

MEMORIA FINAL DE PROYECTO

FitPocket



CICLO FORMATIVO DE GRADO SUPERIOR

Desarrollo de Aplicaciones Web

AUTORES

Sergio,
Víctor
y
Alex

TUTOR

Guillermo

COORDINADOR

José Luis Gallego

Licencia

Esta obra esta bajo una licencia General Public License (GNU) GPL v3

Esta es la última versión de la GPL de GNU: una licencia de software libre con copyleft. Para ver una copia de esta licencia, visite

<https://www.gnu.org/licenses/quick-guide-gplv3.html>

ÍNDICE

1. INTRODUCCIÓN	5
2 ALCANCE DEL PROYECTO	5
2.1 ANÁLISIS PREVIO.....	5
3. ESTUDIO DE VIABILIDAD.....	9
3.1. ESTADO ACTUAL DEL SISTEMA	9
3.2. RESUMEN DE REQUISITOS DEL CLIENTE	9
3.3. POSIBLES SOLUCIONES	9
3.4. SOLUCIÓN ELEGIDA.....	10
3.5. PLANIFICACIÓN TEMPORAL DE LAS TAREAS DEL PROYECTO.....	10
3.6. PLANIFICACIÓN DE LOS RECURSOS A UTILIZAR.....	11
4. ANÁLISIS	12
4.1. MODELO DE DATOS.....	22
4.2. REQUISITOS FUNCIONALES	31
4.3. REQUISITOS NO FUNCIONALES	32
5. DISEÑO	33
5.1. ESTRUCTURA DE LA APLICACIÓN	33
5.2. COMPONENTES DEL SISTEMA / ARQUITECTURA DE RED	34
5.3. HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS	34
6. IMPLEMENTACIÓN.....	35
6.1. Implementación del modelo de datos	35
6.2. Carga de datos.....	35
7. EXPLOTACIÓN.....	36
7.1. PLANIFICACIÓN.....	36
8. DEFINICIÓN DE PROCEDIMIENTOS DE CONTROL Y EVALUACIÓN.....	36
9. CONCLUSIONES	36
10. FUENTES	36

INDICE de tablas e ilustraciones

Tabla 1. Definición de las funcionalidades de reservas

Tabla 2. Definición de las funcionalidades de sesiones

Tabla 3. Definición de las funcionalidades de salas

Tabla 4. Definición de las funcionalidades de actividades

Tabla 5. Definición de las funcionalidades de usuarios

Tabla 6. Definición de tareas

Tabla 7. Definición de la tabla actividad

Tabla 8. Definición de la tabla administrador.

Tabla 9. Definición de la tabla aula.

Tabla 10. Definición de la tabla historico.

Tabla 11. Definición de la tabla historico reservas.

Tabla 12. Definición de la tabla monitor.

Tabla 13. Definición de la tabla reserva.

Tabla 14. Definición de la tabla sesion.

Tabla 15. Definición de la tabla user.

Tabla 16. Definición de la tabla post.

Ilustración 1: Diagrama de casos de uso

Ilustración 2: Modelo de datos

...

1. INTRODUCCIÓN

Este documento recoge el trabajo realizado para el módulo de Proyecto del CFGS en Desarrollo de Aplicaciones Web.

Este módulo profesional complementa la formación establecida para el resto de los módulos profesionales que integran el título en las funciones de análisis del contexto, diseño o desarrollo del proyecto y organización de la ejecución.

La idea de poder ayudar a llevar una vida fitness o fisicoculturista y la del planteamiento de poder llevar este proyecto a la realidad gracias a el auge de la aparición de gimnasios. Nos pareció más que suficiente para elegir esta idea de negocio para nuestro proyecto.

La funcionalidad principal del proyecto, es la capacidad de nuestros usuarios de poder reservar un hueco para las sesiones a lo largo de la semana de diferentes actividades. A partir de ahí el administrador tiene la capacidad de generar:

- Monitores para que coordinen dichas sesiones.
- Sesiones donde tendrá cabida una actividad en un aula.
- Aulas dónde se realizarán las sesiones.
- Actividades de las diferentes disciplinas que tiene el fitness y fisicoculturismo.

Nuestros mayores inconvenientes a la realización del proyecto han sido la idea de cómo poder preservar en la base de datos las reservas, y la generación de una interfaz de usuario amigable y easy to use.

2 ALCANCE DEL PROYECTO

El proyecto a desarrollar es una aplicación web enfocada a móviles para un gimnasio. En la que cada usuario pueda reservar un hueco en las actividades que se realizan en el centro.

La definición de esta aplicación se puede resumir en gestionar de manera autónoma la reservas en las diferentes actividades.

El fin de este producto es:

-Reducir las colas en los mostradores del gimnasio: los clientes podrán realizar la reserva en una actividad desde cualquier dispositivo con acceso a internet en cualquier momento a lo largo de la semana.

Entre los objetivos de este proyecto es de que el cliente de este producto, tenga a cargo un gimnasio, donde sus clientes podrá realizar una variedad de actividades y ejercicios.

El desarrollo de éste proyecto se ha llevado a cabo en varias fases: estudio de viabilidad, análisis, diseño, implementación y pruebas, y explotación o ejecución. A continuación, se detallan las actividades/tareas/procedimientos de cada una de estas fases.

2.1 ANÁLISIS PREVIO

El afán de este proyecto junto con la limitación de tiempo, produce que la expectativa de que no completarlo al 100%, sea alta.

La ambición de este proyecto y la limitación de tiempo hacen que la probabilidad de no completarlo al 100% sea alta. En la siguiente tabla se establecen unos puntos mínimos que cumplir y sus prioridades, que hemos dividido en objetos de negocio. Entendemos como cliente a la empresa que compra o está interesada en nuestra aplicación, usuario al visitante de la parte pública de la aplicación y administrador como al usuario registrado que tiene acceso a la parte privada de la aplicación en la que se gestiona el contenido de la misma

Funcionalidad		Prioridad	Motivo
Reservas	Mostrar	Alta	La visualización de las reservas es una de las funcionalidades principales junto con la creación de estas mismas.
	Crear	Alta	La creación de las reservas es una de las funcionalidades principales del usuario. Debido a que, cuando quiera va a poder reservar una plaza en la actividad que desee.
	Modificar	Media	Las reservas están diseñadas para que sean exclusivamente modificadas cuando el usuario quiere dar de baja la reserva.
	Borrar	Baja	Por lo explicado anteriormente, esta funcionalidad no es realmente necesaria.

Tabla 1. Definición de las funcionalidades de reservas

Funcionalidad		Prioridad	Motivo
Sesiones	Mostrar	Alta	Para la realización de una reserva, es necesario anteriormente haber echado un vistazo al catálogo de sesiones de las diferentes actividades.
	Crear	Alta	Para el correcto funcionamiento de la funcionalidad principal de la aplicación es imprescindible que previamente se haya generado unas sesiones.
	Modificar	Baja	Esta funcionalidad no es de vital para la el funcionamiento de esta aplicación, debido a que una sesión una vez creada puede que no necesite modificación de la misma
	Borrar	Media	La eliminación de las sesiones una función necesaria en caso de haberse equivocado a la hora de generación de la sesión.

