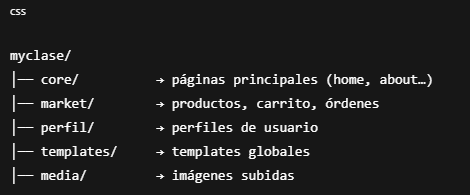
Guia Clase 6 – Estructura del proyecto



**📦 2. Modelo de productos (market/models.py)**

Ampliamos el modelo:

from django.conf import settings

from django.db import models

class Product(models.Model):

seller = models.ForeignKey(

settings.AUTH\_USER\_MODEL,

on\_delete=models.CASCADE,

related\_name="products"

)

title = models.CharField(max\_length=200)

description = models.TextField(blank=True)

marca = models.CharField(max\_length=100, blank=True, default="Genérico")

price = models.DecimalField(max\_digits=12, decimal\_places=2)

stock = models.PositiveIntegerField(default=1)

image = models.ImageField(upload\_to="product\_images/", blank=True, null=True)

active = models.BooleanField(default=True)

created\_at = models.DateTimeField(auto\_now\_add=True)

updated\_at = models.DateTimeField(auto\_now=True)

def \_\_str\_\_(self):

return self.title

def is\_available(self):

return self.active and self.stock > 0

**🛒 3. Modelo de carrito (market/models.py)**

**Agregamos un carrito simple:**

class Cart(models.Model):

user = models.OneToOneField(settings.AUTH\_USER\_MODEL, on\_delete=models.CASCADE)

def total(self):

return sum(item.subtotal() for item in self.items.all())

class CartItem(models.Model):

cart = models.ForeignKey(Cart, on\_delete=models.CASCADE, related\_name="items")

product = models.ForeignKey(Product, on\_delete=models.CASCADE)

quantity = models.PositiveIntegerField(default=1)

def subtotal(self):

return self.product.price \* self.quantity

**📋 4. Vistas principales (market/views.py)**

* Listar productos
* Crear producto
* Agregar al carrito
* Checkout con Mercado Pago

from django.shortcuts import render, redirect, get\_object\_or\_404

from django.contrib.auth.decorators import login\_required

from .models import Product, Cart, CartItem

from .forms import ProductForm

@login\_required

def product\_list(request):

products = Product.objects.filter(active=True).order\_by("-created\_at")

return render(request, "market/product\_list.html", {"products": products})

@login\_required

def product\_create(request):

if request.method == "POST":

form = ProductForm(request.POST, request.FILES)

if form.is\_valid():

product = form.save(commit=False)

product.seller = request.user

product.save()

return redirect("product-list")

else:

form = ProductForm()

return render(request, "market/product\_form.html", {"form": form})

@login\_required

def add\_to\_cart(request, product\_id):

product = get\_object\_or\_404(Product, id=product\_id)

cart, created = Cart.objects.get\_or\_create(user=request.user)

item, created = CartItem.objects.get\_or\_create(cart=cart, product=product)

item.quantity += 1

item.save()

return redirect("view-cart")

@login\_required

def view\_cart(request):

cart, created = Cart.objects.get\_or\_create(user=request.user)

return render(request, "market/cart.html", {"cart": cart})

**💳 5. Integración con Mercado Pago**

Según guía de **MERCADO PAGO.docx**

**Instalar SDK**

pip install mercadopago

**settings.py**

import os

MERCADOPAGO\_ACCESS\_TOKEN = os.getenv("MERCADOPAGO\_ACCESS\_TOKEN")

**views.py**

import mercadopago

from django.conf import settings

from django.http import JsonResponse

@login\_required

def create\_preference\_cart(request):

    cart, created = Cart.objects.get\_or\_create(user=request.user)

    sdk = mercadopago.SDK(settings.MERCADOPAGO\_ACCESS\_TOKEN)

    items = []

    for item in cart.items.all():

        items.append({

            "title": item.product.title,

            "quantity": item.quantity,

            "unit\_price": float(item.product.price),

            "currency\_id": "ARS",

        })

    preference\_data = {

        "items": items,

        "back\_urls": {

            "success": request.build\_absolute\_uri("/pago-exitoso/"),

            "failure": request.build\_absolute\_uri("/pago-fallido/"),

        },

        "auto\_return": "approved",

    }

    preference = sdk.preference().create(preference\_data)

    return JsonResponse({"init\_point": preference["response"]["init\_point"]})

