

QUIZ: SISTEMA DE GESTIÓN DE COLAS Y UNA APLICACIÓN PARA TRÁFICO EN CARRETERAS CON PEAJE

QUIZ: QUEUE MANAGEMENT SYSTEM AND AN APPLICATION FOR TRAFFIC OPTIMIZATION ON ROADS WITH TOLL

Gipsy Miguel Ángel Arrunátegui Angulo¹
Sebastián Miranda Manrique²

RECEPCIÓN: 11 DE SETIEMBRE DEL 2019

ACEPTACIÓN: 08 DE OCTUBRE DEL 2019

RESUMEN

El software Quiz [1] fue realizado para automatizar el proceso de múltiples colas, escenarios y permitir un flujo óptimo de los clientes. Desarrollado en el año 2002, este sistema ha sido revisado como posible optimizador de tráfico en carreteras con peajes, asignando tiques virtuales a los autos y guiándolos hacia los carriles menos congestionados. Los autos, al pasar el peaje, reciben un tique virtual en sus celulares, el cual se gestionará a lo largo de su trayectoria, usando el modelo emisor-aviso-servicio-promoción del sistema Quiz. La clasificación y priorización de los carriles, las realizaría el servidor de este programa, mediante el uso de algoritmos ad-hoc, marcado con feromonas o inteligencia artificial, según se desee.

Palabras clave: cola, cuestionario, carriles, tráfico, software, optimización, feromona

ABSTRACT

The Quiz software [1] was developed to manage the process of multiple queues, scenarios and an optimal flow of customers. Developed in 2002, this system has been revised towards a possible traffic optimizer on toll roads, assigning virtual tickets to cars and guiding them to the less congested lanes. The cars, when passing the toll, receive a virtual ticket on their cell phones, which will be driven along their trajectory, using the issuer-publishing-service-promotion model of the Quiz system. The classification and prioritization of the road lanes, would be done by the Quiz server of this program, through the use of ad-hoc algorithms, marked with pheromones or artificial intelligence, as desired.

Keywords: queue, quiz, lanes, traffic, software, optimization, pheromone

1. INTRODUCCIÓN

Las líneas de espera, también llamadas colas, constituyen un fenómeno omnipresente en casi cualquier sector de la actividad humana. Resolver la atención de un cliente en un tiempo adecuado es materia de preocupación de aquellos que son responsables de brindar un servicio.

Estos fenómenos, aunque de repente no son percibidos directamente, aparecen continua y cotidianamente en la vida. Ya sea cuando toca esperar por el tono de marcar en el teléfono, formar la cola del autobús, esperar la cola de los bancos, acceder al e-mail e internet, matricularse, asistir a un evento, subir a un avión, aguardar en el consultorio del médico, buscar la aprobación de un crédito u otras actividades, que sería largo enumerar, todas pasan por tener que esperar en algún momento para recibir el servicio esperado.

Estos fenómenos son aplicables no solamente a seres individuales, sino a casi cualquier ente susceptible de recibir un servicio o ser materia de una transformación o proceso. Así tenemos, también, colas para la materia prima en una fábrica antes de ser utilizada como insumo, para un paquete de datos que espera su turno en la red antes de viajar y llegar a su destino, para un avión en el uso de la pista, para proyectos de ley que esperan su debate y aprobación en el congreso y hasta los bebés, en un parto múltiple, esperan su turno para nacer.

Los fenómenos de líneas de espera, al estar presentes y en contacto con nuestras vidas, ameritan un estudio y análisis cuidadoso. En la medida que sean mejor comprendidos, sus beneficios nos beneficiarán a todos por igual.

El software Quiz permite la gestión de líneas de espera de distinta naturaleza: personas, vehículos, materiales, datos y cualquier otro elemento susceptible de ser puesto en espera y competir por un servicio. Asimismo, abarca desde las funciones operativas, administrativas y de control, hasta el análisis y simulación de escenarios para la toma de decisiones.

Este software se desarrolló como proyecto profesional de la carrera de Ingeniería de Sistemas en los años 2001 al 2002. El presente documento constituye una aplicación de la estructura y conceptos del mismo a un problema de tráfico en carreteras con peajes.

2. FUNDAMENTACIÓN TEÓRICA

2.1. Las líneas de espera

Para comprender mejor los fenómenos de las líneas de espera, es necesario conocer algunos conceptos y definiciones que pasamos a explicar.

En todo sistema de colas, existen al menos tres elementos clave: clientes que arriban al sistema, servidores que atienden a los clientes, y procesos que definen el comportamiento y flujo entre ellos.

Un **cliente** es un ente que ingresa al sistema para recibir un servicio. Puede ser un ser viviente, un material o un ente abstracto. Sea cual sea su naturaleza, al ingresar, compete con otros entes para recibir el servicio que necesita en el menor tiempo posible.

Un **servidor** es un ente que tiene la capacidad de brindar el servicio que el cliente necesita. En el sistema puede haber uno o más servidores y los mismos pueden tener características similares o diferentes.

El **proceso de llegada o arribo** define la forma como aparecen o se acercan los clientes al sistema para solicitar el servicio. Analíticamente, se trata de encontrar una distribución estadística que refleje este comportamiento de arribo de los clientes. Asimismo, se pueden definir parámetros asociados al

proceso, como la cantidad de clientes que arriban en un determinado instante de tiempo (normalmente, uno), o la actitud del cliente para sumarse a una cola, cambiarse de cola o abandonar el sistema.

El **proceso de salida o servicio** describe el comportamiento del servidor para atender al cliente. Es decir, se trata de estimar el tiempo que se tardará el servidor en atender a un determinado cliente. Al igual que en el proceso de llegada, analíticamente se apela a una distribución estadística para modelar este comportamiento.

Dentro del ordenamiento físico que pueden tener los servidores, se pueden observar dos modelos típicos: **servidores en serie** (como en la línea de ensamblaje de un artículo, en la que el cliente —el artículo— pasa por una serie de procesos) y **servidores en paralelo** (como los cajeros de un banco, en que el cliente pasa por uno de los varios servidores disponibles y luego sale del sistema). También, son posibles las combinaciones de estos, como sucede en el caso de la **serie-paralelo**, por ejemplo, de la gestión de pasaportes o de las revisiones técnicas, casos en los hay que pasar por varias etapas en serie, pero, en cada una de ellas, existe un conjunto de servidores en paralelo.

La **disciplina o arbitraje** de la cola es el método que se usa para elegir al siguiente cliente en cola para ser atendido y, dependiendo de la política de la empresa, existen varios criterios. El más simple y usado es el criterio **PEPS** (FIFO, FCFS en inglés) que significa «el primero en entrar es el primero en salir» (*First-In First-Out* o *First Come First Served*), lo cual quiere decir que el cliente que entró primero a la cola es el que primero será atendido. Otros métodos son el **UEPS** (LIFO, LCFS), «el último en entrar es el primero en salir» (*Last In First Out* o *Last come First Served*); el de priorización, según el cual el siguiente cliente en ser atendido es el que tiene mayor prioridad en la cola, de acuerdo a algún criterio preestablecido; y el aleatorio (**SIRO** - *Service In Random Order*), en el que el siguiente cliente es elegido al azar [2].

