

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

FACULTAD DE PRODUCCIÓN Y SERVICIOS
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



Asignatura:

Programación Web 2

INFORME DEL PROYECTO “ PÁGINA WEB E-COMMERCE LAPC.COM”

Docente: Carlos Corrales

Presentado por:

- Paulo Hidalgo Chinchay
- Ivan Huayhua Mayta
- Jose Jaita Chura
- Ronald Garcia Valdivia

AREQUIPA – PERÚ

2023

1. Objetivos

- Desarrollo de una página web E-Commerce para la empresa de electrónicos LaPC enfocada en la venta de laptops, y otros componentes, usando Django.
- Esta interfaz permite a los administradores del sitio administrar varios aspectos del sitio web de comercio electrónico, como agregar y actualizar productos, administrar cuentas de usuario y monitorear pedidos.

2. Requisitos de la Empresa

- Plataforma de comercio electrónico: la empresa debe elegir la plataforma de comercio electrónico adecuada para su tienda en línea. La plataforma debe poder manejar un gran catálogo de productos y brindar una experiencia de compra perfecta para los clientes.
- Catálogo de productos: la empresa necesita crear un catálogo de productos completo que incluya descripciones detalladas, especificaciones e imágenes de cada computadora portátil. El catálogo debe ser fácil de navegar y buscar.
- Carrito de compras: la empresa debe proporcionar a los clientes una función de carrito de compras que les permita agregar computadoras portátiles a su carrito.
- Gestión de pedidos: la empresa debe contar con un sistema para gestionar los pedidos, incluido el procesamiento, el seguimiento y el cumplimiento de pedidos.

3. Equipos, materiales y temas

- Sistema Operativo (GNU/Linux de preferencia).
- GNU Vim.
- Python 3.
- Git.
- Cuenta en GitHub con el correo institucional.
- Entorno virtual.
- Django 4.
- djangorestframework.

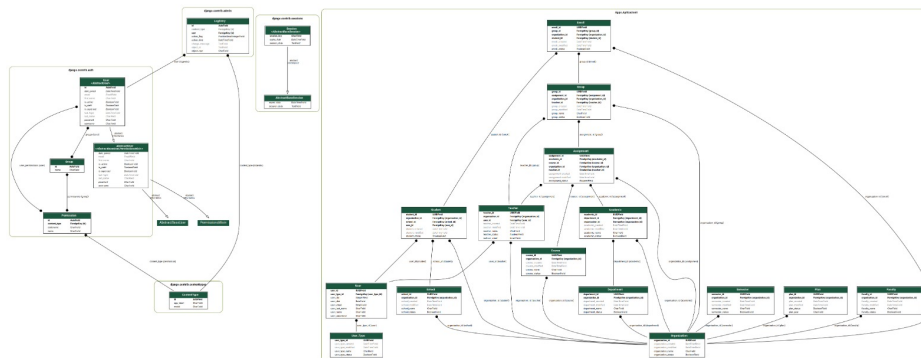
4. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- <https://github.com/123ihuayhua/PWEB2-PROYECTO-FINAL.git>

5. Admin, user y tokens

- Admin:
 - Usuario: admin
 - Contraseña: 12345
 - Token: eef53ad0f9a274f2d2f27426410abb4fe3fb855
- User:
 - Usuario: Alfonso_M
 - Contraseña: alfonso
 - Token: 8aff82557e15d93fe60bfacf6f86fa9ae34fc3c8

6. Modelo ERD



7. Código

- El siguiente código es una implementación de enrutamiento utilizando la biblioteca 'restframework' en Django. Define un conjunto de rutas para diferentes recursos de la API, como vendedores, clientes, pedidos, marcas, tipos de artículos, carritos y artículos. Esta estructura de enrutamiento es esencial para el funcionamiento de una API RESTful, ya que permite acceder a diferentes endpoints para realizar operaciones CRUD (Crear, Leer, Actualizar y Eliminar) en la base de datos a través de solicitudes HTTP. Es un componente clave para construir una API potente y escalable en Django.

Listing 1: urls.py

```

1 from rest_framework import routers
2 from .api import *
3
4 router = routers.DefaultRouter()
5
6 router.register('api/vendedor', VendedorViewSet, 'vendedor')
7 router.register('api/cliente', ClienteViewSet, 'cliente')
8 router.register('api/pedido', PedidoViewSet, 'pedido')
9 router.register('api/marca', MarcaViewSet, 'tienda')
10 router.register('api/tipoArt', TipoArticuloViewSet, 'tienda')
```

```
11 router.register('api/carrito', CarritoViewSet, 'Carrito')
12 router.register('api/articulo', ArticuloViewSet, 'articulo')
13
14 urlpatterns = router.urls
```

- Se han creado clases VendedorAdmin, ClienteAdmin y ArticuloAdmin para definir los campos a mostrar y sus respectivos nombres en el panel. Además, se ha registrado una variedad de modelos en el panel de administración, lo que permite una gestión más eficiente de los datos del sistema. El enfoque en la personalización del panel de administración ha facilitado la visualización y edición de información relevante, optimizando la experiencia de usuario en el sitio web

Listing 2: admin.py

```
1 from django.contrib import admin
2 from .models import *
3 #Mostrar detalles en Vendedor
4 class VendedorAdmin(admin.ModelAdmin):
5     list_display = ('nombre', 'apellido', 'dni')
6
7     def nombre(self, obj):
8         return obj.VenNom
9     nombre.short_description = 'Nombre'
10
11     def apellido(self, obj):
12         return obj.VenApePat
13     apellido.short_description = 'Apellido'
14
15     def dni(self, obj):
16         return obj.VenDNI
17     dni.short_description = 'DNI'
18
19 #Mostrar detalles en Cliente
20 class ClienteAdmin(admin.ModelAdmin):
21     list_display = ('nombre', 'apellido', 'dni')
22
23     def nombre(self, obj):
24         return obj.CliNom
25     nombre.short_description = 'Nombre'
26
27     def apellido(self, obj):
28         return obj.CliApePat
29     apellido.short_description = 'Apellido'
30
31     def dni(self, obj):
32         return obj.CliDNI
33     dni.short_description = 'DNI'
34
35 #Mostrar detalles en Articulo
36 class ArticuloAdmin(admin.ModelAdmin):
37     list_display = ('nombre_articulo', 'marca', 'stock', 'precio_unitario')
38
39     def nombre_articulo(self, obj):
40         return obj.ArtNom
41     nombre_articulo.short_description = 'Nombre'
42
```

```
43 def marca(self, obj):
44     return obj.ArtMarCod
45 marca.short_description = 'Marca'
46
47 def stock(self, obj):
48     return obj.ArtSto
49 stock.short_description = 'Cantidad en Stock'
50
51 def precio_unitario(self, obj):
52     return f'S/ {obj.ArtPreUni:,.2f}'
53 precio_unitario.short_description = 'Precio Unitario'
54
55
56 #Vistas y modelos
57 admin.site.register(Vendedor, VendedorAdmin)
58 admin.site.register(Cliente, ClienteAdmin)
59 admin.site.register(Marca)
60 admin.site.register(TipoArticulo)
61 admin.site.register(Articulo, ArticuloAdmin)
62 admin.site.register(Pedido)
63 admin.site.register(Carrito)
```

- Home(View): Muestra una vista principal que lista todas las marcas y artículos disponibles en la página de inicio.

Signup(View): Permite a los usuarios registrarse en el sitio web mediante un formulario de registro.

Signin(View): Proporciona la funcionalidad para que los usuarios inicien sesión en el sitio web utilizando el formulario de inicio de sesión.

Signout(View): Permite a los usuarios cerrar sesión y redirigirlos a la página de inicio.

Index(View): Muestra una lista de los pedidos realizados por el usuario que ha iniciado sesión.

DetallePedido(View): Muestra los detalles de un pedido específico realizado por el usuario.

CarritoView(View): Muestra el contenido del carrito de compras del usuario que ha iniciado sesión.

AgregarAlCarrito(View): Permite agregar artículos al carrito de compras del usuario.

EliminarDelCarrito(View): Permite eliminar artículos del carrito de compras del usuario.

ActualizarCantidad(View): Actualiza la cantidad de un artículo en el carrito de compras del usuario.

CancelarCarrito(View): Permite cancelar el carrito de compras del usuario.

GuardarPedido(View): Guarda el pedido realizado por el usuario en el carrito y crea una cabecera de pedido con los detalles del carrito.

PedidoDetalle(View): Muestra los detalles de un pedido específico realizado por el usuario.

Productos(View): Muestra una página con la lista de todas las marcas y artículos disponibles.

generar token: Es una vista basada en función que se utiliza como endpoint para generar un token de autenticación para un usuario que utiliza Django REST Framework.

