

**“LABORATORIO DE PROGRAMACIÓN PARA SERVIDORES WEB”**  
**TERCER EXAMEN PARCIAL - PRÁCTICO**

<b>NOMBRE DEL DOCENTE</b>	<b>HARIM CASTELLANOS ALTAMIRANO</b>		
<b>NOMBRE DEL ESTUDIANTE</b>	<b>MATIAS RODRIGUEZ HECTOR EDGAR</b>		
<b>PORCENTAJE ASIGNADO AL EXAMEN</b>	<b>35%</b>	<b>NÚMERO DE REACTIVOS</b>	<b>4</b>

**I. INSTRUCCIONES:**

- a) El uso del celular está prohibido, por favor guárdelo en su mochila ya que en caso de usarlo o tenerlo a la vista su calificación será cero y su examen será retirado
- b) El tiempo para responder el presente examen es de 90 minutos
- c) Para contestar el examen solo requiere su computadora destinada para la clase donde valido su ambiente de desarrollo configurado mediante una tarea, su entorno de desarrollo y su USB
- d) No está permitido salir una vez iniciado el examen
- e) Cuando termine su examen entregarlo al profesor mediante su USB, tiene un lapso de 10 min para lo que usted requiera fuera del aula.
- f) La Memoria USB se quedará con el profesor hasta 10 min antes del término de la clase. Si no cuenta con su USB el profesor copiará directamente su examen y será penalizado con el 25% de su calificación correspondiente al examen práctico.
- g) El horario para presentar el examen es de 18:15 a 19:45 hrs. con fecha Martes 30 de Mayo 2017, entregar el examen después del horario señalado tendrá una calificación de cero
- h) El presente documento debe ser entregado en formato pdf con nombre de archivo su nombre completo. La entrega valida la aceptación de haber contestado su examen práctico y ser acreedor a una calificación. En caso de no entregarlo su calificación será cero.
- i) Responda lo que se indique en cada pregunta

1. Crea un proyecto Web Dinámico en Eclipse Neon utilizando Tomcat 9 que tenga el NOMBRE del estudiante seguido de U3 como nombre del proyecto. Ejemplo <HarimU3>
2. Utilizar los recursos de la carpeta 1\_actividadUno y copiar las carpetas *css* y *vendor*, así como *index.html* y *favicon.ico* a la carpeta de WebContent



css



vendor



favicon.ico




index.html

3. Ejecutar el proyecto. El navegador debe mostrar la plantilla codificada en index.html sin errores

**Llegar a este punto= +0.5 puntos**


4. Agregar la carpeta *img* que están dentro de la carpeta 2\_actividadDos en WebContent
5. Modificar la página *index.jsp* para que se visualice como sigue:




[Inicio](#)
[Productos](#)
[Salir](#)

## La Ropería

[Mujer](#)
[Hombre](#)
[Accesorios](#)






**Blusa Paola Bossi PB1528 manga corta-azul**

**\$130.99**

Blusa de manga corta, en color azul marino, detalle de malla en la espalda y sublimado de flores en bajos esquinas, excelente para esta primavera

★★★★☆




**Blusa Manga Larga Milan Mujer Ejecutivo -Tinto**

**\$382.99**

MILAN DACACHE. Diseño moderno con el fit perfecto para la mujer de hoy, combinado con cualquier elemento haciendo brillar tu look empresarial.

★★★★☆




**Camisa Chic Turn Down-Vino Rojo**

**\$162.22**

Elegante gira el collar abajo camisa de manga larga color puro para los hombres

★★★★☆




**Camisa Casual Botón Pulsado (azul Marino)**

**\$289.88**

Camisa de vestir modelo de la tela clásica botón casual Camisa de los hombres

★★★★☆




**Reloj Montblanc Sport Chronograph**

**\$409,400.00**

Este reloj cronógrafo satisface las más altas exigencias tanto en términos de deporte extremo como en apariencia. Elaborado de tantalio y oro rosa de 18K.

★★★★☆



**Gafas de sol Christian Dior**

**\$289.99**

SKU: CH344FA190OECLMX, Modelo: 5886538L, Color: Plateado

★★★★☆

Copyright © Sinda Corp

6. Crear la página *libreria.jsp* dentro de WebContent

7. Copiar todo el contenido dentro de <head>...</head> con excepción del tag <title> de la página *index.jsp* en *libreria.jsp*

8. Incluye la página *librería.jsp* dentro de los tag <head>...</head> de la página *index.jsp*

9. Crear la página *menu.jsp* dentro de WebContent

10. Copiar todo el tag <nav>...</nav> de la página *index.jsp* en *menu.jsp*

11. Incluye la página *menu.jsp* dentro de los tag <body>...</body> de la página *index.jsp* de forma correcta

12. Crear la página *pie.jsp* dentro de WebContent
13. Copiar todo el tag `<footer>...</footer>` y `<script>...</script>` de la página *index.jsp* en *pie.jsp*
14. Incluye la página *pie.jsp* dentro de los tag `<body>...</body>` de la página *index.jsp* de forma correcta

