UNIVERSIDAD MIGUEL HERNÁNDEZ

GRADO EN ESTADÍSTICA EMPRESARIAL

TRABAJO FIN DE GRADO

SISTEMAS DE APOYO A LA DECISIÓN PARA EL DISEÑO DE UN SERVICIO DE URGENCIAS HOSPITALARIAS

CURSO ACADEMICO: 2021-2022

Departamento de Estadística, Matemáticas e Informática

Facultad de Ciencias Sociales y Jurídicas de Elche

AUTOR: Raúl Artigues Femenia

TUTORES: Joaquín Sánchez Soriano, Juan Carlos Gonçalves Dosantos

Resumen.

La crisis de la COVID-19 tuvo un impacto negativo directo en el ámbito de gestión sanitaria, aumentando significativamente los tiempos de espera y la saturación en los servicios de urgencias hospitalarias tanto públicas como privadas.

Fundamentando en este problema, se ha procedido a realizar un algoritmo matemático en Rsoftware capaz de analizar por simulación un servicio de urgencias en un hospital. Para ello se desarrolla una herramienta para la toma de decisiones que permite analizar los tiempos de servicio de los pacientes que acuden a urgencias dependiendo de los parámetros que definan dicho servicio. Con esta herramienta se puede optimizar el diseño y dimensionamiento de las unidades que componen el servicio de urgencias en un hospital.

La interpretación de los resultados obtenidos del simulador se basa en los modelos de colas, concretamente, en la *teoría de colas,* es por eso, que es de vital importancia tener claros los conceptos que se desarrollan en el presente proyecto. Tanto las medidas de eficacia de la teoría de colas como las trazabilidades de los distintos pacientes del sistema se observan en una interfaz básica que ofrece la posibilidad de realizar consultas y sus respectivas conclusiones.

Palabras clave: Teoría de colas, simulación, Rsoftware, hospital, urgencias.

Abstract.

The COVID-19 crisis had a direct negative impact on the field of health management, significantly increasing waiting times in the emergency services of public and private hospitals as their saturation.

Based on this problem, this project deals with the realization of a RSoftware mathematical algorithm capable of analyzing an emergency department in a hospital by simulation. For this purpose, a decision-making tool is developed to analyze the service times of patients attending the emergency department depending on the service-defining parameters. This tool can be used to optimize the design and sizing of the units that make up a hospital's emergency department.

The interpretation of the results obtained from the simulator is based on queuing models, specifically, on *queuing theory*. Both the efficiency measures of the queuing theory and the traceability of the different patients in the system are displayed in a basic interface that offers the possibility of making queries and their respective conclusions.

Keywords: Queuing theory, simulation, Rsoftware, hospital, emergencies.

Índice General.

Índice de Figuras.

Índice de Tablas.

CAPÍTULO 1

Introducción.

Además de los sistemas informáticos y tecnológicos, las colas son una gran parte de la vida de todos los individuos, para comprar el pan, para realizar un trámite administrativo, para ser atendido en un sistema de *urgencias hospitalarias*, etc. En todas estas se tiene que esperar un determinado tiempo, siendo un paso normalizado en la sociedad, pero que en la gran mayoría veces resulta incómodo. Según informan varios medios digitales españoles[1], los días de espera en servicios hospitalarios aumenta de forma considerable al paso de los años, incluso meses.

A veces las demoras son buenas. Hacen visualizar la importancia del servicio o producto al que se pretende aspirar/adquirir. Pero en general, los clientes no les gusta esperar, lo cual es lógico. Pues bien, mientras se aguarda a ser atendido pueden surgir varias preguntas como: ¿Por qué hay que esperar? ¿Cuánto tiempo tengo que estar en lista de espera?

Estas preguntas tienen una respuesta bastante sencilla, en algún momento del horizonte temporal, la demanda es mayor a la oferta, es decir, hay más personas que llegan al sistema y quieren ser atendidos que personal disponible. Por tanto, se demuestra una vez más, la gran importancia que tiene la planificación, organización y optimización del diseño y dimensionamiento de las unidades que componen el servicio de cualquier sistema.

Por esta razón, los modelos de colas son importantes analizarlos y evaluarlos, para así determinar la manera más eficiente de operar. Con esto se pretende identificar el nivel óptimo de capacidad del sistema para minimizar el coste global del mismo, evaluar el posible impacto de las diferentes alternativas de modificación que tendría en el coste total del sistema, entre otras muchas.

[1] Puedes consultar más información sobre la saturación hoy en día en el siguiente enlace

Este documento da una solución a través de la programación de un *simulador* para analizar un servicio de *urgencias en un hospital*, mediante redes colas aplicando *teoría de colas.* Dicho sistema estará compuesto por n servidores en cada uno de los nodos que componen el servicio. El algoritmo facilitará la identificación de cuellos de botella, es decir, la saturación en cada uno de los nodos, la trazabilidad de los pacientes y aquellas localizaciones del servicio hospitalario con poca o nula *saturación*.

1.1 Estructura del proyecto

La estructura de este proyecto es la siguiente:

Capítulo 1: Introducción, se da una introducción a la simulación de redes de colas y a la explicación de la nomenclatura y diferentes modelos de la *teoría de Colas*.

Capítulo 2: Descripción del problema, se describo el problema en cuestión, cuáles son los objetivos y él porque de esta simulación para un sistema de *urgencias hospitalarias*.

Capítulo 3: Simulación, se centra en la explicación de la simulación, de la generación de números aleatorios y se expone un ejemplo de los métodos de Monte Carlo.

Capítulo 4: Diseño e implementación, se presenta como fue construido el *simulador* programado y las bases que se siguieron para llevarlo a cabo.

Capítulo 5: Análisis de simulación, se muestran los resultados de los diferentes *escenarios* simulados mediante el algoritmo matemático programado.

Capítulo 6: Discusión y conclusiones, se presentan las conclusiones de este proyecto y algunas sugerencias a mejorar en un futuro.

Capítulo 7: Bibliografía, se exponen los diversos documentos consultados para poder llevar a cabo este documento.

Capítulo 8: Anexo, se encuentra el código realizado en *Rstudio* que está dotado del *simulador* y la interfaz básica.

1.2 Descripción de un sistema de colas

La *teoría de colas* es una disciplina, dentro de las matemáticas, más específicamente en los procesos estocásticos, que permite estudiar de forma científica la demora que deben esperar los "clientes" cuando están demandando un servicio. De esta manera, en la mayoría de los casos, los usuarios realizarán la cola si el servicio no es inmediato. Las fórmulas para cada modelo indican como se desempeña cada sistema de colas, incluyendo la media de la cantidad de espera que sucederá y otras variedades de mediciones.

Debido a esto, los modelos de colas son de gran ayuda para determinar cómo manejar los sistemas de una forma más eficaz.

1.2.1 Características de los sistemas de colas

En la vida real es muy común la formación de líneas de espera o, también denominadas, colas. Suele ocurrir cuando la demanda de un servicio es superior a la oferta o a la capacidad que existe para dar dicho servicio.

Estas suceden en muchos ámbitos, desde el mundo tecnológico como las telecomunicaciones hasta la atención de clientes en un establecimiento comercial en la ciudad.

Situaciones como las mencionadas anteriormente, entre otras muchas, tienen ciertas características similares que dan lugar al modelo de sistema de colas.

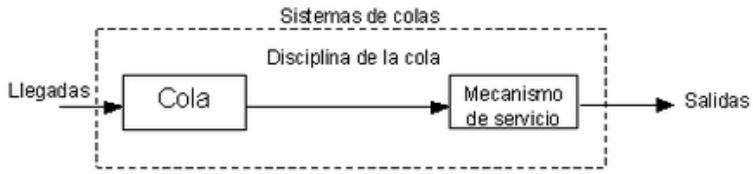


Figura 1.1 Sistema de una cola. Fuente: Wikiwand

Como se puede observar en la Figura 1.1, existen distintos parámetros o elementos que componen un sistema de colas.

En primer lugar, se encuentra las *llegadas o fuente de entrada*. Se trata de un conjunto de individuos que pueden llegar a solicitar un servicio en cuestión. Se puede considerar infinita o finita, pero en el ámbito de la vida real, se trata de llegadas finitas con tamaños muy grandes.

A continuación, se define el *Cliente*. Es aquel individuo de la población potencial que solicita el servicio. Se considera que los tiempos de las llegadas de los usuarios consecutivos son $0 < t_1 < t_2 < \cdots$, será importante conocer el patrón de probabilidad según el cual la fuente de entrada genera consumidores. Lo más común es tomar como referencia los tiempos entre llegadas de dos usuarios consecutivos: $\tau_k = t_k - t_{k-1}$, fijando una distribución de probabilidad. Cuando una fuente de ingreso es finita, la distribución de los τ_k cambia según el número de clientes en el proceso de ser servidos.

Seguidamente se encuentra la *Capacidad de la cola*. Esta indica el número máximo de clientes que pueden estar en cola, antes de estar siendo servidos. Nuevamente, existen dos posibilidades, considerarla infinita o finita. Para hacer los cálculos muchos más simples, se considera ilimitada, es decir, que puede haber innumerables usuarios en cola esperando a ser servidos.

En cuanto a la *Disciplina en cola,* se puede decir que es el modo en el que los usuarios son seleccionados para ser servidos. Las disciplinas más comunes son:

- FIFO: First-In First-Out (Primero en entrar, primero en salir). En esta disciplina se atiende por orden de llegada. En este tipo de cola, los clientes que llegan deben ponerse al final de esta y esperar a su turno. Además, todos los usuarios tienen el mismo tratamiento, es decir, no existen prioridades.

- LIFO: Last-In First-Out (Último en entrar, primero en salir). Se trata de una disciplina en la que el último cliente en entrar será el primero en salir. Este tipo de cola se asemeja a una pila de objetos.
- RSS (colas Random). Se da cuando se seleccionan los clientes de forma totalmente aleatoria.
- Priority. Este tipo de disciplina se da cuando existe dentro de la cola alguna prioridad a la hora de servir a los usuarios.

El *mecanismo de servicio.* Es el procedimiento por el cual se otorga auxilio a los usuarios que lo necesitan/solicitan. Se puede distinguir entre mecanismos de servicio multicanal o monocanal.

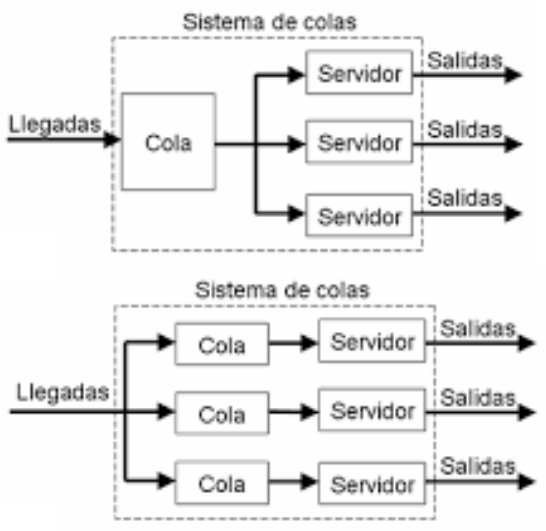


Figura 1.2. Sistema de cola multicanal. Fuente: Redalyc

Para poder determinar el mecanismo de servicio se debe conocer el número de servidores de este mecanismo y la distribución de probabilidad del tiempo de servicio de cada servidor.

En la Figura 1.1 se expone como es un sistema monocanal o sistema de colas simple. Mientras que en la Figura 1.2 se puede observar dos tipos de servicios multicanal. El primero de ellos es una única cola para los n servidores que dispone el sistema. El segundo modelo trata de tener tantas colas como servidores haya. Este último es el más fácil de implementar, pero no es el óptimo.

Cabe destacar que, dentro del mecanismo de servicio, existen las *etapas de servicio.* Estos ciclos pueden ser unietapa o multietapa. El sistema unietapa es aquel en el que los clientes entrar y salen del sistema únicamente pasando por un único servidor. Por lo contrario, el ciclo multietapa, indica que, para poder salir del sistema de colas, es obligatorio pasar por más de un servidor, un claro ejemplo sería el servicio de *urgencias hospitalarias*, ya que como mínimo hay que pasar por dos servidores, el triaje y la consulta. Está claro que este ejemplo se puede expandir mucho más y, posteriormente, se explicará.

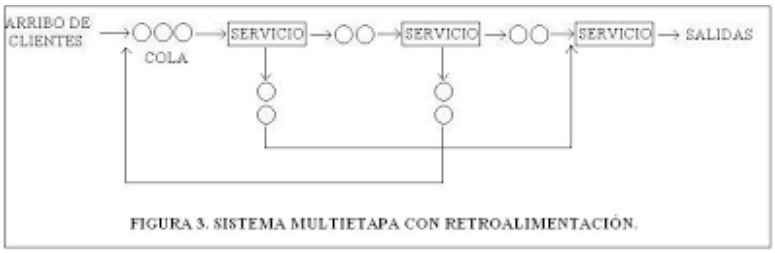


Figura 1.3. Sistema multietapa. Fuente: rephip

En la Figura 1.3 se puede observar cómo es un sistema de colas multietapa.

Otro elemento del sistema de colas, como su nombre indica es la *cola*. Esta nomenclatura hace referencia a los clientes que esperan para ser atendidos.

Por último, se encuentra *el sistema de la cola.* Este es el conjunto formado por la cola, el mecanismo de servicio y la disciplina de la cola, la cual nos proporciona el criterio de elección del cliente para entrar al mecanismo de servicio.

1.2.2 Terminología

A continuación, se presenta el conjunto de variables que se utilizarán como notación en este informe, todas estas relativas al estudio y a la *teoría de colas*:

$N(t)$= Número de clientes en el sistema en el momento t. Esta variable es un proceso estocástico con espacios de estados discretos y en tiempo continuo.
$N_q(t) =$ Longitud de la cola, denota el número de usuarios en la cola en el instante t.
s= Indica el número de servidores en el sistema.
$P_n(t)$= Denota la probabilidad de que, en el momento t, se hallen n clientes en el sistema.

μ_n= Es el número medio de clientes a los que se le da servicio completo, por unidad de tiempo, cuando ya existen n usuarios en el sistema. También se conoce como tasa de servicio promedio. Cabe destacar que, si en todos los servidores del sistema la distribución del tiempo de servicio es la misma, se suele denotar como μ. Como efecto, se observa que $\mu_n = n * \mu$ si $n = 1,2,3, \dots, s$ y $\mu_n = s * \mu$ para $n \geq s$.

λ_n= Denota la tasa de llegadas promedio o, lo que es lo mismo, el número esperado de llegadas por unidad de tiempo. Cuando estas son constantes, es decir, que no dependen de n, se representa como λ.

ρ= Representa el factor de utilización del sistema o, lo que es lo mismo, la fracción de recursos del sistema que es consumida por los clientes.

Se define, como

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

Cuando los λ_n son una constante para todo n, se denota como λ. $s * \mu$ es el número medio de usuarios a los que se le puede dar servicio, ya que es la multiplicación entre el número de servidores del sistema por la tasa de servicio medio.

Se menciona que en un largo periodo de tiempo el sistema de cola llega a un estado estable. Esto se puede alcanzar cuando dicho aspecto es independiente del estado inicial y del tiempo transcurrido. Se implanta que cualquier situación puede llegar a su condición de estado estable si $\rho = \frac{\lambda}{s\mu} < 1$. La *teoría de colas* se enfoca ampliamente en esta restricción de etapa estable. Si esta no se cumple, será más difícil estudiar el sistema de forma analítica.

Cuando el sistema es estable, tiene sentido definir los siguientes conceptos:

N = Variable aleatoria que cuenta el número de usuarios en el sistema.

N_q= Variable casual que contabiliza el número de clientes en cola

L= Denota el número promedio de clientes en el sistema. Se calcula hacienda la esperanza de N, $L = \mathrm{E}(N)$.

L_q= Representa el número promedio de usuarios en la cola, $L_q = \mathrm{E}(N_q)$.

$\mathcal{W}$= Variable aleatoria que indica el tiempo que pasa un cliente en el sistema.

$\mathcal{W}_q$= Denota el tiempo que espera un cliente en cola.

W= Tiempo promedio que el usuario está dentro del sistema. $W = \mathrm{E}(\mathcal{W})$.

W_q= Representa el tiempo medio de espera en cola de los usuarios. $W_q = \mathrm{E}(\mathcal{W}_q)$.

Para poder realizar una clasificación de todos los posibles escenarios de los sistemas de colas, se deben especificar las características que determinan los elementos que lo componen. La nomenclatura más habitual hoy en día para los sistemas de colas es A/B/s/K/H/Z, donde:

- A es la distribución del tiempo entre llegadas. Las abreviaturas más utilizadas son: M(Exponencial), D(Determinista), E_k(Erlang), U(Uniforme) o G(Genérica).
- B es la distribución del tiempo de servicio. Las abreviaturas son las mismas que en el caso del A.
- S es el número de servidores en el sistema, como ya se ha mencionado anteriormente.
- Se denota K como la capacidad de la cola, pudiendo ser finita o infinita.
- H es el tamaño de la población.
- Z es la disciplina de la cola. Algunas de las abreviaturas son FIFO, LIFO, entre otras muchas.

1.2.3 Distribuciones que se utilizan para la Teoría de Colas

A continuación, se van a explicar las diferentes distribuciones que se han utilizado para realizar la simulación de un sistema de *urgencias hospitalarias.* En este caso se han utilizado 6 tipos de distribuciones, las cuales son:

1.2.3.1 Distribución exponencial

Cuando se habla de distribución exponencial normalmente se refiera a la cantidad de tiempo que sucede hasta que se produce algún evento en específico[2].

[2] Definición y explicación de la Distribución exponencial

Una de las características de esta distribución probabilística es que la variable aleatoria correspondiente, aporta más valores pequeños que grandes.

En la gran mayoría de casos, esta distribución se aplica a cálculos que están relacionados con el tiempo, como por ejemplo, el tiempo entre llegadas en un sistema de colas. A continuación, se explicará, en formato matemático, la relación entre la distribución exponencial y la *teoría de colas*.

M= Variable aleatoria del tiempo entre llegadas o del tiempo de servicio.

$f_M(t) = \begin{cases} \alpha e^{-\alpha t} & t \geq 0 \\ 0 & t < 0 \end{cases}$ estrictamente decreciente en t

$$E(M) = \frac{1}{\alpha}$$

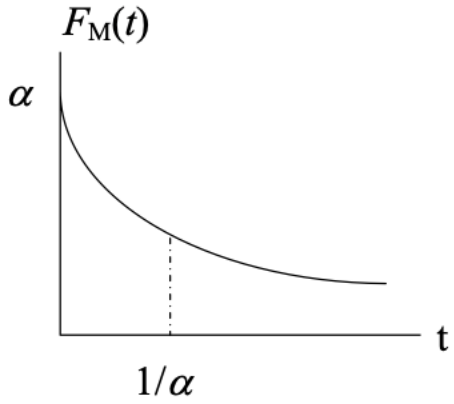


Figura 1.4. Distribución Exponencial. Fuente: Universidad de Valencia

Una de las principales propiedades de esta distribución es que no tiene memoria. Esto significa que la distribución de la probabilidad del tiempo que falta para que suceda el evento es siempre la misma independientemente del tiempo que haya pasado.

$$P\{M > t + \Delta t \mid M > \Delta t\} = e^{-\alpha t}$$

1.2.3.2 Procesos de Poisson

La distribución de Poisson es una distribución probabilística discreta que se aplica a sucesos durante un intervalo de tiempo determinado[3]. La variable aleatoria asociada representa una cifra de ocurrencias de un acontecimiento en un horizonte temporal.

Si los tiempos entre llegadas y tiempos entre servicios se distribuyen según una exponencial, el número de llegadas o servicios hasta un determinado horizonte es un proceso de Poisson.

Denotamos $S(t)$ como el número de ocurrencias tanto de llegadas como de servicios en el momento t, siendo $t \geq 0$. Esta se distribuye según una Poisson con parámetro $\alpha(t)$, siendo α el número promedio de ocurrencias por unidad de tiempo

$$P\{S(t) = n\} = \frac{(\alpha t)^n e^{-\alpha t}}{n!} \qquad n = 0,1,2,\ldots. \qquad E[S(t)] = \alpha t$$

Por la falta de memoria de la distribución exponencial, la probabilidad de que ocurra un suceso en un intervalo pequeño de tiempo de longitud Δt sabiendo que no se ha producido hasta el período t es directamente proporcional a $\alpha \Delta t$.

1.2.3.3 Distribución Uniforme

La distribución uniforme[4] se puede considerar como proveniente de un proceso de extracción aleatoria. El planteamiento radica en el hecho de que la probabilidad se distribuye uniformemente a lo largo de un intervalo. Por tanto, dada una variable aleatoria continua, x, definida en el intervalo [a, b] de la recta real, se dice que x tiene una distribución uniforme en dicho intervalo cuando su función de densidad para
X → $U[a, b]$ es:

[3] Definición y explicación de la Distribución Poisson
[4] Definición y explicación de la Distribución Uniforme

$f(x) = \frac{1}{b-a}$ para x [a, b].

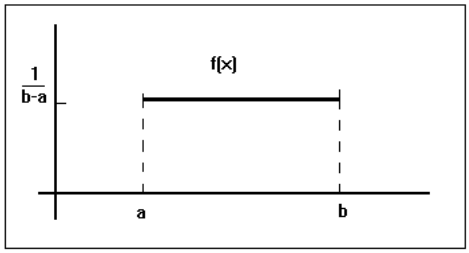


Figura 1.5. Función de densidad de la Distribución Uniforme. Fuente: Universidad de Valencia

De tal forma que su función de distribución será la siguiente:

$$F(X) = P(x \leq X) = \begin{cases} 0 & si\ X < a \\ \int_a^X \frac{1}{b-a} dx = \frac{x-a}{b-a} & si\ a \leq X \leq b \\ 1 & si\ X \geq b \end{cases}$$

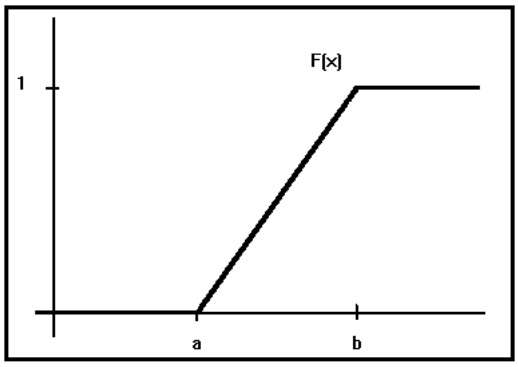


Figura 1.6. Función de distribución de la Uniforme: Fuente: Universidad de Valencia

1.2.3.4 Distribución Beta

La Distribución probabilística beta[5] se define en un intervalo [0,1] que se parametriza con dos variantes positivas, denotadas como α y β. Estas aparecen como exponentes de una variable aleatoria para controlar la forma de la distribución.

[5] Definición y explicación de la Distribución Beta

Dadas dos variables aleatorias $Y =^d \gamma(1,p)$ y $Z =^d \gamma(1,q)$, independientes, se dice,

$$X = \frac{Y}{Y+Z}$$

Tiene distribución $\beta(p,q)$, beta de parámetros p y q. Se puede demostrar que en la definición podrían haberse tomado variables con distribuciones $Y =^d \gamma(a,p)$ y $Z =^d \gamma(a,p)$, ya que la distribución de la X resultante no depende del a elegido. La función de densidad de una $\beta(p,q)$ viene dada por

$$f(x) = \begin{cases} \dfrac{x^{p-1}(1-x)^{q-1}}{\beta(p,q)} & si\ x\ \in [0,1] \\ 0 & en\ caso\ contrario \end{cases}$$

Siendo $\beta(p,q) = \int_0^1 x^{p-1}(1-x)^{q-1}dx$

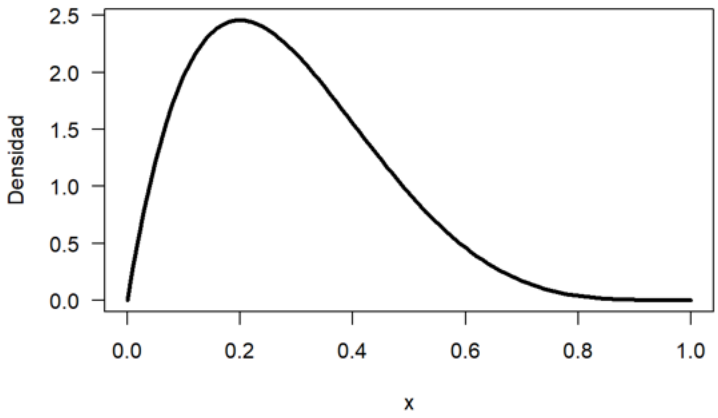


Figura 1.7. Función de densidad de una Distribución Beta. Fuente: github

1.2.3.5 Distribución Log-Normal

Se dice que una variable aleatoria se distribuye según una log-normal[6] si el logaritmo de la variable aleatoria tiene una distribución normal. Debido a esto, hay muchas similitudes matemáticas entre las dos distribuciones.

[6] Definición y explicación de la Distribución Log-Normal

La función de distribución viene dada por:

$$P(X) = \frac{1}{S\sqrt{2\pi}} e^{-\frac{(\ln(x)-M)^2}{2S^2}}$$

La distribución de densidad de probabilidad y función de probabilidad vienen dadas por

$$P(X) = \frac{a - b^2}{x^{a+1}}$$

$P(X) = 1 - \left(\frac{b}{x}\right)^a$ con $x \geq b$

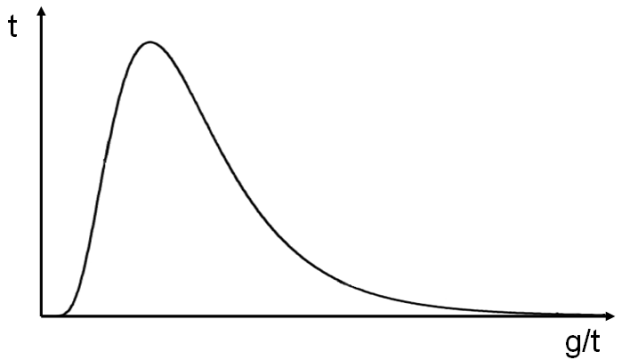


Figura 1.8. Función de densidad de probabilidad de la Log-Normal. Fuente; Estudiarfisica

1.2.3.6 Distribución Triangular

Una distribución triangular[7] se denomina cuando viene dada por tres parámetros, que representan el valor mínimo y el valor máximo de la variable, y el valor del punto en el que triángulo toma su altura máxima. En este caso, el triángulo no tiene por qué ser equilátero.

[7] Definición y explicación de la Distribución Triangular

La función de densidad de probabilidad para la distribución triangular es la siguiente:

$$f(x|a,b,c) = \begin{cases} \dfrac{2(x-a)}{(b-a)(c-a)} & para\ a \leq x \leq c, \\ \dfrac{2}{b-a}\ para\ x = c, \\ \dfrac{2(b-x)}{(b-a)(b-c)}\ para\ c < x \leq b \end{cases}$$

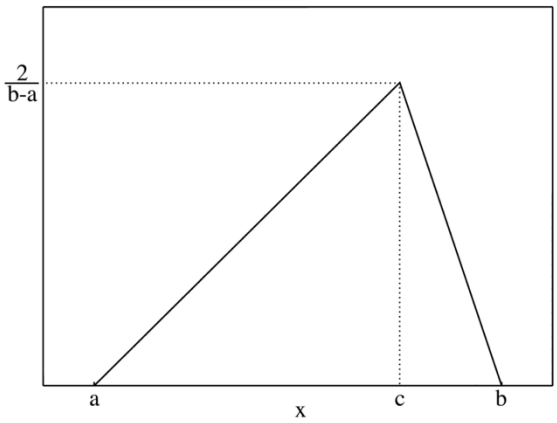


Figura 1.9. Función de densidad de probabilidad de la Triangular. Fuente; Wikipedia

1.2.4 Ley de Little

En esta sección se establecerá las llamadas fórmulas de Little[8]. Dichas ecuaciones se dan en los modelos con distribución del tiempo entre llegadas y distribución del servicio exponencial, pero también en las más generales llamadas ergódicas. Esta ley relaciona el número medio de clientes en el sistema de colas con el número promedio de llegadas por unidad de tiempo y la tasa media que un cliente está dentro del sistema.

Cuando las tasas de llegadas no son constantes (λ_n) se expresa como:

$$L_q = \lambda * W_q$$

[8] Ley de Little

Mientras que si esta es constante ($\lambda_n = \lambda$) para todo $n = 1,2,3, \ldots$

$$L = \lambda * W$$

Para poder entender la validez de estas fórmulas, es considerar que un usuario llega al sistema en este mismo momento. Pasado un tiempo (W), este será servido y saldrá del sistema. Como el número promedio de clientes que llegan al sistema por unidad de tiempo es igual a la tasa de llegadas (λ) y el número medio de usuarios que habrá llegado desde que dicho cliente entró al sistema hasta que salió, es $\lambda * W$.

Si las λ_n no son constantes, para poder validar las fórmulas expuestas anteriormente, se han de generalizar de la siguiente manera:

$$L = \bar{\lambda} * W$$
$$L_q = \bar{\lambda} * W_q$$

Siendo $\bar{\lambda} = \sum_{n=0}^{inf} \lambda_n * p_n$

La última fórmula importante de la ley de Little es la relación entre W y W_q:

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

1.2.5 Modelos de colas simples

La finalidad de este apartado es exponer los diferentes modelos de colas simples. Concretamente se hará un recorrido por los modelos M/M/1, M/M/C/C y los modelos más generales G/G/1 y G/G/C.

1.2.5.1 El sistema M/M/1

Como su nombre indica es aquel en el que la distribución del tiempo entre llegadas de los pacientes en el sistema sigue una distribución exponencial, λ. Los tiempos de servicio siguen la misma distribución que la tasa anterior, eso sí, cambiando el parámetro a μ. La última peculiaridad de este modelo es que únicamente tiene un servidor[9]

Pues bien, teniendo en cuenta los parámetros definidos anteriormente y los libros de *teoría de colas* adjuntados en la bibliografía resulta que:

- Las tasas de llegada se definen como $\lambda_n = \lambda\ , \forall\ n = 0,1,2, \ldots$
- Mientras que los tiempos de servicio se definen como $\mu_n = \mu\ , \forall\ n = 1,2,3, \ldots$

Gracias a estas dos, se puede calcular la probabilidad de que haya n usuarios en el sistema y, por ende, el nivel de *saturación*.

$P_n = (1 - \rho) * \rho^n$ donde $\rho = \frac{\lambda}{\mu}$ (Nivel de *saturación* del sistema)

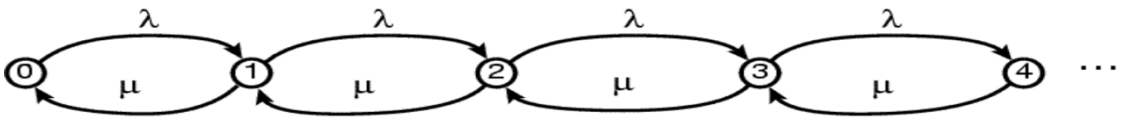


Figura 1.10. Sistema M/M/1. Fuente: Universidad de Murcia

Como se puede observar en la Figura 1.10, cada nodo representa un estado, cada arco o línea representa la transición de los clientes en el sistema y, en estos, está representado tanto la tasa de llegadas (λ) como las de servicios (μ).

1.2.5.2 Modelo M/M/C/C

[9]Consulta más información sobre el modelo M/M/1

Existe un caso especial con este modelo, ya que no genera cola debido a que el límite de capacidad y el número de servidores coinciden. Este sistema de colas sigue una distribución exponencial para ambas tasas. Se considera que el número de fuentes es infinito, por lo que el período de llegadas es constante. Cada uno de los múltiples servidores tendrá una duración de servicio[10].

Como ya se ha mencionado anteriormente, la característica más llamativa de este sistema es que no general cola, por tanto, los usuarios que encuentren los servidores ocupados se perderán (salida del sistema) sin tener una probabilidad de poder ser almacenadas.

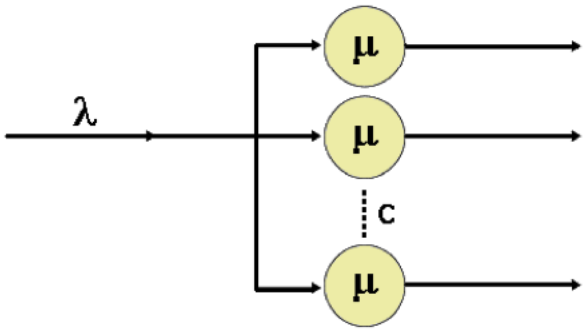


Figura 1.11. Representación gráfica del modelo M/M/C/C. Fuente: Universidad Militar de Granada

Al tener la peculiaridad de la cola, los valores de los parámetros L_q y W_q serán 0, mientras que el tiempo total del sistema coincide con el tiempo de servicio de los múltiples servidores, $W_s = \frac{1}{\mu}$.

1.2.5.3 Modelos generales de colas (G/G/C)

En este apartado se explicarán algunas situaciones en las que las distribuciones tanto de las tasas de llegada como los períodos de servicio no siguen una exponencial. En la vida real, rara vez estas siguen una exponencial o Poisson. Es por eso, la importancia de estos modelos generales.

[10] Consulta más información sobre el modelo M/M/C/C

Existen tres modelos generales, M/G/1, G/G/1 y, el más general de todos G/G/C. El primer de ellos indica que los usuarios llegan según una distribución exponencial/Poisson, pero se asume que son servidos mediante un proceso más general de duración media $\left(\frac{1}{\mu}\right)$ y desviación estándar σ.

Los científicos Pollaczek y Khintchine[11] demostraron la fórmula denominada "P-K" que permite evaluar la longitud media de la cola del sistema.

$$L = \frac{\lambda}{\mu} + \frac{\left(\frac{\lambda}{\mu}\right)^2 + \lambda^2 * \sigma^2}{2 * \left(1 - \frac{\lambda}{\mu}\right)}$$

Además, a partir de la anterior ecuación se extrae que:

$$W_q = \frac{\left(\frac{\lambda}{\mu}\right)^2 + \lambda^2 * \sigma^2}{2 * \left(1 - \frac{\lambda}{\mu}\right)}$$

Cuando la tasa de llegadas de clientes no sigue una distribución exponencial, se denomina G/G/1. Para poder calcular el tiempo promedio de espera en cola en función de los coeficientes de variación, se utiliza la difusión de Kingman[12].

Esta aproximación se formula de la siguiente manera:

$$W_q(G|G|1) \approx \left(\frac{\lambda^2 * \sigma_e^2 + \mu^2 * \sigma_s^2}{2}\right) * W_q(M|M|1)$$

Por último, pero el más importante, se encuentra el modelo G/G/C. Este es el más utilizado en la vida real y, además, se ha utilizado en el proyecto presente.

