



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO EN INFORMÁTICA

Título del proyecto:

“Desarrollo de un módulo para la gestión deportiva de un gimnasio dentro de la aplicación gestora de complejos deportivos Nubapp”

Fco Javier Luquin Oroz

Jesús Villadangos Alonso

Pamplona, 5 de Septiembre de 2012

Agradecimientos

Me gustaría agradecer a todas las personas que de una manera u otra me han apoyado durante este último año de carrera y en la realización de este proyecto. Por compartir mis dudas, preocupaciones, alegrías y sobre todo el día a día.

A mis padres, por aguantarme en algunos días difíciles, por estar siempre ahí, por darme la oportunidad de estudiar una carrera universitaria y por creer en mí.

A mis compañeros de clase, con los que he compartido tantos momentos durante la carrera, por escucharme con los problemas del proyecto y por las alegrías vividas juntos.

A mis compañeros del departamento de proyectistas, donde a lo largo de este año he ido realizando el proyecto, por trabajar juntos, por interesarse en mis dudas y por el buen ambiente de trabajo que teníamos.

A los profesores de la carrera, por hacer de mi quien soy, por esas asignaturas duras de las que tanto me he quejado y por las que tanto he aprendido.

A Jesús Villadangos por aceptar ser mi tutor en este proyecto y orientarme cuando era necesario.

A Leyre, por aguantarme durante este último año de carrera, por ser comprensiva con mi carga de trabajo y apoyarme a su manera.

A Nubapp, y en especial medida a Sergio Fernández y Víctor Acín por darme la oportunidad de realizar el proyecto en su empresa. Por resolver mis dudas cuando las tenía, por mostrarse siempre pacientes y por dedicar parte de su escaso tiempo libre a orientarme y aconsejarme.

A Alberto Imízcoz y Joseba Campos, por prestarse desinteresadamente a leer la memoria, sugerirme cambios léxicos y señalarme faltas ortográficas.

Índice

1 Introducción	6
1.1 Nubapp	6
1.2 Aplicación nitro de Nubapp	6
2 Estudio inicial del sistema de información	7
2.1 Objetivo de la aplicación	7
2.2 Identificación inicial de funcionalidades	7
2.3 Magnitudes o características principales de la aplicación nitro	8
2.4 Estructura de la aplicación y etapas de desarrollo planificadas	9
2.5 Recursos utilizados para la realización del PFC	10
3 Análisis del sistema de información	11
3.1 Definición del sistema	11
3.1.1 Determinación del alcance del sistema	11
3.1.2 Identificación del entorno tecnológico	12
3.1.3 Especificación de estándares y normas	12
3.2 Establecimientos de requisitos	12
3.2.1 Obtención de requisitos	12
3.3 Especificación de casos de uso	15
3.3.1 Sección rutinas	18
3.3.2 Sección ejercicios	21
3.3.3 Sección historial	23
3.3.4 Sección trabajo pendiente	24
3.3.5 Sección citas	26
3.3.6 Mensajes	27
3.3.7 Settings/Configuración	29
3.3.8 Usuarios	29
3.4 Análisis de requisitos	30
3.5 Definición de las interfaces de usuario	31
3.5.1 Principios generales de la interfaz	31
3.5.2 Sección rutinas	32
3.5.3 Sección ejercicios	35
3.5.4 Sección trabajo pendiente	37
3.5.5 Sección historial	39
3.5.6 Settings	40
3.5.7 Mensajes	40

3.5.8 Front-end (usuarios)	41
4 Diseño del sistema de información	46
4.1 Definición de la arquitectura del sistema	46
4.2 Diseño de clases	48
4.3 Diseño físico de datos	54
4.4 Diseño de la migración y carga inicial de datos	57
4.5 Diseño del plan de pruebas	58
5 Conclusiones	62
6 Líneas futuras	63
Anexo I: Páginas web dinámicas	64
Anexo II: jQuery y Ajax	68
Bibliografía	70

1. Introducción

1.1 Nubapp

Nubapp es una start-up radicada en Londres. Surge con la colaboración entre Sergio Fernández y su antiguo jefe, quien se encarga de la financiación.

El objetivo de la empresa es el desarrollo, mantenimiento y comercialización de productos y aplicaciones cloudcomputing.

1.2 Aplicación Nitro de Nubapp

Nitro es una aplicación de servicios en la nube desarrollada por Nubapp para la gestión de complejos deportivos de alto nivel. Consta de una serie de módulos con diferentes funcionalidades tales como reservas online y gestión de reservas, gestión de usuarios, gestión de acceso, pagos de usuarios, aplicación móvil personalizada, gestión de actividades y de las instalaciones, gestión de la domótica, estadísticas de uso de instalaciones, gestión de personal, programa de gestión del gimnasio por parte del monitor, gestión de torneos entre usuarios, etc.... Todo ello desde un enfoque social y basado en la web.

De esta manera, se provee al cliente de una interfaz web desde donde poder gestionar su complejo de manera online y segura, y de una página front-end desde donde los usuarios del complejo pueden efectuar reservas, pagos o comunicarse entre ellos. Además, también dispone de una aplicación móvil para Android e iOS desde la cual los usuarios pueden realizar las tareas explicadas anteriormente.

Actualmente, dado que hay muchos módulos en fase de desarrollo, se comercializa el servicio de reservas online, vía Facebook, front-end, App móvil o SMS bajo el nombre <http://pro.bookpadel.co.uk/>, orientado al creciente mundo del pádel en Inglaterra.

2. Estudio inicial del Sistema de Información

2.2. Objetivo de la aplicación

El objetivo principal del presente trabajo fin de carrera es el desarrollo del módulo de gestión deportiva de un gimnasio, como parte del proyecto de aplicación modular en la nube Nitro.

2.2. Identificación inicial de funcionalidades

Las funcionalidades básicas del módulo de gestión deportiva del gimnasio que se van a desarrollar son:

Desde el punto de vista de la organización:

La parte de gestión propia de un entrenador personal o monitor. Gestión de citas con los usuarios. Creación de tablas y rutinas de entrenamiento tanto genéricas como personalizadas y asignarlas a los usuarios, creación y modificación de nuevos ejercicios que podrán ser utilizados para las tablas de entrenamiento, edición de rutinas de entrenamiento ya creadas, ver el historial de los entrenamientos seguidos por parte del usuario. Responder vía mensaje a las posibles dudas que puedan surgir a los usuarios, etc.

De manera más resumida y esquemática:

- ✓ Gestión de citas y trabajo pendiente.
- ✓ Gestión de ejercicios.
- ✓ Gestión de usuarios.
- ✓ Gestión de rutinas.
- ✓ Gestión del historial de asignaciones (entrenamiento seguido por los usuarios).
- ✓ Gestión de la comunicación con los usuarios.

Desde el punto de vista del usuario de la instalación:

Al ser una aplicación en la nube, la aplicación provee de acceso tanto a la administración del club e instalaciones de cara a la gestión en general como a los usuarios de dicho club.

Desde un navegador web y tras loguearse con sus datos de usuario, se podrán efectuar reservas de instalaciones o por ejemplo crear torneos. La parte correspondiente al módulo gimnasio a implementar es la siguiente:

- ✓ Ver la rutina de entrenamiento actual, guardarla como *pdf* o imprimirla, si así lo deseamos.
- ✓ Mandar un mensaje al monitor, comunicándole dudas, pidiendo una cita o una nueva tabla de entrenamiento.
- ✓ Ver los mensajes que nos haya mandado el monitor.
- ✓ Ver información sobre los ejercicios de nuestro plan de entrenamiento.

Desde el punto de vista de la aplicación móvil:

Como parte del servicio que ofrece Nitro con su aplicación para smartphones:

- ✓ Se deberá crear el API correspondiente para que la aplicación pueda acceder a las características propias del módulo.
- ✓ Se diseñará la interfaz correspondiente al módulo gimnasio, el objetivo primordial es el uso del Smartphone para la consulta del entrenamiento a seguir, marcando los ejercicios ya realizados o accediendo a más información sobre los mismos.

2.3. Magnitudes o características principales de la aplicación:

- ✓ Es una aplicación modular. Se utilizan tanto clases comunes para todos los módulos como clases propias de cada módulo. Debido a las primeras es necesario el uso de una herramienta de control de versiones. Evitándose así posibles conflictos y posibilitando que los desarrolladores programen siempre en la última versión global y funcional. Para el control de versiones se utilizara el repositorio online de pago Github, que utiliza el sistema de control de versiones Git.
- ✓ Es una aplicación multidioma. Se dispondrá de archivos diccionario, todas y cada una de las variables texto que mostraremos deberán corresponderse a una de las variables diccionario de nuestro modulo, dicha variable será la que contenga el texto. En función del idioma de la aplicación, se cargarán unos diccionarios u otros. En concreto, durante el desarrollo de este proyecto deberemos realizar los archivos diccionario correspondientes a Inglés y Español.
- ✓ El código se desarrollará en una versión de repositorios demo con servidor propio, una vez implementada una funcionalidad completa se probará en una versión test almacenada en otro servidor. Una vez testada se podrá añadir a la aplicación comercial, alojada en nubapp.com.
- ✓ Es una aplicación en la nube pero trata de ser interactiva como una aplicación gestora corriendo de manera nativa. Es por ello que la inserción y consulta de

datos deberá hacerse de una manera dinámica por parte del navegador cliente, es decir, sin salir de la página con el objetivo de que quien la utilice no vea ninguna diferencia práctica frente a otros programas (sin navegar de una página a otra ni recargando las mismas). Para ello, se utilizara la tecnología RIA (Rich internet applications), AJAX y jQuery.

- ✓ En relación con el punto anterior, el código generado deberá estar optimizado al máximo, con el objetivo de que el trabajo con la aplicación sea fluido.

2.4. Estructura de la aplicación y etapas de desarrollo planificadas

Para la consecución de los objetivos de este proyecto dividiremos su desarrollo en 4 etapas:

Etapa Inicial:

- ✓ Análisis y diseño global de la aplicación.

Etapa 1: Back - END

- ✓ Diseño del Back - END
- ✓ Gestión de Ejercicios.
- ✓ Gestión de Rutinas.
- ✓ Gestión de usuarios.
- ✓ Gestión de trabajo (rutinas) pendiente.
- ✓ Gestión del historial.
- ✓ Gestión de la comunicación con los usuarios.
- ✓ Gestión de citas.
- ✓ Testeo y pruebas

Etapa 2: Front – END

- ✓ Diseño del front - END
- ✓ Consulta y gestión de la rutina de entrenamiento actual.
- ✓ Gestión de la comunicación con el monitor.
- ✓ Acceso y documentación de los ejercicios de nuestro plan de entrenamiento.
- ✓ Testeo y pruebas.

Etapa 3: Aplicación Móvil

- ✓ Diseño de la aplicación móvil.
- ✓ Creación del API móvil.
- ✓ Testeo y pruebas.

Etapa 4:

- ✓ Testeo y pruebas.
- ✓ Resolución de bugs derivados del uso por parte de clientes.

2.5 Recursos utilizados para la realización del PFC

Para la realización de este proyecto necesitaremos un ordenador con acceso a internet. Los programas utilizados para el desarrollo de la aplicación han sido:

- ✓ Notepad ++ y DreamWeaver, para la edición de código PHP, HTML, SQL, Ajax y JQuery [2][3][4][5].
- ✓ MySQLWorkBench, para la gestión de la BBDD [6].
- ✓ Git-Bash y Tortoise, para el control de versiones y gestión del repositorio común online [10][11].
- ✓ Xampp, como servidor web para el desarrollo local de la aplicación [1][9].
- ✓ Navegadores web Chrome y Firefox, con sus correspondientes consolas de código.
- ✓ Tanto el estudio inicial del PFC, como la planificación, el análisis y el diseño del sistema de información han sido elaborados basándose en la metodología Métrica v3 [8].

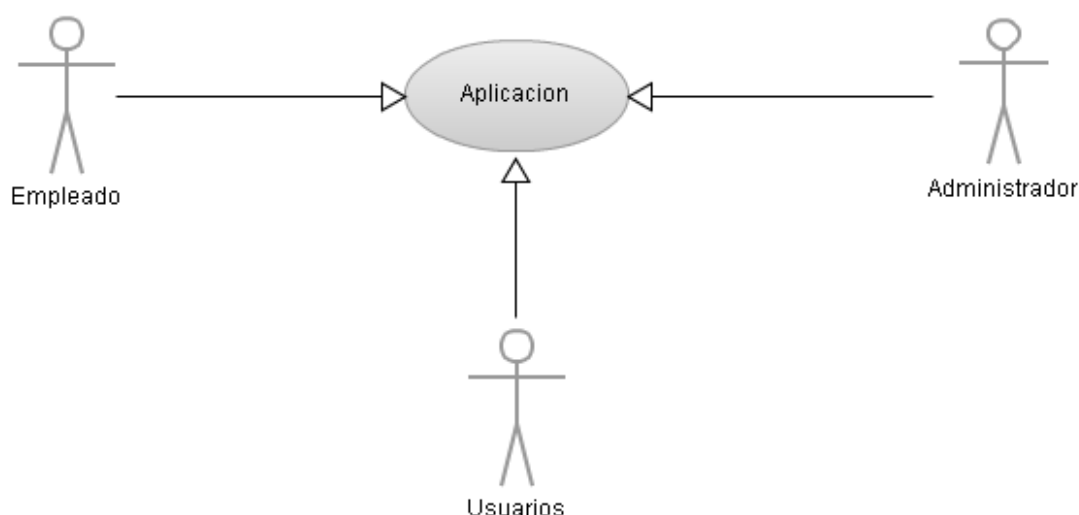
3. Análisis del sistema de información

3.1. Definición del sistema

3.1.1. Determinación del alcance del sistema

El sistema afectará a todo el club deportivo. En esta sección describiremos de manera breve cual será el alcance del sistema y la influencia en los usuarios.

Como podemos ver en la imagen, las entidades que actuarán sobre nuestro sistema serán las siguientes:



Administradores: personas encargadas del mantenimiento y configuración de la aplicación.

Empleados: personal que utilizará la aplicación, en nuestro caso, el monitor o monitores del gimnasio/sala fitness.

Socios: Los socios utilizarán la aplicación para acceder a sus datos de entrenamiento y para comunicarse con el monitor.

3.1.2 Identificación del entorno tecnológico

La aplicación web está escrita en PHP y JavaScript, por lo tanto es imprescindible un servidor Apache con al menos la versión 5 de PHP. Como se trata de un lenguaje interpretado orientado a la creación de páginas web no presenta problemas de compatibilidad con las diferentes plataformas existentes en el mercado.

Es recomendable para el uso de la aplicación un equipo con sistema operativo Windows XP o superior y preferentemente los navegadores Safari, Firefox y Chrome pues el código HTML y CSS está optimizado para dichos navegadores.

En lo referente a la App móvil, será compatible con smartphones, Android e iOS.

3.1.3 Especificación de estándares y normas

Las funcionalidades que se van a implementar en este PFC vienen englobadas dentro de una aplicación general, dado que dicha aplicación va a almacenar información personal de los clientes, esta deberá ser tratada de manera que cumpla lo dispuesto en las leyes Británicas y Españolas, teniendo en cuenta especialmente las siguientes en el caso de España:

Ley Orgánica 15/1999 de Protección de Datos de Carácter Personal.

Real Decreto 994/1999 de reglamento de medidas de seguridad de los ficheros automatizados que contengan datos de carácter personal.

En lo dispuesto en tales documentos se entienden que los datos que va a almacenar el sistema son de nivel básico y como tales, deberán cumplir las medidas de seguridad requeridas para tal nivel en el reglamento de medidas de seguridad.

3.2 Establecimiento de requisitos

3.2.1 Obtención de requisitos

En esta sección se definen detalladamente los requisitos del sistema, a cada uno de ellos se le asociará un código de identificación y se establecerá su prioridad.

FUNCIONALIDADES			
ID	Descripción	Prioridad	Dificultad
R1	Ejecución de la aplicación	Alta	Alta
	La aplicación se ejecutará en cualquiera de los navegadores más utilizados. Cumplirá con las normas de estilo W3C.		

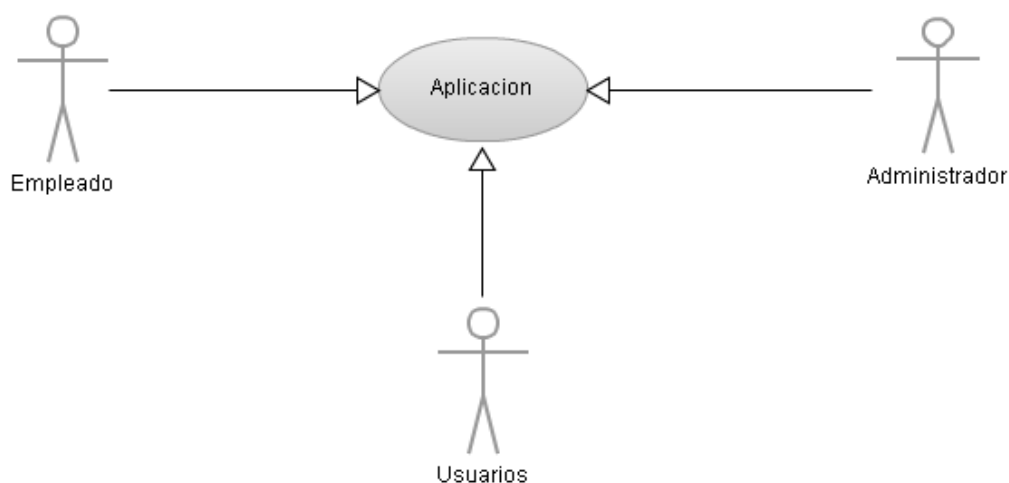
R2	Gimnasio -- Rutinas	Alta	Alta
	<p>Se creará una interfaz para ver, imprimir, crear, modificar, asignar y borrar rutinas. Las rutinas constarán de:</p> <ul style="list-style-type: none"> ✓ Título de la rutina ✓ Descripción de la misma ✓ Ejercicios de la rutina, se podrán añadir, editar y borrar diferentes ejercicios. ✓ Series. ✓ Repeticiones. ✓ Días, será posible realizar diferentes planes de entrenamiento para diferentes días dentro de una misma rutina. 		
R3	Gimnasio – Ejercicios	Alta	Media
	<p>Se creará una interfaz para ver, editar, crear y borrar ejercicios. De cada ejercicio podremos almacenar la siguiente información:</p> <ul style="list-style-type: none"> ✓ Nombre del ejercicio ✓ Grupo muscular ✓ Descripción del ejercicio ✓ Imagen ✓ Video 		
R4	Gimnasio – Trabajo pendiente	Alta	Alta
	<p>Se creará una interfaz para consultar, almacenar y controlar las peticiones de planes de entrenamiento por parte de los usuarios. Desde aquí el monitor podrá asignar directamente rutinas ya creadas a los usuarios. Se quiere almacenar la siguiente información:</p> <ul style="list-style-type: none"> ✓ Nombre del usuario ✓ Descripción 		
R5	Gimnasio – Historial	Alta	Media
	<p>Para poder documentarse correctamente y preparar un plan de entrenamiento correcto, el monitor podrá consultar el historial de entrenamientos realizados por los diferentes usuarios. Se guardaran los siguientes datos:</p> <ul style="list-style-type: none"> ✓ Nombre del usuario que realiza el entrenamiento. ✓ Título del entrenamiento asignado ✓ Fecha ✓ Hora 		
R6	Gimnasio -- Citas	Media	Alta
	<p>Se dispondrá de un apartado con calendario en el que el monitor podrá gestionar las diferentes citas con los usuarios. Se almacenarán los siguientes datos:</p> <ul style="list-style-type: none"> ✓ Nombre del usuario 		

