



Universidad
Zaragoza

Trabajo Fin de Grado

GymGest: Gestión de gimnasios.

Autor/es

Ricardo Ibáñez Fuentes

Director/es

Gabriel Fuertes Muñoz

Escuela Universitaria Politécnica de Teruel
2015

Resumen

GymGest es una aplicación de gestión para los gimnasios, con la cual se pretende tener un mayor conocimiento y rendimiento sobre estos. Así cada uno de los gimnasios conocerá con mayor detalle el trabajo y seguimiento que hacen los monitores a sus clientes, así como tener una mayor información de ellos.

Además esta aplicación mejora la imagen de la empresa respecto a sus clientes, obteniendo de esta forma una mayor rentabilidad tanto por parte del gimnasio (atrayendo a más clientes) como por parte de los clientes sintiéndose más valorados. Hay que tener en cuenta que cada más vez la competencia es mayor y se debe ofrecer algo distinto para abarcar más cuota de mercado.

Las tablas que genera esta aplicación, hace que ponerse en forma sea más fácil para los clientes que se inicien en este mundo, ya que únicamente poseerá los ejercicios necesarios de cada persona, sin distraerlo con otros ejercicios irrelevantes para mejorar su estado físico (como por ejemplo las tablas genéricas que se dan actualmente). El programa ya está cualificado para que no existan errores a la hora de asignar un ejercicio para un grupo muscular al que no pertenece. De esta manera conseguimos que el personal perteneciente al gimnasio tenga más conocimientos sobre los ejercicios que manda a sus clientes, dotándole mayor valor a un entrenador personal.

Se ha realizado una página web adaptativa con el fin de que cualquier persona pueda acceder a ella desde cualquier tipo de dispositivo, ya que en la actualidad el uso de dispositivos móviles ha aumentado notablemente y es casi un requisito fundamental.

Por último se ha pretendido que los administradores de la aplicación, puedan recopilar más información sobre el uso que se le da a esta por parte de todos los usuarios con el fin de poder seguir mejorándola día a día y teniendo también un mayor control de los clientes incluso a nivel mundial.

Abreviaturas

A continuación se explican los términos relacionados con la aplicación.

- **BD:** Base de datos.
- **CU:** Caso de uso.
- **DNI:** Documento nacional de identidad.
- **NIF:** Número de identificación fiscal.
- **RF:** Requisitos funcionales.
- **RI:** Requisitos de interfaz
- **RS:** Requisitos del software
- **JS:** abreviatura común de JavaScript.
- **MVC:** Modelo vista controlador.
- **SQL:** Lenguaje de consulta estructurado.
- **URL:** Identificador de recursos uniforme.
- **TFG:** Trabajo final de grado.
- **HTTP:** Protocolo de transferencia de hipertexto.
- **PDF:** Formato de documento portátil.
- **DS:** Diagrama de actividad.
- **DA:** Diagrama de actividad.

Contenido

Resumen	2
Abreviaturas.....	3
Índice de Ilustraciones	6
1. Introducción.....	9
1.1 Motivación	9
1.2 Objetivos.....	10
1.3 Visión general del documento	10
1.4 Agradecimientos	11
2. Tecnologías usadas	11
3. Estado del arte	12
3.1 FitnessBliss	12
3.2 Xculture	13
3.3 Conclusiones.....	13
4. Planificación	14
5. Análisis	16
5.1 Propósito	16
5.2 Alcance	16
5.3 Perspectivas del producto	17
5.4 Funciones del producto.....	18
5.4.1 Casos de Uso	18
5.4.2 Características de los usuarios	18
5.4.3 Suposiciones y dependencias.....	18
5.4.4 Evolución previsible del sistema	18
5.5 Especificación de requisitos.....	18
Requisitos de interfaz	19
Requisitos del software.....	19
6. Diseño de la aplicación.....	20
6.1 Diseño de la base de datos	20
6.2 Diseño de interfaces de usuario	24
6.3 Diagramas de actividad.....	30
6.4 Diagramas de secuencia.....	34
6.1 Diagramas de clases	38
7. Conclusiones.....	39
8. Líneas de trabajo futuro.....	40
9. Bibliografía.....	41

Anexo I	43
Anexo II – Manual de Usuario	58
Autenticación de usuario	58
Página principal	61
Administrador	61
Gimansio	78
Monitor	82
Anexo III – Tecnologías Usadas	86
9.1 HTML	87
9.2 CSS3	87
9.3 JavaScript	88
9.3.1 AJAX	88
9.3.2 jQuery	89
9.3.3 jQuery Mapael	89
9.3.4 Morris.JS	89
9.4 JSON	89
9.5 Bootstrap	90
9.6 php	91
9.7 MySQL	92
9.8 MySQL Workbench	93
9.9 NeetBeans	93
9.10 Codeigniter	94
Anexo IV – Requisitos Funcionales	96
Requisitos Funcionales	96
Anexo V – Diagramas de Clases	106
Controlador	106
Modelos	115

Índice de Ilustraciones

Ilustración 3-1 FitnessBliss	12
Ilustración 3-2 Xculpture	13
Ilustración 4-1 Planificación Inicial	14
Ilustración 4-2 Planificación Final	15
Ilustración 6-1 Login	24
Ilustración 6-2 Recuperar Password	25
Ilustración 6-3 Recuperar Password	25
Ilustración 6-4 Nueva Contraseña	26
Ilustración 6-5 Dashboard	26
Ilustración 6-6 Formulario	27
Ilustración 6-7 Listar	27
Ilustración 6-8 Listar Tipo	28
Ilustración 6-9 Nuevo Tipo	29
Ilustración 6-10 Eliminar Tipo	29
Ilustración 6-11 Autenticarse	30
Ilustración 6-12 Crear Administrador	31
Ilustración 6-13 Listar Administradores	32
Ilustración 6-14 Buscar Administradores	33
Ilustración 6-15 Iniciar Sesión	35
Ilustración 6-16 Listar	36
Ilustración 6-17 Crear	37
Ilustración 6-18 Modificar	38
Ilustración 0-1 Sudo	54
Ilustración 0-2 Gimnasio	55
Ilustración 0-3 Monitor	57
Ilustración 0-1 Login	58
Ilustración 0-2 Error	58
Ilustración 0-3 Recuperar Password	59
Ilustración 0-4 Correo	59
Ilustración 0-5 Éxito	59
Ilustración 0-6 Recuperar Password	60
Ilustración 0-7 Restablecer Password	60
Ilustración 0-8 Error	60
Ilustración 0-9 Dashboard Sudo	61
Ilustración 0-10 Crear Administrador	62
Ilustración 0-11 Error	62
Ilustración 0-12 Éxito al crear	63
Ilustración 0-13 Ver Password	63
Ilustración 0-14 Imprimir PDF	63
Ilustración 0-15 Listado Administradores	64
Ilustración 0-16 Buscador Administradores	64
Ilustración 0-17 Exportar a PDF	65
Ilustración 0-18 Ver Administrador	65
Ilustración 0-19 Editar Administrador	66
Ilustración 0-20 Listados	66
Ilustración 0-21 Exportar a PDF	66
Ilustración 0-22 Modal	67

Ilustración 0-23 Acciones	67
Ilustración 0-24 Éxito	67
Ilustración 0-25 Modal	67
Ilustración 0-26 Acciones	68
Ilustración 0-27 Tipos Administradores	68
Ilustración 0-28 Menú Lateral	69
Ilustración 0-29 Países	69
Ilustración 0-30 Mapa	70
Ilustración 0-31 Comunidad	70
Ilustración 0-32 Comunidades	71
Ilustración 0-33 Provincias	71
Ilustración 0-34 Provincias	72
Ilustración 0-35 Deportes	72
Ilustración 0-36 Nuevo Deporte	73
Ilustración 0-37 Éxito	73
Ilustración 0-38 Listado	73
Ilustración 0-39 Edición	74
Ilustración 0-40 Éxito Edición	74
Ilustración 0-41 Modal	74
Ilustración 0-42 Éxito Eliminado	75
Ilustración 0-43 Grupos Musculares	75
Ilustración 0-44 Listado Ejercicios	76
Ilustración 0-45 Buscador Ejercicios	76
Ilustración 0-46 Edición Ejercicios	77
Ilustración 0-47 Edición Ejercicios	77
Ilustración 0-48 Modificación	78
Ilustración 0-49 Listado	78
Ilustración 0-50 Gimnasio	78
Ilustración 0-51 Monitores	79
Ilustración 0-52 Tabla Cliente	79
Ilustración 0-53 Clientes	80
Ilustración 0-54 Insertar Deportes	80
Ilustración 0-55 Ver Clientes	81
Ilustración 0-56 Ver Tablas Cliente	81
Ilustración 0-57 Ver Tablas Cliente	81
Ilustración 0-58 Tablas Vencidas	82
Ilustración 0-59 Consulta Clientes	82
Ilustración 0-60 Modal	82
Ilustración 0-61 Consulta Cliente	83
Ilustración 0-62 Ver cliente	83
Ilustración 0-63 Gestión de tablas	83
Ilustración 0-64 Alta tabla	84
Ilustración 0-65 Alta tabla	84
Ilustración 0-66 Alta Tabla	85
Ilustración 0-67 Calendario Tabla	85
Ilustración 0-68 Ver datos Cliente	85
Ilustración 0-69 Final Tabla	86
Ilustración 0-1 - Sistema de cuadrícula en Bootstrap	90
Ilustración 0-1 Welcome	106

Ilustración 0-2 Drashboar Sudo	107
Ilustración 0-3 Drashboar Gimnasio.....	109
Ilustración 0-4 Drashboard Monitor	109
Ilustración 0-5 Mundo	112
Ilustración 0-6 Deporte	113
Ilustración 0-7 Músculo.....	113
Ilustración 0-8 Ejercicio	114
Ilustración 0-9 Tabla.....	114
Ilustración 0-10 Drashboard sudo model	115
Ilustración 0-11 Drashboar gimnasio model	116
Ilustración 0-12 Drashboard monitor model	116
Ilustración 0-13 Administrador model	117
Ilustración 0-14 General model	118
Ilustración 0-15 Deporte model.....	119
Ilustración 0-16 Músculo model.....	120
Ilustración 0-17 Ejercicio model	120
Ilustración 0-18 Mundo model	121
Ilustración 0-19 Tabla model.....	122

1. Introducción

En este punto se va especificar la motivación y objetivos de la aplicación. También se realizará un pequeño recorrido sobre los apartados que contiene el documento.

Por último, se mostrará un pequeño reconocimiento a las personas que han hecho posible la realización de este proyecto.

1.1 Motivación

Son muchas las personas que han intentado en repetidas ocasiones incorporar el deporte y el ejercicio a su rutina vital, y no menos las que han fracasado en el intento. Pero hoy, cada vez somos más las personas que practican deporte, y esto es porque, el deporte está de moda. El mundo de los gimnasios y las disciplinas deportivas no es ajeno a las modas; aparecen nuevas prácticas, nuevos dispositivos y se reinventan ejercicios para estar a la última.

Además el deporte y la tecnología cada vez van más unidos, desde el calzado deportivo, pulseras cuantificadores, aplicaciones móviles... La creatividad y conocimientos técnicos para concebir material mejor y más seguro en áreas de la excelencia deportiva. Eso se ha traducido en un mejor rendimiento, en un equipo mejor, más seguro y más eficaz para la práctica del deporte, una medición precisa del rendimiento.

Otro aspecto a tener en cuenta son los beneficios de tener los servicios profesionales de un entrenador personal certificado ofrece muchas ventajas frente a hacer ejercicios de forma autodidacta, ya sea en el gimnasio, durante una clase grupal o por cuenta propia, esto se debe a que los monitores están capacitados para evaluar, prescribir y desarrollar planes de entrenamiento físico totalmente personalizados de manera sistemática, ordenada y segura dependiendo de los objetivos y necesidades de cada cliente.

Teniendo en cuenta los puntos anteriores así como mis experiencias en el mundo deportivo me he embarcado en este proyecto, por considerar que con esta aplicación se podrá dar un servicio más personalizado a los clientes de un gimnasio que deseen ponerse en forma, simplificando las tablas que a estos se les proporciona intentado que haya más personas que consideren seguir haciendo deporte, ya que aunque el monitor no esté totalmente centrado en él, aun así pueda sentirse más apoyado por su parte.

Del mismo modo el entrenador podrá llevar un seguimiento más específico sobre todos clientes que estén a su cargo, observando su evolución asistencia así como datos que puedan facilitarle la creación de sus rutinas de entrenamiento.

El ultimo motivo que me ha impulsado a elegir este proyecto es la de usar lenguajes de programación web como HTML, PHP, JavaScript... De esta forma profundizaré más en lenguajes de programación que apenas han sido tratados durante la carrera y así adquirir por mi cuenta conocimientos más detallados de estos.

1.2 Objetivos

El uso de tecnologías web cada vez es más amplio, pero no por ello todas las aplicaciones están bien desarrolladas. Adoptar un patrón de diseño MVC, asegura que esta aplicación pueda seguir desarrollándose en un futuro por otras personas y que no se encuentren con un “caos” en su labor.

En toda aplicación web la base de datos es algo fundamental, por ese motivo debe desarrollarse para que pueda ser adaptada, para un posible crecimiento del sistema, así como para la creación de aplicaciones móviles que complementen la aplicación principal.

Por último los objetivos que se buscan con la aplicación se pueden resumir en cuatro:

1. **Por parte de los administradores:** Llevar una gestión total de los gimnasios y obtener información relevante y bien estructurada de los mismos para ayudar a mejorar cualquier tipo de actividad deportiva en un gimnasio. De forma que se pueda seguir mejorando el producto y ofertando actualizaciones a los gimnasios o centros deportivos.
2. **Por parte de los gimnasios:** El objetivo de estos será tener la información más detallada de los monitores y clientes, asegurándose que los monitores realicen bien su trabajo y sean constantes con el cliente. También obtendrán información de los clientes cierta información necesaria para los monitores así como otras actividades que puedan realizar en conjunto con un gimnasio.
3. **Por parte de los monitores:** Llevar un control más personalizado y diario de los clientes que le han sido asignados o ha decidido controlar el. De forma que este le cree rutinas y sea consciente siempre de lo que un cliente está haciendo.
4. **Por parte de clientes:** Sentirse apoyado por un monitor a la hora de realizar su actividad como si de un entrenador personal se tratará. Este tendrá una información más clara y detallada de la actividad que debe realizar en el día y de esta forma no andar perdido, sin saber qué es lo que debe de hacer.

1.3 Visión general del documento

En este punto se va a explicar todo los apartados que nos podremos encontrar a lo largo del documento.

- **Tecnologías utilizadas:** Aquí se van a detallar todos los lenguajes de programación así como las herramientas usadas para el desarrollo de la aplicación.
- **Estado del arte:** Se pretende comparar la aplicación realizada con otras disponibles en el sistema, y compararlas con el fin de observar si la nuestra ofrece algo diferente a las ya existentes.

- **Planificación:** En este apartado se mostrará la planificación inicial de todas las tareas del proyecto, así como su cambio tras la finalización.
- **Análisis:** Contiene todas las restricciones que ha de cumplirse para su correcto desarrollo.
- **Diseño de la aplicación:** Se muestra la aplicación de una forma mucho más detallada.
- **Conclusiones:** En este apartado se pretende examinar el trabajo realizado, sacando una serie de opiniones.
- **Líneas de trabajo futuro:** Se detallarán posibles mejoras y ampliaciones de este proyecto, así como ideas para trabajos futuros.
- **Bibliografía:** Se expondrán los lugares de los cuales hemos extraído información para el desarrollo del proyecto.
- **Anexos:** Documentos complementarios al principal.

1.4 Agradecimientos

Hay ciertas personas y empresas que han hecho posible la creación de este proyecto, por ello me gustaría citarlas reconociéndole su labor en la aplicación.

En primer lugar agradecer a la empresa “**Starglob Soluciones Informáticas**” por dejarme formar parte de su equipo de desarrolladores durante la creación de este proyecto. Esta empresa está formado por una serie de miembros que en mayor en menor medida han aportado algo a este proyecto y sin ellos no hubiera sido posible. A continuación se detalla las pequeñas aportaciones:

- **Gabriel Fuertes Muñoz:** Fundador de la empresa y director de este proyecto. Ha puesto su confianza en mí y me ha ayudado y apoyado en todo momento.
- **Javier Fuertes Latorre:** Me ha ayudado en sus inicios a moverme en el mundo php, además hay que destacar su ayuda por los conocimientos aportados sobre bases de datos.
- **Franks Edison González Landero:** Apoyarme en todo momento y ayudarme cuando así lo he necesitado.
- **Luis Cote Sáez:** Me ha aportado amplios conocimientos sobre el mundo deportivo, además de preocuparse por que la aplicación salga adelante.

Por último, me gustaría agradecer a mi familia por apoyarme durante la creación de este proyecto. Sin ellos nada de esto hubiera sido posible

2. Tecnologías usadas

Con el fin de realizar una exposición más profunda de las tecnologías usadas, debido al número de ellas, las englobaremos en el Anexo III.

3. Estado del arte

Antes de comenzar con los detalles de la aplicación se ha realizado una búsqueda en internet, con el fin de ver productos similares y disponibles en el mercado. De manera que podamos ver diferencias y por así decirlo ver los retos a los que se debe presentar esta aplicación.

3.1 FitnessBliss

Este sitio dispone de más de 750 rutinas y ejercicios de entrenamiento perfectamente explicadas, que puedes llevar a cabo en casa.

Es una plataforma multi-idioma, incluyendo el español que no requiere que descargues o instales ninguna aplicación, sólo necesitas un navegador. Incluso dispone de una versión adaptada para los navegadores de los teléfonos móviles por si accedes desde cualquiera de estos dispositivos.

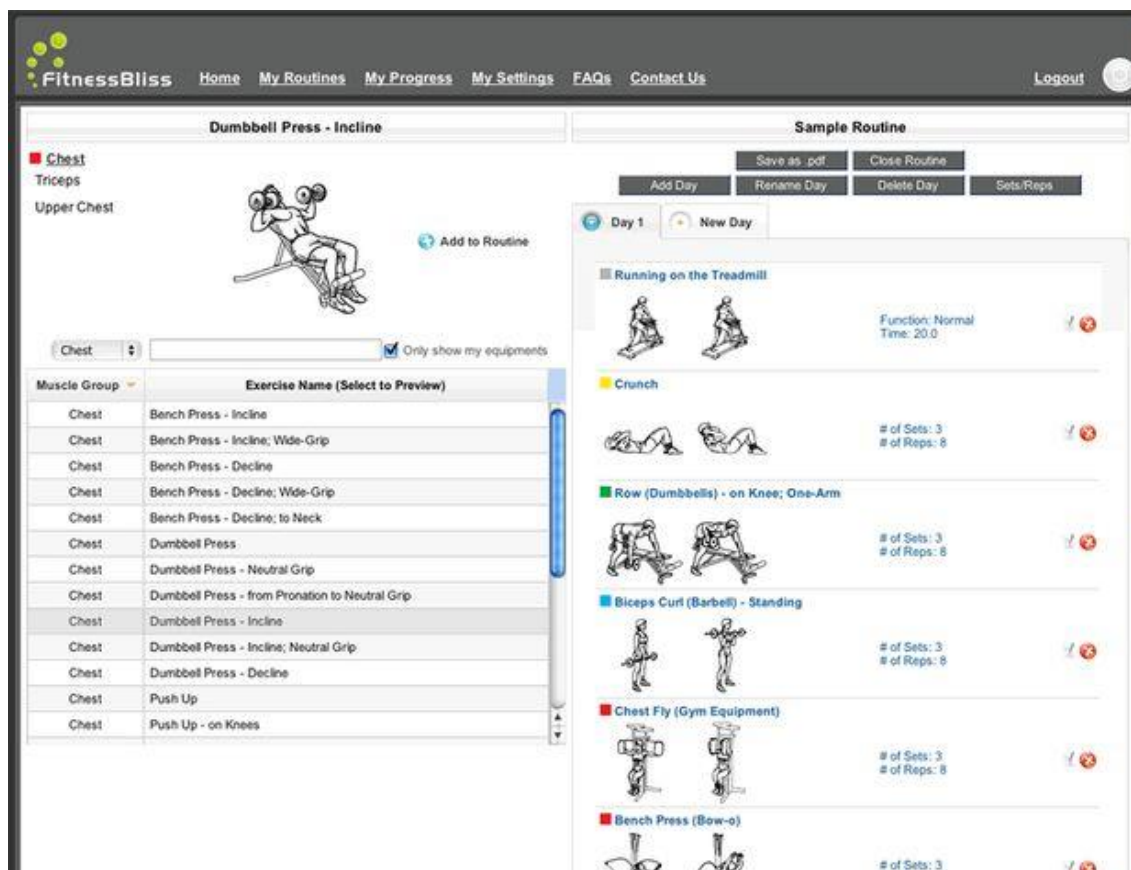


Ilustración 3-1 FitnessBliss

Como se puede observar esta página web persigue objetos similares a los que ofrece nuestra aplicación. Sin embargo, esta solo permite la creación y seguimiento de tablas.

3.2 Xculture

Esta aplicación consta de página web y aplicación móvil. Estas nos permiten crear nuestros propios ejercicios, ver la evolución diaria y conoce a otras personas dedicadas al fitness en la red social de Xculpture, en ella los aficionados y “profesionales” te ayudan y comparten sus conocimientos.

Con Xculpture podrás hacer tus rutinas y ejercicios para poder utilizarlos en el gimnasio. Además, dispone de una herramienta donde se puede seguir tus rutinas paso a paso y guardar los resultados de tu entrenamiento sincronizándolos con el sitio web.



Ilustración 3-2 Xculpture

3.3 Conclusiones

Si bien es cierto que el mercado hay aplicaciones que permiten crear tablas personalizadas. Hemos visto una serie de diferencias importantes respecto a nuestra aplicación. Estas diferencias se exponen a continuación:

- En las aplicaciones web encontradas, el cliente es quien se hace las rutinas, mientras que en nuestra aplicación es un monitor experimentado el que las hace. Este te recomendará los mejores ejercicios según tu condición física actual y tus objetivos.
- Con nuestra aplicación los monitores son los encargados de llevar un control total de todos y cada uno de los clientes. En las aplicaciones observadas es el cliente la persona encargada de hacerlo.

Considero que el uso de las herramientas vistas debería ser usado por expertos. Un usuario normal no sabe que es lo que tiene que hacer, y esto puede dar lugar a lesiones. Por este motivo pienso que estas herramientas no tendrían que ser usadas por personas que se estén iniciando, ya que pueden llegar a ser lesivas.

4. Planificación

En este apartado se pretende mostrar cómo ha sido afrontada la creación de este proyecto. Para ello se hizo una planificación inicial, con el fin de poder tener unas pautas a seguir. De esta manera se decidió que la elaboración de esta aplicación así como toda su documentación debía durar únicamente cuatro meses trabajando en ella diariamente.

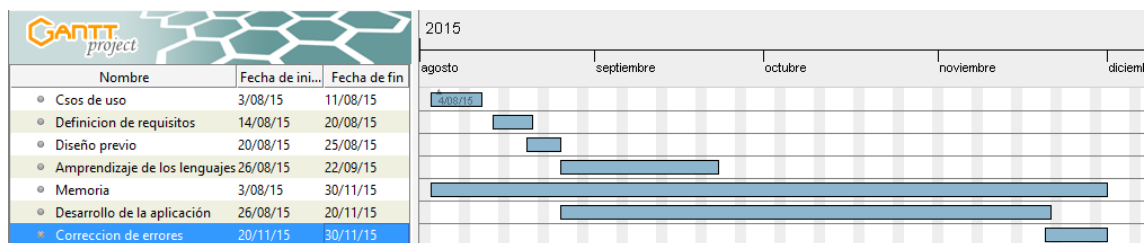
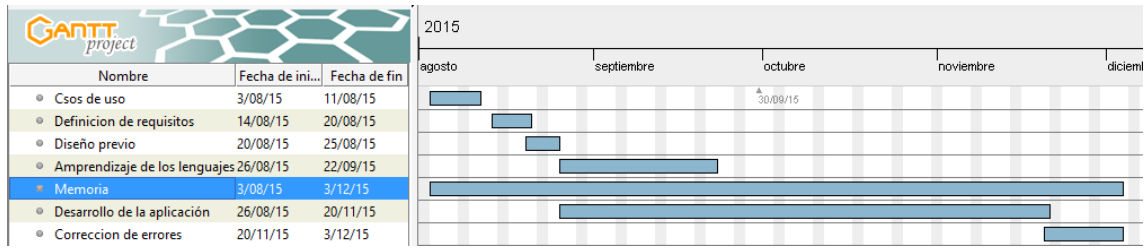


Ilustración 4-1 Planificación Inicial

Las tareas que se decidieron añadir en la planificación son las que quizás más tiempo lleven de elaboración. Globalizando otras tareas menos repetitivas como “memoria”. Todas ellas se detallan a continuación

- **Casos de uso:** descripción de los pasos o las actividades que deberán realizarse para llevar a cabo algún proceso. Los personajes o entidades que participarán en un caso de uso se denominan actores.
- **Requisitos:** Contiene todos a un nivel de detalle suficiente como para permitir a los diseñadores elaborar un sistema que satisfaga estos requisitos, y que permita al equipo de pruebas planificar y realizar las pruebas que demuestren si el sistema satisface, o no, los requisitos.
- **Aprendizaje de los lenguajes:** Aquí veremos los conceptos más básicos acerca de la web y nos iniciamos en el mundo del desarrollo de aplicaciones web. Aprenderemos sobre los lenguajes necesarios, como HTML, CSS y Javascript...
- **Diseño previo:** Elaboración de una serie de bocetos, con el único objetivo de que el programador sepa cómo tiene que ser la interfaz de usuario
- **Memoria:** Documento explicativo, que contiene todo lo referente a la aplicación, con un nivel de detalla muy alto.
- **Desarrollo de la aplicación:** Elaboración de la misma, usando el documento redactado y lo aprendido anteriormente.
- **Corrección de errores:** Tras el desarrollo de la aplicación, tiene que haber un periodo de prueba con el objetivo de detectar y solucionar la mayor parte de errores.

Al terminar la elaboración de la aplicación se puede observar como los plazos han sido cumplidos a excepción de la memoria y la corrección de errores que se alargaron algo más apurando la fecha límite para la entrega final.



5. Análisis

En este punto se va a detallar y especificar la aplicación GymGest a un nivel más detallado.

5.1 Propósito

A continuación se van a detallar todos los requisitos funcionales y no funcionales, así como las restricciones que la aplicación deberá satisfacer en todos los aspectos expuestos en los siguientes puntos para la correcta creación de la aplicación.

Este apartado está enfocado a todas las personas interesadas en conocer profundamente el funcionamiento de la aplicación.

5.2 Alcance

La meta es conseguir un diseño, desarrollo e implementación de aplicación. Esta es una aplicación de gestión que funcionará online, es decir, será una aplicación web.

La aplicación debe permitir crear y gestionar de manera sencilla las cuentas de los administradores, gimnasios, monitores y clientes.

