

<!--Universidad Industrial de Santander-->

Simulación de una epidemia

Autómatas Celulares {

<Por="Diego Barragan y Juan
David Leon"/>

}

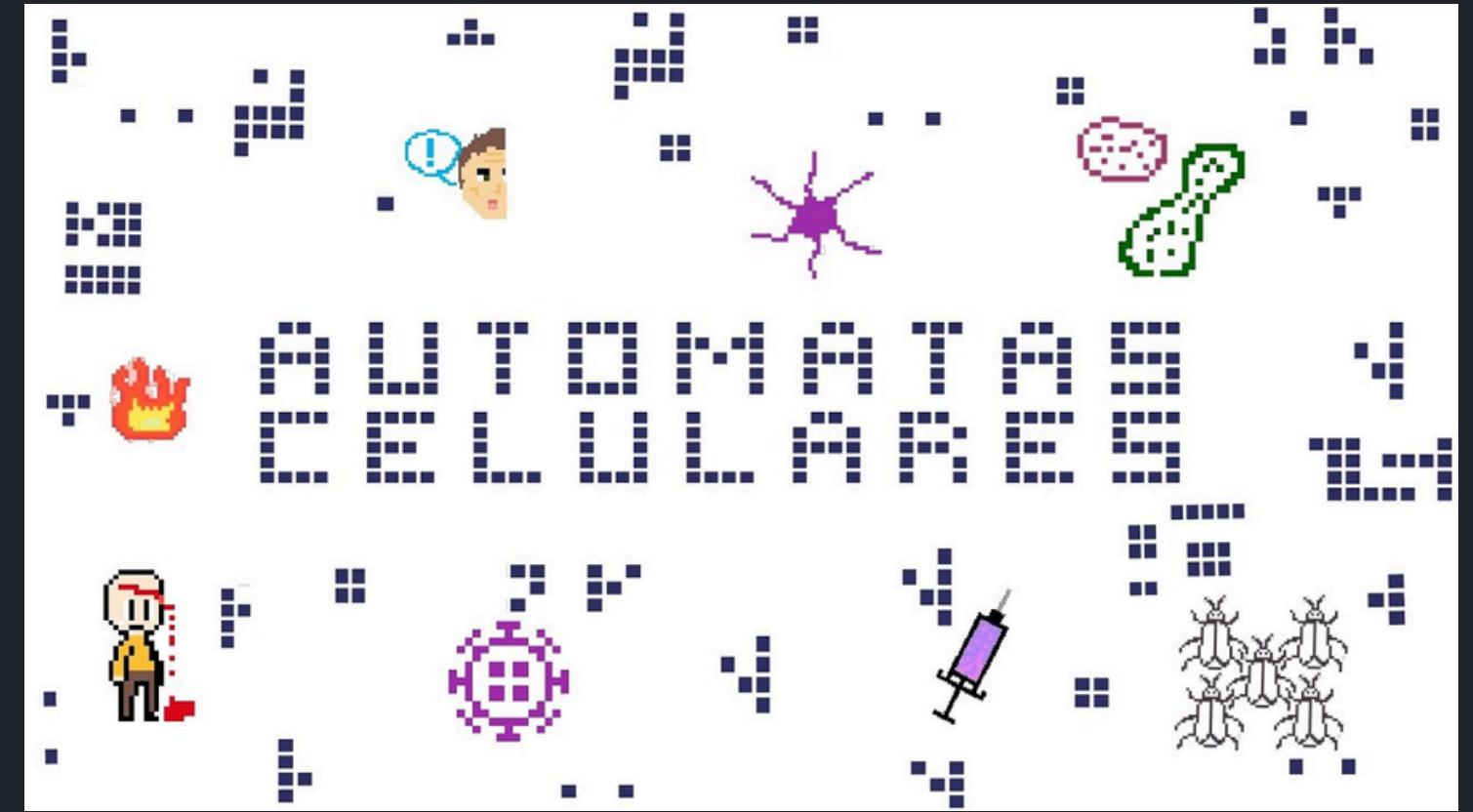


Contenidos

- 01 Abstract (resumen).
- 02 Introducción.
- 03 Propuesta.
- 04 Resultados.
- 05 Conclusiones.

