INTRODUCIÓN Á PROGRAMACIÓN LINEAL

Resolución de problemas de programación lineal mediante *software* libre

Laura Davila Pena, Alejandro Saavedra Nieves e Balbina Casas Méndez
Departamento de Estatística, Análise Matemática e Optimización

I. NOCIÓNS BÁSICAS DE PROGRAMACIÓN MATEMÁTICA

Un **problema de optimización** vén dado por un par *(F, c)*, onde *F* é o conxunto de puntos factibles e *c* é a función de custo. O problema consiste en atopar un punto factible $x \in F$ tal que, para todo $y \in F$, se cumpra que $c(x) \leq c(y)$. Dise que o punto *x* nestas condicións é unha **solución óptima** do problema.
Nese contexto, un **problema de programación matemática** consiste en atopar unha solución ao problema:

$$\begin{array}{ll} \text{Minimizar}_{x \in R^n} & c(x) \\ \text{suxeito a} & g_i(x) \leq 0, \text{ para cada } i = 1, \dots, m \\ & h_j(x) = 0, \text{ para cada } j = 1, \dots, l, \end{array}$$

onde:

- $x = (x_1, \dots, x_n) \in R^n$ son as **variables de decisión** para as que buscamos unha configuración de valores óptima.
- A función *c* representa a **función de custo** a minimizar (ou beneficio a maximizar) asociado a cada combinación das variables de decisión.
- As **restricións** son funcións que representan que configuracións de valores das variables $x_1, \dots, x_n$ son factibles. O conxunto de puntos *x* que as cumpren denomínase **rexión factible**. Distínguense dous tipos de restricións:
  - **Restricións de desigualdade**, $g_i(x) \leq 0$, para cada $i = 1, \dots, m$.
  - **Restricións de igualdade**, $h_j(x) = 0$, para cada $j = 1, \dots, l$.

A PROGRAMACIÓN LINEAL

Un **problema de programación lineal** é un caso particular de problema de programación matemática no que as funcións usadas como función obxectivo a minimizar e as restricións son lineais.

Os elementos que o compoñen son:

- **Vector de custos**, $c \in R^n$.
- **Matriz de restricións**, $A \in R^{m \times n}$, cos seus elementos da forma a_{ij}.
- **Vector de lados dereitos**, $b \in R^m$.

$$\begin{array}{ll} \text{minimizar} & c_1x_1 + c_2x_2 + \cdots + c_nx_n \\ \text{suxeito a} & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \leq b_1 \\ & a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \leq b_2 \\ & \quad \dots\dots\dots\dots\dots \\ & a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \leq b_m \\ & x_i \geq 0,\ i \in \{1, \dots, n\}. \end{array}$$

e, equivalentemente en forma matricial,

$$\begin{array}{ll} \text{minimizar} & c \cdot x \\ \text{suxeito a} & Ax \leq b \\ & x \geq 0. \end{array}$$

Cómpre destacar que aínda que se formulou o problema de programación lineal en termos da minimización de custos, a maximización de beneficios tamén admite unha modelización análoga ao considerar o vector *c* como un vector de ganancias asociadas a cada unha das variables de decisión.

A PROGRAMACIÓN LINEAL NA HISTORIA

Os problemas de programación lineal formúlanse como modelos matemáticos e xorden durante a Segunda Guerra Mundial, no século XX, co obxectivo de planificar os problemas loxísticos e estratéxicos do exército americano. Por exemplo, usáronse para a planificación do traslado de tropas dos Estados Unidos a Europa ou deseñar a dieta dos soldados nunha contorna de escaseza de recursos a mínimo custo ou para determinar o tamaño dos convois americanos.

A PROGRAMACIÓN LINEAL NA ACTUALIDADE

A programación lineal recibiu un gran pulo grazas ao desenvolvemento de ordenadores, o que permitiu resolver problemas nesta clase cada vez máis grandes e usalos en contextos moi diferentes. Entre as súas aplicacións, destaca o seu emprego en eidos como o da enxeñaría, o da economía, ou o da bioloxía, para deseñar rutas de vehículos, optimizar fluxos en redes, deseñar sistemas de control de tráfico, manexar de maneira óptima carteiras financeiras, optimizar o uso da información en bioinformática ou o seu emprego como técnica de aprendizaxe automática no eido da intelixencia artificial.