El método de selección de cola es el criterio que empleará el cliente para ingresar a una determinada cola —como se suele hacer en el supermercado al elegir la cola más corta o la de cajas rápidas— o para cambiarse de cola (*jockeying* o *switching*), de acuerdo con su conveniencia y si ello está permitido.

2.1.1. El proceso de llegada o arribo

En el proceso de llegada o arribo, se trata de modelar el comportamiento en el tiempo del número de clientes que llegan al sistema. Este número, calculado por unidad de tiempo, no es constante y depende de muchos otros factores como son la hora del día, la temporada, las promociones, etc.

Definimos el parámetro λ como la tasa de arribos, que es el número de arribos por hora. Se define:

$$\mu = \frac{1}{\lambda}$$

como el tiempo promedio entre arribos expresado en minutos, esto es el tiempo entre 2 arribos continuos

En la mayoría de aplicaciones de estudio de las líneas de espera, se usa la distribución estadística exponencial Poisson con parámetro λ , el cual tiene la propiedad de no-memoria, es decir, que el tiempo del siguiente arribo no tiene nada que ver con el del arribo previo. Esta independencia simplifica sustancialmente los cálculos del modelo matemático.

Otra distribución importante es la de Erlang, que se basa en una variable aleatoria T, cuya función de densidad está especificada por dos parámetros: R (ratio) y k (shape). Cuando k = 1, esta distribución es igual a la de Poisson [2].

En la vida real, el comportamiento de los arribos no puede circunscribirse a la distribución estadística arriba mencionada, pues este comportamiento es simplemente un caso de distribución de los muchos posibles, pero que, sin embargo, ha permitido el nivel de abstracción necesario para obtener un modelo matemático funcional.

Cuando se aplican diversas distribuciones de frecuencia para el arribo de los clientes, ya no es posible modelar matemáticamente el sistema sin incurrir en una gran complejidad. Es aquí donde aparece, con toda su fuerza, la técnica de simulación de sistemas.

Realizando un modelo de simulación del sistema, es posible asignar diversas distribuciones de frecuencia para modelar el proceso de arribo de los clientes. Mediante la técnica de generación de variables aleatorias, es posible generar una serie de arribos de cliente cuya distribución de frecuencia corresponda a una distribución específica. Por ejemplo, se pueden generar datos que tengan una distribución normal con media μ y desviación estándar σ predeterminados.

2.1.2. El proceso de servicio

El proceso de servicio consiste en representar el comportamiento del servidor típicamente a través del tiempo que le demora prestar el servicio. Para el tratamiento matemático, se suele tomar el tiempo promedio del servicio. Esto, no obstante, se vuelve más complejo cuando, por razones de simplificación, se asume que los servidores —cuando se cuenta con varios— tienen tiempos de servicio iguales.

El proceso de servicio requiere de una variable aleatoria S , que debe tener una distribución acumulada con parámetro $\mu = \frac{1}{\lambda}$, en donde μ es el número de clientes que se atiende en 1 hora (suponiendo que haya demanda) y su tiempo promedio de servicio es $\frac{1}{\mu} = \lambda$.

Al igual que en el proceso de arribo, el considerar que S sigue una distribución exponencial simplifica notablemente los cálculos, aunque se puede utilizar una distribución de Erlang para evitar la falta de memoria de la distribución exponencial, tal cual se mencionará en el acápite del proceso de arribos.

2.2. El estudio y análisis de las líneas de espera

El estudio de las colas ha sido tratado tradicionalmente mediante el uso de las matemáticas, construyendo modelos de los diferentes tipos de colas de manera muy simplificada y con resultados suficientemente precisos para modelos simples. Sin embargo, el mundo real y las variables que lo componen son mucho más complejos y muchos han considerado conveniente, dado el avance y difusión de los medios computacionales, estudiar y analizar las colas mediante la simulación de estos fenómenos para obtener los resultados de un determinado modelo de líneas de espera.

En el presente proyecto del sistema de manejo de colas, se utilizan ambos tratamientos, el matemático y el de simulación, para obtener lo mejor de cada técnica y así poder brindar al usuario del software la capacidad de utilizar ambos y usufructuar los mejores beneficios de cada uno. De esa manera, se logrará un mejor análisis, predicción y comparación de los resultados obtenidos.

2.2.1. El estudio matemático de las líneas de espera

El estudio matemático de las líneas de espera tiene ya muchos años de investigación y supone el tratamiento de los fenómenos mediante la identificación de las colas como un prototipo o modelo para el cual se desarrollan fórmulas para su tratamiento analítico. El estudio de las líneas de espera desde el punto de vista matemático es extenso y complicado. Aquí, solo brindaremos información lacónica y enumerativa, más que explicativa, de los modelos de colas pertinentes al tipo de sistema que vamos a desarrollar.

Notación para los sistemas de líneas de espera

La notación de Kendall-Lee se ha convertido en un estándar para describir los sistemas de colas. Esta notación es de la forma:

$1/2/3/4/5/6$ en donde:

- 1: Especifica la naturaleza del proceso de arribo, de acuerdo con los siguientes valores:
 - M: Tiempos entre arribos independientes e idénticamente distribuidos con variables aleatorias distribuidas exponencialmente
 - D: Tiempos de arribo determinísticos independientes e idénticamente distribuidos
 - E_k : Tiempos de arribo independientes e idénticamente distribuidos con distribución de Erlang de forma k
 - GI, GD: Distribución General y tiempos de arribo independientes e idénticamente distribuidos

- 2: Especifica la naturaleza del proceso de servicio de acuerdo con los siguientes valores:
 - M: Tiempos entre arribos independientes e idénticamente distribuidos con variables aleatorias distribuidas exponencialmente
 - D: Tiempos de arribo determinísticos independientes e idénticamente distribuidos
 - E_k : Tiempos de arribo independientes e idénticamente distribuidos con distribución de Erlang de forma k
 - G: Distribución general

- 3: Especifica la cantidad de servidores en paralelo.

- 4: Especifica la disciplina de la cola:
 - (PEPS, FIFO) = primero en entrar primero en salir
 - (UEPS, LIFO) = último en entrar primero en salir
 - (SIRO) = servicio en orden aleatorio
 - GD = disciplina general

- 5: Especifica el máximo número de clientes en el sistema.

- 6: Especifica el universo de población que puede acceder potencialmente al sistema.

Por ejemplo: $M / E2 / 8 / FIFO / 10 / \infty$

Es una representación de un modelo de colas que podría representar a un consultorio médico en donde los pacientes arriban exponencialmente (M), existen 8 doctores que atienden con una distribución de Erlang de 2 fases, se administra de acuerdo con el orden de llegada (FIFO), existe una capacidad total de 10 pacientes y cualquier ciudadano puede enfermarse y acudir (∞).

