

Ejercicio Parametrizado – Producción de Sillas y Mesas

Cada estudiante resolverá un problema único, generado automáticamente a partir de **la suma de los dos últimos números de su código estudiantil**.

Enunciado

La empresa **MuebleFácil S.A.S.** fabrica sillas y mesas. Cada una requiere diferentes cantidades de madera, pintura y mano de obra.

Tu objetivo es determinar cuántas unidades de cada producto fabricar para **maximizar el ingreso**, considerando las limitaciones de recursos.

Los valores del modelo se generan automáticamente a partir de la suma de los dos últimos números de su código estudiantil. Ejecuta la siguiente celda para personalizar tu modelo.

```
!pip install gurobipy
```

```
from gurobipy import Model, GRB
```

```
import pandas as pd
```

```
# Ingresa aquí
```

```
d = int(input("Ingresa la suma de los 2 últimos números de su código estudiantil: "))
```

```
# Parámetros personalizados
```

```
a1 = 30 + d % 3
```

```
a2 = 40 + d % 4
```

```
b1 = 20 + d % 2
```

$b2 = 30 + (d + 1) \% 3$

$c1 = 40 + d \% 2$

$c2 = 50 + (d + 2) \% 3$

$M = 1000 + 100 * d$

$P = 600 + 500 * d$

$H = 800 + 70 * d$

$u1 = 400 + 20 * d$

$u2 = 700 + 30 * d$

Mostrar parámetros

`print(f"Parámetros personalizados para el estudiante con ID terminado en {d}:")`

`print(f"Silla - Madera: {a1}, Pintura: {b1}, Mano de obra: {c1}, Precio: ${u1}")`

`print(f"Mesa - Madera: {a2}, Pintura: {b2}, Mano de obra: {c2}, Precio: ${u2}")`

`print(f"Disponibilidad - Madera: {M}, Pintura: {P}, Mano de obra: {H}")`

Crear modelo

`model = Model("Sillas_Mesas_Parametrizado")`

Variables de decisión: $x1$ = sillas, $x2$ = mesas

`x1 = model.addVar(vtype=GRB.INTEGER, name="Sillas", lb=0)`

`x2 = model.addVar(vtype=GRB.INTEGER, name="Mesas", lb=0)`

Función objetivo

```
model.setObjective(u1 * x1 + u2 * x2, GRB.MAXIMIZE)
```

```
# Restricciones
```

```
model.addConstr(a1 * x1 + a2 * x2 <= M, name="Madera")
```

```
model.addConstr(b1 * x1 + b2 * x2 <= P, name="Pintura")
```

```
model.addConstr(c1 * x1 + c2 * x2 <= H, name="ManoObra")
```

```
# Resolver modelo
```

```
model.optimize()
```

```
# Mostrar resultados
```

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

```
    print("\n 📋 Resultados óptimos:")
```

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

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

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

```
else:
```

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