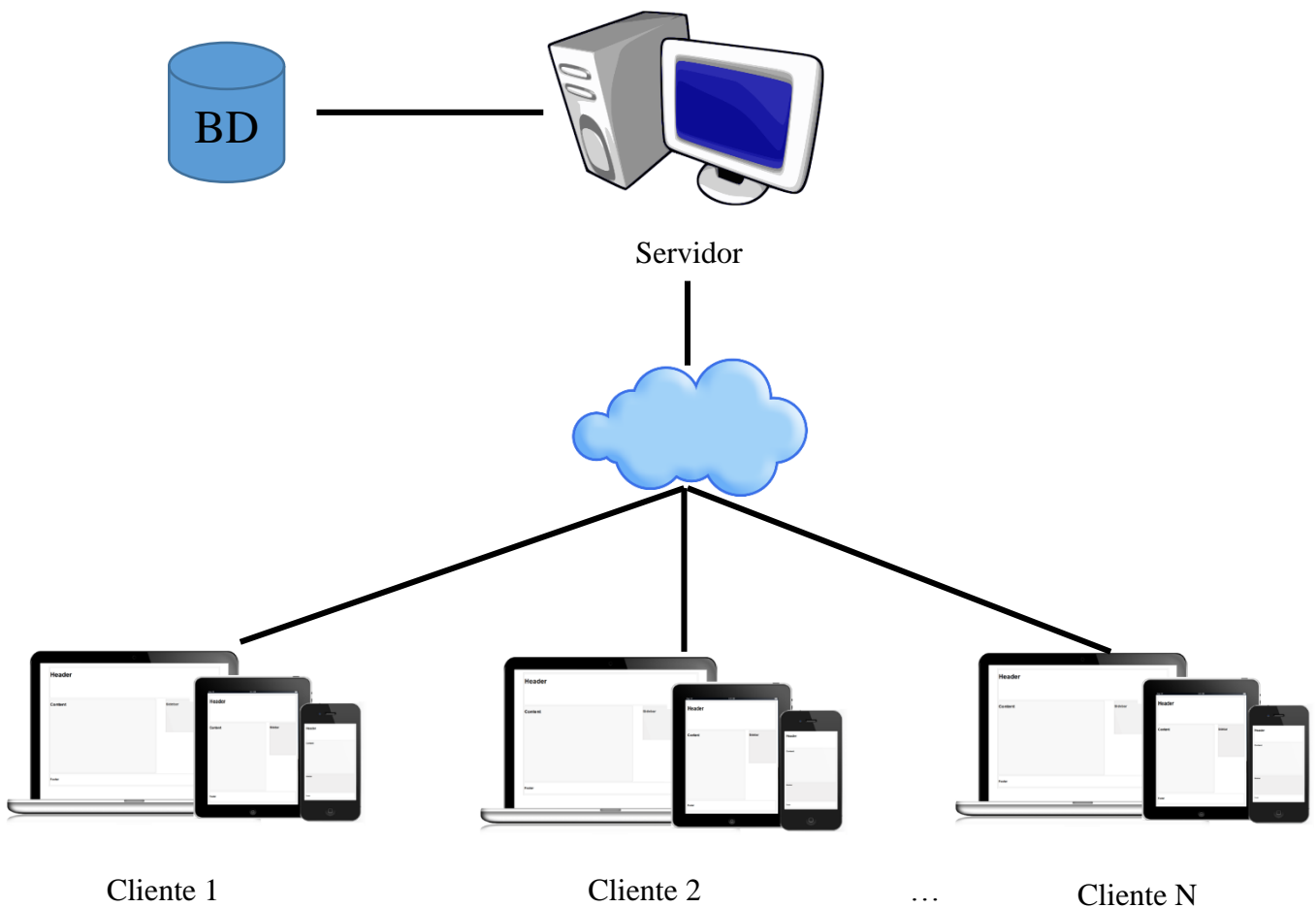
La gestión de administradores y gimnasios será realizada por los administradores principales. De igual manera el gimnasio deberá gestionar monitores y clientes. Por último el monitor gestionará únicamente las rutinas deportivas a los clientes.

5.3 Perspectivas del producto

La aplicación a desarrollar, se puede subdividir en tres partes las cuales se detallan a continuación.

1. **Clientes:** Inicialmente el sistema solo estará preparado para que puedan acceder tres tipos de clientes. Estos podrán acceder al sistema usando cualquier navegador web. Los clientes se detallan a continuación:
 - a. **Administrador.**
 - b. **Gimnasio.**
 - c. **Monitor.**
2. **Servidor:** Este es el encargado de gestionar todas las comunicaciones por parte de los usuarios, interactuando también con la base de datos.
3. **Base de datos:** En esta se almacena toda la información referente a los clientes así como datos que ayudan a la gestión de los mismos.

La estructura general de la aplicación es la siguiente, en ella se observa como el servidor hace de puente intermedio entre los clientes y la base de datos. Siendo el servidor el que suministra la información solicitada por los clientes.



5.4 Funciones del producto

En términos generales, la aplicación deberá abordar los siguientes casos de uso.

5.4.1 Casos de Uso

Debido al elevado número de casos de uso presentes los agruparé todos en el Anexo I.

5.4.2 Características de los usuarios

Los usuarios que accedan a la aplicación deberán estar familiarizados únicamente con el uso de un navegador web, ya que toda la aplicación se usa a través de este.

El usuario es la persona encargada de interactuar continuamente con la aplicación. Los usuarios con acceso al sistema son tres:

- **Administrador principal:** Se encarga de la gestión total del sistema, pudiendo crear otros administradores que le ayuden así como dar de alta gimnasios para que puedan acceder al sistema, además podrá gestionar todos los tipos de ejercicios que este podrá utilizar.
- **Gimnasio:** Encargado de controlar y gestionar todos sus monitores y clientes.
- **Monitor:** Este podrá visualizar y asignar tablas a los clientes, además podrá llevar un seguimiento completo sobre cada una de las rutinas asignadas al cliente.

5.4.3 Suposiciones y dependencias

La aplicación será implementada bajo los lenguajes de programación HTML, CSS3, JavaScript, AJAX, JQuery... Esto significa a la aplicación no es dependiente de ninguna plataforma en concreto, si no que únicamente es dependiente del uso de un navegador web.

5.4.4 Evolución previsible del sistema

Se prevé que la aplicación pueda ser mejorada en un futuro, dotándola a la misma de más funcionalidades así como el desarrollo de una aplicación para dispositivos móviles, la cual permita un uso más adaptado.

5.5 Especificación de requisitos

Al igual que ocurre en los casos de uso y en las tecnologías usadas, debido al gran número de requisitos del proyecto los agruparemos todos en el Anexo IV.

Requisitos de interfaz

- **RI01:** La aplicación web deberá ser adaptativa.
- **RI02:** La interfaz seguirá ciertos criterios de usabilidad y accesibilidad para que sea intuitiva y de fácil acceso.
- **RI03:** La página siempre que se pueda deberá estar dotada de patrones de interacción ya adoptados por la sociedad.
- **RI04:** Cada usuario accederá únicamente a sus tareas.
- **RI05:** La página no dispondrá de logo, en su lugar habrá unas letras con el nombre de aplicación. Cuando estas sean pulsadas se redirigirá al usuario al dashboard.
- **RI06:** Se respetará el principio de diseño de software KISS (“Keep It Simple, Stupid” en español “Mantenlo simple, estúpido”).
- **RI07:** El acceso a la secciones se hará desde un menú en la parte izquierda.
- **RI08:** La sesión se podrá cerrar en cualquier momento desde el menú superior.
- **RI09:** Las filas de las tablas se diferenciarán con colores para una mejor visualización.
- **RI10:** Se usará colores relajantes.

Requisitos del software

- **RS01:** Se accederá a través de un navegador.
- **RS02:** La aplicación únicamente funcionará cuando se disponga de acceso a internet.
- **RS03:** Solo podrán acceder a la aplicación aquellos usuarios que estén dados de alta.
- **RS04:** El usuario podrá acceder a las funciones siempre y cuando este logueado. De lo contrario será redirigido a la página de Disney.
- **RS05:** El sistema garantizará el cifrado de las contraseñas de usuario.

6 Diseño de la aplicación

En este punto se pretende detallar como debe estar diseñada la aplicación. Se especificará de manera clara para que el programador pueda llevar a cabo su tarea. Se iniciará explicando el funcionamiento de la base de datos, ya que a partir de aquí se desarrollará el resto de la aplicación. Por último se detallarán aspectos clave de la aplicación, además de cómo debe estar construida.

6.1 Diseño de la base de datos

La base de datos es el pilar fundamental de la aplicación al igual que ocurre con muchas de las páginas web. Es por ello que la aplicación se puede resumir en un conjunto consultas realizadas por el usuario en función de su necesidad, siendo la aplicación una forma de visualizar la información de las tablas de la BD de manera clara y ordenada.

Por este motivo a continuación se va detallar las distintas tablas existentes en la base de datos, intentando minimizar al máximo todas las posibles dudas.

Tabla País

Contiene los nombres de los países, así como sus coordenadas. Es decir, contiene su latitud y longitud. Además se ha creado un campo por si en un futuro se deseará introducir su código “**Alpha-3 Code**”, Este campo hace referencia a la abreviatura de un país con tres caracteres.

Ejemplo: El código “Alpha-3 Code” de ESPAÑA es ESP.

Tabla ccaa

Esta tabla contendrá nombres de las comunidades autónomas. Cada comunidad posee una clave ajena a la tabla países, es decir, una comunidad pertenece a un país.

Tabla provincia

Es la encargada de almacenar todas las provincias de una comunidad autónoma. Esta tabla contendrá el nombre de la provincia, así como una clave ajena a la tabla ccaa.

Para resumir de manera simple las tres tablas anteriores se puede decir que una provincia pertenece a una comunidad autónoma, y a su vez una comunidad pertenece a un país.

Tipo de administrador

Almacena todos los tipos de administradores que pueden gestionar la aplicación. A un administrador se le asignará posteriormente un tipo de administrador.

Cada tipo de administrador tendrá un cierto número de privilegios en el sistema. Como ya se ha mencionado en otros apartados, inicialmente solo habrá un tipo de administrador. Por este motivo se puede decir que esta tabla en su inicio tendrá una fila con el nombre de *Sudo* (Administrador total).

Esta tabla ha sido creada pensando en un posible crecimiento del sistema. De esta manera, si fuera necesaria la contratación de nuevos administradores, podríamos mandarles faenas menos críticas eliminando el acceso a datos que no le sean relevantes.

Tabla tipo de gimnasio

Contiene los nombres de los tipos de gimnasio que pueden crearse. Estos ya han sido definidos anteriormente (CU TIPOS GIMNASIO).

Tabla Tipo de deporte

Esta tabla contiene los nombres de todos los deportes que existan en la actualidad, como podría ser, futbol, boloncesto, tenis, karate...

Estos deportes se le podrán asignar a un cliente, como ya iremos viendo más adelante.

Tabla Deporte

Esta tabla sirve como puente intermedio entre un cliente y un tipo de deporte. Esta, contiene dos claves ajenas una con el id del cliente y otra con el id de deporte. Por ello se puede decir que por cada uno de los deportes que practique el usuario habrá una fila en esta tabla.

Tabla Articulación

Esta tabla está pensada para varias cosas, por eso puede ser que el nombre no sea el más adecuado según el uso inicial de la aplicación.

Inicialmente contendrá los nombres de todos los grupos musculares existentes a la hora de trabajar en un gimnasio. Aunque en un futuro podrá contener el nombre de las articulaciones con el fin de saber si el usuario sufre dolores en alguna articulación.

Tabla Entrenamiento

Contendrá los nombres de las tablas creadas por los monitores, además del nombre de la tabla tendrá su fecha de inicio, fecha de finalización, el identificador del cliente al que pertenece, el identificador del grupo muscular y si esta activa o no.

Entrando un poco más en detalle esta tabla tendrá una fila por cada grupo muscular que trabaje el cliente. Es decir, si un monitor crea una tabla llamada "mi_primera_tabla", y quiere el usuario haga cardio y biceps. Tendrá dos filas. Ejemplo:

Nombre	Fecha	Fecha_cierre	Fk_cliente	Fk_articulación	cerrar
mi_primera_tabla	2015-11_01	2015-11_30	1	1	0
mi_primera_tabla	2015-11_01	2015-11_30	1	2	0

Tabla sesión

Esta tabla la podemos dividir de dos maneras.

- En caso de que sea una tabla générica el "numero_sesión" corresponderá con el día de la semana que debe realizarse el ejercicio, es decir, si es lunes, el ejercicio se hará el primer día, que en el caso de la programación correspondería con el día 0, además tendrá una clave ajena al entrenamiento que pertenece dicha sesión.
- Si por el contrario es una tabla diaria, en el campo fecha se detallará la fecha del día justo en el que se tiene que trabajar el grupo muscular. Además tendrá una clave ajena a un horario, este se especifica en la tabla horario.

Tabla ejercicio

Esta tabla tiene todos los ejercicios que debe realizar un cliente, para ello se especifica la sesión en la que debe realizar el ejercicio, también tendrá una clave ajena al ejercicio a realizar (el ejercicio a realizar se detalla en la tabla descripción_ejercicio). Por último cada ejercicio tiene información sobre cómo debe realizarse. Estos datos son, número de series, repeticiones, peso recomendado...

Tabla Administrador, gimnasio, monitor y cliente.

Estas tablas se definen todas juntas debido a su grán similitud. Todas ellas contienen información sobre un usuario. Estos datos se pueden clasificar de la siguiente manera:

- **Datos sobre el usuario:** Nombre, apellidos, DNI, teléfono, email...
- **Datos sobre situación geográfica:** Pías, comunidad, provincia, localidad.
- **Datos para el sistema:**
 - **Contraseña:** Esta se almacenará de manera cifrada. De esta manera, si la información de un administrador es robada, no se podrá acceder a la aplicación.
 - **Numero_de_acceos:** Contiene el número de accesos fallidos al sistema.
 - **Fecha_quinto_acceso:** Se almacena la hora del último fallo producido al intentar acceder al sistema (Complementa el punto anterior).
 - **Deshabilitar:** En caso de que este campo contenga un 0 el usuario puede acceder al sistema, si por el contrario tiene un 1 este estará deshabilitado.
 - **url_recuperar_pass:** Contiene el código que se le ha enviado a un usuario que intenta recuperar su contraseña.
 - **Fecha_url_repucperar_pass:** Contiene la hora en que fue enviado el código (Este campo complementa el punto anterior).
 - **Leido:** Un usuario creado recientemente tendrá este campo a cero, hasta que un administrador del sistema visualice sus datos.

6.2 Diseño de interfaces de usuario

En este punto se van mostrar los prototipos de interfaz para que el diseñador pueda observar cómo deben ser cada una de las interfaces que se van a mostrar al usuario. Se pide a los diseñadores que se adapten de la mejor manera posible.

Se sabe que la interfaz es un punto muy importante en todo proyecto, ya que es la forma en la que el usuario maneja el sistema y de ella depende en gran medida el éxito o el fracaso de la aplicación.

Registro de usuario

A hand-drawn prototype of a web browser window titled "A Web Page". The address bar shows "http://www.GymGest.com". The main content area displays a login form for "GymGest". The form has a title bar, followed by "Nombre de usuario" with a text input field, "Contraseña" with a text input field labeled "Password", and two links: "Recuperar contraseña" and "Codigo de recuperación de contraseña". At the bottom of the form is a "Registrar" button.

Ilustración 6-1 Login

En esta interfaz el usuario podrá meter el nombre de usuario y contraseña para acceder al sistema.

En caso de olvido de la contraseña por parte de un usuario, este podrá recuperarla usando para ello los enlaces que hay justo debajo de la casilla password. Las interfaces de recuperación se detallan a continuación.

Recuperar contraseña

La recuperación del password se divide en unos sencillos pasos:

Paso 1: El usuario introducirá en la siguiente pantalla el correo electrónico para que se le envíe el código oportuno para poder restablecer su contraseña.



Ilustración 6-2 Recuperar Password

Paso 2: Una vez el usuario disponga del código de recuperación de contraseña, deberá introducirlo junto a su correo electrónico en la siguiente interfaz.

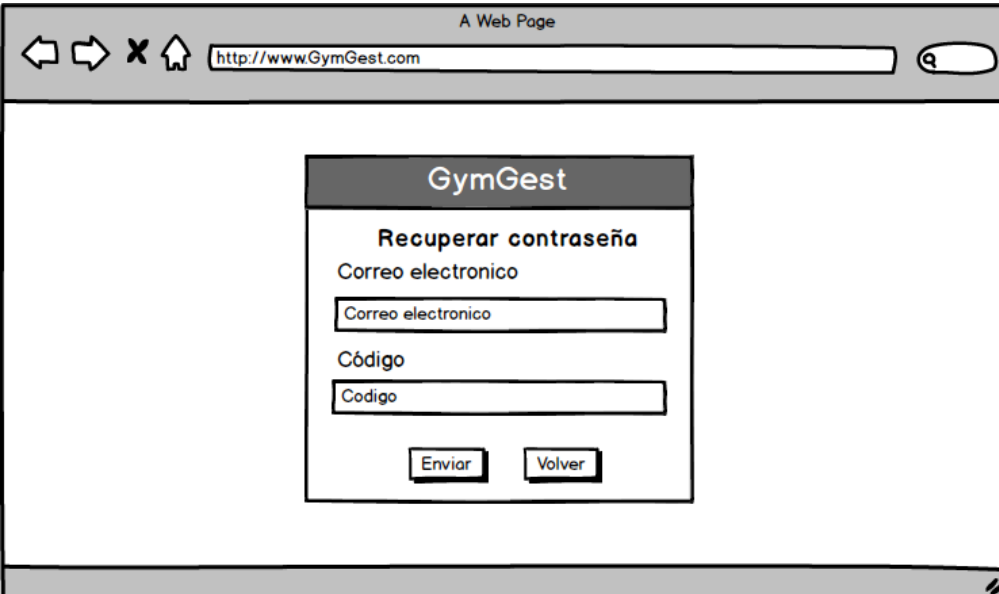


Ilustración 6-3 Recuperar Password

Paso 3: Si el correo electrónico y contraseña introducidos en el paso 2 son correctos, aparecerá la siguiente pantalla.

A Web Page

http://www.GymGest.com

GymGest

Nueva contraseña

Correo electronico

Contraseña

Repetir contraseña

Enviar Volver

Ilustración 6-4 Nueva Contraseña

En esta pantalla, el usuario podrá introducir su nueva contraseña. Para evitar equivocaciones deberá hacerlo dos veces.

Las interfaces que se van a mostrar a continuación corresponden con la gestión de administración.

El fin no es mostrar al programador todas las interfaces de la aplicación, si no que se pueda hacer una idea general de cómo tienen que ser las interfaces.

Dashboard

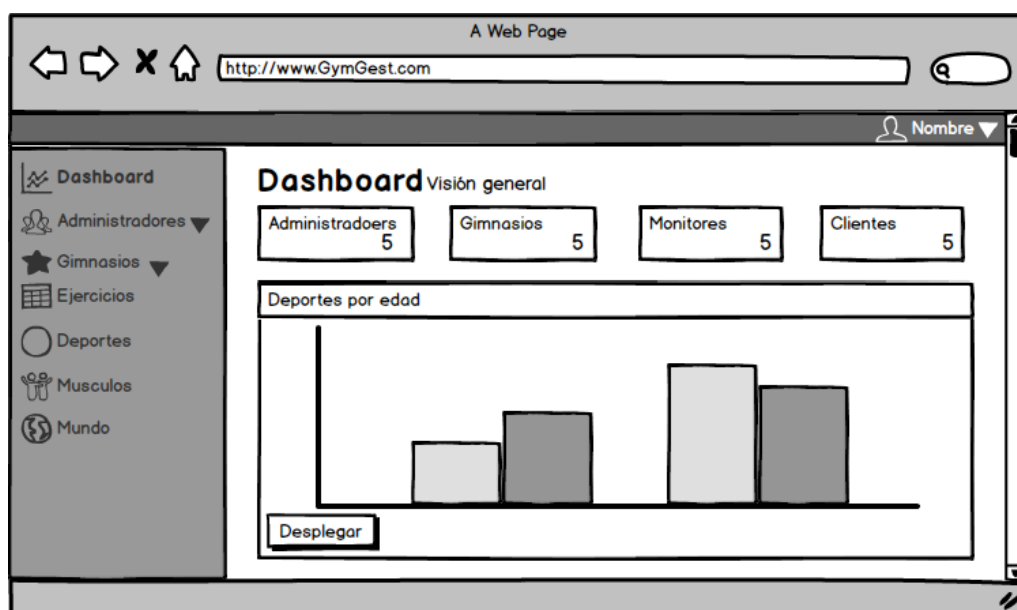


Ilustración 6-5 Dashboard

Esta imagen corresponde con la página principal de los administradores. En ella se podrá observar todo lo que ocurre en el sistema. De esta forma los administradores serán conscientes de todo lo ocurrido y tengan una buena información para poder seguir mejorando el sistema y la experiencia con el usuario.

Formulario para introducir datos en la BD

Ilustración 6-6 Formulario

En esta ilustración se puede ver los diferentes campos que debe tener un formulario; como se puede observar, hay diferentes formas de introducir los datos; de manera que se pretende minimizar la introducción de errores por parte de los administradores.

Listar datos









Nombre	Pais	Activo	Acción
Luis Perez	España	Si	   
Mario Fernandez	España	NO	   

Ilustración 6-7 Listar

El dibujo anterior pertenece a una lista de administradores. En ella se puede observar como tenemos un buscador, un botón para generar un fichero PDF con todos los administradores y una paginación justo al terminar la tabla.

Por otra parte, en la tabla se puede apreciar como hay un campo llamado “acción”; este campo contiene diversas opciones que un administrador puede realizar sobre cada uno de los usuarios. En este caso las acciones a realizar son: ver, modificar, generar documento PDF y dar de alta o de baja en función de si el usuario está o no activo en el sistema.

Listar tipos



Ilustración 6-8 Listar Tipo

Esta imagen únicamente tiene una tabla y un botón. Al no tener gran cantidad de información, casi todo se puede hacer de forma dinámica. (Estas las veremos en las siguientes ilustraciones).

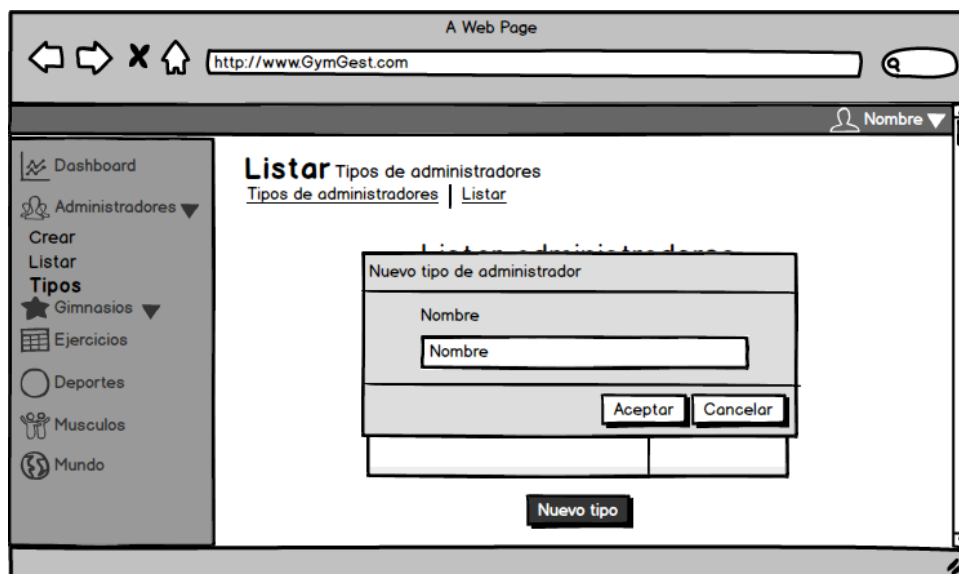


Ilustración 6-9 Nuevo Tipo

Podemos observar cómo ha sido pulsado el botón “nuevo tipo”. Ya que para crear un nuevo tipo de administrador solo debemos rellenar un campo, el formulario se hará en una ventana de este tipo, es decir, en una ventana modal.

El administrador en este caso tiene dos opciones: la primera corresponde con aceptar y enviar los datos introducidos y la segunda sería cancelar cerrando de esta manera la ventana.



Ilustración 6-10 Eliminar Tipo

En la imagen anterior se aprecia cómo se ha pulsado el botón de eliminación, ubicado en la tabla.

Para eliminar, dar de baja o dar de alta cualquier dato el usuario siempre tendrá que aceptar esta ventana. Se hará así para evitar que se eliminen datos de manera equivocada.

6.3 Diagramas de actividad

A continuación se muestran los diagramas de actividad más relevantes del diseño. Gran parte de los subcasos de uso realizan las mismas actividades, por lo que únicamente se presentan algunos diagramas.

Autenticarse

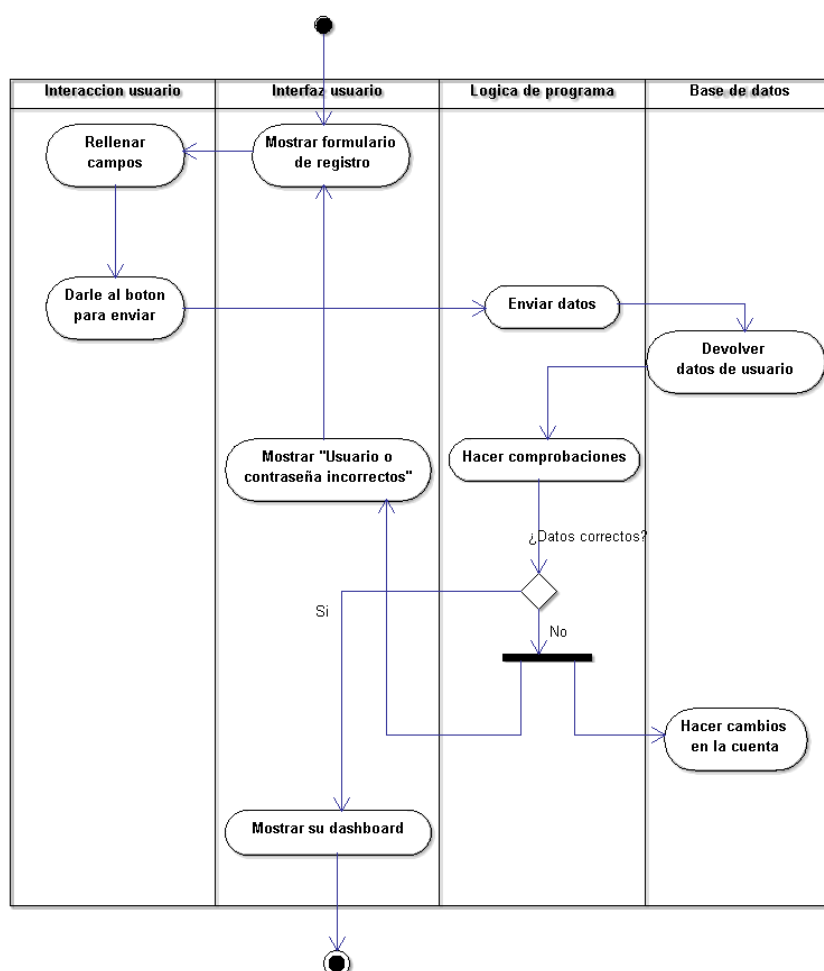


Ilustración 6-11 Autenticarse

Este diagrama corresponde con el subcaso de uso CU01: INICIAR SESIÓN - Se mostrará el formulario de inicio de sesión al usuario. Una vez el usuario rellene los campos oportunos y pulse sobre el botón Registrar, se enviarán los datos al servidor, este pedirá a la BD, los datos de este usuario. Si el usuario existe hará las siguientes comprobaciones:

- Comprobará que el usuario este activo.
- Acto seguido comprobará que no tenga más de 5 accesos fallidos al sistema.

