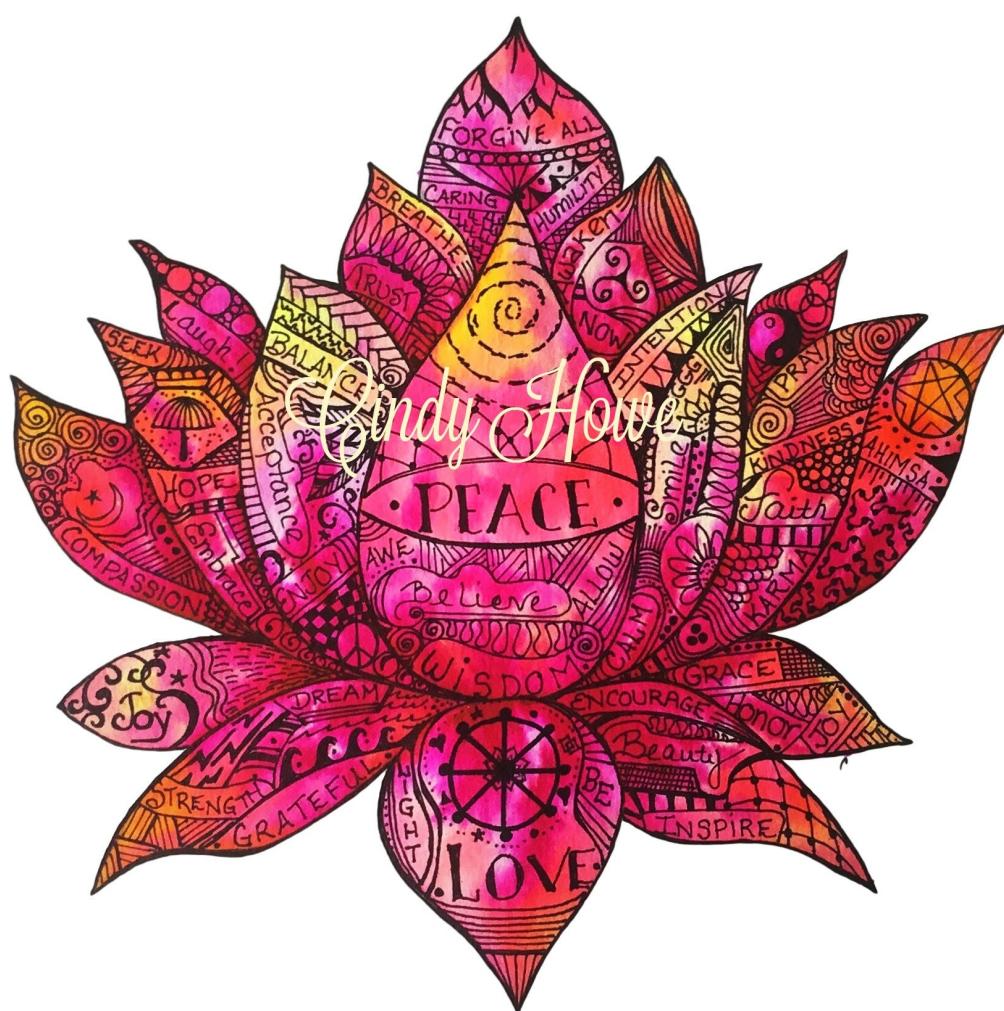


Vicente Villarreal Hinojo

2ºDAW



Para poder empezar esta tarea vamos a clonar el repositorio de Chisela en nuestro workspace y vamos a comenzar con la instalación, ejecutaremos el script en base de datos y procederemos a primero probar la aplicación.

Antes de nada, en el Build Path tenemos que cambiar la librería del servidor Tomcat a nuestra versión Tomcat, puesto que la compañera tenía una versión del Tomcat 9.0.55 y Yo utilice la 9.0.62, en el Project Facets cambiamos también el runtimes de la versión 55 a la 62.

En la instalación en el momento de ejecutar el script en la base de datos, he encontrado un problema en la línea 2 de los insert into de la tabla FLOWER, ya que los datos introducidos en el script son muy largos para la capacidad de la columna, para cambiarlo he modificado el campo **description** de (100) a (200).

No entiendo por qué al registrarte con un usuario la fecha tiene que ser inferior a 2010. Supongo que la edad mínima sería de 12 años.

En el caso de no tener stock algún producto, debería informarnos con un mensaje de que no tiene stock.

```
if(flower.getStock()>0) {
    buttonsDiv.append("<form action='CartServlet' method='post'>
        + "<input type='number' name='quantity' min='1' max='" + flower.getStock() + "' size='4'>
        + "<button name='flower' value='" + flower.getCode() + "'><img src='media/cart.svg' width='
        + "</form>");}
else []
    buttonsDiv.append("<h1><font color='black' size='5'>Sin stock</font></h1>");}
buttonsDiv.append("</div>");
```

Su programa estaba diseñado para que cuando no hubiese stock se quedase vacío en blanco, por lo cuál añadiendo ese else le mostramos al cliente que no tenemos stock.

Los artículos añadidos no se pueden comprar con un usuario administrador, y con otro que no es administrador, tampoco. Esto es debido a que se crean siempre sin stock, ya que en el formulario no lo recoge ese campo, y por tanto a la hora de crear el artículo tampoco recupera el parámetro.

