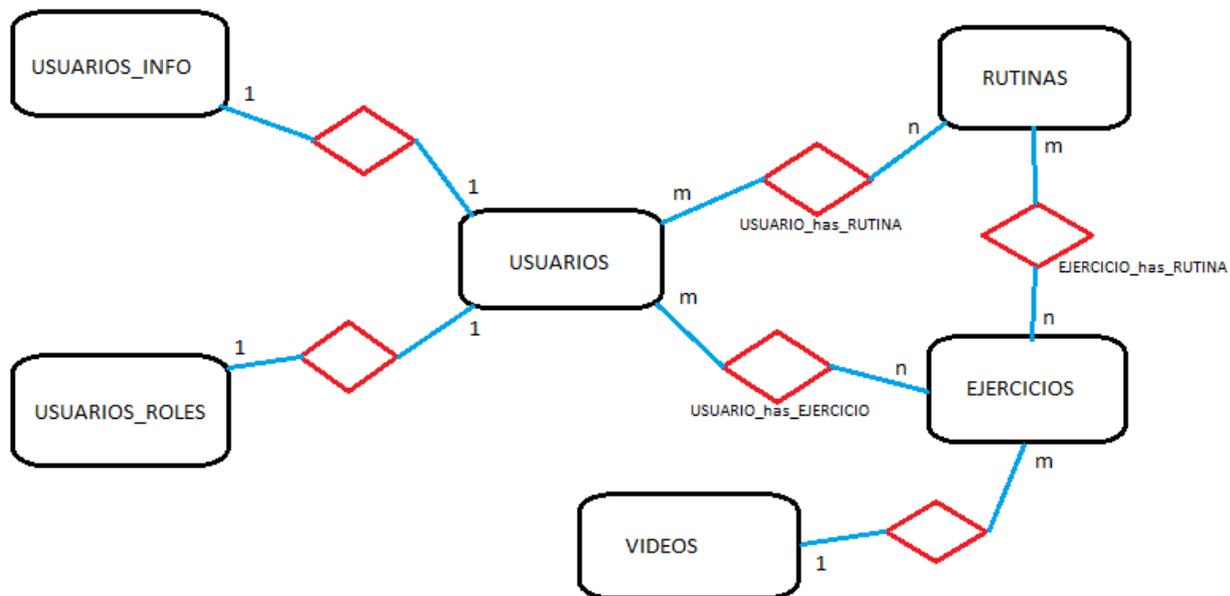


Figura 4.22: Diagrama ER

4.3.1 Detalle de los elementos del diagrama ER

4.3.1.1 Tabla USUARIOS

Descripción	Almacena la dupla Email/Contraseña de los usuarios utilizada para la autenticación en el proceso de login por Spring.
Email	(Cadena de caracteres) Email del usuario. Actúa como identificador único del mismo. Clave primaria de esta tabla y del resto que poseen un campo "Email" ya que son claves foráneas de este. No nulo
Password	(Cadena de caracteres) Contraseña del usuario. No nulo
Enabled	Es un valor booleano que indica si el usuario está activo o no en el sistema. Lo usa Spring Security para su sistema de autenticación. No nulo.
Telefono	(Cadena de caracteres) N° de teléfono móvil del usuario. Clave primaria. Empleado para el login por SMS. No nulo

4.3.1.2 Tabla USUARIOS_INFO

<i>Descripción</i>	<i>Almacena la información personal de todos aquellos usuarios dados de alta en la aplicación.</i>
USUARIOS_Email	<i>(Cadena de caracteres) Clave foránea del campo "Email" de la tabla USUARIOS. Clave primaria de esta tabla. No nulo.</i>
Nombre	<i>(Cadena de caracteres) Nombre del usuario.</i>
Telefono	<i>(Cadena de caracteres) Clave foránea del campo "Telefono" de la tabla USUARIOS. No Nulo</i>
Fecha_Nacimiento	<i>(Date) Fecha de nacimiento del usuario.</i>

4.3.1.3 Tabla USUARIOS_ROLES

<i>Descripción</i>	<i>Almacena el rol ligado a cada usuario de la aplicación. Dicha tabla es usada por Spring para el proceso de autenticación.</i>
USUARIOS_Email	<i>(Cadena de caracteres) Clave foránea del campo "Email" de la tabla USUARIOS. Clave primaria de esta tabla. No nulo.</i>
Role	<i>(Cadena de caracteres) Rol que desempeña el usuario en la aplicación. Utilizado por Spring para configuración de permisos.</i>

4.3.1.4 Tabla RUTINA

<i>Descripción</i>	<i>Almacena la información referente a las rutinas</i>
rut_id	<i>(Entero) Identificador único de rutina. Clave primaria. Autoincremental. No nulo</i>
Nombre	<i>(Cadena de caracteres) Nombre de la rutina</i>
Descripción	<i>(Cadena de caracteres) Descripción de la rutina.</i>

Info_Rutina	<i>(Cadena de caracteres) Nombre del creador de la rutina</i>
Pub_priv	<i>Es un valor booleano que indica si las rutinas a mostrar son las propias del usuario que ha accedido a la aplicación o las del resto.</i>
USUARIOS_Email	<i>(Cadena de caracteres) Clave foránea del campo "Email" de la tabla USUARIOS. Email del creador de la rutina. Clave primaria. No nulo</i>

4.3.1.5 Tabla EJERCICIO

Descripción	<i>Almacena la información referente a los ejercicios</i>
ej_id	<i>(Entero) Identificador único de ejercicio. Clave primaria. Autoincremental. No nulo</i>
Nombre	<i>(Cadena de caracteres) Nombre del ejercicio</i>
Título	<i>(Cadena de caracteres) Nombre del ejercicio.</i>
Subtítulo	<i>(Cadena de caracteres) Nombre del creador del ejercicio.</i>
Descripción	<i>(Cadena de caracteres) Descripción del ejercicio.</i>
Estado_forma	<i>(Cadena de caracteres) Estado de forma del paciente.</i>
Repeticiones	<i>(Entero) N° de veces a realizar el ejercicio</i>
Pub_priv	<i>Es un valor booleano que indica si los ejercicios a mostrar son los propios del usuario que ha accedido a la aplicación o los del resto.</i>
RUTINA_USUARIOS_Email	<i>(Cadena de caracteres) Clave foránea del campo "Email" de la tabla USUARIOS. Email del creador del ejercicio. Clave primaria. No nulo</i>

4.3.1.6 Tabla EJERCICIO_has_RUTINA

Descripción	<i>Almacena los identificadores de los ejercicios que se añaden a una rutina y el identificador de esta. La asociación entre ejercicios y rutinas es “de muchos a muchos”, ya que un ejercicio puede estar en varias rutinas y una rutina puede tener varios ejercicios.</i>
EJERCICIO_ej_id	<i>(Entero) Identificador único de ejercicio. Clave foránea del campo “ej_id” de la tabla EJERCICIOS. Clave primaria. No nulo</i>
RUTINA_rut_id	<i>(Entero) Identificador único de rutina. Clave foránea del campo “rut_id” de la tabla RUTINAS. Clave primaria. No nulo</i>
USUARIOS_Email	<i>(Cadena de caracteres) Email del usuario que añade el ejercicio a la rutina</i>

4.3.1.7 Tabla USUARIO_has_RUTINA

Descripción	<i>Almacena los identificadores de las rutinas que se asocian a un usuario y el identificador de este. La asociación entre usuarios y rutinas es “de muchos a muchos”, ya que una rutina la pueden tener varios usuarios y un usuario puede tener varias rutinas.</i>
usuario_email	<i>(Cadena de caracteres) Clave foránea del campo “Email” de la tabla USUARIOS. No nulo.</i>
rutina_id	<i>(Entero) Clave foránea del campo “rut_id” de la tabla RUTINAS. No nulo</i>

4.3.1.8 Tabla USUARIO_has_EJERCICIO

Descripción	<i>Almacena los identificadores de los ejercicios que se asocian a un usuario y el identificador de este. La asociación entre usuarios y ejercicios es “de muchos a muchos”, ya que un ejercicio lo pueden tener varios usuarios y un usuario puede tener varios ejercicios.</i>
usuario_email	<i>(Cadena de caracteres) Clave foránea del campo “Email” de la tabla USUARIOS. No nulo.</i>
ejercicio_id	<i>(Entero) Clave foránea del campo “ej_id” de la tabla EJERCICIOS. No nulo</i>

4.4.1.9 Tabla VIDEO

<i>Descripción</i>	Almacena las URLs de los videos de Youtube asociados a los ejercicios y los identificadores de estos.
Nombre	<i>(Cadena de caracteres) Nombre del video.</i>
URL	<i>(Cadena de caracteres) URL del video.</i>
VIDEOS_EJERCICIOS_ej_id	<i>(Entero) Clave foránea del campo “ej_id” de la tabla EJERCICIOS. Clave primaria. No nulo</i>
USUARIOS_Email	<i>(Cadena de caracteres) Email del usuario que añade el video al ejercicio.</i>

5 INTERFAZ DE USUARIO Y FUNCIONALIDAD

5.1 Introducción

En este apartado se va a proceder a mostrar dos de los aspectos más importantes para el usuario como son la interfaz gráfica y la funcionalidad de la aplicación. Se ha intentado crear una interfaz gráfica amigable al usuario, intuitiva y fácil de comprender, así como implantar diversas funciones bastante útiles en esta. Las funciones que puede realizar cada usuario difieren según el rol que posean dentro de la aplicación.

5.2 Autenticación o Inicio de Sesión

El usuario tras acceder a la página de inicio de sesión deberá acreditarse con sus credenciales, que en este caso son su correo y su contraseña. El usuario tendrá la opción de autenticarse mediante SMS pulsando sobre el enlace “Regístrate por SMS”.

También, desde aquí, podrá descargarse la app para móvil de MyTraining.

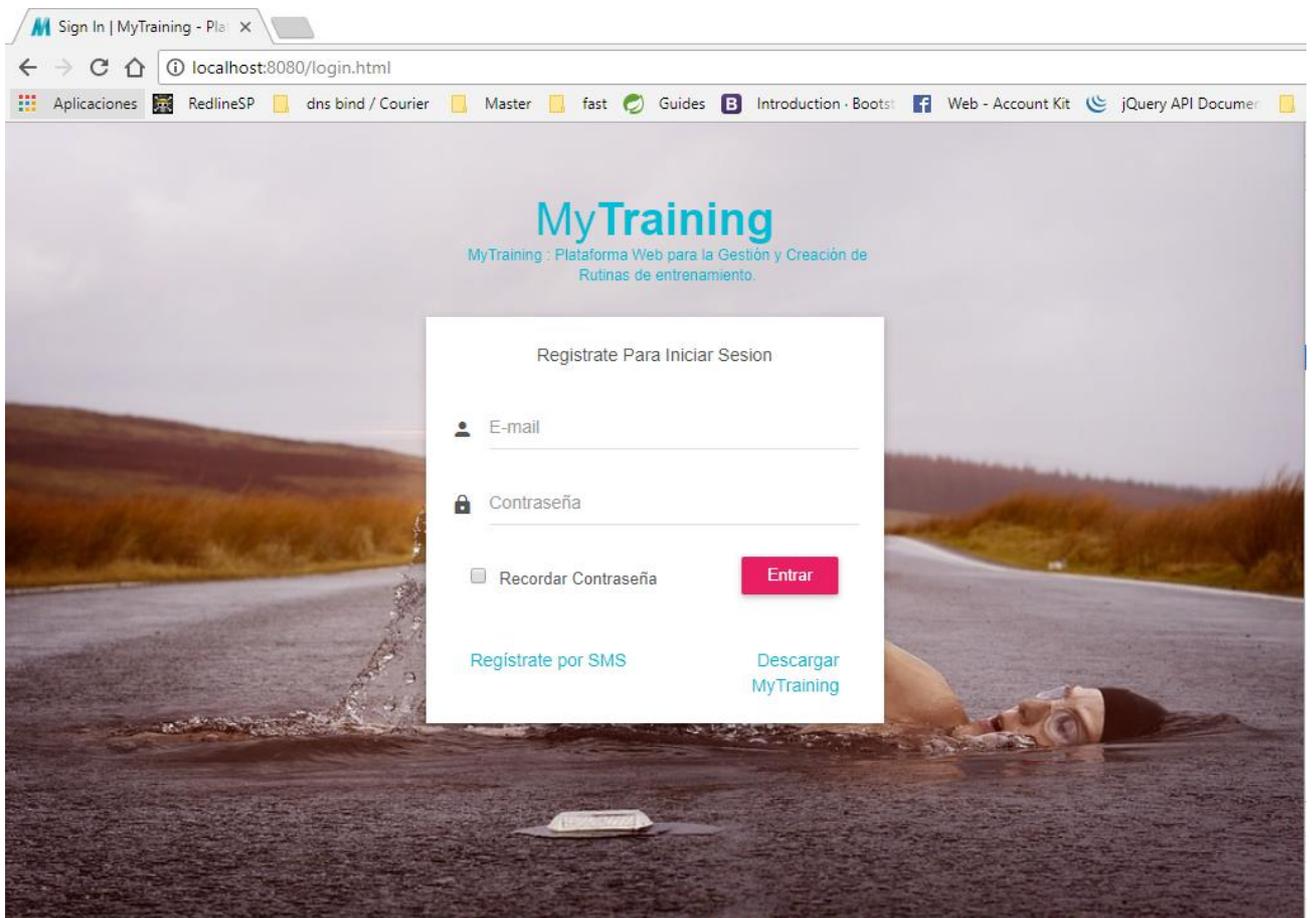


Figura 5.1: Página de Inicio de Sesión.

En el inicio de sesión por SMS el usuario tendrá primero que introducir su correo electrónico y teléfono móvil y acto seguido aparecerá una ventana emergente que solicitará que confirme su n° de teléfono móvil. Una vez introducido, la página solicitará un código el cual habrá sido enviado al móvil correspondiente. Dicho código es producido de forma aleatoria. Tras introducir dicho código, si el correo electrónico y el n° de teléfono coinciden con el mismo usuario, el usuario será redirigido a la página principal.

