

Universidad Politécnica Salesiana

Alumno: *Rayner Palta*

Materia: *Simulación.*

|

▼ Datos:

Provincia escogida: **Azuay**

Cantón: **Cuenca**

Plantel educativo: **Manuel Jota Calle**

Docentes: 208

Alumnos: 5673

```
pip install simpy
```

```
Collecting simpy
```

```
  Downloading https://files.pythonhosted.org/packages/20/f9/874b0bab83406827d1
```

```
Installing collected packages: simpy
```

```
Successfully installed simpy-4.0.1
```

```
import simpy
```

```
import random as rd
```

```
totalDocentes= 208
```

```
totalAlumnos = 5673
```

```
print(totalAlumnos, totalDocentes)
```

```
total_poblacion= (totalDocentes+totalAlumnos)
```

```
docentes_vacunados = int((totalDocentes*90/100))
```

```
print('docentes vacunados', docentes_vacunados)
```

```
alumnosPCR = int((totalAlumnos*10/100))
```

```
print('total alumnos a prueba PCR', alumnosPCR)
```

```
alumnos_no_asisten = int((totalAlumnos*rd.randint(5,10)/100))
```

```
print('alumnos con problemas de salud o no vacunados', alumnos_no_asisten)
```

```
#Horario_atncion
horarioClases=6
horarioReceso=30
diasLaborales=5
diasMes =31
```

```
# Tiempo de simulación
tiempoSimulacion = ((60*6)*5)*31
alumnos_contagiadosPCR =[]
alumnos_contagiadosReceso =[]
alumnos_mes =[]
docentes_mes=[]
```

```
5673 208
docentes vacunados 187
total alumnos a prueba PCR 567
alumnos con problemas de salud o no vacunados 510
```

```
class procesoRegresoClases(object):
    def __init__(self, environment, lugar):
        self.env=environment
        self.lugarRegreso = simpy.Resource(environment)
        self.lugar = lugar

    #espera 31 periodo prueba
    def analisisPrevioMes(self, alumnos):
        print('ENTRO A PERIODO DE PRUEBA')
        yield self.env.timeout(31)
        print('El',alumnos,'está en periodo de prueba de un mes' )
        alumnos_mes.append(alumnos)
        #Proceso prueba pcr a los alumnos
    def analisisPCR(self, alumnos):
        print('---Inicio PCR-----')
        yield self.env.timeout(5)
        alumnos = alumnosPCR
        alumnos_contagiados2 = int((alumnos*2/100))
        alumnos_contagiadosPCR.append(alumnos_contagiados2)
        print('el alumno',alumnos, 'sale de su prueba PCR')

    def receso(self, alumnos):
        print('receso 30 min')
        yield self.env.timeout(horarioReceso)
        focoContagioAlumnos = int((totalAlumnos*2/100))
        alumnos_contagiadosReceso.append(focoContagioAlumnos)
        print('se acaba el receso para el alumno', alumnos)

    def clases (self, alumnos, docentes):
        print('clases')
        docentes_mes.append(docentes)
        yield self.env.timeout(3)
        receso = procesoRegresoClases.receso
        yield self.env.timeout(3)
```

```

print('acaba clases')

def proceso_regreso_alumnos(env, alumno,punto, docentes):
    print(' ')
    print("Llega el alumno",alumno, "a las",str(env.now))
    with punto.lugarRegreso.request() as lugar:
        yield lugar
        print(' ')
        print('Ingresa al periodo de prueba el alumno',alumno, "a las:",str(env.now))
        yield env.process(punto.analisisPrevioMes(alumno))
        print(' ')
        print('Ingresa al analisis de prueba PCR el',alumno, "a las:",str(env.now))
        yield env.process(punto.analisisPCR(alumno))
        print(' ')
        print('El alumno ',alumno,'entro al clases a las: ',str(env.now),'con el docent
        yield env.process(punto.clases(alumno, docentes))
        print(' ')
        print('El alumno ',alumno,'entro al receso a las: ',str(env.now))
        yield env.process(punto.receso(alumno))

def principal(env, alumno, docente):
    #ejecutamos la clase
    lugar = procesoRegresoClases(env,'Manuel Jota Calle')

    alumnosSinRegreso = alumnos_no_asisten
    print('Total de alumnos que no asisten a clases son: ',alumnosSinRegreso)

    for total in range(alumno):
        env.process(proceso_regreso_alumnos(env,'Alumno %d'%(total+1), lugar, docente))
    while True:
        #yield env.timeout(rd.expovariate(1.5))
        yield env.timeout(5) #5 min por alumno
        total+=1
        env.process(proceso_regreso_alumnos(env,'Alumno %d'%(total+1), lugar,docente))

print("Regreso a clases")
env=simpy.Environment()
print("Total de alumnos => ", totalAlumnos)
print("Total de docentes => ", totalDocentes)

env.process(principal(env,totalAlumnos,totalDocentes))
env.run(until = tiempoSimulacion)

↳ Llega el alumno Alumno 16797 a las 55620
   el alumno 567 sale de su prueba PCR

   El alumno Alumno 773 entro al clases a las: 55620 con el docente 208
   clases

   Llega el alumno Alumno 16798 a las 55625
   acaba clases

