Desarrollo e implementación de un modelo de simulación basado en eventos para determinar recursos hospitalarios

Wilson Hernán Rodas Pérez

Carlos Homero Morocho Medina

Ingeniería en Sistemas

Universidad Politécnica Salesiana

Azuay, Cuenca, Ecuador

***Abstract* -** **The simulation is one of the most used prediction techniques that we have today, its use and versatility to predict or test risk factors is extremely useful in any field of engineering and in other sciences, this document describes the analysis and implementation of a simulation model in order to predict the necessary resources that a hospital should have for the global crisis of COVID19 based on discrete events.**

***Index Terms – regression, simulation, discrete events***

I. INTRODUCCIÓN

Desde los años 40’ el mundo ha introducido una serie de herramientas de análisis matemático que han ayudado a seguir practicando uno de los más antiguos artes de la humanidad moderna, el modelado, una de las formas de poder comprender las realidades que nos rodea mediante un entorno totalmente controlado y gestionado por nosotros, esta práctica hizo que actualmente todo fenómeno, proceso o sistema se pueda modelar en diferentes entornos sean físicos como maquetas, prototipos, etc, o lógicos como software de simulación.

El poder crear modelos en entornos computacionales permite simular diferentes tipos de procesos en diferentes ambientes, que pueden ser ajustados según sea necesario, actualmente poseemos una gama infinita de software de simulación tanto pagado como opensource, ya depende del tipo de simulación que queramos realizar para poder escoger una herramienta indicada.

Existen dos maneras de poder realizar simulaciones en entornos computadillas, la una es utilizando un lenguaje de programación con extensiones que permita crear entornos de simulación, estos pueden ser:

* Java
* Python
* C++
* etc

Este tipo de lenguajes generalmente tiene librerías que me permiten crear entornos de simulación, son mucho más manipulables ya que nosotros creamos el código necesario de implementación.

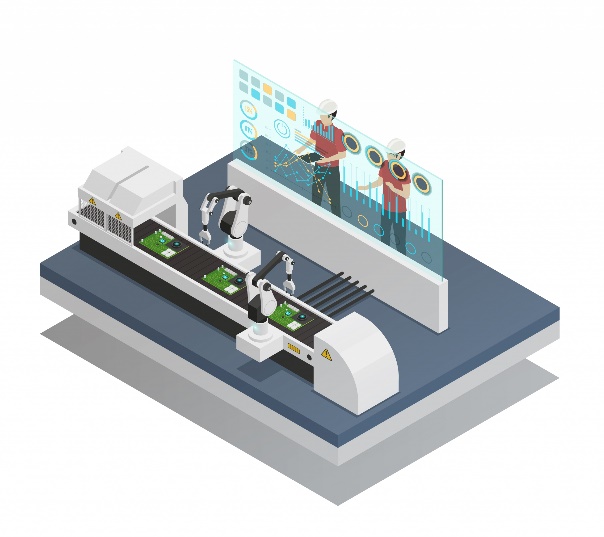
Los siguientes son software específicamente de simulación, es decir no tenemos que codificar nada sino más bien utilizar un lenguaje de simulación especificado por la herramienta y generalmente son entornos más sencillos de manipular, estos pueden ser.

* Arena
* Promodel
* Freecad
* etc

Independientemente de la herramienta que escojamos para simular debemos tener claro los conceptos de simulación, o lo que es necesario para cubrir o entender este análisis, por lo que a continuación agregaremos un par de definiciones que nos harán entrar en contexto.

**¿Qué es simulación?**

Simulación es una técnica numérica para conducir experimentos en una computadora digital. Estos experimentos comprenden ciertos tipos de relaciones matemáticas y lógicas, las cuales son necesarias para describir el comportamiento y la estructura de sistemas complejos del mundo real a través de largos períodos.



**¿Qué es software de simulación?**

El software de simulación se basa en el proceso de modelar un fenómeno real con un conjunto de fórmulas matemáticas. Es, en esencia, un programa que permite al usuario observar una operación a través de la simulación sin realmente realizar esa operación

Los dos conceptos son calve para poder entender este análisis y poder continuar con la implementación, además como sabemos que existe un monto de técnicas o tipos de simulación que no podemos tratarlas aquí, ya que ese no es el objetivo, procederemos con la definición del tipo de simulación que se implementara en este análisis.

