

ISS-07 — INFOGRAFÍA: SERIALIZERS Y JSON CAMELCASE

StoreLab REST API Architecture • Transición Backend (snake_case) ↔ Cable HTTP (camelCase)

¿QUÉ CONSTRUIMOS EN ISS-07?

- **Django REST Framework (DRF):** Incorporación de serializers para Client, ProductType, Product, ProductSale y Sale.
- **Transparencia de Nombres:** El código Python y la BD conservan snake_case (created_at), mientras que la API emite y recibe camelCase (createdAt).
- **CORS Headers:** Integración de CorsMiddleware para permitir consumo cliente seguro.

CONFIGURACIÓN EN SETTINGS

rest_framework & **corsheaders** en INSTALLED_APPS

CorsMiddleware antecede a SecurityMiddleware

REST_FRAMEWORK Config:

- Renderers: CamelCaseJSONRenderer, CamelCaseBrowsableAPIRenderer
- Parsers: CamelCaseJSONParser, CamelCaseFormParser, CamelCaseMultiPartParser
- AllowAny (Permiso abierto Fase I)

FLUJO DE TRANSFORMACIÓN EN EL BORDE HTTP

[Python / Django ORM] created_at → [DRF Serializer] Python dict → [CamelCaseJSONRenderer] createdAt → [Cable HTTP JSON Payload] {"createdAt": "2026-10-01..."}

El proceso inverso ocurre automáticamente en solicitudes POST/PUT mediante CamelCaseJSONParser.

ClientSerializer

- Normaliza email a minúsculas en validate_email.
- Convierte cadenas vacías ("") de address/phone a None.

ProductSerializer

- product_type_id (escribable).
- product_type_name (solo lectura, source='product_type.name').

SaleSerializer

- Incluye client_name.
- Anida detalles de línea mediante lines = ProductSaleSerializer(many=True, read_only=True).

VERIFICACIÓN, EVIDENCIA Y COMPUERTA GATE

Criterio de Aceptación	Verificación	Evidencia	Resultado
AC-07-01: Cuerpo HTTP con createdAt	Aserción sobre response.content en ClientApiTests	EVI-07-01	PASS
AC-07-02: Email normalizado	Persistencia de email en minúsculas en BD	Test de cliente	PASS
AC-07-03: Nombre del tipo de producto	Presencia de productTypeName sin query extra	Test de producto	PASS