[11] Científicos Pollaczek y Khintchine
[12] Consulta más información sobre la difusión de Kingman

Si el primer modelo general no es una generalización exacta, menos aún será el G/G/C. Debido al minucioso error que proporciona dicho modelo, resulta interesante la ecuación siguiente ya que permite el cálculo del tiempo medio en cola para cualquier sistema:

$$W_q(G|G|C) \approx \left(\frac{\lambda^2 * \sigma_e^2 + \mu^2 * \sigma_s^2}{2}\right) * \frac{\rho^{\sqrt{2*c+2}-1}}{c(1-\rho)} * \frac{1}{\mu}$$

CAPÍTULO 2

Descripción del problema.

2.1 Planteamiento del problema

Hoy en día en muchos establecimientos tanto físicos como digitales se puede observar que se tiene que esperar un determinado tiempo para poder finalmente ser atendido. Este fenómeno se observa en establecimientos para realizar una compra, en la peluquería, centros de atención telefónicos, entre otros muchos.

Cabe destacar que algunas de las empresas con más volumen de ventas del mundo invierten mucho capital en el estudio del comportamiento de su sistema de atención, para así poder responder a las siguientes cuestiones que se plantean: ¿Cuánto tiempo el cliente está siendo atendido? ¿Cuál es el período en el que el usuario está en lista de espera? ¿Cuántos clientes están siendo atendidos y cuántos usuarios están esperando a ser servidos?

Este informe trata de responder a estas preguntas planteadas por todos los directivos de las compañías. Pero, como es lógico, cada sector económico tiene su infraestructura y procedimientos para satisfacer las necesidades que demandan los clientes. Es por ello, que este proyecto va dirigido a un sector en específico, el sanitario, concretamente, al sistema de *urgencias hospitalarias*.

En general, los servicios de urgencias tienen la misma estructura interna independientemente de si se trata de un ente público o privado como se observa en la Figura 2.1.

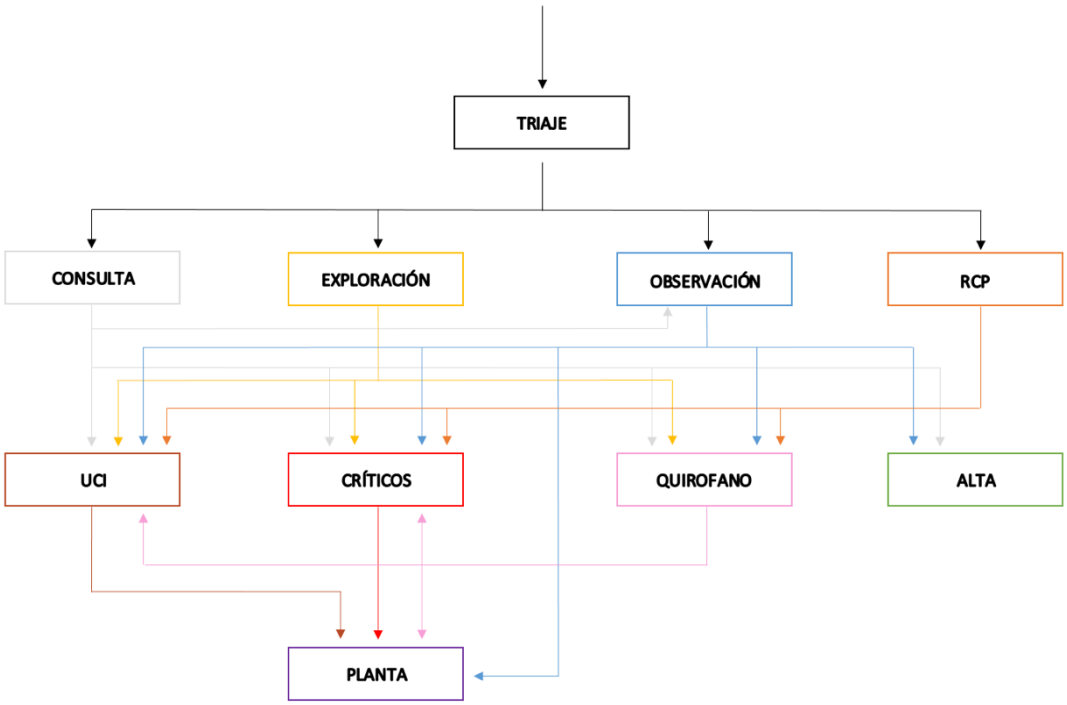


Figura 2.1. Grafo de un sistema de urgencias hospitalarias. Fuente: Elaboración propia

Para la realización del sistema se ha consultado la página web del Ministerio de Sanidad del Reino de España[13] para así obtener una representación lo más realista posible, con sus respectivos estándares.

El sistema está separado en 4 bloques, para así hacerlo más fácil de entender. El primer bloque, punto de entrada, es Triaje. El segundo está formado por la consulta, exploración, observación y RCP. El tercer bloque está dotado de la UCI, críticos y quirófano. Por último, se encuentran los puntos de salida del sistema, los cuales son alta y planta. Este último lugar del hospital es el más importante, ya que más adelante se comprobará el nivel de *saturación* para cada uno de los distintos *escenarios* que se puedan dar.

Hay que recalcar que tanto el número de servidores como el tiempo de servicio en cada uno de los nodos de la Figura 2.1, que se procede a explicar a continuación, son parámetros elegibles, es decir, se puede indicar el número de servidores disponibles para cada uno de los días y apuntar que tipo de distribución se necesita para generar los tiempos de servicio. Además, cada período de servicio en los nodos es distinto, como es lógico.

[13] Ministerio de Sanidad en el Reino de España

Explicados los matices anteriores, el primer nodo al que los pacientes llegan al sistema es el triaje, el único punto de entrada al servicio de urgencias. En este, se clasifica a los individuos según su nivel de gravedad e intervención. Las rutas existentes para el triaje son: consulta; exploración; observación y RCP. Por ejemplo, si un cliente llega con síntomas de deshidratación, se le indicará que tendrá que ir a exploración, eso sí, esperando en cola hasta que el anterior cliente termine con su respectivo servicio.

En segundo lugar, se encuentra la consulta, la más concurrida de todos los sitios existentes en el hospital. De este eje los usuarios pueden dirigirse a UCI; críticos; quirófano; observación o alta.

Seguidamente, cuando el origen del paciente sea triaje y su destino sea exploración o RCP, la próxima ruta a seguir será la misma para ambos casos. Podrán ir a UCI; críticos o quirófano. Cabe destacar que, para cada uno de los nodos iniciales, es decir, exploración y RCP, tendrán una cola, tiempo de espera en cola y períodos de servicio distintos.

Para terminar con los nodos del segundo bloque del sistema, se encuentra observación. Este tiene la peculiaridad de tener dos accesos a él. Por una parte, están los individuos que entrar desde el nodo inicial/triaje. Y por la otra parte, se encuentran aquellos pacientes que vienen desde consulta, debido a que los profesionales han observado alguna anomalía. Por tanto, se debe tener en cuenta que los tiempos en cola varían en función de cuantos pacientes entran al sistema y cuantos de ellos después de pasar por consulta se dirigen a observación.

De forma similar son las rutas del bloque 3. Los ejes de UCI y críticos pueden recibir pacientes de exploración, observación, RCP y quirófano. Pero, el primer nodo mencionado tiene una entrada más, la gente que viene directamente de consulta. Cuando los pacientes terminan sus respectivos servicios en ambos lugares del hospital son llevadas a planta, que tendrán su tiempo en cola (si la hay) y sus tiempos de servicio.

Al contrario que los dos ejes anteriores del bloque 3, el quirófano está caracterizado en este sistema por “dar” pacientes a lugares de su mismo sitio, es decir, desde el quirófano se puede ir a UCI y a críticos, además de subir directamente a planta.

El último bloque que se encuentra representado en la Figura 2.1, son los puntos de salida del sistema, denotado como planta y alta. Como ya se ha mencionado anteriormente, el eje con el que más presión e incertidumbre por la *saturación* hay es planta. La no congestión de esta es tan importante como la satisfactoriedad de una operación. Es por esa razón por la que su situación se estudia minuciosamente en todo el horizonte temporal. Por el contrario, el alta no está involucrado en el estudio de este informe, por lo que solo sirve para saber cuántos pacientes son admitidos como aptos para poder salir del sistema de *urgencias hospitalarias*.

Para poder realizar una correcta evaluación de la red de la Figura 2.1, existen 2 formas, eso sí, asumiendo unos recursos ilimitados.

Por un lado, se puede recrear las circunstancias de la red en la realidad, lo que conlleva un coste económico y un gasto del personal enorme. El esfuerzo que se tendría que hacer para llevar a cabo la contabilidad de los pacientes que entrar, el tiempo que dedican dentro del sistema es desorbitado. Por lo que no es lo óptimo.

Por el otro lado, y lo más sensato, es realizar una simulación mediante, valga la redundancia, un *simulador*. Es por eso, que se presenta un algoritmo capaz de simular un sistema de *urgencias hospitalarias* mediante la aplicación de *teoría de colas* con una interfaz de usuario para facilitar el análisis de los resultados.

2.2 Motivación

El tiempo de espera ha sido y es un problema que afecta a la gran mayoría de sistemas tanto públicos como privados del sector sanitario. Dicha cuestión no solamente se habla dentro del área de salud sino también en las calles de la sociedad española.

Si se hiciera una encuesta a la población que reside en este país sobre cuantas horas tienen que esperar para poder ser atendidos en urgencias, esta saldría muy desfavorable.

Estos tiempos de espera afectan a la paciencia de las personas que se encuentran en el hospital para ser atendidos, pero también, a la relación médico-paciente. Además, los gestores sanitarios, a los que les recae el peso de la resolución, indican en muchas de las manifestaciones que realizan al cabo del año, que necesitan más personal y medios para que no haya tanta insatisfacción.

Si los tiempos de espera eran bastante desorbitados, fue a partir del mes de marzo del 2020, que este tema se descontroló. Debido a la pandemia de la COVID-19, acudir al sistema de urgencias de cualquier hospital de España, era frustrante, ya que si por desgracia una persona había enfermado debido a este virus (que eran muchas) el tiempo que había que estar esperando a ser atendido era desorbitado. Por no comentar, los niveles de *saturación* que se llegaron a alcanzar.

Por estas dos razones comentadas anteriormente, se ha procedido a realizar un algoritmo de simulación matemática para un sistema de *urgencias hospitalarias* general para así, poder dimensionar y optimizar la infraestructura de dicho servicio y conseguir la menor *saturación* posible.

Cabe destacar que la opción de la simulación es la más indicada, porque realizar el conteo de los pacientes que puedan llegar al sistema y los distintos cálculos no es viable en el sentido del tiempo ni del gasto que pueda suponer. El beneficio principal que aporta el simulador científico, junto a otros, es la rapidez en la obtención de los datos finales para su correcta interpretación, además de realizar modificaciones de los parámetros del sistema en cualquier instante.

2.3 Objetivos y materiales y métodos

2.3.1 Objetivos

Analizar por simulación un servicio de urgencias en un hospital. Para ello se desarrolla una herramienta para la toma de decisiones que permite analizar los tiempos de servicio de los pacientes que acuden a urgencias dependiendo de los parámetros que definan dicho servicio. Con esta herramienta se puede optimizar el diseño y dimensionamiento de las unidades que componen el servicio de urgencias en un hospital.

Para lo cual es necesario:

1. Comprender el modelo matemático, *teoría de colas*.
2. Programar un algoritmo científico para realizar la simulación.
3. Llevar a cabo una interfaz gráfica de usuario fácil de utilizar.

2.3.2 Materiales y métodos

Se programó un *simulador* de eventos de urgencias de un hospital en *Rstudio*, con la capacidad de permitir al usufructuario configurar los parámetros de una red de colas. Este algoritmo tiene la característica de ofrecer una interfaz de usuario rápida para analizar de forma gráfica y numérica los resultados obtenidos. Además, da la posibilidad de descarga de un archivo XLSX, el cual contiene la trazabilidad de los clientes en el sistema, para en un informe futuro realizar análisis estadísticos, como por ejemplo, predecir cuantos pacientes acuden a consulta en un periodo de 2 días.

Para la creación del *simulador* matemático se siguieron una serie de pasos que a continuación se proceden a explicar.

Conviene subrayar, que antes de empezar a programar hace falta tener un esquema de cuáles son los objetivos del proyecto y como se puede conseguir esos “targets”, además de poseer un grafo (Figura 2.1) del sistema a simular.

Una vez aclarados los objetivos y la estructura, en este caso, del sistema de *urgencias de un hospital* en general, se procede a entender los teoremas y colorarios que explican la *teoría de colas*. Posteriormente, ya se puede acudir al software a realizar el algoritmo matemático.

En el presente proyecto, se ha decidido utilizar el software *Rstudio*[14] el cual utiliza el lenguaje de programación R[15]. La misión que tiene dicha aplicación informática es la de "crear un software libre y de código abierto para la ciencia de datos, la investigación científica y la comunicación técnica para facilitar la colaboración y la investigación reproducible".

Este software utiliza el lenguaje R, que es el padre de la ciencia de datos en la actualidad junto con Python[16].

Mientras se programa el *simulador* se tiene que ir verificando si las rutas y resultados del sistema de urgencias coinciden con lo que se había modelado. Por último, cuando el algoritmo no proporciona ningún error en sintaxis, se procede a realizar la validación del modelo. Si esta es favorable, la simulación estará terminada y se podrán realizar los cálculos pertinentes de la *teoría de colas.*

[14] Consulta más información sobre Rstudio en este enlace
[15] Consulta más información sobre el lenguaje R en este enlace
[16] Consulta más información sobre Python en este enlace

CAPÍTULO 3

Simulación.

Muchos de los problemas de la vida cuotidiana no se pueden resolver mediante métodos analíticos. Una de las principales causas es por la existencia de unos patrones no estandarizados de entrada y de servicio que generan una gran complejidad del sistema a estudiar.

En ocasiones, los resultados analíticos son para un estado estacionario que nunca se alcanza, debido a la interrupción que se produce antes de abandonar o salir del estado.

En estos casos, el estudio de las colas mediante una simulación puede ser la clave para poder encontrar de forma óptima los resultados.

3.1 Introducción a la simulación

La simulación es una técnica empleada para la solución de modelos estocásticos, además de ser una herramienta muy útil para el análisis del desempeño de los sistemas computacionales. Independientemente de si el sistema está implementado o no, utilizar un *simulador* es conveniente para hacer comparaciones con diferentes alternativas y probar opciones de cargas y ambientes de trabajo.

A pesar de sus múltiples ventajas, los modelos de simulación pueden malograrse o producir resultados que no son útiles. Esto se debe, por ejemplo, a un mal modelaje del nivel de detalle conforme a las mediciones que se quieran detallar. Esta cota de detalle para realizar su estructura conlleva un costo muy grande en cuanto al tiempo de desarrollo, para encontrar erratas en el código y, incluso, un período de computación. Aunque un *simulador* con más precisión proporcionará mejores resultados, se puede dar el caso que realizar suposiciones erróneas con algunos de las aclaraciones en los datos del modelo, puedan ocasionar resultados incoherentes, debido al gran nivel de exactitud y conocimientos de los métodos simulados.

Para realizar una correcta simulación, se tendrá que empezar por un nivel bajo de detalle y, una vez comprobado la validez del modelo, aumentar su exactitud en los resultados. Además, se tiene que considerar que lenguaje de programación se utilizará y, si este va a ser uno de uso general o específico.

Si se pretende generar un *simulador* con modelos estocásticos que dependen de variables aleatorias, se deberá tener especial atención a la hora de generar estos valores y de la semilla que se utilice.

3.2 Generación de números aleatorios

Los números aleatorios son una de las principales bases de la simulación[17]. Habitualmente, toda aleatoriedad en un modelo se obtiene en función de un generador de números aleatorios que produce una sucesión de resultados supuestamente ocasionados por una secuencia de variables independientes e idénticamente distribuidas según una uniforme (0,1).

Los valores aleatorios no son generados por un algoritmo matemático determinístico, es por eso por lo que en el ámbito de la computación se tienen que realizar mediante números pseudo-aleatorios. Para la correcta generación de este tipo de valores, se crean una serie de números en un subconjunto finito de N, usualmente $\{0, \dots, m-1]$, donde $m \in N$. A partir de estas series, se procesan valores aleatorios según una distribución probabilística uniforme (0,1).

3.3 Métodos de Monte Carlo

El término Monte Carlo se sobrepone a un conjunto de métodos matemáticos que se empezaron a utilizar en el 1940[18] para el desarrollo de armas nucleares. Consiste en dar una solución a un problema mediante la intervención de juegos de azar cuyo comportamiento simula algún fenómeno real según una distribución probabilística.

[17] Consulta más información sobre la generación de números aleatorios
[18] Consulta más información sobre los métodos de Monte Carlo

Matemáticamente, se denomina método de Monte Carlo a aquellos procesos estocásticos cuya evolución viene determinada por sucesos aleatorios.

Para entender mejor estos modelos, se expone un ejemplo de las agujas de Buffon para estimar π.

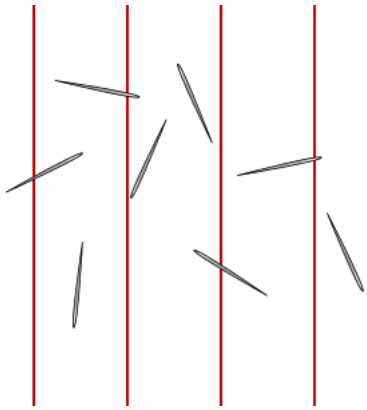


Figura 3.1. Experimento de las agujas de Buffon para estimar π. Fuente: Madrimasd

George Louis Leclerc en 1773, conde de Buffon, propuso el planteamiento del siguiente problema. Se considera un papel horizontal, en el que se ha trazado rectas paralelas separadas por una distancia d, sobre el que se dejan caer de forma aleatoria agujas de longitud $L < d$ (Figura 3.1). Entonces, este se preguntó: ¿Cuál es la probabilidad que las agujas crucen alguna de las rectas ilustradas?

Pues bien, la respuesta a la pregunta planteada la resolvió él mismo en el 1777, dando lugar a la siguiente solución:

$$P = \frac{2 * L}{\pi * d}$$

Esta fórmula permite estimar el valor de π, aunque la convergencia es muy lenta.

CAPÍTULO 4

Diseño e implementación.

Como ya se ha explicado en el capítulo 2 de este proyecto, se ha diseñado e implementado un algoritmo matemático capaz de simular una red del sistema de *urgencias hospitalarias*. Dicha aplicación tiene la característica que puede ser modelado por los usuarios, es decir, se pueden cambiar una serie de parámetros en función de las disponibilidades de esta. Por tanto, se explicarán las particularidades de la programación y como se puede implementar en la vida real.

El algoritmo realizado en *Rstudio*[19] y llamado "SSUH" (Simulador de un Sistema de Urgencias Hospitalarias) está compuesto por tres grandes partes. La primera de ellas hace referencia a los parámetros cambiantes, por ejemplo: número de días a simular; número de servidores en cada nodo, etc.

Por el otro lado, se encuentra el *simulador* de la red a estudiar. Este bloque está dotado de todas las herramientas necesarias para la correcta simulación.

La última parte de todo el conjunto del algoritmo matemático es la realización de los cálculos de la *teoría de colas,* trazabilidad de los pacientes y la interfaz básica para el usuario.

4.1 Bloque 1. Inputs

En este apartado se procede a explicar el código necesario para poder realizar los inputs de los parámetros esenciales para la simulación de un sistema de *urgencias hospitalarias*.

[19] Consulta más información sobre Rstudio

Para el correcto funcionamiento del algoritmo es necesario la instalación de una serie de librerías en *Rstudio*[20]. Para más información de dichos paquetes informáticos, consultar la bibliografía.

Una vez hayan sido descargados y cargados de forma correcta los distintos paquetes matemáticos, se indicarán los valores de los inputs. El primero de ellos hace referencia al número de días a simular. No existe una cuantía límite en este, pero hay que tener en cuenta que a mayor número de días a simular mayor será el coste computacional de esta. Por lo que se recomienda simular como máximo 31 días, es a partir de este valor, que este coste aumenta de forma drástica.

A continuación, habrá que definir qué tipo de *escenario* se elige para la simulación del sistema hospitalario. Esta variable se refiere al número de llegadas de los pacientes al sistema, los cuales seguirán una distribución de Poisson (λ). Hay que matizar que a mayor valor de λ mayor será la cifra de pacientes que llegan. Para que los usuarios de la interfaz básica, que se ha creado, tengan una guía por si su conocimiento de la estadística es nulo, se proporciona tres opciones. El primero de ellos, es el *escenario 1*, el Bajo, el cual tiene un λ de 0.1, lo que significa que llegarán muy pocos pacientes al hospital. Posteriormente, se encuentra la opción Medio con un λ de 0.25. Y por último, el *escenario* Saturado, dotado de un λ de 0.5. Por tanto, dependiendo del tipo de marco que elija el usuario en la interfaz, el sistema tendrá un menor o mayor flujo de pacientes.

Posteriormente a la elección de la cantidad de días a simular y del tipo de *escenario* para el sistema, se encuentra el número de servidores que habrá en cada uno de los nodos que componen la estructura de urgencias. Como se observa en la figura 2.1 del capítulo 2, existen 10 nodos que componen el sistema a estudiar, 9 de los cuales estarán compuestos por n servidores. El único nodo que no tiene servidores y, por ende, no resulta interesante estudiarlo es alta.

Para la correcta implementación de los inputs, es necesario matizar que los valores que se introducen son los servidores disponibles en ese mismo día, no la cuantía global de cada nodo en general.

[20] Información sobre las librerías de Rstudio

Dentro de las 9 entradas de valores posibles, únicamente 1 de ellas hace referencia al número de personal médico disponible para poder atender a los pacientes. Esto se refiere, que el único eje que no depende de la cuantía de camas disponibles en el hospital es triaje o también conocido como recepción.

4.2 Bloque 2. Algoritmo matemático

Una vez explicado los distintos inputs, se proceda a la exposición del algoritmo realizado para la simulación de un sistema de *urgencias hospitalarias*. En este apartado se va a utilizar un leguaje más técnico, por lo que en la bibliografía existe un documento que explica de forma más "normal" dichos tecnicismos para su correcta interpretación.

Definido el número de días a simular, se convierte esa cuantía a minutos, (un día son 1440 minutos), para poder realizar un bucle "for" que dará lugar al cálculo de los momentos de llegadas de cada uno de los pacientes y cuáles son los tiempos entre llegadas de estos. El momento de llegada de los clientes se calcula de la siguiente manera. Para el primer usuario que llega al sistema, resulta fácil, ya que en este no hace falta hacer ningún cálculo porque en la variable "Pos" se encuentra su minuto de llegada. Es a partir del segundo en llegar que se complica. En este instante, se crea un bucle "if" que indica que, si el paciente no es el primero y llegan más de un sujeto en ese horizonte temporal, su momento de llegada será la posición del índice correspondiente en la variable "Pos" y su dimensionamiento será la longitud de la variable "Momento_llegada" + 1 + el cliente anterior en llegar a urgencias hasta la longitud máxima de individuos que entran al sistema. Si por alguna casualidad, en el instante t, únicamente existe una persona que entra al hospital, su momento de llegada será el valor correspondiente al índice en la variable "Pos" y su posición dentro del sistema será aquella en la que la longitud de la variable "Momento Llegada" + el número de clientes, sea inferior a la posición que toma este individuo dentro de la variable "Pos".

Por otra parte, para realizar los cálculos de los tiempos entre llegadas de los pacientes, es necesario haber realizado de forma correcta las deducciones de la variable "Momento_Llegada". Es a partir de este instante que se crea otro bucle "for", eso sí, fuera del rango del anterior, que irá de 1 hasta la longitud máxima de la variable mencionada anteriormente y el cálculo de los tiempos entre llegadas será igual a la suma del momento de llegada del paciente actual + el momento de llegada del paciente anterior.

En este instante, ya se tienen calculados tanto los momentos de llegadas como los tiempos entre llegadas de todos los pacientes que entrarán en los n días. Es por eso, que a continuación se declaran todas las variables necesarias para realizar los sistemas de colas en el hospital. Estos parámetros están ordenados según su orden de implementación en el algoritmo, es decir, las primeras variables declaradas son las correspondientes al nodo triaje y las últimas al nodo de planta.

Es a partir de aquí que empiezan los diferentes cálculos de los sistemas de colas en el hospital. Para la correcta explicación de como se ha elaborado el algoritmo se procede a desglosar el código en 5 bloques.

El primer bloque, es decir, la entrada de pacientes dentro del sistema es el nodo denotado como triaje. Para sus respectivos cálculos se ha seguido el siguiente esquema. Se crea un bucle "for" que su rango será desde 1 hasta la longitud máxima que tiene la variable "nclientes". Esta variable contiene la cuantía de todos los pacientes a atender. Dentro de este bucle se crea una variable llamada "Llegada_Triaje", la cual contiene los momentos de llegadas de todos los pacientes, y otra denominada como "P_Triaje" que identifica el ID de estos. Posteriormente, se crear los tiempos de servicio para cada persona que entra dentro de este bucle siguiendo una disciplina FIFO y, estos tiempos seguirán una de las siguientes distribuciones probabilísticas: exponencial; uniforme; Beta; Log-Normal o Triangular. Cabe matizar que dependiendo de que distribución se escoja, los tiempos de servicio cambian y, por ende, los tiempos medios en cola y todos aquellos parámetros que se basan en la *teoría de colas*.

Nada más crear los tiempos de servicio para cada paciente, se procede a entrar en una comparación mediante un bucle "if". Si el número de clientes que ha entrado a triaje es inferior al número de servidores disponible (input), el momento de terminación del paciente j será la suma entre su momento de llegada a triaje más el tiempo que ha sido atendido. Por lo contrario, si el contador de pacientes de este nodo es superior al número de servidores, que en este caso son recepcionistas, se procede a calcular el valor mínimo de la variable "Momento_Termimanción_Triaje" para poder realizar una comparación que indica si existe tiempo de espera en cola o no. Si resulta que el valor mínimo de los momentos de terminación de los pacientes anteriores es mayor al momento de llegada del individuo actual, este tendrá que esperar en cola hasta que uno que está siendo servido termine. Por el otro lado, si el paciente i llega en el instante t y hay algún servidor disponible entrará inmediatamente al servicio. Por lo tanto, los momentos de terminación de aquellos pacientes que tienen que esperar a ser servidores, será la suma entre su momento de llegada, el tiempo de espera y el tiempo de servicio.

Finalmente, cuando el paciente ha sido atendido se puede dirigir a varios nodos del sistema, como se ha estipulado en la Figura 2.1. Para eso, se crea una variable denominada "Clientes_Prioridad" que asignará de forma aleatoria su siguiente paso, pudiendo acceder a consulta, exploración, observación y RCP, mediante una serie de probabilidades que, en este caso serán 0.9; 0.03;0.06;0.01, respectivamente [1].

Una vez los pacientes son atenidos en triaje, su trayectoria pasará por el bloque 2 del sistema de *urgencias hospitalarias*. Esta agrupación está formada por el conjunto de 4 nodos: consulta; observación; exploración y RCP.

Para realizar nuevamente lo cálculos para la asignación de servidores de cada uno de los nodos mencionados anteriormente, se crea un bucle "for" que se le asignará los valores desde 1 hasta el total de clientes dentro del sistema.

Aquellos sujetos cuyo valor de la asignación realizada en el triaje sea "Con", se dirigirán a consulta mediante un bucle "if". Cuando el paciente j entra dentro de la consulta, se le calcula su momento de llegada y su ID de usuario. Seguidamente, se le asigna un tiempo de servicio mediante una función en *Rstudio* "sample", que dependerá de las 5 distribuciones probabilísticas elegidas. A continuación, se realiza el mismo

proceso que en el triaje. Si el contador de pacientes en consulta es menor al número de servidores el tiempo de espera del cliente será 0 y, por tanto, su momento de terminación será el resultado de la suma entre momento de llegada y el tiempo de servicio.

Por el otro lado, si el individuo *j* entra al sistema cuando los servidores están ocupados, tendrá que esperar hasta ser atendido. Por tanto, el momento de terminación de esta persona será la suma entre momento de llegada, tiempo de espera y período de servicio. Cuando el paciente ya ha sido atendido por el personal médico, se crea nuevamente una variable aleatoria, en este caso denotada como "Clientes_Prioridad2", para redirigirle a otro nodo del sistema, concretamente, se le asigna alta; observación, UCI, críticos o quirófano acorde a las siguientes probabilidades: 0.85,0.1,0.01,0.02,0.02 [2].

Tanto el nodo de exploración como RCP funcionan exactamente igual que la consulta, es decir, se aplican dos bucles "if" que compararan cual es la asignación que se le ha realizado al paciente j desde triaje. Si esta resulta que es "Exp", entrará en exploración y tendrá el mismo sistema de atención que en los dos casos anteriores. De igual modo, se hace en RCP. Finalmente, se le asignará a cada sujeto de los dos nodos un valor que indicará su siguiente paso. Esta asignación se realiza mediante la función "sample", en la que se le asigna una serie de probabilidades a cada elemento. Los pacientes que se encuentren en exploración podrán ir a UCI (0.35), críticos (0.4) o quirófano (0.25) [3], mientras que los pacientes de RCP podrán ir a los mismos nodos, con unas probabilidades distintas 0.25,0.6 y 0.15.

Si la variable "Clientes_Prioridad" toma el valor "Obs" para cualquier paciente, significa que este irá desde triaje a observación. Pero, existe la peculiaridad que cualquier individuo que haya estado en consulta puede ir a Observación. Es por eso por lo que antes de calcular los momentos de terminación en este nodo, se realiza un bucle "for" que recorrerá los índices de todos los pacientes del sistema para comprobar cuál de ellos tiene asignado el nodo observación y su nodo anterior sea consulta. Una vez encontrados, se le asigna como momento de llegada a observación su instante de terminación en la consulta.

Por consiguiente, se crea otro bucle "for" pero esta vez de longitud 1 hasta el máximo del contador que se ha realizado en la identificación anterior. Y es a partir de aquí que el proceso es exactamente el mismo que en los nodos anteriores. En cuanto a la asignación de la ruta de los pacientes provenientes de observación, se tiene que podrán ir a UCI, críticos, quirófano, alta o planta con 0.01;0.04;0.1;0.7;0.15, respectivamente.

Como se ha visto en la explicación tanto del primer bloque como la del segundo, el mecanismo de servicio que se utiliza siempre es el mismo. Se trata de un sistema de colas multicanal con una única cola para los n servidores existentes. La figura 4.1 representa este mecanismo.

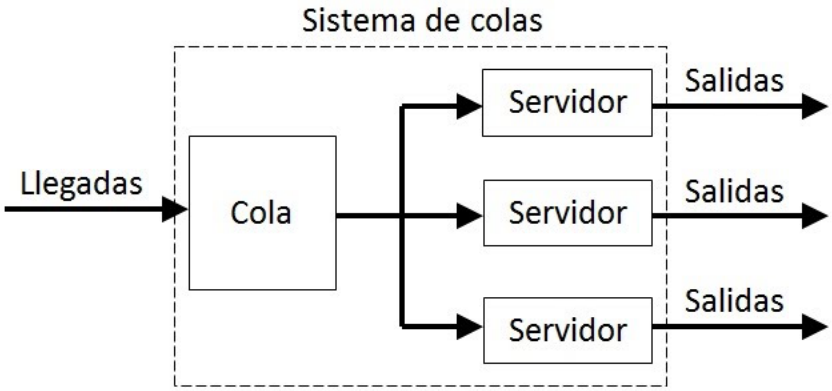


Figura 4.1. Mecanismo Multi-Canal con una única cola para n servidores. Fuente: White Hat Developer

En este momento y explicado el mecanismo de servicio que tiene todo el conjunto de colas del servicio de *urgencias de un hospital* en general, se procede a explicar el código del tercer bloque que está compuesto por UCI, críticos y quirófano.

Anteriormente, el momento de llegada del paciente j en cualquier nodo únicamente era consultar en la variable "Momento_Llegada". En este apartado, el asunto cambia, debido a que ahora el momento de llegada a los 3 nodos será el instante de terminación del paso anterior. Es por eso por lo que se realiza un bucle "for" y varios bucles "if". En este punto, se identifica cuál es su eje anterior, su momento de terminación y, además, se crea una nueva variable "Obs_Nodo_Split" que indica si un paciente ha sido asignado de consulta a observación. Se realiza esta diferenciación porque los horizontes temporales de los pasos son diferentes, es decir, no es lo mismo que un sujeto llega a UCI desde exploración que de observación, ya que si viene de este último habrá que hacer una diferencia para saber si el individuo viene de triaje o consulta.

Calculados los respectivos momentos de llegada, se realizan tres bucles "for", uno para cada nodo de este bloque. El rango de estos será desde 1 hasta el número máximo de pacientes hayan llegado. Posteriormente, se aplica el algoritmo creado para simular un sistema de colas de un mecanismo multicanal con tantos servidores como se hayan establecido al principio del *simulador*.

Una vez los individuos que hayan sido servidos en UCI o críticos se dirigirán obligatoriamente a planta. Pero aquellas personas que han sido atendidas en el quirófano podrán ir a planta, pero también a UCI o críticos mediante una asignación aleatoria con probabilidades 0.5;0.25;0.25, respectivamente. Por esta razón, se calculan de nuevo los momentos de llegada a UCI y críticos mediante la utilización de la función "which" de *Rstudio*.

Para terminar con la explicación del algoritmo matemático encargado de simular un sistema de *urgencias hospitalarias* se encuentra el cuarto bloque compuesto por alta y planta. Como uno de los objetivos principales de este proyecto es saber la *saturación* del último nodo, no se ha calculado el momento de llegada a alta. Únicamente, este eje sirve para saber cuántos pacientes han salido del sistema, es por eso, que se ha realizado un contador para saber cuántas personas se les ha asignado el alta.

Para calcular los momentos de llegada a planta, se ha realizado el siguiente código. Se aplica un bucle "for" de rango 1 hasta el número máximo de clientes para poder averiguar el ID y instante de terminación de aquellos usuarios que han sido atendidos en observación. Esto se realiza mediante un bucle "if" y con las funciones "as.numeric" y "which" integradas en el software utilizado. De igual forma se hace para los pacientes que llegan al nodo planta desde el quirófano.

A continuación, se crea una matriz que tendrá tantas filas como pacientes vayan a planta y 5 columnas. La primera columna será el ID del individuo. La segunda indica el nodo anterior y la tercera el momento de llegada. Para la siguiente columna se ha aplicado una función llamada "Split", esta se encarga de dividir los sujetos que hayan pasado por observación viniendo de consulta. Como ya se ha explicado anteriormente, el horizonte temporal no es el mismo si se llega a planta desde UCI o desde observación habiendo pasado por consulta. Por lo tanto, en esta columna aparecerá el nombre de "NADA" si el

paciente j no ha estado ni en consulta ni observación, "NINGUNO" si solamente ha pasado por observación y "CONSULTA" si ha pasado por la consulta, posteriormente a observación y finalmente a planta pudiendo haber pasado por UCI, críticos o quirófano.

