

Producción de Laptops y Tablets

Cada estudiante resolverá un modelo único generado automáticamente con base en su **suma de sus 3 últimos números de su cédula**.

Enunciado

Una empresa de tecnología produce **laptops** y **tablets** para exportación. Cada uno requiere:

- microchips,
- horas de ensamblaje,
- cantidad de pantalla LCD.

Se desea maximizar los ingresos, respetando los límites de recursos disponibles por semana.

Ejecute la celda siguiente para generar tus parámetros personalizados.

```
!pip install gurobipy
from gurobipy import Model, GRB
import pandas as pd
```

```
# Ingreso del dígito
```

```
d = int(input("Ingresa el último dígito de tu cédula: "))
```

```
# Recursos requeridos por unidad
```

```
chips_laptop = 10 + d % 3
```

```
chips_tablet = 8 + d % 2
```

```
lcd_laptop = 1
```

```
lcd_tablet = 1
```

```
ensam_laptop = 12 + d % 4
```

```
ensam_tablet = 8 + d % 3
```

```
# Recursos disponibles
```

```
chips_total = 1000 + 100 * d
```

```
lcd_total = 800 + 50 * d
```

```
ensam_total = 1200 + 75 * d
```

```
# Ingresos unitarios (en millones)
```

```
precio_laptop = 6 + 0.4 * d
```

```
precio_tablet = 4 + 0.3 * d
```

```
# Mostrar parámetros
```

```
print(f"Datos personalizados para: {d}")
```

```
print(f"Chips disponibles: {chips_total}, LCD: {lcd_total}, Ensamblaje: {ensam_total}")
```

```
print(f"Laptop -> chips: {chips_laptop}, LCD: 1, Ensamblaje: {ensam_laptop}, Precio:  
${precio_laptop:.2f}M")
```

```
print(f"Tablet -> chips: {chips_tablet}, LCD: 1, Ensamblaje: {ensam_tablet}, Precio:  
${precio_tablet:.2f}M")
```

```
# Crear modelo
```

```
model = Model("Laptops_Tablets_Optimización")
```

```
# Variables de decisión
```

```
x1 = model.addVar(vtype=GRB.CONTINUOUS, name="Laptops", lb=0)
```

```
x2 = model.addVar(vtype=GRB.CONTINUOUS, name="Tablets", lb=0)
```

```
# Objetivo
```

```
model.setObjective(precio_laptop * x1 + precio_tablet * x2, GRB.MAXIMIZE)
```

```
# Restricciones
```

```
model.addConstr(chips_laptop * x1 + chips_tablet * x2 <= chips_total, name="Chips")
```

```
model.addConstr(lcd_laptop * x1 + lcd_tablet * x2 <= lcd_total, name="LCD")
```

```
model.addConstr(ensam_laptop * x1 + ensam_tablet * x2 <= ensam_total,  
name="Ensamblaje")
```

```
# Resolver
```

```
model.optimize()
```

```
# Resultados
```

```
if model.status == GRB.OPTIMAL:
```

```
    print("\nResultados:")
```

```
    print(f"Laptops a producir: {x1.X:.0f}")
```

```
    print(f"Tablets a producir: {x2.X:.0f}")
```

```
    print(f"Ingreso total: ${model.ObjVal:.2f} millones")
```

```
else:
```

```
    print("No se encontró solución óptima.")
```