

## CodeVID

*Profesor:* Eddy Ramírez*Estudiantes:*Rodrigo Venegas,  
Johan Herrera,*Carnés:*2017047627  
2016133925

## Introducción:

Uno de los temas que se encuentran en este periodo de pandemia, como una necesidad real donde el trabajo de computación se ha vuelto mucho más especializado, es el análisis y creación de herramientas para atender la emergencia provocada por el nuevo Coronavirus (SARS-CoV-2), responsable de generar la enfermedad llamada COVID-19 (Coronavirus disease 2019), que ha puesto en jaque a muchos países, jefes de estado y ha tenido un impacto muy serio en nuestras vidas y en la economía en general.

En este momento, se tienen computadoras que han hecho modelos tridimensionales de la estructura biológica del virus.

Para entender mejor el modo en que se propaga este nuevos virus y sus reacciones ante diferentes agentes, los estudiantes del curso de Investigación de Operaciones deben de realizar un proyecto programado que simule, la propagación de esta enfermedad tomando en cuenta diferentes configuraciones y agente limitantes.

## Resumen Ejecutivo:

CodeVID es un proyectos que, utilizando la teconología que ofrece Processing y la utilidad del paradigma orientado a objetos, pretende simular diferentes escenarios en los cuales se pueda poner a prueba del desarrollo del nuevo coronavirus. CodeVID simula detalladamente la forma de actuar de la actual pandemia ante diversas situaciones.

Utilizando diversos parámetros de configuración, CodeVID puede ayudar a predecir el comportamiento del nuevo coronavirus en una pequeña población que pueda ser representada con los diferentes agentes que actúan durante la ejecución del programa. Ante la actual pandemia, un porgrama como CodeVID puede ser esencial para mitigar los riesgos y el impactos que podría causar esta enfermedad en el mundo.

## Descripción de la solución:

**Processing:** Es un lenguaje de programación y entorno de desarrollo integrado de código abierto basado en Java, de fácil utilización, y que sirve como medio para la enseñanza y producción de proyectos multimedia e interactivos de diseño digital.[1]

La decisión de tomar este lenguaje, es gracias a su versatilidad con respecto a sus funciones ya integradas para el funcionamiento de animaciones, fue todo un reto, ya que es algo nuevo para nosotros, pero con sus similitud a la sintaxis de Java, fue más sencilla su comprensión.

**Orientación a objetos:** Se tomó la decisión de trabajar con orientación a objetos por la naturaleza del problema, por la manera en que nos es posible modelar cada componente del proyecto y la forma en podemos relacionar cada uno de ellos. Se trabajan diferentes clases para llegar a solución, entre ellas podemos mencionar las más importantes: Punto, Lienzo, Tex.

**Funcionamiento:** Processing nos permite trabajar utilizando Orientación a Objetos con la sintaxis de java (con algunas excepciones). De esta formar se creó la clase **Punto** de la cual extienden las clases **pRecto**, **pEstacional**, **pAleatorio**, **pEstatico**, de esta forma podemos coleccionar todos los puntos como simples objetos de tipo Puntos, además se evita la duplicación de código. Cada puntos se mueve o actúa de una forma diferente por lo que cada uno de ellos presenta una modificación de el método **move()** o **show()**, en su defecto. La clase Lienzo se encargar de modelar toda la ventana, todas las configuraciones del mapa y las paredes utilizando pixeles. En el Lienzo es dónde los Puntos van a actuar y todo el proyecto se va a desarrollar.

La clase Tex, extrae información del proyecto constantemente para generar graficos estadísticos, con el fin de

obtener información detallada del actuar del virus según diferentes configuraciones.

En el **Sketch**, archivo encargado de la ejecución del proyecto, corre un ciclo infinito con el que se da vida a toda la animación. Desde este archivo se obtiene la configuración de los diferentes módulos del proyecto, mapa, agentes, virus. Al mismo tiempo, en variables globales se almacena toda esta información con el objetivo de enviarla a las partes que la estén ocupando.

Para hacer todo más agradable a la vista se agregó un radio fijo, igual a 5, a todos los puntos para hacerlos más grandes y poder observar su colisión con más detalle.

Finalmente, la colisión. Aprovechando el tamaño de los puntos, la colisión no se da en un solo pixel, sino en varios (10, el diámetro de cada punto), de esta manera es posible observar más colisiones durante la ejecución.

