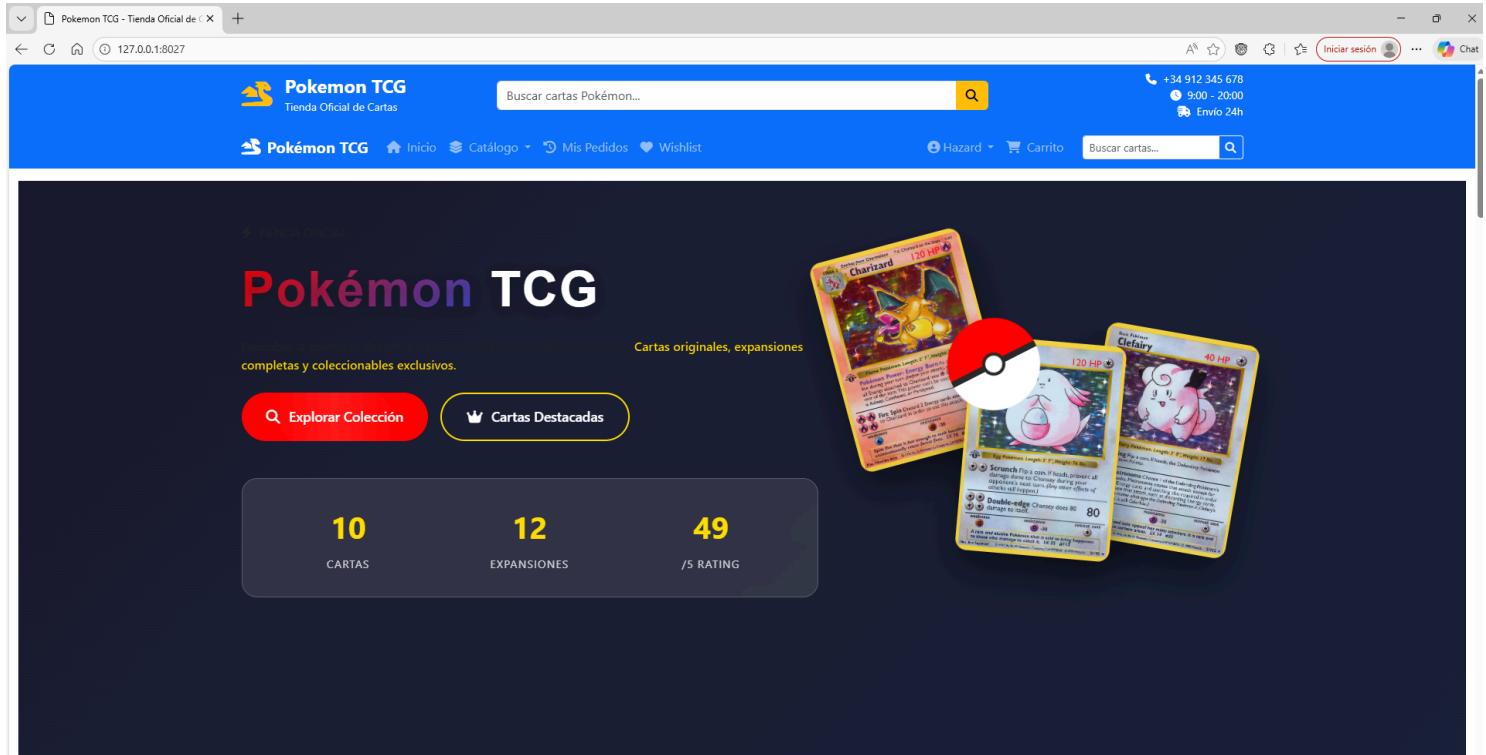


Ejecución



Tienda de Cartas

A screenshot of the 'Catálogo de Cartas Pokémon' (Card Catalogue) page. On the left, there is a sidebar with 'Filtros' (Filters) including 'Búsqueda' (Search), 'Tipo Pokémon' (Pokémon Type), 'Rareza' (Rarity), 'Expansión' (Expansion), 'Precio' (Price), 'Ordenar por' (Sort by), and buttons for 'Aplicar Filtros' (Apply Filters) and 'Limpiar Filtros' (Clear Filters). The main area shows a grid of card listings. Each listing includes the card name, type, rarity, price, availability (number of copies), and a 'Ver Detalles' (View Details) button. Some cards are shown with their front and back sides. Examples include Blastoise, Charizard, Dragonite V, Eevee, Gengar VMAX, Greninja, Lucario VSTAR, Magnezone, Shuppet, and Diancie.

Wishlist

Mi Wishlist - Pokemon TCG

Pokemon TCG Tienda Oficial de Cartas

Buscar cartas Pokémon...

+34 912 345 678
9:00 - 20:00
Envío 24h

Iniciar sesión

Inicio Catálogo Mis Pedidos Wishlist

Hazard Carrito Buscar cartas...

3 cartas

Mi Wishlist

Cartas que te gustaría comprar en el futuro

Imagen	Título	Descripción	Precio	Stock	Opciones
	Blastoise	Rara Holo	17,99€	10 disponibles	<input type="button" value="Ver Detalles"/> <input type="button" value=""/>
	Lucario VSTAR	VASTRO	41,99€	10 disponibles	<input type="button" value="Ver Detalles"/> <input type="button" value=""/>
	Dragonite V	Rara Holo V	29,99€	10 disponibles	<input type="button" value="Ver Detalles"/> <input type="button" value=""/>

Valor total estimado de tu wishlist
3 cartas • 0.00€

Carrito

Mi Carrito - Pokemon TCG

Pokemon TCG Tienda Oficial de Cartas

Buscar cartas Pokémon...

+34 912 345 678
9:00 - 20:00
Envío 24h

Iniciar sesión

Inicio Catálogo Mis Pedidos Wishlist

Hazard Carrito Buscar cartas...

Mi Carrito

4 productos en tu carrito

Productos en el carrito	Precio	Cantidad	Subtotal	Acciones
 Gengar VMAX Rara Holo VMAX Cielos Crecientes 10 unidades	47,99€	- +	239,95€	<input type="button" value=""/> <input type="button" value=""/>
 Greninja Rara Holo Brillante Destino Stock: 10 unidades	17,99€	- +	17,99€	<input type="button" value=""/> <input type="button" value=""/>
 Eevee Común Espada y Escudo Stock: 10 unidades	1,19€	- +	2,38€	<input type="button" value=""/> <input type="button" value=""/>
 Dragonite V Rara Holo V Fusion Strike Stock: 10 unidades	29,99€	- +	59,98€	<input type="button" value=""/> <input type="button" value=""/>

Resumen del Pedido

Subtotal:	320,30€
Envío:	GRATIS
Impuestos (21%):	67,26€
Total:	387,56€

(Añade -220,3€ más para envío gratis)

Tienes un cupón de descuento?
Código de descuento

Métodos de pago aceptados:

Compra 100% segura
Devolución en 30 días
Envío en 24-48h

Información de pedido al pagar

The screenshot shows the 'Checkout - Pokemon TCG' page at the 'Información' (Information) step. The top navigation bar has four steps: 'Carrito' (Cart), 'Información' (Information), 'Pago' (Payment), and 'Confirmación' (Confirmation). The 'Información' step is highlighted with a blue circle.

Información de Envío

Nombre completo *: hazel Falcon
Email *: hazaelink@gmail.com

Teléfono *:
Código Postal *:

Dirección de envío *:
Ciudad *: No especificada
Provincia:

País *: España

Notas adicionales (opcional):
Instrucciones especiales para la entrega, horarios preferidos, etc.
 Guardar esta dirección para futuros pedidos

Resumen del Pedido

Productos (10):

Producto	Cantidad	Precio
5x Gengar VMAX	5	239,95€
1x Greninja	1	17,99€
2x Eevee	2	2,38€
2x Dragonite V	2	59,98€

Subtotal: **320,30€**
Envío: **GRATIS**
Impuestos (21%): **67,26€**

Total: **387,56€**

Método de Pago

Tarjeta de Crédito/Débito

Pagar con tarjeta

PayPal

Transferencia Bancaria

Efectivo

Acepto los [Términos y Condiciones](#) y la [Política de Privacidad](#)

Mis pedidos

The screenshot shows the 'Mis Pedidos - Pokemon TCG' page at 'Mis Pedidos'. The top navigation bar has links for 'Inicio', 'Catálogo', 'Mis Pedidos' (highlighted with a red circle), and 'Wishlist'. The right side of the header has icons for phone number (+34 912 345 678), opening hours (9:00 - 20:00), and delivery (Envío 24h). The top right corner has an 'Iniciar sesión' button.

Mis Pedidos

Historial de todos tus pedidos realizados

+ Nueva Compra

Filtros: Todos los estados, dd/mm/aaaa, Buscar por número o producto..., Limpiar filtros, Limpiar filtros

Nº Pedido #c425cd04-ac06-4051-89b1-9f0cd6d05dcf | **Fecha** 08/12/2025 18:15 | **Productos** 10 productos | **Estado** Pendiente | **Total** 387,56€ | **Acciones** [link]

1 Pedidos totales | Entregados | Pendientes | 0€ Total gastado

Pokemon TCG Tienda oficial de cartas Pokémon Trading Card Game en España. | **Enlaces Rápidos** Catálogo | **Ayuda** Envíos y Entregas | **Contacto** C/ Pokémon, 123, Madrid

Formulario de Registro

The screenshot shows a web browser window for the 'Pokemon TCG' website. The URL in the address bar is 127.0.0.1:8027/register/. The page title is 'Regístrate - Pokemon TCG'. The header includes the 'Pokemon TCG' logo, a search bar ('Buscar cartas Pokémon...'), and links for 'Iniciar Sesión', 'Registrarse', and 'Carrito'. Below the header is a large blue button with the text '+ Crear Cuenta' and the sub-instruction '¡Únete a la comunidad Pokémon!'. The main form area contains fields for 'Nombre de Usuario' (with placeholder 'Ej: entrenador_ash') and 'Correo Electrónico' (with placeholder 'ejemplo@email.com'). There are also fields for 'Nombre' (placeholder 'Ej: Ash'), 'Apellidos' (placeholder 'Ej: Ketchum'), 'Contraseña' (placeholder 'Mínimo 8 caracteres'), and 'Confirmar Contraseña' (placeholder 'Repite la contraseña'). A note below the password fields states: 'La contraseña debe contener al menos 8 caracteres.' Below the form is a checkbox labeled 'Acepto los Términos y Condiciones y la Política de Privacidad'. To the right of the form is a 'Regístrate' button. At the bottom of the page, there are three promotional icons: 'Carrito persistente' (a shopping cart icon), 'Wishlist personal' (a heart icon), and 'Historial de pedidos' (a circular arrow icon). The footer of the page includes contact information: '+34 912 345 678', '9:00 - 20:00', and 'Envío 24h'.

Formulario de Inicio de sesión

The screenshot shows a web browser window for the 'Pokemon TCG' website. The URL in the address bar is 127.0.0.1:8027/login/. The page title is 'Iniciar Sesión - Pokemon TCG'. The header includes the 'Pokemon TCG' logo, a search bar ('Buscar cartas Pokémon...'), and links for 'Iniciar Sesión', 'Registrarse', and 'Carrito'. Below the header is a large blue button with the text '→ Iniciar Sesión'. The main form area contains fields for 'Usuario' (placeholder 'Nombre de usuario') and 'Contraseña' (placeholder 'Contraseña'). Below the password field is a link 'Olvidaste tu contraseña?'. A checkbox 'Recordar sesión' is also present. To the right of the form is a 'Iniciar Sesión' button. Below the form is a link '¿No tienes cuenta?'. At the bottom of the page, there is a section titled 'Credenciales de prueba' containing two entries: 'Usuario admin: admin / admin123' and 'Usuario cliente: demo / demodemo'. The footer of the page includes contact information: '+34 912 345 678', '9:00 - 20:00', and 'Envío 24h'.

Panel de Admin

Pedidos

The screenshot shows the 'Dashboard' section of the Admin Panel. On the left, a sidebar menu includes 'Dashboard', 'Cartas', 'Inventario', 'Pedidos', 'Usuarios', 'Categorías', 'Expansiones', and 'Reseñas'. A button at the bottom left says 'Ir a la tienda' and another says 'Cerrar sesión'. The main area displays a table titled 'Últimos Pedidos' with columns: N° PEDIDO, CLIENTE, FECHA, TOTAL, ESTADO, and ACCIONES. It lists four recent orders. At the bottom right of the dashboard is a 'Nueva Carta' button.

N° PEDIDO	CLIENTE	FECHA	TOTAL	ESTADO	ACCIONES
#c425cd04-ac06-4061-89b1-9f0cd6d05dcf	Hazard	08/12 18:15	387,56€	Pendiente	[Edit]
#0de06b0a-34ad-43c7-bc8f-9fef8d710a22	Yahelfuego	08/12 15:45	92,02€	Pendiente	[Edit]
#4b59a970-aa4d-44f1-ab27-0221fc9fe6cd	victor	08/12 08:59	48,49€	Pendiente	[Edit]
#25d2e38c-9097-4605-92b8-7af8dfca7db2	ash_ketchum	08/12 06:29	0,00€	ENTR	[Edit]

Cartas

The screenshot shows the 'Cartas' (Cards) section of the Admin Panel. The sidebar menu is identical to the previous dashboard. The main area has a search bar with 'Buscar...' and buttons for 'Buscar' and 'Limpiar'. Below is a table with columns: ID, CODIGO, NOMBRE, NUMERO_EN_EXPANSION, DESCRIPCION, and ACCIONES. It lists ten card entries. Each entry has three action buttons at the end.

ID	CODIGO	NOMBRE	NUMERO_EN_EXPANSION	DESCRIPCION	ACCIONES
10	SWSH12-003	Venusaur VMAX	3	Venusaur es un Pokémon de tipo Planta/Veneno. La f...	[Edit] [View] [Delete]
2	SWSH12-020	Charizard	20	Charizard es un Pokémon de tipo Fuego/Volador. Esc...	[Edit] [View] [Delete]
3	SWSH12-030	Blastoise	30	Blastoise es un Pokémon de tipo Agua. Los cañones ...	[Edit] [View] [Delete]
1	SWSH12-050	Pikachu	50	Pikachu es un Pokémon de tipo Eléctrico. Cuando se...	[Edit] [View] [Delete]
8	SWSH12-131	Dragonite V	131	Dragonite es un Pokémon de tipo Dragón/Volador. Pu...	[Edit] [View] [Delete]
7	SWSH11-072	Gengar VMAX	72	Gengar es un Pokémon de tipo Fantasma/Veneno. Se e...	[Edit] [View] [Delete]
6	SWSH10-107	Lucario VSTAR	107	Lucario es un Pokémon de tipo Lucha/Alero. Puede p...	[Edit] [View] [Delete]
9	SWSH9-088	Greninja	88	Greninja es un Pokémon de tipo Agua/Siniestro. Cre...	[Edit] [View] [Delete]
4	SWSH4-079	Mewtwo V	79	Mewtwo es un Pokémon legendario creado por manus...	[Edit] [View] [Delete]

Inventario

The screenshot shows the 'Inventarios' (Inventory) section of the Admin Panel. The left sidebar has a purple header for 'Inventario'. The main area has a search bar and a 'Crear Nuevo' button. A table lists 9 items:

ID	CARTA	CANTIDAD_DISPONIBLE	CANTIDAD_RESERVADA	PRECIO	ACCIONES
1	SWSH12-003 - Venusaur VMAX	10	0	71,98	[Edit, Delete]
2	SWSH12-020 - Charizard	10	2	17,99	[Edit, Delete]
3	SWSH12-030 - Blastoise	10	0	17,99	[Edit, Delete]
4	SWSH12-050 - Pikachu	10	0	1,19	[Edit, Delete]
5	SWSH12-131 - Dragonite V	10	2	29,99	[Edit, Delete]
6	SWSH11-072 - Gengar VMAX	10	5	47,99	[Edit, Delete]
7	SWSH10-107 - Lucario VSTAR	10	0	41,99	[Edit, Delete]
8	SWSH9-088 - Greninja	10	5	17,99	[Edit, Delete]
9	SWSH4-079 - Mewtwo V	10	0	44,98	[Edit, Delete]

The screenshot shows the 'Pedidos' (Orders) section of the Admin Panel. The left sidebar has a purple header for 'Pedidos'. The main area has a search bar and a 'Crear Nuevo' button. A table lists 8 orders:

ID	NUMERO_PEDIDO	CLIENTE	ESTADO	FECHA_PEDIDO	ACCIONES
8	c425cd04-ac06-4061-89b1-9f0cd6d05dcf	Hazard	PENDIENTE	8 de diciembre de 2025 a las 18:15	[Edit, Delete]
7	5030366e-568c-4131-a633-2f1d87c31f7c	Hazard	CARRITO	8 de diciembre de 2025 a las 18:15	[Edit, Delete]
6	0de06b0a-34ad-43c7-bc8f-9efef8d710a22	YahelFuego	PENDIENTE	8 de diciembre de 2025 a las 15:45	[Edit, Delete]
5	1776da92-0348-4cc9-8705-0fa02ecab283	YahelFuego	CARRITO	8 de diciembre de 2025 a las 15:45	[Edit, Delete]
4	4b59a970-aa4d-44f1-ab27-0221fc9fe6cd	victor	PENDIENTE	8 de diciembre de 2025 a las 08:59	[Edit, Delete]
3	ad0e46fe-d219-430e-9b1c-4cf1f51c8c292	admin	CARRITO	8 de diciembre de 2025 a las 06:58	[Edit, Delete]
1	25d2e38c-9097-4605-92b8-7af8dfca7db2	ash_ketchum	ENTR	8 de diciembre de 2025 a las 06:29	[Edit, Delete]

Usuarios

The screenshot shows the Admin Panel for a Pokemon TCG application. The left sidebar has a dark theme with purple highlights for the 'Usuarios' section. The main content area is titled 'usuarios' and contains a table of user data. The table columns are: ID, PASSWORD, LAST_LOGIN, IS_SUPERUSER, USERNAME, and ACCIONES. There are 8 users listed, each with a unique ID and password hash. The last login date varies from 'None' to '8 de diciembre de 2025'. The 'ACCIONES' column contains three icons: a blue eye, an orange edit, and a red delete.

ID	PASSWORD	LAST_LOGIN	IS_SUPERUSER	USERNAME	ACCIONES
1	pbkdf2_sha256\$100000\$KraDc74EyUTLmh9rdJfSq\$ZTp9A...	8 de diciembre de 2025 a las 18:30	True	admin	
2	pbkdf2_sha256\$600000\$SMSZlXwceAdmuZYTQvv5\$gor2X9...	None	False	ash_ketchum	
3	pbkdf2_sha256\$600000\$aJlPrBWODwhQdJlkzDekme\$pgTAwx...	None	False	misty	
4	pbkdf2_sha256\$600000\$sEemFKRtZht9awVNbzncbv\$N8nnki...	None	False	brock	
5	pbkdf2_sha256\$600000\$ldctuHvezdfTC8noGUg1BxSWzPPRX...	None	False	coleccionista	
6	pbkdf2_sha256\$600000\$scrHdqBwdD9rFl3BuOmUn\$HPksoQ...	8 de diciembre de 2025 a las 09:18	False	victor	
7	pbkdf2_sha256\$100000\$C1JV21Hj7LyjOcy3m95fU\$qlU88...	8 de diciembre de 2025 a las 15:45	False	Yahelfuego	
8	pbkdf2_sha256\$100000\$KQ8Vm0ZDD7XF7uel9cRGD\$MxG4l...	8 de diciembre de 2025 a las 18:09	False	Hazard	

Categorías

The screenshot shows the Admin Panel for a Pokemon TCG application. The left sidebar has a dark theme with purple highlights for the 'Categorías' section. The main content area is titled 'Categorías' and contains a table of category data. The table columns are: ID, NOMBRE, DESCRIPCION, ICONO, FECHA_CREACION, and ACCIONES. There are 7 categories listed, each with a unique ID and name. The creation date is '8 de diciembre de 2025 a las 06:29' for all categories. The 'ACCIONES' column contains three icons: a blue eye, an orange edit, and a red delete.