Abstract (resumen) {

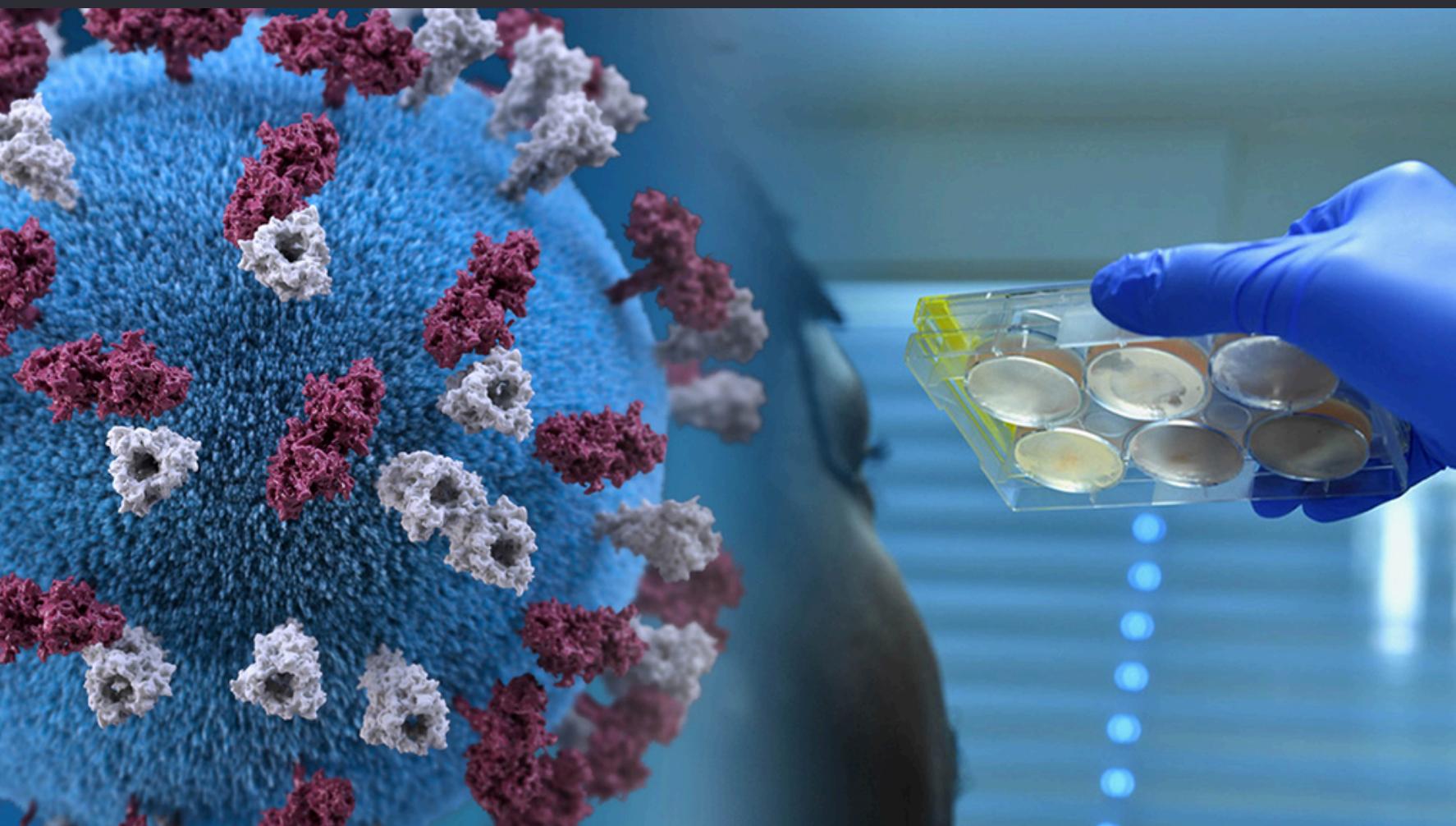
Este proyecto consiste en modelar y analizar la propagación de una epidemia utilizando autómatas celulares.



```
filter: grayscale(100%);  
color:#777;  
header #main-navigation ul li span:hover{  
  box-shadow:  
    -webkit-box-shadow:  
    -moz-box-shadow:  
    background-color:#F9F9F9;  
    color:#444;  
}  
header #main-navigation ul li.active span{  
  color:#b90000;  
}  
header #main-navigation ul li span.dashboard{  
  background:#F5F5F5 url('../img/dashboard/home.png') center  
  #main-navigation ul li span.dashboard.hover{  
    background:#F9F9F9 url('../img/dashboard/home.png') center  
    #main-navigation ul li.active span.dashboard{  
      background:#F9F9F9 url('../img/dashboard/home.png') center  
    }  
}
```

