```
Productos Principales por Ingresos:
      Producto   Ingresos
0    Portátil   7999.92
1    Teléfono   4999.90

Categorías Únicas: 2
Estadísticas de Ingresos:
Mín: 399.80, Máx: 7999.92, Media: 3569.92
```

Aplicación Práctica: Optimizar el inventario enfocándose en productos y categorías de alto ingreso.

12.2. Ejercicio Integrador 2: Revisión de Desempeño de Empleados

Enunciado: Crear un DataFrame de datos de desempeño de empleados con índices personalizados (IDs de empleados). Filtrar empleados con puntajes de desempeño altos (>80) en departamentos específicos (Ventas o Marketing). Calcular el puntaje promedio por departamento e identificar calificaciones únicas de desempeño. Mostrar columnas específicas para los empleados filtrados y sus detalles.

Ejemplo de Entrada:

```
data = {
    'Nombre': ['Alicia', 'Bruno', 'Carlos', 'David'],
    'Departamento': ['Ventas', 'Marketing', 'TI', 'Ventas'],
    'Puntaje': [85, 90, 75, 88],
    'Calificaci n': ['A', 'A', 'B', 'A']
}
df = pd.DataFrame(data, index=['E001', 'E002', 'E003', 'E004'])
```

Salida Esperada:

```
Altos Desempeños en Ventas/Marketing:
      Nombre  Departamento  Puntaje
E001  Alicia      Ventas       85
E002   Bruno   Marketing       90
E004   David      Ventas       88

Puntaje Promedio por Departamento:
Ventas      86.5
Marketing   90.0
TI          75.0
Name: Puntaje, dtype: float64

Calificaciones Únicas: ['A', 'B']
```

Aplicación Práctica: Identificar empleados destacados para promociones y análisis departamental.

12.3. Ejercicio Integrador 3: Patrones de Compra de Clientes

Enunciado: Crear un DataFrame de datos de compras de clientes. Filtrar compras realizadas en el último trimestre (suponiendo que se proporcionan fechas). Calcular el monto total de compras por cliente usando una operación de Serie. Resumir estadísticas de compras y contar categorías únicas de productos. Mostrar los tres principales clientes por monto total de compra.

Ejemplo de Entrada:

```
data = {
    'Cliente': ['C1', 'C2', 'C1', 'C3'],
    'Producto': ['Port til', 'Tel fono', 'Tableta', 'Rat n'],
    'Categor a': ['Electr nica', 'Electr nica', 'Electr nica'
       , 'Accesorios'],
    'Monto': [999.99, 499.99, 299.99, 19.99],
    'Fecha': ['2025-10-01', '2025-09-15', '2025-11-01', '
       2025-08-01']
}
df = pd.DataFrame(data)
```

Salida Esperada:

```
Clientes Principales por Monto Total:
Cliente
C1    1299.98
C2     499.99

Estadísticas de Compras:
Mín: 299.99, Máx: 999.99, Media: 599.99
Categorías Únicas: 1
```

Aplicación Práctica: Dirigir programas de lealtad a clientes de alto valor.

12.4. Ejercicio Integrador 4: Gestión de Inventario de Tiendas

Enunciado: Crear un DataFrame de inventario de tiendas en múltiples ubicaciones. Filtrar artículos con bajo inventario (<10 unidades) en tiendas específicas. Calcular el valor total del inventario por tienda (precio * cantidad). Identificar tipos de productos únicos y sus conteos. Mostrar los artículos con bajo inventario y estadísticas del valor del inventario.

Ejemplo de Entrada:

```
data = {
    'Tienda': ['T1', 'T1', 'T2', 'T2'],
    'Producto': ['Port til', 'Rat n', 'Tel fono', 'Tableta'],
    'Precio': [999.99, 19.99, 499.99, 299.99],
    'Cantidad': [5, 50, 8, 3]
}
df = pd.DataFrame(data)
```

Salida Esperada:

```
Artículos con Bajo Inventario:
   Tienda  Producto  Cantidad
0     T1  Portátil         5
3     T2  Tableta         3

Valor del Inventario por Tienda:
T1    5999.45
T2    4899.93
Name: Valor_Inventario, dtype: float64

Tipos de Productos Únicos: 4
```

Aplicación Práctica: Priorizar el reabastecimiento en tiendas con inventario bajo.