- Por último, cifrará y comparará la contraseña que ha introducido el usuario con la que ha recibido de la base de datos.

Si todos estos pasos se hacen de forma correcta, el usuario tendrá acceso al sistema, llevándolo a su dashboard. Si por el contrario se produce algún error, se le mostrará un mensaje al usuario indicándole lo ocurrido.

Crear administrador

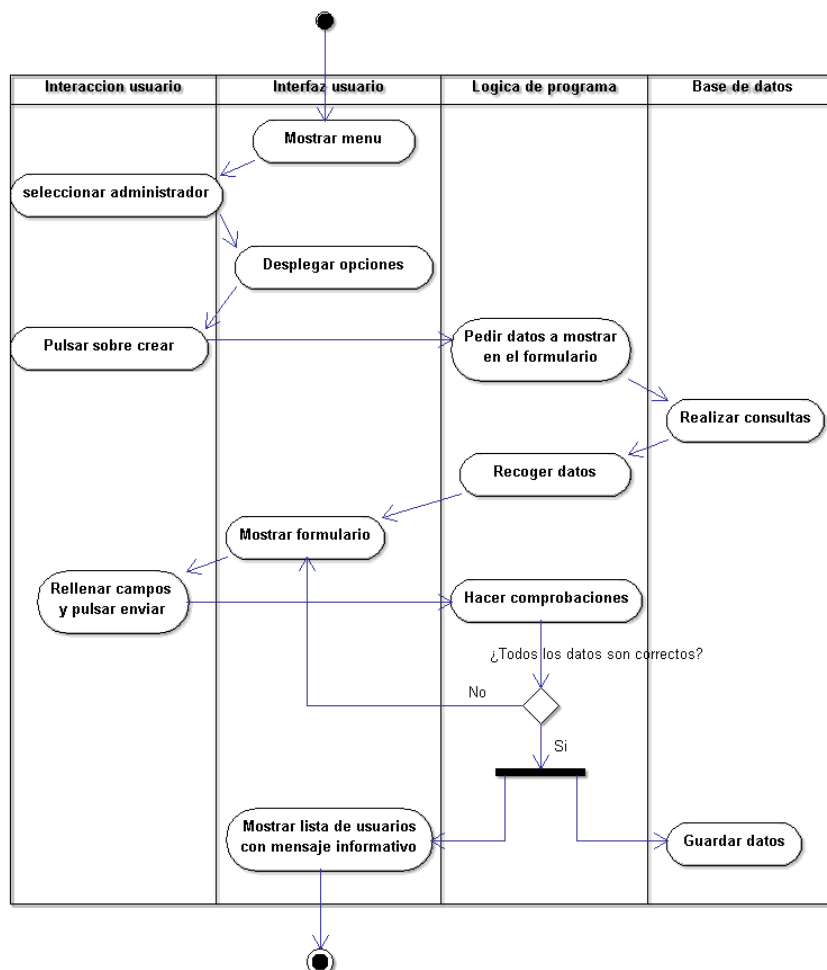


Ilustración 6-12 Crear Administrador

Este diagrama corresponde con el subcaso de uso CU04.5: CREAR ADMINISTRADOR – Una vez el usuario tenga el menú izquierdo en pantalla, podrá seleccionar “crear administrador”, tras esto se enviará una orden al servidor y este recopilará la información a mostrar (Esta puede ser países, comunidades, provincias...), una vez tenga toda la información, se mostrara un formulario para que el usuario pueda introducir, seleccionar, y enviar los datos.

Tras la orden de envío el servidor recogerá toda la información enviada por el formulario y la comprobará, en caso de que sea correcta será almacenada en la base de datos y se mostrará la lista de usuarios con un mensaje indicándole que el usuario se ha creado con éxito, si por lo contrario ha habido algún error, se mostrará el formulario con

los datos que había enviado el administrador y un mensaje de error indicándole los problemas para que pueda solucionarlos.

Listar administradores

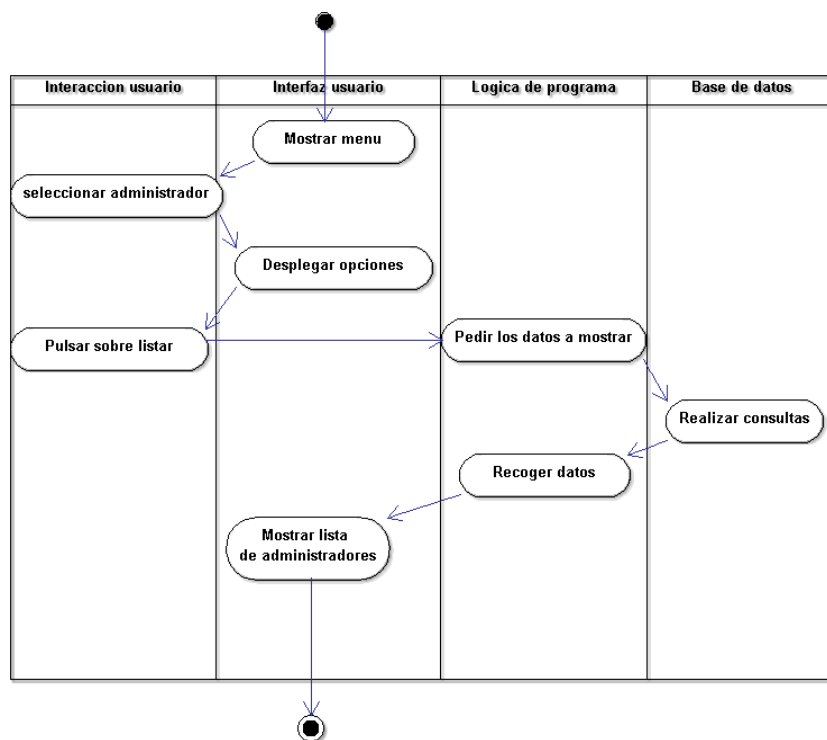


Ilustración 6-13 Listar Administradores

El diagrama anterior corresponde con el subcaso de uso CU04.6: LISTAR ADMINISTRADORES - Para que un administrador pueda seleccionar los administradores del sistema, este deberá de pulsar sobre listar en el desplegable de administradores, una vez hecho esto el sistema recogerá la información sobre los diez primeros administradores existentes en la base de datos, una vez recogida dicha información se le mostrará al administrador en una tabla.

Buscar administrador

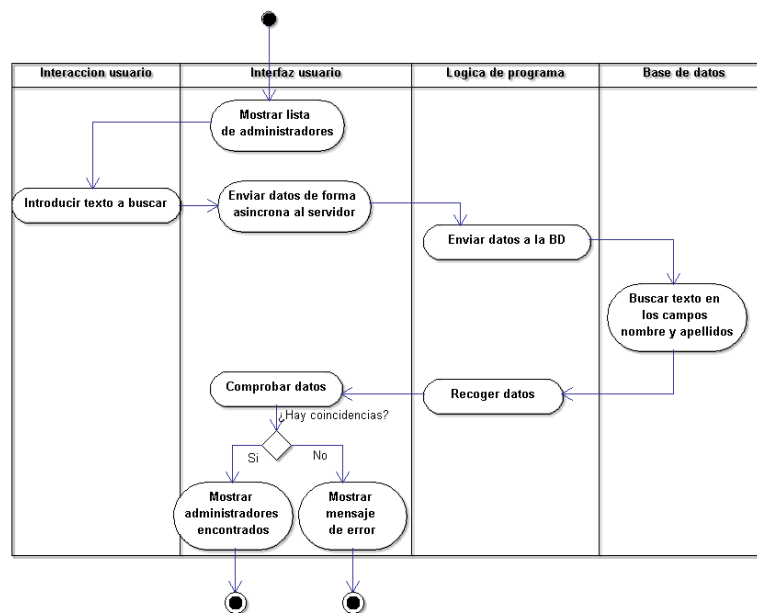


Ilustración 6-14 Buscar Administradores

El diagrama de la imagen anterior corresponde al subcaso de uso CU04.7: BUSCAR ADMINISTRADOR – Este quizás es uno de los más complejos, el usuario cuando este en la vista listar y desee buscar un administrador, tan solo tendrá que introducir unos caracteres o una palabra en el buscador, tras esto (un código AJAX sin cambiar de interfaz) solicitará al controlador que busque coincidencias en la BD, después de la consulta en la BD, el controlador enviará todas las coincidencias. En caso de que no hayan se mostrará un mensaje indicándoselo al usuario (en el lugar donde estaba la tabla), si por el contrario hay usuarios encontrados, se listarán en la tabla para que los pueda ver el administrador.

6.4 Diagramas de secuencia

En este punto procedemos a presentar algunos de los diagramas de secuencia más relevantes para el diseño de nuestra aplicación. Cada uno de los elegidos realiza una función de gestión algo compleja. Estos serán mostrados a continuación.

Iniciar sesión (Para el usuario administrador).

Diagrama correspondiente al caso de uso CU01: INICIAR SESIÓN - Cuando el usuario este situado en la interfaz de registro deberá introducir su correo electrónico y contraseña, este pulse el botón “Registrar” se invocarán los siguientes métodos:

- **Validar:** Encargado de recoger los valores del formulario y llamar a todos los submétodos que le ayudan a comprobar si el administrador puede entrar al sistema.
 - **es_administrador:** Comprueba si el usuario introducido existe en la tabla de administradores de la BD, para ello invoca el método “datos_administrador” este en caso de que exista, devolverá sus datos con los que se harán una serie de comprobaciones. Si el administrador existe y puede acceder al sistema se devolverá “TRUE” al método principal.
 - **iniciar_sesion_administrador:** Recibirá como parámetro el nombre del administrador que está accediendo al sistema. Este recogerá sus datos en la BD e invocará al método “datos_administrador”, con esos datos se iniciará una sesión.

Por ultimo reseteará los accesos fallidos al sistema usando para esto el método “actualizar_quinto_acceo”. Una vez actualizados los datos llamara al controlador “Dashboard_sudo”.

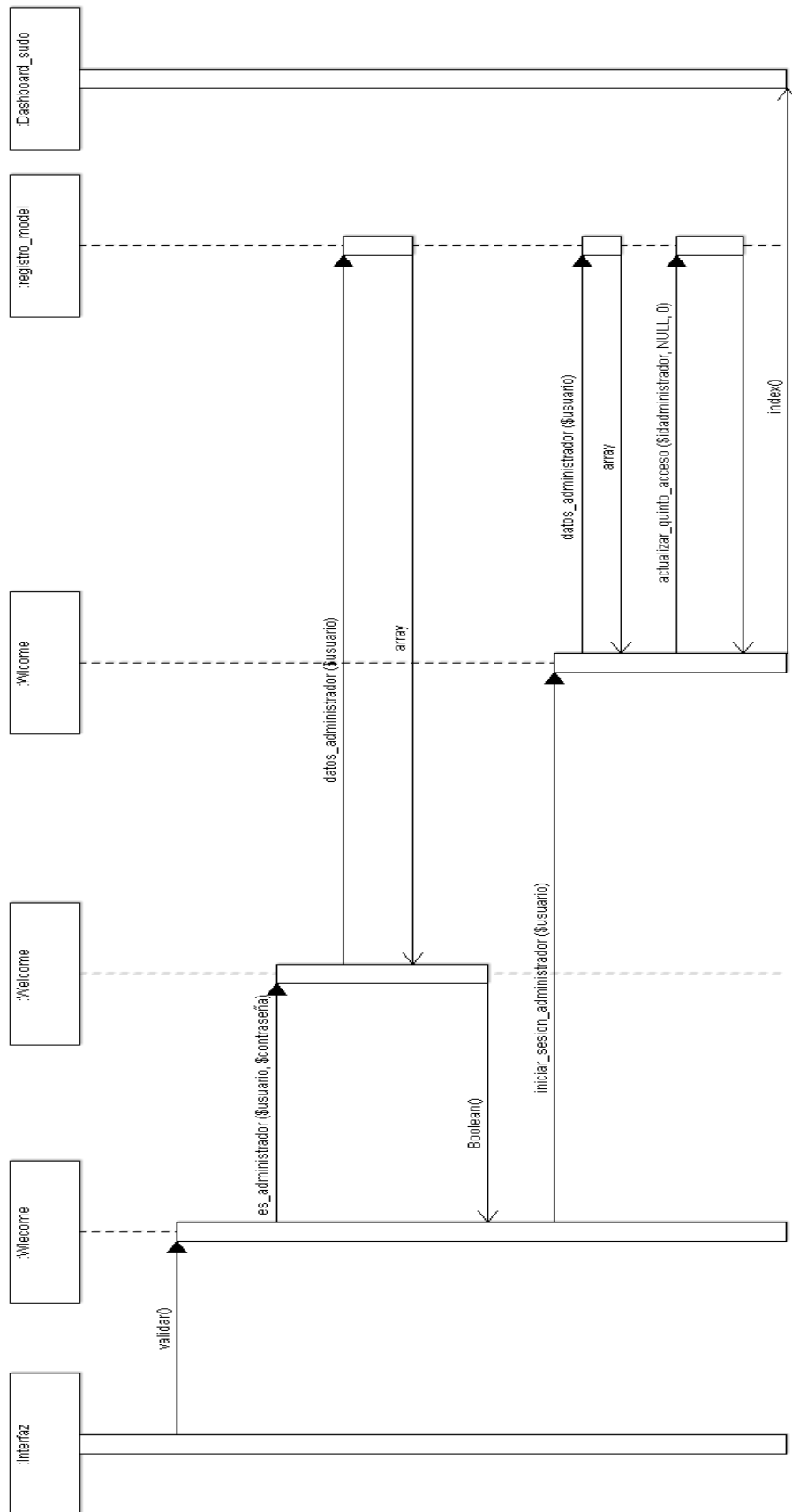


Ilustración 6-15 Iniciar Sesión

Listar

Este diagrama de secuencia corresponde con el subcaso de uso CU04.6: LISTAR ADMINISTRADORES – Podemos observar como el usuario a través de la interfaz selecciona listar administradores, automáticamente se invocará al método “administradores”, este no tendrá parámetros (los recibirá cuando sea invocado desde otros métodos de la misma clase).

El método mencionado anteriormente se encarga de mostrar los administradores de forma paginada, para ello solicitará al modelo el total de filas que tiene la tabla administradores, tras esto invocará a “administradores_paginados” a este se le pasará el número de filas que se desean listar, y la fila en la que debe comenzar (como no se ha pasado ningún parámetro al método “administradores” la fila inicial será la primera).

Tras hacer todo lo anterior se devolverá un vector con todos los datos a la interfaz para que se los muestre al usuario.

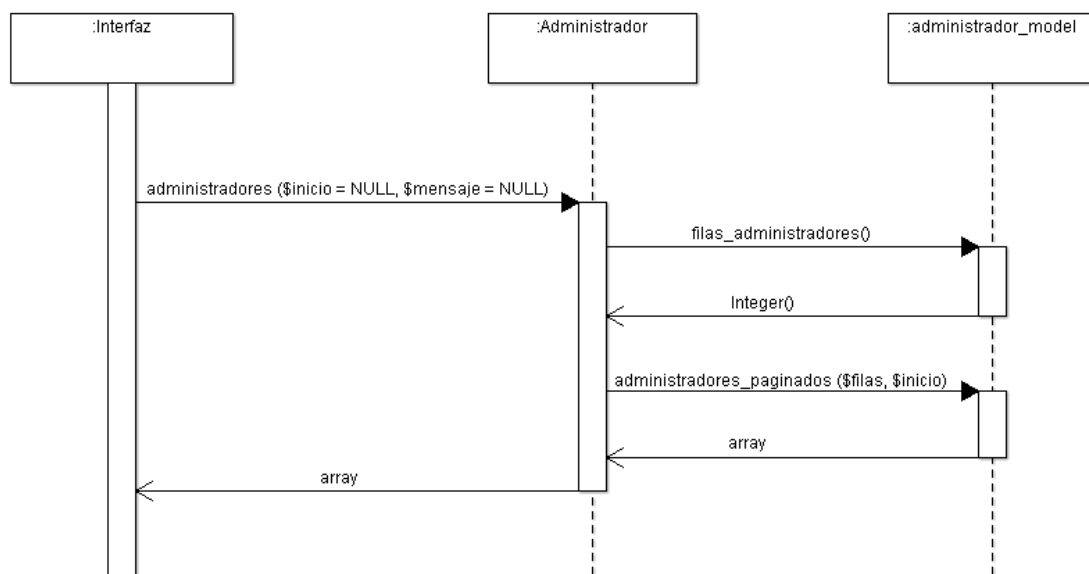


Ilustración 6-16 Listar

Crear

El siguiente diagrama corresponde de con el subcaso de uso CU04.5: CREAR ADMINISTRADOR - Cuando se seleccione crear administrador, se invocará al método formulario_alta de la clase administrador, este recolectará todos los datos que deben ser mostrados en las casillas de selección del formulario.

El administrador podrá rellenar y seleccionar todos los campos, por ultimo deberá pulsar el botón de envío, de esta forma estará invocando el método “crear” que es el encargado de recopilar todos los datos, cuando así lo haya hecho usará dos métodos:

- **crear:** Crea un nuevo usuario en la BD con los datos recibidos del array. Cuando acabe al método de su éxito en la inserción.
- **get_id:** Dado el DNI del administrador creado, nos devolverá su identificador.

Por último se llamará a la clase administradores la cual recibirá un mensaje en forma de vector, el cual deberá ser mostrado al usuario.

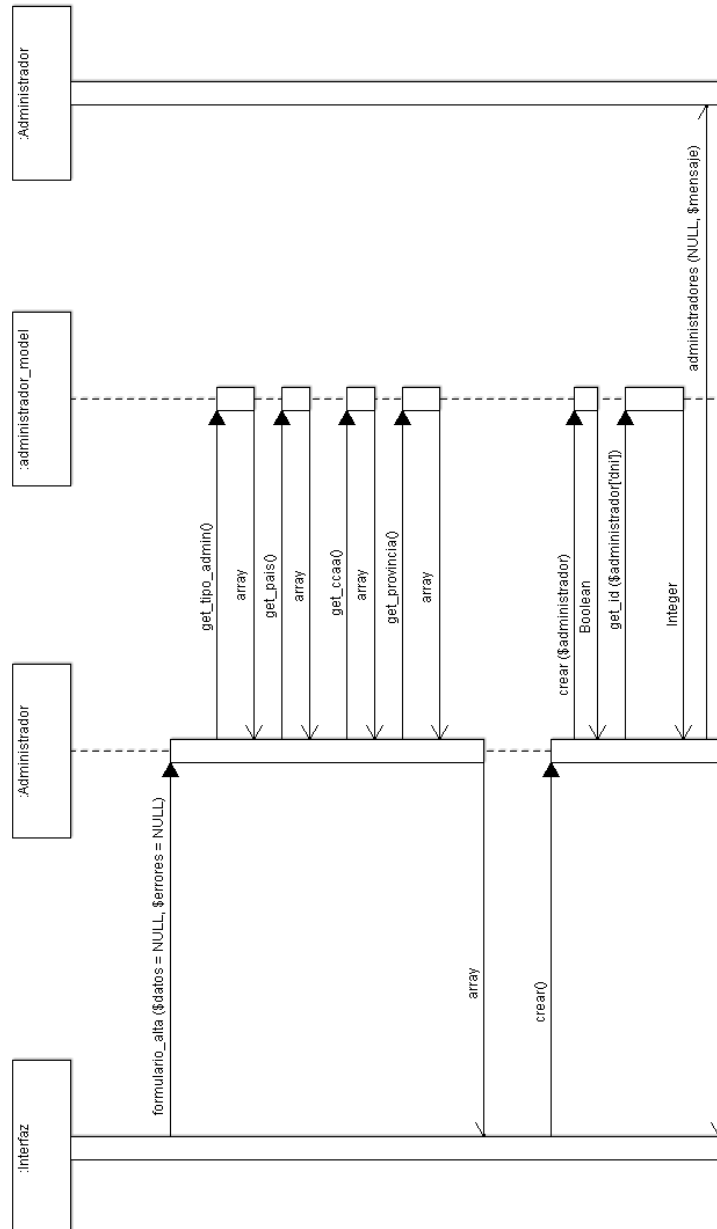


Ilustración 6-17 Crear

Modificar

Este diagrama corresponde con el subcaso de uso CU04.10: EDITAR INFORMACIÓN DE UN ADMINISTRADOR - Cuando sea pulsado el botón encargado de editar un administrador, se invocará método “formulario_editar” de la clase “Administrador”, este recibirá el identificador del administrador a editar, y empezara a llamar a los siguientes métodos del controlador “administrador_model”:

- **get_administradores:** Dado un identificador devuelve todos los datos del administrador al que pertenece.

Los siguientes recogerán información general para mostrarla en el formulario de edición y que puedan ser seleccionados.

- **get_tipo_admin:** Devuelve todos los tipos de administradores.
- **get_pais, get_ccaa y get_provincia:** Retorna todos los países, comunidades y provincias almacenadas en la BD.

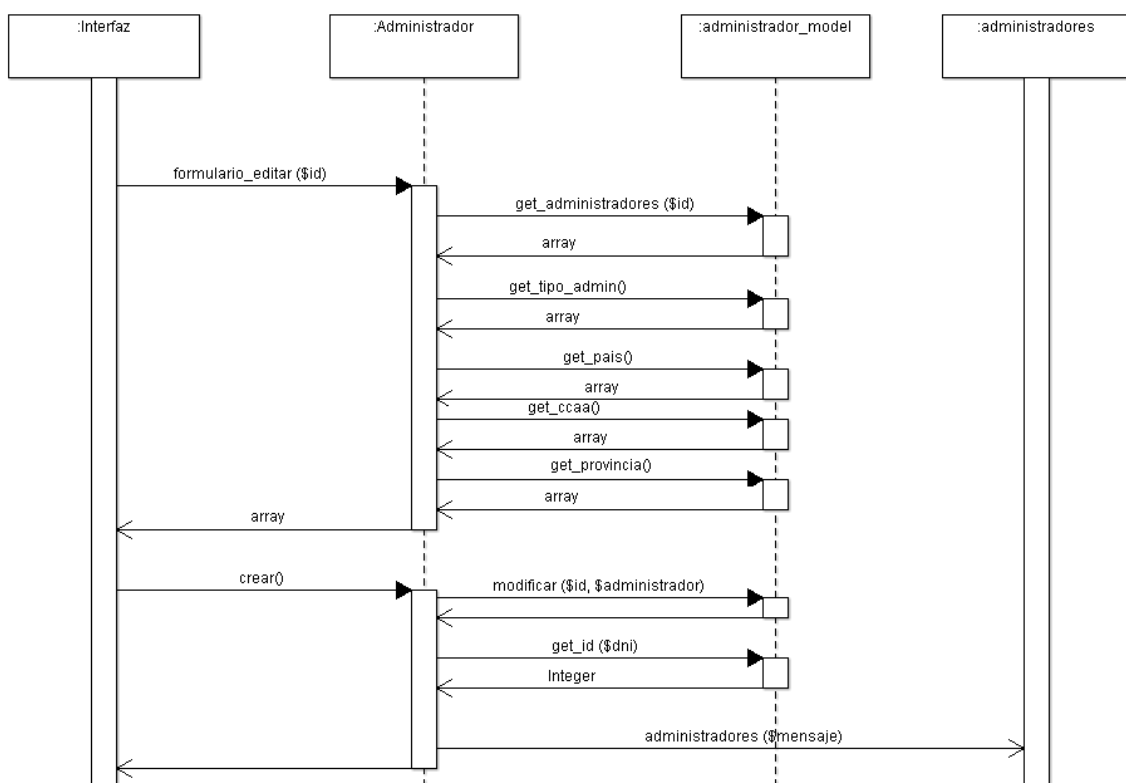


Ilustración 6-18 Modificar

6.1 Diagramas de clases

Para un mayor detalle de los diagramas de clases que se han creado y el completo entendimiento de la aplicación los mostraremos en el Anexo V.

7 Conclusiones

Tras la realización de este proyecto puedo afirmar que los objetivos marcados inicialmente han sido cumplidos en su totalidad, y que la aplicación desarrollada aporta un valor añadido a los gimnasios los cuales tendrán la oportunidad de ofrecer un servicio nuevo a los clientes, así como tener un mayor control sobre las actividades que realizan.

Otra de las características que pueden destacarse de esta aplicación es que el propietario del gimnasio, no solo es capaz de observar la actividad de los clientes, sino que también puede supervisar la correcta labor de los empleados.

Por otro lado el uso de tecnologías como el Framework “CodeIgniter” nos aporta una mayor organización y una mejor estructura dentro los proyectos. Como todas las tecnologías en las que nos introducimos por primera vez debemos superar la curva de aprendizaje tras la cual el desarrollo de los proyectos es extremadamente más ágil permitiéndonos en el futuro un mantenimiento más asequible debido al uso del patrón de diseño MVC.

Esto nos permite que a la hora de usar código desarrollado por otras personas bajo la misma herramienta, su lectura y comprensión es mucho más sencilla permitiéndonos así trabajar varias personas en el mismo proyecto sin invertir un tiempo excesivo en la comprensión del código.

Además del CodeIgniter el uso de otras tecnologías para el desarrollo de este tipo de aplicaciones tales como Bootstrap nos facilita enormemente la creación de diseños atractivos para el usuario ya que es una característica que nuestro ámbito académico no se desarrolla en profundidad.

Para dar por concluidas las tecnologías utilizadas no puedo dejar de nombrar las tecnologías de comunicación asíncrona tales como: JavaScript, AJAX, JQuery, las cuales me han permitido realizar peticiones de datos bajo demanda sin que el usuario fuera consciente de ello.

Finalmente y a título personal considero que embarcarme en un proyecto cuyas tecnologías no han sido expuestas detalladamente durante mis estudios hace que me sienta más preparado para enfrentarme al mundo laboral debido a que he observado que son tecnologías ampliamente usadas en la actualidad.

8 Líneas de trabajo futuro

En este apartado se pretende explicar unas series de mejoras para que este proyecto pueda seguir creciendo y aumente así su funcionalidad y uso.

- **Iteración de los clientes:** Considero que sería adecuado que los clientes pudieran entrar al sistema, además de consultar sus rutinas, sería conveniente que pudieran introducir datos (días que van, peso que levanta actualmente, peso corporal, lesiones producidas recientemente...). De esta manera podrá observar toda su trayectoria así como su evolución.

De la misma manera el monitor también saldrá beneficiado al tener más conocimientos acerca de los clientes que gestiona y podrá generar entrenamientos aún más pendientes.

- **Desarrollo de una aplicación móvil:** Para complementar el punto anterior, estaría bien que el usuario en vez de tener un papel con su tabla pudiera tener también una aplicación que le diga únicamente los ejercicios que tiene que hacer en ese día, además de poder introducir datos en el mismo momento en el que está entrenando.

