



**UNIVERSIDAD POLITECNICA SALESIANA**

SEDE CUENCA

**CARRERA:** INGENIERIA DE SISTEMAS

**Nombre:** Bryam Gabriel Mora Lituma

**Materia:** Simulacion

**Fecha:** 04/07/2021

Diseñe y desarrolle un modelo y/o script que permita simular el siguiente caso real: •En base a los datos del siguiente link <https://educacion.gob.ec/wp-content/uploads/downloads/2012/08/AZUAY11.pdf> (<https://educacion.gob.ec/wp-content/uploads/downloads/2012/08/AZUAY11.pdf>), genere una simulación del ingresos de los estudiantes, para ello debemos escoger un establecimiento y en base a los docentes y estudiantes modelar el reingreso de los estudiantes en base a los siguientes datos.

- Solo se va a tener en cuenta uno de los planteles educativos(Escuela, colegio, universidad dentro del Azuay).
- Se tiene un promedio que el 90% de los docentes han sido vacunados y pueden realizar el proceso de ingreso en cada uno de los cursos.
- Dentro del procesos se tiene que alrededor del 5% - 10% de los estudiantes no podrán asistir debido a no presentar la vacuna/enfermedades adyacentes.
- Los estudiantes solo pertenecen a una sola entidad educativa al igual que los docentes.
- Se va a tener un periodo de prueba de un mes, posterior a ello se realiza al azar al 10% de estudiantes una prueba PCR para validar que no estén contagiados.

- De la ultima el 2% de los estudiantes dan positivo por lo que se cierra el curso completo.
- Los estudiantes asisten cada semana y estos están en un horario de 6 horas ya sea diurno o nocturno.
- Tienen un receso 30 minutos dentro del establecimiento en donde se concentran todos los estudiantes y existe un foco de contagio del 2%.
- El proceso de simulación desarrollado deberá considerar los siguientes aspectos:
  - Generar un cuaderno de Python para el desarrollo y parametrización de graficas, reportes, y animación (Simpy).
  - Generar una animación en 2D/3D del modelo propuesto.
  - 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.

Unidad Educativa: Sudamericano

Estudiantes.

Número total de estudiantes de genero femenino: 212

Número total de estudiantes de genero masculino: 350

Número total de estudiantes del establecimiento: 562

Docentes.

Número de Docentes genero femenino: 24

Número de Docentes genero masculino: 16

Número total de profesores: 40

Aulas: 30

<https://www.infoescuelas.com/ecuador/azuay/unidad-educativa-particular-sudamericano-en-cuenca/>  
(<https://www.infoescuelas.com/ecuador/azuay/unidad-educativa-particular-sudamericano-en-cuenca/>)

### Importamos las librerias

```
In [100]: 1 import random  
          2 import simpy  
          3 import random  
          4 import collections  
          5 import matplotlib.pyplot as pp
```