Introducción {

La propagación de enfermedades infecciosas es un fenómeno complejo que puede modelarse computacionalmente para comprender mejor su dinámica y controlar su impacto . }



Propuesta {

Se busca implementar un modelo computacional desarrollado con diferentes librerias de Python sobre la propagación de una epidemia

}

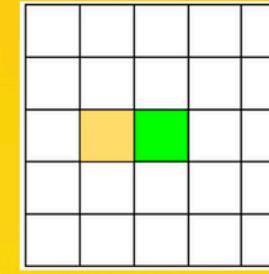
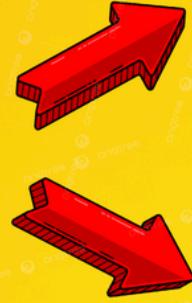
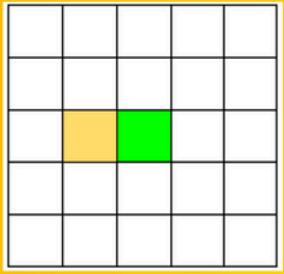


```
31 self._logger = None
32 self.fingerprints = set()
33 self.logdups = True
34 self.debug = debug
35 self.logger = logging.getLogger(__name__)
36 if path:
37     self._file = open(path, 'a', encoding='utf-8')
38     self._file.seek(0)
39     self.fingerprints.update(self._file.read().split())
40
41 @classmethod
42 def from_settings(cls, settings):
43     debug = settings.getbool('FINGERPRINT_DEBUG')
44     return cls(job_dir(settings), debug)
45
46 def request_seen(self, request):
47     fp = self.request_fingerprint(request)
48     if fp in self.fingerprints:
49         return True
50     self.fingerprints.add(fp)
51     if self._file:
52         self._file.write(fp + os.linesep)
53
54 def request_fingerprint(self, request):
55     return request_fingerprint(request)
56
```

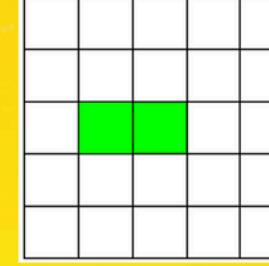

Explicación simulación de una epidemia

Estado sucesivo

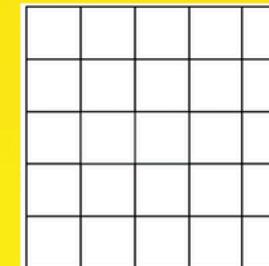
Estado precedente



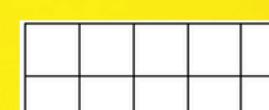
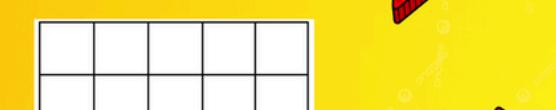
Sujeto sano interactua con infectado (no se infecta)



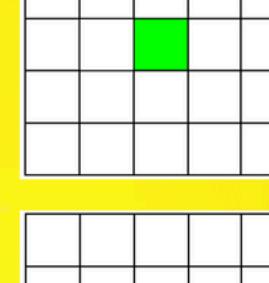
Sujeto sano interactua con infectado (se infecta)



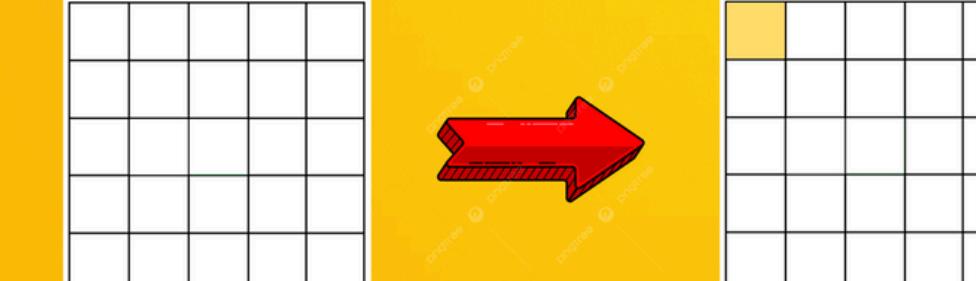
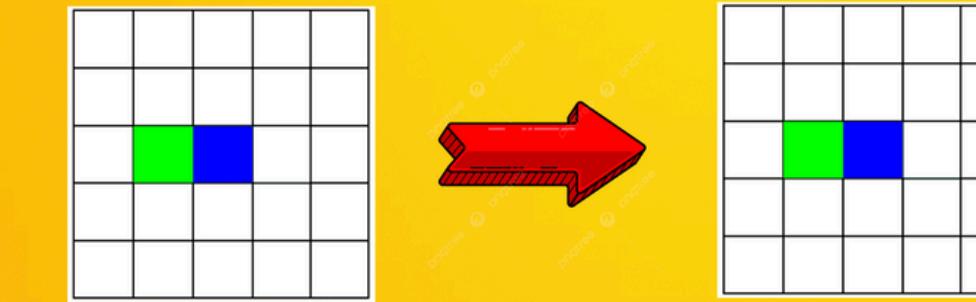
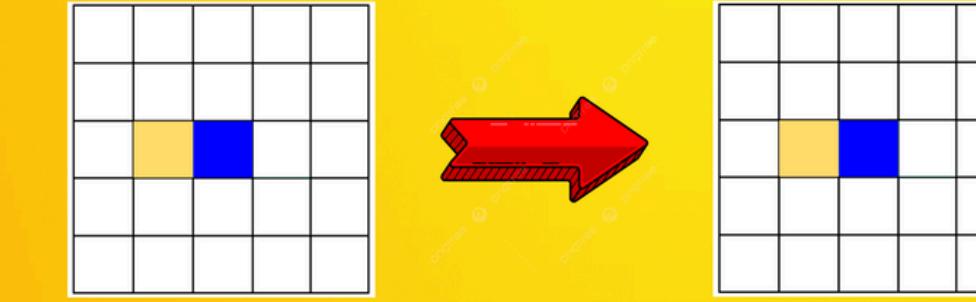
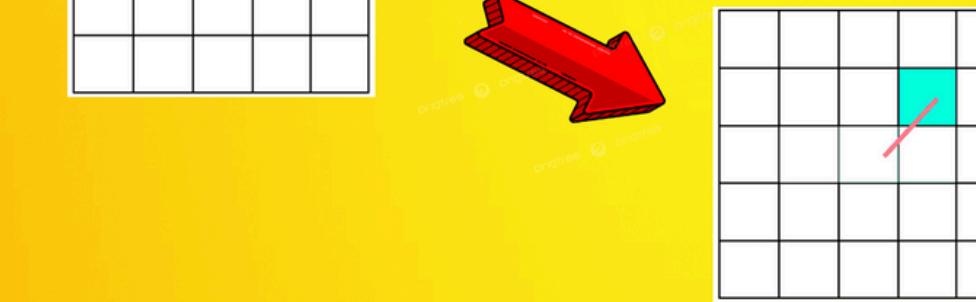
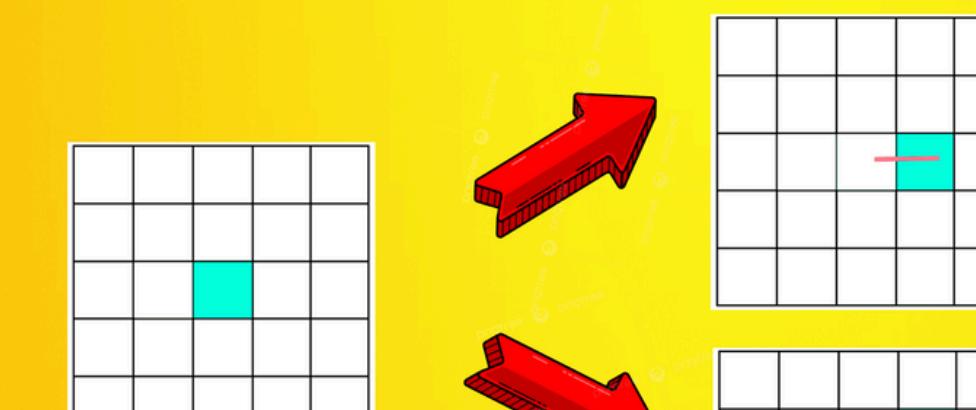
Sujeto infectado muere (queda la casilla en blanco)



