

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

```

```

16 def order_request(request):
17     order = list()
18     if request.method == 'POST':
19         data_order = request.POST['data_order']
20         print(data_order)
21         products = _create_dict(data_order)
22         total = 0
23         for code, qty in products.items():
24             product = {}
25             service = Service.objects.get(pk=code)
26             product['id'] = service.id
27             product['descripcion'] = service.title
28             product['cantidad'] = qty
29             product['precio'] = 100
30             total += qty * 100
31             order.append(product)
32         request.session['total_order'] = float(total)
33         request.session['detail_order'] = order
34     return render(request, 'services/detail_order.html', {'order':order, 'total':total})

```

Crear el archivo detail_order.html

```

3 {% load static %}
4 <div class="container">
5     <div class="row mt-2">
6         <div class="col-md-9">
7             <table class="table table-bordered">
8                 <thead class="text-white">
9                     <tr>
10                         <th>Producto</th>
11                         <th>Cantidad</th>
12                         <th>Precio</th>
13                         <th>Subtotal</th>
14                     </tr>
15                 </thead>
16                 <tbody class="text-white">
17                     {% for product in order %}
18                     <tr>
19                         <td>{{product.descripcion}}</td>
20                         <td style="text-align:right;" >{{product.cantidad}}</td>
21                         <td style="text-align:right;" >{{product.precio}}</td>
22                         <td style="text-align:right;" >{{product.subtotal}}</td>
23                     </tr>
24                     {% endfor %}
25                 </tbody>
26             </table>
27         </div>
28     </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>

```

```

34         <div class="row mt-1">
35             <div class="col-md-9 mx-auto mb-1">
36                 <div id="button_submit" style='text-align:right;' >
37                     <a href="#" class="btn text-white form-btn mx-1">
38                         <span class="spinner-border spinner-border-sm d-none"></span>
39                         <i class="fas fa-paper-plane fa-fw"></i>Confirmar Pedido
40                     </a>
41                 </div>
42             </div>
43         </div>
44     </div>
45 {% endblock content %}

```

Crear el modelo Pedido

```

21 class Pedido(models.Model):
22     nombre = models.CharField(max_length=100, verbose_name='Nombre')
23     direccion = models.CharField(max_length=200, verbose_name='Calle y número',
24     help_text='<span class="font-italic" style="font-size:0.9rem">Nombre de la calle y número</span>')
25     colonia = models.CharField(max_length=200, verbose_name='Colonia o Fraccionamiento',
26     help_text='<span class="font-italic" style="font-size:0.9rem">Nombre de la colonia o fraccionamiento</span>')
27     total = models.DecimalField(verbose_name='Total', max_digits=8, decimal_places=2,
28     help_text='<span class="text-danger" style="font-size:0.9rem">Cantidad a pagar</span>')
29     correo = models.EmailField(verbose_name='Email')
30     fecha = models.DateTimeField(auto_now_add=True, verbose_name='Fecha de creación')
31
32     class Meta:
33         verbose_name = 'Pedido'
34         verbose_name_plural = 'Pedidos'
35         ordering = ['-fecha']
36
37     def __str__(self):
38         return str(self.id)

```

Realizar la migración a la base de datos.

Crear el archivo forms.py

```

1  from django.forms import ModelForm, TextInput, EmailInput
2  from services.models import Service, Pedido
3
4  class PedidoForm(ModelForm):
5      def __init__(self, *args, **kwargs):
6          self.request = kwargs.pop('request', None)
7          #Se obtiene el total de la sesión del usuario
8          total_float = self.request.session.get('total_float')
9          super().__init__(*args, **kwargs)
10         self.fields['total'].initial = total_float
11
12     class Meta:
13         model = Pedido
14         fields = ['nombre', 'direccion', 'colonia', 'correo', 'total']
15         widgets = {
16             'nombre': TextInput(attrs={'class': 'form-control', 'placeholder': 'Nombre'}),
17             'direccion': TextInput(attrs={'class': 'form-control', 'placeholder': 'Calle y número'}),
18             'colonia': TextInput(attrs={'class': 'form-control', 'placeholder': 'Colonia o fraccionamiento'}),
19             'correo': EmailInput(attrs={'class': 'form-control', 'placeholder': 'Email'}),
20             'total': TextInput(attrs={'class': 'form-control', 'readonly': 'readonly'}),
21         }

```

Crear la clase ServiceCreatePedido