### Iniciamos nuestras variables

```
In [101]: 1 PROFESORES = 40
2 P_PROF_VACUNADOS = int(PROFESORES*0.9)
3 VACUNADOS_PROF = PROFESORES - P_PROF_VACUNADOS
4 print("Profesores: ",PROFESORES)
5 print("Profesores vacunados: ",P_PROF_VACUNADOS)
6 print("Profesores no vacunados:",VACUNADOS_PROF)
7 ESTUDIANTES = 562
8 print("Estudiantes: ",ESTUDIANTES)
9 P_ESTUDIANTES_NO = random.randint(5,10)
10 print("Porcentaje de estudiantes que no asisten: ",P_ESTUDIANTES_NO,"%")
11 ESTUDIANTES_NO_ASISTEN = int(ESTUDIANTES * P_ESTUDIANTES_NO /100)
12 print("Numero de estudiantes que no asisten: ",ESTUDIANTES_NO_ASISTEN)
13 ESTUDIANTES_ASISTEN = ESTUDIANTES-ESTUDIANTES_NO_ASISTEN
14 print("Numero de estudiantes que asisten: ",ESTUDIANTES_ASISTEN)
15
16 #Total de estudiantes y profesores vacunados que asisten
17 TOTAL = P_PROF_VACUNADOS + ESTUDIANTES_ASISTEN
18
19 TOTAL_CONTAGIADOS = 0
20
21 print("Total de profesores y estudiantes tanto vacunados y que asisten: ",TOTAL)
22
23 cursos = ['curso 1','curso 2','curso 3','curso 4','curso 5','curso 6']
24 aula1 = {}
25 aula2 = {}
26 aula3 = {}
27 aula4 = {}
28 aula5 = {}
29 aula6 = {}
30
31 #print("Cursos: ",cursos)
32
33 #Para el numero de contagiados al final del
34 contagiados = []
```

```
Profesores: 40
Profesores vacunados: 36
Profesores no vacunados: 4
Estudiantes: 562
Porcentaje de estudiantes que no asisten: 5 %
Numero de estudiantes que no asisten: 28
```

Numero de estudiantes que asisten: 534

Total de profesores y estudiantes tanto vacunados y que asisten: 570

### **Metodos para asignar el aula y la clase**

```
In [102]: 1 def uso_aula(env,estudiante,aula,curso):
2     print(aula)
3     print("Hora Ingreso",env.now)
4     if aula=="Curso 1":
5         aula1[estudiante]=estudiante
6         yield env.timeout(0.01)
7         return
8     if aula=="Curso 2":
9         aula2[estudiante]=estudiante
10        yield env.timeout(0.01)
11        return
12    if aula=="Curso 3":
13        aula3[estudiante]=estudiante
14        yield env.timeout(0.01)
15        return
16    if aula=="Curso 4":
17        aula4[estudiante]=estudiante
18        yield env.timeout(0.01)
19        return
20    if aula=="Curso 5":
21        aula5[estudiante]=estudiante
22        yield env.timeout(0.01)
23        return
24    if aula=="Curso 6":
25        aula6[estudiante]=estudiante
26        yield env.timeout(0.01)
27        return
28    yield env.timeout(0.05)
29    return
30
31 def clases(env, curso):
32     print("Clases")
33     c = 0
34     while True:
35         yield env.timeout(random.expovariate(1/0.005))
36         print("time ",env.now)
37         aula = random.choice(curso.cursos)
38         print("El estudiante pertenece al curso. ",aula)
39         print("Hora: ",env.now)
40         estudiante = curso.estudiantes[c]
41         print("Estudiante. ",estudiante)
42         tot = len(aula1)+len(aula2)+len(aula3)+len(aula4)+len(aula5)+len(aula6)
```

```
43     if tot < curso.n_estudiantes:
44         yield env.process(uso_aula(env, estudiante, aula, curso))
45     else:
46         print("Ya no hay mas estudiantes")
47         print("Tiempo ", env.now)
48         break
49     c+=1
```

```

In [103]: 1 Curso = collections.namedtuple('Curso', 'contador,cursos, capacidad, usado, t_curso_lleno, puesto_usado, estud
2 print("Institucion educativa SUDAMERICANO")
3
4 env = simpy.Environment()
5
6 contador = simpy.Resource(env, capacity=6)
7 cursos = ["Curso 1", "Curso 2", "Curso 3", "Curso 4", "Curso 5", "Curso 6"]
8 capacidad = {curso: int(((ESTUDIANTES_ASISTEN)/len(cursos))+3) for curso in cursos}
9 usado = {curso: env.event() for curso in cursos}
10 #Tiempo en el que el curso se llena
11 t_curso_lleno = {curso: None for curso in cursos}
12
13 #Numero de puestos usados
14 puesto_usado = {curso: 0 for curso in cursos}
15
16 estudiantes =[0] * ESTUDIANTES_ASISTEN
17 for j in range(ESTUDIANTES_ASISTEN):
18     estudiantes[j] = "Estudiante "+str(j)
19
20 n_estudiantes = ESTUDIANTES_ASISTEN
21
22 print("cursos: ", cursos[1])
23
24 curs = Curso(contador, cursos, capacidad, usado, t_curso_lleno, puesto_usado, estudiantes, n_estudiantes)
25 env.process(clases(env, curs))
26 env.run(until=35)