**🔗 6. URLs (market/urls.py)**

from django.urls import path

from . import views

app\_name = "market"

urlpatterns = [

path("", views.product\_list, name="productlist"),

path("create/", views.product\_create, name="product-create"),

path("edit/<int:pk>/", views.product\_edit, name="product-edit"),

path("delete/<int:pk>/", views.product\_delete, name="product-delete"),

path("cart/", views.view\_cart, name="view-cart"),

path("add/<int:product\_id>/", views.add\_to\_cart, name="add-to-cart"),

# Mercado Pago

path("pago/<int:product\_id>/", views.create\_preference, name="crear-preferencia"),

path("pago-carrito/", views.create\_preference\_cart, name="crear-preferencia-carrito"),

]

**🛠 1. Botón para crear productos**

Agregalo en la parte superior de product\_list.html, solo visible para usuarios autenticados:

{% extends "base.html" %}

{% block title %}Productos • Mercadito{% endblock %}

{% block content %}

<h2>Todos los productos</h2>

{% if user.is\_authenticated %}

<p>

<a href="{% url 'product-create' %}" class="btn btn-success">

➕ Publicar producto

</a>

</p>

{% endif %}

<div class="card-grid">

{% for product in products %}

<div class="card">

{% if product.image %}

<img src="{{ product.image.url }}" alt="{{ product.title }}" style="width:100%; max-height:200px; object-fit:cover;">

{% endif %}

<h3>{{ product.title }}</h3>

<p>{{ product.description|truncatewords:20 }}</p>

<p><strong>${{ product.price }}</strong></p>

{% if user.is\_authenticated %}

<a href="{% url 'add-to-cart' product.id %}" class="btn btn-primary">🛒 Agregar al carrito</a>

{% else %}

<a href="{% url 'account\_login' %}" class="btn btn-secondary">Inicia sesión para comprar</a>

{% endif %}

</div>

{% empty %}

<p>No hay productos disponibles.</p>

{% endfor %}

</div>

{% endblock %}

**🛠 2. Formulario para publicar productos**

En market/forms.py:

from django import forms

from .models import Product

class ProductForm(forms.ModelForm):

class Meta:

model = Product

fields = ["title", "description", "price", "stock", "image", "active"]

**En templates/market/product\_form.html:**

{% extends "base.html" %}

{% block title %}Nuevo producto{% endblock %}

{% block content %}

<h2>Publicar producto</h2>

<form method="post" enctype="multipart/form-data">

{% csrf\_token %}

{{ form.as\_p }}

<button type="submit" class="btn btn-success">Guardar</button>

</form>

{% endblock %}

**🛠 3. Carrito y pago con Mercado Pago**

1. Cuando el usuario cliquea **“Agregar al carrito”**, lo mandás a la vista add\_to\_cart.
2. Desde el carrito, agregás un botón **“Pagar con Mercado Pago”** que dispara la vista create\_preference.

Ejemplo de cart.html:

{% extends "base.html" %}

{% block title %}Tu carrito{% endblock %}

{% block content %}

<h2>Carrito de compras</h2>

{% if cart.items.all %}

<ul>

{% for item in cart.items.all %}

<li>{{ item.product.title }} (x{{ item.quantity }}) — ${{ item.subtotal }}</li>

{% endfor %}

</ul>

<p><strong>Total:</strong> ${{ cart.total }}</p>

<!-- Botón de pago de todo el carrito -->

<button id="checkout-btn" class="btn btn-primary">Pagar carrito completo</button>

<script src="https://sdk.mercadopago.com/js/v2"></script>

<script>

const mp = new MercadoPago("{{ PUBLIC\_KEY }}", { locale: 'es-AR' });

document.getElementById("checkout-btn").addEventListener("click", () => {

fetch("{% url 'crear-preferencia-carrito' %}")

.then(response => response.json())

.then(data => {

window.location.href = data.init\_point;

});

});

</script>

{% else %}

<p>Tu carrito está vacío.</p>

{% endif %}

{% endblock %}

En base colocar antes del body

  {% block content %}{% endblock %}

  {% block extra\_js %}{% endblock %}

Vaciar carrito // terminar pago// modificar productos