

INVESTIGACIÓN OPERATIVA SUPERIOR

más de simpy y overleaf ...

Virtual



*muy importante
vamos a utilizar*

“ Normas APA
actualizadas (7ª edición)

overleaf

(para escribir el trabajo final)

DEMO 1

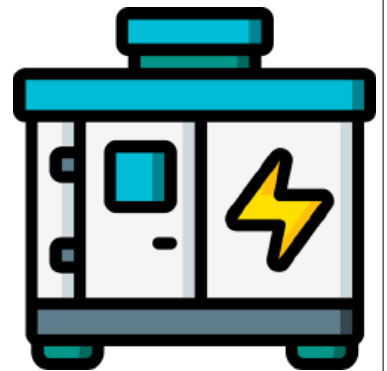


repaso de la clase pasada

generadores

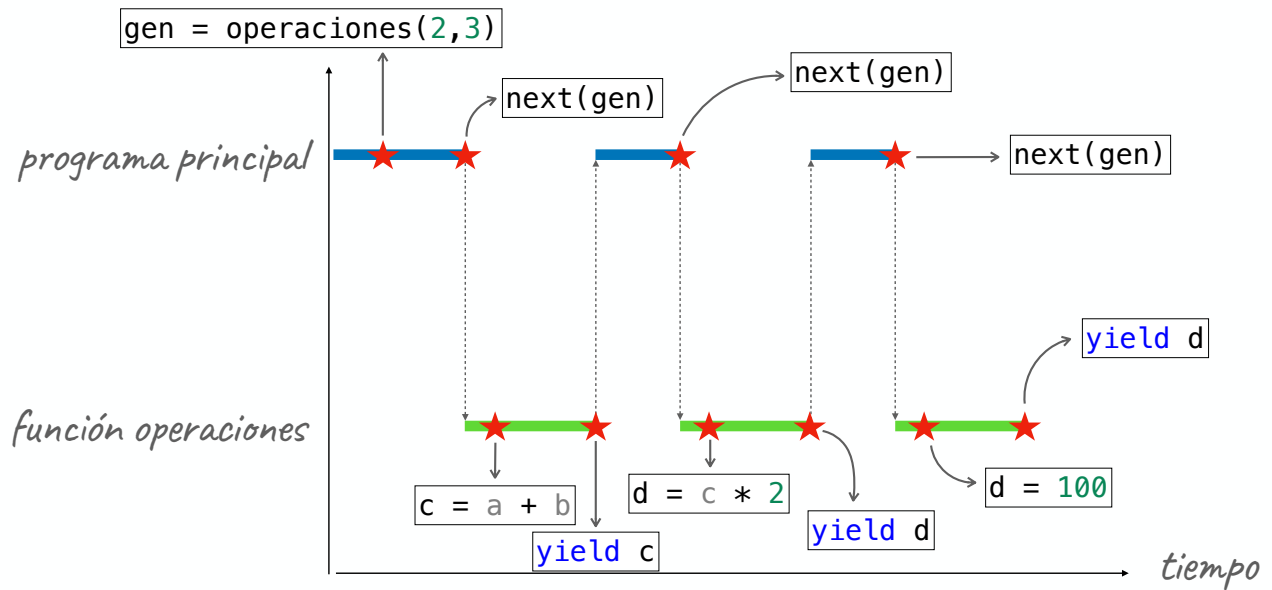
```
def operaciones(a, b):  
    c = a + b  
    yield c  
    d = c * 2  
    yield d  
    d = 100  
    yield d
```

*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



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

```
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...



¿preguntas?

continuamos con
SimPy



uso de recursos



recursos

*crea la disponibilidad de
recursos*



```
operarios = sim.Resource(env, capacity=x)
```



*cantidad de recursos
disponibles*



recursos



```
req_op = operarios.request()
```

*crea una solicitud de
recurso del pool de
recursos*

```
yield req_op | env.timeout(t)
```

espera hasta tener el recurso ...

*o hasta que se
cumpla el tiempo t*

```
operarios.release(req_op)
```

*libera el recurso solicitado
en el pedido req_op*

recursos



