

ABPRO

Módulo	Nivel de Dificultad		
Desarrollo de aplicaciones webdinámicas con Java	Medio		
Tema: El patrón de diseño MVC	MVC		
Intención del aprendizaje o aprendizaje esperado:			
<ul style="list-style-type: none">• Construye un Servlet que rescata información de un request GET/POST utilizando tecnología Java acorde al estándar J2EE.• Construye una vista JSP que despliega información provista por un Servlet utilizando tags JSTL para el formateo acorde al estándar J2EE.• Implementa patrón MVC en un aplicativo web dinámico separando los elementos pertenecientes a cada capa			
Planteamiento del Problema:			
<p>Una empresa de asesorías en prevención de riesgos necesita contar con un sistema de información que le permita administrar los principales procesos que se llevan a cabo en ella día a día.</p> <p>Hasta el momento se han definido las siguientes funcionalidades en el sistema:</p>			
ID	Nombre	Detalle	Actor(es)
1	Inicio	Página de inicio del portal con información relevante del proyecto que se está realizando.	Cliente / Administrativo / Profesional
2	Contacto	Formulario de contacto para realizar consultas.	Cliente
3	Crear Capacitación	Formulario para crear una capacitación en el sistema.	Cliente
4	Listar Capacitaciones	Listado de capacitaciones registradas.	Cliente
5	Listado de Usuarios	Listado con los usuarios existentes en plataforma.	Administrativo
6	Crear Usuario	Formulario que permitirá crear un usuario en sistema.	Administrativo
7	Editar Cliente	Formulario que permite modificar los datos de un usuario de tipo cliente.	Administrativo
8	Editar Administrativo	Formulario que permite modificar los datos de un usuario de tipo administrativo.	Administrativo
9	Editar Profesional	Formulario que permite modificar los datos de un usuario de tipo profesional.	Administrativo

10	Listado Visitas	Lista de todas las visitas realizadas a cada uno de los clientes. Además, posee un formulario que permite agregar nuevas visitas a sistema.	Profesional
11	Responder checklist	Listado de chequeos de una visita.	Profesional
12	Listado Pago	Lista de todos los pagos realizados por los clientes.	Administrativo
13	Crear Pago	Permitirá agregar un pago al sistema.	Administrativo
14	Listado Asesorías	Lista con las asesorías realizadas hasta el momento.	Profesional
15	Crear Asesorías	Contiene un formulario para agregar una nueva asesoría.	Profesional
16	Reportes	Despliega reportes específicos.	Profesional
17	Administrar Asistentes	Administración de asistentes a una capacitación; incluye listarlos, poder agregar asistentes y eliminarlos.	Cliente
18	Login	Contendrá un acceso a un usuario al portal a través de un RUT de usuario y una clave.	Cliente / Administrativo / Profesional
19	Gestionar accidentes	Listado con todos los accidentes registrados en plataforma. Además, permite editarlos, agregar uno nuevo y eliminarlos. En esta sección cada cliente administra sus propios accidentes.	Cliente
20	Administrar chequeos	Permite mostrar los distintos chequeos realizados a cada cliente en una visita a terreno, y permite agregar uno nuevo, y cambiar su estado.	Administrativo

En los ejercicios de las sesiones anteriores, se crearon servlets para los siguientes casos de uso:

- Inicio
- Contacto
- Crear Capacitación
- Listar Capacitación
- Login
- Listado de Usuarios
- Crear Usuario

Además, agregó las clases asociadas a estos casos de uso en el modelo, y se asociaron los estilos y comportamiento dinámico creado en módulos anteriores.

Como parte de este ejercicio se pide realizar lo siguiente:

1. Como primera tarea, en caso de que no haya agregado en alguna de las clases creadas en el paquete “modelo” los accesores, los mutadores, los constructores o el método toString, por favor hágalo.
2. Cree un paquete de nombre “interfaces”.
3. En el paquete anterior, cree una interface para la clase Capacitacion. En ella deben estar declarados dos métodos: uno que retorne una lista de objetos de tipo Capacitacion (no debe

recibir ningún parámetro), y otro que reciba un objeto de tipo Capacitacion y lo “registre” en sistema (no se pide que almacene nada, solo se simulará una acción).

4. Cree un paquete de nombre “implementacion”.
5. En el paquete creado en el paso anterior, agregue una clase que “implemente” la interface creada; esto lo obligará a incorporar acciones para los métodos definidos. Para el caso del método que entrega un listado de capacitaciones, genere una rutina que retorne una lista de al menos tres objetos creados manualmente. En el caso del método que crea una capacitación, genere una rutina que reciba un objeto, y que despliegue sus datos por consola de Java.
6. Modifique el servlet que lista las capacitaciones, y haga una rutina tal que llame a la clase de implementación, y que utilice el método respectivo para obtener el listado de capacitaciones. Dicho listado debe desplegarlo por medio de la vista JSP utilizada en el caso de uso correspondiente.
7. Modifique el servlet que permite crear una capacitación. Dicho servlet debe desplegar el formulario de creación de capacitaciones; al procesar el formulario, sus datos deben ser recuperados por otro servlet, el cual generará un objeto de tipo Capacitacion, y usando dicho objeto llamará al método de creación de un registro en sistema. Una vez llamado al método, debe desplegar un mensaje alusivo a través de una vista JSP especialmente creada para este efecto.

El entregable debe ser un proyecto web dinámico Java comprimido en formato RAR o ZIP.

Nota 1: Este ejercicio es la continuación del ejercicio grupal del día anterior.

Nota 2: Como pudo notar, los ejercicios solicitados no registran datos en una base. Dicha funcionalidad será abordada en ejercicios posteriores.

Nota 3: En las referencias del problema, el último enlace es un ejemplo de aquello que se busca lograr, sin la necesidad de utilizar una base de datos.

El Datos de apoyo al planteamiento

Ejecución: Grupal (equipo de no más 4 personas)

Componentes para evaluar: Se deberá entregar un proyecto Java web dinámico, comprimido en formato ZIP o RAR.

Recursos Bibliográficos:

Qué es Java Enterprise (J2ee, JEE)

<https://www.fundesem.es/bt/publicacion-java-ee-y-el-desarrollo-web-un-enfoque-de-aprendizaje>

Qué son los Servlets

<https://users.dcc.uchile.cl/~jbarrios/servlets/general.html>

Utilizando métodos post y get

<https://www.ecodeup.com/enviar-parametros-servlet-desde-una-vista-utilizando-post-get/>

Sesiones y cookies

<https://ricardogeek.com/manejo-de-sesiones-y-cookies-en-servlets-java/>

Etiquetas JSTL

<http://www.jtech.ua.es/ayto/ayto2008/jsp/sesion07-apuntes.html>

MVC

<https://www.ecodeup.com/patrones-de-diseno-en-java-mvc-dao-y-dto/>

Introducción a MVC con Servlets y JSP

<http://blog.oscarscode.com/es/java-es/introduccion-a-mvc-con-servlets-y-jsp/>