Para realizar el sistema de colas en planta se aplica el mismo método que en los nodos anteriores. Se dispone de n servidores, que en este caso, serán camas y, mientras que la totalidad de estas no estén ocupadas, no habrá cola y, por ende, el tiempo de espera en cola será 0. Los períodos de servicio para el actual nodo seguirán una de las 5 distribuciones explicadas en el capítulo 1, pero eso sí, asumiendo que estos serán mucho más altos que en los 8 nodos restantes del sistema.

En la programación del sistema de colas de la planta, existe una diferencia respecto a los otros. Como uno de los objetivos de este proyecto es averiguar la *saturación* de esta, se ha realizado el siguiente código.

Nada más crear de forma aleatoria los tiempos de servicio, se aplica una serie de bucles "if" para poder averiguar cuantos servidores/camas están ocupados. Para este cálculo es necesario declarar una nueva variable que en el momento inicial tendrá el mismo valor que el número de servidores disponibles en los días simulados, que se introducen al principio del *simulador* mediante un input.

Si en el instante $t \geq 1$ el contador de pacientes en planta es igual a 1, el número de camas disponibles será igual a el número total de servidores disponibles en el instante inicial - 1. Cuando dicho contador es inferior al número de camas, los servidores disponibles serán la resta entre el número de servidores disponible en el anterior paciente -1.

Por lo contrario, si al llegar un paciente, estos están ocupados, los servidores disponibles serán 0, y nada más terminar un individuo su respectivo servicio, este será inmediatamente ocupado y, por tanto, habrá nuevamente 0 camas disponibles. Además, se ha tenido en cuenta en la programación que, si un sujeto sale y no existe ningún individuo esperando en cola, la disponibilidad de las camas será la suma entre el número de servidores disponibles en el instante anterior +1.

El código de programación de este *simulador* se encuentra en el Anexo del proyecto pudiendo acceder cualquier persona a él. Mediante el algoritmo explicado en los anteriores párrafos se consigue una simulación de un sistema de *urgencias de un hospital* en general, aplicando la *teoría de colas* para así conseguir los "targets" propuestos.

4.3 Cálculos

Posteriormente a la simulación del sistema, se realiza el siguiente código tanto para los cálculos necesarios como para la interfaz de Usuario, obteniendo una clara visión de la situación en urgencias.

4.3.1 Medidas de eficacia

En primer lugar, se realizan los cálculos de los parámetros esenciales de la *teoría de colas.* En este momento, no habrá ningún tipo de bucle, ya que los resultados obtenidos en la simulación de cada uno de los nodos se han guardado en un "data.frame". Cabe destacar que para cada una de estas variables habrá 11 registros, además del dato general de este.

Para el tiempo medio de llegadas, también conocido como λ, se calcula el número de pacientes en cada uno de los nodos mediante la función "length" en *Rstudio* y, a continuación se divide por el número total de minutos que se simulan. Se ha de matizar que un día son 1440 minutos, por lo que se multiplicará el número de días por este valor. Por ejemplo, para calcular el λ de UCI, se contabiliza el número de individuos que llegan a él y se divide por t, donde t será 1440*N.º de días. En consecuencia, se obtiene el número promedio de sujetos que llegan por minuto. Para obtener esta tasa en el sistema general, resulta más fácil, ya que este tiene únicamente un punto de entrada, por lo que se calcula el de λ del triaje.

Por el otro lado, el segundo parámetro más importante en la *teoría de cola* es μ o tiempo medio de servicio. En este caso, basta con calcular la media de los tiempos de servicio en cada nodo del sistema. Y, para el sistema en general, hacer la media de las medias, es decir, calcular el promedio de todos los μ.

El tiempo medio en cola aporta mucha información sobre la saturación de los 9 nodos y del sistema conjuntamente. Por tanto, para poder obtener los resultados es necesario calcular la media de los tiempos de espera en cola. Y, para el sistema en general, será hacer la media de las medias. Esto se obtiene mediante la función “mean”.

Otro parámetro importante para la correcta interpretación del dimensionamiento y la *saturación* de urgencias son los números de servidores disponibles en el día de inicialización de la simulación. En este caso, no habrá que hacer ningún calculo previo ya que los valores de esta variable los introduce el propio usuario en la interfaz básica.

Seguidamente, se encuentra el parámetro que nos indica el nivel de *saturación*, factor de utilización. Para su cálculo es necesario haber obtenido anteriormente los valores de λ, μ y el número de servidores. En todos los nodos, incluido el sistema en conjunto, su procedimiento de cálculo es el siguiente: $F.U = \frac{\lambda}{nº\ servidores * \mu}$. Por tanto, se puede decir que un sistema no está congestionando si $\rho(F.U) < 1$.

La computación del tiempo promedio de duración en los 9 nodos se realiza mediante la media de la diferencia entre el momento de terminación y el instante de llegada de cada lugar del hospital. Para el sistema en general las deducciones resultan más complejas. Como existen dos puntos de salida del sistema, alta y planta, habrá calcular por separado las diferencias de los tiempos. En cuanto a alta, únicamente habrá que calcular la media de los momentos de llegada, ya que cuando se va a este nodo, no existe ningún tipo de servicio. Por el lado de planta, se tiene que calcular la media de la diferencia entre el momento de terminación y el instante de llegada. Posteriormente, se realiza la promedio de las dos medias mencionadas y se obtendrá el tiempo medio de duración en el hospital.

En cuanto al número medio de pacientes en cola, se obtiene realizando la multiplicación entre λ y μ. Esta operación matemática indicará cuantos individuos esperan ser atendidos por minuto.

El último parámetro que se calcula referente a la *teoría de colas* es el número medio de servidores ocupados. Se realiza mediante la multiplicación entre ρ y la cuantía de servidores en cada eje. Por tanto, a mayor valor de este, mayor será la *saturación* de los lugares en el hospital.

4.3.2 Interfaz de usuario básica

Nada más terminar con los respectivos cálculos, se procede a realizar la interfaz básica para poder efectuar las conclusiones. Pero, como todo proceso matemático, se necesita una adecuación de los datos. Para ello se crean "data.frame" que engloban las computaciones de los parámetros anteriores y las trazabilidades de los pacientes.

Como existen 5 posibles pasos de los individuos en el horizonte temporal, la explicación de la creación de las trazabilidades se divide en el mismo número de movimientos.

El primer paso está dotado de un bucle "for" que tendrá tantos índices como clientes haya en el sistema. Dentro de este se encuentran bucles "if" para poder diferenciar donde se dirigen en el primer paso. Una vez identificados los ejes, se procede a consultar los ID, momento de llegada, cuando termina y cuáles son los tiempos de servicio de cada uno de los individuos implementado funciones como "which" y "lenght". Además, se calculan los porcentajes de pacientes que irán a consulta, exploración, observación o RCP viniendo de triaje. Estos datos consultados se guardan en una única variable para, posteriormente, crear un conjunto de datos que contenga toda la información de las personas.

Cuando se habla del paso 2, se refiere a las transiciones que realizan los pacientes del bloque 1 al 2, como se observar en la figura 2.1. En esta parte, el código utilizado es extenso, pero tienen muchas similitudes entre los diferentes itinerarios. Por esta razón, se va a explicar la programación de uno de ellos.

Como todo algoritmo, se declaran las variables a utilizar, que en este caso se refieren a la interacción entre consulta y alta. A continuación, se aplica un bucle "for" encargado de realizar un recorrido por los índices, que tendrán una longitud igual al número de personas que realizan esta ruta. Por consiguiente, se realiza un bucle "if" que compara si los pacientes que han llegado a alta vienen de consulta. Si este se cumple, se

realiza un contador para el dimensionamiento de las nuevas variables comentados anteriormente y se consulta el ID y el momento de llegada. Posteriormente y, de la misma manera que en el paso 1, se calcula el porcentaje de individuos que llegan a alta habiendo pasado por consulta. Para estos valores, se utilizan las funciones "order" y "with".

El mismo formato se aplica para realizar las consultas y cálculos del tercer paso. Pero en este caso, cuando ya se tiene la información de las rutas de los pacientes, se crea un nuevo algoritmo para poder anexar esta información a los datos del primer y segundo paso. Este código se encarga de comprobar el dimensionamiento de las interacciones anteriores y adecuarlo a los resultados del tercer paso. En cuanto a la programación, se han empleado varios bucles "for" y "if", además de funciones como "min", "max", "length", "sum", entre otras muchas. Para el correcto funcionamiento, se ha de tener especial atención a la hora de dimensionar las diferentes variables y matrices e ir jugando con los índices. Esto último, se refiere a ir restando o sumando 1 en los respectivos índices. Además, se ha tenido en cuenta que, si por alguna extraña razón no hay ningún tráfico de personas en este paso, el programa lo identifique y lo manifieste en la interfaz gráfica.

Tanto el paso 4 como el 5, utilizan los algoritmos descritos en los tres pasos anteriores, por lo que se utilizaran el mismo número de variables, eso sí, con distintos nombres, y las mismas funciones que proporciona *Rstudio*.

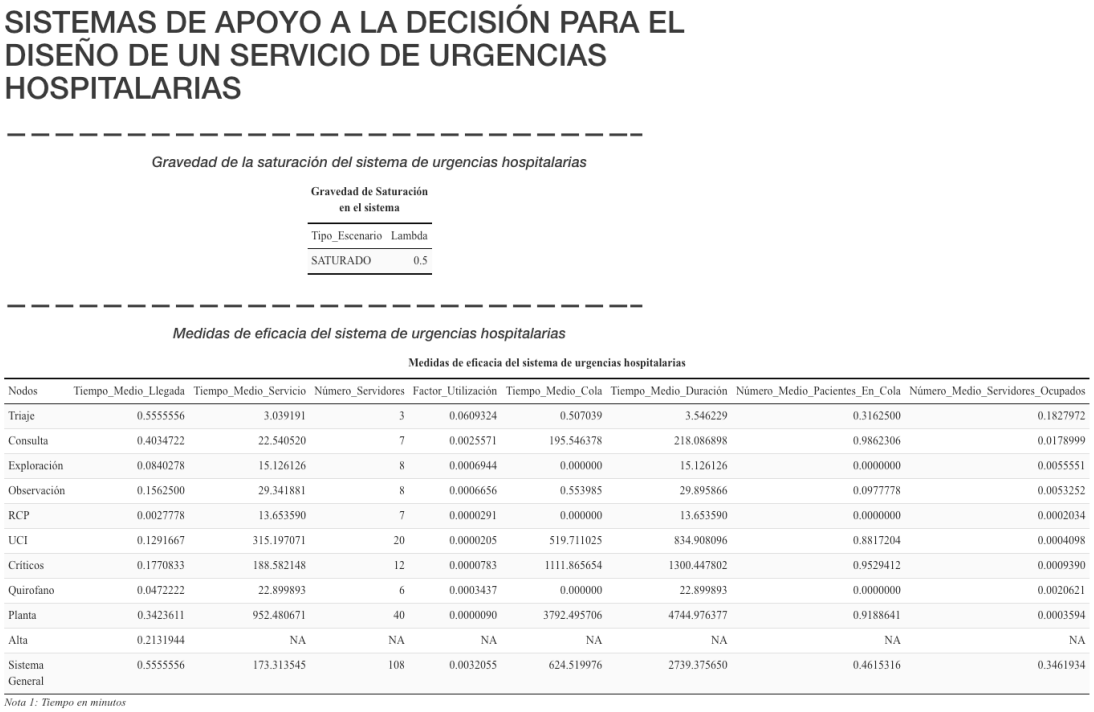
Una vez, explicado cómo se ha realizado el algoritmo de las trazabilidades de los pacientes en el sistema, se procede a la explicación de la interfaz de usuario.

Para la correcta implementación de esta interfaz se ha utilizado la herramienta Knitr[21] que proporciona Rmarkdown[22] en el software *Rstudio*. Esta función es capaz de crear un archivo HTML para poder visualizar todo tipo de gráficas y tablas, además de garantizar una interacción en estas últimas.

Nada más iniciar la aplicación, aparece una tabla, realizada mediante la librería "kable"[23], que proporciona la información del tipo de *escenario* que se ha simulado en el

[21] Consulta más información sobre Knitr
[22] Consulta más información sobre Rmarkdown
[23] Consulta más información sobre la librería Kable

servicio de urgencias. En esta aparece la nomenclatura del ambiente y el λ correspondiente a la tasa de llegadas de los pacientes. A continuación, se encuentra otra tabla, en la que se muestran los resultados de *teoría de colas* obtenidos en la simulación. Estas figuras se pueden observar en la siguiente ilustración.



SISTEMAS DE APOYO A LA DECISIÓN PARA EL DISEÑO DE UN SERVICIO DE URGENCIAS HOSPITALARIAS

Gravedad de la saturación del sistema de urgencias hospitalarias

Gravedad de Saturación en el sistema

Tipo_Escenario	Lambda
SATURADO	0.5

Medidas de eficacia del sistema de urgencias hospitalarias

Medidas de eficacia del sistema de urgencias hospitalarias

Nodos	Tiempo_Medio_Llegada	Tiempo_Medio_Servicio	Número_Servidores	Factor_Utilización	Tiempo_Medio_Cola	Tiempo_Medio_Duración	Número_Medio_Pacientes_En_Cola	Número_Medio_Servidores_Ocupados
Triaje	0.5555556	3.039191	3	0.0609324	0.507039	3.546229	0.3162500	0.1827972
Consulta	0.4034722	22.540520	7	0.0025571	195.546378	218.086898	0.9862306	0.0178999
Exploración	0.0840278	15.126126	8	0.0006944	0.000000	15.126126	0.0000000	0.0055551
Observación	0.1562500	29.341881	8	0.0006656	0.553985	29.895866	0.0977778	0.0053252
RCP	0.0027778	13.653590	7	0.0000291	0.000000	13.653590	0.0000000	0.0002034
UCI	0.1291667	315.197071	20	0.0000205	519.711025	834.908096	0.8817204	0.0004098
Críticos	0.1770833	188.582148	12	0.0000783	1111.865654	1300.447802	0.9529412	0.0009390
Quirofano	0.0472222	22.899893	6	0.0003437	0.000000	22.899893	0.0000000	0.0020621
Planta	0.3423611	952.480671	40	0.0000090	3792.495706	4744.976377	0.9188641	0.0003594
Alta	0.2131944	NA	NA	NA	NA	NA	NA	NA
Sistema General	0.5555556	173.313545	108	0.0032055	624.519976	2739.375650	0.4615316	0.3461934

Nota 1: Tiempo en minutos

Figura 4.2. Inicio de la interfaz. Fuente: Elaboración propia

Los resultados obtenidos tanto en las dos tablas de la figura 4.2 como en las restantes, se explicarán en el capítulo 5 de este proyecto.

Esta interfaz también está dotada de cuatro gráficos que muestran las trazabilidades de los pacientes (figura 4.3). Todos estos están realizados mediante las librerías "ggplot" y "plotly"[24]. En estas figuras se utilizan funciones para el diseño, títulos, leyendas, colores, nombre de los ejes, etc. Además, se ha establecido la función "writeXLS" que da la posibilidad de descarga de los datos de la simulación en formato XLSX (Excel). Finalmente, gracias a todos los conjuntos de datos creados en el algoritmo, se puede realizar consultas de todos los pacientes que entrar en el sistema de *urgencias hospitalarias*.

[24] Consulta más información sobre las librerías ggplot y plotly

Información general del sistema por cada pacientes

Show 10 entries Search:

Información de cada uno de los pacientes en el Sistema de urgencias hospitalarias

	Personas	Momento_Llegada_al_Sistema	Cuando_Termina_el_Triaje	Donde_Llega_Despues_Triaje	Cuando_Llega	Cuando_Termina	Donde_Llega_Segundo_Paso	Cuando_Termina_Segundo_Paso
1	1	0	1.47764734830707	CONSULTA	1.47764734830707	25.1340927942656	ALTA	25.1340927942656
2	2	2	3.49287380278111	CONSULTA	3.49287380278111	25.3705827454105	UCI	322.705072698183
3	3	5	6.10346333589405	CONSULTA	6.10346333589405	24.4812880665995	CRÍTICOS	206.728596275207
4	4	6	8.02220940962434	CONSULTA	8.02220940962434	33.6762711431365	ALTA	33.6762711431365
5	5	6	7.62701119109988	CONSULTA	7.62701119109988	31.5079808612354	ALTA	31.5079808612354
6	6	8	12.76723060105	EXPLORACIÓN	12.76723060105	27.8706042682752	UCI	182.334724141285
7	7	9	13.930577205494	CONSULTA	13.930577205494	42.4632815148216	CRÍTICOS	208.093751111766
8	8	12	15.6188049316406	CONSULTA	15.6188049316406	42.4769318858162	ALTA	42.4769318858162
9	9	12	15.1740766428411	CONSULTA	15.1740766428411	41.1792239628267	ALTA	41.1792239628267
10	10	14	15.0454703718424	CONSULTA	15.0454703718424	47.7341632780153	ALTA	47.7341632780153

Showing 1 to 10 of 800 entries Previous 1 2 3 4 5 … 80 Next

Figura 4. 3. Realización de consultas. Fuente: Elaboración propia

CAPÍTULO 5

Análisis de simulación.

En el presente capítulo se presentan los resultados obtenidos mediante el algoritmo expuesto en los apartados anteriores, además de, realizar una comparación de los niveles de *saturación* del hospital creando distintos *escenarios.*

Con la comparación de los resultados que se observarán en la interfaz de usuario básica proporcionada, se pretende conseguir optimizar el dimensionamiento y controlar el grado de *saturación* tanto de los nodos como del sistema en su totalidad.

5.1 Escenarios simulados

Este análisis está compuesto por la realización de 3 *escenarios* de *saturación*: Bajo, Medio y Saturado, en los cuales habrá cambios en las distribuciones probabilísticas de la tasa de servicio y en el número de servidores disponibles, para ver cómo afectan estos a la *saturación* y dimensionamiento del servicio de *urgencias de un hospital*. Sin embargo, en los 9 *escenarios* totales, el número de días a simular siempre será el mismo, 1. Para la correcta comparación de los resultados de los 3, se utilizarán las mismas distribuciones y cantidad de servidores en cuanto a los subescenarios se refiere.

El primer *escenario* a simular será aquel en la que las tasas de llegadas de los pacientes siguen una distribución Poisson con $\lambda = 0.1$ o lo que es lo mismo una tasa Baja. Una vez estos períodos han sido configurados, se procede a realizar 3 subescenarios que hacen referencia a los tiempos de servicio y al número de servidores en los 9 nodos del sistema.

El subescenario inicial es aquel en el que los períodos de servicio siguen una distribución probabilística uniforme. Esta distribución se caracteriza por poseer valores que se encuentran dentro de un intervalo. Los rangos que puede tomar la variable tiempo de servicio, en minutos, en cada nodo del sistema son los siguientes: Triaje= [1,5][25];

[25] Número de recepcionistas en los hospitales españoles

Consulta= [15,30][26];Exploración= [10,20][27]; Observación= [15,45][28]; RCP= [10,20]; UCI=[1900,5760];Críticos=[1440,4320];Quirófano=[15,30];Planta=[1440,10080]. Por otra parte, se encuentran el número de servidores, que pueden ser personal médico en el caso de triaje o camas en los nodos restantes. Por tanto, los valores para los servidores son 3; 7; 8;8;12;7;20;12;6 y 40, respectivamente[29]. El número de camas en los hospitales varía en función de cuantos habitantes tenga la ciudad donde se ubique, cuanto de grande es y el régimen político de la comunidad autónoma. Es por eso, que las cifras de este subescenario provienen de la media de los hospitales españoles.

A continuación, se procede a explicar la segunda circunstancia. En esta, los únicos parámetros que cambian respecto al anterior escenario son las tasas de servicio, que ahora adoptaran distribuciones probabilísticas distintas dependiendo del bloque del sistema hospitalario, como se mencionó en el capítulo 2 de este mismo documento.

El bloque 1 tendrá una distribución uniforme con unos valores mínimos de 1 minuto y unos máximos de 5. El bloque 2, formado por consulta, exploración, observación y RCP, estarán distribuidas mediante una triangular. Esta distribución se caracteriza por necesitar la introducción de 2 parámetros: valor mínimo y valor máximo. Por lo tanto, los valores mínimos y máximos de los 4 nodos serán los mismo que en la distribución uniforme del primer subescenario. La tercera parte tendrá una distribución log-normal, con una media y desviación típica para cada nodo: UCI con una media de 2430 minutos con una desviación típica de 300; críticos tiene un promedio de 1990 minutos y una desviación estándar de 400 y, por último, quirófano con Media=20 y Desv. Tip. =5. El último bloque, planta, tendrá una distribución Beta ($\beta = 0.5$) multiplicado por el tiempo máximo de servicio correspondiente a la distribución uniforme.

Aplicando las mismas distribuciones anteriores y cambiando el número de servidores de los 9 nodos, se obtiene el tercer subescenario cuando la tasa de llegadas de los pacientes al sistema sigue una distribución Poisson de $\lambda = 0.1$. Ahora la cantidad de camas/personal médico será el siguiente: Triaje=2, Consulta=6, Exploración=5, Observación=10, RCP=3, UCI=10, Críticos=16, Quirófano=4 y Planta=60.

[26] Número de servidores en consulta en un hospital en general
[27] Número de camas en exploración en un hospital en general
[28] Número de camas en observación en un hospital en general
[29] Información del número de servidores en los hospitales españoles

Una vez visto el *escenario* Bajo, se procede a realizar la explicación del Medio. Este se diferencia respecto al primero únicamente con la tasa de llegadas de los pacientes, que ahora tendrá un $\lambda = 0.25$. Los 3 subescenarios de dicho apartado estarán compuestos por los mismos números de servidores y las mismas distribuciones probabilísticas con sus respectivos parámetros. No se cambian los valores de las variables ya que se quieren realizar comparaciones de los resultados obtenidos, y para ver las similitudes o discrepancias, se deben tener los mismos parámetros y distribuciones.

Por último, el *escenario* 3 está compuesto por un $\lambda = 0.5$ y tres subescenarios. Por tanto, la única diferencia existente entre los tres principales escenarios simulados son los valores del parámetro mencionado anteriormente. En este se podrá observar el comportamiento del sistema cuando llegan en un día la escalofriante cifra de 800 pacientes.

Como es lógico, existen miles de combinaciones posibles para los 3 *escenarios*. Incluso, se pueden crear n escenarios, únicamente, cambiando el valor λ de la tasa de llegadas del sistema. Este proyecto tiene la finalidad de averiguar mediante un *simulador* cuales son los períodos medios en cola, tiempos de servicio, entre otros objetivos. Es por eso por lo que cabe destacar que con esta simulación se podrían realizar múltiples análisis estadísticos y de optimización, pero no es la finalidad del presente estudio. Un futuro proyecto podría ser la creación de un algoritmo matemático capaz de optimizar el coste que supone tener x servidores en el sistema, además, de indicar la estructura ideal para los desplazamientos de los pacientes.

5.2 Resultados

En el presente apartado se exponen los diferentes resultados obtenidos gracias a la simulación de 3 *escenarios* dotados de 3 subapartados. Además, se comparan para ver las discrepancias entre ellos y, así, ver de una forma más clara las diferentes saturaciones del sistema de *urgencias hospitalarias*. Por último, se exponen las medidas de eficacia relacionados con una simulación de un mes entero, para ver la *saturación* de este.

Las figuras que se exponen y razonan a continuación, se pueden obtener en la interfaz de usuario proporcionada. Además, en la aplicación existe la posibilidad de interacción en las tablas y gráficos realizando consultas.

Para la correcta interpretación de los distintos subescenarios que se encuentran dentro de los 3 apartados simulados, mediante el cambio de la tasa de llegadas, se procede a identificarlos mediante una serie de colores. Los subapartados 1, el color de fondo será el blanco. En el caso de los segundos, será el ivory. Y, por último, el color identificativo será el azure.

En primer lugar, se estudia el *escenario* 1, saturación Baja, que tendrá la peculiaridad de poseer las mismas distribuciones probabilísticas en todos los nodos.

Medidas de eficacia del sistema de urgencias hospitalarias

Nodos	Tiempo_Medio_Llegada	Tiempo_Medio_Servicio	Número_Servidores	Factor_Utilización	Tiempo_Medio_Cola	Tiempo_Medio_Duración	Número_Medio_Pacientes_En_Cola	Número_Medio_Servidores_Ocupados
Triaje	0.1291667	2.972144	3	0.0144864	0.0000000	2.972144	0.0000000	0.0434591
Consulta	0.1187500	22.162452	7	0.0007655	0.1097351	22.272187	0.0175439	0.0053582
Exploración	0.0027778	14.517010	8	0.0000239	0.0000000	14.517010	0.0000000	0.0001913
Observación	0.0159722	30.487094	8	0.0000655	0.0000000	30.487094	0.0000000	0.0005239
RCP	0.0020833	16.499181	7	0.0000180	0.0000000	16.499181	0.0000000	0.0001263
UCI	0.0034722	3696.547900	20	0.0000000	0.0000000	3696.547900	0.0000000	0.0000009
Críticos	0.0062500	3061.225606	12	0.0000002	0.0000000	3061.225606	0.0000000	0.0000020
Quirofano	0.0041667	25.066942	6	0.0000277	0.0000000	25.066942	0.0000000	0.0001662
Planta	0.0138889	5391.575945	40	0.0000001	0.0000000	5391.575945	0.0000000	0.0000026
Alta	0.1159722	NA	NA	NA	NA	NA	NA	NA
Sistema General	0.1291667	1362.009125	108	0.0000948	0.0121928	3037.576430	0.0019493	0.0102422

Nota 1: Tiempo en minutos

Tabla 5.1. Medidas de eficacia Escenario 1, Subapartado 1: Fuente: Elaboración propia

Como se puede observar en la tabla anterior, el nivel de *saturación* del sistema en general es escaso. Empezando desde el principio, los tiempos medios de llegadas de los pacientes a cada uno de los nodos son muy bajos. El caso donde este valor es el más elevado es en la entrada del sistema, triaje, que por cada minuto que pasa entra 0.13 individuos. Tanto los tiempos medios de servicio como el número de servidores, son resultados que se extraen de forma sencilla, ya que se conocen estas distribuciones y la cuantía de servidores existentes.

Además, se extrae la conclusión de que no existe cola en ningún nodo salvo en consulta con una media de 0.11 minutos. Por ende, se concluye que el número promedio de pacientes en cola no supera en ningún caso 0.05 clientes por minuto.

Cualquier persona que entre al sistema, de media, su servicio en triaje será de alrededor de 3 minutos. Por el contrario, y lo lógico, el tiempo medio que duración del paciente j en planta es de 5391 minutos. Es por esa situación, que el número medio de servidores ocupados en el horizonte temporal simulado, en este caso un día, no supera el 0.05 por minuto.

Para el sistema en su totalidad, el período promedio en cola es de 0.012 minutos con un promedio de 0.019 clientes. Estos valores, son contrastados y afirmados, con el promedio de servidores ocupados, ya que es bajo y reafirma juntamente con $\rho < 0.01$ que el sistema no está congestionado.

Citada la congestión, se presente un gráfico que muestra el nivel de *saturación* en planta durante 24 horas.

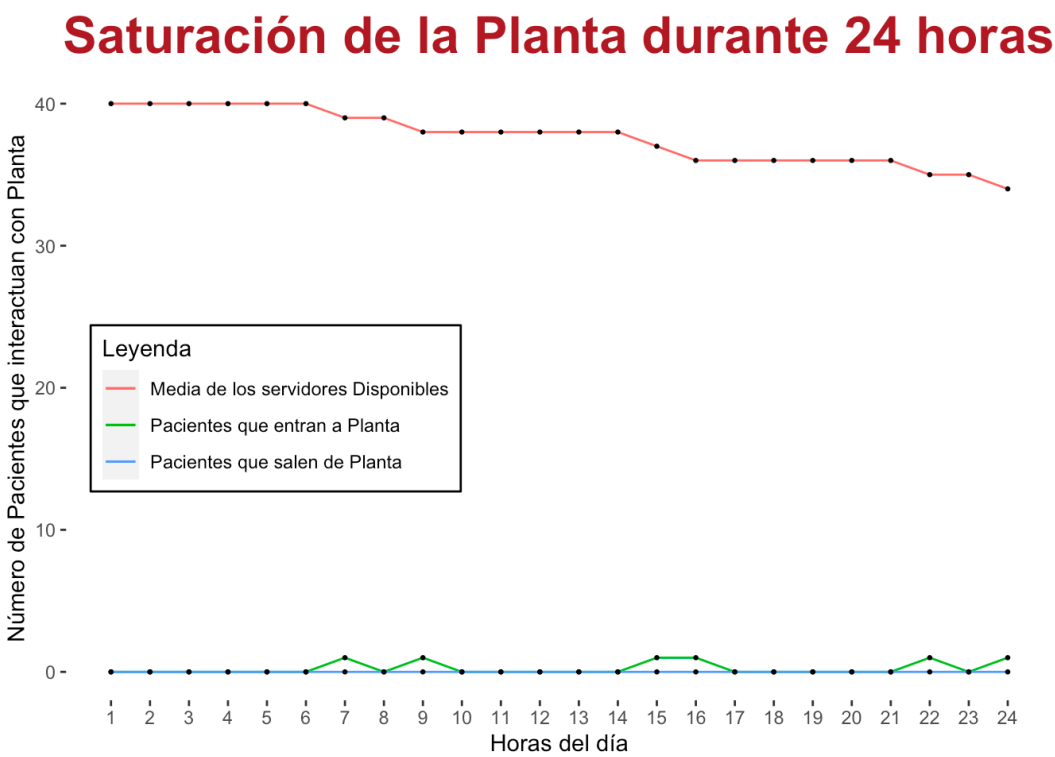


Figura 5.1. Saturación en planta Escenario 1.1. Fuente: Elaboración propia

Mediante la interpretación de la figura 5.1, se puede extraer que el sistema de la planta no se congestiona. Pasadas las 6 horas de inicialización de la simulación el número medio de servidores disponibles empieza a decaer llegando al final de la jornada a unas 35 camas disponibles en promedio.

Además, se observa una pequeña fluctuación en el número de pacientes que entran y pero no los que salen del nodo. Estas oscilaciones resultan interesantes porque se puede observar que cuando el número de pacientes que entran al sistema es mayor al que salen, los servidores disponibles disminuyen. De la misma forma, cuando se alcanza el mayor pico de entradas, se registra la menor salida de individuos. Estas en el modelo de *saturación* Bajo son 0, ya que los pacientes están más de un día en planta.

Estas evasiones ocurren hasta el momento temporal de 24 horas, debido a que se ha simulado únicamente un día y se puede dar el caso, que un paciente entre y salga el mismo día. Del mismo modo, ocurre con el número medio de servidores disponibles y pacientes que entran al sistema, que "terminan" su función pasadas las 30 horas desde la inicialización.

Otro gráfico que exhibe el nivel de *saturación*, pero en este caso de la UCI, es el diagrama de la figura 5.2

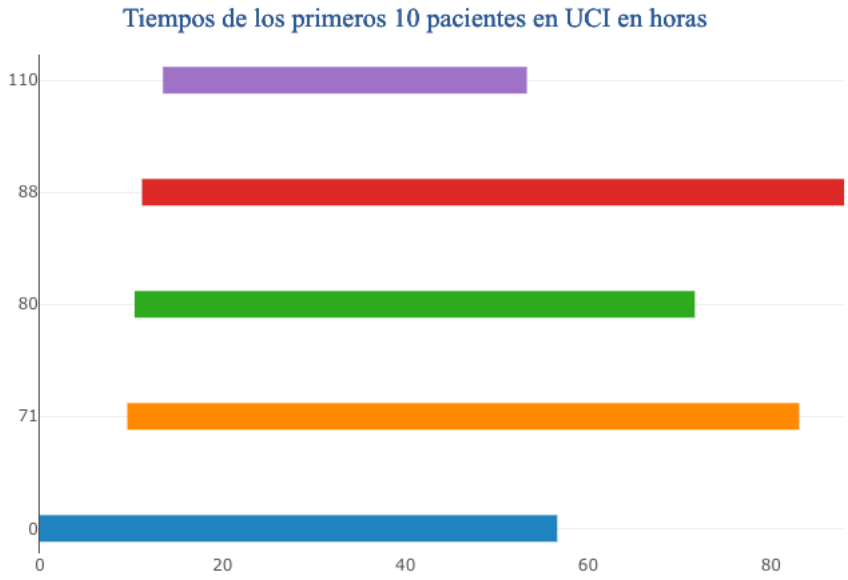


Figura 5.2. Duración de los 10 primeros pacientes en UCI Escenario 1.1. Fuente: Elaboración propia

En este caso, queda ilustrado el tiempo que está esperando a ser atendido, el tiempo de servicio y el momento que termina el paciente j. Este gráfico es muy útil para analizar el trabajo de los médicos que realizan las operaciones y para saber si el dimensionamiento de la UCI es el correcto, ya que si existen un gran número de individuos que esperan, significa que habrá que remodelar este servicio.

Como se observa en la figura 5.2, el paciente con ID 80, es decir, el 80 individuo que llega al sistema en un día, está siendo operado durante casi 6 horas. Haciendo la diferencia entre el momento que termina y el tiempo de servicio, se puede obtener el instante en el que es dirigido a este nodo. En este caso, el paciente categorizado como 80, entró al sistema pasadas las 10 horas desde la apertura del *servicio de urgencias*. Además, como se ha simulado un *escenario* de *saturación* bajo, el número máximo de pacientes que entran a UCI son 5. La interfaz de usuario proporciona el mismo título para todos los supuestos, es por eso por lo que en el encabezado aparece "primeros 10 pacientes".

Como se mencionó en el capítulo 3, el nodo más concurrido del servicio de urgencias propuesto, y en general, es la consulta. Por esta razón, se ha decido crear el siguiente gráfico.