Listing 3: views.py

```
1 from django.shortcuts import render, redirect, get_object_or_404
```

```
2 from django.contrib.auth.forms import AuthenticationForm
3 from django.contrib.auth import login, logout, authenticate
4 from django.db import IntegrityError
5 from .forms import PedidoForm, RegistroForm
6 from .models import Carrito, Pedido, Cliente, Marca, Articulo
7 from django.contrib.auth.decorators import login_required
8 from django.contrib.auth.decorators import login_required
9 from django.http import HttpResponse, JsonResponse
10 from django.contrib import messages
11 from django.contrib.auth import authenticate
12 from rest_framework.decorators import api_view
13 from rest_framework.response import Response
14 from django.views.generic import View
15 from django.utils.decorators import method_decorator
16
17
18 # Vista principal
19
20 class Home(View):
21     def get(self, request, *args, **kwargs):
22         marcas = Marca.objects.all()
23         arts = Articulo.objects.all()
24         context = {'marcas': marcas, 'arts': arts}
25         return render(request, 'home.html', context)
26
27 # Registrarse (usuario)
28 class Signup(View):
29     def get(self, request, *args, **kwargs):
30         form = RegistroForm()
31         return render(request, 'signup.html', {'form': form})
32
33     def post(self, request, *args, **kwargs):
34         print(request.POST)
35         form = RegistroForm(request.POST)
36         if form.is_valid():
37             if request.POST['password1'] == request.POST['password2']:
38                 try:
39                     # Crear usuario
40                     user = Cliente.objects.create_user(
41                         username=request.POST['username'],
42                         password=request.POST['password1'],
43                         CliDNI=request.POST['CliDNI'],
44                         CliApePat=request.POST['CliApePat'],
45                         CliNom=request.POST['CliNom'],
46                         CliEstReg=True
47                     )
48                     login(request, user)
49                     return redirect('home')
50                 except IntegrityError:
51                     return render(request, 'signup.html', {
52                         'form': RegistroForm,
53                         'error': 'El usuario ya existe'
54                     })
55         print(form.errors)
56         return render(request, 'signup.html', {
57             'form': form,
```

```
58     })
59
60 # Logearse
61 class Signin(View):
62     def get(self, request, *args, **kwargs):
63         form = AuthenticationForm()
64         return render(request, 'signin.html', {'form': form})
65
66     def post(self, request, *args, **kwargs):
67         form = AuthenticationForm(data=request.POST)
68         if form.is_valid():
69             user = authenticate(
70                 request, username=request.POST['username'], password=request.POST['password'])
71             if user is not None:
72                 login(request, user)
73                 return render(request, 'home.html')
74             return render(request, 'signin.html', {'form': form, 'error': 'Usuario o contraseña
75                 incorrecta'})
76
77 # Salir de la sesin
78 class Signout(View):
79     def get(self, request):
80         logout(request)
81         return redirect('home')
82
83 # Ver pedidos
84 @method_decorator(login_required, name='dispatch')
85 class Index(View):
86     def get(self, request, *args, **kwargs):
87         pedido = Pedido.objects.filter(PedCabCodCli=request.user)
88         return render(request, 'index.html', {'pedidos': pedido})
89
90 # Detalle pedido
91 @method_decorator(login_required, name='dispatch')
92 class DetallePedido(View):
93     def get(self, request, pedidoID, *args, **kwargs):
94         pedido_cabecera = get_object_or_404(
95             Pedido, pk=pedidoID, PedCabCodCli=request.user)
96         detalles_pedido = Pedido.objects.filter(PedDetCodCab=pedido_cabecera)
97
98         if detalles_pedido.exists():
99             detalle = detalles_pedido.first()
100         else:
101             detalle = Pedido(PedDetCodCab=pedido_cabecera)
102
103         form = PedidoForm(instance=detalle)
104         carrito = request.session.get('carrito', {})
105         articulos = Artículo.objects.filter(pk__in=carrito.keys())
106         detalles = []
107         for carrito in pedido_cabecera.PedDetArtCod.all():
108             # cantidad = carrito[str(articulo.pk)][ 'cantidad' ]
109             # subtotal = cantidad * articulo.ArtPreUni
110             detalles.append({
111                 'articulo': carrito.articulo,
112                 'cantidad': carrito.cantidad,
113                 'subtotal': carrito.subtotal,
```

```

113         'precio_uni': carrito.articulo.ArtPreUni,
114     })
115     print("Se ha registrado productos")
116     return render(request, 'pedido_detalle.html', {
117         'pedido_cabecera': pedido_cabecera,
118         'detalle': detalle,
119         'form': form,
120         'detalles': detalles
121     })
122
123     def post(self, request, pedidoID, *args, **kwargs):
124         pedido_cabecera = get_object_or_404(
125             Pedido, pk=pedidoID, PedCabCodCli=request.user)
126         detalles_pedido = Pedido.objects.filter(PedDetCodCab=pedido_cabecera)
127
128         if detalles_pedido.exists():
129             detalle = detalles_pedido.first()
130         else:
131             detalle = Pedido(PedDetCodCab=pedido_cabecera)
132
133         form = PedidoForm(request.POST, instance=detalle)
134         if form.is_valid():
135             form.save()
136             return JsonResponse({'success': True})
137         else:
138             return JsonResponse({'success': False})
139
140     # Eliminar pedido
141     @method_decorator(login_required, name='dispatch')
142     class EliminarPedido(View):
143         def get(self, request, pedidoID, *args, **kwargs):
144             pedido_cabecera = get_object_or_404(
145                 Pedido, pk=pedidoID, PedCabCodCli=request.user)
146             detalles_pedido = Pedido.objects.filter(pk=pedidoID)
147
148             for carrito in pedido_cabecera.PedDetArtCod.all():
149                 carrito.setstatus(True)
150                 # articulo = detalle.PedDetArtCod
151                 # articulo.ArtSto += detalle.PedDetCantidad
152                 # articulo.save()
153             pedido_cabecera.delete()
154
155             return redirect('index')
156
157     # Marcas
158     class MarcaDetail(View):
159         def get(self, request, marca_id, *args, **kwargs):
160             marca = get_object_or_404(Marca, pk=marca_id)
161             arts = Articulo.objects.filter(ArtMarCod=marca)
162             '''
163             templates = {
164                 'Acer': 'marca_acer.html',
165                 'Apple': 'marca_apple.html',
166                 'Asus': 'marca_asus.html',
167                 'Dell': 'marca_dell.html',
168                 'HP': 'marca_hp.html',

```



```
169         'Lenovo': 'marca_lenovo.html',
170         'MSI': 'marca_msi.html',
171     }
172     templates = templates.get(marca.MarNom, 'marca_default.html')
173     '''
174     context = {'marca': marca, 'arts': arts}
175     return render(request, 'marca_msi.html', context)
176
177 # Carrito de compras
178 @method_decorator(login_required, name='dispatch')
179 class CarritoView(View):
180     def get(self, request, *args, **kwargs):
181         carrito = Carrito.objects.filter(usuario=request.user, status = True)
182         total = sum(item.subtotal for item in carrito)
183         return render(request, 'carrito.html', {'carrito': carrito, 'total': total})
184
185 # Agregar al carrito
186 @method_decorator(login_required, name='dispatch')
187 class AgregarAlCarrito(View):
188     def post(self, request, articulo_id, *args, **kwargs):
189         articulo = get_object_or_404(Articulo, pk=articulo_id)
190         carrito, created = Carrito.objects.get_or_create(
191             usuario=request.user, articulo=articulo)
192         if not created:
193             carrito.cantidad += 1
194             carrito.save()
195         return render(request, 'carrito.html')
196
197 # Eliminar Producto
198 @method_decorator(login_required, name='dispatch')
199 class EliminarDelCarrito(View):
200     def post(self, request, carrito_id, *args, **kwargs):
201         carrito = get_object_or_404(
202             Carrito, pk=carrito_id, usuario=request.user)
203         carrito.delete()
204         return redirect('carrito')
205
206 # Carrito cantidad actualizar
207 class ActualizarCantidad(View):
208     def post(self, request, item_pk, cantidad, *args, **kwargs):
209         try:
210             carrito_item = get_object_or_404(Carrito, pk=item_pk)
211             carrito_item.cantidad = cantidad
212             carrito_item.save()
213             return JsonResponse({'status': 'ok'})
214         except Exception as e:
215             return JsonResponse({'status': 'not ok', 'message': str(e)})
216
217 # Cancelar carrito
218 class CancelarCarrito(View):
219     def post(self, request, *args, **kwargs):
220         carrito = Carrito.objects.filter(usuario=request.user, status = True)
221         # Se cancela si hay 1 producto pero mas de 1 no
222         # request.session['carrito'] = []
223         # request.session.modified = True
224         carrito.delete()
```

```

225         return redirect('carrito')
226
227 #Guardar pedido
228 @method_decorator(login_required, name='dispatch')
229 class GuardarPedido(View):
230     def get(self, request, *args, **kwargs):
231         # messages.warning(request, 'No se permiten solicitudes GET.')
232         return redirect('index')
233
234     def post(self, request, *args, **kwargs):
235         carrito = Carrito.objects.filter(usuario=request.user)
236         if carrito.exists():
237             pedido_cabecera = Pedido(PedCabCodCli=request.user)
238             pedido_cabecera.save()
239             for item in carrito:
240                 articulo = item.articulo
241                 cantidad = item.cantidad
242                 # if cantidad > articulo.ArtSto:
243                 #     messages.error(
244                 #         request, f'La cantidad de "{articulo.ArtNom}" en el carrito supera el
245                 #             stock disponible.')
246                 #     return redirect('carrito')
247                 # subtotal = cantidad * articulo.ArtPreUni
248                 pedido_cabecera.PedDetArtCod.add(item)
249                 # pedido_cabecera.PedDetCantidad = cantidad
250                 # pedido_cabecera.PedDetPreUniArt = articulo.ArtPreUni
251                 # pedido_cabecera.PedDetSubtotal = subtotal
252                 # pedido_cabecera.PedDetTot += subtotal
253                 # articulo.ArtSto -= cantidad
254                 # articulo.save()
255                 item.setstatus(False)
256                 item.save()
257             pedido_cabecera.save()
258             # carrito.delete()
259             return redirect('pedido_detalle', pedidoID=pedido_cabecera.pk)
260         else:
261             messages.warning(request, 'No hay articulos en el carrito.')
262             return redirect('index')
263
264 # Pedido detalle
265 @method_decorator(login_required, name='dispatch')
266 class PedidoDetalle(View):
267     def get(self, request, pedidoID, *args, **kwargs):
268         pedido = get_object_or_404(
269             Pedido, pk=pedidoID, PedCabCodCli=request.user)
270         detalles = []
271         for carrito in pedido.PedDetArtCod.all():
272             detalles.append({
273                 'articulo': carrito.articulo,
274                 'cantidad': carrito.cantidad,
275                 'precio_uni': carrito.articulo.ArtPreUni,
276                 'subtotal': carrito.subtotal,
277             })
278         return render(request, 'pedido_detalle.html', {'pedido': pedido, 'detalles':

```

```

279 # Productos
280 class Productos(View):
281     def get(self, request, *args, **kwargs):
282         marcas = Marca.objects.all()
283         arts = Articulo.objects.all()
284         context = {'marcas': marcas, 'arts': arts}
285         return render(request, 'productos.html', context)
286
287 # Token
288 @api_view(['POST'])
289 def generar_token(request):
290     username = request.data.get('username')
291     password = request.data.get('password')
292     user = authenticate(username=username, password=password)
293     if user is not None:
294         token = user.generate_auth_token()
295         return Response({'token': token})
296     else:
297         return Response({'error': 'Credenciales invalidas'})

```

- Formularios personalizados son parte fundamental de la lógica de registro de usuarios y creación de pedidos en la aplicación web. Proporcionan campos personalizados y validaciones para garantizar la integridad y seguridad de los datos ingresados por los usuarios. Además, el método save personalizado agrega funcionalidad adicional para manejar adecuadamente la creación y almacenamiento de instancias del modelo en la base de datos.

