

Laboratorio de Métodos Cuantitativos Aplicados a la Gestión

✓ Simulacro- Segundo Examen Parcial

14 de Noviembre, 2025

Nombre y Apellido: Alexander Ballera

Número de registro:

Abajo de esta línea, importá/installá todas las *libraries* que consideres relevantes para la resolución de este examen.

```
import sympy as sp          # para álgebra simbólica (integración, derivación, ecuaciones)
import numpy as np           # para operaciones numéricas y crear arrays de datos
import matplotlib.pyplot as plt # para crear visualizaciones y gráficos
import numpy_financial as nf
```

Unidad 4

✓ Ejercicio 1

La función de costo marginal de una empresa está dada por $C'(q) = 300 + 20q - 12q^2$ y los costos fijos ascienden a \$40000. Determinar la función de costos totales y la de costo medio. Graficar las funciones de costo medio y marginal en un mismo gráfico. Interpretar.

```
PASO 1: Definir variables simbólicas
PASO 2: Definir la función marginal (dato del problema)
PASO 3: Integrar para obtener la función total
PASO 4: Determinar la constante de integración
PASO 5: Calcular funciones derivadas (costo medio, beneficio, etc.)
PASO 6: Visualizar las funciones
```

```
# PASO 1: Definir variables simbólicas
# Crear variable de cantidad (q)
# positive=True asegura que q > 0 (no tiene sentido cantidad negativa)
q = sp.symbols('q', positive=True)

# Crear variable para la constante de integración (costo fijo)
# Sin restricciones porque puede ser cualquier valor real
F = sp.symbols('F')
```

```
# PASO 2: Definir la función marginal $C'(q)=300+20q-12q^2$
CMg = 300 + 20 * q - 12 * q**2
print("Función de Costo Marginal C'(q):")
CMg
```

Función de Costo Marginal $C'(q)$:

```
# PASO 3: Integrar para obtener la función total de costo C(q)
# Integrar CMg respecto a q y agregar constante F
# sp.integrate() encuentra la antiderivada
# + F representa el costo fijo (constante de integración)
C = sp.integrate(CMg, q) + F

print("Costo Total (con constante F):")
C
```

Costo Total (con constante F):

```
# Validación: la derivada de C debe ser igual a CMg
test = sp.diff(C, q)
print(f"dC/dq = {test}")
print(f"CMg    = {CMg}")
print(f"¿Son iguales? {test == CMg}")

dC/dq = -12*q**2 + 20*q + 300
CMg    = -12*q**2 + 20*q + 300
¿Son iguales? True
```

PASO 4: Determinar la constante de integración

```
condicion = sp.Eq(C.subs(q, 0), 40000)
print(f"Ecación a resolver: {condicion}")

F_sol = sp.solve(condicion, F)[0]
print(f"\nSolución: F = {F_sol}")
```

Ecuación a resolver: Eq(F, 40000)

Solución: F = 40000

```
# Sustituir el valor de F encontrado en la función C
# .subs(F, F_sol) reemplaza F por su valor numérico
C = C.subs(F, F_sol)
```

```
print("Costo Total final:")
C
```

Costo Total final:

```
# PASO 5: Calcular funciones derivadas (costo medio, beneficio, etc.)
# Calcular costo medio dividiendo costo total entre cantidad
CMed = C / q
print("Costo Medio (sin simplificar):")
CMed
```

Costo Medio (sin simplificar):

```
# Simplificar la expresión algebraica
CMed = sp.simplify(CMed)
print("\nCosto Medio (simplificado):")
CMed
```

Costo Medio (simplificado):

```
# PASO 6: Visualizar las funciones
# 6.1 Crear dominio (rango de valores de q para graficar)
# Rango ampliado para observar el comportamiento de las funciones
q_grid = np.linspace(0.1, 50, 500)
```

```
# 6.2 Convertir expresiones simbólicas a funciones numéricas
CMg_num = sp.lambdify(q, CMg)    # Función numérica para CMg(q)
CMed_num = sp.lambdify(q, CMed)  # Función numérica para CMed(q)
```