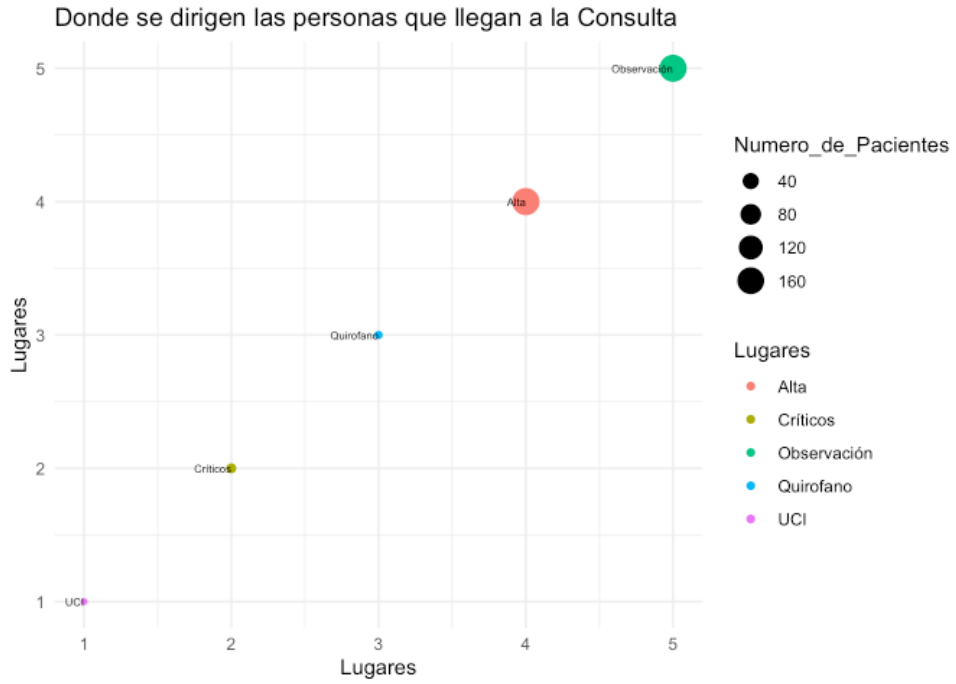


Figura 5.3. Rutas de los pacientes originados en Consulta para el Escenario 1.1. Fuente: Elaboración propia

El presente gráfico proporciona una visión clara de cuáles son los nodos que se dirigen los pacientes después de haber pasado por la consulta. El lugar donde los pacientes no se dirigen prácticamente es a quirófano junto a críticos y UCI. Sin embargo, resulta peculiar que el tamaño de los círculos correspondientes a observación y alta, sean del mismo tamaño. Hay que matizar que, un tamaño grande del círculo significa que el

número de pacientes que realizan esa ruta es elevado. Además, la disposición de estos en la gráfica no implica absolutamente nada en el estudio, simplemente, es diseño.

Del mismo modo que en el capítulo 2, se expuso la estructura/esquema del sistema de urgencias (Figura 2.1), a continuación, se ha realizado un grafo del sistema de urgencias.

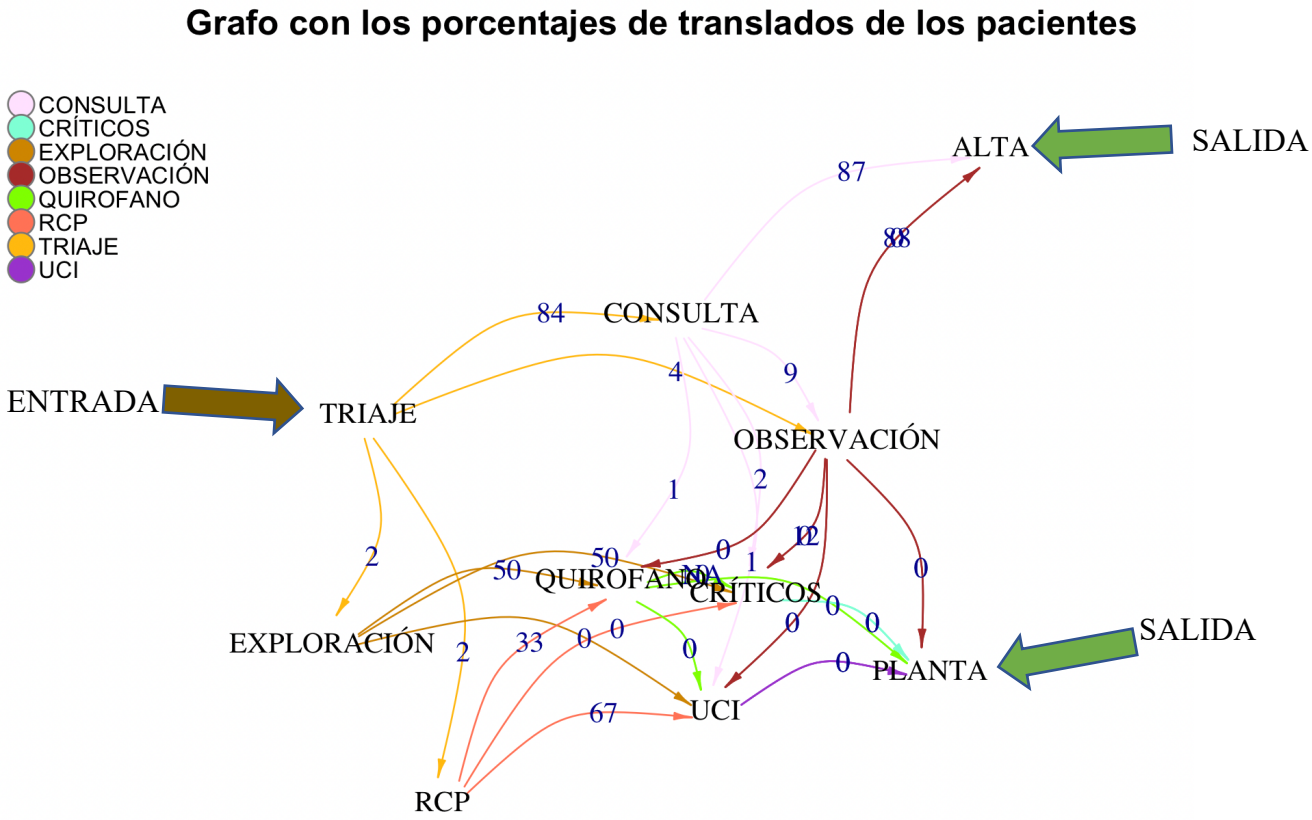


Figura 5.4. Grafo del sistema de urgencias hospitalarias Escenario 1.1. Fuente: Elaboración propia

Como se observa en la Figura 5.4, se han calculado los porcentajes de pacientes que se dirigen de un nodo a otro. La flecha de color marrón, indica el único punto de entrada al sistema. Por el otro lado, las dos flechas verdes recalcan las salidas de urgencias. Un ejemplo de la interpretación del presente gráfico seria que el 84% de los pacientes que llegan al sistema de *urgencias hospitalarias* con un nivel de *saturación* Bajo acaban siendo derivados a consulta, el 4% a observación, el 2% a RCP y también el 2% de los individuos a exploración.

Para el segundo subescenario de este apartado, se van a utilizar una serie de distribuciones probabilísticas distintas para los bloques que componen el sistema como

se ha explicado en el capítulo 5.1. En este caso, se expondrán las medidas de eficacia y la *saturación* en UCI.

Las cifras obtenidas gracias a la aplicación de la *teoría de colas* en esta simulación se encuentran en la siguiente tabla.

Medidas de eficacia del sistema de urgencias hospitalarias

Nodos	Tiempo_Medio_Llegada	Tiempo_Medio_Servicio	Número_Servidores	Factor_Utilización	Tiempo_Medio_Cola	Tiempo_Medio_Duración	Número_Medio_Pacientes_En_Cola	Número_Medio_Servidores_Ocupados
Triaje	0.1291667	2.972144	3	0.0144864	0.0000000	2.972144	0.0000000	0.0434591
Consulta	0.1187500	22.178707	7	0.0007649	0.1015534	22.280261	0.0175439	0.0053542
Exploración	0.0027778	14.563352	8	0.0000238	0.0000000	14.563352	0.0000000	0.0001907
Observación	0.0159722	30.065505	8	0.0000664	0.0000000	30.065505	0.0000000	0.0005312
RCP	0.0020833	15.856924	7	0.0000188	0.0000000	15.856924	0.0000000	0.0001314
UCI	0.0034722	2532.985239	20	0.0000001	0.0000000	2532.985239	0.0000000	0.0000014
Críticos	0.0062500	2005.778416	12	0.0000003	0.0000000	2005.778416	0.0000000	0.0000031
Quirofano	0.0041667	20.353811	6	0.0000341	0.0000000	20.353811	0.0000000	0.0002047
Planta	0.0131944	4987.087755	40	0.0000001	0.0000000	4987.087755	0.0000000	0.0000026
Alta	0.1159722	NA	NA	NA	NA	NA	NA	NA
Sistema General	0.1291667	1069.874412	108	0.0001207	0.0112837	2835.332335	0.0019493	0.0130389

Nota 1: Tiempo en minutos

Tabla 5.2. Medidas de eficacia para el Escenario 1, subapartado 2. Fuente: Elaboración propia

Cuando el sistema se simula mediante una *saturación* de llegadas Baja pero cambiando las distribuciones de servicio, los tiempos medios de asociados a este rondan el valor 1069 minutos. Tanto los períodos entre llegadas como el número de servidores, son exactamente los mismo que en el subescenario 1 del primer apartado explicado en este capítulo. Todos los factores de utilización (grado de congestión) tienen un valor inferior a 0.05, lo cual permite decir, que no existen indicios de llenado.

Los únicos tiempos medios en cola existentes son los respecticos a consulta, que si se comparan con el subescenario anterior, se han visto ligeramente disminuidos. En estos resultados, existen un dato muy peculiar ya que los períodos promedios de servicio en la consulta son prácticamente idénticos a los anteriores, por lo que la distribución probabilística aplicada en este subapartado proporciona unas cifras casi idénticas.

Para el sistema en general, los parámetros elegidos no proporcionan una congestión del servicio, pero si, unos tiempos de duración elevados, ya que de media el paciente estará dentro del sistema unos 2835 minutos. Sin embargo, el número promedio de sujetos que están esperando a ser atendidos en las distintas colas del sistema serán 0.001 por minuto, es decir, prácticamente nadie tendrá que hacer cola para ser atendido.

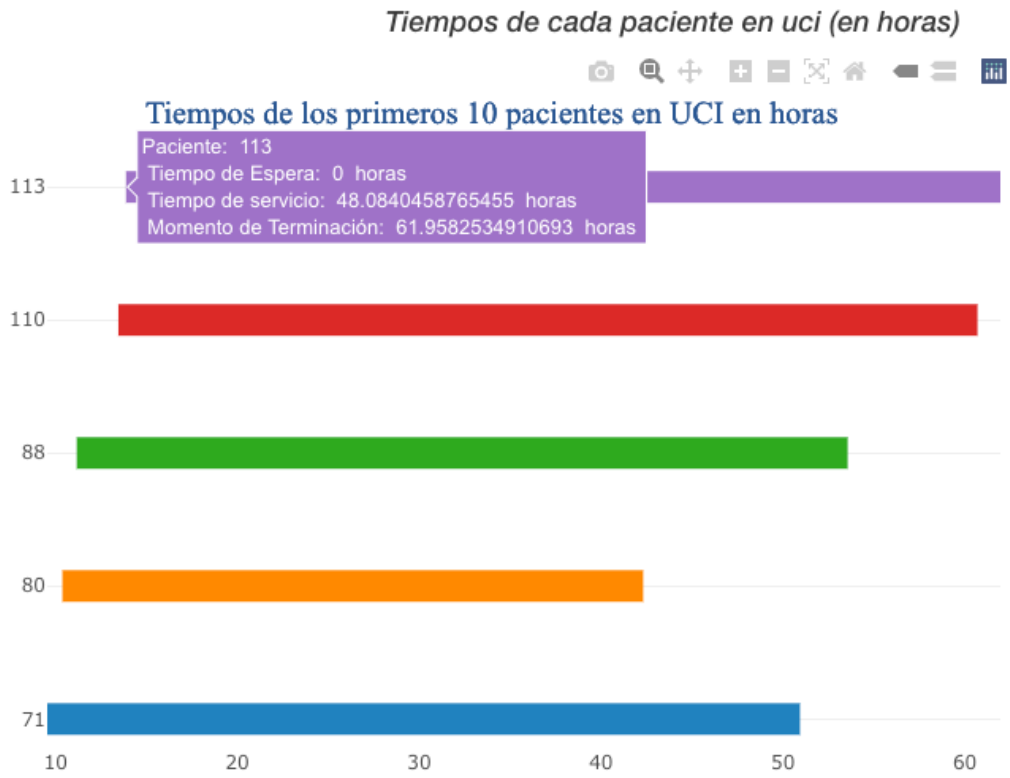


Figura 5.5. Tiempos de servicio en UCI escenario 1.2 Fuente: Elaboración propia

Como se puede observar en la Figura 5.5, los 5 primeros pacientes en entrar a la UCI lo hacen antes de las 9 horas desde haber inicializado el *simulador*. La pestaña que está abierta en dicha figura corresponde al paciente número 113, que espera 0 minutos en cola y es servido en aproximadamente 48 horas de operación. Por tanto, este cliente llegó a las 13 horas al nodo.

La longitud de las barras indica el tiempo de duración de los pacientes dentro de la UCI. Por lo tanto, a mayor longitud de estas, mayor será el tiempo que permanecerán en el eje. Cada individuo se identifica mediante un color y se ordenan en función al tiempo de llegada. Es por eso, que el primer sujeto en ser atendido es el 71 y posteriormente será el 80 y así sucesivamente.

A continuación se procede a la explicación del último subescenario del apartado 1. Este tiene la peculiaridad del cambio del número de servidores en cada nodo, es por eso por lo que se muestra la siguiente figura.

Medidas de eficacia del sistema de urgencias hospitalarias

Nodos	Tiempo_Medio_Llegada	Tiempo_Medio_Servicio	Número_Servidores	Factor_Utilización	Tiempo_Medio_Cola	Tiempo_Medio_Duración	Número_Medio_Pacientes_En_Cola	Número_Medio_Servidores_Ocupados
Triaje	0.1291667	2.972144	2	0.0217295	0.0296191	3.001763	0.0322581	0.0434591
Consulta	0.1187500	22.178707	6	0.0008924	0.1942060	22.372913	0.0350877	0.0053542
Exploración	0.0027778	14.563352	5	0.0000381	0.0000000	14.563352	0.0000000	0.0001907
Observación	0.0159722	30.065505	10	0.0000531	0.0000000	30.065505	0.0000000	0.0005312
RCP	0.0020833	15.856924	3	0.0000438	0.0000000	15.856924	0.0000000	0.0001314
UCI	0.0034722	2532.985239	10	0.0000001	0.0000000	2532.985239	0.0000000	0.0000014
Críticos	0.0062500	2005.778416	16	0.0000002	0.0000000	2005.778416	0.0000000	0.0000031
Quirofano	0.0041667	20.353811	4	0.0000512	0.0000000	20.353811	0.0000000	0.0002047
Planta	0.0131944	4987.087755	60	0.0000000	0.0000000	4987.087755	0.0000000	0.0000026
Alta	0.1159722	NA	NA	NA	NA	NA	NA	NA
Sistema General	0.1291667	1069.874412	114	0.0001207	0.0248694	2835.348830	0.0074829	0.0137633

Nota 1: Tiempo en minutos

Tabla 5.3. Medidas de eficacia para el Escenario 1, subapartado 3. Fuente: Elaboración propia.

En el caso de la baja tasa media de llegadas de los pacientes al sistema, con distribuciones probabilísticas distintas en los 4 bloques y el cambio de número de servidores disponibles en el día de la simulación, se obtienen prácticamente los mismos datos que en los dos supuestos anteriores.

En este caso, el único nodo repercutido ha sido triaje, ya que en los casos antiguos no existen cola alguna y en este el tiempo medio que ha de esperar un cliente en de 0.03 minutos, prácticamente 0.

Analizados los tres subescenarios del apartado 1, se procede a la explicación de los resultados obtenidos mediante una tasa de llegadas según una distribución Poisson con parámetro $\lambda = 0.25$. Dicho supuesto se ha llamado nivel de *saturación* Medio.

Medidas de eficacia del sistema de urgencias hospitalarias

Nodos	Tiempo_Medio_Llegada	Tiempo_Medio_Servicio	Número_Servidores	Factor_Utilización	Tiempo_Medio_Cola	Tiempo_Medio_Duración	Número_Medio_Pacientes_En_Cola	Número_Medio_Servidores_Ocupados
Triaje	0.3076389	2.968353	3	0.0345465	0.0525871	3.02094	0.0564334	0.1036396
Consulta	0.2159722	22.266440	7	0.0013856	1.1733366	23.43978	0.2218650	0.0096994
Exploración	0.0541667	14.902885	8	0.0004543	0.0000000	14.90288	0.0000000	0.0036346
Observación	0.0888889	29.067962	8	0.0003822	0.0000000	29.06796	0.0000000	0.0030580
RCP	0.0027778	15.428181	7	0.0000257	0.0000000	15.42818	0.0000000	0.0001800
UCI	0.0736111	318.646460	20	0.0000116	131.9070477	450.55351	0.6509434	0.0002310
Críticos	0.0937500	194.738192	12	0.0000401	364.1967652	558.93496	0.9111111	0.0004814
Quirofano	0.0284722	21.571207	6	0.0002200	0.0000000	21.57121	0.0000000	0.0013199
Planta	0.1909722	1005.027051	40	0.0000048	2142.0987254	3147.12578	0.8545455	0.0001900
Alta	0.1166667	NA	NA	NA	NA	NA	NA	NA
Sistema General	0.3076389	180.183153	108	0.0017074	293.2698291	1936.06464	0.2994331	0.1843957

Nota 1: Tiempo en minutos

Tabla 5.4. Medidas de eficacia para el Escenario 2, subapartado 1. Fuente: Elaboración propia.

Cuando el sistema se simula mediante una saturación de llegadas Media, los tiempos entre llegadas de los pacientes rondan el valor 0.3. Dicho valor significa que cada minuto que transcurre 0.3 son los nuevos pacientes que entran al sistema. Tanto los períodos medios de servicio como el número de servidores son exactamente los mismo que en el subescenario 1. Todos los factores de utilización (grado de congestión) tienen un valor inferior a 0.05, lo cual permite decir, que no existen indicios de llenado en los nodos.

Los tiempos medios en cola tanto de la planta como de críticos, se han visto incrementados. En este escenario, el período que pasará en cola un paciente si quiere ser atendido en planta y resulta que no existe disponibilidad de camas, será de 2142 minutos. En cuanto a críticos, este tiempo es menor, llegando a esperar de media 364 minutos.

Para el sistema en general, los parámetros elegidos no proporcionan una congestión del servicio, pero si, unos tiempos de espera elevados, ya que de media el paciente espera unos 290 minutos. Sin embargo, el número promedio de individuos que están esperando a ser atendidos en las distintas colas del sistema serán 0.29 por minuto.

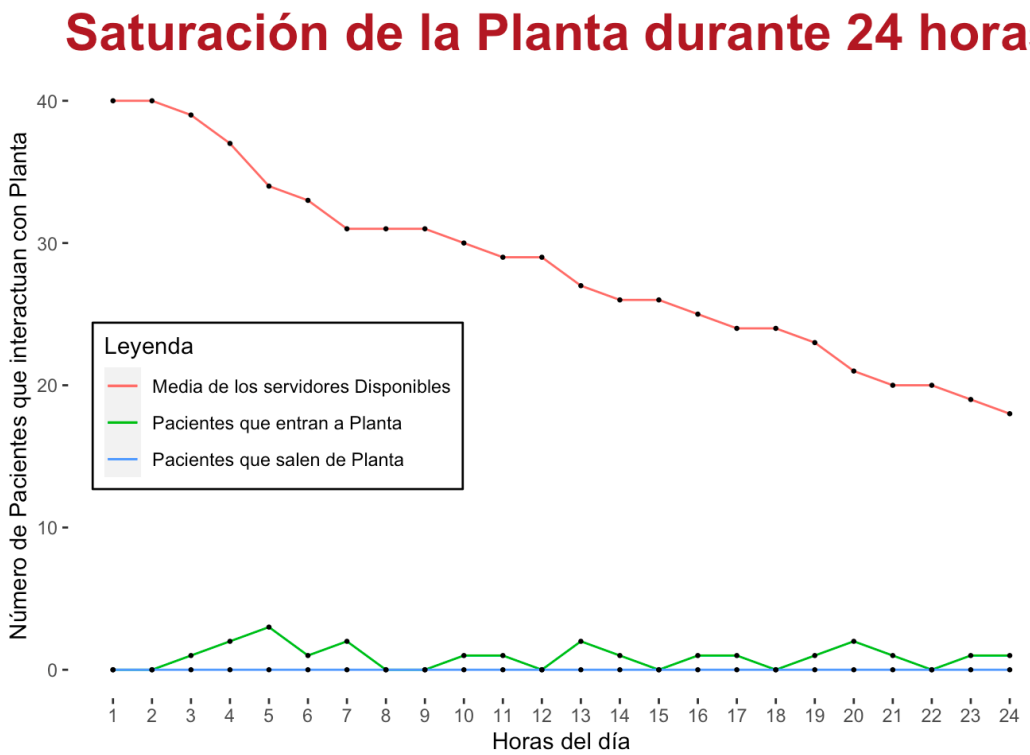


Figura 5.6. Saturación en planta Escenario 2.1. Fuente: Elaboración propia.

En cuanto a la saturación de la planta durante un período de 24 horas, se puede decir que es a partir de las 3 horas desde la inicialización de la simulación que la disponibilidad media de camas va disminuyendo. Esto se debe, a que siempre hay más pacientes que entrar en este nodo que salen, como se puede observar en la Figura 5.6. Además, este fenómeno también se puede demostrar mediante en análisis de los picos tanto de los usuarios que entran como los que salen. El punto máximo de entradas se da en el instante 4, mientras que el de salidas se da en el instante 0. En este último momento, hay que recalcar que como se "plotea" 24 horas y los tiempos de servicio que se dan en planta son como mínimo de 3 días, es por eso que no sale ningún paciente del sistema.

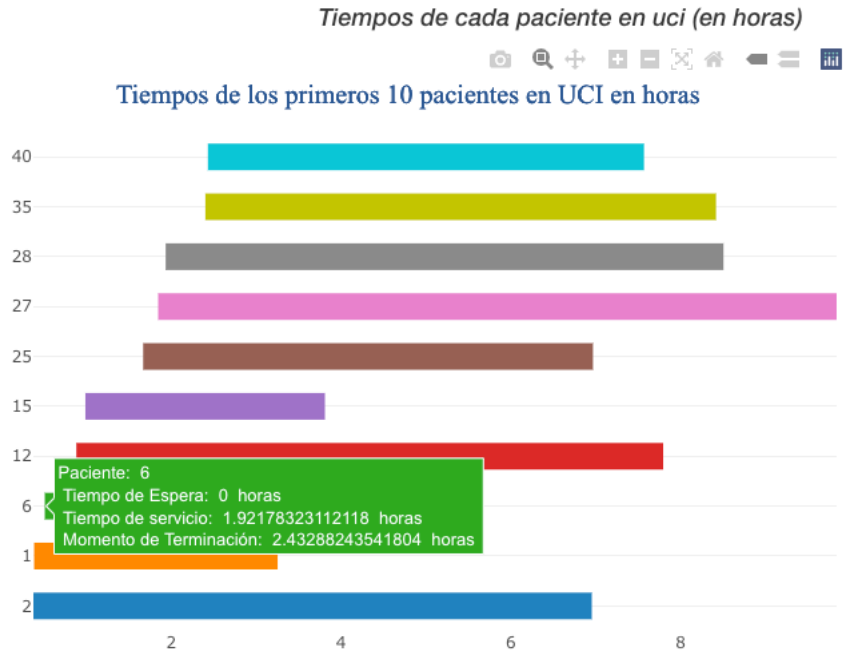


Figura 5.7. Duración de los 10 primeros pacientes en UCI Escenario 2.1. Fuente: Elaboración propia.

Como se puede observar en la Figura 5.7, los 10 primeros pacientes en entrar a la UCI lo hacen antes de las 2 horas desde haber inicializado el *simulador.* La pestaña que está abierta en dicha figura corresponde al paciente número 6, que espera 0 minutos en cola y es servido en aproximadamente 2 horas de operación. Por tanto, este cliente llegó a los 0.52 horas al nodo.

La longitud de las barras indica el tiempo de duración de los pacientes dentro del correspondiente nodo. Por lo tanto, a mayor longitud de estas, mayor será el tiempo que permanecerán en el eje. Cada paciente se identifica mediante un color y se ordenan en función al tiempo de llegada. Es por eso, que el primer individuo en ser atendido es el 2 y posteriormente será el 1 y así sucesivamente.

Como esta es la primera explicación del *escenario* 2, se procede a desarrollar los conceptos relacionados de las rutas que realizan los pacientes que salen de consulta (figura 5.8) y del sistema en general (figura 5.9).

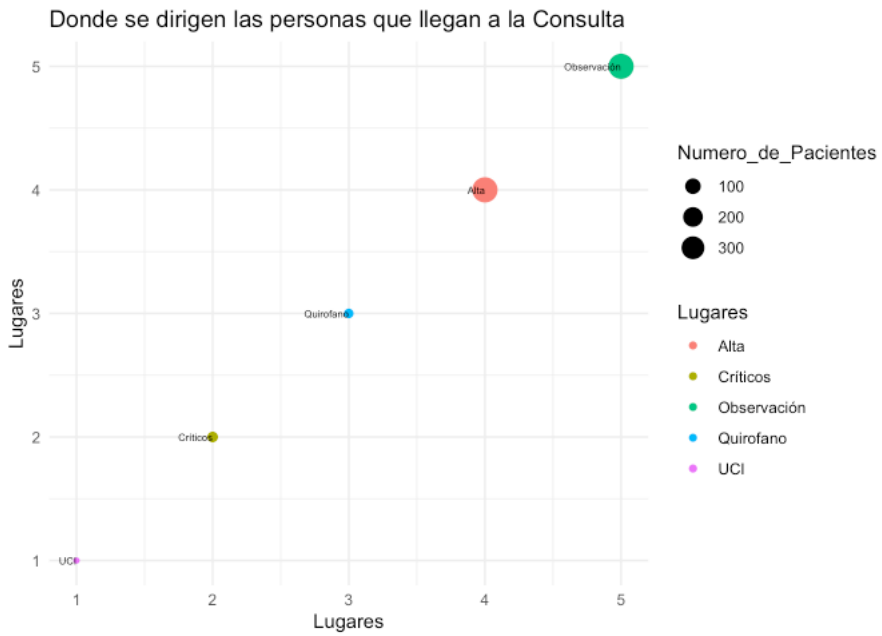


Figura 5.8. Rutas de los pacientes originados en Consulta para el Escenario 2.1. Fuente: Elaboración propia.

En la Figura 5.8, se expone la cantidad de pacientes que realizan las 5 posibles rutas provenientes de consulta. Se puede decir, que el camino más transitado es alta, acompañado de observación. El lugar, cuyo tamaño de círculo indica una cantidad pequeña de individuos, es quirófano y UCI.

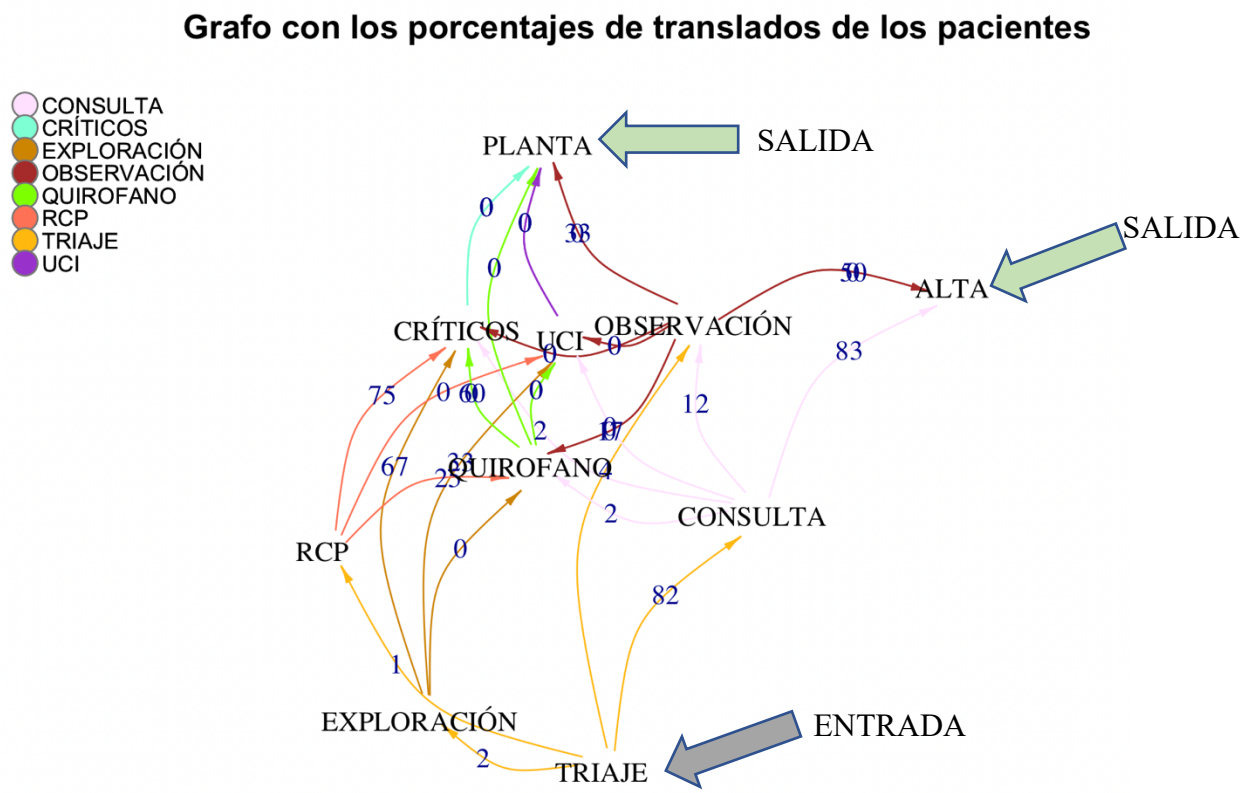


Figura 5.9. Grafo del sistema de urgencias hospitalarias Escenario 2.1. Fuente: Elaboración propia

Como se observa en la figura 5.9, se han calculado los porcentajes de pacientes que se dirigen de un nodo a otro. La flecha de color gris, indica el único punto de entrada al sistema. Por el otro lado, las dos flechas verdes recalcan las salidas de urgencias. Un ejemplo de la interpretación del presente gráfico seria que el 83% de los pacientes que salen de consulta acaban siendo derivados a alta, el 12% a observación, el 2% a críticos, el 2% a quirófano y solamente el 1% de los individuos a UCI.

Además, se puede concluir que dependiendo de las tasas de llegada de los escenarios, los porcentajes de los distintos caminos variaran, ya que en la figura 5.4 el 84% de los pacientes que entraban al sistema eran redirigidos a consulta y, en el caso del segundo *escenario* 1 subescenario, este porcentaje es del 82%.

Para el segundo subescenario del apartado 2, se van a utilizar una serie de distribuciones probabilísticas distintas para los bloques que componen el sistema como se ha explicado en el capítulo 5.1. En este caso, se expondrán las medidas de eficacia y la *saturación* en UCI.

Las cifras obtenidas gracias a la aplicación de la *teoría de colas* en esta simulación se encuentran en la siguiente figura.

Medidas de eficacia del sistema de urgencias hospitalarias

Nodos	Tiempo_Medio_Llegada	Tiempo_Medio_Servicio	Número_Servidores	Factor_Utilización	Tiempo_Medio_Cola	Tiempo_Medio_Duración	Número_Medio_Pacientes_En_Cola	Número_Medio_Servidores_Ocupados
Triaje	0.3076389	2.968353	3	0.0345465	0.0525871	3.02094	0.0564334	0.1036396
Consulta	0.2861111	22.296737	7	0.0018331	21.8709694	44.16771	0.7208738	0.0128320
Exploración	0.0062500	14.609527	8	0.0000535	0.0000000	14.60953	0.0000000	0.0004278
Observación	0.0465278	29.686250	8	0.0001959	0.0000000	29.68625	0.0000000	0.0015673
RCP	0.0027778	15.126801	7	0.0000262	0.0000000	15.12680	0.0000000	0.0001836
UCI	0.0048611	2153.914897	20	0.0000001	0.0000000	2153.91490	0.0000000	0.0000023
Críticos	0.0180556	2013.135967	12	0.0000007	816.7401109	2829.87608	0.5384615	0.0000090
Quirofano	0.0125000	18.790374	6	0.0001109	0.0000000	18.79037	0.0000000	0.0006652
Planta	0.0375000	3464.152375	40	0.0000003	0.0000000	3464.15237	0.0000000	0.0000108
Alta	0.2701389	NA	NA	NA	NA	NA	NA	NA
Sistema General	0.3076389	859.079214	108	0.0003581	93.1848519	2088.85529	0.1461965	0.0386751

Nota 1: Tiempo en minutos

Tabla 5.5. Medidas de eficacia para el Escenario 2.2. Fuente: Elaboración propia.

Observando la tabla 5.5, se puede observar que aplicando las distribuciones explicadas en el capítulo 1 de este proyecto, los tiempos medios en cola cambian drásticamente. Anteriormente, en planta existía un período de espera muy elevado, sin embargo, este parece ser que se ha trasladado a críticos. Resulta especial el número medio de pacientes que esperarán en cola con el sistema actual. De una simulación a otra ha habido una diferencia significativa, ya que se ha pasado de 0 personas de promedio en cola a 0.53 por minuto en este nodo.

Si los nodos, consulta y críticos, reciben gran afluencia de personas e incluso al borde de la congestión, el sistema en su conjunto no se ve repercutido, ya que sus números medio de servidores ocupados es de 0.03 y su factor de utilización es inferior a 0.001.

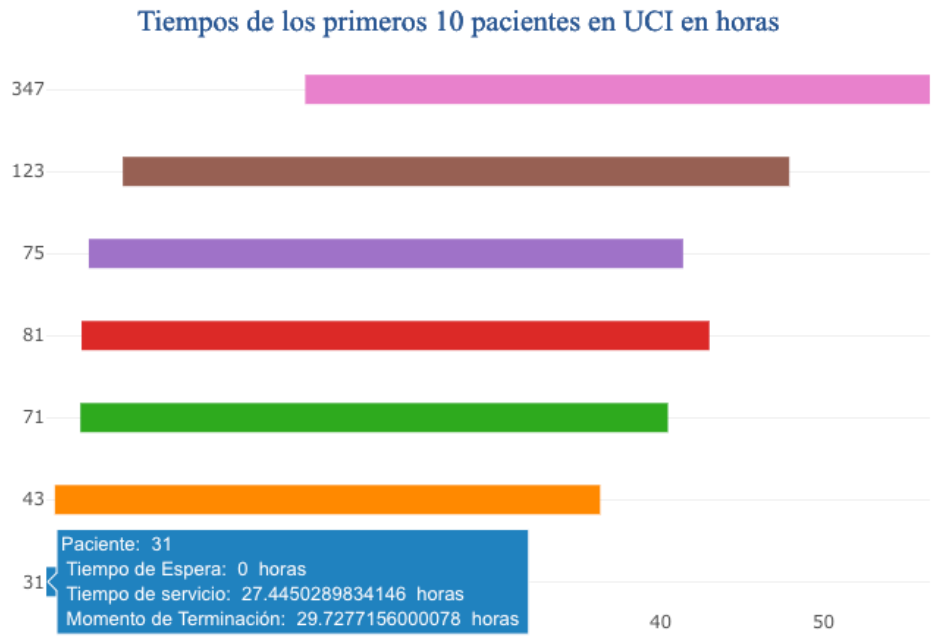


Figura 5.10. Tiempos de servicio en UCI escenario 2.2. Fuente: Elaboración propia.