Tabla 2. Definición de las funcionalidades de sesiones

Funcionalidad		Prioridad	Motivo
Salas	Mostrar	Alta	Las salas deben ser consultables para poder ser utilizadas en las sesiones.
	Crear	Alta	Para que la aplicación sea funcional tiene que tener la capacidad de crear nuevas salas.
	Modificar	Baja	La modificación de las salas resulta poco útil debido a que solamente será de utilidad cuando nos hayamos equivocado a la hora de creación de la sala o en caso de que hubiera una reforma en el gimnasio.
	Borrar	Media	La eliminación de salas es necesaria debido a que en caso de equivocación generando salas, sería útil poder eliminarla para generarla de nuevo.

Tabla 3. Definición de las funcionalidades de salas

Funcionalidad		Prioridad	Motivo
Actividades	Mostrar	Alta	Para que el usuario pueda consultar las diversas actividades del centro y que el administrador pueda consultarlas para creación de nuevas sesiones, esta funcionalidad es vital.
	Crear	Alta	Para que se puedan crear sesiones con actividades que no se han realizado anteriormente esta funcionalidad es de importancia.
	Modificar	Media	La modificación de las actividades es necesaria en caso de que se haya generado la actividad con información errónea o escasa.
	Borrar	Baja	La eliminación de las actividades es necesaria en el caso de que se deje impartir la actividad en el centro. Aunque esto no es de alta importancia debido a que el mayor problema que puede generar la no eliminación de la actividad es de que ocupe espacio en disco

Tabla 4. Definición de las funcionalidades de actividades

Funcionalidad		Prioridad	Motivo
Usuarios	Mostrar	Alta	Funcionalidad básica si queremos que un usuario acceda a sus datos o se identifique ante el sistema.
	Crear	Alta	Al poder registrarse, es posible ofrecer los servicios de la aplicación.
	Admin - >Modificar(Activación y actualización de los diferentes campos)	Alta	Los usuarios se pueden registrar, pero no pueden loguearse en la aplicación hasta que previamente un administrador haya activado el usuario confirmando que es un usuario real del gimnasio. Debido a esto y la probabilidad de que el administrador deba modificar algún dato del usuario hace a esta funcionalidad de alta importancia.
	Usuario - >Modificar (Modificación datos perfil y acceso)	Alta	Dentro de nuestra aplicación es de relevancia que el usuario pueda modificar sus datos. Son los datos del usuario y tiene derecho a configurarlos a gustos.
	Borrar	Media	La opción de eliminar un usuario tiene que estar alcance del propio usuario como del administrador.

Tabla 5. Definición de las funcionalidades de usuarios

3. ESTUDIO DE VIABILIDAD

En esta fase se considera si el proyecto se puede realizar teniendo en cuenta las circunstancias internas y externas de la empresa, las diferentes soluciones posibles y los recursos de los cuales se dispone.

Para ello se hace una valoración del estado actual del sistema y de los requisitos del cliente, se presentará un estudio de soluciones alternativas y la solución elegida por el cliente.

3.1. ESTADO ACTUAL DEL SISTEMA

Estado actual sobre aplicaciones de gimnasios:

Actualmente diversas aplicaciones en relación con la gestión de gimnasios. (Como por ejemplo IsMyGym, Virtuagym, gayatrisoft...) Estas son plataformas que ofrecen sus servicios a otras empresas, tienen buena valoración y demuestran que tienen una gran clientela.

Ofrecen servicios como:

- RESERVAS ONLINE (A través de la web o app móvil)
- PAGO ON-LINE (Tus socios podrán pagar mediante Paypal/Tarjeta de crédito)
- HORARIOS (Consulta y modifica los horarios fácilmente)
- ACTIVIDADES (Da a conocer tus clases y en qué consisten)
- ZONA PRIVADA (Un espacio personalizado para tus trabajadores)
- LISTAS DE ESPERA (Configura listas de espera para tus actividades reservables)
- BONOS DE RESERVAS (Asigna a tus socios bonos de reservas para distintas actividades)
- FACTURAS (Sistema de generación de facturas)
- VENTAS (A través de la app)
- WEB (Puesta en marcha de una web personalizada para cada centro)

3.2. RESUMEN DE REQUISITOS DEL CLIENTE

El cliente busca:

-Una forma de que los clientes reserven un hueco en sus actividades favoritas, así evitando cola en recepción del centro deportivo. Ya sea mediante el uso de algún móvil, ordenador o tabletas por parte del usuario o a disposición de los usuarios en el centro deportivo.

-Un muro donde los clientes puedan postear comentarios, actividades que han realizado con sus compañeros.

-Una vista de las actividades que se practican en el centro, con una corta descripción.

3.3. POSIBLES SOLUCIONES

Las soluciones actuales pasan por:

-Usar algunas de las plataformas anteriormente mencionadas.

Ofrecen gran cantidad de servicios, pruebas gratuitas del programa y unos estilos agradables para este tipo de entornos. A partir del mes de la prueba gratuita, empezarían a cobrarnos una cuota mensual. Hay otras plataformas que son gratuitas lo único que aplican una comisión por cada alta de usuario. Parece ser una buena alternativa en caso de tener una inversión de dinero baja.

-Coger alguna aplicación opensource ya desarrollada y avanzar a partir de ahí. (Como ClubMaster, Gymie)

Esta posibilidad no ofrece tantos servicios como las plataformas de pago, pero si son susceptibles a cambios a gusto del cliente, siempre que tenga un mínimo de desarrolladores capaz de ello. Hay algunas de estas que tienen tanto soporte de pago, como soporte gratuito mediante “Issues” del proyecto en github. Dónde el soporte no será ni 100% fiable ni inmediato.

-El uso de una web dedicada y personalizada a los servicios que el cliente solicita.

Esta solución ofrece una personalización total a requisitos del cliente. Lo que eso conlleva es tener a un equipo de desarrolladores de aplicaciones web. El desarrollo, el cuál podría ser de corta duración (2-3 semanas), en cambio sí se juntan con las prácticas podría alargarse hasta los 3 meses.

3.4. SOLUCIÓN ELEGIDA

La solución elegida es, el uso de una web dedicada y personalizada a los servicios que el cliente solicita.

Debido a que:

-Las plataformas de pago quedan excluidas de nuestras posibilidades.

-El desconocimiento sobre diferentes lenguajes de programación de los proyectos open source, sumado al aprendizaje de estos mismos y el entendimiento de estos proyectos para continuar con su desarrollo, nos hacen descartar esta posibilidad.

De aquí sale la propuesta de crear un nuevo proyecto denominado “FitPocket”

3.5. PLANIFICACIÓN TEMPORAL DE LAS TAREAS DEL PROYECTO

En cuanto a la planificación temporal de las tareas sobre este proyecto, las hemos dividido y repartido entre el equipo de trabajo:

Tareas		
CRUD sesiones	CRUD usuarios	CRUD monitores
CRUD administradores	Documentación	Login/Registro
Incorporación de AJAX en login y registros	Front login user/admin/monitor	Vista diaria de las sesiones a lo largo de una

		semana
Vista de reservas	Vista del histórico de reservas	Vista del administrador

Tabla 6. Definición de tareas

Cada una de estas tareas puede prolongarse de una semana a cuatro semanas dependiendo de la complejidad de cada tarea. Para la repartición de las tareas, los involucrados del proyecto cogerán nuevas tareas en cuanto vayan acabando las previamente cogidas. Se tratará de hacer una repartición del trabajo de forma equitativa, aunque es una cosa difícil de llevar a cabo.