## Ejemplos de prueba:

Para sintetizar los datos, se muestran en una tabla las configuraciones más importantes que se aplican a cada caso, los ejemplos completos se encuentran en los respectivos archivos de configuración.

Todas las pruebas se corren durante 25s

### Ejemplo de prueba 1:

Los archivos de esta prueba son:  
(agentesConfig1, codevidConfig1, mapaConfig1)

<i>Config</i>	<i>Grado</i>
Agentes Rectos	2
Agentes Estacionales	3
Agentes Aleatoios	2
Agentes Estáticos	40
Infectados	3 - Estacionales
Tamaño de mapa	Grande
Paredes	0
Probabilidad de contagio	99.5%
Reinfección	20%
Probabilidad de muerte	8%

### Ejemplo de prueba 2:

Los archivos de esta prueba son:  
(agentesConfig2, codevidConfig2, mapaConfig2)

<i>Config</i>	<i>Grado</i>
Agentes Rectos	15
Agentes Estacionales	15
Agentes Aleatoios	15
Agentes Estáticos	10
Infectados	10 - Estáticos
Tamaño de mapa	Grande
Paredes	2
Probabilidad de contagio	90%
Reinfección	30%
Probabilidad de muerte	5%

### Ejemplo de prueba 3:

Los archivos de esta prueba son:  
(agentesConfig3, codevidConfig3, mapaConfig3)

<i>Config</i>	<i>Grado</i>
Agentes Rectos	2
Agentes Estacionales	40
Agentes Aleatoios	40
Agentes Estáticos	50
Infectados	2 - Rectos
Tamaño de mapa	Grande
Paredes	0
Probabilidad de contagio	95%
Reinfección	0%
Probabilidad de muerte	10%

#### **Ejemplo de prueba 4:**

Los archivos de esta prueba son:  
(agentesConfig4, codevidConfig4, mapaConfig4)

<i>Config</i>	<i>Grado</i>
Agentes Rectos	10
Agentes Estacionales	10
Agentes Aleatoios	10
Agentes Estáticos	10
Infectados	10 - Aleatorios
Tamaño de mapa	Grande
Paredes	2
Probabilidad de contagio	99%
Reinfección	50%
Probabilidad de muerte	30%

#### **Ejemplo de prueba 5:**

Los archivos de esta prueba son:  
(agentesConfig5, codevidConfig5, mapaConfig5)

<i>Config</i>	<i>Grado</i>
Agentes Rectos	5
Agentes Estacionales	20
Agentes Aleatoios	50
Agentes Estáticos	15
Infectados	5 - Rectos
Tamaño de mapa	Grande
Paredes	2
Probabilidad de contagio	60%
Reinfección	1
Probabilidad de muerte	3%

#### **Ejemplo de prueba 6:**

Los archivos de esta prueba son:  
(agentesConfig6, codevidConfig6, mapaConfig6)

<i>Config</i>	<i>Grado</i>
Agentes Rectos	20
Agentes Estacionales	3
Agentes Aleatoios	20
Agentes Estáticos	1
Infectados	1-Estático
Tamaño de mapa	Pequeño
Paredes	0
Probabilidad de contagio	90%
Reinfección	1
Probabilidad de muerte	80%

#### **Ejemplo de prueba 7:**

Los archivos de esta prueba son:  
(agentesConfig7, codevidConfig7, mapaConfig7)

<i>Config</i>	<i>Grado</i>
Agentes Rectos	1
Agentes Estacionales	2
Agentes Aleatoios	1
Agentes Estáticos	1
Infectados	1 - Recto
Tamaño de mapa	Pequeño
Paredes	0
Probabilidad de contagio	90%
Reinfección	0
Probabilidad de muerte	80%

#### **Ejemplo de prueba 8:**

Los archivos de esta prueba son:  
(agentesConfig8, codevidConfig8, mapaConfig8)

<i>Config</i>	<i>Grado</i>
Agentes Rectos	15
Agentes Estacionales	40
Agentes Aleatoios	50
Agentes Estáticos	5
Infectados	5 - Estáticos
Tamaño de mapa	Mediano
Paredes	4
Probabilidad de contagio	90%
Reinfección	1
Probabilidad de muerte	60%

