



Desarrollo web en entorno servidor
Acceso a datos

Actividad integradora 2EV
Desarrollo de una aplicación Web con Spring Boot
8 – Carro de compra sin sesiones

CONTENIDO

1.- Objetivos.....	2
2.- Requisitos.....	2
3.- Observaciones previas sobre la ausencia de usuarios y sesiones en la aplicación.....	2
4.- Tareas a realizar.....	2
4.1.- Definir la clase “producto en el carro”	2
4.2.- Relacionar el “producto en el carro” y productos.....	2
4.4.- Añadir un producto al carro de compra.....	3
4.5.- Listado del carro de compra.....	3
4.6.- Quitar producto del carro de compra.....	3
4.7.- Vaciar carro de compra.....	4

1.- Objetivos

- Objetivos de acceso a datos
 - Practicar el desarrollo de relaciones entre entidades en JPA.
 - Practicar el desarrollo de consultas derivadas o personalizadas en repositorios JPA.
- Objetivos de desarrollo web
 - Practicar el desarrollo de métodos de servicio, métodos de controlador y plantillas.

2.- Requisitos

Haber completado todas las partes anteriores de la actividad, y, más específicamente, disponer de una página de detalle de producto, donde se mostrará un enlace / botón para añadir un producto al carro de compra

3.- Observaciones previas sobre la ausencia de usuarios y sesiones en la aplicación

En este punto de la actividad aún no disponemos de sesiones o de un mecanismo para identificar a los usuarios. Esto tiene algunas consecuencias en lo que respecta al “carro de compra”:

- Los productos del carro de la compra no se asociarán a usuarios o sesiones de la web. Por así decirlo, de momento, habrá un único carro de compra “global”.
- Este carro de compra “global” se compartirá entre todos los usuarios. Si dos usuarios accedieran a la vez a la vez a la aplicación (ejemplo: abrir dos navegadores), y añadieran productos al carro, los dos verían añadidos estos productos.
- Se establecerán en el carro una serie de restricciones (indicadas más adelante) que habrá que modificar en un futuro, cuando se incorporen usuarios al sistema.

4.- Tareas a realizar

4.1.- Definir la clase “producto en el carro”

Para definir esta clase hay que tener en cuenta:

- Se debe elegir un nombre adecuado.
- Hay que definir un identificador de producto en el carro (la clave primaria). Debe ser obligatoriamente de tipo “Long” y debe llamarse “id” o “nombreDeClaseId”.
- Definir el resto de los atributos necesarios para la clase. Como mínimo, será necesario:
 - Número de unidades que el cliente quiere comprar.
 - Fecha y hora en que se añadió al carro.
 - Fecha y hora en que se modificó (se cambiaron unidades).
- No podrá añadirse dos veces al carro el mismo producto. Es decir, la tabla debe tener alguna clase de restricción para que no aparezca dos veces el mismo producto. Esta restricción deberá modificarse en un futuro, cuando se añadan sesiones / usuarios al sistema.

Usar las anotaciones JPA necesarias para que se cree una tabla en el sistema, con las columnas adecuadas, y con las restricciones necesarias.

4.2.- Relacionar el “producto en el carro” y productos

Realizar los cambios necesarios para que se establezca una relación bidireccional entre los productos en el carro y los productos, teniendo en cuenta:

- Un producto en el carro hará referencia a un solo producto (@ManyToOne).
- Un producto puede aparecer múltiples veces en el carro (@OneToMany).

En este sentido, esto puede parecer contradictorio con la restricción indicada en el punto 4.1, y podría parecer que lo más correcto es una relación @OneToOne, pero no lo es. Cuando en un futuro se modifique la restricción, para incluir criterios relacionados con sesiones y usuarios, esta relación @OneToMany Cobrará más sentido.

4.4.- Añadir un producto al carro de compra

Crear los elementos necesarios (controlador, servicio, repositorio, métodos en distintas clases, plantillas, etc.) para que se pueda añadir productos al carro de compra.

Para esto, tener en cuenta:

- La página de detalle de producto debe incluir un enlace para añadir una unidad del producto al carro.
- Si se intenta añadir un producto que ya está en el carro, el proceso no debe fallar. Se deben incrementar en 1 las unidades que se desean comprar.
- Opcionalmente, en los listados de productos (por categoría, todos los productos), se puede añadir un enlace para añadir directamente el producto al carro sin tener que abrir la página de detalle de producto.

4.5.- Listado del carro de compra

Crear los elementos necesarios (controlador, servicio, repositorio, métodos en distintas clases, plantillas, etc.) para que se pueda listar el carro de compra.

Para esto, tener en cuenta:

- En toda la web debe haber siempre un enlace para acceder al carro de compra. Este enlace dirigirá al usuario a una URL específica para el listado de productos en el carro.
- Se mostrarán los productos. De cada producto se mostrará:
 - Nombre
 - Unidades que se han añadido al carro
 - Precio unitario
 - Importe (unidades * precio unitario)
 - Enlace para eliminar el producto del carro de la compra.
- Además, se mostrará de alguna forma el total de la compra.
- Un enlace para vaciar el carro de compra

4.6.- Quitar producto del carro de compra

Crear los elementos necesarios (controlador, servicio, repositorio, métodos en distintas clases, plantillas, etc.) para que se pueda eliminar un producto del carro de la compra.

Para esto, tener en cuenta:

- El usuario hará clic en el enlace mencionado en el punto 5.5 para realizar esta operación.
- Tras eliminarse el producto del carro de la compra, se volverá a mostrar el carro. Para esto, hacer una redirección en el controlador, usando "redirect: ..."

4.7.- Vaciar carro de compra

Crear los elementos necesarios (controlador, servicio, repositorio, métodos en distintas clases, plantillas, etc.) para que se pueda vaciar completamente el carro de la compra.

Esta funcionalidad se activará desde un enlace que aparecerá en algún punto dentro de la página de listado del carro de la compra.

Se vuelve a mostrar el carro de compra (que estará vacío) usando redirect.