```

```

El estudiante pertenece al curso. Curso 3
Hora: 2.7097652316282113
Estudiante. Estudiante 183
Curso 3
Hora Ingreso 2.7097652316282113
time 2.721652815588325
El estudiante pertenece al curso. Curso 4
Hora: 2.721652815588325
Estudiante. Estudiante 184
Curso 4
Hora Ingreso 2.721652815588325
time 2.736750129491289
El estudiante pertenece al curso. Curso 3
Hora: 2.736750129491289
Estudiante Estudiante 185

```



Estudiante: Estudiante 103

Curso 3

Hora Ingreso 2.736750129491289

time 2.768278287435254

El estudiante pertenece al curso. Curso 1

### Estudiantes asignados por aula

```
In [104]: 1 tot = len(aula1)+len(aula2)+len(aula3)+len(aula4)+len(aula5)+len(aula6)
          2
          3 aulas_x = ["aula 1","aula 2","aula 3","aula 4","aula 5","aula 6"]
          4 aulas_y = [len(aula1), len(aula2), len(aula3), len(aula4), len(aula5), len(aula6)]
          5 print("Total estudiantes que deben asistir en el periodo de prueba. ",n_estudiantes)
          6
          7 print("Estudiantes por aula")
          8 print("Aula 1 ",len(aula1))
          9 print("Aula 2 ",len(aula2))
         10 print("Aula 3 ",len(aula3))
         11 print("Aula 4 ",len(aula4))
         12 print("Aula 5 ",len(aula5))
         13 print("Aula 6 ",len(aula6))
         14 print("_____")
         15 print("Total de \nestudiantes que asistieron ",tot)
```

Total estudiantes que deben asistir en el periodo de prueba. 534

Estudiantes por aula

Aula 1 79

Aula 2 92

Aula 3 82

Aula 4 96

Aula 5 92

Aula 6 93

---

Total de

estudiantes que asistieron 534

### Estudiantes contagiados y aula cerrada

```
In [105]: ▶ 1 por_pcr = int(tot*0.10)
2 print("Numero de estudiantes ha hacer la prueba: ",por_pcr," de ",tot)
3 por_pcr = int(len(aula6)*0.02)
4 print("EL numero de aula 6 se cierra")
5 print("Numero de estudiantes que dan positivo de la ultima aula: ",por_pcr," de ",len(aula6))
6 TOTAL_CONTAGIADOS=TOTAL_CONTAGIADOS+por_pcr
7 print("TOTAL CONTAGIADOS: ",TOTAL_CONTAGIADOS)
```

