



Título del Proyecto:

Sport Manager

Autor:

Chorro, Pau

Director:

Carbonell, Vicente

TESINA PARA LA OBTENCIÓN DEL TÍTULO DE:

**Diploma de Especialización en
Desarrollo de Aplicaciones para
Android**

Septiembre del 2017

Contenido

Introducción	3
Descripción del problema	3
Objetivos	4
Motivación	5
Tecnologías utilizadas	6
Arquitectura de la aplicación	10
Esquema del diseño	11
Modelo de datos	12
Rest API. Listado de servicios web	13
Vistas	14
Conclusiones	20
Anexos	21

Introducción

Descripción del problema

El avance en los entrenamientos ha hecho que cada vez sea más importante un control más riguroso del entrenamiento y la competición. Son muchos los métodos e instrumentos a través de los cuales se realiza un control del entrenamiento de los deportistas, algunos de los cuales, pueden llegar a no ser lo suficientemente eficaces debido a multitud de errores que pueden ocurrir (falta de comunicación o ineficaz, marco temporal inadecuado, errores en la toma de datos,...).

Es necesario adaptarse a las nuevas tecnologías de Aplicaciones Web y Móvil para poder realizar el Control de la carga de entrenamiento de manera que permitiría automatizar un control más riguroso, preciso y eficaz del entrenamiento de los deportistas mediante una recogida de datos orientada a las necesidades comunicativas del entrenador y sus deportistas, que tenga como resultado una organización y análisis más eficaz de los datos recogidos.

FICHA DE CONTROL DE CARGA DE ENTRENAMIENTO DE LOS DEPORTISTAS									
Nº DE ENTRENADOR: 1									
FECHA: 10/01/2010									
Nº DE DEPORTISTA	TIPO DE ENTRENAMIENTO	TIPO DE ENTRENAMIENTO	TIPO DE ENTRENAMIENTO	TIPO DE ENTRENAMIENTO	TIPO DE ENTRENAMIENTO	TIPO DE ENTRENAMIENTO	TIPO DE ENTRENAMIENTO	TIPO DE ENTRENAMIENTO	TIPO DE ENTRENAMIENTO
1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9
10	10	10	10	10	10	10	10	10	10
11	11	11	11	11	11	11	11	11	11
12	12	12	12	12	12	12	12	12	12
13	13	13	13	13	13	13	13	13	13
14	14	14	14	14	14	14	14	14	14
15	15	15	15	15	15	15	15	15	15
16	16	16	16	16	16	16	16	16	16
17	17	17	17	17	17	17	17	17	17
18	18	18	18	18	18	18	18	18	18
19	19	19	19	19	19	19	19	19	19
20	20	20	20	20	20	20	20	20	20
21	21	21	21	21	21	21	21	21	21
22	22	22	22	22	22	22	22	22	22
23	23	23	23	23	23	23	23	23	23
24	24	24	24	24	24	24	24	24	24
25	25	25	25	25	25	25	25	25	25
26	26	26	26	26	26	26	26	26	26
27	27	27	27	27	27	27	27	27	27
28	28	28	28	28	28	28	28	28	28
29	29	29	29	29	29	29	29	29	29
30	30	30	30	30	30	30	30	30	30
31	31	31	31	31	31	31	31	31	31
32	32	32	32	32	32	32	32	32	32
33	33	33	33	33	33	33	33	33	33
34	34	34	34	34	34	34	34	34	34
35	35	35	35	35	35	35	35	35	35
36	36	36	36	36	36	36	36	36	36
37	37	37	37	37	37	37	37	37	37
38	38	38	38	38	38	38	38	38	38
39	39	39	39	39	39	39	39	39	39
40	40	40	40	40	40	40	40	40	40
41	41	41	41	41	41	41	41	41	41
42	42	42	42	42	42	42	42	42	42
43	43	43	43	43	43	43	43	43	43
44	44	44	44	44	44	44	44	44	44
45	45	45	45	45	45	45	45	45	45
46	46	46	46	46	46	46	46	46	46
47	47	47	47	47	47	47	47	47	47
48	48	48	48	48	48	48	48	48	48
49	49	49	49	49	49	49	49	49	49
50	50	50	50	50	50	50	50	50	50
51	51	51	51	51	51	51	51	51	51
52	52	52	52	52	52	52	52	52	52
53	53	53	53	53	53	53	53	53	53
54	54	54	54	54	54	54	54	54	54
55	55	55	55	55	55	55	55	55	55
56	56	56	56	56	56	56	56	56	56
57	57	57	57	57	57	57	57	57	57
58	58	58	58	58	58	58	58	58	58
59	59	59	59	59	59	59	59	59	59
60	60	60	60	60	60	60	60	60	60
61	61	61	61	61	61	61	61	61	61
62	62	62	62	62	62	62	62	62	62
63	63	63	63	63	63	63	63	63	63
64	64	64	64	64	64	64	64	64	64
65	65	65	65	65	65	65	65	65	65
66	66	66	66	66	66	66	66	66	66
67	67	67	67	67	67	67	67	67	67
68	68	68	68	68	68	68	68	68	68
69	69	69	69	69	69	69	69	69	69
70	70	70	70	70	70	70	70	70	70
71	71	71	71	71	71	71	71	71	71
72	72	72	72	72	72	72	72	72	72
73	73	73	73	73	73	73	73	73	73
74	74	74	74	74	74	74	74	74	74
75	75	75	75	75	75	75	75	75	75
76	76	76	76	76	76	76	76	76	76
77	77	77	77	77	77	77	77	77	77
78	78	78	78	78	78	78	78	78	78
79	79	79	79	79	79	79	79	79	79
80	80	80	80	80	80	80	80	80	80
81	81	81	81	81	81	81	81	81	81
82	82	82	82	82	82	82	82	82	82
83	83	83	83	83	83	83	83	83	83
84	84	84	84	84	84	84	84	84	84
85	85	85	85	85	85	85	85	85	85
86	86	86	86	86	86	86	86	86	86
87	87	87	87	87	87	87	87	87	87
88	88	88	88	88	88	88	88	88	88
89	89	89	89	89	89	89	89	89	89
90	90	90	90	90	90	90	90	90	90
91	91	91	91	91	91	91	91	91	91
92	92	92	92	92	92	92	92	92	92
93	93	93	93	93	93	93	93	93	93
94	94	94	94	94	94	94	94	94	94
95	95	95	95	95	95	95	95	95	95
96	96	96	96	96	96	96	96	96	96
97	97	97	97	97	97	97	97	97	97
98	98	98	98	98	98	98	98	98	98
99	99	99	99	99	99	99	99	99	99
100	100	100	100	100	100	100	100	100	100



