ELO-329

TAREA 2: Simulación teórica, gráfica y visual del desarrollo de una pandemia.

Joaquín Aguilera
Felipe Bustos
Nicolás Capetillo
Robert Parra

El objetivo de esta tarea es adaptar y extender lo que se realizó en la tarea 1, pero ahora utilizando la librería JavaFX para poder ejecutar el programa fuera de la línea de comando, es decir, con una interfaz gráfica y poder observar el comportamiento de una situación inspirada en una pandemia tipo COVID-19. Para organizar las interfaces se utilizó el modelo vista controlador (MVC), con esto se podrá ver gráficamente las diferentes etapas solicitadas de la tarea.

A medida que se realizaron las etapas correspondientes, aparecen problemáticas que tomaron significante tiempo para darle solución.

Una de las primeras dificultades que se presentaron en el desarrollo de la tarea, fue el abrir otra ventana al presionar el menú “Settings”. Después de que se leyera significante una cantidad de documentación, se procedió a crear un “menuItem”, el cual contenía un “eventHandler” en caso de ser activado. Esto hace que se inicie una nueva escena con su propia ventana (“Stage”), la cual muestra los ajustes correspondientes.

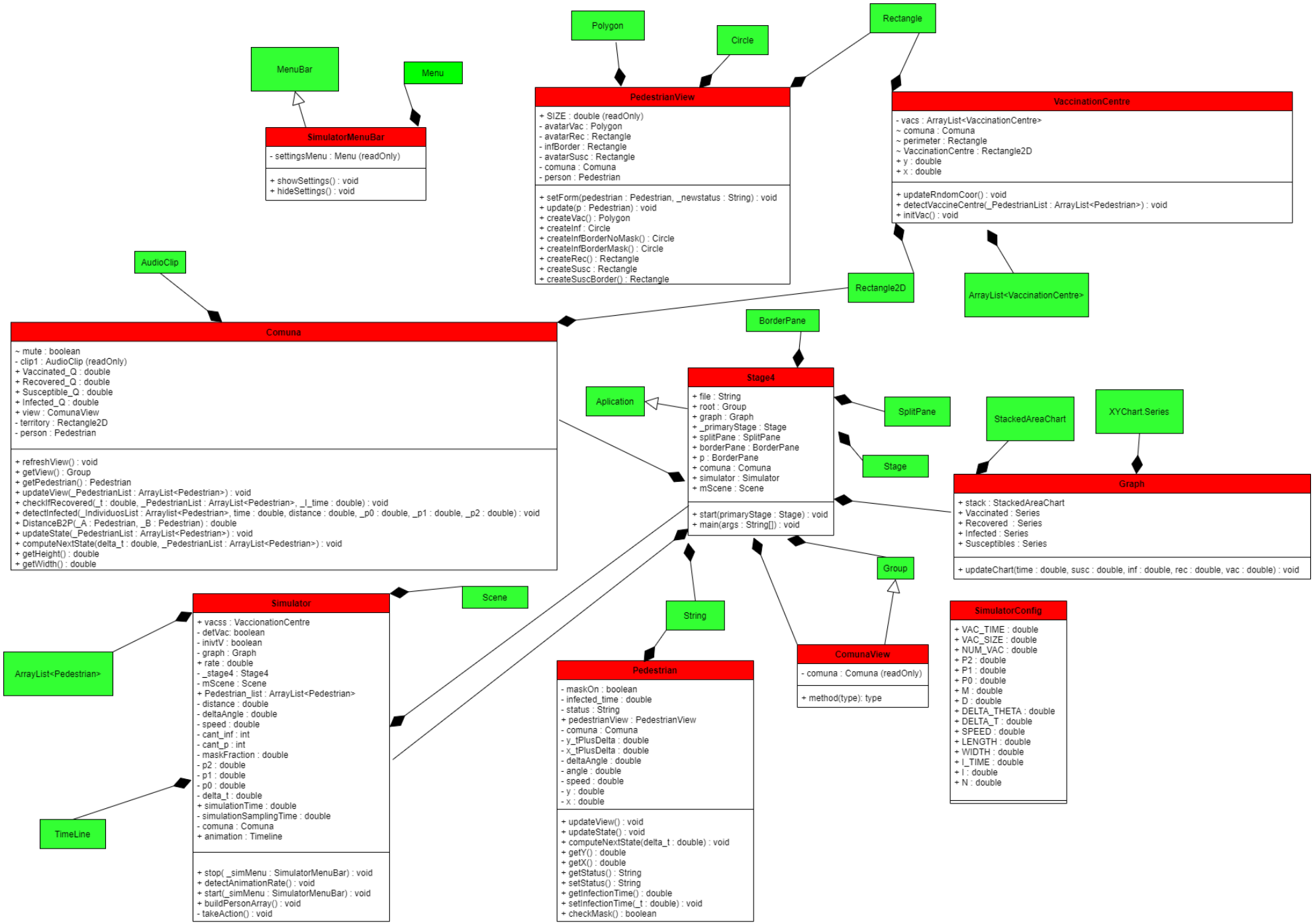
Otra dificultad presente en el desarrollo de las etapas, fue el hacer que la simulación se reinicialice con los nuevos parámetros seleccionados por el usuario en la ventana “Settings”. Posteriormente fue resuelto inicializando una nueva instancia de la clase Simulador, en el mismo espacio de memoria que el objeto anterior, lo que provoca que siga corriendo el programa, pero con distintos parámetros. Lo anteriormente expuesto es posible de visualizar en el “eventHandler” del “menuItem” “Parameters”, de la clase “SimulatorMenuBar”.