```

El alumno Alumno 773 entro al receso a las: 55626
receso 30 min

Llega el alumno Alumno 16799 a las 55630

Llega el alumno Alumno 16800 a las 55635

Llega el alumno Alumno 16801 a las 55640

Llega el alumno Alumno 16802 a las 55645

Llega el alumno Alumno 16803 a las 55650

Llega el alumno Alumno 16804 a las 55655
se acaba el receso para el alumno Alumno 773

Ingresa al periodo de prueba el alumno Alumno 774 a las: 55656
ENTRO A PERIODO DE PRUEBA

Llega el alumno Alumno 16805 a las 55660

Llega el alumno Alumno 16806 a las 55665

Llega el alumno Alumno 16807 a las 55670

Llega el alumno Alumno 16808 a las 55675

Llega el alumno Alumno 16809 a las 55680

Llega el alumno Alumno 16810 a las 55685
El Alumno 774 está en periodo de prueba de un mes

Ingresa al analisis de prueba PCR el Alumno 774 a las: 55687
---Inicio PCR-----

Llega el alumno Alumno 16811 a las 55690
el alumno 567 sale de su prueba PCR

El alumno Alumno 774 entro al clases a las: 55692 con el docente 208
clases

Llega el alumno Alumno 16812 a las 55695
acaba clases

El alumno Alumno 774 entro al receso a las: 55698
receso 30 min

Llega el alumno Alumno 16813 a las 55700

Llega el alumno Alumno 16814 a las 55705

```
totalContagiosMes= len(alumnos_contagiadosReceso)+len(alumnos_contagiadosPCR)
totalAlumnosMes =len(alumnos_mes)
totalDocnteMes =len(docentes_mes)
print('total de alumnos contagiados: ',totalContagiosMes)
print('total de alumnos que asisten al mes: ', totalAlumnosMes)
print('total de docentes por mes:',totalDocnteMes)
```

