Desarrollo de una simulación de ingreso a clases presenciales post covid

> Nombre: Bryam Vega
> Maestro: Ing. Diego Quisi
> Universidad: Universidad Politécnica Salesiana

Introducción

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, 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 cuanta 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.
- 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 es un foco de contagio del 2%.

Desarrollo del modelo de simulación

Mediante los siguientes requerimientos, procedemos a realizar el proceso de simulación requerido. Para ello tomamos en cuenta la **Unidad Educativa Hermano Miguel de la Salle**. Para esta simulación solamente tomaremos en cuenta los datos de la sección básica superior y bachillerato. Con ello en cuenta la instutición cuenta con los siguientes datos:

- **Número de docentes:** 43
- **Número de cursos:** 24
- **Número de estudiantes:** 1080

Con estas características procedemos a inicializar los parámetros necesarios para realizar nuestra simulación:

Parámetros iniciales de la simulación.

Tomando en cuenta los requerimientos se obtienen lso siguientes parámetros:

- Existira un total de 38 docentes ya que el 90% debe asistir al plantel.
- Existira un total de 972 estudiantes que asistirán al plantel educativo ya que el 5%-10% no podra asistir por enfermedad o vacuna.
- Existira un horario de 6 horas, es decir 360 min para realizar la simulación de cada día.
- se tiene un receso de 30 min cada día en donde existira una posibilidad de contagio.

In [1]:
```
NUMERO_DOCENTES              = 38
NUMERO_ESTUDIANTES           = 972
CURSOS                       = 24
NUMERO_ESTUDIANTES_POR_CURSO = NUMERO_ESTUDIANTES//CURSOS
TIEMPO_SIMULACION            = 361*NUMERO_ESTUDIANTES
DIAS                         = 0
MES                          = 'SEPTIEMBRE'
RESCESO                      = 30
```

Creamos los estudiantes que se van a clases y maestros

Mediante la clase Student vamos a crear los estudiantes que van a ingresar en nuestra simulación. Para ello se crea una lista de estudiantes que contiene la clase estudiante.

In [2]:
```
class Estudiante:
    def __init__(self,nombre,curso,infectado,examen):
        self.nombre    = nombre
        self.curso     = curso
        self.infectado = infectado
        self.examen    = examen

class Maestro:
    def __init__(self,nombre,infectado,examen):
        self.nombre      = nombre
        self.curso       = 'indefinido'
        self.institucion = 'Unidad Educativa Hermano Miguel de la Salle'
        self.infectado   = infectado
        self.examen      = examen

def llenarEstudiantes():
    count=1
    lista_estudiantes = {}
    cursos_nivel      = ['octavo','noveno','decimo','primero','segundo','tercero']
    cursos_id         = ['A','B','C','D']
    for i in cursos_nivel:
        for j in cursos_id:
            for z in range(0,NUMERO_ESTUDIANTES_POR_CURSO+3 if i=='tercero' else NUMERO_ESTUDIANTES_POR_CURSO):
                nombre='estudiante_{}'.format(count)
                estudiante = Estudiante(nombre,'{}_{}'.format(i,j),0,0)
                lista_estudiantes[nombre]=estudiante
                count+=1
    return lista_estudiantes

def llenarMaestros():
    count=1
    lista_maestros = {}
    for i in range(0,NUMERO_DOCENTES):
        nombre = 'maestro_{}'.format(count)
        maestro = Maestro(nombre,0,0)
        lista_maestros[nombre]=maestro
        count+=1
    return lista_maestros
```

Llamamos a llenar los estudiantes que van a estar en nuestra simulación y procedemos inicializarlos. A continuación se ve como se crearón los estudiantes.

In [3]:
```
LISTA_ESTUDIANTES = llenarEstudiantes()
LISTA_MAESTROS    = llenarMaestros()
```

Una vez realizado esto unificamos los datos para tener un total de personas que iran al plantel y poder realizar toda la simulación.

In [4]:
```
LISTA_PERSONAS = {**LISTA_ESTUDIANTES,**LISTA_MAESTROS}
```

Ejecución de la simulación

En esta sección se crean los métodos necesarios para realizar el proceso de simulación tomando en cuenta la institución como una clase para nuestra simulación y los métodos del proceso del día de clases y proceso de prueba pcr para probar la simulación.

Importamos librerias

Las librerías que utilizamos son `simpy` para nuestra simulación y `random` para la generación de valores randómicos.

In [5]:
```
import simpy
import random
```

Creación de las clases de simulación

Para este caso contamos con nuestra clase `Colegio` el cual contiene los métodos `proceso_dia_clases` que simula los días de clases y `proceso_pcr` que simula el proceso de las pruebas pcr.

Por otro lado tenemos nuestra clase `Simulacion` que contiene los métodos para correr la simulación.

In [6]:
```
class Colegio():
    def __init__(self,environment):
        self.env          = environment
        self.pcr          = simpy.Resource(self.env,5)

    def proceso_clases(self):
        yield self.env.timeout(165)

    def proceso_receso(self):
        global RESCESO
        yield self.env.timeout(RESCESO)

    def proceso_prueba(self):
        yield self.env.timeout(2)

    def proceso_dia_clases(self,persona):
        yield self.env.process(self.proceso_clases())
        yield self.env.process(self.proceso_receso())
        if(random.randint(1,100)>98):
            persona.infectado=1
            LISTA_PERSONAS[persona.nombre] = persona
        yield self.env.process(self.proceso_clases())

    def proceso_pcr(self,persona):
        with self.pcr.request() as pcr:
            yield pcr
            yield self.env.process(self.proceso_prueba())
            if(random.randint(1,100)>(95 if persona.infectado==1 else 98)):
                persona.infectado=2
                LISTA_PERSONAS[persona.nombre] = persona

class Simulacion():

    def ejecutar_simulacion(self,env,filtrado):
        colegio = Colegio(env)
        global DIAS
        if DIAS < 31:
            print('inrgresan {} personas a la instituación'.format(len(filtrado)))
            for i in filtrado.items():
                yield env.process(colegio.proceso_dia_clases(i[1]))
        else:
            print('se tomo una muestra de: {} para prueba PCR'.format(int(len(LISTA_PERSONAS)*0.1)))
            for i in random.choices(list(LISTA_PERSONAS.keys()),k=int(len(LISTA_PERSONAS)*0.1)):
                env.process(colegio.proceso_pcr(LISTA_PERSONAS[i]))

def filtrar_posibles_contagios():
    filtrado={}
    for i in LISTA_PERSONAS.items():
        if i[1].infectado!=1:
            filtrado[i[0]]=i[1]
    return filtrado
```

