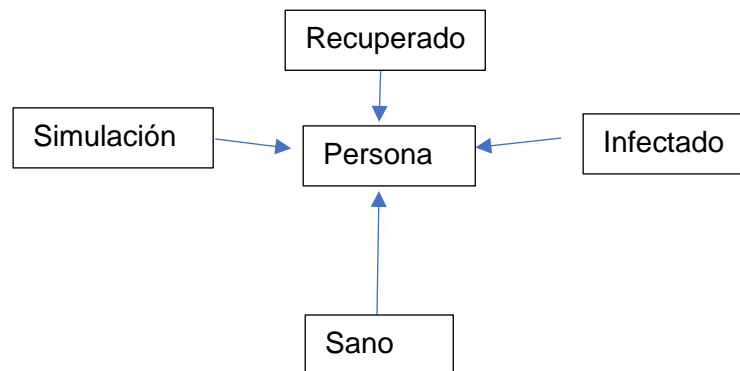


Contexto:

La OMS le ha pedido que simule la propagación de un virus nuevo llamado flojera aguda, bajo las siguientes condiciones, existen tres tipos de personas: sanos, infectados, recuperados. Cada persona sana, al estar en contacto con un infectado tiene el 90% de probabilidad de infectarse. Luego que una persona se infecte pasan 14 segundos (días) hasta que se recupera. El programa no contempla muertes de las personas, ni reinfección de estas.

Entidades:**Requerimientos Funcionales**

Nombre:	R.#1 Manejar diferentes tipos de personas
Resumen:	El programa recibe los datos los datos de la persona mediante un archivo y los carga en la interfaz dependiendo si es una persona sana, infecta o recuperada.
Entradas:	Cantidad de personas sanas, cantidad de personas infectadas y cantidad de personas recuperadas.
Resultados:	Personas cargadas correctamente en el software.

Nombre:	R. #2 Llevar un manejo del estado de salud de las personas
----------------	--

Resumen:	El programa deberá estar en la capacidad de manejar los estados de salud de las personas que se encuentren en el programa contemplando que cualquier persona que no esté infectada tendrá el 90% de posibilidades de infectarse y la persona infectada se recuperará después de 14 días.
Entradas:	Estado actual de la salud de la persona, puede ser sano, infectado o recuperado.
Resultados:	El estado de cada persona se actualiza dependiendo si ésta se infecta o sigue sana y en caso de que se infecte, que muestre si ésta se recupere.

Nombre:	R. #3 Mostrar la interacción de las personas en la interfaz de usuario.
Resumen:	El programa debe estar en la capacidad de mostrar la cantidad de personas en la interfaz y caracterizarlas según su estado de salud.
Entradas:	Ninguna
Resultados:	Las personas son mostradas en la interfaz de usuario.

Nombre:	R. #4 Indicar la cantidad de personas en cada estado de salud.
Resumen:	El programa debe estar en la capacidad de mostrar al usuario la cantidad actual

	de personas según su estado de salud en el momento.
Entradas:	Ninguna
Resultados:	Se muestra en la interfaz de usuario cuántas personas sanas, infectadas y recuperadas hay en ese momento.

Requerimientos no funcionales

Nombre:	RNF. #1 Usabilidad del programa
Resumen:	El programa debe ser intuitivo al uso y las tareas deben estar correctamente explicadas, siendo fácil de usar a simple vista.
Entradas:	Ninguna
Resultados:	El usuario sabe como utilizar el programa.

Nombre:	RNF. #2 Las excepciones deben ser reportadas por la propia aplicación.
Resumen:	El programa debe estar en la capacidad de lanzar una excepción cada vez que una persona se contagie y también

	cuando más del 30% de la población esté infectada. Las dos excepciones anteriormente mencionadas son propias y cada una es independiente de la otra.
Entradas:	Ninguna
Resultados:	Se muestra en la interfaz de usuario la excepción provocada haciéndole saber al usuario la situación ocurrida.

Nombre:	RNF. #3 El programa debe manejar el ordenamiento parcial y el ordenamiento natural con las personas.
Resumen:	El programa debe estar en la capacidad ordenar de forma parcial las personas del programa dependiendo al color del texto de los estados de salud de las mismas. También debe ordenar de forma natural el número de personas en cada estado
Entradas:	Ninguna
Resultados:	Se muestra en la interfaz de usuario el ordenamiento parcial y natural de las personas.

Nombre:	RNF. #La información de las personas debe ser leída por un archivo de texto.
----------------	--

Resumen:	El programa debe estar en la capacidad leer los estados de salud de las personas y la cantidad de las personas en cada estado de salud mediante un archivo de tipo TXT.
Entradas:	Archivo de texto con la información de las personas
Resultados:	El programa lee de manera correcta la información de las personas.