```
req_op = operarios.request()
```

```
yield req_op | env.timeout(t)
```

impaciencia

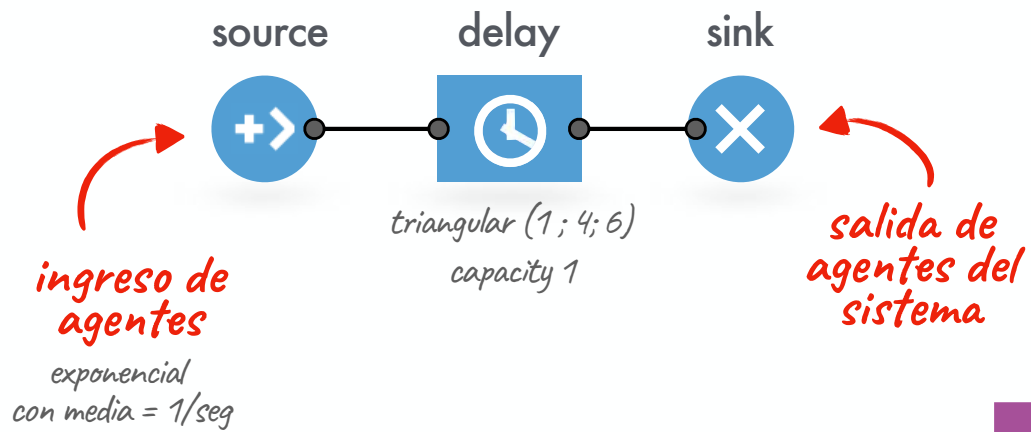
```
if req_op.triggered:
```

```
    ...  
    operarios.release(req_op)
```

```
else:
```

```
    ...  
    req_op.cancel()
```

*muy importante cancelar
el pedido*



DEMO 2

contenedores

contenedores que se cargan y descargan con el mismo material "a granel" ya sea discreto o continuo



contenedores



```
tq = sim.Container( env = env,  
                  init = 0,  
                  capacity = 10 )
```

cantidad inicial (pointing to `init = 0`)

capacidad del contenedor (pointing to `capacity = 10`)

`tq.level` ← *devuelve el nivel actual del contenedor*

`tq.capacity` ← *devuelve la capacidad del contenedor*

contenedores

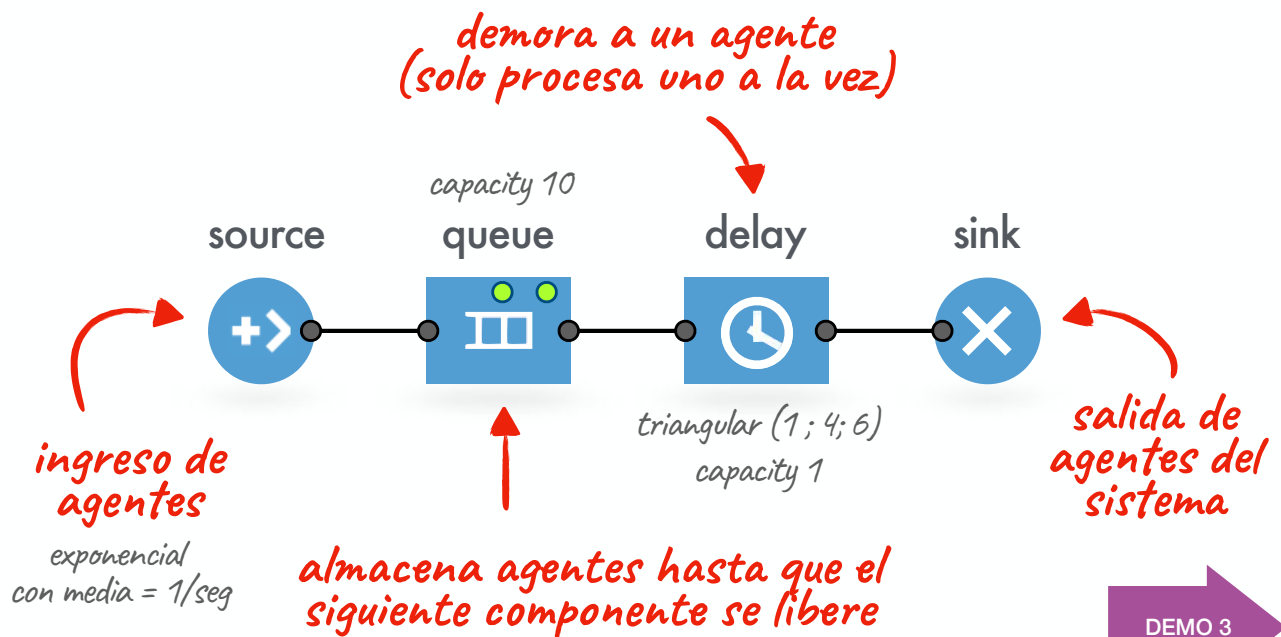
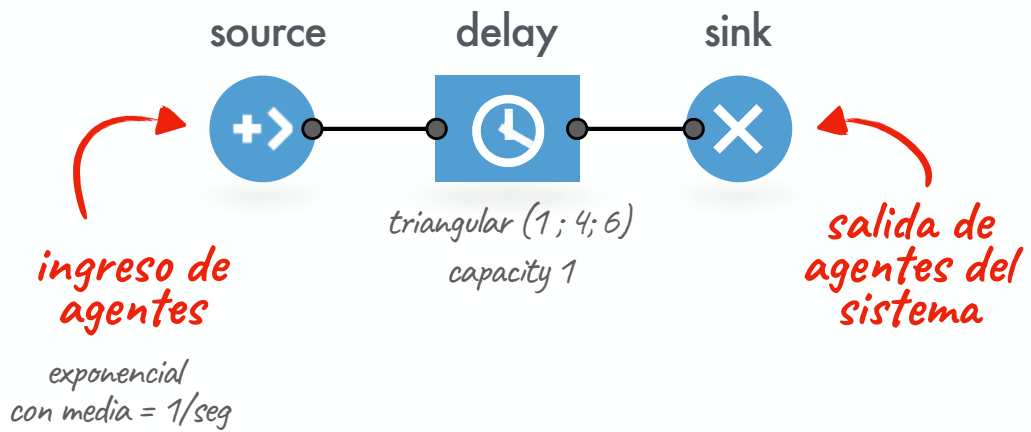


`yield tq.put(q)`

agrega q elementos al contenedor

`yield tq.get(q)`

quita q elementos del contenedor



almacenes

*los almacenes guardan items
nominados*



almacenes

*capacidad del
almacén*

