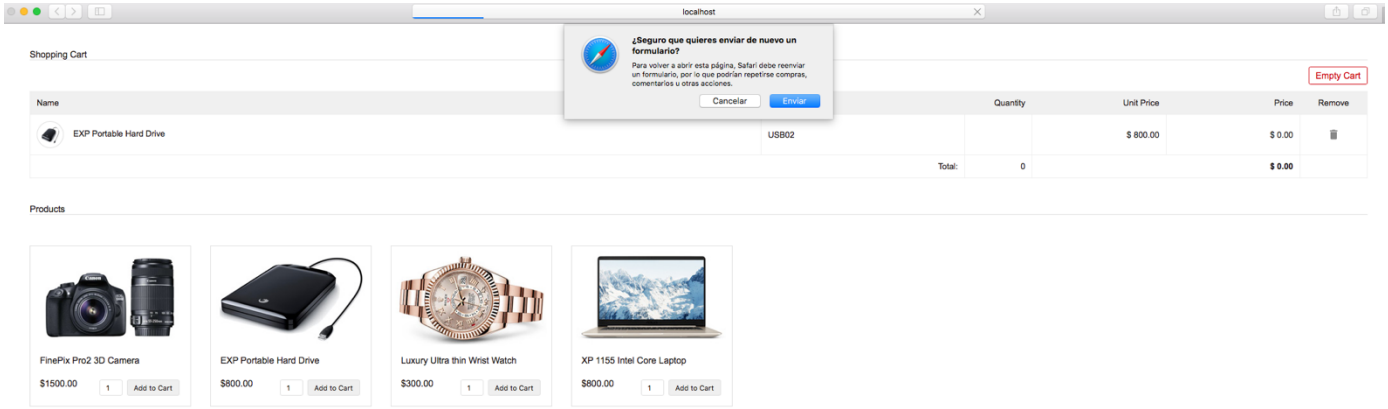
http://jgarmon.byethost4.com/shopping/index.php

Errores detectados

1. Posibilidad de inyección: por ejemplo en el input incluir 1;?><script>alert("¿?")</script><?php

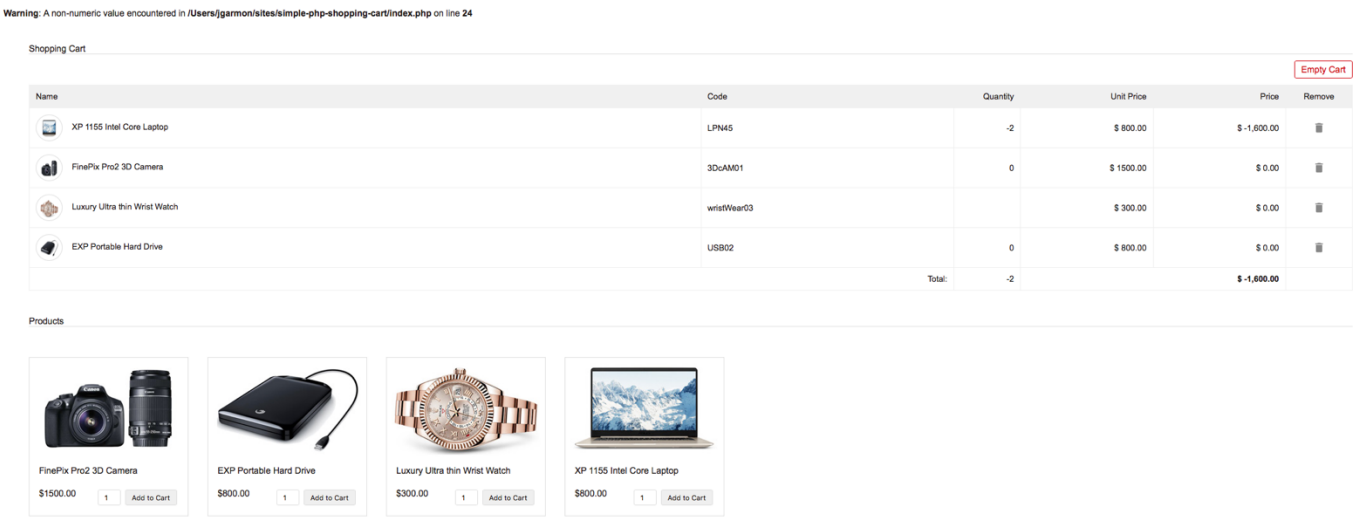


Para ello se hace un filtrado de datos. Una alternativa sería utilizar la sentencia **htmlentities($_POST["quantity"]);**

2. Entrada de datos errónea o no válida

Se pretende controlar que la entrada de datos correspondiente a la cantidad sea válida, esto es:

- Que no sea vacío
- Que no sea ni cero
- Que no sea negativo
- Que sea numérico

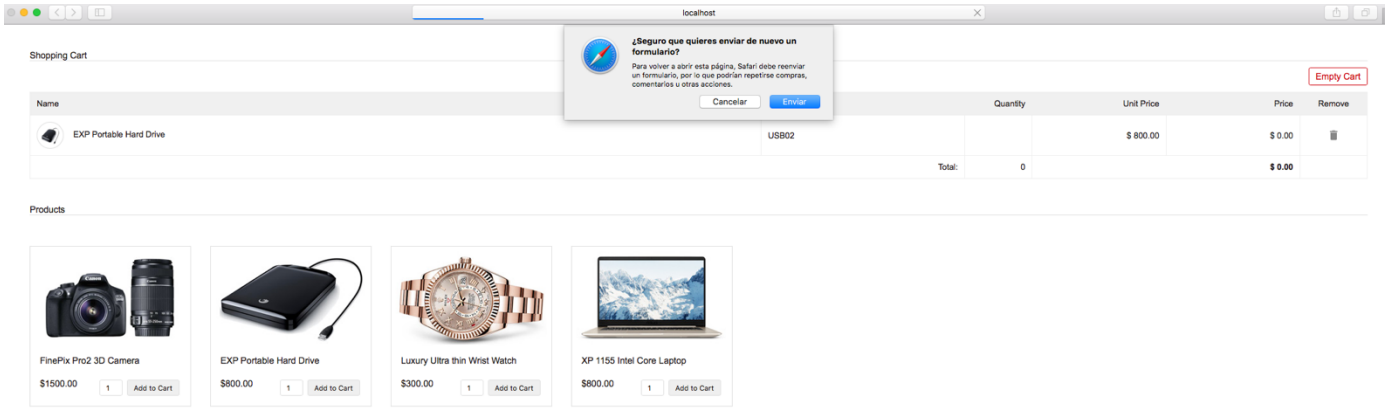


Para ello una forma de hacerlo es mediante el filtrado de entrada del campo $_POST[“quantity”]

```
    // El campo numérico se prepara para eliminar lo que no sea un dígito o un signo
    $quantity = filter_input(INPUT_POST, 'quantity', FILTER_SANITIZE_NUMBER_INT);
```

Existe la posibilidad de utilizar expresiones regulares para la verificación de datos o el filter_input FILTER_VALIDATE_INT con opciones concretas de rango de valores.

3. Recarga de datos que incrementa a cantidad sin control.
   Este error se solventa mediante la aplicación del POST-REDIRECT-GET. Esto es que cada vez que se valida la entrada de dato vía $_POST se redirecciona a la página como GET, para que no se cargue de nuevo el contenido. El hecho de que el almacenamiento del contenido se haga a través de $_SESSION facilita el funcionamiento.



El código corregido sería como sigue:

```
if(!empty($_GET["action"])) {
switch($_GET["action"]) {
    case "add":
        // Verificamos la entrada de datos para evitar inyección por ejemplo
        // El campo numérico se prepara para eliminar lo que no sea un dígito o un signo
        $quantity = filter_input(INPUT_POST, 'quantity', FILTER_SANITIZE_NUMBER_INT);
        //if(!empty($_POST["quantity"])) {
        if(!empty($quantity) && $quantity>0) {      // verificamos que hay contenido, que es mayor que 0
            $productByCode = $db_handle->runQuery("SELECT * FROM tblproduct WHERE code='" . $_GET["code"] . "'");
            $itemArray = array($productByCode[0]["code"]=>array('name'=>$productByCode[0]["name"], 'code'=>$productByCode[0]["code"],
'quantity'=>$quantity/*$_POST["quantity"]*/, 'price'=>$productByCode[0]["price"], 'image'=>$productByCode[0]["image"]));

            if(!empty($_SESSION["cart_item"])) {
                if(in_array($productByCode[0]["code"],array_keys($_SESSION["cart_item"]))) {
                    foreach($_SESSION["cart_item"] as $k => $v) {
                            if($productByCode[0]["code"] == $k) {
                                if(empty($_SESSION["cart_item"][$k]["quantity"])) {
                                    $_SESSION["cart_item"][$k]["quantity"] = 0;
                                }

                                $_SESSION["cart_item"][$k]["quantity"] += $quantity/*htlmentities($_POST["quantity"])*/;
                            }
```

```
                }
            } else {
                $_SESSION["cart_item"] = array_merge($_SESSION["cart_item"],$itemArray);
            }
        } else {
            $_SESSION["cart_item"] = $itemArray;
        }
    }
    header("Location: ".$_SERVER['PHP_SELF']);      // forzamos POST-REDIRECT-GET para evitar incrementar el carro en cada refresco
    break;
    case "remove":
        if(!empty($_SESSION["cart_item"])) {
            foreach($_SESSION["cart_item"] as $k => $v) {
                    if($_GET["code"] == $k)
                        unset($_SESSION["cart_item"][$k]);          // elimina el elemento del carro
                    if(empty($_SESSION["cart_item"]))
                        unset($_SESSION["cart_item"]);              // si el carro está cacio lo elimina
            }
        }
    header("Location: ".$_SERVER['PHP_SELF']);  // Este caso realmente no es necesario porque no recive $_POST/$_GET
    break;
    case "empty":
        unset($_SESSION["cart_item"]);
    header("Location: ".$_SERVER['PHP_SELF']);   // Este caso realmente no es necesario porque no recive $_POST/$_GET
    break;
}
exit;
}
```