

# INVESTIGACIÓN OPERATIVA SUPERIOR

*nuevo módulo: manufactura*

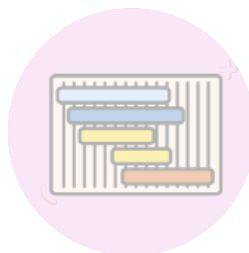
Virtual

## *bienvenidos al tercer módulo*

*proyectos de  
investigación  
operativa*

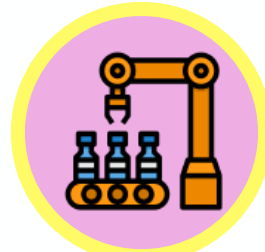
*estadística con  
python*

*visión por  
computadora*



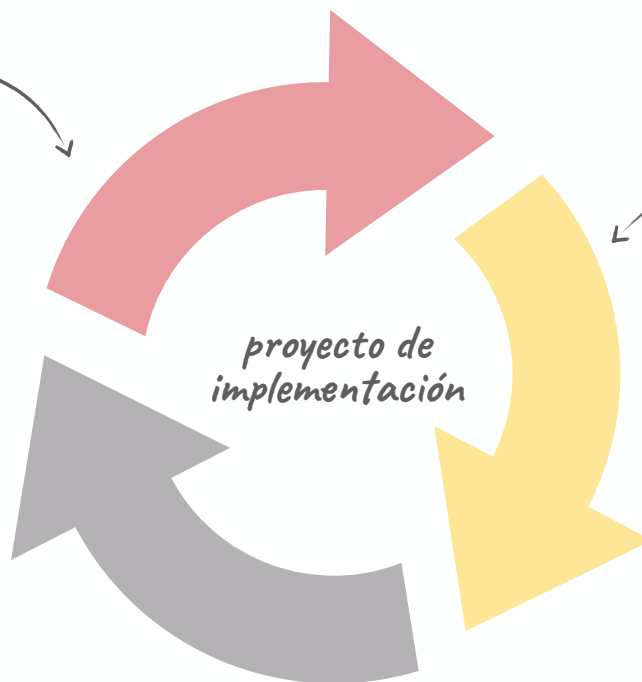
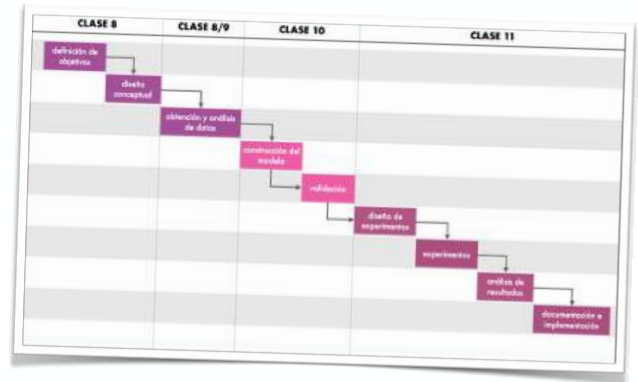
*gemelos digitales  
(digital twins)*

*modelos de simulación  
con python*

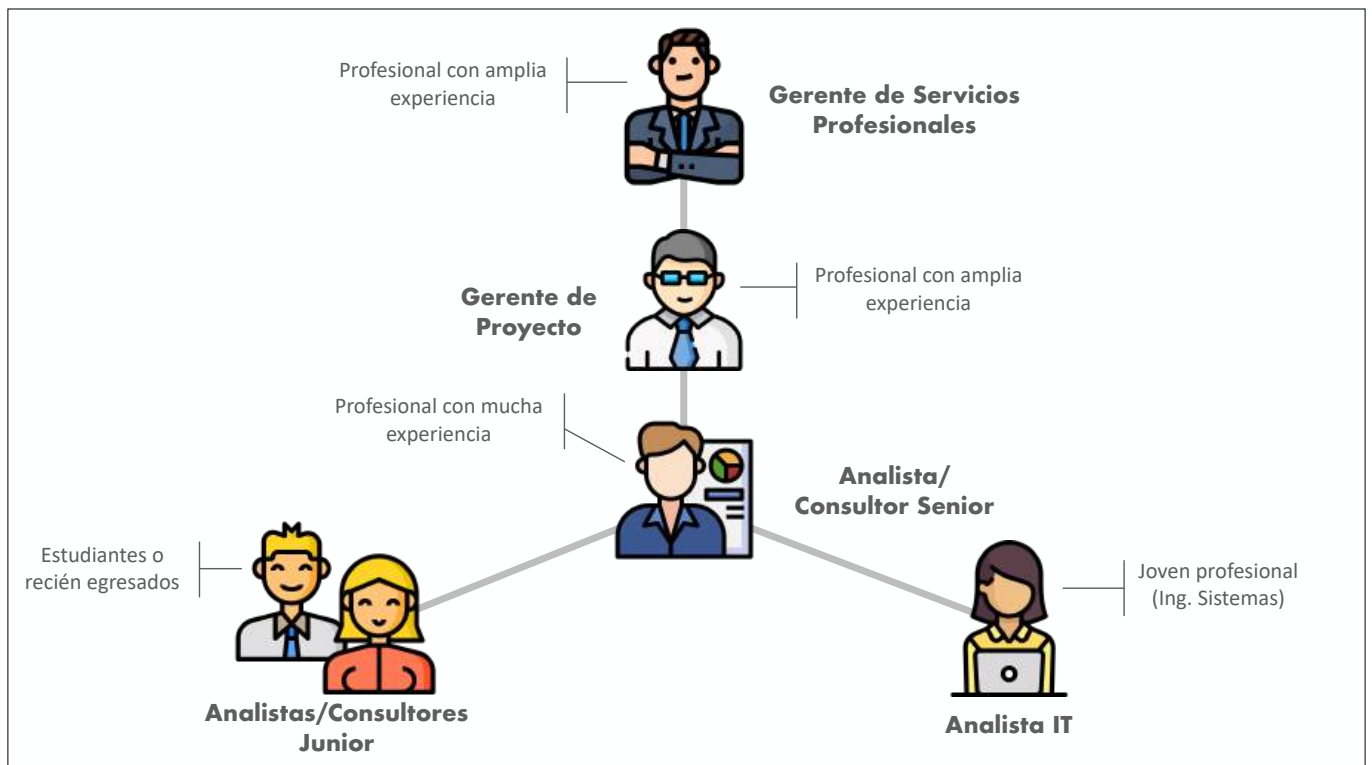


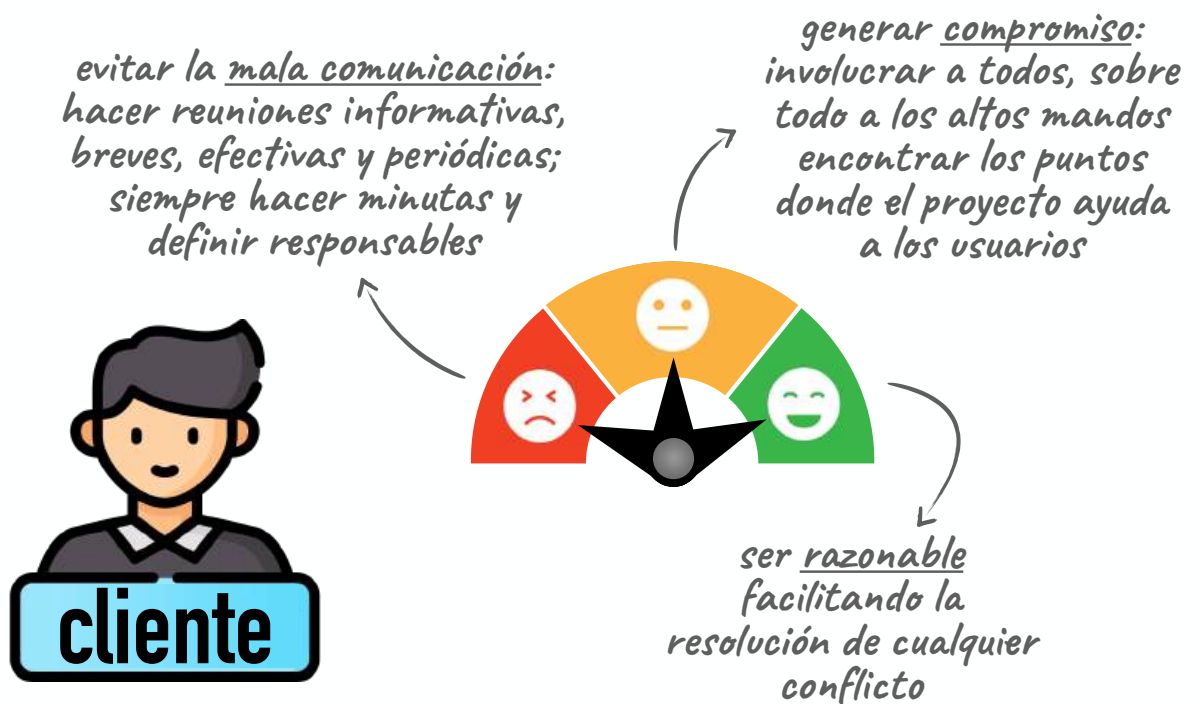
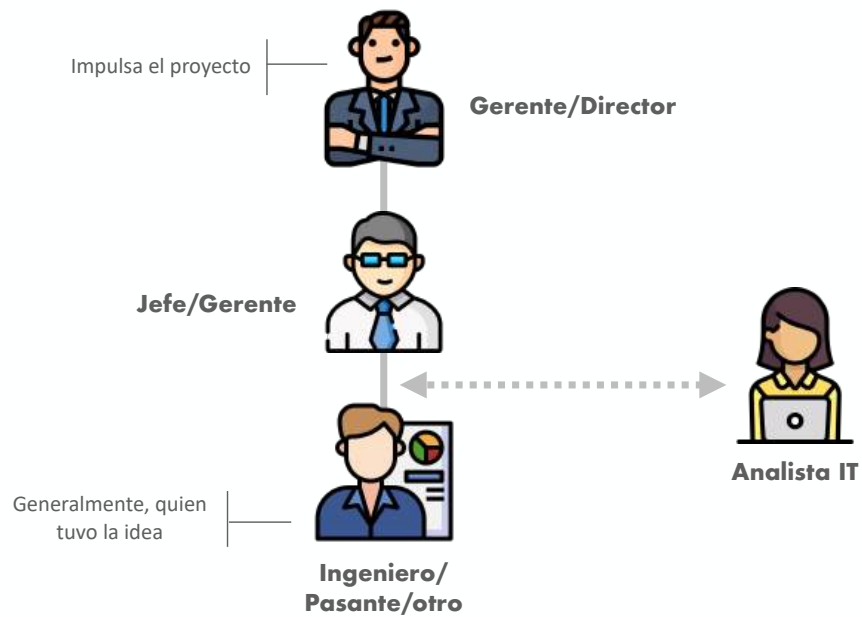
*manufactura*