Modelos de línea de espera

Teniendo en cuenta la notación de Kendall-Lee, la combinación de sus parámetros representa a muchos tipos de modelo de líneas de espera típicos, de acuerdo con la configuración específica de colas y

servidores. Para este proyecto de software de líneas de espera, solo se tratan aquellos que sean de interés y aplicables al proyecto.

Los modelos por reseñar son:

M/M/1/GD/∞/∞

M/M/1/GD/c/∞

M/M/s/GD/∞/∞

M/M/s/GD/∞/∞ (en Tándem)

El modelo M / M / 1 / GD / ∞ / ∞

Este es un modelo que cuenta con un solo servidor y tiene tiempos de arribo y servicio exponenciales.

Número de clientes presentes en el sistema = L

$$L = \frac{\lambda}{(\mu - \lambda)}$$

Número de clientes en espera = L_q

$$L_q = \frac{\lambda^2}{\mu(\mu - \lambda)}$$

Fórmula de Little :

$$L = \lambda W$$

$$L_q = \lambda W_q$$

$$L_s = \lambda W_s$$

Donde:

W = promedio de tiempo del cliente en el sistema

λ = promedio de arribos por unidad de tiempo

L = promedio de clientes en el sistema

L_q = promedio de clientes en cola

L_s = promedio de clientes atendidos

W_q = tiempo promedio en cola

W_s = tiempo promedio en servicio

Algunos ejemplos del modelo:

- Una caseta de peaje que atiende vehículos, los cuales arriban a razón de 275 por hora y que atiende un vehículo cada 12 segundos.
- Un servicio de alquiler de videos con 1 empleado que atiende, donde se registra un arribo de 10 clientes por hora y el empleado se demora 5 minutos en cada cliente.
- Una estación de servicio con autos que arriban al surtidor a razón de 7.5 por hora y son surtidos de gasolina en 4 minutos en promedio.

El modelo M / M / 1 / GD / c / ∞

Este modelo se refiere al de arribos y servicios exponenciales, que solo contiene un servidor y que cuenta con una capacidad limitada de clientes en el sistema “c”. Este es idéntico al sistema M / M / 1 / GD / ∞ / ∞, excepto por el hecho de que, cuando llega al número máximo de clientes “c” (los que están atendándose más los que están en cola), inmediatamente cesa toda admisión al sistema. Es decir, la tasa de arribos se vuelve cero, hasta que se restablezcan las condiciones de operación, verbigracia, cuando el número de clientes es menor que “c”.

Se define ρ como la intensidad de tráfico:

$$\rho = \frac{\lambda}{\mu}$$

Cuando $\lambda \neq \mu$: la probabilidad de que el sistema entre en un estado estable está dada por:

$$\pi_0 = \frac{1 - \rho}{1 - \rho^{c+1}}$$

$$L = \frac{\rho [1 - (c + 1)\rho^c + c\rho^{c+1}]}{(1 - \rho^{c+1})(1 - \rho)}$$

$$W = \frac{L}{\lambda(1 - \pi_c)}$$

$$W_q = \frac{L_q}{\lambda(1 - \pi_c)}$$

Nótese que, en este sistema, λ puede ser mayor que μ, ya que el discriminante c evita que el sistema “explote”, como podría ocurrir en el modelo previo.

Algunos ejemplos:

- Un médico tiene un total de 8 sillas de espera en su consultorio. Los pacientes arriban a una tasa de 3 por hora y el médico se tarda un promedio de 25 minutos por paciente. Los pacientes que encuentran el consultorio lleno se retiran y buscan otro doctor.
- Un Lobby de un cajero automático puede albergar hasta 6 clientes, los cuales arriban a razón de 40 por hora y se atienden a razón de 2 minutos cada uno.

El modelo M / M / s / GD / ∞ / ∞

Este modelo posee similares características a los ya vistos. El número total de clientes en el sistema es j y la diferencia radica en que el número de servidores s es mayor que 1. Es decir, el cliente que espera en cola se atenderá en el primer servidor que se encuentre disponible. Para simplificar el análisis del problema, se asume que el tiempo de atención es similar para todos los servidores. Se define ρ como la intensidad de tráfico:

$$\rho = \frac{\lambda}{s\mu}$$

Cuando $\rho < 1$, se obtiene un estado estacionario, cuya probabilidad de encontrarse vacío es π_0 , que viene dado por:

$$\pi_0 = \frac{1}{\sum_{i=0}^{s-1} \frac{(s\rho)^i}{i!} + \frac{(s\rho)^s}{(1-\rho)s!}}$$

La probabilidad de que todos los servidores (en un estado estable) se encuentren ocupados, viene dado por:

$$P(j \geq s) = \frac{(s\rho)^s \pi_0}{s!(1-\rho)}$$

La Tabla 1 - Probabilidad $P(j \geq s)$ muestra una serie de valores tabulados para distintas situaciones. Tenemos entonces que:

$$L_q = \frac{P(j \geq s)\rho}{1-\rho}$$

$$W_q = \frac{L_q}{\lambda} = \frac{P(j \geq s)}{s\mu - \lambda}$$

$$L = L_q + \frac{\lambda}{\mu}$$

$$W = \frac{L}{\lambda} = W_q + \frac{1}{\mu}$$

Estas fórmulas son solamente válidas para distribuciones de llegada y servicio exponenciales, y tiempos de servicio iguales entre los servidores. Nótese que, si $\rho \geq 1$, no existe estado estable y el sistema “explota”. Algunos ejemplos:

- Una agencia bancaria con 2 ventanillas recibe 80 clientes por hora, los cuales forman una sola cola delante de las ventanillas. El tiempo promedio de servicio es de 1.2 minutos y las llegadas son exponenciales. Determinar el número de clientes en la agencia, el tiempo esperado que el cliente pasará en la agencia y el tiempo ocioso de cada ventanilla.
- Un gerente de banco quiere precisar el número óptimo de ventanillas que debe abrir en un día viernes. Se supone que el costo de hacer esperar 1 minuto a cada cliente es de 5 centavos. Los clientes arriban a razón de 2 por minuto. Las ventanillas atienden a razón de 1 cliente cada 2 minutos y al banco le cuesta \$/ 30 por hora cada ventanilla habilitada. Los tiempos de arribo y servicio son exponenciales. Para minimizar costos, ¿cuántas ventanillas se deben abrir?

ρ	S=2	S=3	S=4	S=5	S=6	S=7
0.1	0.02	0	0	0	0	0
0.2	0.07	0.02	0	0	0	0
0.3	0.14	0.07	0.04	0.02	0.01	0
0.4	0.23	0.14	0.09	0.06	0.04	0.03
0.5	0.33	0.24	0.17	0.13	0.1	0.08
0.55	0.39	0.29	0.23	0.18	0.14	0.11
0.6	0.45	0.35	0.29	0.24	0.2	0.17
0.65	0.51	0.62	0.35	0.3	0.26	0.21
0.7	0.57	0.51	0.43	0.38	0.34	0.6
0.75	0.64	0.57	0.51	0.46	0.42	0.39
0.8	0.71	0.65	0.6	0.55	0.52	0.49
0.85	0.78	0.73	0.69	0.65	0.62	0.6
0.9	0.85	0.83	0.79	0.76	0.74	0.72
0.95	0.92	0.91	0.89	0.88	0.87	0.85

Tabla 1. Probabilidad $P(j \geq s)$. Elaboración propia.