	<ul style="list-style-type: none"> ✓ Comentarios ✓ Día ✓ Hora 		
R7	<p>Configuración de la aplicación</p> <p>Algunos de los parámetros de nuestra aplicación podrán ser configurados desde el apartado general de opciones que se encuentra en la aplicación. Por el momento, se podrá configurar:</p> <ul style="list-style-type: none"> ✓ Habilitar/Deshabilitar el acceso de los usuarios al apartado gimnasio del back-end. ✓ Habilitar/Deshabilitar el uso de ejercicios genéricos por parte de la aplicación. ✓ Habilitar/Deshabilitar el uso de rutinas genéricas por parte de la aplicación 	Alta	Media
R8	<p>Carga inicial de datos</p> <p>La aplicación dispondrá de entrenamientos y ejercicios predefinidos.</p>	Media	Media
R9	<p>Resultados</p> <p>Una vez finalizada una acción, tanto el monitor como el usuario obtendrán información sobre el resultado de dicha acción.</p>	Alta	Media
R10	<p>Logs</p> <p>La aplicación escribirá en el archivo log de la aplicación general mostrando el uso dado por los monitores a la aplicación.</p>	Baja	Media
R11	<p>Permisos</p> <p>Para el acceso al módulo gimnasio de la aplicación, el trabajador deberá tener permitido el uso del mismo.</p> <p>Un administrador asignará permisos a los monitores para que puedan acceder.</p>	Alta	Media
R12	<p>Diseño reutilizable y claridad de código</p> <p>El diseño ha de ser reutilizable y el código debe ser claro y estar debidamente comentado. Se creará un documento explicativo de las diferentes clases que interactúan con el módulo, así como de las que han sido creadas para el mismo módulo. Se intentará optimizar el código tanto como sea posible.</p>	Alta	Alta
R13	<p>Homogeneidad</p> <p>Las pantallas del sistema deberán ser homogéneas y seguir las normas de estilo y estructuración del resto de la aplicación.</p>	Alta	Media
R14	<p>Facilidad de uso</p> <p>El uso del sistema deberá ser fácil e intuitivo</p>	Alta	Alta
R15	<p>Comunicación usuario – monitor</p> <p>Los usuarios y los monitores podrán intercambiar mensajes entre sí para resolver dudas, pedir citas o entrenamientos, etc.</p>	Alta	Alta
R16	Front-end	Alta	Alta

	En la parte front-end de la aplicación se creará un apartado gimnasio desde donde los usuarios podrán ver, imprimir o guardar sus rutinas y contactar con el monitor.		
R17	API Se creara un API para la aplicación móvil que provea acceso a la información.	Alta	Media
R18	Aplicación móvil Se diseñará una aplicación móvil para que el usuario pueda ver su tabla desde su Smartphone. El objetivo principal de esta interfaz es su uso a la hora de entrenar, consultando el plan de entrenamiento, marcando los ejercicios realizados, accediendo a información de los mismos, etc.	Alta	Alta
R19	Dinamismo Al ser un servicio en la nube, la aplicación deberá diferenciar qué cliente o entidad está usando la misma y cargar solo la información de dicho cliente, mostrando solo los ejercicios, configuraciones y rutinas correspondientes al cliente en cuestión.	Alta	Media
R20	Diccionarios Todo el texto que muestre la aplicación serán variables ligadas a un archivo diccionario desde donde se cargara el texto en el idioma correspondiente.	Alta	Media

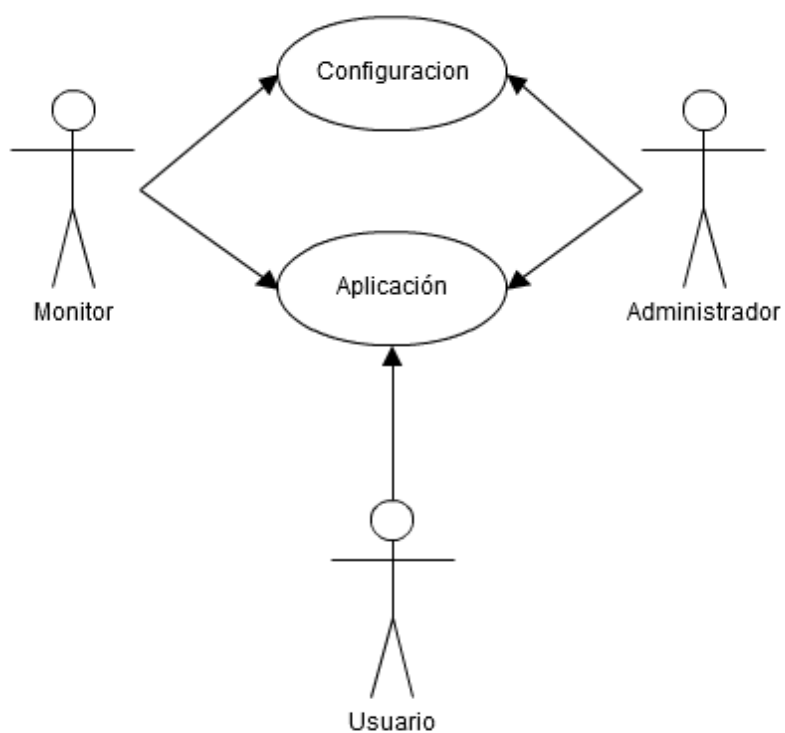
3.3 Especificación de casos de uso

Se ha realizado un análisis de los casos de uso a partir del siguiente diagrama de casos de uso general:

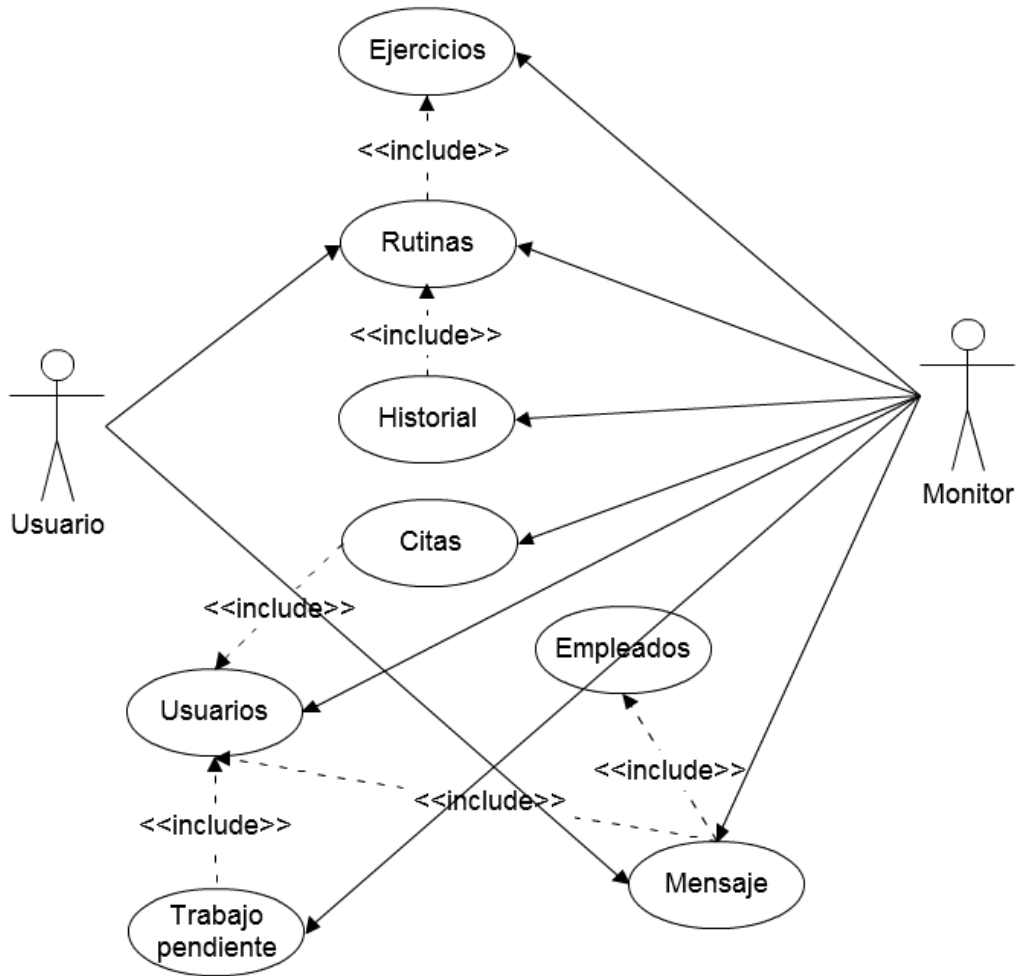


Obteniendo los siguientes diagramas al iterar:

Iteración 1:



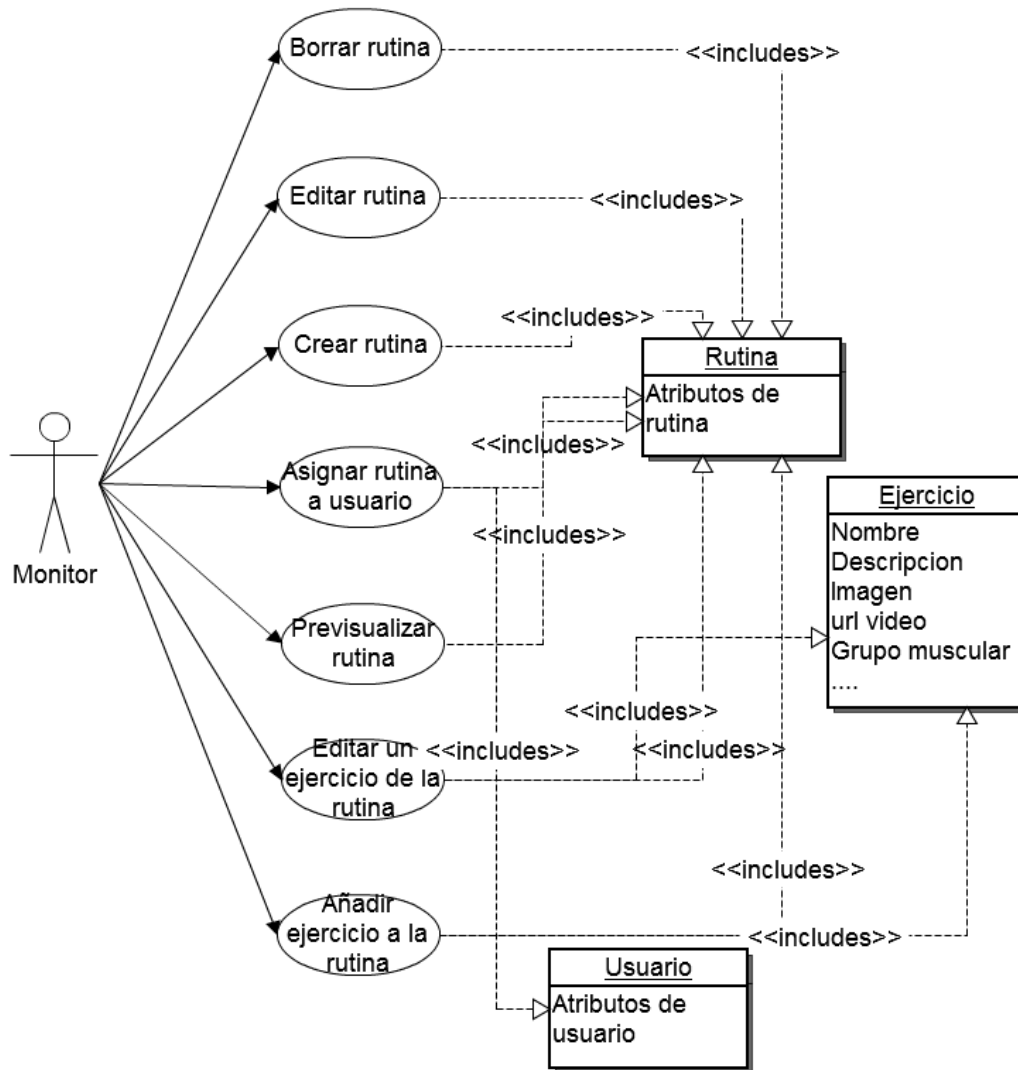
Iteración 2:



A continuación, y como parte de la iteración final, iremos mostrando los diferentes diagramas de casos de uso por secciones realizadas.

Con el objetivo de simplificar los diagramas, se ha omitido en los mismos la figura del administrador que aparecía en anteriores iteraciones. Simplemente sería repetir las relaciones del actor Monitor para el Administrador.

3.3.1. Sección Rutinas



RUTINAS		
ID	Información	
CU1	Descripción	Borrar rutina
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado. Hay rutinas que puedan ser borradas en el sistema
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	<ol style="list-style-type: none"> I. El administrador o monitor accede a la sección de rutinas del módulo gimnasio de la aplicación. II. Decide la rutina a borrar. III. Selecciona la opción borrar. IV. Se actualiza la información en la BBDD. V. El sistema informa del resultado de la ejecución (exitosa o fallo).

RUTINAS		
ID	Información	
CU2	Descripción	Editar rutina
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado. Hay rutinas que pueden ser editadas en el sistema
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	<ol style="list-style-type: none"> I. El administrador o monitor accede a la sección de rutinas del módulo gimnasio de la aplicación. II. Decide la rutina a editar. III. Edita los datos y los guarda. IV. Se actualiza la información en la BBDD. V. El sistema informa del resultado de la ejecución (exitosa o fallo).

RUTINAS		
ID	Información	
CU3	Descripción	Crear rutina
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado.
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	<ol style="list-style-type: none"> I. El administrador o monitor accede a la sección de rutinas del módulo gimnasio de la aplicación. II. Selecciona la opción añadir nueva rutina. III. Introduce los datos de la nueva rutina y le da a guardar. IV. Se actualiza la información en la BBDD. V. El sistema informa del resultado de la ejecución (exitosa o fallo).

RUTINAS		
ID	Información	
CU4	Descripción	Asignar rutina a usuario
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado.
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	<ol style="list-style-type: none"> I. El administrador o monitor accede a la sección de rutinas del módulo gimnasio de la aplicación. II. Selecciona la opción asignar de alguna rutina. III. Selecciona el usuario a quien la desea asignar. IV. Se actualiza la información en la BBDD. V. El sistema informa del resultado de la ejecución (exitosa o fallo).

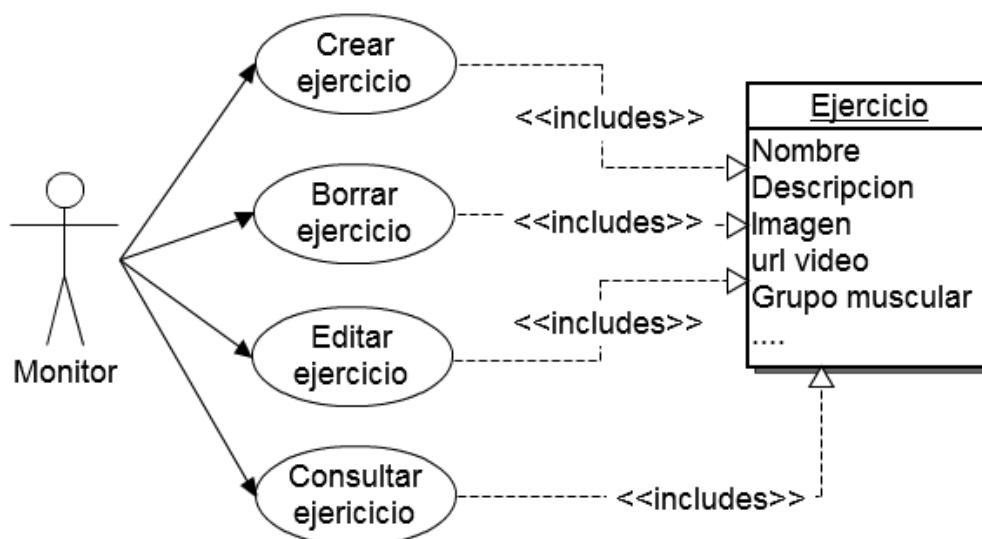
RUTINAS		
ID	Información	
CU5	Descripción	Ver rutina
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado. Hay rutinas creadas.
	Postcondición	El sistema muestra la rutina
	Flujo Básico	<ol style="list-style-type: none"> I. El administrador o monitor accede a la sección de rutinas del módulo gimnasio de la aplicación. II. Selecciona la opción ver rutina de una rutina. III. Se carga la información y se muestra por pantalla. IV. El sistema informa del resultado de la ejecución (exitosa o fallo).

RUTINAS		
ID	Información	
CU6	Descripción	Editar ejercicio
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado y hay alguna rutina creada.
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	<ol style="list-style-type: none"> I. El administrador o monitor accede a la sección de rutinas del módulo gimnasio de la aplicación. II. Selecciona la opción editar de alguna de las rutinas. III. Selecciona el ejercicio a editar, introduce los nuevos datos y clic en actualizar. IV. Se actualiza la información en la BBDD. V. El sistema informa del resultado de la ejecución (exitosa o fallo).

RUTINAS		
ID	Información	
CU7	Descripción	Insertar ejercicio
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado y hay alguna rutina creada.
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	<ol style="list-style-type: none"> I. El administrador o monitor accede a la sección de rutinas del módulo gimnasio de la aplicación. II. Selecciona la opción editar en alguna de las rutinas. III. Selecciona el día, ejercicio, series y repeticiones y clic en insertar. IV. Se actualiza la información en la BBDD. V. El sistema informa del resultado de la ejecución (exitosa o fallo).

RUTINAS		
ID	Información	
CU8	Descripción	Buscar rutina
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado.
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	I. El administrador o monitor accede a la sección rutinas del módulo gimnasio de la aplicación. II. Introduce el nombre de la rutina que desea buscar. III. Se muestran las rutinas cuyo título coincida con el parámetro de búsqueda. IV. El sistema informa del resultado de la ejecución (exitosa o fallo).

3.3.2. Sección Ejercicios



EJERCICIOS		
ID	Información	
CU9	Descripción	Crear ejercicio
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado.
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	I. El administrador o monitor accede a la sección de ejercicios del módulo gimnasio de la aplicación. II. Selecciona la opción crear ejercicio. III. Introduce los datos correspondientes y clicla en guardar.

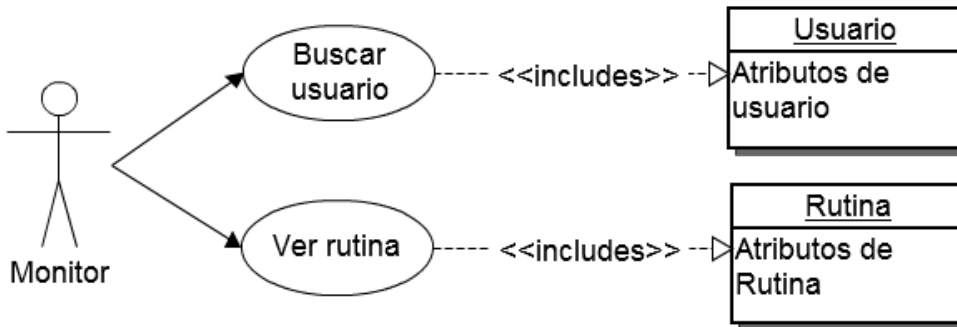
		IV. Se actualiza la información en la BBDD. V. El sistema informa del resultado de la ejecución (exitosa o fallo).
--	--	---

EJERCICIOS		
ID	Información	
CU10	Descripción	Editar ejercicio
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado y hay algún ejercicio creado.
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	<ol style="list-style-type: none"> I. El administrador o monitor accede a la sección de ejercicios del módulo gimnasio de la aplicación. II. Selecciona la opción editar de alguno de los ejercicios. III. Cambia los parámetros pertinentes y clics en guardar. IV. Se actualiza la información en la BBDD. V. El sistema informa del resultado de la ejecución (exitosa o fallo).

EJERCICIOS		
ID	Información	
CU11	Descripción	Borrar ejercicio
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado y hay algún ejercicio creado.
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	<ol style="list-style-type: none"> I. El administrador o monitor accede a la sección de ejercicios del módulo gimnasio de la aplicación. II. Selecciona la opción borrar de alguno de los ejercicios. III. Se actualiza la información en la BBDD. IV. El sistema informa del resultado de la ejecución (exitosa o fallo).

