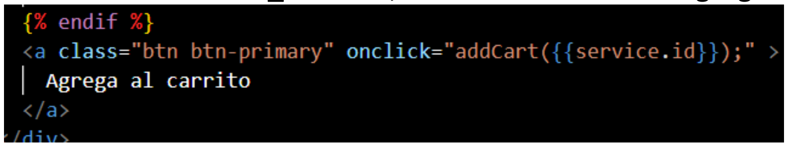
LocalStorage

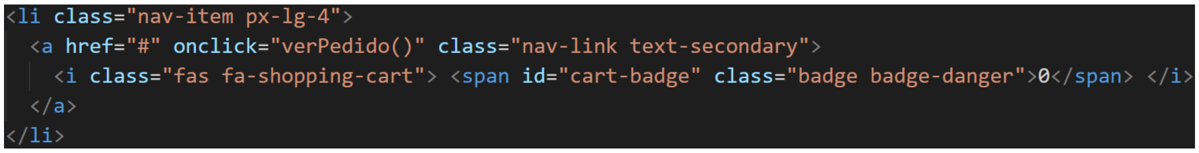
- Los datos persisten entre ventanas/tabs con el mismo origen.
- Las claves y los valores son **siempre cadenas de texto** (ten en cuenta que, al igual que con los objetos, las claves de enteros se convertirán automáticamente en cadenas de texto).
- Ejemplo
  - `localStorage.setItem('miGato', 'Juan');`
  - `var cat = localStorage.getItem('miGato');`
  - `localStorage.removeItem('miGato');`
  - `localStorage.clear(); //Elimina Todos los elementos`

En el archivo service_list.html, colocar el botón de “Agregar al carrito”

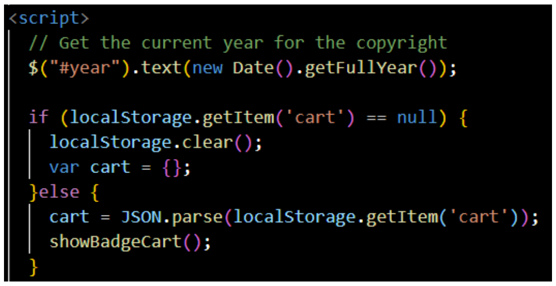


```
{% endif %}
<a class="btn btn-primary" onclick="addCart({{service.id}});" >
  Agrega al carrito
</a>
</div>
```

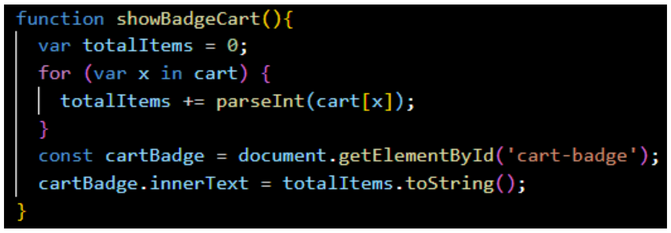
En el archivo base.html



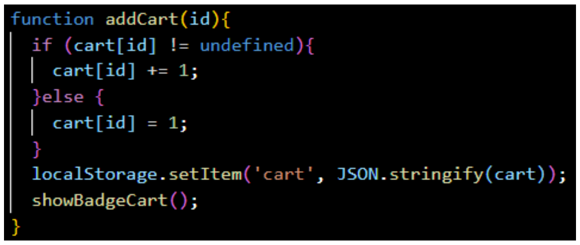
```
<li class="nav-item px-lg-4">
  <a href="#" onclick="verPedido()" class="nav-link text-secondary">
    <i class="fas fa-shopping-cart"> <span id="cart-badge" class="badge badge-danger">0</span> </i>
  </a>
</li>
```