II. FORMULACIÓN E RESOLUCIÓN DUN PROBLEMA DE PROGRAMACIÓN LINEAL

Nesta sección propoñeremos un caso realista que pode ser modelado usando técnicas de programación lineal. O obxectivo é dobre, xa que ademais de ilustrar o exposto na sección anterior, abordarase a súa resolución. Dado que se trata dun problema con dúas únicas variables de decisión, será fácil observar a súa resolución dende o punto de vista gráfico. Ademais, ilustrarase o emprego de *software* libre na resolución desta clase de problemas.

O PROBLEMA DA EMPRESA DE PINTURAS

Unha empresa fabrica pintura para exteriores e pintura para interiores. A empresa debe decidir a cantidade (en toneladas) que vai fabricar de cada tipo de pintura, sabendo que por cada tonelada de pintura de exteriores gañará 5.000 euros, e por cada tonelada de pintura para interiores gañará 4.000 euros. Non obstante, debe respectar certas restricións:

- Hai dúas materias primas A e B, das que só se dispón de 24 e 6 toneladas, respectivamente.
- Cada tonelada de pintura de exteriores require de 6 toneladas da materia prima A e 1 tonelada da materia prima B.
- Cada tonelada de pintura de interiores require de 4 toneladas da materia prima A e 2 toneladas da materia prima B.
- Ademais, unha enquisa de mercado indica que a demanda de pintura para interiores non pode ser maior que 1 tonelada máis que a de pintura para exteriores.
- Por último, sábese que a demanda máxima de pintura para interiores é de 2 toneladas.

OBXECTIVO: Cal é a cantidade óptima que se debe producir de cada tipo de pinturas?

FORMULACIÓN DO PROBLEMA: Se denotamos por x a cantidade de toneladas para producir de pintura de exteriores e por y a cantidade de toneladas de pintura de interiores, a ganancia pola produción é $5x + 4y$, de maneira que hai que determinar os valores de x e y que maximicen a devandita ganancia.

Á vez, as restricións tamén se poden expresar en termos de x e y, de maneira que a formulación completa se poida representar como segue.

$$
\begin{array}{ll}
\text{maximizar} & 5x + 4y \\
\text{suxeito a} & 6x + 4y \leq 24 \\
 & x + 2y \leq 6 \\
 & -x + y \leq 1 \\
 & y \leq 2 \\
 & x \geq 0,\ y \geq 0.
\end{array}
$$

II. i. RESOLUCIÓN MEDIANTE O PROGRAMA R

Nesta sección mostramos os pasos a seguir para resolver un problema de programación lineal no software R, que pode obterse libremente dende https://cloud.r-project.org/. Unha vez instalado, necesitamos dispoñer do paquete `lpSolveAPI`, que será o que nos permita tratar tales problemas. Para iso, abrimos R e escribimos a orde `install.packages("lpSolveAPI")`. Este paquete só necesita instalarse a primeira vez que o imos empregar.

Paso 1: carga da libraría e construción do problema

En primeiro lugar cargamos a libraría `lpSolveAPI`. Logo construímos un problema de programación lineal con 4 restricións e 2 variables, mediante a función `make.lp()`.

```
> library(lpSolveAPI)
> pintura <- make.lp(4,2)
> pintura
Model name:
            C1    C2
Minimize     0     0
R1           0     0  free  0
R2           0     0  free  0
R3           0     0  free  0
R4           0     0  free  0
Kind       Std   Std
Type      Real  Real
Upper      Inf   Inf
Lower        0     0
```

Paso 2: coeficientes da función obxectivo e das restricións

Introducimos os coeficientes das variables na función obxectivo e nas restricións utilizando as funcións `set.objfn()` e `set.row()`, respectivamente.

```
> set.objfn(pintura,c(5,4))
> set.row(pintura,1,c(6,4))
> set.row(pintura,2,c(1,2))
> set.row(pintura,3,c(-1,1))
> set.row(pintura,4,c(0,1))
> pintura
Model name:
            C1    C2
Minimize     5     4
R1           6     4  free  0
R2           1     2  free  0
R3          -1     1  free  0
R4           0     1  free  0
Kind       Std   Std
Type      Real  Real
Upper      Inf   Inf
Lower        0     0
```

