

PROGRAMACIÓN WEB

Antonio Jiménez Jiménez
Gonzalo Márquez de Torres
José María Muñoz López



PRÁCTICA III. GESTIÓN DE CAMPAMENTOS.

1.	Pila de Tecnología.....	3
2.	Mapa de Navegación.....	3
3.	Esquema Relacional y DAOs.....	4
4.	Instrucciones Para Acceder al Sistema.....	6
5.	Decisiones Tomadas y Análisis de Dificultades.....	6
6.	Bibliografía.....	6

Usuario de la Base de Datos: i02jjia

Contraseña: PRPW

PRÁCTICA III. GESTIÓN DE CAMPAMENTOS.

1. Pila de Tecnología.

La pila de tecnología utilizada en nuestro proyecto presenta la siguiente estructura:

- ✚ El lenguaje de programación utilizado en el back-end ha sido principalmente Java, aunque también hemos hecho uso de JavaScript para controlar ciertos aspectos y errores.
- ✚ En cuanto al front-end, hemos hecho uso de CSS como lenguaje de diseño y HTML con el objetivo de desarrollar la estructura de la página web.
- ✚ En cuanto a la base de datos relacional MySQL, haciendo uso de la alojada en la UCO, para almacenar y organizar los datos de la aplicación.
- ✚ Como entorno de programación, hemos hecho uso de Eclipse.
- ✚ Como servidor se ha utilizado Apache Tomcat con el objetivo de ejecutar y desplegar nuestra aplicación.

2. Mapa de Navegación.

Dentro de nuestra aplicación, crearemos el mapa de navegación que se correspondería con el siguiente:

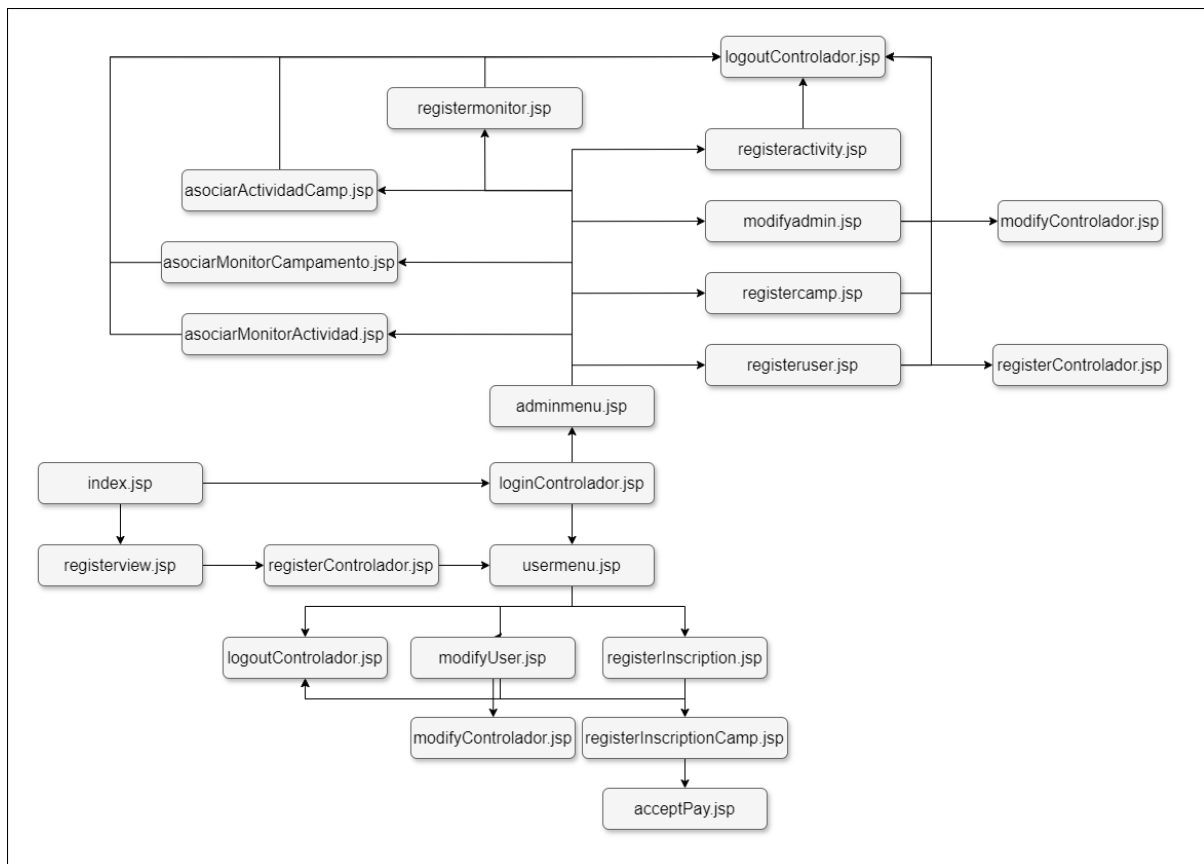


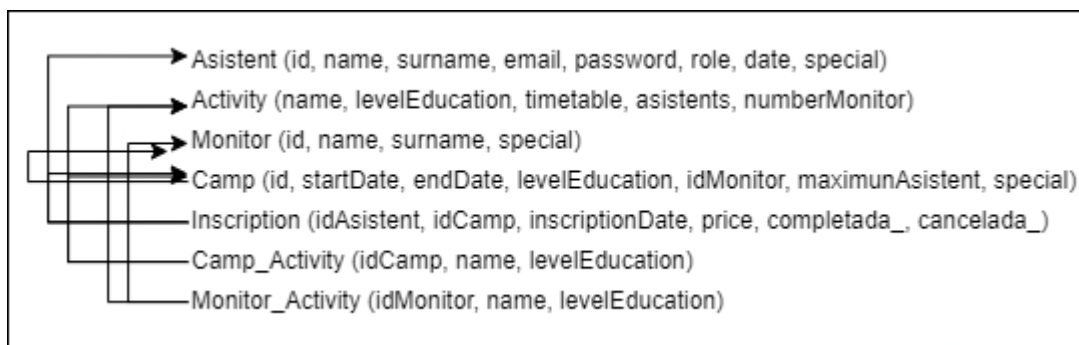
Figura 1. Mapa de Navegación.

En primer lugar, tendremos dos opciones, una de inicio de sesión y otra de registro. Además, en cuanto el inicio de sesión diferenciaremos entre administrador o usuario común.

En caso de acceder como administrador, la página principal, además de las opciones disponibles del menú de administrador, mostrará una tabla con el listado de identificador de campamentos, el número de personas inscritas y el acceso a más información sobre este. En las opciones disponibles tendremos dar de alta actividades, monitores y campamentos, asociar actividades a campamentos, monitores a actividades y a campamentos. En el caso de acceder como usuario, la página principal podrá realizar un registro en un determinado campamento, además de mostrar las inscripciones a la cuales se accede. En ambos casos, tendrá la opción de modificar los datos y cerrar sesión.

El sistema está basado en un patrón Modelo-Vista-Controlador (MVC). A través de los DAOs, se realizan las operaciones con los datos introducidos en el sistema particularmente para cada usuario. El usuario seguirá el flujo de navegación mostrado en el punto anterior mediante el uso de los JSPs.

3. Esquema Relacional y DAOs.



Con respecto a los DAOs, existentes en la capa de datos, tendremos los siguientes con sus respectivos métodos.

ActivityDAO.

- ❖ registrarActividad: registra una nueva actividad.
- ❖ imprimirActividad: imprime todas las actividades disponibles.
- ❖ asociarMonitor: asocia un monitor a una determinada actividad.
- ❖ imprimirActividadNameAndLevel: imprime el nombre y el nivel educativo de una actividad.
- ❖ contarMonitores: cuenta el número de monitores asociados a una actividad.
- ❖ actividadAsociadaACamp: verifica si una actividad está asociada a una campamento.

AsistentDAO.

- ❖ esAsistentExiste: existe el asistente en la base de datos.
- ❖ registrarAsistent: registra un asistente en el sistema.
- ❖ modificarAsistent: modifica los datos de un asistente.
- ❖ getAsistentConEmail: obtiene los datos de un asistente a través del email.
- ❖ esAsistentRegistrado: verifica si el asistente está registrado en el sistema.
- ❖ esAsistentSpecial: verifica si el asistente es especial.

CampDAO.

- ❖ getCamp: obtiene los datos de un campamento a través del identificador.
- ❖ registrarCamp: registra un campamento en el sistema.
- ❖ imprimirCamps: lista los campamentos del sistema.
- ❖ contarActividadesCamp: cuenta las actividades asociadas a un campamento.
- ❖ comprobarActividad: verifica si una actividad ya existe en el campamento.
- ❖ asociarActividad: asocia una actividad a un campamento.
- ❖ asociarMonitor: asocia un monitor a un campamento.
- ❖ asociarSpecialMonitor: asocia un monitor especial a un campamento.
- ❖ buscarCampDisponibles: lista los campamentos disponibles.
- ❖ getActivitiesEnCamp: obtiene las actividades asociadas a un campamento.
- ❖ imprimirCampsID: lista los campamentos disponibles del sistema.