Imágenes de control de la carga de entrenamiento

El objetivo principal de un entrenador es conseguir un estado adecuado de forma para sus atletas. Así, es necesaria una programación precisa y rigurosa que no sería posible sin un

adecuado control de las cargas de entrenamiento. La función principal de la carga de entrenamiento es comparar las cargas diferentes de las sesiones de entrenamiento. Existen multitud de métodos para monitorizar la carga del entrenamiento: para ello se utilizan parámetros o índices internos del organismo (carga interna) como la FC o el Consumo de oxígeno (VO_2), concentración de ácido láctico (LAC), etc., así como los índices externos (carga externa) que muestran cuanto ha trabajado el deportista como son los Km, el tiempo, Velocidad media,...

Por otra parte, otra forma no invasiva se basa en la propia experiencia del deportista. En los métodos de control del entrenamiento que se basan en la percepción del deportista, es necesario que se realice bajo unas directrices rigurosas (experiencia del deportista, feedback, comunicación bidireccional, marco temporal adecuado...) para un correcto control de la carga.

Actualmente, está totalmente reconocido el uso de escalas de RPE con multitud de variantes. Existen varios tipos de escalas RPE. La aplicación desarrollada se basa en la propuesta de Control del entrenamiento mediante RPE

Objetivos

El objetivo principal es desarrollar una aplicación que sirva para registrar el control del entrenamiento en escalas RPE de los deportistas. Además servirá como base de ampliaciones futuras para que la aplicación se convierta en un gestor de deportistas para los entrenadores.

Así la funcionalidad principal de la aplicación consiste en:

- **Registro de tiempo de zonas:** se registrará el tiempo de cada zona según el tipo de deporte. Esto servirá más adelante para realizar cálculos y poder controlar los minutos que se han hecho

en cada deporte, el total en cada zona, etc... Mediante estos datos el entrenador será capaz de prever diferentes estados de forma.

El tiempo en cada zona se determinará siguiendo diferentes criterios, o bien por percepción del propio deportista, o por otro lado siguiendo el registro de un pulsometro (recomendada).

Motivación

Este proyecto lo estoy realizando junto a unos compañeros del Grado de Educación Física. La principal motivación en la realización de este proyecto es la ausencia de aplicaciones para el control de la carga de entrenamiento.