9 Bibliografía

Consultadas agosto del 2015

- [1] <http://www.xculpture.com/>
- [2] <http://my.fitnessbliss.com/fbo/login.seam?cid=18372092>
- [3] <http://www.codeigniter.com/userguide3/index.html>
- [4] http://escodeigniter.com/guia_usuario/libraries/pagination.html#
- [5] <http://getbootstrap.com/>
- [6] <https://fortawesome.github.io/Font-Awesome/icons/>
- [7] <http://php.net/manual/es/index.php>
- [8] <http://morrisjs.github.io/morris.js/>
- [9] <http://www.vincentbroue.fr/mapael/>
- [10] https://librosweb.es/libro/bootstrap_3/
- [11] <http://www.fpdf.org/>
- [12] <http://www.codedrinks.com/como-integrar-la-libreria-fpdf-con-el-framework-codeigniter/>

Consultadas septiembre del 2015

- [13] <https://almsaeedstudio.com/themes/AdminLTE/pages/calendar.html>
- [14] <http://startbootstrap.com/template-overviews/sb-admin/>
- [15] http://www.w3schools.com/bootstrap/bootstrap_modal.asp
- [16] <http://uno-de-piera.com/consultas-agrupadas-en-codeigniter-3/>

Consultadas octubre del 2015

- [17] <http://www.forosdelweb.com/f18/salto-pagina-con-fpdf-488753/>
- [18] <http://www.lostiemposcambian.com/blog/javascript/pasar-parametros-a-ficheros-externos-javascript/>

[19] <https://blog.udemy.com/jquery-vs-javascript-2-cual-es-la-diferencia-en-definitiva/>

[20] <https://geekytheory.com/ajax-obtener-datos-asincronamente/>

Consultadas octubre del 2015

[21] <http://isbin.com/uzosiq/258/embed?html,js,output>

[22] <https://github.com/acornejo/bootstrap-nav-wizard>

[23] http://www.majstro.com/Web/Bedrijf/js/js_Global_String_split1.html

[24] <https://opendata.socrata.com/dataset/Country-List-ISO-3166-Codes-Latitude-Longitude/mnkm-8ram>

[25] <http://uno-de-piera.com/enviar-emails-en-codeigniter-con-smtp-desde-yahoo-y-gmail/>

Consultadas noviembre del 2015

[26] <http://www.desarrolloweb.com/articulos/codeigniter.html>

[27] http://www.w3schools.com/jsref/tryit.asp?filename=tryjsref_table_insertrow

[28] <https://es.wikipedia.org/wiki/JavaScript>

[29] <https://es.wikipedia.org/wiki/HTML>

[30] <https://es.wikipedia.org/wiki/JQuery>

[31] <https://es.wikipedia.org/wiki/AJAX>

[32] <https://es.wikipedia.org/wiki/MySQL>

[33] <https://es.wikipedia.org/wiki/PHP>

Anexo I

A continuación se describen los casos de uso comunes para los tres tipos de usuarios en el sistema.

- **CU01: INICIAR SESION**

El usuario deberá introducir su nombre de usuario y contraseña. Si la consulta tiene éxito, se permitirá al usuario acceder a su cuenta. En caso contrario, se mostrará un mensaje de error.

Si se supera un número determinado de intentos de accesos fallidos se bloqueará la cuenta de usuario de manera temporal.

- **CU02: RECUPERAR CONTRASEÑA**

En caso de olvido, el usuario podrá recuperar su contraseña, para ello deberá introducir su correo electrónico en la pantalla de recuperación de contraseña.

Tras introducir el correo el sistema comprobará que ese correo existe en la base de datos. En caso afirmativo se enviará un código aleatorio al correo para que pueda recuperar su cuenta. De no ser así se le mostrara un mensaje de error y no se enviará correo alguno.

Cuando el usuario disponga del código, éste deberá ser introducido en el apartado disponible para insertarlo. Además del código se requerirá ingresar el correo del usuario.

En caso de que el sistema compruebe que el código coincide con el correo electrónico asociado al administrador, se comprobará que no se haya sobrepasado un periodo de tiempo determinado. En caso de éxito será reconducido a una pantalla donde podrá restablecer la contraseña. De no ser así, mostrará un mensaje de error.

En los puntos siguientes se van a detallar los casos de uso para un usuario tipo administrador. Estos son los siguientes:

- **CU03: VER INFORMACIÓN GENERAL**

El administrador podrá ver un mapa con el número de gimnasios que hay en cada país.

También podrá visualizar la información mostrada de manera gráfica.

Algunas de las gráficas deberán estar dotadas de un botón, donde se podrá ampliar la información, visualizándola en una tabla debidamente estructurada.

- **CU04: GESTIONAR ADMINISTRADORES**

El administrador principal podrá gestionar los administradores creados en la base de datos, para ello se van a definir estas acciones en varios subcasos, los cuales se detallan a continuación:

- **CU04.1: CREAR TIPO DE ADMINISTRADOR**

Inicialmente en el sistema solo existirá un tipo de administrador con todos los privilegios de gestión, no obstante, deberá estar preparado para albergar en un futuro más tipos de administradores con limitaciones respecto al administrador principal. Así pues podremos crear distintos tipos de administradores mediante un formulario.

- **CU04.2: LISTAR TIPOS DE AMINISTRADOR**

El administrador principal podrá ver en una tabla y de forma estructurada todos los tipos de administradores existentes en la base de datos. Dicha tabla, además de la información sobre los tipos de administradores dispondrá de una serie de iconos que nos permitirán gestionar individualmente cada tipo de administrador.

- **CU04.3: EDITAR TIPO DE ADMINISTRADOR**

El administrador principal podrá editar un tipo de administrador. Para ello, tal y como ha sido mencionado en el punto anterior (CUXX.2), este usuario una vez lista los tipos de administradores dispondrán de un botón junto al nombre del tipo de administrador, el cual permita editarlo. Cuando dicho botón sea pulsado el usuario será reconducido a un formulario en el cual podrá modificar los datos y guardar los cambios realizados.

- **CU04.4: ELIMINAR TIPO DE ADMINISTRADOR**

El administrador principal podrá eliminar un tipo de administrador. Al igual que en puntos anteriores (CUXX.2), el usuario una vez haya listado los distintos tipos de administradores dispondrá de un botón junto al nombre del tipo de administrador, el cual le permitirá eliminarlo. Cuando dicho botón sea pulsado se le deberá mostrar un mensaje indicándole que se eliminarán los datos de BD. Este mensaje dispondrá de dos botones, uno de cancelación y otro de aceptación.

En caso de que el usuario pulse sobre el botón aceptar se eliminarán los datos de la BD y se listarán de nuevo los tipos de administradores existentes en la BD. Si por el contrario pulsa sobre el botón cancelar desaparecerá el mensaje sin producirse ningún cambio.

- **CU04.5: CREAR ADMINISTRADOR**

El administrador principal podrá crear nuevos administradores. Para ello deberá rellenar y seleccionar una serie de datos a través de un formulario.

En caso de que todos los datos introducidos o seleccionados sean correctos, se almacenarán en la base de datos. Una vez que el administrador haya sido creado, se mostrara un mensaje indicando el éxito de la operación, pudiendo ver la contraseña que el sistema ha generado, así como imprimir en un fichero PDF todos los datos junto con su password.

En el supuesto caso de que exista algún error en los datos insertados, se detallarán para que el administrador principal pueda subsanarlos. Además el formulario contendrá los datos insertados previos al envío con el fin de poder ser revisados.

- **CU04.6: LISTAR ADMINISTRADORES**

El administrador principal podrá ver en una tabla, de forma estructurada y paginada todos los administradores existentes en la base de datos. Dicha tabla, además de la información sobre los administradores dispondrá de una serie de iconos que nos permitan gestionar individualmente cada uno de ellos.

Los subcasos expuestos a continuación, hacen referencia al subcaso de uso descrito anteriormente (CUXX.X), ya que todos ellos están en la pantalla listar administradores.

- **CU04.7: BUSCAR ADMINISTRADOR**

El administrador principal podrá buscar un administrador en la base de datos. Para ello existirá un buscador en el que se podrá introducir una serie de caracteres a buscar en las tablas asociadas a administradores, se comprobará los caracteres y se devolverá las coincidencias encontradas.

Los elementos que se buscarán mediante los caracteres insertados serán los nombres y apellidos de los administradores.

Finalmente se mostrarán los administradores que coincidan con los caracteres introducidos previamente.

- **CU04.8: IMPRIMIR ADMINISTRADORES**

El administrador principal podrá imprimir un administrador. Para ello dispondrá de un botón que le permita hacerlo. Cuando el botón se pulse se generará un documento “PDF” con la información de los administradores existentes en el sistema. Este documento generado podrá ser visualizado, descargado o impreso.

- **CU04.9: VER INFORMACIÓN DE UN ADMINISTRADOR**

El administrador principal podrá ver la información total de un administrador. Para este fin dispondrán de un botón, de manera que pueda ampliar la información del administrador seleccionado.

- **CU04.10: EDITAR INFORMACIÓN DE UN ADMINISTRADOR**

El administrador principal podrá editar un administrador. Por lo cual tiene que existir un botón que le permita hacerlo. Cuando el botón sea pulsado será reconducido a un formulario en el cual podrá modificar los datos y guardar los cambios realizados.

- **CU04.11: GENERAR PDF DE UN ADMINISTRADOR**

El administrador principal podrá generar un fichero “PDF” con la información total de un administrador. Con este fin deberá tener un botón que le permita generar el fichero “PDF” con información del administrador solicitado.

- **CU04.12: DAR DE BAJA ADMINISTRADOR**

El administrador principal podrá dar de baja en el sistema a un administrador. Esta acción la realizará mediante un botón, el cual le permita dar de baja al administrador solicitado. Previamente se le mostrará un mensaje indicándole los cambios que se efectuarán en el sistema. El usuario tendrá la posibilidad de aceptarlos y dar de baja el administrador o cancelar sin efectuar ningún cambio.

- **CU04.13: DAR DE ALTA ADMINISTRADOR**

El administrador principal podrá dar de alta en el sistema a un administrador, con este fin dispondrá de un botón que le permita hacerlo.

Una vez sea pulsado, se le mostrará un mensaje indicándole los cambios que se efectuaran en el sistema. El administrador podrá aceptar los cambios y dar alta al administrador o cancelar la operación sin realizar modificaciones.

- **CU05: GESTIONAR GIMNASIOS**

El administrador principal puede gestionar los gimnasios existentes. Esto será realizado de manera muy similar a la gestión de administradores, **siendo iguales los subcasos de uso**. Las únicas diferencias a destacar son las siguientes.

- Un gimnasio pertenece a un tipo de gimnasio, de esta manera podremos conocer con más detalle aspectos que puedan distinguir a cada uno de ellos, por ejemplo podríamos distinguir entre:
 - Gimnasio.
 - Gimnasio femenino.
 - Macrogimnasio.
 - Low Cost.
- Por otro lado se puede destacar dos cambios significativos a la hora de trabajar un gimnasio, dichos cambios se detallan a continuación:
 - En el formulario de creación se sustituirá el campo DNI por el campo NIF.
 - Un gimnasio puede tener un logotipo que le represente.

- **CU06: GESTION DE DEPORTES**

En este apartado se podrá gestionar todos los deportes existentes en la actualidad.

- **CU06.1: LISTAR**

Se listará en una tabla todos los deportes existentes en la base de datos. Junto al nombre de cada uno de los deportes, ha de aparecer una serie de botones que permita la gestión de cada uno de los deportes en particular.

Por otro lado, debe haber un botón que nos permita la creación de un nuevo deporte.

Los siguientes subcasos de usos están relacionados con el subcaso de uso listar (CUXX.X). Esto es debido a que los siguientes subcasos que se detallarán a continuación deben efectuarse desde esta pantalla.

- **CU06.2: CREAR**

Se deberá poder crear un nuevo deporte, dado que la información sobre este, se limita a un nombre, no se debe redireccionar a un formulario para la creación. Para ello una vez sea pulsado el botón para la creación de un nuevo deporte, se debe abrir una ventana “modal” y con esto evitaremos que el administrador principal se distraiga con otras cosas que no sean relevantes para este fin.

La ventana contendrá un pequeño formulario que permitirá introducir el nombre de un nuevo deporte. En esta venta el administrador podrá decidir entre, cerrar la ventana sin efectuar ningún cambio o rellenar el campo nombre para su inserción en la base de datos.

- **CU06.3: EDITAR**

Si se desea editar un deporte, se debe redirigir a un formulario tras pulsar el botón oportuno. Dicho formulario tendrá un campo de texto con el nombre del deporte a modificar en su interior. Una vez realizados los cambios oportunos, se podrán guardar.

- **CU06.4: ELIMINAR**

Si por algún motivo el administrador principal opina que un deporte ya no es relevante en la aplicación, podrá eliminarlo, para ello dispondrá de un botón, que una vez sea pulsado se desplegará una ventana indicándole que los datos serán eliminados. El administrador podrá cancelar la operación o eliminar los datos.

- **CU07: GESTIÓN A NIVEL MUNDIAL**

Debido a que esta aplicación está pensada para poder usarse a nivel mundial se deberá poder administrar una serie de datos que nos permita hacerlo.

Estos se definen a continuación:

- **CU07.1: LISTAR PAISES**

Deberán listarse los datos referentes a un país, siendo estos el nombre del país, su latitud y longitud. Además junto a cada uno de ellos aparecerá una serie de botones que nos permita su edición o eliminación.

En la parte inferior debemos tener un botón que nos permita la creación de nuevos países.

- **CU07.2: CREAR PAÍS**

Una vez el administrador pulse sobre botón para añadir un nuevo país, la pantalla actual se sustituirá por un formulario que nos permita hacerlo. Este formulario será diferente al resto, ya que ha de tener un mapa de google y una casilla donde introducir el país a añadir.

Una vez introducido el nombre de un país, deberá ser buscado en el mapa de google, esté nos mostrará su situación en el mapa, además en el formulario se escribirá automáticamente su latitud y longitud. De esta manera se elimina una faena engorrosa al administrador, como sería la de buscar en internet las coordenadas de un país.

Si por algún motivo el administrador no deseara hacer ningún cambio debe disponer de un botón para retroceder a la lista de países.

- **CU07.3: EDITAR PAÍS**

A pesar de que el usuario tiene difícil introducir mal los datos cuando un país es creado, debido a las características excepcionales de este formulario, daremos la posibilidad de editar el nombre del país una vez listados. Junto a los datos de cada país, el administrador dispondrá de un botón que le permita hacerlo. Una vez pulsado, se redireccionará a un formulario donde estará escrito el nombre del país seleccionado. Cuando realice algún tipo de cambio, este podrá ser guardado.

- **CU07.4: ELIMINAR PAÍS**

Un país almacenado en la base de datos, podrá ser eliminado en cualquier momento. Para ello dispondrá de un botón en la lista de países junto a los datos de un país. Una vez sea pulsado este botón, se abrirá una ventana para informar que los datos se eliminarán de la BD, el usuario podrá eliminar el país, o cancelar la operación sin realizar cambio alguno.

Los siguientes casos de uso hacen referencia a la creación de comunidades autónomas para un país. A continuación serán expuestos:

- **CU07.5: LISTAR COMUNIDADES AUTONOMAS**

El administrador podrá ver todas las comunidades autónomas que contiene la base de datos, para ello se mostrará el nombre de la CCAA junto al país al que pertenece. Además tendrá dos botones junto a los datos de cada una de ellas, estos botones corresponden a edición y eliminación.

Por otro lado debe haber un botón que permita la creación de una nueva CCAA.

- **CU07.6: CREAR COMUNIDAD AUTONOMA**

Una vez pulsado el botón correspondiente para crea una comunidad autónoma desaparecerá la lista de las CCAA y en su lugar aparecerá un formulario. Este formulario tendrá una lista desplegable con los nombres de los países y una casilla de texto para introducir el nombre de la CCAA.

El administrador podrá optar entre dos opciones:

La primera será volver a la lista de CCAA, pulsando sobre el botón volver.

La segunda será crear una nueva comunidad. Para ello el administrador debe rellenar los campos y pulsar sobre el botón guardar.

- **CU07.7: EDITAR COMUNIDAD AUTONOMA**

Junto a la descripción de una CCAA dispondremos de un botón que nos permitirá editar la comunidad. Una vez pulsado se debe redirigir a un formulario donde se encontrarán los datos actuales de la comunidad a editar. En él se podrá seleccionar otro país y cambiar el nombre, una vez modificados los cambios se guardarán.

Los siguientes casos de uso hacen referencia a la gestión de provincias que puede tener y una comunidad. Estos se especifican a continuación:

- **CU07.9: ELIMINAR COMUNIDAD AUTONOMA**

Al igual que en los casos anteriores debe haber un botón que nos permita eliminar la comunidad autónoma. Una vez pulsado el botón se nos abrirá una ventana, esta informará al administrador que los datos serán eliminados de la BD. Si realmente desea eliminarla, solo tendrá que

pulsar sobre el botón aceptar, si por el contrario no desea realizar ningún cambio, dispondrá de un botón para cancelar.

- **CU07.10: LISTAR PROVINCIAS**

El administrador principal podrá listar todas las provincias existentes en la base de datos, estas serán mostradas en una tabla junto con la comunidad y país a la que pertenece. Junto a estos datos tendremos dos botones correspondientes a la modificación y eliminación de provincias.

En la parte inferior habrá un botón que nos permitirá la creación de nuevas provincias.

- **CU07.11: CREAR PROVINCIA**

Una vez pulsado el botón correspondiente a crear provincia, aparecerá un formulario en el lugar que ocupa la lista de países, este estará dotado de un campo de selección. Este campo contendrá todas las comunidades autónomas junto al país a la que pertenece, también poseerá una casilla de tipo texto para introducir el nombre de la provincia. Una vez seleccionada la comunidad y rellenado el nombre se podrá guardar los datos.

- **CU07.12: MODIFICAR PROVINCIA**

Cuando el botón para modificar la provincia sea pulsado por el administrador se le llevará a un formulario, este contendrá los datos de la provincia a modificar. El administrador podrá hacer cambios y guardarlos.

- **CU07.13: ELIMINAR PROVINCIA**

El administrador tendrá la posibilidad de eliminar una provincia. Cuando pulse el botón de eliminación, se le desplegará una ventana indicando que los datos serán eliminados, si desea seguir adelante, podrá hacerlo mediante el botón aceptar, si por el contrario no desea eliminarla, dispondrá de un botón que le permita cancelar y volver a la lista de provincias.

- **CU08: GESTIONAR GRUPOS MUSCULARES**

En este apartado un administrador gestionará los grupos musculares. Estos serán usados para la creación de ejercicios (como se verá más adelante). A

continuación se exponen los subcasos de uso referentes a la gestión de un grupo muscular:

- **CU08.1: LISTAR**

Se expondrán los grupos musculares existentes en la base de datos mediante una tabla, en ella se verán los nombres, así como dos botones correspondientes a cada uno. Estos botones corresponden a la edición y eliminación de un grupo muscular en concreto.

La parte inferior alojará un botón, éste nos permitirá crear un nuevo grupo muscular.

- **CU08.2: CREAR**

Cuando el administrador quiera crear un nuevo grupo muscular, únicamente deberá pulsar sobre el botón oportuno (ya definido en el punto anterior), una vez lo pulse, se abrirá una ventana “modal” con un campo para introducir el nombre. Una vez rellenado el campo podrá guardar los cambios. Si no desea insertar un nuevo grupo muscular, cerrará la ventana pulsando sobre el botón cancelar.

- **CU08.3: EDITAR**

Cuando el botón de edición sea pulsado se debe redirigir a un formulario. Este formulario contendrá un campo, el cual estará rellenado con el nombre del grupo muscular a editar. Una vez realizadas modificaciones se podrá guardar los cambios.

- **CU08.4: ELIMINAR**

Una vez el administrador pulse sobre el botón eliminar, se desplegará una ventana, la cual le explicará que los datos serán eliminados. Si desea seguir adelante, podrá aceptar pulsando sobre el botón aceptar, si por el contrario desea volver al listado, dispondrá de un botón de cancelación.

- **CU09: GESTIONAR EJERCICIOS**

El administrador creará ejercicios que serán utilizados por el monitor a la hora de crear sus tablas. La gestión de ejercicios se divide en cinco subcasos de uso, estos se detallan a continuación:

- **CU09.1: CREAR**

El administrador estará en disposición de crear nuevos ejercicios. Esto lo realizará mediante un formulario, en el habrá un campo para el nombre del ejercicio, una casilla para seleccionar a que grupo muscular pertenece, dos “interruptores” para especificar si es de tipo isométrico y si estará habilitado (para su posterior uso), y por último podrá subir una imagen que represente al ejercicio. Este último apartado no será obligatorio ya que puede haber ejercicios sin imágenes.

- **CU09.2: LISTAR**

Los datos podrán ser visualizados en una tabla de forma paginada. Esta contendrá varios campos que son: el nombre del ejercicio, grupo muscular al que pertenece, si es o no isométrico, si está habilitado y si dispone de una imagen. Seguidamente se dispondrá de tres botones que permitirá la visualización total de los datos, la edición y la eliminación.

- **CU09.3: VER**

Una vez pulsado sobre el botón correspondiente para ver el ejercicio se mostrará los datos del mismo, de la misma manera que se mostraba al ser listado, con la diferencia de que ahora aparecerá la imagen del ejercicio (en caso de que la tenga). Si el ejercicio no tiene imagen se especificará con un texto especificando que no hay imagen, en el mismo lugar donde debería ir la imagen.

- **CU09.4: EDITAR**

Una vez se pulse sobre el botón editar, se llevará al administrador a un formulario, este tendrá los datos del ejercicio a modificar, así como su imagen si la tiene. Una vez realizadas las modificaciones en los datos o la imagen, se podrá guardar los cambios.

- **CU09.5: DAR DE BAJA**

Un ejercicio que no sea usado se podrá dar de baja. Una vez se pulse sobre el botón dar de baja, aparecerá una ventana indicando que el ejercicio se deshabilitará. En caso de que el administrador desee hacerlo podrá aceptar el mensaje, si por el contrario el administrador no quiere hacer cambios, tras haber pulsado sobre el botón, este deberá poder cancelar la operación.

○ CU09.6: DAR DE ALTA

Un ejercicio que esta deshabilitado podrá darse de alta en el sistema. Para tal fin habrá un botón en el lugar donde estaba el botón de deshabilitar. Una vez pulsado saldrá una ventana indicándole que se procederá a dar de alta el ejercicio. En caso de que el administrador acepte se dará de alta, si por el contrario no desea darlo de alta, podrá cancelar la operación.

En la siguiente imagen podremos observar de forma gráfica todos los casos de uso que tiene el administrador.

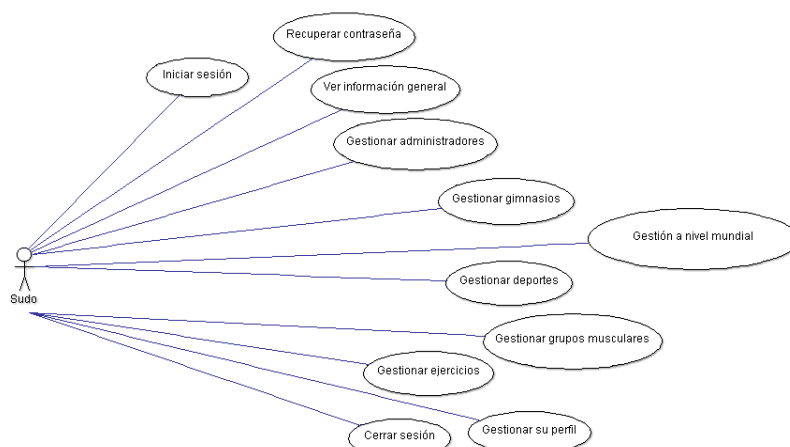


Ilustración 0-1 Sudo

A continuación se exponen los casos de uso para un usuario de tipo gimnasio. Los estos pueden: crear, listar, ver, editar, generar documentos PDF y dar de alta o de baja los clientes y monitores. Ya que los subcasos de uso son idénticos a los expuestos anteriormente en GESTIONAR ADMINISTRADORES (CUXX), no se va a detallar nuevamente cada uno de ellos.

● CU10: GESTIONAR MONITORES

Estos son los encargados de generar las rutinas (tablas de ejercicios), a los clientes de un gimnasio. Además de todos los casos de uso comunes con gestión de administradores también tendrá el siguiente subcaso:

○ CU10.1: VISUALIZAR TABLAS

Junto con los datos del monitor se deberá visualizar todas las tablas que ha realizado a sus clientes.

● CU11: GESTIONAR CLIENTES

Un gimnasio podrá gestionar los diferentes clientes de los cuales dispone, dando nuevas altas en el si así lo desea. Por otro lado tendrá dos subcasos de uso diferente al resto de usuarios ya definidos:

- **CU11.1: GESTIONAR DEPORTES**

El gimnasio podrá gestionar los deportes que cada cliente individualmente práctica con el fin de saber más respecto a ellos. Así pues en el campo acción de la tabla de clientes, deberá haber un botón que le permita hacerlo. Una vez pulsado dicho botón, deberá aparecer en el lado izquierdo de la pantalla todos los deportes que el cliente no tiene asignados y en lado derecho los que sí. Pudiendo cambiarlos de un lado a otro.

- **CU11.2: VER TABLAS**

Además de ver de forma detallada todos sus datos, se podrán visualizar las distintas tablas creadas por los monitores.

La siguiente ilustración detalla todos los casos de uso para el usuario gimnasio.

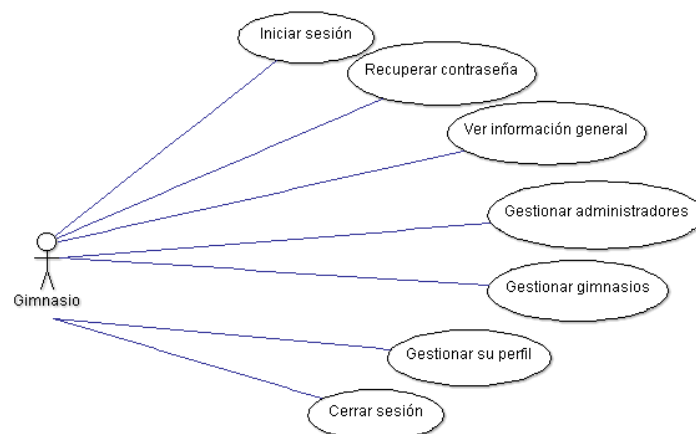


Ilustración 0-2 Gimnasio

Los últimos casos de uso a especificar son los referentes al usuario monitor. Este es el encargado de realizar tablas a los clientes y hacer un seguimiento personalizado sobre cada uno de ellos.

- **CU12: GESTIONAR CLIENTES**

Este caso de uso a su vez lo vamos a detallar en varios subcasos, con el fin de ser más específicos, estos se definen a continuación:

- **CU12.1: LISTAR**

El monitor podrá visualizar en una tabla los datos de los clientes que le han sido establecidos por el gimnasio o que no estén fijados todavía a ningún monitor. Esta tabla además de contener información sobre el

cliente tendrá un campo llamado acción el cual contendrá uno o dos botones por cada usuario, estos dos casos se explican a continuación:

- Si el cliente está no tiene sujeto a ningún monitor, se mostrará un botón para que se lo pueda auto asignar.
 - En caso de que el monitor ya lo tenga fijado, podrá desasignárselo y visualizar sus datos.
- **CU12.2: ASIGNAR CLIENTE**
 Cuando el monitor pulse el botón de asignar cliente, le aparecerá una ventana modal para preguntarle si realmente desea hacerlo, en caso de que así sea se le asignará el cliente y el botón se sustituirá por otro que le permitirá permita hacer los pasos contrarios.
 - **CU12.3: VER**

El monitor podrá visualizar los datos de un cliente junto con los deportes que práctica, además dispondrá de un botón que le permitirá visualizar sus tablas (en caso de que las tenga). Justo debajo de sus rutinas, habrá dos botones para crearle nuevas tablas. Ambas opciones tienen están divididas en tres pequeños pasos, siendo estos dos primeros iguales:

Paso 1: Introducir datos de la tabla, estos son: nombre, fecha de inicio, fecha de finalización y activación.