Paso 3: signo das restricións e lados dereitos

Mediante a función `set.constr.type()`, indicamos que as restricións son de "≤". Logo, introducimos os valores dos lados dereitos das restricións utilizando `set.rhs()`.

```
> set.constr.type(pintura,rep("<=",4))
> set.rhs(pintura,c(24,6,1,2))
> pintura
Model name:
            C1    C2
Minimize     5     4
R1           6     4   <=   24
R2           1     2   <=    6
R3          -1     1   <=    1
R4           0     1   <=    2
Kind       Std   Std
Type      Real  Real
Upper      Inf   Inf
Lower        0     0
```

Paso 4: tipo de problema de optimización

A continuación, debemos indicar que o obxectivo é maximizar, para o cal empregamos a función `lp.control()`.

```
> lp.control(pintura,sense="max")
> pintura
Model name:
            C1    C2
Maximize     5     4
R1           6     4   <=   24
R2           1     2   <=    6
R3          -1     1   <=    1
R4           0     1   <=    2
Kind       Std   Std
Type      Real  Real
Upper      Inf   Inf
Lower        0     0
```

Paso 5: resolución do problema

Empregamos a función `solve()` para resolver o noso problema de optimización.

```
> solve(pintura)
[1] 0
```

Que a saída sexa 0 significa que se atopou unha solución factible óptima ao noso problema.

Paso 6: valores das variables, función obxectivo e lados esquerdos das restricións no óptimo

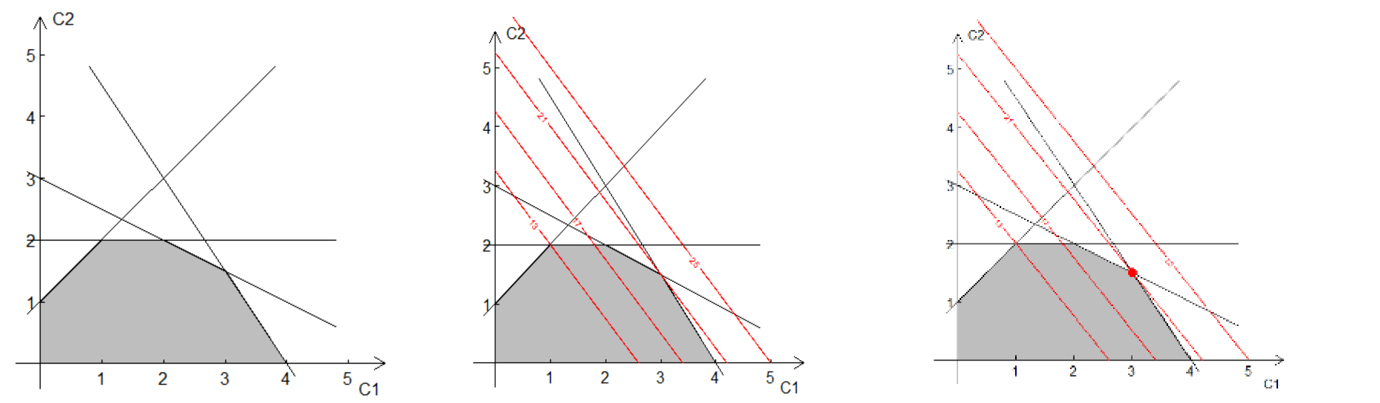
Mediante as funcións `get.variables()`, `get.objective()` e `get.constraints()`, podemos obter os valores das variables, a función obxectivo e os lados esquerdos das restricións, respectivamente, no óptimo.

```
> get.variables(pintura)
[1] 3.0 1.5
> get.objective(pintura)
[1] 21
> get.constraints(pintura)
[1] 24.0  6.0 -1.5  1.5
```