Permanece infectado (no cambia la casilla)



Sujeto infectado se cura (cambia de color la casilla)



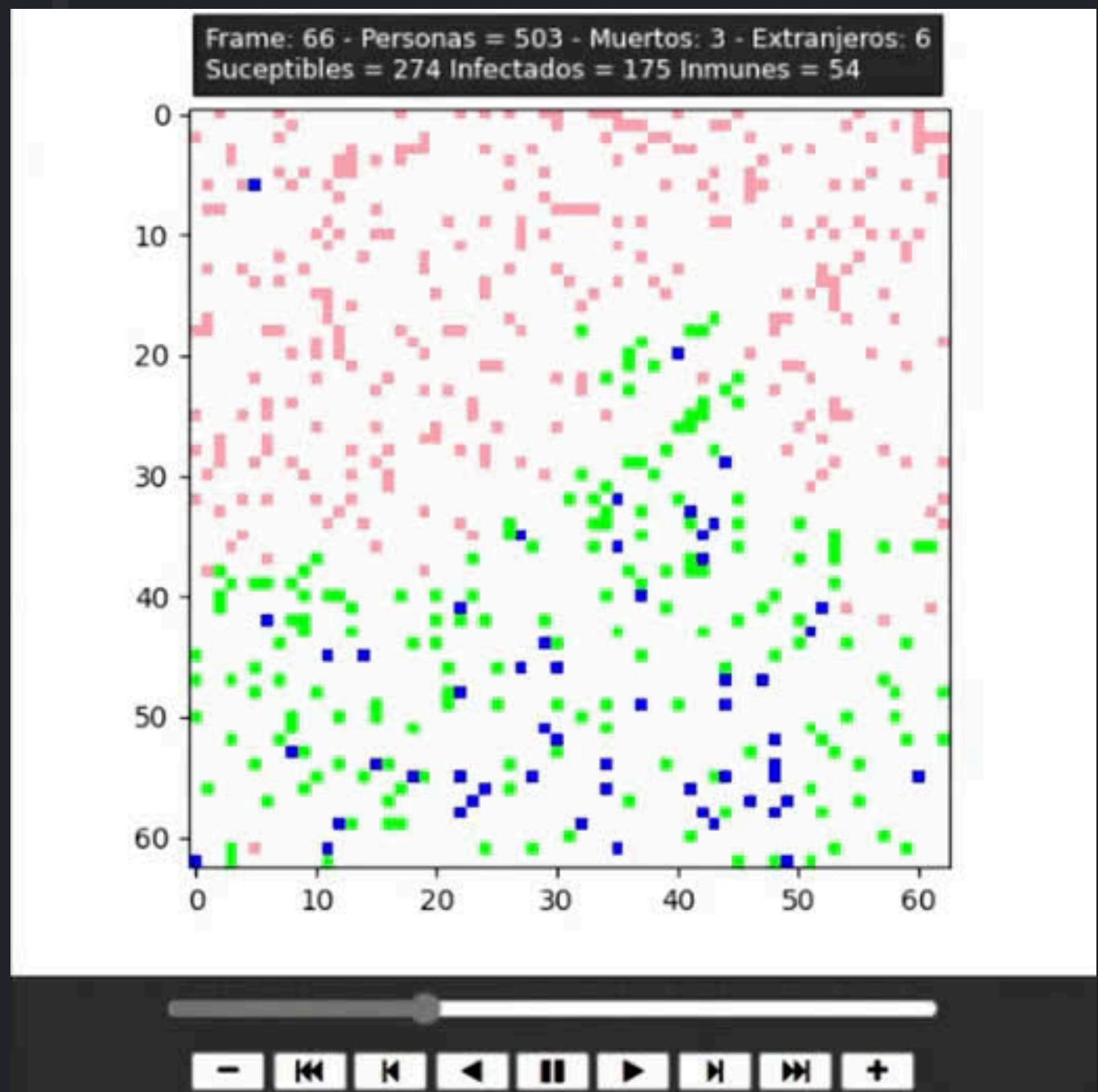
Para cualquier célula:
Se puede mover tanto recto como diagonalmente