Por último, el hecho de que el gráfico se fuese actualizando con los nuevos datos entregados por la simulación no fue tema fácil. Se intentó ocupar distintos métodos que vienen con la librería correspondiente, pero lo que finalmente hizo efecto, fue que el método Simulador añadiera los nuevos datos al gráfico a medida que avanzaba la simulación.

Es importante mencionar que mientras avanzaba el desarrollo de la tarea, se produjeron constantes problemas relacionados con la configuración del proyecto en el IDE (IntelliJ), lo cual corresponde a una dificultad también.

ELO-329

Diagrama UML



ELO-329

Descripción UML

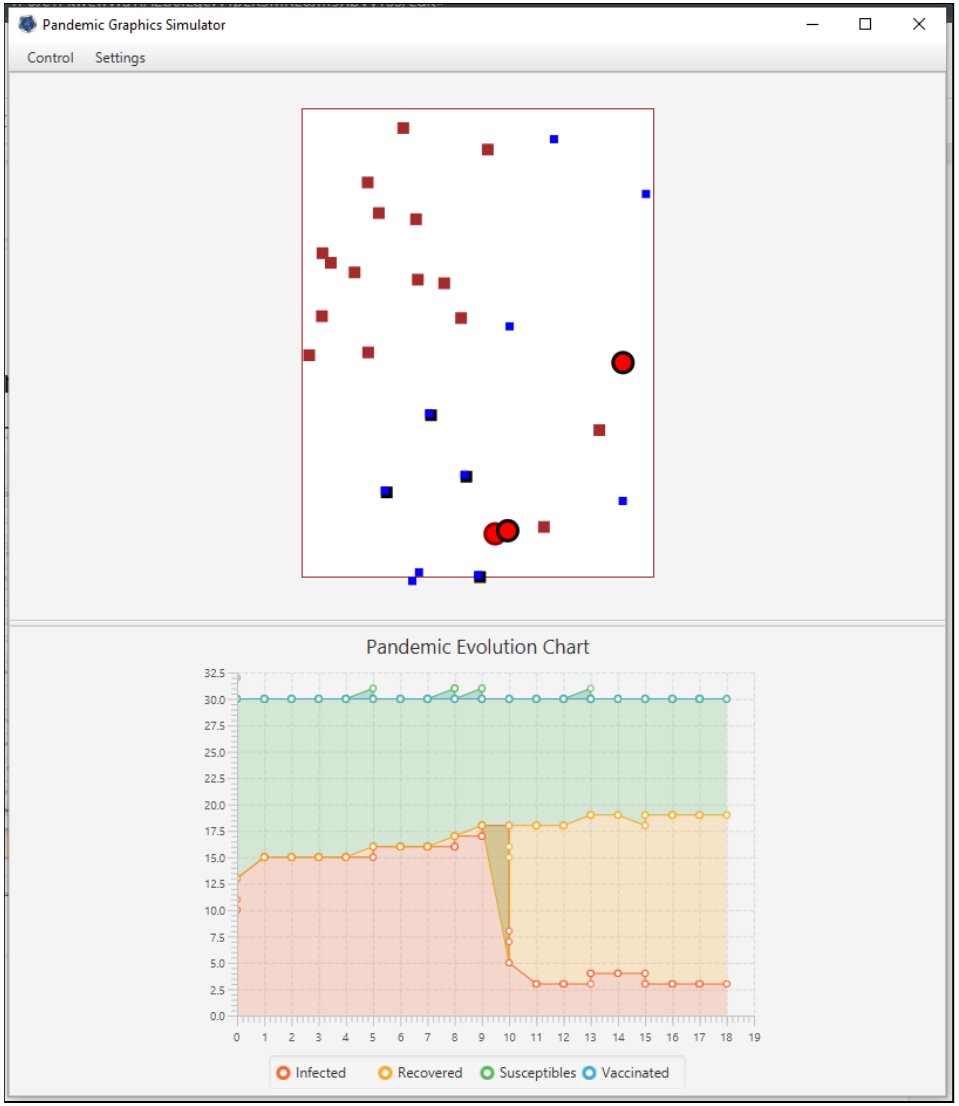
- En este diagrama se contemplan tres tipos de relaciones entre clases: composición, agregado y herencia.
    - Estos tipos existen debido a que hay relaciones entre clases y tipos de datos no primitivos que dependen de su existencia para tener un motivo dentro del código (composición). Por ejemplo, el tipo de dato “Polygon” tiene su motivo de utilización gracias a “PedestrianView”, si esta clase desaparece, “Polygon” desaparece del diagrama.
    - También hay relaciones de tipo de agregación, puesto que hay clases/atributos que son agregados a partir de otros, por ejemplo, el tipo de dato “application” nace a partir de “Stage4”, que es el main encargado de ejecutar el código, y dentro de “Stage4”, se hace uso del tipo de dato “application”.
    - La relación heredada existe debido a que hay clases que se extienden de una clase padre, la cual contiene atributos y métodos a los cuales la clase hija puede acceder, por ejemplo, “SimulatorMenuBar” es una clase heredada de la clase MenuBar.