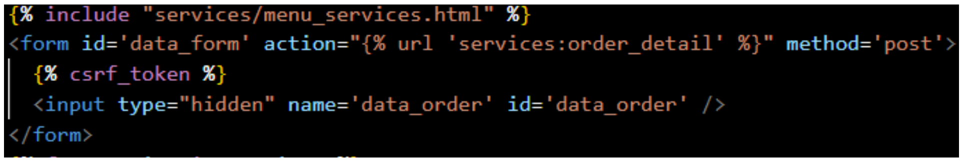
```
<script>
  // Get the current year for the copyright
  $("#year").text(new Date().getFullYear());

  if (localStorage.getItem('cart') == null) {
    localStorage.clear();
    var cart = {};
  }else {
    cart = JSON.parse(localStorage.getItem('cart'));
    showBadgeCart();
  }
```



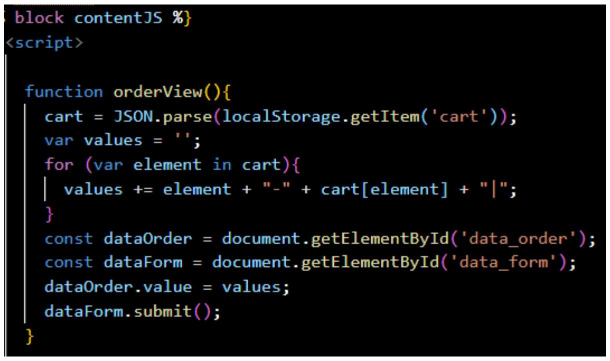
```
function showBadgeCart(){
  var totalItems = 0;
  for (var x in cart) {
    totalItems += parseInt(cart[x]);
  }
  const cartBadge = document.getElementById('cart-badge');
  cartBadge.innerText = totalItems.toString();
}
```



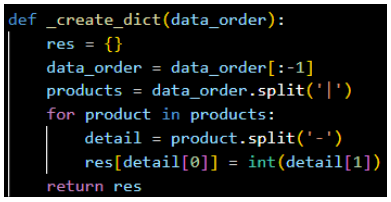
```
function addCart(id){
  if (cart[id] != undefined){
    cart[id] += 1;
  }else {
    cart[id] = 1;
  }
  localStorage.setItem('cart', JSON.stringify(cart));
  showBadgeCart();
}
```



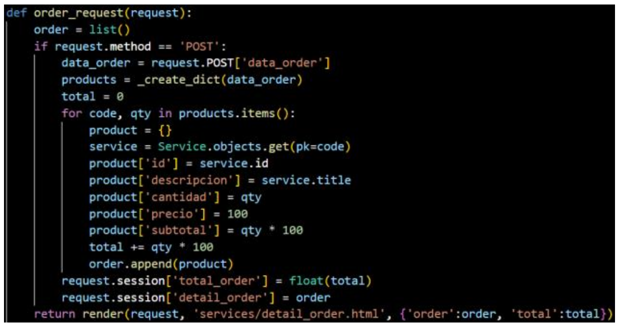
```
{% include "services/menu_services.html" %}
<form id='data_form' action="{% url 'services:order_detail' %}" method='post'>
  {% csrf_token %}
  <input type="hidden" name='data_order' id='data_order' />
</form>
```



```
block contentJS %}
<script>

  function orderView(){
    cart = JSON.parse(localStorage.getItem('cart'));
    var values = '';
    for (var element in cart){
      values += element + "-" + cart[element] + "|";
    }
    const dataOrder = document.getElementById('data_order');
    const dataForm = document.getElementById('data_form');
    dataOrder.value = values;
    dataForm.submit();
  }
```