# proyectos en investigación operativa

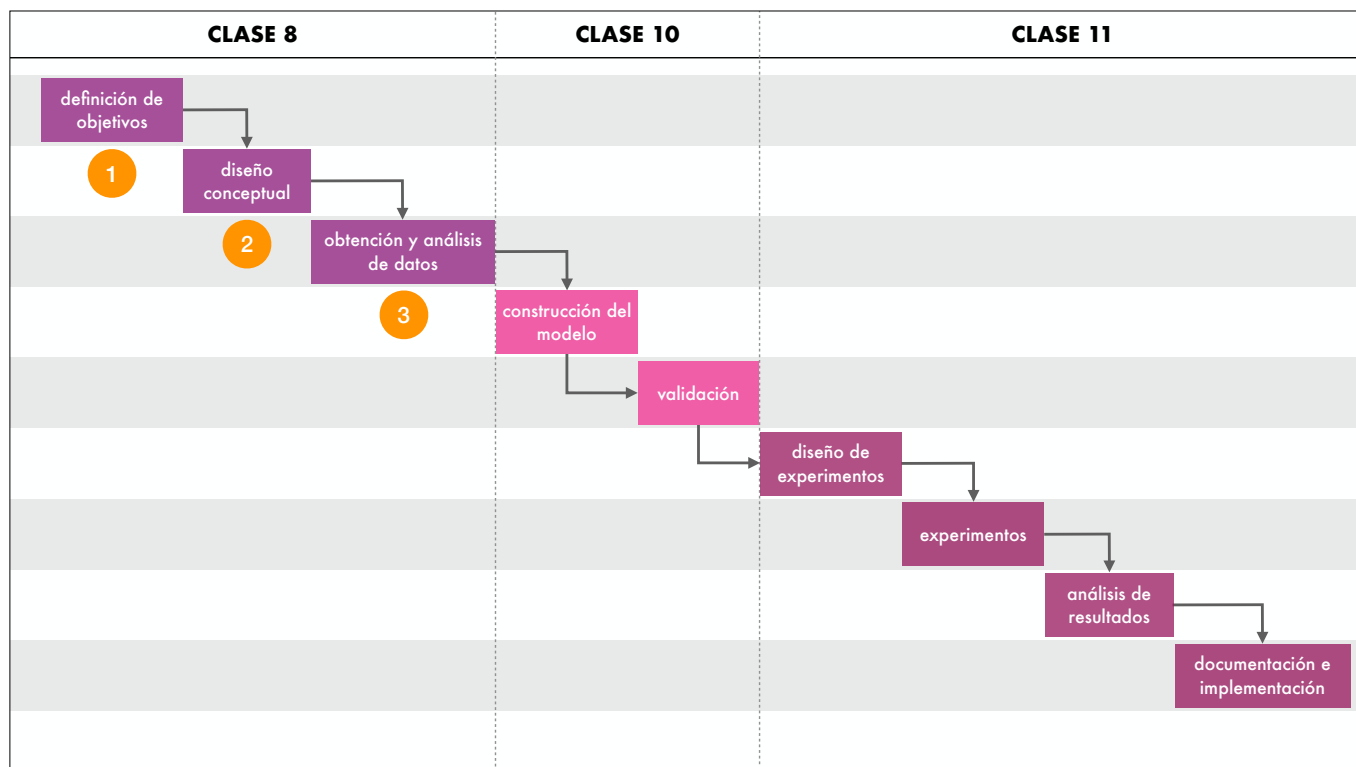


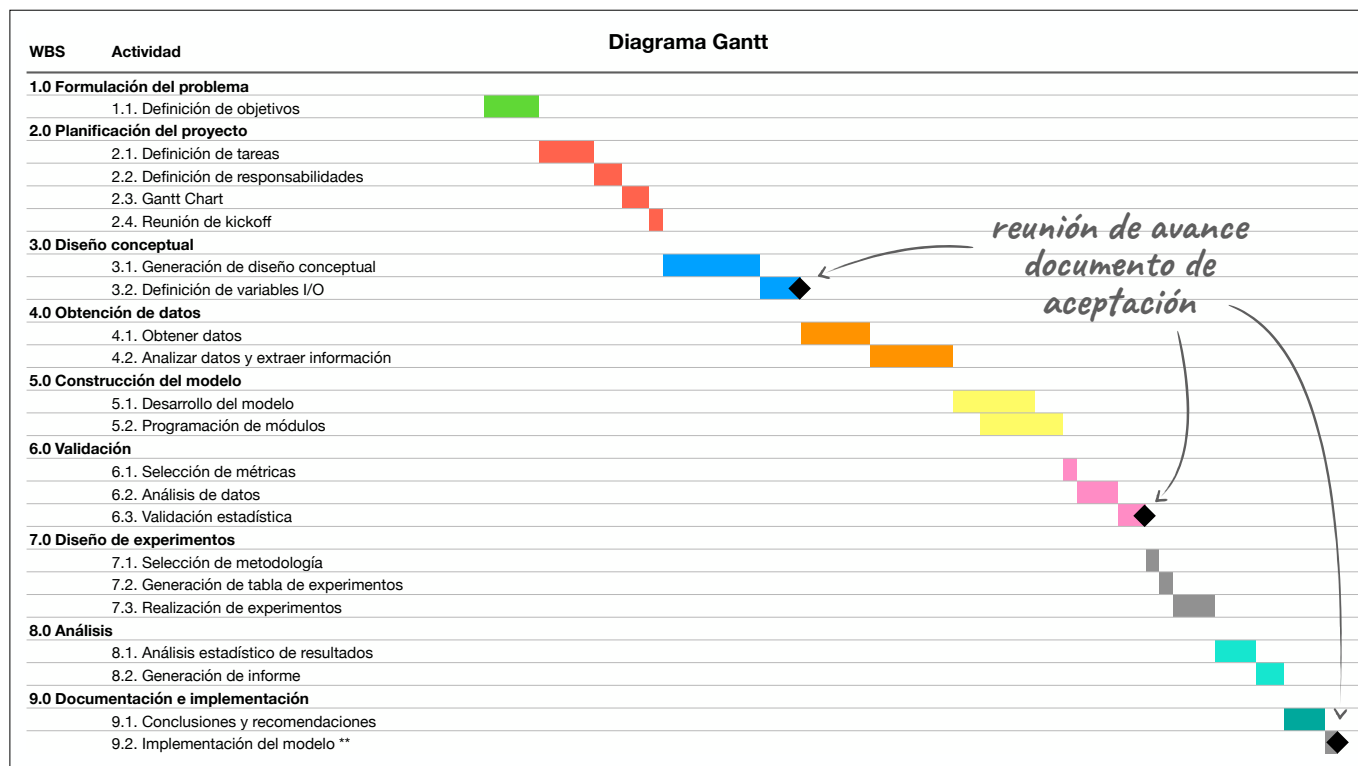
*el equipo*





# la planificación



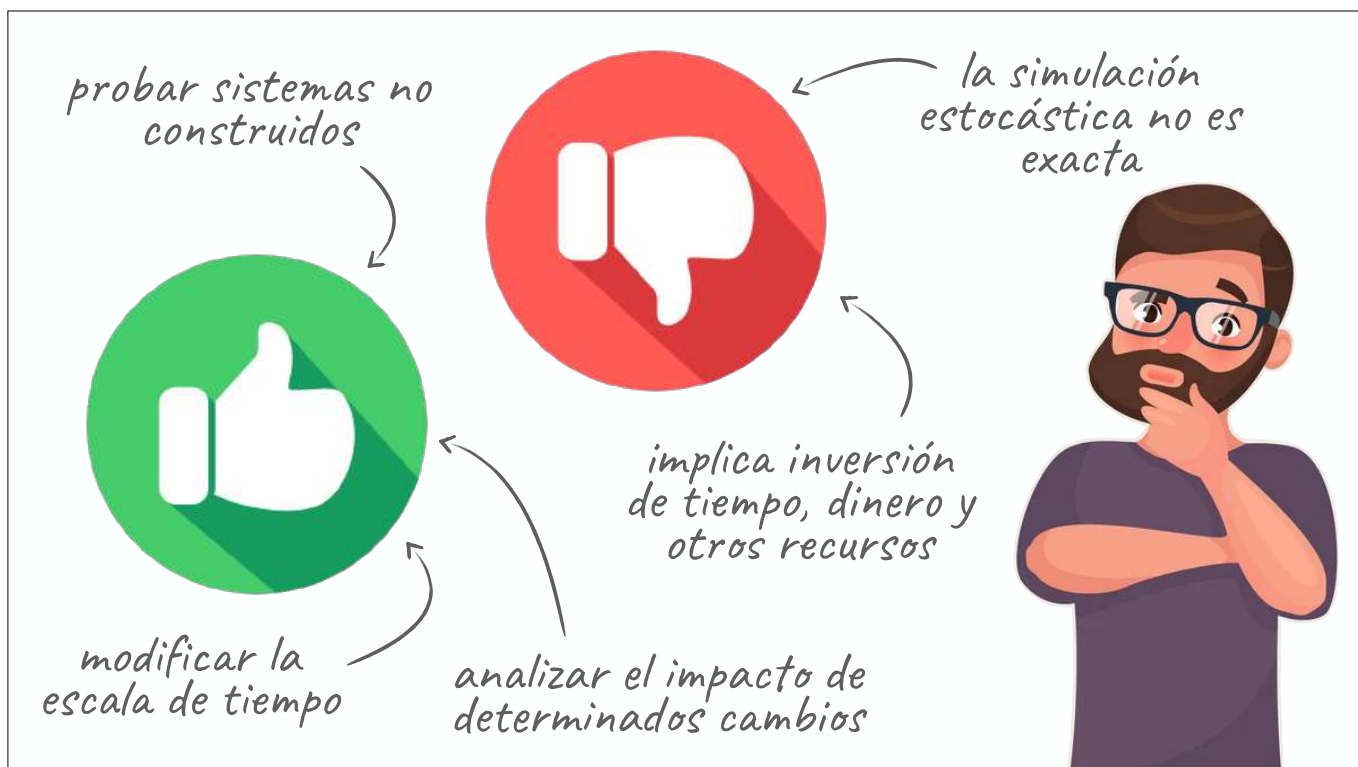
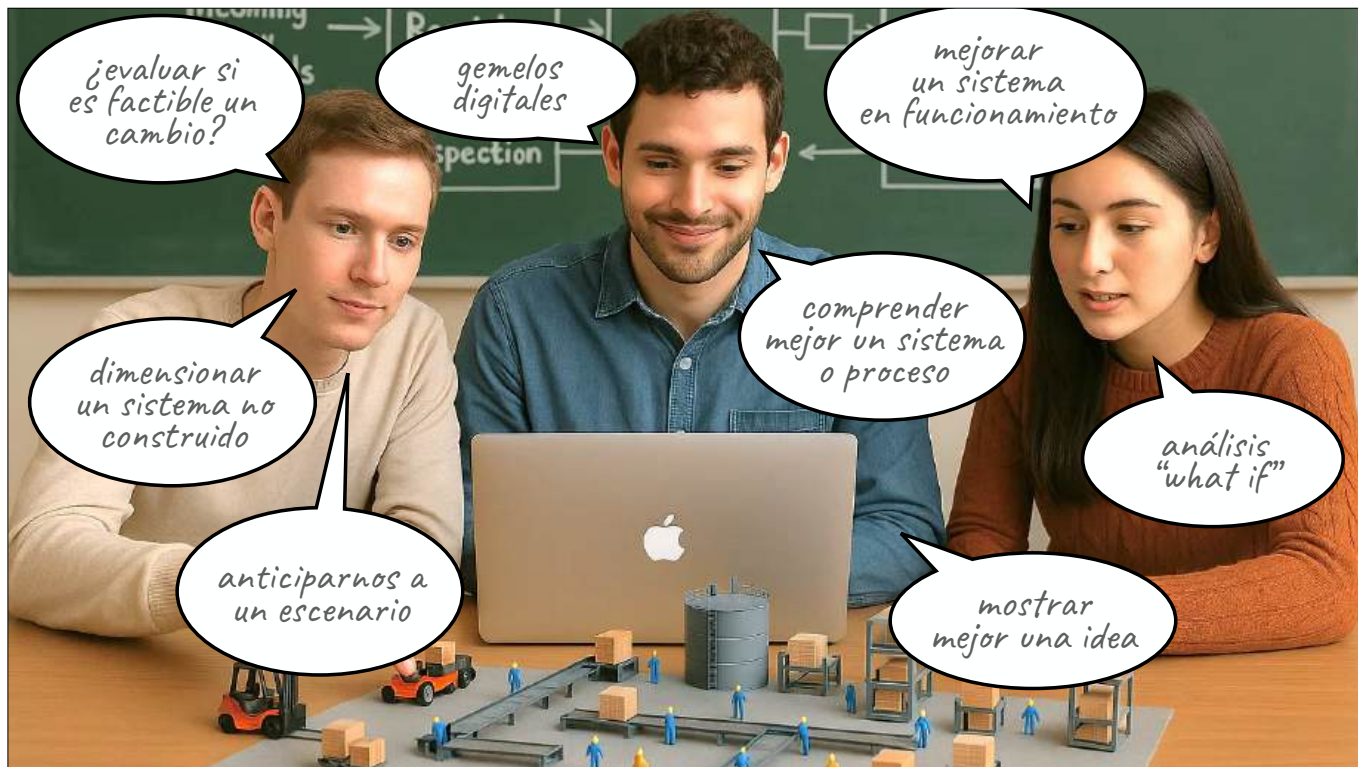


WBS	Actividad	Gerente de Servicios Profesionales	Gerente de proyecto	Analista 1	Analista 2	Analista IT
<b>1.0 Formulación del problema</b>						
1.1.	Definición de objetivos	A	E			
<b>2.0 Planificación del proyecto</b>						
2.1.	Definición de tareas	A	E			
2.2.	Definición de responsabilidades	A	E			
2.3.	Gantt Chart	A	E			
2.4.	Reunión de kickoff	P	E	P	P	P
<b>3.0 Diseño conceptual</b>						
3.1.	Generación de diseño conceptual		R	E	E	
3.2.	Definición de variables I/O		R	E	E	
<b>4.0 Obtención de datos</b>						
4.1.	Obtener datos		R	S	E	E
4.2.	Analizar datos y extraer información		R	S	E	
<b>5.0 Construcción del modelo</b>						
5.1.	Desarrollo del modelo		R	S	E	
5.2.	Programación de módulos		R	S		E
<b>6.0 Validación</b>						
6.1.	Selección de métricas		R	S	E	
6.2.	Análisis de datos		R	S		
6.3.	Validación estadística		R	S		
<b>7.0 Diseño de experimentos</b>						
7.1.	Selección de metodología		R	S		
7.2.	Generación de tabla de experimentos		R	S		
7.3.	Realización de experimentos		R	S		
<b>8.0 Análisis</b>						
8.1.	Análisis estadístico de resultados		R	S		
8.2.	Generación de informe	A	R	S		
<b>9.0 Documentación e implementación</b>						
9.1.	Conclusiones y recomendaciones		R/E	S/E		
9.2.	Implementación del modelo **	A	R	E	E	E

■ Aprueba  
■ Responsable primario  
■ Ejecuta  
■ Responsable secundario  
■ Presencia



