

Práctica 2. JSF

Aplicación web para la participar en los concursos convocados por la UBU.

Objetivo

Desarrollar una aplicación web con nombre `participar` (WebApp) que permita participar o votar como jurado en las convocatorias de concursos registrados previamente en la aplicación `concursos`.

Funcionalidad

Casos de uso 1. Validación: el usuario se valida en el sistema. Si la validación es correcta se pasa a la ventana de concursos disponibles en los que participar/votar.

Casos de uso 2. El usuario visualizará el listado de concursos ya registrados en la herramienta. Podrá seleccionar cualquiera de los existentes para participar como concursante o votar como jurado.

Caso de uso 3: Participar en un concurso: en esta pantalla el usuario podrá adjuntar/modificar la información requerida para participar en el concurso seleccionado. Deberá proporcionar al menos la información siguiente:

- Título
- Descripción
- Sobrenombre o apodo
- Ficheros adjuntos

Caso de uso 4: El usuario introduce el título del fichero y lo adjunta para que sea almacenado en el sistema asociado al concurso_`concursante` correspondiente.

Caso de uso 5: Votar como jurado: El usuario visualizará un listado con todos los participantes del concurso. Podrá descargar los adjuntos de cada participante para valorarlos. Podrá votar a los concursantes que desee. Para simplificarlo no se va a restringir la votación a una sola por concurso / jurado, de modo que la votación simplemente incrementará un contador de votaciones.

Requisitos no funcionales

La aplicación se resolverá utilizando los contenidos vistos en la sesión de prácticas previa *JSF*.

La persistencia de los datos se gestionará mediante conexiones JDBC contra una base de datos MySQL.

Como resultado final se obtendrá un fichero war que deberá obtenerse desde Maven con la opción: `mvn package`

Prototipos

La aplicación web se compone básicamente de cinco pantallas principales (correspondientes a los casos de uso). Estas ventanas se corresponderán con formularios HTML.

Modelo de datos

Se utilizará un modelo de cinco entidades/clases (ampliado sobre la práctica 1). Se presenta a continuación el modelo relacional.

- Usuarios(id, login, password, perfilAcceso)
- Concursos(id, titulo, objetivo, bases, premios, plazo, jurado, *idUsuario*)
- Ficheros(id, titulo, ruta, *idConcurso*)
- Participaciones(id, titulo, descripcion, apodo, *idConcurso*, *idUsuario*, votosAcumulados)
- FichParticipaciones(id, titulo, ruta, *idParticipacion*)

En la tabla Participaciones se tiene como clave foránea *idConcurso* + *idUsuario* que referencian al campo id de la tabla de Usuarios y al campo id de la tabla Concursos. En la tabla FichParticipaciones se tiene como clave foránea a *idParticipacion* que referencia al campo id de la tabla Participaciones.

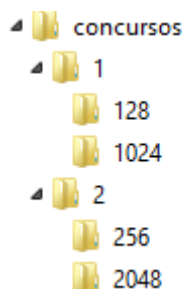
En UBUVirtual se dispone de ejemplos de código Java para el manejo de consultas e inserciones con JDBC.

Simplificaciones

No se solicitan validaciones de los datos del formulario, en el lado del cliente.

Los ficheros subidos al servidor se alojan a partir de un subdirectorio que cuelga del directorio de la aplicación web (*participar*) de nombre *directorio*. Dentro de este subdirectorio, existirá un directorio por concurso, coincidiendo en nombre con el identificador del concurso. Dentro de cada directorio de concurso, existirá a su vez un subdirectorio por cada usuario participante, coincidiendo en nombre con el identificador del usuario.

Ej: suponiendo que en los concursos 1 y 2 han participado los usuarios 128 y 1024 para el concurso 1 y los usuarios 256 y 2048 para el concurso 2:



Alternativa a la creación dinámica de subdirectorios: Estas carpetas estarán **ya creadas previamente** y la aplicación **no se encarga de gestionarlas**, sólo de alojar ficheros en el correspondiente directorio y de insertar la correspondiente entrada en la base de datos.

Condiciones de entrega

Se entregará un fichero `participar.zip` a través de la tarea habilitada en UBUVirtual. Dicho proyecto contendrá un proyecto Eclipse de naturaleza Maven y los scripts de creación de las tablas de base de datos en MySQL.

Se reutilizará el pool de conexiones creado en la práctica anterior.

El proyecto tendrá en su directorio raíz un fichero `pom.xml` que permite el desarrollo completo del fichero `war` a desplegar en un servidor Glassfish. Antes de la entrega se recomienda `mvn clean`.

Adicionalmente se entregará un **video** en el que alumno explica cómo ha desarrollado la práctica, mostrando cómo está organizado el proyecto, arrancándolo y probando que funciona.

Revisar la documentación de la tarea para detalles adicionales y cambios en la fecha y hora de envío límite (**31/12/2021 a las 21:30**). La práctica tiene una valoración máxima de 1 punto.

Anexo 1. Creación del poyecto

Crear el proyecto con *archetype*: `javax.faces-war-archetype`

Cambiar las propiedades del proyecto *Project Facets* para incluir JSF 2.2

Añadir la carpeta `src\main\webapp` seleccionando con el botón derecho, *Build Path / Us as Source Folder* si no está en Java Resources.

Con eso el proyecto debería estar preparado para trabajar con:

- La carpeta `src\main\java` para las clases java (servlets, javabeans, clases de utilidad, etc).
- La carpeta `src\main\webapp` para las desarrollar los .jsp, .html, ficheros de imágenes, CSS, etc.

En UBUVirtual se deja un fichero `pom.xml` de ejemplo, con la configuración que se debería utilizar en el desarrollo de la práctica, con los ajustes que sean necesarios. El usuario administrador de Glassfish debe tener login "admin" y password "admin".

Anexo 2. Componente gráfico <h:dataTable/>

Para la resolución de la práctica se sugiere el uso del componente definido con `<h:dataTable/>`. Este componente define la estructura de una tabla (sus columnas) y permite inyectar una lista con elementos sobre los que iterar para construir las filas, en tiempo de ejecución.

En el siguiente ejemplo se muestra su uso, definiendo una tabla de 3 columnas. La lista de objetos a mostrar se obtiene de `value="#{listaTrabajos.lista}"`, y en cada una de las iteraciones el objeto a acceder está en la variable "trabajo":

```
<h:dataTable border="1" value="#{listaTrabajos.lista}" var="trabajo">
  <h:column id="Autor">
    <p:facet name="header">
      <h:outputText value="Autor"/>
    </p:facet>
    <h:outputText value="#{trabajo.login}"/>
  </h:column>
  <h:column id="Titulo">
    <p:facet name="header">
      <h:outputText value="Titulo"/>
    </p:facet>
```

```
        <h:outputText value="#{trabajo.titulo}"> </h:outputText>
    </h:column>
    <h:column id="Fichero">
        <p:facet name="header">
            <h:outputText value="Fichero"/>
        </p:facet>
        <h:outputText value="#{trabajo.ruta}"> </h:outputText>
    </h:column>
</h:dataTable>
```