12.5. Ejercicio Integrador 5: Análisis de Rendimiento Académico

Enunciado: Crear un DataFrame de resultados de exámenes de estudiantes en múltiples asignaturas. Filtrar estudiantes con un promedio de puntaje superior a 85. Calcular el puntaje promedio por asignatura usando una operación de Serie. Contar categorías de calificaciones únicas (por ejemplo, A, B, C). Mostrar los detalles de los estudiantes filtrados y estadísticas por asignatura.

Ejemplo de Entrada:

```
data = {
    'Estudiante': ['Alicia', 'Bruno', 'Carlos'],
    'Matem ticas': [90, 85, 78],
    'Ciencias': [88, 92, 80],
    'Historia': [85, 88, 75],
    'Calificaci n': ['A', 'A', 'B']
}
df = pd.DataFrame(data)
```

Salida Esperada:

```
Altos Rendimientos (Promedio > 85):
  Estudiante  Matemáticas  Ciencias  Historia Calificación
0    Alicia           90        88        85            A
1     Bruno           85        92        88            A

Promedios por Asignatura:
Matemáticas    84.33
Ciencias       86.67
Historia       82.67
dtype: float64

Calificaciones Únicas: 2
```

Aplicación Práctica: Identificar estudiantes destacados para becas y planificación curricular.

12.6. Ejercicio Integrador 6: Análisis de Campañas de Marketing

Enunciado: Crear un DataFrame de datos de campañas de marketing. Filtrar campañas con una tasa de respuesta alta (>5 %) lanzadas en 2025. Calcular el costo total por campaña usando una operación de Serie. Resumir estadísticas de costos y contar tipos de campañas únicas. Mostrar las campañas filtradas y sus detalles.

Ejemplo de Entrada:

```
data = {
    'Campa a': ['C1', 'C2', 'C3'],
    'Tipo': ['Correo', 'Redes', 'Correo'],
    'Costo': [5000, 8000, 6000],
    'Tasa_Respuesta': [6.5, 4.2, 5.8],
    'Fecha_Lanzamiento': ['2025-01-15', '2024-12-01', '2025-03-01
      ']
}
df = pd.DataFrame(data)
```

Salida Esperada:

```
Campañas de Alta Respuesta en 2025:
  Campaña   Tipo  Costo  Tasa_Respuesta Fecha_Lanzamiento
0     C1  Correo   5000             6.5      2025-01-15
2     C3  Correo   6000             5.8      2025-03-01

Estadísticas de Costos:
Mín: 5000, Máx: 6000, Media: 5500
Tipos de Campañas Únicas: 1
```

Aplicación Práctica: Evaluar estrategias de marketing efectivas para la asignación de presupuesto.

12.7. Ejercicio Integrador 7: Análisis de Pacientes en Salud

Enunciado: Crear un DataFrame de registros de salud de pacientes. Filtrar pacientes con presión arterial alta (>140) que visitaron en 2025. Calcular la edad promedio por género usando una operación de Serie. Contar categorías de diagnóstico únicas. Mostrar los detalles de los pacientes filtrados y estadísticas resumidas.

Ejemplo de Entrada:

```
data = {
    'Paciente': ['P1', 'P2', 'P3'],
    'Edad': [45, 60, 50],
    'G nero': ['M', 'F', 'M'],
    'PA': [145, 130, 150],
    'Diagn stico': ['Hipertensi n', 'Normal', 'Hipertensi n'],
    'Fecha_Visita': ['2025-02-01', '2024-11-15', '2025-01-10']
}
df = pd.DataFrame(data)
```

Salida Esperada:

```
Pacientes con PA Alta en 2025:
  Paciente  Edad Género  PA   Diagnóstico Fecha_Visita
0      P1    45      M  145  Hipertensión  2025-02-01
2      P3    50      M  150  Hipertensión  2025-01-10

Edad Promedio por Género:
F    60.0
M    47.5
Name: Edad, dtype: float64

Diagnósticos Únicos: 2
```

Aplicación Práctica: Identificar pacientes en riesgo para intervenciones de salud dirigidas.