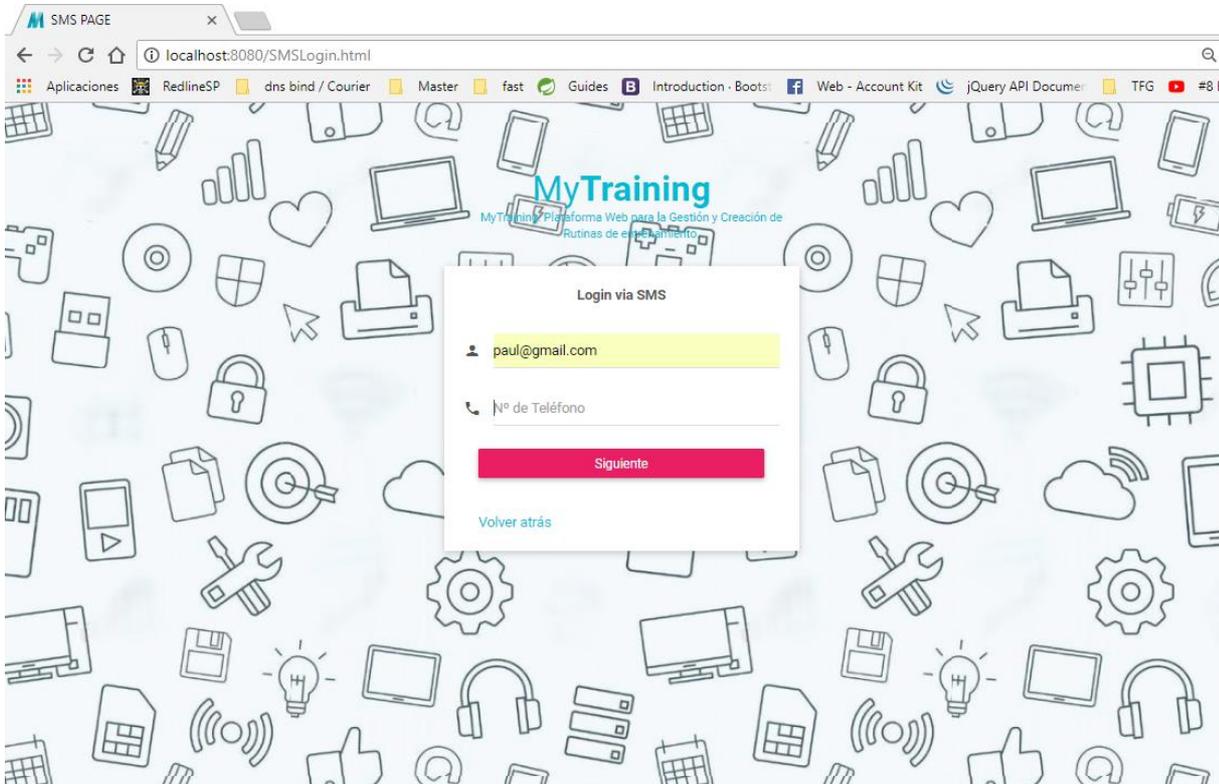


Figura 5.2: Página SMS.

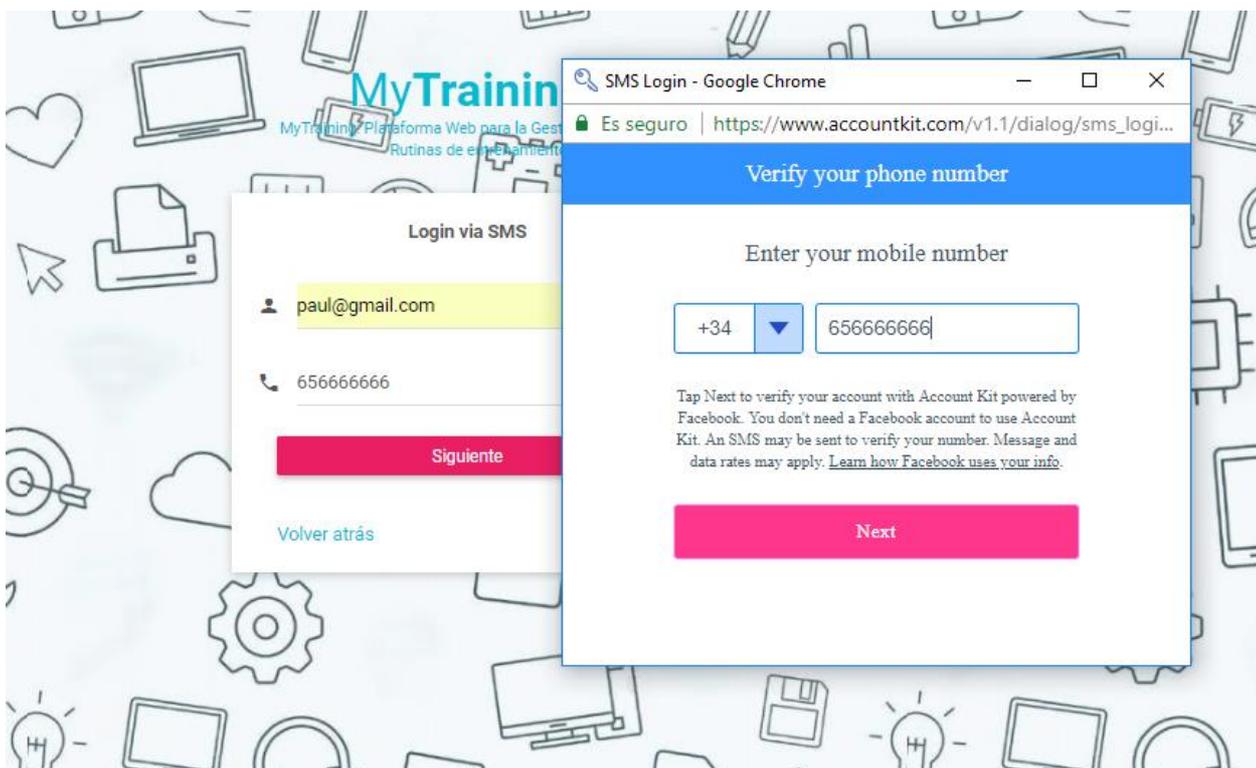


Figura 5.3: Verificación teléfono.

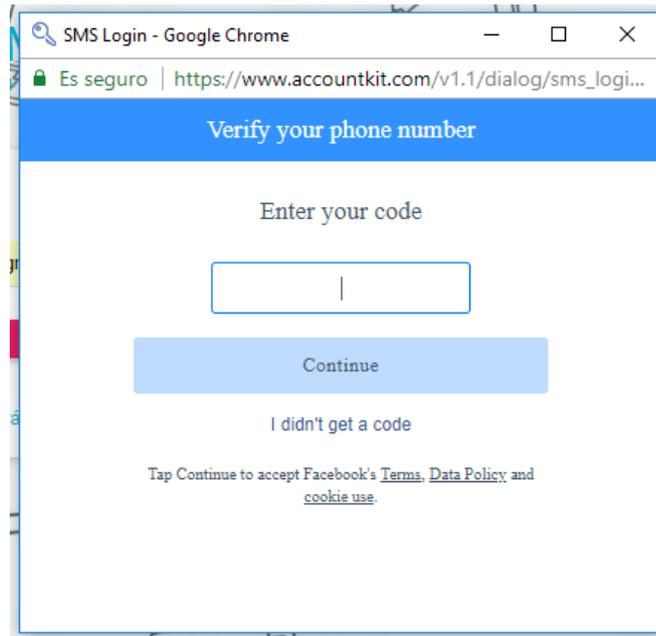


Figura 5.4: Código SMS.



Figura 5.5: Mensaje Código.

Una vez logueado, el usuario, según su rol, será redireccionado a la página principal de la aplicación.

Para que un usuario pueda acceder a la aplicación a través de la página de inicio de sesión debe primero estar registrado. Existen dos tipos de usuarios, los usuarios con rol 'User', y los usuarios con rol 'Especialista'. Estos últimos son los encargados de registrar usuarios de rol 'User', y a los usuarios de rol 'Especialista' los registra el administrador de la aplicación.

Si el usuario introduce credenciales incorrectas, saltará el siguiente mensaje.

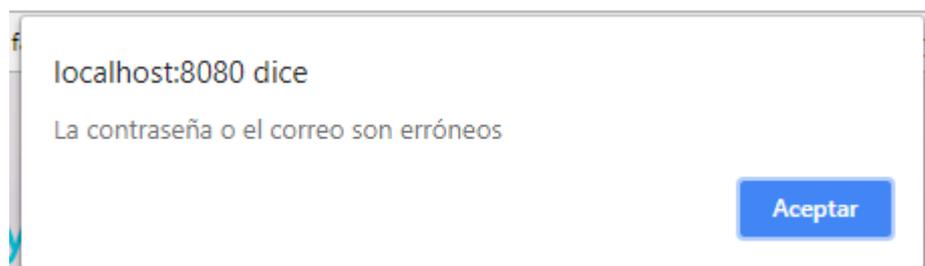


Figura 5.6: Mensaje de Aviso de Credenciales Introducidas Erróneas.

5.3 Vistas y funciones de la web

A continuación, procederemos a mostrar las diferentes vistas de la aplicación:

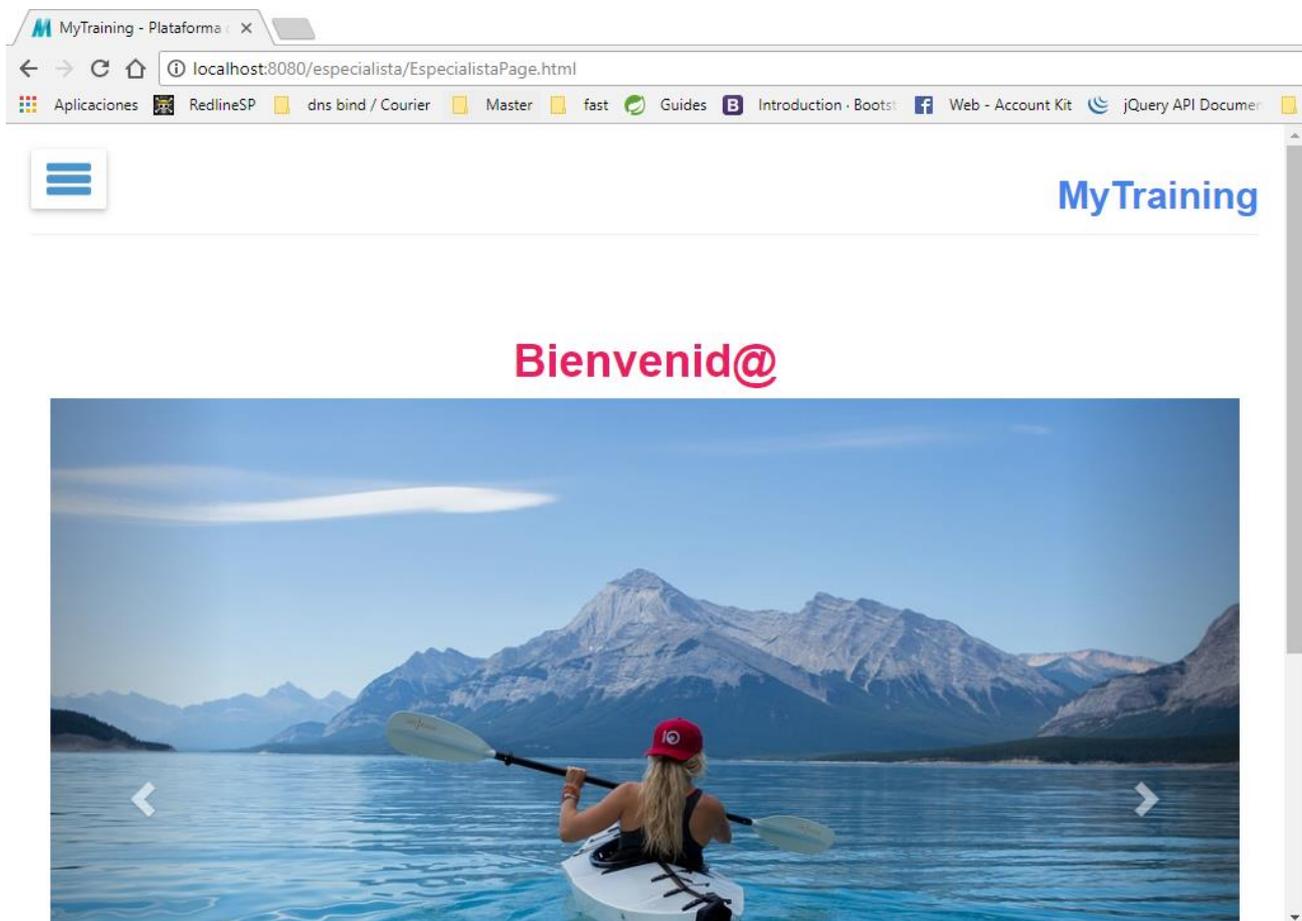


Figura 5.7: Página principal.

Este es el aspecto de la página principal para ambos usuarios. En la parte central podemos ver un slideshow con diversas fotografías relacionadas con el deporte. En la esquina superior izquierda hay un botón cuya función es desplegar un menú lateral que difiere según el rol del usuario. Dicho menú solo se encontrará oculto inicialmente en la página principal, en el resto de páginas podremos encontrarlo ya desplegado con oportunidad de ocultarlo.