Como se ha mencionado anteriormente, la UCI no está congestionada en este escenario, en el que las llegadas siguen una distribución Poisson de $\lambda = 0.25$ y sus tiempos de servicio siguen 5 distribuciones diferentes. Este nodo está dotado de 20 camas, pues bien, en todo un día solo se han utilizado 7.

El primer paciente en llegar a UCI es el 31, con un tiempo de espera de 0 horas, como se ha estipulado anteriormente, y su tiempo de servicio es de 27 horas. Como llega pasadas las 2 horas, este individuo es finalmente operado a las 29 horas.

A continuación se procede a la explicación del 3 y último subescenario del segundo apartado. Este tiene la peculiaridad del cambio del número de servidores en cada nodo, es por eso por lo que se muestra la siguiente tabla.

Medidas de eficacia del sistema de urgencias hospitalarias

Nodos	Tiempo_Medio_Llegada	Tiempo_Medio_Servicio	Número_Servidores	Factor_Utilización	Tiempo_Medio_Cola	Tiempo_Medio_Duración	Número_Medio_Pacientes_En_Cola	Número_Medio_Servidores_Ocupados
Triaje	0.3076389	2.968353	2	0.0518198	0.4257289	3.394082	0.2708804	0.1036396
Consulta	0.2861111	22.296737	6	0.0021387	70.9801645	93.276901	0.9635922	0.0128320
Exploración	0.0062500	14.609527	5	0.0000856	0.0000000	14.609527	0.0000000	0.0004278
Observación	0.0465278	29.686250	10	0.0001567	0.0000000	29.686250	0.0000000	0.0015673
RCP	0.0027778	15.126801	3	0.0000612	0.0000000	15.126801	0.0000000	0.0001836
UCI	0.0048611	2153.914897	10	0.0000002	0.0000000	2153.914897	0.0000000	0.0000023
Críticos	0.0180556	2013.135967	16	0.0000006	395.8682608	2409.004228	0.3846154	0.0000090
Quirofano	0.0125000	18.790374	4	0.0001663	0.0000000	18.790374	0.0000000	0.0006652
Planta	0.0375000	3464.152375	60	0.0000002	0.0000000	3464.152375	0.0000000	0.0000108
Alta	0.2701389	NA	NA	NA	NA	NA	NA	NA
Sistema General	0.3076389	859.079214	114	0.0003581	51.9193505	2089.047662	0.1798987	0.0408237

Nota 1: Tiempo en minutos

Tabla 5.6. Medidas de eficacia para el Escenario 2.3. Fuente: Elaboración propia.

En el caso de la Media tasa promedio de llegadas de los pacientes al sistema, con distribuciones probabilísticas distintas en los 4 bloques y el cambio de número de servidores disponibles en el día de la simulación, se obtienen prácticamente los mismos datos que en los dos supuestos anteriores.

En este caso, ha habido una mejoría respecto al subescenario 2, ya que se ha conseguido disminuir el tiempo medio en cola de críticos y, por ende, el número promedio de pacientes en cola, siendo ahora de 0.34 por minuto

Explicados los tres subescenarios del segundo apartado, se procede a la explicación de los resultados obtenidos mediante una tasa de llegadas según una distribución Poisson con parámetro $\lambda = 0.5$. Dicho medio se ha llamado nivel de congestión Saturado.

Medidas de eficacia del sistema de urgencias hospitalarias

Nodos	Tiempo_Medio_Llegada	Tiempo_Medio_Servicio	Número_Servidores	Factor_Utilización	Tiempo_Medio_Cola	Tiempo_Medio_Duración	Número_Medio_Pacientes_En_Cola	Número_Medio_Servidores_Ocupados
Triaje	0.5555556	3.039191	3	0.0609324	0.507039	3.546229	0.3162500	0.1827972
Consulta	0.4034722	22.540520	7	0.0025571	195.546378	218.086898	0.9862306	0.0178999
Exploración	0.0840278	15.126126	8	0.0006944	0.000000	15.126126	0.0000000	0.0055551
Observación	0.1562500	29.341881	8	0.0006656	0.553985	29.895866	0.0977778	0.0053252
RCP	0.0027778	13.653590	7	0.0000291	0.000000	13.653590	0.0000000	0.0002034
UCI	0.1291667	315.197071	20	0.0000205	519.711025	834.908096	0.8817204	0.0004098
Críticos	0.1770833	188.582148	12	0.0000783	1111.865654	1300.447802	0.9529412	0.0009390
Quirofano	0.0472222	22.899893	6	0.0003437	0.000000	22.899893	0.0000000	0.0020621
Planta	0.3423611	952.480671	40	0.0000090	3792.495706	4744.976377	0.9188641	0.0003594
Alta	0.2131944	NA	NA	NA	NA	NA	NA	NA
Sistema General	0.5555556	173.313545	108	0.0032055	624.519976	2739.375650	0.4615316	0.3461934

Nota 1: Tiempo en minutos

Tabla 5.7. Medidas de eficacia para el Escenario 3.1. Fuente: Elaboración propia.

Observando la tabla 5.7, se puede decir con rotundidad que se trata del modelo Saturado ya que los tiempos promedio en cola son exageradamente elevados. Un ejemplo de este es que para que un paciente puede ser atendido en la consulta, de media debe esperar 195 minutos.

Otro indicador que sugiere una congestión elevada es el número medio de clientes en cola, como por ejemplo en la UCI, que habrán 0.88 individuos por minuto. Es por esta razón que los tiempos medios de duración en cada uno de los nodos son extremadamente cuantiosos.

Tanto los tiempos promedios de servicio como el número de servidores en los 9 nodos del sistema, son exactamente los mismos que los dos primeros subescenarios de los 2 apartados explicados anteriormente en este capítulo.

Si se analiza el sistema en general, se puede decir que los tiempos de llegada al sistema son bastantes grandes, ya que por cada minuto que pase entrarán 0.55 pacientes. El nivel de congestión global es de 0.003 que en un principio se puede extraer la conclusión que el servicio de urgencias no está saturado, pero eso es erróneo porque si se tiene en cuenta los tiempos medios de espera y la cantidad promedio de usuarios en cola se deduce que el hospital está bastante saciado.

Esa congestión mencionada anteriormente, se ve reflejada en el siguiente gráfico que muestra las fluctuaciones de los pacientes que entran y salen de la planta durante solo 38 horas.

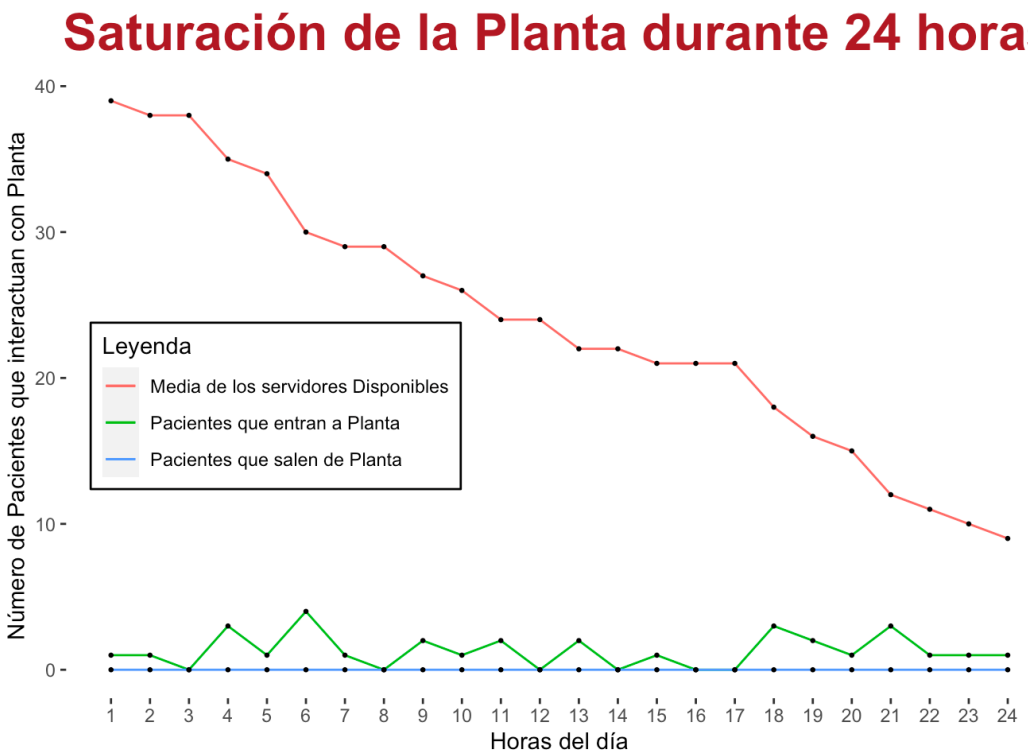


Figura 5.11. Saturación en planta Escenario 3.1. Fuente: Elaboración propia.

En esta representación gráfica, se observa que a partir de las 1 horas desde la inicialización del *simulador*, los servidores medios disponibles optan el valor de 29 Esto se debe a la gran *saturación* que ha soportado el *sistema de urgencias* en este *escenario*. A partir de este instante, los servidores medios disponibles van disminuyendo hasta alcanzar al final del día la cantidad de 10.

Respecto a las fluctuaciones tanto de los pacientes que se incorporan como los que se evaden de planta, existe un patrón muy peculiar. En los instantes 10,14,26 y 36 se observa que estas oscilaciones están relacionadas negativamente, es decir, cuando existen pacientes que salen del nodo, en el mismo momento se produce una llegada insólita de individuos. Sin embargo, como únicamente se ha "ploteado" 24 horas, el total de salidas de este nodo es 0.

Con el gráfico de la figura 5.11, hay que ir con meticulosidad a la hora de realizar interpretaciones ya que si por alguna casualidad el número de pacientes que entrar y salen son valores inferiores a 5 y el número medio de servidores disponibles es 0, no significa que el algoritmo matemático encargado de realizar dicho gráfico este defectuoso. Esta

situación se debe a que en la figura no hay una variable que representa en el instante t cuantos pacientes están siendo atendidos.

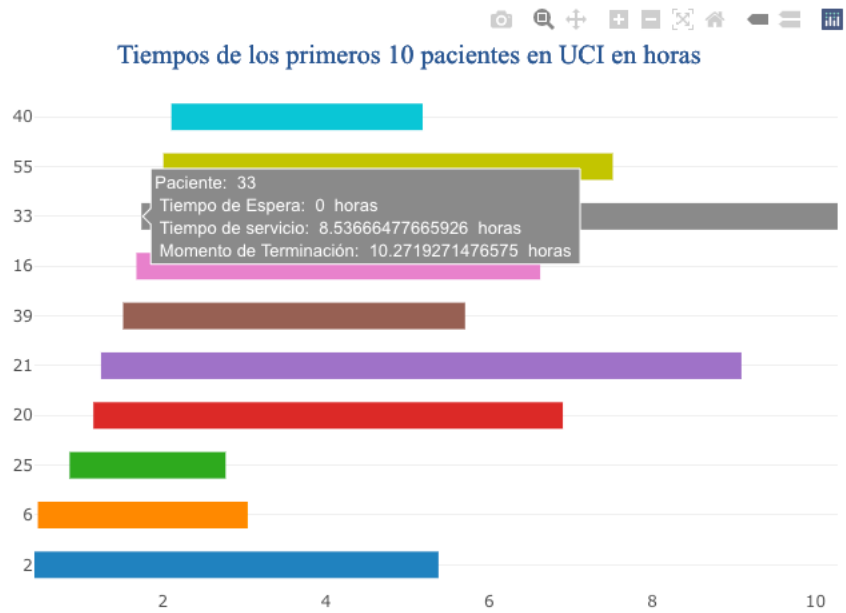


Figura 5.12. Duración de los 10 primeros pacientes en UCI Escenario 3.1. Fuente: Elaboración propia.

Observando la figura 5.12, se puede concluir que los 10 primeros pacientes en acudir a la UCI en el *escenario* 3, cuya tasa de servicio se considera saturada, se encuentran dentro del estándar visto hasta la actualidad, es decir, en todos los *escenarios* analizados anteriormente, los diez individuos siempre han llegado a las primeras dos horas de inicio de la simulación.

El tiempo de duración más extenso corresponde al paciente con ID 33. Este ha estado siendo servido un período de 8.5 horas. Sin embargo, el sujeto con identificación 25, es el que posee menor tiempo de duración ya que está siendo servido 1.91 horas y no tiene que realizar una cola ya que es el tercero en entrar y la UCI está compuesta 20 servidores.

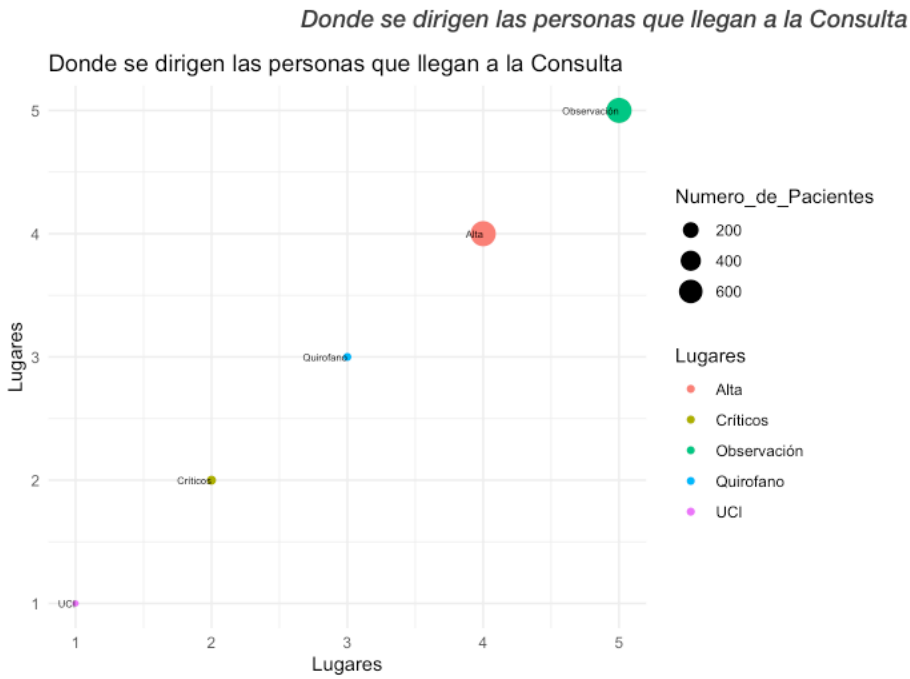


Figura 12. Rutas de los pacientes originados en Consulta para el Escenario 3.1. Fuente: Elaboración propia.

En el anterior gráfico, figura 5.13, se observa la cantidad de pacientes que realizan las 5 posibles rutas provenientes de consulta. Se puede decir, que el camino más transitado es alta, acompañado de observación. El lugar, cuyo tamaño de círculo indica una cantidad pequeña de individuos, es UCI. Para una correcta interpretación del gráfico, se ha ordenador de menor a mayor valor de transiciones en los nodos.

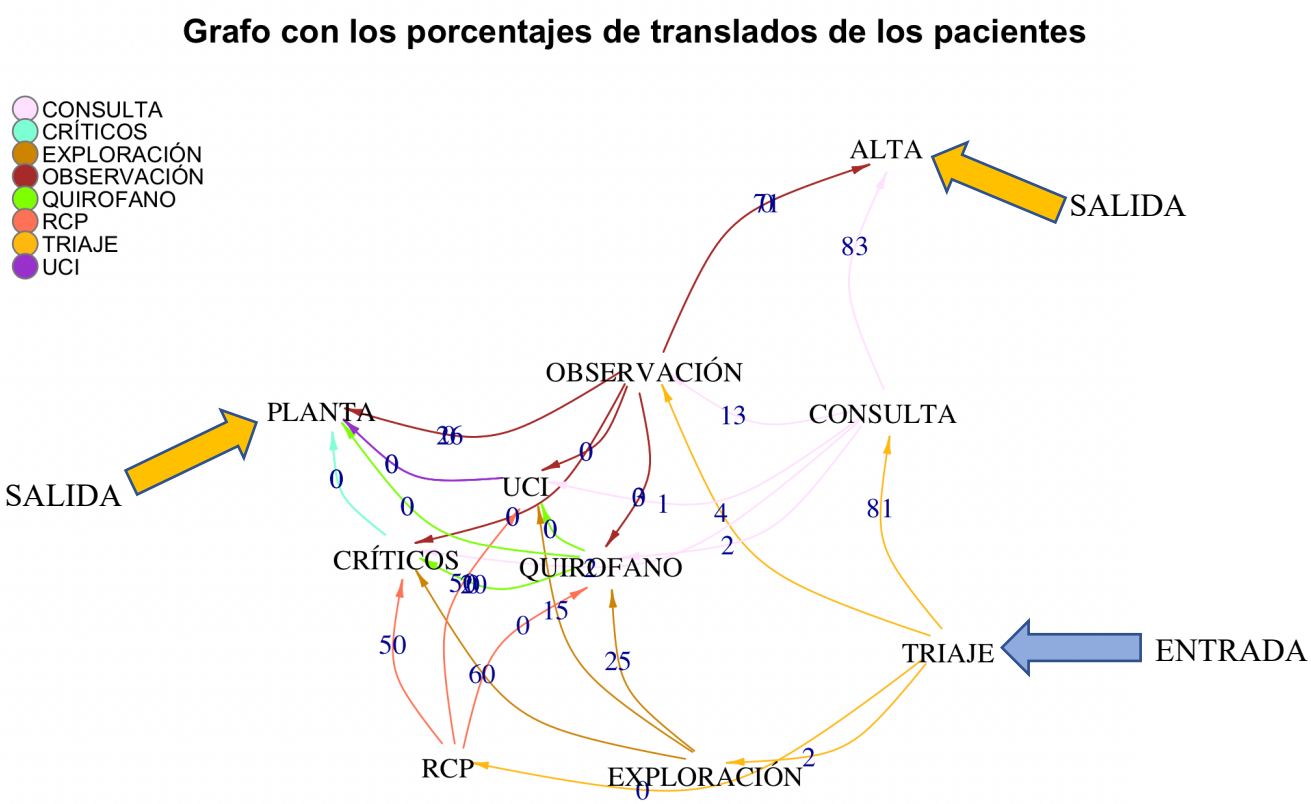


Figura 5.14. Grafo del sistema de urgencias hospitalarias para el Escenario 3.1. Fuente: Elaboración propia.

Como se observa en la figura 5.14, se han calculado los porcentajes de pacientes que se dirigen de un nodo a otro. La flecha de color azul, indica el único punto de entrada al sistema. Por el otro lado, las dos flechas amarillas recalcan las salidas de urgencias. Un ejemplo de la interpretación del presente gráfico seria que el 60% de los pacientes que salen de exploración acaban siendo derivados a críticos, el 25% a quirófano y el 15% de los individuos a observación.

Además, se puede concluir que dependiendo de las tasas de llegada de los *escenarios*, los porcentajes de los distintos caminos variaran, ya que en la figura 5.4 el 84% de los pacientes que entraban al sistema eran redirigidos a consulta, en el caso de la figura 5.9 el porcentaje es del 82% y, en el supuesto de que las tasas de llegada al *sistema de urgencias* siguen una distribución Poisson con parámetro $\lambda = 0.5$, es del 81%.

Para el 2 subescenario del tercer apartado, se van a utilizar una serie de distribuciones probabilísticas distintas para los bloques que componen el sistema como se ha explicado en el capítulo 5.1. En este caso, se expondrán las medidas de eficacia y la *saturación* en UCI.

Las cifras obtenidas gracias a la aplicación de la *teoría de colas* en esta simulación se encuentran en la siguiente tabla.

Medidas de eficacia del sistema de urgencias hospitalarias

Nodos	Tiempo_Medio_Llegada	Tiempo_Medio_Servicio	Número_Servidores	Factor_Utilización	Tiempo_Medio_Cola	Tiempo_Medio_Duración	Número_Medio_Pacientes_En_Cola	Número_Medio_Servidores_Ocupados
Triaje	0.5555556	3.039191	3	0.0609324	0.507039	3.546229	0.3162500	0.1827972
Consulta	0.5145833	22.603701	7	0.0032522	463.376040	485.979741	0.9905533	0.0227654
Exploración	0.0138889	14.929315	8	0.0001163	0.000000	14.929315	0.0000000	0.0009303
Observación	0.0902778	29.582535	8	0.0003815	0.000000	29.582535	0.0000000	0.0030517
RCP	0.0027778	15.867485	7	0.0000250	0.000000	15.867485	0.0000000	0.0001751
UCI	0.0125000	2414.497924	20	0.0000003	0.000000	2414.497924	0.0000000	0.0000052
Críticos	0.0250000	2113.162493	12	0.0000010	1450.742411	3563.904904	0.6666667	0.0000118
Quirofano	0.0201389	19.664012	6	0.0001707	0.000000	19.664012	0.0000000	0.0010241
Planta	0.0652778	4496.870583	40	0.0000004	790.309399	5287.179982	0.4787234	0.0000145
Alta	0.4902778	NA	NA	NA	NA	NA	NA	NA
Sistema General	0.5555556	1014.130894	108	0.0005478	300.548321	3007.976350	0.2724659	0.0591640

Nota 1: Tiempo en minutos

Tabla 5.8. Medidas de eficacia para el Escenario 3.2. Fuente: Elaboración propia.

Observando la tabla 5.8, se puede observar que aplicando las distribuciones explicadas en el capítulo 1 de este proyecto, los tiempos medios en cola cambian drásticamente. Anteriormente, en planta existía un período de espera muy elevado, sin embargo, este parece ser que se ha trasladado a críticos. Resulta especial el número promedio de pacientes que esperarán en cola con el sistema actual. De una simulación a otra ha habido una diferencia significativa, ya que se ha pasado de 0.95 persones de media en cola a 0.67 por minuto en este nodo. Por lo tanto con las distribuciones elegidas se han conseguido bajar los tiempos y el número medio de clientes en cola en planta pero no en críticos.

Si los nodos, consulta y críticos, reciben gran afluencia de personas e incluso al borde de la congestión, el sistema en su conjunto no se ve repercutido, ya que sus números medio de servidores ocupados es de 0.06 y su factor de utilización es inferior a 0.001.

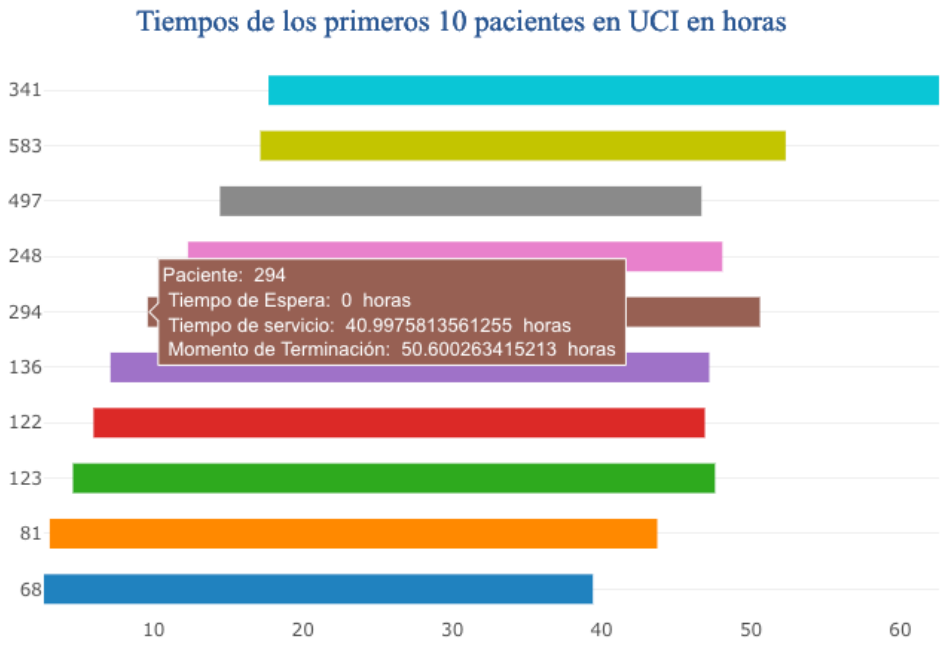


Figura 5.15. Tiempos de servicio en UCI escenario 3.2. Fuente: Elaboración propia.

Como se ha mencionado anteriormente, la UCI no está congestionada en este escenario, en el que las llegadas siguen una distribución Poisson de $\lambda = 0.5$ y sus tiempos de servicio siguen 5 distribuciones diferentes. Este nodo está dotado de 20 camas, pues bien, en todo un día solo se han utilizado 10.

El sexto paciente en llegar a UCI es el 294, con un tiempo de espera de 0 horas, como se ha estipulado anteriormente, y su tiempo de servicio es de 40 horas. Como llega pasadas las 9 horas, este sujeto es finalmente operado a las 50 horas.

A continuación se procede a la explicación del 3 y último subescenario del tercer apartado. Este tiene la peculiaridad del cambio del número de servidores en cada nodo, es por eso por lo que se muestra la siguiente tabla.

Medidas de eficacia del sistema de urgencias hospitalarias

Nodos	Tiempo_Medio_Llegada	Tiempo_Medio_Servicio	Número_Servidores	Factor_Utilización	Tiempo_Medio_Cola	Tiempo_Medio_Duración	Número_Medio_Pacientes_En_Cola	Número_Medio_Servidores_Ocupados
Triaje	0.5555556	3.039191	2	0.0913986	4.785269	7.824459	0.8000000	0.1827972
Consulta	0.5145833	22.603701	6	0.0037942	656.789910	679.393612	0.9919028	0.0227654
Exploración	0.0138889	14.929315	5	0.0001861	0.000000	14.929315	0.0000000	0.0009303
Observación	0.0902778	29.582535	10	0.0003052	0.000000	29.582535	0.0000000	0.0030517
RCP	0.0027778	15.867485	3	0.0000584	0.000000	15.867485	0.0000000	0.0001751
UCI	0.0125000	2414.497924	10	0.0000005	339.077834	2753.575758	0.4444444	0.0000052
Críticos	0.0250000	2113.162493	16	0.0000007	737.858167	2851.020660	0.5555556	0.0000118
Quirofano	0.0201389	19.664012	4	0.0002560	0.000000	19.664012	0.0000000	0.0010241
Planta	0.0652778	4496.870583	60	0.0000002	0.000000	4496.870583	0.0000000	0.0000145
Alta	0.4902778	NA	NA	NA	NA	NA	NA	NA
Sistema General	0.5555556	1014.130894	114	0.0005478	193.167909	2614.885158	0.3102114	0.0624508

Nota 1: Tiempo en minutos

Tabla 5.9. Medidas de eficacia para el Escenario 3.3. Fuente: Elaboración propia.

En el caso de la Saturada tasa promedio de llegadas de los pacientes al sistema, con distribuciones probabilísticas distintas en los 4 bloques y el cambio de número de servidores disponibles en el día de la simulación, se obtienen prácticamente los mismos datos que en los dos supuestos anteriores.

En este caso, ha habido una mejoría respecto al subescenario 2, ya que se ha conseguido disminuir el tiempo medio en cola de críticos y, por ende, el número promedio de pacientes en cola, siendo ahora de 0.55 por minuto

Por consiguiente, gracias a los resultados obtenidos en los 9 escenarios anteriores, cada uno de ellos con tasas de llegadas de las pacientes distintas, se puede realizar una comparación de las medidas de eficacia del modelo de colas, para poder concluir que dimensionamiento es el óptima para los recursos disponibles.

Dichas comparaciones se construyen en base a los subescenarios simulados, es decir, se dimensionan tres tablas en las que se encontraran las cifras calculadas del *Escenario1*, *Escenario2*, *Escenario3* agrupadas por los subapartados. Para entender mejor estos supuestos, se proceden a explicar.

Medidas de eficacia para el sistema general con los Subescenearios 1

	Tiempo_Medio_Llegadas	Factor_Utilización	Tiempo_Medio_en_Cola	Número_Medio_Pacientes_en_Cola	Número_Medio_Servidores_Ocupados
Escenario1	0.1291667	0.0000948	0.0121928	0.0194939	0.0124220
Escenario2	0.3076389	0.0017674	293.2698291	0.2994331	0.1843957
Escenario3	0.5555556	0.0032055	624.5199760	0.4615316	0.3461934

Nota 1: Tiempo en minutos

Tabla 5.10. Comparación Subescenarios 1. Fuente: Elaboración propia.

En la tabla 5.10, están representadas aquellas medidas de eficacia de modelos de colas que se han calculado mediante la simulación de los diferentes subescenarios 1. En general, a mayor *saturación* del sistema, es decir, del *Escenario* 1 al 3, mayor será el tiempo medio de llegadas al sistema, pasando de 0.13 pacientes por minuto a un máximo de 0.55 por minuto.

A simple vista se observa que los factores de utilización poseen unos valores similares, pero esto no significa que el sistema no esté saturado, sino lo contrario. Como existen 9 nodos y en cada uno de ellos las cifras de estos parámetros varían bastante, la media es un poco dispar. Por lo que para poder decir de si se trata o no de un sistema congestionado se tendrá que hacer un análisis de las medidas restantes que se indican en la tabla 5.10.

Las últimas tres variables expuestas en la tabla anterior son las más importantes a la hora de realizar una correcta conclusión. Estas nos indican el tiempo medio que pasará un paciente dentro del sistema, cuantos sujetos habrá delante de él y cuál es el promedio de servidores dando servicio. Un ejemplo claro del aumento progresivo de las cifras de estas variables es el incremento en el tiempo promedio de espera en cola, ya que en el primer subescenario del apartado 1 se obtuvo que los individuos esperaban de media unos 0.012 minutos mientras que en el 1 subsescenario del apartado 3 se obtuvo una media de 624 minutos.

Por tanto, se demuestra que a mayor número de llegadas de pacientes al sistema, mayores serán los tiempos de espera, el número medio de servidores ocupados y la congestión del sistema de *urgencias hospitalarias*.

El siguiente aspecto por contrastar, son las medidas de eficacia del sistema en su totalidad que se coordinan con los subescenarios 2 de los 3 apartados/bloques simulados. En estos los tiempos de servicio se diversifican en 5 distribuciones: uniforme; triangular; beta; log-normal y Poisson, dejando las mismas cuantías de servidores que en el caso anterior.

Medidas de eficacia para el sistema general con los Subescenearios 2

	Tiempo_Medio_Llegadas	Factor_Utilización	Tiempo_Medio_en_Cola	Número_Medio_Pacientes_en_Cola	Número_Medio_Servidores_Ocupados
Escenario1	0.1291667	1e-04	0.0112837	0.0194939	0.0139283
Escenario2	0.3076389	2e-04	156.3855360	0.1461865	0.0323075
Escenario3	0.5555556	4e-03	321.7786530	0.2535534	0.0531591

Nota 1: Tiempo en minutos

Tabla 5.11. Comparación Subescenarios 2. Fuente: Elaboración propia.

En la tabla 5.11, están representadas aquellas medidas de eficacia de modelos de colas que se han calculado mediante la simulación de los diferentes subescenarios 2. En general, a mayor *saturación* del sistema, es decir, del *Escenario* 1 al 3, mayor será el tiempo medio de llegadas al sistema, pasando de 0.13 pacientes por minuto a un máximo de 0.55 por minuto.

A simple vista se observa que los factores de utilización poseen unos valores similares, pero esto no significa que el sistema no esté saturado, sino lo contrario. Como existen 9 nodos y en cada uno de ellos las cifras de estos parámetros varían bastante, la media es un poco dispar. Por lo que para poder decir de si se trata o no de un sistema congestionado se tendrá que hacer un análisis de las medidas restantes que se indican en la tabla 5.11.

Las últimas tres variables expuestas en la tabla anterior son las más importantes a la hora de realizar una correcta conclusión. Estas nos indican el tiempo medio que pasará un paciente dentro del sistema, cuantos individuos habrá delante de él y cuál es el promedio de servidores dando servicio. Un ejemplo claro del aumento progresivo de las cifras de estas variables es el incremento en el tiempo promedio de espera en cola, ya que en el segundo subescenario del apartado 1 se obtuvo que los sujetos esperaban de media unos 0.01 minutos mientras que en el 2 subsescenario del apartado 3 se obtuvo una media de 321 minutos. La diferencia en este caso no es tan drástica como en la comparación anterior, pero sigue siendo una variabilidad muy elevada.

Por tanto, se demuestra que a mayor número de llegadas de pacientes al sistema, mayores serán los tiempos de espera, el número medio de servidores ocupados y la *congestión* del sistema de *urgencias hospitalarias*.

El siguiente aspecto por contrastar, son las medidas de eficacia del sistema en su totalidad que se coordinan con los subescenarios 3 de los tres bloques simulados. En estos los tiempos de servicio son los mismos que en caso anterior, sin embargo, los números de servidores disponibles en la inicialización del *simulador* son distintos, como ya se explicó en el capítulo 5.1.

Medidas de eficacia para el sistema general con los Subescenearios 3

	Tiempo_Medio_Llegadas	Factor_Utilización	Tiempo_Medio_en_Cola	Número_Medio_Pacientes_en_Cola	Número_Medio_Servidores_Ocupados
Escenario1	0.1291667	0.0001290	0.0248694	0.0748290	0.0147021
Escenario2	0.3076389	0.0002991	83.0612425	0.1798170	0.0340971
Escenario3	0.5555556	0.0049220	251.0472830	0.3102114	0.0561240

Nota 1: Tiempo en minutos

Tabla 5.12. Comparación Subescenarios 3. Fuente: Elaboración propia.

En la tabla 5.12, están representadas aquellas medidas de eficacia de modelos de colas que se han calculado mediante la simulación de los diferentes subescenarios 3. En general, a mayor *saturación* del sistema, es decir, del *Escenario* 1 al 3, mayor será el tiempo medio de llegadas al sistema, pasando de 0.13 pacientes por minuto a un máximo de 0.55 por minuto.

A simple vista se observa que los factores de utilización poseen unos valores similares, pero esto no significa que el sistema no esté saturado, sino lo contrario. Como existen 9 nodos y en cada uno de ellos las cifras de estos parámetros varían bastante, la media es un poco dispar. Por lo que para poder decir de si se trata o no de un sistema congestionado se tendrá que hacer un análisis de las medidas restantes que se indican en la tabla 5.12.

Las últimas tres variables expuestas en la tabla anterior son las más importantes a la hora de realizar una correcta conclusión. Estas nos indican el tiempo medio que pasará un paciente dentro del sistema, cuantos sujetos habrá delante de él y cuál es el promedio de servidores dando servicio. Un ejemplo claro del aumento progresivo de las cifras de estas variables es el incremento en el tiempo promedio de espera en cola, ya que en el tercer subescenario del apartado 1 se obtuvo que las personas esperaban de media unos 0.02 minutos mientras que en el 3 subsescenario del apartado 3 se obtuvo una media de 251 minutos. Esta diferencia es la menor respecto a las 3 comparaciones realizadas, por

lo que con las 5 distribuciones distintas y el cambio de la cantidad de servidores disponibles en la inicialización del sistema, se consigue unos menores tiempos de espera.