Paso 2: El monitor seleccionará los ejercicios que un cliente debe hacer, estando todos ellos agrupados por grupos musculares.

Paso 3: Estos son definidos en dos subcasos de uso.

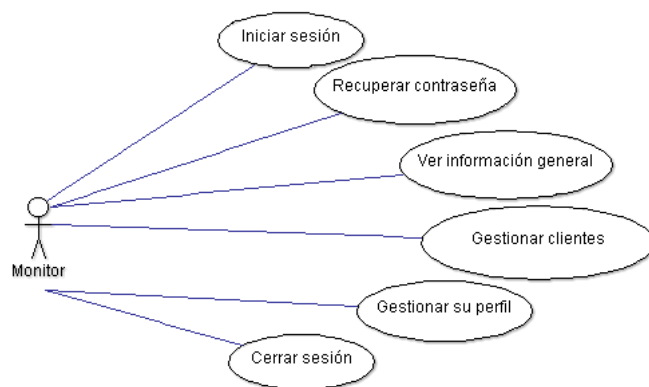
- **CREAR TABLA GENÉRICA**

Se especificarán que día de la semana debe realizarse los ejercicios de un grupo muscular, pudiendo haber varios por día.

- **CREAR TABLA DIÁRIA**

Si se desea hacer una tabla más personalizada acerca de los días exactos en que tiene que trabajar cada grupo muscular, se usará esta opción.

A continuación podremos observar el último dibujo, este hace referencia a los casos de uso del monitor.

*Ilustración 0-3 Monitor*

Anexo II – Manual de Usuario

Autenticación de usuario

El acceso a la aplicación lo podremos realizar a través de un navegador web (Se aconseja usar Chrome para una mejor experiencia). Una vez abierto el navegador, podemos acceder usando el siguiente enlace “<http://www.starglob.com/GymGest>” en el encontraremos un formulario de registro como el mostrado a continuación.

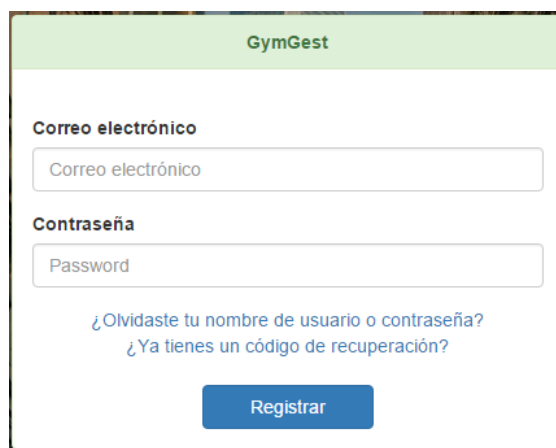


Ilustración 0-1 Login

Como se puede observar este formulario tiene tres opciones

1. Iniciaremos sesión introduciendo el correo electrónico y contraseña, en caso de que ambos datos sean correctos se nos iniciará la sesión, si por el contrario los datos introducidos no son correctos se mostrará el siguiente error.

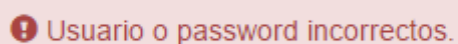


Ilustración 0-2 Error

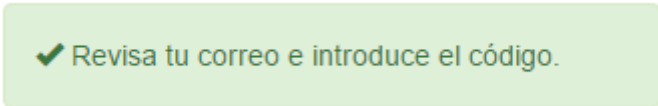
2. En caso de olvido podremos recuperar nuestra contraseña, pulsado sobre el enlace “¿Olvidaste tu nombre de usuario o contraseña?” este nos llevará a la siguiente pantalla.

The screenshot shows a web form titled 'GymGest' with the heading 'Recuperar contraseña'. Below the heading, it says 'Te enviaremos un correo electrónico con código para restablecer tu contraseña.' There is a section labeled 'Correo electrónico' with a text input field containing the placeholder 'Correo electronico'. At the bottom, there are two buttons: 'Enviar' and '← Volver'.*Ilustración 0-3 Recuperar Password*

En ella podremos introducir el correo electrónico que usamos para iniciar sesión, tras ello se nos enviará al correo con un código de recuperación similar al siguiente:

*Ilustración 0-4 Correo*

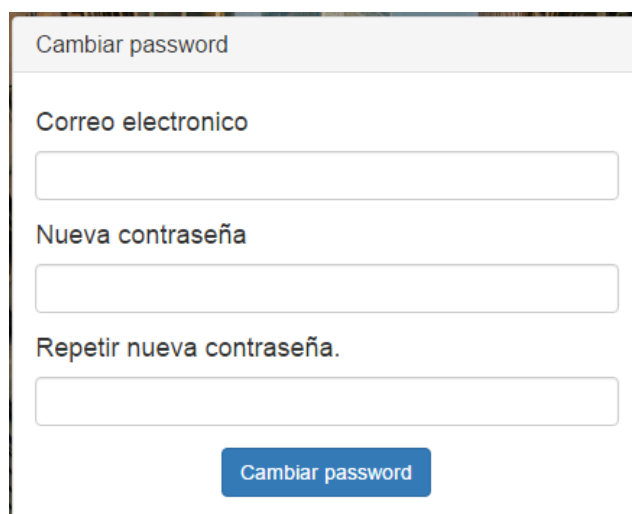
En el correo podemos observar que nos indica claramente lo que debemos de hacer para restablecer la contraseña, además seremos notificados en la interfáz.

A green rectangular box with rounded corners containing a green checkmark icon followed by the text 'Revisa tu correo e introduce el código.'*Ilustración 0-5 Éxito*

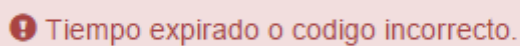
3. Por último si ya tenemos el código de recuperación podemos ir al enlace definido en el correo, y así veremos el formulario mostrado a continuación:

The screenshot shows a web form titled "Recuperar contraseña" (Recover password) under the "GymGest" header. Below the title is the instruction "Introducir el código enviado a tu correo." (Enter the code sent to your email). The form contains two input fields: "Correo electronico" (Email) and "Código" (Code), with placeholder text "Correo electronico" and "Codigo enviado" respectively. At the bottom are two buttons: "Enviar" (Send) and "← Volver" (Return).*Ilustración 0-6 Recuperar Password*

Este formulario nos pide que introduzcamos nuestro correo electrónico junto con el código recibido. Si ambos datos son correctos y el tiempo no ha superado el máximo establecido seremos redirigidos a una pantalla donde podremos restablecer la contraseña y si se hace de forma correcta se nos enviará a nuestra pantalla principal. La pantalla para restablecer la contraseña es la siguiente.

The screenshot shows a web form titled "Cambiar password" (Change password). It contains three input fields: "Correo electronico" (Email), "Nueva contraseña" (New password), and "Repetir nueva contraseña." (Repeat new password). At the bottom is a button labeled "Cambiar password" (Change password).*Ilustración 0-7 Restablecer Password*

Si hay algún error durante el cambio de contraseña seremos notificados con el siguiente mensaje.

The error message is displayed in a red box with a red exclamation mark icon. The text reads: "Tiempo expirado o código incorrecto." (Time expired or code incorrect).*Ilustración 0-8 Error*

Página principal

La página principal y las funciones en el sistema variarán en función del tipo de usuario que inicie sesión. En este punto vamos a detallar el uso de la aplicación para los tres tipos de usuario: administrador, gimnasio y monitor.

Administrador

Es la persona con más privilegios y se encarga de dar acceso al sistema a nuevos administradores y gimnasios así como gestionar los datos este último usará.

Su página principal es la siguiente:



Ilustración 0-9 Dashboard Sudo

En ella visualiza de forma rápida y clara todo lo que ocurre en nuestro sistema. Aparte de esto podrá hacer todas las acciones que tiene en el menú izquierdo, las cuales procedemos a detallar:

Administradores: Podrá crear, listar y gestionar todo lo referente a ellos.

Crear: Aquí puede dar de alta nuevos administradores en el sistema, para ellos solo debemos de rellenar el siguiente formulario.

GymGest - Sudo ricardo

Administradores Crear.

Administradore / Crear

Nombre

Apellidos

DNI

Tipo de administrador

Fecha de nacimiento

País

Comunidad

Provincia

localhost:8080/GymGest/index.php/Administrador/formulario_alta

Ilustración 0-10 Crear Administrador

Si se produce algún tipo de error durante la creación del nuevo administrador, se le notificará para que pueda subsanarlo.

Administradore / Crear

Se ha producido una serie de errores:
Algun campo contiene numeros, asegurese de que esten bien rellenos.
DNI demasiado corto.

Nombre

Apellidos

Ilustración 0-11 Error

En caso de que no se produzca ningún error será reconducido al listado de usuarios, donde se mostrará un mensaje con una serie de opciones, además podrá visualizar el nuevo administrador creado.

GymGest - Sudo ricardo

Administradores Consulta, edita e imprime.

Administradores / Listar

✓ El administrador Alberto Gutierrez Blasco se ha creado con éxito

Imprimir administrador

Administradores

Pdf

Buscar gimnasio

Nombre	Pais	Comunidad	Telefono	Correo	Activo	Accion
ricardo Ibáñez Fuentes	España	Aragon	978000000	richieteruel@gmail.com	SI	
Alberto Gutierrez Blasco	España	Aragon	978000000	yo@gmail.com	SI	

Ilustración 0-12 Éxito al crear

Como se puede ver el mensaje, contiene dos opciones, estas son:

1. **Ver clave generada para el administrador:** Se podrá visualizar el código dejando el puntero del ratón sobre el botón que contiene un ojo.

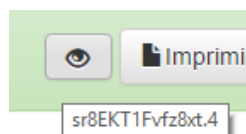


Ilustración 0-13 Ver Password

2. **Imprimir documento PDF junto con su password:** Al pulsar sobre el botón "imprimir" se abrirá una nueva ventana con el siguiente documento:

GymGest

Fecha: 02/12/2015

Nombre: Alberto	Apellidos: Gutierrez Blasco	Fecha de nacimiento: 2015-12-02
País: España	Comunidad: Aragon	Provincia: Teruel
Población: Teruel	Dirección: avd sagunto	Código postal: 44002
Telefono: 978000000	E-Mail: yo@gmail.com	Password: sr8EKT1Fvfz8xt.4

Ilustración 0-14 Imprimir PDF

Listar: Esta pantalla es igual que la descrita anteriormente cuando se ha creado un administrador, con la diferencia de que el mensaje no aparecerá.

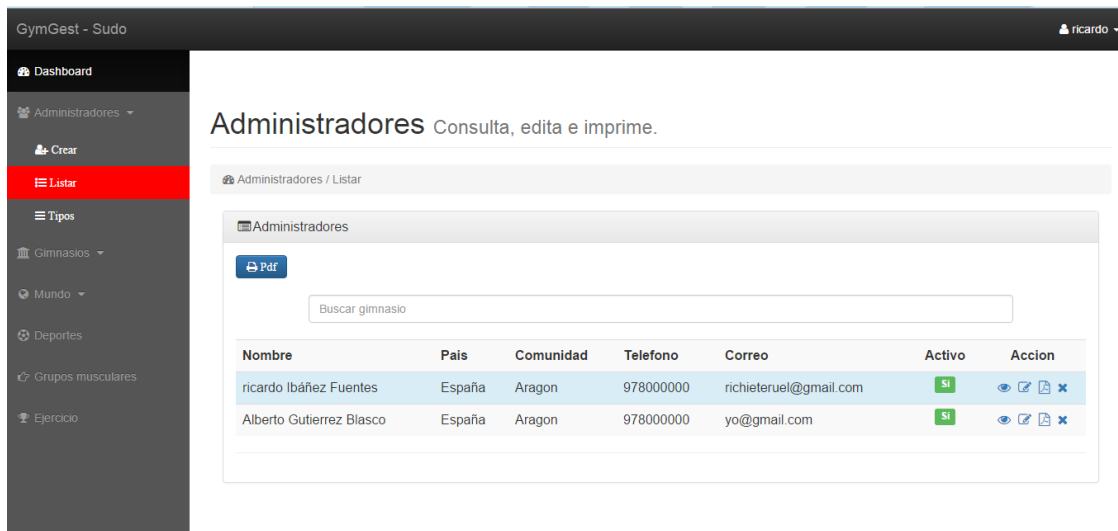


Ilustración 0-15 Listado Administradores

En esta interfáz podemos observar que varias cosas:

- **Lista de adminsitradores:** Contiene una tabla con los todos los administradores existentesn en el sistema (aparecerán paginados cuando la tabla tenga más de diez administradores).
- **Buscador:** Podremos introducir una serie de caracteres a buascar. Para verlo más claro haremoun ejemplo:

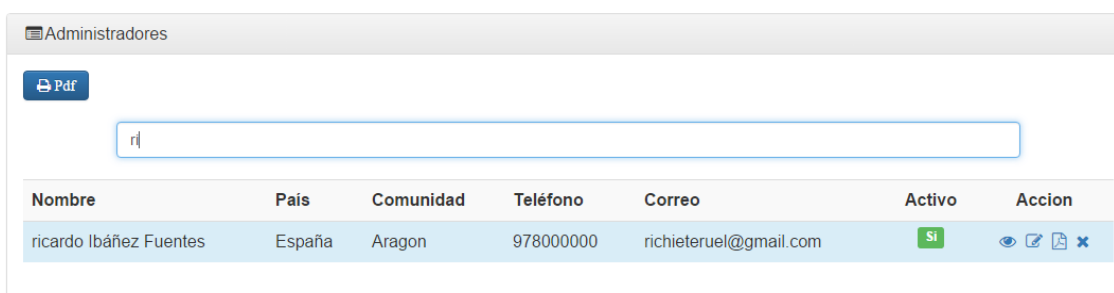


Ilustración 0-16 Buscador Administradores

Podemos observar como el administrador ha introducido los caracteres “ri”, el sistema automáticamente ha buscado sus coincidencias en la BD y ha descartado los administradores que no las tienen (las coincidencias serán buscadas en los campos nombre y apellidos de la base de datos).

- **Botón PDF:** Este botón nos abrirá un documento PDF en una nueva pestaña con los datos básicos de los administradores.



GymGest

Fecha: 02/12/2015

LISTA DE ADMINISTRADORES

NOMBRE	PAÍS	CCAA	TELEFONO	MAIL
ricardo Ibáñez Fuentes	España	Aragon	978000000	richieteruel@gmail.com
Alberto Gutierrez Blasco	España	Aragon	978000000	yo@gmail.com

Ilustración 0-17 Exportar a PDF

- Por ultimo nos queda comentar los cuatro botones que contiene el campo acción:
 - **Ver:** Este nos muestra más información acerca del administrador.


☒ Ver datos.

Datos personales		Datos del sistema	
Nombre	ricardo	Numero accesos	0
Nif	18451238-V	Habilitado	SI
País	España	Tipo de administrador	Sudo
Comunidad	Aragon		
Provincia	Teruel		
Población	Teruel		
Dirección	Calle Falsa		
CP	44002		
Teléfono	978000000		
Email	richieteruel@gmail.com		
CP	44002		

☒ Modificar
 ☒ Volver

Ilustración 0-18 Ver Administrador

- **Editar:** Se nos abrirá un formulario con todos los datos del administrador seleccionado para que este pueda ser modificado.

Administradores [Editar.](#) Administradore / Editar

Nombre <input type="text" value="Alberto"/>	Apellidos <input type="text" value="Gutierrez Blasco"/>
DNI <input type="text" value="48141567-Z"/>	Tipo de administrador <input type="text" value="Sudo"/>
Fecha de nacimiento <input type="text" value="02/12/2015"/>	País <input type="text" value="España"/>

Ilustración 0-19 Editar Administrador

Tras editarlos los datos podremos observar como se han realizado los cambios (En este caso se ha modificado su primer apellido).

Nombre	País	Comunidad	Telefono	Correo	Activo	Accion
ricardo Ibáñez Fuentes	España	Aragon	978000000	richieteruel@gmail.com	<input checked="" type="checkbox"/>	Ver Editar Eliminar
Alberto Gimenez Blasco	España	Aragon	978000000	yo@gmail.com	<input checked="" type="checkbox"/>	Ver Editar Eliminar

Ilustración 0-20 Listados

- **PDF:** Generará un documento PDF igual que el mostrado a la hora de crear el administrador solo que esta vez no se le mostrará la contraseña



GymGest

Fecha: 02/12/2015

Nombre: ricardo	Apellidos: Ibáñez Fuentes	Fecha de nacimiento: 1987-11-18
País: España	Comunidad: Aragon	Provincia: Teruel
Población: Teruel	Dirección: Calle Falsa	Código postal: 44002
Telefono: 978000000	E-Mail: richieteruel@gmail.com	<input type="text" value=""/>

Ilustración 0-21 Exportar a PDF

- **Dar de baja:** Al pulsar este botón se nos mostrará la siguiente ventana. En ella nos pedirá que confirmemos los cambios.

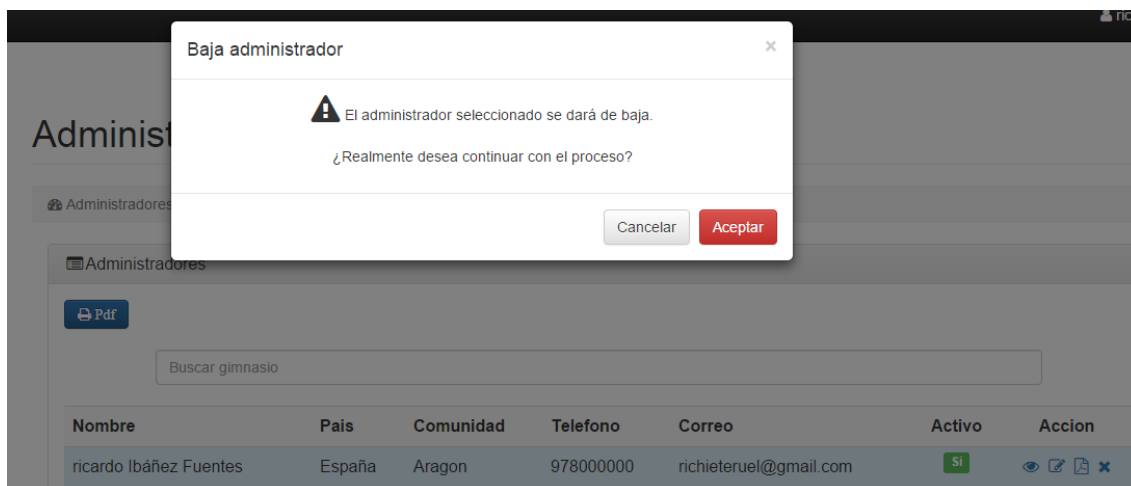


Ilustración 0-22 Modal

Si pulsamos aceptar, veremos los siguientes cambios:

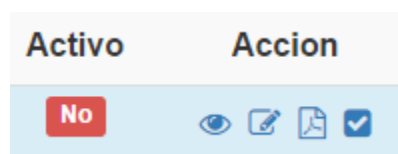


Ilustración 0-23 Acciones

En la imagen anterior visualizamos como el botón para dar de baja a un administrador, se ha sustituido por uno para darlo de alta, además su estado se ha cambiado. También se mostrará el siguiente mensaje.



Ilustración 0-24 Éxito

Si pulsamos el nuevo botón para volver a dar de alta al administrador, se nos mostrará una ventana parecida a la anterior pero haciendo el proceso inverso.

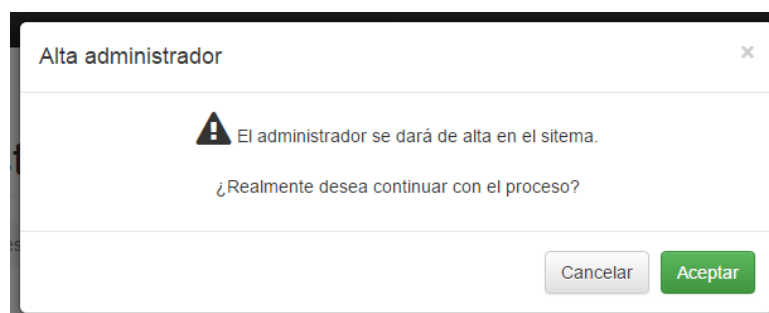


Ilustración 0-25 Modal

Al pulsar el botón aceptar, el administrador volverá a su estado inicial.

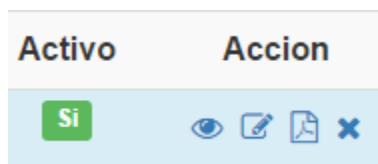


Ilustración 0-26 Acciones

Tipos de administrador

Al crear un nuevo administrador nos aparecía un desplegable para seleccionar el tipo de administrador. La lista que en el aparecía correspondía con los que hay en la tabla de la siguiente imagen:



Ilustración 0-27 Tipos Administradores

Como podemos observar esta tabla solo tiene un tipo de administrador el cual puede ser gestionado. A lo largo del proyecto ya se ha comentado que inicialmente solo hay un tipo de administrador, y aunque se cree un nuevo administrador, este no tendrá ningún efecto en el sistema. Por ello esta interfaz la comentaremos más adelante en interfaces idénticas.

Gimnasios: La gestión de gimnasios es igual a la de administradores. Solo que no tiene tipo de administrador si no que tiene tipo de gimnasio, en esta lista podremos introducir tipos como:

- Gimnasio.
- Gimnasio femenino.
- MacroGimnasio
- LowCost

Mundo: Este apartado nos permite gestionar todo lo referente al mundo, es decir los países, las comunidades y las provincias que se asignan a todos los usuarios para saber más acerca de su procedencia.



Ilustración 0-28 Menú Lateral

En la ilustración anterior podemos observar como hacemos la gestión de cada una de estos de manera independiente. Porcedemos a verlos en detalle:

- **Países:** Inicialmente podremos ver en una tabla todos los países que tenemos dados de alta en nuestra BD, junto a su nombre tenemos su latitud y longitud, estas coordenadas son usadas para mostrá los gimnasios por país en el mapa del dashboard.

En esta tabla tambien tenemos el campo acción similar al de administradores en el que podremos editar un país o **eliminarlo**.

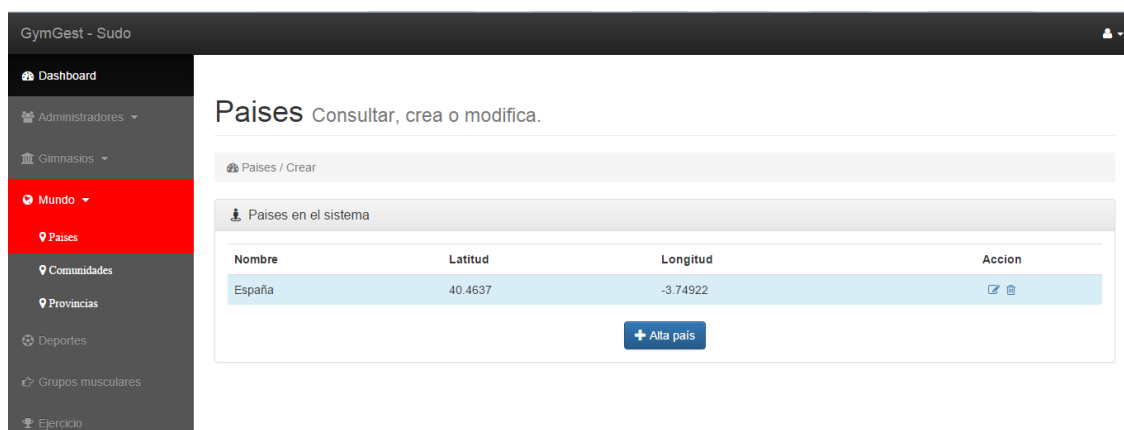


Ilustración 0-29 Países

Para añadir un nuevo país solo tan solo debemo pulsar sobre el botón “Alta país”, de esta forma accederemos a un formulario para su creación (el formulario aparecerá de forma totalmente dinámica)

En la siguiente imagen podremos ver el formulario que aparece tras pulsar el botón. En el podemos ver un mapa de google y una serie de casillas. Para dar de alta un nuevo país solo debemos de poner el nombre en el campo “nombre del país” y pulsar sobre el botón “Obtener coordenadas” ó pulsar “enter”

Ilustración 0-30 Mapa

La imagen anterior corresponde con la búsqueda del país francia, podemos observar como el campo latitud y longitud han sido rellenados automáticamente. Esto minimiza el fallo a la hora de introducir un país, además hace que una faena engorrosa sea una faena simple. Para guardar estos datos tan solo debemos pulsar el botón “Guardar país”.

- **Comunidades:** En esta interfaz podemos visualizar todas las comunidades existentes, así como el país al que pertenece. Junto a su nombre tenemos los correspondientes campos de edición y eliminación.

País	Comunidad	Acción
España	Aragon	[Edit] [Delete]

Ilustración 0-31 Comunidad

Cuando se desee crear una nueva comunidad autónoma el administrador tan solo debe pulsar el botón “Alta país” de esta manera se nos mostrará un formulario para crearla, tal y como podemos apreciar en la siguiente ilustración.

GymGest - Sudo

Dashboard

- Administradores
- Gimnasios
- Mundo
- Deportes
- Grupos musculares
- Ejercicio

Comunidades Consultar, crea o modifica.

Mundo / Comunidades

Crear nueva comunidades autonomas

Pais: España

Nombre de la comunidad:

[Guardar comunidad](#) [Volver](#)

Ilustración 0-32 Comunidades

Podemos observar como ha sido desplegada la lista de paises. Para insertar una nueva comunidad tan solo debemos seleccionar el país al que pertenece, poner su nombre y pulsar sobre el botón “Guardar comunidad”.

- **Provincias:** Aquí podemos ver todas las provincias existentes junto al nombre de la comunidad y país al que pertenecen.

GymGest - Sudo

Dashboard

- Administradores
- Gimnasios
- Mundo
- Países
- Comunidades
- Provincias
- Deportes
- Grupos musculares
- Ejercicio

Provincias Consultar, crea o modifica.

Mundo / Provincias

Comunidades autonomas

Pais	Comunidad	Provincia	Acción
España	Aragon	Teruel	Editar Eliminar

[+ Alta provincia](#)

Ilustración 0-33 Provincias

Al pulsar sobre “Alta provincia” se nos mostrará el siguiente formulario:

Ilustración 0-34 Provincias

En la pantalla anterior podemos observar como para crear una nueva provincia, tan solo debemos introducir su nombre y seleccionar su provincia (esta va acompañada con el nombre del país al que pertenece). Para almacenarla bastará con pulsar “Guardar provincia”.

Deportes: Aquí podemos listar todos los deportes existentes en la base de datos, además también podremos crearlos, editarlos y borrarlos al igual que pasaba con “tipo de administrador”.

Nombre	Acción
Futbol	
Tenis	
Baloncesto	
Karate	

+ Nuevo tipo

Ilustración 0-35 Deportes

Cuando el administrador pulse sobre el botón “Nuevo tipo” se desplegará una ventana modal con un formulario, en ella podremos introducir el nombre del nuevo deporte.

