

# INVESTIGACIÓN OPERATIVA SUPERIOR

*continuamos con scheduling*

Presencial

## ... algunas curiosidades

*a primera vista  
muy bueno,  
luego, casi  
irrecuperable*



deepseek

*el mejorcito  
por ahora*



grok



chatgpt

*mejores  
posibilidades que  
deepseek*

punto #1	✗ no encuentra solución	✓ ojo: no usa las cantidades	✗ error de uso de python
punto #2	✗ error de uso de python	✓ ya lo había resuelto en #1.	✗ error de uso de python
punto #3	✗ error de uso de python	✗ hay un error en uso de tipos	✗ arrastra errores
punto #4	✗ error de uso de python	✗ hay errores y los arrastra	✗ arrastra errores
punto #5	✓ pero arrastra errores	✓ pero arrastra errores	✓ pero arrastra errores
punto #6	✗ error de uso de python	✓ ligero error	✗ error de uso de python



# *limitaciones de licencia*

DEMO 0

## *scheduling* (parte II)

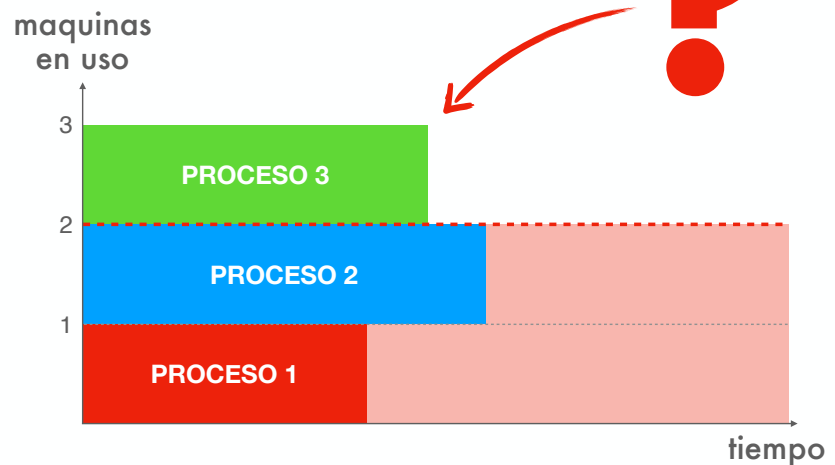
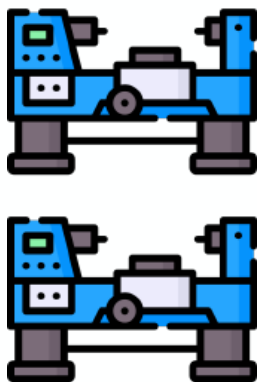




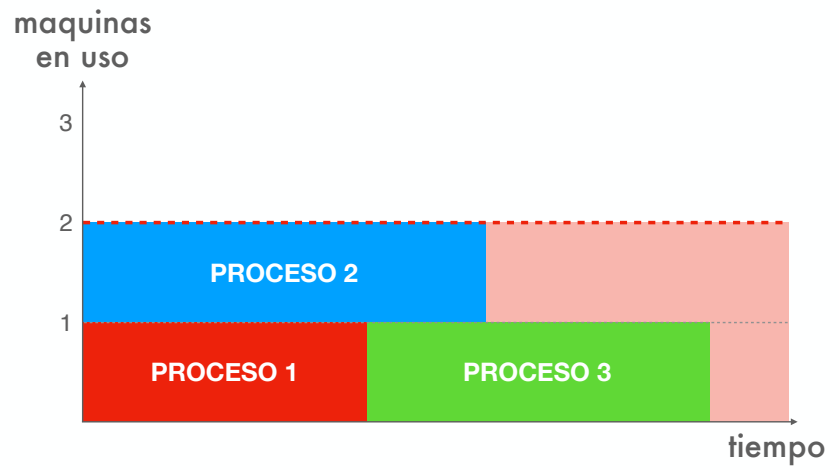
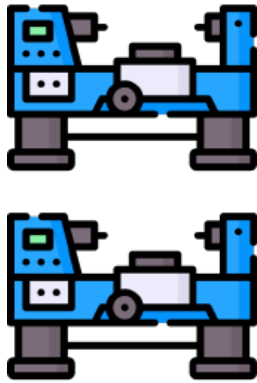


*una forma de  
trabajar con  
recursos es  
pensarlos como  
limitaciones en el  
tiempo*