In [7]:
```
DIAS=0
while (DIAS<=31):
    print("="*10,'DÍA {}'.format(DIAS),"="*10)
    env = simpy.Environment()
    simulacion = Simulacion()
    env.process(simulacion.ejecutar_simulacion(env,filtrar_posibles_contagios()))
    env.run(until=TIEMPO_SIMULACION)
    if(DIAS==31):
        print('infectados confirmados: {}'.format(len([i for i in LISTA_PERSONAS.items() if i[1].infectado ==2])))
    else:
        print('posibles infectados con corte en el día: {}'.format(len([i for i in LISTA_PERSONAS.items() if i[1].in
fectado ==1])))
    DIAS+=1
```

```
========== DÍA 0 ==========
inrgresan 1010 personas a la instituación
posibles infectados con corte en el día: 17
========== DÍA 1 ==========
inrgresan 993 personas a la instituación
posibles infectados con corte en el día: 45
========== DÍA 2 ==========
inrgresan 965 personas a la instituación
posibles infectados con corte en el día: 59
========== DÍA 3 ==========
inrgresan 951 personas a la instituación
posibles infectados con corte en el día: 73
========== DÍA 4 ==========
inrgresan 937 personas a la instituación
posibles infectados con corte en el día: 92
========== DÍA 5 ==========
inrgresan 918 personas a la instituación
posibles infectados con corte en el día: 108
========== DÍA 6 ==========
inrgresan 902 personas a la instituación
posibles infectados con corte en el día: 121
========== DÍA 7 ==========
inrgresan 889 personas a la instituación
posibles infectados con corte en el día: 148
========== DÍA 8 ==========
inrgresan 862 personas a la instituación
posibles infectados con corte en el día: 161
========== DÍA 9 ==========
inrgresan 849 personas a la instituación
posibles infectados con corte en el día: 181
========== DÍA 10 ==========
inrgresan 829 personas a la instituación
posibles infectados con corte en el día: 194
========== DÍA 11 ==========
inrgresan 816 personas a la instituación
posibles infectados con corte en el día: 205
========== DÍA 12 ==========
inrgresan 805 personas a la instituación
posibles infectados con corte en el día: 221
========== DÍA 13 ==========
inrgresan 789 personas a la instituación
posibles infectados con corte en el día: 242
========== DÍA 14 ==========
inrgresan 768 personas a la instituación
posibles infectados con corte en el día: 260
========== DÍA 15 ==========
inrgresan 750 personas a la instituación
posibles infectados con corte en el día: 274
========== DÍA 16 ==========
inrgresan 736 personas a la instituación
posibles infectados con corte en el día: 285
========== DÍA 17 ==========
inrgresan 725 personas a la instituación
posibles infectados con corte en el día: 293
========== DÍA 18 ==========
inrgresan 717 personas a la instituación
posibles infectados con corte en el día: 306
========== DÍA 19 ==========
inrgresan 704 personas a la instituación
posibles infectados con corte en el día: 320
========== DÍA 20 ==========
inrgresan 690 personas a la instituación
posibles infectados con corte en el día: 335
========== DÍA 21 ==========
inrgresan 675 personas a la instituación
posibles infectados con corte en el día: 350
========== DÍA 22 ==========
inrgresan 660 personas a la instituación
posibles infectados con corte en el día: 357
========== DÍA 23 ==========
inrgresan 653 personas a la instituación
posibles infectados con corte en el día: 372
========== DÍA 24 ==========
inrgresan 638 personas a la instituación
posibles infectados con corte en el día: 387
========== DÍA 25 ==========
inrgresan 623 personas a la instituación
posibles infectados con corte en el día: 398
========== DÍA 26 ==========
inrgresan 612 personas a la instituación
posibles infectados con corte en el día: 408
========== DÍA 27 ==========
inrgresan 602 personas a la instituación
posibles infectados con corte en el día: 419
========== DÍA 28 ==========
inrgresan 591 personas a la instituación
posibles infectados con corte en el día: 427
========== DÍA 29 ==========
inrgresan 583 personas a la instituación
posibles infectados con corte en el día: 439
========== DÍA 30 ==========
inrgresan 571 personas a la instituación
posibles infectados con corte en el día: 450
========== DÍA 31 ==========
se tomo una muestra de: 101 para prueba PCR
infectados confirmados: 3
```

Reportes

En esta sección se realiza un reporte para resumir la información de la simulación y analizar resultados

In [14]:
```
import altair as alt
import pandas as pd
import plotly.express as px
import plotly
plotly.offline.init_notebook_mode(connected=True)
```

Estadísticas generales

In [9]:
```
print('Número total de personas en la institución: {}'.format(len(LISTA_PERSONAS)))
print('Número total de estudiantes: {}'.format(len(LISTA_ESTUDIANTES)))
print('Número total de maestros: {}'.format(len(LISTA_MAESTROS)))
print('Número total de personas con posible contagio: {}'.format(len([i for i in LISTA_PERSONAS.items() if i[1].infe
ctado ==1])))
print('Número total de personas contagiadas confirmadas: {}'.format(len([i for i in LISTA_PERSONAS.items() if i[1].i
nfectado ==2])))
print('Cursos que se cierran: {}'.format(list(pd.DataFrame([i[1].curso for i in LISTA_PERSONAS.items() if i[1].infec
tado ==2],columns=['cursos'])['cursos'].unique())))
```

```
Número total de personas en la institución: 1010
Número total de estudiantes: 972
Número total de maestros: 38
Número total de personas con posible contagio: 448
Número total de personas contagiadas confirmadas: 3
Cursos que se cierran: ['noveno_C', 'segundo_D', 'tercero_A']
```