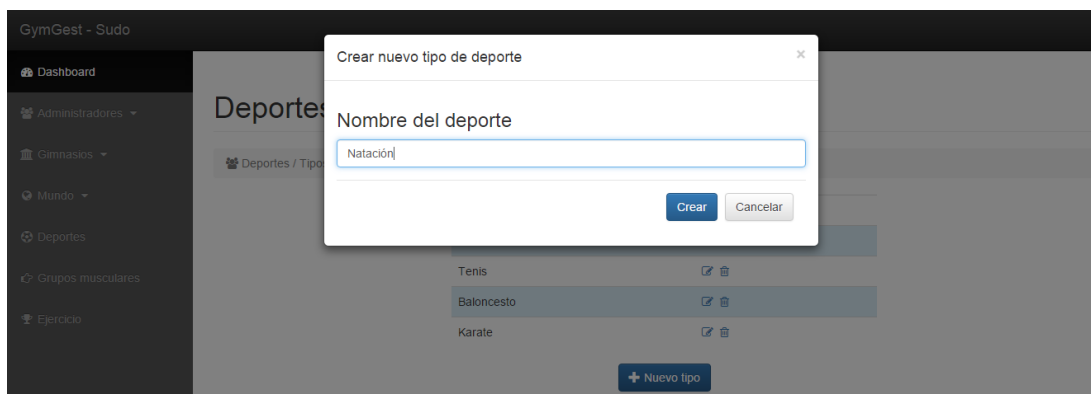


Ilustración 0-36 Nuevo Deporte

En la imagen anterior podemos ver que al pulsar el botón se ha desplegado la ventana donde hemos introducido el nombre del deporte, al pulsar sobre crear se guardará en la base de datos y se mostrará el siguiente mensaje.

✓ Deporte creado con éxito.

Ilustración 0-37 Éxito

Además podemos observar como el deporte ya aparece en la lista.











Nombre	Acción
Futbol	 
Tenis	 
Baloncesto	 
Karate	 
Natación	 

Ilustración 0-38 Listado

Sobre los campos de la tabla podemos realizar dos acciones, estas son las siguientes:

- **Editar:** Al pulsar sobre este botón nos aparecerá el nombre del deporte seleccionado en un formulario para su modificación.

Editar.

Nombre del deporte

Natación

Guardar cambios

Ilustración 0-39 Edición

Una vez se ha realizado los cambios tan solo debemos pulsar sobre el botón existente, quedando así modificados los datos del deporte.

✓ Deporte modificado correctamente.

Nombre	Acción
Fútbol	
Tenis	
Baloncesto	
Karate	
Natación olímpica	

Ilustración 0-40 Éxito Edición

Podemos observar como al ser modificado un deporte nos aparece su correspondiente mensaje y su modificación.

- **Eliminar:** Al pulsar este botón se mostrará un mensaje preguntandonos si realmente deseamos hacerlo.

Eliminar dato

⚠ Los datos serán eliminados de la base de datos de manera permanente.

¿Realmente desea continuar con el proceso?

Cancelar Eliminar

Ilustración 0-41 Modal

Cuando pulsemos el botón eliminar el deporte será suprimido de la lista, además se no aparecerá un mensaje indicandonos su correcta eliminación.

✓ Deporte eliminado correctamente.









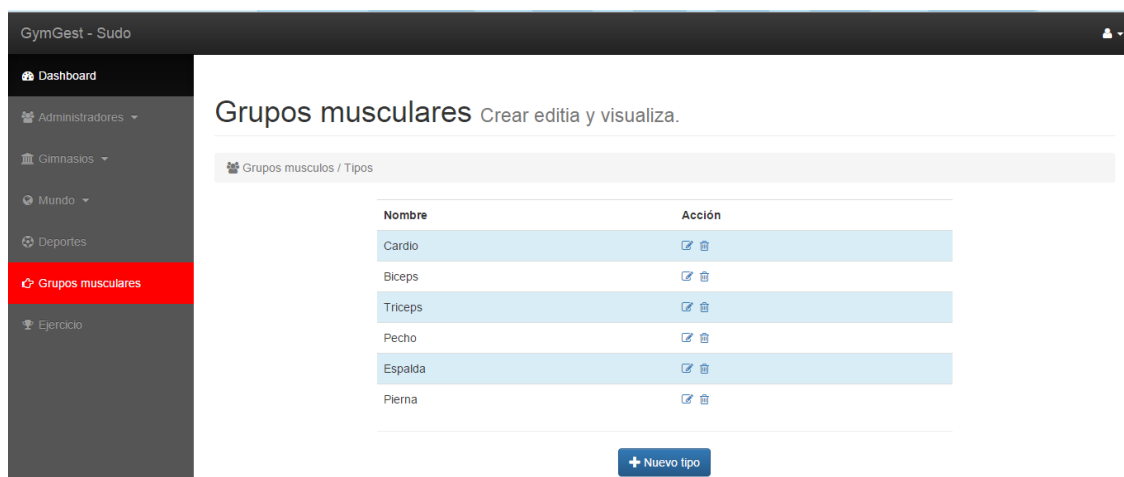
Nombre	Acción
Futbol	 
Tenis	 
Baloncesto	 
Karate	 

Ilustración 0-42 Éxito Eliminado

Grupos musculares: Aquí podemos encontrar todos los grupos musculares que han sido creados en la base de datos, y que se utilizarán para agrupar los ejercicios.



GymGest - Sudo

Dashboard

Administradores

Gimnasios

Mundo

Deportes













Grupos musculares

Ejercicio

Grupos musculares

Crear edita y visualiza.

Grupos musculares / Tipos

Nombre	Acción
Cardio	 
Biceps	 
Triceps	 
Pecho	 
Espalda	 
Pierna	 

+ Nuevo tipo

Ilustración 0-43 Grupos Musculares

En esta pantalla no hay nada que destacar puesto que la gestión se hace igual que la de deportes.

Ejercicio: Aquí se gestionan todos los ejercicios que han sido creados, como podemos ver estos se listan de forma paginada y tendremos un buscador para encontrar con mayor rapidez los ejercicios.

Ejercicios Crear, editar y visualizar deportes.

Deportes / Tipos

Ejercicios

Buscar ejercicio

Nombre	Articulación	Isométrico	Habilitado	Imagen	Acción
Bicicleta	Cardio	NO	SI	SI	
Cinta de correr	Cardio	NO	SI	SI	
Elíptica	Cardio	NO	SI	SI	
Remo	Cardio	NO	SI	SI	
Escaleras	Cardio	NO	SI	SI	
En banco scott	Biceps	NO	SI	SI	
Martillo alterno	Biceps	NO	SI	SI	
En barra	Biceps	NO	SI	SI	
Concentrado	Biceps	NO	SI	SI	
Cruz en polea alta	Biceps	NO	SI	SI	

Nuevo ejercicio

Ilustración 0-44 Listado Ejercicios

- **Buscar:** Al poner el nombre del ejercicio a buscar aparecerán todas sus coincidencias.

pre|

Nombre	Articulación	Isométrico	Habilitado	Imagen	Acción
Press francés	Triceps	No	Si	Si	
Pres banca	Pecho	No	Si	Si	
Pres banca declinado	Pecho	No	Si	Si	
Prensa inclinada	Pierna	No	Si	Si	

Ilustración 0-45 Buscador Ejercicios

En la imagen anterior podemos ver como se ha introducido los caracteres “pre” automáticamente han sido buscadas sus coincidencias en la BD de manera asincrónica mostrándolas en la tabla.

- **Ver:** Podemos observar los datos del ejercicio, también se mostrará su imagen.

Pres banca

Articulacion	Isometrico	Habilitado
Pecho	No	Si



[Editar](#) [Volver](#)

Ilustración 0-46 Edición Ejercicios

En la ilustración anterior visualizamos también un botón para editarlo al igual que en la tabla.

- **Editar:** Se abrirá un formulario en el que se presentarán sus datos, además podremos modificar la imagen en caso de que lo deseemos.


Crear nuevo ejercicio

Nombre del ejercicio:

Articulacion:

Es isometrico: ☐ ☒

Habilitar: ☐ ☒



Cambie la imagen si lo desea.

[Seleccionar archivo](#) | Ningún archivo seleccionado

[Editar ejercicio](#) [Volver a listado](#)

Ilustración 0-47 Edición Ejercicios

- El botón para dar de alta o de baja un ejercicio funciona de la misma manera que se hacía en administradores.

Debajo de la lista de deportes se hay un botón para crear un nuevo deporte. Al pulsarlo aparecerá el formulario correspondiente a la creación de un nuevo ejercicio.

Crear nuevo ejercicio

Nombre del ejercicio:

Articulaion:

Es isometrico: ☐ ☐

Habilitar: ☐ ☒

Seleccione la imagen: elevacion de talones.jpg

Ilustración 0-48 Modificación

Una vez rellenados todos los campos y seleccionado (o no) una imagen podremos crearlo. A continuación podemos observar como ha sido creado correctamente el ejercicio:

Buscar ejercicio

Nombre	Articulaion	Isometrico	Habilitado	Imagen	Acción
Elevación de talones	Pierna	NO	SI	SI	Ver Editar Eliminar

[< First](#)
[<](#)
[2](#)
[3](#)
[4](#)

Ilustración 0-49 Listado

Gimansio

En el dashboard del gimnasio se mostrara el número de monitores y clientes que tienen dados de alta, además podremos observar los que han sido creados hasta que vemos sus datos. En la ilustración siguiente no tenemos creados usuarios recientemente.

GymGest People

Dashboard

Bienvenido a GymGest.

Dashboard

Monitores: 1

Cientes: 2

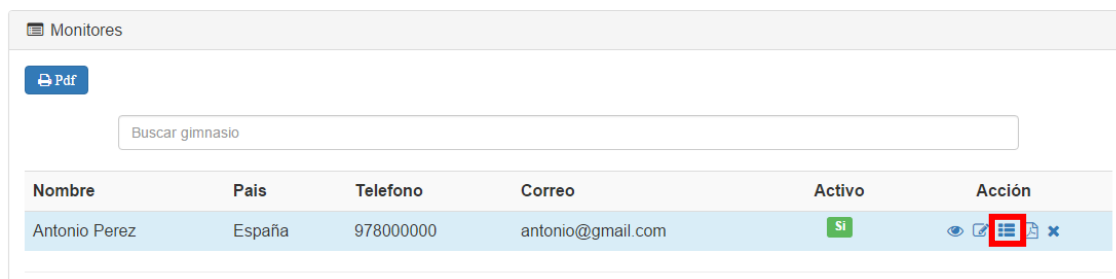
Monitores: No hay nuevos administradores en el sistema. [Ver todos los monitores](#)

Cientes: No hay nuevos gimnasios en el sistema. [Ver todos los clientes](#)

Ilustración 0-50 Gimnasio

Las opciones que dispone para hacer un gimnasio son dos, ambas se gestionan de la misma manera que se hacía con un administrador a excepción de unas opciones extra, que detallamos a continuación:

Gestión de monitores: Aquí solo apreciamos un nuevo icono, este nos permitirá ver las tablas que ha hecho un monitor.

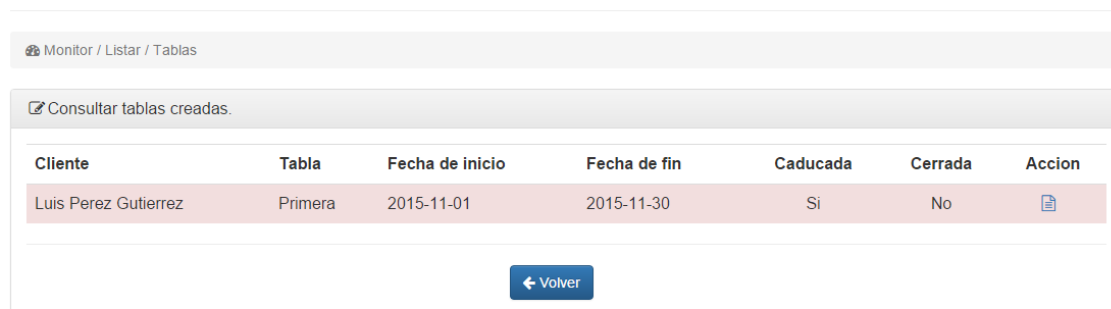


Nombre	Pais	Telefono	Correo	Activo	Acción
Antonio Perez	España	978000000	antonio@gmail.com	Si	[Icons: eye, link, calendar (highlighted), print, close]

Ilustración 0-51 Monitores

Al pulsar el icono veremos la siguiente pantalla.

Monitor Consultar tablas creadas.



Cliente	Tabla	Fecha de inicio	Fecha de fin	Caducada	Cerrada	Accion
Luis Perez Gutierrez	Primera	2015-11-01	2015-11-30	Si	No	[PDF icon]

[← Volver](#)


Ilustración 0-52 Tabla Cliente


Podemos apreciar como el monitor seleccionado solo tiene una tabla además está caducada. En el campo acción tenemos un botón que nos permitirá visualizar en PDF la tabla mencionada (pero este caso lo detallaremos al final).

Gestión de clientes: Sobre los clientes podemos hacer dos cosas más:

- La primera será asignar o modificar deportes y lo haremos mediante un botón situado en el campo acción.

Cientes Consulta, edita e imprime.

 Clientes / Listar

 Pdf

Buscar cliente











Nombre	Telefono	Correo	Monitor	Activo	Accion
Luis Perez Gutierrez	0978000000	yo@gmail.com	Antonio Perez	Si	    
Marta Perez	0978000000	yo@gmail.com	Antonio Perez	Si	    

Ilustración 0-53 Clientes

Al pulsar este botón nos aparecerá un formulario en cual están todos los deportes existentes en la BD situados en la derecha, y los que le han sido establecidos en la parte izquierda.

Seleccione los deportes que practica en la actualidad.

Futbol
Baloncesto

Pasar »
Todos »

« Quitar
« Todos

Karate
Tenis

Procesar formulario

Ilustración 0-54 Insertar Deportes

Podemos observar como el cliente seleccionado ya practica dos deportes, tambien se visualiza como se pasan de manera facil los deportes de un lado al otro.

- La siguiente diferencia se encuentra a la hora de visualizar los datos:

Cliente Ver datos.

Cliente / Ver

☒ Ver datos.

Datos personales

Nombre	Luis	Apellidos	Perez Gutierrez
DNI	123456789-Z	Fecha de nacimiento	1990-11-04
País	España	Comunidad	Aragon
Provincia	Teruel	Población	Teruel
Teléfono	0978000000	Email	yo@gmail.com
Monitor	Antonio Perez	Habilitado	SI

Deportes

Karate

Tenis

[Modificar](#) [Tablas](#) [Volver](#)

Ilustración 0-55 Ver Clientes

Se puede apreciar que hay un nuevo botón llamado “Tablas” al pulsar sobre este se verán todas las tablas que tiene.

Cliente Ver datos.

Cliente / Ver

Gestión de tablas

Tablas del cliente

Nombre	Inicio	Fin	Acción
Primera	2015-11-01	2015-11-30	Ver Editar

[Volver](#)

Ilustración 0-56 Ver Tablas Cliente

En caso de que no tenga ninguna aparecerá el siguiente mensaje:

Cliente Ver datos.

Cliente / Ver

Gestión de tablas

Tablas del cliente

⚠ Este cliente no dispone de ninguna tabla por el momento.

[Volver](#)

Ilustración 0-57 Ver Tablas Cliente

Monitor

En su dashboard únicamente podremos visualizar todas las tablas que tenga caducadas con el fin de que lleve una gestión diaria sobre sus clientes.

Tablas vencidas			
Nombre	Tabla	Fecha de inicio	Fecha de fin
Luis Perez Gutierrez	Primera	2015-11-01	2015-11-30
Alvaro Antón	Buena	2015-12-01	2015-12-01

Ilustración 0-58 Tablas Vencidas

La gestión de los clientes por parte del monitor, se basa unicamente en la visualización, asignación y creación de tablas, estos se esplican a continuación:

GymGest - Monitor Antonio

Dashboard

Clientes

Clientes Consulta, edita e imprime.

Clientes / Listar

Nombre	Monitor	Activo	Accion
Luis Perez Gutierrez	Sin monitor	Si	
Marta Perez	Antonio Perez	Si	
Alvaro Antón	Antonio Perez	Si	
Luis Perez	Sin monitor	Si	

Ilustración 0-59 Consulta Clientes


- **Asignación:** El monitor seguira a los clientes que le han sido impuestos por un gimnasio asi como los clintes que no pertenezcan a ningun monitor. Para asignarselo solo tendra que pulsar sobre el botón del campo acción que le permita hacerlo, tras esto le aparecera una ventana indicandole si realmente desea hacerlo.



Ilustración 0-60 Modal

Tras aceptar el botón se verá como realmente le ha sido asignado.

Cientes Consulta, edita e imprime.

 Cientes / Listar

✓ Un cliente se ha añadido a tu lista de seguimiento.

 Cientes





Nombre	Monitor	Activo	Accion
Luis Perez Gutierrez	Antonio Perez	Si	 

Ilustración 0-61 Consulta Cliente

- Ver: Podrá visualizar los datos del cliente así como los deportes que practica.

 Cliente / Ver

 Ver datos.

Datos personales

Nombre	Luis	Apellidos	Perez Gutierrez
DNI	123456789-Z	Fecha de nacimiento	1990-11-04
Monitor	Antonio Perez	Habilitado	SI

Deportes

Karate


Tenis

 Volver



 Tablas

Ilustración 0-62 Ver cliente

- Para crearle nuevas tablas debemos pulsar sobre el botón “Tablas”.

 Gestión de tablas

Tablas del cliente

Nombre	Inicio	Fin	Accion
Primera	2015-11-01	2015-11-30	 


 Tabla generica

 Tabla diaria

 Volver

Ilustración 0-63 Gestión de tablas

En esta pantalla además de visualizar y poderse la dar de baja, tenemos dos botones para crear dos tipos diferentes de tablas. Como ambos tienen pasos comunes primero los detallamos

Paso 1: Debemos detallar todos los datos de la tabla como si queremos crearla activa, también deberemos introducir su nombre así como su fecha de inicio y su fecha de fin.

GymGest - Monitor Antonio

Dashboard
 Clientes

1 Datos generales 2 Entrenamiento 3 Días a entrenar

Nombre: Comienzo

Activo: No SI

Fecha de inicio: 01/11/2015

Fecha de finalización: 31/12/2015

Siguiente →

Ilustración 0-64 Alta tabla

Paso 2: En este seleccionamos todos los ejercicios que debe hacer el usuario, como se puede ver estos están agrupados en diferentes pestañas.

GymGest - Monitor Antonio

Dashboard
 Clientes

1 Datos generales 2 Entrenamiento 3 Días a entrenar

Cardio Biceps Tríceps Pecho Espalda Pierna

Biceps - Seleccionar ejercicios

Ver: 10 entradas
 Buscar ejercicio...

Nombre	Seleccionar	Series	Repeticiones	Peso
Concentrado	No SI	4	10	10
Cruz en polea alta	No SI	4	10	35
En banco scott	No SI	4	10	20
En barra	No SI	4	12	15
Martillo alterno	No SI	Series	Repeticiones	Recomendado

Página 1 de 5 de entradas

<< 1 Siguiente >>

Volver Siguiente →

Ilustración 0-65 Alta tabla

Paso 3: Este se divide en dos, en función de si se quiere hacer una tabla genérica o diaria.

- **Tabla genérica:** Tendremos una tabla con una fila inicial en la que podremos seleccionar los grupos musculares que debe trabajar cada día de la semana, si deseamos que trabaje más de uno por día, deberemos añadir una nueva fila.

GymGest - Monitor Antonio

Dashboard Clients

1 Datos generales 2 Entrenamiento 3 Dias a entrenar

Selección de días de entrenamiento

Nueva fila Eliminar fila

Lunes	Martes	Miercoles	Jueves	Viernes	Sabado	Domingo
Cardio	Pecho	Cardio	Espalda	Cardio	Sin entrenamiento	Sin entrenamiento
Biceps	Sin entrenamiento	Triceps	Sin entrenamiento	Biceps	Sin entrenamiento	Sin entrenamiento

[Volver](#) [Terminar](#)

Ilustración 0-66 Alta Tabla

- **Tabla diaria:** Podremos seleccionar el día exacto en el que queramos que se trabajen los grupos musculares arrastrandolos de la lista de la izquierda al calendario.

Grupos musculares

- Cardio
- Biceps
- Triceps
- Pecho
- Espalda
- Pierna

December 2015

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	31	1	2
3	4	5	6	7	8	9

Ilustración 0-67 Calendario Tabla

Tras terminar la tabla podremos observar como ha sido creada, tambien podremos visualizarla usando el icono que hay en el campo acción.

Ciente [Ver datos.](#)

Ciente / Ver

✓ La tabla ha sido creada con éxito.

Gestión de tablas

Tablas del cliente

Nombre	Inicio	Fin	Acción
Primera	2015-11-01	2015-11-30	Ver Editar
Comienzo	2015-11-01	2015-12-31	Ver Editar

[Volver](#) [+ Tabla generica](#) [+ Tabla diaria](#)

Ilustración 0-68 Ver datos Cliente



Nombre: Luis Perez Gutierrez

Monitor: Antonio

Desde: 2015-11-01

Hasta: 2015-12-31

Lunes	Martes	Miercoles	Jueves	Viernes	Sabado	Domingo
Cardio Biceps	Pecho	Cardio Triceps	Espalda	Cardio Biceps		



Cardio



Bicicleta
20 minutos.



Cinta de correr
20 minutos.



Elíptica
20 minutos.

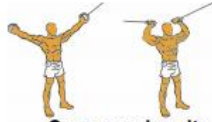


Escaleras
20 minutos.

Biceps



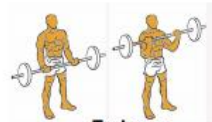
Concentrado
Ser. 4 Rep. 10
Peso: 10



Cruz en polea alta
Ser. 4 Rep. 10
Peso: 35



En banco scott
Ser. 4 Rep. 10
Peso: 20



En barra
Ser. 4 Rep. 12
Peso: 15

Ilustración 0-69 Final Tabla

Anexo III – Tecnologías Usadas

En este punto se pretende detallar todas y cada una de las tecnologías y aplicaciones usadas durante el desarrollo de este proyecto, con el fin de tener una mayor idea de cómo funciona y ha sido elaborado.

9.1 HTML



HyperText Markup Language “lenguaje de marcas de hipertexto”, hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código para la definición de contenido de una página web, como texto, imágenes, videos, entre otros.

Se considera el lenguaje web más importante y es el estándar que se ha impuesto en la visualización de páginas web. La página web contiene sólo texto mientras que recae en el navegador web (interpretador del código) la tarea de unir todos los elementos y visualizar la página final.

9.2 CSS3



Cascading Style Sheets “Hoja de estilo en cascada” o CSS es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

CSS tiene una sintaxis muy sencilla, que usa unas cuantas palabras clave tomadas del inglés para especificar los nombres de varias propiedades de estilo.

Una hoja de estilo se compone de una lista de reglas. Cada regla o conjunto de reglas consiste en uno o más selectores y un bloque de declaración (o «bloque de estilo») con los estilos a aplicar para los elementos del documento que cumplan con el selector que les precede. Cada bloque de estilos se define entre llaves, y está formado por una o varias declaraciones de estilo con el formato “**propiedad : valor**”;

9.3 JavaScript



JavaScript "JS" es un lenguaje de programación interpretado. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente, implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web para interactuar ella.

9.3.1 AJAX



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, mejorando la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página, aunque existe la posibilidad de configurar las peticiones como síncronas de tal forma que la interactividad de la página se detiene hasta la espera de la respuesta por parte del servidor.

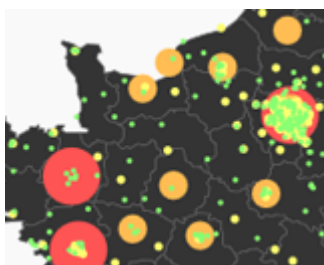
9.3.2 jQuery



jQuery es una biblioteca de JavaScript, que permite simplificar la manera de interactuar con los documentos HTML, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. jQuery es la biblioteca de JavaScript más utilizada.

jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

9.3.3 jQuery Mapael



Se trata de un sencillo plugin para jQuery basado en raphael.js que permite mostrar de forma dinámica mapas vectoriales en pantalla. Es una forma más de expresar javascript para crear presentaciones visualmente atractivas e interactivas.

9.3.4 Morris.JS

morris.js

good-looking charts shouldn't be difficult

Morris.js es una librería javascript que trabaja con jQuery y nos permite de forma muy sencilla crear gráficas de muchos tipos y con muy buena apariencia, gráficas de tipo Area, Line, Bar y Donut

9.4 JSON



(parser) de JSON.

JSON, acrónimo de JavaScript Object Notation, es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript que no requiere el uso de XML.

La simplicidad de JSON ha dado lugar a la generalización de su uso, especialmente como alternativa a XML en AJAX. Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos en este contexto es que es mucho más sencillo escribir un analizador sintáctico

9.5 Bootstrap



Es un framework o conjunto de herramientas de software libre 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.

Características

- Tiene un soporte relativamente incompleto para HTML5 y CSS 3, pero es compatible con la mayoría de los navegadores web.
- Soporta diseños sensibles, esto significa que el diseño gráfico de la página se ajusta dinámicamente, tomando en cuenta las características del dispositivo usado (Computadoras, tabletas, teléfonos móviles).
- Es de código abierto y está disponible en GitHub. Los desarrolladores están motivados a participar en el proyecto y a hacer sus propias contribuciones a la plataforma.

Estructura y Función

Bootstrap es modular y consiste esencialmente en una serie de hojas de estilo que implementan la variedad de componentes de la herramienta.

- **Sistema de cuadrilla y diseño sensible:** Utiliza 12 columnas con distintos layouts predefinidos con estructuras fijas a 940 píxeles o diseños fluidos (con %).



Ilustración 0-1 - Sistema de cuadrícula en Bootstrap

Posibles plantillas de diseño podrían ser usadas para distintos tipos de dispositivos: teléfonos móviles, formato de retrato y paisaje, tabletas y computadoras con baja y alta resolución (pantalla ancha). Esto ajusta el ancho de las columnas automáticamente.

9.6 php



PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor.

Características

- Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo.
- Se considera uno de los lenguajes más flexibles, potentes y de alto rendimiento conocidos hasta el día de hoy
- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- Es considerado un lenguaje fácil de aprender, destacando una amplia documentación en su sitio oficial.
- El código fuente escrito en PHP es invisible al navegador web y al cliente, ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con **MySQL** y PostgreSQL.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos. No requiere definición de tipos de variables aunque sus variables se pueden evaluar también por el tipo que estén manejando en tiempo de ejecución.
- Tiene manejo de excepciones (desde PHP5).

9.7 MySQL



Es un sistema de gestión de bases de datos relacional, multihilo y multiusuario. Desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009 que desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios.

Es muy utilizada en aplicaciones web, como Joomla, Wordpress, Drupal o phpBB, en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla. **Su popularidad como aplicación web está muy ligada a PHP, que a menudo aparece en combinación con MySQL.**

Es una base de datos muy rápida en la lectura cuando utiliza el motor no transaccional, pero puede provocar problemas de integridad en entornos de alta concurrencia en la modificación. En aplicaciones web hay baja concurrencia en la modificación de datos y en cambio el entorno es intensivo en lectura de datos, lo que hace a MySQL ideal para este tipo de aplicaciones.

Características:

- Amplio subconjunto del lenguaje SQL.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

9.8 MySQL Workbench



Es una herramienta visual de diseño de bases de datos que integra desarrollo de software, Administración de bases de datos, diseño de bases de datos, creación y mantenimiento para el sistema de base de datos MySQL.

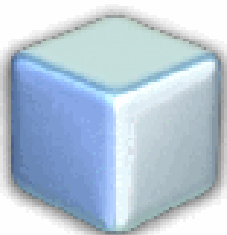
MySQL Workbench es uno de los primeros productos de la familia MySQL que ofrece dos ediciones diferentes:

1. **Community Edition:** Es un producto con todas las funciones que no está paralizado de ninguna manera.
2. **Standard Edition:** Extensión de "Community Edition" con una serie de módulos y plugins.

Características

- Soporte para características de MySQL 5
- Exportación a script SQL CREATE.
- Excelente editor visual de esquemas SQL.

9.9 NetBeans



NetBeans

Es un entorno de desarrollo integrado libre, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extenderlo. Este es un producto libre y gratuito sin restricciones de uso.

Es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos (Actualmente Sun Microsystems es administrado por Oracle Corporation).

NetBeans **permite crear aplicaciones Web** con PHP 5. Al tener también soporte para AJAX, cada vez más desarrolladores de aplicaciones, están utilizando NetBeans como IDE.

9.10 Codeigniter



Es un framework para aplicaciones web de código abierto para crear sitios web dinámicos con PHP. «Su objetivo es permitir que los desarrolladores puedan realizar proyectos mucho más rápido que creando toda la estructura desde cero, brindando un conjunto de bibliotecas para tareas comunes, así como una interfaz simple y una estructura lógica para acceder esas bibliotecas.

Utiliza MVC, que permite a los desarrolladores Web mejorar la forma de trabajar y también agregar mayor velocidad a la hora de crear una página Web.

Codeigniter no es el único framework del mercado, también están zend, cakephp, symphony, entre otros. Pero CI (Codeigniter) se diferencia al resto por lo rápido, lo extensible, lo sencillo de aprender y lo poderoso que puede llegar a ser si lo sabemos usar bien.

Características:

- Utiliza MVC es el acrónimo de Modelo Vista Controlador, que es un patrón para el desarrollo de software que separa la lógica de control, la interfaz del usuario y los datos del sistema. A continuación una explicación un poco más detallada del modelo, vista y controlador:
 - **Modelo:** es el que se encarga de la gestión de la base de datos y toda la lógica del negocio, el modelo permite abstraer al resto de la aplicación de cómo se obtienen o gestionan los datos. Lo cual nos da una ventaja ya que si después se requiere migrar la base de datos o cambiar algo, no va a ser necesario modificar ni la vista ni los controladores, lo cual nos facilita el trabajo. Otra cosa importante de destacar es que el modelo permite la reutilización de código ya que varios controladores pueden usar el mismo modelo sin que eso afecte en nada a los otros controladores y se evita tener que crear un modelo para cada controlador.
 - **Controlador:** es el centro de control de nuestra aplicación, se encarga de recibir la información del usuario procesarla, validarla, sacar o introducir la información lo que sea necesario con la ayuda de los modelos y crear una vista acorde a las necesidades del usuario.
 - **Vista:** La vista va a ser la interfaz para nuestro usuario. En nuestro sistema siempre vamos a tener diversidad de vistas que deberán ser cargadas por el control de acuerdo a las necesidades.
- **Flexibilidad:** Agrega librerías y drivers para ayudarnos a simplificar las tareas en el desarrollo de nuestro sistema, con funciones ya predefinidas que nos facilitan el trabajo aunque no son de uso obligatorio.

- **Versatilidad:** CodeIgniter es bastante menos rígido que otros frameworks. Define una manera de trabajar específica, pero en muchos de los casos podemos seguirla o no y sus reglas de codificación muchas veces las podemos omitir para trabajar como más a gusto nos encontremos.
- **Ligereza:** El núcleo de CodeIgniter es bastante ligero, lo que permite que el servidor no se sobrecargue interpretando o ejecutando grandes porciones de código.
- **Documentación tutorializada:** La documentación de CodeIgniter es fácil de seguir y de asimilar, porque está escrita en modo de tutorial. Esto no facilita mucho la referencia rápida, cuando ya sabemos acerca del framework y queremos consultar sobre una función o un método en concreto, pero para iniciarnos sin duda se agradece mucho.
- **Facilidad de instalación:** La configuración inicial se realiza con apenas la edición de un archivo, donde debemos escribir cosas como el acceso a la base de datos. Durante la configuración no necesitaremos acceso a herramientas como la línea de comandos.

Hay que destacar que CodeIgniter es más rápido que muchos otros entornos. Incluso en una discusión sobre entornos de desarrollo con PHP, Rasmus Lerdorf, el creador de PHP, expresó que le gustaba CodeIgniter

«porque es rápido, ligero y parece poco un entorno»

Anexo IV – Requisitos Funcionales

En este apartado se van a definir una serie de requisitos, todos ellos son obligatorios, no siendo aceptable una aplicación que no cumpla con alguno de los requisitos expuestos aquí. Estos serán especificados de manera clara y sencilla para que sean fáciles de comprender.

Requisitos Funcionales

- **RF001:** El sistema dejará registrarse al usuario.
- **RF002:** Para acceder a la aplicación, el usuario deberá identificarse con su correo electrónico y contraseña.
- **RF003:** El sistema debe comprobar la existencia de este usuario en la base de datos. Para ello lo debe buscar en tres sitios distintos, siendo estos: la tabla administrador, gimnasio, y monitor.
- **RF004:** Si el correo electrónico no existe en la base de datos se le mostrará un mensaje de error.
- **RF005:** En caso de que el usuario exista, pero la contraseña no sea correcta, se le mostrará un mensaje de error. Por otro lado el sistema debe de contar el número de intentos de acceso fallidos, almacenando también la hora en la que ha intentado acceder al sistema.
- **RF006:** En cualquier momento el usuario tendrá la posibilidad de cambiar su contraseña pulsando sobre un texto que diga “¿Olvidaste tu nombre de usuario o contraseña?”. Este texto debe estar situado en la interfaz de inicio.
- **RF007:** Al pulsar sobre “¿Olvidaste tu nombre de usuario o contraseña?”. La interfaz de inicio de sesión desaparecerá. Seguidamente aparecerá una interfaz donde se permita al usuario introducir un correo electrónico. Una vez Introducido el correo el usuario ha de recibir un correo electrónico un código aleatorio para restaurar su contraseña.
- **RF008:** El sistema generará aleatoriamente un código de seis dígitos.
- **RF009:** El código se almacenará en la base de datos y será válido durante 30 minutos.
- **RF010:** Cuando un usuario disponga del código enviado por el sistema, tendrá la opción de introducirlo pulsando sobre el texto “¿Ya tienes un código de recuperación?”.

- **RF011:** Al pulsar sobre el texto de disposición de código, la pantalla de inicio de sesión se difuminará, dando paso a un formulario donde introducir el correo electrónico y el código. Si el código está asociado al correo, será redirigido a un nuevo formulario, en caso contrario, se mostrará un mensaje de error.
- **RF012:** Cuando un usuario esté situado en el formulario de cambio de contraseña, deberá introducir el correo electrónico y dos veces la nueva contraseña. Si ambas contraseñas son iguales se cambiará la contraseña y se mostrará su página principal, si por el contrario las contraseñas no son iguales se mostrará un mensaje de error.
- **RF013:** Si el nombre de usuario y contraseña son correctos, debe distinguir que tipo de usuario es el que quiere acceder al sistema, pudiendo ser: administradores, gimnasios y monitores. También ha de poner el número de intentos fallidos a cero, así como borrar la hora en la que intento acceder al sistema.
- **RF014:** Si el usuario es de tipo administrador, el sistema le mostrará la página principal, siendo esta el dashboard de administradores.
- **RF015:** En caso de que el usuario sea un gimnasio, se le mostrará la página principal el dashboard de gimnasio.
- **RF016:** Si por último el usuario es de tipo monitor, se le mostrará como página principal el dashboard de monitor.
- **RF017:** La página web debe contener dos menús: una en la parte superior y otro en la parte izquierda.
- **RF018:** En el menú izquierdo, el usuario dispondrá de todas las opciones que puede hacer en el sistema.
- **RF019:** El menú superior contendrá un desplegable para acceder a su perfil y cerrar sesión.
- **RF020:** El usuario podrá consultar sus datos personales, accediendo a su perfil.
- **RF021:** Los datos de usuario mostrados en el perfil, podrán ser modificados.
- **RF022:** EL usuario podrá cambiar su contraseña cuando lo desee.
- **RF023:** A la pantalla de información general, se le llamará dashboard.

Los siguientes requisitos hacen referencia a ver información general de administrador.

- **RF024:** Se debe mostrar el número total de administradores, gimnasios, monitores y clientes que hay creados en la base de datos.
- **RF025:** Aparecerá un mapa del mundo. En él se mostrará un punto encima de cada país que use la aplicación. Al poner el ratón encima del punto, se mostrará el número de gimnasios dados de alta en ese país.
- **RF026:** Se mostrarán cuatro gráficas. Esas corresponden a: Ejercicios por edad, deportes por edad, deportes por comunidad y ejercicios según grupo muscular.
- **RF027:** Existirá un botón tras cada tabla, pulsándolo se desplegará una tabla detallando toda la información de las gráficas anteriores.
- **RF028:** Si la tabla que acompaña la gráfica, está desplegada se podrá volver a ocultar.
- **RF029:** Se mostrarán tres gráficas circulares. En ellas se observarán: gimnasios divididos por país, comunidad autónoma y provincia.
- **RF030:** El administrador tiene que ser consciente de los nuevos administradores creados en el sistema. Para ello se mostrará una tabla con los últimos administradores creados. Estos administradores desaparecerá de la tabla una vez el administrador lea su información.
- **RF031:** Si no hay ningún administrador creado recientemente, aparecerá un mensaje en color verde, indicando que no hay nuevos administradores.
- **RF032:** Se mostrarán los gimnasios que se dan de alta en el sistema, de esta forma el administrador tendrá mayor control. Para ello se mostrará una tabla con la información de los últimos gimnasios creados. Esta información desaparecerá una vez se lea su información.
- **RF033:** En caso de que no existan nuevos gimnasios. Se mostrará un mensaje en color verde, indicando que no existe nuevos gimnasios en el sistema.

A continuación se muestran requisitos, pertenecientes a ver información general de un gimnasio:

- **RF034:** El gimnasio podrá ver el número de monitores y clientes que tiene dados de alta.
- **RF035:** Cuando un gimnasio cree un monitor, aparecerá en el dashboard hasta que sea leído.

- **RF036:** En caso de que no tenga monitores para leer se mostrará un mensaje indicándoselo.
- **RF037:** Tras la creación de un nuevo cliente, se mostrará en el dashboard. La información se mostrará hasta que sea leído. Tras ser visualizado este desaparece de la lista.
- **RF038:** En caso de que el usuario no tenga ningún cliente nuevo, se mostrará un mensaje indicándole que no hay clientes por leer.

Los siguientes requisitos hacen referencia a la zona de administración de monitores.

- **RF039:** Se mostrará al monitor, todas las tablas que estén caducadas y no estén cerradas.

Dada la similitud que tienen varios de los casos de uso, se agrupan los requisitos para los casos de uso que corresponden a: tipos de administrador, tipos de gimnasio, País, CCAA, Provincia, Deportes, Grupos musculares.

- **RF040:** Se mostrará una tabla con todos los nombres existentes en la base de datos.
- **RF041:** En la tabla debe haber un campo llamado acción.
- **RF042:** El campo acción tendrá dos botones, uno para editar y otro para eliminar.
- **RF043:** Al pulsar sobre el botón editar, el administrador debe ser reconducido a un formulario donde podrá editar los datos.
- **RF044:** Cuando el administrador haya editado los datos podrá guardarlos mediante un botón disponible para ello al final del formulario.
- **RF045:** Al pulsar sobre el botón eliminar, se abrirá una ventana modal. En ella se explicará al administrador lo que ocurrirá en el sistema.
- **RF046:** La ventana modal de eliminación tendrá dos botones, uno para eliminar los datos y otro para cancelar.
- **RF047:** Cuando se pulse sobre el botón cancelar de la ventana modal será cerrada sin efectuar cambios.
- **RF048:** Se mostrará al administrador un mensaje tras la creación, eliminación y modificación.

Referentes a los casos de uso tipo administrador, tipo gimnasio, deportes y grupos musculares.

- **RF049:** Debajo de la tabla habrá un botón para crear un nuevo tipo.
- **RF050:** Al pulsar el botón para crear un nuevo tipo se abrirá una ventana modal.
- **RF051:** El modal explicado en el punto anterior, tendrá un formulario para introducir el nombre del tipo y un botón para crearlo en la BD.

Los siguientes casos de uso hacen referencia únicamente al caso de uso país.

- **RF052:** La tabla además contener el nombre de país (Ya mencionado anteriormente) deberá tener su latitud y longitud.
- **RF053:** Justo debajo de la tabla tendrá un botón para crear un nuevo país.
- **RF054:** Al pulsar el botón mencionado en el punto anterior desaparecerá la lista de países. Dando lugar a un formulario para añadir un país.
- **RF055:** El formulario para añadir un país tendrá un mapa de google, una casilla para introducir el nombre del país y dos casillas para introducir su latitud y longitud.
- **RF056:** El formulario tendrá tres botones, obtener coordenadas, guardar país, y cancelar.
- **RF057:** Cuando se introduzca el nombre del país y se pulse la tecla “enter”, o se clique sobre el botón obtener coordenadas, debe aparecer el país en el mapa y rellenar las casillas latitud y longitud.
- **RF058:** Si se pulsa sobre el botón guardar, se almacenará el país y se mostrará un mensaje indicando su éxito.
- **RF059:** En caso de que el usuario pulse el botón cancelar el formulario debe de desaparecer, volviendo a aparecer el listado de países.

Los casos de uso hacen referencia únicamente comunidad autónoma.

- **RF060:** En la tabla de comunidades autónomas, además del nombre aparecerá el país al que pertenece.
- **RF061:** Debajo de la tabla habrá un botón para crear una comunidad autónoma.
- **RF062:** Al ser pulsado el botón mencionado en el punto anterior, la lista de comunidades se desvanecerá dando paso al formulario para añadir una nueva.

- **RF063:** El formulario tendrá campos, el primero para seleccionar un país y el segundo para introducir el nombre de la comunidad.
- **RF064:** El formulario tendrá dos botones, uno para volver a lista de comunidades y otro para guardar una nueva comunidad.

A continuación se exponen los requisitos que hacen referencia a las provincias.

- **RF065:** En la tabla junto con el nombre de la provincia aparecerán dos campos más, el primero será el país y el segundo la comunidad autónoma a la que pertenece la provincia.
- **RF066:** Tras la tabla habrá un botón para añadir una nueva provincia.
- **RF067:** Cuando el botón comentado en el punto anterior sea pulsado desaparecerá la tabla de comunidades y en su lugar aparecerá un formulario.
- **RF068:** El formulario tendrá dos campos, el primero para seleccionar la comunidad a la que pertenece y el segundo para introducir el nombre de la provincia.
- **RF069:** Además de los campos, el formulario dispondrá de dos botones, el primero guardará la provincia y el segundo volverá a la tabla provincias.

A continuación se mostrarán los requisitos que hacen referencia a la gestión de clientes por parte del monitor.

- **RF070:** El monitor podrá ver en una tabla todos los clientes del gimnasio que le hayan sido asignados, o no estén asignados a ningún monitor.
- **RF071:** La tabla tendrá un campo llamado acción.
- **RF072:** El campo acción tendrá únicamente un botón, este le permitirá gestionar el cliente.
- **RF073:** Tras pulsar el botón mencionado anteriormente, se mostrará una tabla con información general del cliente, así como los deportes que practica.
- **RF074:** Justo debajo de la información del cliente, habrá dos botones, uno para volver a la lista de clientes y otro para acceder a la sección de tablas.
- **RF075:** Si el cliente pulsa sobre el botón para acceder a la sección tablas, la información general del cliente desaparecerá, mostrando un listado con sus tablas.
- **RF076:** Si el cliente no tiene ninguna tabla se mostrará un mensaje informativo.

- **RF077:** La tabla de rutinas contendrá, su nombre, fecha de inicio y fecha de fin.
- **RF078:** La tabla también tendrá un campo llamado acción, en puede habrá dos botones:
 - **RF078.1:** El primer botón será para visualizar la tabla en un documento PDF. Este botón será diferente en función de si es una tabla genérica o una tabla diaria.
 - **RF078.2:** El segundo botón también será diferente; de una forma o de otra en función de si esta activada o desactivada.
- **RF079:** Tras la tabla que contiene información sobre las rutinas se verán dos botones, el primero permitirá crear una tabla genérica, mientras que el segundo creará una tabla diaria.
- **RF080:** Tras los dos botones para la creación de tablas mencionados anteriormente, ha de aparecer un botón para volver a la información del cliente.

Los siguientes requisitos hacen referencia a la creación de una tabla genérica:

- **RF081:** Cuando el monitor pulse sobre crear un tabla genérica se abrirá un formulario dividido en tres pasos.
- **RF082:** El primer paso tendrá información general sobre la tabla, es decir, su nombre, fecha de inicio y fecha de fin. Junto a estos habrá un interruptor indicando si la tabla se creará activa.
- **RF083:** El segundo paso tendrá unas pestañas correspondientes a cada uno de los grupos musculares existentes en la de datos.
- **RF084:** Para el grupo cardio, se podrá seleccionar los ejercicios y después se deberá especificar el tiempo de duración.
- **RF085:** Para el resto de grupos musculares se podrán seleccionar los ejercicios. Además se deberá especificar el número de series, repeticiones así como el peso recomendado.
- **RF086:** En el tercer y último punto, se seleccionarán los días de la semana que se ejercitará cada uno de los grupos musculares.
- **RF087:** Tras pulsar sobre terminar, se guardará la información y se le notificará al usuario.

Debido a sus grandes similitudes se van a exponer una serie de requisitos de forma agrupada. Estos requisitos hacen referencia a la gestión de administradores, gimnasios, monitores y clientes.

- **RF088:** Los datos más relevantes se mostrarán e una tabla.
- **RF089:** La tabla contendrá un campo que indique si el usuario está activo en el sistema.
- **RF090:** Al final de la tabla debe de haber una casilla llamada acción.
- **RF091:** La casilla acción contendrá como mínimo cuatro botones, estos son, ver, editar, generar documento PDF y activar o desactivar.
- **RF092:** Los datos se mostrarán paginados, siendo el número máximo de usuarios por tabla de diez.
- **RF093:** Cuando el botón ver sea pulsado se redirigirá al usuario a una página, en la que se visualizarán todos los datos del usuario de manera más detallada.
- **RF094:** Si el botón editar es pulsado, se llevará al usuario a un formulario de edición; este estará relleno con los datos del usuario seleccionado. Deberá poder realizar los cambios que estime oportunos y guardarlos.
- **RF095:** En todo momento se debe poder generar un documento PDF, con la información detallada del usuario.
- **RF096:** Los documentos PDF, además de tener la información del usuario, así como el logotipo, el nombre de la aplicación, y la fecha actual.
- **RF097:** Si quiere restringir el acceso al sistema a un usuario, se deberá pulsar el botón para dar de baja.
- **RF098:** Cuando el botón dar de baja se pulse, se abrirá una ventana modal. Esta ventana informara al usuario de los cambios que se van a producir en el sistema, si los acepta, se dará de baja al usuario.
- **RF099:** Si el usuario no tiene acceso al sistema, el botón para darlo de baja se cambiará por uno para darlo de alta.
- **RF100:** Cuando se quiera dar de alta a un usuario en el sistema, se pulsará el botón provisto para ello y se abrirá una ventana modal indicando los cambios que se van a producir.
- **RF101:** La ventana modal contendrá dos botones, el primero será para aceptar la operación y el segundo cerrará la ventana sin efectuar ningún cambio.

- **RF102:** La página además de la tabla dispondrá de una casilla de texto a modo buscador.
- **RF103:** Cuando se introduzca una serie de caracteres en el buscador, estos serán buscados en algunos campos de la base de datos. En la tabla se mostrarán únicamente los usuarios que contengan los caracteres introducidos.
- **RF104:** Deberá contener un botón llamado PDF. Este botón generará un documento PDF, con todos los usuarios en el sistema.

Además de la de los requisitos expuestos anteriormente. La gestión de monitores y la de clientes, tendrá una serie de requisitos, estos se detallan a continuación:

Requisito fundamental, para la gestión de monitores:

- **RF105:** El campo acción contendrá un botón más, este botón mostrará todas las tablas generadas por el monitor a los clientes.

Requisitos para la gestión de clientes.

- **RF106:** Además de ver los datos del cliente seleccionado, se deberá visualizar las tablas que tiene.
- **RF107:** En el campo acción de la tabla, habrá un botón para asignarle deportes.

A continuación se van a detallar todos los requisitos que hacen referencia a la gestión de clientes por parte del monitor.

- **RF108:** El monitor tendrá tabla con los clientes que le ha asignado el gimnasio, así como los clientes que estén sin asignar a ningún monitor.
- **RF109:** El monitor dispondrá de dos botones, uno para ver datos del cliente y otro para asignarse o desasignarse clientes.
- **RF110:** Cuando el botón para la asignación de clientes sea pulsado se mostrará una ventana modal; esta le indicará le informará de los cambios que se van a producir.
- **RF111:** Si ya tiene asignado un cliente, el monitor tendrá un botón para dejar de seguir al usuario en lugar del botón de asignación.
- **RF112:** Cuando el botón para dejar de seguir al usuario sea pulsado, se abrirá una ventana modal.

- **RF113:** La ventana modal dispondrá de dos botones, el primero para aceptar los cambios en el sistema y el segundo para cancelar la operación.
- **RF114:** Cuando se pulse el botón para ver datos del deporte se redirigirá al monitor a otra página, en ella podrá ver los datos del cliente así como los deportes que practica.
- **RF115:** La ventana de visualización de datos tendrá dos botones, el primero para volver a la lista de clientes, y el segundo para ver las tablas del cliente.
- **RF116:** Cuando el botón para ver las tablas del usuario sea pulsado, la información del cliente desaparecerá, apareciendo en su lugar las tablas que el usuario tiene.
- **RF117:** En caso de que el usuario no disponga de ninguna tabla se mostrará un mensaje.
- **RF118:** La información que el monitor dispondrá sobre las tablas es la siguiente: nombre, fecha de inicio y fecha de finalización.
- **RF119:** Junto con la información de las tablas habrá un campo acción.
- **RF120:** El campo acción tendrá dos botones, el primero será para generar un documento PDF con la tabla, y el segundo para darla de alta o de baja.
- **RF121:** Justo debajo de la información de las tablas, el monitor tendrá dos botones, el primero para crear una tabla genérica y el segundo para generar una tabla diaria.

Anexo V – Diagramas de Clases

En este apartado se muestra los diagramas de la aplicación web GymGest.

Antes de entrar en detalle con cada una de las clases se muestran todas las clases que componen la aplicación.

A continuación se va a detallar todas las clases usadas en la aplicación, estas clases están agrupadas en dos grupos, controlador y modelo:

Controlador

Welcome

Gestiona todo lo referente al inicio de sesión por parte de usuario. Debido a la gran cantidad de información que se ha de comprobar para el inicio, el método validar se ha ido dividiendo en métodos más simples con el fin de hacer más legible el código.

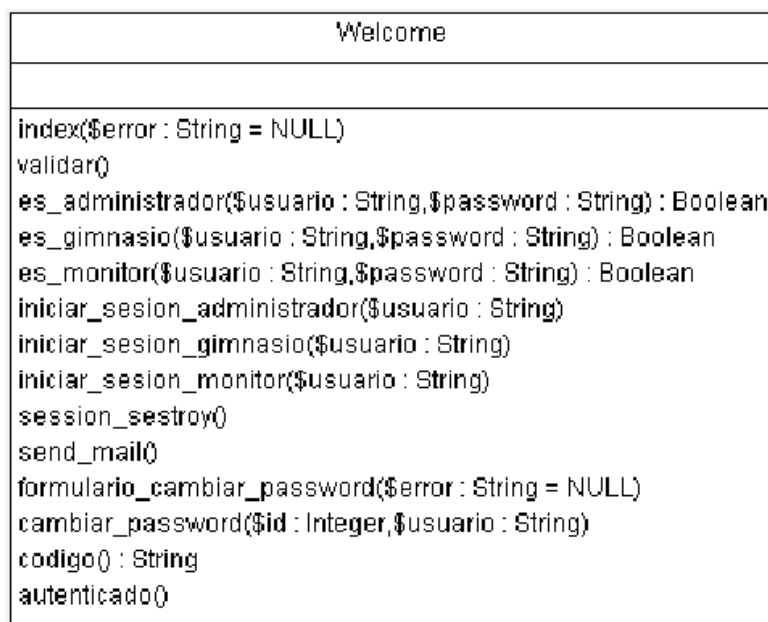


Ilustración 0-1 Welcome

- **Index:** Muestra el formulario para iniciar sesión.
- **Comprobar:** Verifica si el usuario que intenta acceder existe en la BD y a su vez tiene acceso al sistema, en caso afirmativo lo redirigirá a su dashboard, si por el contrario hay algún error, se harán cambios en la BD de datos con el objetivo de llevar un mayor control.

A continuación se detallarán unos submétodos encargados de ayudar a este método.

- **Es_administrador, es_gimnasio y es_monitor:** Comprueba únicamente si el tipo de usuario que intenta acceder al sistema. En caso de que el

primero fuera cierto devolverá “TRUE” y ya no comprobaría los siguientes métodos.

- **Iniciar_sesion_administrador,** **Iniciar_sesion_gimnasio,**
Iniciar_sesion_monitor: Recoge información sobre el usuario que accede al sistema, para guardarlos en sus datos de sesión.
- **sesión_destroy:** Elimina la sesión del usuario y llama al método index cuando este pulse sobre salir.
- **send_mail:** Recoge el correo de un usuario que no puede acceder al sistema para enviarle un código que le permita cambiar su contraseña, además el código y su hora de envió.
- **formulario_cambiar_password:** Cuando un usuario ya disponga del código e intente recuperar su contraseña, este método comprobará si todo es correcto, en caso afirmativo redirigirá al usuario a un formulario para restablecer su contraseña, si por el contrario encuentra algún fallo se lo notificará al usuario.
- **cambiar_password:** Únicamente cambia el password de un determinado usuario.
- **Código:** Método privado que crea un código de 6 dígitos de forma totalmente aleatoria.