**¿Qué es simulación de eventos discretos?**

La simulación por eventos discretos es una técnica informática de modelado dinámico de sistemas. Frente a su homóloga, la simulación de tiempo continuo, esta se caracteriza por un control en la variable del tiempo que permite avanzar a éste a intervalos variables, en función de la planificación de ocurrencia de tales eventos a un tiempo futuro.

II. DESARROLLO

La implementación que veremos a continuación se basa en la problemática que tienen la mayoría de los hospitales actualmente, que es no contar con un aproximado de recursos necesarios (ambulancias, enfermeras, camas, respiradores) para poder atender a los pacientes de COVID19.

En este contexto vamos a tartar de simular mediante eventos discretos la asignación de esos recursos en un proceso normal de hospitalización dentro de un centro médico, es decir los recursos que se ocupan cuando un paciente llama a emergencias, luego cuando es atendido por el hospital y por ultimo los recursos que usa ese paciente mientras se queda internado en el mismo, con el fin de poder definir la capacidad máxima de pacientes que pueden ser atendidos.



# A. Herramienta de simulación

Como primer punto tenemos la elección de la herramienta que utilizaremos para generar la simulación, es este caso hemos escogido una librería de JavaScript llamada SIMJs, esta es una herramienta muy flexible de manejar y me permite generar mis propios entornos de simulación y poder manipular cualquier detalle extra mientras se ejecuta, ya que debemos codificar nuestra simulación.

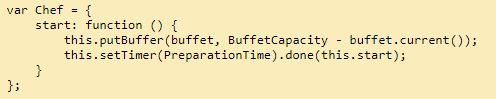
SIM.JS es una biblioteca para modelar sistemas de eventos de tiempo discretos:

* La biblioteca proporciona construcciones para crear Entidades que son los actores activos en el sistema y encapsula el estado y la lógica de los componentes de un sistema.
* Las entidades compiten por recursos, que pueden ser Instalaciones (servicios que solicitan las entidades; admite las disciplinas de servicio FIFO, LIFO con preferencia y Procesador compartido), Buffers (recursos que pueden almacenar una cantidad finita de tokens) y Tiendas (recursos que pueden almacenar JavaScript objetos).
* Las entidades se comunican esperando eventos o enviando mensajes.
* El registro y el análisis de estadísticas lo proporcionan las estadísticas de series de datos (recopilación de observaciones discretas independientes del tiempo), las estadísticas de series temporales (recopilación de observaciones discretas dependientes del tiempo) y las estadísticas de población (el comportamiento del crecimiento y la disminución de la población).
* SIM.JS también proporciona una biblioteca de generación de números aleatorios para generar variantes aleatorias sembradas a partir de varias distribuciones, incluidas uniforme, exponencial, normal, gamma, pareto y otras.

SIM.JS está escrito en JavaScript \* idiomático \*, si necesita mas información sobre la herramienta no olvide revisar la siguiente página.

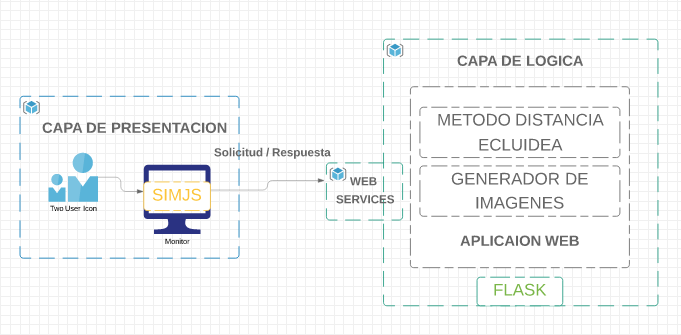
<https://www.codeproject.com/Articles/569143/SIM-JS-Discrete-Event-Simulation-in-JavaScript>

Ejemplo de codigo



# B. Arquitectura de la implementación

Para la implementación emos decido crear una pequeña aplicación web Flask, que además de proveer los archivos .html y .js, nos permitirá generar las imágenes estadísticas obtenidas de la simulación, usando la librería matplotlib de Python.