El modelo en etapas (Tándem), serie de $M / M / s / GD / \infty / \infty$

Este modelo es conocido en español como ‘en tándem’ o ‘en etapas’. Su tratamiento es similar a los anteriores, considerando cada etapa (*stage*) como uno de los modelos anteriores, en donde la entrada a una etapa proviene de la salida de la etapa precedente.

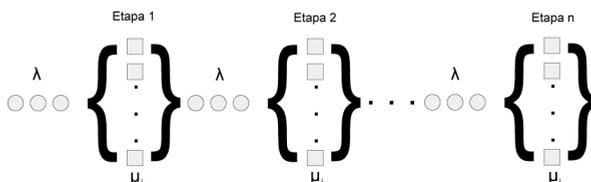


Figura 1. Modelo de colas k-etapas. Elaboración propia.

Cada etapa recibe a los clientes atendidos en la etapa anterior y estos son atendidos por sus i -servidores, que son liberados hacia la siguiente etapa. El modelo asume que las colas intermedias tienen suficiente capacidad para albergar los clientes provenientes de la etapa anterior.

Teorema de Jackson:

«Si los tiempos inter-arribos son exponenciales con parámetro λ , si los tiempos de servicio μ_k para cada etapa son exponenciales y si cada etapa contiene infinita capacidad para aceptar los arribos provenientes de la etapa precedente, los tiempos entre arribos para cada etapa del sistema serán exponenciales con el parámetro λ ».

Algunos ejemplos:

- El sistema peruano de emisión de certificados de antecedentes penales tiene varias etapas secuenciales: recepción de formas, pago de derechos, toma de fotografías, impresión dactilar y entrega del certificado.
- En una línea de ensamblaje de autos, la última etapa del ensamblaje de vehículos es poner el motor y las llantas. Arriban 54 autos por hora. En la primera etapa, se le pone el motor; en la

segunda, se le ponen las llantas. Existe 1 trabajador que pone motores a razón de 6 por hora, y un trabajador que pone llantas a razón de 1 por minuto.

2.2.2. Problemática de las líneas de espera

Dado el carácter universal de los fenómenos de líneas de espera, las instituciones y empresas en general, cualquiera que preste un servicio se preocupará por hacerlo en los mejores términos para sus clientes. Esto supone que se trate de atender al cliente lo mejor y más rápido posible.

Dentro del esquema de competitividad y globalización que enfrentan las empresas hoy, la libertad del cliente para elegir “su proveedor” provoca que exista preocupación por las demoras que sus clientes experimentan en líneas de espera y se trate de minimizar estos tiempos.

Obviamente, la solución más trivial consiste en aumentar el número de servidores ad-infinitum (ventanillas de un banco por ejemplo), pero esto supone una erogación de recursos de los que no se dispone o es muy caro disponer.

Dentro de las preguntas típicas que se formulan los responsables de brindar un servicio figuran las siguientes:

- ¿Cuánto se demora un cliente en mi servicio?
- ¿Cuánto tiempo espera en la cola?
- ¿Cuánto se demora mi servidor en atender a mi cliente?
- ¿Cuántos clientes abandonan mi sistema?
- ¿Cuántos servidores debo tener para atender a mis clientes en un tiempo determinado?
- ¿Debo tener colas separadas o una sola?
- ¿Cuál debe ser la capacidad de mi local?
- ¿Cuánto debo pagar a mis servidores?
- ¿En cuánto mejora mi tiempo de atención si agrego un servidor?
- ¿Agrego más servidores o mejoro los existentes?
- ¿Cómo habría sido la atención del pasado día si hubiera contado con dos servidores más?
- ¿En mi nueva oficina, cuántos servidores debo poner?
- ¿Cuánto me cuesta cada cliente que abandona mis líneas de espera?
- ¿Cuál es el tiempo de atención promedio en todas mis oficinas?
- ¿Existe tiempo ocioso en mis servidores?
- ¿Después de haber capacitado a mis servidores, ha mejorado su tiempo de respuesta?
- ¿Si debo reducir costos, de qué servidores puedo prescindir de modo que se afecta mínimamente el servicio?

Estas y muchas más interrogantes se pueden plantear con la finalidad de optimizar el servicio a los clientes.

Los analistas matemáticos que han desarrollado la «teoría de colas» (*queueing theory*) han tratado de resolver y responder muchas de las preguntas de los sistemas de líneas de espera como las planteadas anteriormente. Estas formulaciones matemáticas tienen en común el apelar a una serie de condiciones asumidas y simplificaciones que les permiten arribar a fórmulas concretas, pero solamente a resultados aproximados.

Estos resultados aproximados son, en muchos casos, suficientes y suficientemente aproximados como para ser tomados en cuenta en un estudio a priori del sistema. Conforme el sistema a analizar se complica, su análisis matemático se complica, hasta llegar a modelos inmanejables y resultados muy cuestionables, todo esto aparejado con la exageración que hay que asumir.

Es en este punto que se hace necesario el uso de otra técnica para analizar el sistema. Una de estas es la simulación de sistemas.

La simulación de sistemas usando computadoras permite encontrar soluciones a problemas matemáticos o de sistemas para los cuales no se conoce una solución analítica satisfactoria. Además, facilita la solución de problemas de sistemas para los cuales la experimentación sería muy costosa y permite investigar posibles estrategias [15].

La simulación de sistemas se viene utilizando desde hace muchos años con éxito para analizar el comportamiento de muchos y diversos sistemas; desde un simple juego de michi (Tic-Tac-Toe) hasta la simulación de un viaje a Marte o la evolución del universo desde el Big-Bang, pasando en el intermedio por juegos, negocios, procesos de manufactura, tráfico, plantas nucleares, bio-sistemas y cerebros artificiales.

Precisamente, en la mayoría de los sistemas mencionados, existen inmersos fenómenos de líneas de espera que se simulan internamente. Por ejemplo, en el proceso de ensamblaje de una computadora existen líneas de montaje que admiten cierta capacidad de partes y piezas, que poseen una determinada velocidad de producción. En una central telefónica, existe una determinada cantidad de líneas y troncales que debe atender a una cantidad mucho mayor de teléfonos de los usuarios.

La simulación de sistemas aplicada a los fenómenos de líneas de espera permite el análisis de sistemas más complejos de los que permite el análisis matemático y, en general, permite mayor flexibilidad en el manejo de las variables y condiciones del problema. Sin embargo, hasta hace poco, la construcción de los modelos computacionales con las herramientas tradicionales de la ingeniería de software y el hardware disponible implicaba un gran esfuerzo, costo y tiempo antes de ser utilizable y llegar a un grado de madurez y confiabilidad [3].