*¿qué es simulación?*

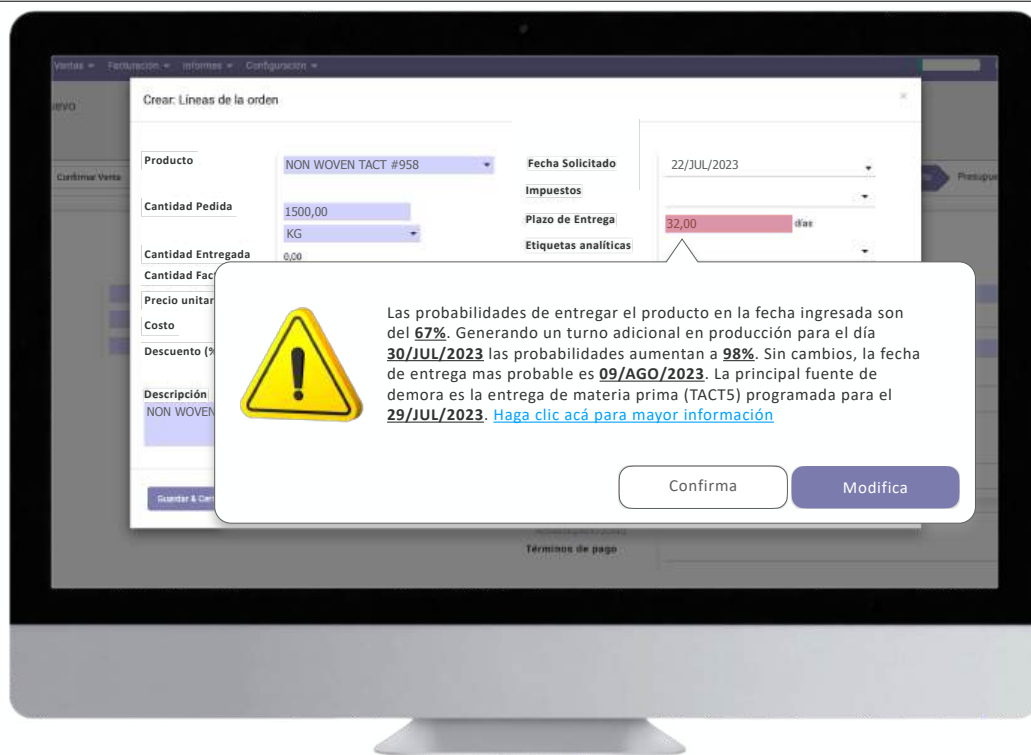






*tipos de  
implementación*





*comencemos un  
proyecto de  
simulación ...*



corte de  
metal



doblado



soldado



corte de  
madera



tapizado



pintura



despacho

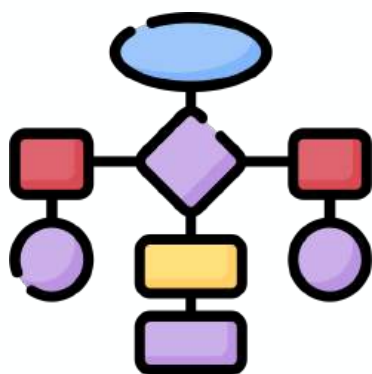
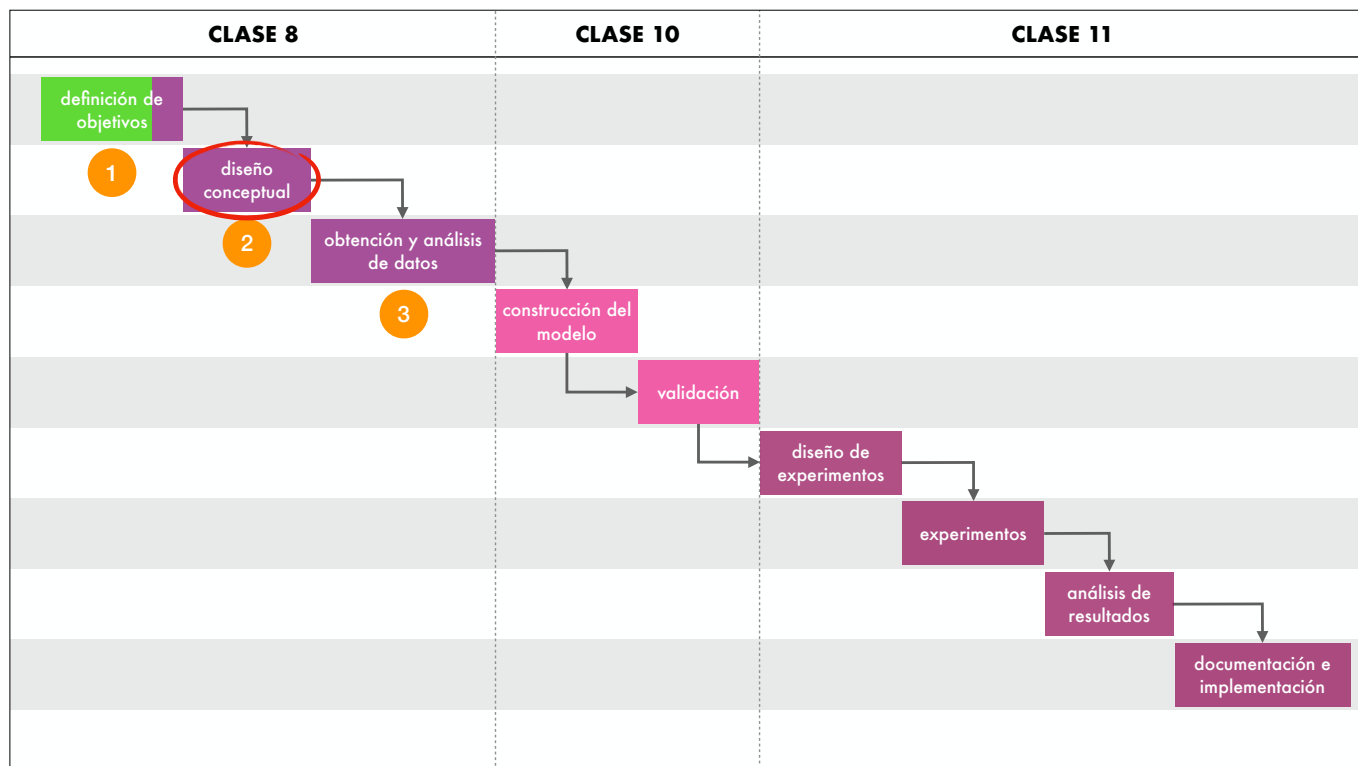
1 **objetivo**: determinar la probabilidad de cumplir con un programa de producción en el tiempo requerido, alertar posibles desvíos y riesgos.