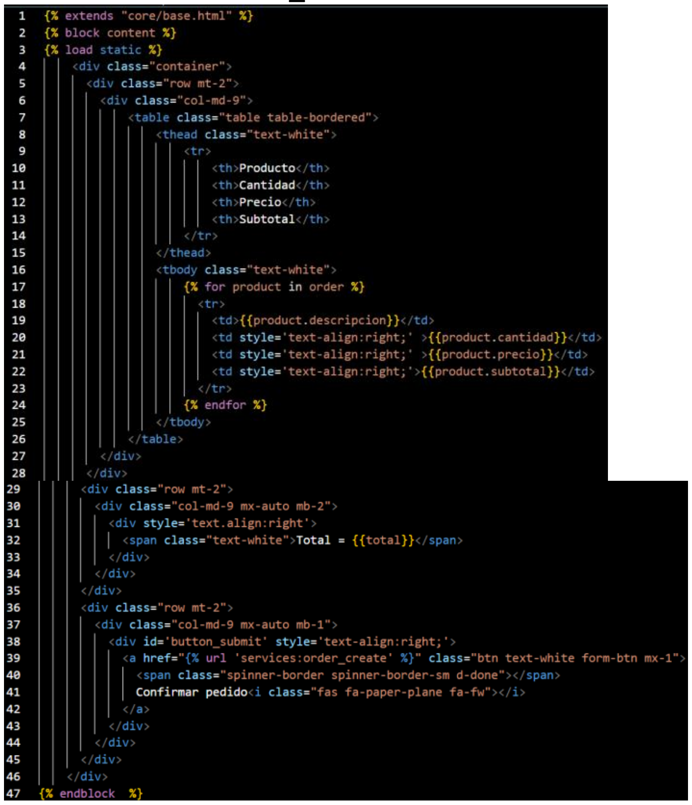


```
def _create_dict(data_order):
    res = {}
    data_order = data_order[:-1]
    products = data_order.split('|')
    for product in products:
        detail = product.split('-')
        res[detail[0]] = int(detail[1])
    return res
```



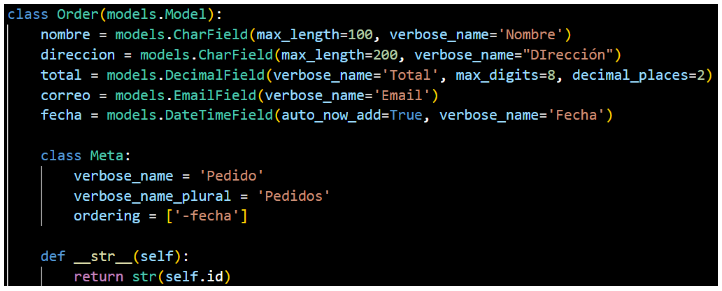
```
def order_request(request):
    order = list()
    if request.method == 'POST':
        data_order = request.POST['data_order']
        products = _create_dict(data_order)
        total = 0
        for code, qty in products.items():
            product = {}
            service = Service.objects.get(pk=code)
            product['id'] = service.id
            product['descripcion'] = service.title
            product['cantidad'] = qty
            product['precio'] = 100
            product['subtotal'] = qty * 100
            total += qty * 100
            order.append(product)
        request.session['total_order'] = float(total)
        request.session['detail_order'] = order
    return render(request, 'services/detail_order.html', {'order':order, 'total':total})
```

Crear el archivo detail_order.html



```
{% extends "core/base.html" %}
{% block content %}
{% load static %}
    <div class="container">
      <div class="row mt-2">
        <div class="col-md-9">
            <table class="table table-bordered">
                <thead class="text-white">
                    <tr>
                        <th>Producto</th>
                        <th>Cantidad</th>
                        <th>Precio</th>
                        <th>Subtotal</th>
                    </tr>
                </thead>
                <tbody class="text-white">
                    {% for product in order %}
                      <tr>
                        <td>{{product.descripcion}}</td>
                        <td style='text-align:right;' >{{product.cantidad}}</td>
                        <td style='text-align:right;' >{{product.precio}}</td>
                        <td style='text-align:right;'>{{product.subtotal}}</td>
                      </tr>
                    {% endfor %}
                </tbody>
            </table>
        </div>
      </div>
      <div class="row mt-2">
        <div class="col-md-9 mx-auto mb-2">
          <div style='text.align:right'>
            <span class="text-white">Total = {{total}}</span>
          </div>
        </div>
      </div>
      <div class="row mt-2">
        <div class="col-md-9 mx-auto mb-1">
          <div id='button_submit' style='text-align:right;'>
            <a href="{% url 'services:order_create' %}" class="btn text-white form-btn mx-1">
              <span class="spinner-border spinner-border-sm d-done"></span>
              Confirmar pedido<i class="fas fa-paper-plane fa-fw"></i>
            </a>
          </div>
        </div>
      </div>
    </div>
{% endblock  %}
```