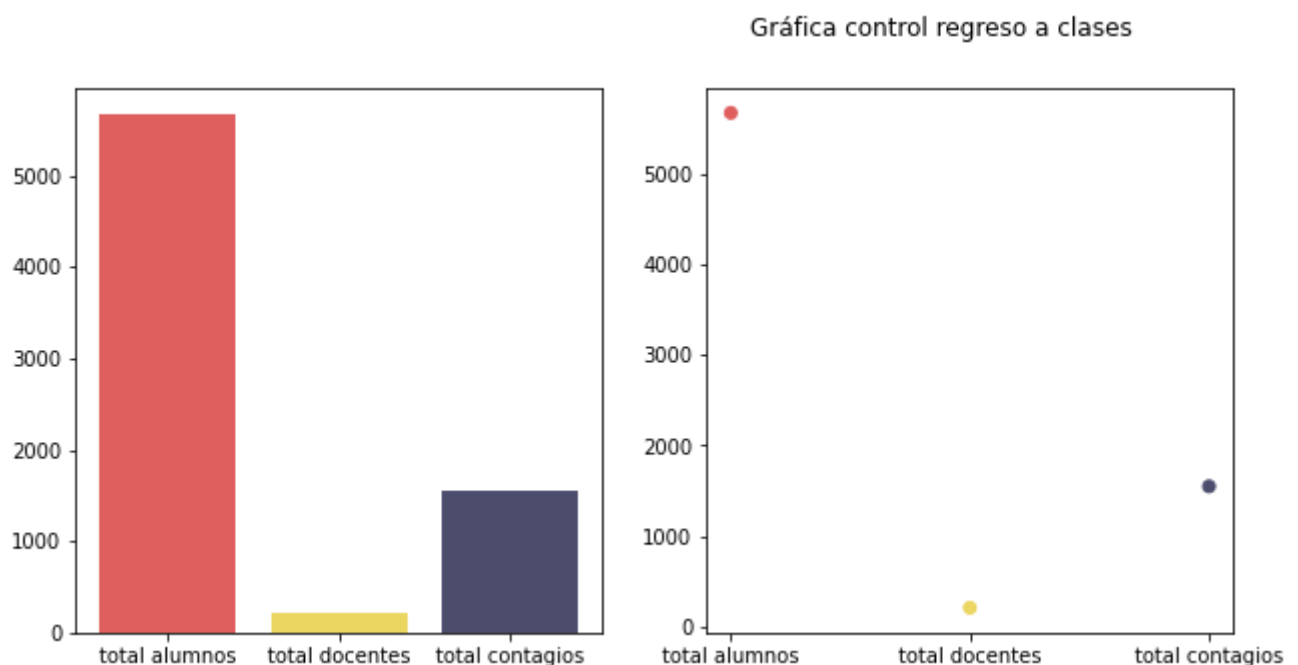
total de alumnos contagiados: 1549

```
total de alumnos que asisten al mes: 775
total de docentes por mes: 775
```

```
import matplotlib.pyplot as plt

nombre2 = [ 'total alumnos','total docentes', 'total contagios']
datos2 = [totalAlumnos,totalDocentes, totalContagiosMes]
fig = plt.figure(figsize=(16,5))
plt.subplot(131)
plt.bar(nombre2, datos2, color=['#DF5E5E', '#ECD662', '#4C4C6D'])
plt.subplot(132)
plt.scatter(nombre2, datos2, color=['#DF5E5E', '#ECD662', '#4C4C6D'])
plt.suptitle('Gráfica control regreso a clases')

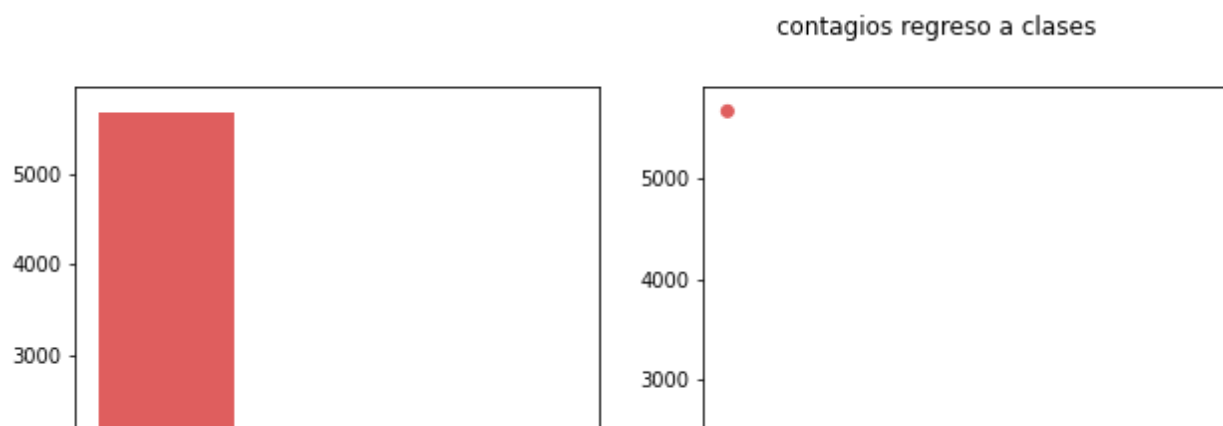
plt.show()
```



```
nombre = [ 'total alumnos','contagios PCR', 'contagios Receso']
datos = [totalAlumnos, len(alumnos_contagiadosPCR), len(alumnos_contagiadosReceso)]
fig = plt.figure(figsize=(16,5))

plt.subplot(131)
plt.bar(nombre, datos, color=['#DF5E5E', '#ECD662', '#4C4C6D'])
plt.subplot(132)
plt.scatter(nombre, datos, color=['#DF5E5E', '#ECD662', '#4C4C6D'])
plt.suptitle('contagios regreso a clases')

plt.show()
```



▼ Recomendaciones

Se recomienda el mantener distanciamiento dentro del plantel educativo en el receso de los alumno

✓ 0s completed at 7:37 PM

● ✕