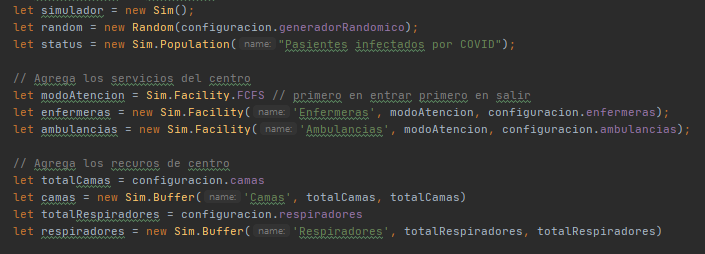


Como podemos observar en la arquitectura, el Simjs se ejecutará obviamente en el cliente por ende debemos generar el script de la simulación con la interfaz necesaria para poder ejecutar las pruebas.

El servidor de Flask nos permitirá servir los archivos estáticos y poder generar las gráficas de simulación, mediante web services.

# C. Implementación de la simulación

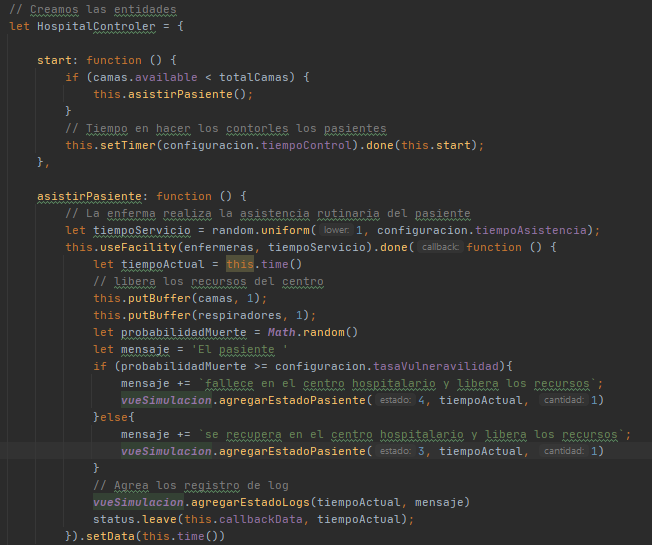
En este punto solo nos centraremos en la implementación del código para realizar la simulación que emos planteado anteriormente, entonces como primer punto debemos inicializar los eventos y recursos que utilizaremos.



Después procedemos a crear las respectivas entidades que interactuaran en la simulación.

* ENTIDAD HOSPITAL

En esta entidad definiremos los eventos respectivos a la asistencia de las enfermeras respecto a los pacientes internados, es decir aquí liberaremos recursos tanto de camas como de respiradores de los internados que se recuperaron o fallecieron.



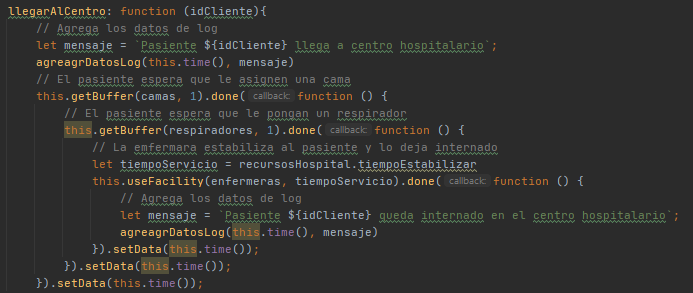
* ENTIDAD PACIENTE

En esta entidad definiremos los métodos donde los clientes hacen uso de las ambulancias y por ende la asignación de los recursos de camas y respiradores

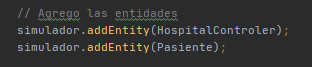
El siguiente Codigo es sobre la generación de pacientes respecto a un determinado tiempo de llegada, aquí se define si el paciente hace uso de una ambulancia o simplemente llega por su cuenta al centro médico.