EJERCICIOS		
ID	Información	
CU12	Descripción	Ver ejercicio
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado y hay algún ejercicio creado.
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	<ol style="list-style-type: none"> I. El administrador o monitor accede a la sección de ejercicios del módulo gimnasio de la aplicación. II. Selecciona la opción ver información de alguno de los ejercicios. III. Se muestra la información detallada de dicho

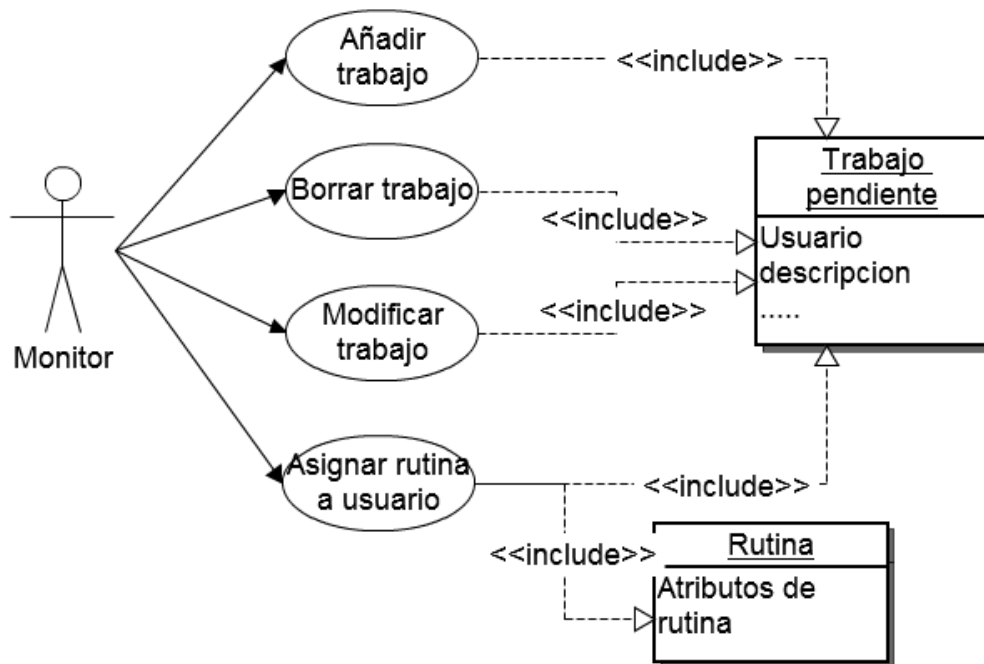
3.3.3. Sección Historial



HISTORIAL		
ID	Información	
CU13	Descripción	Buscar usuario
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado.
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	V. El administrador o monitor accede a la sección de Historial del módulo gimnasio de la aplicación. VI. Introduce el usuario que desea buscar. VII. Se muestran las rutinas asignadas a dicho usuario. VIII. El sistema informa del resultado de la ejecución (exitosa o fallo).

HISTORIAL		
ID	Información	
CU14	Descripción	Ver rutina
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado. Hay rutinas asignadas al usuario.
	Postcondición	El sistema muestra la rutina
	Flujo Básico	I. El administrador o monitor accede a la sección de historial del módulo gimnasio de la aplicación. II. (opcional) Realiza una búsqueda. III. Selecciona la opción ver rutina de una asignación. IV. Se carga la información y se muestra por pantalla. V. El sistema informa del resultado de la ejecución (exitosa o fallo).

3.3.4. Sección Trabajo pendiente



TRABAJO PENDIENTE		
ID	Información	
CU15	Descripción	Añadir trabajo
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado.
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	<ol style="list-style-type: none"> I. El administrador o monitor accede a la sección de trabajo pendiente del módulo gimnasio de la aplicación. II. Selecciona la opción crear trabajo pendiente. III. Introduce los datos del nuevo trabajo pendiente y clicla en guardar. IV. Se actualiza la información de la BBDD. V. El sistema informa del resultado de la ejecución (exitosa o fallo).

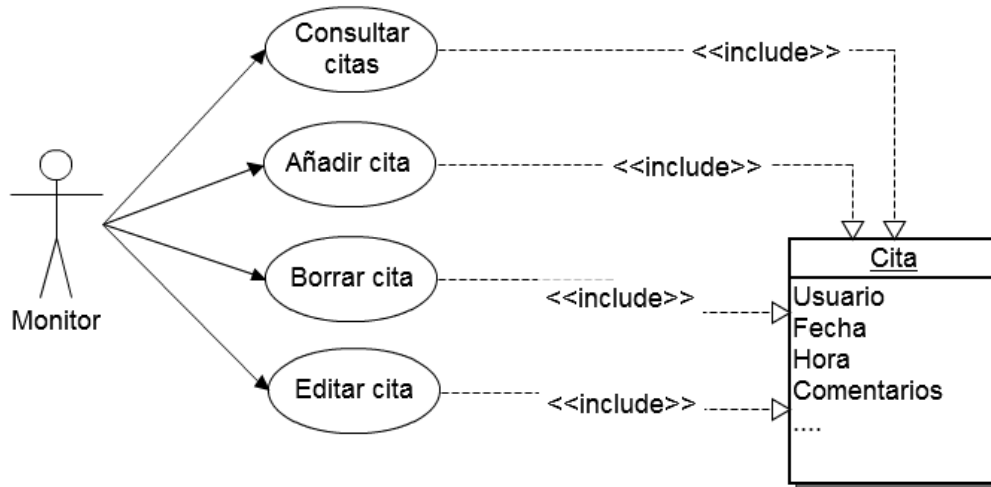
TRABAJO PENDIENTE		
ID	Información	
CU16	Descripción	Borrar trabajo
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado. Hay trabajos pendientes
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	<ol style="list-style-type: none"> I. El administrador o monitor accede a la sección de trabajo pendiente del módulo gimnasio de la

		<p>aplicación.</p> <p>II. Selecciona la opción borrar de alguno de los trabajos pendientes.</p> <p>III. Se actualiza la información de la BBDD.</p> <p>IV. El sistema informa del resultado de la ejecución (exitosa o fallo).</p>
--	--	--

TRABAJO PENDIENTE		
ID	Información	
CU17	Descripción	Editar trabajo
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado. Hay trabajos pendientes.
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	<p>I. El administrador o monitor accede a la sección de trabajo pendiente del módulo gimnasio de la aplicación.</p> <p>II. Selecciona la opción editar de algún trabajo pendiente.</p> <p>III. Introduce los nuevos datos y clic en guardar.</p> <p>IV. Se actualiza la información de la BBDD.</p> <p>V. El sistema informa del resultado de la ejecución (exitosa o fallo).</p>

TRABAJO PENDIENTE		
ID	Información	
CU18	Descripción	Asignar rutina a usuario
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado.
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	<p>I. El administrador o monitor accede a la sección de trabajo pendiente del módulo gimnasio de la aplicación.</p> <p>II. Selecciona la opción asignar rutina de algún trabajo pendiente.</p> <p>III. Selecciona la rutina a asignar.</p> <p>IV. Se actualiza la información de la BBDD.</p> <p>V. El sistema informa del resultado de la ejecución (exitosa o fallo).</p>

3.3.5. Sección Citas



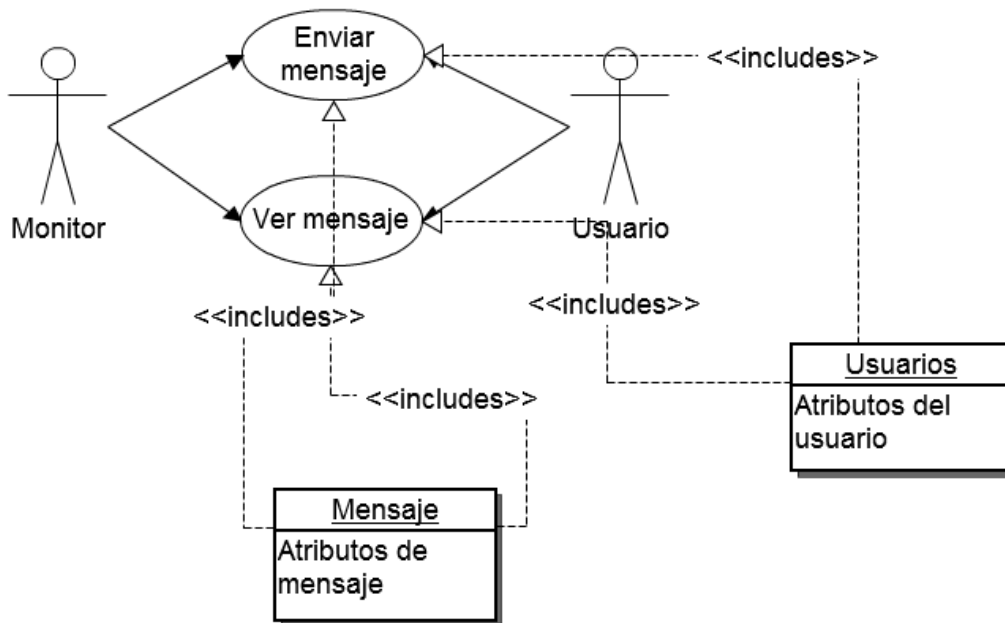
CITAS		
ID	Información	
CU19	Descripción	Consultar citas
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado.
	Postcondición	El sistema muestra las citas en el periodo deseado.
	Flujo Básico	I. El administrador o monitor accede a la sección calendario de la aplicación. II. Selecciona la franja en la cual desea consultar las citas. III. Se muestra la información de la BBDD. IV. El sistema informa del resultado de la ejecución (exitosa o fallo).

CITAS		
ID	Información	
CU20	Descripción	Añadir cita
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado.
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	I. El administrador o monitor accede a la sección calendario de la aplicación. II. Selecciona la opción nueva cita. III. Introduce los datos de la nueva cita y clic en guardar. IV. Se actualiza la información de la BBDD. V. El sistema informa del resultado de la ejecución (exitosa o fallo).

CITAS		
ID	Información	
CU21	Descripción	Editar cita
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado. Hay citas creadas.
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	<ol style="list-style-type: none"> I. El administrador o monitor accede a la sección calendario de la aplicación. II. Selecciona la opción editar de cita. III. Introduce los datos y clics en guardar. IV. Se actualiza la información de la BBDD. V. El sistema informa del resultado de la ejecución (exitosa o fallo).

CITAS		
ID	Información	
CU22	Descripción	Borrar cita
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado. Hay citas creadas.
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	<ol style="list-style-type: none"> I. El administrador o monitor accede a la sección calendario de la aplicación. II. Selecciona la opción borrar de una cita. III. Se actualiza la información de la BBDD. IV. El sistema informa del resultado de la ejecución (exitosa o fallo).

3.3.6. Mensajes



MENSAJES		
ID	Información	
CU23	Descripción	Ver mensaje
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios está autenticado. Hay mensajes.
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	I. El administrador o monitor accede a la sección mensajes de la aplicación. II. Selecciona el mensaje a leer. III. Se carga la información de la BBDD. IV. El sistema informa del resultado de la ejecución (exitosa o fallo).

MENSAJES		
ID	Información	
CU24	Descripción	Enviar mensaje
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios esta autenticado.
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	I. El administrador o monitor accede a la sección mensajes de la aplicación. II. Selecciona el mensaje al cual responder o selecciona la opción nuevo mensaje (en cuyo caso selecciona el destinatario). III. Introduce los datos del mensaje y le da a enviar.

		IV. Se actualiza la información de la BBDD. V. El sistema informa del resultado de la ejecución (exitosa o fallo).
--	--	---

3.3.7. Settings/Configuración

SETTINGS		
ID	Información	
CU25	Descripción	Configurar los diferentes parámetros del módulo.
	Actor	Administrador, empleado
	Precondición	El administrador o un empleado con privilegios esta autenticado.
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	I. El administrador o monitor accede a la sección settings de la aplicación. II. Selecciona la opción del módulo gimnasio. III. Configura algunos parámetros y clics en guardar. IV. Se actualiza la información de la BBDD. V. El sistema informa del resultado de la ejecución (exitosa o fallo).

3.3.8. Usuarios

Tanto en la sección rutinas como en la sección mensajes aparecía el actor usuario. A continuación mostramos los casos de uso correspondientes a dichos diagramas (consultar rutina, ver mensaje, enviar mensaje).

MENSAJES		
ID	Información	
CU26	Descripción	Enviar mensaje
	Actor	USUARIO
	Precondición	El usuario está autenticado.
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	I. El usuario accede a la sección mensajes del front-end. II. Selecciona el mensaje al cual responder o selecciona la opción nuevo mensaje (en cuyo caso selecciona el destinatario). III. Introduce los datos del mensaje y le da a enviar. IV. Se actualiza la información de la BBDD. V. El sistema informa del resultado de la ejecución (exitosa o fallo).

MENSAJES		
ID	Información	
CU27	Descripción	Ver mensaje
	Actor	USUARIO
	Precondición	El usuario esta autenticado. Hay algún mensaje
	Postcondición	El sistema se ha actualizado correctamente
	Flujo Básico	I. El usuario accede a la sección mensajes del front-end. II. Selecciona el mensaje a leer. III. Se carga la información de la BBDD. IV. El sistema informa del resultado de la ejecución (exitosa o fallo).

RUTINAS		
ID	Información	
CU28	Descripción	Ver rutina
	Actor	USUARIO
	Precondición	El usuario esta autenticado. El usuario tiene alguna rutina asignada.
	Postcondición	El sistema muestra la rutina
	Flujo Básico	I. El usuario accede a la sección mi rutina del front-end. II. Se carga la información y se muestra por pantalla. III. El sistema informa del resultado de la ejecución (exitosa o fallo).

3.4 Análisis de requisitos

En esta tarea se asocian los requisitos con los casos de uso que los implementan. Para ello se ha construido una matriz de trazabilidad entre los casos de uso de realización y los requisitos funcionales.

	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10	R11	R12	R13	R14	R15	R16	R17	R18	R19	R20
CU1	x	x						x	x	x	x	x	x	x					x	x
CU2	x	x						x	x	x	x	x	x	x					x	x
CU3	x	x						x	x	x	x	x	x	x					x	x
CU4	x	x						x	x	x	x	x	x	x					x	x
CU5	x	x						x	x	x	x	x	x	x					x	x
CU6	x	x						x	x	x	x	x	x	x					x	x
CU7	x	x						x	x	x	x	x	x	x					x	x
CU8	x	x						x	x	x	x	x	x	x					x	x
CU9	x		x					x	x	x	x	x	x	x					x	x
CU10	x		x					x	x	x	x	x	x	x					x	x
CU11	x		x					x	x	x	x	x	x	x					x	x
CU12	x		x					x	x	x	x	x	x	x					x	x
CU13	x				x				x	x	x	x	x	x					x	x

CU14	x		x		x	x	x	x	x					x	x
CU15	x		x		x	x	x	x	x					x	x
CU16	x		x		x	x	x	x	x					x	x
CU17	x		x		x	x	x	x	x					x	x
CU18	x		x		x	x	x	x	x					x	x
CU19	x		x		x	x	x	x	x					x	x
CU20	x		x		x	x	x	x	x					x	x
CU21	x		x		x	x	x	x	x					x	x
CU22	x		x		x	x	x	x	x					x	x
CU23	x				x	x	x	x	x	x				x	x
CU24	x				x	x	x	x	x	x				x	x
CU25	x		x		x	x	x	x	x					x	x
CU26	x				x	x	x	x	x			x	x	x	x
CU27	x				x	x	x	x	x			x	x	x	x
CU28	x				x	x	x	x	x			x	x	x	x

3.5 Definición de las interfaces de usuario

3.5.1. Principios generales de la interfaz

A continuación se describirán normas de diseño utilizadas.

- ✓ En la aplicación se utilizarán imágenes con formato .png, ya que mantienen el equilibrio entre compresión (tamaño) y calidad.
- ✓ Se utilizarán colores primarios y las combinaciones de los colores de fondo y primer plano tendrán suficiente contraste para que sean vistas por personas con déficit de percepción de color o por pantallas en blanco y negro.
- ✓ No se utilizarán imágenes de fondo que puedan causar distracción respecto del contenido.
- ✓ Los contenidos estarán estructurados facilitando la lectura y comprensión.
- ✓ Deberá existir una completa compatibilidad con todos los sistemas operativos ya que se trata de una aplicación web que cumple con los estándares y funciona en cualquier navegador.
- ✓ Se mantendrá un diseño lógico y uniforme.
- ✓ Los controles que no puedan utilizarse en cada pantalla no estarán presentes, o bien se mostrarán inactivos.
- ✓ Se seguirá una estructura de contenidos y diseño similar a la del resto de la aplicación.

A continuación pasaremos a detallar y explicar los diferentes aspectos de las interfaces implementadas.

3.5.2. Sección Rutinas

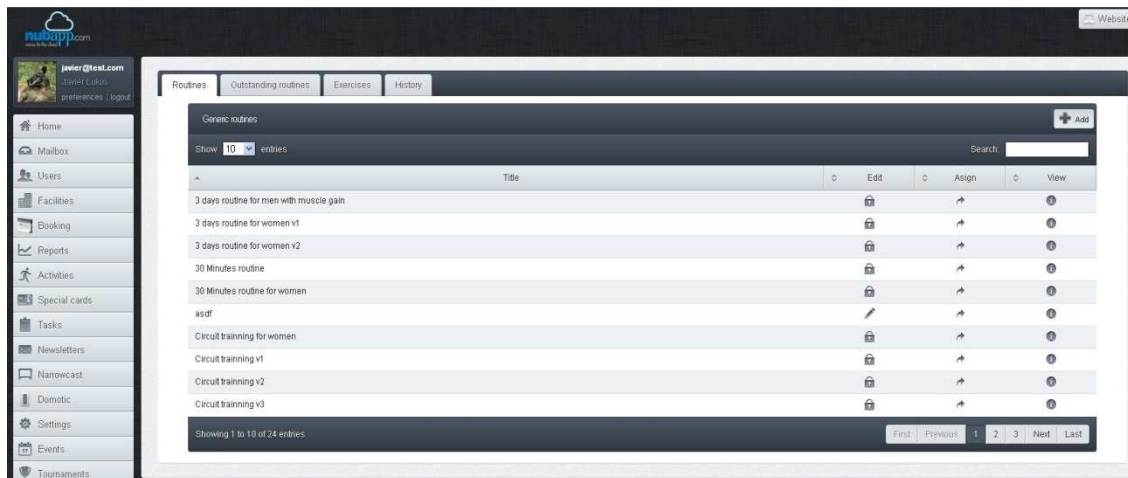


Ilustración 1 Sección principal

Al acceder al módulo gimnasio el sistema carga por defecto el apartado rutinas, desde donde podremos ver un listado de las rutinas creadas. Para cada una de las rutinas, podremos editarlas (si no son predefinidas), asignarlas a un usuario o verlas. Las rutinas están paginadas en una datatable por antigüedad, permitiendo al monitor buscar entre las mismas.

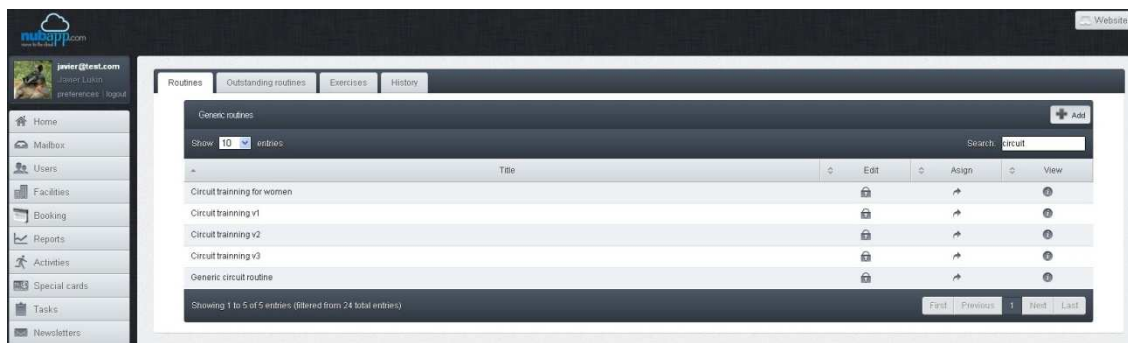


Ilustración 2 Buscar rutinas

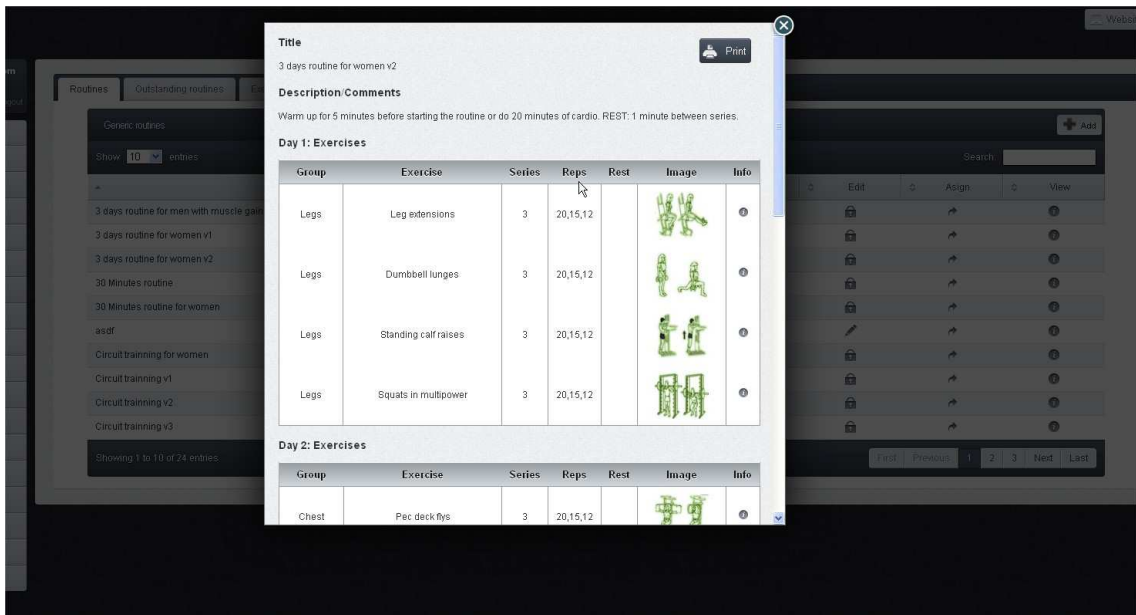


Ilustración 3 Ver rutina

Si deseamos asignar una rutina a un usuario, el sistema carga una tabla donde se listan todos los usuarios de la aplicación. De nuevo y con el objetivo de agilizar el trabajo del monitor, han sido estructurados en una datatable para que el monitor pueda efectuar búsquedas.