Listing 4: forms.py

```

1 from datetime import timezone
2 import json
3 from django import forms
4 from django.db.models.signals import post_save
5 from django.dispatch import receiver
6 from django.forms import ModelForm, ValidationError, formset_factory
7 from django.contrib.auth.forms import UserCreationForm
8 from .models import Pedido, Vendedor
9 from .models import Cliente
10 from django.core.validators import MinLengthValidator
11 from django.utils.translation import gettext as _
12 from django.core.validators import ValidationError, RegexValidator
13 from django.utils.translation import gettext as _
14
15
16 class RegistroForm(UserCreationForm):
17     CliDNI = forms.CharField(max_length=8, required=True)
18     CliApePat = forms.CharField(max_length=50, required=True)
19     CliNom = forms.CharField(max_length=50, required=True)
20     password1 = forms.CharField(
21         label='Contrasea',
22         widget=forms.PasswordInput(attrs={'class': 'form-control', 'placeholder':
23             'Contrasea'}),
24         strip=False,
25         help_text=_('La contraseña debe tener al menos 8 caracteres y no puede ser demasiado
26             comn.'),
27         validators=[

```

```

26         # Validacin de longitud mnima de la contrasea
27         MinLengthValidator(8, message=_('La contrasea debe tener al menos 8 caracteres.')),
28         # Validacin de complejidad de la contrasea
29         RegexValidator(
30             regex=r'^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]+$',
31             message=_('La contrasea debe contener al menos una letra mayuscula, una letra
                minscula, un nmero y un carcter especial.'))
32     ),
33     # Validacin de similitud de la contrasea con el nombre de usuario
34     RegexValidator(
35         regex=r'^[A-Za-z0-9]*([A-Za-z0-9]{1,3}|[A-Za-z0-9]*[A-Za-z0-9][A-Za-z0-9]*)[A-Za-z0-9]*$',
36         message=_('La contrasea no puede ser demasiado similar al nombre de usuario.'))
37     )
38 ]
39 )
40 password2 = forms.CharField(
41     label='Confirmar contrasea',
42     widget=forms.PasswordInput(attrs={'class': 'form-control', 'placeholder': 'Confirmar
        contrasea'}),
43     strip=False,
44     help_text=_('Ingrese la misma contrasea que antes, para verificacin.'),
45 )
46
47 class Meta(UserCreationForm.Meta):
48     model = Cliente
49     fields = ('username', 'password1', 'password2',
50             'CliDNI', 'CliApePat', 'CliNom')
51     labels = {
52         'CliDNI': 'DNI',
53         'CliApePat': 'Apellido Paterno',
54         'CliNom': 'Nombre'
55     }
56     def clean_password2(self):
57         password1 = self.cleaned_data.get('password1')
58         password2 = self.cleaned_data.get('password2')
59         if password1 and password2 and password1 != password2:
60             raise ValidationError(_('Las contraseas no coinciden.'))
61         return password2
62
63
64 class PedidoForm(forms.ModelForm):
65     class Meta:
66         model = Pedido
67         fields = '__all__'
68
69     def __init__(self, *args, **kwargs):
70         super().__init__(*args, **kwargs)
71         initial = kwargs.get('initial')
72         if initial:
73             self.fields['PedDetArtCod'].initial = initial['PedDetArtCod'][0]
74             self.fields['PedDetCantidad'].initial = initial['PedDetCantidad'][0]
75             self.fields['PedDetPreUniArt'].initial = initial['PedDetPreUniArt'][0]
76             self.fields['PedDetSubtotal'].initial = initial['PedDetSubtotal'][0]
77             self.fields['PedDetTot'].initial = initial['PedDetTot'][0]
78
79     def save(self, commit=True):

```

```
80     pedido_cabecera = super().save(commit=False)
81     pedido_cabecera.PedCabFec = timezone.now() # Establecer la fecha actual
82     if commit:
83         pedido_cabecera.save()
84     return pedido_cabecera
85
86     # def save(self, *args, **kwargs):
87     #     if self.pk is None:
88     #         # object is being created
89     #         self.actualizar_stock(self.PedDetArtCod, self.PedDetCantidad)
90     #     else:
91     #         # object is being updated
92     #         old_detalle = Pedido.objects.get(pk=self.pk)
93     #         if old_detalle.PedDetCantidad != self.PedDetCantidad:
94     #             difference = self.PedDetCantidad - old_detalle.PedDetCantidad
95     #             self.actualizar_stock(self.PedDetArtCod, difference)
96     #         super().save(*args, **kwargs)
97
98
99 # @receiver(post_save, sender=Pedido)
100 # def update_stock_quantity(sender, instance, **kwargs):
101 #     articulo = instance.PedDetArtCod
102 #     # articulo.ArtSto = articulo.ArtSto - instance.PedDetCantidad
103 #     articulo.save()
104
105
106 # @receiver(post_save, sender=Pedido)
107 # def actualizar_stock(sender, instance, **kwargs):
108 #     articulo = instance.PedDetArtCod
109 #     cantidad = instance.PedDetCantidad
110 #     if cantidad > articulo.ArtSto:
111 #         # error_message = 'La cantidad solicitada supera el stock disponible'
112 #         # return render(request, 'actualizar_pedido.html', {'form': formatter,
113 #         # 'error_message': error_message})
114 #         raise forms.ValidationError(
115 #             f'No hay suficiente stock de {articulo.ArtNom}')
116 #     articulo.ArtSto = articulo.ArtSto - (cantidad-1)
117 #     articulo.save()
```

- Implementación de varios serializadores utilizando Django REST Framework. Los serializadores se utilizan para facilitar la conversión de instancias de modelos de Django en formato JSON y viceversa. Cada serializador está diseñado para trabajar con un modelo específico y define los campos que se incluirán en la serialización. Además de los serializadores, el código también muestra un método personalizado create_pedido dentro del PedidoSerializer, que se utiliza para crear instancias de pedidos a partir de los datos proporcionados en la solicitud. Este método realiza cálculos y acciones adicionales antes de crear el objeto Pedido, lo que muestra cómo se pueden extender las funcionalidades predeterminadas de los serializadores para adaptarlos a casos específicos.

Listing 5: serializers.py

```
1 from django.shortcuts import get_object_or_404
2 from rest_framework import serializers
3 from .models import *
4
```

```
5 class VendedorSerializer(serializers.ModelSerializer):
6     class Meta:
7         model = Vendedor
8         fields = ('id', 'VenDNI', 'VenApePat', 'VenNom', 'VenEstReg')
9         read_only_fields = ('id',)
10
11 class ClienteSerializer(serializers.ModelSerializer):
12     class Meta:
13         model = Cliente
14         fields = ('id', 'CliDNI', 'CliApePat', 'CliNom', 'CliEstReg', 'password', 'username')
15         read_only_fields = ('id',) #Campo no se actualiza
16
17 class MarcaSerializer(serializers.ModelSerializer):
18     class Meta:
19         model = Marca
20         fields = ('id', 'MarNom', 'MarImg', 'MarEstReg')
21
22 class TipoArticuloSerializer(serializers.ModelSerializer):
23     class Meta:
24         model = TipoArticulo
25         fields = ('id', 'TipArtNom', 'TipArtEstReg')
26         read_only_fields = ('id',)
27
28 class ArticuloSerializer(serializers.ModelSerializer):
29     class Meta:
30         model = Articulo
31         fields = ('id', 'ArtMarCod', 'ArtTipCod', 'ArtNom', 'ArtDes', 'ArtImg', 'ArtSto',
32                 'ArtPreUni', 'ArtEstReg')
33         read_only_fields = ('id',)
34
35 class PedidoSerializer(serializers.ModelSerializer):
36     class Meta:
37         model = Pedido
38         fields = ('id', 'PedCabCodCli', 'PedCabFec', 'PedDetArtCod', 'PedDetCantidad',
39                 'PedDetPreUniArt', 'PedDetSubtotal', 'PedDetTot', 'PedDetEstReg')
40         read_only_fields = ('id', 'PedCabFec',)
41
42     def create_pedido(self, data):
43         # Validar los datos enviados en la solicitud
44         serializer = PedidoSerializer(data=data)
45         serializer.is_valid(raise_exception=True)
46
47         # Obtener los datos validados
48         validated_data = serializer.validated_data
49
50         # Obtener el precio del articulo seleccionado
51         precio_articulo = validated_data['PedDetArtCod'].ArtPreUni
52
53         # Calcular el subtotal y el total
54         subtotal = validated_data['PedDetCantidad'] * precio_articulo
55         total = subtotal
56
57         # Agregar el subtotal y el total a los datos validados
58         validated_data['PedDetSubtotal'] = subtotal
59         validated_data['PedDetTot'] = total
```

```
59     # Disminuir el stock del articulo correspondiente
60     articulo = validated_data['PedDetArtCod']
61     cantidad = validated_data['PedDetCantidad']
62     articulo.disminuir_stock(cantidad)
63
64     # Crear el pedido
65     pedido = Pedido.objects.create(**validated_data)
66
67     return pedido
68
69 class CarritoSerializer(serializers.ModelSerializer):
70     class Meta:
71         model = Carrito
72         fields = ('usuario', 'articulo', 'cantidad', 'pedido')
```

■

Listing 6: urls.py de localstore

```
1  """
2  URL configuration for localstore project.
3
4  The 'urlpatterns' list routes URLs to views. For more information please see:
5      https://docs.djangoproject.com/en/4.2/topics/http/urls/
6  Examples:
7  Function views
8      1. Add an import: from my_app import views
9      2. Add a URL to urlpatterns: path('', views.home, name='home')
10 Class-based views
11     1. Add an import: from other_app.views import Home
12     2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
13 Including another URLconf
14     1. Import the include() function: from django.urls import include, path
15     2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
16 """
17 from django.contrib import admin
18 from django.urls import path
19 from tienda import views
20 from django.conf import settings
21 from django.conf.urls.static import static
22 from django.urls import path, include
23
24
25 urlpatterns = [
26     path('admin/', admin.site.urls),
27     path('', views.Home.as_view(), name='home'),
28     path('home/', views.Home.as_view(), name='home'),
29     path('signup/', views.Signup.as_view(), name='signup'),
30     path('index/', views.Index.as_view(), name='index'),
31     path('index/<int:pedidoID>', views.DetallePedido.as_view(), name='pedido_detalle'),
32     path('index/eliminar/<int:pedidoID>', views.EliminarPedido.as_view(),
33         name='eliminar_pedido'),
34     path('logout/', views.Signout.as_view(), name='logout'),
35     path('signin/', views.Signin.as_view(), name='signin'),
36     path('carrito/', views.CarritoView.as_view(), name='carrito'),
```

```
36 path('carrito/agregar_al_carrito/<int:articulo_id>/', views.AgregarAlCarrito.as_view(),
    name='agregar_al_carrito'),
37 path('carrito/actualizar_cantidad/<int:item_pk>/<int:cantidad>/',
    views.ActualizarCantidad.as_view(), name='actualizar_cantidad'),
38 path('carrito/eliminar_del_carrito/<int:carrito_id>/',
    views.EliminarDelCarrito.as_view(), name='eliminar_del_carrito'),
39 path('carrito/cancelar_carrito/', views.CancelarCarrito.as_view(),
    name='cancelar_carrito'),
40 path('carrito/guardar_pedido/', views.GuardarPedido.as_view(), name='guardar_pedido'),
41 path('pedido_detalle/<int:pedidoID>/', views.PedidoDetalle.as_view(),
    name='pedido_detalle'),
42 path('marca/<int:marca_id>/', views.MarcaDetail.as_view(), name='marca_detail'),
43 path('productos/', views.Productos.as_view(), name='productos'),
44 path('', include('tienda.urls')),
45 path('api-token-auth/', views.generar_token),
46 path('accounts/', include('allauth.urls')),
47 ]
48
49 urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
```

- Este archivo esencial es importante para configurar y personalizar el comportamiento global del proyecto, definiendo diversas opciones para la aplicación, la base de datos, la seguridad, las plantillas y otros aspectos fundamentales.