El avance de las tecnologías de información y la ingeniería de software —en los últimos tiempos— permite que se puedan realizar simulaciones cada vez más complejas. En la actualidad, es posible realizar simulaciones en un computador personal usando herramientas, componentes y librerías alrededor del paradigma de la programación orientada a objetos.

Los avances de la ingeniería de software proporcionan una definición y un manejo abstraccional más preciso de los elementos del problema, además de la reutilización de bloques fundamentales de construcción de los modelos simulados. Asimismo, se conjugan con un manejo más apto del esquema organizativo y relacional de los mismos, lo que mejora la capacidad de simular sistemas más complejos.

2.2.3. La simulación basada en Agentes

Para el estudio de sistemas complejos en los que no se puede tener una aproximación determinística ni la linealidad está necesariamente presente, la simulación basada en agentes constituye una forma muy interesante de estudiar y observar las propiedades emergentes que surgen a partir del comportamiento de agentes distintos en un medio en el cual se interrelacionan, puesto que utilizan reglas de comportamiento relativamente sencillas, pero que originan agrupaciones o comportamientos no esperados y que, en muchos casos, son inteligentes y sorprendentes. En la aplicación de este sistema de colas a la problemática del tráfico, se utilizará el modelado y la simulación basada en agentes, siendo estos los automóviles, camiones, buses y otros vehículos que transitan por una carretera con peajes, donde, al momento de ingresar a la vía, se le asigne una determinada ruta. Esta será determinada por un algoritmo o un sistema de inteligencia artificial basado en el análisis del tráfico actual, es decir, de cuáles son las condiciones que se están dando y cuál sería la mejor ruta que se puede tomar. Para el modelado en agentes, se utilizará el software conocido como NetLogo. A través de él, más adelante, desarrollaremos un modelo para resolver este problema del tráfico en vías con peaje [4].

3 SOLUCIÓN PROPUESTA

Enfrentados a los problemas de las entidades que experimentan el fenómeno de las líneas de espera, la empresa, los trabajadores y los ejecutivos involucrados en la prestación de servicios bajo un fenómeno de encolamiento, se requiere de herramientas computacionales que les permitan gestionar las colas de atención en sus negocios.

Por este motivo, se plantea una solución de *software* para la gestión de líneas de espera. Este se dividirá en los siguientes subsistemas:

- Subsistema de operaciones
- Subsistema de administración y control
- Subsistema de análisis
- Subsistema de simulación

3.1. Concepto de la solución

La solución que plantea el producto de *software* de gestión de líneas de espera, al que, en adelante, denominaremos QUIZ (*Q*ueue *I*nformation *Z*apper), se basa en los siguientes principios:

- a) En toda línea de espera (cola) hay clientes y servidores.
- b) Los clientes deben esperar a que llegue su turno de atención en alguno de los servidores.
- c) Terminada la atención, el servidor llamará a otro cliente de la cola.
- d) Este esquema puede repetirse para derivar al cliente —una vez atendido— hacia una siguiente etapa en la que el esquema anterior (b y c) se repite.
- e) Los principios anteriores deberían generalizarse para varios negocios, escenarios y situaciones.

Se definió una estructura inicial que proporcionará soporte a las operaciones, control, análisis, simulación y toma de decisiones:



Figura 2. Estructura de QUIZ. Elaboración propia.

Aplicando estos principios, se determinó que la parte operativa del sistema Quiz debería contar con los siguientes componentes:

- Emisores de tickets (Emisor)
- Estaciones de aviso (EDA)
- Estaciones de atención (Estación)
- Controlador (Qserver)
- Sistema del cliente (opcional)

Estos componentes se organizarán e interactuarán para brindar los mecanismos que permitan atender organizada y eficientemente la línea de espera. A continuación, definimos propiamente los componentes:

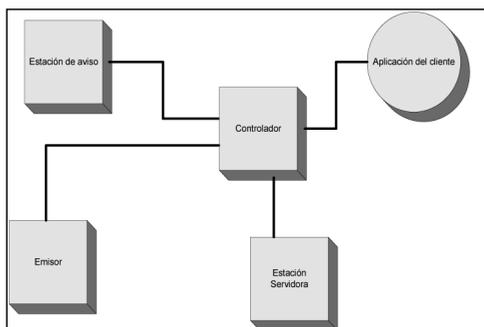


Figura 3. Componentes QUIZ. Elaboración propia.

Emisor: es el componente que permite emitir un tique de atención, con el cual se identificará al cliente y se le guiará hasta la estación que lo atenderá y hacia la siguiente etapa si esta existiere. QUIZ tiene la capacidad de generar tiques virtuales —que no se imprimen— pero que funcionan internamente. La emisión del tique se activará por cualquier medio que sea capaz de iniciar un evento en el emisor: un botón, *touch screen*, lector de banda magnética, lector de huella, sensor de código de barras y cámaras, entre otros.

Estación de aviso (EDA): es el componente que se encargará de publicar el número de tique y la estación de servicio hacia la cual debe dirigirse el cliente para ser atendido. La estación de avisaje puede consistir en cualquier dispositivo que sea capaz de interfazarse vía protocolo de red o de puerto serial o paralelo, recibir mensajes y mostrarlos. En el caso de tickets virtuales, la estación de aviso puede disparar sensores, compuertas y selectores electrónicos que liberen al cliente hacia la estación de servicio que lo atenderá.

Estación: componente que será manejado por los trabajadores que atienden al cliente. Cuando está lista para atender a su próximo cliente, solicita a QUIZ que seleccione al siguiente cliente de la cola, que es notificado a través de la estación de aviso. Cuando el cliente llega, se le atiende y la estación indicará a QUIZ el inicio y el fin de la atención. QUIZ enrutará el ticket hacia la siguiente línea de espera si esta existiera. En el caso de tickets virtuales, la Estación indicará a QUIZ su disponibilidad, y este seleccionará y avisará al cliente virtual para que sea encaminado hacia la estación de servicio.

Controlador (QServer): componente que arbitra todo el proceso, registra, controla y administra a los otros componentes de QUIZ. Administra los mensajes, datos, secuencias y configuraciones. Además, constituye el núcleo del sistema QUIZ. Es el componente sobre el que recae toda la orquestación del proceso de colas y la administración de los datos.

Para interconectar los componentes QUIZ, se configurará una red con protocolo TCP/IP, en la que circularán los mensajes QUIZ. La mensajería QUIZ está definida en un protocolo ad-hoc (ver figura 4), al cual responden los distintos componentes. Todos estos podrán usarse en cualquier número y configuración. También, se interconectan con el controlador (Qserver) vía TCP/IP utilizando puertos y direcciones IP configurables.