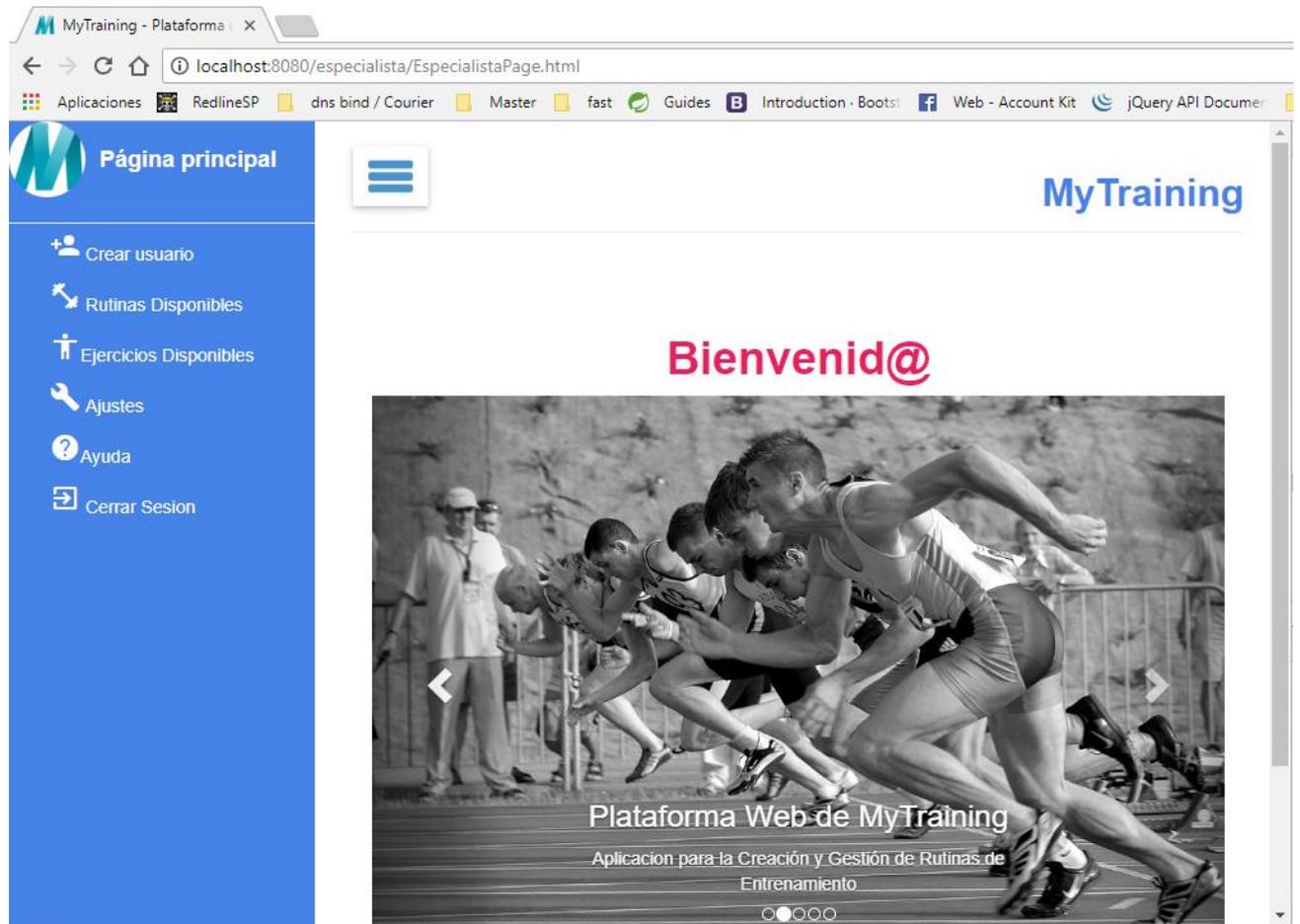


Figura 5.8: Página principal con menú lateral desplegado de Usuario Especialista.

En la imagen anterior se muestran las opciones del menú lateral de un usuario Especialista. Si pinchamos sobre “Página principal” desde cualquier pestaña seremos redirigidos a la página de Bienvenida que se muestra.

Procedemos a ver cada una de las pestañas.

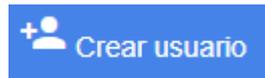


Figura 5.9: Pestaña 'Crear usuario'.

Una captura de pantalla de un navegador web. La barra de direcciones muestra "localhost:8080/especialista/UserAdd.html". El encabezado de la página dice "MyTraining" y "A continuación introduzca los datos personales, que serán utilizados por MyTraining España para crear un nuevo usuario." El formulario "Registro de Nuevo Usuario" contiene los siguientes campos: "Nombre Apellidos" (con ícono de persona), "Email" (con ícono de correo), "dd/mm/aaaa" (con ícono de calendario), "Numero de Telefono" (con ícono de teléfono), "Contraseña" (con ícono de candado) y "Confirmar contraseña" (con ícono de candado). Debajo del formulario hay un botón rojo "Confirmar" y un enlace azul "Ir a la página de inicio".

Figura 5.10: Página 'Crear usuario'

En esta página el usuario Especialista creará un nuevo usuario a partir de los datos del mismo al que vaya a dar de alta en la aplicación.



Figura 5.11: Usuario creado



Figura 5.12: Pestaña ‘Rutinas Disponibles’

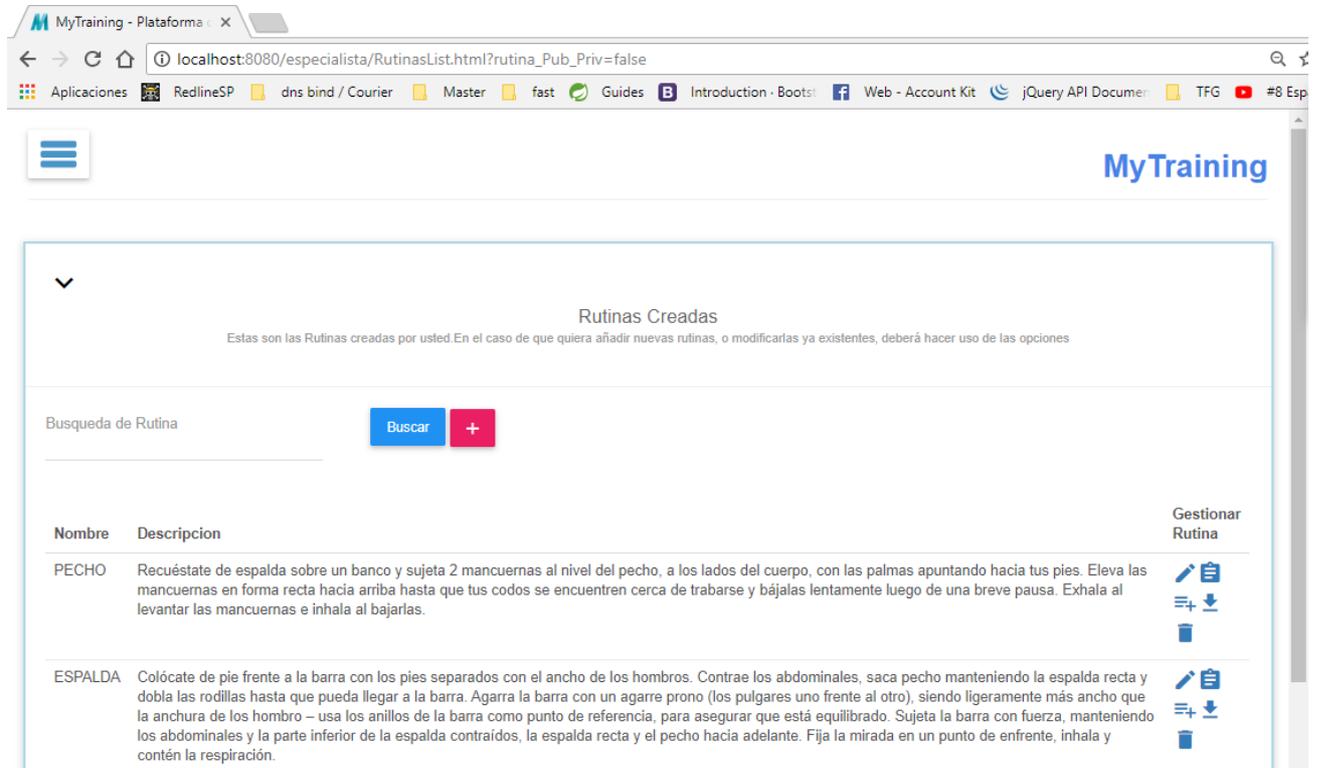


Figura 5.13: Página ‘Rutinas Creadas’

En esta página, el usuario Especialista puede ver las rutinas que él mismo ha creado y las que han creado el resto de Especialistas mediante . Dichas rutinas están compuestas de un nombre y una descripción.

El usuario Especialista puede modificar la rutina , añadir ejercicios a dicha rutina , ver los ejercicios que posee la rutina , descargar un json con la información de la rutina y sus ejercicios y eliminar la rutina .

Las rutinas pueden buscarse a través de su nombre mediante el cuadro de búsqueda

Busqueda de Rutina

Buscar

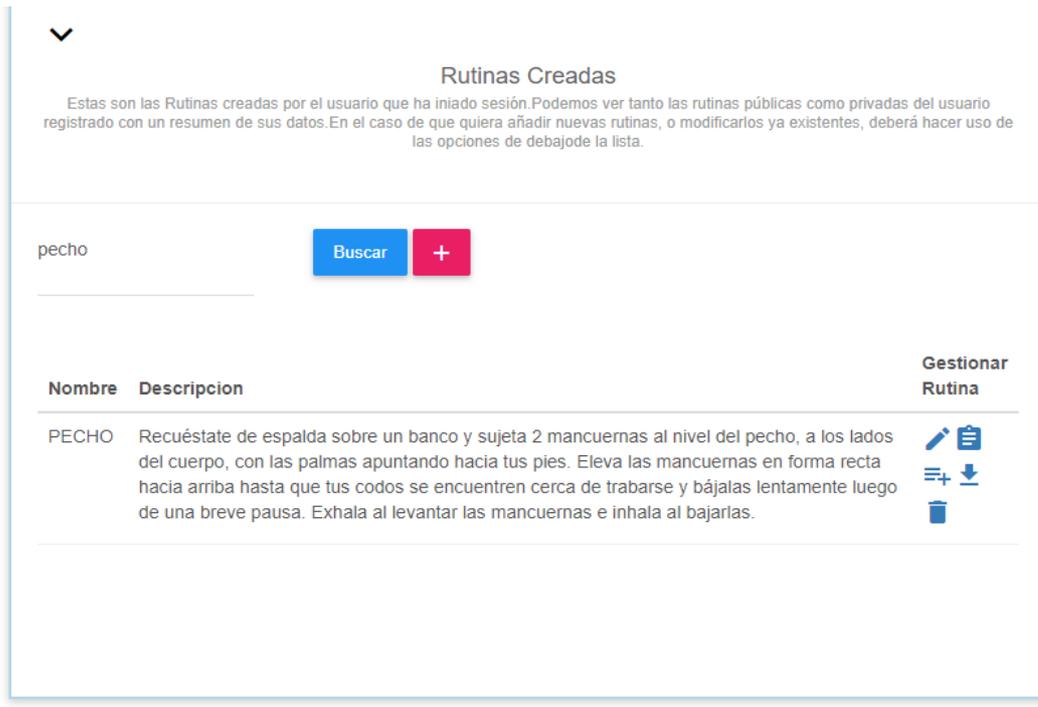


Figura 5.14: Página ‘Rutinas Creadas’ con búsqueda

Y pueden añadirse nuevas mediante el botón

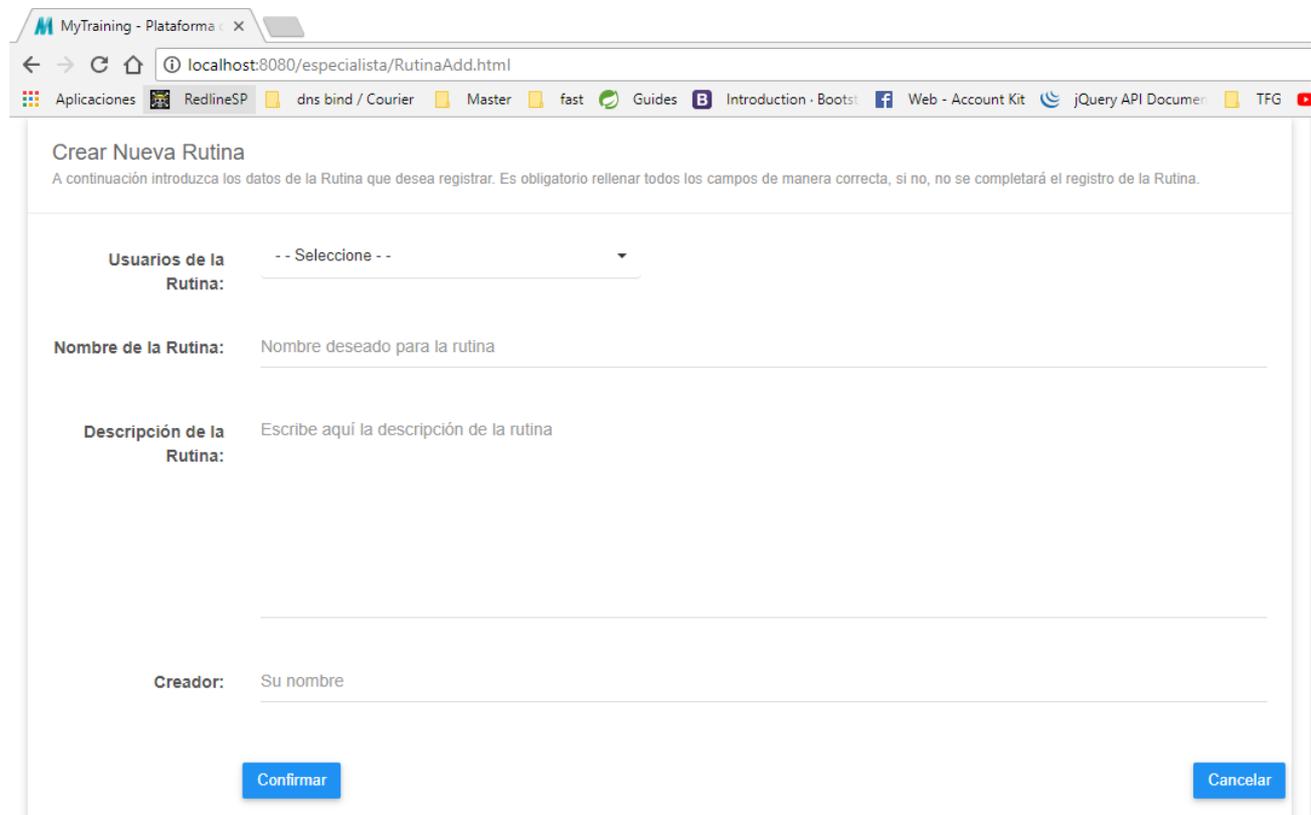


Figura 5.15: Página ‘Añadir rutina’

Campos de la página ‘Añadir rutina’

- **Usuario de la rutina:** Lista desplegable de usuarios a elegir a los que se les quiere asignar dicha rutina (Los Especialistas no aparecen en la lista).

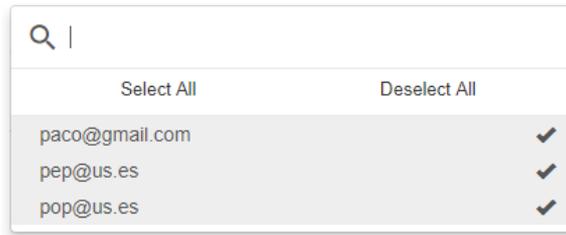


Figura 5.15.1: Lista usuarios

- **Nombre de la rutina**
- **Descripción de la rutina**
- **Creador:** El propio nombre del Especialista que la crea.

