

Taller de Programación

Tarea 2 – parte 2

Curso 2021

1 Información Administrativa

La tarea comienza el **6 de setiembre** y finaliza el **25 de octubre**, y constará de dos partes: **la parte 1 que irá del 6 de setiembre al 27 de setiembre** y **la parte 2 que irá del 4 de octubre al 25 de octubre**. La tarea constará además de múltiples entregas parciales a realizarse a través de actividades en el EVA, y la entrega final de código de cada parte a realizarse hasta el **27 de setiembre a las 23:59 hs para la parte 1**, y hasta el **25 de octubre a las 14:00 hs para la tarea completa** (ver [1]).

2 Objetivos

La parte 2 de la Tarea 2 mantiene los objetivos de aprendizaje por parte del estudiante indicados en la parte 1.

3 Introducción

Basados en los lineamientos definidos en la Descripción del Proyecto [2], en esta segunda etapa, se continuará con el desarrollo del componente Servidor Web. Al finalizar la parte 2 de la tarea 2 los usuarios interactuarán con la aplicación *Entrenamos.uy* desde un Cliente Web (Mozilla Firefox [3]) que accederá al Servidor Web. Tanto la Estación de Trabajo como el Servidor Web usarán la interfaz del Servidor Central para acceder a la lógica de negocio para poder realizar los casos de uso.

En esta tarea se tendrán dos aplicaciones separadas que **ejecutarán de manera independiente y no compartirán datos**. El Servidor Central será empaquetado como una librería (archivo JAR [5]), que será incluido dentro de la Aplicación Web y de la Estación de Trabajo (una copia de la librería en cada aplicación).

Este documento presenta los requerimientos funcionales y especiales para la parte 2 de la Tarea 2, cuyo foco es el desarrollo del Back-end de la Aplicación Web, utilizando tecnologías server-side de Java Enterprise Edition (JSP [10] y Servlets [11]) y en su integración con el Front-end desarrollado en la Parte 1 con tecnologías client-side (HTML5[6], CSS[7] y JavaScript[8])). Al igual que para la Parte 1, el Back-end utilizará la capa lógica del Servidor Central (desarrollada en la Tarea 1) con las modificaciones requeridas en esta tarea. El contenido estático de las entidades de la plataforma será eliminado y sólo se accederá a él mediante el Servidor Central. En esta parte también se debe realizar el *deployment* de la aplicación en el servidor Web Tomcat [4].

4 Visión

La parte 2 de la Tarea 2 mantiene la visión indicada en la parte 1.

5 Actores

La parte 2 de la Tarea 2 mantiene los actores indicados en la parte 1.

6 Requerimientos Funcionales – Parte 2

Los casos de uso seleccionados para su implementación en la parte 2 se presentan a continuación. Se continuará con el desarrollo de la aplicación web, conectando las vistas diseñadas en la parte 1 con la lógica del servidor central mediante tecnologías server-side.

6.1 Servidor Web

Caso de Uso	Alta de Usuario (modificar opcionalmente)
Actores	Visitante
Descripción	El caso de uso comienza cuando un visitante desea darse de alta como nuevo usuario en el sistema. Para ello indica su nickname (único), nombre, apellido, contraseña, confirmación de contraseña, correo electrónico (único), fecha de nacimiento y (opcionalmente) una imagen . Finalmente, el sistema da de alta al usuario. Si el usuario es un profesor se indica además otros datos básicos: la institución a la que pertenece, una descripción general, una breve biografía (opcional) y un link a su sitio web (opcional). Si el nickname o el correo electrónico se encuentran en uso por algún otro usuario, el sistema avisa al visitante, pudiendo éste corregir la información o cancelar el alta.

Caso de uso	Alta de Actividad Deportiva (modificar opcionalmente)
Actores	Profesor
Descripción	El caso de uso comienza cuando un profesor desea dar de alta una actividad deportiva de la institución a la que pertenece. En primer lugar, el profesor indica los datos básicos de la actividad: nombre (único), descripción, duración, costo. Se indican también la/las categorías asociadas a la actividad deportiva. Opcionalmente, el profesor puede elegir además una imagen para la actividad deportiva. En caso de que exista una actividad deportiva con dicho nombre, se puede modificar los datos o cancelar el caso de uso. Finalmente, el sistema da de alta la actividad en la fecha actual y con estado “Ingresada”.