*¿es correcto?*

*el objetivo debe definir  
perfectamente los límites  
(alcance) del proyecto*



*escribir el objetivo  
correcto para la  
próxima clase*

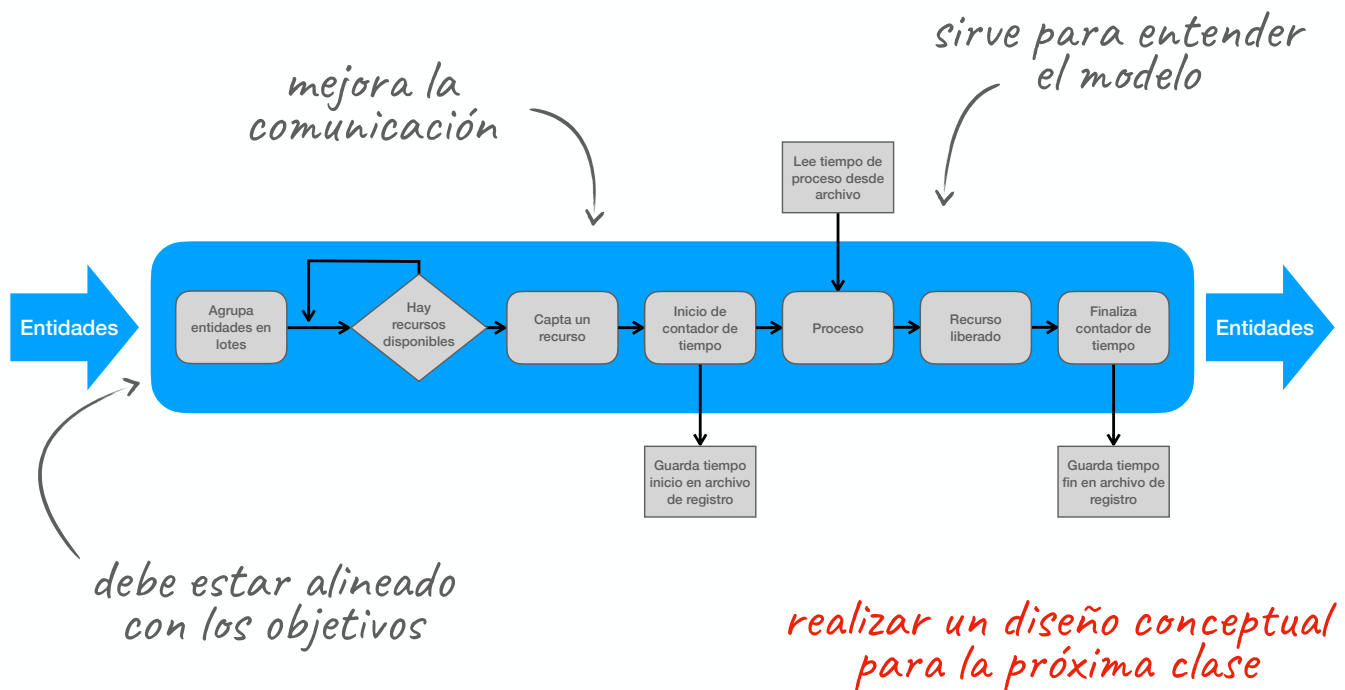


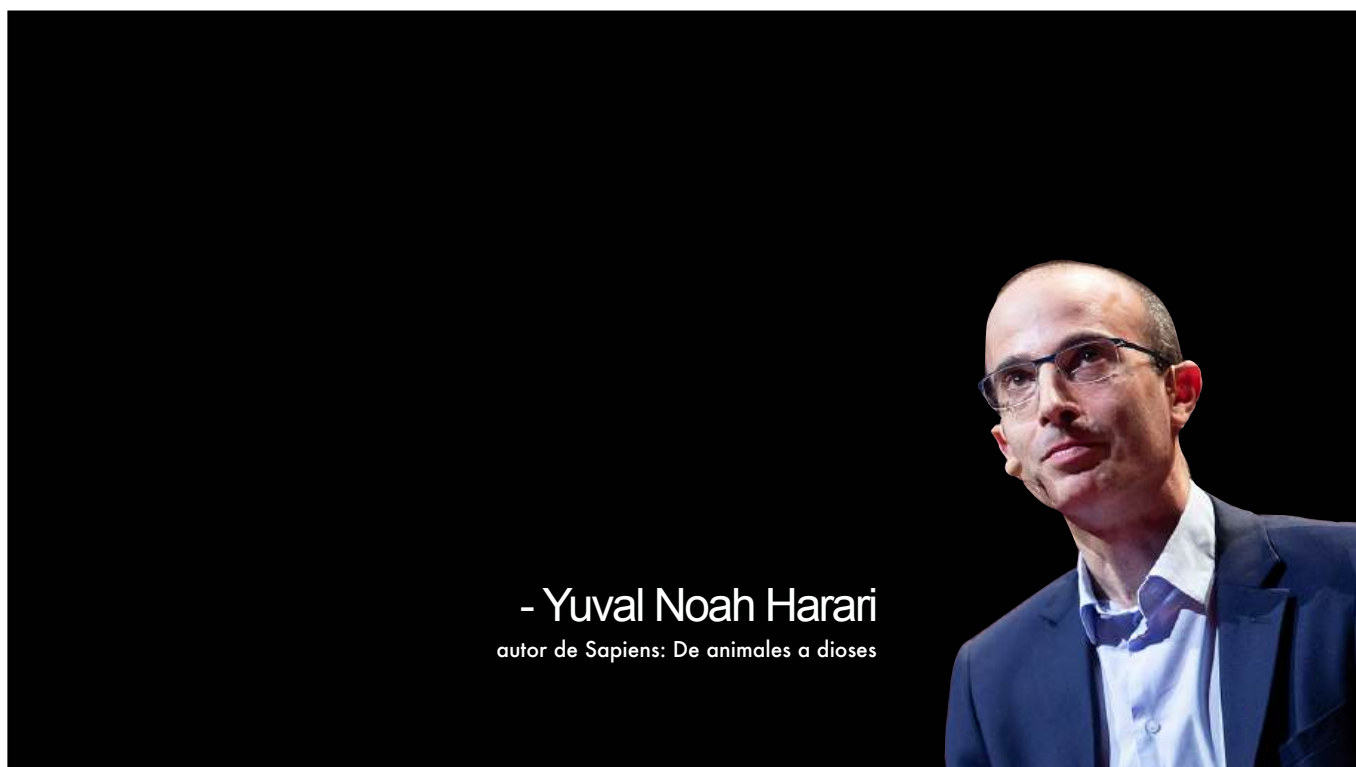
*diseño conceptual*



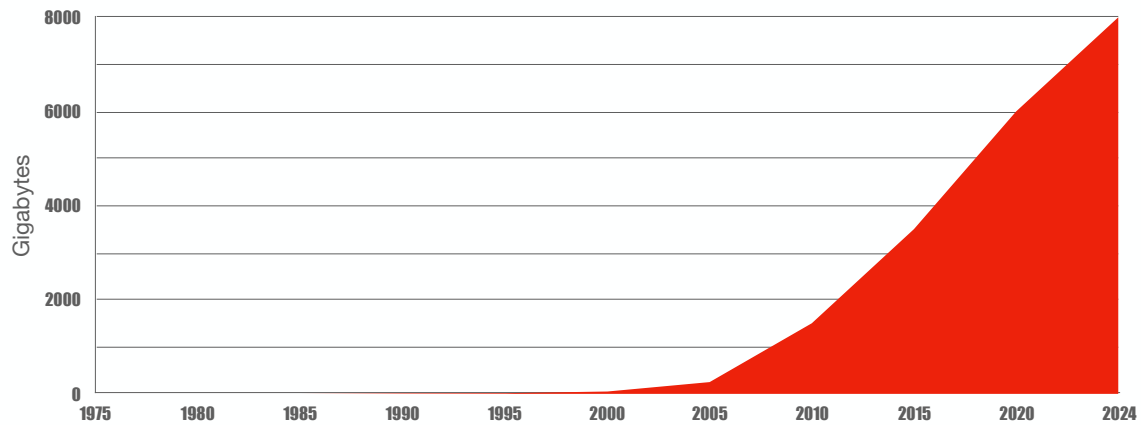
← entender

comunicar →





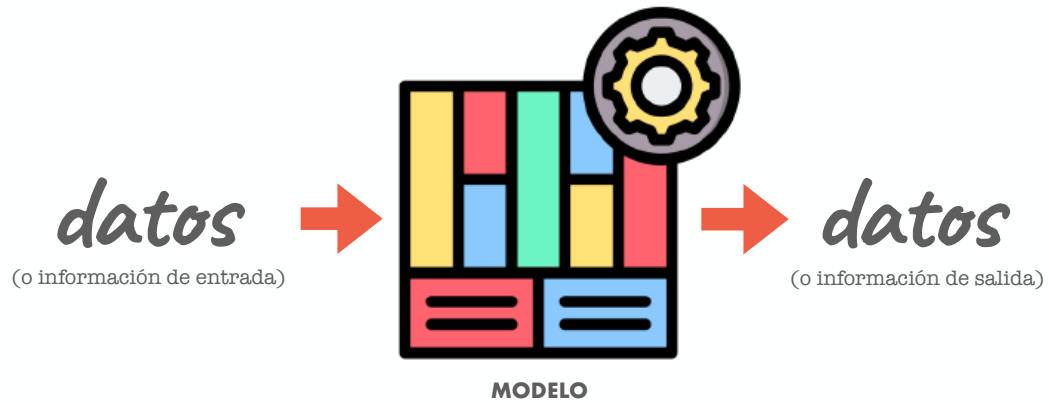
# crecimiento de capacidad de almacenamiento



cantidad de gigabytes que se pueden comprar por 100 usd

# crecimiento de capacidad de almacenamiento





*“un modelo no puede ser  
mejor que los datos”*





extremadamente  
importante

↓  
obtención y  
análisis de  
datos

↖  
muy importante

automatizar,  
utilizar históricos,  
muestras grandes,  
base de datos

¿qué hago con los  
datos "basura" o  
faltantes? ¿criterio!

pruebas estadísticas

obtención  
de datos

análisis de  
datos

limpieza de  
datos

ajuste

validación

búsqueda de  
correlaciones

entender los datos  
que se obtuvieron lo  
mejor posible

identificación de la  
distribución y sus  
parámetros

muy importante a fin  
de que el modelo se  
ajuste a la realidad

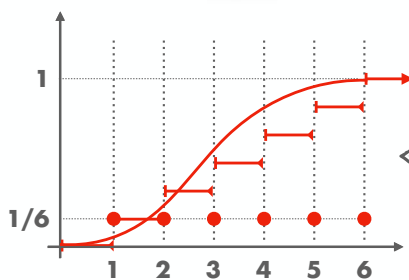
# un breve repaso de estadística

... mitos y realidades