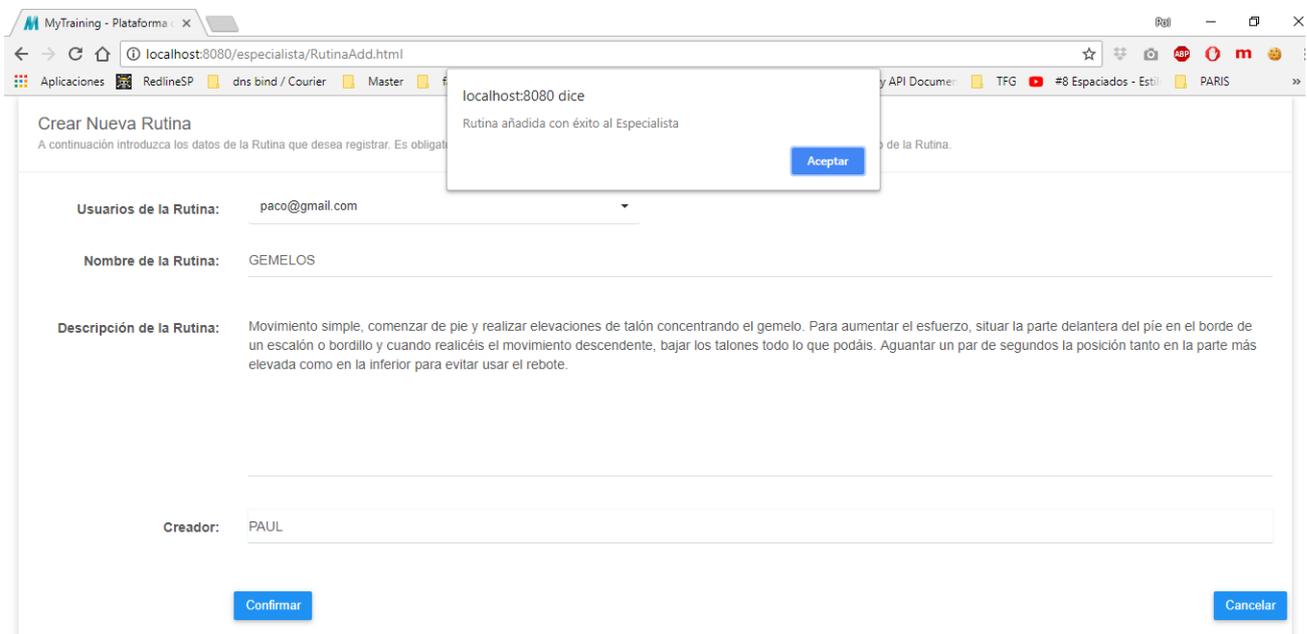


Figura 5.16: Rutina añadida

Para acceder a las rutinas del resto de Especialistas solo hay que utilizar el enlace correspondiente en la lista desplegable.

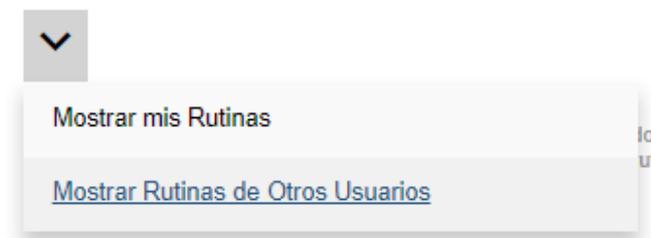


Figura 5.17: Rutinas de otros usuarios

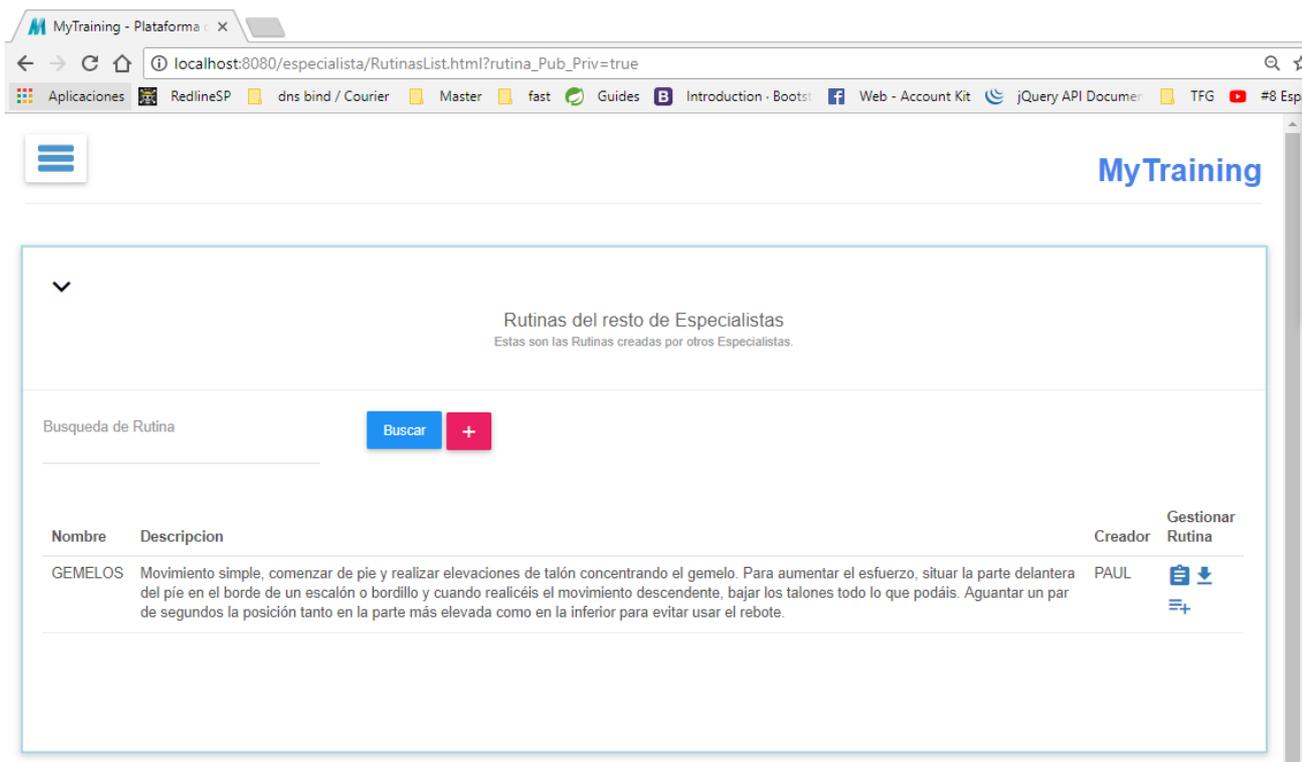


Figura 5.18: Página Rutinas de otros usuarios

La vista de las rutinas de otros usuarios Especialistas es similar a la de las propias, exceptuando que, en este caso, existe una columna “Creador” que indica quién es el dueño de la rutina, así como que en este apartado las rutinas no pueden modificarse ni eliminarse, ya que eso solo concierne al creador de estas. Sin embargo, cualquier usuario Especialista puede añadir ejercicios a cualquier rutina. Dichas rutinas pueden buscarse a través de su nombre o el nombre del creador mediante el cuadro de búsqueda

Ahora se procederán a mostrar el resto de acciones sobre rutinas mencionadas anteriormente.

Modificar rutina

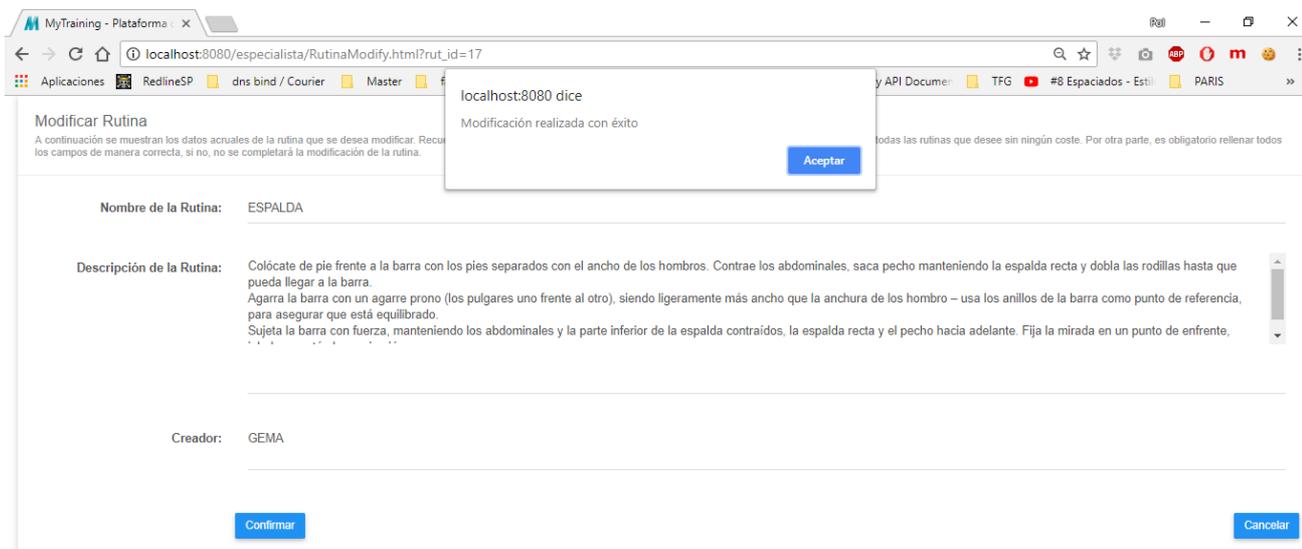


Figura 5.19: Pestaña Modificar Rutina

 Añadir ejercicio a rutina

Para entender mejor este apartado, primero explicaremos la pestaña relacionada con los ejercicios.

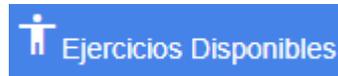


Figura 5.20: Pestaña Ejercicios Disponibles

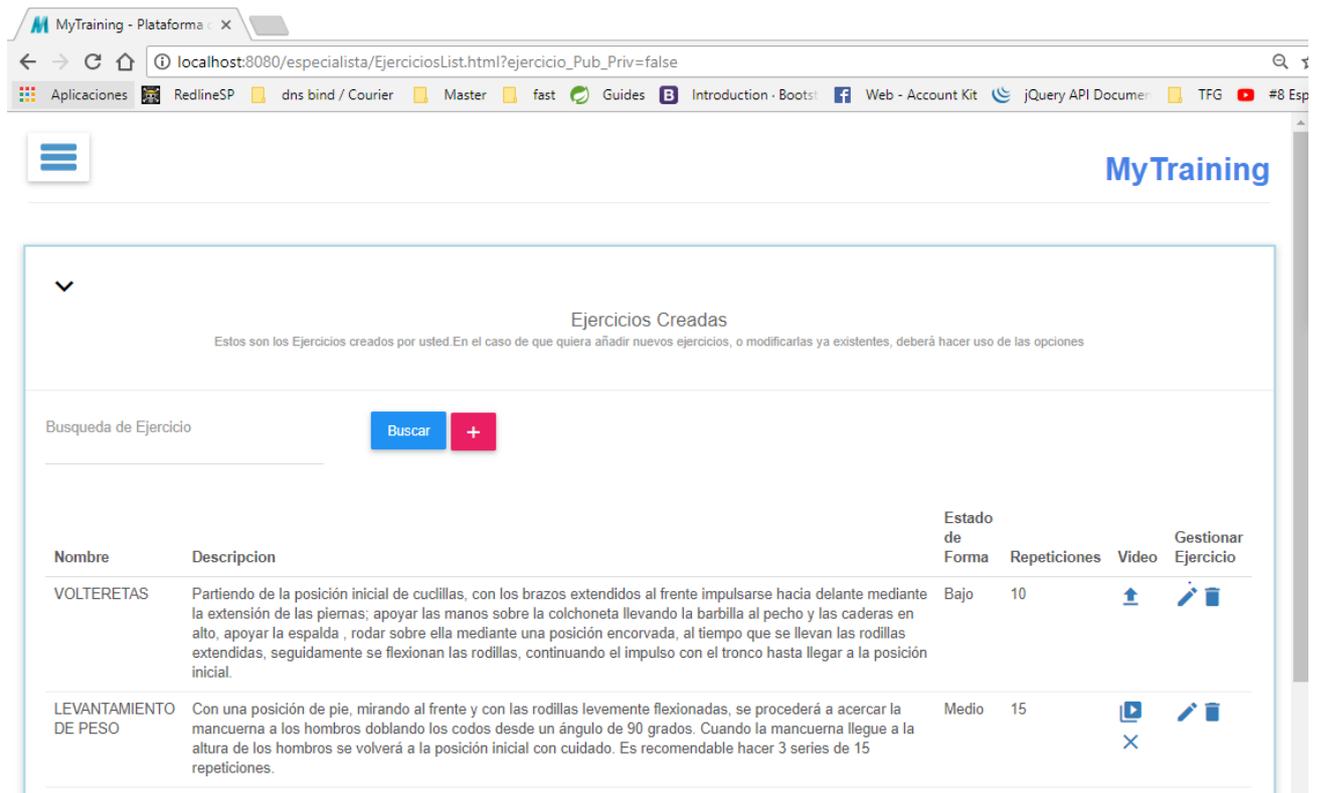


Figura 5.21: Página Ejercicios Creados

En esta página, el usuario Especialista puede ver los ejercicios que él mismo ha creado y los que han creado el resto de Especialistas mediante . Dichos ejercicios están compuestos de un nombre, una descripción, el estado de forma del usuario al que se le ha asignado el ejercicio y las repeticiones que debe realizar del ejercicio.

El usuario Especialista puede modificar el ejercicio , añadir un video al ejercicio  (en el caso de que no posea ya uno) y eliminar el ejercicio . En el caso de que el ejercicio posea un video, este video puede ser visualizado  y/o eliminado .

Al igual que en las rutinas, pueden añadirse nuevos mediante el botón 

Busqueda de Ejercicio

Y buscarlos

Crear Nuevo Ejercicio

A continuación introduzca los de un ejercicio que desea registrar. Recuerde que el hecho de introducir ejercicios es completamente gratuito, por lo que puede añadir todos los ejercicios que desee sin ningún coste. Por otra parte, es obligatorio rellenar todos los campos de manera correcta, si no, no se registrará el evento.