Caso de uso	Alta de Dictado de Clase (modificar opcionalmente)
Actores	Profesor
Descripción	El caso de uso comienza cuando un profesor desea dar de alta una clase para una actividad deportiva. En primer lugar, el sistema lista las actividades deportivas con estado “Aceptada” de la institución a la que pertenece. El profesor elige una de ellas e ingresa los datos básicos de la clase: nombre de la clase (único), fecha y hora de inicio, socios mínimos, socios máximos, URL asociada. Opcionalmente, el profesor puede elegir además una imagen para la clase. En caso de que exista una clase con el nombre ingresado, el profesor puede modificar los datos o cancelar el caso de uso. Finalmente, el sistema da de alta el dictado de la clase en la fecha actual.

Los Casos de Uso **Consulta de Usuario, Modificar Datos de Usuario, Consulta de Actividad Deportiva, Consulta de Dictado de Clase, Registro a Dictado de Clase, Consulta de Cuponera de Actividades Deportivas, Compra de Cuponera, Seguir a un Usuario y Dejar de seguir a un Usuario** mantienen su especificación provista en la parte 1. En caso de no realizar el requerimiento especial **7.2 Imágenes de Usuario/Actividad Deportiva/Dictado de Clase** los casos de uso **Alta de Usuario, Alta de Actividad Deportiva y Alta de Dictado de Clase** mantienen su especificación provista en la parte 1.

7 Requerimientos Especiales – Parte 2

7.1 Calidad del Producto

La definición y ejecución de planes de pruebas adecuados es clave en el desarrollo de un sistema de alta calidad. En este sentido, al igual que lo solicitado en la iteración anterior, se requerirá el mantenimiento y desarrollo de pruebas automáticas para el componente Servidor Central. Estas pruebas deberán ser automatizadas utilizando JUnit [13] y la evaluación del cubrimiento de las mismas deberá realizarse con Eclemma [14]. Se requiere como mínimo que el cubrimiento de sentencias alcance el 80% del código que corresponde a la capa lógica. Cabe aclarar que solamente interesa la realización de pruebas sobre el componente Servidor Central, es decir, no es la intención que se realicen pruebas automáticas de interfaz de usuario ni sobre el componente Servidor Web.

La calidad no depende exclusivamente de la ausencia de errores en el código, sino de otros factores, por ejemplo la calidad del código generado. Cada organización suele definir buenas prácticas y exigir estándares de codificación que permiten, entre otras cosas, obtener un código más legible que facilite la realización de pruebas y el mantenimiento del código. En esta tarea se integrarán diversas herramientas de calidad del software como plugin de Eclipse, en particular las herramientas PMD [15] y CheckStyle [16] que permiten realizar un análisis estático del código (sin ejecutarlo) para identificar diversos problemas y comprobar el cumplimiento de buenas prácticas. En ambos casos se deberá modificar el código para que no aparezcan errores en ninguna de las siguientes categorías de evaluación. Se deberá usar la configuración básica, sin modificar, e incluir los chequeos que sean necesarios si no se encuentran habilitados.

PMD	CheckStyle
AbstractClassWithoutAbstractMethod	AvoidStarImport
LooseCoupling	EmptyCatchBlock
PackageCase	GenericWhitespace
ShortVariable	IllegalCatch
UnusedPrivateField	LeftCurly
	MemberName
	MethodName
	MethodParamPad
	UnnecessaryParentheses
	UnusedImports
	VisibilityModifier
	WhitespaceAfter

7.2 Imágenes de Usuario/Actividad Deportiva/Dictado de Clase

La presencia de contenido multimedia es imprescindible para promocionar los contenidos del sitio entrenamos.uy. Por este motivo, en la parte 1 se les pidió que los usuarios, actividades deportivas, clases y cuponeras puedan tener imágenes asociadas, las que se mostrarán junto con la información de cada elemento. En esta parte, se les solicita que el usuario pueda elegir dicha imagen al momento de crear el usuario/actividad deportiva/clase, tal y como se indica en los casos de uso respectivamente.

