

CURSO : LENGUAJE DE PROGRAMACIÓN II
PROFESOR : JOSE GARCIA LA RIVA
SEMESTRE : 2024 – I
CICLO : CUARTO
FECHA : 25/02/2024
DURACIÓN : 60 MINUTOS

NOTA

ALUMNO (A) : Raul Felix Lopez Villarreal

EVALUACIÓN DE LABORATORIO 2 (T2)

Consideraciones generales:

- Considerar el orden, la limpieza y la claridad de las respuestas.
- NO está permitido el uso o consulta de cuadernos, separatas, libros o cualquier material de la asignatura durante el desarrollo de la evaluación.
- Se bonificará con un punto adicional en caso no se encuentre ninguna falta ortográfica en el desarrollo del examen.
- Todas las respuestas deberán aparecer dentro del espacio señalado para tal fin.

LOGRO DE LA EVALUACION:

Al término de la unidad, el alumno está en la capacidad de analizar e implementar aplicaciones web haciendo uso de Spring Framework con conexión de base de datos con MySQL.

CONSOLIDADO

Pregunta	Puntaje		Llenar solo en caso de Recalificación justificada	
	Máximo	Obtenido	Sustento	Puntaje
1	20			
Total	20			

CARRITO DE COMPRAS BASICO

Al iniciar la aplicación se deberá visualizar todos los productos que se ofertan en la tienda virtual (Imagen 1), al hacer clic en el botón “**Agregar a Carrito**” se deberá añadir al cesto de compras el producto seleccionado, en caso haga clic en el botón “**Comprar**” deberá visualizar la pantalla de la Imagen 2”.

