

Programación Web

Carlos Lucena Robles f92luroc@uco.es
Miguel Raigón Jiménez i92rajim@uco.es
Kamal Abdelkader Haddu i02abhak@uco.es

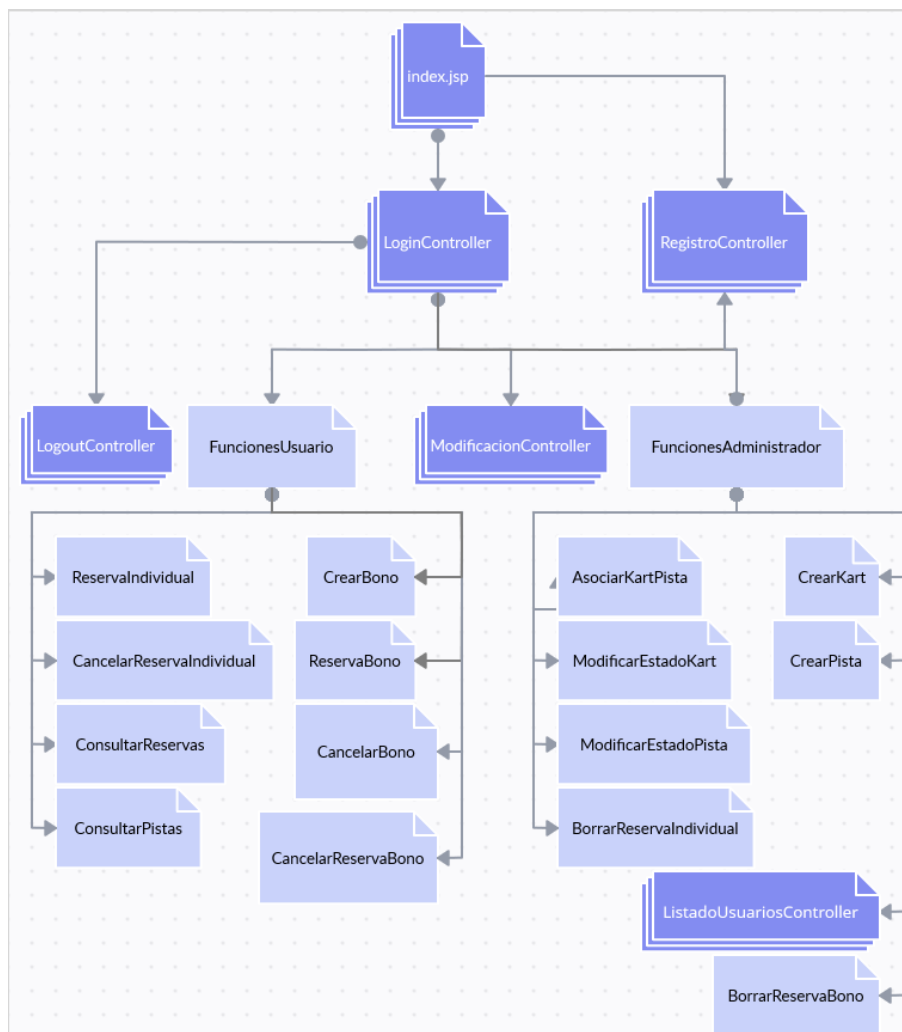
Pablo Roldán Puebla i92ropup@uco.es
Paloma Romero Delgado i92rodep@uco.es

Informe Práctica 3

1. Pila de tecnología

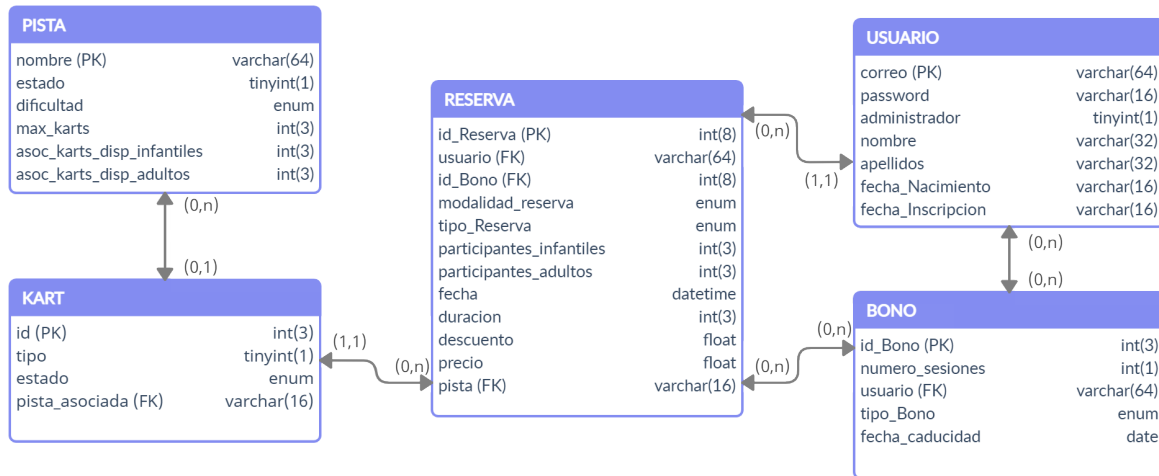
Para el desarrollo back-end de la aplicación se ha usado MySQL, Java y Apache Tomcat, y el desarrollo de front-end se ha llevado a cabo con HTML, CSS y JavaScript.