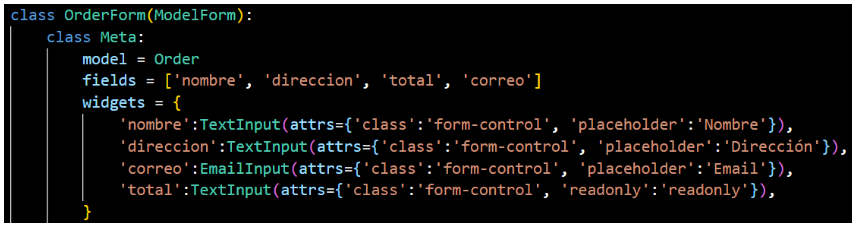
Crear el modelo Pedido



```python
class Order(models.Model):
    nombre = models.CharField(max_length=100, verbose_name='Nombre')
    direccion = models.CharField(max_length=200, verbose_name="DIrección")
    total = models.DecimalField(verbose_name='Total', max_digits=8, decimal_places=2)
    correo = models.EmailField(verbose_name='Email')
    fecha = models.DateTimeField(auto_now_add=True, verbose_name='Fecha')

    class Meta:
        verbose_name = 'Pedido'
        verbose_name_plural = 'Pedidos'
        ordering = ['-fecha']

    def __str__(self):
        return str(self.id)
```

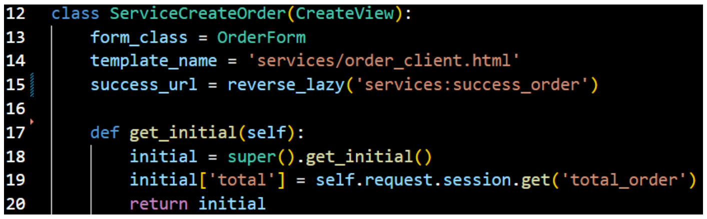
Realizar la migración a la base de datos.

Crear el formulario OrderForm



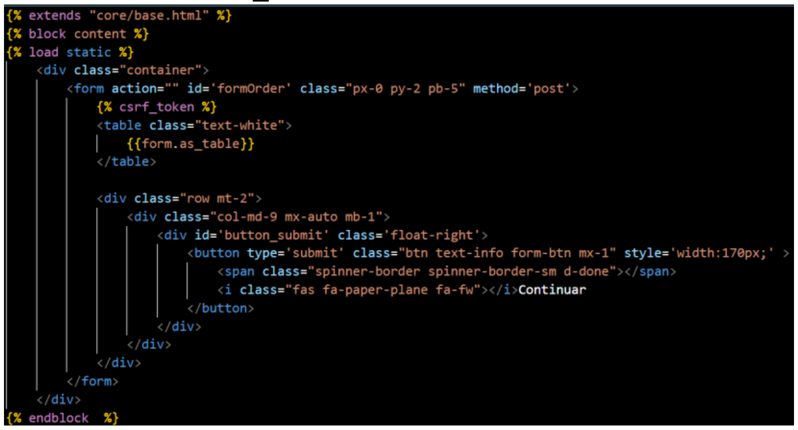
```python
class OrderForm(ModelForm):
    class Meta:
        model = Order
        fields = ['nombre', 'direccion', 'total', 'correo']
        widgets = {
            'nombre':TextInput(attrs={'class':'form-control', 'placeholder':'Nombre'}),
            'direccion':TextInput(attrs={'class':'form-control', 'placeholder':'Dirección'}),
            'correo':EmailInput(attrs={'class':'form-control', 'placeholder':'Email'}),
            'total':TextInput(attrs={'class':'form-control', 'readonly':'readonly'}),
        }
```