ID	NOMBRE	DESCRIPCION	ICONO	FECHA_CREACION	ACCIONES
7	Cartas Promocionales	Cartas especiales de promociones y eventos	None	8 de diciembre de 2025 a las 06:29	
6	Cartas de Energía	Cartas de energía de todos los tipos	None	8 de diciembre de 2025 a las 06:29	
5	Cartas de Entrenador	Cartas de entrenador, objeto y estadio	None	8 de diciembre de 2025 a las 06:29	
1	Pokémon Básico	Cartas de Pokémon en su etapa básica	None	8 de diciembre de 2025 a las 06:29	
4	Pokémon EX/GX/V	Cartas especiales EX, GX, V y similares	None	8 de diciembre de 2025 a las 06:29	
2	Pokémon Evolucionado	Cartas de Pokémon evolucionados (Etapa 1 y 2)	None	8 de diciembre de 2025 a las 06:29	
3	Pokémon Legендario	Cartas de Pokémon legendarios y míticos	None	8 de diciembre de 2025 a las 06:29	

Expansiones

The screenshot shows the Admin Panel for a Pokemon TCG application. The left sidebar has a dark theme with white icons and text. It includes links for Dashboard, Cartas, Inventario, Pedidos, Usuarios, Categorías, Expansiones (which is highlighted in purple), and Reseñas. At the bottom of the sidebar are buttons for 'Ir a la tienda' and 'Cerrar sesión'. The main content area is titled 'Expansiones' and features a search bar with a placeholder 'Buscar...', a 'Buscar' button, and a 'Limpiar' button. Below the search bar is a table with the following columns: ID, CODIGO, NOMBRE, FECHA_LANZAMIENTO, TOTAL_CARTAS, and ACCIONES. The table contains 9 rows of data:

ID	CODIGO	NOMBRE	FECHA_LANZAMIENTO	TOTAL_CARTAS	ACCIONES
1	SWSH12	Fusion Strike	12 de noviembre de 2021	264	[Edit, Delete]
2	SWSH11	Cielos Crecientes	27 de agosto de 2021	198	[Edit, Delete]
3	SWSH10	Choque de Sombras	18 de junio de 2021	198	[Edit, Delete]
4	SWSH9	Brillante Destino	14 de mayo de 2021	190	[Edit, Delete]
6	SWSH7	Relámpagos Ardientes	19 de marzo de 2021	185	[Edit, Delete]
5	SWSH8	Ensalada de Espadas	19 de febrero de 2021	202	[Edit, Delete]
7	SWSH6	Callejón sin Salida	22 de enero de 2021	184	[Edit, Delete]
8	SWSH5	Destello de Venganza	13 de noviembre de 2020	185	[Edit, Delete]
9	SWSH4	Tinieblas Imponentes	6 de noviembre de 2020	189	[Edit, Delete]

Reseñas

The screenshot shows the Admin Panel for a Pokemon TCG application. The left sidebar has a dark theme with white icons and text. It includes links for Dashboard, Cartas, Inventario, Pedidos, Usuarios, Categorías, Expansiones, and Reseñas (which is highlighted in purple). At the bottom of the sidebar are buttons for 'Ir a la tienda' and 'Cerrar sesión'. The main content area is titled 'Reseñas' and features a search bar with a placeholder 'Buscar...', a 'Buscar' button, and a 'Limpiar' button. Below the search bar is a table with the following columns: ID, CARTA, USUARIO, VALORACION, TITULO, and ACCIONES. The table contains 3 rows of data:

ID	CARTA	USUARIO	VALORACION	TITULO	ACCIONES
3	SWSH12-020 - Charizard	brock	5	¡Impresionante!	[Edit, Delete]
2	SWSH12-030 - Blastoise	misty	4	Gran carta de agua	[Edit, Delete]
1	SWSH12-050 - Pikachu	ash_ketchum	5	¡Mi Pikachu favorito!	[Edit, Delete]

Inicio Desde Admin

The screenshot shows the Pokemon TCG Official Store website with an administrator session active. At the top, there's a blue header bar with the logo, a search bar, and contact information (+34 912 345 678, 9:00 - 20:00, Envío 24h). Below the header, the navigation bar includes 'Inicio' and 'Panel Admin'. A green notification box at the top left says '¡Bienvenido de nuevo, admin!'. The main content area features the 'Pokémon TCG' logo and a callout for 'Cartas originales, expansiones completas y colecciónables exclusivos.' It includes a red button for 'Explorar Colección' and a white button for 'Cartas Destacadas'. Below this, a dark grey box displays statistics: '10 CARTAS', '12 EXPANSIONES', and '49 /S RATING'. To the right, there's a large image of several Poké cards, including Charizard, Mewtwo, and Clefairy.

Estructura

The screenshot shows a dark-themed file explorer window with the title "EXPLORER" at the top left. In the center, there is a tree view of a directory structure named "POKEMON_TCG-MAIN". The structure includes a ".venv" folder, a "core" folder containing "management", "migrations", and "templates" subfolders, and several other folders like "admin", "auth", "carrito", "cartas", "dashboard", "errors", "pagos", and "partials". Each folder contains one or more HTML files, indicated by orange diamond icons. The "partials" folder is currently selected, as shown by a dark grey background and white text. The right side of the window has a vertical scrollbar.

```
POKEMON_TCG-MAIN
  > .venv
  < core
    > __pycache__
    > management
    > migrations
  < templates
    < admin
      < base_admin.html
      < confirmar_eliminar.html
      < detalle.html
      < formulario.html
      < lista.html
    < auth
      < login.html
      < register.html
    < carrito
      < checkout.html
      < detalle_pedido.html
      < historial.html
      < ver.html
    < cartas
      < detalle.html
      < lista.html
      < wishlist.html
    < dashboard
      < index.html
    < errors
      < 400.html
      < 403.html
      < 404.html
      < 500.html
    < pagos
      < efectivo.html
      < paypal.html
      < tarjeta.html
      < transferencia.html
  < partials
    < footer.html
    < header.html
    < messages.html
```

```
✓ partials
  ◇ footer.html
  ◇ header.html
  ◇ messages.html
  ◇ navbar.html
  ◇ base.html
  ◇ index.html
> templatetags
> views
  ⚡ __init__.py
  ⚡ admin.py
  ⚡ apps.py
  ⚡ context_processors.py
  ⚡ forms.py
  ⚡ models.py
  ⚡ signals.py
  ⚡ tests.py
  ⚡ urls.py
> media
> pokémon_tcg
> static
  ≡ db.sqlite3
  ⚡ manage.py
  ⚡ populate_database.py
  ⓘ readme.md
  ≡ requirements.txt
```

6

Models

```
core > 🐍 models.py > 📄 Carta > 🎯 precio_estimado
  1  # core/models.py
  2  ✓ from django.db import models
  3  from django.contrib.auth.models import User
  4  from django.core.validators import MinValueValidator, MaxValueValidator
  5  import uuid
  6  from decimal import Decimal
  7  from django.utils import timezone
  8
  9
10 ✓ class Categoría(models.Model):
11     """Modelo para categorías de cartas Pokémon"""
12     nombre = models.CharField(max_length=100, unique=True)
13     descripción = models.TextField(blank=True, null=True)
14     ícono = models.CharField(max_length=50, blank=True, null=True)
15     fecha_creación = models.DateTimeField(auto_now_add=True)
16
17 ✓     class Meta:
18         verbose_name = "Categoría"
19         verbose_name_plural = "Categorías"
20         ordering = ['nombre']
21
22 ✓     def __str__(self):
23         return self.nombre
24
25     @property
26     def total_cartas(self):
27         """Total de cartas en esta categoría"""
28         return self.cartas.count()
29
30
31 ✓ class Expansion(models.Model):
32     """Modelo para las expansiones/sets de cartas Pokémon"""
33     código = models.CharField(max_length=10, unique=True)
34     nombre = models.CharField(max_length=200)
35     fecha_lanzamiento = models.DateField()
36     total_cartas = models.PositiveIntegerField()
37     símbolo = models.ImageField(upload_to='expansiones/símbolos/', blank=True, null=True)
38     activa = models.BooleanField(default=True)
39     descripción = models.TextField(blank=True, null=True)
40
41 ✓     class Meta:
42         verbose_name = "Expansión"
43         verbose_name_plural = "Expansiones"
44         ordering = ['-fecha_lanzamiento']
45
46 ✓     def __str__(self):
47         return f'{self.código} - {self.nombre}'
48
```

```
48
49     @property
50     def cartas_count(self):
51         """Número real de cartas en esta expansión"""
52         return self.cartas.count()
53
54
55 class Carta(models.Model):
56     """Modelo principal para las cartas Pokémon"""
57
58     # Tipos Pokémon
59     TIPOS_POKEMON = [
60         ('Normal', 'Normal'),
61         ('Fuego', 'Fuego'),
62         ('Agua', 'Agua'),
63         ('Planta', 'Planta'),
64         ('Eléctrico', 'Eléctrico'),
65         ('Hielo', 'Hielo'),
66         ('Lucha', 'Lucha'),
67         ('Veneno', 'Veneno'),
68         ('Tierra', 'Tierra'),
69         ('Volador', 'Volador'),
70         ('Psíquico', 'Psíquico'),
71         ('Bicho', 'Bicho'),
72         ('Roca', 'Roca'),
73         ('Fantasma', 'Fantasma'),
74         ('Dragón', 'Dragón'),
75         ('Siniestro', 'Siniestro'),
76         ('Acero', 'Acero'),
77         ('Hada', 'Hada'),
78         ('Incoloro', 'Incoloro'),
79     ]
80
81     # Rarezas
82     RAREZAS = [
83         ('COMUN', 'Común'),
84         ('INFREC', 'Infrecuente'),
85         ('RARO', 'Raro'),
86         ('RARO_HOLO', 'Raro Holo'),
87         ('RARO_LUM', 'Raro Luminosa'),
88         ('ULTRA', 'Raro Ultra'),
89         ('SECRETA', 'Rara Secreta'),
90         ('EX', 'Rara Holo EX'),
91         ('GX', 'Rara GX'),
92         ('V', 'Rara Holo V'),
93         ('VMAX', 'Rara Holo VMAX'),
```

```
core > models.py > Carta > precio_estimado
  55 class Carta(models.Model):
  56     |     ('PROMO', 'Promo'),
  57     |
  58
  59     # Condiciones
  60     CONDICIONES = [
  61         ('NM', 'Casi Perfecta (Near Mint)'),
  62         ('LP', 'Levemente Jugada (Lightly Played)'),
  63         ('MP', 'Moderadamente Jugada (Moderately Played)'),
  64         ('HP', 'Altamente Jugada (Heavily Played)'),
  65         ('D', 'Dañada (Damaged)'),
  66     ]
  67
  68
  69     # Campos básicos
  70     codigo = models.CharField(max_length=20, unique=True)
  71     nombre = models.CharField(max_length=200)
  72     numero_en_expansion = models.PositiveIntegerField()
  73     descripcion = models.TextField()
  74
  75     # Atributos
  76     tipo = models.CharField(max_length=20, choices=TIPOS_POKEMON, blank=True, null=True)
  77     tipo_secundario = models.CharField(max_length=20, choices=TIPOS_POKEMON, blank=True, null=True)
  78     hp = models.PositiveIntegerField(blank=True, null=True)
  79
  80     # Relaciones
  81     expansion = models.ForeignKey(Expansion, on_delete=models.PROTECT, related_name='cartas')
  82     categoria = models.ForeignKey(Categoría, on_delete=models.SET_NULL, null=True, blank=True)
  83
  84     # Detalles
  85     rareza = models.CharField(max_length=20, choices=RAREZAS)
  86     condicion = models.CharField(max_length=2, choices=CONDICIONES, default='NM')
  87     es_holo = models.BooleanField(default=False)
  88     primera_edicion = models.BooleanField(function max_length: Any)
  89     idioma = models.CharField(max_length=20, default='Español')
  90
  91     # Imágenes
  92     imagen_frontal = models.ImageField(upload_to='cartas/frontal/')
  93     imagen_trasera = models.ImageField(upload_to='cartas/trasera/', blank=True, null=True)
  94
  95     # Metadatos
  96     fecha_creacion = models.DateTimeField(auto_now_add=True)
  97     fecha_actualizacion = models.DateTimeField(auto_now=True)
  98     coleccionable = models.BooleanField(default=True)
  99     popularidad = models.IntegerField(default=0) # Para ordenamiento
 100
 101     class Meta:
 102         verbose_name = "Carta"
 103         verbose_name_plural = "Cartas"
```

```
140     verbose_name_plural = "Cartas"
141     ordering = ['expansion', 'numero_en_expansion']
142     indexes = [
143         models.Index(fields=['nombre']),
144         models.Index(fields=['tipo']),
145         models.Index(fields=['rareza']),
146         models.Index(fields=['popularidad']),
147     ]
148
149     def __str__(self):
150         return f"{self.codigo} - {self.nombre}"
151
152     @property
153     def tipos_completos(self):
154         """Devuelve todos los tipos de la carta"""
155         tipos = []
156         if self.tipo:
157             tipos.append(self.tipo)
158         if self.tipo_secundario:
159             tipos.append(self.tipo_secundario)
160         return tipos
161
162     @property
163     def precio_estimado(self):
164         """Calcula precio estimado"""
165         precios_base = {
166             'COMUN': 0.3, 'INFREC': 0.8, 'RARA': 2.0,
167             'RARA_HOLO': 10.0, 'RARA_LUM': 15.0, 'ULTRA': 30.0,
168             'SECRETA': 40.0, 'EX': 20.0, 'GX': 25.0,
169             'V': 20.0, 'VMAX': 35.0, 'PROMO': 5.0,
170         }
171
172         multiplicadores = {
173             'NM': 1.0, 'LP': 0.75, 'MP': 0.5, 'HP': 0.3, 'D': 0.1
174         }
175
176         base = precios_base.get(self.rareza, 1.0)
177         multiplicador = multiplicadores.get(self.condicion, 1.0)
178
179         if self.es_holo:
180             multiplicador *= 1.3
181         if self.primera_edicion:
182             multiplicador *= 2.0
183
184         return round(base * multiplicador, 2)
185
```

```
185
186     def aumentar_popularidad(self, cantidad=1):
187         """Aumenta la popularidad de la carta"""
188         self.popularidad += cantidad
189         self.save(update_fields=['popularidad'])
190
191
192 class Inventario(models.Model):
193     """Gestión de stock de cartas"""
194     carta = models.OneToOneField(Carta, on_delete=models.CASCADE, related_name='inventario')
195     cantidad_disponible = models.PositiveIntegerField(default=0)
196     cantidad_reservada = models.PositiveIntegerField(default=0)
197     precio = models.DecimalField(max_digits=10, decimal_places=2, default=0)
198     precio_promocional = models.DecimalField(max_digits=10, decimal_places=2, blank=True, null=True)
199     en_promocion = models.BooleanField(default=False)
200
201     # Métricas
202     vendidos_total = models.PositiveIntegerField(default=0)
203     valoracion_promedio = models.FloatField(default=0, validators=[MinValueValidator(0), MaxValueValidator(5)])
204
205     # Fechas
206     fecha_ingreso = models.DateTimeField(auto_now_add=True)
207     ultima_actualizacion = models.DateTimeField(auto_now=True)
208
209     class Meta:
210         verbose_name = "Inventario"
211         verbose_name_plural = "Inventarios"
212         ordering = ['-fecha_ingreso']
213
214     def __str__(self):
215         return f"Inventario de {self.carta.nombre}"
216
217     @property
218     def precio_actual(self):
219         """Precio actual (promocional si está en promoción)"""
220         return self.precio_promocional if self.en_promocion and self.precio_promocional else self.precio
221
222     @property
223     def stock_real(self):
224         """Stock disponible real"""
225         return self.cantidad_disponible - self.cantidad_reservada
226
227     @property
228     def disponible(self):
229         """¿Hay stock disponible?"""
230         return self.stock_real > 0
```

```
231
232     def reservar(self, cantidad):
233         """Reserva stock"""
234         if cantidad > self.stock_real:
235             raise ValueError(f"Stock insuficiente. Disponible: {self.stock_real}")
236         self.cantidad_reservada += cantidad
237         self.save()
238
239     def liberar(self, cantidad):
240         """Libera stock reservado"""
241         if cantidad > self.cantidad_reservada:
242             cantidad = self.cantidad_reservada
243         self.cantidad_reservada -= cantidad
244         self.save()
245
246     def vender(self, cantidad):
247         """Registra una venta"""
248         if cantidad > self.stock_real:
249             raise ValueError(f"Stock insuficiente. Disponible: {self.stock_real}")
250         self.cantidad_disponible -= cantidad
251         self.vendidos_total += cantidad
252         self.save()
253
254
255 class Pedido(models.Model):
256     """Pedidos de clientes"""
257
258     ESTADOS = [
259         ('CARRITO', 'En Carrito'),
260         ('PENDIENTE', 'Pendiente'),
261         ('PAGADO', 'Pagado'),
262         ('ENVIADO', 'Enviado'),
263         ('ENTREGADO', 'Entregado'),
264         ('CANCELADO', 'Cancelado'),
265     ]
266
267     METODOS_PAGO = [
268         ('TARJETA', 'Tarjeta'),
269         ('PAYPAL', 'PayPal'),
270         ('TRANSFERENCIA', 'Transferencia'),
271         ('EFFECTIVO', 'Efectivo'),
272     ]
273
274     # Identificación
275     numero_pedido = models.UUIDField(default=uuid.uuid4, editable=False, unique=True)
276     cliente = models.ForeignKey(User, on_delete=models.CASCADE, related_name='pedidos')
277
```

```
278     # Estado y fechas
279     estado = models.CharField(max_length=20, choices=ESTADOS, default='CARRITO')
280     fecha_pedido = models.DateTimeField(auto_now_add=True)
281     fecha_pago = models.DateTimeField(blank=True, null=True)
282     fecha_envio = models.DateTimeField(blank=True, null=True)
283     fecha_entrega = models.DateTimeField(blank=True, null=True)
284
285     # Información de envío
286     nombre_completo = models.CharField(max_length=200)
287     email = models.EmailField()
288     telefono = models.CharField(max_length=20)
289     direccion = models.TextField()
290     ciudad = models.CharField(max_length=100, blank=True, default='')
291     provincia = models.CharField(max_length=100)
292     codigo_postal = models.CharField(max_length=10)
293     pais = models.CharField(max_length=100, default='España')
294
295     # Información de pago
296     metodo_pago = models.CharField(max_length=20, choices=METODOS_PAGO, blank=True, null=True)
297
298     # Totales
299     subtotal = models.DecimalField(max_digits=10, decimal_places=2, default=0)
300     envio = models.DecimalField(max_digits=6, decimal_places=2, default=0)
301     impuestos = models.DecimalField(max_digits=10, decimal_places=2, default=0)
302     descuento = models.DecimalField(max_digits=10, decimal_places=2, default=0)
303     total = models.DecimalField(max_digits=10, decimal_places=2, default=0)
304
305     # Información adicional
306     notas = models.TextField(blank=True, null=True)
307
308 class Meta:
309     verbose_name = "Pedido"
310     verbose_name_plural = "Pedidos"
311     ordering = ['-fecha_pedido']
312     indexes = [
313         models.Index(fields=['estado']),
314         models.Index(fields=['fecha_pedido']),
315         models.Index(fields=['cliente']),
316     ]
317
318     def __str__(self):
319         return f"Pedido #{self.numero_pedido}"
320
321     def calcular_totales(self):
322         """Calcula todos los totales del pedido"""
323         # Calcular subtotal
324         subtotal = Decimal('0.00')
```

```
Click to add a breakpoint    lar subtotal
324     subtotal = Decimal('0.00')
325     for item in self.items.all():
326         item.calcular_subtotal()
327         subtotal += item.subtotal
328
329     # Calcular envío
330     envio = Decimal('0.00') if subtotal >= Decimal('100.00') else Decimal('4.95')
331
332     # Calcular impuestos
333     impuestos = (subtotal * Decimal('0.21')).quantize(Decimal('0.01'))
334
335     # Calcular total
336     total = (subtotal + envio + impuestos - self.descuento).quantize(Decimal('0.01'))
337
338     # Actualizar campos
339     self.subtotal = subtotal
340     self.envio = envio
341     self.impuestos = impuestos
342     self.total = total
343
344     # Guardar sin llamar a save() completo
345     Pedido.objects.filter(id=self.id).update(
346         subtotal=subtotal,
347         envio=envio,
348         impuestos=impuestos,
349         total=total
350     )
351
352     @property
353     def cantidad_items(self):
354         return sum(item.cantidad for item in self.items.all())
355
356     @property
357     def envio_gratis(self):
358         return self.envio == 0
359
360     def procesar_pago(self):
361         """Procesa el pago del pedido"""
362         if self.estado == 'CARRITO':
363             self.estado = 'PAGADO'
364             self.fecha_pago = timezone.now()
365             self.save()
366
367             # Reservar stock
368             for item in self.items.all():
```

```
365     item.inventario.item_cantidad)
366
367
368
369
370
371 class ItemPedido(models.Model):
372     """Items individuales en un pedido"""
373     pedido = models.ForeignKey(Pedido, on_delete=models.CASCADE, related_name='items')
374     carta = models.ForeignKey(Carta, on_delete=models.PROTECT, related_name='items_pedido')
375     inventario = models.ForeignKey(Inventory, on_delete=models.PROTECT)
376     cantidad = models.PositiveIntegerField(default=1)
377     precio_unitario = models.DecimalField(max_digits=10, decimal_places=2)
378     subtotal = models.DecimalField(max_digits=10, decimal_places=2, default=0)
379
380
381     class Meta:
382         verbose_name = "ítem de Pedido"
383         verbose_name_plural = "ítems de Pedido"
384         unique_together = ['pedido', 'carta']
385
386     def __str__(self):
387         return f'{self.cantidad}x {self.carta.nombre}'
388
389     def save(self, *args, **kwargs):
390         # Calcular subtotal antes de guardar
391         self.calcular_subtotal()
392         super().save(*args, **kwargs)
393
394     def calcular_subtotal(self):
395         """Calcula el subtotal del ítem"""
396         if self.precio_unitario and self.cantidad:
397             self.subtotal = self.precio_unitario * self.cantidad
398         else:
399             self.subtotal = Decimal('0.00')
400
401
402 class Resena(models.Model):
403     """Reseñas de cartas"""
404     carta = models.ForeignKey(Carta, on_delete=models.CASCADE, related_name='resenas')
405     usuario = models.ForeignKey(User, on_delete=models.CASCADE, related_name='resenas')
406     valoracion = models.PositiveIntegerField(
407         validators=[MinValueValidator(1), MaxValueValidator(5)])
408     )
409     titulo = models.CharField(max_length=200)
410     comentario = models.TextField()
411     condicion_recibida = models.CharField(max_length=2, choices=Carta.CONDICIONES, blank=True, null=True)
412     recomendado = models.BooleanField(default=True)
413     fecha_creacion = models.DateTimeField(auto_now_add=True)
414     fecha_actualizacion = models.DateTimeField(auto_now=True)
415     aprobada = models.BooleanField(default=False)
416     votos_positivos = models.PositiveIntegerField(default=0)
```

```

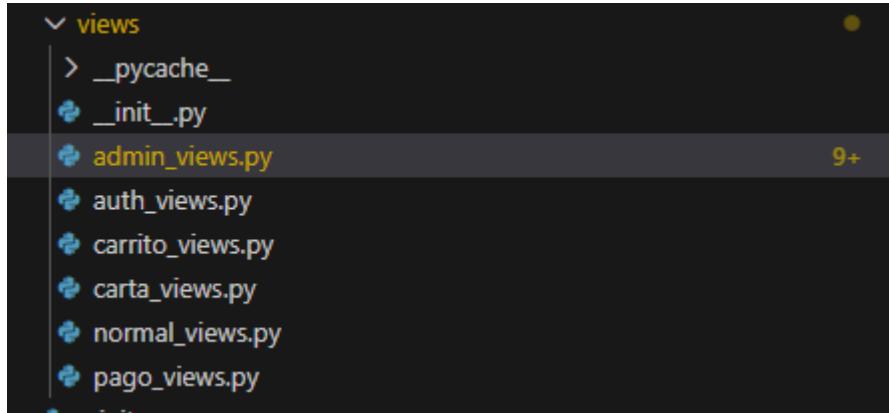
419     class Meta:
420         verbose_name = "Reseña"
421         verbose_name_plural = "Reseñas"
422         ordering = ['-fecha_creacion']
423         unique_together = ['carta', 'usuario']
424
425     def __str__(self):
426         return f"Reseña de {self.usuario.username} para {self.carta.nombre}"
427
428     @property
429     def utilidad(self):
430         return self.votos_positivos - self.votos_negativos
431
432     def votar_positivo(self):
433         self.votos_positivos += 1
434         self.save()
435
436     def votar_negativo(self):
437         self.votos_negativos += 1
438         self.save()
439
440
441     class Coleccion(models.Model):
442         """Colecciones personales de cartas"""
443         usuario = models.ForeignKey(User, on_delete=models.CASCADE, related_name='colecciones')
444         nombre = models.CharField(max_length=200)
445         descripcion = models.TextField(blank=True, null=True)
446         publica = models.BooleanField(default=False)
447         fecha_creacion = models.DateTimeField(auto_now_add=True)
448         fecha_actualizacion = models.DateTimeField(auto_now=True)
449         portada = models.ImageField(upload_to='colecciones/portadas/', blank=True, null=True)
450
451     class Meta:
452         verbose_name = "Colección"
453         verbose_name_plural = "Colecciones"
454         ordering = ['-fecha_creacion']
455         unique_together = ['usuario', 'nombre']
456
457     def __str__(self):
458         return self.nombre
459
460     @property
461     def total_cartas(self):
462         return sum(item.cantidad for item in self.cartas.all())
463
464     @property
465     def valor_estimado(self):

