

# exec\_note\_ESP\_2025102

October 2, 2025

## 1 Dashboard Ejecutivo de Inteligencia de Negocios

Insights automatizados para decisiones basadas en datos

```
[1]: input_file = 'data/comercializadora/comercializadora_transactions.csv' # Input
    ↪ CSV file

config = {
    'project_name': 'comercializadora', # Project name
    'out_dir' : 'outputs', # Output directory

    # Data mapping
    'date_col': 'fecha', # Required
    'product_col': 'producto', # Required
    'description_col': 'glosa', # Optional, set to None if not available
    'revenue_col': 'total', # Required
    'quantity_col': 'cantidad', # Required
    'transaction_col': 'trans_id', # Required
    'cost_col': 'costo', # Optional, set to None if not
    ↪ available
    'customer_col': 'customer_id', # Optional, set to None if not available

    # Analysis settings
    'analysis_date': '2025-03-01', # Or 'current' for today
    'top_products_threshold': 0.2, # Top products threshold (e.g., 0.2 for
    ↪ top 20%)
    'dead_stock_days': 30, # Days of inactivity to consider stock
    ↪ as dead
    'currency_format': 'CLP', # 'USD', 'EUR', 'CLP', etc.
    'language': 'ESP', # 'ENG' or 'ESP'

    # Logging and performance
    'log_level': 'INFO', # 'DEBUG', 'INFO', 'WARNING', 'ERROR',
    ↪ 'CRITICAL'
}

save = 0 # True or 1 to save outputs, False or 0 to just print
```

## 1.1 Configuración y Resumen de Datos

*Cargando sus datos de negocio y configurando parámetros de análisis*

```
[2]: from modules.business_analytics import BusinessAnalyzer
from modules.dashboard import ExecutiveDashboard
from modules.advanced_analytics import AdvancedAnalytics
from modules.reports import *
from modules.utils import *

# Initialize logging from config
from modules.logger import setup_logging
setup_logging(log_level=config.get('log_level', 'INFO'), config=config)

# Helper function for translated filenames (bound to config language)
from modules.translations import create_filename_helper
fn = create_filename_helper(config)

# Initialize modules
analyzer = BusinessAnalyzer(data_source=input_file, config=config)
dashboard = ExecutiveDashboard(analyzer)
advanced = AdvancedAnalytics(analyzer)
```

```
Logging [INFO] to: logs\comercializadora_20251002_173429.log
Data date range: 2024-12-01 to 2025-02-28
Recommended analysis_date: 2025-03-01 or later
```

## 1.2 Resumen Ejecutivo

*Su negocio de un vistazo: métricas clave, acciones críticas e insights principales*

```
[3]: summary = dashboard.create_quick_summary()
print_info(summary, analyzer.out_dir, fn('DASH', 'quick_summary'), save=save)
```

```
=====
RESUMEN DEL DASHBOARD
=====
```

### MÉTRICAS CLAVE:

- Ingresos Totales: \$ 1.194.033.014
- Tasa de Crecimiento: -16.6%
- Transacciones: 49,372

### ACCIONES CRÍTICAS:

- Ingresos decreciendo en 16.6%
  - Revisión urgente de estrategia de ventas necesaria

### INSIGHTS CLAVE:

- Top 20% productos = 65.5% ingresos

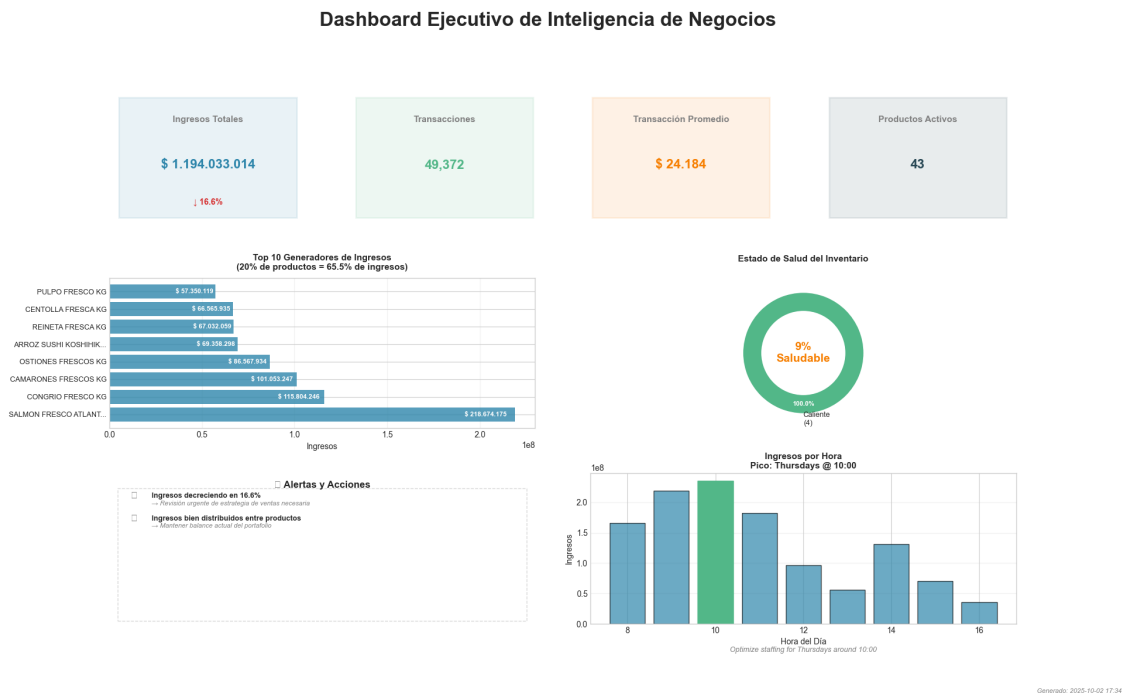
- Salud de Inventario: 9% healthy
- Producto Sin Movimiento: 0 productos