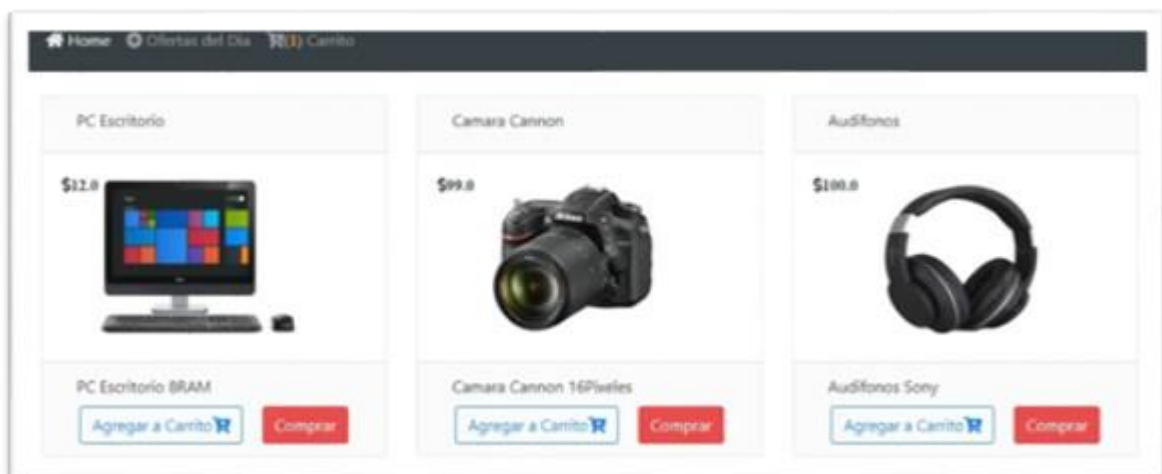


Imagen 1

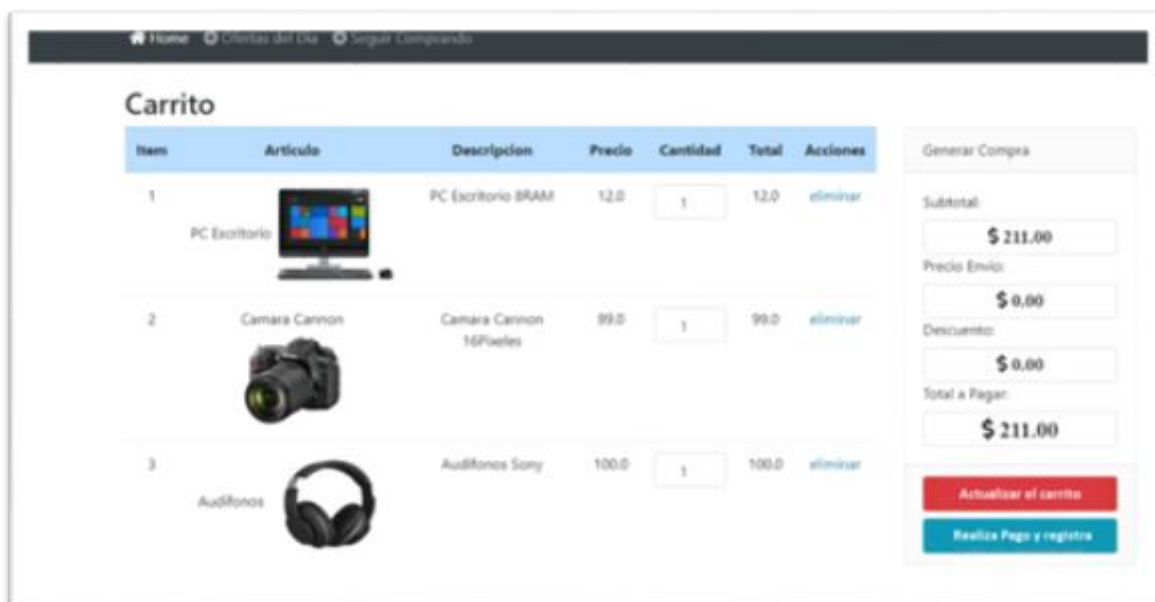


Imagen 2

Generar los códigos correspondientes (Entidades) para crear las tablas necesarias en MySQL para que el carrito visualice los productos que vende, en la vista previa del carrito deberá calcular el sub-total, si hubiese algún cálculo de precio envío y/o descuento deberá calcularlo y visualizarlo según indica la Imagen 2.

En la caja de texto de cada artículo se podrá cambiar la cantidad adquirir y deberá realizarse un recalcu lo al presionar el botón “**Actualizar el carrito**”. La opción “**Realizar pago y registrar**” solo deberá visualizar un mensaje que se proceso y además deberá grabar la información del o de los productos adquiridos en las tablas correspondientes.

RUBRICA DE EVALUACIÓN				
	5 puntos	3 puntos	2 puntos	0 puntos
5 puntos	Muestra en la vista los datos solicitados en forma correcta.	Mapea las entidades según el modelo solicitado de forma correcta pero no muestra en la vista los datos solicitados.	Mapea las entidades según el modelo solicitado, pero con errores	No existe ninguna implementación.
	6 puntos	3 puntos	2 puntos	0 puntos

6 puntos	Implementa el proceso de actualizar correctamente mostrando el mensaje de éxito.	Muestra en un modal los datos del supervisor en forma completa.	Muestra en un modal los datos del supervisor en forma parcial.	No existe ninguna implementación.
	6 puntos	3 puntos	2 puntos	0 puntos
6 puntos	Implementa el proceso de eliminar correctamente mostrando los mensajes respectivos.	Implementa la ruta en el controlador para eliminar, lo invoca desde la vista, pero no realiza el proceso.	Implementa la ruta en el controlador para eliminar.	No existe ninguna implementación.
	3 puntos	2 puntos	1 punto	0 puntos
3 puntos	Muestra de forma correcta la posición actual del registro en la tabla y el total de registro en la tabla.	Muestra de forma parcial la posición actual del registro en la tabla y el total de registro en la tabla.	Implementa lo solicitado de forma correcta.	No existe ninguna implementación.

```

package org.cibertec.edu.pe.controller;

import java.util.ArrayList;
import java.util.List;

import org.cibertec.edu.pe.model.Detalle;
import org.cibertec.edu.pe.model.Producto;
import org.cibertec.edu.pe.model.Venta;
import org.cibertec.edu.pe.repository.IDetalleRepository;
import org.cibertec.edu.pe.repository.IProductoRepository;
import org.cibertec.edu.pe.repository.IVentaRepository;
import org.springframework.beans.factory.annotation.Autowired;
import org.springframework.stereotype.Controller;
import org.springframework.ui.Model;
import org.springframework.web.bind.annotation.GetMapping;
import org.springframework.web.bind.annotation.ModelAttribute;
import org.springframework.web.bind.annotation.PathVariable;
import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.SessionAttributes;

@Controller
@SessionAttributes({"carrito", "total"})

public class ProductoController {
    @Autowired
    private IProductoRepository productoRepository;
    @Autowired
    private IVentaRepository ventaRepository;
    @Autowired
    private IDetalleRepository detalleRepository;

```

```

@GetMapping("/index")
public String listado(Model model) {
    List<Producto> lista = new ArrayList<>();
    lista = productoRepository.findAll();
    model.addAttribute("productos", lista);
    return "index";
}

@GetMapping("/agregar/{idProducto}")
public String agregar(Model model, @PathVariable(name = "idProducto",
required = true) int idProducto) {
    // Codigo para agregar un producto
    Producto maquina =
productoRepository.findById(idProducto).orElse(null);
    if (maquina != null) {
        Detalle detalle = new Detalle();
        detalle.setProducto(maquina);
        detalle.setCantidad(1);
        detalleRepository.save(detalle);

        List<Detalle> carrito = (List<Detalle>)
model.getAttribute("carrito");
        if (carrito == null) {
            carrito = new ArrayList<Detalle>();
            model.addAttribute("carrito", carrito);
        }
        carrito.add(detalle);

        double total = (double) model.getAttribute("total");
        if (maquina != null) {
            total += maquina.getPrecio();
            model.addAttribute("total", total);
        }
    }
    return "redirect:/index";
}

@GetMapping("/carrito")
public String carrito() {
    return "carrito";
}

@GetMapping("/pagar")
public String pagar(Model model) {
    // Codigo para pagar
    return "pagar";
}

@PostMapping("/actualizarCarrito")
public String actualizarCarrito(Model model) {
    // Codigo para actualizar el carrito
    return "carrito";
}

// Inicializacion de variable de la sesion
@ModelAttribute("carrito")

```

```
public List<Detalle> getCarrito() {  
    return new ArrayList<Detalle>();  
}  
  
@ModelAttribute("total")  
public double getTotal() {  
    return 0.0;  
}  
}
```