Sin embargo, se demuestra que a mayor número de llegadas de pacientes al sistema, mayores serán los tiempos de espera, el número medio de servidores ocupados y la congestión del sistema de *urgencias hospitalarias*.

Concluyendo con las múltiples comparaciones de la simulación, se propone una simulación de un mes entero, 31 días, mediante los 3 *escenarios* con los subapartados 1, ya que estos son los más habituales en urgencias.

Para este contraste se utilizará el algoritmo realizado, y la única modificación/diferencia que habrá que hacer es indicar en el parámetro "Días" la cantidad a simular. En la interfaz gráfica se pueden realizar las respectivas consultas de los pacientes, el grafo del sistema y las saturaciones en planta y en UCI.

En la figura 5.16 se expone el nivel de *saturación* que soporta planta en esta simulación en un entorno Bajo, es decir, con un λ de 0.1.

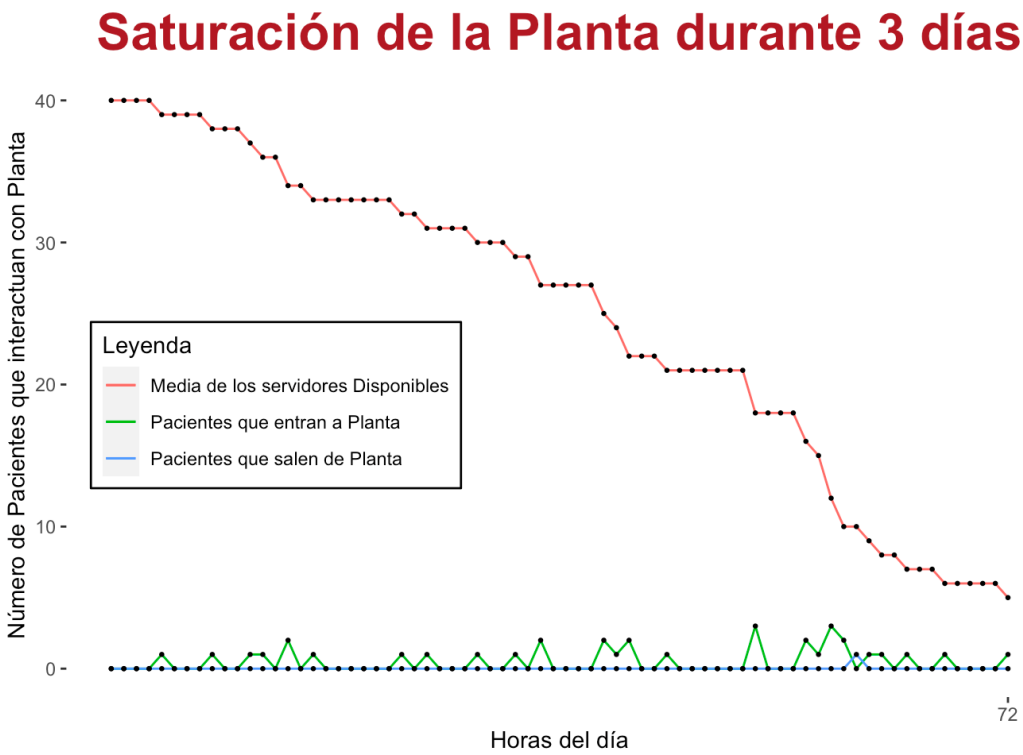


Figura 5.16. Saturación en planta de 3 días habiendo simulado 1 mes

Medidas de eficacia para el sistema general en el período de 31 días

	Tiempo_Medio_Llegadas	Factor_Utilización	Tiempo_Medio_en_Cola	Número_Medio_Pacientes_en_Cola	Número_Medio_Servidores_Ocupados
Escenario1	0.1291667	0.0000926	4280.667	0.2536347	0.0099987
Escenario2	0.3076389	0.0022191	17180.255	0.4047827	0.0236627
Escenario3	0.5555556	0.0039690	35981.420	0.4736210	0.0428610

Nota 1: Tiempo en minutos

Tabla 5.13. Medidas de eficacia para los 3 Escenarios período de 31 días. Fuente: Elaboración propia.

En la tabla 5.13, están representadas aquellas medidas de eficacia de modelos de colas que se han calculado mediante la simulación de los diferentes *escenarios*. En general, a mayor *saturación* del sistema, es decir, del *Escenario* 1 al 3, mayor será el tiempo medio de llegadas al sistema, pasando de 0.13 pacientes por minuto a un máximo de 0.55 por minuto.

A simple vista se observa que los factores de utilización poseen unos valores similares, pero esto no significa que el sistema no esté saturado, sino lo contrario. Como existen 9 nodos y en cada uno de ellos las cifras de estos parámetros varían bastante, la media es un poco dispar. Por lo que para poder decir de si se trata o no de un sistema congestionado se tendrá que hacer un análisis de las medidas restantes que se indican en la tabla 5.13.

Las últimas tres variables expuestas en la tabla anterior son las más importantes a la hora de realizar una correcta conclusión. Estas nos indican el tiempo medio que pasará un paciente dentro del sistema, cuantos usuarios habrá delante de él y cuál es el promedio de servidores dando servicio. Un ejemplo claro del aumento progresivo de las cifras de estas variables es el incremento en el tiempo promedio de espera en cola, ya que en el *escenario* 1 se obtuvo que los individuos esperaban de media unos 4280 minutos mientras que en el tercer *escenario* se obtuvo una media de 35981 minutos. Estos períodos son tan elevados ya que se está simulando un mes entero, y el algoritmo calcula la media de esos días. Además, se puede observar una gran diferencia de tiempos dependiendo de si se trata de un sistema de congestión Bajo, Medio o Saturado. Por otro lado, tanto el número medio de pacientes en cola como del promedio de servidores/camas ocupadas aumenta de forma constante en función del grado de *saturación*. A simple vista, se podría decir que 0.4 clientes esperando por minuto en cola no parece tan grave, pero si se da la casualidad de que esto le pasa a cualquier individuo que este leyendo este proyecto, se pondría nervioso y con principios de ansiedad.

En la figura 5.16, se puede observar que la *saturación* en planta cada vez que transcurre el tiempo empeora, por lo que se tendría que revisar el sistema interno de esta. Además, los pacientes que salen durante estos 3 días es 0. Por esa razón, la congestión del nodo empeora conforme aumenta el tiempo de simulación.

Por lo tanto, queda argumentado que a mayor número de llegadas de pacientes al sistema, mayores serán los tiempos de espera, el número medio de servidores ocupados y la congestión del sistema de *urgencias hospitalarias*.

CAPÍTULO 6

Discusión y conclusiones.

Las listas de espera son uno de los principales problemas del sistema público sanitario español. Estas se crean inicialmente por la necesidad de planificación y por el uso incorrecto de los recursos disponibles. Es a partir de aquí, que la percepción de estabilidad de los pacientes se ve claramente dañada.

Por eso, el objetivo del presente proyecto era analizar por simulación un *servicio de urgencias* en un hospital. Para ello se desarrolló una herramienta para la toma de decisiones que permite analizar los tiempos de servicio de los pacientes que acuden a urgencias dependiendo de los parámetros que definan dicho servicio. Con esta herramienta se puede optimizar el diseño y dimensionamiento de las unidades que componen el servicio de *urgencias en un hospital*.

En este proyecto se explicó un ambiente de desarrollo para la modelización y simulación de redes de colas en un hospital que cuenta con una interfaz de usuario en el lenguaje R lo que permite obtener los resultados del simulador en cualquier sistema operativo. Esta simulación se fundamenta en un *simulador* de eventos que proporciona unos valores que hacen referencia a la *teoría de colas* mediante el uso de la generación de números aleatorios.

Mediante la herramienta informática establecida se ha podido crear 9 tipos de marcos en un hospital, además de la realización de 3 *escenarios* en un período de 31 días. Estas situaciones se han creado implementando distribuciones de llegada Poisson, y con 5 distribuciones a la hora de promulgar los tiempos de servicio en cada uno de los 9 nodos que componen el sistema de urgencias.

Los resultados obtenidos mediante este *simulador* proporcionan una visión del tráfico existente en el hospital, cuanto tiempo estará de media un individuo dentro de este y cuál es la *saturación* de aquellos nodos más importantes.

Independientemente, de las tasas de llegadas, es decir, del parámetro λ elegido, el factor de utilización del sistema en su conjunto nunca llega a sobrepasar los 0.01, lo que es una muy buena noticia. Sin embargo, los tiempos medios en cola y el número promedio de pacientes en espera si se ven repercutidos, ya que en un *escenario* Saturado estos períodos se disparan llegando a valores de incluso horas de espera. Los nodos más importantes para el análisis de la simulación son UCI y planta. En los marcos en que las tasas de llegada son bajas o medias y la simulación realizada es de únicamente un día, estas no se ven repercutidas, en cuanto a la *saturación* se refiere, pero si al tráfico de pacientes.

Sin embargo, cuando se simula un mes, estos dos ejes se ven sobrepasados, y se tendría que pensar sobre la disponibilidad de las camas, incluso fijando un número de servidores más elevado para evitar a toda costa esta congestión.

Por tanto, como esta herramienta no necesita un gran nivel de hardware, se podrían hacer simulaciones cada dos días para "predecir" si la planificación actual de cualquier hospital es capaz de soportar una gran afluencia de pacientes. Si en los hospitales, se sobrepone este algoritmo matemático, se observaría el nivel óptimo en camas tanto en UCI como en planta.

De la totalidad de las comparaciones realizadas se extrae la conclusión que la saturación del hospital en general es muy complicada que se dé, pero la congestión en los nodos es más fácil que ocurra. Los ejes que se han visto más repercutidos son consulta, críticos, UCI y planta. Por lo que el director del sistema hospitalario tendrá que realizar una gestión más compleja y efectiva del sistema de *urgencias hospitalarias* que tutela.

Por último, hay que matizar que existe trabajo futuro a la hora de realizar un sistema de urgencias más complejo. Al final, cuantas más interacciones haya entre los distintos nodos mayor será la precisión de los resultados obtenidos en el *simulador* en comparación a la vida real. De la misma forma, la interfaz de usuario se podría ver mejorada, creando una aplicación en el propio ordenador, que se encarga de obtener una serie de inputs e imprimir los resultados en pocos minutos. Esta "app" estaría en los ordenadores de los responsables de la planificación y gestión del sistema de *urgencias hospitalarias*.

CAPÍTULO 7

Bibliografía.

[1] P. Borja Velázquez Martí. Aplicación de modelos de teoría de colas a la gestión asistencial en los centros de salud. *Universidad Técnica de Ambato, 2017*

[2] P. Ricardo Cao Abad. Introducción a la Simulación y a la Teoría de Colas. *Facultad de Informática. Universidad de Coruña. 2002.*

[3] P. José León Paniagua. Unidad de urgencias hospitalaria. *Instituto de salud Carlos III, Madrid, 2016*

[4] P. Frank Kronthaler. Data Analysis with RStudio. *2020*

[5] P. François Chollet. Deep learning with R. *2018*

CAPÍTULO 8

Anexo.

