

67. Usando cookies en Laravel para resolver el carrito de compras

Con el método implementado como hasta ahora, cada vez que pulsemos el botón se va a crear un nuevo carrito. Esto, por supuesto, no es lo que queremos, ya que es extremadamente ineficiente.

✓ AY MI COOKIE

Por suerte, las cookies han llegado para ayudarnos.

Creamos y anexamos la cookie

En ProductCartController:

```
<?php

namespace App\Http\Controllers;
use App\Models\Cart;
use App\Models\Product;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Cookie;

class ProductCartController extends Controller
{
    // BORRAMOS TODAS LAS FUNCIONES MENOS STORE Y DESTROY
    // Las funciones se crean automáticamente

    /**
     * Store a newly created resource in storage.
     *
     * @param  \Illuminate\Http\Request  $request
     * @param  \App\Models\Product  $product
     * @return \Illuminate\Http\Response
     */
    public function store(Request $request, Product $product)
    {
        // Creo que el nombre lo explica to bien
        $cart = $this->getFromCookieOrCreate();

        $cart = Cart::create();
        // Se accede a la cantidad de un determinado
        // producto que hay en el carrito
        $quantity = $cart->products()
            ->find($product->id)
            ->pivot
            ->quantity ?? 0;

        // Se crea un carrito cada vez: no muy eficiente
        $cart->products()->syncWithoutDetaching([
            $product->id => ['quantity' => $quantity + 1],
        ]);

        // Nombre, valor, tiempo de vida (minutos)
        $cookie = Cookie::make('cart', $cart->id, 7 * 24 * 60);
```

```

    // Redirigimos pero con la cookie creada
    return redirect()->back()->cookie($cookie);
}

/**
 * Remove the specified resource from storage.
 *
 * @param \App\Models\Product $product
 * @param \App\Models\Cart $cart
 * @return \Illuminate\Http\Response
 */
public function destroy(Product $product, Cart $cart)
{
    //
}

public function getFromCookieOrCreate() {
    // Solo si la cookie existe y además coincide con un carrito
    // reconocido por el sistema (id) no se crea una nueva instancia

    // Se busca la cookie llamada cart
    $cartId = Cookie::get('cart');
    // Si la cookie no existe, esto será null
    $cart = Cart::find($cartId);
    // Si eso es null, o tiene una id no reconocida,
    // se ejecutará la parte derecha de las ?? y se
    // creará una nueva instancia de Cart
    return $cart ?? Cart::create();
}
}
?>

```