Si bien existen muchas aplicaciones desarrolladas por las principales marcas *Polar*, *Garmin*, etc. En el cual te permite llevar un control de los entrenamientos de tus deportistas, éste no cumple con rigor el uso de alguno de los principales métodos de carga de entrenamiento sino que parece más enfocado a informar al usuario de a pie.

Es por ello que la realización de este proyecto tiene como principal objetivo paliar la necesidad de muchos entrenadores en cuanto al control de la carga de entrenamiento se refiere.

Tecnologías utilizadas

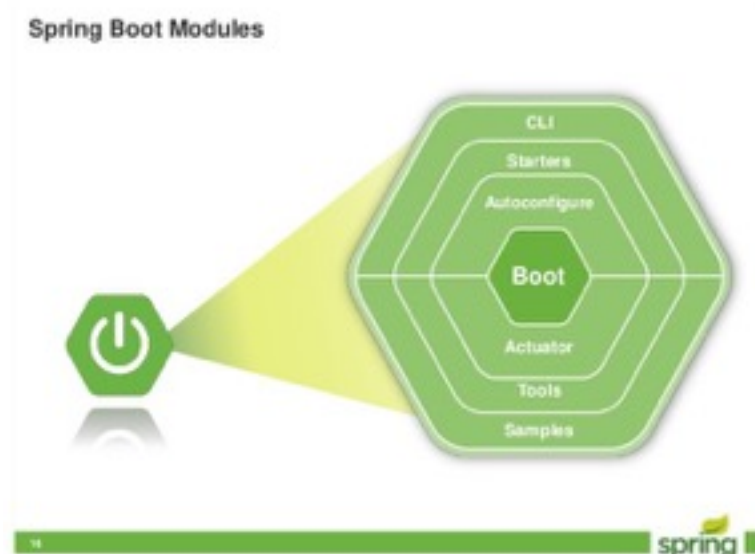
- **Java 8:** es la versión más reciente de Java que incluye nuevas características, mejoras y correcciones de bugs para mejorar la eficacia en el desarrollo



- **J2EE:** Permite utilizar arquitecturas de N capas distribuidas y se apoya ampliamente en componentes de software modulares ejecutándose sobre un servidor de aplicaciones.



- **Spring Boot:** simplifica la creación de aplicaciones de aplicaciones y servicios Spring extendiendo el concepto de Runtime configurando por debajo el AppServer, Contextos de Spring, monitorización,... todo basado en una aproximación Convention-Over-Configuration.
- **Spring MVC:** framework diseñado alrededor de DispatcherServlet que despacha solicitudes a un manejador.



- **Spring security:** proporciona servicios de seguridad para aplicaciones de software empresariales basados en J2EE
- **Thymeleaf:** es una librería Java que implementa un motor de plantillas de XML/XHTML/HTML5 (también extensible a otros formatos) que puede ser utilizado tanto en modo web como en otros entornos no web. Se acopla muy bien para trabajar en la capa vista del MVC de aplicaciones web, pero puede procesar cualquier archivo XML, incluso en entornos desconectados.



- **HTML5:** HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web.



- **JavaScript:** es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.



- **CSS3:** es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML.



- **Bootstrap:** es un framework o conjunto de herramientas de Código abierto para diseño de sitios y aplicaciones web. Contiene plantillas de diseño con tipografía, formularios, botones, cuadros, menús de navegación y otros elementos de diseño basado en HTML y CSS, así como, extensiones de JavaScript opcionales adicionales.



- **JQuery:** es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.



- **MongoDB:** base de datos NoSQL. Guarda estructuras de datos en documentos tipo JSON con un esquema dinámico (BSON), haciendo que la integración de los datos en ciertas aplicaciones sea más fácil y rápida



Arquitectura de la aplicación

La arquitectura de la aplicación Sport Manager se basa en tres capas que corren en una instancia EC2 de Ubuntu server en AWS configurado para ello. Si bien dos se mezclan entre sí están bien diferenciadas.

Parte Cliente:

En primer lugar, tenemos la capa de aplicación del cliente o lo que más bien son los recursos estáticos que forman la aplicación con la que interactuará el cliente, es decir: las páginas Web con Html5, JS y Javascript.