6.3 Evaluar funciones en todos los puntos del dominio

```
CMg_vals = CMg_num(q_grid)  # Valores de costo marginal
CMed_vals = CMed_num(q_grid) # Valores de costo medio
```

6.4 Crear visualización profesional
fig, ax = plt.subplots(figsize=(12, 7))

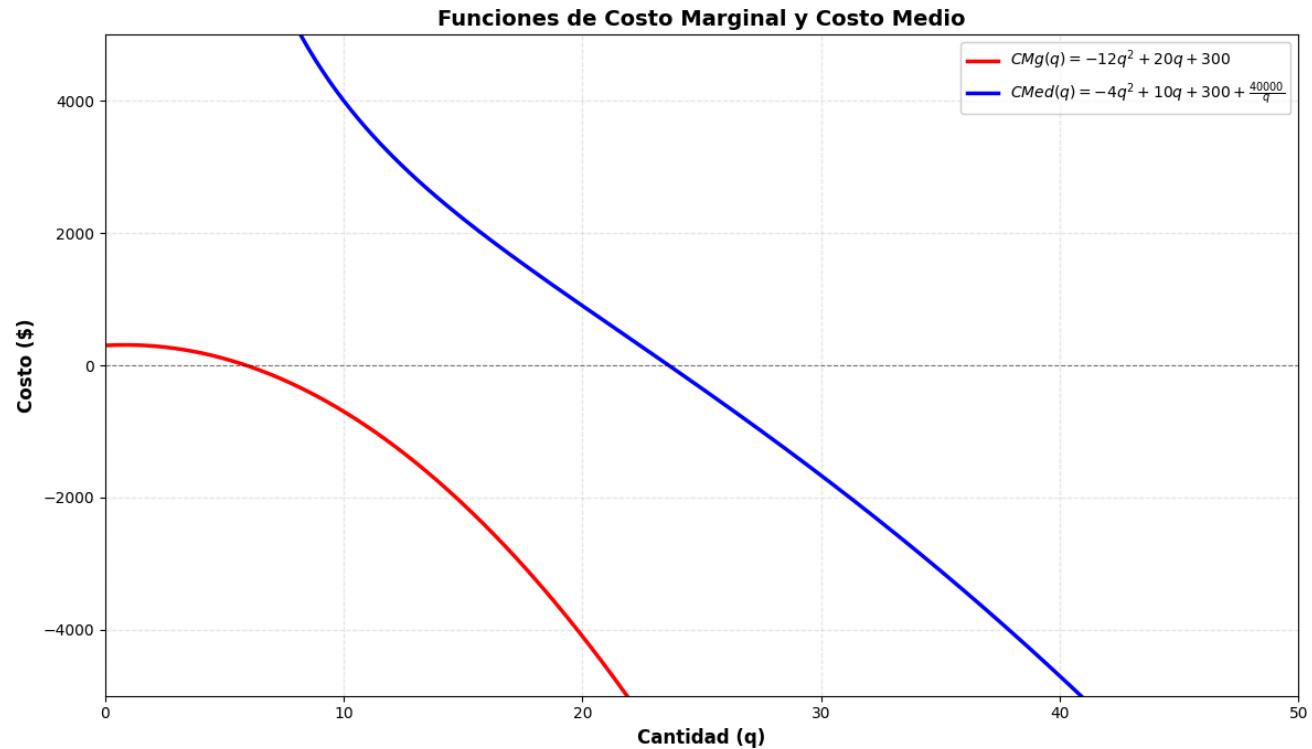
```
# Graficar ambas funciones con las fórmulas correctas
ax.plot(q_grid, CMg_vals, label='$CMg(q) = -12q^2 + 20q + 300$', linewidth=2.5, color='red')
ax.plot(q_grid, CMed_vals, label='$CMed(q) = -4q^2 + 10q + 300 + \frac{40000}{q}$', linewidth=2.5, color='blue')

# Línea de referencia en y=0
ax.axhline(0, color='black', linewidth=0.8, linestyle='--', alpha=0.5)
```

```
# Configurar el gráfico
ax.grid(True, alpha=0.3, linestyle='--')
ax.set_xlabel('Cantidad (q)', fontsize=12, fontweight='bold')
ax.set_ylabel('Costo ($)', fontsize=12, fontweight='bold')
ax.set_title('Funciones de Costo Marginal y Costo Medio', fontsize=14, fontweight='bold')
ax.legend(fontsize=10, loc='upper right', framealpha=0.9)

# Ajustar límites para observar ambas funciones claramente
ax.set_xlim(0, 50)
ax.set_ylim(-5000, 5000)

plt.tight_layout()
plt.show()
```



