

29/12/20

Solución propuesta:

La arquitectura utilizada para la solución corresponde al modelo-vista-controlador (MVC), en donde el usuario interactúa con el controlador para modificar el modelo, reflejándose estas interacciones en la vista. La arquitectura descrita se puede apreciar en la ilustración 1.

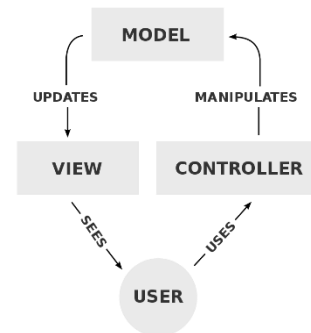


Ilustración 1 Arquitectura MVC

Para la construcción del modelo se consideró que una persona puede estar en cuatro estados:

- Sano: No padece la enfermedad, por lo que es propenso a enfermarse. Es representado con el color **azul**.
- Enfermo: Padece la enfermedad y es propenso a morir con probabilidad 'Death_rate'. Luego de 'Days_to_heal' días la persona se cura pudiendo convertirse en una persona curada con probabilidad 'healPbb' o en una persona sana con probabilidad 1- 'healPbb'. Es representado con el color **rojo**.
- Muerto: Ha muerto a causa del virus. Es representado con el color **amarillo**.
- Curado: Luego de padecer la enfermedad, ha desarrollado inmunidad, por lo que ya no se puede enfermar. Es representando con el color **verde**.

Para la construcción del universo de estudio, se consideraron 6 poblaciones aisladas en las que por simplicidad se asumió que en cada una había una persona infectada al comienzo de la simulación. Cada población contiene 'iny(Initial_population/6' personas. Cada persona es generada aleatoriamente en cada comunidad, estas tienen un movimiento aleatorio en este espacio, a menos que se presenten las siguientes situaciones:

- Fiesta: Las personas que tengan entre 15 y 40 años, se dirigen a un club simbólico generado aleatoriamente. Su probabilidad de suceder es de 'eventPbb = 0.005', y cuando sucede se cambia el color de la comunidad a gris claro.
- Cuarentena: Cuando las personas enfermas de una población alcanzan el 'quarantienCriterion = 0.05' % de la comunidad total, se entra en cuarentena. En ella, cada persona tiene asignada una familia del 0 al 8, por lo que se dirigen a la casa que le corresponde.

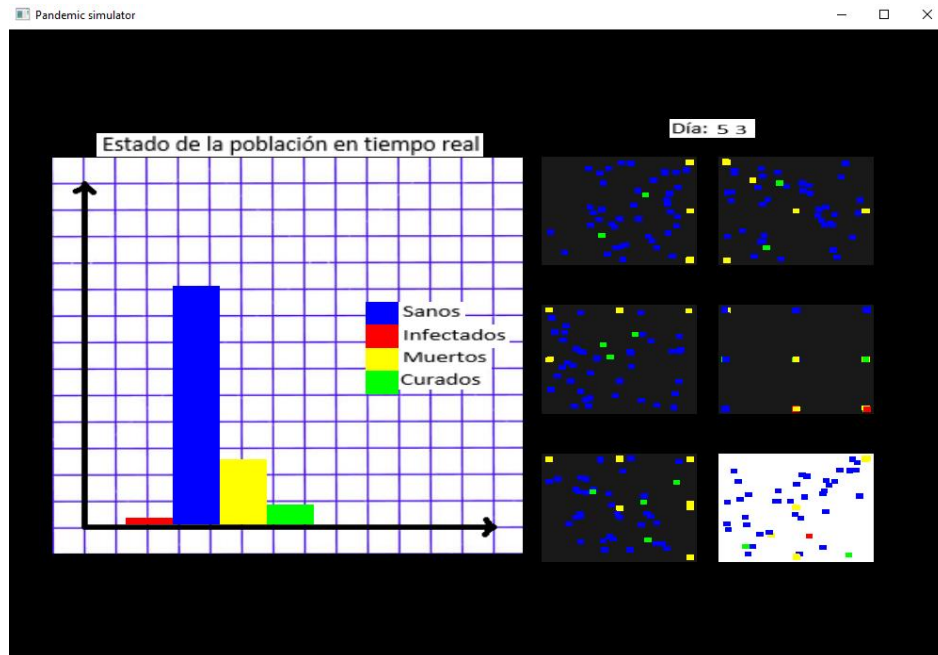
El tiempo transcurre de forma continua, por lo que se muestran los resultados en tiempo real en la parte izquierda de la vista, esta gráfica está construida con primitivas de 'OpenGL'. Al presionar la tecla 'p', se termina la simulación y se despliega el resultado histórico en un gráfico aparte, construido con la biblioteca 'matplotlib'.

Por otra parte, las letras, números y background del gráfico, fueron construidos con texturas, mientras que en todos los demás elementos se utilizaron cuadrados escalados y trasladados.

Importante: Se recomienda utilizar el archivo 'virus.json' provisto en la entrega de la tarea, ya que el programa no se encuentra del todo optimizado, por lo que al aumentar la población inicial pierde fotogramas por segundo. En cuanto a las probabilidades, puede cambiar los valores.

Resultados:

El resultado final se puede apreciar en la siguiente figura:



En la figura se puede apreciar que la hay una comunidad en fiesta (background gris claro), y que otra comunidad está en cuarentena (personas aisladas en sus casas). Finalmente, el grafico con la información histórica desplegado se puede apreciar en la siguiente figura:

