# IDO- Tarea 4

Marcelino Sánchez Rodríguez 191654 Diana Espinosa Ruiz

2024-02-12

## TAREA 1 PPL

## Problema 2

```
using JuMP, HiGHS
model = Model(HiGHS.Optimizer)
@variable(model, x[1:3] >= 0)

@objective(model, Max, 3x[1] + 4x[2] - 3x[3] )

@constraint(model, x[1] + 5x[2] + 8x[3] <= 40)

optimization_result = optimize!(model)

value.(x)
```

```
Running HiGHS 1.6.0: Copyright (c) 2023 HiGHS under MIT licence terms
Presolving model
1 rows, 1 cols, 1 nonzeros
0 rows, 0 cols, 0 nonzeros
Presolve : Reductions: rows 0(-1); columns 0(-3); elements 0(-3) - Reduced to empty
Solving the original LP from the solution after postsolve
Model   status      : Optimal
Objective value     :  1.2000000000e+02
HiGHS run time      :          0.00

3-element Vector{Float64}:
 40.0
```

```
  0.0
  0.0
```

## Problema 3

```
using JuMP, HiGHS
model = Model(HiGHS.Optimizer)
@variable(model, x[1:3] >= 0)

@objective(model, Max, 3x[1] + 4x[2] - 3x[3] )

@constraint(model, x[1] + 5x[2] + 8x[3] <= 40)

optimization_result = optimize!(model)

value.(x)
```

```
Running HiGHS 1.6.0: Copyright (c) 2023 HiGHS under MIT licence terms
Presolving model
1 rows, 1 cols, 1 nonzeros
0 rows, 0 cols, 0 nonzeros
Presolve : Reductions: rows 0(-1); columns 0(-3); elements 0(-3) - Reduced to empty
Solving the original LP from the solution after postsolve
Model   status      : Optimal
Objective value     :  1.2000000000e+02
HiGHS run time      :          0.00

3-element Vector{Float64}:
 40.0
  0.0
  0.0
```

## Problema 4

```
using JuMP, HiGHS

model = Model(HiGHS.Optimizer)
N = 4
M= 4
```

```julia
@variable(model, x[1:N, 1:N], Bin)

# Restricciones de filas y columnas (ya las tienes)
@constraint(model, [i=1:N], sum(x[i,j] for j=1:N) <= 1)  # Restricción de fila
@constraint(model, [j=1:N], sum(x[i,j] for i=1:N) <= 1)  # Restricción de columna

# Restricciones de las diagonales principales
for d = -(N-1):(N-1)
    @constraint(model, sum(x[i, i+d] for i=1:N if 1 <= i+d <= N) <= 1)
end

# Restricciones de las diagonales secundarias
for d = 2:(2*N)
    @constraint(model, sum(x[i, d-i] for i=1:N if 1 <= d-i <= N) <= 1)
end

@constraint(model, sum(x[i,j] for i=1:N, j=1:N) <= M)


@objective(model, Max, sum(x[i,j] for i=1:N, j=1:N))

optimize!(model)
```

```
Running HiGHS 1.6.0: Copyright (c) 2023 HiGHS under MIT licence terms
Presolving model
19 rows, 16 cols, 76 nonzeros
19 rows, 16 cols, 94 nonzeros
Objective function is integral with scale 1

Solving MIP model with:
   19 rows
   16 cols (16 binary, 0 integer, 0 implied int., 0 continuous)
   94 nonzeros

        Nodes      |    B&B Tree     |            Objective Bounds              |  Dynamic Co
     Proc. InQueue |  Leaves   Expl. | BestBound       BestSol              Gap |   Cuts   In

         0       0         0   0.00%   5               -inf                inf        0
 T       0       0         0   0.00%   5               4                 25.00%       0

Solving report
  Status            Optimal
```

```
Primal bound       4
Dual bound         4
Gap                0% (tolerance: 0.01%)
Solution status    feasible
                   4 (objective)
                   0 (bound viol.)
                   0 (int. viol.)
                   0 (row viol.)
Timing             0.00 (total)
                   0.00 (presolve)
                   0.00 (postsolve)
Nodes              1
LP iterations      17 (total)
                   0 (strong br.)
                   0 (separation)
                   0 (heuristics)
```

```
value.(x)
```

```
4×4 Matrix{Float64}:
 -0.0  -0.0   1.0   0.0
  1.0  -0.0   0.0  -0.0
  0.0   0.0  -0.0   1.0
 -0.0   1.0  -0.0   0.0
```

## Problema 5

```
using JuMP, HiGHS
model = Model(HiGHS.Optimizer)
@variable(model, x[1:3] >= 0)

@variable(model, u >= 0 , Bin)

@objective(model, Max, -3x[1] - 4x[2] + 3x[3] )
M=1000

@constraint(model, x[1] + x[2] + 4x[3] <= 60)
@constraint(model, -x[1] + 2x[2] + x[3] >= 12)
@constraint(model, x[2] + x[3] <= M*u)
@constraint(model, x[1] + x[3] <= 54 + M*(1-u))
```

```
@constraint(model, x[2] + x[3] >= u)

optimization_result = optimize!(model)

value.(x)
```

```
Running HiGHS 1.6.0: Copyright (c) 2023 HiGHS under MIT licence terms
Presolving model
5 rows, 3 cols, 12 nonzeros
4 rows, 2 cols, 8 nonzeros
3 rows, 2 cols, 6 nonzeros

Solving MIP model with:
   3 rows
   2 cols (0 binary, 0 integer, 0 implied int., 2 continuous)
   6 nonzeros

         Nodes      |    B&B Tree     |            Objective Bounds              |  Dynamic Co
     Proc. InQueue |  Leaves   Expl. | BestBound       BestSol              Gap |   Cuts   In

         0       0         0   0.00%   45              -inf                 inf        0

Solving report
  Status            Optimal
  Primal bound      45
  Dual bound        45
  Gap               0% (tolerance: 0.01%)
  Solution status   feasible
                    45 (objective)
                    0 (bound viol.)
                    0 (int. viol.)
                    0 (row viol.)
  Timing            0.00 (total)
                    0.00 (presolve)
                    0.00 (postsolve)
  Nodes             1
  LP iterations     0 (total)
                    0 (strong br.)
                    0 (separation)
                    0 (heuristics)

3-element Vector{Float64}:
```

```
 0.0
 0.0
15.0
```

## Problema 6

```julia
using JuMP, HiGHS
model = Model(HiGHS.Optimizer)
@variable(model, x[1:3] >= 0)

@objective(model, Max, -3x[1] - 4x[2] + 3x[3] )

@constraint(model, x[1] + x[2] + 4x[3] <= 60)
@constraint(model, -x[1] + 2x[2] + x[3] >= 12)

optimization_result = optimize!(model)

value.(x)
```

```
Running HiGHS 1.6.0: Copyright (c) 2023 HiGHS under MIT licence terms
Presolving model
2 rows, 2 cols, 4 nonzeros
0 rows, 0 cols, 0 nonzeros
Presolve : Reductions: rows 0(-2); columns 0(-3); elements 0(-6) - Reduced to empty
Solving the original LP from the solution after postsolve
Model   status      : Optimal
Objective value     :  4.5000000000e+01
HiGHS run time      :          0.00

3-element Vector{Float64}:
 0.0
 0.0
15.0
```