```

```

466         def valor_estimado(self):
467             total = Decimal('0.00')
468             for item in self.cartas.all():
469                 total += item.carta.precio_estimado * item.cantidad
470             return total
471
472     class ColeccionCarta(models.Model):
473         """Cartas dentro de una colección"""
474         colección = models.ForeignKey(Coleccion, on_delete=models.CASCADE, related_name='cartas')
475         carta = models.ForeignKey(Carta, on_delete=models.CASCADE, related_name='en_colecciones')
476         cantidad = models.PositiveIntegerField(default=1)
477         estado = models.CharField(max_length=2, choices=Carta.CONDICIONES, blank=True, null=True)
478         notas = models.TextField(blank=True, null=True)
479         fecha_agregado = models.DateTimeField(auto_now_add=True)
480
481     class Meta:
482         verbose_name = "Carta en Colección"
483         verbose_name_plural = "Cartas en Colecciones"
484         unique_together = ['colección', 'carta']
485
486     def __str__(self):
487         return f"{self.carta.nombre} en {self.colección.nombre}"

```



Admin views

```
1  from django.shortcuts import render, get_object_or_404, redirect
2  from django.contrib.auth.decorators import login_required, user_passes_test
3  from django.contrib import messages
4  from django.core.paginator import Paginator
5  from django.db.models import Q, Sum, Count, Avg
6  from django.utils import timezone
7  from datetime import timedelta
8  from ..models import (
9      Carta, Expansion, Categoria, Pedido, Inventario,
10     Resena, Coleccion, User
11 )
12 from django import forms
13 from django.forms import ModelForm
14
15 def staff_required(view_func):
16     """Decorador para verificar que el usuario es staff"""
17     decorated_view_func = login_required(user_passes_test(
18         lambda u: u.is_staff,
19         login_url='/login/'
20     )(view_func))
21     return decorated_view_func
22
23 @staff_required
24 def dashboard_view(request):
25     """Dashboard de administración con estadísticas"""
26
27     # Estadísticas generales
28     total_ventas = Pedido.objects.filter(
29         estado='ENTR',
30     ).aggregate(total=Sum('total'))['total'] or 0
31
32     pedidos_pendientes = Pedido.objects.filter(estado='PAGADO').count()
33     stock_bajo = Inventario.objects.filter(cantidad_disponible_lte=5).count()
34
35     # Ventas por día (últimos 30 días)
36     fecha_inicio = timezone.now() - timedelta(days=30)
37     ventas_recientes = Pedido.objects.filter(
38         fecha_pedido_gte=fecha_inicio,
39         estado='ENTR'
40     ).order_by('fecha_pedido')
41
42     # Últimos pedidos
43     ultimos_pedidos = Pedido.objects.exclude(estado='CARRITO').order_by(
44         '-fecha_pedido'
45     )[:10]
46
47     context = {
48         'total_ventas': total_ventas,
```

```
47     context = {
48         'total_ventas': total_ventas,
49         'pedidos_pendientes': pedidos_pendientes,
50         'stock_bajo': stock_bajo,
51         'ventas_recientes': ventas_recientes,
52         'ultimos_pedidos': ultimos_pedidos,
53     }
54     return render(request, 'dashboard/index.html', context)
55
56 @staff_required
57 def lista_admin(request, model_name):
58     """Vista genérica para listar objetos de cualquier modelo"""
59
60     # Mapeo de nombres de modelo a clases
61     model_map = {
62         'carta': Carta,
63         'expansion': Expansion,
64         'categoria': Categoria,
65         'pedido': Pedido,
66         'inventario': Inventario,
67         'resena': Resena,
68         'coleccion': Coleccion,
69         'usuario': User,
70     }
71
72     if model_name not in model_map:
73         messages.error(request, f'Modelo {model_name} no encontrado')
74         return redirect('admin_dashboard')
75
76     model_class = model_map[model_name]
77
78     # Filtrar por búsqueda
79     query = request.GET.get('q', '')
80     objetos = model_class.objects.all()
81
82     if query:
83         # Búsqueda genérica por campos comunes
84         if hasattr(model_class, 'nombre'):
85             objetos = objetos.filter(nombre__icontains=query)
86         elif hasattr(model_class, 'username'):
87             objetos = objetos.filter(username__icontains=query)
88         elif hasattr(model_class, 'codigo'):
89             objetos = objetos.filter(codigo__icontains=query)
90
91     # Página
92     paginator = Paginator(objetos, 25)
93     page_number = request.GET.get('page')
```

In 46 Col 5 Spaces: 0

```
92     paginator = Paginator(objetos, 25)
93     page_number = request.GET.get('page')
94     page_obj = paginator.get_page(page_number)
95
96     # Obtener campos para mostrar en la tabla
97     campos = []
98     if objetos.exists():
99         obj = objetos.first()
100        campos = [field.name for field in obj._meta.fields[:5]] # Primeros 5 campos
101
102    context = {
103        'model_name': model_name,
104        'model_name_display': model_class._meta.verbose_name_plural,
105        'page_obj': page_obj,
106        'campos': campos,
107        'query': query,
108    }
109    return render(request, 'admin/lista.html', context)
110
111 @staff_required
112 def detalle_admin(request, model_name, obj_id):
113     """Vista genérica para ver detalle de un objeto"""
114
115     model_map = {
116         'carta': Carta,
117         'expansion': Expansion,
118         'categoria': Categoria,
119         'pedido': Pedido,
120         'inventario': Inventario,
121         'resena': Resena,
122         'coleccion': Coleccion,
123         'usuario': User,
124     }
125
126     if model_name not in model_map:
127         messages.error(request, f'Modelo {model_name} no encontrado')
128         return redirect('admin_dashboard')
129
130     model_class = model_map[model_name]
131     objeto = get_object_or_404(model_class, id=obj_id)
132
133     context = {
134        'model_name': model_name,
135        'model_name_display': model_class._meta.verbose_name,
136        'objeto': objeto,
137    }
138    return render(request, 'admin/detalle.html', context)
```

```
138     return render(request, 'admin/detalle.html', context)
139
140 @staff_required
141 def crear_admin(request, model_name):
142     """Vista genérica para crear un objeto usando formularios de Django"""
143
144     model_map = {
145         'carta': Carta,
146         'expansion': Expansion,
147         'categoria': Categoria,
148         'inventario': Inventario,
149         'resena': Resena,
150         'colección': Colección,
151     }
152
153     if model_name not in model_map:
154         messages.error(request, f'Modelo {model_name} no encontrado')
155         return redirect('admin_dashboard')
156
157     model_class = model_map[model_name]
158
159     # Crear formulario dinámico
160     class DynamicForm(forms.ModelForm):
161         class Meta:
162             model = model_class
163             fields = '__all__'
164
165             # Widgets personalizados
166             widgets = {
167                 'descripcion': forms.Textarea(attrs={'rows': 3}),
168                 'comentario': forms.Textarea(attrs={'rows': 3}),
169                 'notas': forms.Textarea(attrs={'rows': 3}),
170                 'dirección': forms.Textarea(attrs={'rows': 2}),
171             }
172
173     if request.method == 'POST':
174         form = DynamicForm(request.POST, request.FILES)
175         if form.is_valid():
176             obj = form.save()
177             messages.success(request, f'{model_class._meta.verbose_name} creado exitosamente')
178             return redirect('detalle_admin', model_name=model_name, obj_id=obj.id)
179         else:
180             messages.error(request, 'Por favor corrige los errores del formulario')
181     else:
182         form = DynamicForm()
183
```

```
184     context = {
185         'model_name': model_name,
186         'model_name_display': f'Crear {model_class._meta.verbose_name}',
187         'form': form, # ¡ESTO ES LO QUE FALTABA!
188         'action': 'crear',
189     }
190
191     # Pasar datos adicionales según el modelo
192     if model_name == 'carta':
193         context['expansions'] = Expansion.objects.all()
194         context['categorias'] = Categoria.objects.all()
195     elif model_name == 'inventario':
196         context['cartas'] = Carta.objects.all()
197
198     return render(request, 'admin/formulario.html', context)
199
200 @staff_required
201 def editar_admin(request, model_name, obj_id):
202     """Vista genérica para editar un objeto usando formularios de Django"""
203
204     model_map = {
205         'carta': Carta,
206         'expansion': Expansion,
207         'categoria': Categoria,
208         'pedido': Pedido,
209         'inventario': Inventario,
210         'resena': Resena,
211         'coleccion': Coleccion,
212         'usuario': User,
213     }
214
215     if model_name not in model_map:
216         messages.error(request, f'Modelo {model_name} no encontrado')
217         return redirect('admin_dashboard')
218
219     model_class = model_map[model_name]
220     objeto = get_object_or_404(model_class, id=obj_id)
221
222     # Crear formulario dinámico
223     class DynamicForm(forms.ModelForm):
224         class Meta:
225             model = model_class
226             fields = '__all__'
227
228             # Widgets personalizados para campos específicos
229             widgets = {
```

```
230     'descripcion': forms.Textarea(attrs={'rows': 3}),
231     'comentario': forms.Textarea(attrs={'rows': 3}),
232     'notas': forms.Textarea(attrs={'rows': 3}),
233     'direccion': forms.Textarea(attrs={'rows': 2}),
234   }
235
236   if request.method == 'POST':
237     form = DynamicForm(request.POST, request.FILES, instance=objeto)
238     if form.is_valid():
239       form.save()
240       messages.success(request, f'{model_class._meta.verbose_name} actualizado exitosamente')
241       return redirect('detalle_admin', model_name=model_name, obj_id=obj_id)
242     else:
243       messages.error(request, 'Por favor corrige los errores del formulario')
244   else:
245     form = DynamicForm(instance=objeto)
246
247   context = {
248     'model_name': model_name,
249     'model_name_display': f'Editar {model_class._meta.verbose_name}',
250     'objeto': objeto,
251     'form': form, # ¡ESTO ES LO QUE FALTABA!
252     'action': 'editar',
253   }
254
255   # Pasar datos adicionales según el modelo
256   if model_name == 'carta':
257     context['expansions'] = Expansion.objects.all()
258     context['categorias'] =Categoria.objects.all()
259   elif model_name == 'inventario':
260     context['cartas'] = Carta.objects.all()
261   elif model_name == 'pedido':
262     context['estados'] = Pedido.ESTADOS
263     context['metodos_pago'] = Pedido.METODOS_PAGO
264
265   return render(request, 'admin/formulario.html', context)
266
267 @staff_required
268 def eliminar_admin(request, model_name, obj_id):
269   """Vista genérica para confirmar eliminación"""
270
271   model_map = {
272     'carta': Carta,
273     'expansion': Expansion,
274     'categoria': Categoria,
```

```

275     'pedido': Pedido,
276     'inventario': Inventario,
277     'resena': Resena,
278     'coleccion': Coleccion,
279   }
280
281   if model_name not in model_map:
282     messages.error(request, f'Modelo {model_name} no encontrado')
283     return redirect('admin_dashboard')
284
285   model_class = model_map[model_name]
286   objeto = get_object_or_404(model_class, id=obj_id)
287
288   if request.method == 'POST':
289     try:
290       objeto_nombre = str(objeto)
291       objeto.delete()
292       messages.success(request, f'{model_class._meta.verbose_name} "{objeto_nombre}" eliminado exitosamente')
293       return redirect('lista_admin', model_name=model_name)
294     except Exception as e:
295       messages.error(request, f'Error al eliminar: {str(e)}')
296       return redirect('detalle_admin', model_name=model_name, obj_id=obj_id)
297
298   context = {
299     'model_name': model_name,
300     'model_name_display': model_class._meta.verbose_name,
301     'objeto': objeto,
302   }
303   return render(request, 'admin/confirmar_eliminar.html', context)
304
305 @staff_required
306 def confirmar_pedido(request, pedido_id):
307   """Confirmar un pedido pendiente"""
308   pedido = get_object_or_404(Pedido, numero_pedido=pedido_id)
309
310   if pedido.estado == 'PEND':
311     pedido.estado = 'PROC'
312     pedido.save()
313     messages.success(request, f'Pedido #{pedido.numero_pedido} confirmado y en proceso')
314   else:
315     messages.warning(request, 'El pedido no está en estado pendiente')
316
317   return redirect('detalle_admin', model_name='pedido', obj_id=pedido.id)
318
319 @staff_required
320 def marcar_enviado(request, pedido_id):
321   """Marcar un pedido como enviado"""
322   pedido = get_object_or_404(Pedido, numero_pedido=pedido_id)
323
324   if pedido.estado == 'PROC':
325     pedido.estado = 'ENVI'
326     pedido.fecha_envio = timezone.now()
327     pedido.numero_seguimiento = request.POST.get('numero_seguimiento', '')
328     pedido.save()
329     messages.success(request, f'Pedido #{pedido.numero_pedido} marcado como enviado')
330   else:
331     messages.warning(request, 'El pedido no está en proceso')
332
333   return redirect('detalle_admin', model_name='pedido', obj_id=pedido.id)
334
335 @staff_required
336 def marcar_completado(request, pedido_id):
337   """Marcar un pedido como completado"""
338   pedido = get_object_or_404(Pedido, numero_pedido=pedido_id)
339
340   if pedido.estado == 'ENVI':
341     pedido.estado = 'ENTR'
342     pedido.save()
343     messages.success(request, f'Pedido #{pedido.numero_pedido} marcado como entregado')
344   else:
345     messages.warning(request, 'El pedido no está enviado')
346
347   return redirect('detalle_admin', model_name='pedido', obj_id=pedido.id)

```

Views de login y registro

```
1  from django.shortcuts import render, redirect
2  from django.contrib.auth import login, authenticate, logout
3  from django.contrib.auth.decorators import login_required
4  from django.contrib import messages
5  from django.contrib.auth.models import User
6  from ..forms import CustomUserCreationForm
7
8  def login_view(request):
9      if request.user.is_authenticated:
10          return redirect('home')
11
12      if request.method == 'POST':
13          username = request.POST.get('username')
14          password = request.POST.get('password')
15          user = authenticate(request, username=username, password=password)
16
17          if user is not None:
18              login(request, user)
19              messages.success(request, f'¡Bienvenido de nuevo, {user.username}!')
20
21          # Verificar si es admin
22          if user.is_staff:
23              return redirect('admin_dashboard')
24          return redirect('home')
25      else:
26          messages.error(request, 'Usuario o contraseña incorrectos')
27
28  return render(request, 'auth/login.html')
29
30 def logout_view(request):
31     logout(request)
32     messages.success(request, 'Has cerrado sesión correctamente')
33     return redirect('home')
34
35 def register_view(request):
36     if request.user.is_authenticated:
37         return redirect('home')
38
39     if request.method == 'POST':
40         form = CustomUserCreationForm(request.POST)
41         if form.is_valid():
42             user = form.save()
43
44             # Autenticar y loguear al usuario
45             login(request, user)
46             messages.success(request, f'¡Cuenta creada exitosamente! ¡Bienvenido, {user.username}!')
47             return redirect('home')
48     else:
49         form = CustomUserCreationForm()
50
51     return render(request, 'auth/register.html', {'form': form})
52
53 def error_404(request, exception=None):
54     return render(request, 'errors/404.html', status=404)
55
56 def error_500(request):
57     return render(request, 'errors/500.html', status=500)
```

Views Carrito

```
1 # core/views/carrito_views.py
2 from django.shortcuts import render, get_object_or_404, redirect
3 from django.contrib.auth.decorators import login_required
4 from django.contrib import messages
5 from django.db import transaction
6 from ..models import Carta, Pedido, ItemPedido, inventario
7 from decimal import Decimal
8 from django.utils import timezone
9
10
11 @login_required
12 def ver_carrito(request):
13     """Vista del carrito de compras"""
14     try:
15         # Buscar carrito existente
16         carrito = Pedido.objects.filter(
17             cliente=request.user,
18             estado='CARRITO'
19         ).first()
20
21         # Si no existe, crear uno nuevo CON TODOS LOS CAMPOS REQUERIDOS
22         if not carrito:
23             nombre_usuario = f"{request.user.first_name or ''} {request.user.last_name or ''}".strip()
24             if not nombre_usuario:
25                 nombre_usuario = request.user.username
26
27             carrito = Pedido.objects.create(
28                 cliente=request.user,
29                 estado='CARRITO',
30                 nombre_completo=nombre_usuario,
31                 email=request.user.email or '',
32                 telefono='',
33                 direccion='',
34                 ciudad='No especificada', # VALOR POR DEFECTO REQUERIDO
35                 provincia='',
36                 codigo_postal='',
37                 pais='España',
38                 subtotal=Decimal('0.00'),
39                 envio=Decimal('0.00'),
40                 impuestos=Decimal('0.00'),
41                 total=Decimal('0.00')
42             )
43             messages.info(request, "Se ha creado un nuevo carrito para ti.")
44
45         # Calcular totales
46         carrito.calcular_totales()
47
48         context = {
49             'carrito': carrito,
50             'carta': Carta.objects.all(),
51             'ultimo_pedido': Pedido.objects.filter(cliente=request.user).order_by('-fecha').first(),
52             'ultimo_pedido_id': ultimo_pedido.id if ultimo_pedido else None
53         }
54
55         return render(request, 'carrito.html', context)
56     except Exception as e:
57         print(f"Error en la vista ver_carrito: {str(e)}")
58         messages.error(request, "Ocurrió un error al cargar el carrito. Por favor, inténtalo de nuevo más tarde.")
59         return redirect('ver_carrito')
60
61     return render(request, 'carrito.html', {'carrito': carrito})
```

```
49     'carrito': carrito,
50     'items : carrito.items.all().select_related('carta', 'inventario'),
51   }
52
53   return render(request, 'carrito/ver.html', context)
54
55 except Exception as e:
56   messages.error(request, f"Error al cargar el carrito: {str(e)}")
57   return redirect('lista_cartas')
58
59
60 @login_required
61 def agregar_al_carrito(request, carta_id):
62   """Aregar una carta al carrito"""
63   try:
64     carta = get_object_or_404(Carta, id=carta_id)
65     cantidad = int(request.POST.get('cantidad', 1))
66
67     # Verificar stock
68     if not hasattr(carta, 'inventario') or carta.inventario is None:
69       messages.error(request, f'No hay inventario para {carta.nombre}')
70       return redirect('detalle_carta', carta_id=carta_id)
71
72     if carta.inventario.stock_real < cantidad:
73       messages.error(request, f'Stock insuficiente para {carta.nombre}')
74       return redirect('detalle_carta', carta_id=carta_id)
75
76     # Obtener o crear carrito CON TODOS LOS CAMPOS REQUERIDOS
77     carrito = Pedido.objects.filter(
78       cliente=request.user,
79       estado='CARRITO'
80     ).first()
81
82     if not carrito:
83       nombre_usuario = f'{request.user.first_name or ''} {request.user.last_name or ''}'.strip()
84       if not nombre_usuario:
85         nombre_usuario = request.user.username
86
87       carrito = Pedido.objects.create(
88         cliente=request.user,
89         estado='CARRITO',
90         nombre_completo=nombre_usuario,
91         email=request.user.email or '',
92         telefono='',
93         direccion='',
94         ciudad='No especificada', # VALOR POR DEFECTO REQUERIDO
95         provincia='',
96         codig postal=''
```

In 1 Col 1 - Source: 4 - LITE

```
96     codigo_postal='',
97     pais='España',
98     subtotal=Decimal('0.00'),
99     envio=Decimal('0.00'),
100    impuestos=Decimal('0.00'),
101    total=Decimal('0.00')
102 )
103
104 # Verificar si ya existe el item
105 item_existente = carrito.items.filter(carta=carta).first()
106
107 if item_existente:
108     # Actualizar cantidad
109     nueva_cantidad = item_existente.cantidad + cantidad
110     if nueva_cantidad <= carta.inventario.stock_real:
111         item_existente.cantidad = nueva_cantidad
112         item_existente.save()
113         messages.success(request, f'Cantidad actualizada para {carta.nombre}')
114     else:
115         messages.error(request, f'Stock insuficiente para agregar más unidades de {carta.nombre}')
116 else:
117     # Crear nuevo item
118     ItemPedido.objects.create(
119         pedido=carrito,
120         carta=carta,
121         inventario=carta.inventario,
122         cantidad=cantidad,
123         precio_unitario=carta.inventario.precio_actual
124     )
125     messages.success(request, f'{carta.nombre} añadida al carrito')
126
127 # Recalcular totales
128 carrito.calcular_totales()
129
130 return redirect('ver_carrito')
131
132 except Exception as e:
133     messages.error(request, f'Error: {str(e)}')
134     return redirect('detalle_carta', carta_id=carta_id)
135
136 def crear_carrito_usuario(user):
137     """Crea un nuevo carrito para un usuario con todos los campos requeridos"""
138     nombre_usuario = f'{user.first_name or ''} {user.last_name or ''}'.strip()
139     if not nombre_usuario:
140         nombre_usuario = user.username
141
142     return Pedido.objects.create(
```

```
144     estado='CARRITO',
145     nombre_completo=nombre_usuario,
146     email=user.email or '',
147     telefono='',
148     direccion='',
149     ciudad='No especificada', # Campo requerido
150     provincia='',
151     codigo_postal='',
152     pais='España',
153     subtotal=Decimal('0.00'),
154     envio=Decimal('0.00'),
155     impuestos=Decimal('0.00'),
156     total=Decimal('0.00')
157 )
158
159
160 @login_required
161 def ver_carrito(request):
162     """Vista del carrito de compras - Versión simplificada"""
163     try:
164         # Buscar carrito existente
165         carrito = Pedido.objects.filter(
166             cliente=request.user,
167             estado='CARRITO'
168         ).first()
169
170         # Si no existe, crear uno nuevo usando la función auxiliar
171         if not carrito:
172             carrito = crear_carrito_usuario(request.user)
173
174         # Calcular totales
175         carrito.calcular_totales()
176
177         Click to add a breakpoint
178         context = {
179             'carrito': carrito,
180             'items': carrito.items.all().select_related('carta', 'inventario'),
181         }
182
183         return render(request, 'carrito/ver.html', context)
184
185     except Exception as e:
186         messages.error(request, f"Error al cargar el carrito: {str(e)}")
187         return redirect('lista_cartas')
188
189     @login_required
190     def actualizar_carrito(request, item_id):
191         """Actualizar el carrito de compras"""
192
193         carrito = request.user.pedido
194
195         item = carrito.items.get(id=item_id)
196
197         if 'action' in request.POST:
198             if request.POST['action'] == 'remove':
199                 carrito.items.remove(item)
200             elif request.POST['action'] == 'update':
201                 cantidad = int(request.POST['cantidad'])
202                 if cantidad > 0:
203                     item.cantidad = cantidad
204                     item.save()
205
206         carrito.calcular_totales()
207
208         return redirect('ver_carrito')
```

```
190     """Actualizar cantidad de un item en el carrito"""
191     try:
192         item = get_object_or_404(ItemPedido, id=item_id, pedido_cliente=request.user)
193
194         if request.method == 'POST':
195             cantidad = int(request.POST.get('cantidad', 1))
196
197             if cantidad <= 0:
198                 item.delete()
199                 messages.success(request, 'Item eliminado del carrito')
200             elif cantidad <= item.inventario.stock_real:
201                 item.cantidad = cantidad
202                 item.save()
203                 messages.success(request, 'Cantidad actualizada')
204             else:
205                 messages.error(request, 'Stock insuficiente')
206
207         return redirect('ver_carrito')
208
209     except Exception as e:
210         messages.error(request, f"Error: {str(e)}")
211         return redirect('ver_carrito')
212
213
214 @login_required
215 def eliminar_del_carrito(request, item_id):
216     """Eliminar un item del carrito"""
217     try:
218         item = get_object_or_404(ItemPedido, id=item_id, pedido_cliente=request.user)
219         carta_nombre = item.carta.nombre
220         pedido = item.pedido
221         item.delete()
222         messages.success(request, f'{carta_nombre} eliminado del carrito')
223
224         # Recalcular totales
225         pedido.calcular_totales()
226
227     return redirect('ver_carrito')
228
229 except Exception as e:
230     messages.error(request, f"Error: {str(e)}")
231     return redirect('ver_carrito')
232
233
234 @login_required
```

```
235 def vaciar_carrito(request):
236     """Vaciar completamente el carrito"""
237     try:
238         carrito = Pedido.objects.filter(cliente=request.user, estado='CARRITO').first()
239         if carrito:
240             carrito.items.all().delete()
241             carrito.calcular_totales()
242             messages.success(request, 'Carrito vaciado')
243         return redirect('ver_carrito')
244
245     except Exception as e:
246         messages.error(request, f"Error: {str(e)}")
247         return redirect('ver_carrito')
248
249
250 @login_required
251 def checkout_view(request):
252     """Vista de checkout para completar la compra"""
253     try:
254         carrito = Pedido.objects.filter(cliente=request.user, estado='CARRITO').first()
255
256         if not carrito or carrito.items.count() == 0:
257             messages.warning(request, 'Tu carrito está vacío')
258             return redirect('lista_cartas')
259
260         # Verificar stock
261         for item in carrito.items.all():
262             if item.cantidad > item.inventario.stock_real:
263                 messages.error(request, f'Stock insuficiente para {item.carta.nombre}')
264                 return redirect('ver_carrito')
265
266         if request.method == 'POST':
267             # Procesar información (usar nombres CORRECTOS de campos)
268             carrito.nombre_completo = request.POST.get('nombre_completo')
269             carrito.email = request.POST.get('email')
270             carrito.telefono = request.POST.get('telefono')
271             carrito.direccion = request.POST.get('direccion')
272             carrito.ciudad = request.POST.get('ciudad')
273             carrito.provincia = request.POST.get('provincia')
274             carrito.codigo_postal = request.POST.get('codigo_postal')
275             carrito.pais = request.POST.get('pais', 'España')
276             carrito.metodo_pago = request.POST.get('metodo_pago')
277             carrito.notas = request.POST.get('notas')
278             carrito.estado = 'PENDIENTE' # Cambiar estado a pendiente
279             carrito.save()
280
281             # Reservar stock
282             for item in carrito.items.all():
```

```
283     item.inventario.reservar(item.cantidad)
284
285     messages.success(request, '¡Pedido realizado con éxito!')
286     return redirect('mis_pedidos')
287
288     context = {
289         'carrito': carrito,
290     }
291     return render(request, 'carrito/checkout.html', context)
292
293 except Exception as e:
294     messages.error(request, f"Error: {str(e)}")
295     return redirect('ver_carrito')
296
297 @login_required
298 def mis_pedidos(request):
299     """Historial de pedidos del usuario"""
300     try:
301         pedidos = Pedido.objects.filter(
302             cliente=request.user
303         ).exclude(estado='CARRITO').order_by('-fecha_pedido')
304
305         context = {
306             'pedidos': pedidos,
307         }
308         return render(request, 'carrito/historial.html', context)
309
310     except Exception as e:
311         messages.error(request, f"Error: {str(e)}")
312         return redirect('lista_cartas')
313
314
315 @login_required
316 def detalle_pedido(request, pedido_id):
317     """Detalle de un pedido específico"""
318     try:
319         pedido = get_object_or_404(
320             Pedido,
321             numero_pedido=pedido_id,
322             cliente=request.user
323         )
324
325         context = {
326             'pedido': pedido,
327         }
328         return render(request, 'carrito/detalle_pedido.html', context)
329
330
331         return render(request, 'carrito/detalle_pedido.html', context)
332
333 except Exception as e:
334     messages.error(request, f"Error: {str(e)}")
335     return redirect('mis_pedidos')
```

Carta Views

```
core > views > carta_views.py > buscar_cartas
 1  from django.shortcuts import render, get_object_or_404
 2  from django.contrib.auth.decorators import login_required
 3  from django.http import JsonResponse
 4  from django.db.models import Q
 5  from ..models import Carta, Categoria, Expansion, Coleccion, ColeccionCarta
 6
 7  def home_view(request):
 8      """Vista principal/presentación del sitio - VERSIÓN CORREGIDA"""
 9      try:
10          # Cartas destacadas (las más populares) - CORREGIDO
11          cartas_destacadas = Carta.objects.filter(
12              | colecciónable=True
13          ).select_related('inventario', 'expansion').order_by('-popularidad')[:6]
14
15          # Nuevas llegadas
16          nuevas_cartas = Carta.objects.filter(
17              | colecciónable=True
18          ).select_related('inventario', 'expansion').order_by('-fecha_creacion')[:4]
19
20          # Ofertas especiales
21          cartas_oferta = Carta.objects.filter(
22              | inventario_en_promocion=True,
23              | colecciónable=True
24          ).select_related('inventario', 'expansion')[:3]
25
26          # Estadísticas
27          total_cartas = Carta.objects.filter(colecciónable=True).count()
28          total_expansiones = Expansion.objects.filter(activa=True).count()
29          categorias = Categoria.objects.all()[:6]
30
31      except Exception as e:
32          # En caso de error
33          cartas_destacadas = []
34          nuevas_cartas = []
35          cartas_oferta = []
36          total_cartas = 0
37          total_expansiones = 0
38          categorias = []
39
40      context = {
41          'cartas_destacadas': cartas_destacadas,
42          'nuevas_cartas': nuevas_cartas,
43          'cartas_oferta': cartas_oferta,
44          'total_cartas': total_cartas,
45          'total_expansiones': total_expansiones,
46          'categorias': categorias,
47      }
48
49      return render(request, 'index.html', context)
--
```

```
49
50     def lista_cartas(request):
51         """Lista completa de cartas con filtros - VERSIÓN CORREGIDA"""
52         try:
53             # Obtener cartas con inventario
54             cartas = Carta.objects.filter(
55                 |   coleccionable=True
56             ).select_related('expansion', 'inventario')
57
58             # Aplicar filtros
59             tipo = request.GET.get('tipo')
60             rareza = request.GET.get('rareza')
61             expansion_id = request.GET.get('expansion')
62             query = request.GET.get('q', '')
63
64             if query:
65                 cartas = cartas.filter(
66                     Q(nombre__icontains=query) |
67                     Q(descripcion__icontains=query)
68                 )
69
70             if tipo and tipo != 'all':
71                 cartas = cartas.filter(Q(tipo=tipo) | Q(tipo_secundario=tipo))
72
73             if rareza and rareza != 'all':
74                 cartas = cartas.filter(rareza=rareza)
75
76             if expansion_id and expansion_id != 'all':
77                 cartas = cartas.filter(expansion_id=expansion_id)
78
79             # Ordenamiento CORREGIDO
80             orden = request.GET.get('orden', 'nombre')
81
82             if orden == 'precio_asc':
83                 cartas = cartas.order_by('inventario_precio')
84             elif orden == 'precio_desc':
85                 cartas = cartas.order_by('-inventario_precio')
86             elif orden == 'nombre':
87                 cartas = cartas.order_by('nombre')
88             elif orden == 'nuevo':
89                 cartas = cartas.order_by('-fecha_creacion')
90             elif orden == 'popularidad':
91                 # CORREGIDO: popularidad está en Carta, no en Inventario
92                 cartas = cartas.order_by('-popularidad')
93             else:
94                 cartas = cartas.order_by('nombre')
95
96             # Paginación
```

```
97     from django.core.paginator import Paginator
98     paginator = Paginator(cartas, 12)
99     page_number = request.GET.get('page')
100    page_obj = paginator.get_page(page_number)
101
102    # Obtener opciones de filtro
103    tipos = Carta.TIPOS_POKEMON
104    rarezas = Carta.RAREZAS
105    expansiones = Expansion.objects.filter(activa=True)
106
107    context = {
108        'cartas': page_obj,
109        'tipos': tipos,
110        'rarezas': rarezas,
111        'expansiones': expansiones,
112        'filtros_activos': {
113            'tipo': tipo,
114            'rareza': rareza,
115            'expansion': expansion_id,
116            'query': query,
117            'orden': orden,
118        }
119    }
120    return render(request, 'cartas/lista.html', context)
121
122 except Exception as e:
123     # En caso de error, mostrar cartas básicas
124     cartas = Carta.objects.filter(coleccionable=True)[:12]
125     context = {
126         'cartas': cartas,
127         'tipos': Carta.TIPOS_POKEMON,
128         'rarezas': Carta.RAREZAS,
129         'expansiones': Expansion.objects.filter(activa=True),
130         'error': str(e)
131     }
132     return render(request, 'cartas/lista.html', context)
133
134 def detalle_carta(request, carta_id):
135     """Vista detallada de una carta"""
136     carta = get_object_or_404(
137         Carta.objects.select_related('expansion', 'inventario', 'categoria'),
138         id=carta_id
139     )
140
141     # Incrementar popularidad
142     if hasattr(carta, 'inventario'):
143         carta.inventario.save()
```

```

145     # Cartas relacionadas
146     cartas_relacionadas = Carta.objects.filter(
147         Q(expansion=carta.expansion) | Q(tipo=carta.tipo) | Q(rareza=carta.rareza)
148     ).exclude(id=carta.id)[:4]
149
150     # Reseñas aprobadas
151     reseñas = carta.resenas.filter(aprobada=True)[:5]
152
153     context = {
154         'carta': carta,
155         'cartas_relacionadas': cartas_relacionadas,
156         'reseñas': reseñas,
157     }
158     return render(request, 'cartas/detalle.html', context)
159
160 def filtrar_cartas(request):
161     """Vista AJAX para filtrar cartas"""
162     if request.headers.get('X-Requested-With') == 'XMLHttpRequest':
163         # Lógica de filtrado similar a lista_cartas
164         cartas = Carta.objects.filter(
165             inventario__cantidad_disponible__gt=0
166         ).select_related('expansion', 'inventario')
167
168         # Aplicar filtros
169         tipo = request.GET.get('tipo')
170         if tipo:
171             cartas = cartas.filter(Q(tipo=tipo) | Q(tipo_secundario=tipo))
172
173         # Preparar datos para respuesta JSON
174         data = []
175         for carta in cartas[:20]: # Limitar a 20 resultados
176             data.append({
177                 'id': carta.id,
178                 'nombre': carta.nombre,
179                 'precio': str(carta.inventario.precio_actual),
180                 'imagen': carta.imagen_frontal.url if carta.imagen_frontal else '',
181                 'tipo': carta.tipo,
182                 'rareza': carta.rareza,
183                 'expansion': carta.expansion.nombre,
184                 'url': carta.get_absolute_url(),
185             })
186
187         return JsonResponse({'cartas': data})
188
189     return JsonResponse({'error': 'Solicitud no válida'}, status=400)
190
191 @login_required

```

```

193     """Vista de la wishlist del usuario"""
194     try:
195         wishlist, created = Coleccion.objects.get_or_create(
196             usuario=request.user,
197             nombre='Wishlist',
198             defaults={'descripcion': 'Mis cartas deseadas', 'publica': False}
199         )
200         cartas_wishlist = wishlist.cartas.all().select_related('carta_inventario')
201     except:
202         cartas_wishlist = []
203
204     context = {
205         'wishlist': cartas_wishlist,
206     }
207     return render(request, 'cartas/wishlist.html', context)
208
209     @login_required
210     def agregar_wishlist(request, carta_id):
211         """Aregar carta a wishlist"""
212         if request.headers.get('X-Requested-With') == 'XMLHttpRequest':
213             carta = get_object_or_404(Carta, id=carta_id)
214
215             try:
216                 wishlist, created = Coleccion.objects.get_or_create(
217                     usuario=request.user,
218                     nombre='Wishlist',
219                     defaults={'descripcion': 'Mis cartas deseadas', 'publica': False}
220                 )
221
222                 # Verificar si ya está en la wishlist
223                 if not wishlist.cartas.filter(carta=carta).exists():
224                     ColeccionCarta.objects.create(
225                         colección=wishlist,
226                         carta=carta,
227                         cantidad=1
228                     )
229                     wishlist.actualizar_estadisticas()
230                     return JsonResponse({'success': True, 'message': 'Carta añadida a wishlist'})
231                 else:
232                     return JsonResponse({'success': False, 'message': 'La carta ya está en tu wishlist'})
233             except Exception as e:
234                 return JsonResponse({'success': False, 'message': str(e)})
235
236             return JsonResponse({'error': 'Solicitud no válida'}, status=400)
237
238     @login_required
239     def eliminar_wishlist(request, carta_id):

```

```
240     """Eliminar carta de wishlist"""
241     if request.headers.get('X-Requested-With') == 'XMLHttpRequest':
242         carta = get_object_or_404(Carta, id=carta_id)
243
244     try:
245         wishlist = Coleccion.objects.get(
246             usuario=request.user,
247             nombre='Wishlist'
248         )
249         item = wishlist.cartas.filter(carta=carta).first()
250         if item:
251             item.delete()
252             wishlist.actualizar_estadisticas()
253             return JsonResponse({'success': True, 'message': 'Carta eliminada de wishlist'})
254         else:
255             return JsonResponse({'success': False, 'message': 'Carta no encontrada en wishlist'})
256     except Coleccion.DoesNotExist:
257         return JsonResponse({'success': False, 'message': 'No tienes wishlist'})
258     except Exception as e:
259         return JsonResponse({'success': False, 'message': str(e)})
260
261     return JsonResponse({'error': 'Solicitud no válida'}, status=400)
262
263 # core/views/carta_views.py - Agrega esta función:
264 def buscar_cartas(request):
265     """Vista para buscar cartas"""
266     query = request.GET.get('q', '')
267
268     if query:
269         cartas = Carta.objects.filter(
270             Q(nombre__icontains=query) |
271             Q(descripcion__icontains=query) |
272             Q(tipo__icontains=query)
273         ).select_related('inventario', 'expansion')
274     else:
275         cartas = Carta.objects.none()
276
277     context = {
278         'cartas': cartas,
279         'query': query,
280         'resultados_count': cartas.count(),
281     }
282     return render(request, 'cartas/buscar.html', context)
```

Views Normal

```
pre > views > ● normal_views.py > home_view
 1  from django.shortcuts import render, get_object_or_404
 2  from django.contrib.auth.decorators import login_required
 3  from django.http import JsonResponse
 4  from django.db.models import Q
 5  from ..models import Carta, Categoria, Expansion, Coleccion, ColeccionCarta
 6
 7  def home_view(request):
 8      return render(request, 'base.html')
```