Usuario del ejercicio: -- Seleccione --

Nombre del Ejercicio: Nombre deseado para el ejercicio

Creador: Su nombre

Descripción del Ejercicio: Escribe aquí la descripción del ejercicio

Estado de Forma: -- Seleccione una opción --

Repeticiones del Ejercicio: Nº de repeticiones

Confirmar Cancelar

Figura 5.22: Página Añadir Ejercicio

Campos de la página ‘Añadir ejercicio’

- **Usuario de la rutina:** Lista desplegable de usuarios a elegir a los que se les quiere asignar dicho ejercicio (Los Especialistas no aparecen en la lista).
- **Nombre del ejercicio**
- **Creador:** El propio nombre del Especialista que la crea.
- **Descripción del ejercicio**
- **Estado de forma:** Estado de forma del usuario del ejercicio
- **Repeticiones del ejercicio**



Figura 5.23: Ejercicios de otros usuarios

MyTraining - Plataforma

localhost:8080/especialista/EjerciciosList.html?ejercicio_Priv=true

Aplicaciones RedlineSP dns bind / Courier Master fast Guides Introduction - Boots Web - Account Kit jQuery API Document TFG #8 Esp

MyTraining

Ejercicios del resto de Especialistas
Estas son los Ejercicios creados por otros Especialistas.

Busqueda de Ejercicio

Nombre	Creador	Descripcion	Estado de Forma	Repeticiones	Video
BICEPS	PAUL	hhhh	Medio	9	Este ejercicio no tiene videos disponibles.

© 2018 © MyTraining España. Todos los derechos reservados.
Version: 3.0.0

Figura 5.24: Página Ejercicios de otros usuarios

La vista de los ejercicios de otros usuarios Especialistas es similar a la de los propios, exceptuando que, en este caso, existe una columna “Creador” que indica quién es el creador del ejercicio, así como que en este apartado no pueden añadirse videos a los ejercicios, y que estos no pueden modificarse ni eliminarse, ya que eso solo concierne al creador de estos. Si el ejercicio posee un video, se ofrecerá la opción de visualizarlo .

Ahora procederemos a explicar el resto de acciones a realizar sobre un ejercicio.

Modificar Ejercicio

MyTraining - Plataforma

localhost:8080/especialista/EjercicioModify.html?ej_id=11

Aplicaciones RedlineSP dns bind / Courier Master fast Guides Introduction - Boots Web - Account Kit jQuery API Document TFG #8 Espacios - Estil PARIS

Modificar Ejercicio
A continuación introduzca los datos del local que desea registrar. Recuerde que el hecho de registrar un local no garantiza su inscripción en el local. Por otra parte, es obligatorio rellenar todos los campos de texto.

Nombre del Ejercicio: VOLTERETAS

Creador: GEMA

Descripción del Ejercicio
Partiendo de la posición inicial de cucullas, con los brazos extendidos al frente impulsarse hacia delante mediante la extensión de las piernas; apoyar las manos sobre la colchoneta llevando la barbilla al pecho y las caderas en alto, apoyar la espalda, rodar sobre ella mediante una posición encorvada, al tiempo que se llevan las rodillas extendidas, seguidamente se flexionan las rodillas, continuando el impulso con el tronco hasta llegar a la posición inicial.

Estado de Forma:

Repeticiones del Ejercicio: 10

localhost:8080 dice
Modificación realizada con éxito

Figura 5.25: Página Modificar Ejercicio

Eliminar ejercicio

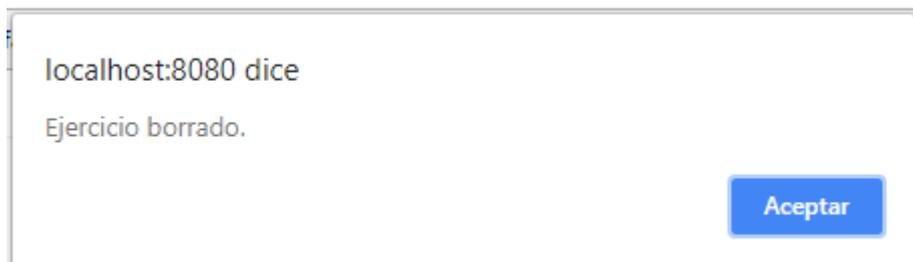
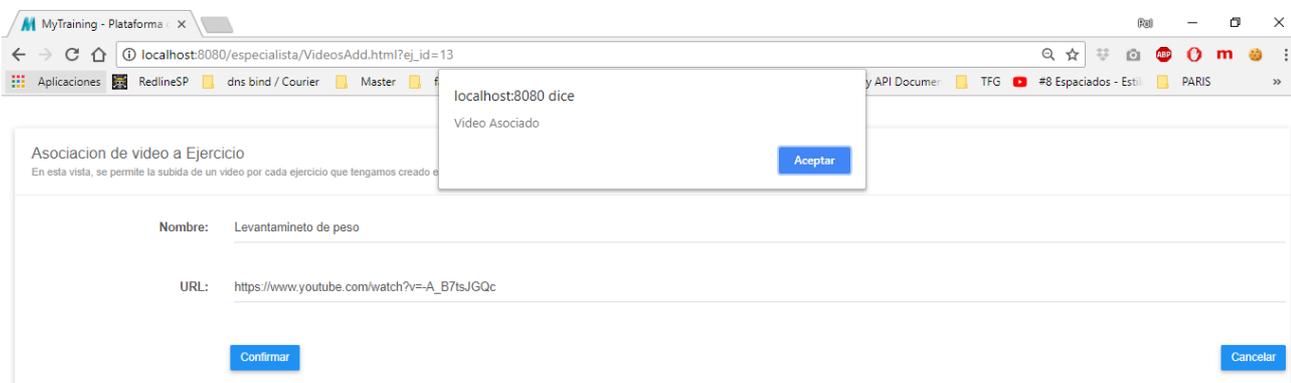


Figura 26: Eliminar Ejercicio

Añadir video

Para añadir el video, el usuario Especialista deberá rellenar el campo URL con la url perteneciente al video del ejercicio, el cuál debe estar alojado en Youtube.



© 2018 © MyTraining España. Todos los derechos reservados.
Version: 3.0.0

Figura 5.27: Página Añadir video

Ver video

En esta página podrá visualizarse el video de Youtube asociado al ejercicio. No es necesario salir de MyTraining a Youtube para ello ya que el video se encuentra embebido en la aplicación web.



© 2018 © MyTraining España. Todos los derechos reservados.
Version: 3.0.0

Figura 5.28: Página Ver video

✕ Eliminar video

MyTraining - Plataforma

localhost:8080/especialista/EjerciciosList.html?ejercicio_Pub_Priv=false

Aplicaciones RedlineSP dns bind / Courier Master localhost:8080 dice Video borrado. Aceptar

Página principal

- Crear usuario
- Rutinas Disponibles
- Ejercicios Disponibles
- Ajustes
- Ayuda
- Cerrar Sesión

MyTraining

Mis Ejercicios

Estos son los Ejercicios creados por el usuario que ha iniciado sesión. Podemos ver tanto los ejercicios públicos como privados del usuario registrado con un resumen de sus datos. En caso de que quiera añadir nuevos ejercicios, o modificar los ya existentes, deberá hacer uso de las opciones de debajo de la lista.

Busqueda de Ejercicio

Nombre	Descripción	Estado de Forma	Repeticiones	Video	Gestionar Ejercicio
VOLTERETAS	Partiendo de la posición inicial de cuclillas, con los brazos extendidos al frente impulsarse hacia delante mediante la extensión de las piernas; apoyar las manos sobre la colchoneta llevando la barbilla al pecho y las caderas en alto, apoyar la espalda, rodar sobre ella mediante una posición encorvada, al tiempo que se llevan las rodillas extendidas, seguidamente se flexionan las rodillas, continuando el impulso con el tronco hasta llegar a la posición inicial.	Bajo	10	<input checked="" type="checkbox"/>	<input type="button" value="✕"/> <input type="button" value="✎"/>
LEVANTAMIENTO DE PESO	Con una posición de pie, mirando al frente y con las rodillas levemente flexionadas, se procederá a acercar la mancuerna a los hombros doblando los codos desde un ángulo de 90 grados. Cuando la mancuerna llegue a la altura de los hombros se volverá a la posición inicial con cuidado. Es recomendable hacer 3 series de 15 repeticiones.	Medio	15	<input type="checkbox"/>	<input type="button" value="✕"/> <input type="button" value="✎"/>

Figura 5.29: Eliminar video

Una vez explicados los ejercicios y las acciones que pueden realizarse sobre ellos, retomamos las rutinas.

Añadir ejercicio a rutina

Añadir Ejercicio a Rutina

Estos son los Ejercicios disponibles para añadir a una rutina, con un resumen de sus datos. En el caso de que quiera añadir nuevos ejercicios, o modificar los ya existentes, deberá hacer uso de las opciones de debajo de la lista.

Busqueda de Rutina

Nombre	Creador	Descripción	Estado de Forma	Repeticiones	Añadir Ejercicio
VOLTERETAS	GEMA	Partiendo de la posición inicial de cuclillas, con los brazos extendidos al frente impulsarse hacia delante mediante la extensión de las piernas; apoyar las manos sobre la colchoneta llevando la barbilla al pecho y las caderas en alto, apoyar la espalda, rodar sobre ella mediante una posición encorvada, al tiempo que se llevan las rodillas extendidas, seguidamente se flexionan las rodillas, continuando el impulso con el tronco hasta llegar a la posición inicial.	Bajo	10	<input type="button" value="📎"/>
BICEPS	PAUL	hhhh	Medio	9	<input type="button" value="📎"/>
LEVANTAMIENTO DE PESO	GEMA	Con una posición de pie, mirando al frente y con las rodillas levemente flexionadas, se procederá a acercar la mancuerna a los hombros doblando los codos desde un ángulo de 90 grados. Cuando la mancuerna llegue a la altura de los hombros se volverá a la posición inicial con cuidado. Es recomendable hacer 3 series de 15 repeticiones.	Medio	15	<input type="button" value="📎"/>

Figura 5.30: Página Añadir Ejercicio a Rutina

Se muestran todos los ejercicios disponibles de todos los usuarios Especialistas para añadir. Aquellos ya añadidos a esa rutina no aparecerán en la lista.

Los ejercicios se añaden pulsando .

localhost:8080 dice
Ejercicio Asociado.

Figura 5.31: Ejercicio asociado

Si el ejercicio que se va a añadir a la rutina no tiene video aparecerá el siguiente mensaje

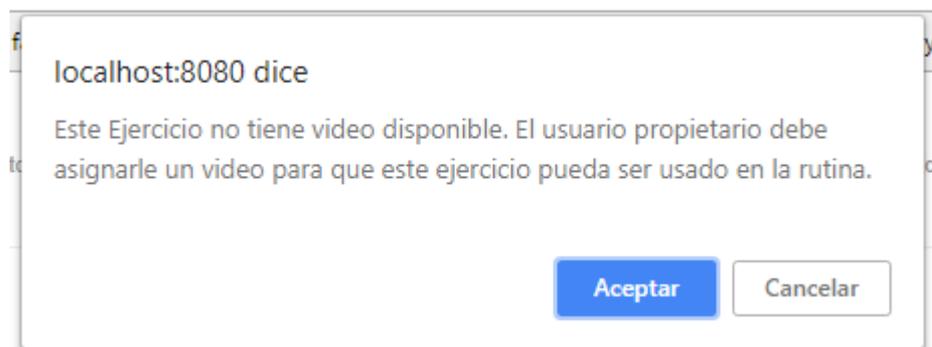


Figura 5.32: Ejercicio sin video

Ver ejercicios de rutina

Ejercicios de Rutinas

Estos son los Ejercicios que tiene una Rutina, con un resumen de sus datos. En el caso de que quiera asociar nuevos ejercicios, o modificar los ya existentes, deberá hacer uso de las opciones de debajo de la lista.

Nombre	Creador	Descripción	Estado de Forma	Repeticiones	Video	Quitar Ejercicio
LEVANTAMIENTO DE PESO	GEMA	Con una posición de pie, mirando al frente y con las rodillas levemente flexionadas, se procederá a acercar la mancuerna a los hombros doblando los codos desde un ángulo de 90 grados. Cuando la mancuerna llegue a la altura de los hombros se volverá a la posición inicial con cuidado. Es recomendable hacer 3 series de 15 repeticiones.	Medio	15		

Añadir Ejercicios Volver atrás

Figura 5.33: Página Ver Ejercicios de Rutina

Desde esta página, aparte de ver los ejercicios que tiene una rutina, así como sus datos y acceder a su vídeo asociado, se pueden quitar dichos ejercicios  y añadir nuevos .

Eliminar rutina

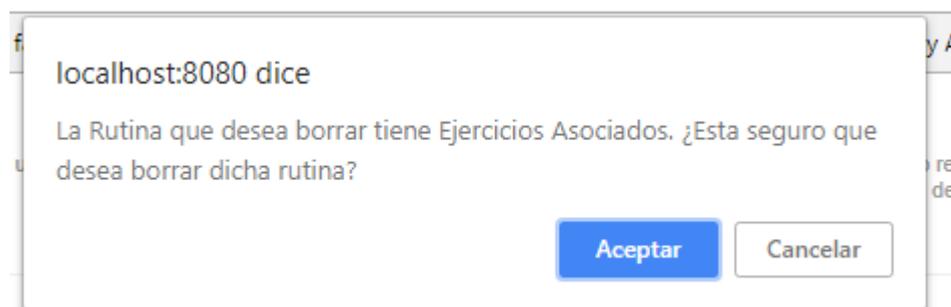


Figura 5.34: Eliminar rutina 1

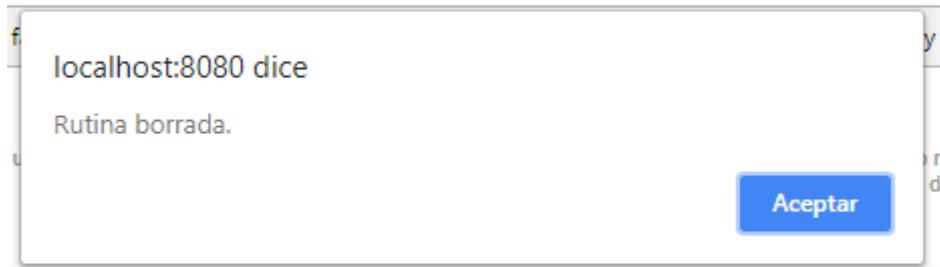


Figura 5.35: Eliminar rutina 2

↓ Descargar rutina

La descarga de la rutina en un json tiene como finalidad su uso en la aplicación móvil homóloga a esta aplicación Web.