Se comparte de forma libre el código empleado en *Rstudio* para los cálculos pertinentes en este proyecto.

```
########### INSTALAMOS LAS LIBRERÍAS NECESARIAS ###################
install.packages("ggplot2")
install.packages("tidyverse")
install.packages("plotly")
install.packages("ggraph")
install.packages("ggthemes")
install.packages("dplyr")
install.packages("igraph")
install.packages("pander")
install.packages("shiny")
install.packages("quantmod")
install.packages("DT")
install.packages("kableExtra")
install.packages("gt")
install.packages("WriteXLS")
install.packages("triangle")

########## CARGAMOS LAS LIBRERÍAS NECESARIAS ######################
library(ggplot2)
library(tidyverse)
library(plotly)
library(ggraph)
library(ggthemes)
library(dplyr)
library(igraph)
library(pander)
library(shiny)
library(quantmod)
library(DT)
library(kableExtra)
library(gt)
library(WriteXLS)
library(triangle)
############################## PARAMETROS A INTRODUCIR ##############
Días<- 1
Escenario<- 3 #1=Bajo; 2=Medio; 3=Saturado
Disponible_Tri<- 3 #Número de Servidores para TRIAJE #2 #3
Disponible_Con<- 7 #Número de Servidores para CONSULTA #6 #7
Disponible_Exp<- 8 #Número de Servidores para EXPLORACIÓN #5 #8
Disponible_RCP<- 7 #Número de Servidores para RCP # 3 #7
Disponible_Obs<- 8 #Número de Servidores para OBSERVACIÓN #10 #8
Disponible_QUI<- 6 #Número de Servidores para QUIROFANO #4 #6
Disponible_CRI<- 12 #Número de Servidores para CRÍTICOS #16 #12
```

```
Disponible_UCI<- 20 #Número de Servidores para UCI #10 #20
S_Disponible_PLA2<- 40 #Número de Servidores para PLANTA #60 #40

#########################################################################
###
{
if(Escenario==1){
  lambda=0.1
  Escenarios<- "BAJO"

}
if(Escenario==2){
  lambda=0.25
  Escenarios<- "MEDIO"
}
if(Escenario==3){
  lambda=0.5
  Escenarios<- "SATURADO"
}

Tipo_Escenario<- data.frame(Tipo_Escenario=Escenarios, Lambda=lambda)

set.seed(57)
t<- 1440*Días
clientes<- round(rpois(1:100,lambda)) #Dist Poisson
nclientes<- NULL
data<- NULL
Pacientes<- 0
Pos<- 0
si<-0
Tiempo_Entre_Llegadas<- 0
Momento_Llegada<- 0
for (i in 1:t) {
  #Para cada minuto que pase cuantos clientes llegan
  nclientes[i]<- c(sample(clientes,1))
  Pos[i]<- i
  #TIEMPO ENTRE LLEGADAS DE LOS DIFERENTES PACIENTES
  #Para saber que clientes llegan en cada minuto. MOMENTO DE LLEGADA
  si[i]<- nclientes[i]
  if (i==1){
    if (si[i]>1){
      Momento_Llegada[i:si[i]]<- Pos[i]
    }
  }
  if(i>1){
    if (si[i]>1){
      Momento_Llegada[length(Momento_Llegada)+nclientes[i-1]+1:si[i]]<
- Pos[i]
    }
    if (si[i]==1){
      Momento_Llegada[length(Momento_Llegada)+nclientes[i-1]+si[i]]<-
Pos[i]
    }
  }
```

```
  Momento_Llegada<- na.omit(Momento_Llegada)
}
  #Tiempo entre llegadas
  for (k in 1:length(Momento_Llegada)) {
    Tiempo_Entre_Llegadas[1]<- 0
    if(k>1){
      Tiempo_Entre_Llegadas[k]<- Momento_Llegada[k]-Momento_Llegada[k-
1]
      i<- i +1
    }
  }
  Tiempo_Entre_Llegadas
#Declaración de nuevas variables
Clientes_prioridad<- NULL
Clientes_prioridad2<- NULL
Clientes_prioridad3<- NULL
Pacientes<- 1:sum(nclientes)
Primer_Step<- 0
Segundo_Step<- 0
Segundo_Step_QUI<- 0
Segundo_Step_CRI<- 0
Segundo_Step_UCI<- 0

#VARIBALES TRIAJE

contadorTRI<- 0; P_Triaje<- 0; Momento_Terminación_Triaje<- 0; Tiempo_
Espera_Triaje<- 0
Tiempo_servicio_Triaje<- 0; Llegada_Triaje<- 0; minimo_Triaje<- 0
reajueste_Triaje<- 0; Tiempo_reajustado_Triaje<- 0; Tiempo_para_reajus
te_Triaje<- 0

#VARIABLES CONSULTA

contadorCon<- 0; P_Consulta<- 0; Momento_Terminación_Consulta<- 0; Tie
mpo_Espera_Consulta<- 0
Tiempo_servicio_Consulta<- 0; Llegada_Consulta<- 0; minimo_Consulta<-
0
reajueste_Consulta<- 0; Tiempo_reajustado_Consulta<- 0; Tiempo_para_re
ajuste_Consulta<- 0

contador_pacientes_Con<-0; Pacientes_Cola_Con<- 0

#VARIABLES EXPLORACIÓN

contadorExp<- 0; P_Exploracion<- 0; Momento_Terminación_Exploracion<-
0; Tiempo_Espera_Exploracion<- 0
Tiempo_servicio_Exploracion<- 0; Llegada_Exploracion<- 0; minimo_Explo
racion<- 0
reajueste_Exploracion<- 0; Tiempo_reajustado_Exploracion<- 0; Tiempo_p
ara_reajuste_Exploracion<- 0

#VARIABLES OBSERVACIÓN

contadorObs<- 0; P_Observacion<- 0; Momento_Terminación_Observacion<-
```

```
0; Tiempo_Espera_Observacion<- 0
Tiempo_servicio_Observacion<- 0; Llegada_Observacion<- 0; minimo_Obser
vacion<- 0
reajueste_Observacion<- 0; Tiempo_reajustado_Observacion<- 0; Tiempo_p
ara_reajuste_Observacion<- 0

c<- 0; c2<- 0; v<- 0; L_Obs<- 0; Momento_Llegada1<- 0; P_O<- 0; P_O2<-
0; Nodo_anterior_Obs1<- 0
Nodo_anterior_Obs2<- 0; Nodo_anterior_Obs3<- 0

#VARIABLES RCP

contadorRCP<- 0; P_RCP<- 0; Momento_Terminación_RCP<- 0; Tiempo_Espera
_RCP<- 0
Tiempo_servicio_RCP<- 0; Llegada_RCP<- 0; minimo_RCP<- 0
reajueste_RCP<- 0; Tiempo_reajustado_RCP<- 0; Tiempo_para_reajuste_RCP
<- 0

#VARIABLES QUIROFANO
contadorQUI<- 0; Llegada_QUI<- 0; Nodo_anterior_QUI<- 0; P_QUI<- 0

contadorQUI2<-0; P_QUI2<- 0; Momento_Terminación_QUI2<- 0; Tiempo_Espe
ra_QUI2<- 0
Tiempo_servicio_QUI2<- 0; Llegada_QUI2<- 0; minimo_QUI2<- 0
reajueste_QUI2<- 0; Tiempo_reajustado_QUI2<- 0; Tiempo_para_reajuste_Q
UI2<- 0
p_qui<- 0; Obs_QUI_SPLIT<- 0; nada_split<-0

#VARIABLES CRÍTICOS
contadorCRI<- 0; Llegada_CRI<- 0; Nodo_anterior_CRI<- 0; P_CRI<- 0

contadorCRI2<-0; P_CRI2<- 0; Momento_Terminación_CRI2<- 0; Tiempo_Espe
ra_CRI2<- 0
Tiempo_servicio_CRI2<- 0; Llegada_CRI2<- 0; minimo_CRI2<- 0
reajueste_CRI2<- 0; Tiempo_reajustado_CRI2<- 0; Tiempo_para_reajuste_C
RI2<- 0
p_cri<- 0; Obs_CRI_SPLIT<- 0

contadorCRI3<- 0; P_CRI3<- 0; Llegada_CRI3<- 0; Nodo_anterior_CRI3<- 0
QUI_CRI_SPLIT<- 0; anterior_CRI<- 0

#VARIABLES UCI
contadorUCI<- 0; Llegada_UCI<- 0; Nodo_anterior_UCI<- 0; P_UCI<- 0

contadorUCI2<-0; P_UCI2<- 0; Momento_Terminación_UCI2<- 0; Tiempo_Espe
ra_UCI2<- 0
Tiempo_servicio_UCI2<- 0; Llegada_UCI2<- 0; minimo_UCI2<- 0
reajueste_UCI2<- 0; Tiempo_reajustado_UCI2<- 0; Tiempo_para_reajuste_U
CI2<- 0
p_uci<- 0; Obs_UCI_SPLIT<- 0

contadorUCI3<- 0; P_UCI3<- 0; Llegada_UCI3<- 0; Nodo_anterior_UCI3<- 0
QUI_UCI_SPLIT<- 0; anterior_UCI<-0
```

```
#VARIABLES ALTA
contadorALT<- 0; Llegada_ALT<- 0; Nodo_anterior_ALT<- 0; P_ALT<- 0
Obs_ALT_SPLIT<- 0; Momento_Llegada_Incial_ALT<- 0

#VARIABLES PLANTA
contadorPLA<- 0; Llegada_PLA<- 0; Nodo_anterior_PLA<- 0; P_PLA<- 0

contadorPLA2<-0; P_PLA2<- 0; Momento_Terminación_PLA2<- 0; Tiempo_Espe
ra_PLA2<- 0
Tiempo_servicio_PLA2<- 0; Llegada_PLA2<- 0; minimo_PLA2<- 0
reajueste_PLA2<- 0; Tiempo_reajustado_PLA2<- 0; Tiempo_para_reajuste_P
LA2<- 0
p_pla<- 0

P_OBS_PLA<- 0; MOMENTO_OBS_PLA<- 0; NODO_OBS_PLA<- 0; contadorOBS_PLA<
- 0
P_QUI_PLA<- 0; MOMENTO_QUI_PLA<- 0; NODO_QUI_PLA<- 0; contadorQUI_PLA<
- 0

Obs_PLA_SPLIT<- 0; QUI_PLA_SPLIT<-0; Disponible_PLA2<-0
######################## EMPIEZAN LOS CÁLCULOS ###############
######################## CÁLCULOS PRIMER STEP ##################
#SE EMPIEZA REALIZANDO EL TRIAJE.
#Con: Consulta; Exp: Exploracion; Obs: Observación; RCP: RCP
#SIENDO LA PRIMERA LA MENOS GRAVE Y LA ÚLTIMA LA MÁS GRAVE.
for (j in 1:sum(nclientes)) {
  contadorTRI<- contadorTRI+1
  Llegada_Triaje[contadorTRI]<- Momento_Llegada[j]

  P_Triaje[j]<- Pacientes[j]

  Tiempo_servicio_Triaje[contadorTRI]<- runif(1, min = 1, max = 5)
  {
    if (contadorTRI<=Disponible_Tri){
      Momento_Terminación_Triaje[contadorTRI]<- na.omit(Momento_Llegad
a[j] + Tiempo_servicio_Triaje[contadorTRI])
      Tiempo_Espera_Triaje[contadorTRI]<- 0
      reajueste_Triaje[contadorTRI]<- Momento_Terminación_Triaje[conta
dorTRI]
    }
    else{
      minimo_Triaje<-min(reajueste_Triaje)
      ordenar_Triaje<- order(minimo_Triaje,decreasing = T)[1]

      if(Momento_Llegada[j]==Momento_Llegada[ordenar_Triaje]){
        Tiempo_Espera_Triaje[contadorTRI]=minimo_Triaje
      }
      else{
        Tiempo_Espera_Triaje[contadorTRI]<- minimo_Triaje - Momento_Ll
egada[j]
        if(Tiempo_Espera_Triaje[contadorTRI]<0){
          Tiempo_Espera_Triaje[contadorTRI]<- 0
        }
      }
```

```
    }
    Momento_Terminación_Triaje[contadorTRI]<- Momento_Llegada[j]+Tiemp
o_Espera_Triaje[contadorTRI]+Tiempo_servicio_Triaje[contadorTRI]
    Tiempo_para_reajuste_Triaje<- reajueste_Triaje
    Tiempo_reajustado_Triaje<- replace(Tiempo_para_reajuste_Triaje,Tie
mpo_para_reajuste_Triaje==minimo_Triaje ,Momento_Terminación_Triaje[co
ntadorTRI])
    reajueste_Triaje<- Tiempo_reajustado_Triaje
  }
  #Para cada cliente que entra en el sistema, indicamos su prioridad,
esto es la ruta del primer paso de cada paciente que entra al sistema
  Clientes_prioridad[j]<- sample(c("Con","Exp","Obs","RCP"), size = 1,
replace = FALSE, prob = c(0.9,0.03,0.06,0.01))
}
#PARA VER TODOS LOS PACIENTES DE TRIAJE EN CADA UNO DE LOS SERVIDORES
Pacientes_Triaje<- data.frame(P_Triaje=P_Triaje,Llegada_Triaje=Llegada
_Triaje,
                              Tiempo_Espera_Triaje=Tiempo_Espera_Triaj
e,
                              Tiempo_servicio_Triaje=Tiempo_servicio_T
riaje,
                              Momento_Terminación_Triaje=Momento_Termi
nación_Triaje)
#Pacientes_Triaje
##########################################################################
###
for (j in 1:sum(nclientes)) {
########################## PRIORIDAD Con --> CONSULTA ############
    if (Clientes_prioridad[j]=="Con"){
      contadorCon<- contadorCon +1 #para poder realizar los cálculos
      Llegada_Consulta[contadorCon]<- Pacientes_Triaje$Momento_Termina
ción_Triaje[j]
      Primer_Step[j]<- "Con"

      P_Consulta[j]<- Pacientes_Triaje$P_Triaje[j]
      P_Consulta[P_Consulta==0]<- NA
      P_Consulta<- na.omit(P_Consulta)

      Tiempo_servicio_Consulta[contadorCon]<- runif(1, min = 15, max =
30) #rtriangle(1,15,30)
      {
        if (contadorCon<=Disponible_Con){
          Momento_Terminación_Consulta[contadorCon]<- na.omit(Paciente
s_Triaje$Momento_Terminación_Triaje[j] + Tiempo_servicio_Consulta[cont
adorCon])
          Tiempo_Espera_Consulta[contadorCon]<- 0
          reajueste_Consulta[contadorCon]<- Momento_Terminación_Consul
ta[contadorCon]
        }
        else{
          minimo_Consulta<-min(reajueste_Consulta)
          ordenar_Consulta<- order(minimo_Consulta,decreasing = T)[1]
          if(Pacientes_Triaje$Momento_Terminación_Triaje[j]==Pacientes
_Triaje$Momento_Terminación_Triaje[ordenar_Consulta]){
```

```
            Tiempo_Espera_Consulta[contadorCon]=minimo_Consulta
          }
          else{
            Tiempo_Espera_Consulta[contadorCon]<- minimo_Consulta - Pa
cientes_Triaje$Momento_Terminación_Triaje[j]
            if(Tiempo_Espera_Consulta[contadorCon]<0){
              Tiempo_Espera_Consulta[contadorCon]<- 0
            }
          }
        }
        Momento_Terminación_Consulta[contadorCon]<- Pacientes_Triaje$M
omento_Terminación_Triaje[j]+Tiempo_Espera_Consulta[contadorCon]+Tiemp
o_servicio_Consulta[contadorCon]
        Tiempo_para_reajuste_Consulta<- reajueste_Consulta
        Tiempo_reajustado_Consulta<- replace(Tiempo_para_reajuste_Cons
ulta, Tiempo_para_reajuste_Consulta==minimo_Consulta ,Momento_Terminac
ión_Consulta[contadorCon])
        reajueste_Consulta<- Tiempo_reajustado_Consulta
      }
      #INDICAMOS LOS PACIENTES DE PRIORIDAD Con DONDE IRAN DESPUES:
      # ALT: ALTA, OBS: OBSERVACION, UCI: UCI, CRI: CRITICOS, QUI: QUI
ROFANO
      Clientes_prioridad2[j]<- sample(c("ALT","OBS","UCI","CRI","QUI")
, size = 1,replace = FALSE, prob = c(0.85,0.1,0.01,0.02,0.02))
    }
###################### PRIORIDAD Exp --> EXPLORACIÓN ################
#
    if (Clientes_prioridad[j]=="Exp"){
      contadorExp<- contadorExp +1 #para poder realizar los cálculos
      Llegada_Exploracion[contadorExp]<-Pacientes_Triaje$Momento_Termi
nación_Triaje[j]
      Primer_Step[j]<- "Exp"

      P_Exploracion[j]<- Pacientes_Triaje$P_Triaje[j]
      P_Exploracion[P_Exploracion==0]<- NA
      P_Exploracion<- na.omit(P_Exploracion)

      Tiempo_servicio_Exploracion[contadorExp]<- runif(1, min = 10, ma
x = 20) #rtriangle(1,10,20)
      {
      if (contadorExp<=Disponible_Exp){
        Momento_Terminación_Exploracion[contadorExp]<- na.omit(Pacient
es_Triaje$Momento_Terminación_Triaje[j] + Tiempo_servicio_Exploracion[
contadorExp])
        Tiempo_Espera_Exploracion[contadorExp]<- 0
        reajueste_Exploracion[contadorExp]<- Momento_Terminación_Explo
racion[contadorExp]
      }
      else{
        minimo_Exploracion<-min(reajueste_Exploracion)
        ordenar_Exploracion<- order(minimo_Exploracion,decreasing = T)
[1]
        if(Pacientes_Triaje$Momento_Terminación_Triaje[j]==Pacientes_T
riaje$Momento_Terminación_Triaje[ordenar_Exploracion]){
```

```
          Tiempo_Espera_Exploracion[contadorExp]=minimo_Exploracion
        }
        else{
          Tiempo_Espera_Exploracion[contadorExp]<- minimo_Exploracion-
Pacientes_Triaje$Momento_Terminación_Triaje[j]
          if(Tiempo_Espera_Exploracion[contadorExp]<0){
            Tiempo_Espera_Exploracion[contadorExp]<- 0
          }
        }
      }
        Momento_Terminación_Exploracion[contadorExp]<- Pacientes_Triaj
e$Momento_Terminación_Triaje[j]+Tiempo_Espera_Exploracion[contadorExp]
+Tiempo_servicio_Exploracion[contadorExp]
        Tiempo_para_reajuste_Exploracion<- reajueste_Exploracion
        Tiempo_reajustado_Exploracion<-replace(Tiempo_para_reajuste_Ex
ploracion, Tiempo_para_reajuste_Exploracion==minimo_Exploracion ,Momen
to_Terminación_Exploracion[contadorExp])
        reajueste_Exploracion<- Tiempo_reajustado_Exploracion
      }
      #INDICAMOS LOS PACIENTES DE PRIORIDAD Exp DONDE IRAN DESPUES:
      # UCI: UCI, CRI: CRITICOS, QUI: QUIROFANO
      Clientes_prioridad2[j]<- sample(c("UCI","CRI","QUI"), size = 1,r
eplace = FALSE, prob = c(0.35,0.4,0.25))
    }
#################### PRIORIDAD RCP --> RCP #################
  if (Clientes_prioridad[j]=="RCP"){
    contadorRCP<- contadorRCP +1 #para poder realizar los cálculos
    Llegada_RCP[contadorRCP]<- Pacientes_Triaje$Momento_Terminación_Tr
iaje[j]
    Primer_Step[j]<- "RCP"
    P_RCP[j]<- Pacientes_Triaje$P_Triaje[j]
    P_RCP[P_RCP==0]<- NA
    P_RCP<- na.omit(P_RCP)
    Tiempo_servicio_RCP[contadorRCP]<- runif(1, min = 10, max = 20) #r
triangle(1,10,20)
    {
    if (contadorRCP<=Disponible_RCP){
      Momento_Terminación_RCP[contadorRCP]<- na.omit(Pacientes_Triaje$
Momento_Terminación_Triaje[j] + Tiempo_servicio_RCP[contadorRCP])
      Tiempo_Espera_RCP[contadorRCP]<- 0
      reajueste_RCP[contadorRCP]<- Momento_Terminación_RCP[contadorRCP
]
    }
    else{
      minimo_RCP<-min(reajueste_RCP)
      ordenar_RCP<- order(reajueste_RCP,decreasing = T)[1]
      {
        if(Pacientes_Triaje$Momento_Terminación_Triaje[j]==Pacientes_T
riaje$Momento_Terminación_Triaje[ordenar_RCP]){
          Tiempo_Espera_RCP[contadorRCP]=minimo_RCP
        }
        else{
          Tiempo_Espera_RCP[contadorRCP]<- minimo_RCP- Pacientes_Triaj
e$Momento_Terminación_Triaje[j]
```

```
          if(Tiempo_Espera_RCP[contadorRCP]<0){
            Tiempo_Espera_RCP[contadorRCP]<- 0
          }
        }
      }
    }
    Momento_Terminación_RCP[contadorRCP]<- Pacientes_Triaje$Momento_Te
rminación_Triaje[j]+Tiempo_Espera_RCP[contadorRCP]+Tiempo_servicio_RCP
[contadorRCP]
    Tiempo_para_reajuste_RCP<- reajueste_RCP
    Tiempo_reajustado_RCP<- replace(Tiempo_para_reajuste_RCP, Tiempo_p
ara_reajuste_RCP==minimo_RCP ,Momento_Terminación_RCP[contadorRCP])
    reajueste_RCP<- Tiempo_reajustado_RCP
  }
  #INDICAMOS LOS PACIENTES DE PRIORIDAD RCP DONDE IRAN DESPUES:
  # UCI: UCI, CRI: CRITICOS, QUI: QUIROFANO
  Clientes_prioridad2[j]<- sample(c("UCI","CRI","QUI"), size = 1,repla
ce = FALSE, prob = c(0.25,0.6,0.15))
  }
# CÁLCULAMOS LOS CLIENTES PRIORIDAD2 PARA OBSERVACIÓN, PARA QUE NO DE
ERROR A LA HORA DE CALCULAR LOS MOMENTOS DE LLEGADA Y TERMINACIÓN DE O
BSERVACIÓN
  if (Clientes_prioridad[j]=="Obs"){
  #INDICAMOS LOS PACIENTES DE PRIORIDAD Obs DONDE IRAN DESPUES:
  # UCI: UCI, CRI: CRITICOS, QUI: QUIROFANO, ALT: ALTA, PLA: PLANTA
    Clientes_prioridad2[j]<- sample(c("UCI","CRI","QUI","ALT","PLA"),
size = 1,replace = FALSE, prob = c(0.01,0.04,0.1,0.7,0.15))
    Primer_Step[j]<- "Obs"
    }
}
###### CÁLCULAMOS LOS MOMENTOS DE LLEGADA A OBSERVACIÓN ##############
for (j in 1:sum(nclientes)) {
    if(Clientes_prioridad2[j]=="OBS"){
      c<- c +1
      P_O[c]<- Pacientes[j]
      P_O[P_O==0]<- NA
      P_O<- na.omit(P_O)
      L_Obs[c]<- Momento_Terminación_Consulta[which(P_Consulta== P_O[c
])]
      L_Obs[L_Obs==0]<- NA
      L_Obs<- na.omit(L_Obs)
      Nodo_anterior_Obs2[c]<- "CONSULTA"
      Clientes_prioridad2[j]<- sample(c("UCI","CRI","QUI","ALT","PLA")
, size = 1,replace = FALSE, prob = c(0.01,0.04,0.1,0.7,0.15))
      Primer_Step[j]<- "Obs"
    }
    else{
      if (Clientes_prioridad[j]=="Obs"){
        c2<- c2 +1
        Momento_Llegada1[c2]<- Pacientes_Triaje$Momento_Terminación_Tr
iaje[j]
        Momento_Llegada1[Momento_Llegada1==0]<- NA
        Momento_Llegada1<- na.omit(Momento_Llegada1)
        P_O2[c2]<- Pacientes[j]
```

```
        P_O2[P_O2==0]<- NA
        P_O2<- na.omit(P_O2)
        Nodo_anterior_Obs3[c2]<- "NINGUNO"
      }
    }
  v<- c(Momento_Llegada1,L_Obs)
  P_Observacion1<- c(P_O2,P_O)
  Nodo_anterior_Obs1<- c(Nodo_anterior_Obs3,Nodo_anterior_Obs2)
}
Momento_Llegada1<- v
Pacientes_Observacion1<- data.frame(P_Observacion=P_Observacion1,Momen
to_Llegada1=Momento_Llegada1,Nodo_anterior_Obs=Nodo_anterior_Obs1)
Pacientes_Observacion1<- Pacientes_Observacion1[with(Pacientes_Observa
cion1, order(Pacientes_Observacion1$Momento_Llegada1)), ]
#################### PRIORIDAD Obs --> OBSERVACIÓN #############
for (j in 1:length(Pacientes_Observacion1$P_Observacion)) {
############ PRIORIDAD Obs --> OBSERVACIÓN ############################
  contadorObs<- contadorObs + 1 #para poder realizar los cálculos
  Tiempo_servicio_Observacion[contadorObs]<- runif(1, min = 15, max =
45) #rtriangle(1,15,45)
  {
    if (contadorObs<=Disponible_Obs){
      Momento_Terminación_Observacion[contadorObs]<- na.omit(Pacientes
_Observacion1$Momento_Llegada1[contadorObs] + Tiempo_servicio_Observac
ion[contadorObs])
      Tiempo_Espera_Observacion[contadorObs]<- 0
      reajueste_Observacion[contadorObs]<- Momento_Terminación_Observa
cion[contadorObs]
    }
    else{
      minimo_Observacion<-min(reajueste_Observacion)
      ordenar_Observacion<- order(minimo_Observacion,decreasing = T)[1
]
      if(Pacientes_Observacion1$Momento_Llegada1[contadorObs]==Pacient
es_Observacion1$Momento_Llegada1[ordenar_Observacion]){
        Tiempo_Espera_Observacion[contadorObs]=minimo_Observacion
      }
      else{
        Tiempo_Espera_Observacion[contadorObs]<- minimo_Observacion- P
acientes_Observacion1$Momento_Llegada1[contadorObs]
        if(Tiempo_Espera_Observacion[contadorObs]<0){
          Tiempo_Espera_Observacion[contadorObs]<- 0
        }
      }
    }
    Momento_Terminación_Observacion[contadorObs]<- Pacientes_Observaci
on1$Momento_Llegada1[contadorObs]+Tiempo_Espera_Observacion[contadorOb
s]+Tiempo_servicio_Observacion[contadorObs]
    Tiempo_para_reajuste_Observacion<- reajueste_Observacion
    Tiempo_reajustado_Observacion<- replace(Tiempo_para_reajuste_Obser
vacion, Tiempo_para_reajuste_Observacion==minimo_Observacion ,Momento_
Terminación_Observacion[contadorObs])
    reajueste_Observacion<- Tiempo_reajustado_Observacion
  }
```

```
}
########################## RESULTADOS PRIMER STEP ##################
#
#PARA VER TODOS LOS PACIENTES DE PRIOPRIDAD Con EN CADA UNO DE LOS SER
VIDORES
Pacientes_Consulta<- data.frame(P_Consulta=P_Consulta,Llegada_Consulta
=Llegada_Consulta,Tiempo_Espera_Consulta=Tiempo_Espera_Consulta,Tiempo
_servicio_Consulta=Tiempo_servicio_Consulta,Momento_Terminación_Consul
ta=Momento_Terminación_Consulta)
#Pacientes_Consulta
#PARA VER TODOS LOS PACIENTES DE PRIOPRIDAD Exp EN CADA UNO DE LOS SER
VIDORES
Pacientes_Exploracion<-data.frame(P_Exploracion=P_Exploracion,Llegada_
Exploracion=Llegada_Exploracion,Tiempo_Espera_Exploracion=Tiempo_Esper
a_Exploracion, Tiempo_servicio_Exploracion=Tiempo_servicio_Exploracion
, Momento_Terminación_Exploracion=Momento_Terminación_Exploracion)
#Pacientes_Exploracion
#PARA VER TODOS LOS PACIENTES DE PRIOPRIDAD Obs EN CADA UNO DE LOS SER
VIDORES
Pacientes_Observacion<-data.frame(P_Observacion=Pacientes_Observacion1
$P_Observacion ,Llegada_Observacion=Pacientes_Observacion1$Momento_Lle
gada1,            Tiempo_Espera_Observacion=Tiempo_Espera_Observacion,
Tiempo_servicio_Observacion=Tiempo_servicio_Observacion,
Momento_Terminación_Observacion=Momento_Terminación_Observacion,Nodo_a
nterior_Obs1=Pacientes_Observacion1$Nodo_anterior_Obs)
#Pacientes_Observacion
#PARA VER TODOS LOS PACIENTES DE PRIOPRIDAD RCP EN CADA UNO DE LOS SER
VIDORES
Pacientes_RCP<- data.frame(P_RCP=P_RCP,Llegada_RCP=Llegada_RCP,Tiempo_
Espera_RCP=Tiempo_Espera_RCP,Tiempo_servicio_RCP=Tiempo_servicio_RCP,
Momento_Terminación_RCP=Momento_Terminación_RCP)
#Pacientes_RCP
######################## MOMENTOS LLEGADA SEGUNDO STEP ##########
for (q in 1:sum(nclientes)) {

########################## MOMENTO LLEGADA UCI ################
    if (Clientes_prioridad2[q]=="UCI"){
      contadorUCI<- contadorUCI +1 #para poder realizar los cálculos
      {
      if(Primer_Step[q]=="Con"){
        P_UCI[contadorUCI]<- Pacientes[q]
        Llegada_UCI[contadorUCI]<- as.numeric(Pacientes_Consulta$Momen
to_Terminación_Consulta[which(Pacientes_Consulta$P_Consulta==P_UCI[con
tadorUCI])])
        Nodo_anterior_UCI[contadorUCI]<- "Con"
        Obs_UCI_SPLIT[contadorUCI]<- "NADA"
      }
      if(Primer_Step[q]=="Exp") {
        P_UCI[contadorUCI]<- Pacientes[q]
        Llegada_UCI[contadorUCI]<- as.numeric(Pacientes_Exploracion$Mo
mento_Terminación_Exploracion[which(Pacientes_Exploracion$P_Exploracio
n==P_UCI[contadorUCI])])
        Nodo_anterior_UCI[contadorUCI]<- "Exp"
        Obs_UCI_SPLIT[contadorUCI]<- "NADA"
```

```
      }
      if(Primer_Step[q]=="Obs"){
        P_UCI[contadorUCI]<- Pacientes[q]
        Llegada_UCI[contadorUCI]<- as.numeric(Pacientes_Observacion$Mo
mento_Terminación_Observacion[which(Pacientes_Observacion$P_Observacio
n==P_UCI[contadorUCI])])
        Nodo_anterior_UCI[contadorUCI]<- "Obs"
        Obs_UCI_SPLIT[contadorUCI]<- Pacientes_Observacion$Nodo_anteri
or_Obs1[which(Pacientes_Observacion$P_Observacion==P_UCI[contadorUCI])
]
      }
      if(Primer_Step[q]=="RCP"){
        P_UCI[contadorUCI]<- Pacientes[q]
        Llegada_UCI[contadorUCI]<- as.numeric(Pacientes_RCP$Momento_Te
rminación_RCP[which(Pacientes_RCP$P_RCP==P_UCI[contadorUCI])])
        Nodo_anterior_UCI[contadorUCI]<- "RCP"
        Obs_UCI_SPLIT[contadorUCI]<- "NADA"
      }
      }
    }
###################### MOMENTO LLEGADA CRITICOS ####################
  if (Clientes_prioridad2[q]=="CRI"){
    contadorCRI<- contadorCRI +1 #para poder realizar los cálculos
    {
      if(Primer_Step[q]=="Con"){
        P_CRI[contadorCRI]<- Pacientes[q]
        Llegada_CRI[contadorCRI]<- as.numeric(Pacientes_Consulta$Momen
to_Terminación_Consulta[which(Pacientes_Consulta$P_Consulta==P_CRI[con
tadorCRI])])
        Nodo_anterior_CRI[contadorCRI]<- "Con"
        Obs_CRI_SPLIT[contadorCRI]<- "NADA"
      }
      if(Primer_Step[q]=="Exp") {
        P_CRI[contadorCRI]<- Pacientes[q]
        Llegada_CRI[contadorCRI]<- as.numeric(Pacientes_Exploracion$Mo
mento_Terminación_Exploracion[which(Pacientes_Exploracion$P_Exploracio
n==P_CRI[contadorCRI])])
        Nodo_anterior_CRI[contadorCRI]<- "Exp"
        Obs_CRI_SPLIT[contadorCRI]<- "NADA"
      }
      if(Primer_Step[q]=="Obs"){
        P_CRI[contadorCRI]<- Pacientes[q]
        Llegada_CRI[contadorCRI]<- as.numeric(Pacientes_Observacion$Mo
mento_Terminación_Observacion[which(Pacientes_Observacion$P_Observacio
n==P_CRI[contadorCRI])])
        Nodo_anterior_CRI[contadorCRI]<- "Obs"
        Obs_CRI_SPLIT[contadorCRI]<- Pacientes_Observacion$Nodo_anteri
or_Obs1[which(Pacientes_Observacion$P_Observacion==P_CRI[contadorCRI])
]
      }
      if(Primer_Step[q]=="RCP"){
        P_CRI[contadorCRI]<- Pacientes[q]
        Llegada_CRI[contadorCRI]<- as.numeric(Pacientes_RCP$Momento_Te
rminación_RCP[which(Pacientes_RCP$P_RCP==P_CRI[contadorCRI])])
```

```
        Nodo_anterior_CRI[contadorCRI]<- "RCP"
        Obs_CRI_SPLIT[contadorCRI]<- "NADA"
      }
    }
  }
#################### MOMENTO LLEGADA QUIROFANO ##################
  if (Clientes_prioridad2[q]=="QUI"){
    contadorQUI<- contadorQUI +1 #para poder realizar los cálculos
    {
      if(Primer_Step[q]=="Con"){
        P_QUI[contadorQUI]<- Pacientes[q]
        Llegada_QUI[contadorQUI]<- as.numeric(Pacientes_Consulta$Momen
to_Terminación_Consulta[which(Pacientes_Consulta$P_Consulta==P_QUI[con
tadorQUI])])
        Nodo_anterior_QUI[contadorQUI]<- "Con"
        Obs_QUI_SPLIT[contadorQUI]<- "NADA"
      }
      if(Primer_Step[q]=="Exp") {
        P_QUI[contadorQUI]<- Pacientes[q]
        Llegada_QUI[contadorQUI]<- as.numeric(Pacientes_Exploracion$Mo
mento_Terminación_Exploracion[which(Pacientes_Exploracion$P_Exploracio
n==P_QUI[contadorQUI])])
        Nodo_anterior_QUI[contadorQUI]<- "Exp"
        Obs_QUI_SPLIT[contadorQUI]<- "NADA"
      }
      if(Primer_Step[q]=="Obs"){
        P_QUI[contadorQUI]<- Pacientes[q]
        Llegada_QUI[contadorQUI]<- as.numeric(Pacientes_Observacion$Mo
mento_Terminación_Observacion[which(Pacientes_Observacion$P_Observacio
n==P_QUI[contadorQUI])])
        Nodo_anterior_QUI[contadorQUI]<- "Obs"
        Obs_QUI_SPLIT[contadorQUI]<- Pacientes_Observacion$Nodo_anteri
or_Obs1[which(Pacientes_Observacion$P_Observacion==P_QUI[contadorQUI])
]
      }
      if(Primer_Step[q]=="RCP"){
        P_QUI[contadorQUI]<- Pacientes[q]
        Llegada_QUI[contadorQUI]<- as.numeric(Pacientes_RCP$Momento_Te
rminación_RCP[which(Pacientes_RCP$P_RCP==P_QUI[contadorQUI])])
        Nodo_anterior_QUI[contadorQUI]<- "RCP"
        Obs_QUI_SPLIT[contadorQUI]<- "NADA"
      }
    }
  }
  ################################## MOMENTO LLEGADA ALTA ##########
  if (Clientes_prioridad2[q]=="ALT"){
    contadorALT<- contadorALT +1 #para poder realizar los cálculos
    if(Primer_Step[q]=="Con"){
      P_ALT[contadorALT]<- Pacientes[q]
      Llegada_ALT[contadorALT]<- as.numeric(Pacientes_Consulta$Momento
_Terminación_Consulta[which(Pacientes_Consulta$P_Consulta==P_ALT[conta
dorALT])])
      Nodo_anterior_ALT[contadorALT]<- "Con"
      Obs_ALT_SPLIT[contadorALT]<- "NADA"
```

```
      Momento_Llegada_Incial_ALT[contadorALT]<- Pacientes_Triaje$Momen
to_Terminación_Triaje[q]
    }
    if(Primer_Step[q]=="Obs"){
      P_ALT[contadorALT]<- Pacientes[q]
      Llegada_ALT[contadorALT]<- as.numeric(Pacientes_Observacion$Mome
nto_Terminación_Observacion[which(Pacientes_Observacion$P_Observacion=
=P_ALT[contadorALT])])
      Nodo_anterior_ALT[contadorALT]<- "Obs"
      Obs_ALT_SPLIT[contadorALT]<- Pacientes_Observacion$Nodo_anterior
_Obs1[which(Pacientes_Observacion$P_Observacion==P_ALT[contadorALT])]
      Momento_Llegada_Incial_ALT[contadorALT]<- Pacientes_Triaje$Momen
to_Terminación_Triaje[q]
    }
  }
}
################ MOMENTO LLEGADA SEGUNDO STEP ##################
#PARA VER LOS MOMENTOS DE LLEGADAS DE LOS PACIENTES A LA UCI(SEGUNDO S
TEP)
Momento_Llegada_UCI<- data.frame(P_UCI=P_UCI,Nodo_anterior_UCI=Nodo_an
terior_UCI,Llegada_UCI=Llegada_UCI,Obs_UCI_SPLIT=Obs_UCI_SPLIT)
Momento_Llegada_UCI<- Momento_Llegada_UCI[with(Momento_Llegada_UCI, or
der(Momento_Llegada_UCI$Llegada_UCI)), ]
#Momento_Llegada_UCI
#PARA VER LOS MOMENTOS DE LLEGADAS DE LOS PACIENTES A CRITICOS(SEGUNDO
STEP)
Momento_Llegada_CRI<- data.frame(P_CRI=P_CRI,Nodo_anterior_CRI=Nodo_an
terior_CRI,Llegada_CRI=Llegada_CRI,Obs_CRI_SPLIT=Obs_CRI_SPLIT)
Momento_Llegada_CRI<- Momento_Llegada_CRI[with(Momento_Llegada_CRI, or
der(Momento_Llegada_CRI$Llegada_CRI)), ]
#Momento_Llegada_CRI
#PARA VER LOS MOMENTOS DE LLEGADAS DE LOS PACIENTES A QUIROFANO(SEGUND
O STEP)
Momento_Llegada_QUI<- data.frame(P_QUI=P_QUI,Nodo_anterior_QUI=Nodo_an
terior_QUI,Llegada_QUI=Llegada_QUI,Obs_QUI_SPLIT=Obs_QUI_SPLIT)
Momento_Llegada_QUI<- Momento_Llegada_QUI[with(Momento_Llegada_QUI, or
der(Momento_Llegada_QUI$Llegada_QUI)), ]
#Momento_Llegada_QUI
#PARA VER LOS MOMENTOS DE LLEGADAS DE LOS PACIENTES A ALTA(SEGUNDO STE
P)
Momento_Llegada_ALT<- data.frame(P_ALT=P_ALT,Nodo_anterior_ALT=Nodo_an
terior_ALT,Llegada_ALT=Llegada_ALT,                                Obs_
ALT_SPLIT=Obs_ALT_SPLIT,Momento_Llegada_Incial_ALT=Momento_Llegada_Inc
ial_ALT)
Momento_Llegada_ALT<- Momento_Llegada_ALT[with(Momento_Llegada_ALT, or
der(Momento_Llegada_ALT$Llegada_ALT)), ]
#Momento_Llegada_ALT
############################ CÁLCULOS SEGUNDO STEP ###############
############## PRIORIDAD QUI --> QUIROFANO ###################
for (w in 1:contadorQUI){ #lo hacemos desde 1 hasta el numero de pacie
ntes que
  #vayan a quirofano
  contadorQUI2<- contadorQUI2 +1 #para saber cuantos servidores hay
  Llegada_QUI2[w]<- Momento_Llegada_QUI$Llegada_QUI[w]
```

```
  Segundo_Step_QUI[w]<- "QUI"
  nada_split[w]<- Momento_Llegada_QUI$Obs_QUI_SPLIT[w]

  #para saber de que pacientes se trata
  p_qui<- which(Momento_Llegada_QUI$Llegada_QUI==Momento_Llegada_QUI$L
legada_QUI[w])
  P_QUI2[w]<- Momento_Llegada_QUI$P_QUI[p_qui]

  Tiempo_servicio_QUI2[w]<- runif(1, min = 15, max = 30) #max(Días*10,
rnorm(1,20,5))
  {
    if (contadorQUI2<=Disponible_QUI){
      Momento_Terminación_QUI2[w]<- na.omit(Momento_Llegada_QUI$Llegad
a_QUI[w]+ Tiempo_servicio_QUI2[w])
      Tiempo_Espera_QUI2[w]<- 0
      reajueste_QUI2[w]<- Momento_Terminación_QUI2[w]
    }
    else{
      minimo_QUI2<-min(reajueste_QUI2)
      ordenar_QUI2<- order(minimo_QUI2,decreasing = T)[1]
      if(Momento_Llegada_QUI$Llegada_QUI[w]==Momento_Llegada_QUI$Llega
da_QUI[ordenar_QUI2]){
        Tiempo_Espera_QUI2[w]=minimo_QUI2
      }
      else{
        Tiempo_Espera_QUI2[w]<- minimo_QUI2 - Momento_Llegada_QUI$Lleg
ada_QUI[w]
        if(Tiempo_Espera_QUI2[w]<0){
          Tiempo_Espera_QUI2[w]<- 0
        }
      }
    }
    Momento_Terminación_QUI2[w]<- Momento_Llegada_QUI$Llegada_QUI[w] +
Tiempo_Espera_QUI2[w]+Tiempo_servicio_QUI2[w]
    Tiempo_para_reajuste_QUI2<- reajueste_QUI2
    Tiempo_reajustado_QUI2<- replace(Tiempo_para_reajuste_QUI2, Tiempo
_para_reajuste_QUI2==minimo_QUI2 ,Momento_Terminación_QUI2[w])
    reajueste_QUI2<- Tiempo_reajustado_QUI2
  }
  #INDICAMOS LOS PACIENTES DE PRIORIDAD QUI DONDE IRAN DESPUES:
  # UCI: UCI, CRI: CRITICOS, PLA: PLANTA
  Clientes_prioridad3[w]<- sample(c("UCI","CRI", "PLA"), size = 1,repl
ace = FALSE, prob = c(0.25,0.25,0.5))
}
################################## INCISO ####################
######### MOMENTOS DE LLEGADA A CRITICOS Y UCI DESDE EL QUIROFANO ####
for (o in 1:contadorQUI) {
  if(Clientes_prioridad3[o] =="UCI"){
    contadorUCI3<- contadorUCI3 + 1
    P_UCI3[contadorUCI3]<- P_QUI2[o]
    Llegada_UCI3[contadorUCI3]<- as.numeric(Momento_Terminación_QUI2[w
hich(P_QUI2==P_UCI3[contadorUCI3])])
    Nodo_anterior_UCI3[contadorUCI3]<- "QUI_P3"
    QUI_UCI_SPLIT[contadorUCI3]<- nada_split[which(P_QUI2==P_UCI3[cont
```

```
adorUCI3])]
  }
  if(Clientes_prioridad3[o] =="CRI"){
    contadorCRI3<- contadorCRI3 + 1
    P_CRI3[contadorCRI3]<- P_QUI2[o]
    Llegada_CRI3[contadorCRI3]<- as.numeric(Momento_Terminación_QUI2[w
hich(P_QUI2==P_CRI3[contadorCRI3])])
    Nodo_anterior_CRI3[contadorCRI3]<- "QUI_P3"
    QUI_CRI_SPLIT[contadorCRI3]<- nada_split[which(P_QUI2==P_CRI3[cont
adorCRI3])]
  }
}
# UNIMOS LOS MOMENTOS DE LLEGADAS CALCULADOS ANTERIORMENTE CON LOS NUE
VOS
# PACIENTES QUE VIENEN DESDE QUIROFANO
######################### MOMENTO LLEGADAS CRI ################
Momento_Llegada_CRI1<- data.frame(P_CRI=c(P_CRI,P_CRI3),Nodo_anterior_
CRI=c(Nodo_anterior_CRI,Nodo_anterior_CRI3),Llegada_CRI=c(Llegada_CRI,
Llegada_CRI3),SPLIT_CRI1=c(Obs_CRI_SPLIT,QUI_CRI_SPLIT))
Momento_Llegada_CRI1<- Momento_Llegada_CRI1[with(Momento_Llegada_CRI1,
order(Momento_Llegada_CRI1$Llegada_CRI)), ]
#Momento_Llegada_CRI1
######################### MOMENTO LLEGADAS UCI ##################
Momento_Llegada_UCI1<- data.frame(P_UCI=c(P_UCI,P_UCI3),Nodo_anterior_
UCI=c(Nodo_anterior_UCI,Nodo_anterior_UCI3),Llegada_UCI=c(Llegada_UCI,
Llegada_UCI3),SPLIT_UCI1=c(Obs_UCI_SPLIT,QUI_UCI_SPLIT))