Pago Views

```
 1  from django.shortcuts import render, redirect, get_object_or_404
 2  from django.contrib.auth.decorators import login_required
 3  from django.contrib import messages
 4  from django.utils import timezone
 5  import uuid
 6  from ..models import Pedido
 7  from ..forms import PagoTarjetaForm
 8  import json
 9  from django.http import HttpResponseRedirect, JsonResponse
10
11 @login_required
12 def pago_efectivo(request):
13     """Formulario de pago en efectivo"""
14     carrito = Pedido.objects.filter(cliente=request.user, estado='CARRITO').first()
15
16     if not carrito or carrito.items.count() == 0:
17         messages.warning(request, 'Tu carrito está vacío')
18         return redirect('lista_cartas')
19
20     if request.method == 'POST':
21         # Confirmar pago en efectivo
22         carrito.estado = 'PEND'
23         carrito.metodo_pago = 'EFEC'
24         carrito.fecha_pago = timezone.now()
25         carrito.save()
26
27         messages.success(request, '¡Pedido confirmado! Deberás pagar en efectivo al recibir tu pedido.')
28         return redirect('detalle_pedido', pedido_id=carrito.numero_pedido)
29
30     context = {
31         'carrito': carrito,
32     }
33     return render(request, 'pagos/efectivo.html', context)
34
35 @login_required
36 def pago_tarjeta(request):
37     """Formulario de pago con tarjeta"""
38     carrito = Pedido.objects.filter(cliente=request.user, estado='CARRITO').first()
39
40     if not carrito or carrito.items.count() == 0:
41         messages.warning(request, 'Tu carrito está vacío')
42         return redirect('lista_cartas')
43
44     form = PagoTarjetaForm()
45
46     if request.method == 'POST':
47         form = PagoTarjetaForm(request.POST)
48         if form.is_valid():
```

```
49     # Aquí iría la integración con pasarela de pago real
50     # Por ahora simulamos el pago exitoso
51     carrito.estado = 'PEND'
52     carrito.metodo_pago = 'TARJ'
53     carrito.fecha_pago = timezone.now()
54     carrito.transaccion_id = f"TARJ_{uuid.uuid4().hex[:16].upper()}"
55     carrito.save()
56
57     messages.success(request, '¡Pago con tarjeta procesado exitosamente!')
58     return redirect('detalle_pedido', pedido_id=carrito.numero_pedido)
59
60     context = {
61         'carrito': carrito,
62         'form': form,
63     }
64     return render(request, 'pagos/tarjeta.html', context)
65
66 @login_required
67 def pago_paypal(request):
68     """Simulación de pago con PayPal"""
69     carrito = Pedido.objects.filter(cliente=request.user, estado='CARRITO').first()
70
71     if not carrito or carrito.items.count() == 0:
72         messages.warning(request, 'Tu carrito está vacío')
73         return redirect('lista_cartas')
74
75     if request.method == 'POST':
76         # Simular procesamiento de PayPal
77         carrito.estado = 'PEND'
78         carrito.metodo_pago = 'PAYP'
79         carrito.fecha_pago = timezone.now()
80         carrito.transaccion_id = f"PAYP_{uuid.uuid4().hex[:16].upper()}"
81         carrito.save()
82
83         messages.success(request, '¡Pago con PayPal procesado exitosamente!')
84         return redirect('detalle_pedido', pedido_id=carrito.numero_pedido)
85
86     context = {
87         'carrito': carrito,
88     }
89     return render(request, 'pagos/paypal.html', context)
90
91 @login_required
92 def pago_transferencia(request):
93     """Información para pago por transferencia bancaria"""
94     carrito = Pedido.objects.filter(cliente=request.user, estado='CARRITO').first()
```

```

96     if not carrito or carrito.items.count() == 0:
97         messages.warning(request, 'Tu carrito está vacío')
98         return redirect('lista_cartas')
99
100    if request.method == 'POST':
101        # Confirmar que se realizará transferencia
102        carrito.estado = 'PEND'
103        carrito.metodo_pago = 'TRANS'
104        # No marcamos como pagado hasta confirmar la transferencia
105        carrito.save()
106
107        messages.success(request, '¡Pedido confirmado! Por favor realiza la transferencia bancaria.')
108        return redirect('detalle_pedido', pedido_id=carrito.numero_pedido)
109
110    # Datos bancarios (en producción deberían estar en settings)
111    datos_bancarios = {
112        'titular': 'Pokemon TCG S.L.',
113        'iban': 'ES12 3456 7890 1234 5678 9012',
114        'bic': 'ABCDESMXXX',
115        'concepto': f'PEDIDO-{carrito.numero_pedido}',
116        'importe': carrito.total,
117    }
118
119    context = {
120        'carrito': carrito,
121        'datos_bancarios': datos_bancarios,
122    }
123    return render(request, 'pagos/transferencia.html', context)
124
125 @login_required
126 def procesar_pago(request, metodo):
127     """Endpoint para procesar pagos (simulación)"""
128     carrito = Pedido.objects.filter(cliente=request.user, estado='CARRITO').first()
129
130     if not carrito:
131         return JsonResponse({'success': False, 'message': 'Carrito no encontrado'})
132
133     # Simular procesamiento de pago
134     carrito.estado = 'PEND'
135     carrito.metodo_pago = metodo
136
137
138
139     carrito.fecha_pago = timezone.now()
140     carrito.transaccion_id = f'{metodo}_{uuid.uuid4().hex[:16].upper()}'
141     carrito.save()
142

```

```

143
144     return JsonResponse({
145         'success': True,
146         'message': 'Pago procesado exitosamente',
147         'pedido_id': str(carrito.numero_pedido)
148     })

```

Templates

Admin base admin

```
 !DOCTYPE html 
{%
  load static %
}
<html lang="es">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>{% block title %}Admin - Pokemon TCG{% endblock %}</title>

  <!-- CSS -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
  <link rel="stylesheet" href="{% static 'css/admin.css' %}">
  <style>
    /* =====
VARIABLES Y CONFIGURACIÓN GENERAL
===== */
:root {
  /* Colores del tema Admin */
  --admin-dark: #1a1f36;
  --admin-darker: #121629;
  --admin-primary: #4a6cf7;
  --admin-secondary: #6c757d;
  --admin-success: #14b8a6;
  --admin-danger: #ef4444;
  --admin-warning: #f59e0b;
  --admin-info: #0ea5e9;
  --admin-light: #f8fafc;
  --admin-border: #e2e8f0;

  /* Gradientes */
  --gradient-admin: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
  --gradient-success: linear-gradient(135deg, #10b981 0%, #059669 100%);
  --gradient-danger: linear-gradient(135deg, #ef4444 0%, #dc2626 100%);
  --gradient-warning: linear-gradient(135deg, #f59e0b 0%, #d97706 100%);

  /* Sombras */
  --shadow-sm: 0 1px 3px rgba(0, 0, 0, 0.12);
  --shadow-md: 0 4px 6px rgba(0, 0, 0, 0.1);
  --shadow-lg: 0 10px 25px rgba(0, 0, 0, 0.15);
  --shadow-xl: 0 20px 40px rgba(0, 0, 0, 0.2);

  /* Bordes */
  --radius-sm: 6px;
  --radius-md: 10px;
  --radius-lg: 16px;
  --radius-xl: 24px;
}
```

```
47  }
48  /* Transiciones */
49  --transition-fast: 0.2s ease;
50  --transition-normal: 0.3s ease;
51  --transition-slow: 0.5s ease;
52 }
53
54 /* ===== */
55 RESET Y BASE
56 ===== */
57 {
58   margin: 0;
59   padding: 0;
60   box-sizing: border-box;
61 }
62
63 body.admin-body {
64   font-family: 'Segoe UI', 'Roboto', 'Helvetica Neue', Arial, sans-serif;
65   background-color: #f5f7fb;
66   color: #333;
67   line-height: 1.6;
68   min-height: 100vh;
69   overflow-x: hidden;
70 }
71
72 /* ===== */
73 LAYOUT PRINCIPAL
74 ===== */
75 .admin-body {
76   display: flex;
77   min-height: 100vh;
78 }
79
80 /* ===== */
81 SIDEBAR
82 ===== */
83 .admin-sidebar {
84   width: 250px;
85   background: var(--admin-darker);
86   color: #fff;
87   position: fixed;
88   top: 0;
89   left: 0;
90   bottom: 0;
91   z-index: 1000;
92   box-shadow: var(--shadow-lg);
93   transition: transform var(--transition-normal);
```

```
95  .sidebar-header {  
96    padding: 25px 20px;  
97    border-bottom: 1px solid □rgba(255, 255, 255, 0.1);  
98    background: □rgba(0, 0, 0, 0.2);  
99  }  
100  
101  .sidebar-header h3 {  
102    font-size: 1.3rem;  
103    font-weight: 700;  
104    margin-bottom: 5px;  
105    display: flex;  
106    align-items: center;  
107  }  
108  
109  
110  .sidebar-header small {  
111    font-size: 0.85rem;  
112    color: □rgba(255, 255, 255, 0.6);  
113  }  
114  
115  .sidebar-menu {  
116    padding: 20px 0;  
117    height: calc(100vh - 180px);  
118    overflow-y: auto;  
119  }  
120  
121  .sidebar-menu::-webkit-scrollbar {  
122    width: 6px;  
123  }  
124  
125  .sidebar-menu::-webkit-scrollbar-track {  
126    background: □rgba(255, 255, 255, 0.1);  
127  }  
128  
129  .sidebar-menu::-webkit-scrollbar-thumb {  
130    background: □rgba(255, 255, 255, 0.2);  
131    border-radius: 3px;  
132  }  
133  
134  .sidebar-menu .nav {  
135    padding: 0 15px;  
136  }  
137  
138  .sidebar-menu .nav-item {  
139    margin-bottom: 5px;  
140  }
```

```
.sidebar-menu .nav-link {
  color: ■rgba(255, 255, 255, 0.8);
  padding: 12px 15px;
  border-radius: var(--radius-sm);
  text-decoration: none;
  display: flex;
  align-items: center;
  transition: all var(--transition-fast);
  font-weight: 500;
}

.sidebar-menu .nav-link:hover {
  background: □rgba(255, 255, 255, 0.1);
  color: ■#fff;
  transform: translateX(5px);
}

.sidebar-menu .nav-link.active {
  background: var(--gradient-admin);
  color: ■#fff;
  box-shadow: var(--shadow-md);
}

.sidebar-menu .nav-link i {
  width: 20px;
  margin-right: 12px;
  text-align: center;
  font-size: 1.1rem;
}

.sidebar-footer {
  position: absolute;
  bottom: 0;
  left: 0;
  right: 0;
  padding: 20px;
  border-top: 1px solid □rgba(255, 255, 255, 0.1);
  background: □rgba(0, 0, 0, 0.2);
}

.sidebar-footer .btn {
  margin-bottom: 10px;
  font-weight: 500;
}
```

```
/* =====
CONTENIDO PRINCIPAL
=====
*/
.admin-main {
  flex: 1;
  margin-left: 250px;
  min-height: 100vh;
  display: flex;
  flex-direction: column;
  transition: margin-left var(--transition-normal);
}

/* =====
TOPBAR
=====
*/
.admin-topbar {
  background: ■#ffff;
  border-bottom: 1px solid var(--admin-border);
  padding: 15px 30px;
  box-shadow: var(--shadow-sm);
  position: sticky;
  top: 0;
  z-index: 100;
}

.admin-topbar .container-fluid {
  padding: 0;
}

#sidebarToggle {
  width: 40px;
  height: 40px;
  border-radius: var(--radius-md);
  display: none;
  align-items: center;
  justify-content: center;
  padding: 0;
}

.badge.bg-danger {
  font-size: 0.75rem;
  padding: 4px 8px;
  animation: pulse 2s infinite;
}
```

```
/* =====
CONTENIDO
=====
*/
.admin-content {
  flex: 1;
  padding: 30px;
  background: #f5f7fb;
}

/* =====
FOOTER
=====
*/
.admin-footer {
  background: #fff;
  border-top: 1px solid var(--admin-border);
  padding: 20px 30px;
  font-size: 0.85rem;
  color: var(--admin-secondary);
}

.admin-footer small {
  font-size: 0.8rem;
}

/* =====
COMPONENTES GENERALES
=====
*/

/* Breadcrumb */
.breadcrumb {
  background: transparent;
  padding: 0;
  margin-bottom: 20px;
}

.breadcrumb-item a {
  color: var(--admin-primary);
  text-decoration: none;
  font-weight: 500;
}

.breadcrumb-item a:hover {
  color: #3a56d6;
  text-decoration: underline;
}
```

```
/* Cards */
.card {
    border: 1px solid var(--admin-border);
    border-radius: var(--radius-lg);
    box-shadow: var(--shadow-sm);
    transition: box-shadow var(--transition-normal);
    margin-bottom: 30px;
}

.card:hover {
    box-shadow: var(--shadow-md);
}

.card-header {
    background: ■#ffff;
    border-bottom: 1px solid var(--admin-border);
    padding: 20px;
    font-weight: 600;
    font-size: 1.1rem;
}

.card-header.bg-primary {
    background: var(--gradient-admin) !important;
    color: ■#ffff;
    border: none;
}

.card-header.bg-danger {
    background: var(--gradient-danger) !important;
    color: ■#ffff;
    border: none;
}

.card-body {
    padding: 25px;
}

/* Botones */
.btn {
    border-radius: var(--radius-md);
    font-weight: 500;
    padding: 10px 20px;
    transition: all var(--transition-fast);
    display: inline-flex;
    align-items: center;
    justify-content: center;
}
```

```
.btn-primary {
  background: var(--gradient-admin);
  border: none;
}

.btn-primary:hover {
  transform: translateY(-2px);
  box-shadow: var(--shadow-md);
}

.btn-warning {
  background: var(--gradient-warning);
  border: none;
  color: ■#fff;
}

.btn-danger {
  background: var(--gradient-danger);
  border: none;
}

.btn-outline-secondary {
  border: 2px solid var(--admin-border);
  color: var(--admin-secondary);
}

.btn-outline-secondary:hover {
  background: var(--admin-border);
  color: var(--admin-dark);
}

.btn-group .btn {
  border-radius: var(--radius-md);
}

/* Tablas */
.table-responsive {
  border-radius: var(--radius-md);
  overflow: hidden;
  box-shadow: var(--shadow-sm);
}

.table {
  margin-bottom: 0;
  border-collapse: separate;
  border-spacing: 0;
}
```

```
.table th {
  background: ■#f8fafc;
  border-bottom: 2px solid var(--admin-border);
  font-weight: 600;
  color: var(--admin-dark);
  padding: 15px;
  text-transform: uppercase;
  font-size: 0.85rem;
  letter-spacing: 0.5px;
}

.table td {
  padding: 15px;
  vertical-align: middle;
  border-bottom: 1px solid var(--admin-border);
  transition: background-color var(--transition-fast);
}

.table tbody tr:hover {
  background-color: □rgba(74, 108, 247, 0.05);
}

.table tbody tr:last-child td {
  border-bottom: none;
}

/* Formularios */
.form-control {
  border: 2px solid var(--admin-border);
  border-radius: var(--radius-md);
  padding: 10px 15px;
  transition: all var(--transition-fast);
  font-size: 0.95rem;
}

.form-control:focus {
  border-color: var(--admin-primary);
  box-shadow: 0 0 0 3px □rgba(74, 108, 247, 0.1);
}

.form-label {
  font-weight: 500;
  margin-bottom: 8px;
  color: var(--admin-dark);
}
```

```
458 .form-control:focus {
459   box-shadow: 0 0 0 3px var(--admin-primary);
460 }
461 }
462
463 .form-label {
464   font-weight: 500;
465   margin-bottom: 8px;
466   color: var(--admin-dark);
467 }
468
469 .form-check-input:checked {
470   background-color: var(--admin-primary);
471   border-color: var(--admin-primary);
472 }
473
474 .form-text {
475   font-size: 0.85rem;
476   color: var(--admin-secondary);
477 }
478
479 /* Alertas */
480 .alert {
481   border-radius: var(--radius-md);
482   border: none;
483   padding: 15px 20px;
484   margin-bottom: 20px;
485   font-weight: 500;
486 }
487
488 .alert-warning {
489   background: linear-gradient(135deg, #fef3c7 0%, #fde68a 100%);
490   color: #92400e;
491   border-left: 4px solid #f59e0b;
492 }
493
494 /* Página */
495 .pagination {
496   margin-bottom: 0;
497 }
498
499 .page-item:first-child .page-link,
500 .page-item:last-child .page-link {
501   border-radius: var(--radius-md);
502 }
503
504 .page-link {
505   color: var(--admin-primary);
506   border: 1px solid var(--admin-border);
```

```
.page-link {
    color: var(--admin-primary);
    border: 1px solid var(--admin-border);
    padding: 8px 15px;
    font-weight: 500;
}

.page-link:hover {
    background-color: ■#f8fafc;
    color: var(--admin-primary);
}

.page-item.active .page-link {
    background: var(--gradient-admin);
    border-color: var(--admin-primary);
}

/* Badges */

.badge {
    font-size: 0.75rem;
    padding: 4px 10px;
    border-radius: 20px;
    font-weight: 600;
}

.badge.bg-success {
    background: var(--gradient-success) !important;
}

.badge.bg-danger {
    background: var(--gradient-danger) !important;
}

.badge.bg-secondary {
    background: ■#e2e8f0 !important;
    color: □#475569;
}

/* =====
PÁGINAS ESPECÍFICAS
===== */

/* Dashboard */

.dashboard-stats {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
    gap: 25px;
    margin-bottom: 30px;
```

```
.stat-card {
  background: #fff;
  border-radius: var(--radius-lg);
  padding: 25px;
  box-shadow: var(--shadow-sm);
  transition: transform var(--transition-normal);
  border-left: 4px solid var(--admin-primary);
}

.stat-card:hover {
  transform: translateY(-5px);
  box-shadow: var(--shadow-md);
}

.stat-card i {
  font-size: 2rem;
  margin-bottom: 15px;
  display: block;
}

.stat-card .stat-number {
  font-size: 2rem;
  font-weight: 700;
  color: var(--admin-dark);
  margin-bottom: 5px;
}

.stat-card .stat-label {
  color: var(--admin-secondary);
  font-size: 0.9rem;
  text-transform: uppercase;
  letter-spacing: 0.5px;
}

/* Lista de elementos */
.card.mb-4 .card-body {
  padding: 20px;
}

/* Formulario de creación/edición */
.card.border-0.shadow-lg {
  border-radius: var(--radius-xl);
  overflow: hidden;
}

.card.border-0.shadow-lg .card-body {
  padding: 40px;
}
```

```
.form-check {
    margin-bottom: 15px;
}

.form-check-input {
    width: 18px;
    height: 18px;
    margin-top: 0.2em;
}

.form-check-label {
    margin-left: 5px;
    font-weight: 500;
}

/* Detalle de objeto */
.table.table-bordered {
    border: 1px solid var(--admin-border);
}

.table.table-bordered th {
    background: ■#f8fafc;
    width: 30%;
    font-weight: 600;
}

.table.table-bordered td {
    background: ■#fff;
}

.table.table-bordered .bg-light {
    background: ■#f8fafc !important;
}

.card.border-danger {
    border: 2px solid var(--admin-danger) !important;
}

.card.border-danger .card-body {
    padding: 20px;
}
```



```


    <i class="fas fa-user me-1"></i>
    {{ user.username }}
    {% if user.is_superuser %}
        <span class="badge bg-danger ms-2">Super Admin</span>
    {% endif %}

</span>
</div>

<div class="d-flex align-items-center">
    <div class="me-3">
        <span class="badge bg-primary">
            <i class="fas fa-bell me-1"></i>
            <span id="notification-count">0</span>
        </span>
    </div>
    <div class="dropdown">
        <button class="btn btn-sm btn-outline-primary dropdown-toggle" type="button" data-bs-toggle="dropdown">
            <i class="fas fa-cog me-1"></i>Opciones
        </button>
        <ul class="dropdown-menu dropdown-menu-end">
            <li><hr class="dropdown-divider"></li>
            <li><a class="dropdown-item" href="{% url 'logout' %}">
                <i class="fas fa-sign-out-alt me-2"></i>Cerrar sesión
            </a></li>
        </ul>
    </div>
</div>
</div>
</nav>

<!-- Content --&gt;
&lt;div class="admin-content"&gt;
    {% block content %}{% endblock %}
&lt;/div&gt;

<!-- Footer --&gt;
&lt;footer class="admin-footer"&gt;
    &lt;div class="container-fluid"&gt;
        &lt;div class="row align-items-center"&gt;
            &lt;div class="col-md-6"&gt;
                &lt;small class="text-muted"&gt;
                    © {{ now | Y }} Pokemon TCG - Panel de Administración
                &lt;/small&gt;
            &lt;/div&gt;
            &lt;div class="col-md-6 text-end"&gt;
                &lt;small class="text-muted"&gt;
                    &lt;i class="fas fa-server me-1"&gt;&lt;/i&gt;
                    v1.0.0 | Última actualización: {{ now | "d/m/Y H:i" }}
                &lt;/small&gt;
            &lt;/div&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/footer&gt;
&lt;/div&gt;

<!-- JavaScript --&gt;
&lt;script src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.min.js"&gt;&lt;/script&gt;
&lt;script src="{{ STATIC_URL }}js/admin.js"&gt;&lt;/script&gt;

{% block extra_js %}{% endblock %}

&lt;/body&gt;
&lt;/html&gt;
</pre>

```

CONFIRMAR ELIMINAR

```
1  {% extends 'admin/base_admin.html' %} 
2
3  {% block content %}
4  <div class="container-fluid">
5      <div class="row justify-content-center">
6          <div class="col-md-6 col-lg-5">
7              <div class="card border-0 shadow-lg">
8                  <div class="card-header bg-danger text-white text-center py-4">
9                      <h3 class="mb-0">
10                         <i class="fas fa-exclamation-triangle me-2"></i>
11                         Confirmar Eliminación
12                     </h3>
13                 </div>
14                 <div class="card-body p-4 p-md-5 text-center">
15                     <!-- Icono de advertencia -->
16                     <div class="mb-4">
17                         <i class="fas fa-trash-alt fa-4x text-danger"></i>
18                     </div>
19
20                     <!-- Mensaje de confirmación -->
21                     <h4 class="mb-3">¿Estás seguro?</h4>
22                     <p class="text-muted mb-4">
23                         Estás a punto de eliminar el siguiente {{ model_name_display|lower }}:
24                     </p>
25
26                     <!-- Información del objeto -->
27                     <div class="card border-danger mb-4">
28                         <div class="card-body">
29                             <h5 class="card-title mb-2">
30                                 {% if objeto.nombre %}
31                                     {{ objeto.nombre }}
32                                 {% elif objeto.username %}
33                                     {{ objeto.username }}
34                                 {% elif objeto.codigo %}
35                                     {{ objeto.codigo }}
36                                 {% else %}
37                                     {{ objeto }}
38                                 {% endif %}
39                             </h5>
40
41                                 {% if objeto.descripcion %}
42                                     <p class="card-text small text-muted mb-0">
43                                         {{ objeto.descripcion|truncatechars:100 }}
44                                     </p>
45                                 {% endif %}
46
47                                 {% if objeto.fecha_creacion %}
48                                     <p class="card-text small text-muted mt-2 mb-0">
```

```


<i class="fas fa-calendar me-1"></i>
    Creado: {{ objeto.fecha_creacion|date:"d/m/Y" }}