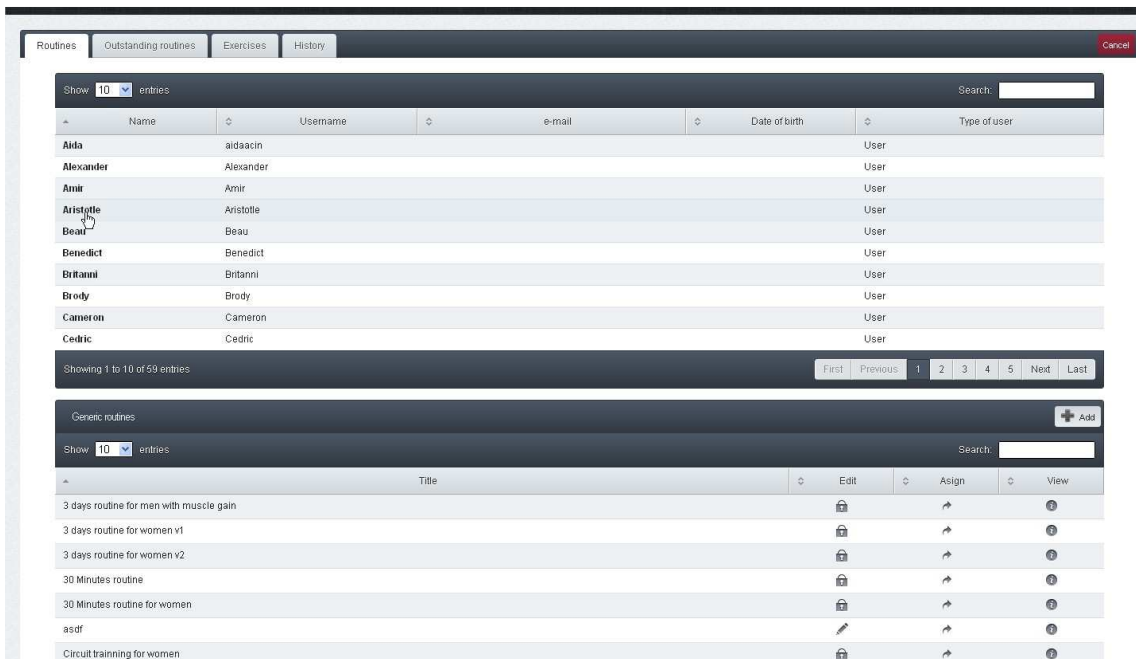


Ilustración 4 Asignar rutina a usuario

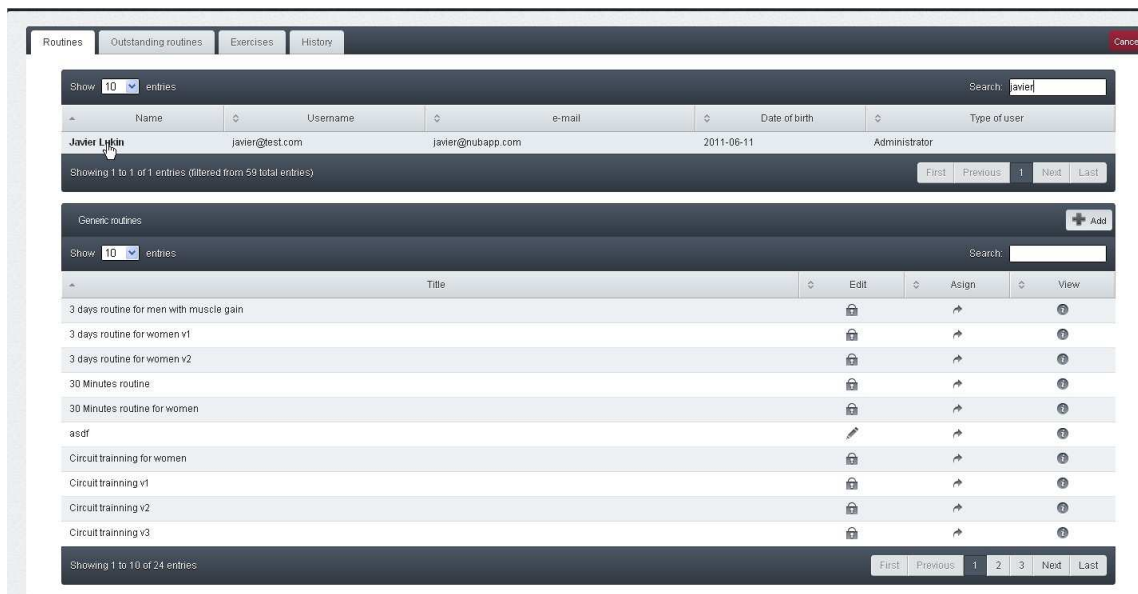


Ilustración 5 Asignar rutina a usuario buscado

Para crear una nueva rutina deberemos clicar sobre el botón Add. Cuando lo hagamos, se desplegará un formulario donde introduciremos los datos correspondientes a nuestra rutina y podremos añadir, editar o borrar ejercicios. También es posible añadir, editar o borrar ejercicios en rutinas ya creadas si clicamos en su correspondiente botón editar.

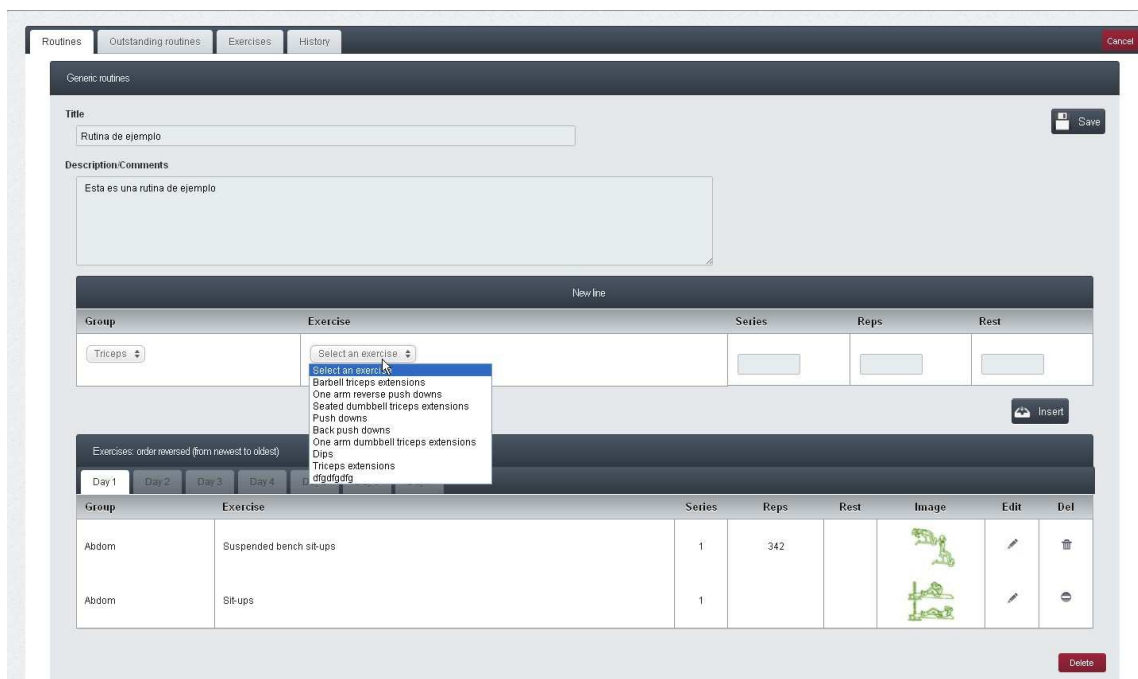


Ilustración 6 Nueva rutina

A la hora de añadir una línea, seleccionamos el ejercicio e introducimos las series, las repeticiones y el descanso. La línea se insertara en el día que se encuentre seleccionado en ese momento.

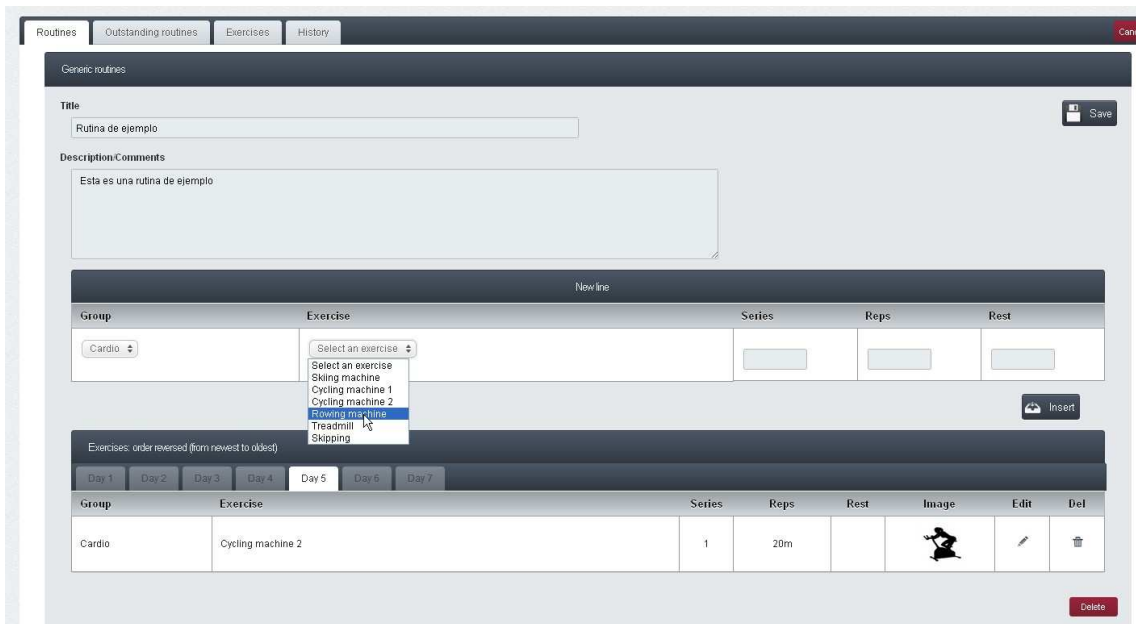


Ilustración 7 Insertar línea

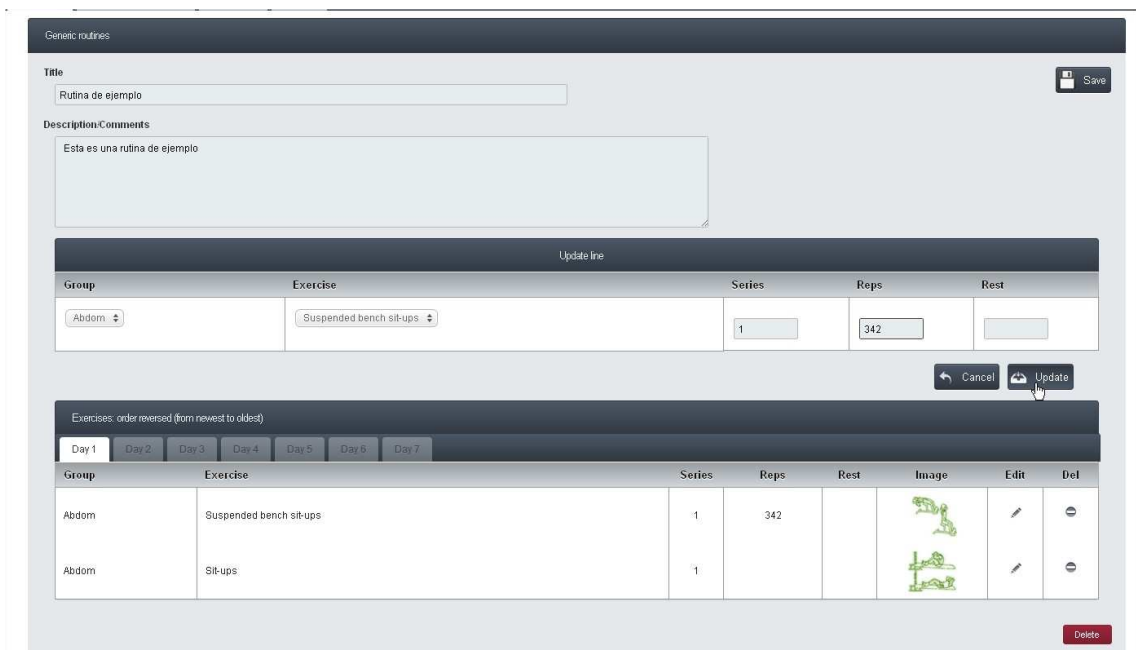


Ilustración 8 Editar línea

3.5.3. Sección ejercicios

Si accedemos a la sección ejercicios veremos los ejercicios de los que dispone nuestra aplicación agrupados por grupos musculares. Desde aquí podremos consultar la información correspondiente a los ejercicios y editar los mismos. Al igual que en la sección rutinas, solo podremos editar los ejercicios creados por nuestra aplicación.



Ilustración 9 Sección ejercicios

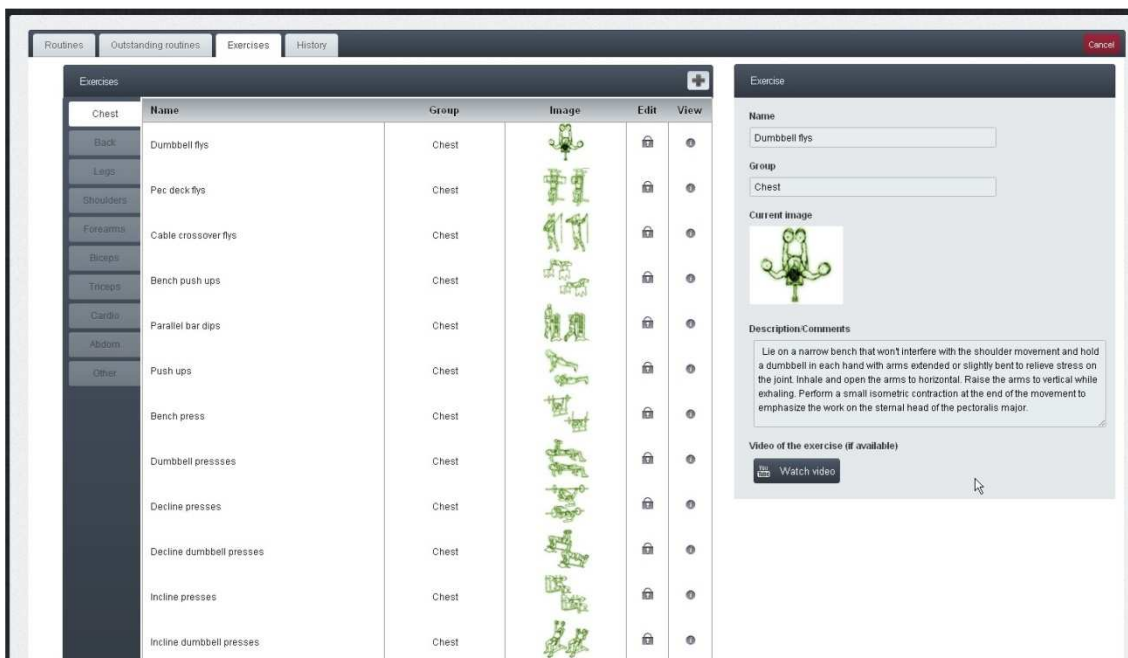


Ilustración 10 Consultar ejercicio

Si el ejercicio seleccionado para consultar su información dispone de un vídeo explicativo podremos cargarlo para verlo.

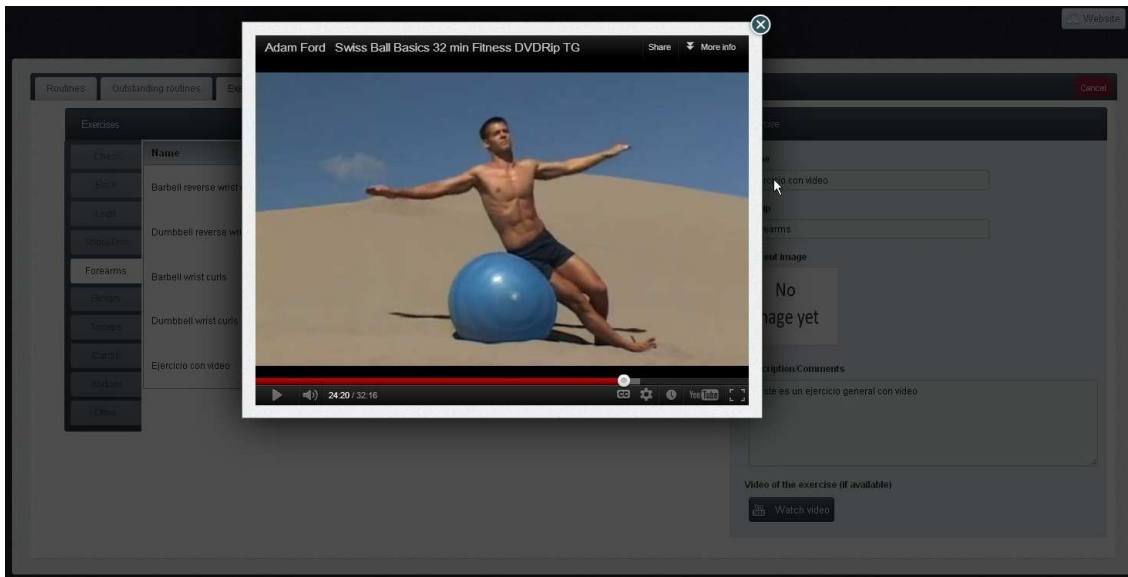


Ilustración 11 Ejercicio con video

Si clicamos en añadir un nuevo ejercicio, aparecerá un formulario donde introduciremos la información del mismo. También podremos cargar una imagen para dicho ejercicio desde nuestro ordenador o añadir un vídeo explicativo del mismo.

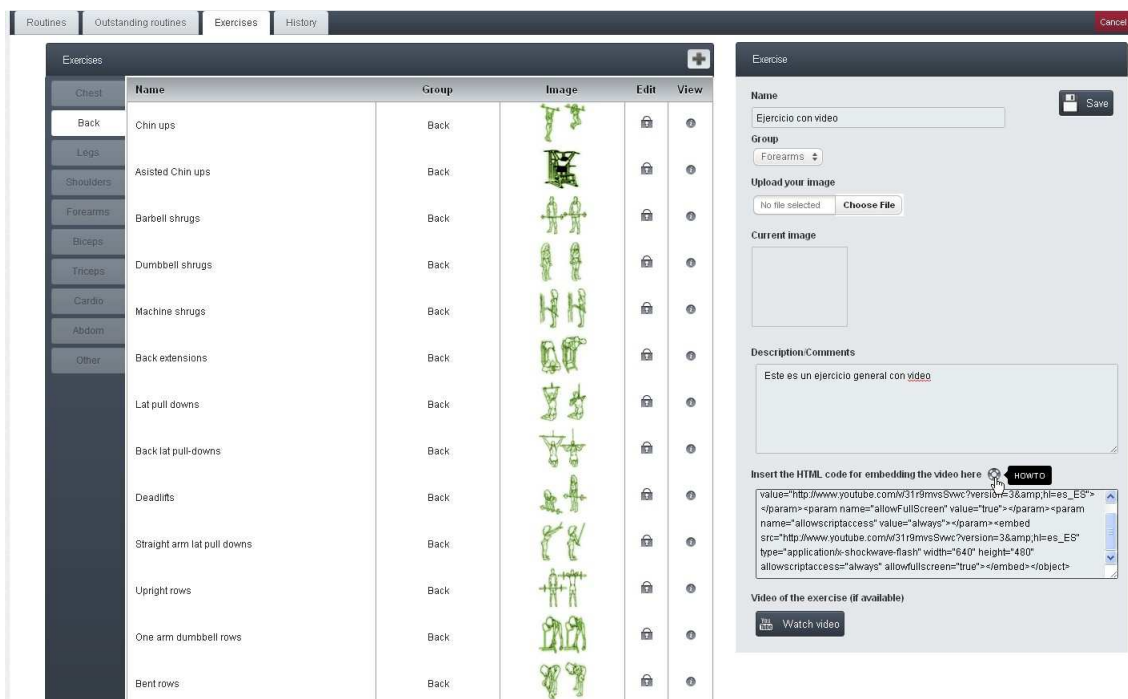


Ilustración 12 Añadir ejercicio

3.5.4. Sección Trabajo pendiente

Al acceder a la sección trabajo pendiente, veremos una tabla con un listado de rutinas por hacer. Estarán ordenadas de mayor a menor antigüedad y el monitor podrá editarlas o asignarlas. Cuando realizamos una asignación, el trabajo pendiente correspondiente desaparece de la lista al haber sido realizado.