Listing 7: settings.py

```
1 """
2 Django settings for localstore project.
3
4 Generated by 'django-admin startproject' using Django 4.2.2.
5
6 For more information on this file, see
7 https://docs.djangoproject.com/en/4.2/topics/settings/
8
9 For the full list of settings and their values, see
10 https://docs.djangoproject.com/en/4.2/ref/settings/
11 """
12
13 import os
14 from pathlib import Path
15
16 # Build paths inside the project like this: BASE_DIR / 'subdir'.
17 BASE_DIR = Path(__file__).resolve().parent.parent
18
19
20 # Quick-start development settings - unsuitable for production
21 # See https://docs.djangoproject.com/en/4.2/howto/deployment/checklist/
22
23 # SECURITY WARNING: keep the secret key used in production secret!
24 SECRET_KEY = 'django-insecure-tu2ymog3e$qw$xdw**88g4$2pb+xqb&(6bp6~nz70#((qew62y'
25
26 # SECURITY WARNING: don't run with debug turned on in production!
27 DEBUG = True
28
29 ALLOWED_HOSTS = []
```



```
30 SITE_ID = 4
31
32 # Application definition
33
34 INSTALLED_APPS = [
35     'django.contrib.admin',
36     'django.contrib.auth',
37     'django.contrib.contenttypes',
38     'django.contrib.sessions',
39     'django.contrib.messages',
40     'django.contrib.staticfiles',
41     #Agregando aplicacin
42     'tienda',
43     'fontawesomefree',
44     'rest_framework',
45     'rest_framework.authtoken',
46     'django_extensions',
47     'corsheaders',
48     'django_filters',
49     #google
50     'django.contrib.sites',
51     'allauth',
52     'allauth.account',
53     'allauth.socialaccount',
54     'allauth.socialaccount.providers.google',
55 ]
56 SOCIALACCOUNT_PROVIDERS = {
57     "google": {
58         "SCOPE": [
59             "profile",
60             "email"
61         ],
62         "AUTH_PARAMS": {"access_tyoe": "online"}
63     }
64 }
65 REST_FRAMEWORK = {
66     'DEFAULT_AUTHENTICATION_CLASSES': [
67         'rest_framework.authentication.TokenAuthentication',
68     ],
69 }
70 }
71
72 MIDDLEWARE = [
73     'corsheaders.middleware.CorsMiddleware',
74     'django.middleware.security.SecurityMiddleware',
75     'django.contrib.sessions.middleware.SessionMiddleware',
76     'django.middleware.common.CommonMiddleware',
77     'django.middleware.csrf.CsrfViewMiddleware',
78     'django.contrib.auth.middleware.AuthenticationMiddleware',
79     'django.contrib.messages.middleware.MessageMiddleware',
80     'django.middleware.clickjacking.XFrameOptionsMiddleware',
81 ]
82
83 ROOT_URLCONF = 'localstore.urls'
84
85 TEMPLATES = [
```

```
86     {
87         'BACKEND': 'django.template.backends.django.DjangoTemplates',
88         'DIRS': [],
89         'APP_DIRS': True,
90         'OPTIONS': {
91             'context_processors': [
92                 'django.template.context_processors.debug',
93                 'django.template.context_processors.request',
94                 'django.contrib.auth.context_processors.auth',
95                 'django.contrib.messages.context_processors.messages',
96             ],
97         },
98     },
99 ]
100
101 WSGI_APPLICATION = 'localstore.wsgi.application'
102
103
104 # Database
105 # https://docs.djangoproject.com/en/4.2/ref/settings/#databases
106
107 DATABASES = {
108     'default': {
109         'ENGINE': 'django.db.backends.sqlite3',
110         'NAME': BASE_DIR / 'db.sqlite3',
111     }
112 }
113
114
115 # Password validation
116 # https://docs.djangoproject.com/en/4.2/ref/settings/#auth-password-validators
117
118 AUTH_PASSWORD_VALIDATORS = [
119     {
120         'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
121     },
122     {
123         'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
124     },
125     {
126         'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
127     },
128     {
129         'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
130     },
131 ]
132
133 GRAPH_MODELS = {
134     'all_applications': True,
135     'graph_models': True,
136 }
137
138
139 # Internationalization
140 # https://docs.djangoproject.com/en/4.2/topics/i18n/
141
```

```

142 LANGUAGE_CODE = 'es'
143
144 TIME_ZONE = 'America/Lima'
145
146 USE_I18N = True
147
148 USE_TZ = True
149
150 CORS_ORIGIN_ALLOW_ALL = True
151
152 APPEND_SLASH = False
153
154 # Static files (CSS, JavaScript, Images)
155 # https://docs.djangoproject.com/en/4.2/howto/static-files/
156
157 STATIC_URL = '/static/'
158 STATIC_ROOT = os.path.join(BASE_DIR, 'static')
159 MEDIA_ROOT = BASE_DIR / 'media'
160 MEDIA_URL = '/public/'
161
162 # Default primary key field type
163 # https://docs.djangoproject.com/en/4.2/ref/settings/#default-auto-field
164
165 DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
166 AUTH_USER_MODEL = 'tienda.Cliente'
167 LOGIN_URL = '/signin'
168 AUTHENTICATION_BACKENDS = (
169     'django.contrib.auth.backends.ModelBackend',
170     'allauth.account.auth_backends.AuthenticationBackend'
171 )
172
173 LOGIN_REDIRECT_URL = '/'
174 LOGOUT_REDIRECT_URL = '/'

```

- Este archivo representa un formulario de registro de usuario con campos para proporcionar información personal y credenciales de inicio de sesión. El formulario está diseñado para ser seguro y compatible con la protección CSRF.

Listing 8: signup.html

```

1 {% extends 'base.html' %}
2 {% load static %}
3 {% block content %}
4 <section class="vh-130" style="background-color: #eee;">
5     <div class="container h-80">
6         <div class="row d-flex justify-content-center align-items-center h-100">
7             <div class="col-lg-12 col-xl-11">
8                 <div class="card text-black" style="border-radius: 25px;">
9                     <div class="card-body p-md-5">
10                         <div class="row justify-content-center">
11                             <div class="col-md-10 col-lg-6 col-xl-5 order-2 order-lg-1">
12
13                                 <p class="text-center h1 fw-bold mb-5 mx-1 mx-md-4 mt-4">Sign up</p>
14                                 {% if form.errors %}
15                                 <div class="alert alert-danger">

```

```

16         <ul>
17             {% for field in form %}
18             {% for error in field.errors %}
19                 <li>{{ error }}</li>
20             {% endfor %}
21             {% endfor %}
22         </ul>
23     </div>
24 {% endif %}
25 <form class="mx-1 mx-md-4" action="/signup/" method="POST">
26     {% csrf_token %}
27     <!-- Nombre -->
28     <div class="d-flex flex-row align-items-center mb-4">
29         <i class="fas fa-user fa-lg me-3 fa-fw"></i>
30         <div class="form-outline flex-fill mb-0">
31             <label class="form-label" id="id_CliNom">Name</label>
32             <input type="text" name="CliNom" id="id_CliNom"
33                 class="form-control" placeholder="Write your Name"/>
34         </div>
35     </div>
36     <!-- Apellido -->
37     <div class="d-flex flex-row align-items-center mb-4">
38         <i class="fas fa-user fa-lg me-3 fa-fw"></i>
39         <div class="form-outline flex-fill mb-0">
40             <label class="form-label" id="id_CliApePat">Surname</label>
41             <input type="text" name="CliApePat" id="id_CliApePat"
42                 class="form-control" placeholder="Write your Surname"/>
43         </div>
44     </div>
45     <!-- dni -->
46     <div class="d-flex flex-row align-items-center mb-4">
47         <i class="fas fa-user fa-lg me-3 fa-fw"></i>
48         <div class="form-outline flex-fill mb-0">
49             <label class="form-label" id="id_CliDNI">DNI</label>
50             <input type="number" maxLength="8" name="CliDNI" id="id_CliDNI"
51                 class="form-control" placeholder="Write your Surname"/>
52         </div>
53     </div>
54     <!-- Usuario -->
55     <div class="d-flex flex-row align-items-center mb-4">
56         <i class="fas fa-user fa-lg me-3 fa-fw"></i>
57         <div class="form-outline flex-fill mb-0">
58             <label class="form-label" id="id_username">Username</label>
59             <input type="text" name="username" id="id_username"
60                 class="form-control" placeholder="Write your Surname"/>
61         </div>
62     </div>
63     <!-- Contrasea -->
64     <div class="d-flex flex-row align-items-center mb-4">
65         <i class="fas fa-lock fa-lg me-3 fa-fw"></i>
66         <div class="form-outline flex-fill mb-0">
67             <label class="form-label" id="id_password1">Password</label>
68             <input type="password" name="password1" id="id_password1"
69                 class="form-control" placeholder="Write your password"/>
70         </div>
71     </div>
72 </form>

```

```

67         <!-- Contrasea2 -->
68         <div class="d-flex flex-row align-items-center mb-4">
69             <i class="fas fa-lock fa-lg me-3 fa-fw"></i>
70             <div class="form-outline flex-fill mb-0">
71                 <label class="form-label" for="id_password2">Password</label>
72                 <input type="password" name="password2" id="id_password2"
73                     class="form-control" placeholder="Repite your password"/>
74             </div>
75         </div>
76
77         <!-- Boton Register -->
78         <div class="d-flex justify-content-center mx-4 mb-3 mb-lg-4">
79             <button type="submit" class="btn btn-primary btn-lg">Register</button>
80         </div>
81     </div>
82     <div class="col-md-10 col-lg-6 col-xl-7 d-flex align-items-center order-1
83         order-lg-2">
84         
86     </div>
87 </div>
88 </div>
89 </div>
90 </div>
91 </div>
92 </section>
93 {% endblock %}

```

- Página de inicio de sesión donde los usuarios pueden ingresar sus credenciales para acceder al sitio web. Además, ofrece la opción de iniciar sesión con una cuenta de Google a través de la autenticación social.