**Llegar a este punto= +2 puntos**

15. Crear un paquete llamado *modelo*, y dentro una clase con nombre *ProductoModelo.java*
16. Crear y encapsular los siguientes atributos privados de la clase *ProductoModelo.java*  
String *pathImg*, String *nombreProducto*, Double *precio*, String *descripcion*
17. Crear un paquete llamado *beans* en *src*. Dentro del paquete, crear un *JavaBean* llamado *CargalInicialBean.java*
18. En *CargalInicialBean* declare los siguientes atributos como privados:

```
List<ProductoModelo> listaProductos = new ArrayList<ProductoModelo>();
String cargaInicial;
```

19. Crear los métodos *get* y *set* correspondientes los atributos de *CargalInicialBean*.
20. Dentro del cuerpo del método *get* del atributo *cargalInicial* debe crear un objeto *ProductoModelo* por cada producto mostrado en la página *index.jsp* en total son 6 productos. Por cada *ProductoModelo* creado debe agregarlo al objeto *listaProductos*, ejemplo:

```
ProductoModelo blusa1 = new ProductoModelo();
blusa1.setPathImg("img/blusa1.jpg");
blusa1.setNombreProducto("Blusa Paola Bossi PB1528 manga corta-azul");
blusa1.setPrecio(130.99);
blusa1.setDescripcion("Blusa de manga corta, en color azul marino, detalle de malla en la espalda y sublimado de flores en bajos esquinas, excelente para esta primavera");
listaProductos.add(blusa1);
```

21. La página *index.jsp* utiliza un bloque de código html para mostrar cada producto de la forma siguiente:

```
<div class="col-lg-4 col-md-6 mb-4">
  <div class="card h-100">
    <a href="#"></a>
    <div class="card-block">
      <h4 class="card-title"><a href="#">Blusa Paola Bossi PB1528 manga corta-azul</a></h4>
      <h5>$130.99</h5>
      <p class="card-text">Blusa de manga corta, en color azul marino, detalle de malla en la
      espalda y sublimado de flores en bajos esquinas, excelente para esta primavera</p>
    </div>
    <div class="card-footer">
      <small class="text-muted">&#9733; &#9733; &#9733; &#9733; &#9734;</small>
    </div>
  </div>
</div>
```

Debe crear una página jsp con nombre *productos.jsp* dentro de WebContent/vista para utilizar como *JavaBean* a *CargalInicialBean.java* con id *carga*, llamar a la propiedad *cargalInicial* y agregar el siguiente código:

```
<%
List<ProductoModelo> listaProductos = carga.getListProductos();
for(ProductoModelo pm: listaProductos) {
%>

<div class="col-lg-4 col-md-6 mb-4">
  <div class="card h-100">
    <a href="#"></a>
    <div class="card-block">
      <h4 class="card-title"><a href="#"><%=pm.getNombreProducto() %></a></h4>
      <h5><%=pm.getPrecio() %></h5>
      <p class="card-text"><%=pm.getDescripcion() %></p>
    </div>
    <div class="card-footer">
      <small class="text-muted">&#9733; &#9733; &#9733; &#9733;&#9734;</small>
    </div>
  </div>
</div>

<%
}&
%>
```

22. Sustituir todo el código HTML en index.jsp que haga referencia a mostrar los productos por el llamado a la página *vista/productos.jsp* de forma correcta. La página index.jsp debe visualizarse de la misma forma.

**Llegar a este punto= +3.5 puntos**

23. Utilice el siguiente script de base de datos para el manejador MySQL

```
CREATE DATABASE tienda_en_linea;

USE tienda_en_linea;

CREATE TABLE productos(
  id INT AUTO_INCREMENT PRIMARY KEY,
  pathImg CHAR(20) NOT NULL,
  nombreProducto CHAR(20) NOT NULL,
  precio double,
  descripcion CHAR(200) NOT NULL
);
```

24. Realice todo lo requerido en código java para que al utilizar a un método llamado ***getProductosEnDB()***; en la JSP *productos.jsp* del JavaBean *CargaInicialBean* obtenga los seis productos mostrados anteriormente. Para lograr este último punto debe realizar las sentencias *insert* y *select* necesarias, así como, utilizar su conexión a la base de datos. Parte del código de la JSP *productos.jsp* se visualizaría de la siguiente forma:

```
<%
List<ProductoModelo> listaProductos = carga.getProductosEnDB();
for(ProductoModelo pm: listaProductos) {
%>
```

```
<div class="col-lg-4 col-md-6 mb-4">
  <div class="card h-100">
    <a href="#"></a>
    <div class="card-block">
      <h4 class="card-title"><a href="#"><%=pm.getNombreProducto() %></a></h4>
      <h5><%=pm.getPrecio() %></h5>
      <p class="card-text"><%=pm.getDescripcion() %></p>
    </div>
    <div class="card-footer">
      <small class="text-muted">&#9733; &#9733; &#9733; &#9733;&#9734;</small>
    </div>
  </div>
</div>

<%
}
%>
```

**Llegar a este punto= +4 puntos**