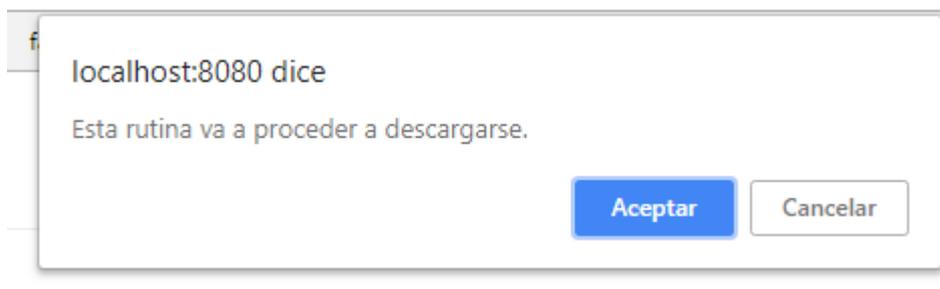


Figura 5.36: Descargar rutina

```
[{"Nombre Rutina":"PECHO","Creador":"GEMA","Descripcion Rutina":"RecuÃ©state de espalda sobre un banco y sujeta 2 mancuernas al nivel del pecho, a los lados del cuerpo, con las palmas apuntando hacia tus pies.\n\nEleva las mancuernas en forma recta hacia arriba hasta que tus codos se encuentren cerca de trabarse y bÃ¡jalas lentamente luego de una breve pausa.\n\nExhala al levantar las mancuernas e inhala al bajarlas."}, {"Informacion Ejercicios":[{"Nombre":"BICEPS","Subtitulo":"PAUL","Descripcion":"hhhh","Estado de forma":"Medio","Repeticiones":9}]}]
```

Figura 5.37: JSON rutina



Figura 5.38: Pestaña Ajustes

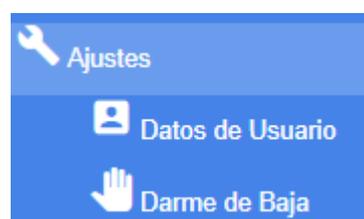


Figura 5.39: Pestaña Ajustes 2

Datos de Usuario

En esta página el usuario puede modificar sus datos personales

MyTraining - Plataformas

localhost:8080/especialista/UserMod.html

Aplicaciones RedlineSP dns bind / Courier Master fast Guides Introduction - Boots! Web - Account Kit jQuery API Document TFG #8 Espaciados - Esti PARIS

Página principal

MyTraining

Modificar mi perfil

A continuación se muestran los datos de usuario, pudiéndose modificar aquellos campos que se necesiten modificar.

E-mail: gema@gmail.com

Nombre: Gema Perez Sal

Fecha de nacimiento: 07/01/1982

Número de teléfono (opcional): 666555444

Contraseña: Contraseña

Confirmar Contraseña: Confirmar contraseña

Confirmar Cancelar

Figura 5.40: Página Datos de Usuario

Darme de Baja

En esta página el usuario podrá darse de baja borrándose de esta manera todos los datos relacionados con este de la aplicación.

Darse de baja

Si se da de baja en el servicio, dejará de disfrutar de todas las facilidades que le da Rutina App para la Gestión de Rutinas de Entrenamiento, debido a que se borrarán todos los datos relacionados con su cuenta personal. Podrá darse de alta posteriormente de nuevo, pero deberá registrar de nuevo sus datos.

Confirmar Cancelar

Figura 5.41: Página Darse de Baja

localhost:8080 dice

Perfil borrado. Baja dada con éxito.

Aceptar

Figura 5.42: Perfil borrado



En esta página se puede contactar mediante un mensaje de correo con personal que resolverá las dudas de los usuarios.

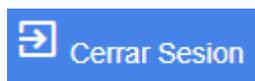
Ayuda y contacto
El precio de la ayuda en línea y otros servicios relacionados con la atención al usuario, vienen incluidos en el precio de tarifa del servicio, por lo que no tendrá que abonar ningún importe adicional. Le recomendamos que utilice dichos servicios en el caso de que necesite asistencia en cualquier aspecto relacionado con nuestra aplicación.

Mándanos un correo Cancelar

Figura 5.43: Página Ayuda

A screenshot of an email composition interface. On the left is a vertical blue sidebar with icons for menu, add, user, and folder. The main area has a top toolbar with "Formato", "Insertar", "Opciones", "Descartar", and "Enviar". Below the toolbar are text formatting options (N, K, S), list options, a subject field with "Título 1", and a "Deshacer" button. The email body shows "De: paulcarmona19@gmail.com", "Para: MyTrainingcontacto@gmail.com;" with "CC y CCO" options, and a "Tema" field.

Figura 5.44: Correo



localhost:8080 dice
Ha cerrado su sesión con éxito. Esperamos verle de nuevo.

Aceptar

Figura 5.45: Cerrar sesión

La vista del usuario es muy similar a la del Especialista, exceptuando algunas funciones que vamos a detallar:



Este es el menú lateral de un usuario. Como puede observarse, la opción de añadir un usuario no se encuentra ya que esa acción concierne al Especialista, y las pestañas de 'Ajustes' y 'Ayuda' son exactamente iguales.

Sin embargo, las pestañas de rutinas y ejercicios disponibles cambian.

Figura 5.46: Menú usuario

Un usuario solo podrá ver las rutinas y ejercicios que le asignen los Especialistas, así como descargar las rutinas, ver los ejercicios que posea cada una de ellas y los videos de los respectivos ejercicios.

MyTraining

Mis Rutinas

Estas son las Rutinas creadas por el usuario que ha iniciado sesión. Podemos ver tanto las rutinas públicas como privadas del usuario registrado con un resumen de sus datos. En el caso de que quiera añadir nuevas rutinas, o modificarlos ya existentes, deberá hacer uso de las opciones de debajo de la lista.

Busqueda de Rutina

Nombre	Descripcion	Creador	Gestionar Rutina
PECHO	Recuéstate de espalda sobre un banco y sujeta 2 mancuernas al nivel del pecho, a los lados del cuerpo, con las palmas apuntando hacia tus pies. Eleva las mancuernas en forma recta hacia arriba hasta que tus codos se encuentren cerca de trabarse y bájalas lentamente luego de una breve pausa. Exhala al levantar las mancuernas e inhala al bajarlas.	GEMA	 
ESPALDA	Colócate de pie frente a la barra con los pies separados con el ancho de los hombros. Contrae los abdominales, saca pecho manteniendo la espalda recta y dobla las rodillas hasta que pueda llegar a la barra. Agarra la barra con un agarre prono (los pulgares uno frente al otro), siendo ligeramente más ancho que la anchura de los hombro – usa los anillos de la barra como punto de referencia, para asegurar que está equilibrado. Sujeta la barra con fuerza, manteniendo los abdominales y la parte inferior de la espalda contraídos, la espalda recta y el pecho hacia adelante. Fija la mirada en un punto de enfrente, inhala y contén la respiración.	GEMA	 
GEMELOS	Movimiento simple, comenzar de pie y realizar elevaciones de talón concentrando el gemelo. Para aumentar el esfuerzo, situar la parte delantera del pie en el borde de un escalón o bordillo y cuando realicéis el movimiento descendente, bajar los talones todo lo que podáis. Aguantar un	PAUL	 

Figura 5.47: Rutinas de usuario

Ejercicios Asociados a Rutinas

Estos son los Ejercicios que tiene Asociados una Rutina, con un resumen de sus datos. En el caso de que quiera asociar nuevos ejercicios, o modificar los ya existentes, deberá hacer uso de las opciones de debajo de la lista.

Nombre	Creador	Descripcion	Estado de Forma	Repeticiones	Video
LEVANTAMIENTO DE PESO	GEMA	Con una posición de pie, mirando al frente y con las rodillas levemente flexionadas, se procederá a acercar la mancuerna a los hombros doblando los codos desde un ángulo de 90 grados. Cuando la mancuerna llegue a la altura de los hombros se volverá a la posición inicial con cuidado. Es recomendable hacer 3 series de 15 repeticiones.	Medio	15	

[Volver atrás](#)

Figura 5.48: Ejercicios de rutinas de usuario

Mis Ejercicios

Estos son los Ejercicios creados por el usuario que ha iniciado sesión. Podemos ver tanto los ejercicios públicos como privados del usuario registrado con un resumen de sus datos. En caso de que quiera añadir nuevos ejercicios, o modificar los ya existentes, deberá hacer uso de las opciones de debajo de la lista

Busqueda de Rutina [Buscar](#)

Nombre	Creador	Descripcion	Estado de Forma	Repeticiones	Video
VOLTERETAS	GEMA	Partiendo de la posición inicial de cuclillas, con los brazos extendidos al frente impulsarse hacia delante mediante la extensión de las piernas; apoyar las manos sobre la colchoneta llevando la barbilla al pecho y las caderas en alto, apoyar la espalda, rodar sobre ella mediante una posición encorvada, al tiempo que se llevan las rodillas extendidas, seguidamente se flexionan las rodillas, continuando el impulso con el tronco hasta llegar a la posición inicial.	Bajo	10	No hay video disponible
BICEPS	PAUL	hhhh	Medio	9	No hay video disponible
LEVANTAMIENTO DE PESO	GEMA	Con una posición de pie, mirando al frente y con las rodillas levemente flexionadas, se procederá a acercar la mancuerna a los hombros doblando los codos desde un ángulo de 90 grados. Cuando la mancuerna llegue a la altura de los hombros se volverá a la posición inicial con cuidado. Es recomendable	Medio	15	

Figura 5.49: Ejercicios de usuario

6 CONCLUSIONES

Finalmente, tras tiempo de dedicación a entender y aprender a utilizar las tecnologías expuestas anteriormente y tener claros los objetivos de este proyecto, se ha conseguido dar solución a estos.

Se ha logrado crear una aplicación web basada en un servicio web REST que permite autenticarse tanto de la forma habitual con el uso de una contraseña como a través del móvil. Los permisos de acceso a las distintas secciones de la web son distintos según el tipo de rol del usuario. El usuario con más privilegios es el especialista, encargado de dar de alta a los ‘pacientes’ en la web, crear, modificar, asociar y eliminar las rutinas y ejercicios relacionados con el entrenamiento individual de cada “paciente”. Además de gestionar los videos explicativos de la realización de los distintos ejercicios para ayudar a su correcta ejecución.

Los pacientes, además de recibir rutinas personalizadas para su entrenamiento, tendrán disponible asesoramiento vía Email del personal y la posibilidad de darse de baja cuando gusten.

Este proyecto ofrece la posibilidad de obtener un entrenamiento personalizado, asistido por especialistas de una forma cómoda y virtual, sin desplazamientos innecesarios ni excesivas pérdidas de tiempo. Es una manera eficaz de promover un estilo de vida saludable.

Durante el proceso de realización de este proyecto he aprendido mucho acerca de las herramientas y tecnologías empleadas. Muchas de ellas ya las conocía, pero no he tenido la oportunidad de trabajar con ellas como he hecho aquí, con lo que he podido comprender mejor su funcionamiento y adquirir más conocimientos.

En definitiva, me alegro de haber elegido este proyecto ya que uno de los temas que más me han interesado a lo largo de la carrera han sido los servicios web y programación web dinámica, los cuales han estado presentes en este proyecto.

REFERENCIAS

[1] En relación a MySQL:

[1][1] Web Oficial de MySQL: <https://www.mysql.com/>

[1][2] Documentación de XAMPP: <https://www.apachefriends.org/community.html>

[2] En relación a Spring Framework y Módulos Relacionados:

[2][1] Documentación de Spring: <https://spring.io/docs>

<http://forum.spring.io/forum/spring-projects/security>

[2][2] Guías de Spring: <https://spring.io/guides>

[2][3] API REST: <https://openehr.atlassian.net/wiki/display/spec/openEHR+REST+APIs>

<https://codezone4.wordpress.com/2012/11/08/restful-web-services-java-mysql-and-json/>

[2][4] Tutoriales de Propósito General: <http://www.mkyong.com/tutorials/>

[2][5] Tutoriales Específicos en Spring: <http://www.baeldung.com/>

[3] En relación a Java:

[3][1] General: <https://javapointers.com/>

[3][1] Creación de JSON en Java: <https://www.java2novice.com/java-json/javax.json/create-json-array/>

[4] En relación a BootStrap(HTML5, CSS3 y JS):

[4][1] Referencia del W3C de HTML5: <https://dev.w3.org/html5/html-author/>

[4][2] Guía de HTML5 en MDN: <https://developer.mozilla.org/en-US/docs/Web/Guide/HTML/HTML5>

[4][3] Tutoriales de HTML5 de W3C: <http://www.w3schools.com/html/>

[4][4] Tutoriales de CSS3 de W3C: http://www.w3schools.com/css/css3_intro.asp

[4][5] Tutoriales y Documentación de BootStrap: <https://getbootstrap.com/docs/4.0/getting-started>

[5] En relación a JavaScript y librerías relacionadas:

[5][1] Tutoriales de JavaScript de W3C: <http://www.w3schools.com/js/>

[5][2] Web Oficial de JQuery: <https://jquery.com/>

[5][3] Web Oficial de JQuery Validation Plugin: <https://jqueryvalidation.org/>

[5][4] Perfil de Github de Klaus Hartl (librerías de cookies): <https://github.com/carhartl/>

[5][5] Ajax: <https://www.webucator.com/tutorial/learn-ajax/jquery/ajax-with-jquery-reading.cfm>

[6] Account Kit de Facebook: <https://developers.facebook.com/docs/accountkit/webjs>

[7] Cuestiones varias: <https://stackoverflow.com/>

ANEXO A: MANUAL DE INSTALACIÓN Y DESPLIEGUE DE LA APLICACIÓN

En este anexo se explicará el proceso de instalación y despliegue de la aplicación sobre el sistema operativo sobre el que se ha desarrollado el proyecto, Windows 10 de 64 bits.

A.1 Instalación de Java JDK 8

1. Descargar JDK para Windows a través de la página <http://www.oracle.com/technetwork/java/javase/downloads/jdk10-downloads-4416644.html>
2. Ejecutar el .exe para instalarlo y escoger una ruta para la instalación. Dicha ruta deberemos recordarla para un paso posterior.