Listing 9: signin.html

```

1  {% extends 'base.html' %}
2  {% load static %}
3  {% block content %}
4      <section class="vh-100">
5          <div class="container py-5 h-100">
6              <div class="row d-flex align-items-center justify-content-center h-100">
7                  <div class="col-md-8 col-lg-7 col-xl-6">
8                      
10                 </div>
11                 <div class="col-md-7 col-lg-5 col-xl-5 offset-xl-1">
12                     <form action="/signin/" method="POST">
13                         {% csrf_token %}
14                         <h1 class="text-center fw-bold mb-2 text-uppercase">Signin</h1>
15                         {{error}} {% csrf_token %}
16                         <!-- Username -->
17                         <div class="form-outline mb-4">
18                             <label for="username">Username</label>
19                             <input type="text" name="username" id="username"

```

```

class="form-control form-control-lg" placeholder="Write your
username"/>
20 </div>
21 <!-- Contrasea -->
22 <div class="form-outline mb-4">
23   <label class="password" for="password">Password</label>
24   <input type="password" name="password" id="password"
     class="form-control form-control-lg" placeholder="Write your
     password"/>
25 </div>
26
27 <div class="d-flex justify-content-around align-items-center mb-4">
28   <!-- Checkbox -->
29   <div class="form-check">
30     <input class="form-check-input" type="checkbox" value=""
       id="form1Example3" checked />
31     <label class="form-check-label" for="form1Example3"> Remember me
       </label>
32   </div>
33   <a href="/signup">Don't have an account?</a>
34 </div>
35 <!-- Boton -->
36 <button type="submit" class="btn btn-primary btn-lg btn-block">Sign
    in</button>
37 {% load socialaccount %}
38 <a href="{% provider_login_url 'google' %}?next=/" class="btn
    btn-primary btn-lg btn-block">Login with google</a>
39
40 </form>
41 </div>
42 </div>
43 </section>
44 {% endblock %}

```

- Este archivo muestra detalles de un pedido realizado por un cliente, incluyendo información relevante como el cliente, la fecha, el total y la lista de artículos con sus cantidades y precios unitarios. La información es presentada en una tabla para facilitar la lectura y comprensión.

Listing 10: pedido_detalle.html

```

1 {% extends 'base.html' %}
2
3 {% block content %}
4 <div class="container">
5   <div class="container">
6     <h2 class="text-center fw-bolder">DATOS DEL PEDIDO</h2>
7     <table class="table">
8       <tbody>
9         <tr>
10          <th scope="row">CLIENTE</th>
11          <td>{{ pedido.PedCabCodCli }}</td>
12        </tr>
13        <tr>
14          <th scope="row">FECHA</th>
15          <td>{{ pedido.PedCabFec }}</td>

```

```

16     </tr>
17     <tr>
18         <th scope="row">TOTAL</th>
19         <td>S/. {{ pedido.total }}</td>
20     </tr>
21 </tbody>
22 </table>
23 </div>
24
25 <div class="container">
26     <table class="table">
27         <thead class="table-dark">
28             <tr>
29                 <th>Articulo</th>
30                 <th>Cantidad</th>
31                 <th>Precio unitario</th>
32                 <th>Subtotal</th>
33             </tr>
34         </thead>
35         <tbody class="table-group-divider">
36             {% for carrito in detalles %}
37             <tr>
38                 <td>{{ carrito.articulo }}</td>
39                 <td>{{ carrito.cantidad }}</td>
40                 <td>S/. {{ carrito.precio_uni }}</td>
41                 <td>S/. {{ carrito.subtotal }}</td>
42             </tr>
43             {% endfor %}
44         </tbody>
45     </table>
46 </div>
47 </div>
48 {% endblock %}

```

- La base proporciona una barra de navegación y una estructura básica para el diseño de la interfaz de una tienda en línea. El contenido específico de cada página se colocará dentro del bloque `content`, y se incluirá un pie de página común en todas las páginas utilizando la plantilla `'footer.html'`. Esto permite mantener un diseño coherente en toda la tienda y reutilizar componentes comunes en diferentes páginas.

Listing 11: base.html

```

1 <!DOCTYPE html>
2 <html lang="en">
3
4 <head>
5     <meta charset="UTF-8">
6     <meta name="viewport" content="width=device-width, initial-scale=1.0">
7     <link rel="stylesheet"
8         href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css">
9     <link rel="stylesheet"
10        href="https://cdn.jsdelivr.net/npm/bootstrap-icons@1.7.2/font/bootstrap-icons.css">
11     <title>Tienda</title>
12 </head>

```

```

12 <body>
13   {% load static %}
14   <nav class="navbar navbar-expand-lg bg-dark navbar-dark">
15     <div class="container">
16       <!-- Agrega el logotipo -->
17       <a class="navbar-brand" href="/">
18         
20       </a>
21       <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
22         data-bs-target="#navbarNav"
23         aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle navigation">
24         <span class="navbar-toggler-icon"></span>
25       </button>
26       <div class="collapse navbar-collapse" id="navbarNav">
27         <ul class="navbar-nav ms-auto">
28           <li class="nav-item">
29             <a class="nav-link" href="/">
30               Home
31               <i class="bi bi-house-door-fill" style="font-size: 2rem;"></i>
32             </a>
33           </li>
34           <li class="nav-item">
35             <a href="/productos/" class="nav-link">
36               Productos
37               <i class="bi bi-laptop" style="font-size: 2rem; color: green;"></i>
38             </a>
39           </li>
40           {% if user.is_authenticated %}
41           <li class="nav-item">
42             <a href="/index/" class="nav-link">
43               Pedidos
44               <i class="bi bi-list-ul" style="font-size: 2rem; color: white;"></i>
45             </a>
46           </li>
47           <li class="nav-item">
48             <a href="/carrito/" class="nav-link">
49               Carrito
50               <i class="bi bi-cart-fill" style="font-size: 2rem; color: orange;"></i>
51             </a>
52           </li>
53           <li class="nav-item">
54             <a href="/logout/" class="nav-link">
55               Logout
56               <i class="bi bi-power" style="font-size: 2rem; color: red;"></i>
57             </a>
58           </li>
59           {% else %}
60           <li class="nav-item">
61             <a href="/signup/" class="nav-link">
62               Signup
63               <i class="bi bi-person-lines-fill" style="font-size: 2rem; color: skyblue;"></i>
64             </a>
65           </li>
66           <li class="nav-item">
67             <a href="/signin/" class="nav-link">

```



```

66         Signin
67         <i class="bi bi-box-arrow-right" style="font-size: 2rem; color: white;"></i>
68         </a>
69     </li>
70     {% endif %}
71 </li>
72 </ul>
73 </div>
74 </div>
75 </nav>
76 {% block content %}
77 {% endblock %}
78 {% include 'footer.html' %}
79 </body>
80
81 </html>

```

- La página de “Carrito de Compras” de la tienda en línea, donde se muestran los productos agregados al carrito y se permite al usuario interactuar con el carrito para modificar cantidades, eliminar productos, guardar el pedido o cancelar el carrito. Además, se utilizan mensajes de alerta para proporcionar retroalimentación visual al usuario en ciertas acciones.

Listing 12: carrito.html

```

1  {% extends 'base.html' %}
2  {% load static %}
3  {% block content %}
4  <link rel="stylesheet"
5      href="https://cdn.jsdelivr.net/npm/sweetalert2@11.1.4/dist/sweetalert2.min.css">
6  <div class="banner">
7      
9      <h1 class="text-center text-warning text-uppercase fw-bolder"
10         style="position: absolute; top: 50%; left: 50%; transform: translate(-50%,
11             -50%);">Carrito de Compras</h1>
12  </div>
13  <div>
14      {% if messages %}
15      <div class="container">
16          <div class="row justify-content-center">
17              <div class="col-md-8">
18                  {% for message in messages %}
19                  <div class="alert alert-danger alert-dismissible fade show" role="alert">
20                      {{ message }}
21                      <button type="button" class="btn-close" data-bs-dismiss="alert"
22                          aria-label="Close"></button>
23                  </div>
24                  {% endfor %}
25              </div>
26          </div>
27      </div>
28      {% endif %}
29  </div>
30  <div class="container">
31      <br>

```

```

28     {% csrf_token %}
29     {% if carrito %}
30     <table class="table">
31         <thead>
32             <tr>
33                 <th>Producto</th>
34                 <th>Cantidad</th>
35                 <th>Precio Unitario</th>
36                 <th>Subtotal</th>
37                 <th>Acciones</th>
38             </tr>
39         </thead>
40         <tbody>
41             {% for item in carrito %}
42             <tr>
43                 <td>{{ item.articulo.ArtNom }}</td>
44                 <td>
45                     <div class="input-group">
46                         <button class="btn btn-outline-secondary" type="button"
47                             onclick="restarCantidad({{ item.pk }})">-</button>
48                         <input type="number" class="form-control" value="{{ item.cantidad }}"
49                             min="1"
50                             max="{{ item.articulo.ArtSto }}" onchange="actualizarCantidad({{
51                                 item.pk }}, this.value)">
52                         <button class="btn btn-outline-secondary" type="button"
53                             onclick="sumarCantidad({{ item.pk }}, {{ item.articulo.ArtSto
54                                 }})">+</button>
55                     </div>
56                 </td>
57                 <td>S/.{{ item.articulo.ArtPreUni }}</td>
58                 <td>S/.{{ item.subtotal }}</td>
59                 <td>
60                     <a href="{% url 'eliminar_del_carrito' item.pk %}" class="btn btn-danger"
61                         onclick="eliminarProducto({{ item.pk }})">Eliminar</a>
62                 </td>
63             </tr>
64             {% endfor %}
65         </tbody>
66         <tfoot>
67             <tr>
68                 <td colspan="3"></td>
69                 <td>Total:</td>
70                 <td>S/.{{ total }}</td>
71             </tr>
72         </tfoot>
73     </table>
74     <div class="d-flex justify-content-around">
75         <a href="{% url 'productos' %}" class="btn btn-primary">Seguir comprando</a>
76         <button class="btn btn-success" onclick="guardarPedido()">Guardar pedido</button>
77         <button class="btn btn-danger" onclick="cancelarCarrito()">Cancelar</button>
78     </div>
79     {% else %}
80     <br>
81     <p>No hay productos en el carrito.</p>
82     <script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>
83     <script>