*ejemplo:*



*ejemplo:*



`m.pulse(T1, valor)`

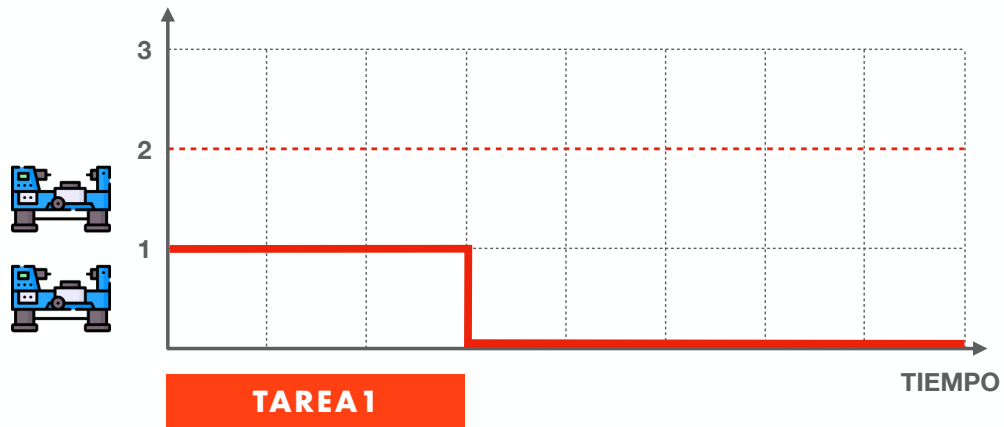
valor

**TAREA 1**

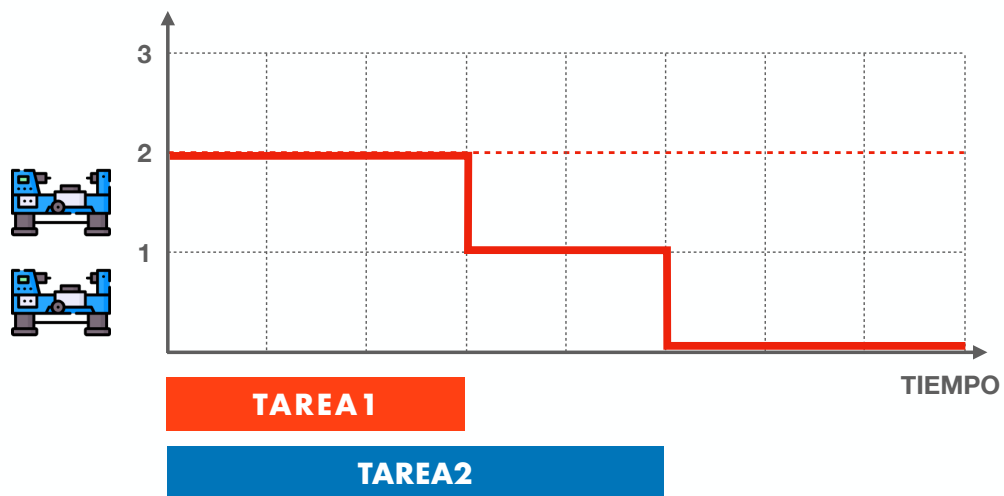
*para los ejemplos de esta clase:*

`m = CpoModel()`

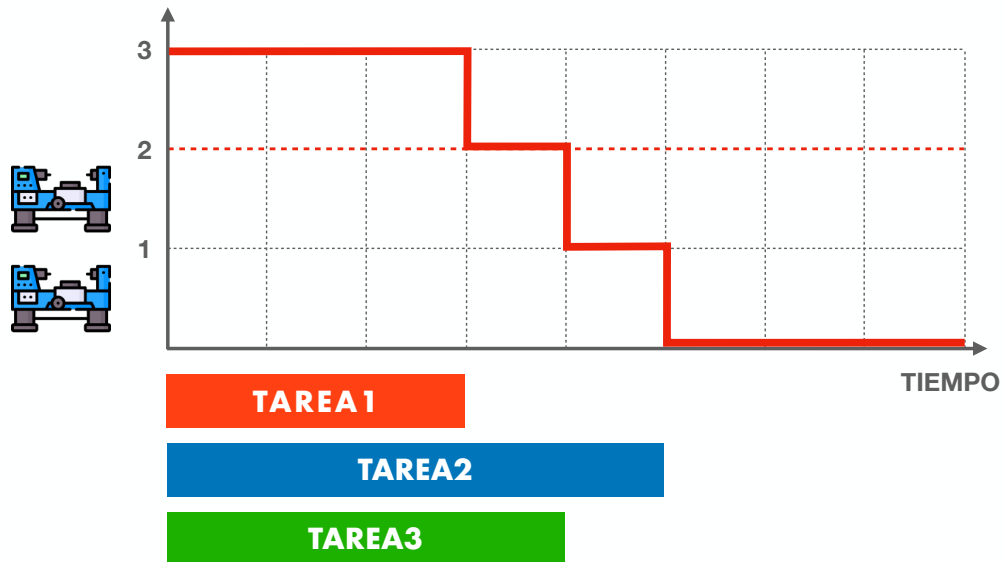