Para ello hemos utilizado una herramienta que se denomina GitKraken Glo, la cual permite crear iteraciones, organizarlas en columnas, asignar un responsable, introducir detalles, imágenes, tareas a realizar, fechas límite... Esta aplicación también tiene la capacidad de controlar las ramas git del repositorio dónde esté alojado nuestro proyecto, lo cual facilita el trabajo con git, si no estás del todo acostumbrado a trabajar con esta herramienta.

3.6. PLANIFICACIÓN DE LOS RECURSOS A UTILIZAR

Para solventar los problemas que plantea el proyecto FitPocket puede ser necesario:

- Contratar personal, sale gratis por los 3 voluntarios del módulo de desarrollo de aplicaciones web que han querido hacer este proyecto.
- Compra de equipos, para el hosting de este proyecto para el desarrollo de manera temporal cada uno de los desarrolladores se montará un servidor local. Para realizar una muestra cara el cliente, el alquiler de un servidor con las capacidades necesarias para alojar el proyecto y la compra de un dominio, sería más que suficiente.

4. ANÁLISIS

Para poder sacar el máximo rendimiento a esta aplicación es necesario que el gimnasio disponga de un equipo ya sea móvil, tableta u ordenador con un software interno compatible con la aplicación del centro.

Diagrama de casos de uso.

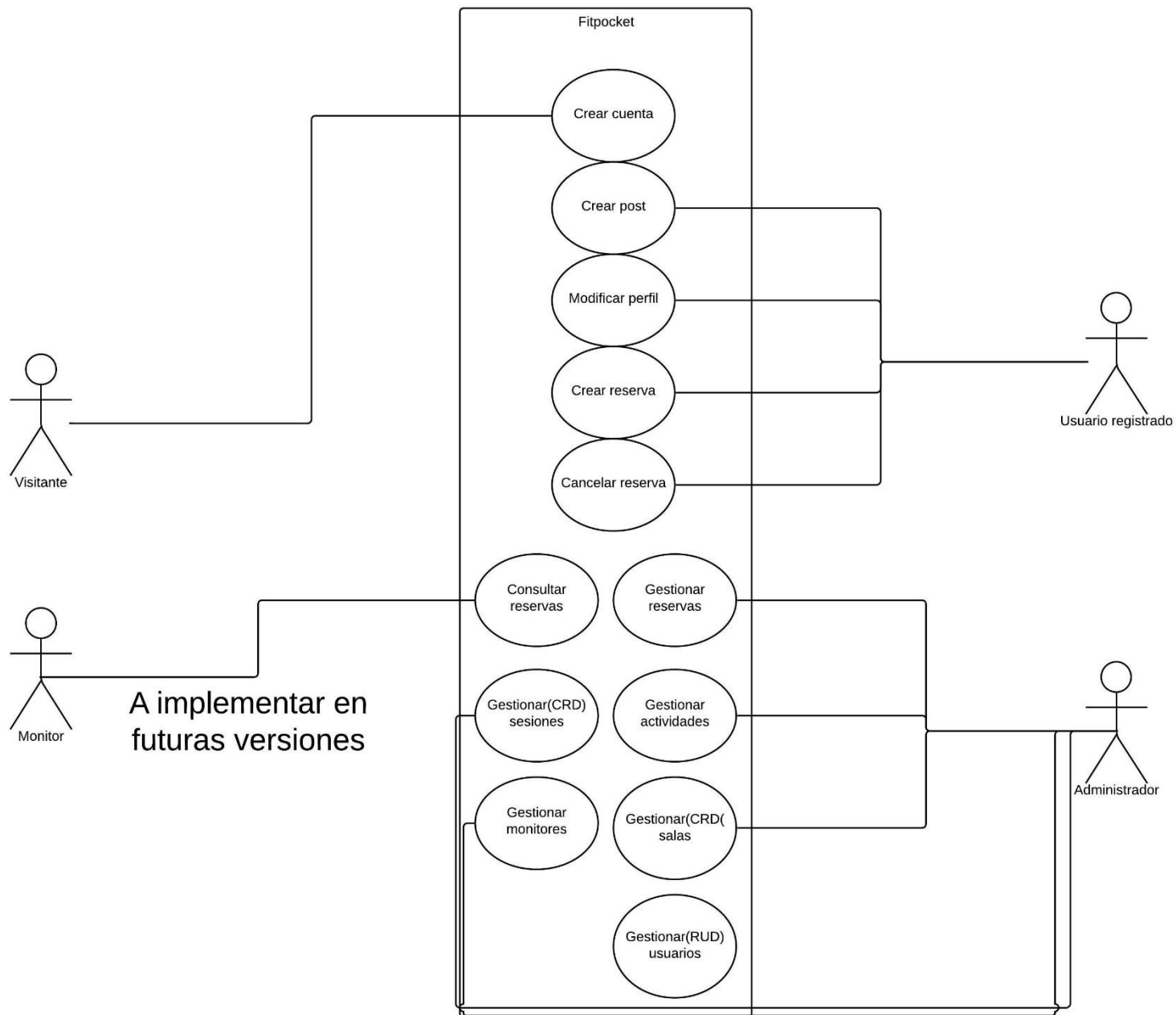


Ilustración 1. Diagrama de casos de uso

Casos de uso:-Casos de uso UC001

Escenario de éxito: El visitante crea una cuenta

Actores: Visitante

Precondición: El visitante no debe estar autenticado en el sistema

1. El visitante indica al sistema que quiere registrarse.
2. El sistema muestra el formulario de registro.
3. El visitante rellena el formulario de registro.
4. El sistema registra los datos introducidos.

Extensiones

- 3a. La contraseña introducida no cumple los requisitos de seguridad.
 1. El sistema indica que la contraseña introducida no es válida y muestra los requisitos mínimos.
 2. El visitante introduce una nueva contraseña.
 3. Se repiten 1 y 2 de esta extensión hasta que la contraseña sea válida.
 4. Continúa en el punto 4.
- 3b. La dirección de email introducido ya está registrada.
 1. El sistema indica que el email introducido ya está en uso.
 2. El visitante introduce nueva dirección de correo.
 3. Se repiten los puntos 1 y 2 de esta extensión hasta que la dirección de email sea una no utilizada.
 4. Continúa en el punto 4.
- 3c. La dirección de email introducida no cumple con los requisitos de correo.
 1. El sistema indica que el email introducido no es válido y muestra los requisitos.
 2. El visitante introduce una nueva contraseña.
 3. Se repiten 1 y 2 de esta extensión hasta que la contraseña sea válida.
 4. Continúa el punto 4.

-Casos de uso UC002

Escenario de éxito: El usuario modifica su perfil.

Actores: Usuario registrado

Precondición: Estar previamente acreditado ante en el sistema

1. El usuario indica al sistema que quiere modificar su perfil
2. El sistema muestra el formulario con los datos de perfil del usuario
3. El usuario cumplimenta el formulario con los datos correspondientes.
4. El usuario confirma los datos agregados.
5. El sistema registra los datos introducidos.

Extensiones:

- 4a. Los datos introducidos no son válidos
 1. Repetir puntos 2 y 3, hasta introducción de datos válidos.
 2. Continúa en punto 5.