## variable aleatoria

$X =$    $\leftarrow$  ~~asumo que representa el tiempo de~~  
~~experimento aleatorio~~ ~~entre dos ciudades~~

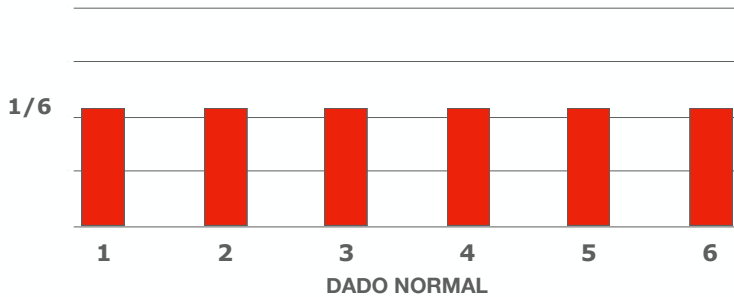
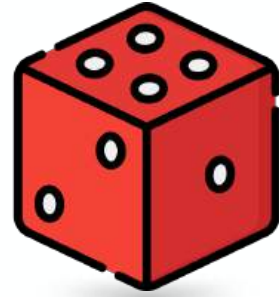


$\leftarrow$  función de distribución (cdf)

$$F_X(x) = \mathbb{P}(X \leq x)$$

# función de probabilidad

probability mass function (pmf)

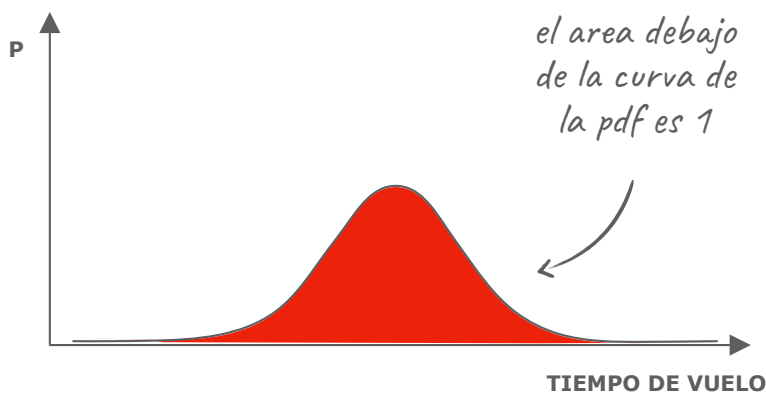
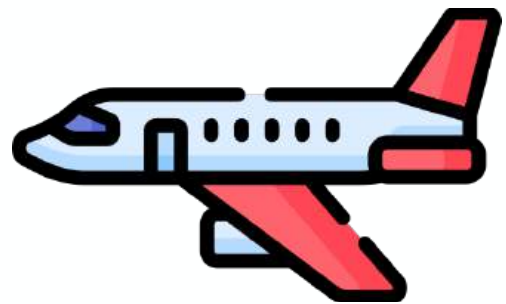


$$p_X(x) = \mathbb{P}(X = x)$$

función de probabilidad nos da la probabilidad de que una variable tome un valor determinado (en este caso discreta)

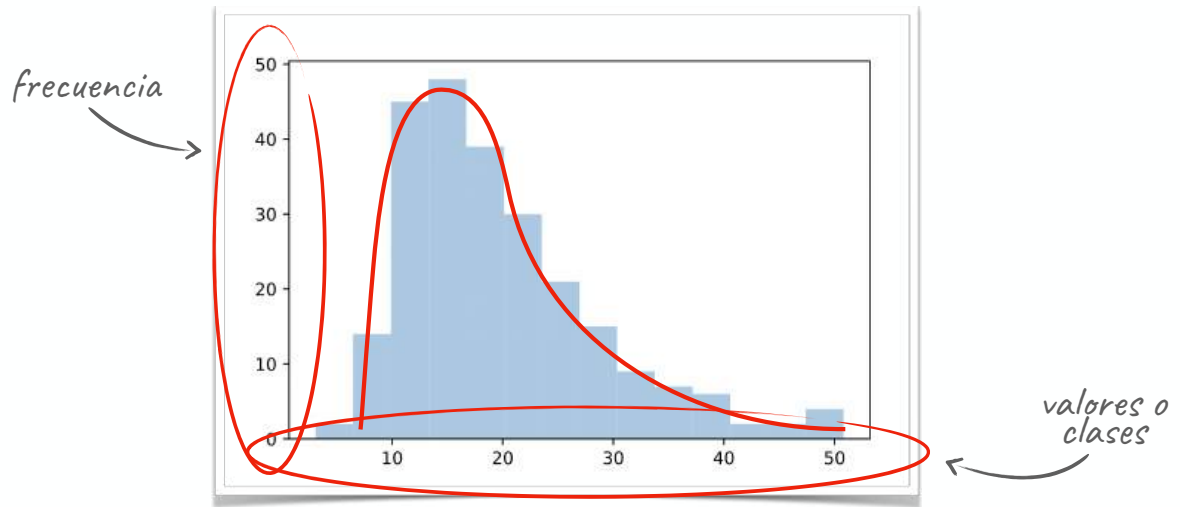
# función de densidad

probability density function (pdf)



función de densidad de probabilidad nos da la probabilidad de que una variable "tome" un valor determinado (en este caso continua)

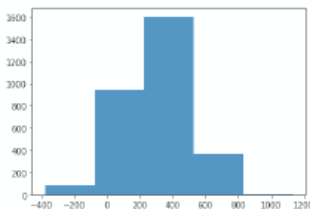
# histograma



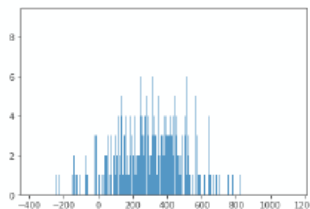
con este diagrama ya podemos comenzar a observar la forma que va a tener la función de densidad/probabilidad

# histograma

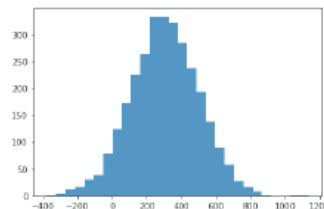
la clave de un histograma es determinar la cantidad de intervalos que vamos a utilizar