Comenzaremos añadiendo el campo en el formulario y por lo tanto, lo siguiente que haremos será recuperar el parámetro y cambiar en el constructor del artículo (flores) añadir el campo stock.

```

    </div>
    <div class="form-group">
        <label class="form-label" for="stock">stock</label> <input type="number"
            class="form-input" placeholder="1" name="stock" required>
    </div>
<form>

```

Añadimos el input del stock.

```

public Flower(String name, String description, float price, Color color, int stock) {
    this.name = name;
    this.description = description;
    this.price = price;
    this.color = color;
    this.stock = stock;
}

```

Le añadimos el campo stock.

```

    Integer stock = Integer.valueOf(request.getParameter("stock"));
    //Creo el objeto y lo guardo en la base de datos
    Flower newFlower = new Flower(name, description, price, color, stock);
    try{
        FlowerControl.addFlower(newFlower);%
        Artículo añadido
    }

```

y añadimos finalmente el artículo con el stock.

Cuando no se añade nada al carrito y navegamos directamente al carrito, en este último se muestra el botón de comprar, que si le damos a comprar la aplicación falla por que el mapa es nulo, se queda en el servlet y no redirige ni te deja hacer alguna otra opción.

Se ha arreglado de forma sencilla, añadiendo un if en la página del jsp de carrito y controlando que no me pasen valores nulos o vacíos, si es así no me mostrará el botón de comprar.

```

<%}>
<div class="cart-buttons">
    <a href="LoginServlet">Volver al catálogo</a>
    <form method="post" action="PurchaseServlet">
        <p>Total: <%= total %>€</p>
        <%if(cart==null||cart.getItems()==null||cart.getItems().keySet().isEmpty()||
           cart.getItems().keySet()==null||cart.getItems().isEmpty()){>
            <br>
        }else{%
            <button>Comprar</button>
        }
    <% }%>

```

Por lo tanto si no añadimos ningún artículo no me muestra la opción de comprar.



En cambio si añadimos al carrito algún artículo, ya me mostraría la opción de comprar.



Además como se puede ver en la imagen de arriba a parte de mostrarte el botón de comprar, le hemos añadido un botón de vaciar el carrito para que el usuario no se vea en la obligación de comprar. Este botón simplemente redirige a un jsp donde se recupera el carrito de la sesión y se limpia.

```
<p>Total: <%= total %>€</p>
<%if(cart==null||cart.getItems()==null||cart.getItems().keySet().isEmpty()||
   cart.getItems().keySet()==null||cart.getItems().isEmpty()){

}else{%
    <button>Comprar</button>
}<% }%>

<!-- <button>Comprar</button> -->
</form>
<a href="vaciarCarrito.jsp"><button>Vaciar carrito</button></a>
</div>
```

Botón de vaciar el carrito.

```

<title>Insert title here</title>
</head>
<body>

<%
//Se recupera el mapa (carrito) guardado en la session.
Cart carrito = (Cart) session.getAttribute("cart");

//Se recuperan los Items y se limpian.
carrito.getItems().keySet().clear();

//Se redirige a la lista de flores.
response.sendRedirect("LoginServlet");

%>

```

Clase Jsp de vaciar el carrito.

Para hacer la tarea descrita en la actividad, en lo relativo a la fecha, solo se podrán registrar usuarios mayores de 18 años. por lo cuál cambiaremos en el **setDateOfBirth** del Usuario la siguiente línea:

```

public void setDateOfBirth(LocalDateTime dateOfBirth) throws UserException {
    LocalDateTime MIN = LocalDateTime.parse("1920-01-01T00:00:00.00");
    LocalDateTime MAX = LocalDateTime.parse("2010-01-01T00:00:00.00");
    LocalDate MIN = LocalDate.parse("1920-01-01");
    LocalDate MAX = LocalDate.of((LocalDate.now().getYear()-18), LocalDate.now().getMonth(), LocalDate.now().getDayOfMonth());
    if(dateOfBirth == null) {
        throw new UserException("La fecha introducida no es válida");
    }else {
        if(dateOfBirth.isBefore(MIN) || dateOfBirth.isAfter(MAX)) {
            throw new UserException("La fecha introducida no puede ser menor que " + MIN + " ni mayor que " + MAX);
        }else {
            this.dateOfBirth = dateOfBirth;
        }
    }
}

```

Hemos cambiado la línea del LocalDate MAX, añadiendo lo siguiente para que nuestro programa perdure en el tiempo.

Entonces con esta modificación si intentamos registrar un usuario más joven que por ejemplo el día de hoy del año 2004 que sería cuando justamente ha cumplido 18 años, nos dice lo siguiente:

Error

La fecha introducida no puede ser menor que 1920-01-01 ni mayor que 2004-11-30

[volver](#)

Para cambiar el nombre del proyecto, hemos marcado el nombre del proyecto y le hemos dado a refactor rename y hemos cambiado el nombre del proyecto a miTiendaModernaVillarrealHinojoVicente.

Posteriormente en el pom.xml le hemos cambiado el nombre del **groupId** y del **artifactId** a nuestro nuevo nombre del proyecto.

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <modelVersion>4.0.0</modelVersion>
  <groupId>miTiendaModernaVillarrealHinojoVicente</groupId>
  <artifactId>miTiendaModernaVillarrealHinojoVicente</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <packaging>war</packaging>
  <dependencies>
    <dependency>
      <groupId>mysql</groupId>
      <artifactId>mysql-connector-java</artifactId>
```