En el siguiente Codigo realizaremos el proceso donde el paciente llega al centro medico y se le asigna los recursos de cama y respiradores y obviamente usando los servicios de las enfermeras



Por último, agregamos las entidades a nuestro simulador

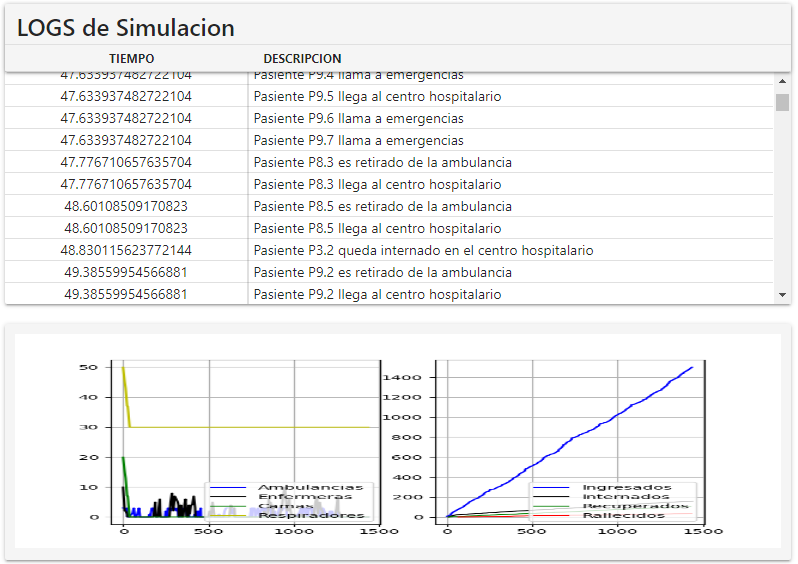


# D. Interfaz de simulación

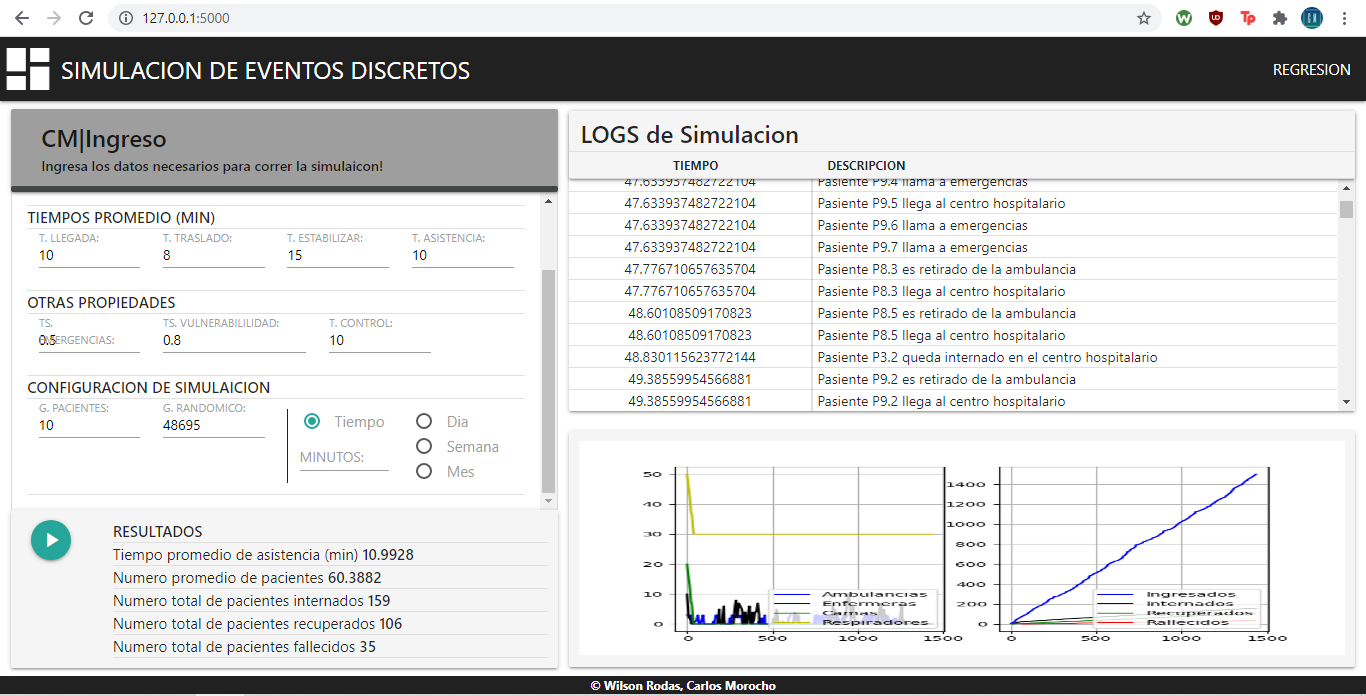
En esta fase no daremos mayor detalle ya que no es el objetivo de esta práctica, sino más bien dar un vistazo de la interfaz final que emos implementado en este análisis de simulación.



