

Descripción de APIs principales – Proyecto Medusa JS con Gráficas para Administrador

UNIVERSIDAD CENTROAMERICANA “JOSÉ SIMEÓN CAÑAS”

Facultad de Ingeniería y Arquitectura

Aplicaciones de código abierto - Ciclo 01/2025



Grupo 06

Integrantes:

Orellana Vividor, Gerardo Andre	00053520	Product Owner
Christian Alejandro Paz Escobar	00132720	Scrum Master
Rodriguez Rodriguez, David Neftali	00218621	Tech Lead
Alas Moscoso, Noe Bladimir	00262020	Developer
Zacatales López, César Adonay	00223021	Developer

Antiguo Cuscatlán, lunes 26 de mayo de 2025

MEDUSAJS V 2.8.2

Índice

1. Introducción.....	3
2. Autenticación.....	3
3. Endpoints personalizados para graficas.....	3
3.1 Obtener estadísticas de ventas.....	3
3.2 Obtener productos más vendidos.....	3
3.3 Obtener nuevos clientes por mes.....	4
4. Códigos de estado HTTP.....	4
5. Consideraciones técnicas.....	4

1. Introducción

Este documento presenta una descripción técnica de las principales APIs disponibles en el backend de MedusaJS personalizado, el cual ha sido extendido para ofrecer datos analíticos al panel de administración. Estas APIs exponen información sobre ventas, productos más vendidos y comportamiento de clientes para su análisis visual.

2. Autenticación

Todas las rutas están protegidas mediante JWT (JSON Web Token). Para acceder a las APIs es necesario:

- Autenticarse con credenciales válidas.
- Incluir el token en la cabecera Authorization:

Authorization: Bearer <token>

3. Endpoints personalizados para graficas

3.1 Obtener estadísticas órdenes por región

- URL: /admin/routes/region_order
- Método: GET
- Descripción: Devuelve datos de órdenes por región.

Respuesta:

```
[
  {
    "region_name": "Mexico",
    "order_count": 10
  },
  {
    "region_name": "Canada",
    "order_count": 20
  },
  {
    "region_name": "Spain",
    "order_count": 31
  },
  {
    "region_name": "South Korea",
    "order_count": 39
  },
  {
    "region_name": "Singapore",
    "order_count": 49
  }
]
```

```

    },
    {
      "region_name": "New Zealand",
      "order_count": 50
    },
    {
      "region_name": "Argentina",
      "order_count": 102
    },
    {
      "region_name": "Chile",
      "order_count": 8
    },
    {
      "region_name": "Norway",
      "order_count": 5
    },
    {
      "region_name": "Nigeria",
      "order_count": 4
    }
  ]

```

3.2 Obtener productos más vendidos

- URL: /admin/routes/client_order
- Método: GET
- Descripción: Lista los pedidos totales por cliente.

Respuesta:

```

[
  {
    "customer": "Juan Pérez",
    "orders": 9
  },
  {
    "customer": "María González",
    "orders": 20
  },
  {
    "customer": "Carlos Martínez",
    "orders": 31
  },
  {

```

```

        "customer": "Ana Rodríguez",
        "orders": 39
    },
    {
        "customer": "Luis Hernández",
        "orders": 49
    },
    {
        "customer": "Sofía López",
        "orders": 50
    },
    {
        "customer": "Pedro Díaz",
        "orders": 1
    },
    {
        "customer": "Patricia Flores",
        "orders": 111
    },
    {
        "customer": "Oscar Valdez",
        "orders": 8
    }
]

```

3.3 Obtener nuevos clientes por mes

- URL: /admin/routes/collection_product
- Método: GET
- Descripción: Devuelve la cantidad de productos por tipo de colección.

Respuesta:

```

[
    {
        "collection_name": "Camisas",
        "product_count": 13
    },
    {
        "collection_name": "Chaquetas",
        "product_count": 17
    },
    {
        "collection_name": "Vestidos",
        "product_count": 10
    }
]

```

```

    },
    {
      "collection_name": "Blusas",
      "product_count": 15
    },
    {
      "collection_name": "Pantalones",
      "product_count": 6
    },
    {
      "collection_name": "Sin colección",
      "product_count": 4
    }
  ]

```

4. Códigos de estado HTTP

Código	Significado
200	OK (respuesta exitosa)
401	No autorizado (sin token o inválido)
500	Error interno del servidor

5. Consideraciones técnicas

- Todas las rutas se encuentran bajo el prefijo /admin/stats/.
- La estructura de respuesta es siempre un objeto JSON con la clave data.
- El backend aplica filtros de seguridad y validación de roles para acceso a estos datos.
- Las rutas están implementadas como controladores separados en el sistema MedusaJS, siguiendo buenas prácticas REST.
- Se pueden aplicar filtros por fecha usando parámetros como ?start_date= y ?end_date= (extensible a futuro).
- Las estadísticas se calculan usando agregaciones SQL optimizadas para rendimiento.
- Se recomienda consumir estas APIs desde el frontend utilizando Axios u otra librería de HTTP en React.

- Todas las respuestas siguen un formato uniforme, ideal para su integración en componentes gráficos.
- En caso de errores, se devuelve un mensaje estructurado con clave message explicando el motivo del fallo.