Crear la clase ServiceCreatePedido



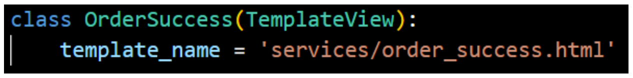
```python
class ServiceCreateOrder(CreateView):
    form_class = OrderForm
    template_name = 'services/order_client.html'
    success_url = reverse_lazy('services:success_order')

    def get_initial(self):
        initial = super().get_initial()
        initial['total'] = self.request.session.get('total_order')
        return initial
```

Crear el html order_cliente.html



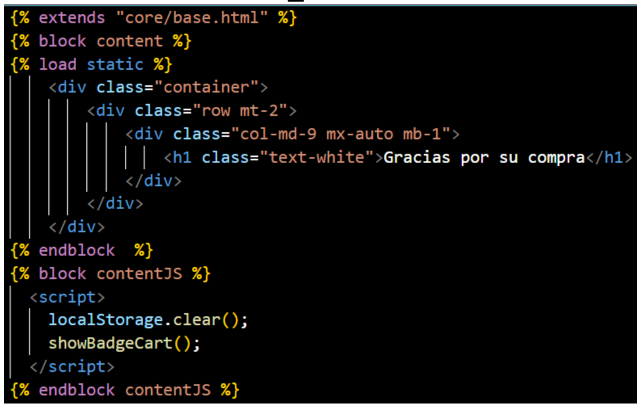
```
{% extends "core/base.html" %}
{% block content %}
{% load static %}
    <div class="container">
        <form action="" id='formOrder' class="px-0 py-2 pb-5" method='post'>
            {% csrf_token %}
            <table class="text-white">
                {{form.as_table}}
            </table>

            <div class="row mt-2">
                <div class="col-md-9 mx-auto mb-1">
                    <div id='button_submit' class='float-right'>
                        <button type='submit' class="btn text-info form-btn mx-1" style='width:170px;' >
                            <span class="spinner-border spinner-border-sm d-done"></span>
                            <i class="fas fa-paper-plane fa-fw"></i>Continuar
                        </button>
                    </div>
                </div>
            </div>
        </form>
    </div>
{% endblock  %}
```

Crear la clase PedidoSuccess



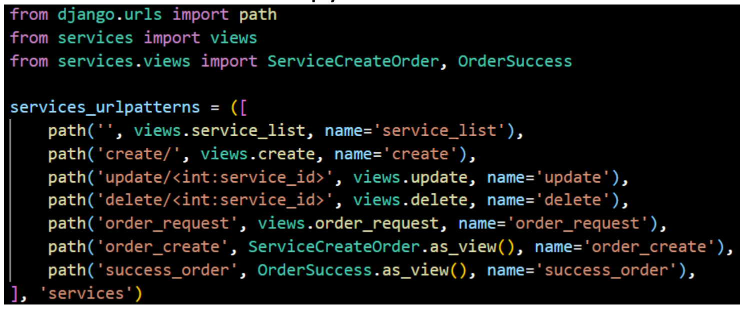
```
class OrderSuccess(TemplateView):
    template_name = 'services/order_success.html'
```

Crear el archivo order_success.html



```
{% extends "core/base.html" %}
{% block content %}
{% load static %}
    <div class="container">
        <div class="row mt-2">
            <div class="col-md-9 mx-auto mb-1">
                <h1 class="text-white">Gracias por su compra</h1>
            </div>
        </div>
    </div>
{% endblock  %}
{% block contentJS %}
  <script>
    localStorage.clear();
    showBadgeCart();
  </script>
{% endblock contentJS %}
```

Verificar el archivo urls.py



```
from django.urls import path
from services import views
from services.views import ServiceCreateOrder, OrderSuccess

services_urlpatterns = ([
    path('', views.service_list, name='service_list'),
    path('create/', views.create, name='create'),
    path('update/<int:service_id>', views.update, name='update'),
    path('delete/<int:service_id>', views.delete, name='delete'),
    path('order_request', views.order_request, name='order_request'),
    path('order_create', ServiceCreateOrder.as_view(), name='order_create'),
    path('success_order', OrderSuccess.as_view(), name='success_order'),
], 'services')
```