```
m.add(m.pulse(T1, 1) <= 2)
```



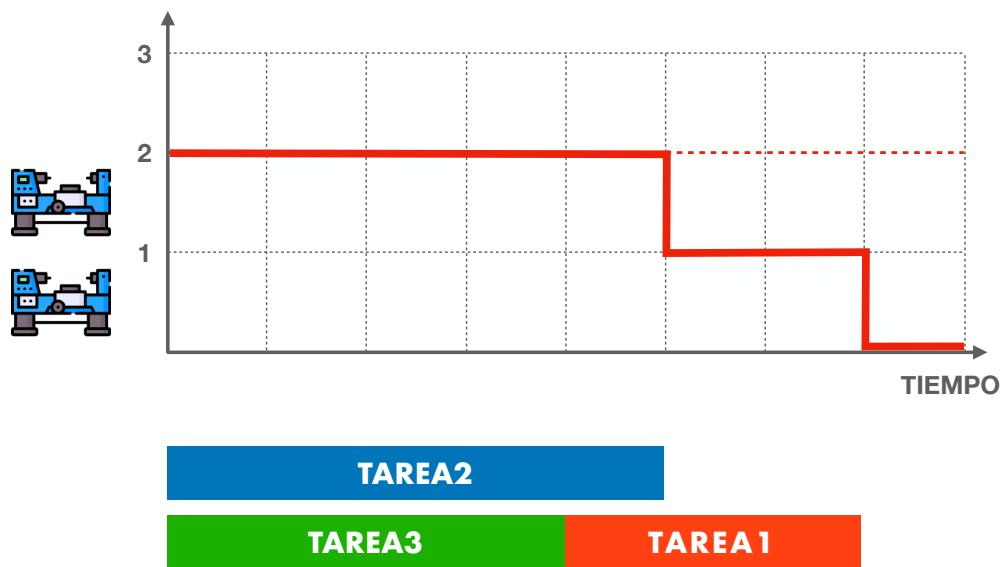
```
m.add(m.pulse(T1, 1) + m.pulse(T2, 1) <= 2)
```



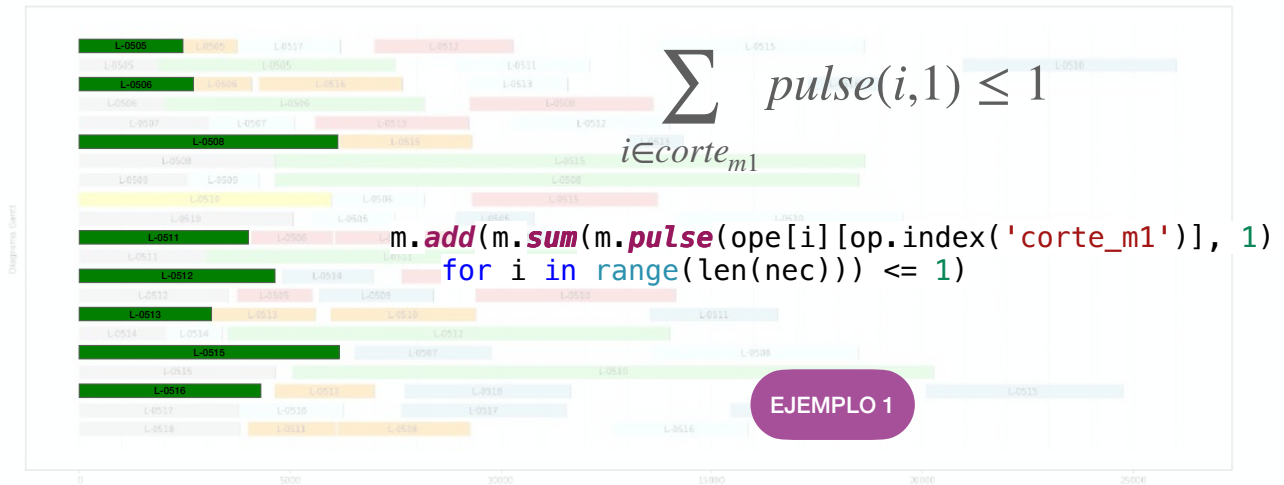
`m.add(m.pulse(T1, 1) + m.pulse(T2, 1) + m.pulse(T3, 1) <= 2)`