#### **Ejemplo de prueba 9:**

Los archivos de esta prueba son:  
(agentesConfig9, codevidConfig9, mapaConfig9)

<i>Config</i>	<i>Grado</i>
Agentes Rectos	15
Agentes Estacionales	40
Agentes Aleatoios	50
Agentes Estáticos	5
Infectados	5 - Estáticos
Tamaño de mapa	Mediano
Paredes	4
Probabilidad de contagio	90%
Reinfección	1
Probabilidad de muerte	4%

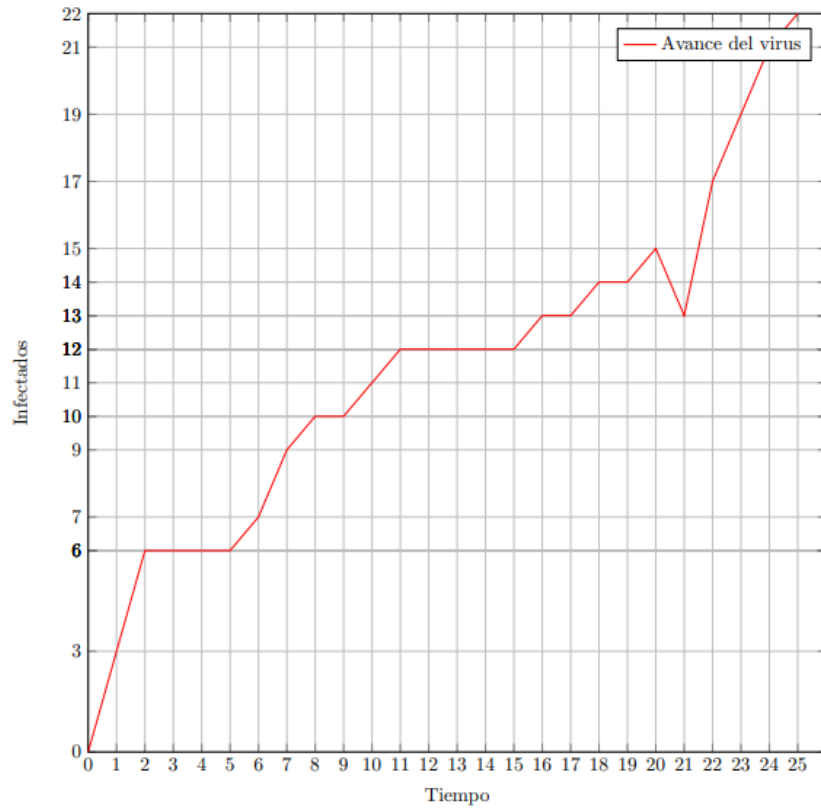
### **Ejemplo de prueba 10:**

Los archivos de esta prueba son:  
(agentesConfig10, codevidConfig10, mapaConfig10)

<i>Config</i>	<i>Grado</i>
Agentes Rectos	1
Agentes Estacionales	2
Agentes Aleatoios	1
Agentes Estáticos	1
Infectados	1 - Recto
Tamaño de mapa	Pequeño
Paredes	0
Probabilidad de contagio	90%
Reinfección	0
Probabilidad de muerte	4%

## **Resultados de las pruebas:**

### **Resultados - Ejemplo de prueba 1:**



Resultados - Ejemplo de prueba 2:

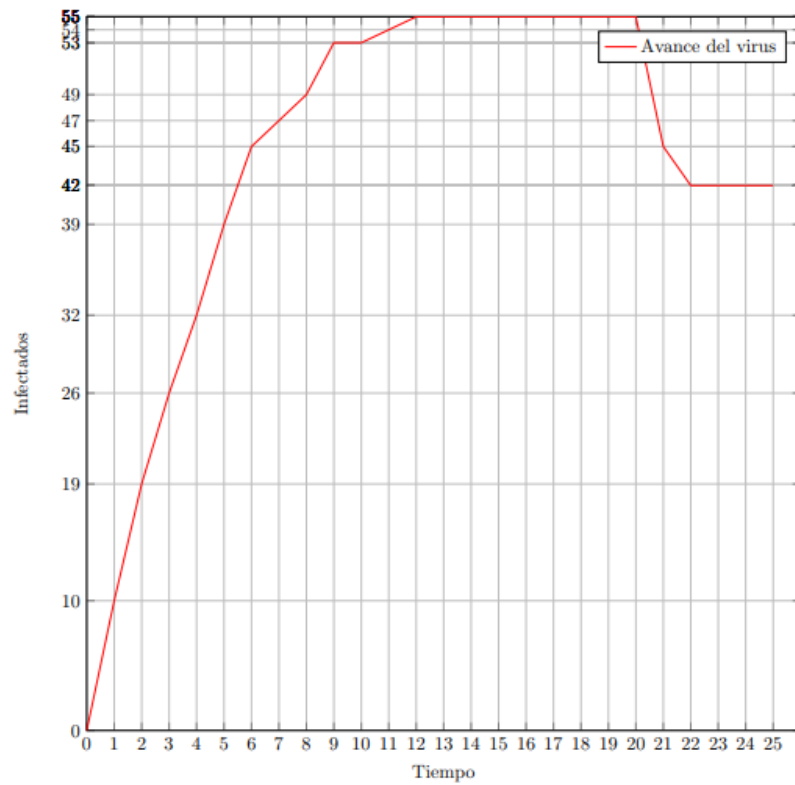
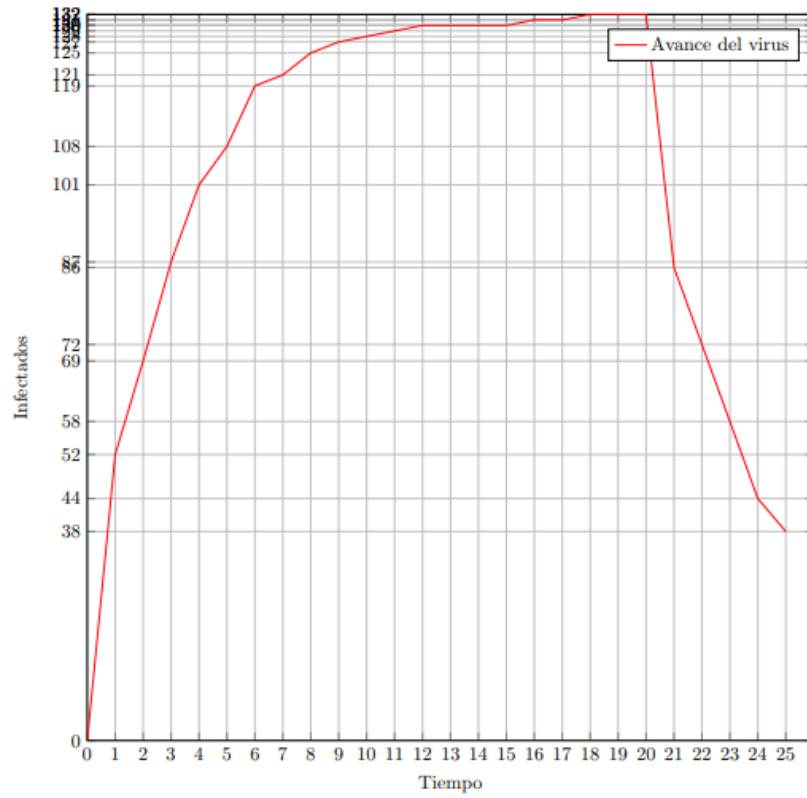
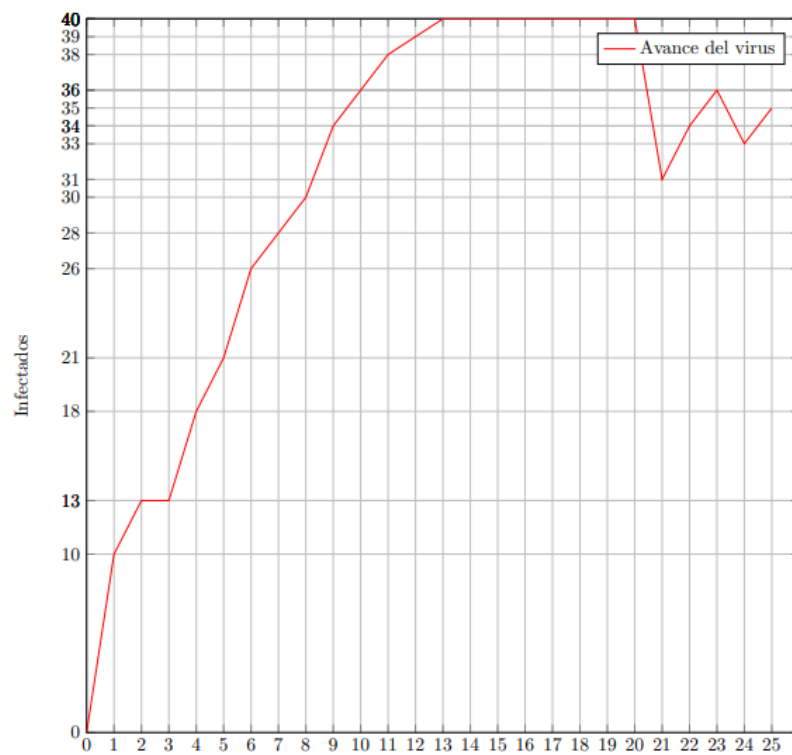


