

Proceso de compra de vacunas segun la llegada

Utilizando la tarea de predicion de llegada de vacunas y el recieneto de vacunacion , realizar un sistema que permita simular y correlacionar el proceso de llegada/compras de vacuna con el proceso de vacunacion, en donde si no se tiene un stock/numero de vacunas las personas tendran que esperar/reasignar a otro dia en donde exista vacunas dentro del establecimiento y realizar el proceso de vacunacion.

```
In [43]: import simpy
import random
import datetime as dt
from datetime import datetime

In [44]: MESAS = 10
TIEMPO_VACUNACION = 5
SEMANAS = 2
TIEMPO_SIMULACION = SEMANAS *3*9*60
CONTROL_SIGNOS = 1
TIEMPO_POST_VACUNA = 20

tiempo_vacunacion={}
cont={0:1}
FECHA_ACTUAL = datetime.now().strftime('%d/%m/%Y')

In [45]: personas_reasignadas={}
tiempo_vacunacion={}
estado_personas={}

In [46]: class VacunasEstado():
    def __init__(self, env):
        self.env = env
        self.dispensador = simpy.Resource(env, capacity=MESAS)
        self.bodega = simpy.Container(env, init=350, capacity=800)
        self.monitoreo = env.process(self.monitoreo_bodega())
    def monitoreo_bodega(self,):
        while True:
            if self.bodega.level <= 10:
                self.logs_procesos('++++','Solicitar nuevas vacunas ', self.env.now)
                env.process(self.solicitar_vacunas())
            yield self.env.timeout(15)
    def solicitar_vacunas(self,):
        yield self.env.timeout(50)
        self.logs_procesos('','Llega suministro vacunas ',self.env.now)
        nivel = self.bodega.capacity - self.bodega.level
        if nivel ==0:
            nivel+=1
        yield self.bodega.put(nivel)
    def logs_procesos(self,icon, accion, hora):
        print(" %s accion: [%s] hora: [%d] "%(icon, accion, hora))

In [47]: class Vacunacion():
    def __init__(self, env, nombre, estadoVacunas):
        self.env = env
        self.nombre = nombre
        self.estado_vacunas = estadoVacunas
        self.proceso = env.process(self.proceso_vacunacion())
        self.personas_vacunadas = 0
    def control_signos(self):
        yield self.env.timeout(random.randint(CONTROL_SIGNOS, CONTROL_SIGNOS+2))
    def logs_procesos(self,icon,nombre, accion, hora):
        print(" %s nombre: [%s] accion: [%s] hora: [%d] "%(icon,nombre, accion, hora))
    def aplicar_vacuna(self):
        yield self.env.timeout(random.randint(TIEMPO_VACUNACION, TIEMPO_VACUNACION+5))
    def post_vacuna(self, tiempo):
        yield self.env.timeout(tiempo)
    def proceso_vacunacion(self,):
        while True:
            nombre= 'persona '+str(cont[0])
            cont[0]=cont[0]+1
            self.logs_procesos("---->",nombre," llega al recinto",self.env.now)
            estado_personas[nombre]='cola'
            yield env.process(self.control_signos())
            if random.randint(1,100) > 10:
                self.logs_procesos("", nombre, "pasa el control", env.now)
                estado_personas[nombre]='pasa control'
                inicio_vacunacion = self.env.now
                if self.estado_vacunas.bodega.level >=1:
                    yield env.process(self.aplicar_vacuna())
                    estado_personas[nombre]='vacunado'
                    self.estado_vacunas.bodega.get(1)
                    tiempo_vacunacion[nombre] = self.env.now - inicio_vacunacion
                    self.logs_procesos("", nombre, "fue vacunada ", self.env.now)
                    yield env.process(self.post_vacuna(TIEMPO_POST_VACUNA))
                    self.logs_procesos("", nombre, "sin complicaciones", self.env.now)
                    self.personas_vacunadas += 1
                    estado_personas[nombre]='completo'
            else:
                prox_cita = datetime.strptime(datetime.strptime(FECHA_ACTUAL, '%d/%m/%Y')-dt.timedelta(days=3), '%d/%m/%Y')
                print("--XX-- No hay vacunas para %s, reasignado fecha para: %s"%(nombre, prox_cita))
                personas_reasignadas[nombre]='reassignado'
                estado_personas[nombre]='reassignado'
            else:
                self.logs_procesos("", nombre,"no pasa el control", self.env.now)
                estado_personas[nombre]='no control'
```