Al conjunto Emisor-EDA-Estación que operará una línea de espera lo denominaremos «zona». Cada zona estará constituida mínimamente por una estación de aviso y una estación de servicio, y para todo el sistema, debe existir por lo menos un emisor y un controlador Qserver. Por ejemplo:

Una oficina bancaria pequeña puede tener:

- 1 Emisor
- 1 Estación de Aviso
- 3 Estaciones de servicio

y una oficina mayor:

- 2 Emisores
- 3 Estaciones de aviso
- 12 Estaciones de servicio

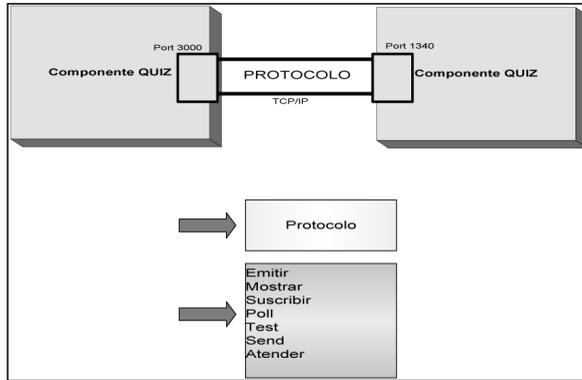


Figura 4. El protocolo QUIZ. Elaboración propia.

Por otro lado, QUIZ puede configurarse para multizona, es decir, manejar una serie de atenciones a un mismo cliente en una secuencia de pasos. Un solo controlador QUIZ puede administrar múltiples zonas, cada una con sus correspondientes componentes. Por ejemplo, en el servicio de emisión de pasaportes, QUIZ podría configurarse de la siguiente forma:

# Zona	Nombre	Emisor	EDA	Estación	QServer
1	Entrega de formularios	2	1	3	1
2	Caja		1	2	
3	Fotos		1	2	
4	Huellas		2	5	
5	Entrega		1	1	

Tabla 2. Ejemplo de configuración multizona. Elaboración propia.

Nótese que la emisión del tique solamente se hace al inicio del trámite. El cliente, luego de ser atendido en la primera zona, es enrutado por QUIZ hacia la siguiente, en donde esperará su turno, será llamado y atendido. Luego, pasará a la siguiente zona y así, sucesivamente, hasta finalizar.

El arbitraje de la cola, es decir, las reglas de negocio que gobiernan la selección del siguiente cliente a ser atendido, pueden configurarse en QUIZ directamente o puede ser delegado a una aplicación del cliente —un sistema experto, por ejemplo— a la que QUIZ le pasa la lista de clientes que están en espera. Por su parte, el sistema del cliente decide e informa a QUIZ cuál es el siguiente que se debe atender.

Los componentes QUIZ pueden configurarse en cualquiera de los esquemas de colas mencionados en el acápite correspondiente y en muchos otros, aún más complicados.

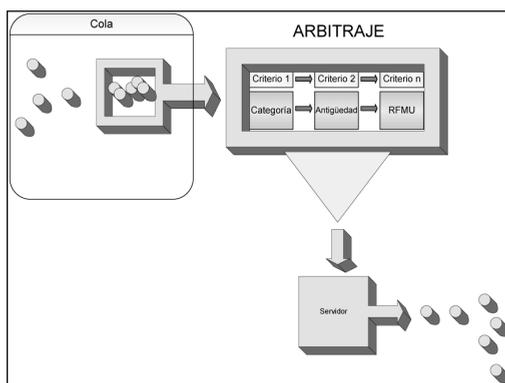


Figura 5. Quiz arbitraje de la línea de espera. Elaboración propia.

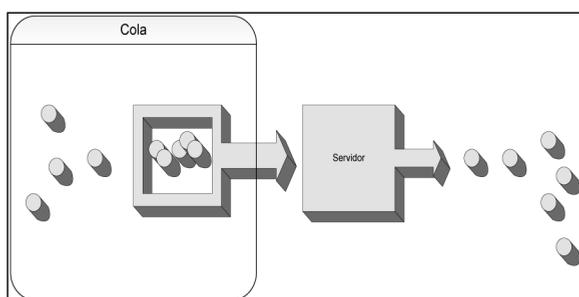


Figura 6. Zona QUIZ con una sola estación de servicio. Elaboración propia.

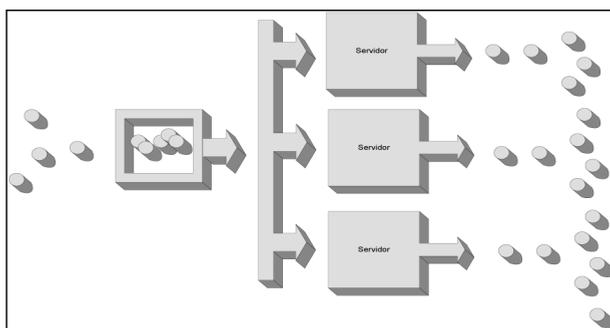


Figura 7. Zona QUIZ con múltiples estaciones de servicio. Elaboración propia.

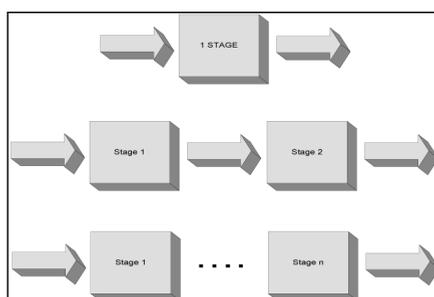


Figura 8. Modelo Multi Etapa (múltiples zonas QUIZ). Elaboración propia.

3.2 Metodología

La metodología de desarrollo para el ciclo de vida de este producto fue una combinación de *Rational Unified Process* (RUP) y programación extrema (XP), como se explica más adelante. A esta combinación se le ha denominado RUXP, como un acrónimo que reúne los acrónimos de las metodologías citadas [5][6].

3.3. Mecanismos

3.3.1. Plataforma para simulación

Se decidió utilizar el producto BaseSim de SolutionsOne como *framework* y librería de componentes de simulación. La decisión se tomó por dos factores: integración con la plataforma de desarrollo Delphi y costo cero.

BaseSim

El producto BaseSim es un conjunto de componentes orientados a la simulación discreta usando orientación a objetos, que corre en la plataforma de desarrollo Delphi, bajo la forma de componentes visuales, los cuales permiten construir modelos con eficiencia y rapidez. Cuenta con una interfase gráfica y animaciones, componentes para recopilación de datos, y gráficos de las estadísticas del modelo.

Netlogo

NetLogo es un entorno de simulación multiagente utilizado por miles de estudiantes e investigadores en todo el mundo. Permite simular el comportamiento de gran cantidad de agentes individuales, cada uno con comportamientos, y relaciones con el entorno y con otros agentes, para analizar las complejas propiedades que pueden emerger.

Para el caso de estudio, se ha utilizado Netlogo para simular como agentes los vehículos y las garitas de peaje.

3.3.2 Plataforma de interconexión

El producto QUIZ hace uso extensivo de la programación con SOCKETS TCP/IP como medio de interconexión y mensajería entre los componentes QUIZ.

