

```
In [134.] !pip install simpy
Requirement already satisfied: simpy in c:\users\fvassq\anaconda3\lib\site-packages (4.0.1)
```

PRUEBA DE SIMULACIÓN SEGUNDO INTERCICLO

VÁSQUEZ FAJARDO FRANKLIN JOEL

SIMULACIÓN

El golpe económico de la crisis sanitaria del corona virus no va a ser cosa de semanas, sino de meses. Dentro de una de las etapas importantes posteriores a las elecciones presidenciales son la re activación de los pequeños comercios y/o centros educativos. Para ello se plantea realizar un modelo de simulación de eventos discretos que permita determinar cual son los principales factores para la re activación económica en el ingreso a las aulas de los estudiantes

Obtener los siguientes análisis:

- Cuantos contagiados tengo al final del mes.
- Cuantos cursos debo cerrar.
- Cuantos estudiantes y docentes ingresan y salen al final del mes.
- Conclusiones
- Recomendaciones.

Conclusión

Durante el desarrollo de la practica se pudo simular el regreso moderado de los alumnos a clases, esto como el nuevo plan de gobierno , durante el cual podemos ver resultados no tan alentadores ya que al komentod e realizar la simulación nos pudimos dar cuenta que el retorno masivo puede ocasionar mayor contagios y a us vez que se cierren mas aulas.

Recomendación

Se recomienda tener en cuenta esto al momento de realizar el retorno a clases, ya que mediante estos medios de simulación de eventos podemos evitar que los contagios sigan y sigan avanzando.

```
In [265.] import simpy
import random
import matplotlib.pyplot as pp
print('LIBRERIAS IMPORTADAS')
```

LIBRERIAS IMPORTADAS

```
In [275.] inicio = 10
tiempo_simulacion = 360000
estudiantes=250
docentes=30
cursos=40
dias=0
```

```
In [276.] alumnos = {}
def dato():
    for i in range(estudiantes):
        alumnos[i+1] = 'almn_{}'.format(i+1),random.randint(1,cursos)
    return alumnos
```

```
In [277.] class prueba():
    def __init__(self,entorno):
        self.env = entorno

    def llega(self,tiempo):
        yield self.env.timeout(tiempo)

    def primera(self,tiempo):
        yield self.env.timeout(tiempo)

    def descanso(self,tiempo):
        yield self.env.timeout(tiempo)

    def segunda(self,tiempo):
        yield self.env.timeout(tiempo)

    def salir(self,tiempo):
        yield self.env.timeout(tiempo)

    def covid(self,listado):
        estudiantes_pcr = []
        for i in range(int(len(listado.keys()))):
            if((random.randint(1,100))<=14):
                estudiantes_pcr.append(1)
        return estudiantes_pcr

    def cls(self,persona):

        llega = random.uniform(0.016,0.032)
        yield self.env.process(self.llega(llega))

        prim = 3
        yield self.env.process(self.primera(prim))

        desc = 0.5
        yield self.env.process(self.descanso(desc))

        segu = 2.5
        yield self.env.process(self.segunda(segu))

        salir = 18
        yield self.env.process(self.salir(salir))
```

```
In [278.] casos = []
contagiadosNuevo = []
caso = []
class Simulacion():
    def sim(self,env,listadoEstudiantes):
        proceso = procesoClases(env)
        if dias < 31:
            print('Asisten {} estudiantes a clases'.format(len(listadoEstudiantes)))
            for i in listadoEstudiantes.items():
                yield env.process(proceso.clases(i))
        else:
            contagiadosNuevo = proceso.covid(listadoEstudiantes)
            caso.append(len(contagiadosNuevo))
            for i in range(len(contagiadosNuevo)):
                for j in range(int(len(listadoEstudiantes.keys()))):
                    if(contagiadosNuevo[i]==j):
                        casos.append(listadoEstudiantes.get(j)[1])

for i in range(32):
```

```
In [279.] print('Dia de clases : {}'.format(i))
env = simpy.Environment()
simulacion = Simulacion()
env.process(simulacion.sim(env,dato()))
env.run(until=tiempo_simulacion)
dias+=1
```

Dia de clases : 0
Asisten 250 estudiantes a clases
Dia de clases : 1
Asisten 250 estudiantes a clases
Dia de clases : 2
Asisten 250 estudiantes a clases
Dia de clases : 3
Asisten 250 estudiantes a clases
Dia de clases : 4
Asisten 250 estudiantes a clases
Dia de clases : 5
Asisten 250 estudiantes a clases
Dia de clases : 6
Asisten 250 estudiantes a clases
Dia de clases : 7
Asisten 250 estudiantes a clases
Dia de clases : 8
Asisten 250 estudiantes a clases
Dia de clases : 9
Asisten 250 estudiantes a clases
Dia de clases : 10
Asisten 250 estudiantes a clases
Dia de clases : 11
Asisten 250 estudiantes a clases
Dia de clases : 12
Asisten 250 estudiantes a clases
Dia de clases : 13
Asisten 250 estudiantes a clases
Dia de clases : 14
Asisten 250 estudiantes a clases
Dia de clases : 15
Asisten 250 estudiantes a clases
Dia de clases : 16
Asisten 250 estudiantes a clases
Dia de clases : 17
Asisten 250 estudiantes a clases
Dia de clases : 18
Asisten 250 estudiantes a clases
Dia de clases : 19
Asisten 250 estudiantes a clases
Dia de clases : 20
Asisten 250 estudiantes a clases
Dia de clases : 21
Asisten 250 estudiantes a clases
Dia de clases : 22
Asisten 250 estudiantes a clases
Dia de clases : 23
Asisten 250 estudiantes a clases
Dia de clases : 24
Asisten 250 estudiantes a clases
Dia de clases : 25
Asisten 250 estudiantes a clases
Dia de clases : 26
Asisten 250 estudiantes a clases
Dia de clases : 27
Asisten 250 estudiantes a clases
Dia de clases : 28
Asisten 250 estudiantes a clases
Dia de clases : 29
Asisten 250 estudiantes a clases
Dia de clases : 30
Asisten 250 estudiantes a clases
Dia de clases : 31

```
In [280.] print('Personas:')
print(estudiantes+docentes)

Personas:
280
```

```
In [281.] print('Total de Contagiados:')
print(caso)

Total de Contagiados:
[36]
```

```
In [282.] print('Cursos Contagiados:')
print(set(casos))

Cursos Contagiados:
{1, 3, 6, 7, 12, 13, 14, 16, 17, 20, 23, 24, 25, 26, 27, 28, 29, 30, 31, 34, 38, 40}
```

```
In [283.] langs = ['Cursos', ' Cerrados']
curso = [cursos, len(cerrar)]
ax.plot(langs,curso)
fig, axes = pp.subplots(figsize=(12,5))
axes.plot(langs, curso, 'r')
axes.set_xlabel('x')
axes.set_ylabel('y')
axes.set_title('title');
```