Representación gráfica

Para representar o conxunto factible debemos empregar a función `plot.lpExtPtr()`. Unha vez construído o conxunto factible, definimos as curvas de nivel da función obxectivo e representámolas sobre a gráfica que contén o conxunto factible mediante a función `contour`. Finalmente, engadimos sobre esta gráfica o punto onde se alcanza a solución óptima, (3, 1.5), para o cal empregamos a función `points`.

```
> plot.lpExtPtr(pintura)
> xs <- seq(0,10,length=1000)
> ys <- seq(0,10,length=1000)
> f <- function(x,y){5*x+4*y}
> zs <- outer(xs,ys,FUN=f)
> contour(xs,ys,zs,levels=c(13,17,21,25),
+          col="red",add=TRUE)
> points(3,1.5,col="red",pch=16,cex=1.5)
```

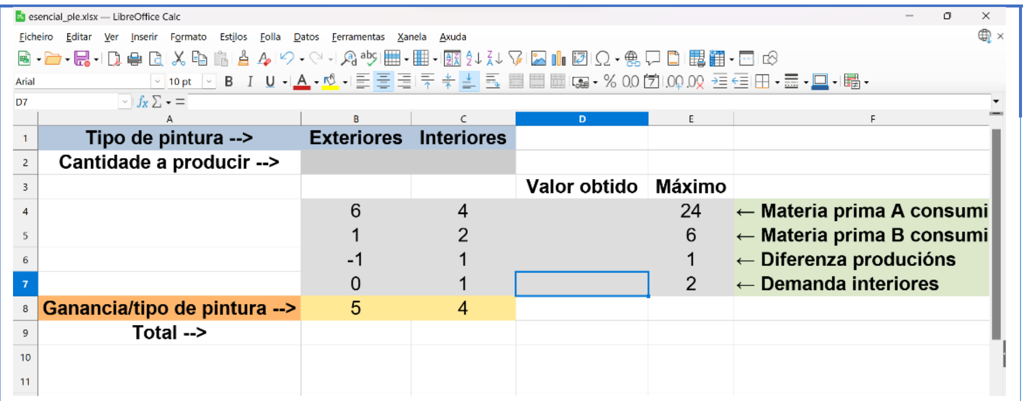


II. ii. RESOLUCIÓN MEDIANTE A FOLLA DE CÁLCULO DE *LIBREOFFICE*

Nesta sección mostramos os pasos a seguir para resolver un problema de programación lineal mediante a folla de cálculo da ferramenta *LibreOffice*, que pode obterse libremente dende https://es.libreoffice.org/.

Paso 1: novo ficheiro e introdución dos datos do problema

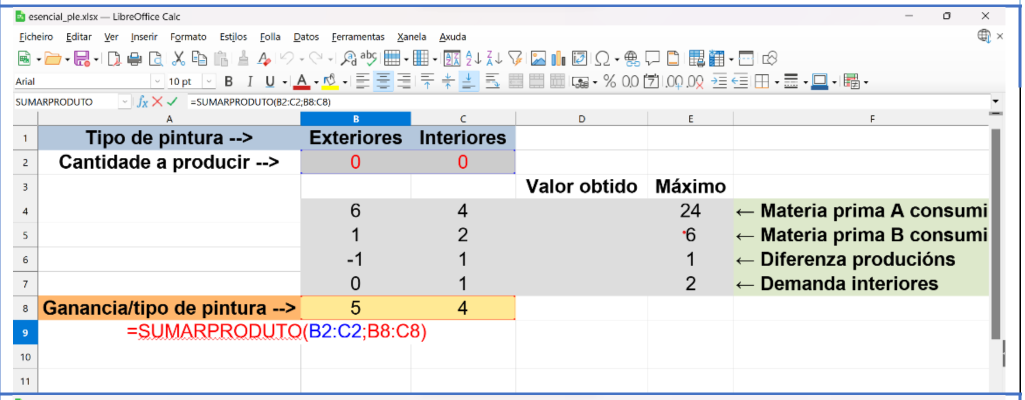
En primeiro lugar, abrimos unha folla de cálculo nova en *LibreOffice* e introducimos os **datos** do problema: coeficientes das restricións e da función obxectivo. É útil titular as celas da folla e mesmo usar cores.