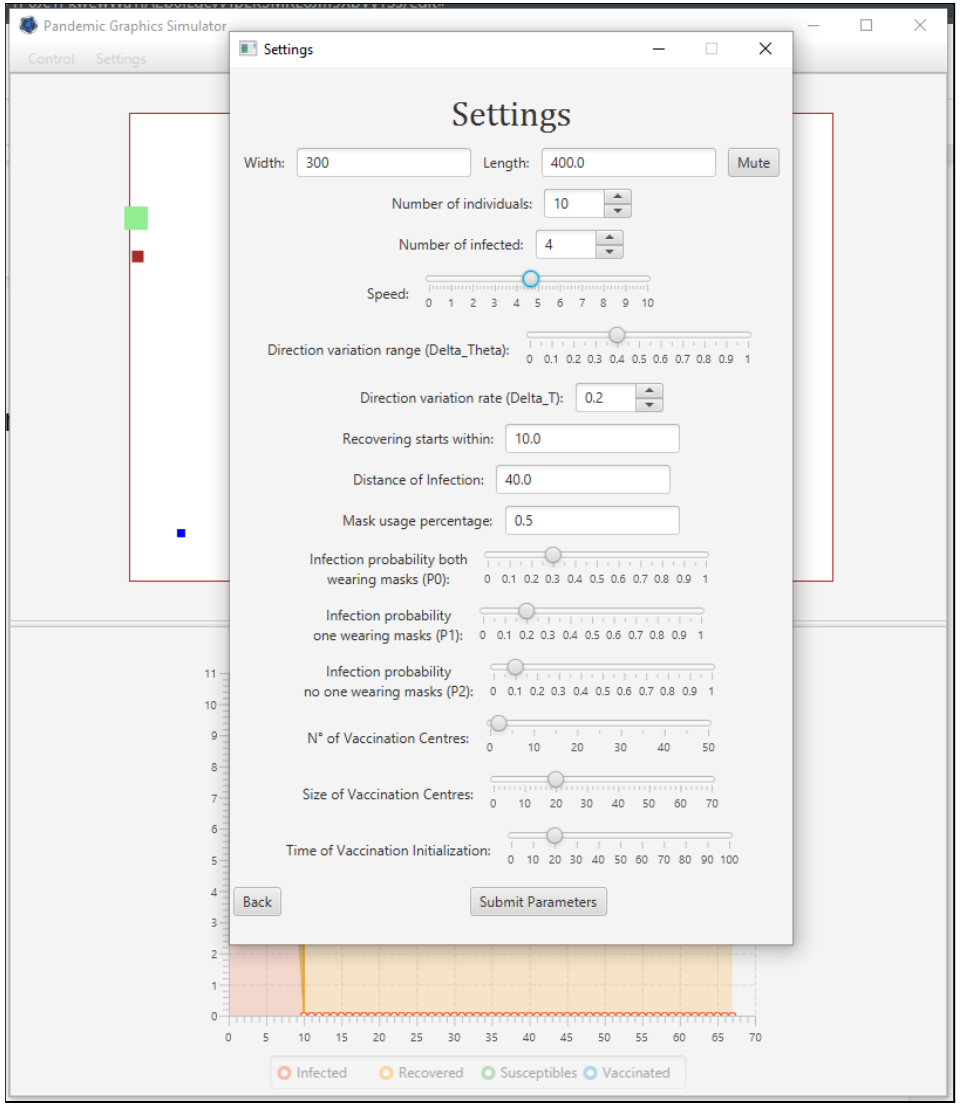
- Todas las clases existen independientes de sí, a excepción de las clases heredadas, esto quiere decir que su uso dentro del código está dentro del Stage4 (main) y no dentro de otra clase, lo que significa que no habrían relaciones de composición entre clases definidas por quien desarrolla el código.

- Los métodos constructores no fueron incluidos dentro del diagrama UML, sin embargo los métodos getters y setters si fueron incluidos con el propósito de evitar un diagrama colapsado de texto y hacerlo lo más legible posible.