Parte Servidor

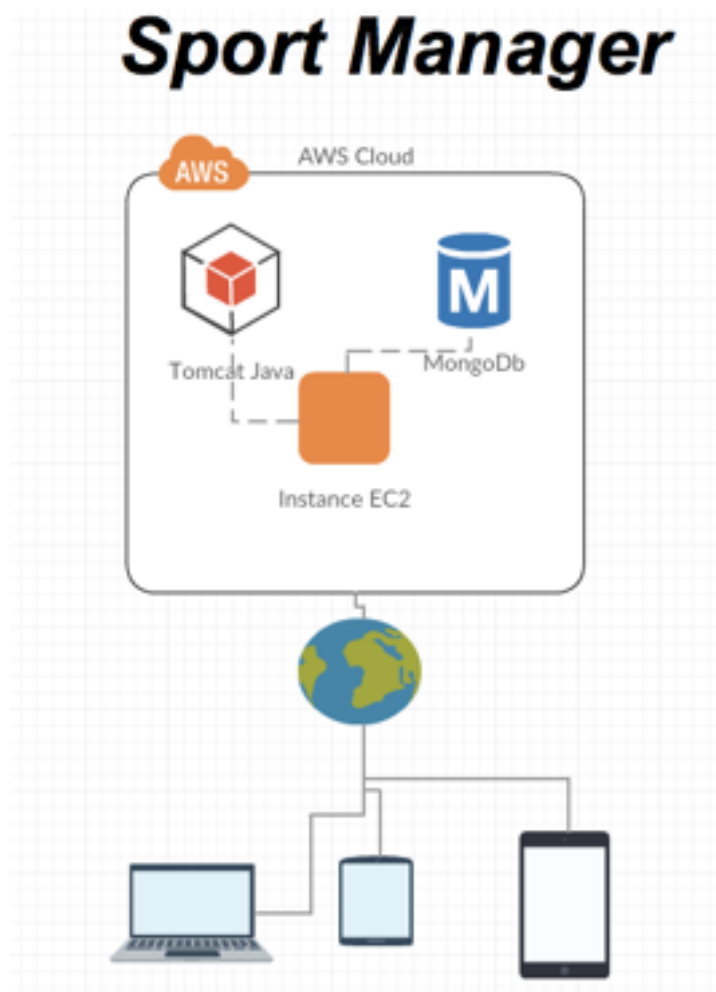
Por otro lado, la parte de backend o lógica de negocio, que a su vez se ha dividido en dos subcapas para modularizar la aplicación consta de dos capas diferentes:

- **API Rest:** lógica de negocio principal mediante las diferentes aplicaciones interactuarán. Está totalmente independizado de el bloque de vistas, por lo que es totalmente reutilizaban para futuros desarrollos, por lo que se pueden realizar diferentes aplicaciones como móviles y consumir estos servicios independientemente de la lógica de vistas.
- **Lógica de vistas:** hace llamadas a los endpoints de la API Rest inyectando vía dependencias los controladores de la API Rest para modularizar estos dos bloques. Y mediante **Thymeleaf** genera recursos estáticos **HTML5/CSS3/JS** que son las vistas mediante las que el usuario interactúa.

Base de datos

Base de datos NoSQL a través de la cual se almacena la información o datos con los que la aplicación interactúa. La base de datos se basa en colecciones de documentos en formato Bson (una modificación de Json) que facilita el desarrollo de aplicaciones.