	A	B	C	D	E	F
1	Tipo de pintura -->	Exteriores	Interiores			
2	Cantidade a producir -->					
3				Valor obtido	Máximo	
4		6	4		24	← Materia prima A consumi
5		1	2		6	← Materia prima B consumi
6		-1	1		1	← Diferenza producións
7		0	1		2	← Demanda interiores
8	Ganancia/tipo de pintura -->	5	4			
9	Total -->					
10						
11						

Paso 2: inicialización das variables e función obxectivo

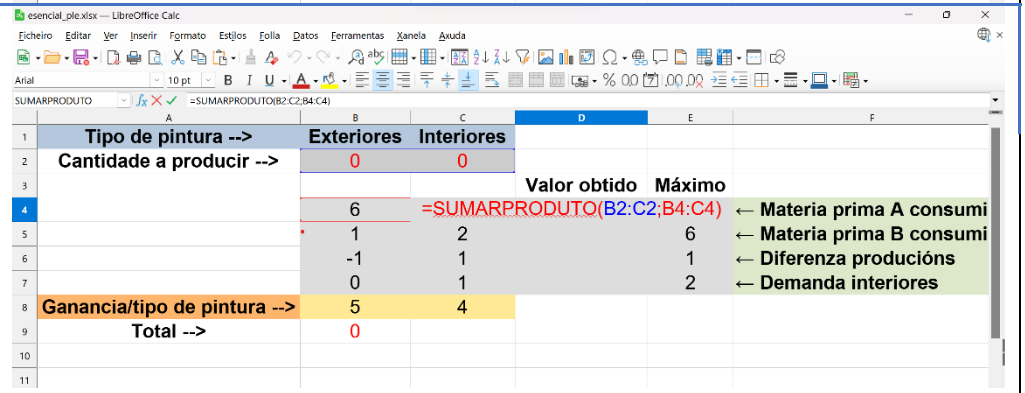
Inicializamos as **variables** do noso problema a 0. Na cela B9, por exemplo, introducimos a **función obxectivo** por medio do comando SUMARPRODUTO.



	A	B	C	D	E	F
1	Tipo de pintura -->	Exteriores	Interiores			
2	Cantidade a producir -->	0	0			
3				Valor obtido	Máximo	
4		6	4		24	← Materia prima A consumi
5		1	2		6	← Materia prima B consumi
6		-1	1		1	← Diferenza producións
7		0	1		2	← Demanda interiores
8	Ganancia/tipo de pintura -->	5	4			
9		=SUMARPRODUTO(B2:C2;B8:C8)				
10						
11						

Paso 3a: a función dunha restrición

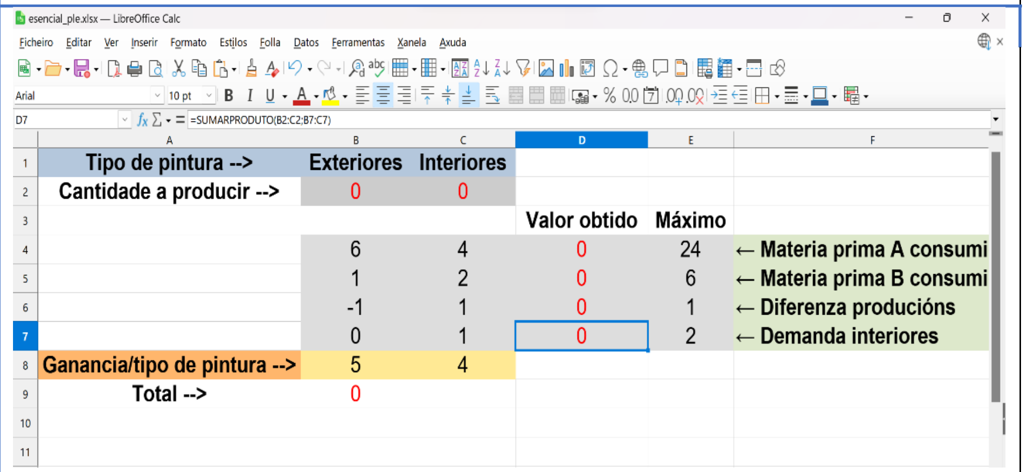
Na cela D4 introducimos a función que define a **primeira restrición** novamente por medio do comando SUMARPRODUTO.