-Casos de uso UC003

Escenario de éxito: El usuario crea reserva

Actores: Usuario registrado

Precondición: Estar previamente acreditado ante el sistema, haber consultado previamente la sesión para saber para cual quiere reservar y saber que no tiene reserva ya hecha.

1. El usuario indica al sistema que quiere consultar las sesiones de un día.

2. El sistema muestra las sesiones del día
3. El usuario indica al sistema que quiere reservar una plaza en una sesión
4. El sistema pide confirmación al cliente
5. El usuario confirma la reserva
6. El sistema registra la reserva

Extensiones:

- 5ª. El usuario cancela en la confirmación de la reserva
 1. Repetir pasos 2, si es necesario y si no 3 y 4.
 2. Continúa en el punto 6.

-Casos de uso UC004

Escenario de éxito: El usuario cancela reserva

Actores: Usuario registrado

Precondición: Estar previamente acreditado ante el sistema, haber consultado previamente las reservas para saber para cual quiere cancelar.

1. El usuario indica al sistema que quiere consultar sus reservas.
2. El sistema muestra las reservas.
3. El usuario indica que quiere cancelar una reserva.
4. El sistema pide confirmación al usuario.
5. El usuario confirma la cancelación de reserva.
6. El sistema registra la cancelación de reserva.

Extensiones:

- 5ª. El usuario cancela en la confirmación de cancelar reserva.
 1. Repetir puntos 3 y 4.
 2. Continúa en el punto 6.

-Casos de uso UC005

Escenario de éxito: El usuario reactiva reserva cancelada

Actores: Usuario registrado

Precondición: Estar previamente acreditado ante el sistema, haber consultado previamente las reservas para saber para cual quiere reactivar.

1. El usuario indica al sistema que quiere consultar sus reservas.
2. El sistema muestra las reservas.
3. El usuario indica que quiere reactivar una reserva cancelada.
4. El sistema pide confirmación al usuario.
5. El usuario confirma la reactivación de la reserva cancelada.
6. El sistema registra la reactivación de la reserva cancelada.

Extensiones:

- 5ª. El usuario cancela en la confirmación de reactivación de la reserva cancelada.
 3. Repetir puntos 3 y 4.
 4. Continúa en el punto 6.

-Casos de uso UC006

Escenario de éxito: El usuario crea post

Actores: Usuario registrado

Precondición: Estar previamente acreditado ante el sistema.

1. El usuario indica al sistema que quiere crear un post
2. El sistema muestra un formulario de creación del post.
3. El usuario cumplimenta el formulario.

4. El usuario indica al sistema que quiere publicar el post.

-Casos de uso UC007

Escenario de éxito: El usuario consulta sesiones

Actores: Usuario registrado

Precondición: Estar previamente acreditado ante el sistema, conocimiento de que día quiere consultar sesión.

1. El usuario indica al sistema que quiere consultar sesiones.
2. El sistema muestra las reservas del primer día de la semana.

Extensiones:

3. El usuario indica al sistema que quiere ver las sesiones de un día que no es el primer día de la semana.
4. El sistema muestra las sesiones del día seleccionado por el usuario.

-Casos de uso UC008

-UC008

Escenario de éxito: El usuario consulta reservas, canceladas o no.

Actores: Usuario registrado

Precondición: Estar previamente acreditado ante al sistema y tenencia de alguna reserva realizada o cancelada.

1. El usuario indica al sistema que quiere consultar las reservas
2. El sistema muestra las reservas del usuario

-Casos de uso UC009

-UC009

Escenario de éxito: El usuario consulta reservas que están en el histórico.

Actores: Usuario registrado

Precondición: Estar previamente acreditado ante el sistema y tenencia de alguna reserva en el histórico.

1. El usuario indica al sistema que quiere consultar las reservas
2. El sistema muestra las reservas del usuario
3. El usuario indica al sistema que quiere consultar las reservas en el histórico

-Casos de uso UC010

-UC010

Escenario de éxito: El usuario consulta las actividades.

Actores: Usuario registrado

Precondición: Estar previamente acreditado ante el sistema.

1. El usuario indica al sistema que quiere consultar las actividades
2. El sistema muestra las actividades

-Casos de uso UC011

-UC011

Escenario de éxito: El administrador crea un administrador.

Actores: Administrador

Precondición: Estar previamente autenticado ante el sistema.

1. El administrador indica al sistema que quiere crear un nuevo administrador.
2. El sistema muestra un formulario con los datos necesarios.
3. El administrador indica al sistema los datos necesarios y confirma el modal.

4. El sistema registra un nuevo administrador con los datos dados.

Extensiones

- 4^a. El sistema indica al administrador que hay campos que no cumplen los requisitos
 1. Repetir punto 3 hasta rellenar con datos correctos.
 2. Continúa en punto 4.

-Casos de uso UC012

-UC012

Escenario de éxito: El administrador modifica un administrador

Actores: Administrador

Precondición: Estar previamente autenticado ante el sistema, conocimiento del administrador que se quiere modificar.

1. El administrador indica al sistema que quiere modificar un administrador.
2. El sistema muestra un formulario a cumplimentar con los datos del administrador.
3. El administrador rellena el formulario y confirma el registro.
4. El sistema registra los cambios efectuados.

Excepciones:

- 4^a. El sistema indica que los que hay cambios que no cumplen los requisitos.
 1. Repetir punto 3 hasta rellenar con datos correctos.
 2. Continúa en punto 4.

-Casos de uso UC013

-UC013

Escenario de éxito: El administrador elimina un administrador

Actores: Administrador

Precondición: Estar previamente autenticado ante el sistema, conocimiento del Administrador a eliminar.

1. El administrador indica al sistema que quiere eliminar un administrador.
2. El sistema pide confirmación al administrador.
3. El administrador confirma la eliminación.
4. El sistema elimina el administrador.

Excepciones

- 3^a. El administrador cancela la confirmación.
 1. Repetir puntos 1,2, hasta confirmación afirmativa.
 2. Continúa en punto 4.

-Casos de uso UC014

-UC014

Escenario de éxito: El administrador consulta los administradores.

Actores: Administrador

Precondición: Estar previamente autenticado ante el sistema.

1. El administrador indica al sistema que quiere consultar los administradores
2. El sistema muestra los administradores

-Casos de uso UC015

-UC015

Escenario de éxito: El administrador consulta los usuarios.

Actores: Administrador

Precondición: Estar autenticado ante el sistema.

1. El administrador indica que quiere consultar los usuarios.
2. El sistema muestra los usuarios.

-Casos de uso UC016-UC016

Escenario de éxito: El administrador modifica un usuario

Actores: Administrador

Precondición: Estar autenticado previamente ante el sistema y tener conocimiento del usuario que se quiere modificar.

1. El administrador indica al sistema que quiere consultar los usuarios.
2. El sistema muestra los usuarios.
3. El administrador indica al sistema que usuario quiere modificar.
4. El sistema muestra un formulario con los datos del usuario.
5. El administrador rellena el formulario con los datos correspondientes y confirma la modificación.
6. El sistema registra los cambios del usuario.

Excepciones:

- 6ª. El sistema indica que los datos introducidos son inválidos.
 1. Repetir puntos 4 y 5, hasta inserción correcta.
 2. Continúa en punto 6.

-Casos de uso UC017-UC00X

Escenario de éxito: El administrador elimina un usuario.