Esquema del diseño





Modelo de datos

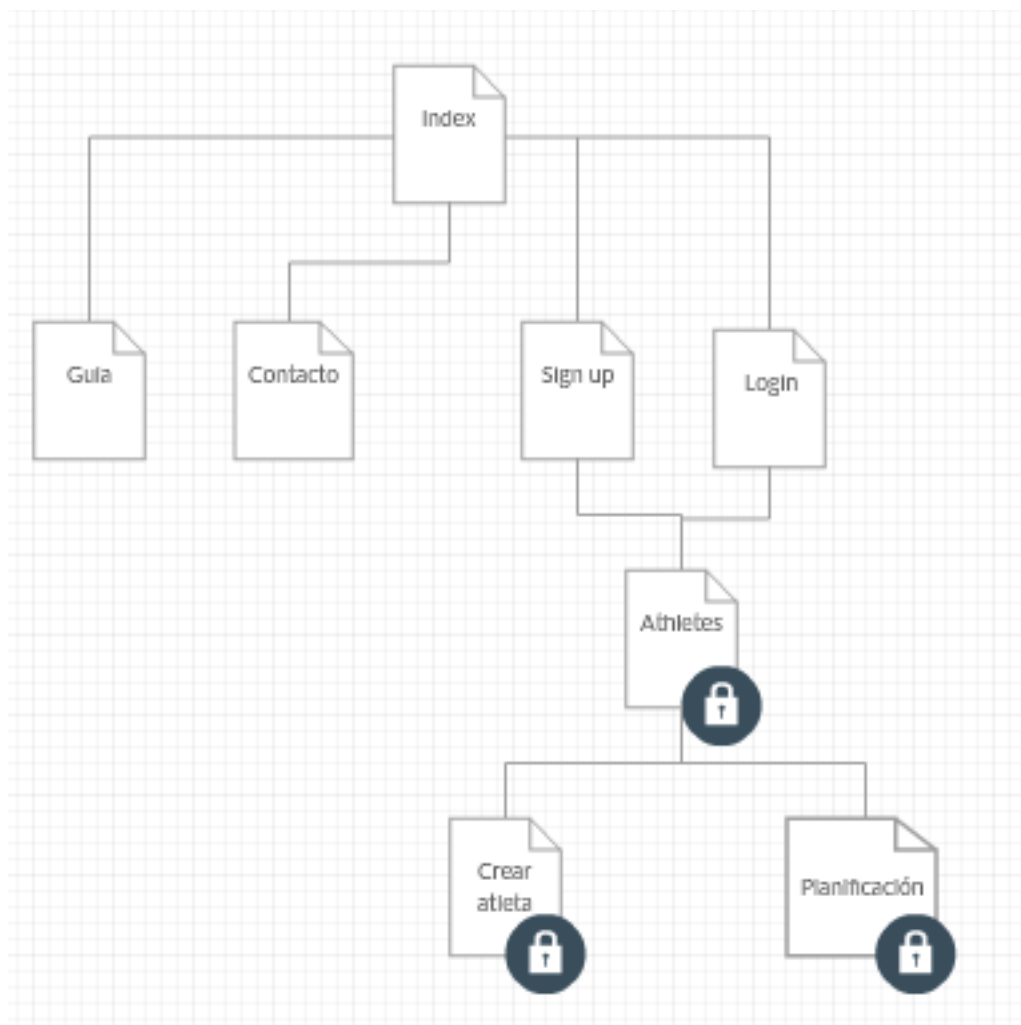


Rest API. Listado de servicios web

- [/v1/athletes], methods=[**POST**], consumes=[application/json;charset=UTF-8], produces=[application/json;charset=UTF-8]
- [/v1/athletes], methods=[**PUT**], consumes=[application/json;charset=UTF-8], produces=[application/json;charset=UTF-8]
- [/v1/athletes/{email}/], methods=[**GET**], produces=[application/json;charset=UTF-8]
- [/v1/athletes/{coach}/], methods=[**GET**], consumes=[application/json;charset=UTF-8], produces=[application/json;charset=UTF-8]
- [/v1/athletes/{email}/], methods=[**PUT**], consumes=[application/json;charset=UTF-8], produces=[application/json;charset=UTF-8]
- [/v1/coaches/{email}/], methods=[**GET**], produces=[application/json;charset=UTF-8]
- [/v1/coaches], methods=[**POST**], consumes=[application/json;charset=UTF-8], produces=[application/json;charset=UTF-8]
- [/v1/coaches], methods=[**PUT**], consumes=[application/json;charset=UTF-8], produces=[application/json;charset=UTF-8]
- [/v1/users/authentication], methods=[**POST**], produces=[application/json;charset=UTF-8]

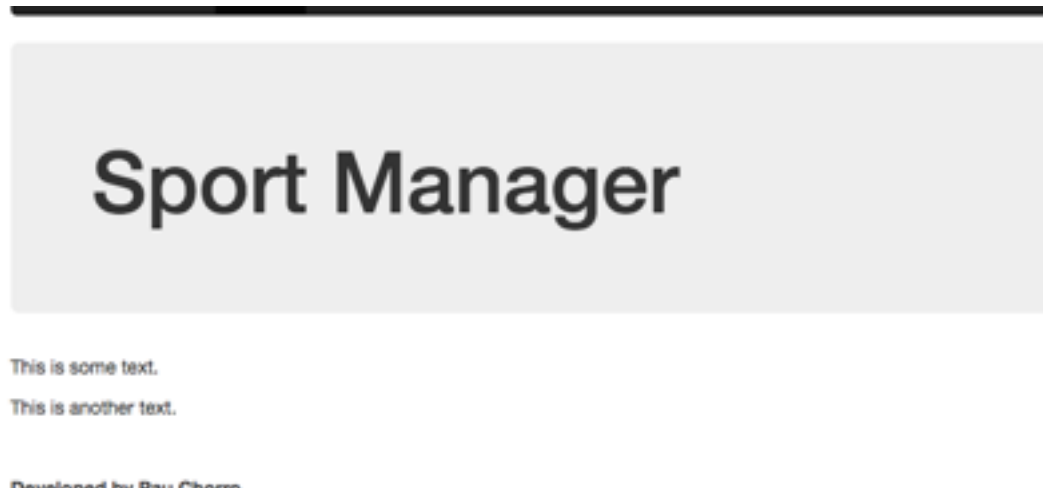
Vistas

El diagrama de navegación es el siguiente:



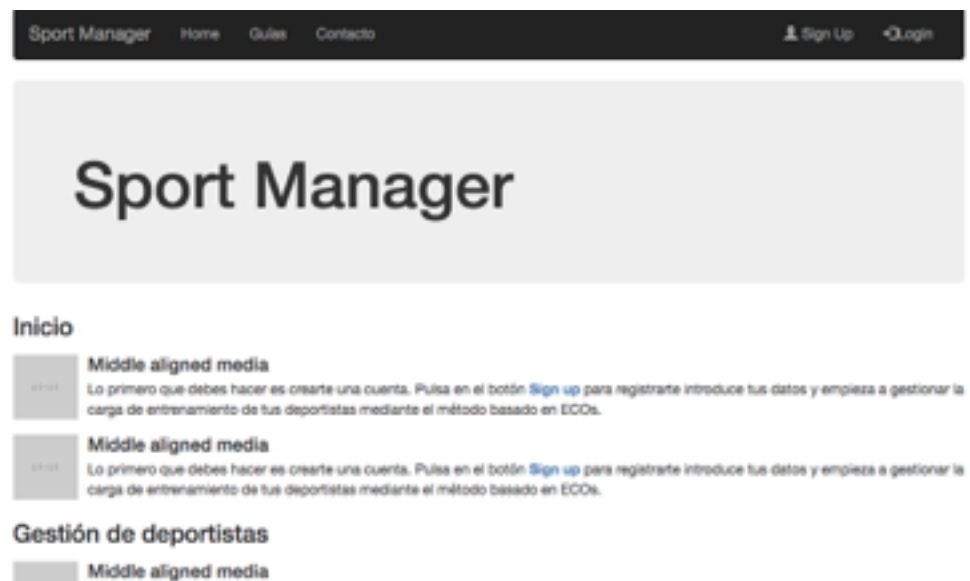
Index

Página de inicio, muestra la información general sobre que trata la aplicación. Puedes ir a la página que desees desde el navegador que hay en la cabecera.