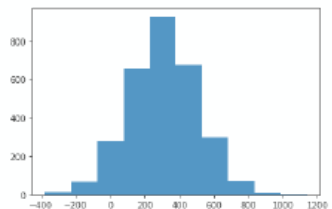
$s = 5$



$s = 3000$



$s = 28$



$s = \text{auto}$

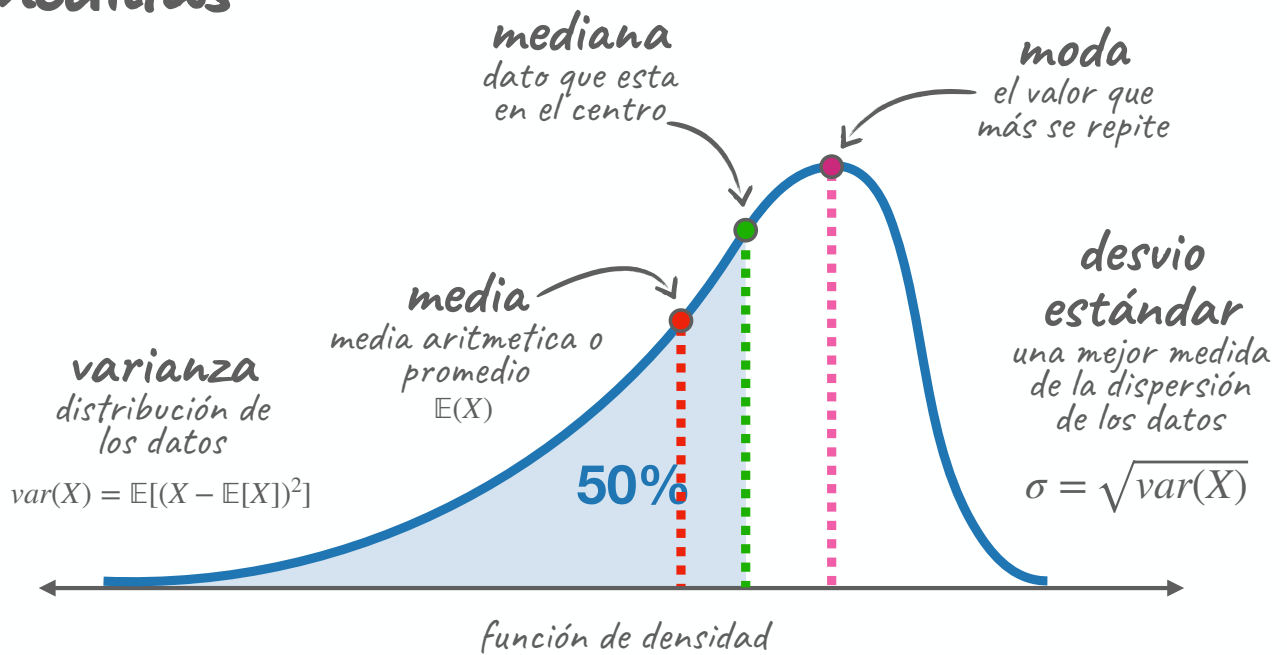
(python decide)

$$s = \frac{\text{val}_{\max} - \text{val}_{\min}}{\sqrt{n}}$$

$n$  : cantidad de datos

$s$  : cantidad de intervalos

# medidas



## coeficiente de correlación lineal



=

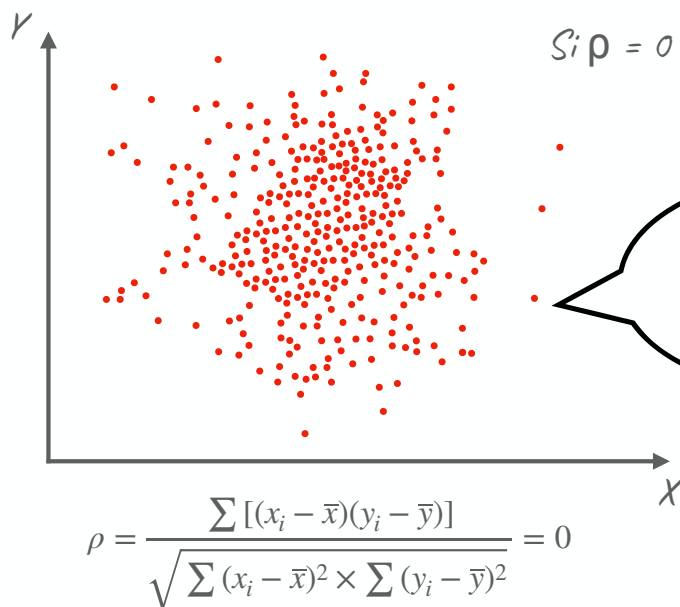


# coeficiente de correlación lineal

coeficiente de correlación  $\rho = \frac{\sum [(x_i - \bar{x})(y_i - \bar{y})]}{\sqrt{\sum (x_i - \bar{x})^2 \times \sum (y_i - \bar{y})^2}}$

media de la variable  $X$   $\bar{x}$  media de la variable  $Y$   $\bar{y}$

covarianza de  $X$  e  $Y$



Si  $\rho = 1$  correlación perfecta

Si  $\rho > 0$  hay cierta correlación

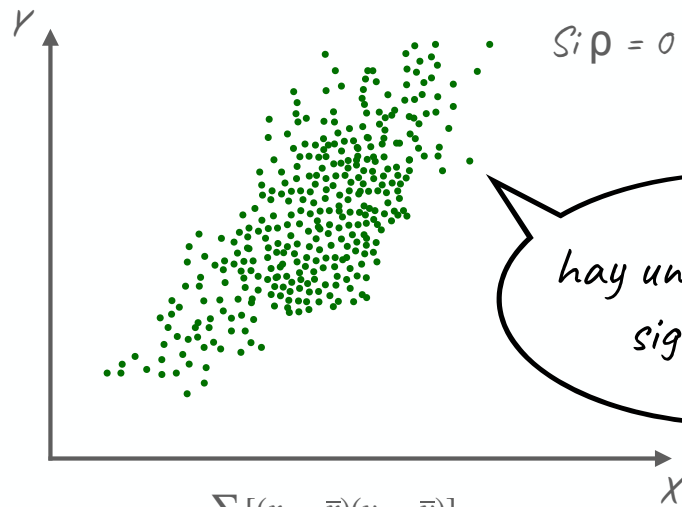
Si  $\rho = 0$  no hay correlación lineal

no hay  
correlación

Si  $\rho = 1$  correlación perfecta

Si  $\rho > 0$  hay cierta correlación

Si  $\rho = 0$  no hay correlación lineal

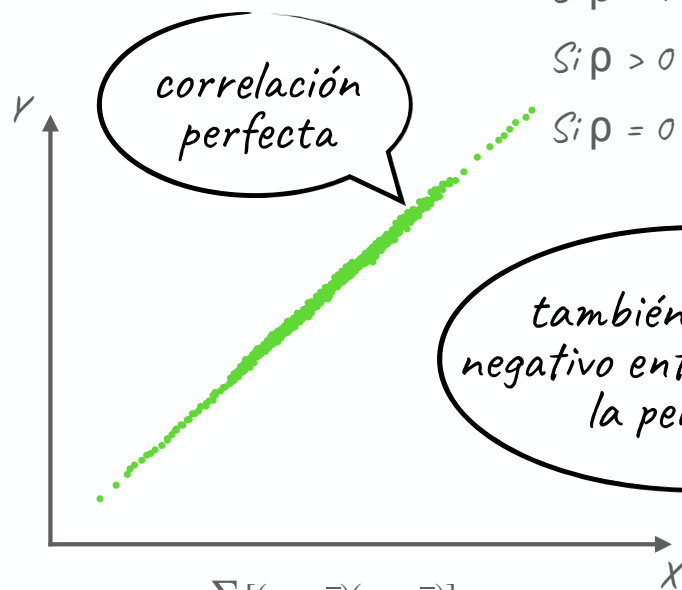


$$\rho = \frac{\sum [(x_i - \bar{x})(y_i - \bar{y})]}{\sqrt{\sum (x_i - \bar{x})^2 \times \sum (y_i - \bar{y})^2}} = 0,7$$

Si  $\rho = 1$  correlación perfecta

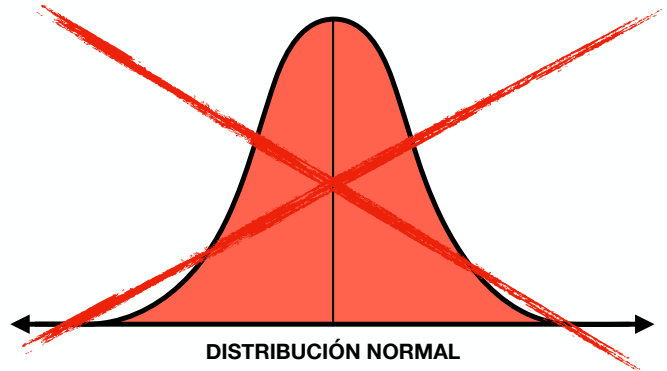
Si  $\rho > 0$  hay cierta correlación

Si  $\rho = 0$  no hay correlación lineal

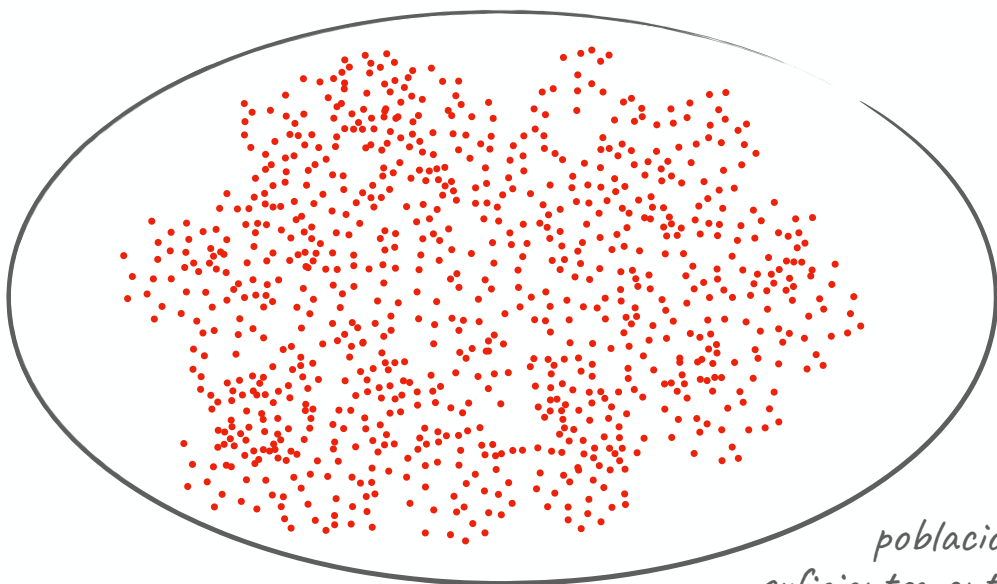


$$\rho = \frac{\sum [(x_i - \bar{x})(y_i - \bar{y})]}{\sqrt{\sum (x_i - \bar{x})^2 \times \sum (y_i - \bar{y})^2}} = 0,97$$