```
a1 = sim.store( env = env, capacity = 10 )
```

```
yield a1.put('1010')
```

```
print(a1.items)
```

```
prod = yield a1.get()
```

```
print(prod.value)
```

*agrega un elemento al almacén
(el elemento puede ser cualquier
cosa, no solo string)*

*quita un elemento del
almacén (fifo)*



funciones lambda



```
def suma(a, b):  
    return a + b
```

```
suma(2,3)
```

equivale a ...

```
suma = lambda a, b: a + b
```

```
print(suma(2,3))
```

DEMO 4a

almacenes



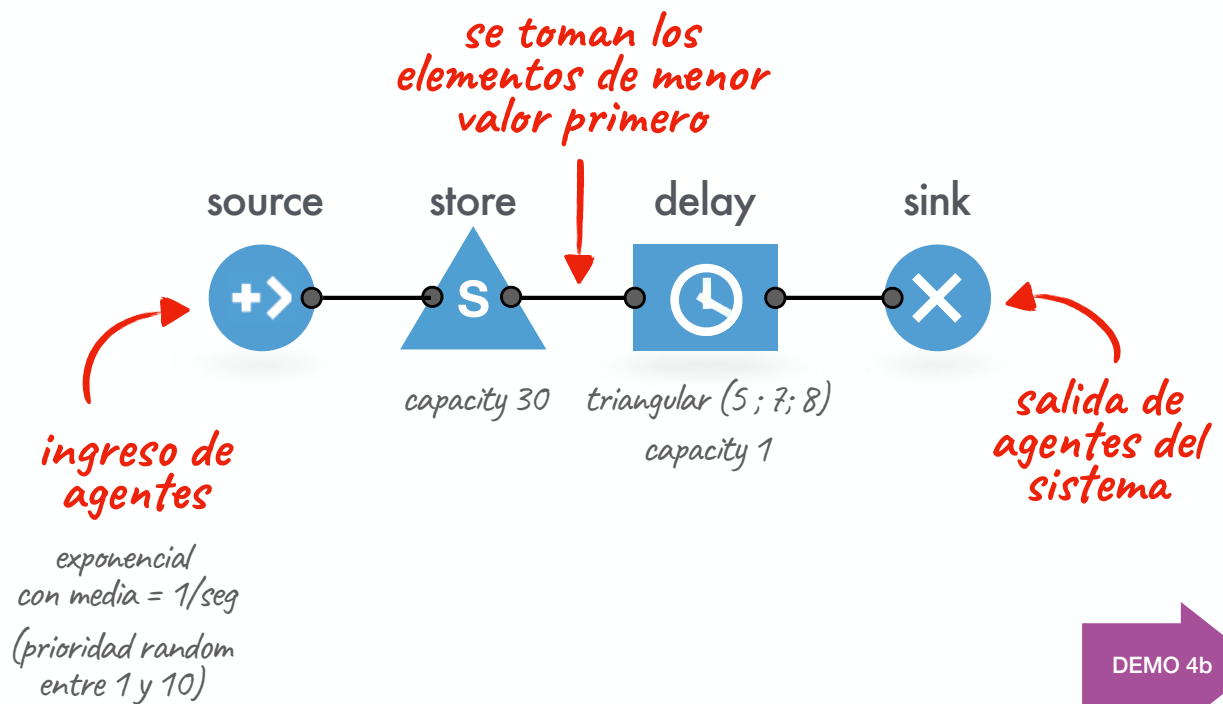
```
a1 = sim.filterStore(env = env,  
                     capacity = 10)
```