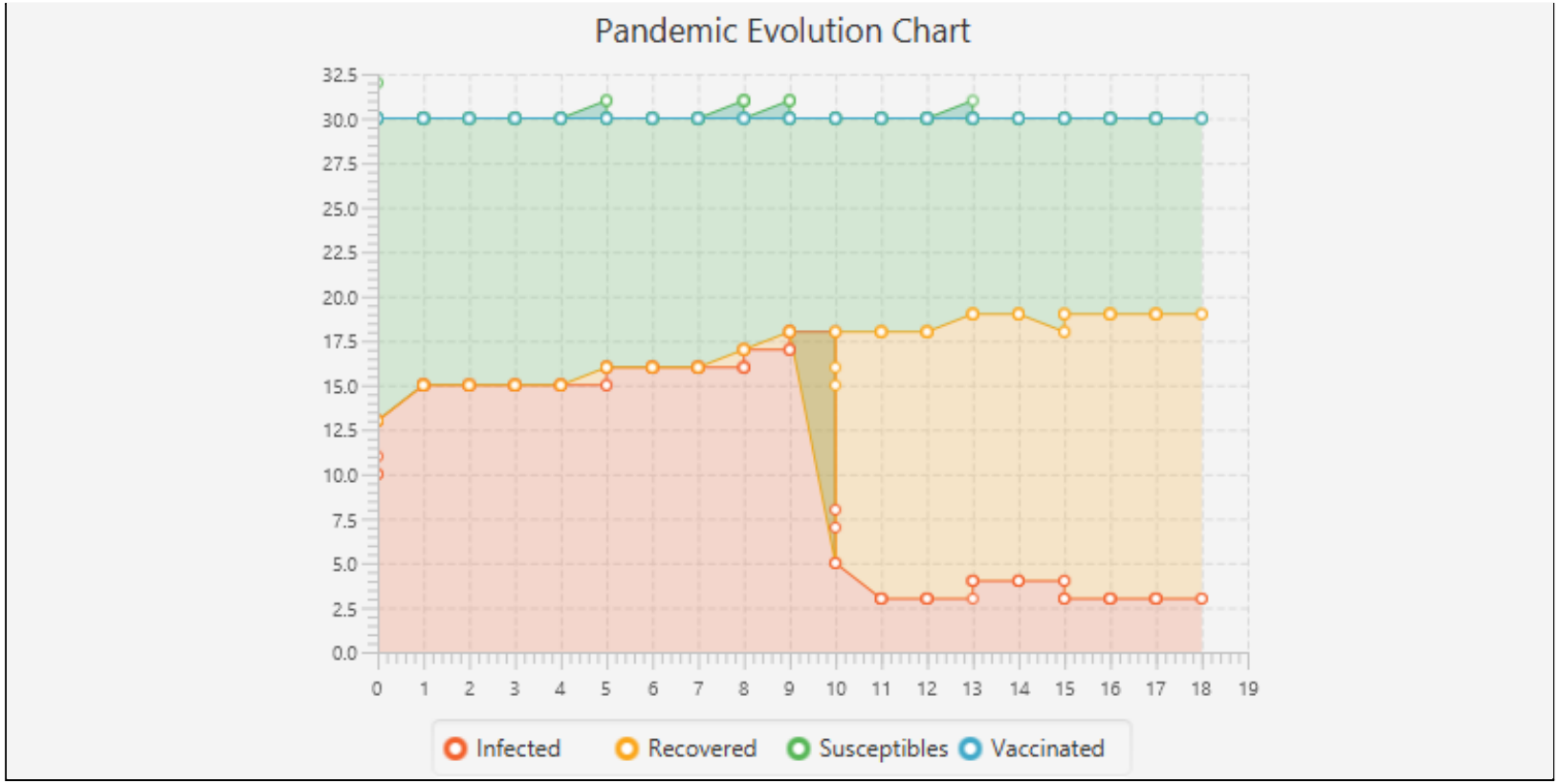
- El modelo fue hecho con Draw.io, con apoyo del diagrama que se proyecta mediante el plugin “UML Design Tool”, pero se le realizaron algunas modificaciones, eliminando tipos de datos innecesarios y sacando de los métodos constructores de cada clase definida.

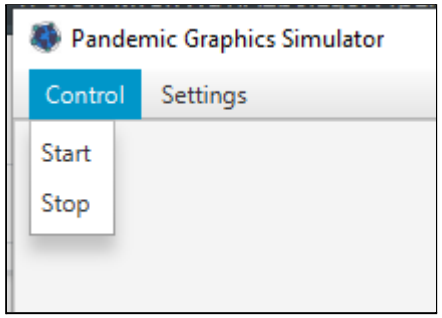
Imagenes del funcionamiento





ELO-329


Pandemic Evolution Chart
32.5
30.0
27.5
25.0
22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
Infected
Recovered
Susceptibles
Vaccinated


Pandemic Graphics Simulator
Control
Settings
Start
Stop