12.8. Ejercicio Integrador 8: Optimización de Entregas Logísticas

Enunciado: Crear un DataFrame de datos de entregas en varias regiones. Filtrar entregas con retrasos (>2 días) en regiones específicas (Norte, Este). Calcular el costo promedio de entrega por región usando una operación de Serie. Contar tipos de vehículos únicos utilizados. Mostrar las entregas retrasadas y estadísticas de costos.

Ejemplo de Entrada:

```
data = {
    'ID_Entrega': ['E1', 'E2', 'E3'],
    'Regi n': ['Norte', 'Este', 'Sur'],
    'D as_Retraso': [3, 1, 4],
    'Costo': [200, 150, 250],
    'Veh culo': ['Cami n', 'Furgoneta', 'Cami n']
}
df = pd.DataFrame(data)
```

Salida Esperada:

```
Entregas Retrasadas en Norte/Este:
  ID_Entrega Región  Días_Retraso  Costo  Vehículo
0         E1  Norte             3    200   Camión

Costo Promedio por Región:
Este     150.0
Norte    200.0
Sur      250.0
Name: Costo, dtype: float64

Vehículos Únicos: 2
```

Aplicación Práctica: Optimizar operaciones logísticas abordando retrasos y costos.

12.9. Ejercicio Integrador 9: Análisis de Satisfacción del Cliente

Enunciado: Crear un DataFrame de datos de satisfacción del cliente recopilados a través de encuestas. Filtrar clientes con calificaciones altas (>4) que hayan comprado en 2025. Calcular el promedio de calificaciones por categoría de producto usando una operación de Serie. Contar los tipos de comentarios únicos (por ejemplo, Positivo, Negativo). Mostrar los detalles de los clientes filtrados y estadísticas resumidas.

Ejemplo de Entrada:

```
data = {
    'Cliente': ['C1', 'C2', 'C3'],
    'Producto': ['Port til', 'Tel fono', 'Tableta'],
    'Categor a': ['Electr nica', 'Electr nica', 'Electr nica'
      ],
    'Calificaci n': [4.5, 3.8, 4.2],
    'Comentario': ['Positivo', 'Negativo', 'Positivo'],
    'Fecha_Compra': ['2025-01-10', '2024-12-15', '2025-02-01']
}
df = pd.DataFrame(data)
```

Salida Esperada:

```
Clientes con Alta Satisfacción en 2025:
  Cliente  Producto  Categoría  Calificación Comentario Fecha_Compra
0     C1  Portátil Electrónica          4.5  Positivo  2025-01-10
2     C3  Tableta Electrónica          4.2  Positivo  2025-02-01

Calificación Promedio por Categoría:
Electrónica    4.35
Name: Calificación, dtype: float64

Comentarios Únicos: 2
```

Aplicación Práctica: Mejorar la experiencia del cliente enfocándose en productos con alta satisfacción.

12.10. Ejercicio Integrador 10: Análisis de Producción Industrial

Enunciado: Crear un DataFrame de datos de producción industrial en varias plantas. Filtrar productos con tasas de defectos altas (>3 %) fabricados en 2025. Calcular el costo total de producción por planta usando una operación de Serie. Contar los tipos de productos únicos fabricados. Mostrar los productos con defectos altos y estadísticas de costos.

Ejemplo de Entrada:

```
data = {
    'Planta': ['P1', 'P1', 'P2'],
    'Producto': ['Motor', 'Frenos', 'Motor'],
    'Costo': [10000, 5000, 12000],
```

```
    'Tasa_Defectos': [4.0, 2.5, 3.5],
    'Fecha_Producci n': ['2025-01-20', '2024-11-10', '2025-03-15
        ']
}
df = pd.DataFrame(data)
```

Salida Esperada:

```
Productos con Altos Defectos en 2025:
  Planta Producto  Costo  Tasa_Defectos Fecha_Producción
0    P1   Motor  10000           4.0     2025-01-20
2    P2   Motor  12000           3.5     2025-03-15

Costo Total por Planta:
P1    15000
P2    12000
Name: Costo, dtype: int64

Productos Únicos: 2
```

Aplicación Práctica: Mejorar la calidad de producción identificando productos defectuosos y optimizando costos.