Java SE Development Kit 10 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

See also:

- [Java Developer Newsletter](#): From your Oracle account, select **Subscriptions**, expand **Technology**, and subscribe to Java.
- [Java Developer Day hands-on workshops \(free\) and other events](#)
- [Java Magazine](#)

JDK 10.0.1 [checksum](#)

Java SE Development Kit 10.0.1

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Accept License Agreement Decline License Agreement

Product / File Description	File Size	Download
Linux	305.97 MB	jdk-10.0.1_linux-x64_bin.rpm
Linux	338.41 MB	jdk-10.0.1_linux-x64_bin.tar.gz
macOS	395.46 MB	jdk-10.0.1_osx-x64_bin.dmg
Solaris SPARC	206.63 MB	jdk-10.0.1_solaris-sparcv9_bin.tar.gz
Windows	390.19 MB	jdk-10.0.1_windows-x64_bin.exe

Figura A.1: JDK

A.2 MAVEN

1. Descargamos el archivo .zip de Maven de la página

<https://maven.apache.org/download.cgi>

	Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.5.3-bin.tar.gz	apache-maven-3.5.3-bin.tar.gz.sha1	apache-maven-3.5.3-bin.tar.gz.asc
Binary zip archive	apache-maven-3.5.3-bin.zip	apache-maven-3.5.3-bin.zip.sha1	apache-maven-3.5.3-bin.zip.asc
Source tar.gz archive	apache-maven-3.5.3-src.tar.gz	apache-maven-3.5.3-src.tar.gz.sha1	apache-maven-3.5.3-src.tar.gz.asc
Source zip archive	apache-maven-3.5.3-src.zip	apache-maven-3.5.3-src.zip.sha1	apache-maven-3.5.3-src.zip.asc

Figura A.2: MAVEN

2. Lo descomprimos con WinRAR o similar y lo ubicamos en la ruta que escogamos, a ser preferible una cercana a la raíz C:\, que tendremos que recordar más adelante.

A.3 Creación de variables de entorno

En caso de no tener configuradas las variables de entorno necesarias para que Maven funcione correctamente, habrá que seguir estos pasos:

1. Acceder a “Este equipo->Configuración avanzada del sistema->variables de entorno

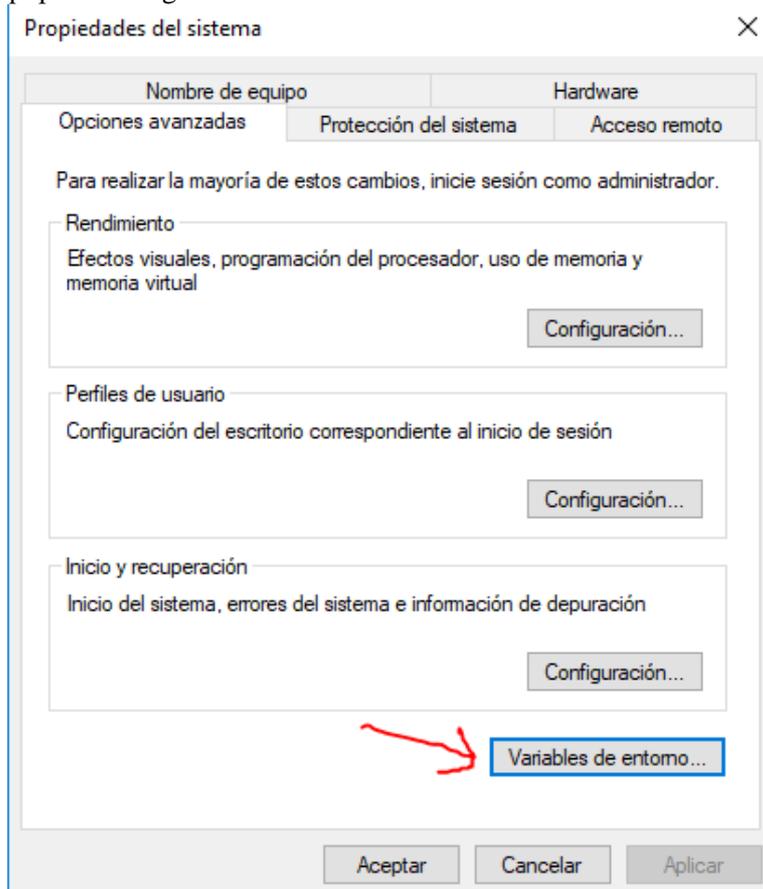


Figura A.3: Variables de entorno

2. Ahora necesitaremos crear la variable de entorno ‘JAVA_HOME’ que contendrá la ruta que escogimos al instalar el JDK.
3. Incluimos la ruta de ‘JAVA_HOME’ y la ruta de Maven a la variable de entorno ‘Path’. A la ruta de Maven deberá añadirse \bin, es decir, ‘tu_ruta_de_maven\bin’.

A.4 Comprobación en cmd e instalación

Abrimos la consola de Windows y ejecutamos el comando “mvn -version”. Si la salida es parecida a la siguiente todo está correcto.

```
C:\WINDOWS\system32>mvn -version
Apache Maven 3.5.2 (138edd61fd100ec658bfa2d307c43b76940a5d7d; 2017-10-18T09:58:13+02:00)
Maven home: C:\Program Files\apache-maven-3.5.2\bin\..
Java version: 1.8.0_161, vendor: Oracle Corporation
Java home: C:\Program Files\Java\jdk1.8.0_161\jre
Default locale: es_ES, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"
```

Figura A.4: versión Maven/java

Acto seguido ejecutamos “mvn install” para que se seintale y configure nuestro repositorio local.

A.5 Instalación de Spring Tools Suite (STS)

Descargamos el zip correspondiente de la página de Spring <https://spring.io/tools/sts/all>

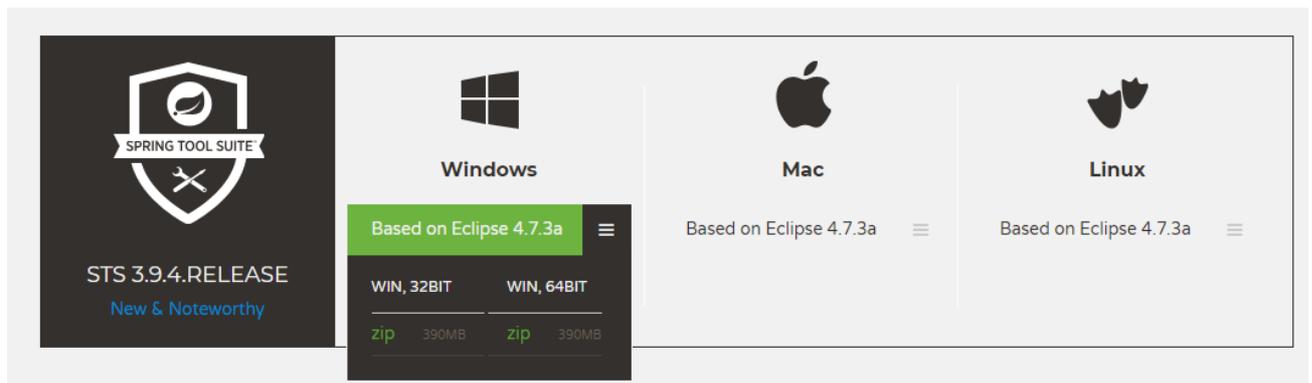


Figura A.5: Descarga STS

Una vez ejecutado e instalado, iniciamos la aplicación.



Figura A.6: Inicio STS

Y por último, escogemos un directorio para el workspace de la aplicación

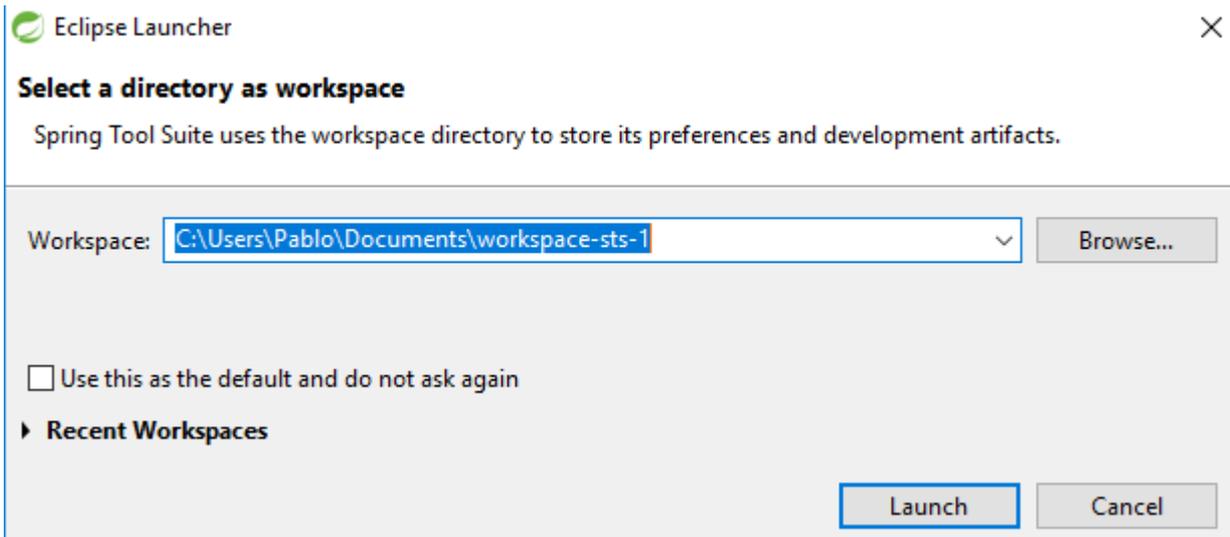


Figura A.7: Workspace

A.6 Instalación de XAMPP

Descargamos XAMPP de la página <https://www.apachefriends.org/es/index.html> y lo instalamos.



Figura A.8: Descarga de XAMPP

Una vez instalado iniciamos la aplicación y activamos el servidor Mysql.

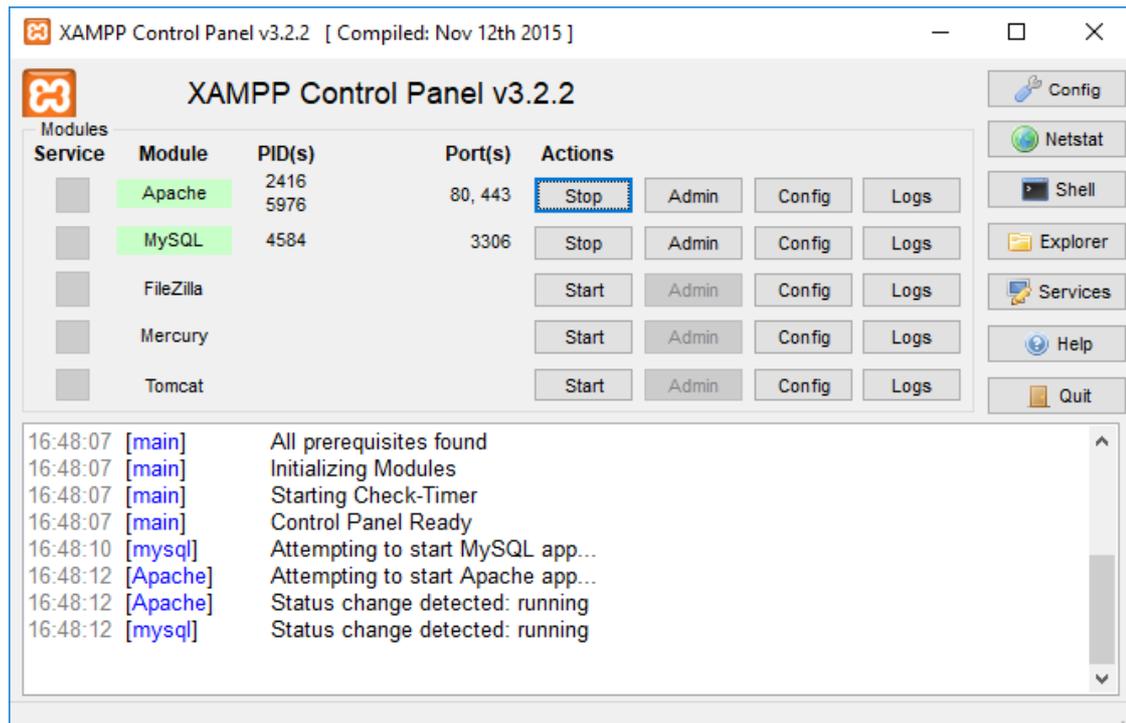


Figura A.9: Interfaz XAMPP

Acto seguido pulsamos sobre “Admin” en la fila de MySQL para dirigirnos a la interfaz gráfica de la base de datos “phpMyAdmin” donde podremos gestionar nuestras bases de datos

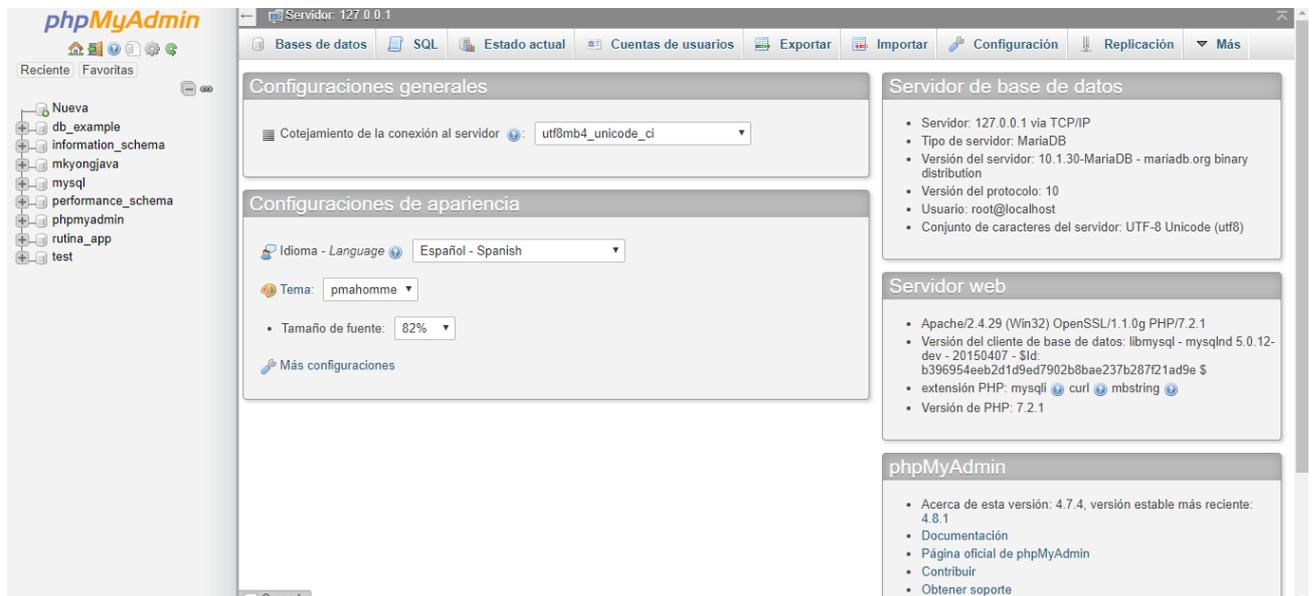


Figura A.10: phpMyAdmin XAMPP

A.7 Despliegue de la aplicación

- I. Para crear la base de datos que utilizará la aplicación web, en la interfaz de phpMyAdmin, elegimos la opción “Importar” y escogemos el archivo ‘rutina_app.sql’.