Los *sockets* TCP confiable aseguran la recepción y orden de los mensajes enviados y recibidos [7].

Los *sockets* TCP son del tipo conectado y pueden ser clientes o servidores. El *socket* servidor trabaja esperando que los clientes se conecten e inicien la conexión para empezar a intercambiar mensajes, hasta que alguno de ellos termine la conexión. Asimismo, para que dos sockets se conecten, es necesario que sepan dónde están ubicados. Esto se logra mediante la especificación de una dirección IP (nn.nn.nn.nn) y un puerto determinado (por ejemplo, para conversar en protocolo http, se usa el puerto 80).

3.3.3. Plataforma de desarrollo

Como plataforma de desarrollo, se utilizó el entorno integrado Delphi 6.0 de la compañía Borland. Esta elección se debe a sus características, que proveen un lenguaje orientado a objetos (*Object Pascal*) y ofrecen un entorno integrado más avanzado y consistente.

Este documento es un ejemplo del formato con los márgenes y la colocación del texto que está disponible en formato de WORD. Contiene las líneas y los párrafos con los márgenes y área de impresión.

Se recomienda utilizar el presente documento como plantilla de MS Word[®] para la preparación del reporte. Se puede descargar, abrir la plantilla, y copiar su texto y las ilustraciones en la plantilla. Las características generales del texto deben de respetar los siguientes criterios:

Los escritos deben ser impresos en hojas tamaño carta, (21.5 cm x 27.9 cm).

Los márgenes externos deben de respetar los siguientes criterios:

- Margen izquierdo: 2.5 cm.
- Margen derecho: 2.0 cm.
- Margen superior (página 1): 2.5 cm.
- Margen superior (páginas 2-8): 2cm.
- Margen inferior: 2.0 cm.

Además, las figuras deben nombrarse con una abreviatura, por ejemplo, «Fig. 2», a menos que sea al inicio de la oración.

4. DISEÑO

4.1. Arquitectura de alto nivel

El sistema QUIZ está estructurado en niveles de gestión que van desde los subsistemas operacionales hasta los sistemas de soporte de decisiones. En esta versión 1.1, se han implementado los subsistemas de operaciones, administración y control, y el análisis de teoría de colas y simulación. El subsistema de soporte a las decisiones será implementado mediante un *Datamart* que podrá desarrollarse en un futuro.

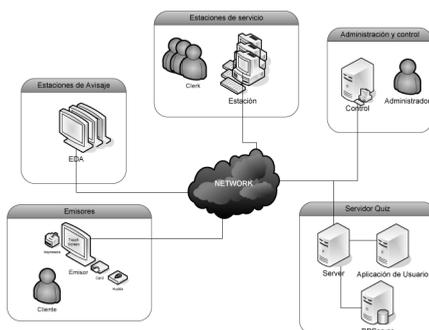


Figura 9. QUIZ, Arquitectura conceptual. Elaboración propia.

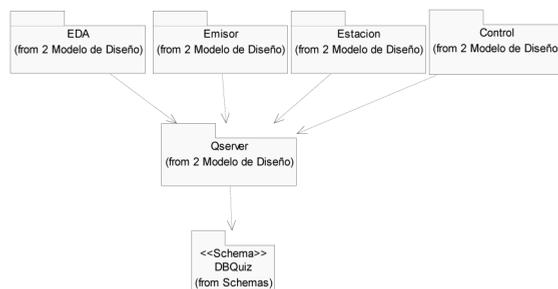


Figura 10. QUIZ, paquetes del subsistema de operaciones. Elaboración propia.

4.2. Vista de UC arquitectónicos

La vista de caso de uso se ha subdividido en cuatro subsistemas: operaciones, administración y control, y el análisis de teoría de colas y simulación.

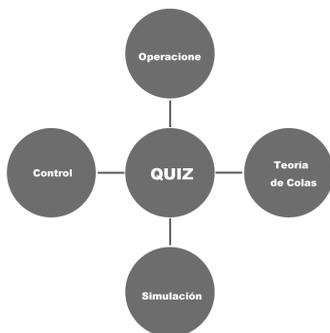


Figura 11. QUIZ, sub-sistemas. Elaboración propia.

Los casos de usos arquitectónicos seleccionados fueron:

Llamar ticket

Cerrar ticket

Probar Emisor

Trabajar con teoría de colas: escenario modelo $M/M/gd/1/\infty/\infty$

4.3. Vistas arquitectónicas

4.3.1. Vista lógica

Modelo de análisis

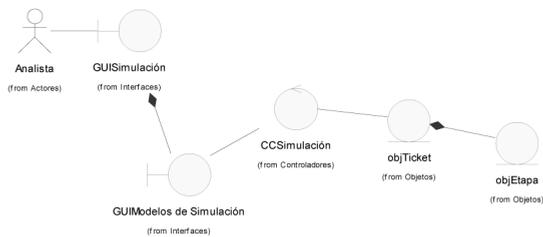


Figura 12. Subsistema de simulación. Elaboración propia.

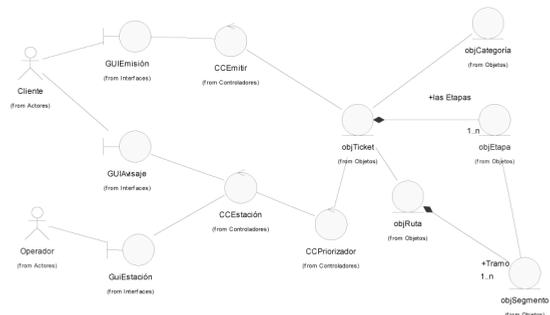


Figura 13. Subsistema de operaciones. Elaboración propia.

5. MODELO DE TRÁFICO CON NETLOGO

Se construyó un modelo de simulación basado en agentes, en donde se representó un tramo de una carretera con casetas de peaje que actúan como los emisores de tiques en el sistema QUIZ, emitiendo un tique virtual a cada vehículo que pasa por la caseta de peaje [4].

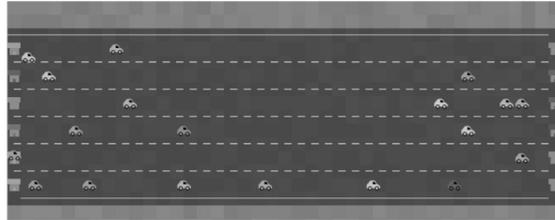


Figura 17. Modelo de autopista con seis carriles. Elaboración propia.

Al pasar el vehículo por la caseta de peaje, se le asigna un carril de preferencia, por el cual, opcionalmente, puede discurrir en menor tiempo. La asignación del carril preferente puede hacerse bajo distintos algoritmos en búsqueda de optimizar el tránsito de cada vehículo.