```

```
81     Swal.fire({
82         icon: 'info',
83         title: 'Carrito vaco',
84         text: 'No hay productos en el carrito.',
85         showConfirmButton: false,
86         timer: 2000
87     });
88     </script>
89     {% endif %}
90 </div>
91 <br>
92 <script>
93     function sumarCantidad(itemPk, maxCantidad) {
94         var inputCantidad = document.querySelector('[data-item-pk="{{itemPk}}"]');
95         var cantidad = parseInt(inputCantidad.value);
96         if (cantidad < maxCantidad) {
97             cantidad++;
98             inputCantidad.value = cantidad;
99             actualizarCantidad(itemPk, cantidad);
100         }
101     }
102
103     function restarCantidad(itemPk) {
104         var inputCantidad = document.querySelector('[data-item-pk="{{itemPk}}"]');
105         var cantidad = parseInt(inputCantidad.value);
106         if (cantidad > 1) {
107             cantidad--;
108             inputCantidad.value = cantidad;
109             actualizarCantidad(itemPk, cantidad);
110         }
111     }
112
113     function actualizarCantidad(itemPk, cantidad) {
114         fetch('/carrito/actualizar_cantidad/{{itemPk}}/{{cantidad}}/', {
115             method: 'POST',
116             headers: {
117                 'Content-Type': 'application/json',
118                 'X-CSRFToken': '{{csrf_token}}'
119             },
120             body: JSON.stringify({})
121         })
122         .then(response => {
123             if (response.ok) {
124                 location.reload();
125             } else {
126                 console.error('Error:', response.statusText);
127             }
128         })
129         .catch(error => {
130             console.error('Error:', error);
131         });
132     }
133     function eliminarProducto(itemPk) {
134         event.preventDefault();
135         const swalWithBootstrapButtons = Swal.mixin({
136             customClass: {
```

```
137         confirmButton: 'btn btn-success',
138         cancelButton: 'btn btn-danger'
139     },
140     buttonsStyling: false
141 });
142
143 swalWithBootstrapButtons.fire({
144     title: 'Ests seguro de que quieres eliminar este producto del carrito?',
145     text: "Esta accin no se puede deshacer.",
146     icon: 'warning',
147     showCancelButton: true,
148     confirmButtonText: 'S, eliminar',
149     cancelButtonText: 'No, cancelar',
150     reverseButtons: true,
151     timer: null
152 }).then((result) => {
153     if (result.isConfirmed) {
154         fetch('/carrito/eliminar_del_carrito/${itemPk}/', {
155             method: 'POST',
156             headers: {
157                 'Content-Type': 'application/json',
158                 'X-CSRFToken': '{ csrf_token }'
159             },
160             body: JSON.stringify({})
161         })
162         .then(response => {
163             if (response.ok) {
164                 swalWithBootstrapButtons.fire({
165                     icon: 'success',
166                     title: 'El producto ha sido eliminado del carrito.',
167                     showConfirmButton: true,
168                     backdrop: 'rgba(30, 12, 204, 0.49)'
169                     url("https://media.giphy.com/media/3oEjI6SIIHBdRxXI40/giphy.gif")
170                     center center
171                     no-repeat'
172
173                 }).then(() => {
174                     location.reload();
175                 });
176             } else {
177                 console.error('Error:', response.statusText);
178             }
179         })
180         .catch(error => {
181             console.error('Error:', error);
182         });
183     } else if (
184         result.dismiss === Swal.DismissReason.cancel
185     ) {
186         swalWithBootstrapButtons.fire({
187             icon: 'error',
188             title: 'El producto no ha sido eliminado del carrito.',
189             backdrop: 'rgba(255, 47, 79, 0.42)'
190             url("https://media.giphy.com/media/3oEjI6SIIHBdRxXI40/giphy.gif")
191             center center
192             no-repeat'
```

```
193     });
194   }
195   });
196 }
197 function guardarPedido() {
198   fetch('/carrito/guardar_pedido/', {
199     method: 'POST',
200     headers: {
201       'Content-Type': 'application/json',
202       'X-CSRFToken': '{{ csrf_token }}'
203     },
204     body: JSON.stringify({})
205   })
206   .then(response => {
207     if (response.ok) {
208       Swal.fire({
209         icon: 'success',
210         title: 'El pedido ha sido guardado exitosamente.',
211         showConfirmButton: true,
212         backdrop: 'rgba(0, 128, 0, 0.49)'
213         url("https://media.giphy.com/media/3oEjI6SIHBdRxXI40/giphy.gif")
214         center center
215         no-repeat'
216       }).then(() => {
217         location.href = '{% url "guardar_pedido" %}';
218       });
219     } else {
220       console.error('Error:', response.statusText);
221     }
222   })
223   .catch(error => {
224     Swal.fire({
225       icon: 'error',
226       title: 'Error al guardar el pedido',
227       text: 'La cantidad en el pedido supera el stock disponible.',
228       showConfirmButton: true
229     });
230     console.error('Error:', error);
231   });
232 }
233
234 function cancelarCarrito() {
235   const swalWithBootstrapButtons = Swal.mixin({
236     customClass: {
237       confirmButton: 'btn btn-success',
238       cancelButton: 'btn btn-danger'
239     },
240     buttonsStyling: false
241   });
242
243   swalWithBootstrapButtons.fire({
244     title: ' ¿ Ests  seguro de que quieres cancelar el carrito?',
245     text: 'Se eliminarn todos los productos del carrito.',
246     icon: 'warning',
247     showCancelButton: true,
248     confirmButtonText: 'S, cancelar',
```

```

249         cancelButtonText: 'No, mantener',
250         reverseButtons: true
251     }).then((result) => {
252         if (result.isConfirmed) {
253             fetch('/carrito/cancelar_carrito/', {
254                 method: 'POST',
255                 headers: {
256                     'Content-Type': 'application/json',
257                     'X-CSRFToken': '{ csrf_token }'
258                 },
259                 body: JSON.stringify({})
260             })
261             .then(response => {
262                 if (response.ok) {
263                     swalWithBootstrapButtons.fire({
264                         icon: 'success',
265                         title: 'El carrito ha sido cancelado.',
266                         showConfirmButton: true,
267                         backdrop: 'rgba(30, 12, 204, 0.49)'
268                         url("https://media.giphy.com/media/3oEjI6SIIHBdRxXI40/giphy.gif")
269                         center center
270                         no-repeat'
271                     }).then(() => {
272                         location.reload();
273                     });
274                 } else {
275                     console.error('Error:', response.statusText);
276                 }
277             })
278             .catch(error => {
279                 console.error('Error:', error);
280             });
281         } else if (result.dismiss === Swal.DismissReason.cancel) {
282             swalWithBootstrapButtons.fire({
283                 icon: 'error',
284                 title: 'El carrito no ha sido cancelado.',
285                 backdrop: 'rgba(255, 47, 79, 0.42)'
286                 url("https://media.giphy.com/media/3oEjI6SIIHBdRxXI40/giphy.gif")
287                 center center
288                 no-repeat'
289             });
290         }
291     });
292 }
293 </script>
294 <script
295     src="https://cdn.jsdelivr.net/npm/sweetalert2@11.1.4/dist/sweetalert2.min.js"></script>
296 {% endblock %}

```

■

Listing 13: encabezado.html

```

1  {% load static %}
2  <section>

```

```

3 <nav class="navbar navbar-expand-lg navbar-light bg-light">
4   <div class="container-fluid">
5     <a class="navbar-brand" href="/"></a>
6     <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
7       data-bs-target="#navbarNav" aria-controls="navbarNav" aria-expanded="false"
8       aria-label="Toggle navigation">
9       <span class="navbar-toggler-icon"></span>
10    </button>
11    <div class="collapse navbar-collapse d-flex justify-content-around" id="navbarNav">
12      <ul class="navbar-nav">
13        <li class="nav-item">
14          <a class="nav-link" href="{% url 'marca_detail' marca_id=1 %}">
15            
17          </a>
18        </li>
19        <li class="nav-item">
20          <a class="nav-link" href="{% url 'marca_detail' marca_id=2 %}">
21            
23          </a>
24        </li>
25        <li class="nav-item">
26          <a class="nav-link" href="{% url 'marca_detail' marca_id=3 %}">
27            
29          </a>
30        </li>
31        <li class="nav-item">
32          <a class="nav-link" href="{% url 'marca_detail' marca_id=4 %}">
33            
35          </a>
36        </li>
37        <li class="nav-item">
38          <a class="nav-link" href="{% url 'marca_detail' marca_id=5 %}">
39            
41          </a>
42        </li>
43        <li class="nav-item">
44          <a class="nav-link" href="{% url 'marca_detail' marca_id=6 %}">
45            
47          </a>
48        </li>
49        <li class="nav-item">
50          <a class="nav-link" href="{% url 'marca_detail' marca_id=7 %}">
51            
53          </a>
54        </li>
55        <li class="nav-item">
56          <a class="nav-link" href="{% url 'marca_detail' marca_id=8 %}">
57            
59          </a>
60        </li>
61      </ul>
62    </div>
63  </div>
64 </nav>

```

```

49         </a>
50     </li>
51     <li class="nav-item">
52         <a class="nav-link" href="{% url 'marca_detail' marca_id=9 %}">
53             
55         </a>
56     </li>
57     <li class="nav-item">
58         <a class="nav-link" href="{% url 'marca_detail' marca_id=10 %}">
59             
61         </a>
62     </li>
63     <li class="nav-item">
64         <a class="nav-link" href="{% url 'marca_detail' marca_id=11 %}">
65             
67         </a>
68     </li>
69     <li class="nav-item">
70         <a class="nav-link" href="{% url 'marca_detail' marca_id=12 %}">
71             
73         </a>
74     </li>
75     <li class="nav-item">
76         <a class="nav-link" href="{% url 'marca_detail' marca_id=13 %}">
77             
79         </a>
80     </li>
</ul>
</div>
</div>
</nav>
</section>

```

- Representa el footer de la empresa "LaPCz" contiene información relevante sobre la empresa, enlaces a diferentes marcas de laptops, detalles de contacto y su logotipo. También incluye un enlace al sitio web principal y presenta un estilo estético acorde con el diseño del sitio web.