Actores: Administrador

Precondición: Estar autenticado previamente ante el sistema y tener conocimiento del usuario que se quiere eliminar

1. El administrador indica al sistema que quiere consultar los usuarios.
2. El sistema muestra los usuarios.
3. El administrador indica al sistema que usuario quiere eliminar.
4. El sistema muestra una confirmación a espera del administrador
5. El administrador confirma la eliminación del usuario
6. El sistema registra la eliminación del usuario

Excepciones:

- 5ª. El administrador cancela en la confirmación
 1. Repetir puntos 3y 4, hasta confirmar afirmativamente a la eliminación del usuario.
 2. Continúa en punto 6.

-Casos de uso UC018-UC018

Escenario de éxito: El administrador activa/desactiva un usuario

Actores: Administrador

Precondición: Estar autenticado ante el sistema y tener conocimiento del usuario que se quiere activar.

1. El administrador indica al sistema que quiere consultar los usuarios.
2. El sistema muestra los usuarios.
3. El administrador indica al sistema que usuario quiere activar/desactivar.
4. El sistema registra la eliminación del usuario.

-Casos de uso UC019

-UC019

Escenario de éxito: El administrador consulta los monitores

Actores: Administrador

Precondición: Estar previamente autenticado ante el sistema.

1. El administrador indica al sistema que quiere consultar los monitores
2. El sistema muestra los monitores

-Casos de uso UC020**-UC020**

Escenario de éxito: El administrador crea un monitor

Actores: Administrador

Precondición: Estar previamente autenticado ante el sistema.

1. El administrador indica al sistema que quiere crear un nuevo monitor.
2. El sistema muestra un formulario con los datos necesarios.
3. El administrador indica al sistema los datos necesarios y confirma el modal.
4. El sistema registra un nuevo monitor con los datos dados.

Extensiones

- 4^a. El sistema indica al administrador que hay campos que no cumplen los requisitos
 1. Repetir punto 3 hasta rellenar con datos correctos.
 2. Continúa en punto 4.

-Casos de uso UC021**-UC021**

Escenario de éxito: El administrador modifica un monitor

Actores: Administrador

Precondición: Estar autenticado previamente ante el sistema y tener conocimiento del monitor que se quiere modificar.

1. El administrador indica al sistema que quiere consultar los monitores.
2. El sistema muestra los monitores.
3. El administrador indica al sistema que monitor quiere modificar.
4. El sistema muestra un formulario con los datos del monitor.
5. El administrador rellena el formulario con los datos correspondientes y confirma la modificación.
6. El sistema registra los cambios del monitor.

Excepciones:

- 6^a. El sistema indica que los datos introducidos son inválidos.
 1. Repetir puntos 4 y 5, hasta inserción correcta.
 2. Continúa en punto 6.

-Casos de uso UC022**-UC022**

Escenario de éxito: El administrador elimina un monitor

Actores: Administrador

Precondición: Estar autenticado previamente ante el sistema y tener conocimiento del monitor que se quiere eliminar

1. El administrador indica al sistema que quiere consultar los monitores.
2. El sistema muestra los monitores.
3. El administrador indica al sistema que monitor quiere eliminar.
4. El sistema muestra una confirmación a espera del administrador.
5. El administrador confirma la eliminación del monitor.

6. El sistema registra la eliminación del monitor.

Excepciones:

- 5ª. El administrador cancela en la confirmación.

1. Repetir puntos 3y 4, hasta confirmar afirmativamente a la eliminación del usuario.

2. Continúa en punto 6.

-Casos de uso UC023

-UC023

Escenario de éxito: El administrador consulta actividades

Actores: Administrador

Precondición: Estar previamente autenticado ante el sistema.

1. El administrador indica al sistema que quiere consultar las actividades
2. El sistema muestra las actividades

-Casos de uso UC024

-UC024

Escenario de éxito: El administrador crea una actividad

Actores: Administrador

Precondición: Estar previamente autenticado ante el sistema.

1. El administrador indica al sistema que quiere crear una nueva actividad.
2. El sistema muestra un formulario con los datos necesarios.
3. El administrador indica al sistema los datos necesarios y confirma el modal.
4. El sistema registra un nuevo administrador con los datos dados.

Extensiones

- 4ª. El sistema indica al administrador que hay campos que no cumplen los requisitos
 1. Repetir punto 3 hasta rellenar con datos correctos.
 2. Continúa en punto 4.

-Casos de uso UC025

-UC025

Escenario de éxito: El administrador modifica una actividad

Actores: Administrador

Precondición: Estar autenticado previamente ante el sistema y tener conocimiento de la actividad que se quiere modificar.

1. El administrador indica al sistema que quiere consultar las actividades.
2. El sistema muestra las actividades
3. El administrador indica al sistema que actividad quiere modificar.
4. El sistema muestra un formulario con los datos de la actividad.
5. El administrador rellena el formulario con los datos correspondientes y confirma la modificación.

6. El sistema registra los cambios de la actividad.

Excepciones:

- 6ª. El sistema indica que los datos introducidos son inválidos.
 1. Repetir puntos 4 y 5, hasta inserción correcta.
 2. Continúa en punto 6.

-Casos de uso UC026

-UC026

Escenario de éxito: El administrador elimina una actividad

Actores: Administrador

Precondición: Estar autenticado previamente ante el sistema y tener conocimiento de la actividad que se quiere eliminar

1. El administrador indica al sistema que quiere consultar las actividades.
2. El sistema muestra las actividades.
3. El administrador indica al sistema que actividad quiere eliminar.
4. El sistema muestra una confirmación a espera del administrador.
5. El administrador confirma la eliminación de la actividad.
6. El sistema registra la eliminación de la actividad.

Excepciones:

- 5ª. El administrador cancela en la confirmación.
 1. Repetir puntos 3y 4, hasta confirmar afirmativamente a la eliminación del usuario.
 2. Continúa en punto 6.

-Casos de uso UC027

-UC027

Escenario de éxito: El administrador consulta las salas.

Actores: Administrador

Precondición: Estar previamente autenticado ante el sistema.

1. El administrador indica al sistema que quiere consultar las salas.
2. El sistema muestra las salas.

-Casos de uso UC028

-UC028

Escenario de éxito: El administrador crea una sala

Actores: Administrador

Precondición: Estar previamente autenticado ante el sistema.

1. El administrador indica al sistema que quiere crear una nueva sala.
2. El sistema muestra un formulario con los datos necesarios.
3. El administrador indica al sistema los datos necesarios y confirma el modal.
4. El sistema registra un nuevo administrador con los datos dados.

Extensiones

- 4ª. El sistema indica al administrador que hay campos que no cumplen los requisitos
 1. Repetir punto 3 hasta rellenar con datos correctos.
 2. Continúa en punto 4.

-Casos de uso UC029

-UC030

Escenario de éxito: El administrador modifica una sala

Actores:

Precondición:

-Casos de uso UC030

-UC030

Escenario de éxito: El administrador elimina una sala

Actores: Administrador

Precondición: Estar autenticado previamente ante el sistema y tener conocimiento de la sala que se quiere eliminar.

1. El administrador indica al sistema que quiere consultar las salas.

2. El sistema muestra las salas.
3. El administrador indica al sistema que sala quiere eliminar.
4. El sistema muestra una confirmación a espera del administrador
5. El administrador confirma la eliminación de la sala.
6. El sistema registra la eliminación de la sala.