Toda transición ocurre con cierta probabilidad la cual define el usuario antes de simular

Resultados {

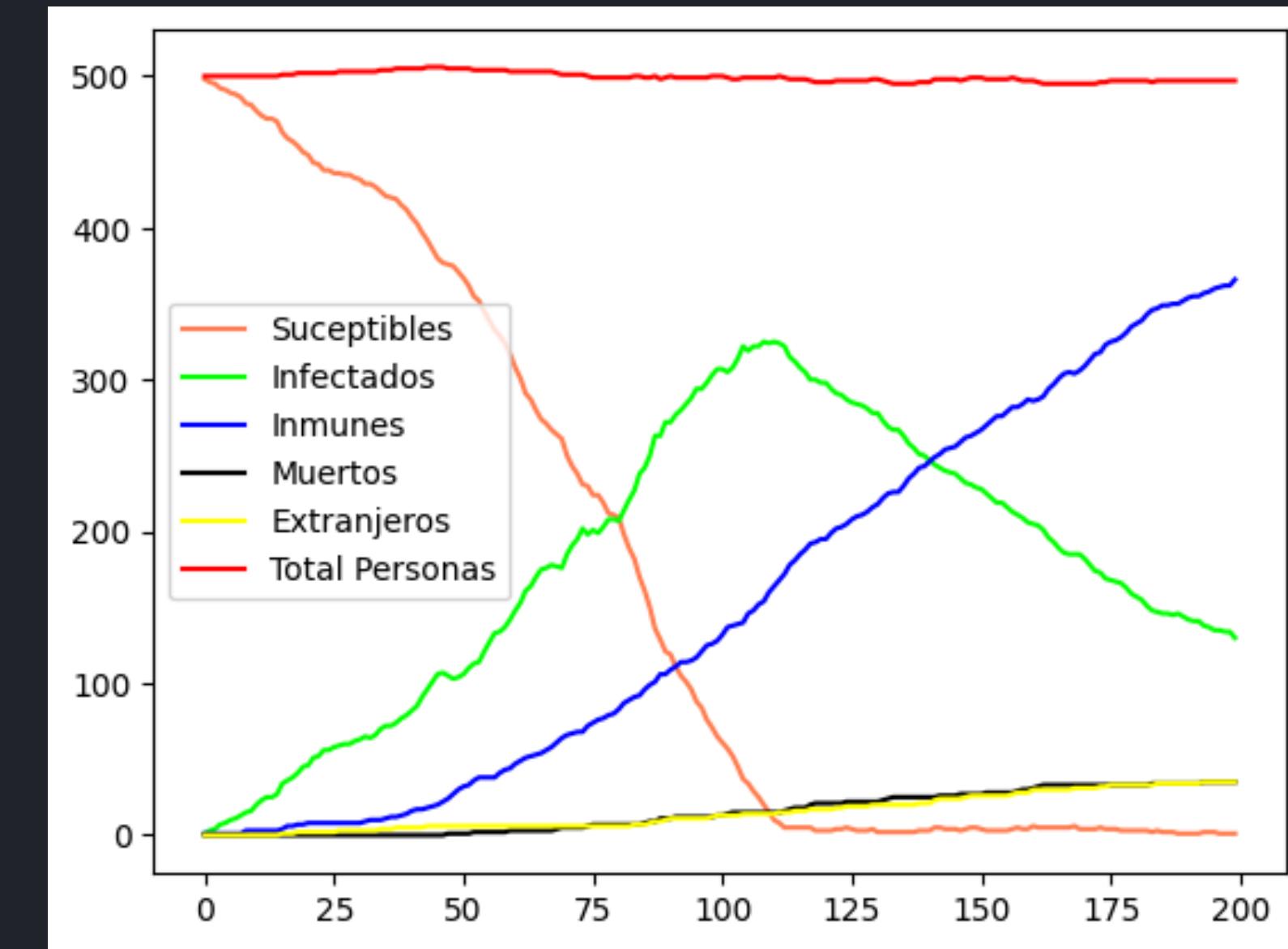
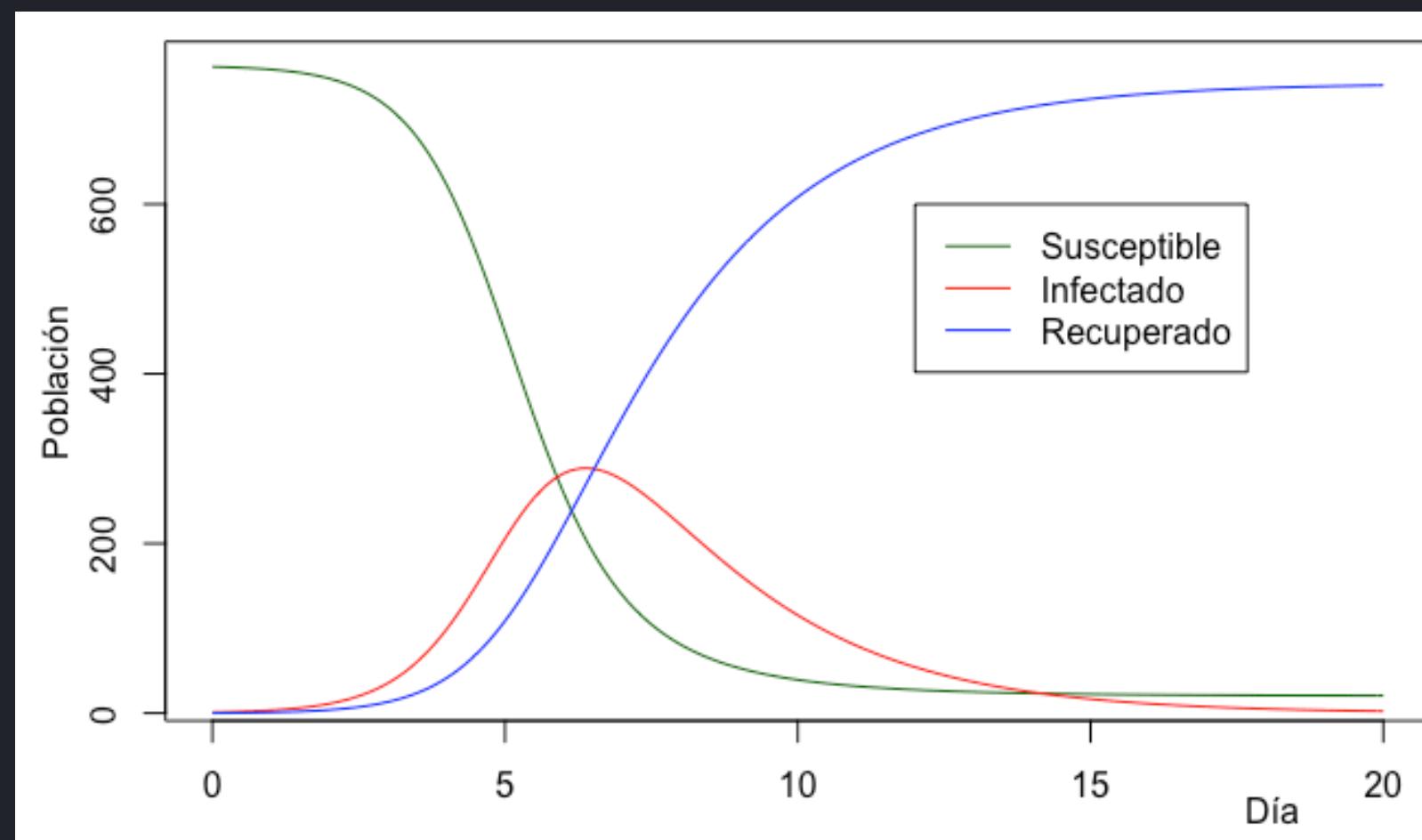
Personas Totales:	500
Infectados:	1
Inmunes:	1
Densidad:	8
Probabilidad de Contagio:	0,8
Probabilidad de Muerte:	0,001
Probabilidad de Sanidad:	0,01
Probabilidad de Nacimiento:	0,001
Pasos:	200

Simular



}

Conclusiones {



{}

<!--Universidad Industrial de Santander-->

Gracias {

<Por="Diego Barragan y Juan
David Leon"/>

}