*¿a qué distribución aproximamos  
cuando contamos con una gran  
cantidad de datos?*



*teorema central del límite*

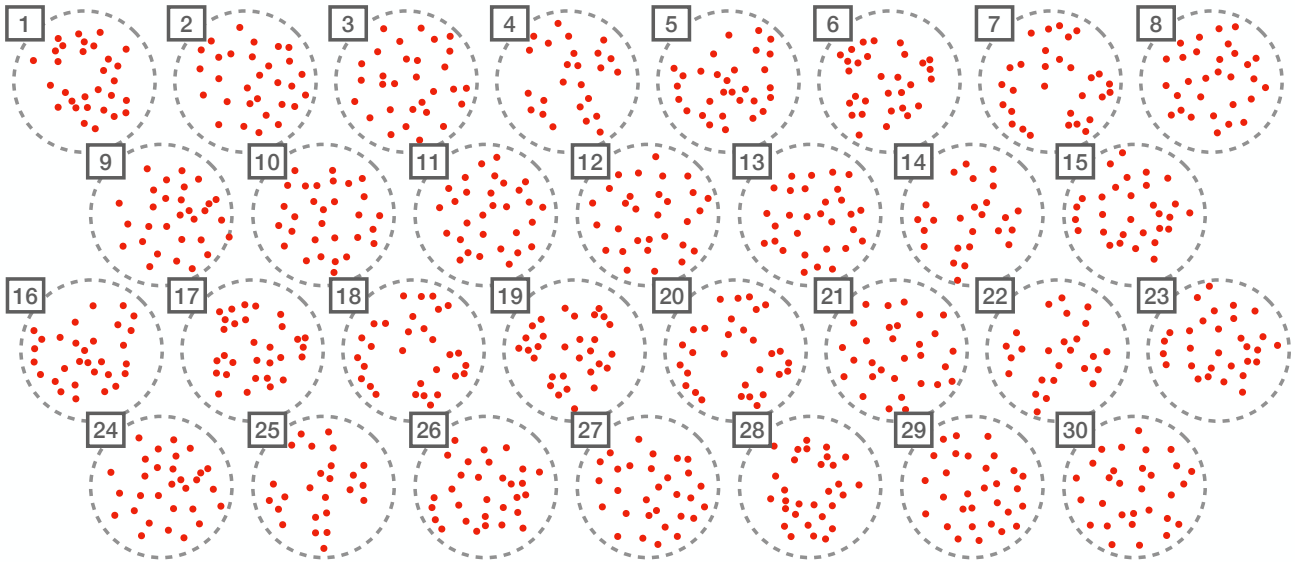


*población  
suficientemente grande*



# teorema central del límite

$$k \geq 30$$
$$n \geq 30$$



# teorema central del límite

si tengo una población suficientemente grande y tomo  $k$  conjuntos de  $n$  elementos, luego,

$$\bar{X} \approx N\left(\mu_p, \frac{\sigma_p}{\sqrt{n}}\right)$$



¿a qué distribución aproximamos  
cuando contamos con una gran  
cantidad de datos?



¡la que corresponda!

## prueba chi-cuadrado

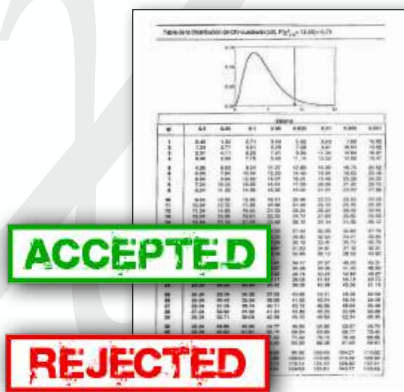
$H_0$  : la muestra tiene una distribución determinada

$H_1$  : la muestra no tiene la distribución determinada

$$\chi^2_{\text{calculado}} = \sum_{i=1}^k \frac{(O_i - E_i)^2}{E_i}$$

$$\chi^2_{\text{calculado}} < \chi^2_{(\alpha, n) \text{ critico}}$$

$$\chi^2_{\text{calculado}} > \chi^2_{(\alpha, n) \text{ critico}}$$



(1916) Karl Pearson

# prueba kolmogorov-smirnov

$H_0$  : la muestra tiene una distribución determinada

$H_1$  : la muestra no tiene la distribución determinada

$$D_n = \sup_x |F_o(x) - F_n(x)|$$

estadístico de prueba

de la distribución elegida

de los datos observada

$$D_n < D_{(\alpha,n)\text{critico}}$$

ACCEPTED

$$D_n > D_{(\alpha,n)\text{critico}}$$

REJECTED



Nikolai Smirnov



Andrei Kolmogorov

TABLE 1. CRITICAL VALUES OF THE KOLMOGOROV-SMIRNOV STATISTIC

One-sided test

n	0.10	0.05	0.025	0.01	0.005
1	0.393	0.424	0.456	0.485	0.500
2	0.424	0.456	0.485	0.500	0.510
3	0.456	0.485	0.500	0.510	0.515
4	0.485	0.500	0.510	0.515	0.518
5	0.500	0.510	0.515	0.518	0.520
6	0.510	0.515	0.518	0.520	0.521
7	0.515	0.518	0.520	0.521	0.522
8	0.518	0.520	0.521	0.522	0.523
9	0.520	0.521	0.522	0.523	0.524
10	0.521	0.522	0.523	0.524	0.525
11	0.522	0.523	0.524	0.525	0.526
12	0.523	0.524	0.525	0.526	0.527
13	0.524	0.525	0.526	0.527	0.528
14	0.525	0.526	0.527	0.528	0.529
15	0.526	0.527	0.528	0.529	0.530
16	0.527	0.528	0.529	0.530	0.531
17	0.528	0.529	0.530	0.531	0.532
18	0.529	0.530	0.531	0.532	0.533
19	0.530	0.531	0.532	0.533	0.534
20	0.531	0.532	0.533	0.534	0.535
21	0.532	0.533	0.534	0.535	0.536
22	0.533	0.534	0.535	0.536	0.537
23	0.534	0.535	0.536	0.537	0.538
24	0.535	0.536	0.537	0.538	0.539
25	0.536	0.537	0.538	0.539	0.540
26	0.537	0.538	0.539	0.540	0.541
27	0.538	0.539	0.540	0.541	0.542
28	0.539	0.540	0.541	0.542	0.543
29	0.540	0.541	0.542	0.543	0.544
30	0.541	0.542	0.543	0.544	0.545
31	0.542	0.543	0.544	0.545	0.546
32	0.543	0.544	0.545	0.546	0.547
33	0.544	0.545	0.546	0.547	0.548
34	0.545	0.546	0.547	0.548	0.549
35	0.546	0.547	0.548	0.549	0.550
36	0.547	0.548	0.549	0.550	0.551
37	0.548	0.549	0.550	0.551	0.552
38	0.549	0.550	0.551	0.552	0.553
39	0.550	0.551	0.552	0.553	0.554
40	0.551	0.552	0.553	0.554	0.555
41	0.552	0.553	0.554	0.555	0.556
42	0.553	0.554	0.555	0.556	0.557
43	0.554	0.555	0.556	0.557	0.558
44	0.555	0.556	0.557	0.558	0.559
45	0.556	0.557	0.558	0.559	0.560
46	0.557	0.558	0.559	0.560	0.561
47	0.558	0.559	0.560	0.561	0.562
48	0.559	0.560	0.561	0.562	0.563
49	0.560	0.561	0.562	0.563	0.564
50	0.561	0.562	0.563	0.564	0.565

# prueba anderson-darling

$H_0$  : la muestra tiene una distribución determinada

$H_1$  : la muestra no tiene la distribución determinada

$$A_n^2 = - \left[ n + \frac{1}{n} \sum_{i=1}^n (2i-1) [\ln(F(Y_i)) + \ln(1 - F(Y_{n+1-i}))] \right]$$

estadístico de prueba

$$A_n^2 < A_{(\alpha,n)\text{critico}}^2$$

ACCEPTED

$$A_n^2 > A_{(\alpha,n)\text{critico}}^2$$

REJECTED



Theodore Anderson



Donald A. Darling



*¿preguntas?*

DEMO 1

*introducción a*  
**SimPy**



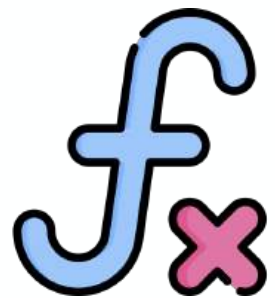


## *funciones vs generadores*

### *funciones ...*

```
def suma(a, b):  
    c = a + b  
    return c  
    print("resultado = ", c)
```

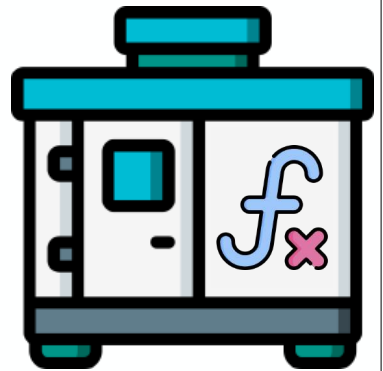
suma(2, 3) → 5



# generadores

```
def operaciones(a, b):  
    c = a + b  
    yield c  
    d = c * 2  
    yield d  
    d = 100  
    yield d
```