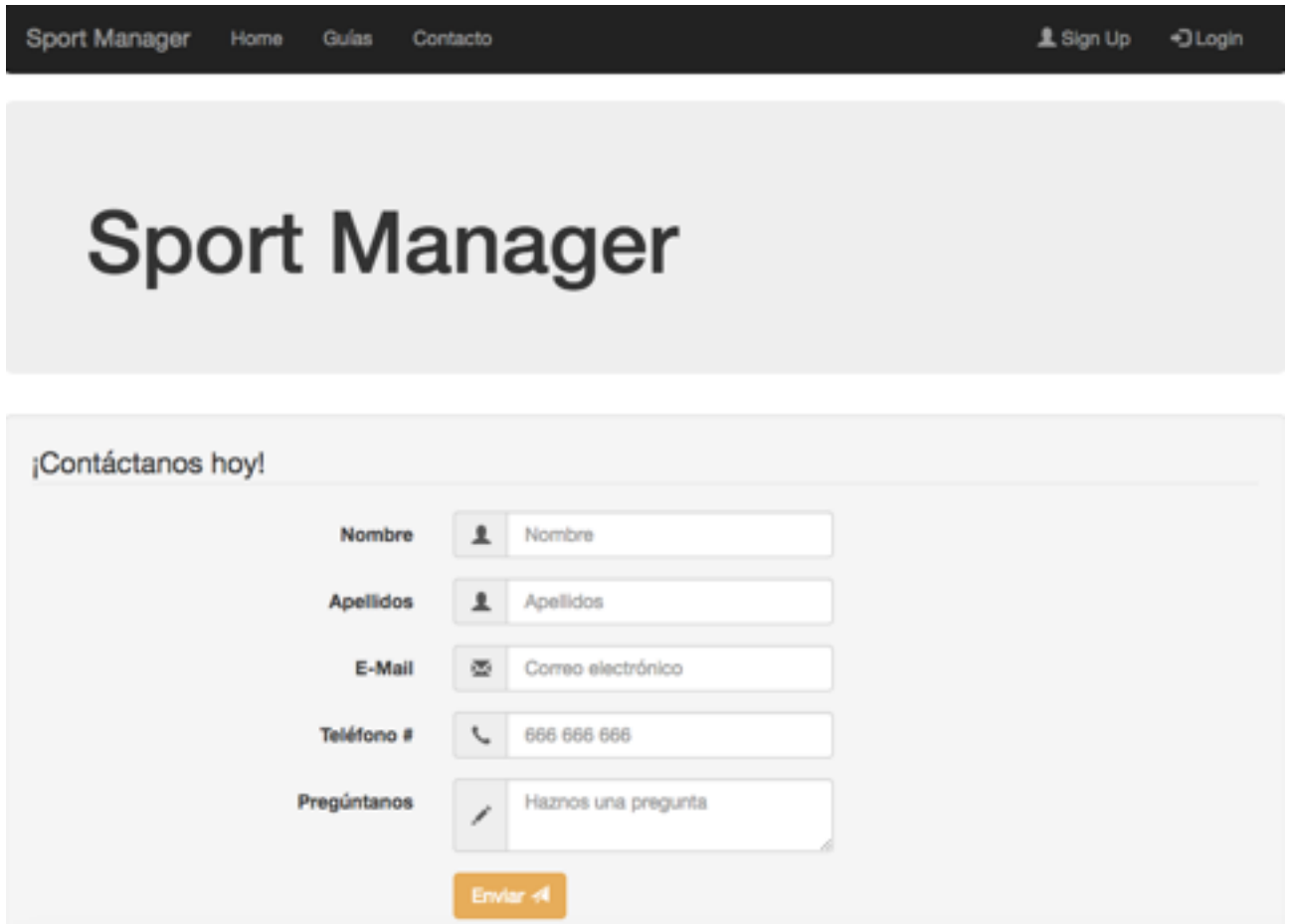
Guia

Explica el uso de la aplicación. Para ello se muestran diferentes explicaciones con su imagen correspondiente y así facilitar la familiarización del uso de la aplicación.



Contacto

Permite al usuario contactar con el desarrollador para poder enviar sugerencias, quejas o cualquier otro tema que el cliente crea conveniente que se deba tratar.



The screenshot shows the 'Sport Manager' website's contact page. At the top is a dark navigation bar with links for 'Sport Manager', 'Home', 'Guías', and 'Contacto', along with 'Sign Up' and 'Login' buttons. Below this is a large grey banner with the text 'Sport Manager'. The main content area is titled '¡Contáctanos hoy!' and contains a contact form with five input fields: 'Nombre' (with a person icon), 'Apellidos' (with a person icon), 'E-Mail' (with an envelope icon), 'Teléfono #' (with a phone icon), and 'Pregúntanos' (with a pencil icon). Each field has a placeholder text. At the bottom of the form is an orange 'Enviar' button with a right-pointing arrow.

Signup

Página de creación de usuarios a través de la cual podremos registrarnos como entrenador. Debemos introducir como mínimo nuestro email y contraseña.



Registration

First Name

First Name, : Smith, Harry

Surname

Last Name, : Targaryen, Potter

Email

Password

☐ I accept [terms](#)

Enviar

Register

Login

Página de login a partir de la cual podremos conectarnos a la aplicación con nuestro usuario ya creado y tener acceso a la funcionalidad principal de la aplicación