`m.add(m.pulse(T1, 1) + m.pulse(T2, 1) + m.pulse(T3, 1) <= 2)`



DEMO 1A



# disponibilidad por área

EJEMPLO 2



corte de metal



doblado



soldado



corte de madera



tapizado



pintura

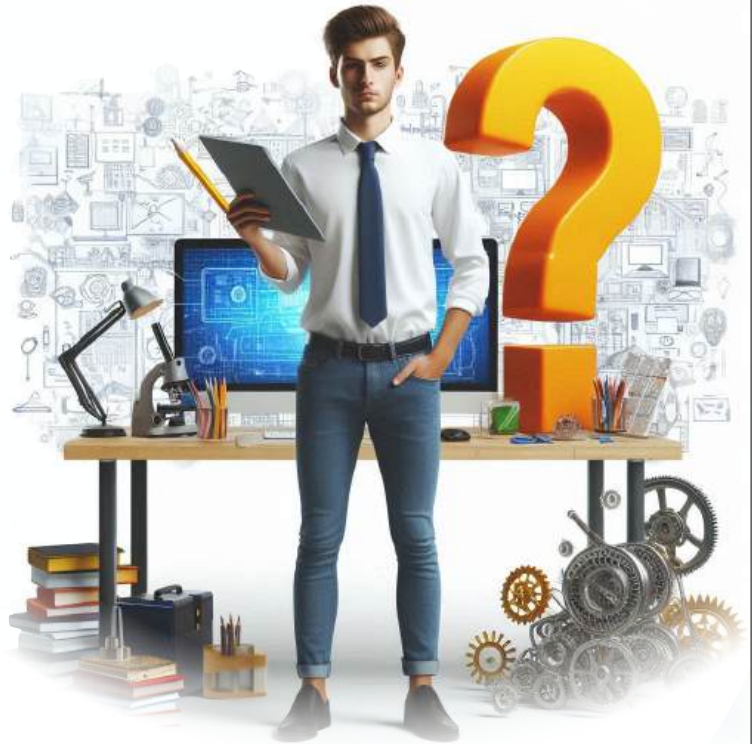


despacho





*¿preguntas?*



`m.step_at_start(T2, valor)`



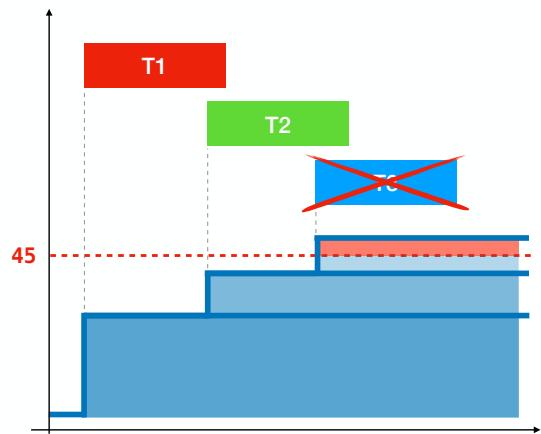
**T2**



```

...
m.add(
  m.step_at_start(T1, 28) +
  m.step_at_start(T2, 12) +
  m.step_at_start(T3, 10) <= 45
)
...

```



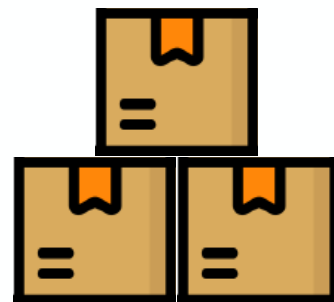
```

m.step_at_end(T3, valor)

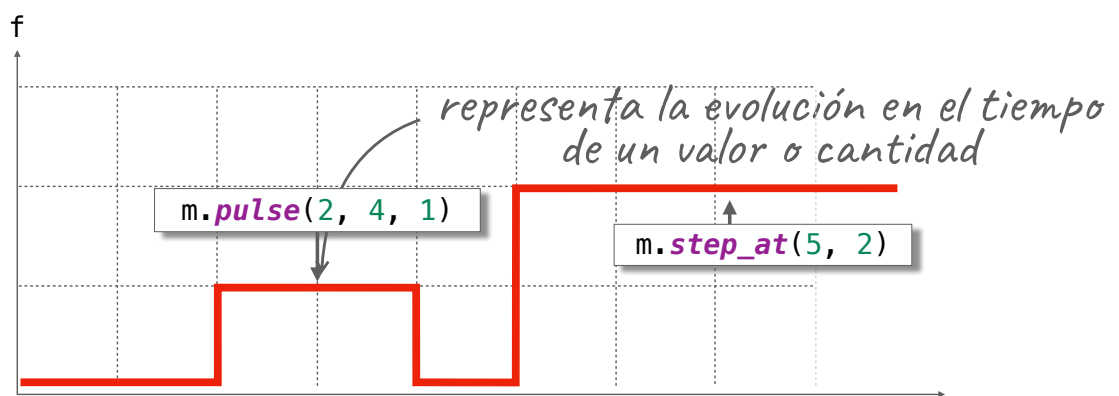
```