Listing 14: footer.html

```

1  {% load static %}
2  <footer class="text-center text-lg-start text-white" style="background-color: black">
3      <!-- Grid container -->
4      <div class="container">
5          <!-- Section: Links -->
6          <section class="">
7              <!--Grid row-->
8              <div class="row">
9                  <!-- Grid column -->
10                 <div class="col-md-3 col-lg-3 col-xl-3 mx-auto mt-5">
11                     <h6 class="text-uppercase mb-4 font-weight-bold">

```



```

12         LaPC
13     </h6>
14     <p>
15         En nuestra empresa, nos dedicamos a ofrecer soluciones tecnologicas de
            alta calidad para satisfacer las necesidades de nuestros clientes.
            Nos enfocamos en la venta de laptops de ltima generacin, brindando
            una amplia variedad de marcas y modelos para que nuestros clientes
            puedan encontrar la opcin que mejor se adapte a sus necesidades.
16     </p>
17 </div>
18 <!-- Grid column -->
19
20 <hr class="w-100 clearfix d-md-none" />
21
22 <!-- Grid column -->
23 <div class="col-md-2 col-lg-2 col-xl-2 mx-auto mt-5">
24     <h6 class="text-uppercase mb-4 font-weight-bold">Marcas</h6>
25     <li><a href="{% url 'marca_detail' marca_id=1 %}" class="text-white"
26         style="text-decoration:none">Acer</a></li>
27     <li><a href="{% url 'marca_detail' marca_id=2 %}" class="text-white"
28         style="text-decoration:none">Apple</a></li>
29     <li><a href="{% url 'marca_detail' marca_id=3 %}" class="text-white"
30         style="text-decoration:none">Asus</a></li>
31     <li><a href="{% url 'marca_detail' marca_id=4 %}" class="text-white"
32         style="text-decoration:none">Dell</a></li>
33     <li><a href="{% url 'marca_detail' marca_id=5 %}" class="text-white"
34         style="text-decoration:none">HP</a></li>
35     <li><a href="{% url 'marca_detail' marca_id=6 %}" class="text-white"
36         style="text-decoration:none">Lenovo</a></li>
37     <li><a href="{% url 'marca_detail' marca_id=7 %}" class="text-white"
38         style="text-decoration:none">MSI</a></li>
39     <li><a href="{% url 'marca_detail' marca_id=8 %}" class="text-white"
40         style="text-decoration:none">Corsair</a></li>
41     <li><a href="{% url 'marca_detail' marca_id=9 %}" class="text-white"
42         style="text-decoration:none">Teros</a></li>
43     <li><a href="{% url 'marca_detail' marca_id=10 %}" class="text-white"
44         style="text-decoration:none">Thermaltake</a></li>
45     <li><a href="{% url 'marca_detail' marca_id=11 %}" class="text-white"
46         style="text-decoration:none">Microsoft</a></li>
47     <li><a href="{% url 'marca_detail' marca_id=12 %}" class="text-white"
48         style="text-decoration:none">Logitech</a></li>
49     <li><a href="{% url 'marca_detail' marca_id=13 %}" class="text-white"
50         style="text-decoration:none">JBL</a></li>
51 </div>
52 <!-- Grid column -->
53
54 <hr class="w-100 clearfix d-md-none" />
55
56 <!-- Grid column -->
57 <div class="col-md-4 col-lg-3 col-xl-3 mx-auto mt-5">
58     <h6 class="text-uppercase mb-4 font-weight-bold">Contactos</h6>
59     <p><i class="fas fa-home mr-3"></i> Av.Alianza 430, ASA, Arequipa</p>
60     <p><i class="fas fa-envelope mr-3"></i> LaPC@gmail.com</p>

```

```

51         <p><i class="fas fa-phone mr-3"></i> +51 931316402</p>
52         <p><i class="fas fa-print mr-3"></i> +51 965789531</p>
53     </div>
54     <div class="col-md-3 col-lg-2 col-xl-2 mx-auto mt-5">
55         
56     </div>
57 </div>
58 </section>
59
60 </div>
61
62
63 <!-- Copyright -->
64 <div class="text-center p-3" style="background-color: rgba(0, 0, 0, 0.2)">
65     2023 Copyright:
66     <a class="text-white" href="/">lapc.com</a>
67 </div>
68
69 </footer>

```

- La funcionalidad de este archivo permite a los clientes visualizar rápidamente sus pedidos anteriores, acceder a los detalles de cada pedido y tener la opción de eliminar un pedido si así lo desean. Esto mejora la experiencia del usuario al facilitar la gestión de sus compras pasadas en la tienda online.

Listing 15: index.html

```

1 {% extends 'base.html' %}
2
3 {% block content %}
4     <div class="container">
5         <h1 class="text-center mt-3 mb-3 display-3 text-primary fw-bolder">MIS PEDIDOS</h1>
6         <div class="row justify-content-center">
7             <div class="col-md-8">
8                 <div class="card">
9                     <div class="card-body">
10                         <table class="table table-bordered text-center">
11                             <thead class="thead-dark table-dark">
12                                 <tr>
13                                     <th class="col-4">CLIENTE</th>
14                                     <th class="col-4">FECHA PEDIDO</th>
15                                     <th class="col-4">DETALLES</th>
16                                 </tr>
17                             </thead>
18                             <tbody class="table-light">
19                                 {% for pedido in pedidos %}
20                                 <tr>
21                                     <td class="col-4">{{ pedido.PedCabCodCli }}</td>
22                                     <td class="col-4">{{ pedido.PedCabFec }}</td>
23                                     <td class="col-4">
24                                         <a href="{% url 'pedido_detalle' pedido.id %}">
25                                             <button type="button" class="btn btn-info
26                                                 btn-rounded" >
27                                                 VER
28                                             </button>

```

```

28         </a>
29         <a href="{% url 'eliminar_pedido' pedido.id %}">
30             {% csrf_token %}
31             <button type="submit" class="btn btn-danger
32                 btn-rounded ">
33                 ELIMINAR
34             </button>
35         </a>
36     </td>
37 </tr>
38 {% endfor %}
39 </tbody>
40 </table>
41 </div>
42 </div>
43 </div>
44 </div>
45 <br><br><br><br>
46 {% endblock %}

```

- Carrusel de imagenes atractiva y dinámica para la tienda online, mostrando un mensaje de bienvenida personalizado y presentando una presentación de diapositivas llamativa para captar la atención del usuario y promover ciertos contenidos importantes.

Listing 16: home.html

```

1 {% extends 'base.html' %}
2 {% block content %}
3 {% load static %}
4 {% if user.is_authenticated %}
5     <h2 class="text-center mb-4">Bienvenido {{ user.username }}</h2>
6 {% endif %}
7 <!-- Carrusel de imagenes -->
8 <div id="carouselExampleIndicators" class="carousel slide" data-bs-ride="carousel">
9     <ol class="carousel-indicators">
10         <li data-bs-target="#carouselExampleIndicators" data-bs-slide-to="0"
11             class="active"></li>
12         <li data-bs-target="#carouselExampleIndicators" data-bs-slide-to="1"></li>
13         <li data-bs-target="#carouselExampleIndicators" data-bs-slide-to="2"></li>
14     </ol>
15     <div class="carousel-inner">
16         <div class="carousel-item active">
17             
19         </div>
20         <div class="carousel-item">
21             
23         </div>
24         <div class="carousel-item">
25             
27         </div>
28     </div>
29 </div>

```

```

25 <a class="carousel-control-prev" href="#carouselExampleIndicators" role="button"
    data-bs-slide="prev">
26 <span class="carousel-control-prev-icon" aria-hidden="true"></span>
27 <span class="visually-hidden">Previous</span>
28 </a>
29 <a class="carousel-control-next" href="#carouselExampleIndicators" role="button"
    data-bs-slide="next">
30 <span class="carousel-control-next-icon" aria-hidden="true"></span>
31 <span class="visually-hidden">Next</span>
32 </a>
33 </div>
34 <br>
35 <h3 class="mt-5 mb-5 text-center">Marcas</h3>
36 <!-- Marcas -->
37 <div class="row row-cols-1 row-cols-md-4 g-1">
38   {% for marc in marcas %}
39     <div class="col mb-4">
40       <div class="card shadow-sm">
41         <a href="{% url 'marca_detail' marca_id=marc.pk %}">
42           <div class="rounded-circle overflow-hidden mx-auto d-flex
               justify-content-center align-items-center" style="width: 150px; height:
               150px;">
43             
44           </div>
45         </a>
46       </div>
47     </div>
48     {% if forloop.counter|divisibleby:4 and not forloop.last %}
49       </div><div class="row row-cols-1 row-cols-md-4 g-6">
50     {% endif %}
51     {% endfor %}
52 </div>
53 <script
    src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"></script>
54 {% endblock %}

```

8. Ejecución

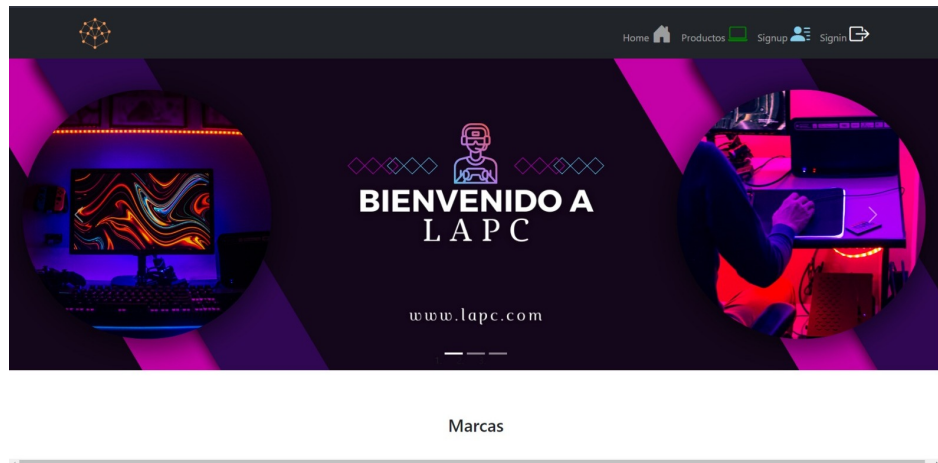


Figura 1: Página principal.

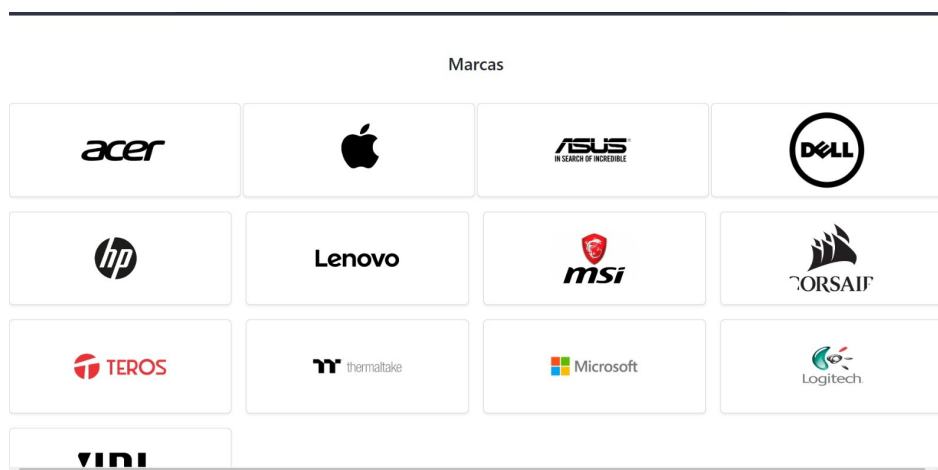


Figura 2: Página principal.