7.3 Búsqueda de Actividades Deportivas y Cuponeras

Debido a la gran cantidad de actividades deportivas que existen en la plataforma y al número de consultas que se pueden realizar, el sistema debe permitir organizar la información y habilitar el filtrado de los resultados para mejorar la respuesta a sus usuarios.

Particularmente, se pide que haya un cuadro de texto (Figura 2) en el cual el usuario pueda buscar y filtrar información a partir del contenido del cuadro. Serán listadas actividades deportivas y cuponeras de actividades deportivas cuyo nombre o descripción contengan el contenido del cuadro de texto (o todos si no se indica nada). El cuadro deberá estar disponible en todo momento en las páginas del Sitio Web (en el cabezal).



Figura 2 - Buscar Actividad Deportiva, Cuponera de Actividad Deportivas

Al ejecutar la búsqueda se irá a una página especial que muestra los contenidos que cumplen con los criterios de búsqueda. La página de resultados tendrá opciones para filtrar y ordenar los resultados. Un esquema posible de la página de resultados se sugiere en la Figura 3, mostrando las opciones para realizar el orden solicitado.

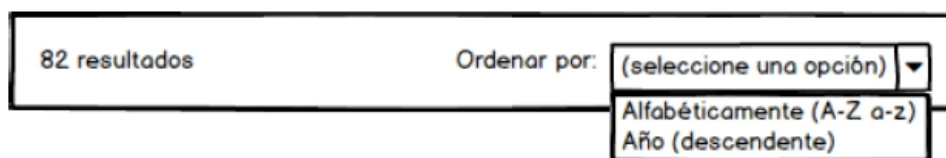


Figura 3 - Página de resultados de búsqueda con opciones de orden

Las opciones de filtrado y ordenamiento serán las siguientes:

- Filtrado por categoría, institución (no se muestra en la Figura 3)
- Ordenar alfabéticamente (en forma ascendente)
- Ordenar por fecha de publicación (en forma descendente)

8 Se Pide

- Documentación: Análisis y diseño de los aspectos más relevantes de la realidad
- Gestión: Registro de horas de actividad de cada integrante del grupo
- Desarrollo:
 - Desarrollo de una interfaz gráfica de usuario Web utilizando las tecnologías server-side JSP [10] y Servlets [11].
 - Desarrollo en Java de la lógica del Servidor Central que deba agregarse para cumplir con los nuevos casos de uso Web.
 - Desarrollo de los Requerimientos Especiales.
 - Desarrollo de casos de prueba con JUnit que deban agregarse para probar los cambios en la lógica y las nuevas funcionalidades, cumpliendo con el cubrimiento de sentencias del 80% en Eclemma.

En la sección Laboratorio link "Calendario de entregas y evaluaciones" del sitio EVA del curso se encuentra la indicación de entregas intermedias, final y sus fechas asociadas, así como los requerimientos obligatorios y opcionales para su aprobación.

9 Referencias

[1]	Estándar de Entregas https://eva.fing.edu.uy/course/view.php?id=315
[2]	Descripción General del Proyecto https://eva.fing.edu.uy/course/view.php?id=315
[3]	Mozilla Firefox http://www.mozilla.com/es-AR/
[4]	Apache Tomcat http://tomcat.apache.org/
[5]	JAR File Specification http://docs.oracle.com/javase/6/docs/technotes/guides/jar/jar.html
[6]	HTML5 https://www.w3schools.com/html/html5_intro.asp
[7]	CSS https://www.w3schools.com/css/
[8]	JavaScript https://www.w3schools.com/js/ https://developer.mozilla.org/es/JavaScript
[9]	Static Web Project https://help.eclipse.org/2019-03/topic/org.eclipse.wst.webtools.doc.user/topics/twcresta.html
[10]	Java Server Pages Technology http://www.oracle.com/technetwork/java/javaee/jsp/index.html
[11]	Java Servlet Technology http://www.oracle.com/technetwork/java/javaee/servlet/index.html
[12]	Dynamic Web Project https://help.eclipse.org/2019-03/topic/org.eclipse.wst.webtools.doc.user/topics/twcreprj.html

13]	JUnit Testing Framework http://www.junit.org/
[14]	EclEmma http://www.eclemma.org/
[15]	PMD http://pmd.sourceforge.net/eclipse/
[16]	Checkstyle http://eclipse-cs.sourceforge.net/