Ejecucion de la simulacion

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```

    nombre: [
---> nombre
    nombre: [
---> nombre

```

```

nombre: (persona 1300) accion: [pasa el control] hora: [3187]
nombre: (persona 1301) accion: [pasa el control] hora: [3187]
nombre: (persona 1290) accion: [sin complicaciones] hora: [3180]
--> nombre: (persona 1303) accion: [ llega al recinto] hora: [3180]
nombre: (persona 1302) accion: [pasa el control] hora: [3180]
nombre: (persona 1303) accion: [pasa el control] hora: [3181]
nombre: (persona 1293) accion: [sin complicaciones] hora: [3184]
--> nombre: (persona 1304) accion: [ llega al recinto] hora: [3184]
nombre: (persona 1302) accion: [pasa el control] hora: [3180]
nombre: (persona 1300) accion: [fue vacunada ] hora: [3187]
nombre: (persona 1301) accion: [fue vacunada ] hora: [3187]
nombre: (persona 1304) accion: [pasa el control] hora: [3187]
nombre: (persona 1295) accion: [sin complicaciones] hora: [3188]
--> nombre: (persona 1305) accion: [ llega al recinto] hora: [3188]
nombre: (persona 1298) accion: [sin complicaciones] hora: [3188]
--> nombre: (persona 1306) accion: [ llega al recinto] hora: [3188]
nombre: (persona 1296) accion: [sin complicaciones] hora: [3189]
--> nombre: (persona 1307) accion: [ llega al recinto] hora: [3189]
nombre: (persona 1297) accion: [sin complicaciones] hora: [3189]
--> nombre: (persona 1308) accion: [ llega al recinto] hora: [3189]
nombre: (persona 1302) accion: [fue vacunada ] hora: [3190]
nombre: (persona 1303) accion: [fue vacunada ] hora: [3190]
nombre: (persona 1305) accion: [pasa el control] hora: [3191]
nombre: (persona 1306) accion: [pasa el control] hora: [3191]
nombre: (persona 1307) accion: [pasa el control] hora: [3191]
nombre: (persona 1308) accion: [no pasa el control] hora: [3192]
--> nombre: (persona 1309) accion: [ llega al recinto] hora: [3192]
nombre: (persona 1299) accion: [sin complicaciones] hora: [3194]
--> nombre: (persona 1310) accion: [ llega al recinto] hora: [3194]
nombre: (persona 1309) accion: [pasa el control] hora: [3196]
nombre: (persona 1310) accion: [pasa el control] hora: [3196]
nombre: (persona 1304) accion: [fue vacunada ] hora: [3197]
nombre: (persona 1306) accion: [fue vacunada ] hora: [3197]
nombre: (persona 1307) accion: [fue vacunada ] hora: [3198]
nombre: (persona 1305) accion: [fue vacunada ] hora: [3200]
nombre: (persona 1310) accion: [fue vacunada ] hora: [3201]
nombre: (persona 1309) accion: [fue vacunada ] hora: [3201]
nombre: (persona 1300) accion: [sin complicaciones] hora: [3207]
--> nombre: (persona 1311) accion: [ llega al recinto] hora: [3207]
nombre: (persona 1312) accion: [ llega al recinto] hora: [3207]
nombre: (persona 1311) accion: [pasa el control] hora: [3209]
nombre: (persona 1312) accion: [pasa el control] hora: [3209]
nombre: (persona 1302) accion: [sin complicaciones] hora: [3210]
--> nombre: (persona 1313) accion: [ llega al recinto] hora: [3210]
nombre: (persona 1303) accion: [sin complicaciones] hora: [3210]
nombre: (persona 1314) accion: [ llega al recinto] hora: [3210]
nombre: (persona 1314) accion: [no pasa el control] hora: [3211]
--> nombre: (persona 1315) accion: [ llega al recinto] hora: [3211]
nombre: (persona 1315) accion: [pasa el control] hora: [3212]
nombre: (persona 1313) accion: [pasa el control] hora: [3213]
nombre: (persona 1311) accion: [fue vacunada ] hora: [3215]
nombre: (persona 1304) accion: [sin complicaciones] hora: [3217]