Estudiantes y docentes que entran y salen al fin del mes

In [10]:
```
filtrado_estudiantes = {}
filtrado_maestros    = {}
for i in LISTA_PERSONAS.items():
    if(i[0].find('maestro')!=-1):
        filtrado_maestros[i[0]]=i[1]
    else:
        filtrado_estudiantes[i[0]]=i[1]

no_infectados = 0
infectados    = 0
po_infectados = 0
for i in filtrado_maestros.items():
    if i[1].infectado==1:
        po_infectados+=1
    elif i[1].infectado==2:
        infectados+=1
    else:
        no_infectados+=1
df_maestro = pd.DataFrame([('no infectados',no_infectados),('posibles infectados',po_infectados),
                          ('infectados',infectados)],columns=['description','value'])
df_maestro
```

Out[10]:

	description	value
0	no infectados	25
1	posibles infectados	13
2	infectados	0

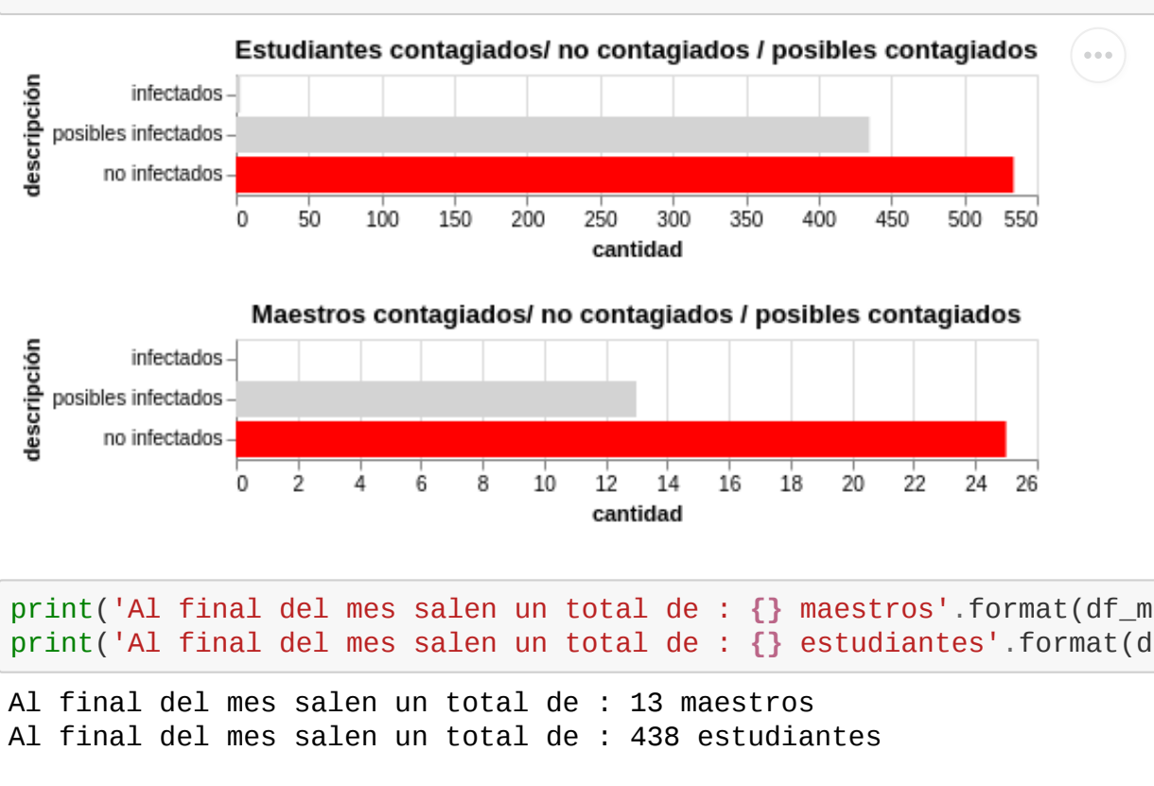
In [11]:
```
no_infectados = 0
infectados    = 0
po_infectados = 0
for i in filtrado_estudiantes.items():
    if i[1].infectado==1:
        po_infectados+=1
    elif i[1].infectado==2:
        infectados+=1
    else:
        no_infectados+=1
df_estudiantes = pd.DataFrame([('no infectados',no_infectados),('posibles infectados',po_infectados),
                          ('infectados',infectados)],columns=['description','value'])
df_estudiantes
```