Excepciones:

- 5ª. El administrador cancela en la confirmación
 1. Repetir puntos 3y 4, hasta confirmar afirmativamente a la eliminación del usuario.
 2. Continúa en punto 6.

-Casos de uso UC031

-UC031

Escenario de éxito: El administrador consulta las sesiones

Actores: Administrador

Precondición: Estar previamente autenticado ante el sistema.

1. El administrador indica al sistema que quiere consultar las salas
2. El sistema muestra las salas.
3. El administrador indica al sistema que sesiones quiere visualizar de las diversas salas.
4. El sistema muestra las sesiones del primer día de la semana.

Excepciones:

5. El administrador indica que quiere consultar las sesiones de un día diferente.

-Casos de uso UC032

-UC031

Escenario de éxito: El administrador crea una sesión

Actores: Administrador

Precondición: Estar previamente autenticado ante el sistema.

1. El administrador indica al sistema que quiere consultar las salas
2. El sistema muestra las salas.
3. El administrador indica al sistema que sesiones quiere visualizar de las diversas salas.
4. El sistema muestra las sesiones del primer día de la semana.
5. El administrador indica al sistema que quiere agregar una sesión
6. El sistema muestra el formulario.
7. El administrador rellena el formulario con los datos correspondientes y confirma.
8. El sistema registra la nueva sesión.

Excepciones:

- 5ª. El administrador indica que quiere consultar las sesiones de un día diferente.
 1. Continúa en el punto 5
- 8ª. El sistema informa que los datos introducidos en el formulario son incorrectos.
 1. Repetición del punto 7, hasta inserción correcta de datos.

4.1. MODELO DE DATOS

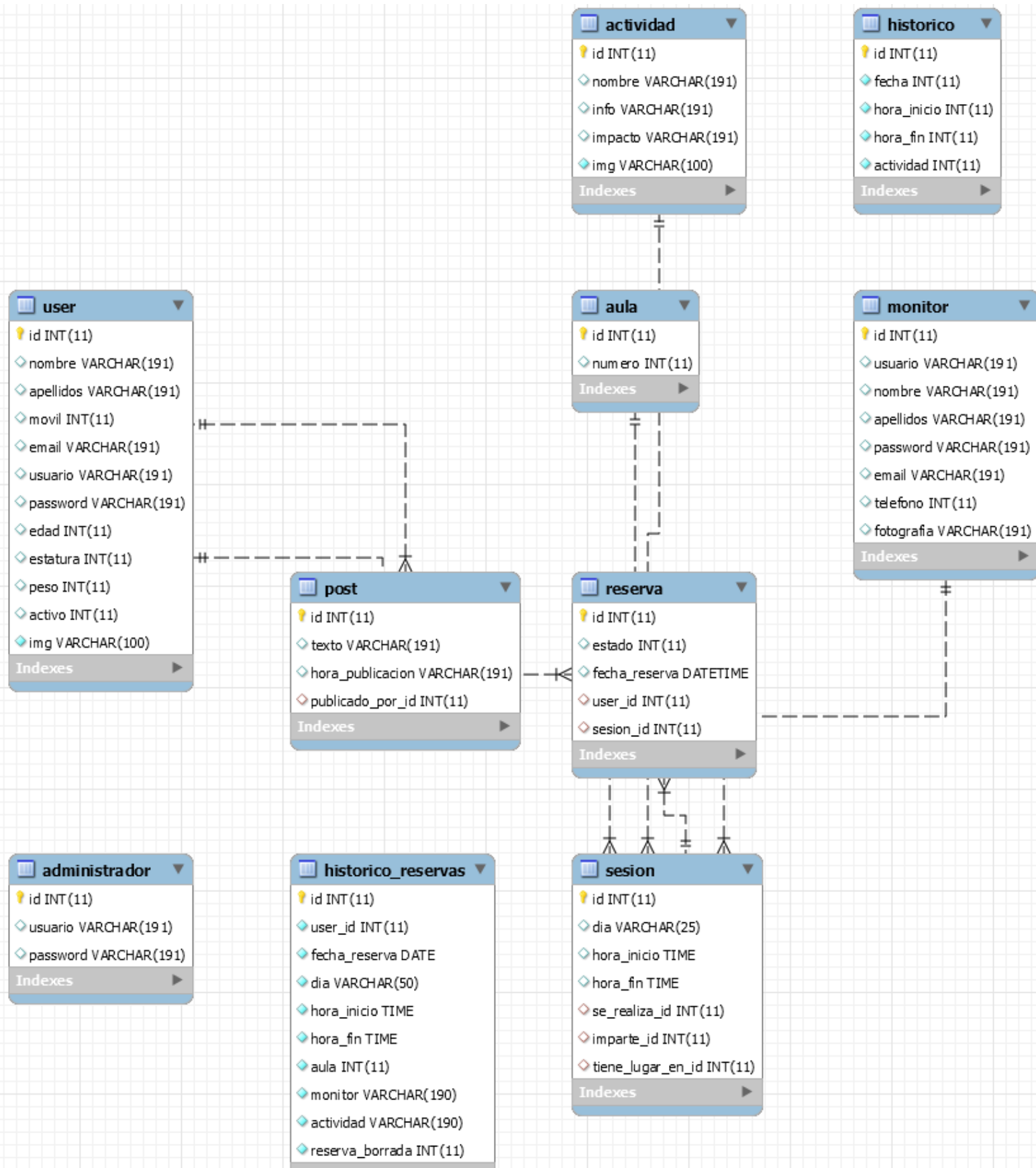


Ilustración 2. Modelo de datos

DEFINICIÓN DE TABLAS

Definición de tablas ordenadas por orden alfabético:

-Tabla actividad

Actividad				
Almacena los datos de las actividades registradas.				
Campo	Tipo	Descripción	Restricciones	Nullable
id	INT(11)	Identificador de la actividad. Auto-generado	Primary key	no
Nombre	VARCHAR(191)	Nombre de la actividad.		no
Info	VARCHAR(191)	Descripción corta sobre la actividad.		no
Impacto	VARCHAR(191)	Nivel de exigencia de la actividad		no
Img	VARCHAR(100)	Ruta de acceso a la imagen vinculada al ejercicio.		no

Tabla 7. Definición de la tabla actividad.

-Tabla administrador

Administrador				
Almacena los datos de los administradores				
Campo	Tipo	Descripción	Restricciones	Nullable
id	INT(11)	Identificador de la actividad. Auto-generado	Primary key	no
usuario	VARCHAR(191)	Nombre del administrador con el que acceder al panel del control.		no
password	VARCHAR(191)	Contraseña de acceso al administrador cifrada con md5.		no

Tabla 8. Definición de la tabla administrador.

-Tabla aula

Aula				
Almacena los datos de las aulas.				
Campo	Tipo	Descripción	Restricciones	Nullable
id	INT(11)	Identificador de la actividad. Auto-generado	Primary key	no
numero	INT(11)	Número asociado a la aula.		no

Tabla 9. Definición de la tabla aula.