User	Requirements	Description of the routine	Date	Edit	Assign
Aristotle		Debo hacer un plan de entrenamiento para Aristotle. Quiere venir 3 dias a la semana y seguir en progresion a sus entrenamientos anteriores.	09-08-2012		
Benedict		adsfasdfdas	22-06-2012		

Ilustración 13 Trabajo pendiente

Al elegir un usuario cuando editamos o añadimos un trabajo pendiente, el sistema despliega una datatable con los usuarios de la aplicación. De nuevo, podremos efectuar búsquedas para agilizar el proceso.

Name	Username	e-mail	Date of birth	Type of user
Aida	aidaacin			User
Alexander	Alexander			User
Amir	Amir			User
Aristotle	Aristotle			User
Beau	Beau			User
Benedict	Benedict			User
Britanni	Britanni			User
Brody	Brody			User
Cameron	Cameron			User
Cedric	Cedric			User

Ilustración 14 Selección de usuario para el trabajo pendiente

User	Requirements	Description of the routine	Date	Edit	Assign
Aristotle		Debo hacer un plan de entrenamiento para Aristotle. Quiere venir 3 dias a la semana y seguir en progresion a sus entrenamientos anteriores.	09-08-2012		
Benedict		adsfasdfdas	22-06-2012		

Ilustración 15 Editar trabajo pendiente

A la hora de asignar una rutina a un trabajo pendiente, el sistema desplegará una datatable similar a la de la sección rutinas, donde se mostraran las diferentes rutinas y la descripción de las mismas. Asimismo, el monitor podrá buscar la rutina a

asignar al usuario o consultar una rutina en concreto para comprobar si es la que más se adecua a la petición del usuario.

The screenshot shows a web application interface for managing fitness routines. At the top, there are navigation tabs: 'Routines', 'Outstanding routines', 'Exercises', and 'History'. The main content area is titled 'Routines' and includes a search bar and a 'Show 10 entries' dropdown. Below this is a table listing various routines such as '3 days routine for men with muscle gain', '3 days routine for women v1', etc. Each row has a 'View' button. Below the list is a pagination control showing 'Showing 1 to 10 of 24 entries'. Underneath is a section for 'Outstanding routines' with an 'Add' button. This section contains a table with columns: 'User', 'Requirements', 'Description of the routine', 'Date', 'Edit', and 'Asign'. The 'Asign' column contains arrows pointing right, indicating that routines are assigned to users.

Ilustración 16 Asignar rutina a usuario

3.5.5. Sección Historial

En esta sección, el monitor podrá consultar las rutinas que han sido asignadas a los usuarios. Siguiendo la idea de secciones anteriores, las asignaciones están estructuradas en una datatable. El monitor podrá buscar un usuario y listar todas y cada una de las asignaciones que han sido realizadas a dicho usuario. Para cada asignación, dispondrá de un botón para ver la rutina en cuestión.

The screenshot shows the 'History' section of the application. It features a search bar and a 'Show 10 entries' dropdown. The main table has columns: 'Username', 'Routine', 'Date', 'Time', and 'View'. The table lists historical assignments for various users, including Alexander, Aristotle, Javier Lukin, and Sergio, with their respective routine names and timestamps. At the bottom, there is a pagination control showing 'Showing 1 to 10 of 14 entries'.

Ilustración 17 Sección historial

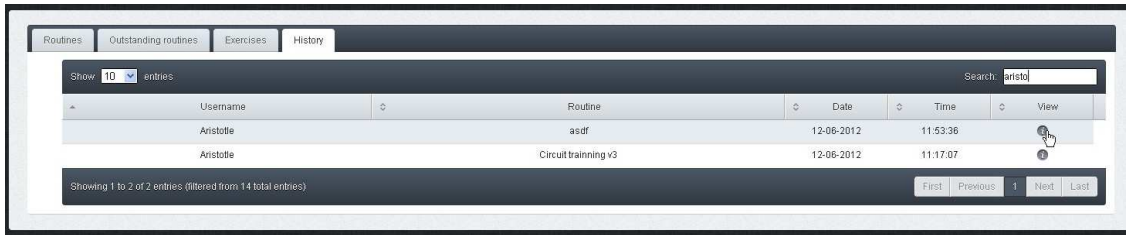


Ilustración 18 Buscar asignaciones a usuario

3.5.6. Settings

Al acceder al apartado settings de la aplicación y seleccionar el apartado gimnasio, aparecerán los tres parámetros que pueden ser configurados de nuestro modulo.



Ilustración 19 Settings

Podremos elegir si activamos la sección gimnasio en el front-end para los usuarios (la entidad cliente puede no disponer de gimnasio así que no le interesaría mostrar la información relativa al mismo a los usuarios) y si usamos o no las rutinas y ejercicios que vienen predefinidos con la aplicación.

3.5.7. Mensajes

El monitor puede intercambiar mensajes con los usuarios. Para crear y consultar los mensajes deberá acceder a la sección mensajes de la aplicación. Desde allí, el monitor podrá seleccionar el usuario al que desea remitir un mensaje (si se trata de un mensaje nuevo) o responder a un mensaje.

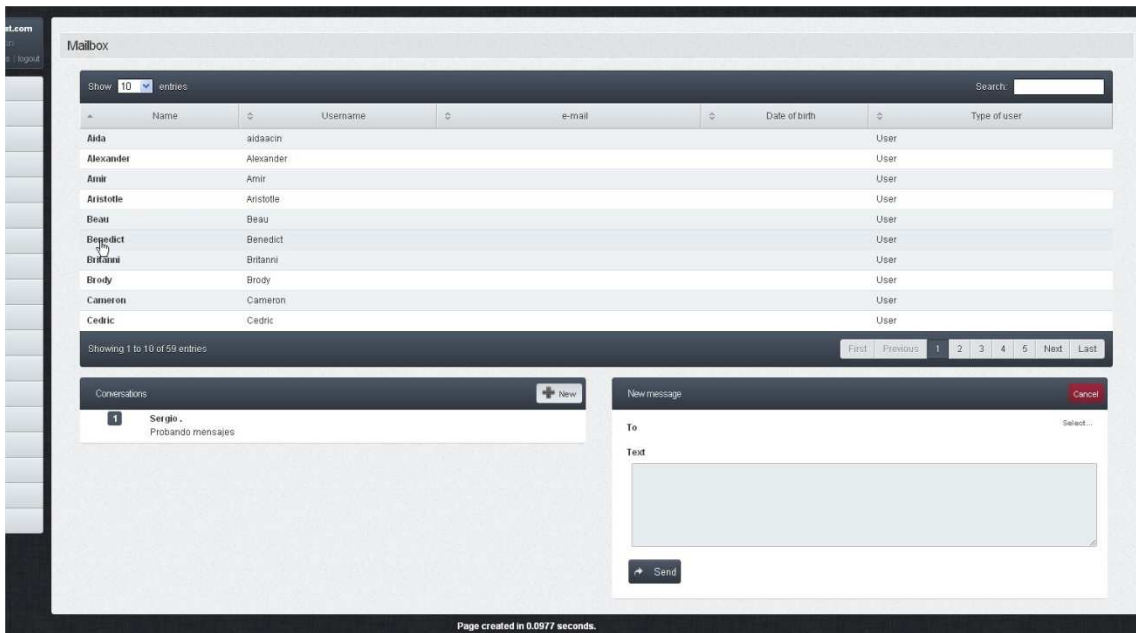


Ilustración 20 Mensajes, seleccionar destinatario

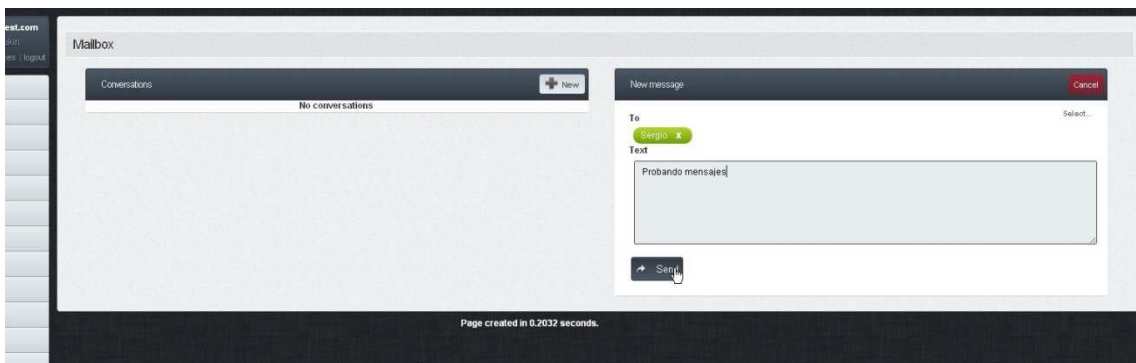


Ilustración 21 Enviar mensaje

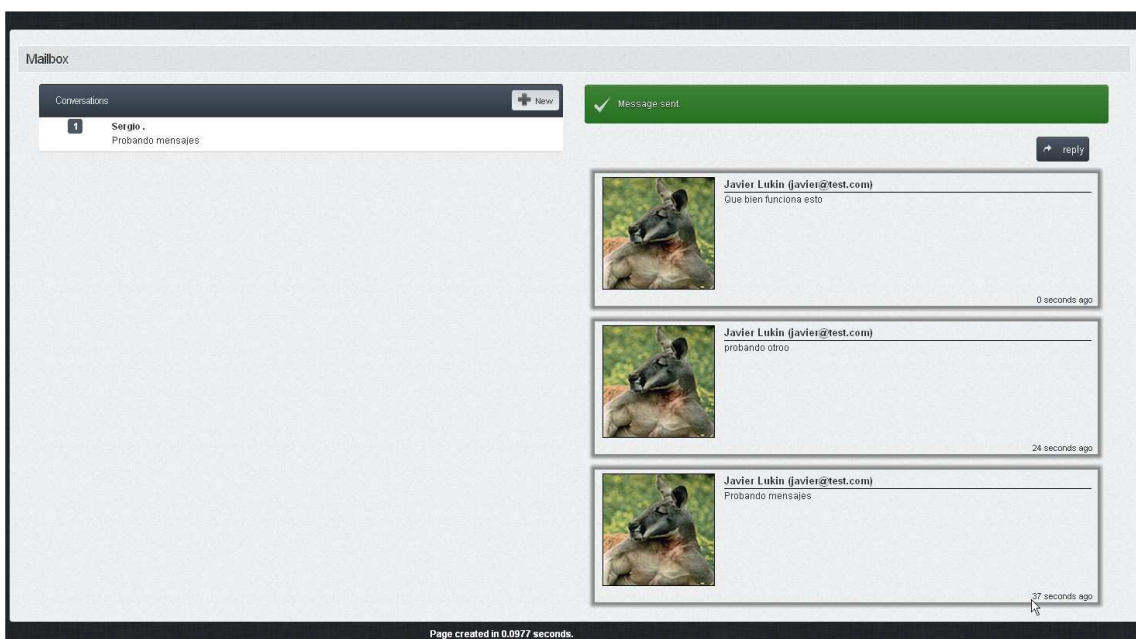


Ilustración 22 Cadena de mensajes

3.5.8. Front-end (usuarios)

El apartado front-end es el correspondiente a los usuarios. Desde la página web que Nitro provee a la entidad cliente los usuarios de dicha entidad pueden interactuar con las instalaciones de la misma. Una vez logueados con sus datos de usuario, si acceden a la sección rutinas y ya tienen alguna rutina asignada verán lo siguiente.

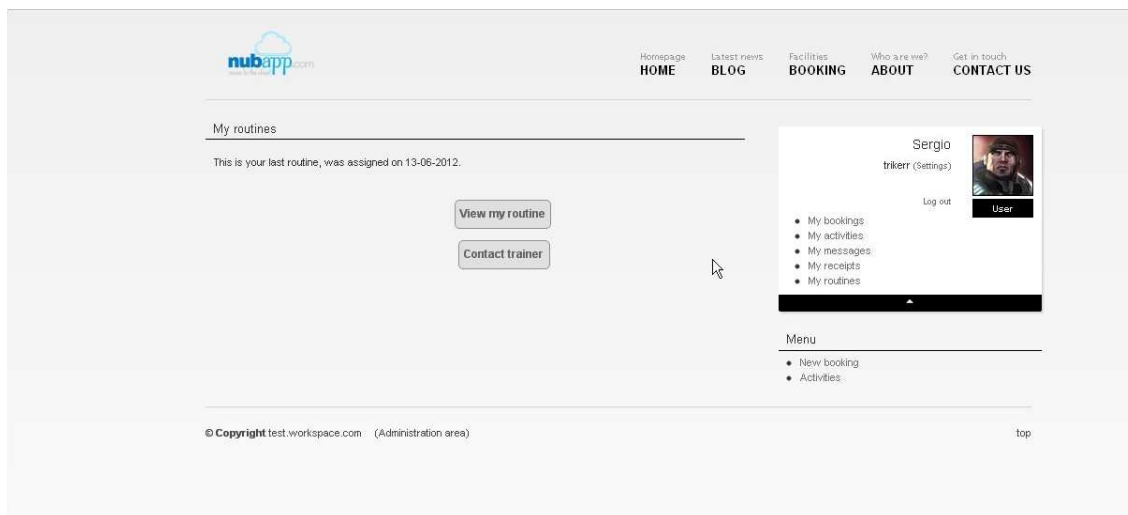


Ilustración 23 Usuario con rutina asignada

Desde aquí, los usuarios podrán contactar con el monitor (el sistema creará una nueva conversación con el monitor y redirigirá al usuario a la sección mensajes) o ver e imprimir su rutina.

[nubapp.com](#)
[HOME](#)
[BLOG](#)
[BOOKING](#)
[ABOUT](#)
[CONTACT US](#)

My routines

This is your last routine, was assigned on 13-06-2012.

[View my routine](#)

Title
German volume training month one

[Print without images](#) [Print](#)

Description/Comments
Do not stay more than 4 weeks with this routine.

Day 1: Exercises

Group	Exercise	Series	Reps	Rest	Image	Info
Legs	Squats	3	10	60 s		ⓘ
Legs	Front squats	3	10	60 s		ⓘ
Legs	Hack squats	3	10	60 s		ⓘ
Legs	Leg extensions	3	10	60 s		ⓘ

Day 2: Exercises

Group	Exercise	Series	Reps	Rest	Image	Info
Chest	Incline presses	3	10	60 s		ⓘ
Chest	Decline presses	3	10	60 s		ⓘ

Sergio trikerr (Settings)
[Log out](#) [User](#)

- My bookings
- My activities
- My messages
- My receipts
- My routines

Menu

- New booking
- Activities

Ilustración 24 Mi rutina

Y consultar la información disponible sobre los diferentes ejercicios.

[nubapp.com](#)
[HOME](#)
[BLOG](#)
[BOOKING](#)
[ABOUT](#)
[CONTACT US](#)

My routines

This is your last routine, was assigned on 13-06-2012.

[View my routine](#)

Title
German volume training month one

[Print without images](#) [Print](#)

Description/Comments
Do not stay more than 4 weeks with this routine.

Day 1: Exercises

Group	Exercise	Series	Reps	Rest	Image	Info
Legs	Squats	3	10	60 s		ⓘ

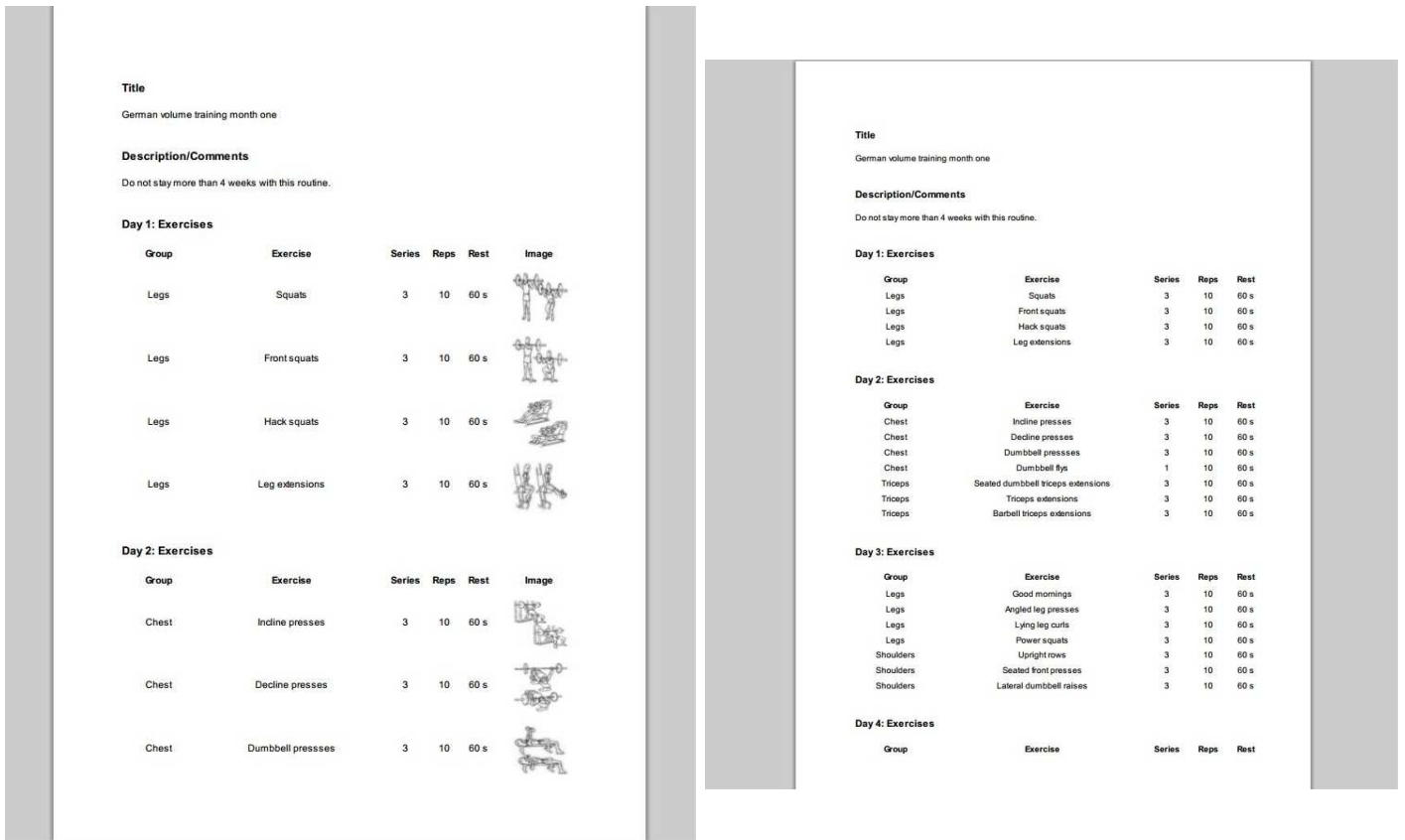
With the barbell resting on a stand, slide under the bar and place it on the trapezius a bit higher than the posterior deltoid. Grasp the bar firmly with the hands at a comfortable width and the elbows back. Inhale deeply, slightly arch the back by rotating the pelvis forward, contract the abdominal core, look straight ahead, and remove the barbell from the stand. Step back one or two steps and stop with both feet parallel to each other and about shoulder-width apart. Bend forward from the hips and avoid rounding the back in order to prevent injury. When the thighs are horizontal to the floor, straighten the legs and lift the torso to return to the initial position. Exhale at the end of the movement.

