

## Ejemplo 1

**Ejemplo 1.** En una oficina se necesita para cada día de la semana el número de trabajadores/as a tiempo completo de la siguiente tabla:

Día	Trabajadores/as
1. Lunes	15
2. Martes	13
3. Miércoles	15
4. Jueves	18
5. Viernes	14
6. Sábado	16
7. Domingo	10

```
from pulp import LpMinimize, LpProblem, LpVariable, LpStatus

# Crear un problema de minimización
prob = LpProblem("Programacion_Trabajo_Semanal", LpMinimize)

# Definir las variables de decisión (número de trabajadores que
empiezan cada día)
x = {i: LpVariable(f"x_{i}", lowBound=0, cat='Integer') for i in
range(1, 8)}

# Definir la función objetivo (minimizar el número total de
trabajadores)
prob += sum(x[i] for i in range(1, 8)), "Total de trabajadores"

# Definir las restricciones diarias
prob += x[1] + x[5] + x[6] + x[7] + x[2] >= 15, "Lunes"
prob += x[1] + x[2] + x[6] + x[7] + x[3] >= 13, "Martes"
prob += x[1] + x[2] + x[3] + x[7] + x[4] >= 15, "Miércoles"
prob += x[2] + x[3] + x[4] + x[1] + x[5] >= 18, "Jueves"
prob += x[3] + x[4] + x[5] + x[2] + x[6] >= 14, "Viernes"
prob += x[4] + x[5] + x[6] + x[3] + x[7] >= 16, "Sábado"
```

```

prob += x[5] + x[6] + x[7] + x[4] + x[1] >= 10, "Domingo"

# Resolver el problema
prob.solve()

# Mostrar el estado de la solución
print(f"Estado de la solución: {LpStatus[prob.status]}")

# Mostrar los valores óptimos de las variables
for i in range(1, 8):
    print(f"Trabajadores que empiezan el día {i}: {x[i].varValue}")

# Mostrar el número total de trabajadores
print(f"Total de trabajadores necesarios: {prob.objective.value()}")

```

**Ejemplo 2. El problema de la mochila.** Una mochila con capacidad de 12 kg se quiere llenar con objetos de peso y valor fijado. En la siguiente tabla se da el peso y el valor de cada uno de los cuatro objetos.

	1	2	3	4
Peso (kg)	3	6	5	5
Valor (euros)	15	25	12	10

```

from pulp import LpMaximize, LpProblem, LpVariable, lpSum, LpStatus

# Crear un problema de maximización
prob = LpProblem("Problema_de_la_Mochila", LpMaximize)

# Definir las variables de decisión (si se selecciona un objeto o no)
x1 = LpVariable("x1", 0, 1, cat="Integer")
x2 = LpVariable("x2", 0, 1, cat="Integer")
x3 = LpVariable("x3", 0, 1, cat="Integer")
x4 = LpVariable("x4", 0, 1, cat="Integer")

# Lista de valores y pesos
valores = [15, 25, 12, 10]
pesos = [3, 6, 5, 5]
capacidad = 12

```

```
# Función objetivo: maximizar el valor de los objetos seleccionados
prob += 15 * x1 + 25 * x2 + 12 * x3 + 10 * x4, "Valor_Total"

# Restricción: la suma de los pesos de los objetos seleccionados no
# debe exceder la capacidad
prob += 3 * x1 + 6 * x2 + 5 * x3 + 5 * x4 <= capacidad,
"Restriccion_Capacidad"

# Resolver el problema
prob.solve()

# Mostrar el estado de la solución
print(f"Estado de la solución: {LpStatus[prob.status]}")

# Mostrar qué objetos se seleccionan
print(f"x1 (objeto 1): {x1.varValue}")
print(f"x2 (objeto 2): {x2.varValue}")
print(f"x3 (objeto 3): {x3.varValue}")
print(f"x4 (objeto 4): {x4.varValue}")

# Mostrar el valor total
print(f"Valor total máximo: {prob.objective.value()}")
```