```

{% endif %}

```

<!-- Advertencia -->


<i class="fas fa-exclamation-circle me-2"></i>
    Advertencia: Esta acción no se puede deshacer.
    {% if model_name == 'carta' %}
        Se eliminarán también los registros relacionados en inventario y reseñas.
    {% elif model_name == 'usuario' %}
        Se eliminarán todos los pedidos, reseñas y colecciones del usuario.
    {% endif %}


```

```

<!-- Formulario de confirmación -->
<form method="post">
    {% csrf_token %}

    <div class="d-grid gap-2">
        <button type="submit" class="btn btn-danger btn-lg">
            <i class="fas fa-trash me-2"></i>Sí, eliminar
        </button>
        <a href="{% url 'detalle_admin' model_name=model_name obj_id=objeto.id %}"
            class="btn btn-outline-secondary btn-lg">
            <i class="fas fa-times me-2"></i>Cancelar
        </a>
    </div>
</form>

```

```

<!-- Información adicional -->


<p class="mb-1">
        <i class="fas fa-info-circle me-1"></i>
        Esta acción será registrada en el historial del sistema.
    </p>
    {% if model_name == 'inventario' %}
        <p class="mb-0">
            <i class="fas fa-box me-1"></i>
            La carta seguirá disponible en el catálogo, pero sin información de stock.
        </p>
    {% endif %}


```

```

91 |     <i class="fas fa-spinner fa-spin me-2"></i>Eliminando...
92 | 
93 |     {% endif %}
94 | 
95 |     </div>
96 | 
97 | </div>
98 | 
99 </div>
100 
101 {% block extra_js %}
102 <script>
103 document.addEventListener('DOMContentLoaded', function() {
104     // Confirmación adicional al enviar
105     const form = document.querySelector('form');
106     form.addEventListener('submit', function(e) {
107         const submitBtn = this.querySelector('button[type="submit"]');
108         submitBtn.innerHTML = '<i class="fas fa-spinner fa-spin me-2"></i>Eliminando...';
109         submitBtn.disabled = true;
110     });
111 });
112 </script>
113 {% endblock %}
114 {% endblock %}

```

Detalle

```
% extends 'admin/base_admin.html' %}

{%- block content %}

<div class="container-fluid">
    <!-- Encabezado -->
    <div class="d-flex justify-content-between align-items-center mb-4">
        <div>
            <h1 class="h3 mb-0">{{ model_name_display }}</h1>
            <nav aria-label="breadcrumb">
                <ol class="breadcrumb">
                    <li class="breadcrumb-item"><a href="{% url 'admin_dashboard' %}">Dashboard</a></li>
                    <li class="breadcrumb-item"><a href="{% url 'lista_admin' model_name=model_name %}">{{ model_name_display }}</a></li>
                    <li class="breadcrumb-item active">Detalle</li>
                </ol>
            </nav>
        </div>
        <div class="btn-group" role="group">
            <a href="{% url 'editar_admin' model_name=model_name obj_id=objeto.id %}" class="btn btn-warning">
                <i class="fas fa-edit me-2"></i>Editar
            </a>
            <a href="{% url 'eliminar_admin' model_name=model_name obj_id=objeto.id %}" class="btn btn-danger">
                <i class="fas fa-trash me-2"></i>Eliminar
            </a>
            <a href="{% url 'lista_admin' model_name=model_name %}" class="btn btn-secondary">
                <i class="fas fa-arrow-left me-2"></i>Volver
            </a>
        </div>
    </div>
    <!-- Detalle del objeto -->
    <div class="row">
        <div class="col-12">
            <div class="card">
                <div class="card-header bg-primary text-white">
                    <h5 class="mb-0">Información detallada</h5>
                </div>
                <div class="card-body">
                    <div class="table-responsive">
                        <table class="table table-bordered">
                            <tbody>
                                <!-- Información básica de la carta -->
                                <tr>
                                    <th width="30%" class="bg-light">Nombre</th>
                                    <td>{{ objeto.nombre }}</td>
                                </tr>
                                <tr>
                                    <th class="bg-light">Código</th>
                                    <td>{{ objeto.codigo }}</td>
                                </tr>
                                <tr>
                                    <th class="bg-light">Expansión</th>
                                    <td>{{ objeto.expansion.nombre }}</td>
                                </tr>
                                <tr>

```

```
 Número | objeto.numero_en_expansion || Tipo | objeto.tipo if objeto.tipo_secundario else endif |
| HP | objeto.hp|default:"- |
| Rareza | objeto.get_rareza_display |
| Condición | objeto.get_condicion_display |
| Holográfica | if objeto.es_holo span badge bg-success">Sí else span badge bg-secondary">No endif |
| Primera edición | if objeto.primera_edicion span badge bg-success">Sí else span badge bg-secondary">No endif |
| Idioma | objeto.idioma |
| Descripción | objeto.descripcion|linebreaks |
| Coleccionable | if objeto.coleccionable span badge bg-success">Sí else span badge bg-secondary">No endif |

```

```

                </td>
            </tr>
            <tr>
                <th class="bg-light">Popularidad</th>
                <td>{{ objeto.popularidad }}</td>
            </tr>
            <tr>
                <th class="bg-light">Precio estimado</th>
                <td>{{ objeto.precio_estimado }}€</td>
            </tr>
            <tr>
                <th class="bg-light">Imagen frontal</th>
                <td>
                    {% if objeto.imagen_frontal %}
                        
                    {% else %}
                        <span class="text-muted">Sin imagen</span>
                    {% endif %}
                </td>
            </tr>
            <tr>
                <th class="bg-light">Imagen trasera</th>
                <td>
                    {% if objeto.imagen_trasera %}
                        
                    {% else %}
                        <span class="text-muted">Sin imagen</span>
                    {% endif %}
                </td>
            </tr>
        </tbody>
    </table>
</div>

<!-- Relaciones --&gt;
&lt;div class="mt-4"&gt;
    &lt;h5 class="mb-3"&gt;Relaciones&lt;/h5&gt;
    &lt;div class="row"&gt;
        &lt;div class="col-md-6"&gt;
            &lt;div class="card"&gt;
                &lt;div class="card-header bg-light"&gt;
                    &lt;h6 class="mb-0"&gt;Inventario&lt;/h6&gt;
                &lt;/div&gt;
                &lt;div class="card-body"&gt;
                    {% if objeto.inventario %}
                        &lt;table class="table table-sm"&gt;
                            &lt;tr&gt;&lt;th&gt;Stock disponible:&lt;/th&gt;&lt;td&gt;{{
objeto.inventario.cantidad_disponible }}&lt;/td&gt;&lt;/tr&gt;
                            &lt;tr&gt;&lt;th&gt;Stock reservado:&lt;/th&gt;&lt;td&gt;{{
objeto.inventario.cantidad_reservada }}&lt;/td&gt;&lt;/tr&gt;
                            &lt;tr&gt;&lt;th&gt;Precio:&lt;/th&gt;&lt;td&gt;{{ objeto.inventario.precio
}}€&lt;/td&gt;&lt;/tr&gt;
                            &lt;tr&gt;&lt;th&gt;Vendidos total:&lt;/th&gt;&lt;td&gt;{{
objeto.inventario.vendidos_total }}&lt;/td&gt;&lt;/tr&gt;
                            &lt;tr&gt;&lt;th&gt;Disponible:&lt;/th&gt;
                            &lt;td&gt;
                                {% if objeto.inventario.disponible %}
                                    &lt;span class="badge bg-success"&gt;Sí&lt;/span&gt;
                                {% else %}
                                    &lt;span class="badge bg-danger"&gt;No&lt;/span&gt;
                                {% endif %}
                            &lt;/td&gt;
                        &lt;/table&gt;
                    {% endif %}
                &lt;/div&gt;
            &lt;/div&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/div&gt;
</pre>

```

```

                {%
                    else %}
                    <span class="badge bg-danger">No</span>
                {%
                    endif %}
            </td>
        </tr>
    </table>
    {%
        else %}
        <p class="text-muted mb-0">Sin inventario registrado</p>
    {%
        endif %}
    </div>
</div>
<div class="col-md-6">
    <div class="card">
        <div class="card-header bg-light">
            <h6 class="mb-0">Reseñas</h6>
        </div>
        <div class="card-body">
            <p class="mb-2">Total: {{ objeto.resenas.count }}</p>
            {% if objeto.inventario %}
                <p class="mb-0">Valoración promedio: {{
objeto.inventario.valoracion_promedio|floatformat:1 }}/5</p>
            {% else %}
                <p class="mb-0 text-muted">Sin valoraciones</p>
            {% endif %}
        </div>
    </div>
</div>
</div>
</div>

<!-- Fechas --&gt;
&lt;div class="card-footer bg-light"&gt;
    &lt;div class="row"&gt;
        &lt;div class="col-md-6"&gt;
            &lt;small class="text-muted"&gt;
                &lt;i class="fas fa-calendar-plus me-1"&gt;&lt;/i&gt;
                Creado: {{ objeto.fecha_creacion|date:"d/m/Y H:i" }}
            &lt;/small&gt;
        &lt;/div&gt;
        {% if objeto.fecha_actualizacion %}
            &lt;div class="col-md-6 text-end"&gt;
                &lt;small class="text-muted"&gt;
                    &lt;i class="fas fa-calendar-check me-1"&gt;&lt;/i&gt;
                    Última actualización: {{ objeto.fecha_actualizacion|date:"d/m/Y H:i" }}
                &lt;/small&gt;
            &lt;/div&gt;
        {% endif %}
    &lt;/div&gt;
&lt;/div&gt;
&lt;/div&gt;
&lt;/div&gt;
{%
    endblock %}
</pre>

```

formulario

```
{% extends 'admin/base_admin.html' %}

{% block content %}
<div class="container-fluid">
    <!-- Encabezado -->
    <div class="d-flex justify-content-between align-items-center mb-4">
        <div>
            <h1 class="h3 mb-0">{{ model_name_display }}</h1>
            <nav aria-label="breadcrumb">
                <ol class="breadcrumb">
                    <li class="breadcrumb-item"><a href="{% url 'admin_dashboard' %}">Dashboard</a></li>
                    <li class="breadcrumb-item"><a href="{% url 'lista_admin' model_name=model_name %}">{{ model_name_display }}</a></li>
                    <li class="breadcrumb-item active">{{ action|title }}</li>
                </ol>
            </nav>
        </div>
        <div>
            <a href="{% url 'lista_admin' model_name=model_name %}" class="btn btn-secondary">
                <i class="fas fa-arrow-left me-2"></i>Cancelar
            </a>
        </div>
    </div>
</div>

<!-- Formulario -->
<div class="row">
    <div class="col-12">
        <div class="card">
            <div class="card-header bg-primary text-white">
                <h5 class="mb-0">
                    <i class="fas fa-{{ if action == 'crear' }}plus{{ else }}edit{{ endif }} me-2"></i>
                    {{ action|title }} {{ model_name_display }}
                </h5>
            </div>
            <div class="card-body">
                <form method="post" enctype="multipart/form-data">
                    {% csrf_token %}

                    <!-- DEBUG: Verificar si hay form -->
                    {% if not form %}
                    <div class="alert alert-danger">
                        ERROR: No se recibió el formulario. Verifica que la vista esté pasando
el contexto correctamente.
                    </div>
                    {% endif %}

                    <!-- MOSTRAR ERRORES GENERALES DEL FORMULARIO -->
                    {% if form.non_field_errors %}
                    <div class="alert alert-danger">
                        {% for error in form.non_field_errors %}
                        <p class="mb-1">{{ error }}</p>
                        {% endfor %}
                    </div>
                    {% endif %}
                </form>
            </div>
        </div>
    </div>
</div>
```

```

<!-- Usar el formulario de Django COMPLETO -->
{%
    if form %
        <div class="row">
            {%
                for field in form.visible_fields %
                    <div class="col-md-6 mb-3">
                        <label for="{{ field.id_for_label }}" class="form-label">
                            {{ field.label }}
                            {%
                                if field.field.required %
                                    <span class="text-danger">*</span>
                                {%
                                    endif %
                                }
                            </label>

                            <!-- USAR {{ field }} para que Django maneje todo -->
                            {{ field }}

                            {%
                                if field.help_text %
                                    <div class="form-text">{{ field.help_text }}</div>
                                {%
                                    endif %
                                }
                            }

                            {%
                                if field.errors %
                                    <div class="text-danger small mt-1">
                                        {%
                                            for error in field.errors %
                                                <div>{{ error }}</div>
                                                {%
                                                    endfor %
                                                }
                                            </div>
                                            {%
                                                endif %
                                            }
                                        </div>
                                        {%
                                            endfor %
                                        }
                            }

                            <!-- Campos ocultos -->
                            {%
                                for hidden in form.hidden_fields %
                                    {{ hidden }}
                                {%
                                    endfor %
                                }
                            </div>
                            {%
                                else %
                            <!-- FORMULARIO MANUAL como respaldo -->
                            <div class="alert alert-warning">
                                Usando formulario manual. Esto podría no guardar correctamente.
                            </div>

                            {%
                                if model_name == 'carta' %
                                    <!-- Formulario manual para Carta -->
                                    <div class="row">
                                        <div class="col-md-6 mb-3">
                                            <label class="form-label">Código <span
class="text-danger">*</span></label>
                                                <input type="text" name="codigo" class="form-control"
                                                       value="{{ objeto.codigo|default:'' }}" required>
                                            </div>
                                            <div class="col-md-6 mb-3">
                                                <label class="form-label">Nombre <span
class="text-danger">*</span></label>
                                                <input type="text" name="nombre" class="form-control"
                                                       value="{{ objeto.nombre|default:'' }}" required>
                                            </div>
                                            <div class="col-md-6 mb-3">
                                                <label class="form-label">Número en Expansión <span
class="text-danger">*</span></label>
                                                <input type="number" name="numero_en_expansion" class="form-control"
                                                       value="{{ objeto.numero_en_expansion|default:'' }}" required>
                                            </div>
                                    </div>
                            
```

```

        </div>
        <div class="col-md-6 mb-3">
            <label class="form-label">Expansión <span
class="text-danger">*</span></label>
            <select name="expansion" class="form-control" required>
                <option value="">Seleccionar expansión</option>
                {%
                    for expansion in expansions %
                %}
                <option value="{{ expansion.id }}"
                    {% if objeto.expansion.id == expansion.id %}selected{%
                endif %}>
                    {{ expansion.nombre }}
                </option>
                {%
                    endfor %
                }
            </select>
        </div>
        <div class="col-md-6 mb-3">
            <label class="form-label">Tipo</label>
            <select name="tipo" class="form-control">
                <option value="">Seleccionar tipo</option>
                <option value="Normal" {% if objeto.tipo == 'Normal' %}selected{%
                endif %}>Normal</option>
                <option value="Fuego" {% if objeto.tipo == 'Fuego' %}selected{%
                endif %}>Fuego</option>
                <option value="Agua" {% if objeto.tipo == 'Agua' %}selected{%
                endif %}>Agua</option>
                <option value="Planta" {% if objeto.tipo == 'Planta' %}selected{%
                endif %}>Planta</option>
                <option value="Eléctrico" {% if objeto.tipo == 'Eléctrico' %}selected{%
                endif %}>Eléctrico</option>
                <option value="Hielo" {% if objeto.tipo == 'Hielo' %}selected{%
                endif %}>Hielo</option>
                <option value="Lucha" {% if objeto.tipo == 'Lucha' %}selected{%
                endif %}>Lucha</option>
                <option value="Veneno" {% if objeto.tipo == 'Veneno' %}selected{%
                endif %}>Veneno</option>
                <option value="Tierra" {% if objeto.tipo == 'Tierra' %}selected{%
                endif %}>Tierra</option>
                <option value="Volador" {% if objeto.tipo == 'Volador' %}selected{%
                endif %}>Volador</option>
                <option value="Psíquico" {% if objeto.tipo == 'Psíquico' %}selected{%
                endif %}>Psíquico</option>
                <option value="Bicho" {% if objeto.tipo == 'Bicho' %}selected{%
                endif %}>Bicho</option>
                <option value="Roca" {% if objeto.tipo == 'Roca' %}selected{%
                endif %}>Roca</option>
                <option value="Fantasma" {% if objeto.tipo == 'Fantasma' %}selected{%
                endif %}>Fantasma</option>
                <option value="Dragón" {% if objeto.tipo == 'Dragón' %}selected{%
                endif %}>Dragón</option>
                <option value="Siniestro" {% if objeto.tipo == 'Siniestro' %}selected{%
                endif %}>Siniestro</option>
                <option value="Acero" {% if objeto.tipo == 'Acero' %}selected{%
                endif %}>Acero</option>
                <option value="Hada" {% if objeto.tipo == 'Hada' %}selected{%
                endif %}>Hada</option>
                <option value="Incoloro" {% if objeto.tipo == 'Incoloro' %}selected{%
                endif %}>Incoloro</option>
            </select>
        </div>
        <div class="col-md-6 mb-3">

```

```

        <label class="form-label">Rareza <span
class="text-danger">*</span></label>
        <select name="rareza" class="form-control" required>
            <option value="">Seleccionar rareza</option>
            <option value="COMUN" {& if objeto.rareza == 'COMUN'
%}selected{& endif %}>Común</option>
            <option value="INFREC" {& if objeto.rareza == 'INFREC'
%}selected{& endif %}>Infrecuente</option>
            <option value="RARA" {& if objeto.rareza == 'RARA' %}selected{%
endif %}>Rara</option>
            <option value="RARA_HOLO" {& if objeto.rareza == 'RARA_HOLO'
%}selected{& endif %}>Rara Holo</option>
            <option value="RARA_LUM" {& if objeto.rareza == 'RARA_LUM'
%}selected{& endif %}>Rara Luminosa</option>
            <option value="ULTRA" {& if objeto.rareza == 'ULTRA'
%}selected{& endif %}>Rara Ultra</option>
            <option value="SECRETA" {& if objeto.rareza == 'SECRETA'
%}selected{& endif %}>Rara Secreta</option>
            <option value="EX" {& if objeto.rareza == 'EX' %}selected{%
endif %}>Rara Holo EX</option>
            <option value="GX" {& if objeto.rareza == 'GX' %}selected{%
endif %}>Rara GX</option>
            <option value="V" {& if objeto.rareza == 'V' %}selected{& endif
%}>Rara Holo V</option>
            <option value="VMAX" {& if objeto.rareza == 'VMAX' %}selected{%
endif %}>Rara Holo VMAX</option>
            <option value="PROMO" {& if objeto.rareza == 'PROMO'
%}selected{& endif %}>Promo</option>
        </select>
    </div>
    <div class="col-md-12 mb-3">
        <label class="form-label">Descripción <span
class="text-danger">*</span></label>
        <textarea name="descripcion" class="form-control" rows="3"
required>{{ objeto.descripcion|default: '' }}</textarea>
    </div>
    <div class="col-md-6 mb-3">
        <label class="form-label">Imagen frontal</label>
        <input type="file" name="imagen_frontal" class="form-control"
accept="image/*">
        {& if objeto.imagen_frontal %
            <div class="mt-2">
                
                <div class="form-check mt-2">
                    <input type="checkbox" name="imagen_frontal-clear"
id="imagen_frontal-clear" class="form-check-input">
                    <label for="imagen_frontal-clear"
class="form-check-label">Eliminar imagen actual</label>
                </div>
            </div>
        {& endif %
    </div>
    <div class="col-md-6 mb-3">
        <label class="form-label">Imagen trasera</label>
        <input type="file" name="imagen_trasera" class="form-control"
accept="image/*">
        {& if objeto.imagen_trasera %
            <div class="mt-2">