**TAREA 3**

veremos un  
ejemplo en un  
rato



# funciones escalonadas



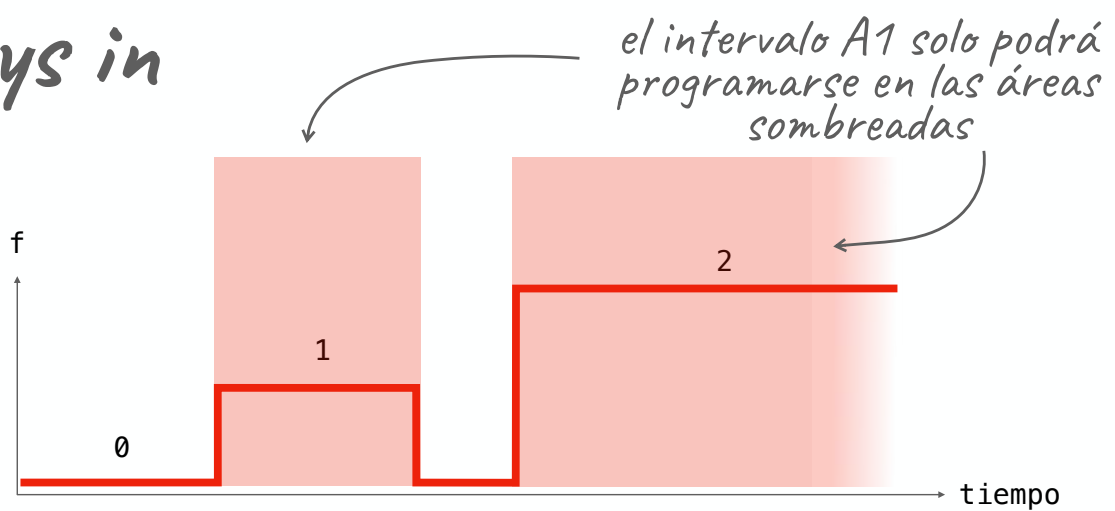
construcción manual

$f1 = m.pulse(2, 4, 1) + m.step\_at(5, 2)$

$f2 = m.pulse(A1, 1) + m.step\_at\_end(A2, 2)$

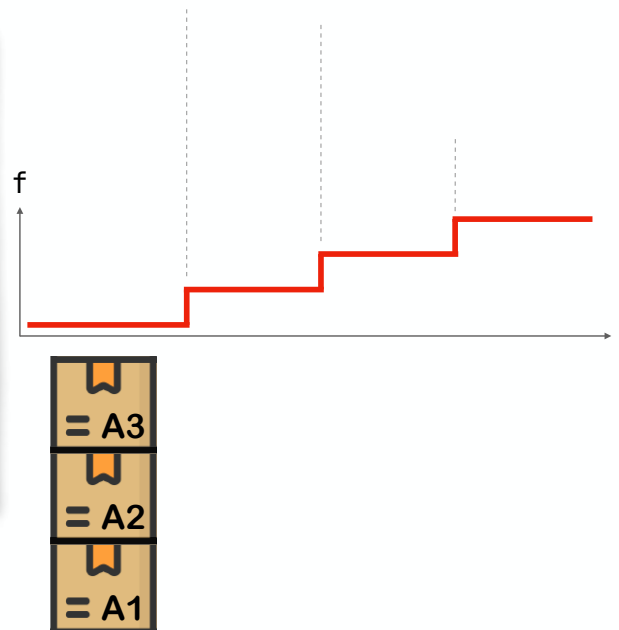
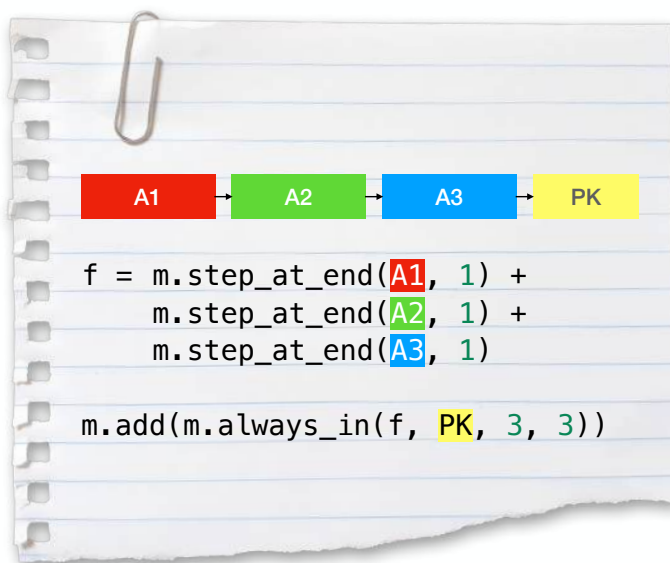
construcción con intervalos

*always in*

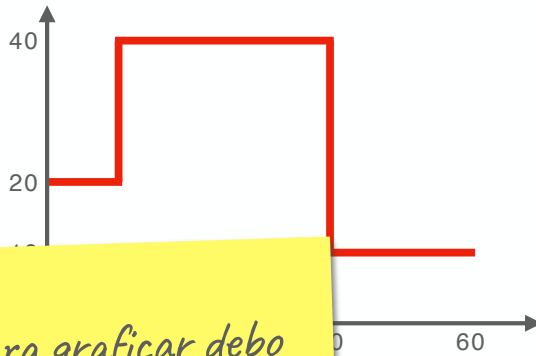


```
m.add(m.always_in(f, A1, 1, 2))
```

*veremos más de estas instrucciones luego ...*



*para graficar ...*



*para graficar debo  
crear una función  
tipo step*

```
f = CpoStepFunction()
f.add_value(0, 10, 20)
f.add_value(10, 40, 40)
f.add_value(40, 60, 40)
visu.function(
    segments = f,
    style = 'area',
    origin = 0,
    horizon = 30,
    color = 'red')
```

DEMO 1C

## CRONOGRAMA DE TRABAJO

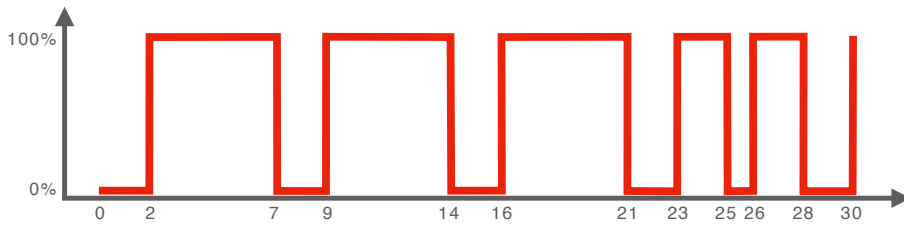
LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES	346400
	Ana	Luis	Pedro	Javier	Javier
	Ana	Luis	Pedro	Ana	Javier
	Ana	Luis	Pedro	Javier	Javier
	Ana	Luis	Ana	Javier	Javier
					Javier
					Javier
					Javier
					Javier
					Javier