```
i1 = a1.get(lambda i: i=='1010')
```

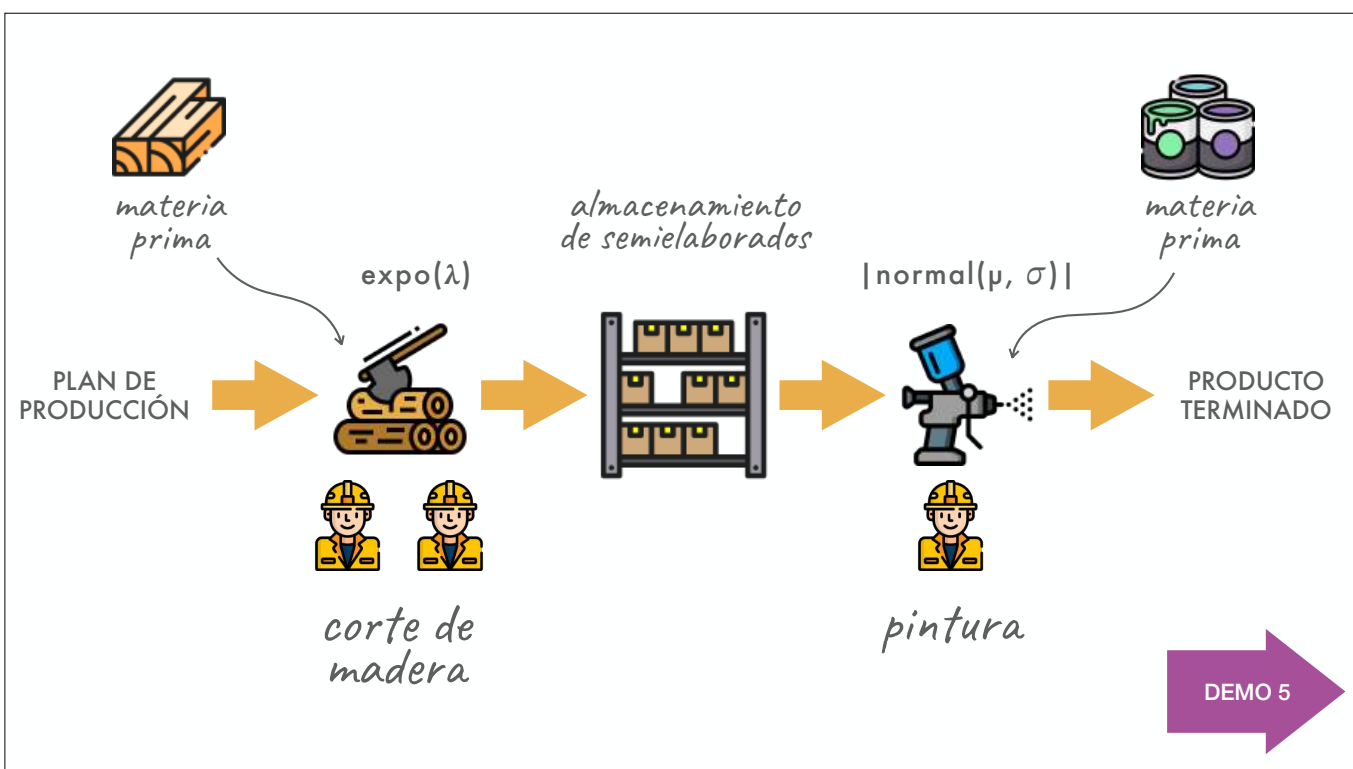