```

```

    <div class="form-check mt-2">
        <input type="checkbox" name="imagen_trasera-clear"
id="imagen_trasera-clear" class="form-check-input">
            <label for="imagen_trasera-clear"
class="form-check-label">Eliminar imagen actual</label>
        </div>
    </div>
    {% endif %}
</div>
<div class="col-md-3 mb-3">
    <div class="form-check">
        <input type="checkbox" name="es_holo" id="es_holo"
class="form-check-input">
            {% if objeto.es_holo %}checked{% endif %}
        <label for="es_holo"
class="form-check-label">Holográfica</label>
    </div>
</div>
<div class="col-md-3 mb-3">
    <div class="form-check">
        <input type="checkbox" name="primera_edicion"
id="primera_edicion" class="form-check-input">
            {% if objeto.primera_edicion %}checked{% endif %}
        <label for="primera_edicion" class="form-check-label">Primera
edición</label>
    </div>
</div>
<div class="col-md-3 mb-3">
    <div class="form-check">
        <input type="checkbox" name="coleccionable" id="coleccionable"
class="form-check-input">
            {% if objeto.coleccionable %}checked{% endif %}
        <label for="coleccionable"
class="form-check-label">Coleccionable</label>
    </div>
</div>
    {% endif %}
    {% endif %}

    <!-- Botones -->
    <div class="mt-4">
        <div class="d-flex justify-content-end gap-2">
            <a href="{% url 'lista_admin' model_name=model_name %}" class="btn
btn-secondary">
                Cancelar
            </a>
            <button type="submit" class="btn btn-primary">
                <i class="fas fa-save me-2"></i>
                {% if action == 'crear' %}Crear{% else %}Guardar cambios{% endif
%}
            </button>
        </div>
    </div>
</form>
</div>
</div>
</div>
```

```

        </div>
    </div>

{%- block extra_js %}

<script>
document.addEventListener('DOMContentLoaded', function() {
    // Añadir clases Bootstrap automáticamente a los campos generados por Django
    const formElements = document.querySelectorAll('form input, form select, form textarea');

    formElements.forEach(element => {
        if (element.type === 'checkbox' || element.type === 'radio') {
            if (!element.classList.contains('form-check-input')) {
                element.classList.add('form-check-input');
            }
        } else if (element.tagName === 'SELECT') {
            if (!element.classList.contains('form-control')) {
                element.classList.add('form-control');
            }
        } else if (element.tagName === 'TEXTAREA') {
            if (!element.classList.contains('form-control')) {
                element.classList.add('form-control');
            }
        } else if (!element.classList.contains('form-control')) {
            element.classList.add('form-control');
        }
    });
});

// Validación de formulario
const form = document.querySelector('form');
form.addEventListener('submit', function(e) {
    let isValid = true;
    const requiredFields = form.querySelectorAll('[required]');

    requiredFields.forEach(field => {
        if (field.type === 'checkbox') {
            if (!field.checked) {
                isValid = false;
                showError(field, 'Este campo es obligatorio');
            }
        } else if (field.type === 'file') {
            // Para archivos, solo validar si estamos creando
            const isEdit = '{{ action }}' === 'editar';
            if (!field.value && !isEdit && field.required) {
                isValid = false;
                showError(field, 'Este archivo es obligatorio');
            }
        } else if (!field.value.trim()) {
            isValid = false;
            showError(field, 'Este campo es obligatorio');
        } else {
            removeError(field);
        }
    });

    if (!isValid) {
        e.preventDefault();
        showNotification('error', 'Por favor, completa todos los campos obligatorios');
    }
});

```

```
function showError(field, message) {
    field.classList.add('is-invalid');
    let errorDiv = field.parentNode.querySelector('.invalid-feedback');
    if (!errorDiv) {
        errorDiv = document.createElement('div');
        errorDiv.className = 'invalid-feedback';
        errorDiv.textContent = message;
        field.parentNode.appendChild(errorDiv);
    }
}

function removeError(field) {
    field.classList.remove('is-invalid');
    field.classList.add('is-valid');
    const errorDiv = field.parentNode.querySelector('.invalid-feedback');
    if (errorDiv) {
        errorDiv.remove();
    }
}

function showNotification(type, message) {
    // Eliminar notificaciones existentes
    const existingNotifications = document.querySelectorAll('.alert.position-fixed');
    existingNotifications.forEach(notification => notification.remove());

    const notification = document.createElement('div');
    notification.className = `alert alert-${type} alert-dismissible fade show position-fixed`;
    notification.style.cssText = 'top: 20px; right: 20px; z-index: 9999; min-width: 300px;';
    notification.innerHTML =
        `${message}
        <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
    `;
    document.body.appendChild(notification);

    setTimeout(() => {
        notification.remove();
    }, 5000);
}
});
```

</script>

{% endblock %}

{% endblock %}

Lista

```
{% extends 'admin/base_admin.html' %}

{% load admin_tags %} <!-- Añade esta linea -->

{% block content %}


<div class="d-flex justify-content-between align-items-center mb-4">
        <h1 class="h3 mb-0">{{ model_name_display }}


```

```

        {%
        endif %}
        {%
        endwith %}
    </td>
    {%
    endfor %}
    <td>
        <div class="btn-group" role="group">
            <a href="{% url 'detalle_admin' model_name=model_name
obj_id=objeto.id %}">
                class="btn btn-sm btn-info">
                    <i class="fas fa-eye"></i>
                </a>
            <a href="{% url 'editar_admin' model_name=model_name
obj_id=objeto.id %}">
                class="btn btn-sm btn-warning">
                    <i class="fas fa-edit"></i>
                </a>
            <a href="{% url 'eliminar_admin' model_name=model_name
obj_id=objeto.id %}">
                class="btn btn-sm btn-danger">
                    <i class="fas fa-trash"></i>
                </a>
            </div>
        </td>
    </tr>
    {%
    empty %}
    <tr>
        <td colspan="{{ campos|length|add:1 }}" class="text-center">
            No se encontraron resultados
        </td>
    </tr>
    {%
    endfor %}
</tbody>
</table>
</div>

<!-- Paginación --&gt;
{%
if page_obj.paginator.num_pages &gt; 1 %}
&lt;nav aria-label="Page navigation" class="mt-4"&gt;
    &lt;ul class="pagination justify-content-center"&gt;
        {%
        if page_obj.has_previous %}
            &lt;li class="page-item"&gt;
                &lt;a class="page-link" href="?page=1&amp;q={{ query }}&gt;Primera&lt;/a&gt;
            &lt;/li&gt;
            &lt;li class="page-item"&gt;
                &lt;a class="page-link" href="?page={{ page_obj.previous_page_number }}&amp;q={{ query }}&gt;
                    Anterior
                &lt;/a&gt;
            &lt;/li&gt;
        {%
        endif %}
        {%
        for num in page_obj.paginator.page_range %}
            {%
            if page_obj.number == num %}
                &lt;li class="page-item active"&gt;
                    &lt;a class="page-link" href="?page={{ num }}&amp;q={{ query }}&gt;{{ num }}&lt;/a&gt;
                &lt;/li&gt;
            {%
            elif num &gt; page_obj.number|add:'-3' and num &lt; page_obj.number|add:'3' %}
                &lt;li class="page-item"&gt;
                    &lt;a class="page-link" href="?page={{ num }}&amp;q={{ query }}&gt;{{ num }}&lt;/a&gt;
                &lt;/li&gt;
            {%
            endif %}
        {%
        endfor %}
    &lt;/ul&gt;
&lt;/nav&gt;
{%
endif %}
</pre>

```

```
        {%- endif %}
    {%- endfor %}

    {%- if page_obj.has_next %}
- >
        Siguiente
    </a>
- >
        Última
    </a>

{%- endif %}
</ul>
</nav>
{%- endif %}
</div>
</div>
</div>
{%- endblock %}
```

Logins

login

```
{% extends 'base.html' %}  
{% load crispy_forms_tags %}  
  
{% block title %}Iniciar Sesión - Pokemon TCG{% endblock %}  
  
{% block content %}  
<div class="container py-5">  
    <div class="row justify-content-center">  
        <div class="col-md-6 col-lg-5">  
            <div class="card shadow-lg border-0">  
                <div class="card-header bg-primary text-white text-center py-4">  
                    <h2 class="mb-0">  
                        <i class="fas fa-sign-in-alt me-2"></i>Iniciar Sesión  
                    </h2>  
                </div>  
                <div class="card-body p-4 p-md-5">  
                    <form method="post" novalidate>  
                        {% csrf_token %}  
  
                        <div class="mb-4">  
                            <label for="username" class="form-label fw-bold">  
                                <i class="fas fa-user me-2"></i>Usuario  
                            </label>  
                            <input type="text"  
                                name="username"  
                                id="username"  
                                class="form-control form-control-lg"  
                                placeholder="Nombre de usuario"  
                                required  
                                autofocus>  
                        </div>  
  
                        <div class="mb-4">  
                            <label for="password" class="form-label fw-bold">  
                                <i class="fas fa-lock me-2"></i>Contraseña  
                            </label>  
                            <input type="password"  
                                name="password"  
                                id="password"  
                                class="form-control form-control-lg"  
                                placeholder="Contraseña"  
                                required>  
                            <div class="form-text">  
                                <a href="#" class="text-decoration-none">¿Olvidaste tu  
contraseña?</a>  
                            </div>  
                        </div>  
  
                        <div class="mb-4 form-check">  
                            <input type="checkbox" class="form-check-input" id="remember"  
name="remember">  
                            <label class="form-check-label" for="remember">  
                                Recordar sesión  
                            </label>  
                        </div>  
                </div>  
            </div>  
        </div>  
    </div>  
</div>
```

```
<div class="d-grid gap-2 mb-4">
    <button type="submit" class="btn btn-pokemon btn-lg">
        <i class="fas fa-sign-in-alt me-2"></i>Iniciar Sesión
    </button>
</div>

<div class="text-center">
    <p class="mb-0">¿No tienes cuenta?</p>
    <a href="{% url 'register' %}" class="btn btn-outline-primary btn-lg mt-2">
        <i class="fas fa-user-plus me-2"></i>Crear Cuenta
    </a>
</div>
</form>

{% if form.errors %}
<div class="alert alert-danger mt-4">
    <strong>Error:</strong> Usuario o contraseña incorrectos.
</div>
{% endif %}

<!-- Demo credentials -->
<div class="card mt-4 border-info">
    <div class="card-header bg-info text-white">
        <i class="fas fa-info-circle me-2"></i>Credenciales de prueba
    </div>
    <div class="card-body">
        <small class="text-muted">
            <strong>Usuario admin:</strong> admin / admin123<br>
            <strong>Usuario cliente:</strong> demo / demodemo
        </small>
    </div>
</div>
</div>
</div>
</div>
<% endblock %}
```

registro

```
{% extends 'base.html' %}  
{% load crispy_forms_tags %}  
  
{% block title %}Registrarse - Pokemon TCG{% endblock %}  
  
{% block content %}  
<div class="container py-5">  
    <div class="row justify-content-center">  
        <div class="col-md-8 col-lg-6">  
            <div class="card shadow-lg border-0">  
                <div class="card-header bg-primary text-white text-center py-4">  
                    <h2 class="mb-0">  
                        <i class="fas fa-user-plus me-2"></i>Crear Cuenta  
                    </h2>  
                    <p class="mb-0 mt-2">¡Únete a la comunidad Pokémon!</p>  
                </div>  
                <div class="card-body p-4 p-md-5">  
                    <form method="post" novalidate>  
                        {% csrf_token %}  
  
                        <div class="row">  
                            <div class="col-md-6 mb-3">  
                                <label for="username" class="form-label fw-bold">  
                                    <i class="fas fa-user me-2"></i>Nombre de Usuario  
                                </label>  
                                <input type="text"  
                                       name="username"  
                                       id="username"  
                                       class="form-control"  
                                       placeholder="Ej: entrenador_ash"  
                                       required  
                                       value="{{ form.username.value|default:' ' }}">  
                                {% if form.username.errors %}  
                                    <div class="text-danger small mt-1">  
                                        {{ form.username.errors.0 }}  
                                    </div>  
                                {% endif %}  
                            </div>  
  
                            <div class="col-md-6 mb-3">  
                                <label for="email" class="form-label fw-bold">  
                                    <i class="fas fa-envelope me-2"></i>Correo Electrónico  
                                </label>  
                                <input type="email"  
                                       name="email"  
                                       id="email"  
                                       class="form-control"  
                                       placeholder="ejemplo@email.com"  
                                       required  
                                       value="{{ form.email.value|default:' ' }}">  
                                {% if form.email.errors %}  
                                    <div class="text-danger small mt-1">  
                                        {{ form.email.errors.0 }}  
                                    </div>  
                                {% endif %}  
                            </div>  
                        </div>  
                    </form>  
                </div>  
            </div>  
        </div>  
    </div>  
</div>
```

```

<div class="row">
    <div class="col-md-6 mb-3">
        <label for="first_name" class="form-label fw-bold">
            <i class="fas fa-id-card me-2"></i>Nombre
        </label>
        <input type="text"
            name="first_name"
            id="first_name"
            class="form-control"
            placeholder="Ej: Ash"
            required
            value="{{ form.first_name.value|default:'' }}">
    </div>

    <div class="col-md-6 mb-3">
        <label for="last_name" class="form-label fw-bold">
            <i class="fas fa-id-card me-2"></i>Apellidos
        </label>
        <input type="text"
            name="last_name"
            id="last_name"
            class="form-control"
            placeholder="Ej: Ketchum"
            required
            value="{{ form.last_name.value|default:'' }}">
    </div>
</div>

<div class="row">
    <div class="col-md-6 mb-3">
        <label for="password1" class="form-label fw-bold">
            <i class="fas fa-key me-2"></i>Contraseña
        </label>
        <input type="password"
            name="password1"
            id="password1"
            class="form-control"
            placeholder="Minimo 8 caracteres"
            required>
        <div class="form-text">
            <small>La contraseña debe contener al menos 8
            caracteres.</small>
        </div>
        {% if form.password1.errors %}
        <div class="text-danger small mt-1">
            {{ form.password1.errors.0 }}
        </div>
        {% endif %}
    </div>

    <div class="col-md-6 mb-3">
        <label for="password2" class="form-label fw-bold">
            <i class="fas fa-key me-2"></i>Confirmar Contraseña
        </label>
        <input type="password"
            name="password2"
            id="password2"
            class="form-control"
            placeholder="Repite la contraseña"
            required>
    </div>
</div>

```

```

        {% if form.password2.errors %}
            <div class="text-danger small mt-1">
                {{ form.password2.errors.0 }}
            </div>
        {% endif %}
    </div>

    <div class="mb-4 form-check">
        <input type="checkbox" class="form-check-input" id="terms" name="terms"
required>
        <label class="form-check-label" for="terms">
            Acepto los
            <a href="#" class="text-decoration-none">Términos y Condiciones</a>
            y la
            <a href="#" class="text-decoration-none">Política de Privacidad</a>
        </label>
    </div>

    <div class="d-grid gap-2">
        <button type="submit" class="btn btn-pokemon btn-lg">
            <i class="fas fa-user-plus me-2"></i>Registrarse
        </button>
        <a href="{% url 'login' %}" class="btn btn-outline-secondary btn-lg">
            <i class="fas fa-sign-in-alt me-2"></i>Ya tengo cuenta
        </a>
    </div>
</form>

<div class="text-center mt-4">
    <p class="mb-2">Regístrate y obtén:</p>
    <div class="row">
        <div class="col-md-4">
            <div class="text-center">
                <i class="fas fa-shopping-cart fa-2x text-primary mb-2"></i>
                <p class="small mb-0">Carrito persistente</p>
            </div>
        </div>
        <div class="col-md-4">
            <div class="text-center">
                <i class="fas fa-heart fa-2x text-danger mb-2"></i>
                <p class="small mb-0">Wishlist personal</p>
            </div>
        </div>
        <div class="col-md-4">
            <div class="text-center">
                <i class="fas fa-history fa-2x text-success mb-2"></i>
                <p class="small mb-0">Historial de pedidos</p>
            </div>
        </div>
    </div>
</div>
</div>
</div>
<% block extra_js %>
<script>
```

```

document.addEventListener('DOMContentLoaded', function() {
    // Validación de contraseñas en tiempo real
    const password1 = document.getElementById('password1');
    const password2 = document.getElementById('password2');

    function validatePasswords() {
        if (password1.value && password2.value) {
            if (password1.value !== password2.value) {
                password2.setCustomValidity('Las contraseñas no coinciden');
                password2.classList.add('is-invalid');
            } else {
                password2.setCustomValidity('');
                password2.classList.remove('is-invalid');
                password2.classList.add('is-valid');
            }
        }
    }

    password1.addEventListener('input', validatePasswords);
    password2.addEventListener('input', validatePasswords);
});
</script>
{%
  endblock %}
{%
  endblock %}

```

carrito checkout

```

{% extends 'base.html' %}
{% load crispy_forms_tags %}
{% load custom_filters %}

{% block title %}Checkout - Pokemon TCG{% endblock %}

{% block content %}


1



Carrito



2



Información



3



Pago



4



Confirmación


```

```

<div class="row">
    <!-- Información del envío -->
    <div class="col-lg-8 mb-4">
        <div class="card">
            <div class="card-header bg-primary text-white">
                <h5 class="mb-0">
                    <i class="fas fa-shipping-fast me-2"></i>Información de Envío
                </h5>
            </div>
            <div class="card-body">
                <div class="row">
                    <div class="col-md-6 mb-3">
                        <label for="nombre_completo" class="form-label">Nombre completo *</label>
                        <input type="text"
                            class="form-control"
                            id="nombre_completo"
                            name="nombre_completo"
                            value="{{ carrito.nombre_completo|default:user.get_full_name }}"
                            required>
                    </div>

                    <div class="col-md-6 mb-3">
                        <label for="email" class="form-label">Email *</label>
                        <input type="email"
                            class="form-control"
                            id="email"
                            name="email"
                            value="{{ carrito.email|default:user.email }}"
                            required>
                    </div>
                </div>
            </div>

            <div class="row">
                <div class="col-md-6 mb-3">
                    <label for="telefono" class="form-label">Teléfono *</label>
                    <input type="tel"
                        class="form-control"
                        id="telefono"
                        name="telefono"
                        value="{{ carrito.telefono|default:' ' }}"
                        required>
                </div>

                <div class="col-md-6 mb-3">
                    <label for="codigo_postal" class="form-label">Código Postal *</label>
                    <input type="text"
                        class="form-control"
                        id="codigo_postal"
                        name="codigo_postal"
                        value="{{ carrito.codigo_postal|default:' ' }}"
                        required>
                </div>
            </div>

            <div class="mb-3">
                <label for="direccion" class="form-label">Dirección de envío *</label>
                <textarea class="form-control"
                    id="direccion"
                    name="direccion"
                    rows="3"
                    required>{{ carrito.direccion|default:' ' }}</textarea>
            </div>

            <div class="row">
                <div class="col-md-6 mb-3">
                    <label for="ciudad" class="form-label">Ciudad *</label>
                    <input type="text"
                        class="form-control"
                        id="ciudad"
                        name="ciudad">
                </div>
            </div>
        </div>
    </div>

```

```

        value="{{ carrito.ciudad|default:'' }}"
        required>
    </div>

    <div class="col-md-6 mb-3">
        <label for="provincia" class="form-label">Provincia</label>
        <input type="text"
            class="form-control"
            id="provincia"
            name="provincia"
            value="{{ carrito.provincia|default:'' }}">
    </div>
</div>

<div class="mb-3">
    <label for="pais" class="form-label">País *</label>
    <select class="form-select" id="pais" name="pais" required>
        <option value="España" {% if carrito.pais == 'España' %}selected{% endif %}>España</option>
        <option value="Portugal" {% if carrito.pais == 'Portugal' %}selected{% endif %}>Portugal</option>
        <option value="Francia" {% if carrito.pais == 'Francia' %}selected{% endif %}>Francia</option>
        <option value="Italia" {% if carrito.pais == 'Italia' %}selected{% endif %}>Italia</option>
        <option value="Alemania" {% if carrito.pais == 'Alemania' %}selected{% endif %}>Alemania</option>
        <option value="Otro" {% if carrito.pais == 'Otro' %}selected{% endif %}>Otro</option>
    </select>
</div>

<div class="mb-3">
    <label for="notas" class="form-label">Notas adicionales (opcional)</label>
    <textarea class="form-control"
        id="notas"
        name="notas"
        rows="2">{{ carrito.notas|default:'' }}</textarea>
    <div class="form-text">
        Instrucciones especiales para la entrega, horarios preferidos, etc.
    </div>
</div>

<div class="form-check mb-3">
    <input class="form-check-input" type="checkbox" id="save_address" checked>
    <label class="form-check-label" for="save_address">
        Guardar esta dirección para futuros pedidos
    </label>
</div>
</div>
</div>

<!-- Método de envío --&gt;
&lt;div class="card mt-4"&gt;
    &lt;div class="card-header bg-light"&gt;
        &lt;h5 class="mb-0"&gt;
            &lt;i class="fas fa-truck me-2"&gt;&lt;/i&gt;Método de Envío
        &lt;/h5&gt;
    &lt;/div&gt;
    &lt;div class="card-body"&gt;
        &lt;div class="form-check mb-3"&gt;
            &lt;input class="form-check-input" type="radio" name="metodo_envio" id="envio_estandar"
value="estandar" checked&gt;
            &lt;label class="form-check-label w-100" for="envio_estandar"&gt;
                &lt;div class="d-flex justify-content-between"&gt;
                    &lt;div&gt;
                        &lt;strong&gt;Envío Estándar&lt;/strong&gt;
                        &lt;p class="mb-0 small text-muted"&gt;Entrega en 3-5 días laborables&lt;/p&gt;
                    &lt;/div&gt;
                &lt;div class="text-end"&gt;
</pre>