=====

### 1.3 Dashboard Visual de Rendimiento

*Resumen visual completo de ingresos, tendencias y salud operacional*

```
[4]: # Create and display the executive dashboard
fig = dashboard.create_full_dashboard(figsize=(20, 12))
print_fig(fig, dashboard.analyzer.out_dir, fn('DASH', 'executive', 'png'),
save=save)
```



### 1.4 Análisis de Concentración de Ingresos

*Identifique sus productos estrella y exposición al riesgo de portafolio*

```
[5]: pareto = analyzer.get_pareto_insights()
print_info(analyzer.print_pareto(), analyzer.out_dir, fn('BA', 'pareto'),
save=save)
```

INSIGHT PRINCIPAL: ¡Tus 8 productos principales (20.0% del catálogo) generan 65.5% de los ingresos!

Nivel de Riesgo de Concentración: Medium

Top 5 5 Generadores de Ingresos:

1. SALMON FRESCO ATLANTICO KG: \$ 218.674.175
2. CONGRIO FRESCO KG: \$ 115.804.246
3. CAMARONES FRESCOS KG: \$ 101.053.247
4. OSTIONES FRESCOS KG: \$ 86.567.934
5. ARROZ SUSHI KOSHIHIKARI 5KG: \$ 69.358.298

Regla 80/20: El Top 20% = 65.5% de ingresos

## 1.5 Chequeo de Salud de Inventario

*Detecte stock muerto y optimice la inversión en inventario*

```
[6]: inventory = analyzer.get_inventory_health()
print_info(analyzer.print_inventory_health(), analyzer.out_dir, fn('BA', 'inventory'), save=save)
```

Puntuación de Salud de Inventario: 9%

Alerta de Producto Sin Movimiento: 0 productos

## 1.6 Eficiencia Operacional

*Optimice personal y recursos basándose en patrones de demanda pico*

```
[7]: peak_times = analyzer.get_peak_times()
print_info(analyzer.print_peak_times(), analyzer.out_dir, fn('BA', 'peak_times'), save=save)
```

Ventanas de Máximo Rendimiento:

- Mejor Día: Thursdays
- Hora Pico: 10:00
- Día Más Lento: Saturdays

Optimize staffing for Thursdays around 10:00

## 1.7 Alertas y Acciones

*Problemas críticos que requieren atención inmediata y oportunidades estratégicas*

```
[8]: alerts = analyzer.get_alerts()
print_info(analyzer.print_alerts(), analyzer.out_dir, fn('BA', 'alerts'), save=save)
```

ACCIONES CRÍTICAS REQUERIDAS:

Ingresos decreciendo en 16.6%

Impacto: Tendencia negativa del negocio

Acción: Revisión urgente de estrategia de ventas necesaria

INDICADORES DE ÉXITO:

Ingresos bien distribuidos entre productos

Siguiente Paso: Mantener balance actual del portafolio

## 1.8 Exportar Datos de Resumen

*Guarde métricas clave para rastrear tendencias en el tiempo*

```
[9]: # Save executive summary as CSV for easy tracking over time
import pandas as pd

summary_dict = analyzer.get_executive_summary_dict()
summary_df = pd.DataFrame([summary_dict])

if save:
    save_path = os.path.join(analyzer.out_dir, fn('BA', 'executive_summary',
↪ 'csv'))
    os.makedirs(os.path.dirname(save_path), exist_ok=True)
    summary_df.to_csv(save_path, index=False)
    print(f" Executive summary exported to {save_path}")
else:
    print(summary_df.to_string(index=False))
```

Date	Total Revenue	Revenue Growth %	Total Transactions	Top 20% Revenue
2025-03-01	1194033014	-16.586181	49372	
Share	Dead Stock Count	Inventory Health %		
65.52633	0	9.302326		