```
a1 = sim.priorityStore(env = env,  
                      capacity = 10)
```

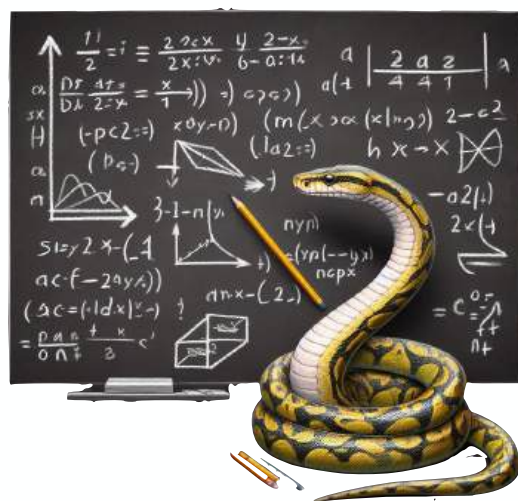
```
a1.put([8, '1010'])
```

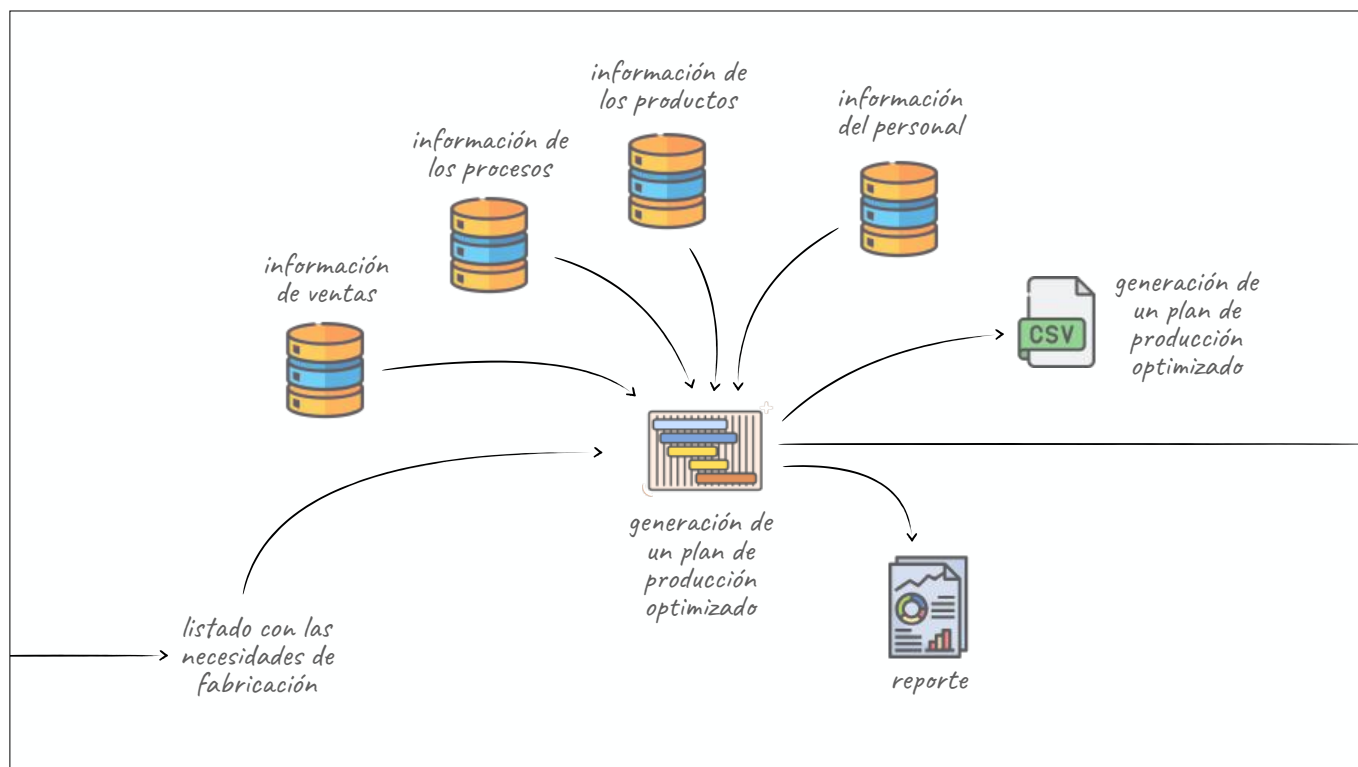
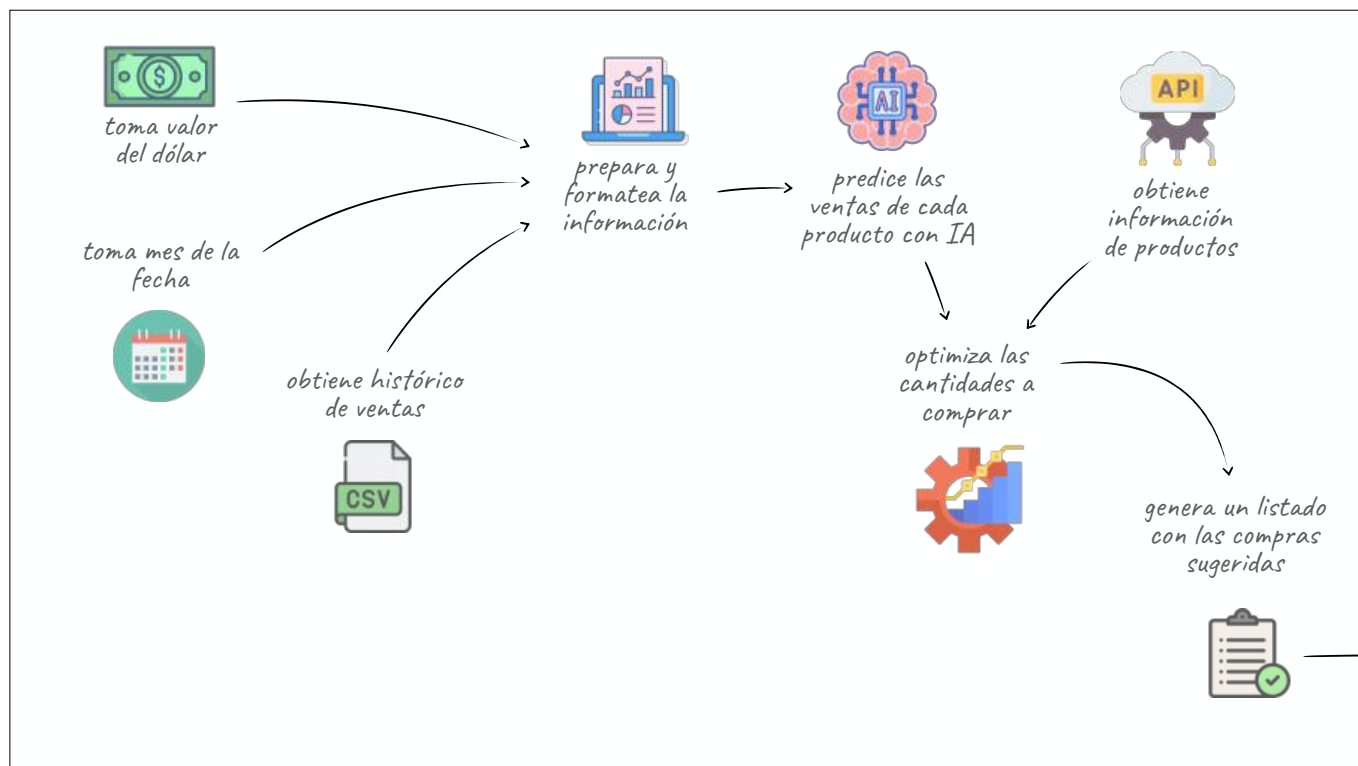


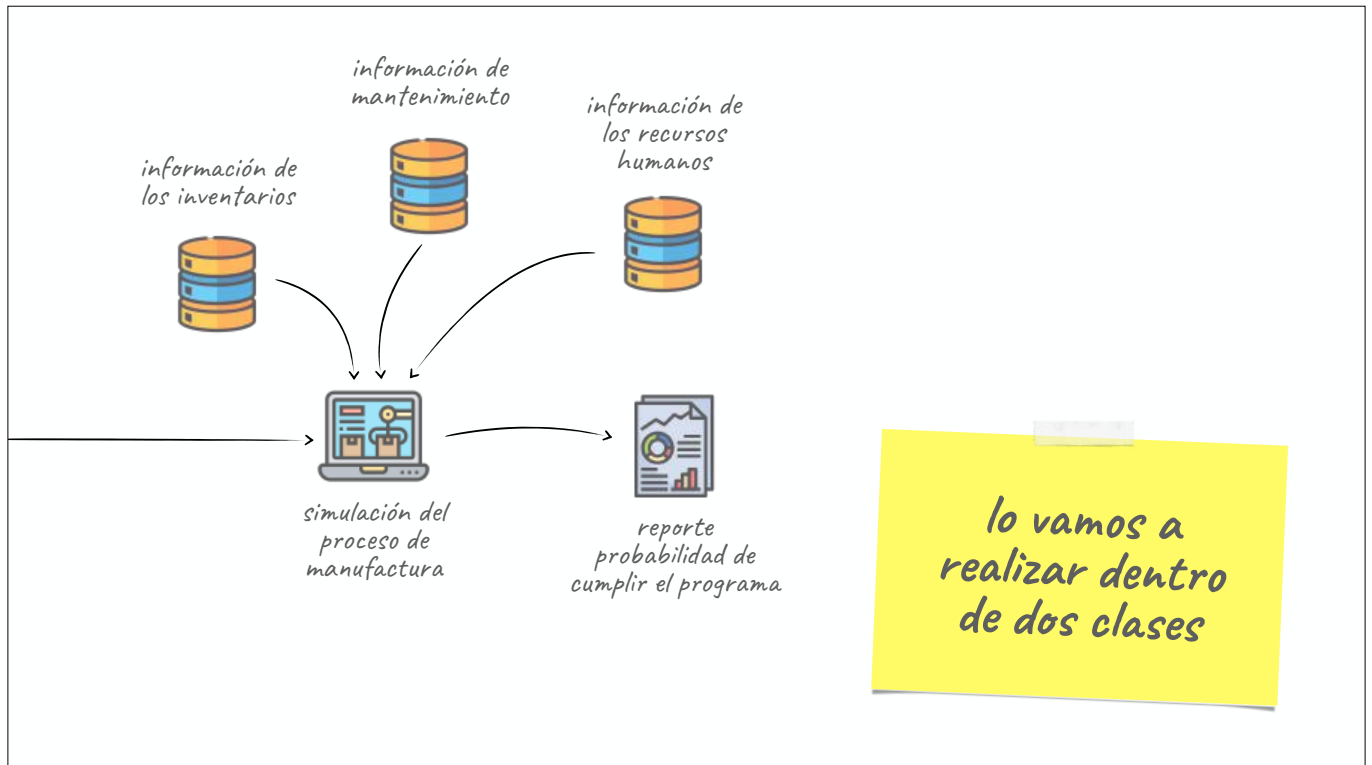
demo 5



presentación del
problema
integrador







algunas practicas optativas ...

(1) del demo 5 ¿qué indicadores útiles podemos obtener?

(2) hacer el demo 5 desde cero



bibliografía y otros ...

[TF] Overleaf:

<https://www.overleaf.com>

<https://es.overleaf.com/learn>



Español

[Videos]:

<https://www.youtube.com/watch?v=Xgq0DvsVrrw>

<https://www.youtube.com/watch?v=xcTN4F3l9Ds>



Inglés



próxima clase:

clase presencial

*reprogramamos la clase 9
visión por computadora aplicado a
modelos de investigación operativa*

INVESTIGACIÓN OPERATIVA SUPERIOR

¡muchas gracias!