*asociación entre  
funciones  
escalonadas e  
intervalos*

*intensidad  
calendarios y disponibilidad*



# ¿cómo lo defino?



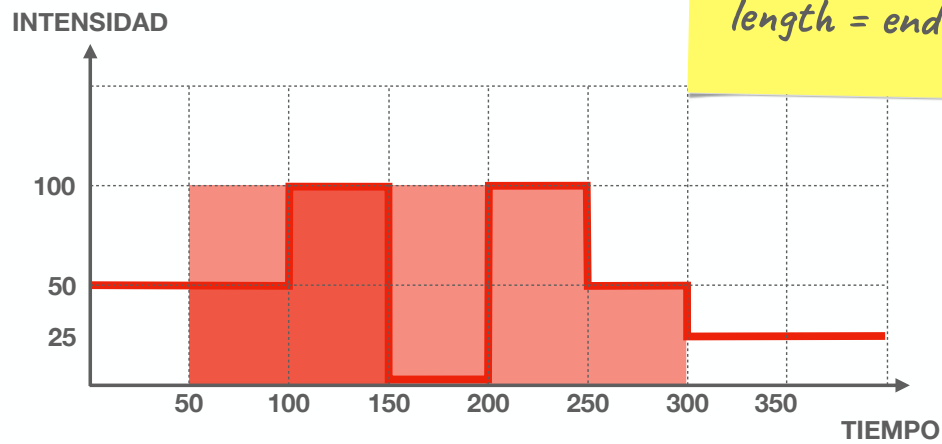
```
dic = CpoStepFunction()
dic.set_value(0, 31, 100)
for i in range(0, 31, 7):
    dic.set_value(i, i+2, 0)
for i in [8, 25]:
    dic.set_value(i, i+1, 0)
```

## intensity

```
t = m.interval_var(size=11, intensity=f, name = 't')
```

la tarea se realizará acorde  
a la función  $f$  (en %)

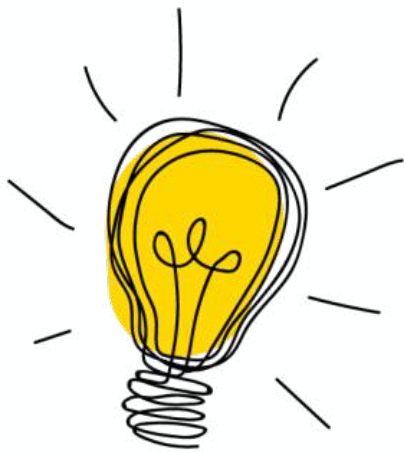
# size vs length



$length = end - inicio$

size = 150

length 50 250



TIP!