nombre: (persona 1316) accion: [ llega al recinto] hora: [3217]
nombre: (persona 1306) accion: [sin complicaciones] hora: [3217]
--> nombre: (persona 1317) accion: [ llega al recinto] hora: [3217]
nombre: (persona 1297) accion: [sin complicaciones] hora: [3218]
--> nombre: (persona 1318) accion: [ llega al recinto] hora: [3218]
nombre: (persona 1312) accion: [fue vacunada ] hora: [3219]
nombre: (persona 1317) accion: [pasa el control] hora: [3219]
nombre: (persona 1318) accion: [pasa el control] hora: [3219]
nombre: (persona 1305) accion: [sin complicaciones] hora: [3220]
--> nombre: (persona 1319) accion: [ llega al recinto] hora: [3220]
nombre: (persona 1316) accion: [pasa el control] hora: [3220]
nombre: (persona 1310) accion: [sin complicaciones] hora: [3221]
--> nombre: (persona 1320) accion: [ llega al recinto] hora: [3221]
nombre: (persona 1313) accion: [fue vacunada ] hora: [3221]
nombre: (persona 1309) accion: [sin complicaciones] hora: [3222]
--> nombre: (persona 1321) accion: [ llega al recinto] hora: [3222]
nombre: (persona 1315) accion: [fue vacunada ] hora: [3222]
nombre: (persona 1319) accion: [pasa el control] hora: [3222]
nombre: (persona 1320) accion: [pasa el control] hora: [3222]
nombre: (persona 1321) accion: [pasa el control] hora: [3223]
nombre: (persona 1316) accion: [fue vacunada ] hora: [3226]
nombre: (persona 1318) accion: [fue vacunada ] hora: [3228]
nombre: (persona 1320) accion: [fue vacunada ] hora: [3228]
nombre: (persona 1317) accion: [fue vacunada ] hora: [3229]
nombre: (persona 1321) accion: [fue vacunada ] hora: [3229]
nombre: (persona 1319) accion: [fue vacunada ] hora: [3231]
nombre: (persona 1311) accion: [sin complicaciones] hora: [3235]
--> nombre: (persona 1322) accion: [ llega al recinto] hora: [3235]
nombre: (persona 1322) accion: [pasa el control] hora: [3236]
nombre: (persona 1312) accion: [sin complicaciones] hora: [3239]
--> nombre: (persona 1323) accion: [ llega al recinto] hora: [3239]
Simulación realizada despues de 2 semanas
[Mesa 0] ha vacunado a 108 personas
[Mesa 1] ha vacunado a 109 personas
[Mesa 2] ha vacunado a 108 personas
[Mesa 3] ha vacunado a 108 personas
[Mesa 4] ha vacunado a 109 personas
[Mesa 5] ha vacunado a 106 personas
[Mesa 6] ha vacunado a 106 personas
[Mesa 7] ha vacunado a 107 personas
[Mesa 8] ha vacunado a 107 personas
[Mesa 9] ha vacunado a 107 personas

```

Resultados

```

In [16]: no_control={key for key in list(estado_personas.keys()) if estado_personas[key]=="no control"}
vacunados={key for key in list(estado_personas.keys()) if estado_personas[key]=="completo"}
reassignados={key for key in list(estado_personas.keys()) if estado_personas[key]=="reassignado"}
print("-"*70,"\nTotal personas: ",len(estado_personas))
print("Personas que no pasaron el control: ",len(no_control))
print("Personas vacunadas: ",len(vacunados))
print("Personas que fueron reasignados por falta de stock: ",len(reassignados))

```

```

-----
Total personas: 2302
Personas que no pasaron el control: 224
Personas vacunadas: 1786
Personas que fueron reasignados por falta de stock: 242

```

```

In [74]: import matplotlib.pyplot as plt

plt.figure(figsize=(15,8))
real = [len(no_control), len(vacunados), len(reassignados)]
etiquetas = ['No pasaron control', 'Vacunados', 'Reasignados']

plt.bar(etiquetas, real,color=['black', 'red', 'green'])

plt.title("Dificultades procesos vacunacion")
plt.show()

```

Categoría	Valor
No pasaron control	224
Vacunados	1786
Reasignados	242