	A	B	C	D	E	F
1	Tipo de pintura -->	Exteriores	Interiores			
2	Cantidade a producir -->	0	0			
3				Valor obtido	Máximo	
4		6	=SUMARPRODUTO(B2:C2;B4:C4)			← Materia prima A consumi
5		1	2		6	← Materia prima B consumi
6		-1	1		1	← Diferenza producións
7		0	1		2	← Demanda interiores
8	Ganancia/tipo de pintura -->	5	4			
9	Total -->	0				
10						
11						

Paso 3b: as funcións das restantes restricións

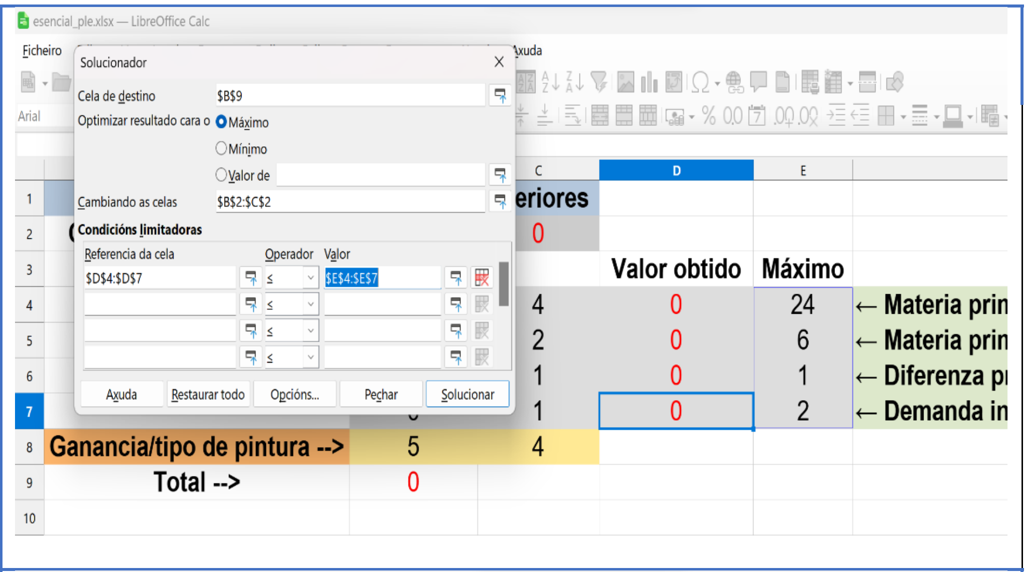
De forma análoga introdúcense as funcións que definen as **restantes restricións**.



	A	B	C	D	E	F
1	Tipo de pintura -->	Exteriores	Interiores			
2	Cantidade a producir -->	0	0			
3				Valor obtido	Máximo	
4		6	4	0	24	← Materia prima A consumi
5		1	2	0	6	← Materia prima B consumi
6		-1	1	0	1	← Diferenza producións
7		0	1	0	2	← Demanda interiores
8	Ganancia/tipo de pintura -->	5	4			
9	Total -->	0				
10						
11						

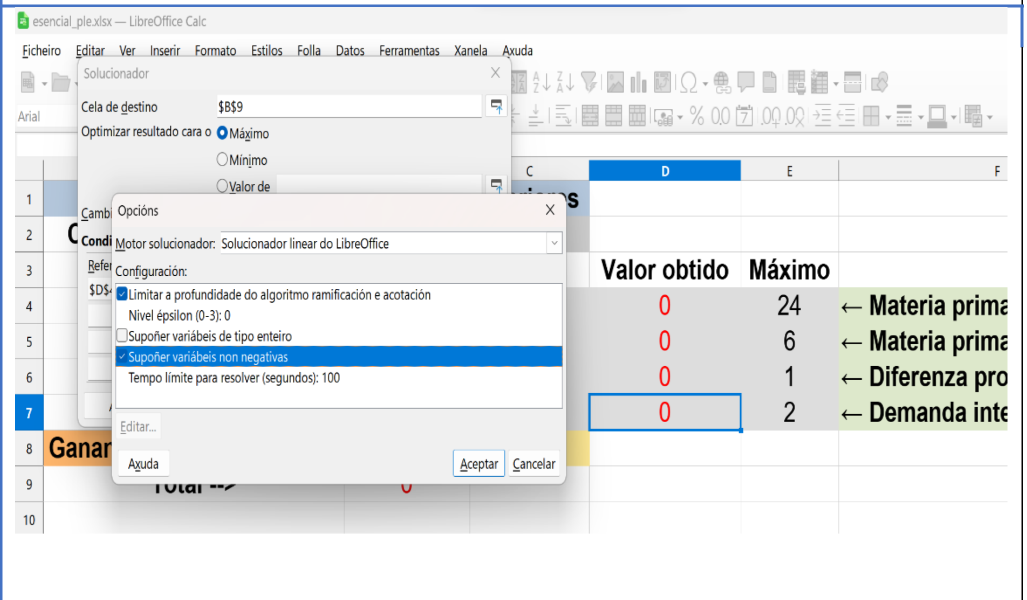
Paso 4: ferramenta *solucionador* e identificación de variables, tipo de obxectivo e restricións