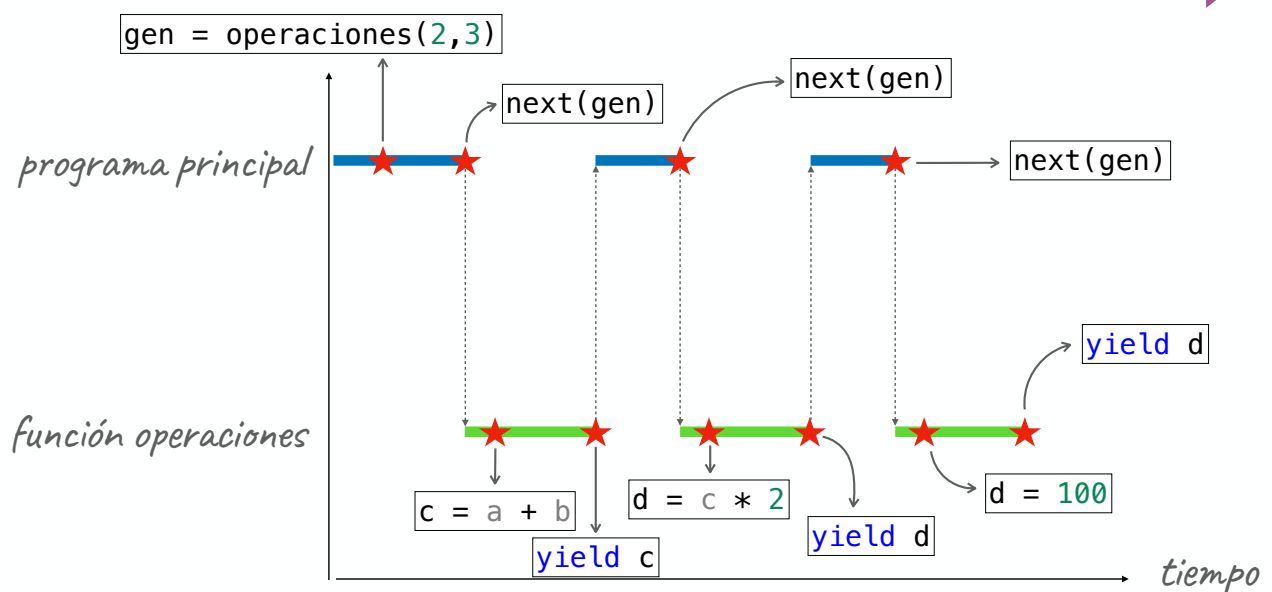
*se podría decir que  
yield devuelve el control  
a quien llamo a la  
función*



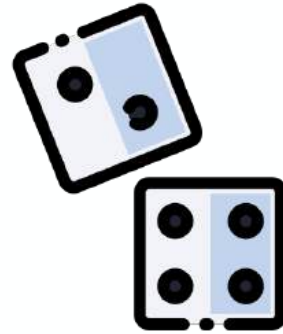
```
z = operaciones(2, 3)  
print(next(z)) → 5  
print(next(z)) → 10  
print(next(z)) → 100
```

# generadores

DEMO 2



# generación de números aleatorios



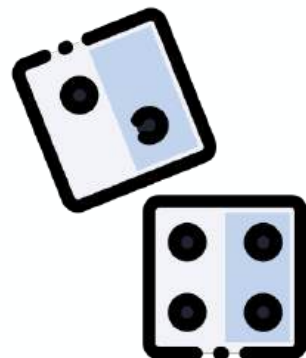
## con numpy ...

*devuelve alguna de las alternativas donde las probabilidades indicadas*

```
np.random.choice(a=[1,2,3,4,5,6])
```

```
np.random.choice(a = [1,2,3],  
                 size = 5,  
                 p = [0.7,0.2,0.1])
```

*devuelve cinco valores seleccionando alguna de las alternativas a según las probabilidades p*





*con numpy ...*

```
np.random.exponential(scale=1.7, size=10)
```

*genera 10 valores según una distribución  
exponencial con media 1,7*

```
np.random.normal(loc=10, scale=2, size=4)
```

*genera cuatro valores según una  
distribución normal con media 10  
y desvio 2*

*ahora simply ...*

*instrucciones  
básicas*





<div> <div> UBAfiuba FACULTAD DE INGENIERÍA </div> <div> SimPy CheatSheet </div> </div>		
<b>Comandos básicos</b>	<b>Recursos</b>	<b>Almacenes</b>
<i>Instala la biblioteca</i> pip install simpy	<i>Allocación de recursos (creación del pool)</i> A = sim.Resource(env, q)	<i>Creación del almacén</i> S = sim.Store(env, Qmax)
<i>Importa la librería SimPy</i> import simpy as sim	<i>Solicitud de recurso</i> r = A.request()	<i>Agrega un ítem al almacén</i> S.put(item)
<i>Crea el ambiente de simulación</i> env = sim.Environment()	<i>Liberación de recurso</i> A.release(r)	<i>Saca un ítem del almacén (FIFO)</i> S.get()
<i>Vincula una función al ambiente de simulación</i> env.process(función())	<i>Estado de una solicitud</i> r.triggered	<i>Lista los ítems del almacén</i> S.items
<i>Inicia la simulación hasta el tiempo t</i> env.run(until=t)	<i>Cantidad de recursos utilizados</i> A.count	<i>Retorna la capacidad del almacén (Qmax)</i> S.capacity
<b>Números aleatorios (numpy)</b>	<b>Tiempo</b>	<b>Contenedores</b>
<i>Devuelve alguna de las n opciones en forma aleatoria</i> np.random.choice(['F','M'], p=[0.5, 0.5])	<i>Retorna el tiempo de la simulación</i> env.now	<i>Creación del contenedor</i> T = sim.Container(env, Qmax)
<i>Genera valores según una distribución normal</i> np.random.normal(media, desvio)	<i>Genera una demora de tiempo t</i> yield env.timeout(t)	<i>Agrega contenido</i> T.put()
<i>Genera valores según una distribución Gama</i> np.random.gamma(shape, scale=1.0)	<i>Generación de un ambiente de tiempo real *</i> env = sim.RealtimeEnvironment(factor=1)	<i>Consumo contenido</i> T.get()
<i>Genera valores según una distribución uniforme</i> np.random.uniform(min, max)	<i>Pausa un proceso (en un generador)*</i> return None	<i>Retorna el nivel del contenedor</i> T.level
Investigación Operativa 3		

# básicos ...

```
env = sim.Environment()
```

↑  
instancio el ambiente  
de simulación

```
env.run(until=10)
```

↖  
inicio la simulación por 10  
unidades de tiempo



atención: simpy no tiene  
interfaz gráfica

# tiempo

`env.now` ← devuelve el tiempo actual  
de la simulación



`env.timeout(t)` ← se usa mucho con  
yield ya que simpy  
controla el tiempo  
↑  
genera una demora en la  
simulación por un tiempo *t*

# procesos en simpy

un proceso de simpy es un generador  
donde básicamente hago cálculos o  
derivaciones y genero demoras



# procesos en simpy

```
def operacion(a, b):  
    c = a + b  
    yield env.timeout(c)
```

*normalmente los procesos deben retornar con alguna función de simpy*

...

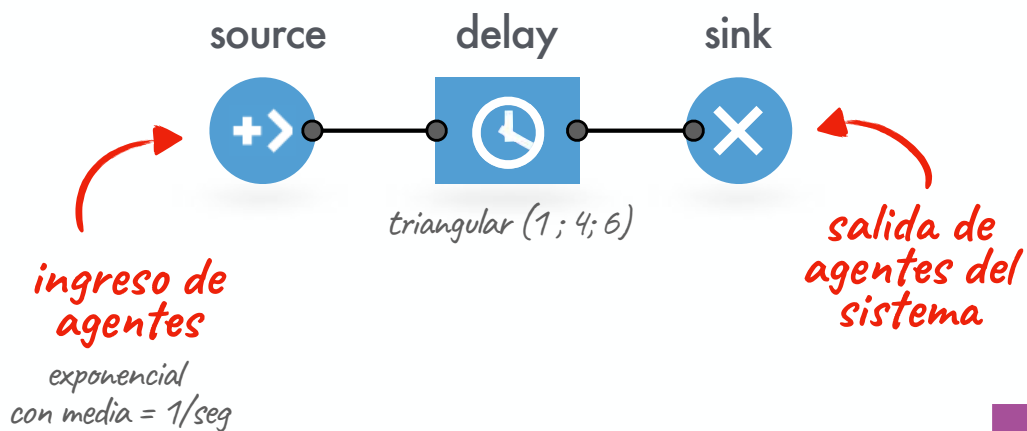
```
env.process(operacion(2,5))
```

...

*se la llama por medio de una función de simpy*



*hagamos un ejercicio ...*



DEMO 3

# *bibliografía y otros ...*

## **[Python] Bibliotecas:**

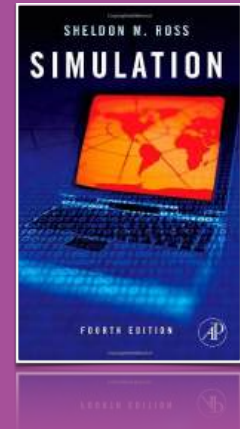
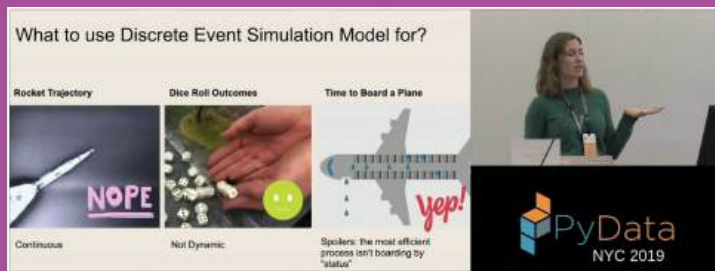
<https://simpy.readthedocs.io>

<https://docs.python.org/3/library/random.html>

<https://fitter.readthedocs.io/en/latest/>

## **[Videos]:**

<https://www.youtube.com/watch?v=693UiPq6mll>



## **Simulation**

Autor: Sheldon M. Ross

Editorial: Academic Press

A man in a white hard hat and blue shirt, carrying a brown briefcase, stands on a rocky peak, looking out over a vast landscape under a bright sky.

*próxima clase:*

# *clase presencial*

*visión por computadora aplicado a  
modelos de investigación operativa*

# INVESTIGACIÓN OPERATIVA SUPERIOR

*¡muchas gracias!*