2. Mapa de navegación de la aplicación



3. Base de datos

3.1. Esquema relacional



3.2. Los DAOs que se han implementado son KartDAO, PistaDAO, UsuarioDAO, ReservaAdultosDAO, ReservaDAO, ReservaFamiliarDAO y ReservaInfantilDAO. Por ejemplo el DAO de kart accede a la tabla KART de la base de datos para hacer diferentes consultas como son las requeridas por los métodos crearKart, actualizarPistaKart, eliminarPistaKart, entre otros. Debido a la gran cantidad de métodos que tiene cada uno de ellos se aconseja ver desde el propio programa los comentarios explicados.

4. Instrucciones de acceso

4.1. Acceso a phpMyAdmin:

4.1.1. Usuario: f92luroc

4.1.2. Contraseña: pw2223

4.2. Despliegue de la aplicación web: ejecutar en un servidor tomcat el archivo .war. Para abrir la aplicación en el navegador web se hace con: <http://localhost:8080/WebProyectoPW/index.jsp>

4.3. Usuarios de prueba:

4.3.1. Usuario administrador:

4.3.1.1. Correo electrónico: admin@admin.es

4.3.1.2. Contraseña: admin

4.3.2. Usuario cliente:

4.3.2.1. Correo electrónico: jpp@gmail.com

4.3.2.2. Contraseña: pass

5. Decisiones de diseño e implementación

5.1. Aspectos sobre usuarios:

- 5.1.1. Respecto a usuarios, para distinguir entre un usuario administrador y un cliente se tiene en la base de datos el atributo administrador en la tabla USUARIO. Este resulta ser true (1) cuando es administrador y false (0) cuando no lo es. Además, se tiene el campo password que guarda la contraseña del usuario.
- 5.1.2. Se decidió que un administrador solo puede registrar a otros administradores, por tanto se tiene un administrador inicial introducido manualmente en la base de datos.
- 5.1.3. Respecto a la modificación de los datos de un usuario, el campo contraseña sirve tanto para confirmar el cambio como para establecer una nueva contraseña.

5.2. Aspectos sobre pistas y karts:

- 5.2.1. Sobre la creación de pistas se ha considerado que el número mínimo de máximo de karts sea 5 por pista.
- 5.2.2. En la tabla PISTAS de la base de datos se tienen los atributos `asoc_karts_disp_infantiles` y `asoc_karts_disp_adultos` que indican el número de karts disponibles de cada tipo que pueden circular por la pista al mismo tiempo. La suma de ambos valores no puede superar a los karts máximos permitidos en la pista. Esto último se encuentra controlado por js.
- 5.2.3. Para la modificación del estado de karts se ha tomado que se pueda cambiar sus estados de reservado a disponible o de reservado a mantenimiento. Por tanto, estos karts se retiran de la reserva en la que estuviesen y la pista asociada pasa a null. Sin embargo, no se permite el cambio de disponible a reservado ni de mantenimiento a reservado. El cambio de disponible a reservado solo se efectuará en caso de realizar reservas.
- 5.2.4. Se ha considerado poder cambiar el estado de uno o varios karts a la vez, con la restricción que el cambio sea para todos el mismo, por ejemplo, todos los karts que se seleccionen pasan a estar disponibles.
- 5.2.5. Para la modificación del estado de pistas, las reservas ya realizadas no se ven afectadas independientemente de su cambio de estado. Se ha considerado poder cambiar el estado de una o varias pistas a la vez,

con la restricción que el cambio sea para todas el mismo, por ejemplo, todas las pistas seleccionadas pasan a estar disponibles.

- 5.2.6. A la hora de realizar la consulta de las pistas disponibles se ha filtrado por tipo y número mínimo de karts disponibles. Además, se ha contemplado poder seleccionar ambos. En cambio, no se ha llevado a cabo la consulta según fecha.