InscriptionDAO.

- ❖ registrarInscription: registra una inscripción en el sistema.
- ❖ imprimirInscriptions: existe el asistente en la base de datos.
- ❖ contarInscriptionsCamp: contar el número de inscripciones de un determinado campamento.
- ❖ getInscriptionsById: obtiene las inscripciones de un determinado asistente.
- ❖ eliminarInscriptionByAsistentAndIdCamp: elimina la inscripción de un determinado asistente a un campamento concreto.
- ❖ calculaPrecioCompleto: calcula el precio de un campamento de una inscripción completa.
- ❖ calculaPrecioParcial: calcula el precio de un campamento de una inscripción parcial.
- ❖ getAsistentsEnCamp: obtiene una lista del identificador de los asistentes que asisten a un campamento.

MonitorDAO.

- ❖ registrarMonitor: registra un nuevo monitor en el sistema.
- ❖ imprimirMonitores: imprime los monitores del sistema.
- ❖ esMonitorAsociadaAActividad: verifica si un monitor está asociado a una actividad concreta.

A la hora de iniciar el proyecto, uno de los problemas fue el limitado conocimiento a la hora de crear la base de datos inicialmente. A medida que nos familiarizamos con los conceptos, gracias a las clases prácticas y la exploración en la web. La similitud entre el nuevo lenguaje y

otros previamente utilizados en aspectos como bucles y manejo de excepciones facilitó una adaptación, agilizando el proceso de aprendizaje.

4. Instrucciones Para Acceder al Sistema.

Para acceder a nuestra aplicación, diferenciaremos entre dos tipos de usuarios, por un lado, el administrador y, por otro lado, el usuario que interactúa con el sistema, como hemos comentado con anterioridad. Por ello podrá acceder con la siguientes credenciales:


Administrador	Correo: lucia@gmail.com
	Contraseña: password
Usuario	Correo: paco@gmail.com
	Contraseña: paco1234


5. Decisiones Tomadas y Análisis de Dificultades.

Por una lado, decidimos introducir una única tabla de Asistentes, donde diferenciaremos entre el administrador y el usuario, puesto que nos simplificará las tareas y llamadas al sistema mediante el atributo *role*.

En el lado de dificultades, hemos encontrado el gestión de la seguridad, mediante los sistemas de autenticación y autorización utilizando filtros de seguridad mediante el uso de *userBean*. Además, de la adaptación al entorno de Eclipse, adherido a la dificultad del lenguaje Java, junto al uso de los *servlets*.

6. Bibliografía.

-  Apache Tomcat Project. (n.d.). Apache tomcat®. Apache.org. Retrieved August 27, 2024, from <https://tomcat.apache.org/>
-  Apuntes:Jdbc [acceso a datos]. (n.d.). Codeandcoke.com. Retrieved August 27, 2024, from <https://datos.codeandcoke.com/apuntes:jdbc>
-  Castellano, F. J. G. (n.d.). Tutorial Servlets y JDBC. Ugr.Es. Retrieved August 27, 2024, from <https://ccia.ugr.es/~figc/docencia/2005-2006/2/servlets/>
-  Contenidos Prácticos y Teóricos de Programación Web. (n.d.). Uco.es. <https://moodle.uco.es/m2324/>
-  Setting and getting connection property values. (n.d.). Vertica.com. Retrieved August 27, 2024, from <https://www.vertica.com/docs/9.3.x/HTML/Content/Authoring/ConnectingToVertica/ClientJDBC/SettingAndGettingConnectionPropertyValues.htm>
-  Squirrels, J. (2023, July 21). Tu primera aplicación usando servlets de Java. CodeGym. <https://codegym.cc/es/groups/posts/es.255.tu-primera-aplicacion-usando-servlets-de-java>
-  Stack Overflow - where developers learn, share, & build careers. (n.d.). Stack Overflow. Retrieved August 27, 2024, from <https://stackoverflow.com/>

 UserBean (oracle WebCenter sites java API reference). (2017, January 20). Oracle.com.
<https://docs.oracle.com/middleware/12212/wcs/java-ref/com/fatwire/rest/beans/UserBean.html>