Figure 1: Diagrama Final

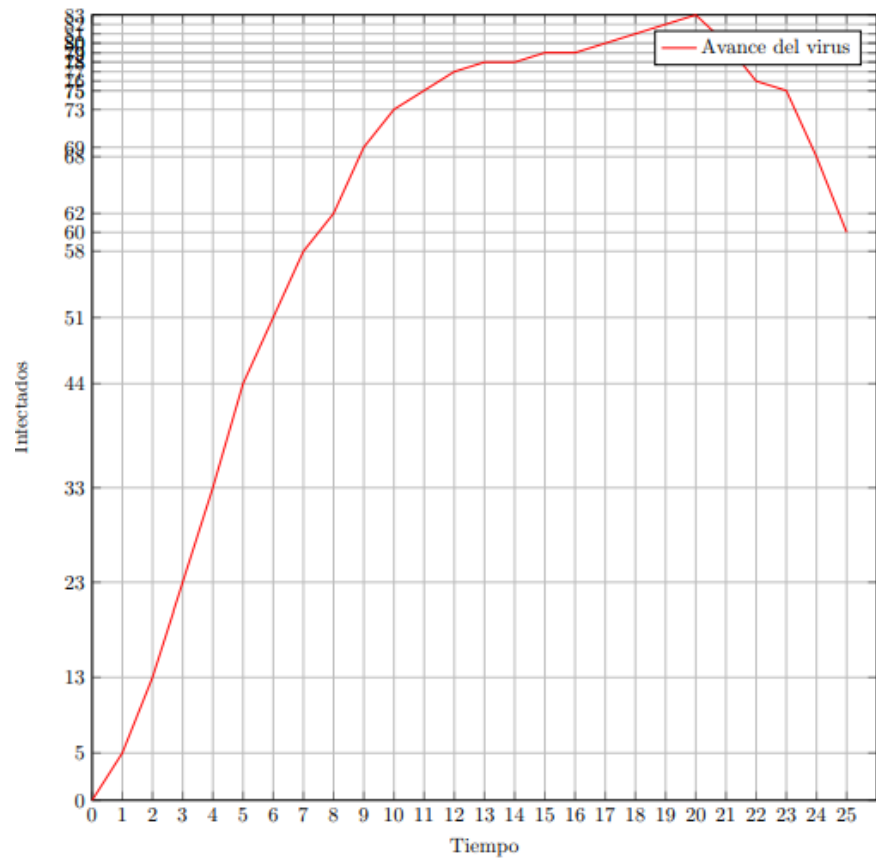
Resultados - Ejemplo de prueba 3:



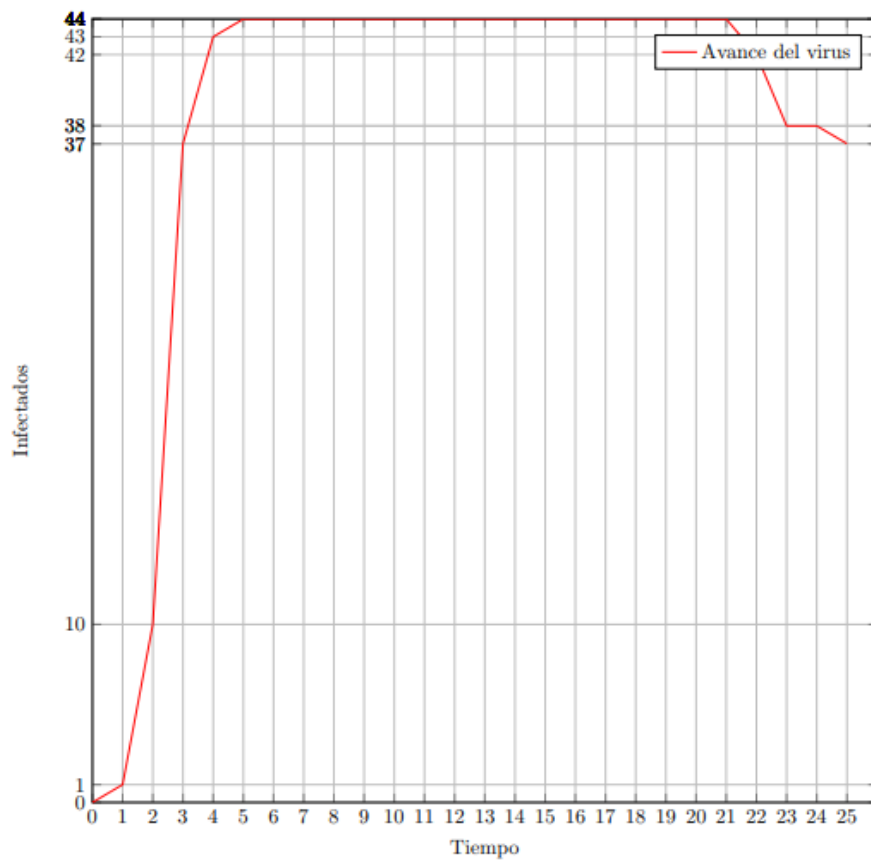
Resultados - Ejemplo de prueba 4:



Resultados - Ejemplo de prueba 5:

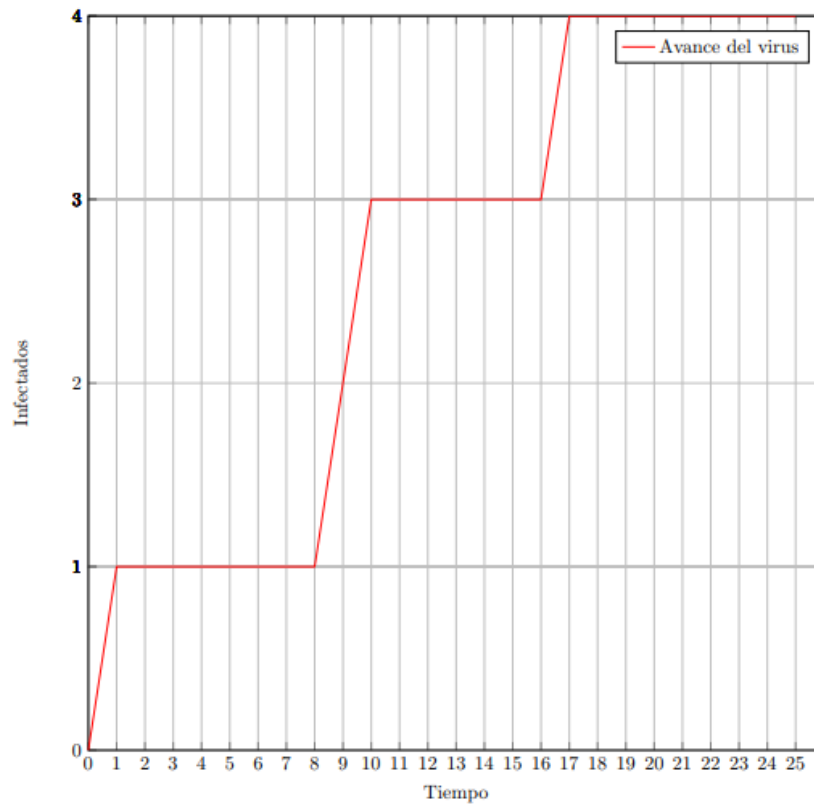


Resultados - Ejemplo de prueba 6:

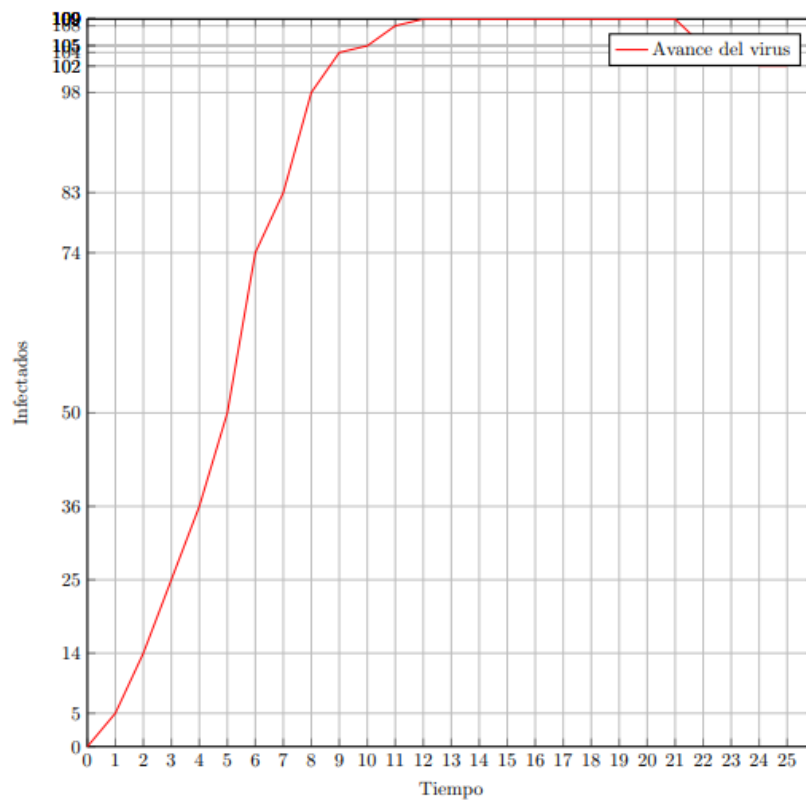


Resultados - Ejemplo de prueba 7:

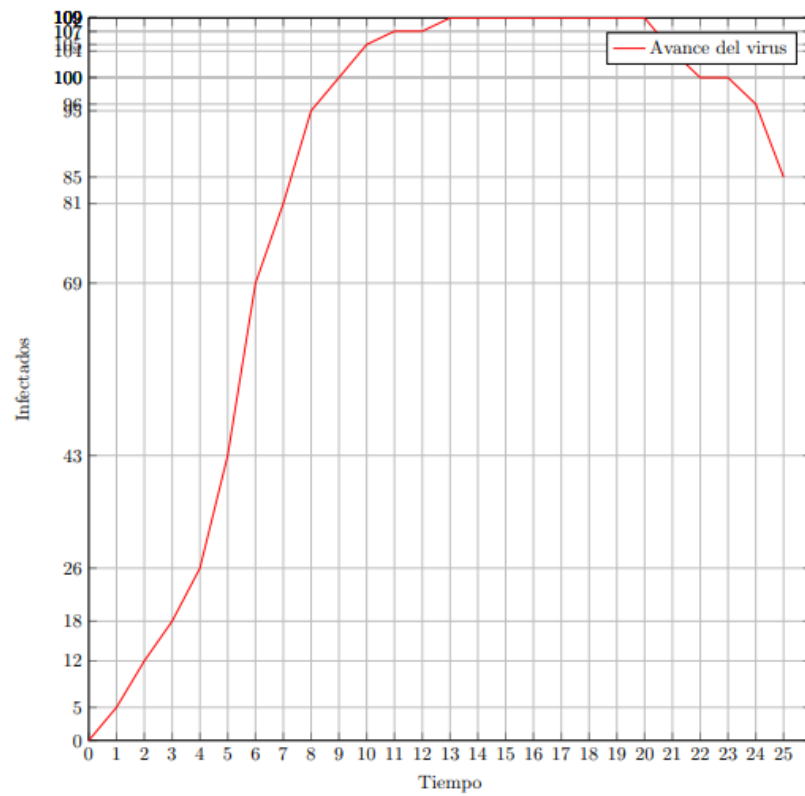




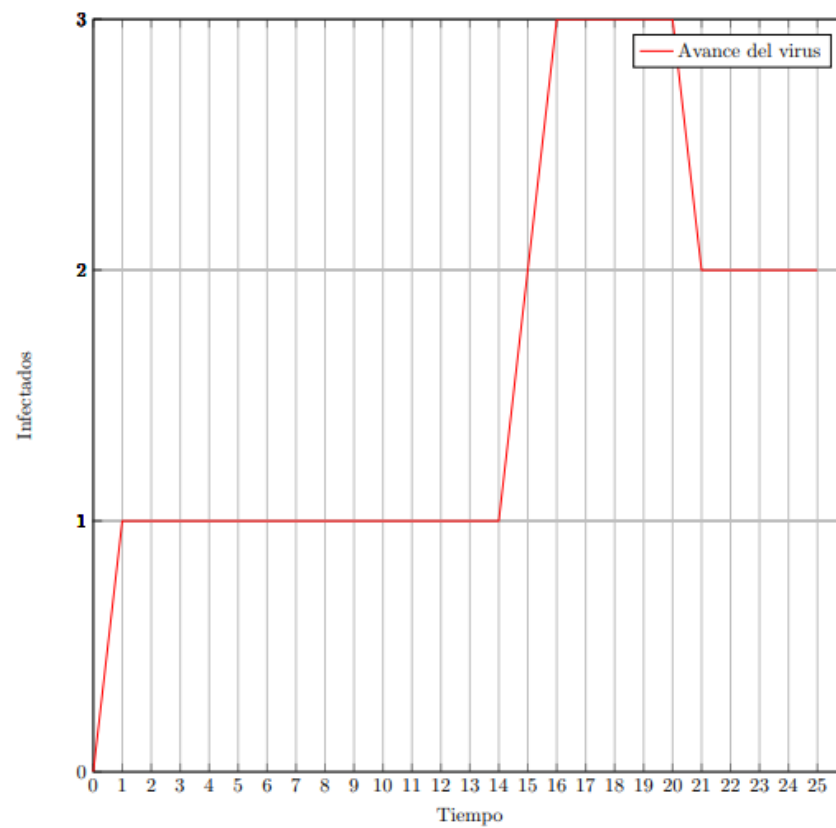
Resultados - Ejemplo de prueba 8:



Resultados - Ejemplo de prueba 9:



Resultados - Ejemplo de prueba 10:



## Conclusiones:

Durante el proyecto, los integrantes del equipo aprendieron en todos los sentidos, se aprendió a trabajar con

**Processing**, se trabajó por primera vez con una animación y un ciclo infinito. Además los estudiantes pudieron poner en práctica sus habilidades de optimización de algoritmos al enfrentarse con un programa que podría ralentizarse ante la presencia de muchos ciclos. Además de todo el aprendizaje, el proyecto ayudó al equipo a entender mejor el modo de propagación y actuar del nuevo Coronavirus.

Con respecto al proyecto en general, fue todo un reto, ya que al utilizar un nuevo lenguaje, fue bastante complicado llegar a una solución concreta, aparte de esto, se fue sobrellevando de una manera correcta. También podríamos decir que los beneficios que brinda este lenguaje fue de bastante provecho para el desarrollo del mismo. Se puede ver que la animación cumple su funcionalidad de infección, aparte se llega a ver como los resultados de una pandemia real se logra apreciar en el programa.

## Recomendaciones:

Con las recomendaciones de este proyecto, es tomar algún lenguaje ya utilizado sencillo y fácil de entender para la creación del mismo, ya que nosotros a pesar de que usamos un lenguaje nuevo, fue sencillo de comprender puesto que se trabajó con una extensión de Java.

También la división de trabajo fue buena, y cuando se trabajó en equipo fue aún más satisfactorio los avances, por lo que se recomienda asignar tareas cortas individualmente y asignar tareas grandes con pequeñas reuniones para la toma de decisión del diseño.

Además recomendamos tomarse el tiempo de leer de manera precisa la documentación, ya que tuvimos unos fallos a la hora de no implementar unas funcionalidades pequeñas que dejamos pasar, pero por suerte nos dimos cuenta a tiempo.

Se recomienda modular el problema en subproblemas, y hacer la división de manera que se logre apreciar bien el embolado para que se sepa cual es la correcta división de clases, para evitar la duplicación de código.

## Bibliografía:

Books Processing.org. (2014, diciembre). Recuperado de <https://processing.org/books/>

## Referencias:

colaboradores de Wikipedia. (2020, 16 abril). Processing. Recuperado de <https://es.wikipedia.org/wiki/Processing>  
[1]