[Watch video](#)

Legs	Front squats	3	10	60 s		ⓘ
Legs	Hack squats	3	10	60 s		ⓘ
Legs	Leg extensions	3	10	60 s		ⓘ

Ilustración 25 Mi rutina, ver información

Los usuarios podrán imprimir y guardar sus rutinas de dos maneras diferentes, con y sin imágenes.



Por último, si el usuario que accede al front-en no dispone de ninguna rutina asignada, dispondrá de un formulario para pedir una directamente. Una vez enviada, la información de dicho cuestionario aparecerá directamente en la sección trabajo pendiente del monitor. Cabe destacar que esto no supe la posibilidad de un intercambio de mensajes entre el usuario y el monitor. Se trata de una funcionalidad diferente cuyo objetivo es agilizar el proceso en sí.



Ilustración 26 Pedir rutina

Outstanding routines					
User	Requirements	Description of the routine	Date	Edit	Asign
Sergio		Hola, Me gustaria que se me asignara un plan de entrenamiento para ir tres dias a la semana. Si quieres podemos concretar una cita, de todas maneras, te comento lo que quiero por si acaso. Tengo 23 años u peso 80 kilos, soy de complecion ancha y nunca he ido al gimnasio. Me gustaria adelgazar y tonificar mi cuerpo. Un saludo	13-08-2012		

Page created in 0.2913 seconds.

Ilustración 27 Pedir rutina enviada a trabajos pendientes

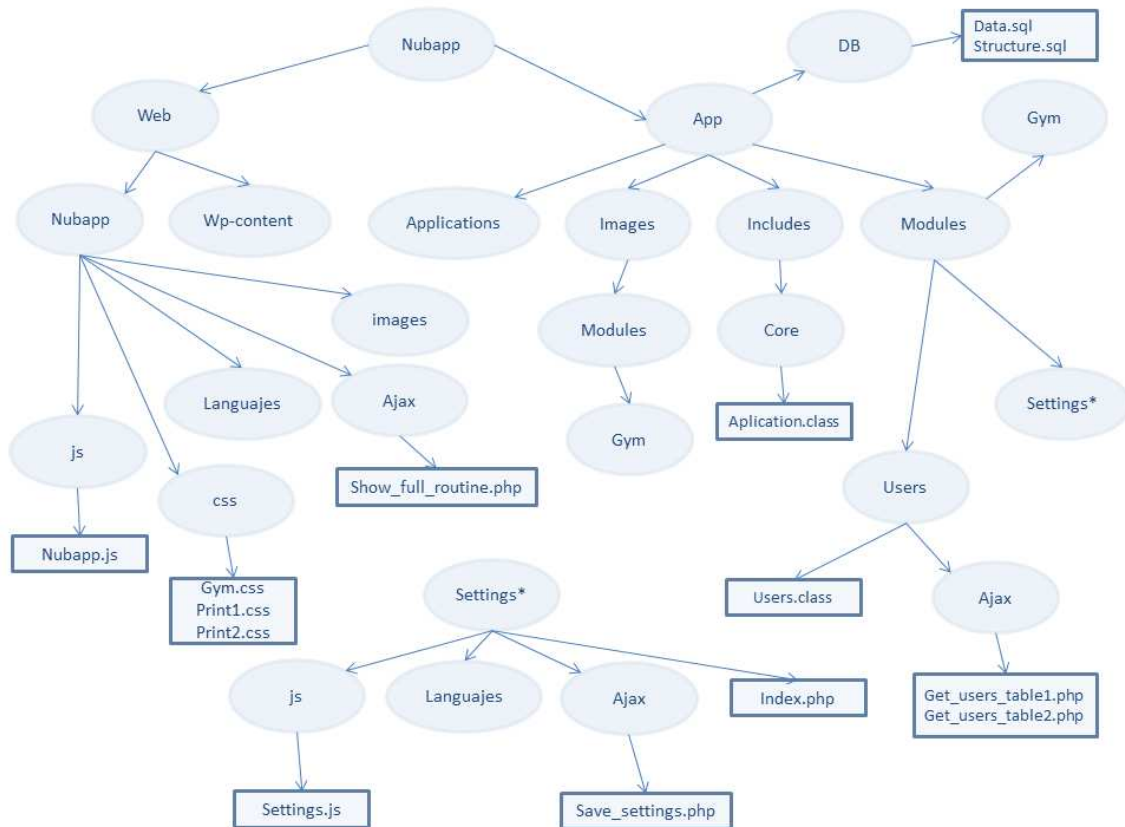
4. Diseño del sistema de información

4.1 Definición de la arquitectura del sistema

Nitro es una aplicación modular. Se utilizan tanto clases comunes para todos los módulos como clases propias de cada módulo. Como ya se comentó en la introducción, es una aplicación multidioma. Debido a ello se dispondrá de archivos diccionario, y todas y cada una de las variables de texto que mostraremos deberán corresponderse a una de las variables diccionario de nuestro modulo, siendo dicha variable será la que contenga el texto. En función del idioma de la aplicación, se cargarán unos diccionarios u otros.

El código se desarrollará en una versión de repositorios demo con servidor propio, una vez implementada una funcionalidad completa se probará en una versión test almacenada en otro servidor. Una vez testada en test se podrá añadir a la aplicación comercial, alojada en nubapp.com

A continuación se muestra un árbol con la estructura de directorios y archivos del programa. Al ser parte de una aplicación con mas funcionalidades a parte de la implementada, se muestran solo las creadas o modificadas de manera significativa por nuestro módulo.

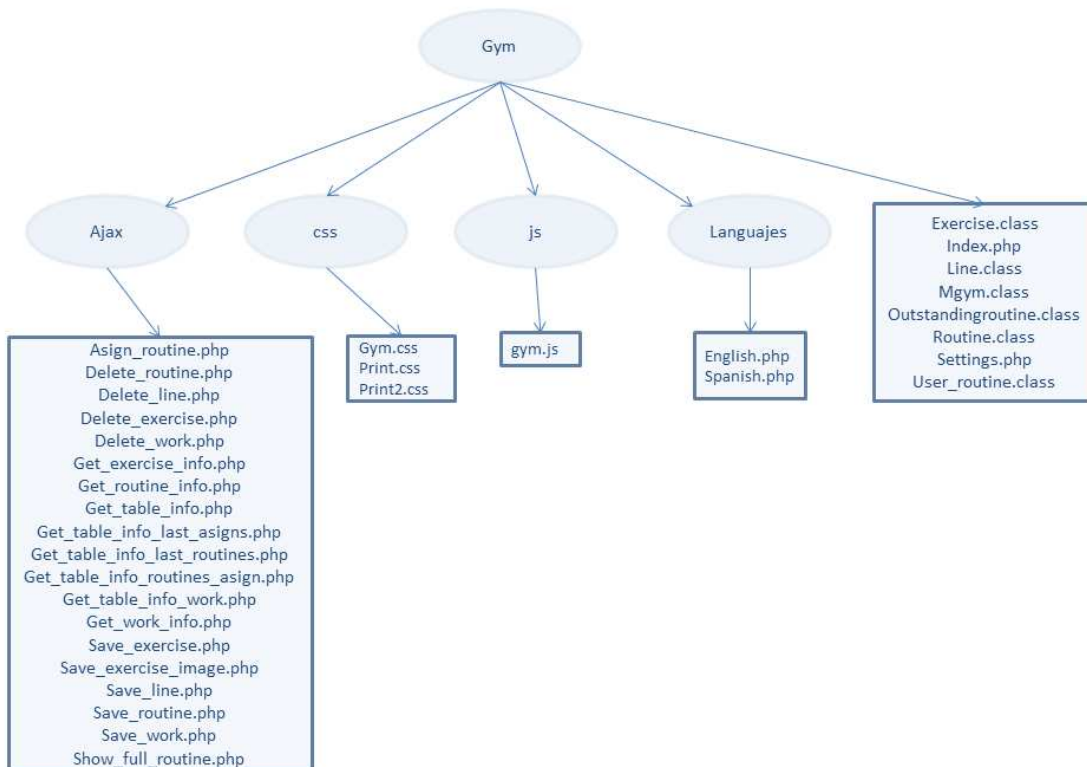
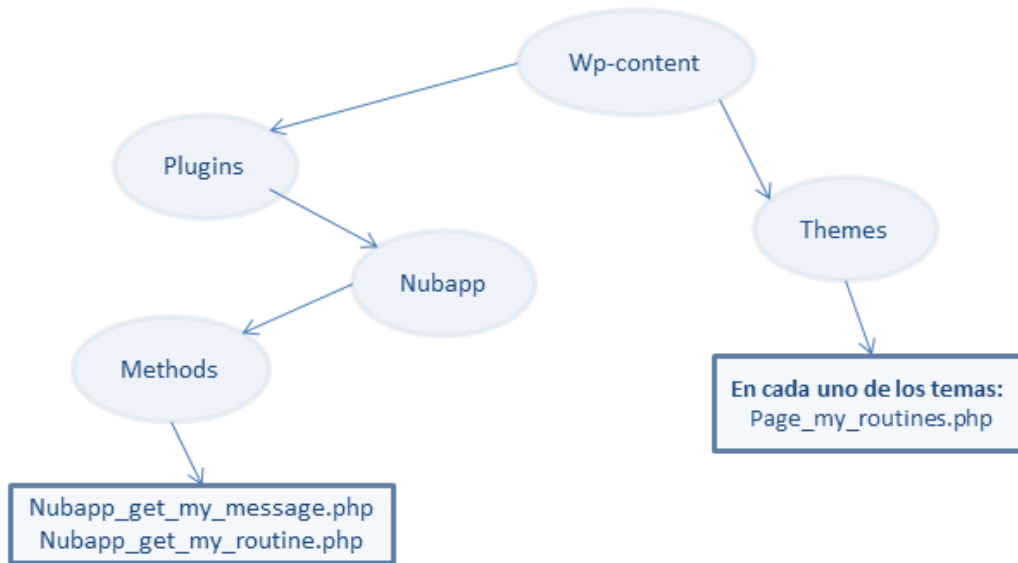


Web es la parte correspondiente al front-end mientras que App es la parte correspondiente a la aplicación en si o back-end. No es nuestro objetivo explicar que hace cada uno de los archivos implementados, sin embargo, comentaremos algunos detalles que pueden no quedar claros a simple vista.

En Nubapp->App->applications habrá una carpeta por cada una de las aplicaciones cliente que han contratado los servicios de la aplicación, dentro de dicha carpeta nuestro módulo guardará las imágenes que hayan sido creadas por dicha entidad cliente.

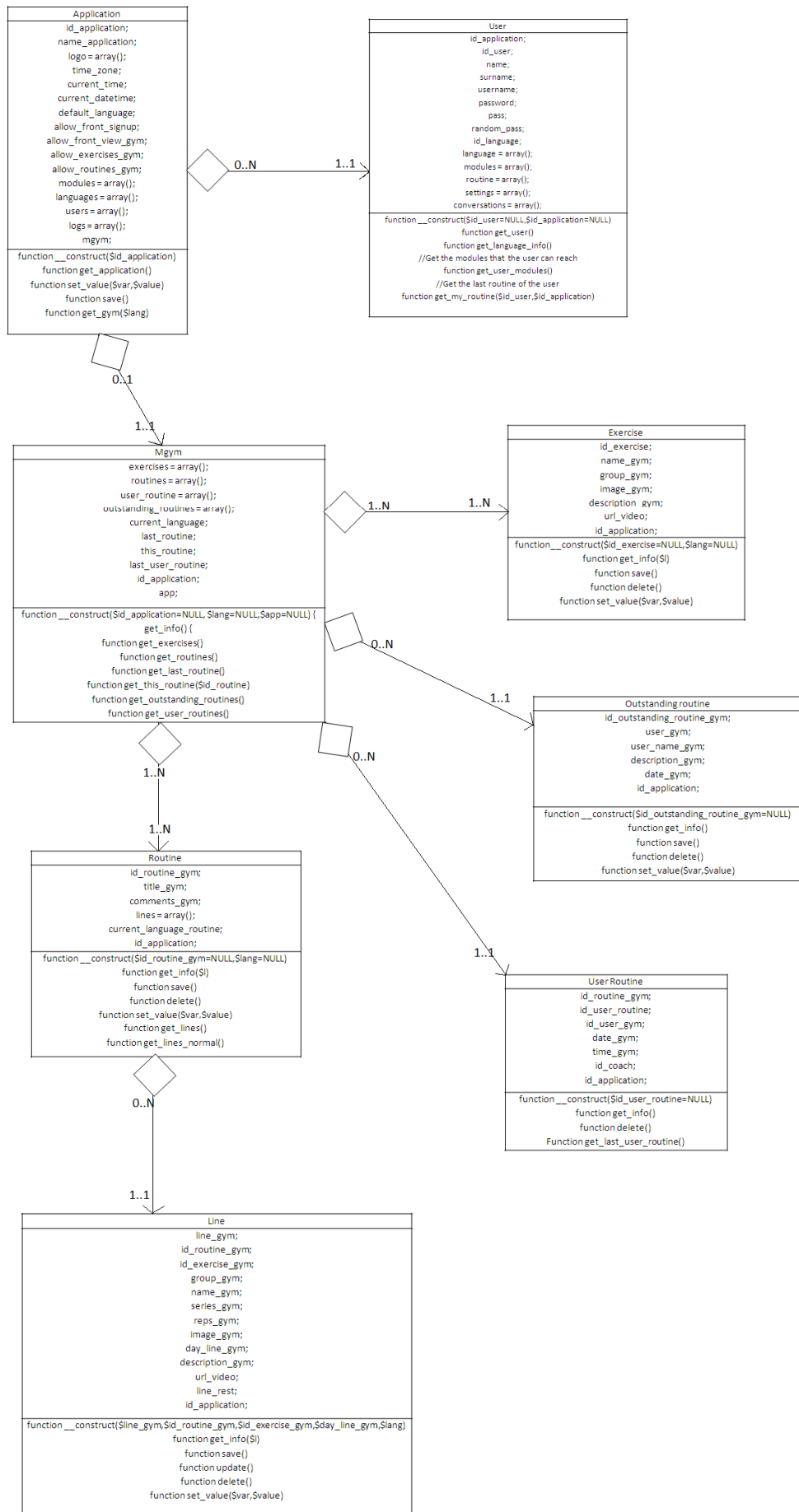
En Nubapp->App->images->modules->gym se guardan todas las imágenes que utiliza nuestro programa para las rutinas y ejercicios que vienen predefinidos en el sistema.

Las ramificaciones Nubapp->App->modules->gym y Nubapp->web->wp-content no se han mostrado por motivos de espacio, las detallamos a continuación:



4.2 Diseño de clases

A continuación se muestra un diagrama con el diseño detallado de cada una de las clases que forman parte de nuestro módulo, sus atributos, operaciones, relaciones y métodos, y la relación jerárquica del mismo. Son clases finales y han sido obtenidas a partir de las especificaciones y casos de uso.



Clase application

Application
<pre>id_application; name_application; logo = array(); time_zone; current_time; current_datetime; default_language; allow_front_signup; allow_front_view_gym; allow_exercises_gym; allow_routines_gym; modules = array(); languages = array(); users = array(); logs = array(); mgym;</pre>
<pre>function __construct(\$id_application) function get_application() function set_value(\$var,\$value) function save() function get_gym(\$lang)</pre>

La aplicación general dispone nada más loguearse de un objeto application que carga toda la información relativa al sistema. Mediante dicho objeto sabremos la hora y fecha local de dicha aplicación, el logo y nombre del cliente que debe ser mostrado, el lenguaje utilizado por el cliente o los módulos contratados (permitiendo o no el acceso a los mismos). Este objeto dispone asimismo de un array de objetos de la clase user, desde donde podremos acceder y manejar todos los usuarios de dicha aplicación.

Entre las funciones, y por su relación con nuestro modulo, cabe destacar la función get_gym, encargada de crear un objeto de clase mgym con toda la información de nuestro módulo gimnasio (la carga de datos se realiza en base a los atributos id_application y default_lenguaje de la clase application) de la entidad cliente que se ha logueado.

Clase Mgym

Mgym
<pre>exercises = array(); routines = array(); user_routine = array(); outstanding_routines = array(); current_language; last_routine; this_routine; last_user_routine; id_application; app;</pre>

```

function __construct($id_application=NULL, $lang=NULL,$app=NULL)
    get_info()
    function get_exercises()
    function get_routines()
    function get_last_routine()
function get_this_routine($id_routine)
function get_outstanding_routines()
function get_user_routines()

```

La clase Mgyim permite acceder y cargar toda la información correspondiente al módulo gimnasio de una aplicación dada. De manera general cargando toda la información (get_info) o de manera más específica para los diferentes apartados del mismo (get_routines(), get_exercises()...), estando enfocadas estas últimas funciones en la recarga dinámica de datos de los diferentes apartados, por ejemplo, si introducimos una nueva rutina, no hará falta recargar toda la información del módulo, sino que recargaremos la información correspondiente a las rutinas.

Para almacenar la información relativa a rutinas, ejercicios, trabajo pendiente, etc., se han declarado diferentes arrays de objetos. De esta manera por ejemplo, en el atributo routines tendremos un array con objetos rutinas de la clase routine. La clase también dispone de otras funciones como por ejemplo acceder a la información de una rutina en concreto (get_this_routine).

Clase outstandingroutine

Outstanding_routine
id_outstanding_routine_gym; user_gym; user_name_gym; description_gym; date_gym; id_application;
function __construct(\$id_outstanding_routine_gym=NULL) function get_info() function save() function delete() function set_value(\$var,\$value)

Esta clase permite acceder, modificar, guardar o borrar objetos de la sección trabajo pendiente.

Clase exercise

Exercise

<pre> id_exercise; name_gym; group_gym; image_gym; description_gym; url_video; id_application; </pre>
<pre> function __construct(\$id_exercise=NULL,\$lang=NULL) function get_info(\$l) function save() function delete() function set_value(\$var,\$value) </pre>

Esta clase permite acceder, modificar, guardar o borrar objetos de la sección ejercicios

Clase Routine

Routine
<pre> id_routine_gym; title_gym; comments_gym; lines = array(); current_language_routine; id_application; </pre>
<pre> function __construct(\$id_routine_gym=NULL,\$lang=NULL) function get_info(\$l) function save() function delete() function set_value(\$var,\$value) function get_lines() function get_lines_normal() </pre>

Esta clase permite acceder, modificar, guardar o borrar objetos de la sección rutinas. Entre sus atributos debemos destacar el atributo lines, se trata de un array que contiene las líneas de la rutina, el array está formado por objetos de la clase line.

En relación con esto se dispone de dos funciones (get_lines() y get_lines_normal()) que cargan las líneas de la rutina en orden ascendente o descendente. Este último aspecto es necesario dado que al crear la rutina vemos las líneas que hemos insertado recientemente las primeras, mientras que al mostrar una rutina se cargan en el orden en que han sido insertadas.

Clase Line

Line
<pre>line_gym; id_routine_gym; id_exercise_gym; group_gym; name_gym; series_gym; reps_gym; image_gym; day_line_gym; description_gym; url_video; line_rest; id_application;</pre>
<pre>function __construct(\$line_gym,\$id_routine_gym,\$id_exercise_gym,\$day_line_gym,\$lang) function get_info(\$l) function save() function update() function delete() function set_value(\$var,\$value)</pre>

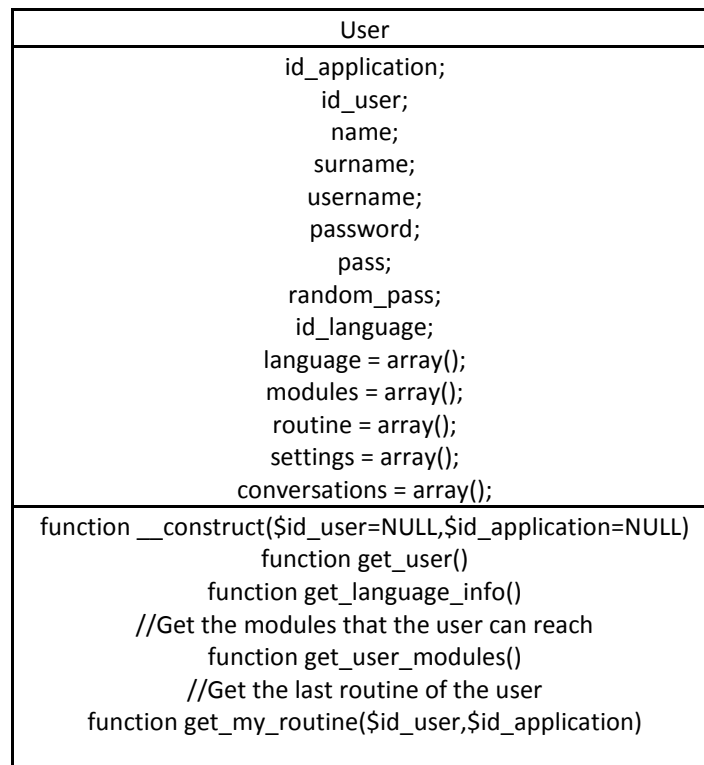
Esta clase permite acceder, modificar, guardar o borrar objetos línea que forman parte de distintas rutinas.