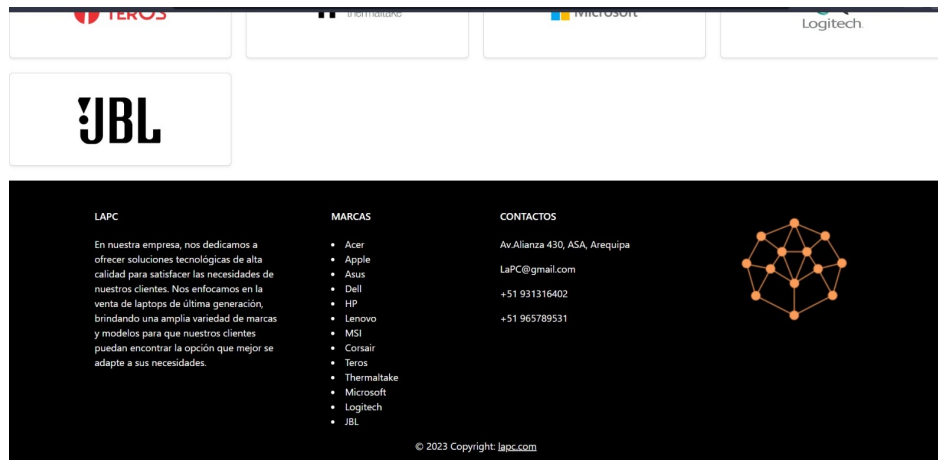


Figura 3: Página principal.

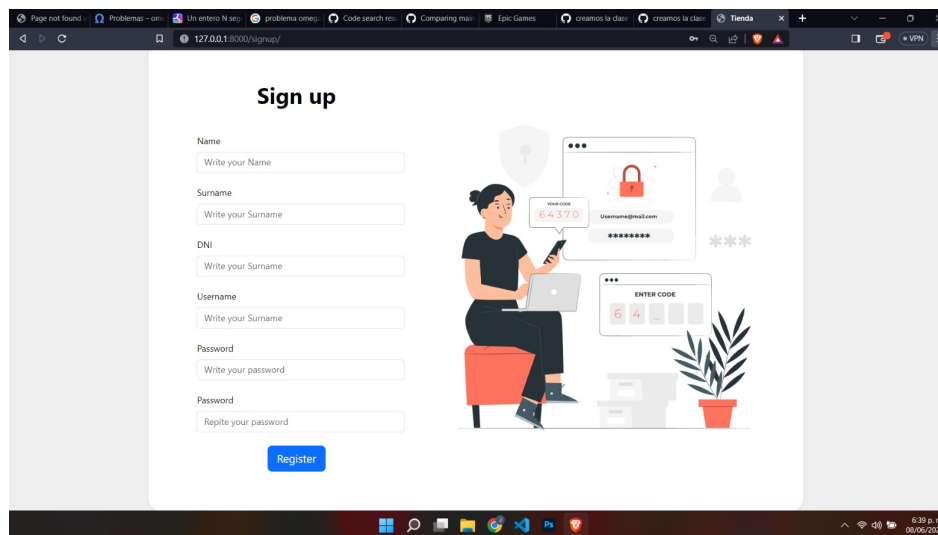


Figura 4: Sign up.

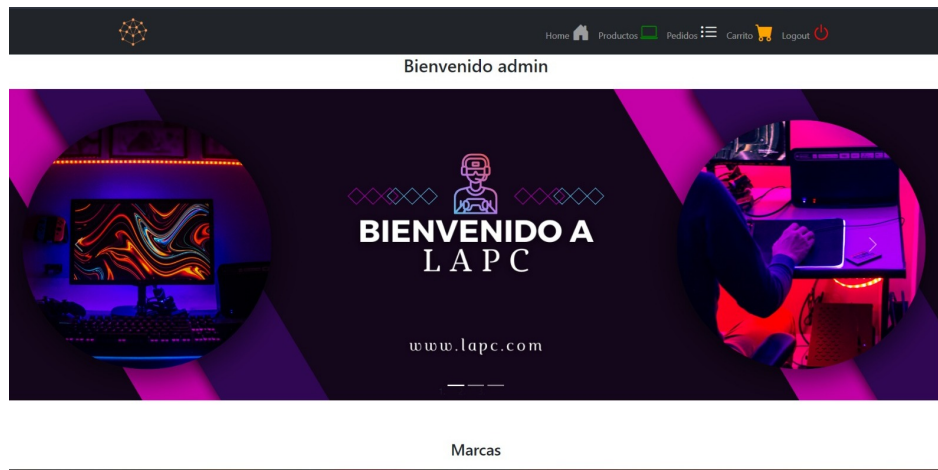


Figura 5: Al loguearnos nos muestra nuevas opciones.



Figura 6: Agregamos varios productos.

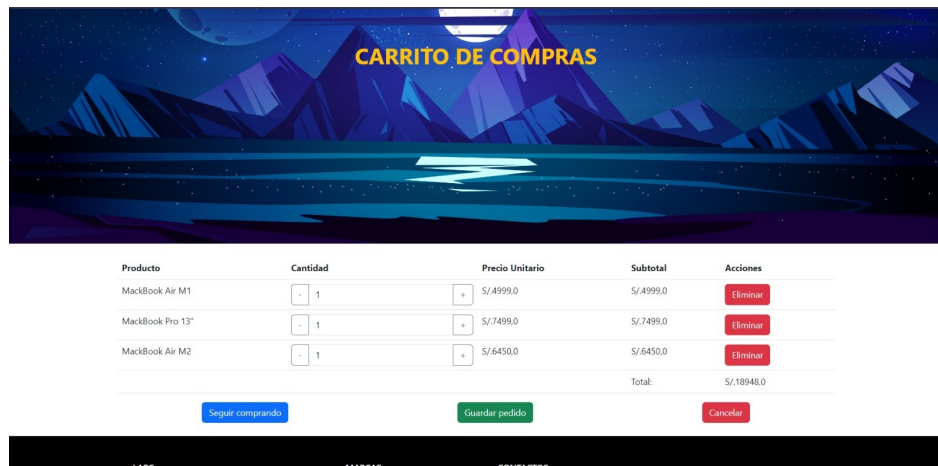


Figura 7: Carrito de compras.

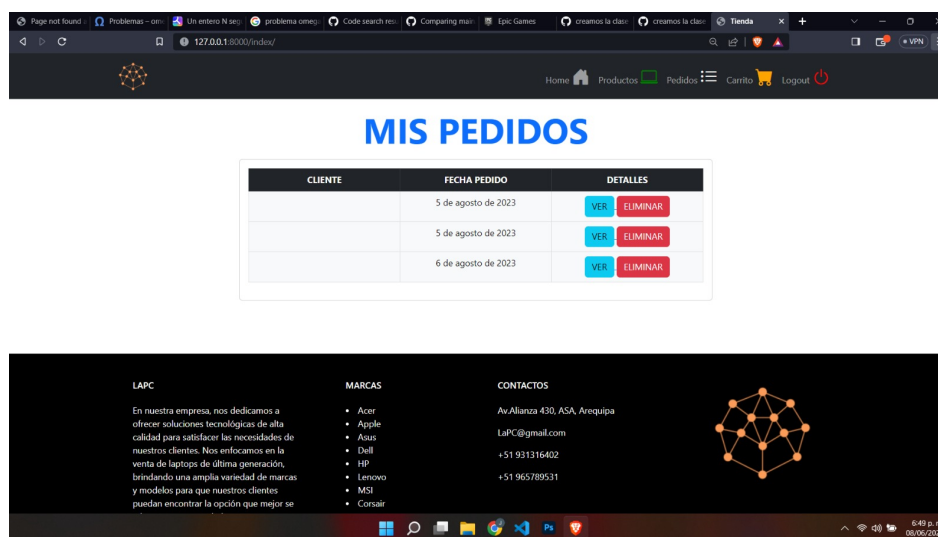


Figura 8: Pedidos.

DATOS DEL PEDIDO			
CLIENTE			
FECHA	6 de agosto de 2023		
TOTAL	S/. 18948.0		
Artículo	Cantidad	Precio unitario	Subtotal
MackBook Air M1	1	S/. 4999.0	S/. 4999.0
MackBook Pro 13"	1	S/. 7499.0	S/. 7499.0
MackBook Air M2	1	S/. 6450.0	S/. 6450.0

LAPC En nuestra empresa, nos dedicamos a ofrecer soluciones tecnológicas de alta calidad para satisfacer las necesidades de nuestros clientes. Nos enfocamos en la venta de laptops de última generación, brindando una amplia variedad de marcas y modelos para que nuestros clientes puedan encontrar la opción que mejor se adapte a sus necesidades.	MARCAS <ul style="list-style-type: none"> Acer Apple Asus Dell HP Lenovo MSI Corsair Tenso Thermaltake Microsoft Logitech 	CONTACTOS Av. Alianza 430, ASA, Arequipa laPC@gmail.com +51 931316402 +51 965789531	
--	---	--	---

Figura 9: Datos de nuestro pedido.

9. Calificación Individual

- Paulo Hidalgo Chinchay - ¿15
- Ivan Huayhua Mayta -¿15
- Jose Jaita Chura -¿15
- Ronald Garcia Valdivia -¿15

10. Referencias

- https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Tutorial_local_library_website
- <https://github.com/rescobedoq/pw2/tree/main/labs/lab05>
William S. Vincent. (2022). Django for Beginners: Build websites with Python. Django 4.0. leanpub.com. [URL]
<https://docs.djangoproject.com/en/4.1/ref/models/fields/>
https://docs.djangoproject.com/en/4.0/topics/db/examples/many_to_many/
https://docs.djangoproject.com/en/4.0/topics/db/examples/many_to_one/
<https://blog.hackajob.co/djangos-new-database-constraints/>
<https://stackoverflow.com/questions/3330435/is-there-an-sqlite-equivalent-to-mysqls-describe-table>
<https://docs.djangoproject.com/en/4.1/ref/validators/#how-validators-are-run>
<https://docs.djangoproject.com/en/4.1/ref/models/instances/>
<https://www.youtube.com/watch?v=rHux0gMZ3Eg>
<https://www.youtube.com/watch?v=OTmQQjsl0eg>
<https://tex.stackexchange.com/questions/34580/escape-character-in-latex>
<https://www.sqlitetutorial.net/sqlite-show-tables/>
<https://www.wplogout.com/export-database-diagrams-erd-from-django/>