Log in

Email

Email

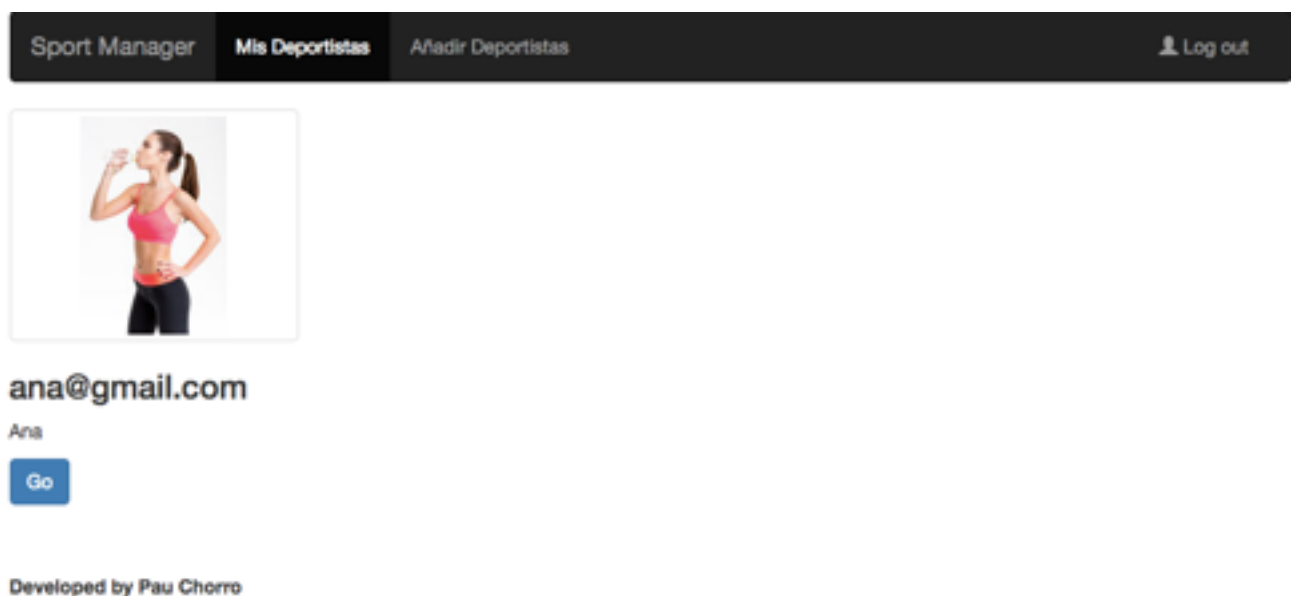
Password

Password

Log in

Athletes

Muestra una lista de nuestros deportistas y un link a su página de planificación para gestionar el control de la carga de entrenamiento.



Crear Ateta

Permite crear un deportista para poder permitirnos gestionar su contenido, debemos añadir una imagen en formato url así como una contraseña que se usará más adelante en una futura continuación del desarrollo.

Sport Manager
Mis Deportistas
Añadir Deportistas
Log out

Registration

First Name

First Name, -: Smith, Harry

Surname

Last Name, -: Targaryen, Potter

Url Image

Url Image online, -: dropbox, drive, google images,...

Email

Password

Wheight

Registrar Athlete

Planificación

Permite gestionar la carga de entrenamiento del deportista que hayamos seleccionado. Ésta página es la funcionalidad principal de la aplicación.

Sport Manager
Mis Deportistas
Añadir Deportistas
Athlete Planification: Pablo Mira
Log out

Planification table

Planification table planification

- Planification

Fecha	Carga	Comentarios
2016-09-18	asdas	asdasdas

Carga: Entrenamiento resistencia

zone	swim	bike	run		
1	100	0	0	✖	↑ ↓
2	0	0	0	✖	↑ ↓
3	0	0	0	✖	↑ ↓

Guardar datos

Éxito! Datos introducidos correctamente

Conclusiones

La aplicación que se planteo en un principio constaba de más funcionalidades comparado con el estado actual de la aplicación.

Por otro lado se planteo el uso de frameworks como AngularJs para el desarrollo de las pantallas aunque finalmente se opto por el uso del motor de plantillas Thymeleaf el cual es adoptado por Spring Boot como el principal motor de plantillas para el uso de Spring MVC lo que indica la calidad de esta librería. Si a eso sumamos que ha sido desarrollado en España por Daniel Fernandez, Jose Samper y Emanuel Reina y a si poder dar a conocer este proyecto de código libre con marca nacional. Si bien en un futuro el propósito es hacer una migración a AngularJS de manera que las vistas queden totalmente modularizadas e independiente de la parte servidor. Es por ello que el desarrollo actual se ha centrado en individualizar las dependencias entre la API Rest y la capa de vistas.

En cuanto al uso la autenticación, debido a que decidí usar la última versión de Thymeleaf y Spring Boot, esto me ha acabado penalizando de manera que no he podido encontrar mucha información al respecto sobre la última versión de **Spring security** y su integración con Thymeleaf haciendo que esta parte no quede completa ya que no he podido solucionar los diferentes problemas que me han aparecido así que toca esperar a que se haya extendido más el uso de la versión 1.4 de Spring así como la Thymeleaf 3, para poder encontrar más información al respecto y realizar la integración más rápidamente.

Finalmente, destacar el uso de MongoDB para la persistencia de datos con el que quiero destacar la facilidad con la que se adapta a las necesidades reales de nuestro modelo de datos. Comparado a las base de datos relacionales te permite agilizar mucho el desarrollo puesto que permite que el modelo de datos sea

un reflejo de las necesidades reales de la lógica de negocio así como evitar muchas de las restricciones que las Base de Datos relacionales suponen y que muchas veces entorpecen el desarrollo.

Anexos

Guía de uso en youtube: <https://www.youtube.com/watch?v=IUSbBDYeoNY&feature=youtu.be>