prohibe el comienzo de la tarea cuando  $f=0$

`m.add(m.forbid_start(t, f))`

prohibe el fin de la tarea cuando  $f=0$

`m.add(m.forbid_end(t, f))`

`m.add(m.forbid_extent(t, f))`

prohibe la tarea  $f=0$

DEMO 1D



# calendario operarios

esto indica que  
posiblemente nuestra  
udm sea hora

EJEMPLO 3



*lunes*

*martes*

*miércoles*

*jueves*

*viernes*

8 a 17hs

8 a 17hs

8 a 17hs

8 a 17hs

8 a 17hs

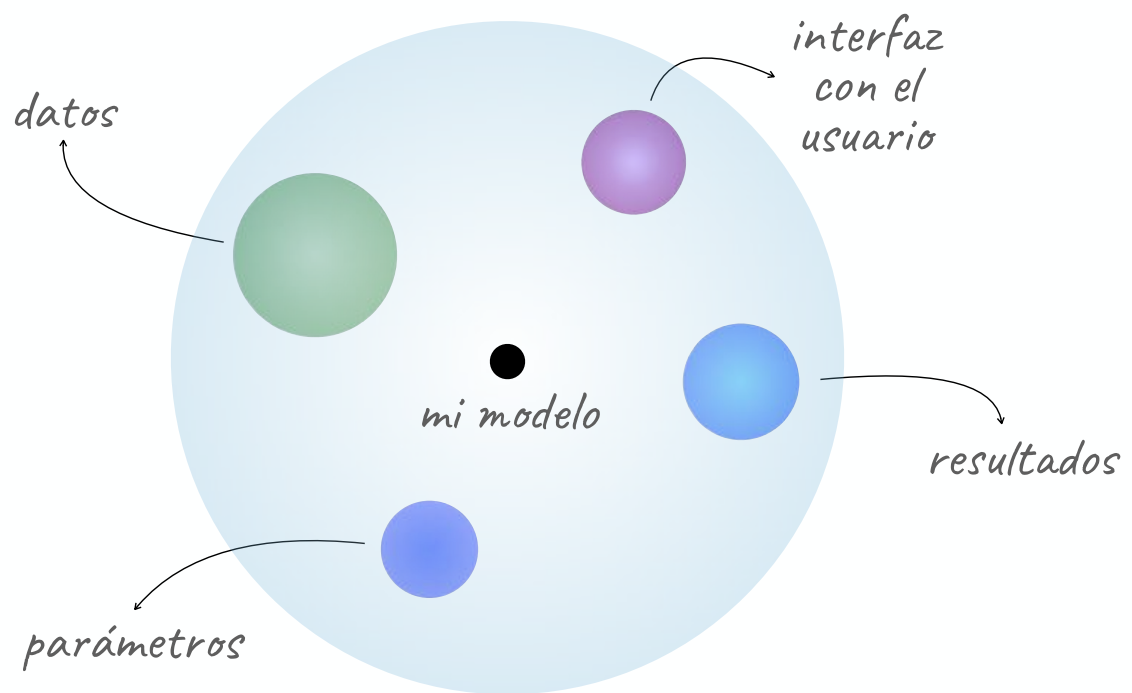


todos los días cuentan con una hora de almuerzo

*¿preguntas?*







presentación  
de resultados



*genera confianza  
permitiendo evaluar y  
comparar los  
resultados*



*es una muy buena  
herramienta de  
ventas*

## importancia



*permite corregir  
y mejorar el  
modelo*



*facilita la  
evaluación de la  
performance del  
modelo*

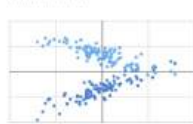


# Google Charts

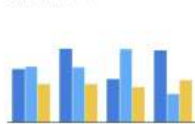
Geo Chart



Scatter Chart



Column Chart



Pie Chart



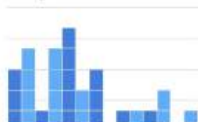
Bubble Chart



Donut Chart



Histogram



Bar Chart



Combo Chart



Org Chart



Treemap



Table

	Name	Salary	Full Time
1	Marie	\$26,700	✓
2	Albert	\$25,200	✗
3	Enrico	\$25,700	✓
4	John	\$16,400	✓

Area Chart



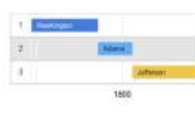
Stepped Area Chart



Line Chart



Timeline



Gauge



Candlestick Chart



puede ser creado por un diseñador gráfico

dentro de un marco html

se generan en javascript



o si hacemos algo de magia ... en un pdf

se que genera un archivo html



se visualizan en un navegador

# javascript

carga la biblioteca

llamado a la función que dibuja

función que dibuja

carga los datos

carga las opciones

instancia el gráfico

lo dibuja

```
google.charts.load('current', {'packages':['?']});
```

```
google.charts.setOnLoadCallback(drawChart);
```

```
function drawChart() {
```

```
    var datos = google.visualization.arrayToDataTable([?]);
```

```
    var opciones = {?};
```

```
    var grafico = new google.visualization.PieChart(  
        document.getElementById('?');
```

```
    grafico.draw(datos, opciones);
```

```
}
```

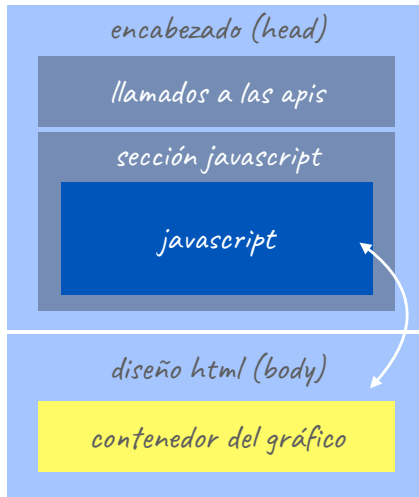
la biblioteca que corresponde según el gráfico

depende del tipo de gráfico, se requiere ver la documentación

diccionario con opciones

el contenedor que definimos en el html

# html (básico)



```
<html>
<head>
  <script type="text/javascript"
    src="https://www.gstatic.com/charts/loader.js"></script>
  <script type="text/javascript">
```

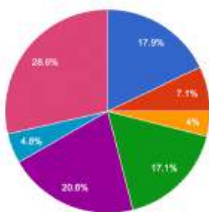
javascript

</script>
</head>
<body>
 <div id="piechart" style="width: 900px; height: 500px;"></div>
</body>
</html>

explicado en el slide anterior

id utilizado en el javascript

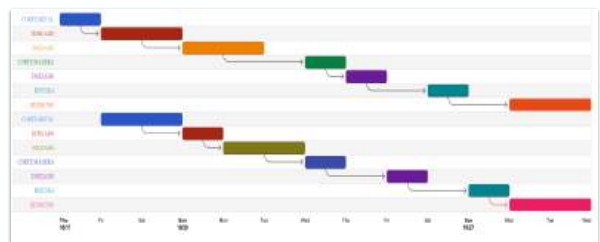
## ejemplos



EJEMPLO 4



EJEMPLO 5



EJEMPLO 6

hay muchos gráficos más ...  
y muchas bibliotecas diferentes ...

genero el html como un  
string

lo grabo como un  
archivo html



*¿cómo integrarlo con python?*

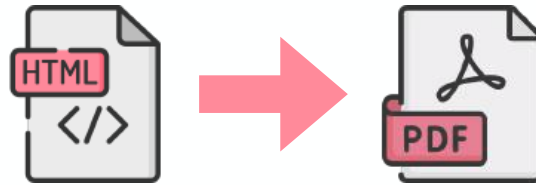
o ... lo visualizo en  
python con un  
interprete de código  
html