```
44 class ServiceCreatePedido(CreateView):
45     form_class = PedidoForm
46     template_name = 'services/pedido_cliente.html'
47     success_url = reverse_lazy('services:success_pedido')
48
49     def form_valid(self, form):
50         #Se guardan los datos del pedido
51         pedido_nuevo = form.save(commit=False)
52         pedido_nuevo.save()
53         return super().form_valid(form)
54
55     def get_form_kwargs(self):
56         #Se coloca el request, para poderlo obtener en la forma
57         kwargs = super(ServiceCreatePedido, self).get_form_kwargs()
58         kwargs['request'] = self.request
59         return kwargs
```

Crear el html pedido_cliente.html

```
1 {% extends "core/base.html" %}
2 {% load static %}
3 {% block content %}
4     <div class="container mb-4">
5         <form id='formPedido' action="" class="px-0 py-2 pb-5" method='post'>
6             {% csrf_token %}
7             <table class="text-white">
8                 {{form.as_table}}
9             </table>
10            <div class="row mt-0">
11                <div class="col-md-9 mx-auto mb-1">
12                    <div id='button_submit' class="float-right">
13                        <button type='submit' class="btn text-info form-btn mx-1" style='width:170px;' >
14                            <span class="spinner-border spinner-border-sm d-none"></span>
15                            <i class="fas fa-paper-plane fa-fw"></i>Continuar
16                        </button>
17                    </div>
18                </div>
19            </div>
20        </form>
21    </div>
22 {% endblock content %}
```

Crear la clase PedidoSuccess

```
61 class PedidoSuccess(TemplateView):
62     template_name = 'services/pedido_success.html'
```

Crear el archivo pedido_success.html

```
1 {% extends "core/base.html" %}
2 {% load static %}
3 {% block content %}
4     <div class="container mb-4">
5         <h1 class='text-white'>Gracias por su compra!</h1>
6     </div>
7 {% endblock content %}
8
9 {% block contentJS %}
10    <script>
11        localStorage.clear();
12        show_bagde.cart();
13    </script>
14 {% endblock contentJS %}
```

Verificar el archivo urls.py

```
1 from django.urls import path
2 from services import views
3 from services.views import ServiceListView, ServiceDetailView, ServiceCreateView, S
4
5 services_urlpatterns = ([
6     path('create_pedido', ServiceCreatePedido.as_view(), name='create_pedido'),
7     path('success_pedido', PedidoSuccess.as_view(), name='success_pedido'),
8     path('detalle_pedido/', views.realiza_pedido, name='detalle_pedido'),
9     path('detail/<pk>', ServiceDetailView.as_view(), name='service_detail'),
10    path('', ServiceListView.as_view(), name='service_list'),
11    path('create/', ServiceCreateView.as_view(), name='create'),
12    path('update/<int:pk>', ServiceUpdateView.as_view(), name='update'),
13    #path('create/', views.create, name='create'),
14    # path('update/<int:service_id>', views.update, name='update'),
15 ], 'services')
```

Realizar el cambio de href en el archivo detalle_pedido.html

```
<div class="row mt-1">
  <div class="col-md-9 mx-auto mb-1">
    <div id="button_submit" style='text-align:right;' >
      <a href="{% url 'services:create_pedido' %}" class="btn te
        <span class="spinner-border spinner-border-sm d-none">
        <i class="fas fa-paper-plane fa-fw"></i>Confirmar Pedido
      </a>
    </div>
  </div>
</div>
```