Esta sección es donde se representan los datos de la simulación.



Una vista completa de la interfaz.

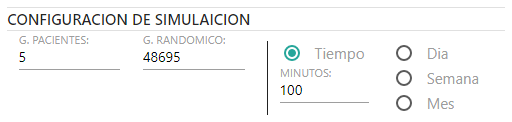


# D. Pruebas de la simulación

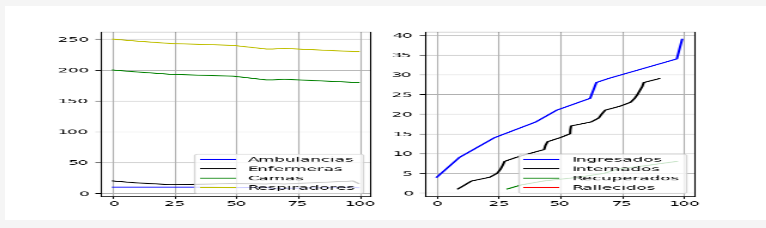
Una le las configuraciones que emos realizado como ejemplo es la siguiente, con esta deducimos que es la optima para que un hospital trabaje de forma eficiente y los recursos no se comprometan de golpe, lo que permitirá la atención de todos los pacientes infectados por COVID19.

Las siguientes imágenes son sobre la configuración.





En las siguientes imágenes se muestran los gráficos estadísticos de los resultados obtenidos, donde podemos ver la tendencia que tienen los recursos respecto a la simulación.



# E. Resultados y análisis.

Con las pruebas que realizamos anteriormente, definimos que a mayor numero de pacientes, requerimos un mayor número de camas y respiradores, pero lo interesante es que analizamos que a mayor número de enfermeras atendiendo el hospital la mayoría es atendida a tiempo, ya que, si bien no existe los recursos necesarios en el hospital para internarse, al menos se logra estabilizar al paciente en la zona de emergencia y esperar hasta tener recursos.

III. RECOMENDACIONES

Como recomendaciones debemos decir que el huso de una herramienta de simulación implementado por un lenguaje de programación como es JavaScript suele ser muy tedioso de comprender, e incluso a veces suele ser mejor utilizar un software ya creado de simulación, en este caso nosotros queríamos experimentar con la herramienta para definir nuestro propio proceso, ya depende del tiempo que tengamos y tipo de simulación que necesitemos realizar.

Sin más, recomendamos visitar nuestro blog donde explicamos algunas cosas puntuales de lo emos revisado.

<https://morochorodas.blogspot.com/2020/12/tutorial-de-uso-de-simjs.html?fbclid=IwAR0yXVoPJCbKP-tzFQVyy6eqejbBdxn_1kJ3ik3Ft5aI-xsGUDli5-arpuI>

IV. CONCLUSIONES

En conclusión, el uso del software de simulación supone una gran ventaja en cualquier ámbito profesional que se lo aplica, pues su buen diseño y construcción aseguran una información válida para la toma de decisiones empresariales, además no hay que olvidar que las librerías de simulación que proveen los lenguajes de programación estándar, nos pueden ser de mucha utilidad cuando los sistemas de simulación no son suficientes o carecen de personalización respecto al rol de negocio que maneja la empresa.

REFERENCIAS

1. " *Discrete-event simulation", wikipedia.com,*

*2020. [Online]. Available:*

*https://en.wikipedia.org/wiki/Discrete-event\_simulation**[[Accessed: 10 - Dec - 2020].*

1. *"* *Discrete Event Simulation in JavaScript”,* *codeproject.com 2013. [Online]. Available*

*https://www.codeproject.com/Articles/569143/SIM-JS-Discrete-Event-Simulation-in-JavaScript*

*[Accessed: 08 - Dec - 2020].*