Clase UserRoutine

UserRoutine
<pre>id_routine_gym; id_user_routine; id_user_gym; date_gym; time_gym; id_coach; id_application;</pre>
<pre>function __construct(\$id_user_routine=NULL) function get_info() function delete() Function get_last_user_routine()</pre>

Esta clase permite acceder, modificar, guardar o borrar objetos de la sección historial.

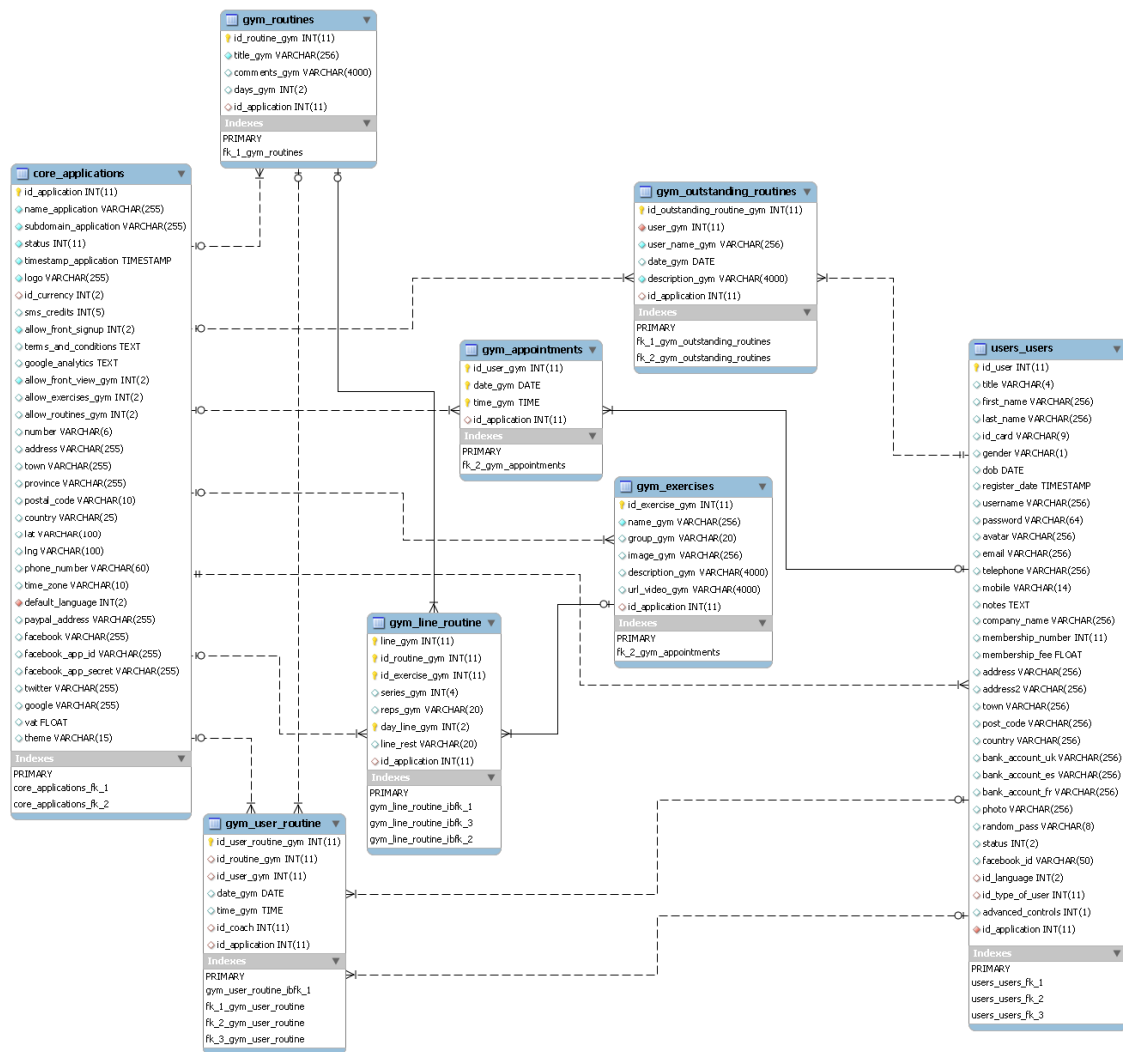
Clase user



Esta clase permite acceder, modificar, guardar o borrar objetos usuario que forman parte de la aplicación de una entidad que haya contratado los servicios de Nubapp. El sistema considera usuarios tanto a trabajadores (monitor) como a clientes de las instalaciones. La diferencia entre usuarios radica en los privilegios de los mismos. Es por ello que disponemos de una función `get_user_modules` que nos determinará a que secciones tiene acceso el usuario. Por otro lado, si se trata de un usuario de tipo cliente, en el atributo `routine` tendrá almacenada la rutina que le ha sido asignada (este atributo es usado en el front-end para mostrar la rutina asignada al mismo).

4.3 Diseño físico de datos

En el siguiente esquema podemos ver el modelo físico de datos de nuestro sistema, para simplificarlo, se ha omitido la estructura correspondiente a los mensajes, que se mostrará posteriormente:

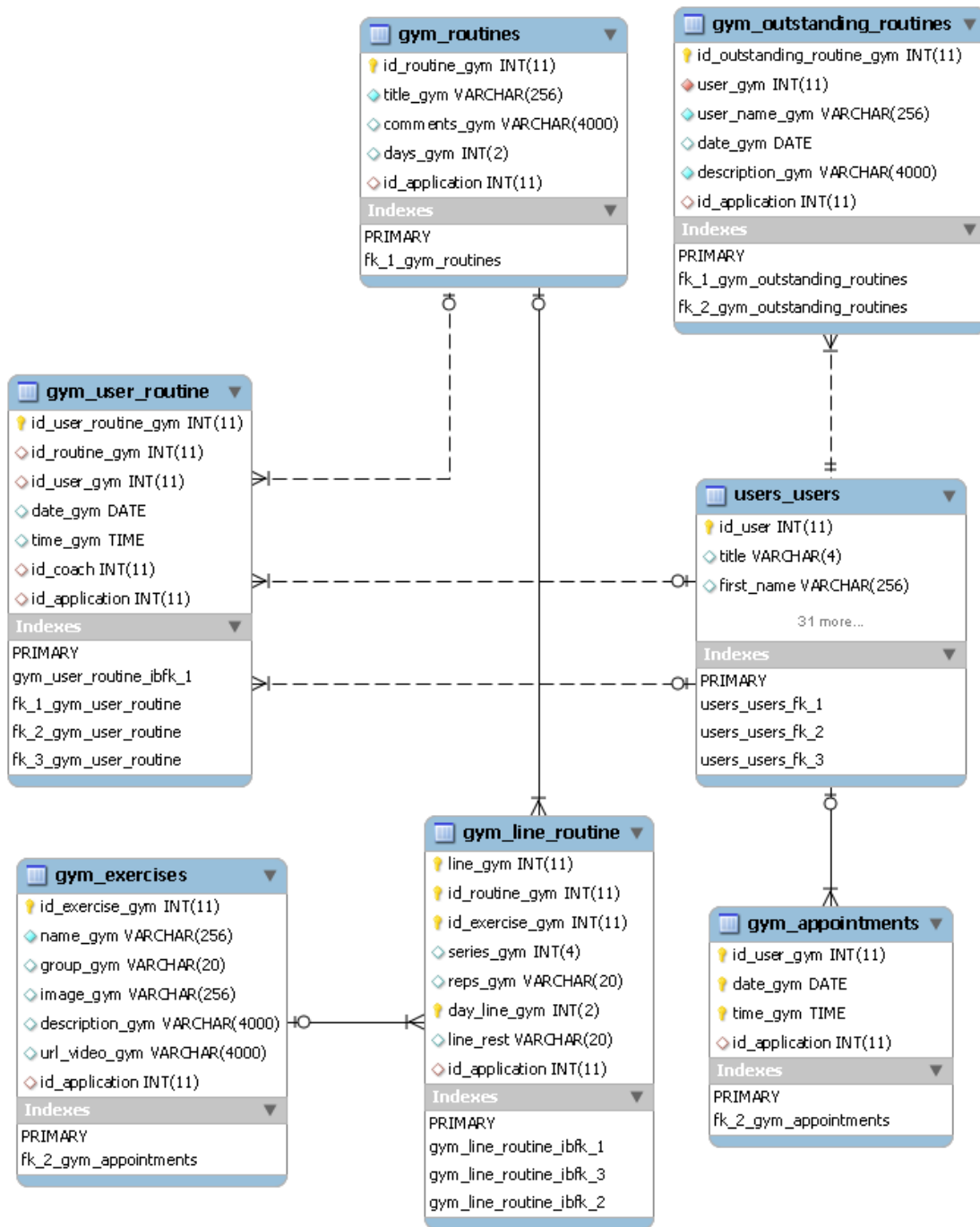


Como podemos observar, basándonos en el diseño de clases realizado, hemos optado por un mapeado del tipo “una clase, una tabla”, implementando de esta manera una tabla por cada una de las clases utilizadas en el sistema. En lo referente a rutinas, podríamos haberlas agrupado en una tabla en lugar de utilizar dos (línea rutina y rutina). No obstante, dado que se decidió que el usuario del sistema podría crear rutinas sin líneas, y considerando el tráfico de datos correspondiente a insertar y editar líneas sucesivamente en una misma rutina y sus posteriores cargas para refrescar el sistema, se ha optado por la utilización de dos tablas en lugar de una.

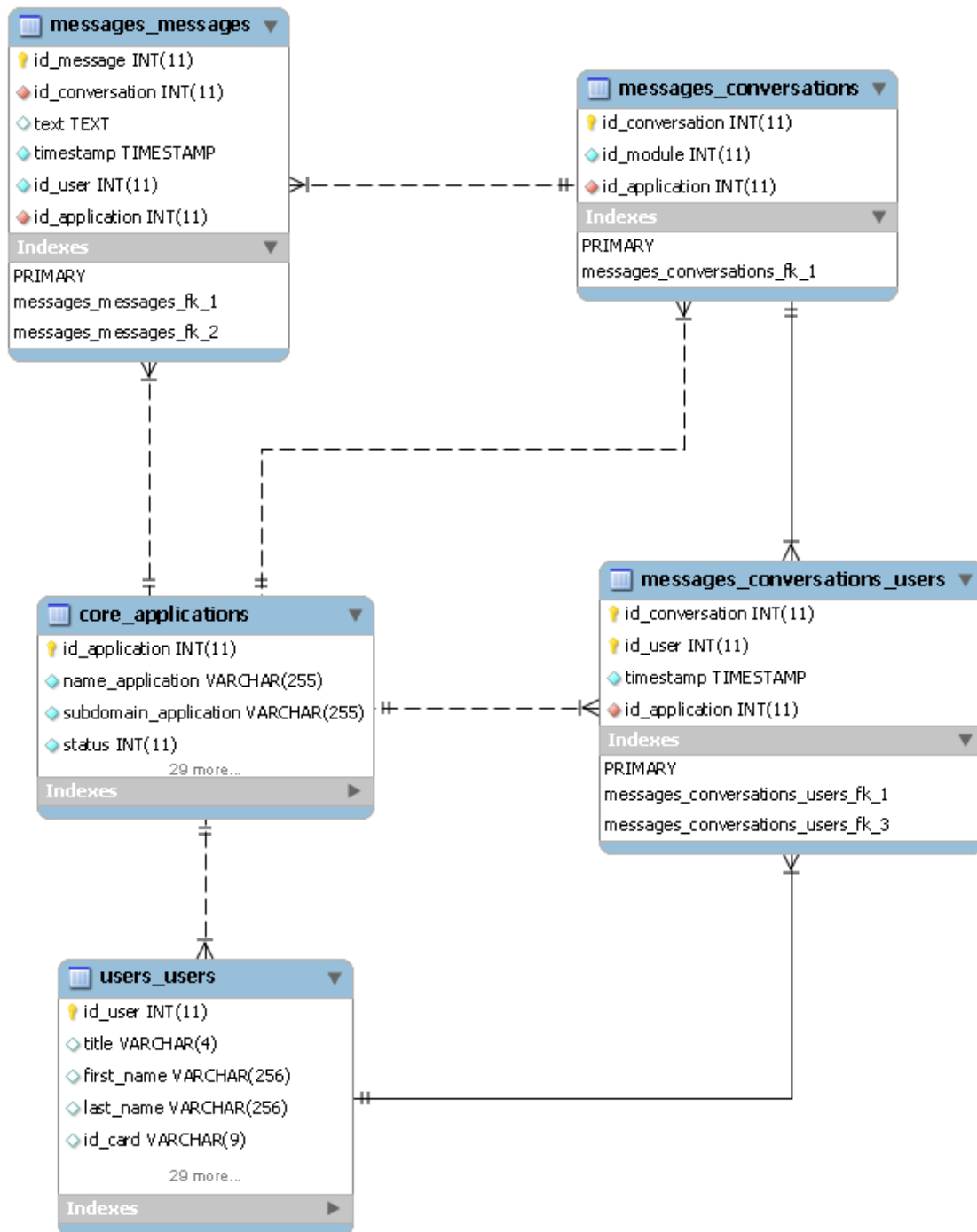
De esta manera, dado que no tendremos necesidad de enlazar las tablas rutina y línea rutina muy a menudo, optimizamos la actualización de datos desde la propia BD al acceder siempre a una tabla con menos atributos.

Como se ha comentado en apartados anteriores, deberemos diferenciar entre diferentes tipos de aplicaciones en función de la entidad cliente que use el mismo, es por ello que todas las tablas creadas tienen un atributo id_application, que referencia a la tabla core_applicacion.

En esta vista podemos ver el modelo físico de datos simplificado, pues hemos omitido la referencia a core_applications:



Por último, el modelo físico de datos correspondiente al módulo mensajes es el siguiente:



4.4 Diseño de la migración y carga inicial de datos

En el apartado de especificación de requisitos, vimos como uno de ellos establecía que la aplicación debía contar con rutinas y ejercicios predefinidos:

R8	Carga inicial de datos
	La aplicación dispondrá de entrenamientos y ejercicios predefinidos.

Dicho de otra manera, un usuario futuro de la aplicación dispondrá de suficiente información como para realizar sus tareas sin necesidad de introducir él mismo ninguna información. Para ello, se han cargado rutinas y ejercicios predefinidos.

Dado que el sistema trabaja con archivos diccionario, se ha creado un programa que carga la información relativa a rutinas y ejercicios predefinidos desde dichos archivos en función del idioma de la aplicación. Asimismo, se han creado los scripts SQL correspondientes a la inserción inicial de datos con la información relativa a los ejercicios y rutinas predefinidas.

Los ejercicios predefinidos que vienen con la aplicación son 100 y su información ha sido obtenida de la “Guía de movimientos de musculación” de Charles Delavier. Los correspondientes a otros idiomas (inglés y francés) han sido obtenidos de la correspondiente versión idiomática de dicho libro.

Por último, el sistema también incorpora 20 rutinas de entrenamiento predefinidas.

4.5 Diseño del plan de pruebas

Una vez realizada nuestra aplicación, es necesario definir un plan de pruebas para la misma con el objetivo de encontrar posibles errores. A continuación, se detallarán esquemáticamente las pruebas realizadas.

Requisito	Descripción	Prueba
R1	Ejecución de la aplicación	P1
	Pruebas: Se realizaran todas las posibles ejecuciones del programa en los tres navegadores comprobando su correcto funcionamiento en los mismos y solucionando los problemas en caso de haberlos.	
R2	Gimnasio -- Rutinas	P2.1 P2.2 P2.3
	Pruebas: Para todos y cada uno de los formularios de la sección, se comprobará que no se pueden meter datos incorrectos, y que no se pueda enviar información inconclusa o que no cumpla con los requisitos de funcionalidad del sistema (por ejemplo, en el campo series solo se permiten enteros, mientras que en el campo repeticiones se permiten caracteres). Se comprobará que el despliegue y ocultación de las diferentes ventanas se hace de la manera correcta. Ocultando las antiguas siempre que accedamos a una nueva.	

	<p>La carga y actualización de las tablas y datos se realiza de manera correcta tanto en la base de datos como en la aplicación. Mostrando correctamente la nueva entrada en caso de que hayamos introducido los datos correctamente. Se comprobara que el SGBD se actualiza de manera correcta.</p>	
R3	<p>Gimnasio – Ejercicios</p> <p>Pruebas: Para todos y cada uno de los formularios de la sección, se comprobará que no se pueden meter datos incorrectos, y que no se pueda enviar información inconclusa o que no cumpla con los requisitos de funcionalidad del sistema.</p> <p>Se comprobará que el despliegue y ocultación de las diferentes ventanas se hace de la manera correcta. Ocultando las antiguas siempre que accedamos a una nueva.</p> <p>La carga y actualización de las tablas y datos se realiza de manera correcta tanto en la base de datos como en la aplicación. Mostrando correctamente la nueva entrada en caso de que hayamos introducido los datos correctamente. Se comprobará que el SGBD se actualiza de manera correcta.</p>	<p>P3.1 P3.2 P3.3</p>
R4	<p>Gimnasio – Trabajo pendiente</p> <p>Pruebas: Para todos y cada uno de los formularios de la sección, se comprobará que no se pueden meter datos incorrectos, y que no se pueda enviar información inconclusa o que no cumpla con los requisitos de funcionalidad del sistema.</p> <p>Se comprobará que el despliegue y ocultación de las diferentes ventanas se hace de la manera correcta. Ocultando las antiguas siempre que accedamos a una nueva.</p> <p>La carga y actualización de las tablas y datos se realiza de manera correcta tanto en la base de datos como en la aplicación. Mostrando correctamente la nueva entrada en caso de que hayamos introducido los datos correctamente. Se comprobará que el SGBD se actualiza de manera correcta.</p>	<p>P4.1 P4.2 P4.3</p>
R5	<p>Gimnasio – Historial</p> <p>Pruebas: Se comprobará que el despliegue y ocultación de las diferentes ventanas se hace de la manera correcta. Ocultando las antiguas siempre que accedamos a una nueva.</p> <p>La carga y actualización de las tablas y datos se realiza de manera correcta tanto en la base de datos como en la aplicación. Mostrando correctamente la nueva entrada en caso de que hayamos introducido los datos correctamente. Se comprobará que el SGBD se actualiza de manera correcta.</p>	<p>P5.1 P5.2 P5.3</p>
R6	<p>Gimnasio -- Citas</p> <p>Pruebas:</p>	<p>P6.1 P6.2</p>

	<p>Para todos y cada uno de los formularios de la sección, se comprobará que no se pueden meter datos incorrectos, y que no se pueda enviar información inconclusa o que no cumpla con los requisitos de funcionalidad del sistema.</p> <p>Se comprobará que el despliegue y ocultación de las diferentes ventanas se hace de la manera correcta. Ocultando las antiguas siempre que accedamos a una nueva.</p> <p>La carga y actualización de las tablas y datos se realiza de manera correcta tanto en la base de datos como en la aplicación. Mostrando correctamente la nueva entrada en caso de que hayamos introducido los datos correctamente. Se comprobará que el SGBD se actualiza de manera correcta.</p>	P6.3
R7	<p>Configuración de la aplicación</p> <p>Pruebas: Se comprobará que la aplicación carga los datos en función de los parámetros de configuración seleccionados en el apartado settings.</p>	P7
R8	<p>Carga inicial de datos</p> <p>Pruebas: La aplicación carga, muestra y utiliza los ejercicios y rutinas predefinidas de manera correcta y en el lenguaje correspondiente a la aplicación.</p>	P8
R9	<p>Resultados</p> <p>Pruebas: Comprobaremos que se obtiene información tanto de éxito como de error en las diferentes acciones que actualicen la BBDD (a excepción de la inserción de líneas).</p>	P9
R10	<p>Logs</p> <p>Pruebas: Se comprobará en el archivo log que el modulo ha escrito sus entradas de manera correcta cuando era necesario.</p>	P10
R11	<p>Permisos</p> <p>Pruebas: Comprobaremos que no es posible acceder a apartados que nos son restringidos. Intentaremos, logueados como monitor acceder a otros módulos. Como usuario y de manera externa intentaremos acceder a nuestro modulo.</p>	P11
R12	<p>Diseño reutilizable y claridad de código</p> <p>Pruebas: Revisaremos el código comprobando que está debidamente comentado y clarificado.</p>	P12
R13	<p>Homogeneidad</p> <p>Pruebas: Comprobaremos que todas y cada una de las interfaces siguen el esquema de estilos desarrollado por Nubapp.</p> <p>Comprobaremos que los elementos y botones comunes a otras partes de la aplicación siguen las mismas normas de estilo, situación y acción que los del resto de la aplicación.</p>	P13.1 P13.2
R14	Facilidad de uso	P14