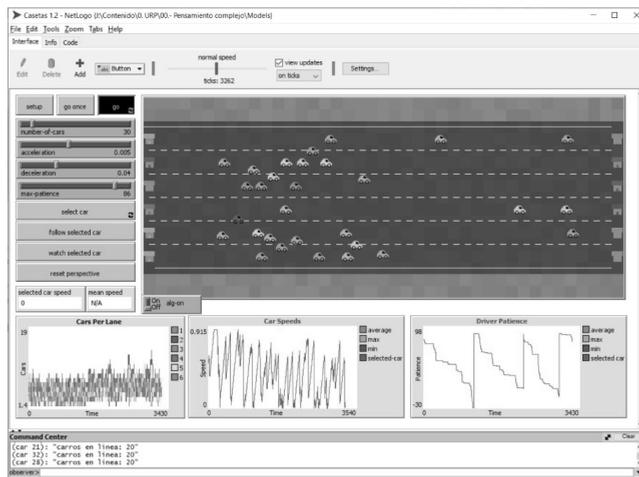


Figura 18. Ejecución del modelo. Elaboración propia.

El modelo desarrollado es una ampliación del modelo de dos carriles de la biblioteca de modelos del Netlogo [4], el cual se ha ampliado a seis carriles con casetas de peaje a los extremos.

6. CONCLUSIONES

- La metodología XP nos permitió adquirir la experiencia de implementar una alternativa novedosa y que produjo resultados. Esto se pudo concretar gracias al profundo involucramiento del cliente.
- El concepto de diseño por contrato (programar las pruebas antes), apoyado en la unidad de pruebas Dunit, ayudó a detectar errores de manera crucial, cuando inadvertidamente se había deteriorado algún componente al aplicar *refactoring* o agregar funcionalidad.

- La arquitectura del sistema, estrictamente hablando, no se obtuvo como parte del proceso con XP, sino, más bien, en una revisión final para obtener la versión 1.1 del producto. Sin embargo, la aproximación inicial a la que se llegó con XP fue bastante satisfactoria.
- La implementación y uso de un protocolo de comunicaciones ad-hoc simplificó en gran medida la interacción entre los componentes. Su facilidad de expansión, a través de la definición de nuevos verbos, permite ampliar la funcionalidad a discreción.
- El subsistema de simulación mediante el uso del *framework* SolutionsBase nos permitió crear unos premodelos que simplifican el trabajo del analista, ya que no necesita conocer programación, solamente configurar los premodelos y ejecutar las simulaciones.
- El producto QUIZ tiene posibilidades de utilización en muchos escenarios más allá de lo previsto inicialmente. Su utilización en sistemas de tiempo real, por ejemplo, en una línea de producción de alta velocidad, es factible, pero debe ser estudiada y cuidadosamente analizada. A pesar de que el diseño arquitectónico garantiza excelentes tiempos de respuesta, hay partes sensibles (la aplicación del cliente y el acceso a la base de datos) que necesitarían evaluarse en tal sentido.
- El uso de un protocolo de comunicaciones y de un servidor central que opera sobre el mismo es un concepto que podría aplicarse a labores de integración de sistemas, encolando sus mensajes.
- Muchos de los conceptos e ideas que se plasmaron en este producto se han visto confirmados años después, en los tiempos actuales (2009): el uso de arquitecturas orientadas a los servicios, la mensajería estándar, los protocolos estándar, el bus de servicios, las colas de mensajes, los servicios Web, las sesiones *stateless*, entre otros.
- Extender el concepto de QUIZ a los tickets virtuales para carreteras con peajes, mostró que es posible trabajar estos esquemas para investigar los problemas del tráfico: congestiones, elusiones, rutas u otros.
- El modelado basado en agentes demostró que es una herramienta viable y con muchas posibilidades para el estudio de los sistemas complejos, por ejemplo, el tráfico.

RECOMENDACIONES

- Evaluar distintos algoritmos de selección del carril preferente, como asignar el carril adyacente más vacío, asignar el carril más vacío si está $n\%$ más vacío que el actual, limitar la cantidad de autos haciendo *jockeying*, activar el *jockeying* a partir de un $\%$ de ocupación, etc.
- Expandir el modelado con Agentes inteligentes para Realizar y validar modelos de predicción, ruteo o evasión de congestiones de tráfico utilizando mecanismos de marcado por feromonas virtuales esparcidas por los demás vehículos.[8]
- Validar los modelos realizados con un *dataset* de tráfico (conjunto de datos) consistente. [9]

REFERENCIAS

- [1] Arrunátegui, Miguel. *Quiz: Sistema de gestión de líneas de espera*. Tesis de titulación. Universidad Peruana de Ciencias Aplicadas. 2009. url: <http://hdl.handle.net/10757/273732>
- [2] W. Winston, *Operations Research Applications and Algorithms*. Belmont, CA: Duxbury Press, 1993.
- [3] J. Banks, «Introduction to Simulation». *Proceedings of the 2000 Winter Simulation Conference*, Eds. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 1999.
- [4] U. Wilensky y W. Rand, *An introduction to Agent Based Modeling*. MIT Press, 2015.

- [5] R. Jeffries et al., *Extreme Programming Installed*. Boston: Addison-Wesley, 2001.
- [6] P. Krutchen, «The 4+1 View Model of Architecture», en *IEEE Software*, vol. 12, no. 6, 1995, pp. 42-50.
- [7] Indy Internet Direct, «Descripción de los productos y librerías Indy» [En línea]. Disponible en <http://www.indyproject.org> [Accedido: Marzo de 2008]
- [8] Y. Ando, Y. Fukazawa, O. Masutani, H. Iwasaki, S. Honiden, «Performance of Pheromone Model for predicting Traffic Congestion», en *AAMAS '06: Proceedings of the fifth international joint conference on Autonomous agents and multiagent systems*, 2006, pp. 73–80.
- [9] R. Horiguchi, Y. Toshio, A. Hirokazu, M. Kuwahara, K. Masahiko, Ozaki, H. Oguchi y Takashi. *A benchmark data set for validity evaluation of road network simulation models*, 2019
- [10] G. Eppen, F. J. Gould y C. P. Schmidt, *Investigación de Operaciones en la Ciencia Administrativa*, 3ª. Edición, Prentice-Hall, 1992.
- [11] M. Chon Ho, J. Mun-Yee Lim, K. Lun Soon, C. Yong Chong. «An improved pheromone-based vehicle rerouting system to reduce traffic congestion», en *Applied Soft Computing* 84:105702, August 2019
- [12] A. Law, y D. Kelton, *Simulation Modeling & Analysis*. New Jersey: McGraw-Hill, 2000.
- [13] H. Taha, *Investigación de Operaciones - Una Introducción*. 6a. Edición. Prentice-Hall, 1998.
- [14] N. Van Dijk, «On Hybrid Combination Of Queueing And Simulation», en *Proceedings of the 2000 Winter Simulation Conference* eds. J. A. Joines, R. R. Barton, K. Kang, and P. A. Fishwick, 2000.
- [15] M. Anu, «Introduction To Modeling and Simulation» en *Proceedings of the 1997 Winter Simulation Conference*, Eds. S. Andradóttir, K. J. Healy, D. H. Withers, and B. L. Nelson, 1997.