-Tabla historico

Histórico				
Almacena los datos de las sesiones que se han realizado.				
Campo	Tipo	Descripción	Restricciones	Nullable
id	INT(11)	Identificador de la actividad. Auto-generado	Primary key	no
fecha	INT(11)	Fecha en la que se ha realizado la sesión.		no
hora_inicio	INT(11)	Data la hora de inicio de la sesión		no
hora_fin	INT(11)	Data la hora de fin de la sesión		no
actividad	int(11)	Relaciona el histórico de una sesión con una actividad		

Tabla 10. Definición de la tabla historico.

-Tabla historico reservas

historico_reservas				
Almacena los datos de las sesiones que se han realizado.				
Campo	Tipo	Descripción	Restricciones	Nullable
id	INT(11)	Identificador de la actividad. Auto-generado	Primary key	no
user_id	INT(11)	Identificador del usuario que ha realizado la reserva.		no
fecha_reserva	DATE	Fecha de creación de la reserva de una plaza en una sesión.		no
dia	VARCHAR(50)	Data la hora de fin de la sesión		no
hora_inicio	TIME			no
hora_fin	TIME			no
aula	INT(11)	Identificador de la aula, dónde se va a realizar la sesión.		no
monitor	VARCHAR(190)			no
actividad	VARCHAR(190)			no
reserva_borrada	INT(11)			

Tabla 11. Definición de la tabla historico_reservas.

-Tabla monitor

monitor				
Almacena los datos relativos al monitor.				
Campo	Tipo	Descripción	Restricciones	Nullable
id	INT(11)	Identificador del monitor. Auto-generado	Primary key	no
usuario	VARCHAR(191)	Nombre de usuario con el que identificarse ante el sistema.		no
nombre	VARCHAR(191)	Nombre real del monitor en cuestión.		no
apellidos	VARCHAR(191)	Apellidos del monitor en cuestión.		no
password	VARCHAR(191)	Contraseña de acceso para identificarse ante el sistema. Cifrada en md5.		no
email	VARCHAR(191)	Correo asociado al monitor.		no
telefono	INT(11)	Número de teléfono asociado al monitor	Unique	no
fotografia	VARCHAR(191)	Ruta de acceso a la fotografía asociada al monitor.		

Tabla 12. Definición de la tabla monitor.

-Tabla reserva

Reserva				
Almacena los datos de las reservas realizadas por los usuarios.				
Campo	Tipo	Descripción	Restricciones	Nullable
id	INT(11)	Identificador de la reserva. Auto-generado	Primary key	no
estado	INT(11)	Estado de la reserva.		no
fecha_reserva	DATETIME			
user_id	INT(11)	Relación de la reserva con el identificador de un usuario en concreto.	Foreing_key id(usuario)	no
sesion_id	INT(11)	Relación de la reserva con el identificador de una sesión.	Foreing_key id(sesion)	no

Tabla 13. Definición de la tabla reserva.

-Tabla sesion

Sesion				
Almacena los datos de las sesiones.				
Campo	Tipo	Descripción	Restricciones	Nullable
id	INT(11)	Identificador de la reserva. Auto-generado	Primary key	no
dia	VARCHAR(25)	Fecha de realización de la sesión.		
hora_inicio	TIME	Hora de inicio de la sesión		
hora_fin	TIME	Hora de finalización de la sesión		
se_realiza_id	INT(11)	Identificado de la actividad que se imparte en la sesión.	Foreing key id (actividad)	no
imparte_id	INT(11)	Identificador del monitor que imparte la sesión	Foreing key id(monitor)	no
tiene_lugar_en_id	INT(11)	Identificador de la sala en la que se imparte la sesión	Foreing key id (sala)	no

Tabla 14. Definición de la tabla sesion.

-Tabla user

User				
Almacena los datos de los usuarios registrados				
Campo	Tipo	Descripción	Restricciones	Nullable
id	INT(11)	Identificador de la reserva. Auto-generado	Primary key	no
email	VARCHAR(191)	Email identificador del usuario y con la contraseña forma parte de los datos de acceso.		no
password	VARCHAR(191)	Contraseña del usuario para identificarse ante el sistema. Esta encriptada en md5.		no
nombre	VARCHAR(191)	Nombre del usuario registrado.		no
apellidos	VARCHAR(191)	Apellidos del usuario registrado.		no
movil	VARCHAR(191)	Número de teléfono asociado al usuario.		no
usuario	VARCHAR(191)	Nombre de usuario		no
edad	VARCHAR(191)	Edad de usuario		no
estatura	VARCHAR(191)	Estatura del usuario		no
peso	VARCHAR(191)	Peso del usuario		no
activo	int(11)			

Tabla 15. Definición de la tabla user.

POST				
Almacena los datos de los post de los usuarios				
Campo	Tipo	Descripción	Restricciones	Nullable
id	INT(11)	Identificador del post. Auto-generado	Primary key	no
Texto	VARCHAR(191)	Texto dentro del post		No
Hora_publicacion	VARCHAR(191)	Hora de publicación del post.		No
Publicado_por_id	INT(11)	Identificado del usuario que ha generado el post	Foreing_key Id (usuario)	

Tabla 16. Definición de tabla post.

4.2. REQUISITOS FUNCIONALES

Son aquellos que determinan qué tareas tiene que hacer el sistema.

Por parte de los usuarios registrados y activados estas son sus funcionalidades:

-Reservar y cancelar la plaza en una sesión de una actividad.

El usuario debe poder reservar y cancelar una plaza en la sesión de la actividad que quiera realizar o dejar de realizar.

-Listar reservas próximas

El usuario debe poder consultar las reservas realizadas y las canceladas.

-Listar reservas históricas

El usuario debe poder consultar las reservas a las que ha acudido.

-Modificación de los datos del perfil

El usuario debe poder modificar los datos de su propio perfil.

Por parte de los administradores:

-CRUD de las actividades

El administrador debe tener la capacidad de crear, consultar, modificar y eliminar actividades.

-CRD de las sesiones

El administrador debe tener la capacidad de crear, consultar y eliminar sesiones.

-RUD de los usuarios

El administrador debe tener la capacidad de consultar, modificar y eliminar usuarios. Entre las funciones más importantes se encuentra la activación de los usuarios, debido a que en la creación se generan deshabilitados y es trabajo del administrador activar estos perfiles.

-CRUD de los administradores

El administrador debe tener la capacidad de crear, consultar, modificar y eliminar administradores.

-CRUD de los monitores

El administrador debe tener la capacidad de crear, consultar, modificar y eliminar monitores.

-Independizar usuarios de la aplicación de los administradores.

Por seguridad y rapidez los usuarios de la aplicación son independientes de los usuarios administradores, cada parte tiene su acceso independiente, la persistencia de sus datos está dividida en diferentes tablas de la base de datos y en ningún momento se cruzan.

4.3. REQUISITOS NO FUNCIONALES

Diseño enfocado a móviles y tabletas

Debido a que los usuarios hacen gran uso de dispositivos móviles o a través de las tabletas de los establecimientos. Es conveniente realizar un diseño *mobile first* en el que se prioriza su maquetación adaptada a pantallas pequeñas. Pero a su vez se puede utilizar en otro tipo de plataformas.

Diseño easy to use

Según la petición del cliente si lo que quiere es dejar de tener colas en el mostrador del centro deportivo, no tiene que complicar al usuario las funcionalidades que quiere servir. Por ello esto tiene relevancia

5.DISEÑO

En esta fase se realiza una aproximación al diseño tecnológico de la solución.