	<p>Pruebas: Tanto nosotros como alguien ajeno a la empresa comprobaremos el funcionamiento del programa y si el mismo es intuitivo o no por medio de una valoración personal.</p>	
R15	<p>Comunicación usuario – monitor</p> <p>Pruebas: Comprobaremos que el envío de los mensajes se realiza de manera correcta. No dejaremos enviar un mensaje sin contenido o sin destinatario. La carga y respuesta a mensajes anteriores se realizará de manera correcta y comprobaremos que se informa al usuario del éxito o fracaso de la operación. Se comprobará que no se puede añadir un usuario que no esté registrado.</p>	P15
R16	<p>Front-end</p> <p>Pruebas: Para el formulario de contacto con el monitor, se redirigirá al usuario a la sección mensajes. Es por ello que nos remitiremos a las pruebas de la sección anterior.</p> <p>Se comprobará que el despliegue y ocultación de las diferentes ventanas se hace de la manera correcta. Ocultando las antiguas siempre que accedamos a una nueva.</p> <p>La carga y vista de la rutina se efectúa de manera correcta. El contenido multimedia no presenta errores si la información sobre el mismo se encuentra vacía o incompleta</p>	P16.1 P16.2 P16.3
R17	<p>API</p> <p>Pruebas: Se comprobará que el API funciona correctamente ante llamadas al mismo autorizadas y que los datos son devueltos de manera correcta.</p>	P17
R18	<p>Aplicación móvil</p> <p>Pruebas: No se han diseñado ni realizado pruebas para este requisito al tratarse de un diseño no implementado todavía.</p>	P18
R19	<p>Dinamismo</p> <p>Pruebas: Se comprobará que la aplicación carga toda la información relativa a la entidad cliente. Los elementos predefinidos y creados por dicho cliente serán los únicos datos a mostrar. No se podrá acceder a la información creada por otra entidad cliente.</p>	P19
R20	<p>Diccionarios</p> <p>Pruebas: Se comprobará que en función del idioma de la aplicación cargamos un diccionario u otro de manera correcta. Las variables y textos del diccionario se revisarán para que no presenten faltas de ortografía ni de gramática.</p>	P20

5. Conclusiones

Esta ha sido mi primera experiencia laboral dentro del mundo de la informática. Cuando Sergio Fernández me propuso realizar el PFC con Nubapp durante el curso académico me mostré reticente a compaginar las asignaturas de 5º y el PFC. Al final acepté su propuesta y la experiencia ha merecido la pena pues durante todo este año he aprendido mucho. El desarrollo de esta aplicación ha sido complejo, pues al tratarse de una aplicación en la nube que suple la utilización de un software local, nuestro sistema debía ser dinámico y no tratarse de una serie de formularios que se van cargando página tras página. Gracias al uso de Ajax y jQuery, la interacción usuario—sistema es fluida y estática (siempre nos encontramos en la misma página) y la aplicación carga y envía datos sin que el usuario lo note.

El desarrollo de esta aplicación en la nube ha sido desde luego una experiencia muy enriquecedora que me ha permitido poner en práctica algunos de los conocimientos adquiridos durante estos años en la universidad y además hacerlo en el entorno laboral real de la empresa. Como es lógico, durante todo este tiempo y en cuanto a conocimientos técnicos, he mejorado notablemente el nivel de conocimientos en PHP, AJAX y JavaScript. A la par que he mejorado mis habilidades de comunicación y he adquirido experiencia en un entorno real de desarrollo joven y con muchas ganas de trabajar.

Por otro lado, durante la realización del PFC he constatado que para desarrollar un proyecto, la aplicación de una metodología no implica la desaparición de los impedimentos, simplemente permite detectarlos lo antes posible y gestionarlos de la mejor manera.

La computación en la nube uno de los conceptos más repetidos en el mundo de la informática de los últimos tiempos, siendo ya una realidad muy atractiva para las empresas como un método para mejorar su infraestructura por su versatilidad, disponibilidad, facilidad de uso y por la reducción de costes e inversión. Es por ello en definitiva que considero este proyecto enriquecedor en el plano profesional al haberme permitido adquirir experiencia en un campo en auge que tiende a expandirse.

6. Líneas futuras

Durante el desarrollo de la aplicación han surgido muchos y variados aspectos que no tenían cabida en el desarrollo del PFC, ya sea por prioridades temporales o por cuestión de establecer un límite de finalización del mismo. A continuación listaremos los más importantes:

- ✓ Se ha desarrollado un API móvil y se ha ideado y diseñado la interfaz correspondiente a la aplicación móvil de la aplicación. A fecha de entrega de la documentación de este proyecto, Nubapp tiene intención de añadir dicha funcionalidad a su aplicación móvil a pesar de no haber iniciado su desarrollo todavía por motivos de personal y prioridades.
- ✓ La interfaz general desarrollada ha sido testeada por numerosos usuarios de mi entorno y por el personal de Nubapp. Nos parece un interfaz atractivo, funcional y dinámico. No obstante es normal que durante el uso del mismo que hagan los diferentes clientes del servicio Nitro de Nubapp surjan nuevas sugerencias, dudas y cambios.
- ✓ En un futuro y si como parece, sigue teniendo interés comercial, se desea implementar un módulo para la asignación de planes dietéticos junto con las rutinas.
- ✓ Por último, actualmente estamos trabajando con el API de google para generación y lectura de códigos QR con el objetivo de utilizarlos en el gimnasio. De esta manera y gracias a la codificación QR, el monitor instalaría los códigos en las diferentes maquinas deportivas, brindando a los usuarios la posibilidad de acceder a la información de los mismos a través de su aplicación móvil.

Anexo I: Páginas web dinámicas

Una de las características más interesantes de Internet es la facilidad para publicar información. Sin embargo, esta facilidad es relativa. En los inicios del www y de las páginas HTML, cada vez que se cambiaba la información de una página, lo que se hacía era editar el fichero HTML, modificando el texto pero procurando no tocar los tags que daban forma a la página.

Sin embargo, este sistema (que aún utilizan muchos “webmasters”) es engorroso y poco práctico. Además si el contenido (textos, banners, imágenes) depende del trabajo de varias personas el coste de realización y publicación puede ser muy alto en tiempo, dinero y organización. Cuando nos encontramos con una página de estas características decimos que es una página estática, porque el contenido no se actualiza con una cierta frecuencia, sino que pueden pasar meses de una actualización a la siguiente.

Las páginas estáticas suelen tener una utilidad relativa. Una página web con la información de un Club Deportivo donde se explican cuatro reglas sobre el deporte y una dirección de contacto será útil para algunas personas, pero una vez visiten la página no volverán a ella. Sin embargo, si además de la información sobre el deporte se explica cuándo serán las próximas competiciones, la clasificación de la liga, etc. se da un valor añadido a la página que facilita que los visitantes vuelvan, y dado el caso a la hora de escoger club escojan aquél que más conocen. De alguna manera podríamos decir que aquí la clave es el servicio que se presta al usuario.

Las bases de datos, sin duda, están entre las herramientas informáticas más usadas hoy en día. En una página web dinámica la base de datos proporciona la información que se presentará al usuario, con la gran ventaja de que la información puede variar según el usuario, según la hora del día o según lo que se desee. La forma de presentar los datos es indiferente, es más, podemos tener diferentes plantillas que nos permitan presentar los mismos datos, de manera que el usuario pueda ver la página como más le guste sin perder por ello nada de información.

Hoy en día existen muchas maneras de crear páginas web dinámicas. Como ejemplos podríamos decir que Microsoft tiene su plataforma .NET, que soporta diferentes lenguajes con la tecnología ASP, Macromedia tiene su servidor ColdFusion que permite trabajar con aplicaciones ASP, PHP y JSP, y de software libre tenemos Apache con PHP o Apache con Java y JSP... son sólo ejemplos incompletos. Tecnologías y posibilidades hay muchas, sólo hay que escoger la que más nos guste o nos interese.

Pero para crear una página web dinámica no nos podemos quedar sólo con la tecnología del servidor de Internet.

Existe una segunda parte muy importante que es el gestor de bases de datos. De nuevo tenemos numerosas opciones a escoger dependiendo de nuestros intereses. Podemos hablar de Oracle, PostgreSQL, Microsoft SQL Server o MySQL, por hablar de los más conocidos. Todos ellos pueden usarse para páginas webs dinámicas o aplicaciones en Internet, aunque dependiendo de la plataforma y del lenguaje utilizado se usa más una u otra. Así, en los servidores Windows se acostumbra a usar Microsoft SQL Server, aunque también se usa MySQL como alternativa de bajo coste. En entornos Apache con programación en PHP la base de datos más utilizada, sin duda, es MySQL. De hecho se habla de la tecnología LAMP (Linux – Apache – MySQL – PHP) como una de las tecnologías con más futuro en la Red. En entornos de programación JSP (generalmente sobre Apache y Tomcat), se suele utilizar PostgreSQL o MySQL. Finalmente, una combinación clásica es la de Sun Solaris con iPlanet y Oracle, usando OC4J (Oracle Containerfor Java) como servidor de aplicaciones Java.

La creación de páginas web con contenido dinámico exige la utilización de bases de datos (que serán las encargadas de almacenar la información) y programación. Por tanto, a la hora de realizar un sitio es necesario tener claro qué queremos hacer con él para poder decidir si será dinámico o estático. Una vez tengamos claro que nos interesa un sitio dinámico (y acostumbra a ser lo más conveniente), deberemos definir qué tecnología usar y qué base de datos.

En general la tecnología Microsoft es suficientemente fiable, la programación en ASP tiene la gran ventaja de permitir la utilización de muchos lenguajes diferentes (C, Visual BASIC, Pascal, Cobol,...), y SQL Server es un gestor de bases de datos potente, flexible y fiable. En cambio, es una tecnología cara. Por otro lado, la tecnología LAMP es gratuita y Linux y Apache son posiblemente de lo mejor en sus correspondientes ámbitos.

Sin embargo, MySQL tiene algunas limitaciones (no tiene algunas funcionalidades que sí tienen SQL Server o PostgreSQL), aunque resulta muy potente y útil para prácticamente cualquier tipo de aplicación. Y en cuanto a PHP, como lenguaje de programación es un lenguaje específico para Internet (a diferencia de los lenguajes que comentaba de ASP.NET) que cada vez está teniendo más aceptación, y del que podemos encontrar numerosas aplicaciones gratuitas fácilmente adaptables a nuestras necesidades. Finalmente, la arquitectura Java (que incluye JSP, servlets, aplicaciones Java y J2EE) con la combinación de Linux, Apache, Tomcat o Jboss (para J2EE) y PostgreSQL es potente y robusta, al mismo precio que LAMP, y se usa mucho en desarrollos profesionales. Tal vez la dificultad en este caso sea que Java es un lenguaje

menos conocido y algo más complejo que los usados por los otros dos grandes sistemas.

W3C

El Consorcio World Wide Web (W3C) es una comunidad internacional donde las organizaciones miembro, personal a tiempo completo y el público en general trabajan conjuntamente para desarrollar estándares Web. Liderado por el inventor de la Web Tim Berners-Lee y el Director Ejecutivo (CEO) Jeffrey Jaffe, la misión del W3C es guiar la Web hacia su máximo potencial. La principal actividad del W3C es desarrollar protocolos y directrices que aseguren el crecimiento de la Web a largo plazo. Los estándares del W3C definen las partes claves que hacen que la World Wide Web funcione.

El equipo del W3C está distribuido por todo el mundo, pero muchas de estas personas se concentran en Cambridge, Massachusetts, Sophia-Antipolis y Tokio.

El W3C desarrolla especificaciones técnicas y directrices a través de un proceso que ha sido diseñado para maximizar el consenso sobre el contenido de un informe técnico, de forma que se pueda asegurar la alta calidad técnica y editorial, así como obtener un mayor apoyo desde el W3C y desde la comunidad en general.

Diseño y Aplicaciones Web

Incluye a los estándares para la construcción y representación de las páginas Web, incluyendo HTML5, CSS, SVG, Ajax y otras tecnologías para las Aplicaciones Web (“WebApps”). Esta sección también incluye información sobre cómo hacer páginas accesibles para personas con discapacidades (WCAG), aplicar internacionalización y trabajar sobre dispositivos móviles.

Arquitectura Web

Se centra en las tecnologías y principios fundamentales sobre los que se sostiene la Web, incluyendo URIs y HTTP. Los siguientes principios guían el trabajo del W3C:

Web para todo el mundo: El valor social que aporta la Web, es que ésta hace posible la comunicación humana, el comercio y las oportunidades para compartir conocimiento. Uno de los objetivos principales del W3C es hacer que estos beneficios estén disponibles para todo el mundo, independientemente del hardware, software, infraestructura de red, idioma, cultura, localización geográfica, o habilidad física o mental.

Web desde cualquier dispositivo: La cantidad de dispositivos diferentes para acceder a la Web ha crecido exponencialmente. Actualmente, los teléfonos móviles, teléfonos inteligentes, PDAs, sistemas de televisión interactiva, sistemas de respuesta de voz, puntos de información e incluso algunos pequeños electrodomésticos pueden acceder a la Web. Visión: La visión del W3C para la Web incluye la participación, compartir conocimiento y, de esta forma, construir confianza a gran escala.

Web de los Autores y Consumidores: La Web fue creada como una herramienta de comunicación para permitir el intercambio de información entre todo el mundo y desde cualquier lugar. Durante muchos años, para muchas personas la Web fue una herramienta de "solo lectura". Los blogs y wikis trajeron más autores a la Web y las redes sociales emergieron del próspero mercado para crear contenido y personalizar las experiencias en la Web. Los estándares del W3C han apoyado esta evolución gracias a la robusta arquitectura y a los principios de diseño.

Web de los Datos y Servicios: Algunas personas ven la Web como un repositorio gigante de datos enlazados mientras otros como un conjunto enorme de servicios que intercambian mensajes. Ambas vistas son complementarias y los requisitos de cada aplicación pueden ser los mejores determinantes para decidir que aproximación elegir para solucionar progresivamente los problemas complejos mediante tecnología Web.

Web de Confianza: La Web ha cambiado la forma en la que nos comunicamos. Al ocurrir esto, la naturaleza de nuestras relaciones sociales ha cambiado también. En la actualidad, las personas se "conocen en Internet", y llevan a cabo relaciones personales y comerciales sin haberse visto en persona anteriormente. El W3C reconoce que la confianza es un fenómeno social, pero el diseño de las tecnologías puede fomentar la confianza y la responsabilidad. A medida que cualquier actividad se hace a través de la Web, cada vez es más importante apoyar las interacciones complejas entre distintas partes alrededor del mundo.

Anexo II: JQuery y Ajax

jQuery es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. 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.

Características

- ✓ Selección de elementos DOM.
- ✓ Interactividad y modificaciones del árbol DOM, incluyendo soporte para CSS 1-3 y un plugin básico de XPath.
- ✓ Eventos.
- ✓ Manipulación de la hoja de estilos CSS.
- ✓ Efectos y animaciones.
- ✓ Animaciones personalizadas.
- ✓ AJAX.
- ✓ Soporta extensiones.
- ✓ Utilidades varias como obtener información del navegador, operar con objetos y vectores, funciones para rutinas comunes, etc.
- ✓ Compatible con los navegadores Mozilla Firefox 2.0+, Internet Explorer 6+, Safari 3+, Opera 10.6+ y Google Chrome 8+.

La característica principal de la biblioteca es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del árbol DOM y peticiones AJAX. Para ello utiliza las funciones `$()` o `jQuery()`.

Ajax

AJAX, acrónimo de *Asynchronous Java Script And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (*Rich Internet Applications*). 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, lo que significa aumentar 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. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante *XMLHttpRequest*, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores dado que está basado en estándares abiertos como JavaScript y DocumentObjectModel (DOM).

Tecnologías incluidas en Ajax

Ajax es una combinación de cuatro tecnologías ya existentes:

- ✓ XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- ✓ DocumentObjectModel (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- ✓ El objeto XMLHttpRequest para intercambiar datos de forma asíncrona con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.
- ✓ XML es el formato usado generalmente para la transferencia de datos solicitados al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML.

Como el DHTML, LAMP o SPA, Ajax no constituye una tecnología en sí, sino que es un término que engloba a un grupo de éstas que trabajan conjuntamente.

Bibliografía

- [1] Documentación relativa a XAMPP: <http://www.apachefriends.org/es/faq-xampp.html>
- [2] Manuales PHP: <http://php.net/manual/es/index.php> y <http://www.php.net/>
- [3] jQuery TOOLS: <http://jquerytools.org/documentation/>
- [4] jQuery: http://docs.jquery.com/Main_Page
- [5] W3C: <http://www.w3c.es/Divulgacion/>
- [6] MySQL: <http://dev.mysql.com/doc/refman/5.0/es/index.html>
- [7] Páginas web dinámicas: <http://www.desarrolloweb.com/manuales/7/>
- [8] Métrica v3:
http://administracionelectronica.gob.es/?_nfpb=true&_pageLabel=P60085901274201580632&langPae=es
- [9] Configuración de subdominios y workspace: <http://www.tildemark.com/setting-up-virtual-hosts-with-xampp-running-on-windows-xp/>
- [10] Git para Windows: <http://msysgit.github.com/>
- [11] Repositorio online Github: <https://help.github.com/>

«Desarrollo de un módulo para la gestión deportiva de un gimnasio dentro de la aplicación gestora de complejos deportivos Nubapp»



Javier Luquin
Jesús Villadangos

Introducción

- ❑ Empresa: Nubapp
 - ❑ Centrada en el desarrollo de aplicaciones en la nube.

- ❑ Aplicación: Nitro
 - ❑ Gestión de complejos polideportivos.
 - ❑ Reservas, usuarios, acceso, instalaciones, actividades, gimnasio, torneos, etc.
 - ❑ Tres interfaces: aplicación, página web, app móvil.

Introducción

- ❑ Módulo gimnasio
 - ❑ Aplicación:
 - ❑ Creación y edición de rutinas y ejercicios.
 - ❑ Gestión de usuarios del modulo.
 - ❑ Control del trabajo pendiente y rutinas por hacer.
 - ❑ Citas con los usuarios.
 - ❑ Comunicación con los usuarios.
 - ❑ Página web:
 - ❑ Consulta del entrenamiento por los usuarios.
 - ❑ Contacto con el monitor.
 - ❑ App móvil:
 - ❑ Desarrollo de la API del módulo
- ❑ Herramientas utilizadas para el desarrollo de la aplicación.
 - ❑ PHP, HTML, CSS y JQUERY
 - ❑ MySQL
 - ❑ GIT

Aplicación

- ❑ Video presentación de la aplicación realizada
 - ❑ Settings
 - ❑ Ejercicios
 - ❑ Crear una rutina
 - ❑ Asignar una rutina
 - ❑ Añadir trabajo pendiente
 - ❑ Historial
 - ❑ Front end y comunicación usuario monitor.

Líneas futuras

- ❑ Desarrollo de la aplicación móvil.
- ❑ Nuevas sugerencias, dudas y cambios provenientes del uso comercial de la aplicación.
- ❑ Planes dietéticos.
- ❑ Generación de códigos QR con información sobre los ejercicios.

Conclusiones

- ❑ Primera experiencia laboral dentro del mundo de la informática.
- ❑ Puesta en práctica de algunos de los conocimientos adquiridos en la carrera. Adquisición de conocimientos técnicos en lenguajes web.
- ❑ Adquisición de experiencia en un campo (aplicaciones en la nube) en auge que tiende a expandirse.

Muchas gracias