A continuación, no botón “Ferramentas” do menú superior seleccionamos “*Solucionador*” e aparece un cadro de diálogo. No cadro, indícase que a cela de destino (obxectivo) atópase na cela B9, que o **tipo** de obxectivo é de Máximo e que as celas cambiantes (variables) son B2 e C2. En canto ás Condicións limitadoras (restricións), indicamos que as referencias de celas (funcións) están entre as celas D4 e D7, todas as restricións son de **tipo** “≤” e os valores (lados dereitos das restricións) atópanse entre as celas E4 e E7.



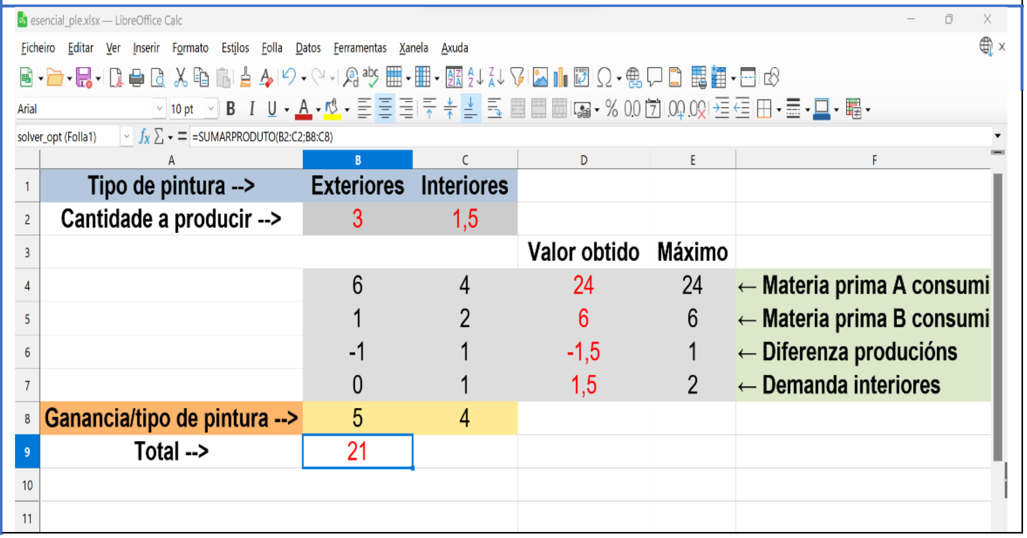
Paso 5: opcións da ferramenta *solver*

Xa só nos queda entrar en Opcións (parte inferior do cadro de diálogo), marcar en Supoñer variábeis non negativas e seleccionar un Solucionador linear.



Paso 6: visualización da solución óptima

Unha vez aceptadas as opcións seleccionadas e tras pulsar no botón de Solucionar e Manter o resultado, nas celas correspondentes ás variables do problema podemos ver a súa **solución óptima** (3 e 1,5), a cela da función obxectivo mostrará o seu valor óptimo (21) e nas celas dos lados esquerdos das restricións, os seus valores no óptimo.



Referencias:

[1] Hillier, F. & Lieberman, G. (2010). Introducción a la investigación de operaciones. McGraw-Hill.