

84. Usa transacciones de la base de datos para operaciones complejas

En OrderController:

Queremos que, si un pedido falla al rellenarse de productos, no quede huérfano. Necesitamos una **transacción de SQL**, que deshará los cambios si algo falla dentro de la transacción.

En OrderPaymentController:

También.

```
/**
 * Store a newly created resource in storage.
 *
 * @param \Illuminate\Http\Request $request
 * @param \App\Order $order
 * @return \Illuminate\Http\Response
 */
public function store(Request $request, Order $order)
{
    return DB::transaction(function() use($order) {
        //PaymentService::handlePayment();

        $this->cartService->getFromCookie()->products()->detach();

        $order->payment()->create([
            'amount' => $order->total,
            'payed_at' => now(),
        ]);

        $order->status = 'payed';
        $order->save();

        return redirect()
            ->route('main')
            ->withSuccess("Thanks! Your payment for \${$order->total} was successful.");
    }, 5);
}
```

Cosas olvidadas

Crear el OrderPaymentController

No estaba creado, así que tuve que ir al GitHub del man y copiarlo.

Ruta main en web.php

Tampoco la tenía, por alguna razón, y había que poner:

```
Route::get('/', 'MainController@index')->name('main');
```