Dashboard_sudo

Dashboard_sudo
index() newOperation() movimientos_gimnasios() : array movimientos_gimnasios() : array centrosPorPais() : array centrosPorComunidad() : array centrosPorProvincia() : array paises_activos() : array ejerciciosEdad() : array deportesComunidad() : array ejerciciosDeporte() : array ejerciciosArticulacion() calculaedad(\$fecha_nacimiento : String) : Integer

Ilustración 0-2 Drashboar Sudo

- **Index:** Envía una serie de datos básicos al dashboard, esto es debido al gran volumen de información que esta necesita para mostrar las gráficas, así que el resto de métodos serán invocados de manera asíncrona desde la interfaz.
- **Movimientos_gimnasio:** Recoge todos los gimnasios que han sido creados recientemente en el sistema.
- **Movimientos_administradores:** Busca todos los administradores que se han creado en el sistema y cuya casilla de leído este a cero.
- **CentrosPorPaís:** Comprueba y almacena el número de gimnasios que hay un país.
- **CentrosPorComunidad:** Recoge el número de gimnasios que hay en las comunidades.
- **CentrosPorProvincia:** Recopila información sobre el número de gimnasios que hay en cada provincia.
- **Países_activos:** Se extrae información sobre los países que están activos y el total de gimnasios que tiene cada uno de ellos.
- **EjerciciosEdad:** Recogemos los ejercicios que se practican según un rango de edad.
- **deportesComunidad:** Se observa que deportes se practican en cada comunidad autónoma.
- **ejerciciosDeporte:** Extraemos todos los ejercicios que se realizan según un rango de edad.
- **ejerciciosArticulacion:** observamos todos los ejercicios que se realizan de forma agrupada.
- **calcularEdad:** Método privado encargado de recoger una fecha nos devuelve un número con la edad actual del usuario.

Dashboard_gimnasio y Dashboard_Monitor

Estas clases mostrarán el dashboard a los usuarios correspondientes. Solo tiene dos métodos:

- **index:** Envía todos los datos a la interfaz.
- **autenticado:** Comprueba que el usuario está registrado y

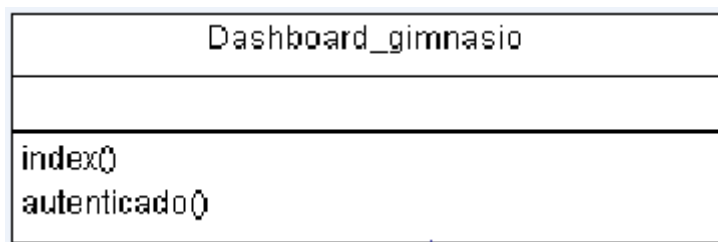


Ilustración 0-3 Drashboar Gimnasio

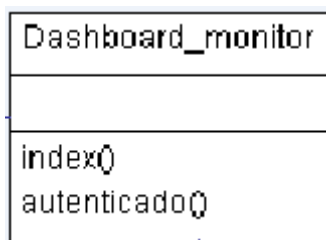


Ilustración 0-4 Drashboard Monitor

Clases: Administrador, Gimnasio, monitor y cliente.

Debido a su similitud, estas clases se agrupan para ser explicadas, cada una de ellas se encarga de la gestión de un tipo de usuario. A continuación, se van a detallar todos y cada uno de los métodos que contienen:

- **administradores, gimnasios, monitores e index:** Métodos encargados de listar todos los usuarios existentes en la BD de forma paginada. Además será invocada por otros métodos de la misma clase, pudiendo recibir o no un mensaje para mostrárselo al usuario.
- **formulario_alta:** Se ocupa de recoger todos los datos que se mostrarán en el formulario de alta del usuario, esta opcionalmente puede recoger los datos que un usuario ya había metido en un formulario, así como los errores producidos a la hora de intentar crearlo, estos datos se recogen para informar al usuario de los errores producidos y que pueda hacer los cambios en los datos introducidos.
- **crear:** Recibe todos los datos del formulario e intentará crear un nuevo usuario en caso de que todos los datos sean correctos. Si por el contrario hay algún dato incorrecto, invocará a la clase “formulario_alta”, pasándole todos los errores producidos y los datos introducidos por el usuario para que no tenga que volver a introducirlos.
- **ver:** Recibe el identificador de un usuario, y se encargará de recoger toda la información de dicho usuario para ser mostrarla de forma ordenada en el navegador.

- **formulario editar:** Recibe el identificador de un usuario y mostrará los datos en un formulario similar al usado para su creación. De esta manera los datos podrán ser modificados.
- **editar:** Este método será el encargado de recopilar todos los datos del formulario de edición, así como de realizar los cambios oportunos en la BD.
- **Baja:** Dado el id de un usuario, lo dará de baja en el sistema.
- **Alta:** Para dar de alta a un usuario nuevamente, se usará este método, similar al anterior.
- **marcar_leído:** Inicialmente el usuario se creará como no leído. Cuando un gimnasio o un administrador lea sus datos, este estado cambiará.
- **autocompletar:** Se encarga de recibir una serie de caracteres a buscar, y devolverle al usuario las coincidencias encontradas.
- **validar_dni_spanish:** Método privado que comprueba si el DNI introducido a la hora de crear o modificar un usuario es correcto.
- **Imprimir:** Genera un documento PDF con datos de los usuario.
- **imprimir “Usuario”:** Genera un documento PDF con todos los datos de un único usuario. Puede recibir una contraseña, si es así la escribirá también en el documento.
- **Autenticado:** Comprueba que el usuario tenga la sesión iniciada antes de enviarle datos.

Los métodos descritos a continuación se encuentran únicamente en las clases administrador y gimnasio. Estos métodos se encargan de la gestión de los tipos de administradores y de los tipos de gimnasio respectivamente.

- **Tipos:** Encargado de listar todos los tipos, podrá recibir un mensaje para mostrárselo al administrador.
- **crearTipo:** Recibe el nombre de un nuevo tipo y lo crea. Cuando haya sido creado llamara al método anterior pasándole un mensaje para que se le muestre al administrador.
- **eliminar tipo:** Recogerá el identificador de un tipo creado en el sistema y se encargará de eliminarlo.

- **formulario_editar_tipo:** Solicita a la BD los datos de un tipo para enviarlos a un formulario de edición.
- **editar_tipo:** Recibe los datos de un tipo y se encargará de hacer los cambios en la BD.

Los siguientes métodos se encargan de editar los perfiles de los usuarios que pueden acceder al sistema, siendo estos: administradores, gimnasios y monitores.

- **ajustes:** Muestra el perfil del usuario conectado. Además recopila todos los datos necesarios para que el administrador pueda modificarlos.
- **cambiar_password:** Encargado de cambiar el password del administrador conectado.
- **password:** Método privado encargado de generar de forma aleatoria un password.
- **editar_pefil:** Modifica los datos de un usuario.

Mundo

Esta clase se creará para ayudar a realizar una gestión de usuarios a nivel mundial, es decir, gestiona los países que hay en el sistema, así como las comunidades y provincias.

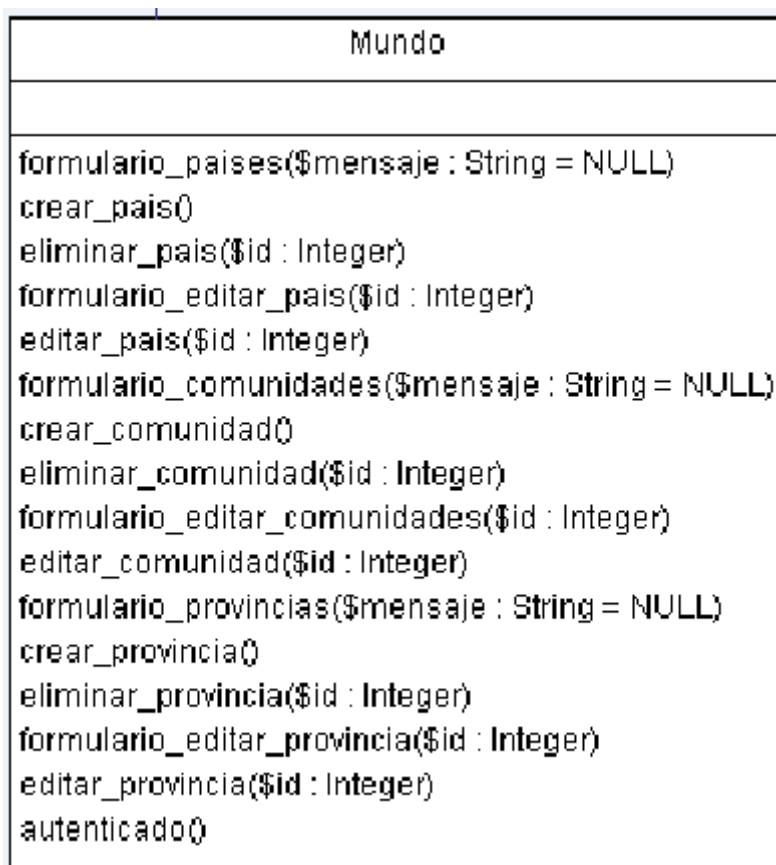


Ilustración 0-5 Mundo

Como ya se ha comentado anteriormente, ésta gestiona: países, comunidades y provincias. Cada uno de estos tiene las mismas clases. Por este motivo, tan solo se va a detallar la gestión de países.

- **formulario_paises:** Agrupa todos los datos necesarios para pasárselos a la vista y de esta forma el usuario podrá ver los países creados en el sistema así como crear otros nuevos.
- **crear_pais:** Recoge los datos enviados por un administrador para crear un nuevo país. Además invocará de nuevo el método anterior pasándole un mensaje para indicarle al usuario la creación de un nuevo país.
- **eliminar_pais:** Dado el identificador de un país lo elimina de la BD e Informará al usuario de ello.
- **formulario_editar_pais:** Recoge los datos disponibles en la BD sobre un determinado país y los envía a un formulario de edición para que el usuario pueda modificarlos.
- **editar_pais:** Recoge el identificador y los datos de un país a modificar y hace los cambios en la BD.

Deporte

Esta clase se encarga de la gestión de todos los deportes. Los métodos que contiene esta clase son los encargados de listar, crear, eliminar, y editar un deporte; ya que esta clase no contiene nada especial que deba ser comentado, no se va a detallar sus métodos.

Deporte
<pre>tipos_de_deporte(\$mensaje : String = NULL) crear_tipo() eliminar_tipo(\$id : Integer) formulario_editar(\$id : Integer) editar(\$id : Integer) autenticado()</pre>

Ilustración 0-6 Deporte

Musculo

Clase encargada de gestionar los grupos musculares almacenados en la BD. Al igual que pasa con la clase anterior únicamente contiene los métodos necesarios para listar, crear, eliminar y editar los grupos musculares.

Musculo
<pre>index(\$mensaje : String = NULL) crear() eliminar(\$id : Integer) formulario_editar(\$id : Integer) editar(\$id : Integer) autenticado()</pre>

Ilustración 0-7 Músculo

Ejercicio

Se encarga de la gestión de un tipo de ejercicio, esta tiene las mismas clases de siempre, la única diferencia a destacar es:

- Crear ahora se llama **do_upload**: El método se encargará de recoger el ejercicio junto con su imagen y de guardar los datos. En caso de no haber foto el campo de la tabla “URL_IMG” se pondrá a NULL.

Ejercicio
<pre> index(\$mensaje : String = NULL) formulario(\$error : String = NULL,\$datos : array = NULL) baja(\$id : Integer) alta(\$id : Integer) do_upload() ver(\$id : Integer) formulario_editar(\$id : Integer) editar(\$id : Integer) autenticado() </pre>

Ilustración 0-8 Ejercicio

Tabla

Tabla
<pre> cerrar(\$nombre : String) abrir(\$nombre : String) crear(\$id_cliente : Integer) newOperation() crear_ejercicios_cardio(\$id_sesion : Integer,\$identificadores_cardio : array,\$tiempo_cardio : array) crear_ejercicios_biceps(\$id_sesion : Integer,\$identificadores_biceps : array,\$series_biceps : array,\$repeticiones_biceps : array,\$peso_biceps : array) crear_ejercicios_triceps(\$id_sesion : Integer,\$identificadores_triceps : array,\$series_triceps : array,\$repeticiones_triceps : array,\$peso_triceps : array) crear_ejercicios_pecho(\$id_sesion : Integer,\$identificadores_pecho : array,\$series_pecho : array,\$repeticiones_pecho : array,\$peso_pecho : array) crear_ejercicios_espalda(\$id_sesion : Integer,\$identificadores_espalda : Integer,\$series_espalda : array,\$repeticiones_espalda : array,\$peso_espalda : array) crear_ejercicios_pierna(\$id_sesion : Integer,\$identificadores_pierna : array,\$series_pierna : array,\$repeticiones_pierna : array,\$peso_pierna : array) generar_pdf_tabla(\$id : Integer,\$nombre : String,\$id_gimnasio : Integer = NULL) generar_pdf_calendario(\$id : Integer,\$nombre : String,\$id_gimnasio : Integer = NULL) </pre>

Ilustración 0-9 Tabla

- **Cerrar:** Cierra el nombre una tabla.
- **Abrir:** Hace el paso contrario al punto anterior.
- **Crear:** crea una tabla genérica.
- **crear_tabla_diaria:** Crear una tabla diaria.
- Los siguientes cinco métodos se encarga de crear los ejercicios que tiene que hacer un cliente.

- **generar_pdf_tabla y generar_pdf_calendario:** Crea un documento PDF con sus respectivas tablas.

Modelos

En este apartado se definirán todas las clases del modelo. Todas estas clases terminan con “_model” ya que CodeIgniter así nos lo exige. Estas clases hacen de puente intermedio entre el controlador y la base de datos.

Dashboard_sudo_model

Esta clase es la encargada de generar todas las consultas a la BD que hace la clase Dashboard_sudo.

Dashboard_sudo_model
<code> contarPaises(\$idPais : Integer) : Integer contarComunidades(\$idComunidad : Integer) : Integer contarProvincias(\$idProvincia : Integer) : Integer gimnasios_por_pais(\$idPais : Integer) : Integer total_gimnasios() : Integer total_administradores() : Integer total_monitores() : Integer total_clientes() : Integer get_gimnasios(\$id : Integer = FALSE) : array get_administradores(\$id : Integer = FALSE) : array get_tipo_deporte(\$idDeporte : Integer = FALSE) : array get_clientes() : array get_deporte(\$idPaciente : Integer) : array get_descripcion_ejercicio() : array get_entrenamiento(\$id : Integer) : array get_sesion(\$id : Integer) : array count_ejercicios(\$id : Integer) : Integer get_ejercicios(\$idSesion : Integer) : array get_nombre_ejercicio(\$idEjercicio : Integer) : array get_gimnasio_cliente(\$id : Integer) : array get_articulacion(\$id : Integer = FALSE) : array get_articulacion(\$id : Integer = FALSE) : String get_entrenamiento_articulacion(\$idArticulacion : Integer) : array autenticado() </code>

Ilustración 0-10 Dashboard sudo model

Se encarga de gestionar todos los datos necesarios para mostrar al dashboard del administrador, junto con su correspondiente controlador se puede decir que son quizás las clases mas complejas, debidas al grán volumen de iformación que manejan. Empezamos a comentar los metodos que contienen:

- Los primeros ocho métodos al igual que “count_ejercicios” se encargan de contar el número de filas que contiene diversas tablas, esas tablas se pueden deducir leyendo el nombre del método.
- Todos los métodos que contienen inicialmente la palabra “_get” se pueden clasificar de la siguiente manera.

- Cuando no recibe ningún parametro devuelve todos los datos de una tabla.
- Si recibe un identificador busca todos los datos correspondientes a el y a su vez estos métodos se pueden dividir en dos:
 - Si devuelve retorna un String devuelve el campo nombre.
 - En caso de que retorne un array será con todos los datos referentes a ese id.

Dashboard_gimnasio_model

Dashboard_gimnasio_model
<pre>monitores_gimnasio(\$id : Integer) : array clientes_gimnasio(\$id : integer) : array total_monitores(\$id_gimnasio : Integer) : Integer total_clientes(\$id_gimnasio : Integer) : Integer</pre>

Ilustración 0-11 Dashboard gimnasio model

Contiene los metodos necesarios para recopilar los datos que se van a mostrar al dashboard del gimnasio. Estos unicamente corresponden al numero de monitores y clientes y una lista con todos los que no han sido leidos todavia, es decir el campo de la tabla leido este a 0.

Dashboard_monitor_model

Dashboard_monitor_model
<pre>id_de_gimnasio(\$id : Integer) : Integer clientes_paginados_monitor(\$id_monitor : Integer,\$fk_gimnasio : Integer) : array get_tablas_caducadas(\$id_cliente : Integer) : array</pre>

Ilustración 0-12 Dashboard monitor model

Solicita a la BD los datos necesarios para el dashboard monitor. Estos métodos se encargan de:

- **Id_de_gimnasio:** Dado el id de un monitor retornamos el gimnasio al que pertenece.
- **Clientes_paginados_monitor:** Devuelve los clientes de un determinado gimnasio.

- **Get_tablas_caducadas:** Comprueba todas las tablas cuyas fechas están vencidas en la base de datos.

Administrador_model

Encargada únicamente de suministrar la información, que necesite el controlador Administrador

Administrador_model
<pre> crear(\$data : array) modificar(\$id : Integer,\$data : array) get_administradores(\$id : Integer = FALSE) : array administradores_paginados(\$filas : Integer,\$inicio : Integer) : array filas_administradores() : Integer get_id(\$dni : String) : Integer search(\$cadena : String) : array insertar_tipo(\$data : array) modificar_tipo(\$id : Integer,\$data : array) : Boolean get_tipos(\$id : Integer = False) : array eliminar_tipo(\$id : Integer) </pre>

Ilustración 0-13 Administrador model

- **Crear:** Recibe un vector con todos los datos de un administrador y lo crear en la base de datos.
- **Modificar:** Realiza cambios en los datos del administrador, cuyo identificador coincide con el recibido.
- **Get_administradores:** Se encarga de dos funciones. Estas son:
 - Si recibe un id, devuelve un vector con toda la información de un administrador.
 - En caso de no recibir ningún parámetro, devuelve un vector con todos los administradores existentes en la base de datos.
- **Administradores_paginados:** Devuelve los administradores de la base de datos, de manera paginada.
- **Filas_administradores:** Retorna el número de administradores existentes en la base de datos.
- **Get_id:** Dado el DNI de un administrador, devuelve el identificador que se le asigne en la base de datos.

- **Search:** Encargado de buscar coincidencias en el nombre y apellidos de los administradores.
- **Insert_tipo:** Dado un vector con los datos de un nuevo tipo de administrador, se encarga de guardarlo en la base de datos.
- **Modificar_tipo:** Modifica un tipo de administrador, para ello recibe los datos y el identificador del tipo de administrador a modificar.
- **Get_tipos:** Este método cumple dos funciones, estas son:
 - Si recibe un identificador, busca el tipo de administrador al que pertenece y retorna sus datos.
 - Si no recibe ningún identificador, devuelve todos los tipos de administradores existentes en la base de datos.
- **Eliminar_tipo:** Elimina el tipo de administrador al cual corresponde el identificador recibido.

General_model:

Esta clase se encarga de tareas que hacen varios controladores, de ahí el nombre de general.

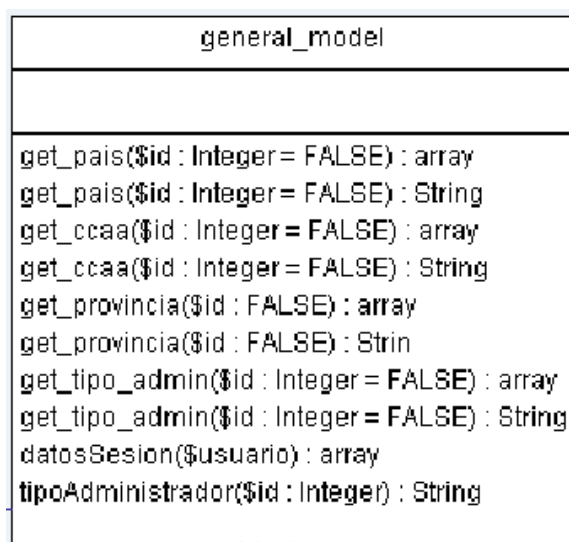


Ilustración 0-14 General model

Los ocho primeros métodos se encargan de dos tareas, estos están duplicados ya que se así se hace el polimorfismo en PHP. Los datos serán buscados en tabla de la BD cuyo nombre coincide con la palabra que hay después de “get_”

- Si reciben un identificador devuelve su nombre, el identificador será buscado en la tabla de la BD
- Si no reciben ningún dato devuelven todos los datos de la tabla de la BD.
- **tipoAdministrador:** Dado el id de un administrador, buscar el tipo de administrador al cual pertenece dicho administrador.
- **datosSesion:** Retorna varios del administrador, relevantes para iniciar una sesión.

Deporte_model:

Deporte_model
tipos_deportes_paginados (\$filas : Integer,\$inicio : Integer) : array filas_tipos_deportes () : Integer crear_tipo (\$data : array) : Boolean modificar_tipo (id,data) eliminar_tipo (\$id : Integer) : Boolean campo_tipo_deporte (\$id : Integer = FALSE) : Boolean get_tipo_deporte (\$id : Integer = FALSE) : String

Ilustración 0-15 Deporte model

Encargado de la comunicación con la tabla “tipo_de_deporte”. Los metodos que contiene son los siguientes:

- **tipos_deportes_paginados:** Recibe el número de filas a devolver y el número de fila a través de la cual ha de empezarse a buscar.
- **filas_tipos_deportes:** Retorna el total de filas que tiene la tabla.
- **Crear_tipo:** Crear un nuevo tipo de deporte.
- **Eliminar_tipo:** Dado un identificador elimina esa fila de la BD.
- **Campo_tipo_deporte:** Dad un identificador devuelve todos sus datos.
- **get_tipo_deporte:** Devuelve el nombre de un de un deporte.

Musculo_model

Musculo_model
<pre> musculos_paginados(\$filas : Integer,\$inicio : Integer) : array filas_musculos() : Integer get_musculo(\$id : Integer = FALSE) : array crear(\$data : array) : Boolean modificar(id,data) : Boolean eliminar(\$id : Integer) : Boolean </pre>

Ilustración 0-16 Músculo model

Esta clase se encarga de gestionar la tabla articulación. Como ya se ha comentado en la descripción de la BD. Esta inicialmente no tendrá articulaciones si no grupos musculares. Todos los métodos encargados de la gestión son iguales a la gestión de los deportes. Contiene unicamente los métodos encargados de listar, crear, modificar y eliminar los datos.

Ejercicio_model

Ejercicio_model
<pre> get_ejercicio(\$id : Integer = FALSE) : array get_articulaciones(\$id : Integer = FALSE) : array get_articulaciones(\$id : Integer = FALSE) : String ejercicios_paginados(\$filas : Integer,\$inicio : Integer) : array filas_ejercicios() : Integer ejercicio_por_id(\$id : Integer) : array crear(\$data : array) : array modificar(id,data) : Boolean eliminar(\$id : Integer) : Boolean </pre>

Ilustración 0-17 Ejercicio model

Esta clase gestiona la tabla “descripcion_ejercicio” . Al igual que ocurre en la clase anterior hace el mismo tipo de gestión.

Mundo_model

Esta clase es la encargada de la comunicación entre la base de datos y el controlador “Mundo”. Gestiona los datos de los Países, comunidades autónomas y provincias.

Mundo_model
<pre> get_paises(\$id : Integer = FALSE) : array get_paises(\$id : Integer = FALSE) : String get_pais(\$id : Integer = FALSE) : array crear_pais(\$data : array) modificar_pais(\$id : Integer,\$data : array) : Boolean eliminar_pais(\$id : Integer) : Boolean get_comunidades(\$id : Integer = FALSE) : array get_comunidades(\$id : Integer = FALSE) : String get_comunidad(\$id : Integer) : array crear_comunidad(\$data : array) modificar_comunidad(id,data) eliminar_comunidad(\$id : Integer) : Boolean get_provincias(\$id : Integer = FALSE) : array get_provincias(\$id : Integer = FALSE) : String get_provincia(\$id : Integer) : array crear_provincia(\$data : array) modificar_provincia(\$id : Integer,\$data : array) : Boolean eliminar_provincia(\$id : Integer) : Boolean id_pais(\$id : Integer = FALSE) : Integer </pre>

Ilustración 0-18 Mundo model

Podemos agrupar los quince primeros métodos en grupos de cinco. Estos grupos pertenecen a, país, comunidad y provincia. Dado que todos los métodos hacen lo mismo solo se va a detallar los cinco primeros.

- **Get_pais:** Método encargado de dos tareas:
 - Si recibe el identificador de un país, devolverá su nombre.
 - Si no recibe ningún id. Retornará todos los países existentes en la BD.
- **get_pais:** A través del identificador de un país, devolvemos todos sus datos.
- **Crear_pais:** Recibido un vector con los datos de un país, lo almacenamos en la BD.
- **Modificar_pais:** Modificamos los datos de un país cuyo identificador corresponda al recibido.

Eliminar_pais: Dado un identificador eliminamos el país al que pertenece.

Tabla_model

Encargada de la comunicación con la base de datos para la gestión de tablas.

tabla_model
newAttr: Integer
get_entrenamiento_nombre(\$nombre : String = FALSE) is_calendar(\$id_entrenamiento : Integer) modificar_entrenamiento(\$id : Integer,\$data : array) get_articulacion(\$id : Integer = FALSE) get_tipo_ejercicio(\$id : Integer = FALSE) crear_entrenamiento(\$data : array) crear_sesion(\$data : array) ver_tablas(\$id_cliente : Integer) get_entrenamiento(\$id_cliente : Integer,\$nombre : String) nombre_grupo(\$id : Integer) get_sesion(\$id_entrenamiento : Integer,\$dia : String) get_ejercicios(\$id_sesion : Integer) get_descripciones(\$id_descripcion_ejercicio : Integer) get_imagen_gimnasio(\$id_monitor : Integer) get_logo_gimnasio(\$id : Integer) nombre_monitor_cliente(\$id_cliente : Integer,\$id_gimnasio : Integer) nombre_cliente(\$id : Integer) nombre_monitor(\$id : Integer) get_numeros_sesion(\$id_entrenamiento : Integer)

Ilustración 0-19 Tabla model

- **get_entrenamiento_nombre:** Recibe el nombre del entrenamiento que a consultar.
- **is_calendar:** Dado el identificador de un entrenamiento devuelve si es una tabla de tipo calendario.
- **modificar_entrenamiento:** Modifica un entrenamiento ya creado.
- **get_articulacion:** Este método hará dos consultas:
 - Si recibe un id devolverá su fila.
 - En caso de que no reciba ningún parámetro devolverá todos los campos.
- **get_tipo_ejercicio:** Dado el identificador de un ejercicio devolvemos su descripción.
- **crear_entrenamiento:** Crear un entrenamiento y devolvemos su identificador.
- **crear_sesion:** Crea una nueva sesión y se retorna su identificador.

- **crear_ejercicio:** Creamos un nuevo ejercicio.
- **ver_tablas:** Veremos todas las tablas para un determinado cliente.
- **get_entrenamiento:** dado un identificador y un nombre dovelemos todas sus coincidencias.
- **nombre_grupo:** Recibe el nombre de un grupo muscular y nos retorna su nombre.
- **get_sesion:** Dado del id de un entrenamiento y su día nos devuelve sus sesiones.
- **get_imagen_gimnasio:** Cuando reciba el id de un monitor devolverá el logotipo del gimnasio en el que trabaja.
- **Get_logo_gimnasio:** Dado id de un gimnasio devuelve su logotipo.
- **nombre_monitor_cliente:** Recibe el id de un cliente y el de un gimnasio y devuelve el nombre del monitor.
- **nombre_cliente:** Retorna el nombre de un monitor.
- **get_numeros_sesion:** Devuelve todas las sesiones de un entrenamiento.