▼ Ejercicio 2

Dadas las siguientes funciones de oferta y demanda:

$$qd = \sqrt{16 - p}$$

$$qs = p/2 - 1/2$$

Calcular EC y EP Graficar ambas funciones en un mismo gráfico, donde el eje vertical debe ser el precio y el horizontal la cantidad.

Sombrear las áreas correspondientes a los excedentes encontrados indicando en el gráfico cuál es el excedente del consumidor y cuál el del productor

- Recalcular los excedentes teniendo en cuenta que se implementa un impuesto de suma fija $t = 3$ a los productores. Interpretar.

Unidad 5

▼ Ejercicio 3

A partir del cash flow MENSUAL establecido en la línea de código propuesta a continuación y considerando un costo de capital anual del 9% ,realice un análisis que contenga:

1- Métricas financieras: VAN, TIR, TIRM y PAYBACK DESCONTADO e interprete cada una de ellas.

2- Grafique el Van a diferentes tasas de descuento de costo del capital (marque en dicho gráfico la TIR y explíquelo)

3- Considere que el gerente financiero pronostica un aumento posible del costo de capital a la hora de iniciar el proyecto conformando un costo total del 21% anual. Interprete los resultados del nuevo análisis y haga un gráfico comparativo de ambas situaciones.

```
inversion_inicial = -70000
cash_flows = [
    13000, -10500, -10000, 18000, 25800, -22000, 27000, 18000, -27000, 29000,
    15000, -10000, 10000, -18000, 20000, -21000, 24000, -27000, 28000, -10000,
    -8000, 40000, 18000, -13000, 25000, 18000, -26000, 25000, 42000
]
nuevo_cash_flow = [inversion_inicial] + cash_flows
r = 0.09
m = 12
r = (1 + r / m) ** m - 1
r = r / 12
```

```
# 1- Métricas financieras: VAN, TIR, TIRM y PAYBACK DESCONTADO e interprete cada una de ellas.
# VAN
```

```
VAN = round(nf.npv(r, nuevo_cash_flow), 2)
print(VAN)
```

```
97148.95
```

```
# TIR
TIR = round(nf.irr(nuevo_cash_flow), 2)
print(TIR)
```

```
0.07
```

```
# TIRM
TIRM = round(nf.mirr(nuevo_cash_flow, finance_rate=r, reinvest_rate=TIR), 2)
print(TIRM)
```

```
0.05
```

```
# PAYBACK DESCONTADO
cumulative_discounted_cash_flow = 0
prd = None
for t in range(1, len(cash_flows)):
    # Descontar el flujo de caja actual
    discounted_flow = cash_flows[t] / (1 + r)**t # aaaaa pero aca cambia se descuentan los flujos periodo periodo pc
    cumulative_discounted_cash_flow += discounted_flow

    # Verificar si el flujo de caja descontado acumulado es mayor o igual a la inversión inicial
    if cumulative_discounted_cash_flow >= inversion_inicial:
        # Si es así, calcular el período de recuperación descontado
        # El período de recuperación descontado es el período actual 't' más la fracción del flujo descontado del úl
        remaining_investment = inversion_inicial - (cumulative_discounted_cash_flow - discounted_flow)
        prd = t - 1 + (remaining_investment / discounted_flow)
        break

if prd is not None:
    print(f"El Periodo de Recuperación Descontado (PRD) es: {prd:.2f} años")
else:
    print("La inversión inicial no se recupera dentro del período del análisis descontado.")
```

```
El Periodo de Recuperación Descontado (PRD) es: 6.72 años
```

▼ Ejercicio 4

Importe la siguiente base de datos cripto_1.csv y realice la exploración inicial, gráficas relevantes.