Numero de estudiantes ha hacer la prueba: 53 de 534

Aula 6 se cierra

Numero de estudiantes que dan positivo de la ultima aula: 1 de 93

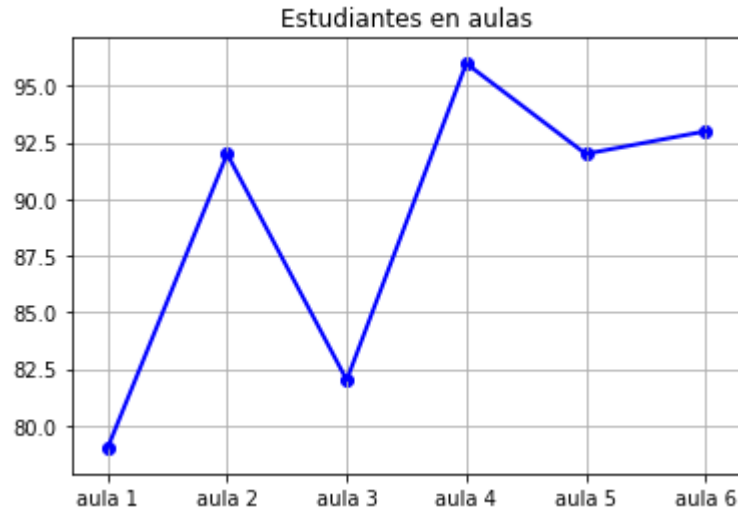
TOTAL CONTAGIADOS: 1

**Grafica de los estudiantes con relacion al numero de aulas**

In [106]:

```
1 x = aulas_x
2 y = aulas_y
3
4 print(x)
5 print(y)
6
7 pp.plot(x,y, linewidth=2,color='blue')
8 pp.scatter(x,y,color='blue')
9 pp.title("Estudiantes en aulas")
10 pp.grid(True)
11 pp.show()
```

```
['aula 1', 'aula 2', 'aula 3', 'aula 4', 'aula 5', 'aula 6']
[79, 92, 82, 96, 92, 93]
```



**Metodos para simular el porceos de regreso a clases**

```
In [107]: 1 def inicio_clases(env, resource, espera, prio):
2     yield env.timeout(espera)
3     with resource.request(priority=prio) as req:
4         yield req
5         print("Inicio clases: ",env.now)
6         yield env.timeout(3.5)
7         return
8     print("Fin primera jornada: ",env.now)
9     return
10
11 def receso(env,n_est,resource,espera,prio):
12     yield env.timeout(espera)
13     with resource.request(priority=prio) as req:
14         print("Inicio de receso: ",env.now)
15         yield req
16         print("est",n_est)
17         contg = int(n_est * 0.02)
18         contg = contg
19         print("Numero de contagiados 2% en receso. ",contg)
20         yield env.timeout(0.5)#0.5 representa media hora (30 min)
21         return
22     print("Fin receso: ",env.now)
23     return
24
25 def fin_clases(env, resource, espera, prio):
26     yield env.timeout(espera)
27     with resource.request(priority=prio) as req:
28         yield req
29         print("Inicio segunda jornada: ",env.now)
30         yield env.timeout(2)
31         return
32     print("Fin segunda jornada: ",env.now)
33     return
```

## Resultado del proceso

```
In [108]: ▶ 1 env = simpy.Environment()
2 res = simpy.PriorityResource(env, capacity=1)
3
4 t_est = len(aula1)+len(aula2)+len(aula3)+len(aula4)+len(aula5)
5 #print(t_est)
6 jornada1 = env.process(inicio_clases(env,res, espera=0,prio=0))
7 receso = env.process(receso(env,t_est,res, espera=3.5,prio=1))
8 jornada2 = env.process(fin_clases(env,res, espera=2,prio=2))
9
10 env.run()
```

```
441
Inicio clases: 0
Inicio de receso: 3.5
est 441
Numero de contagiados 2% en receso. 8
Inicio segunda jornada: 4.0
```

### Estudiantes contagiados durante el regreso a clases

```
In [109]: ▶ 1 contg = int(t_est*0.02)
2 t=contg+TOTAL_CONTAGIADOS
3 print("Número de estudiantes de última aula cerrada: ",len(aula6))
4 print("Numero de contagiados con prueba pcr: ",TOTAL_CONTAGIADOS)
5 print("Numero de contagiados en el receso de 4 aulas: ",contg)
6 print("Numero de contagiados en total: ",t)
```

```
Número de estudiantes de última aula cerrada: 93
Numero de contagiados con prueba pcr: 1
Numero de contagiados en el receso de 4 aulas: 8
Numero de contagiados en total: 9
```

```
In [ ]: ▶ 1
```