Out[11]:

	description	value
0	no infectados	534
1	posibles infectados	435
2	infectados	3

In [12]:
```
alt.Chart(df_estudiantes).mark_bar().encode(
    y=alt.Y('description',sort=alt.EncodingSortField(field="value",order='ascending'),title='descripción'),
    x=alt.X('value',title='cantidad'),
    color= alt.condition(
            alt.datum.value == max(df_estudiantes['value']),
            alt.value('red'),
            alt.value('lightgrey')
            )
).properties(
    title='Estudiantes contagiados/ no contagiados / posibles contagiados'
)&alt.Chart(df_maestro).mark_bar().encode(
    y=alt.Y('description',sort=alt.EncodingSortField(field="value",order='ascending'),title='descripción'),
    x=alt.X('value',title='cantidad'),
    color= alt.condition(
            alt.datum.value == max(df_maestro['value']),
            alt.value('red'),
            alt.value('lightgrey')
            )
).properties(title='Maestros contagiados/ no contagiados / posibles contagiados')
```

Out[12]:



In [13]:
```
print('Al final del mes salen un total de : {} maestros'.format(df_maestro[1:3]['value'].sum()))
print('Al final del mes salen un total de : {} estudiantes'.format(df_estudiantes[1:3]['value'].sum()))
```

```
Al final del mes salen un total de : 13 maestros
Al final del mes salen un total de : 438 estudiantes
```

Porcentaje de contagiados / no contagiados / posibles contagiados en la institución

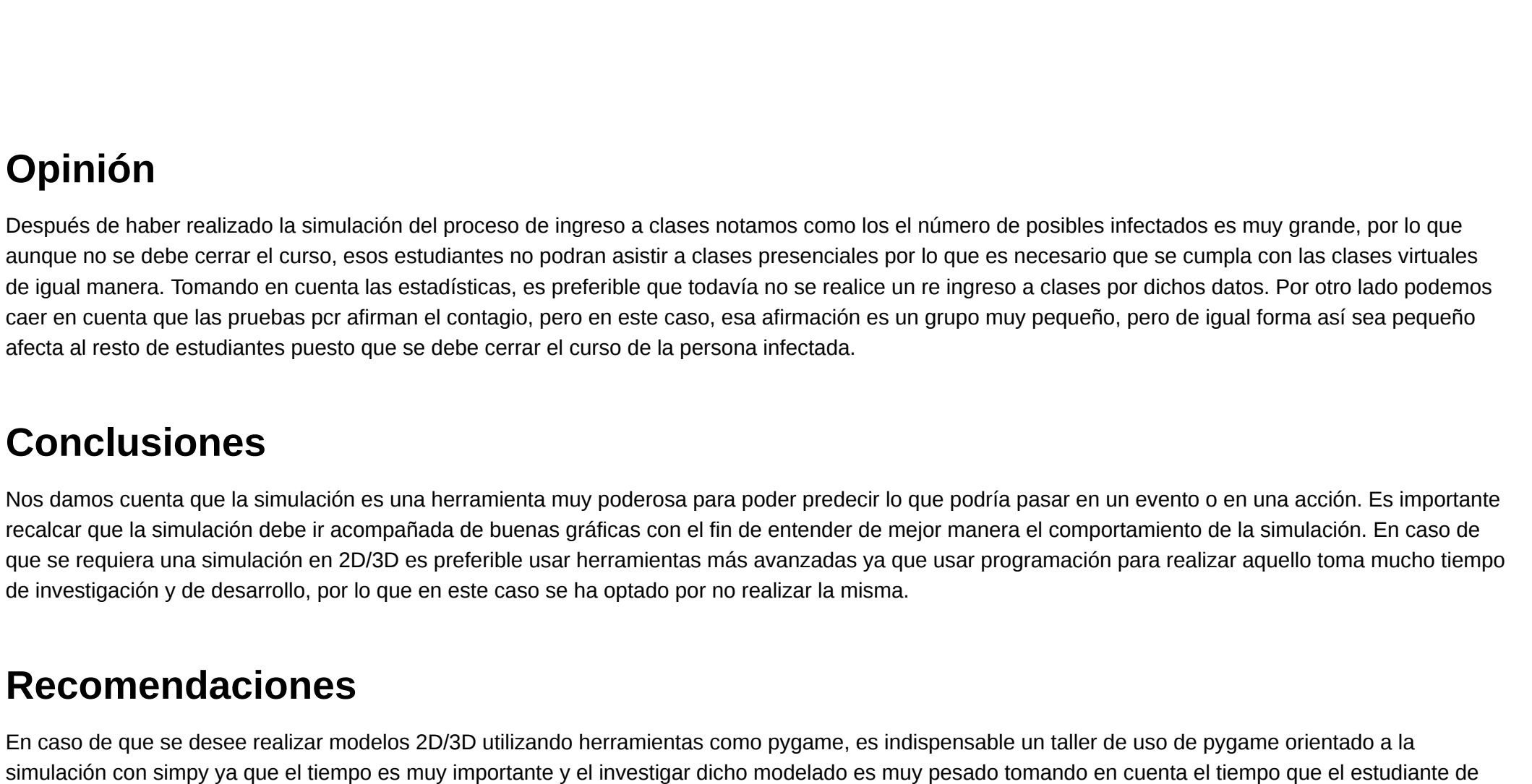
In [15]:
```
no_infectados = 0
infectados    = 0
po_infectados = 0
for i in LISTA_PERSONAS.items():
    if i[1].infectado==1:
        po_infectados+=1
    elif i[1].infectado==2:
        infectados+=1
    else:
        no_infectados+=1

df_general = pd.DataFrame([('no infectados',no_infectados/len(LISTA_PERSONAS)),
                           ('posibles infectados',po_infectados/len(LISTA_PERSONAS)),
                          ('infectados',infectados/len(LISTA_PERSONAS))],columns=['description','value'])
```