5.1.ESTRUCTURA DE LA APLICACIÓN

Se trata del desarrollo de una aplicación Multipage Application (MPA). Son el tipo más convencional de aplicaciones web. Cada petición que se realiza al servidor refresca una nueva página. Una MPA tiende a ser soportada en la mayor de navegadores web y puede trabajar sin necesidad de JavaScript. A diferencia de las Single Web Application (SPA) que son un tipo de aplicación web donde todas las pantallas se muestran en la misma página, sin necesidad de recargar el navegador. Las SPA surgieron de la necesidad de mejorar la experiencia de usuario y el flujo de trabajo. Google, Facebook o Trello utilizan SPA's.

Los pros de las MPA son:

- Hay muchos más recursos disponibles, ejemplos y frameworks disponibles. Lo que significa que es fácil encontrar desarrolladores, ya que la mayoría de ellos tendrá experiencia en MPA.

- El posicionamiento en buscadores (SEO) es realmente sencillo ya que cada solicitud devuelve el contenido completo. Cualquier rastreador verá exactamente lo que ve el usuario.

- Desarrollo sencillo: las MPA son generalmente más fáciles de desarrollar ya que requieren menores tecnologías.

- La carga de la página inicial es más rápida: la página se representa en el lado del servidor antes de enviarla al navegador. Cuando se recibe, el contenido está listo para ser mostrado.

- Las MPA son generalmente más baratas de desarrollar debido a la disponibilidad de los desarrolladores y el uso de tecnologías.

Entre los contras de las MPA:

- Las MPA son más lentas para cargar páginas. El navegador espera al nuevo contenido y no hará nada mientras tanto. Cuanto más contenido o páginas complejas solicite, más tendrá que esperar el usuario para que se muestre su contenido.

- Frontend y backend están estrechamente acoplados: en una arquitectura MVC típica, la lógica del lado del servidor está en Controladores y Modelos, mientras que la frontend está en Vistas. Sin embargo, interactúan estrechamente entre sí, ya que las vistas deben estar al tanto del backend.

- Desarrollo móvil: el código no se puede reutilizar fácilmente para una versión móvil de la aplicación. Necesita desarrollar un nuevo backend para ello o agregar puntos finales de API y lógica en la aplicación existente, lo que resulta en un tiempo de desarrollo adicional.

5.2.COMPONENTES DEL SISTEMA / ARQUITECTURA DE RED

Servidor Web

Los servidores web son los que hacen posible alojar nuestra aplicación web, es decir tener nuestra aplicación en funcionamiento. La principal función de un servidor web es almacenar los datos de nuestro sitio web y revotarlos cuando los usuarios realicen peticiones a los diversos apartados de nuestra web.

En nuestro caso hemos utilizado XAMPP debido a su gratuidad y fácil integración en cualquier equipo Windows, Linux...

Sistema gestor de base de datos

Un sistema gestor de base de datos (SGDB) es un sistema que permite la creación, gestión y administración de bases de datos, así como la elección y manejo de las estructuras necesarias para el almacenamiento y búsqueda de la información del modo más eficiente posible.

En nuestro caso hemos utilizado MySQL porque ya viene incorporado con XAMPP. Como característica principal y que lo hace muy atractivo es que es gratuito. Además, es muy sencillo de utilizar, es rápido, posee buena seguridad y es compatible con varios SO dado que es de código abierto. Todas estas ventajas hacen que MySQL sea uno de los SGDB más usados en el mundo en lo que a bases de datos relacionales se refiere.

Navegadores web

Un navegador web es un software, aplicación o programa que permite acceder a la web interpretando el conjunto de archivos proporcionados por un sitio web para que el usuario sea capaz de visualizarlos. La funcionalidad básica de un navegador web es permitir la visualización de documentos de texto, generalmente escritos en lenguaje HTML o JSP. Además, permite visitar páginas web y hacer actividades en ella, es decir, enlazar un sitio con otro, imprimir, enviar y recibir correo, entre otras funcionalidades más. A esto se le llama navegar. Actualmente existen numerosos navegadores web, aunque las preferencias de los usuarios están muy definidas, pero cabe destacar los siguientes:

- Google Chrome
- Mozilla Firefox
- Opera
- Safari
- Opera
- Internet Explorer

5.3.HERRAMIENTAS Y TECNOLOGÍAS UTILIZADAS

Se está utilizando un servidor XAMPP (Apache + MariaDB + PHP + Perl). Nos garantiza un entorno seguro y cómodo de desplegar. Este pack viene con PhpMyAdmin integrado otro plus.

En relación de lenguajes de programación, estamos usando en el Back-end PHP siguiendo el patrón nativo MVC con el framework Codeigniter. Integrando RedBeanPHP para la persistencia de datos. También estamos utilizando JavaScript, JQuery, JQueryUI,PL\SQL.

6. IMPLEMENTACIÓN

6.1. Implementación del modelo de datos

Partiendo del diseño, en esta fase se construye el sistema.

En el proyecto no hay implementación del modelo de datos, si esto significa que tenemos algún software que permite una migración a diferentes SGDB.

6.2. Carga de datos

Para poner en funcionamiento nuestra aplicación en la base de datos, tenemos un script SQL que genera los procedimientos y las tablas con contenido incluido.

Configuraciones realizadas en el sistema

Para este proyecto no se necesita mucha configuración de nuestro servidor XAMPP.

Con hacer un “git clone <https://github.com/victhorius/fitpocket.git>” en la carpeta htdocs ya debería ser funcional.

7. EXPLOTACIÓN

7.1. PLANIFICACIÓN

La implantación es la fase más crítica del proyecto ya que el sistema entra en producción, es decir opera en un entorno real, con usuarios reales.

PLANIFICACIÓN

Una vez alcanzados unos objetivos mínimos en cuanto a funcionalidad de la aplicación, esta se subirá al servidor elegido.

Este servidor se elegirá en función del tiempo disponible y compatibilidad con nuestro sistema.

Los objetivos mínimos son los indicados en los requisitos funcionales y no funcionales.

8. DEFINICIÓN DE PROCEDIMIENTOS DE CONTROL Y EVALUACIÓN

A lo largo del ciclo de vida del proyecto hemos estado utilizando una Github como gestor de control de versiones. Víctor tiene Github Premium lo que nos permitio tener el repositorio en oculto.

También hemos estado utilizando Gitkraken que es una potente y elegante interfaz gráfica multiplataforma para git desarrollada con Electron. De forma muy sencilla podemos llevar el completo seguimiento de nuestros repositorios, ver ramas, tags, crear nuevos, todo el historial de nuestro trabajo, commits etcétera.

9. CONCLUSIONES

En el equipo de trabajo ha faltado una organización inicial del proyecto, lo cual ha producido que acabemos de prisa y corriendo los últimos retoques. Nos hemos quedado con ganas de incorporar más funcionalidades, una mejor documentación, pruebas unitarias...

10. FUENTES

DAW

Enseñanzas mínimas: Real Decreto 686/2010, de 20 de mayo (BOE 12/06/2010)

http://pdf/IFCS03/titulo/RD20100686_TS_Desarrollo_Aplicaciones_Web.pdf

Currículo: Decreto 1/2011, de 13 de enero (BOCM 31/01/2011)

http://pdf/IFCS03/curriculo/D20110001_TS_Desarrollo_Aplicaciones_Web.pdf