DEMO 3A



*generación de  
reportes pdf*

# pyhtml2pdf

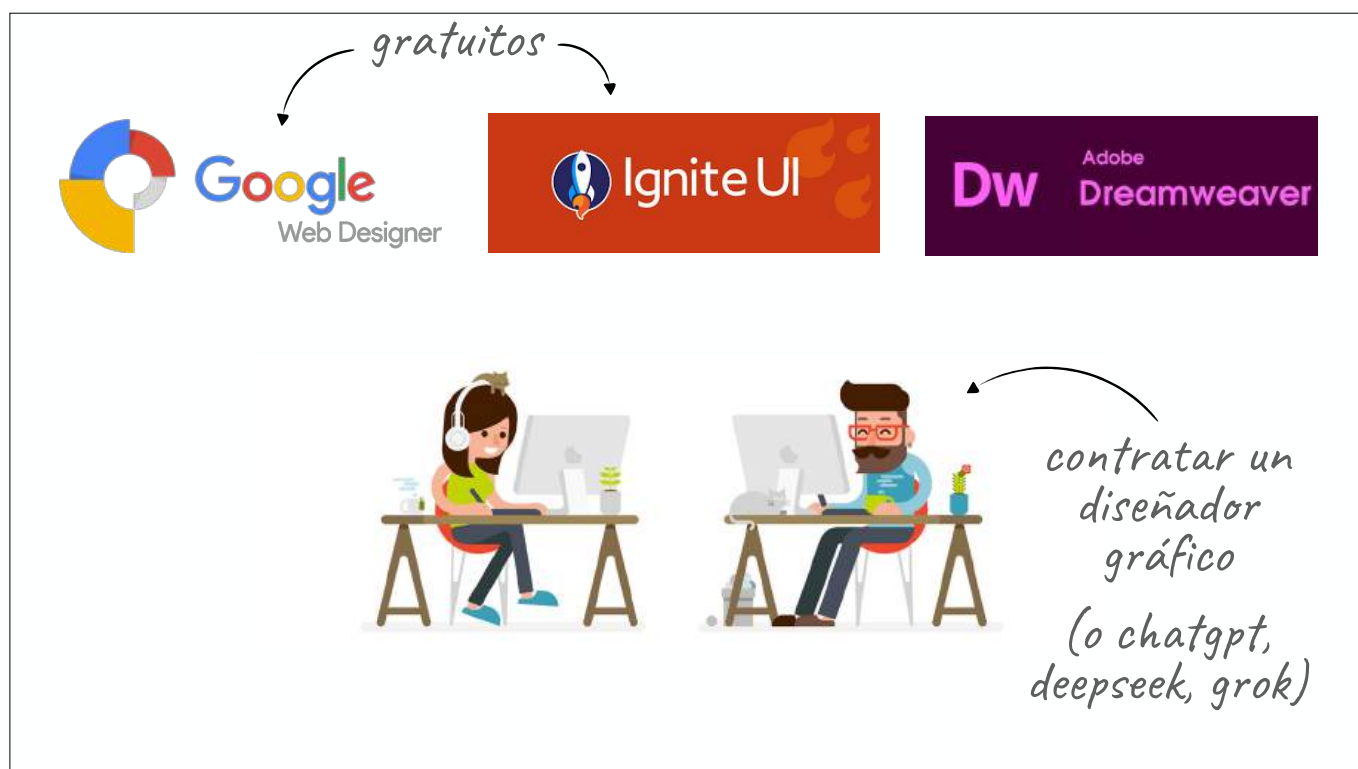


```
import os  
from pyhtml2pdf import converter  
path = os.path.abspath('reporte.html')  
converter.convert(f'file:/// {path}', 'reporte.pdf')
```

DEMO 3B



*finalmente el  
diseño ...*

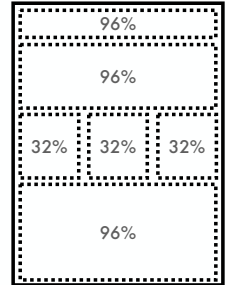


# algunas practicas optativas ...

tip: pueden pintar de diferentes colores los DIV para diferenciarlos



(1) generar una estructura html principalmente con `<div id="?"></div>` como la que se propone a la derecha.



(2) probar cargar distintos gráficos en el html.

(3) incorporar todo a python y transformarlo a pdf.

## bibliografía y otros ...

### [Python] bibliotecas:

<https://pypi.org/project/ipython/>  
<https://docs.python.org/3/library/os.html>  
<https://pypi.org/project/html/>  
<https://pypi.org/project/pyhtml2pdf/>

### [Diseño] links:

<https://designer.igniteui.com/>  
<https://www.adobe.com/>  
<https://webdesigner.withgoogle.com/>

para la próxima clase ...



Español



Español





*próxima clase:*

## *clase virtual*

*scheduling 3, introducción a base de  
datos, otras bibliotecas de CP y  
presentación del problema  
integrador*

# INVESTIGACIÓN OPERATIVA SUPERIOR

*¡muchas gracias!*