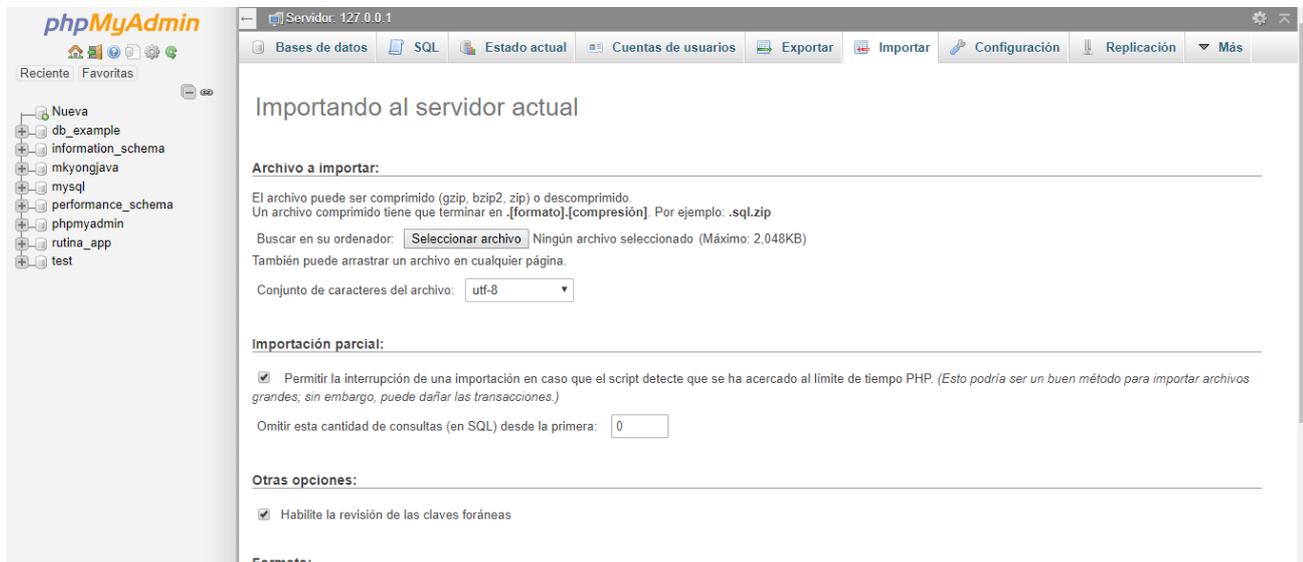


Figura A.11: Importar BD

- II. Descomprimos el archivo ‘MyTraining.zip’ y colocamos la carpeta ‘MyTraining’ en el workspace de la aplicación definido anteriormente.
- III. Iniciamos STS y seleccionamos la opción ‘File->Import->Existing Maven projects’

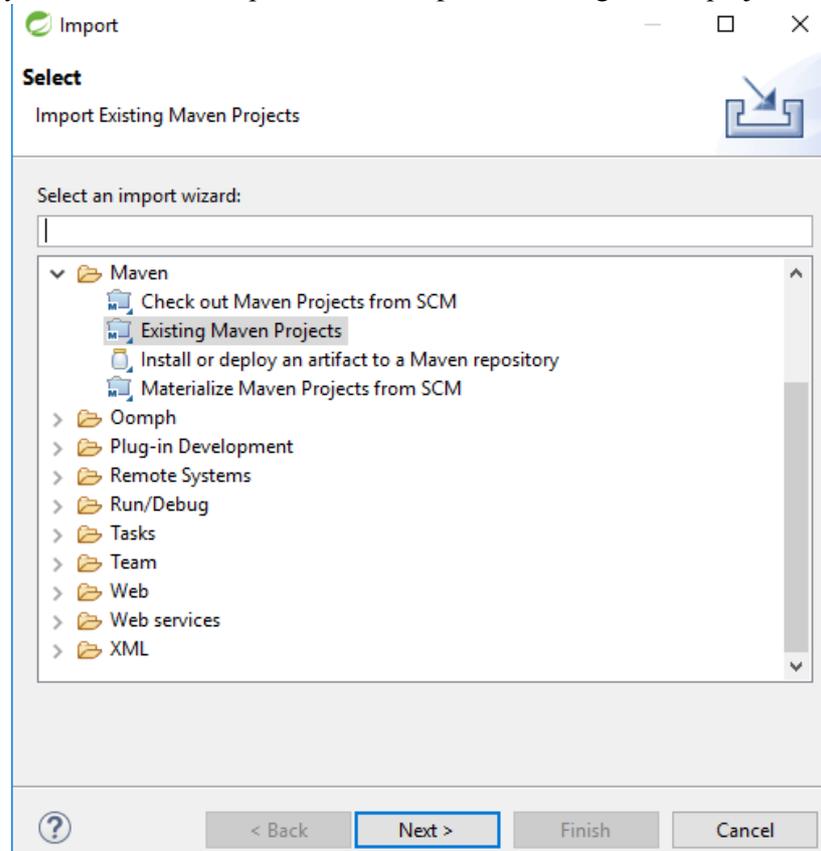


Figura A.12: Importar proyecto

- IV. Escogemos como Root Directory la carpeta del proyecto anteriormente colocada en el workspace.

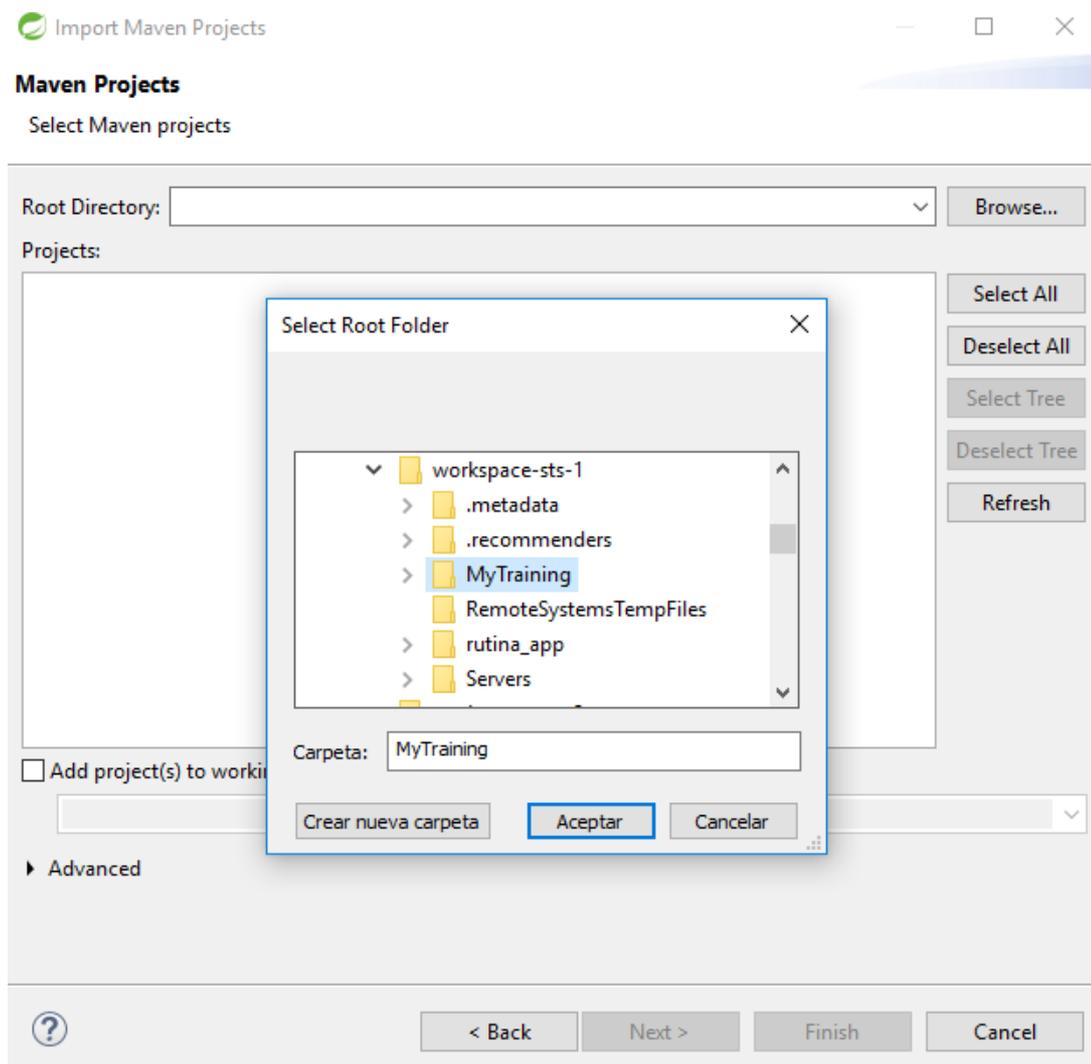


Figura A.13: Root Directoy

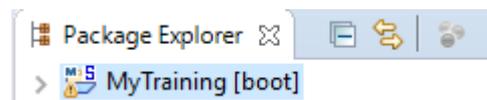


Figura A.14: Proyecto importado

A.8 Account kit de Facebook

Hemos empleado esta herramienta de Facebook para proporcionar la autenticación de usuario a través de SMS para la cual hace falta crearse una cuenta de desarrollador de Facebook.

Una guía rápida de creación y configuración de ello puede verse aquí:

<https://developers.facebook.com/docs/accountkit/webjs>

The screenshot shows the 'Account Kit para web (JavaScript): inicio rápido' page on the Facebook for developers site. The page is in Spanish and features a dark header with navigation links: 'Documentos', 'Herramientas', 'Ayuda', and a search bar. The main content area is divided into three columns. The left column contains a sidebar with links for 'Account Kit', 'Información general', 'iOS', 'Android', 'Web', and various API-related topics. The middle column has a large heading, a sub-heading, a paragraph, a callout box with an information icon, and two numbered steps: '1. Inicio de sesión' and '2. Definir la configuración de la aplicación'. The right column lists 'En esta página' with five numbered items and a 'Sigüientes pasos' section. A blue 'Entrar en Facebook' button is visible below the first step.

Account Kit

Información general
iOS
Android
Web
Personalizar la interfaz de usuario para web
Personalizar el dominio y los códigos de país
Referencia de la API
Web básica
Personalizar el correo electrónico
Identificadores de acceso
Uso de la API Graph
Códigos de país para SMS
Idiomas admitidos
Eventos de la aplicación y análisis
Changelog
Best Practices
Preguntas frecuentes

Account Kit para web (JavaScript): inicio rápido

En la sección Sigüientes pasos encontrarás información sobre las opciones de personalización disponibles.

Para obtener información sobre problemas comunes relacionados con la integración, consulta la sección de preguntas frecuentes.

1. Inicio de sesión

Entra en Facebook para crear aplicaciones o regístrate como desarrollador.

[Entrar en Facebook](#)

2. Definir la configuración de la aplicación

Decide si quieres permitir el inicio de sesión con SMS y correo electrónico, y selecciona la configuración.

En esta página

1. Inicio de sesión
2. Definir la configuración de la aplicación
3. Configurar la página HTML de inicio de sesión
4. Ejemplo de implementación de PHP en tu servidor web
5. Ejemplo de implementación de Node.js en tu servidor web

Sigüientes pasos

Figura A.15: Account kit

Una vez creada nuestra aplicación tendremos la opción de personalizarla desde el menú de configuración de nuestra cuenta.

The screenshot shows the 'Mis aplicaciones' page on the Facebook for developers site. The page is in Spanish and features a dark header with navigation links: 'Documentos', 'Herramientas', 'Ayuda', 'Mis aplicaciones', and a search bar. The main content area is divided into two columns. The left column has a search bar and a large 'Add a New App' button. The right column shows a card for 'MyTraining' with its application ID and status. Below the card, there are two checkboxes: 'Account Kit' (checked) and 'Messenger' (unchecked), with a 'Finish Setup' button next to the Messenger checkbox.

facebook for developers

Documentos Herramientas Ayuda Mis aplicaciones

Buscar en developers.facebook.com

Buscar aplicaciones

[+ Añadir una nueva aplicación](#)

MyTraining
Identificador de la aplicación:
1826347254092121
Estado: 🇺🇸 En desarrollo

Account Kit

Messenger [Finish Setup](#)

Figura A.16: Account kit 2

The screenshot displays the 'Configuración del inicio de sesión web' (Web Login Configuration) page in the Account Kit interface. On the left is a navigation sidebar with options like 'Panel', 'Configuración', 'Roles', 'Alertas', 'Revisión de la aplicación', 'PRODUCTOS', 'Account Kit', 'Configuración', 'Inicio rápido', 'Messenger', and 'Registro de actividad'. The main content area is titled 'Configuración del inicio de sesión web' and contains several sections:

- Dominios del servidor que se utilizan para el SDK para web:** A text input field containing 'http://localhost:8080/'. Below it, a note states: 'Introduce los dominios del servidor en los que alojas la integración con Account Kit. Los inicios de sesión se aprobarán desde estos dominios.'
- URL de direccionamiento que se utilizan para el inicio de sesión con correo electrónico en el SDK para web o para el inicio de sesión con número de teléfono o correo electrónico en el SDK básico:** A text input field containing 'http://localhost:8080/'. Below it, a note states: 'Introduce la URL de redireccionamiento en la que esperas recibir respuesta de Account Kit. Al iniciar sesión solo se redireccionará a estas URL.'
- Lista de códigos de país admitidos:** A text input field. Below it, a note states: 'Se permitirá el inicio de sesión web para los números de teléfono con estos códigos de país. Para permitir el inicio de sesión con todos los códigos de país, deja este campo en blanco.'
- Lista de códigos de país no admitidos:** A text input field. Below it, a note states: 'No se permitirá el inicio de sesión web para los números de teléfono con estos códigos de país.'

At the bottom, there are two sections:

- Personaliza tu interfaz:** Two radio buttons: 'Opciones básicas' (unselected) and 'Opciones avanzadas' (selected).
- Imagen de fondo:** A text input field with the placeholder text 'Introduce la URL'.
- Vista previa:** A preview window showing a login screen with the text 'Inicio de sesión con SMS' and 'SMS login (no input)', and an error message 'Error de inicio de sesión' below it.

Figura A.17: Account kit 3