```

```

        {%
            if carrito.envio_gratis %}
            <span class="text-success">GRATIS</span>
        {% else %}
            <strong>{{ carrito.envio }}€</strong>
        {% endif %}
    </div>
</div>
</label>
</div>

<div class="form-check">
<input class="form-check-input" type="radio" name="metodo_envio" id="envio_express"
value="express">
<label class="form-check-label w-100" for="envio_express">
<div class="d-flex justify-content-between">
<div>
<strong>Envio Express</strong>
<p class="mb-0 small text-muted">Entrega en 24-48 horas</p>
</div>
<div class="text-end">
<strong>
            {% if carrito.envio_gratis %}
                5€
            {% else %}
                {{ carrito.envio|add:5 }}€
            {% endif %}
            </strong>
</div>
</div>
</label>
</div>
</div>
</div>
</div>

<!-- Resumen y pago -->
<div class="col-lg-4">
<div class="card sticky-top" style="top: 20px;">
<div class="card-header bg-primary text-white">
<h5 class="mb-0"><i class="fas fa-shopping-bag me-2"></i>Resumen del Pedido</h5>
</div>
<div class="card-body">
<!-- Productos -->
<div class="mb-3">
<h6 class="mb-2">Productos {{ carrito.cantidad_items }}</h6>
<div class="cart-items">
            {% for item in carrito.items.all %}
                <div class="d-flex justify-content-between align-items-center mb-2">
                    <div class="small">
                        {{ item.cantidad }}x {{ item.carta.nombre|truncatechars:20 }}
                    </div>
                    <div class="small">{{ item.subtotal }}€</div>
                </div>
            {% endfor %}
        </div>
</div>

<!-- Totales -->
<div class="border-top pt-3">
<div class="d-flex justify-content-between mb-2">
<span>Subtotal:</span>
<strong>{{ carrito.subtotal }}€</strong>
</div>
<div class="d-flex justify-content-between mb-2">
<span>Envío:</span>
<strong id="envio-cost">
            {% if carrito.envio_gratis %}
                <span class="text-success">GRATIS</span>
            {% else %}

```

```

        {{ carrito.envio }}€
        {% endif %}
    </strong>
</div>

<div class="d-flex justify-content-between mb-2">
    <span>Impuestos (21%):</span>
    <strong>{{ carrito.impuestos }}€</strong>
</div>

<hr>

<div class="d-flex justify-content-between mb-4">
    <span class="h5 mb-0">Total:</span>
    <span class="h4 text-primary mb-0" id="total-cost">{{ carrito.total }}€</span>
</div>
</div>

<!-- Método de pago -->
<div class="mb-4">
    <h6 class="mb-3">Método de Pago</h6>
    <div class="accordion" id="paymentMethods">
        <!-- Tarjeta -->
        <div class="accordion-item">
            <h2 class="accordion-header">
                <button class="accordion-button" type="button" data-bs-toggle="collapse"
data-bs-target="#collapseTarjeta">
                    <i class="fab fa-cc-visa me-2 text-primary"></i>
                    <i class="fab fa-cc-mastercard me-2 text-danger"></i>
                    Tarjeta de Crédito/Débito
                </button>
            </h2>
            <div id="collapseTarjeta" class="accordion-collapse collapse show"
data-bs-parent="#paymentMethods">
                <div class="accordion-body">
                    <div class="form-check">
                        <input class="form-check-input" type="radio" name="metodo_pago"
id="pago_tarjeta" value="TARJETA" checked>
                        <label class="form-check-label" for="pago_tarjeta">
                            Pagar con tarjeta
                        </label>
                    </div>
                </div>
            </div>
        </div>
    </div>

    <!-- PayPal -->
    <div class="accordion-item">
        <h2 class="accordion-header">
            <button class="accordion-button collapsed" type="button"
data-bs-toggle="collapse" data-bs-target="#collapsePayPal">
                <i class="fab fa-paypal me-2 text-info"></i>
                PayPal
            </button>
        </h2>
        <div id="collapsePayPal" class="accordion-collapse collapse"
data-bs-parent="#paymentMethods">
            <div class="accordion-body">
                <div class="form-check">
                    <input class="form-check-input" type="radio" name="metodo_pago"
id="pago_paypal" value="PAYPAL">
                    <label class="form-check-label" for="pago_paypal">
                        Pagar con PayPal
                    </label>
                </div>
            </div>
        </div>
    </div>

    <!-- Transferencia -->

```

```

        <div class="accordion-item">
            <h2 class="accordion-header">
                <button class="accordion-button collapsed" type="button"
data-bs-toggle="collapse" data-bs-target="#collapseTransferencia">
                    <i class="fas fa-university me-2 text-success"></i>
                    Transferencia Bancaria
                </button>
            </h2>
            <div id="collapseTransferencia" class="accordion-collapse collapse"
data-bs-parent="#paymentMethods">
                <div class="accordion-body">
                    <div class="form-check">
                        <input class="form-check-input" type="radio" name="metodo_pago"
id=" pago_transferencia" value="TRANSFERENCIA">
                        <label class="form-check-label" for=" pago_transferencia">
                            Pagar por transferencia
                        </label>
                    </div>
                </div>
            </div>

        <!-- Efectivo -->
        <div class="accordion-item">
            <h2 class="accordion-header">
                <button class="accordion-button collapsed" type="button"
data-bs-toggle="collapse" data-bs-target="#collapseEfectivo">
                    <i class="fas fa-money-bill-wave me-2 text-success"></i>
                    Efectivo
                </button>
            </h2>
            <div id="collapseEfectivo" class="accordion-collapse collapse"
data-bs-parent="#paymentMethods">
                <div class="accordion-body">
                    <div class="form-check">
                        <input class="form-check-input" type="radio" name="metodo_pago"
id=" pago_efectivo" value="EFFECTIVO">
                        <label class="form-check-label" for=" pago_efectivo">
                            Pagar en efectivo al recibir
                        </label>
                    </div>
                </div>
            </div>
        </div>

        <!-- Términos y condiciones -->
        <div class="mb-4">
            <div class="form-check">
                <input class="form-check-input" type="checkbox" id="accept_terms" required>
                <label class="form-check-label small" for="accept_terms">
                    Acepto los
                    <a href="#" class="text-decoration-none">Términos y Condiciones</a>
                    y la
                    <a href="#" class="text-decoration-none">Política de Privacidad</a>
                </label>
            </div>
        </div>

        <!-- Botón de pago -->
        <div class="d-grid">
            <button type="submit" class="btn btn-pokemon btn-lg">
                <i class="fas fa-lock me-2"></i>Continuar con el Pago
            </button>
        </div>

        <!-- Seguridad -->
        <div class="text-center mt-4">
            <small class="text-muted">

```

```
                <i class="fas fa-shield-alt text-success me-1"></i>
                Pago 100% seguro SSL encriptado
            </small>
        </div>
    </div>
</div>
</div>
</form>
</div>

{%- block extra_css %}
<style>
.steps {
    display: flex;
    justify-content: space-between;
    position: relative;
    margin-bottom: 30px;
}

.steps::before {
    content: '';
    position: absolute;
    top: 20px;
    left: 0;
    right: 0;
    height: 2px;
    background-color: #dee2e6;
    z-index: 1;
}

.step {
    position: relative;
    z-index: 2;
    text-align: center;
    flex: 1;
}

.step-circle {
    width: 40px;
    height: 40px;
    border-radius: 50%;
    background-color: #dee2e6;
    color: #6c757d;
    display: flex;
    align-items: center;
    justify-content: center;
    margin: 0 auto 10px;
    font-weight: bold;
}

.step.active .step-circle {
    background-color: #3B4CCA;
    color: white;
    box-shadow: 0 0 0 5px rgba(59, 76, 202, 0.2);
}

.step.completed .step-circle {
    background-color: #28a745;
    color: white;
}

.step-label {
    font-size: 0.9rem;
    color: #6c757d;
}

.step.active .step-label {
    color: #3B4CCA;
    font-weight: bold;
}
```

```

}

</style>
{%- endblock %}

{%- block extra_js %}
<script>

document.addEventListener('DOMContentLoaded', function() {
    // Cambiar coste de envío
    const envioEstandar = document.getElementById('envio_estandar');
    const envioExpress = document.getElementById('envio_express');
    const envioCost = document.getElementById('envio-cost');
    const totalCost = document.getElementById('total-cost');

    function updateShippingCost() {
        let envio = parseFloat('{{ carrito.envio }}') || 0;

        if (envioExpress.checked) {
            envio += 5;
            envioCost.innerHTML = envio.toFixed(2) + '€';
        } else {
            if (envio === 0) {
                envioCost.innerHTML = '<span class="text-success">GRATIS</span>';
            } else {
                envioCost.innerHTML = envio.toFixed(2) + '€';
            }
        }

        // Recalcular total
        const subtotal = parseFloat('{{ carrito.subtotal }}') || 0;
        const impuestos = parseFloat('{{ carrito.impuestos }}') || 0;
        const total = subtotal + impuestos + envio;

        totalCost.innerHTML = total.toFixed(2) + '€';
    }

    envioEstandar.addEventListener('change', updateShippingCost);
    envioExpress.addEventListener('change', updateShippingCost);

    // Validación del formulario
    const form = document.getElementById('checkoutForm');
    form.addEventListener('submit', function(e) {
        const acceptTerms = document.getElementById('accept_terms');

        if (!acceptTerms.checked) {
            e.preventDefault();
            showNotification('warning', 'Debes aceptar los términos y condiciones');
            acceptTerms.focus();
            return;
        }

        // Validar método de pago seleccionado
        const metodoPago = document.querySelector('input[name="metodo_pago"]:checked');
        if (!metodoPago) {
            e.preventDefault();
            showNotification('warning', 'Debes seleccionar un método de pago');
            return;
        }

        // Mostrar mensaje de procesamiento
        const submitBtn = form.querySelector('button[type="submit"]');
        submitBtn.innerHTML = '<i class="fas fa-spinner fa-spin me-2"></i>Procesando...';
        submitBtn.disabled = true;
    });
}

// Autocompletar código postal
const codigoPostalInput = document.getElementById('codigo_postal');
const ciudadInput = document.getElementById('ciudad');
const provinciaInput = document.getElementById('provincia');

codigoPostalInput.addEventListener('blur', function() {

```

```

const cp = this.value.trim();

if (cp.length === 5 && /^d+$/.test(cp)) {
    // Simulación de búsqueda de ciudad por CP
    const ciudades = {
        '28001': { ciudad: 'Madrid', provincia: 'Madrid' },
        '08001': { ciudad: 'Barcelona', provincia: 'Barcelona' },
        '41001': { ciudad: 'Sevilla', provincia: 'Sevilla' },
        '46001': { ciudad: 'Valencia', provincia: 'Valencia' },
        '48001': { ciudad: 'Bilbao', provincia: 'Vizcaya' },
    };

    if (ciudades[cp]) {
        ciudadInput.value = ciudades[cp].ciudad;
        provinciaInput.value = ciudades[cp].provincia;
    }
}
});

function showNotification(type, message) {
    const notification = document.createElement('div');
    notification.className = `alert alert-${type} alert-dismissible fade show position-fixed`;
    notification.style.cssText = 'top: 20px; right: 20px; z-index: 9999; min-width: 300px;';
    notification.innerHTML = `
        ${message}
        <button type="button" class="btn-close" data-bs-dismiss="alert"></button>
    `;

    document.body.appendChild(notification);

    setTimeout(() => {
        notification.remove();
    }, 5000);
}

});
</script>
{%
    endblock %}
{%
    endblock %}

```

Detalle Pedido

```

{% extends 'base.html' %}

{% block title %}Pedido #{{ pedido.numero_pedido }} - Pokemon TCG{% endblock %}

{% block content %}


# Pedido #{{ pedido.numero_pedido|slice:'8' }}



Realizado el {{ pedido.fecha_pedido|date:"d/m/Y H:i" }}



Volver a pedidos


```



```

        <h6>Entregado</h6>
        <p class="text-muted small mb-0">
            {% if pedido.estado == 'ENTR' %}
                Entregado
            {% else %}
                En camino
            {% endif %}
        </p>
    </div>
</div>

<!-- Estado actual -->
<div class="mt-4">
    <span class="badge estado-{{ pedido.estado|lower }} fs-6">
        {{ pedido.get_estado_display }}
    </span>
    {% if pedido.fecha_entrega_estimada %}
        <p class="mt-2 mb-0">
            <i class="fas fa-calendar-alt me-2"></i>
            Entrega estimada: {{ pedido.fecha_entrega_estimada|date:"d/m/Y" }}
        </p>
    {% endif %}
</div>
</div>

<!-- Productos -->
<div class="card mb-4">
    <div class="card-header bg-light">
        <h5 class="mb-0">Productos ({{ pedido.cantidad_items }})</h5>
    </div>
    <div class="card-body p-0">
        <div class="table-responsive">
            <table class="table table-hover mb-0">
                <thead class="table-light">
                    <tr>
                        <th>Producto</th>
                        <th class="text-center">Precio</th>
                        <th class="text-center">Cantidad</th>
                        <th class="text-end">Subtotal</th>
                    </tr>
                </thead>
                <tbody>
                    {% for item in pedido.items.all %}
                    <tr>
                        <td>
                            <div class="d-flex align-items-center">
                                <div class="flex-shrink-0">
                                    
                                </div>
                                <div class="flex-grow-1 ms-3">
                                    <h6 class="mb-1">
                                        <a href="{% url 'detalle_carta' item.carta.id %}" class="text-decoration-none text-dark">

```

```

                {{ item.carta.nombre }}
            </a>
        </h6>
        <div class="small text-muted">
            <span class="badge badge-tipo tipo-{{ item.carta.tipo|lower }}">
                {{ item.carta.tipo }}
            </span>
            <span class="ms-2">{{ item.carta.expansion.nombre }}</span>
        </div>
    </td>
    <td class="text-center align-middle">
        {{ item.precio_unitario }}€
    </td>
    <td class="text-center align-middle">
        {{ item.cantidad }}
    </td>
    <td class="text-end align-middle">
        <strong>{{ item.subtotal }}€</strong>
    </td>
</tr>
{% endfor %}
</tbody>
</table>
</div>
</div>
</div>

<!-- Acciones --&gt;
{% if pedido.estado == 'ENTR' %}
&lt;div class="card mb-4"&gt;
    &lt;div class="card-header bg-light"&gt;
        &lt;h5 class="mb-0"&gt;¿Cómo fue tu compra?&lt;/h5&gt;
    &lt;/div&gt;
    &lt;div class="card-body"&gt;
        &lt;p class="mb-3"&gt;Valora los productos recibidos y comparte tu experiencia&lt;/p&gt;
        &lt;div class="d-flex gap-2"&gt;
            &lt;a href="#" class="btn btn-outline-primary"&gt;
                &lt;i class="fas fa-star me-2"&gt;&lt;/i&gt;Dejar reseña
            &lt;/a&gt;
            &lt;a href="#" class="btn btn-outline-success"&gt;
                &lt;i class="fas fa-redo me-2"&gt;&lt;/i&gt;Volver a comprar
            &lt;/a&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/div&gt;
{% endif %}
&lt;/div&gt;

<!-- Información de envío y pago --&gt;
&lt;div class="col-lg-4"&gt;
    <!-- Resumen --&gt;
    &lt;div class="card mb-4"&gt;
        &lt;div class="card-header bg-primary text-white"&gt;
            &lt;h5 class="mb-0"&gt;Resumen del pedido&lt;/h5&gt;
        &lt;/div&gt;
        &lt;div class="card-body"&gt;
</pre>

```

```

<div class="mb-3">
  <div class="d-flex justify-content-between mb-2">
    <span>Subtotal:</span>
    <strong>{{ pedido.subtotal }}€</strong>
  </div>

  {%
    if pedido.descuento > 0 %
  }
  <div class="d-flex justify-content-between mb-2 text-success">
    <span>Descuento:</span>
    <strong>-{{ pedido.descuento }}€</strong>
  </div>
  {%
    endif %
  }

  <div class="d-flex justify-content-between mb-2">
    <span>Envío:</span>
    <strong>
      {%
        if pedido.envio == 0 %
      }
      <span class="text-success">GRATIS</span>
      {%
        else %
      }
      {{ pedido.envio }}€
      {%
        endif %
      }
    </strong>
  </div>

  <div class="d-flex justify-content-between mb-2">
    <span>Impuestos (21%):</span>
    <strong>{{ pedido.impuestos }}€</strong>
  </div>

  <hr>

  <div class="d-flex justify-content-between">
    <span class="h5 mb-0">Total:</span>
    <span class="h4 text-primary mb-0">{{ pedido.total }}€</span>
  </div>
</div>

<!-- Información de pago -->
<div class="mt-4">
  <h6>Información de pago</h6>
  <table class="table table-sm">
    <tr>
      <th>Método:</th>
      <td>{{ pedido.get_metodo_pago_display }}</td>
    </tr>
    <tr>
      <th>Estado:</th>
      <td>
        {%
          if pedido.pagado %
        }
        <span class="badge bg-success">Pagado</span>
        {%
          else %
        }
        <span class="badge bg-warning">Pendiente</span>
        {%
          endif %
        }
      </td>
    </tr>
    {%
      if pedido.pagado and pedido.fecha_pago %
    }
    <tr>
      <th>Fecha pago:</th>
      <td>{{ pedido.fecha_pago|date:"d/m/Y H:i" }}</td>
    </tr>
  
```

```

        {%
        endif %}
        {% if pedido.transaccion_id %}
        <tr>
            <th>ID transacción:</th>
            <td><small>{{ pedido.transaccion_id }}</small></td>
        </tr>
        {% endif %}
    </table>
</div>
</div>
</div>

<!-- Información de envío --&gt;
&lt;div class="card mb-4"&gt;
    &lt;div class="card-header bg-light"&gt;
        &lt;h5 class="mb-0"&gt;
            &lt;i class="fas fa-truck me-2"&gt;&lt;/i&gt;Envío
        &lt;/h5&gt;
    &lt;/div&gt;
    &lt;div class="card-body"&gt;
        &lt;address class="mb-0"&gt;
            &lt;strong&gt;{{ pedido.nombre_completo }}&lt;/strong&gt;&lt;br&gt;
            {{ pedido.direccion_envio }}&lt;br&gt;
            {{ pedido.codigo_postal }} {{ pedido.ciudad_envio }}&lt;br&gt;
            {% if pedido.provincia_envio %}{{ pedido.provincia_envio }}&lt;br&gt;{% endif %}
            {{ pedido.pais_envio }}&lt;br&gt;
            &lt;i class="fas fa-phone me-1"&gt;&lt;/i&gt; {{ pedido.telefono }}&lt;br&gt;
            &lt;i class="fas fa-envelope me-1"&gt;&lt;/i&gt; {{ pedido.email }}
        &lt;/address&gt;

        {% if pedido.metodo_envio %}
        &lt;hr class="my-3"&gt;
        &lt;div&gt;
            &lt;strong&gt;Método:&lt;/strong&gt;
            {{ pedido.metodo_envio|title }}
        &lt;/div&gt;
        {% endif %}

        {% if pedido.notas_cliente %}
        &lt;hr class="my-3"&gt;
        &lt;div&gt;
            &lt;strong&gt;Notas:&lt;/strong&gt;&lt;br&gt;
            &lt;small&gt;{{ pedido.notas_cliente }}&lt;/small&gt;
        &lt;/div&gt;
        {% endif %}
    &lt;/div&gt;
&lt;/div&gt;

<!-- Acciones del pedido --&gt;
&lt;div class="card"&gt;
    &lt;div class="card-header bg-light"&gt;
        &lt;h5 class="mb-0"&gt;Acciones&lt;/h5&gt;
    &lt;/div&gt;
    &lt;div class="card-body"&gt;
        &lt;div class="d-grid gap-2"&gt;
            {% if pedido.estado == 'PEND' %}
            &lt;button class="btn btn-outline-danger"
                onclick="if(confirm('¿Cancelar este pedido?')) location.href='#!'&gt;
                &lt;i class="fas fa-times me-2"&gt;&lt;/i&gt;Cancelar pedido
            &lt;/button&gt;
        </pre>

```

```
{% endif %}

    <a href="{% url 'lista_cartas' %}" class="btn btn-pokemon">
        <i class="fas fa-shopping-cart me-2"></i>Seguir comprando
    </a>

    <button class="btn btn-outline-primary" onclick="window.print()">
        <i class="fas fa-print me-2"></i>Imprimir pedido
    </button>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<% block extra_css %>
<style>
.timeline {
    position: relative;
    padding-left: 30px;
}

.timeline::before {
    content: '';
    position: absolute;
    left: 15px;
    top: 0;
    bottom: 0;
    width: 2px;
    background-color: #dee2e6;
}

.timeline-item {
    position: relative;
    margin-bottom: 30px;
}

.timeline-item:last-child {
    margin-bottom: 0;
}

.timeline-point {
    position: absolute;
    left: -30px;
    top: 5px;
    width: 12px;
    height: 12px;
    border-radius: 50%;
    background-color: #dee2e6;
    border: 2px solid #fff;
    box-shadow: 0 0 0 3px #dee2e6;
}

.timeline-item.completed .timeline-point {
    background-color: #28a745;
    box-shadow: 0 0 0 3px rgba(40, 167, 69, 0.2);
}

.timeline-item:not(.completed) .timeline-point {
```

```
background-color: #ffc107;
box-shadow: 0 0 0 3px rgba(255, 193, 7, 0.2);
}

.timeline-content h6 {
    margin-bottom: 5px;
}

@media print {
    .btn, .card-header, .timeline::before {
        display: none !important;
    }

    .card {
        border: none !important;
    }
}
</style>
{% endblock %}
{% endblock %}
```