In [16]:
```
df_general
```

Out[16]:

	description	value
0	no infectados	0.553465
1	posibles infectados	0.443564
2	infectados	0.002970

In [18]:
```
fig = px.pie(df_general, values="value",
             names="description",
             color_discrete_sequence=['red','lightgrey','darkgrey'],width=1000,height=500)
fig.update_layout(title_text='contagiados / no contagiados / posibles contagiados en la institución de la Salle', ti
tle_x=0.5)
```



Opinión

Después de haber realizado la simulación del proceso de ingreso a clases notamos como los el número de posibles infectados es muy grande, por lo que aunque no se debe cerrar el curso, esos estudiantes no podran asistir a clases presenciales por lo que es necesario que se cumpla con las clases virtuales de igual manera. Tomando en cuenta las estadísticas, es preferible que todavía no se realice un re ingreso a clases por dichos datos. Por otro lado podemos caer en cuenta que las pruebas pcr afirman el contagio, pero en este caso, esa afirmación es un grupo muy pequeño, pero de igual forma así sea pequeño afecta al resto de estudiantes puesto que se debe cerrar el curso de la persona infectada.

Conclusiones

Nos damos cuenta que la simulación es una herramienta muy poderosa para poder predecir lo que podría pasar en un evento o en una acción. Es importante recalcar que la simulación debe ir acompañada de buenas gráficas con el fin de entender de mejor manera el comportamiento de la simulación. En caso de que se requiera una simulación en 2D/3D es preferible usar herramientas más avanzadas ya que usar programación para realizar aquello toma mucho tiempo de investigación y de desarrollo, por lo que en este caso se ha optado por no realizar la misma.

Recomendaciones

En caso de que se desee realizar modelos 2D/3D utilizando herramientas como pygame, es indispensable un taller de uso de pygame orientado a la simulación con simpy ya que el tiempo es muy importante y el investigar dicho modelado es muy pesado tomando en cuenta el tiempo que el estudiante de último ciclo tiene para la realización del mismo.