5.3. Aspectos sobre reservas:

- 5.3.1. El campo fecha de la tabla RESERVA de la base de datos se ha establecido como tipo datetime, para poder tener en cuenta las horas de las reservas.
- 5.3.2. Para reservas individuales, la fecha de hacer una reserva no puede ser en el mismo día, esto se encuentra controlado por JS. Además, no se permite hacer una reserva a menos de 2 horas de la última reserva realizada para la misma fecha. Se ha tomado 2 horas por comodidad, ya que una reserva puede durar como máximo este tiempo (120 min).
- 5.3.3. El número de participantes máximo por reserva es de 20 personas, está controlado por HTML.
- 5.3.4. En la tabla BONOS el atributo número de sesiones indica la cantidad de sesiones que tienen los bonos actualmente. Como máximo se establece 5 sesiones y como mínimo 0. Para borrar un bono es necesario borrar previamente todas las reservas del bono.
- 5.3.5. Se toma en cuenta las fechas de caducidad de las reservas de bono, por lo que no se permite hacer una reserva posterior a la fecha de caducidad. Por defecto, se ha tomado la fecha de caducidad de 01/01/1970.
- 5.3.6. Para la consulta de reservas entre una fecha de inicio y fin, se muestra una tabla en la que se diferencian las reservas finalizadas y futuras por el color de fondo, siendo gris para aquellas reservas finalizadas.

5.4. Otros aspectos:

- 5.4.1. Las funcionalidades requeridas por cada usuario se muestran por una serie de botones para poder acceder a la operación que desee. Se han usado botones, tablas, selectores, entre otros, para aportar mayor facilidad de uso al usuario.
- 5.4.2. Mediante JavaScript se ha realizado validaciones de registro, entre las que están que no se puedan registrar menores de edad y que los campos de nombre, apellidos y contraseña cumplan con un número de caracteres válido. Además, validación de modificación de datos en concreto del nombre, apellidos y contraseña. Otros controles son fechas correctas, validación de reservas para que se hagan con al

menos un día de antelación, control para que el nombre de una pista creada sea válido y para la consulta a las pistas disponibles.

6. Dificultades

Las dificultades encontradas a lo largo de la realización de la práctica han sido las siguientes:

1. A pesar de habernos planificado mejor que en las anteriores prácticas, con más tiempo podríamos haber pulido más el código, haber desarrollado más el front-end y haber implementado la consulta de pistas por fecha.
2. Numerosos fallos durante su realización. Esto se debe en cierta medida a la falta de experiencia y a la extensión del proyecto.
3. Problemas para configurar Apache Tomcat en Eclipse.

7. Referencias

1. Atta. "Web Development Tutorials & Protips"
<<https://attacomsian.com/blog/>>
2. MCLibre, Bartolomé Sintés Marco (2022). "Páginas web HTML y hojas de estilo CSS"
<<https://www.mclibre.org/consultar/htmlcss/>>
3. DeveloperMozilla (2022), "HTML: Lenguaje de etiquetas de hipertexto"
<<https://developer.mozilla.org/es/docs/Web/HTML>>
4. MySQL(2022), "Date and Time Functions"
<<https://dev.mysql.com/doc/refman/8.0/en/date-and-time-functions.html>>
5. W3Schools (1999-2022). "SQL CHECK Constraint"
<https://www.w3schools.com/sql/sql_check.asp>
6. W3Schools (1999-2022). "HTML Tutorial"
<<https://www.w3schools.com/html/default.asp>>
7. W3Schools (1999-2022). "HTML Tutorial"
<<https://www.w3schools.com/css/default.asp>>
8. Autor desconocido. "Uso combinado de JavaBeans, servlets, y JSP"
<http://gssi.det.uvigo.es/users/agil/public_html/LRO/jsp.pdf>
9. Sun Microsystems, Inc. (1999). "jsp:setProperty"
<<https://tecfa.unige.ch/guides/jsp/jsp-ref10/syntaxref10.fm13.html>>
10. Stack Overflow, usuario FeCH(2017)
<<https://stackoverflow.com/questions/43153424/parsing-html-datetime-local-type-input-into-java-localdatetime>>
11. Stack Overflow, usuario Kirn (2010)
<<https://stackoverflow.com/questions/4195688/how-can-i-read-context-parameter-web-xml-values-in-a-non-servlet-java-file>>
12. Stack Overflow, usuario Mr Man(2011)
<<https://stackoverflow.com/questions/5953212/passing-multiple-request-parameters-with-same-name-from-table-rows>>