Momento_Llegada_UCI1<- Momento_Llegada_UCI1[with(Momento_Llegada_UCI1,
order(Momento_Llegada_UCI1$Llegada_UCI)), ]
#Momento_Llegada_UCI1
########################## CONTINUAMOS CALCULANDO ################
#################### PRIORIDAD CRI --> CRITICOS ##################
for (e in 1:nrow(Momento_Llegada_CRI1)){ #lo hacemos desde 1 hasta el
numero de pacientes que vayan a criticos
  contadorCRI2<- contadorCRI2 +1 #para saber cuantos servidores hay
  Llegada_CRI2[e]<- Momento_Llegada_CRI1$Llegada_CRI[e]
  Segundo_Step_CRI[e]<- "CRI"
  anterior_CRI[e]<- Momento_Llegada_CRI1$Nodo_anterior_CRI[e]

  #para saber de que pacientes se trata
  p_cri<- which(Momento_Llegada_CRI1$Llegada_CRI==Momento_Llegada_CRI1
$Llegada_CRI[e])
  P_CRI2[e]<- Momento_Llegada_CRI1$P_CRI[p_cri]
  Tiempo_servicio_CRI2[e]<- runif(1, min = 1440, max = 4320) #max(Días
*1440,rnorm(1,1990,400))
  {
    if (contadorCRI2<=Disponible_CRI){
      Momento_Terminación_CRI2[e]<- na.omit(Momento_Llegada_CRI1$Llega
da_CRI[e]+ Tiempo_servicio_CRI2[e])
      Tiempo_Espera_CRI2[e]<- 0
      reajueste_CRI2[e]<- Momento_Terminación_CRI2[e]
    }
    else{
      minimo_CRI2<-min(reajueste_CRI2)
      ordenar_CRI2<- order(minimo_CRI2,decreasing = T)[1]
```

```
      if(Momento_Llegada_CRI1$Llegada_CRI[e]==Momento_Llegada_CRI1$Lle
gada_CRI[ordenar_CRI2]){
        Tiempo_Espera_CRI2[e]=minimo_CRI2
      }
      else{
        Tiempo_Espera_CRI2[e]<- minimo_CRI2 - Momento_Llegada_CRI1$Lle
gada_CRI[e]
        if(Tiempo_Espera_CRI2[e]<0){
          Tiempo_Espera_CRI2[e]<- 0
        }
      }
    }
    Momento_Terminación_CRI2[e]<- Momento_Llegada_CRI1$Llegada_CRI[e]
+ Tiempo_Espera_CRI2[e]+Tiempo_servicio_CRI2[e]
    Tiempo_para_reajuste_CRI2<- reajueste_CRI2
    Tiempo_reajustado_CRI2<- replace(Tiempo_para_reajuste_CRI2, Tiempo
_para_reajuste_CRI2==minimo_CRI2 ,Momento_Terminación_CRI2[e])
    reajueste_CRI2<- Tiempo_reajustado_CRI2
  }
}
#################### PRIORIDAD UCI --> UCI ########################
for (l in 1:nrow(Momento_Llegada_UCI1)){ #lo hacemos desde 1 hasta el
numero de pacientes que vayan a la uci
  contadorUCI2<- contadorUCI2 +1 #para saber cuantos servidores hay
  Llegada_UCI2[l]<- Momento_Llegada_UCI1$Llegada_UCI[l]
  Segundo_Step_UCI[l]<- "UCI"
  anterior_UCI[l]<- Momento_Llegada_UCI1$Nodo_anterior_UCI[l]

  #para saber de que pacientes se trata
  p_uci<- which(Momento_Llegada_UCI1$Llegada_UCI==Momento_Llegada_UCI1
$Llegada_UCI[l])
  P_UCI2[l]<- Momento_Llegada_UCI1$P_UCI[p_uci]
  Tiempo_servicio_UCI2[l]<- runif(1, min = 1900, max =5760) #max(Días*
1440,rnorm(1,2430,300))
  {
    if (contadorUCI2<=Disponible_UCI){
      Momento_Terminación_UCI2[l]<- na.omit(Momento_Llegada_UCI1$Llega
da_UCI[l]+ Tiempo_servicio_UCI2[l])
      Tiempo_Espera_UCI2[l]<- 0
      reajueste_UCI2[l]<- Momento_Terminación_UCI2[l]
    }
    else{
      minimo_UCI2<-min(reajueste_UCI2)
      ordenar_UCI2<- order(minimo_UCI2,decreasing = T)[1]
      if(Momento_Llegada_UCI1$Llegada_UCI[l]==Momento_Llegada_UCI1$Lle
gada_UCI[ordenar_UCI2]){
        Tiempo_Espera_UCI2[l]=minimo_UCI2
      }
      else{
        Tiempo_Espera_UCI2[l]<- minimo_UCI2 - Momento_Llegada_UCI1$Lle
gada_UCI[l]
        if(Tiempo_Espera_UCI2[l]<0){
          Tiempo_Espera_UCI2[l]<- 0
        }
```

```
      }
    }
    Momento_Terminación_UCI2[l]<- Momento_Llegada_UCI1$Llegada_UCI[l]
+ Tiempo_Espera_UCI2[l]+Tiempo_servicio_UCI2[l]
    Tiempo_para_reajuste_UCI2<- reajueste_UCI2
    Tiempo_reajustado_UCI2<- replace(Tiempo_para_reajuste_UCI2, Tiempo
_para_reajuste_UCI2==minimo_UCI2 ,Momento_Terminación_UCI2[l])
    reajueste_UCI2<- Tiempo_reajustado_UCI2
  }
}
#################### RESULTADOS SEGUNDO STEP ####################
#PARA VER TODOS LOS PACIENTES DE PRIOPRIDAD QUI EN CADA UNO DE LOS SER
VIDORES
Pacientes_QUI2<- data.frame(P_QUI2=P_QUI2,Llegada_QUI2=Llegada_QUI2,Ti
empo_Espera_QUI2=Tiempo_Espera_QUI2,Tiempo_servicio_QUI2=Tiempo_servic
io_QUI2,Momento_Terminación_QUI2=Momento_Terminación_QUI2)
#Pacientes_QUI2
#PARA VER TODOS LOS PACIENTES DE PRIOPRIDAD CRI EN CADA UNO DE LOS SER
VIDORES
Pacientes_CRI2<- data.frame(P_CRI2=P_CRI2,Llegada_CRI2=Llegada_CRI2,Ti
empo_Espera_CRI2=Tiempo_Espera_CRI2,Tiempo_servicio_CRI2=Tiempo_servic
io_CRI2,Momento_Terminación_CRI2=Momento_Terminación_CRI2,Anterior_CRI
2=anterior_CRI)
#Pacientes_CRI2
#PARA VER TODOS LOS PACIENTES DE PRIOPRIDAD UCI EN CADA UNO DE LOS SER
VIDORES
Pacientes_UCI2<- data.frame(P_UCI2=P_UCI2,Llegada_UCI2=Llegada_UCI2,Ti
empo_Espera_UCI2=Tiempo_Espera_UCI2,Tiempo_servicio_UCI2=Tiempo_servic
io_UCI2,Momento_Terminación_UCI2=Momento_Terminación_UCI2,Anterior_UCI
2=anterior_UCI,Llegada_Hora_Paciente=round(Llegada_UCI2/60,0),Salida_H
ora_Paciente=round(Momento_Terminación_UCI2/60,0))
#Pacientes_UCI2
####################### MOMENTOS LLEGADA TERCER STEP ##########
####################### MOMENTOS LLEGADA PLANTA ######################
#
#HACEMOS ESTE BUCLE PARA IDENTIFICAR LOS PACIENTES QUE VIENEN DE OBSER
VACIÓN A PLANTA
for (p in 1:sum(nclientes)) {
  if(Clientes_prioridad2[p]=="PLA"){
    contadorOBS_PLA<- contadorOBS_PLA + 1
    P_OBS_PLA[contadorOBS_PLA]<- Pacientes[p]
    MOMENTO_OBS_PLA[contadorOBS_PLA]<- as.numeric(Pacientes_Observacio
n$Momento_Terminación_Observacion[which(Pacientes_Observacion$P_Observ
acion==P_OBS_PLA[contadorOBS_PLA])])
    NODO_OBS_PLA[contadorOBS_PLA]<- "OBS"
    Obs_PLA_SPLIT[contadorOBS_PLA]<- Pacientes_Observacion$Nodo_anteri
or_Obs1[which(Pacientes_Observacion$P_Observacion==P_OBS_PLA[contadorO
BS_PLA])]
  }
}
#HACEMOS ESTE BUCLE PARA IDENTIFICAR LOS PACIENTES QUE VIENEN DE QUIRO
FANO A PLANTA
for (a in 1:contadorQUI) {
  if(Clientes_prioridad3[a]=="PLA"){
```

```
    contadorQUI_PLA<- contadorQUI_PLA + 1
    P_QUI_PLA[contadorQUI_PLA]<- P_QUI2[a]
    MOMENTO_QUI_PLA[contadorQUI_PLA]<- as.numeric(Pacientes_QUI2$Momen
to_Terminación_QUI2[which(Pacientes_QUI2$P_QUI2==P_QUI_PLA[contadorQUI
_PLA])])
    NODO_QUI_PLA[contadorQUI_PLA]<- "QUI_P3"
    QUI_PLA_SPLIT[contadorQUI_PLA]<- Momento_Llegada_QUI$Obs_QUI_SPLIT
[which(Momento_Llegada_QUI$P_QUI==P_QUI_PLA[contadorQUI_PLA])]
  }
}
#CREAMOS UNA MATRIZ PARA PODER CALCULAR LOS MOMENTOS DE LLEGADAS PARAR
EL TERCER STEP
longitud1<- length(Segundo_Step_CRI) + length(Segundo_Step_UCI) + leng
th(P_OBS_PLA) + length(P_QUI_PLA)

Momento_Llegada_PLA<- data.frame(P_PLA, Nodo_anterior_PLA= NA, Llegada
_PLA= longitud1, SPLIT=NA, ANTERIOR_P4=NA)

#PACIENTES DE CRITICOS
Momento_Llegada_PLA[1:length(Segundo_Step_CRI),1]<- Pacientes_CRI2$P_C
RI2
Momento_Llegada_PLA[1:length(Segundo_Step_CRI),2]<- Segundo_Step_CRI
Momento_Llegada_PLA[1:length(Segundo_Step_CRI),3]<- Pacientes_CRI2$Mom
ento_Terminación_CRI2
Momento_Llegada_PLA[1:length(Segundo_Step_CRI),4]<- Momento_Llegada_CR
I1$SPLIT_CRI1
Momento_Llegada_PLA[1:length(Segundo_Step_CRI),5]<- Pacientes_CRI2$Ant
erior_CRI2

#PACIENTES DE UCI
añadir1<- length(Segundo_Step_CRI) + 1
longitud2<- length(Segundo_Step_CRI) + length(Segundo_Step_UCI)
Momento_Llegada_PLA[añadir1:longitud2,1]<- Pacientes_UCI2$P_UCI2
Momento_Llegada_PLA[añadir1:longitud2,2]<- Segundo_Step_UCI
Momento_Llegada_PLA[añadir1:longitud2,3]<- Pacientes_UCI2$Momento_Term
inación_UCI2
Momento_Llegada_PLA[añadir1:longitud2,4]<- Momento_Llegada_UCI1$SPLIT_
UCI1
Momento_Llegada_PLA[añadir1:longitud2,5]<- Pacientes_UCI2$Anterior_UCI
2

#PACIENTES DE OBSERVACIÓN
añadir2<- nrow(Momento_Llegada_PLA) + 1
longitud3<- nrow(Momento_Llegada_PLA) + length(P_OBS_PLA)
Momento_Llegada_PLA[añadir2:longitud3,1]<- P_OBS_PLA
Momento_Llegada_PLA[añadir2:longitud3,2]<- NODO_OBS_PLA
Momento_Llegada_PLA[añadir2:longitud3,3]<- MOMENTO_OBS_PLA
Momento_Llegada_PLA[añadir2:longitud3,4]<- Obs_PLA_SPLIT

#PACIENTES DE QUIROFANO
añadir3<- nrow(Momento_Llegada_PLA) + 1
longitud4<- nrow(Momento_Llegada_PLA) + length(P_QUI_PLA)
Momento_Llegada_PLA[añadir3:longitud4,1]<- P_QUI_PLA
Momento_Llegada_PLA[añadir3:longitud4,2]<- NODO_QUI_PLA
```

```
Momento_Llegada_PLA[añadir3:longitud4,3]<- MOMENTO_QUI_PLA
Momento_Llegada_PLA[añadir3:longitud4,4]<- QUI_PLA_SPLIT

Momento_Llegada_PLA<- Momento_Llegada_PLA[with(Momento_Llegada_PLA, or
der(Momento_Llegada_PLA$Llegada_PLA)), ]
#Momento_Llegada_PLA
########################## CÁLCULOS TERCER STEP #################
################################## PLANTA ########################
for (u in 1:nrow(Momento_Llegada_PLA)){ #lo hacemos desde 1 hasta el n
umero de pacientes que vayan a planta
  contadorPLA<- contadorPLA +1 #para saber cuantos servidores hay
  Llegada_PLA2[u]<- Momento_Llegada_PLA$Llegada_PLA[u]

  #para saber de que pacientes se trata
  p_pla<- which(Momento_Llegada_PLA$Llegada_PLA==Momento_Llegada_PLA$L
legada_PLA[u])
  P_PLA2[u]<- Momento_Llegada_PLA$P_PLA[p_pla]
  Tiempo_servicio_PLA2[u]<-  runif(1, min = 1440, max = 10080) #rbeta(
1,0.5,0.5)*9000
  {
    if (contadorPLA<=S_Disponible_PLA2){
      Momento_Terminación_PLA2[u]<- na.omit(Momento_Llegada_PLA$Llegad
a_PLA[u]+ Tiempo_servicio_PLA2[u])
      Tiempo_Espera_PLA2[u]<- 0
      reajueste_PLA2[u]<- Momento_Terminación_PLA2[u]
      if(contadorPLA==1){
        Disponible_PLA2[u]<- S_Disponible_PLA2-1
      }
      if(contadorPLA>1){
        if(min(reajueste_PLA2)>Llegada_PLA2[contadorPLA]){
          Disponible_PLA2[u]<- Disponible_PLA2[u-1] - 1
        }
        if(min(reajueste_PLA2)<Llegada_PLA2[contadorPLA]){
          if(Disponible_PLA2[u-1]<S_Disponible_PLA2){
          Disponible_PLA2[u]<- Disponible_PLA2[u-1] -1
          }
          if(Disponible_PLA2[u-1]==S_Disponible_PLA2){
            Disponible_PLA2[u]<- Disponible_PLA2[u-1]
          }
        }
      }
    }
    else{
      minimo_PLA2<-min(reajueste_PLA2)
      ordenar_PLA2<- order(minimo_PLA2,decreasing = T)[1]
      if(Momento_Llegada_PLA$Llegada_PLA[u]==Momento_Llegada_PLA$Llega
da_PLA[ordenar_PLA2]){
        Tiempo_Espera_PLA2[u]=minimo_PLA2
      }
      else{
        Tiempo_Espera_PLA2[u]<- minimo_PLA2 - Momento_Llegada_PLA$Lleg
ada_PLA[u]
        Disponible_PLA2[u]<- Disponible_PLA2[u-1] - 1
        if(Disponible_PLA2[u-1]==0){
```

```
          Disponible_PLA2[u]<- 0
        }
        if(Tiempo_Espera_PLA2[u]<=0){
          Tiempo_Espera_PLA2[u]<- 0
          if(Disponible_PLA2[u-1]<S_Disponible_PLA2){
          Disponible_PLA2[u]<- Disponible_PLA2[u-1] +1
          }
          if(Disponible_PLA2[u-1]==S_Disponible_PLA2){
            Disponible_PLA2[u]<- Disponible_PLA2[u-1]
          }
        }
      }
    }
    Momento_Terminación_PLA2[u]<- Momento_Llegada_PLA$Llegada_PLA[u] +
Tiempo_Espera_PLA2[u]+Tiempo_servicio_PLA2[u]
    Tiempo_para_reajuste_PLA2<- reajueste_PLA2
    Tiempo_reajustado_PLA2<- replace(Tiempo_para_reajuste_PLA2, Tiempo
_para_reajuste_PLA2==minimo_PLA2 ,Momento_Terminación_PLA2[u])
    reajueste_PLA2<- Tiempo_reajustado_PLA2
  }
}
############################### RESULTADOS TERCER STEP #############
#PARA VER TODOS LOS PACIENTES DE PRIOPRIDAD PLA EN CADA UNO DE LOS SER
VIDORES
Pacientes_PLA2<- data.frame(P_PLA2=P_PLA2,Llegada_PLA2=Llegada_PLA2,Ti
empo_Espera_PLA2=Tiempo_Espera_PLA2,Tiempo_servicio_PLA2=Tiempo_servic
io_PLA2,Momento_Terminación_PLA2=Momento_Terminación_PLA2,Disponible_P
LA2=Disponible_PLA2,Llegada_Hora_Paciente= round(Llegada_PLA2/60,0),Sa
lida_Hora_Paciente=round(Momento_Terminación_PLA2/60,0))
#Pacientes_PLA2
}
############## CÁLCULOS DE LOS PARAMETROS DE TEORÍA DE COLAS #########
{
######################### TIEMPO MEDIA DE LLEGADA #################
Tiempo_Medio_Llegada_Tri<- (length(Pacientes_Triaje$P_Triaje)/t)
Tiempo_Medio_Llegada_Con<- (length(Pacientes_Consulta$P_Consulta)/t)
Tiempo_Medio_Llegada_Exp<- (length(Pacientes_Exploracion$P_Exploracion
)/t)
Tiempo_Medio_Llegada_Obs<- (length(Pacientes_Observacion$P_Observacion
)/t)
Tiempo_Medio_Llegada_RCP<- (length(Pacientes_RCP$P_RCP)/t)
Tiempo_Medio_Llegada_UCI<- (length(Pacientes_UCI2$P_UCI2)/t)
Tiempo_Medio_Llegada_CRI<- (length(Pacientes_CRI2$P_CRI2)/t)
Tiempo_Medio_Llegada_QUI<- (length(Pacientes_QUI2$P_QUI2)/t)
Tiempo_Medio_Llegada_PLA<- (length(Pacientes_PLA2$P_PLA2)/t)
Tiempo_Medio_Llegada_ALT<- (length(Momento_Llegada_ALT$P_ALT)/t)
Tiempo_Medio_Llegada_SIST<- (length(Pacientes_Triaje$P_Triaje)/t)
######################### TIEMPO MEDIA DE SERVICIO ############
Tiempo_Medio_Ser_Tri<- mean(Pacientes_Triaje$Tiempo_servicio_Triaje)
Tiempo_Medio_Ser_Con<- mean(Pacientes_Consulta$Tiempo_servicio_Consult
a)
Tiempo_Medio_Ser_Exp<- mean(Pacientes_Exploracion$Tiempo_servicio_Expl
oracion)
Tiempo_Medio_Ser_Obs<- mean(Pacientes_Observacion$Tiempo_servicio_Obse
```

```
rvacion)
Tiempo_Medio_Ser_RCP<- mean(Pacientes_RCP$Tiempo_servicio_RCP)
Tiempo_Medio_Ser_UCI<- mean(Pacientes_UCI2$Tiempo_servicio_UCI2)
Tiempo_Medio_Ser_CRI<- mean(Pacientes_CRI2$Tiempo_servicio_CRI2)
Tiempo_Medio_Ser_QUI<- mean(Pacientes_QUI2$Tiempo_servicio_QUI2)
Tiempo_Medio_Ser_PLA<- mean(Pacientes_PLA2$Tiempo_servicio_PLA2)
Tiempo_Medio_Ser_ALT<- NA
Tiempo_Medio_Ser_SIST<- (Tiempo_Medio_Ser_Con+Tiempo_Medio_Ser_Exp+
                           Tiempo_Medio_Ser_Obs+Tiempo_Medio_Ser_RCP
+
                           Tiempo_Medio_Ser_UCI+Tiempo_Medio_Ser_CRI
+
                           Tiempo_Medio_Ser_QUI+Tiempo_Medio_Ser_PLA
)/9
####################### TIEMPO MEDIO EN COLA #################
Tiempo_Medio_Cola_Tri<- mean(Pacientes_Triaje$Tiempo_Espera_Triaje)
Tiempo_Medio_Cola_Con<- mean(Pacientes_Consulta$Tiempo_Espera_Consulta
)
Tiempo_Medio_Cola_Exp<- mean(Pacientes_Exploracion$Tiempo_Espera_Explo
racion)
Tiempo_Medio_Cola_Obs<- mean(Pacientes_Observacion$Tiempo_Espera_Obser
vacion)
Tiempo_Medio_Cola_RCP<- mean(Pacientes_RCP$Tiempo_Espera_RCP)
Tiempo_Medio_Cola_UCI<- mean(Pacientes_UCI2$Tiempo_Espera_UCI2)
Tiempo_Medio_Cola_CRI<- mean(Pacientes_CRI2$Tiempo_Espera_CRI2)
Tiempo_Medio_Cola_QUI<- mean(Pacientes_QUI2$Tiempo_Espera_QUI2)
Tiempo_Medio_Cola_PLA<- mean(Pacientes_PLA2$Tiempo_Espera_PLA2)
Tiempo_Medio_Cola_ALT<- NA
Tiempo_Medio_Cola_SIST<- (Tiempo_Medio_Cola_Tri+Tiempo_Medio_Cola_Con+
Tiempo_Medio_Cola_Exp+
                            Tiempo_Medio_Cola_Obs+Tiempo_Medio_Cola_RC
P+Tiempo_Medio_Cola_UCI+
                            Tiempo_Medio_Cola_CRI+Tiempo_Medio_Cola_QU
I+Tiempo_Medio_Cola_PLA)/9
####################### NÚMERO DE SERVIDORES #################
Número_Servidores_Tri<- Disponible_Tri
Número_Servidores_Con<- Disponible_Con
Número_Servidores_Exp<- Disponible_Exp
Número_Servidores_Obs<- Disponible_Obs
Número_Servidores_RCP<- Disponible_RCP
Número_Servidores_UCI<- Disponible_UCI
Número_Servidores_CRI<- Disponible_CRI
Número_Servidores_QUI<- Disponible_QUI
Número_Servidores_PLA<- S_Disponible_PLA2
Número_Servidores_ALT<- NA
Número_Servidores_SIST<- sum(Número_Servidores_Con,Número_Servidores_E
xp,
                           Número_Servidores_Obs,Número_Servidores_R
CP,
                           Número_Servidores_UCI,Número_Servidores_C
RI,
                           Número_Servidores_QUI,Número_Servidores_P
LA)
####################### FACTOR DE UTILIZACIÓN #################
```

```
Factor_Uti_Tri<- Tiempo_Medio_Llegada_Tri/(Número_Servidores_Tri*Tiemp
o_Medio_Ser_Tri)
Factor_Uti_Con<- Tiempo_Medio_Llegada_Con/(Número_Servidores_Con*Tiemp
o_Medio_Ser_Con)
Factor_Uti_Exp<- Tiempo_Medio_Llegada_Exp/(Número_Servidores_Exp*Tiemp
o_Medio_Ser_Exp)
Factor_Uti_Obs<- Tiempo_Medio_Llegada_Obs/(Número_Servidores_Obs*Tiemp
o_Medio_Ser_Obs)
Factor_Uti_RCP<- Tiempo_Medio_Llegada_RCP/(Número_Servidores_RCP*Tiemp
o_Medio_Ser_RCP)
Factor_Uti_UCI<- Tiempo_Medio_Llegada_UCI/(Número_Servidores_UCI*Tiemp
o_Medio_Ser_UCI)
Factor_Uti_CRI<- Tiempo_Medio_Llegada_CRI/(Número_Servidores_CRI*Tiemp
o_Medio_Ser_CRI)
Factor_Uti_QUI<- Tiempo_Medio_Llegada_QUI/(Número_Servidores_QUI*Tiemp
o_Medio_Ser_QUI)
Factor_Uti_PLA<- Tiempo_Medio_Llegada_PLA/(Número_Servidores_PLA*Tiemp
o_Medio_Ser_PLA)
Factor_Uti_ALT<- NA
Factor_Uti_SIST<- Tiempo_Medio_Llegada_SIST/ Tiempo_Medio_Ser_SIST
######################## TIEMPO MEDIO DE DURACIÓN ###################
Tiempo_Medio_Duración_Tri<- mean(Pacientes_Triaje$Momento_Terminación_
Triaje - Pacientes_Triaje$Llegada_Triaje)
Tiempo_Medio_Duración_Con<- mean(Pacientes_Consulta$Momento_Terminació
n_Consulta - Pacientes_Consulta$Llegada_Consulta)
Tiempo_Medio_Duración_Exp<- mean(Pacientes_Exploracion$Momento_Termina
ción_Exploracion - Pacientes_Exploracion$Llegada_Exploracion)
Tiempo_Medio_Duración_Obs<- mean(Pacientes_Observacion$Momento_Termina
ción_Observacion - Pacientes_Observacion$Llegada_Observacion)
Tiempo_Medio_Duración_RCP<- mean(Pacientes_RCP$Momento_Terminación_RCP
- Pacientes_RCP$Llegada_RCP)
Tiempo_Medio_Duración_UCI<- mean(Pacientes_UCI2$Momento_Terminación_UC
I2 - Pacientes_UCI2$Llegada_UCI2)
Tiempo_Medio_Duración_CRI<- mean(Pacientes_CRI2$Momento_Terminación_CR
I2 - Pacientes_CRI2$Llegada_CRI2)
Tiempo_Medio_Duración_QUI<- mean(Pacientes_QUI2$Momento_Terminación_QU
I2 - Pacientes_QUI2$Llegada_QUI2)
Tiempo_Medio_Duración_PLA<- mean(Pacientes_PLA2$Momento_Terminación_PL
A2 - Pacientes_PLA2$Llegada_PLA2)
Tiempo_Medio_Duración_ALT<- NA
Tiempo_Medio_Duración_SIST<- (mean(Momento_Llegada_ALT$Momento_Llegada
_Incial_ALT)+                        mean(Pacientes_PLA2$Momento_Termin
ación_PLA2-Pacientes_PLA2$Llegada_PLA2))/2
#################### NÚMERO MEDIO DE PACIENTES EN COLA ##########
Número_Medio_Pa_Cola_Tri<-length(which(Pacientes_Triaje$Tiempo_Espera_
Triaje>0))/length(Pacientes_Triaje$P_Triaje)
Número_Medio_Pa_Cola_Con<-length(which(Pacientes_Consulta$Tiempo_Esper
a_Consulta>0))/length(Pacientes_Consulta$P_Consulta)
Número_Medio_Pa_Cola_Exp<-length(which(Pacientes_Exploracion$Tiempo_Es
pera_Exploracion>0))/length(Pacientes_Exploracion$P_Exploracion)
Número_Medio_Pa_Cola_Obs<-length(which(Pacientes_Observacion$Tiempo_Es
pera_Observacion>0))/length(Pacientes_Observacion$P_Observacion)
Número_Medio_Pa_Cola_RCP<-length(which(Pacientes_RCP$Tiempo_Espera_RCP
>0))/length(Pacientes_RCP$P_RCP)
```

```
Número_Medio_Pa_Cola_UCI<-length(which(Pacientes_UCI2$Tiempo_Espera_UC
I2>0))/length(Pacientes_UCI2$P_UCI2)
Número_Medio_Pa_Cola_CRI<-length(which(Pacientes_CRI2$Tiempo_Espera_CR
I2>0))/length(Pacientes_CRI2$P_CRI2)
Número_Medio_Pa_Cola_QUI<-length(which(Pacientes_QUI2$Tiempo_Espera_QU
I2>0))/length(Pacientes_QUI2$P_QUI2)
Número_Medio_Pa_Cola_PLA<-length(which(Pacientes_PLA2$Tiempo_Espera_PL
A2>0))/length(Pacientes_PLA2$P_PLA2)
Número_Medio_Pa_Cola_ALT<- NA
Número_Medio_Pa_Cola_SIST<- (Número_Medio_Pa_Cola_Tri+Número_Medio_Pa_
Cola_Con+
                             Número_Medio_Pa_Cola_Exp+Número_Medio_Pa_
Cola_Obs+
                              Número_Medio_Pa_Cola_RCP+Número_Medio_Pa
_Cola_UCI+
                             Número_Medio_Pa_Cola_CRI+Número_Medio_Pa_
Cola_QUI+
                               Número_Medio_Pa_Cola_PLA)/9
################ NÚMERO MEDIO DE SERVIDORES OCUPADOS #################
Número_Medio_Ser_Ocu_Tri<- Factor_Uti_Tri*Número_Servidores_Tri
Número_Medio_Ser_Ocu_Con<- Factor_Uti_Con*Número_Servidores_Con
Número_Medio_Ser_Ocu_Exp<- Factor_Uti_Exp*Número_Servidores_Exp
Número_Medio_Ser_Ocu_Obs<- Factor_Uti_Obs*Número_Servidores_Obs
Número_Medio_Ser_Ocu_RCP<- Factor_Uti_RCP*Número_Servidores_RCP
Número_Medio_Ser_Ocu_UCI<- Factor_Uti_UCI*Número_Servidores_UCI
Número_Medio_Ser_Ocu_CRI<- Factor_Uti_CRI*Número_Servidores_CRI
Número_Medio_Ser_Ocu_QUI<- Factor_Uti_QUI*Número_Servidores_QUI
Número_Medio_Ser_Ocu_PLA<- Factor_Uti_PLA*Número_Servidores_PLA
Número_Medio_Ser_Ocu_ALT<- NA
Número_Medio_Ser_Ocu_SIST<-  Factor_Uti_SIST*Número_Servidores_SIST
############## CREAMOS EL DATA.FRAME DE LOS DATOS ####################
Parametros<- data.frame(Nodos=c("Triaje","Consulta","Exploración","Obs
ervación",
                                "RCP","UCI","Críticos","Quirofano",
                                "Planta","Alta","Sistema General"),
Tiempo_Medio_Llegada=c(Tiempo_Medio_Llegada_Tri,Tiempo_Medio_Llegada_C
on,Tiempo_Medio_Llegada_Exp,Tiempo_Medio_Llegada_Obs,Tiempo_Medio_Lleg
ada_RCP,Tiempo_Medio_Llegada_UCI,Tiempo_Medio_Llegada_CRI,Tiempo_Medio
_Llegada_QUI,Tiempo_Medio_Llegada_PLA,Tiempo_Medio_Llegada_ALT,Tiempo_
Medio_Llegada_SIST),
                        Tiempo_Medio_Servicio=c(Tiempo_Medio_Ser_Tri,T
iempo_Medio_Ser_Con,Tiempo_Medio_Ser_Exp,Tiempo_Medio_Ser_Obs,Tiempo_M
edio_Ser_RCP,Tiempo_Medio_Ser_UCI,Tiempo_Medio_Ser_CRI,Tiempo_Medio_Se
r_QUI,Tiempo_Medio_Ser_PLA,Tiempo_Medio_Ser_ALT,Tiempo_Medio_Ser_SIST)
,
                        Número_Servidores=c(Número_Servidores_Tri,Núme
ro_Servidores_Con,Número_Servidores_Exp,Número_Servidores_Obs,Número_S
ervidores_RCP,Número_Servidores_UCI,Número_Servidores_CRI,Número_Servi
dores_QUI,Número_Servidores_PLA,Número_Servidores_ALT,Número_Servidore
s_SIST),
                        Factor_Utilización=c(Factor_Uti_Tri,Factor_Uti
_Con,Factor_Uti_Exp,Factor_Uti_Obs,Factor_Uti_RCP,Factor_Uti_UCI,Facto
r_Uti_CRI,Factor_Uti_QUI,Factor_Uti_PLA,Factor_Uti_ALT,Factor_Uti_SIST
),
```

```
                        Tiempo_Medio_Cola=c(Tiempo_Medio_Cola_Tri,Tiem
po_Medio_Cola_Con,Tiempo_Medio_Cola_Exp,Tiempo_Medio_Cola_Obs,Tiempo_M
edio_Cola_RCP,Tiempo_Medio_Cola_UCI,Tiempo_Medio_Cola_CRI,Tiempo_Medio
_Cola_QUI,Tiempo_Medio_Cola_PLA,Tiempo_Medio_Cola_ALT,Tiempo_Medio_Col
a_SIST),
                        Tiempo_Medio_Duración=c(Tiempo_Medio_Duración_
Tri,Tiempo_Medio_Duración_Con,Tiempo_Medio_Duración_Exp,Tiempo_Medio_D
uración_Obs,Tiempo_Medio_Duración_RCP,Tiempo_Medio_Duración_UCI,Tiempo
_Medio_Duración_CRI,Tiempo_Medio_Duración_QUI,Tiempo_Medio_Duración_PL
A,Tiempo_Medio_Duración_ALT,Tiempo_Medio_Duración_SIST),
                        Número_Medio_Pacientes_En_Cola=c(Número_Medio_
Pa_Cola_Tri,Número_Medio_Pa_Cola_Con,Número_Medio_Pa_Cola_Exp,Número_M
edio_Pa_Cola_Obs,Número_Medio_Pa_Cola_RCP,Número_Medio_Pa_Cola_UCI,Núm
ero_Medio_Pa_Cola_CRI,Número_Medio_Pa_Cola_QUI,Número_Medio_Pa_Cola_PL
A,Número_Medio_Pa_Cola_ALT,Número_Medio_Pa_Cola_SIST),
                        Número_Medio_Servidores_Ocupados=c(Número_Medi
o_Ser_Ocu_Tri,Número_Medio_Ser_Ocu_Con,Número_Medio_Ser_Ocu_Exp,Número
_Medio_Ser_Ocu_Obs,Número_Medio_Ser_Ocu_RCP,Número_Medio_Ser_Ocu_UCI,N
úmero_Medio_Ser_Ocu_CRI,Número_Medio_Ser_Ocu_QUI,Número_Medio_Ser_Ocu_
PLA,Número_Medio_Ser_Ocu_ALT,Número_Medio_Ser_Ocu_SIST))
}
##################### TRAZABILIDAD DE LOS PACIENTES ##############
############################### PRIMER PASO ##################
{
contadorDatos<- 0
Donde_Llega_Despues_Triaje<- 0
Cuando_Llega<- 0
Cuando_Termina<- 0
Personas<- 0
Donde_se_va<- 0
Momento_Llegada_al_Sistema<-0
Cuando_Termina_el_Triaje<- 0
contador_por_Con<-0
contador_por_Exp<-0
contador_por_RCP<- 0
contador_por_Obs<- 0
for (s in 1:sum(nclientes)) {
  contadorDatos<- contadorDatos + 1
  if(Primer_Step[s]== "Con"){
    Personas[contadorDatos]<- contadorDatos
    Momento_Llegada_al_Sistema[contadorDatos]<- Pacientes_Triaje$Llega
da_Triaje[which(Pacientes_Triaje$P_Triaje==Personas[contadorDatos])]
    Cuando_Termina_el_Triaje[contadorDatos]<- Pacientes_Triaje$Momento
_Terminación_Triaje[which(Pacientes_Triaje$P_Triaje==Personas[contador
Datos])]
    Donde_Llega_Despues_Triaje[contadorDatos]<- "CONSULTA"
    Cuando_Llega[contadorDatos]<- Pacientes_Consulta$Llegada_Consulta[
which(Pacientes_Consulta$P_Consulta==Personas[contadorDatos])]
    Cuando_Termina[contadorDatos]<- Pacientes_Consulta$Momento_Termina
ción_Consulta[which(Pacientes_Consulta$P_Consulta==Personas[contadorDa
tos])]
    contador_por_Con<- contador_por_Con+1
  }
  if(Primer_Step[s]== "Exp"){
```

```
    Personas[contadorDatos]<- contadorDatos
    Momento_Llegada_al_Sistema[contadorDatos]<- Pacientes_Triaje$Llega
da_Triaje[which(Pacientes_Triaje$P_Triaje==Personas[contadorDatos])]
    Cuando_Termina_el_Triaje[contadorDatos]<- Pacientes_Triaje$Momento
_Terminación_Triaje[which(Pacientes_Triaje$P_Triaje==Personas[contador
Datos])]
    Donde_Llega_Despues_Triaje[contadorDatos]<- "EXPLORACIÓN"
    Cuando_Llega[contadorDatos]<- Pacientes_Exploracion$Llegada_Explor
acion[which(Pacientes_Exploracion$P_Exploracion==Personas[contadorDato
s])]
    Cuando_Termina[contadorDatos]<- Pacientes_Exploracion$Momento_Term
inación_Exploracion[which(Pacientes_Exploracion$P_Exploracion==Persona
s[contadorDatos])]
    contador_por_Exp<- contador_por_Exp +1
  }
  if(Primer_Step[s]== "RCP"){
    Personas[contadorDatos]<- contadorDatos
    Momento_Llegada_al_Sistema[contadorDatos]<- Pacientes_Triaje$Llega
da_Triaje[which(Pacientes_Triaje$P_Triaje==Personas[contadorDatos])]
    Cuando_Termina_el_Triaje[contadorDatos]<- Pacientes_Triaje$Momento
_Terminación_Triaje[which(Pacientes_Triaje$P_Triaje==Personas[contador
Datos])]
    Donde_Llega_Despues_Triaje[contadorDatos]<- "RCP"
    Cuando_Llega[contadorDatos]<- Pacientes_RCP$Llegada_RCP[which(Paci
entes_RCP$P_RCP==Personas[contadorDatos])]
    Cuando_Termina[contadorDatos]<- Pacientes_RCP$Momento_Terminación_
RCP[which(Pacientes_RCP$P_RCP==Personas[contadorDatos])]
    contador_por_RCP<- contador_por_RCP +1
  }
  if(Primer_Step[s]== "Obs"){
    Personas[contadorDatos]<- contadorDatos
    Momento_Llegada_al_Sistema[contadorDatos]<- Pacientes_Triaje$Llega
da_Triaje[which(Pacientes_Triaje$P_Triaje==Personas[contadorDatos])]
    Cuando_Termina_el_Triaje[contadorDatos]<- Pacientes_Triaje$Momento
_Terminación_Triaje[which(Pacientes_Triaje$P_Triaje==Personas[contador
Datos])]
    Donde_Llega_Despues_Triaje[contadorDatos]<- Pacientes_Observacion$
Nodo_anterior_Obs1[which(Pacientes_Observacion$P_Observacion==Personas
[contadorDatos])]
    if(Donde_Llega_Despues_Triaje[contadorDatos]=="NINGUNO"){
      Donde_Llega_Despues_Triaje[contadorDatos]<- "OBSERVACIÓN"
      Cuando_Llega[contadorDatos]<- Pacientes_Observacion$Llegada_Obse
rvacion[which(Pacientes_Observacion$P_Observacion==Personas[contadorDa
tos])]
      Cuando_Termina[contadorDatos]<- Pacientes_Observacion$Momento_Te
rminación_Observacion[which(Pacientes_Observacion$P_Observacion==Perso
nas[contadorDatos])]
      contador_por_Obs<- contador_por_Obs+1
    }
    else{
    Cuando_Llega[contadorDatos]<- Pacientes_Consulta$Llegada_Consulta[
which(Pacientes_Consulta$P_Consulta==Personas[contadorDatos])]
    Cuando_Termina[contadorDatos]<- Pacientes_Consulta$Momento_Termina
ción_Consulta[which(Pacientes_Consulta$P_Consulta==Personas[contadorDa
```

```
tos])]
    }
  }
}
Porcentaje_Consulta_Paso1<- (contador_por_Con/sum(nclientes))*100
Porcentaje_Exploracion_Paso1<- (contador_por_Exp/sum(nclientes))*100
Porcentaje_RCP_Paso1<- (contador_por_RCP/sum(nclientes))*100
Porcentaje_Observacion_Paso1<- (contador_por_Obs/sum(nclientes))*100
Origen_Paso1<- c("TRIAJE","TRIAJE","TRIAJE","TRIAJE")
Destino_Paso1<- c("CONSULTA","EXPLORACIÓN","RCP","OBSERVACIÓN")
Porcentaje_Paso1<- c(Porcentaje_Consulta_Paso1,Porcentaje_Exploracion_
Paso1,
                     Porcentaje_RCP_Paso1,Porcentaje_Observacion_Paso1
)

datos<- data.frame(Personas=Pacientes,Momento_Llegada_al_Sistema=Momen
to_Llegada_al_Sistema,Cuando_Termina_el_Triaje=Cuando_Termina_el_Triaj
e,Donde_Llega_Despues_Triaje=Donde_Llega_Despues_Triaje,Cuando_Llega=C
uando_Llega,Cuando_Termina=Cuando_Termina)
}
############################## SEGUNDO PASO ####################
{
#DE CONSULTA A ALTA
contador_Con_ALT<- 0
nodo_Con_ALT<- 0
numero_paciente_Con_ALT<- 0
Momento_terminacion_Con_ALT<- 0
for (d in 1:length(Momento_Llegada_ALT$P_ALT)) {
  if(Momento_Llegada_ALT$Nodo_anterior_ALT[d]=="Con"){
    contador_Con_ALT<- contador_Con_ALT + 1
    nodo_Con_ALT[contador_Con_ALT]<- "ALTA"
    numero_paciente_Con_ALT[contador_Con_ALT]<- Momento_Llegada_ALT$P_
ALT[d]
    Momento_terminacion_Con_ALT[contador_Con_ALT]<- Momento_Llegada_AL
T$Llegada_ALT[d]
  }
}
Porcentaje_Con_Alta_Paso2<- (contador_Con_ALT/length(Pacientes_Consult
a$P_Consulta))*100

Con_ALT<- data.frame(numero_paciente_Con_ALT=numero_paciente_Con_ALT,
                     Momento_terminacion_Con_ALT=Momento_terminacion_C
on_ALT,
                   nodo_Con_ALT=nodo_Con_ALT)
Con_ALT<- Con_ALT[with(Con_ALT, order(Con_ALT$numero_paciente_Con_ALT)
), ]

#DE CONSULTA A OBSERVACIÓN
contador_Con_OBS<- 0
nodo_Con_OBS<- 0
numero_paciente_Con_OBS<- 0
Momento_terminacion_Con_OBS<- 0
for (f in 1:length(Pacientes_Observacion1$P_Observacion)) {
  if(Pacientes_Observacion1$Nodo_anterior_Obs[f]=="CONSULTA"){
```

```
    contador_Con_OBS<- contador_Con_OBS + 1
    nodo_Con_OBS[contador_Con_OBS]<- "OBSERVACIÓN"
    numero_paciente_Con_OBS[contador_Con_OBS]<- Pacientes_Observacion1
$P_Observacion[f]
    Momento_terminacion_Con_OBS[contador_Con_OBS]<- Pacientes_Observac
ion$Momento_Terminación_Observacion[which(Pacientes_Observacion$P_Obse
rvacion==numero_paciente_Con_OBS[contador_Con_OBS])]
  }
}
Porcentaje_Con_Observacion_Paso2<- (contador_Con_OBS/length(Pacientes_
Consulta$P_Consulta))*100

Con_OBS<- data.frame(numero_paciente_Con_OBS=numero_paciente_Con_OBS,
                     Momento_terminacion_Con_OBS=Momento_terminacion_C
on_OBS,
                     nodo_Con_OBS=nodo_Con_OBS)
Con_OBS<- Con_OBS[with(Con_OBS, order(Con_OBS$numero_paciente_Con_OBS)
), ]

#DE CONSULTA A UCI
contador_Con_UCI<- 0
nodo_Con_UCI<- 0
numero_paciente_Con_UCI<- 0
Momento_terminacion_Con_UCI<- 0
for (g in 1:length(Momento_Llegada_UCI1$P_UCI)) {
  if(Momento_Llegada_UCI1$Nodo_anterior_UCI[g]=="Con"){
    if(Momento_Llegada_UCI1$SPLIT_UCI1[g]!="CONSULTA"){
      contador_Con_UCI<- contador_Con_UCI + 1
      nodo_Con_UCI[contador_Con_UCI]<- "UCI"
      numero_paciente_Con_UCI[contador_Con_UCI]<- Momento_Llegada_UCI1
$P_UCI[g]
      Momento_terminacion_Con_UCI[contador_Con_UCI]<- Pacientes_UCI2$M
omento_Terminación_UCI2[which(Pacientes_UCI2$P_UCI2==numero_paciente_C
on_UCI[contador_Con_UCI])]
    }
  }
}
Porcentaje_Con_UCI_Paso2<- (contador_Con_UCI/length(Pacientes_Consulta
$P_Consulta))*100

Con_UCI<- data.frame(numero_paciente_Con_UCI=numero_paciente_Con_UCI,
                     Momento_terminacion_Con_UCI=Momento_terminacion_C
on_UCI,
                     nodo_Con_UCI=nodo_Con_UCI)
Con_UCI<- Con_UCI[with(Con_UCI, order(Con_UCI$numero_paciente_Con_UCI)
), ]

#DE CONSULTA A CRÍTICOS
contador_Con_CRI<- 0
nodo_Con_CRI<- 0
numero_paciente_Con_CRI<- 0
Momento_terminacion_Con_CRI<- 0
for (h in 1:length(Momento_Llegada_CRI1$P_CRI)) {
  if(Momento_Llegada_CRI1$Nodo_anterior_CRI[h]=="Con"){
```

```
    if(Momento_Llegada_CRI1$SPLIT_CRI1[h]!="CONSULTA"){
      contador_Con_CRI<- contador_Con_CRI + 1
      nodo_Con_CRI[contador_Con_CRI]<- "CRÍTICOS"
      numero_paciente_Con_CRI[contador_Con_CRI]<- Momento_Llegada_CRI1
$P_CRI[h]
      Momento_terminacion_Con_CRI[contador_Con_CRI]<- Pacientes_CRI2$M
omento_Terminación_CRI2[which(Pacientes_CRI2$P_CRI2==numero_paciente_C
on_CRI[contador_Con_CRI])]
    }
  }
}
Porcentaje_Con_CRI_Paso2<- (contador_Con_CRI/length(Pacientes_Consulta
$P_Consulta))*100

Con_CRI<- data.frame(numero_paciente_Con_CRI=numero_paciente_Con_CRI,
                     Momento_terminacion_Con_CRI=Momento_terminacion_C
on_CRI,
                     nodo_Con_CRI=nodo_Con_CRI)
Con_CRI<- Con_CRI[with(Con_CRI, order(Con_CRI$numero_paciente_Con_CRI)
), ]

#DE CONSULTA A QUIROFANO
contador_Con_QUI<- 0
nodo_Con_QUI<- 0
numero_paciente_Con_QUI<- 0
Momento_terminacion_Con_QUI<- 0
for (ñ in 1:length(Momento_Llegada_QUI$P_QUI)) {
  if(Momento_Llegada_QUI$Nodo_anterior_QUI[ñ]=="Con"){
    if(Momento_Llegada_QUI$Obs_QUI_SPLIT[ñ]!="CONSULTA"){
      contador_Con_QUI<- contador_Con_QUI + 1
      nodo_Con_QUI[contador_Con_QUI]<- "QUIROFANO"
      numero_paciente_Con_QUI[contador_Con_QUI]<- Momento_Llegada_QUI$
P_QUI[ñ]
      Momento_terminacion_Con_QUI[contador_Con_QUI]<- Pacientes_QUI2$M
omento_Terminación_QUI2[which(Pacientes_QUI2$P_QUI2==numero_paciente_C
on_QUI[contador_Con_QUI])]
    }
  }
}
Porcentaje_Con_QUI_Paso2<- (contador_Con_QUI/length(Pacientes_Consulta
$P_Consulta))*100

Con_QUI<- data.frame(numero_paciente_Con_QUI=numero_paciente_Con_QUI,
                     Momento_terminacion_Con_QUI=Momento_terminacion_C
on_QUI,
                     nodo_Con_QUI=nodo_Con_QUI)
Con_QUI<- Con_QUI[with(Con_QUI, order(Con_QUI$numero_paciente_Con_QUI)
), ]

#DE EXPLORACIÓN A UCI
contador_Exp_UCI<- 0
nodo_Exp_UCI<- 0
numero_paciente_Exp_UCI<- 0
Momento_terminacion_Exp_UCI<- 0
```

```
for (z in 1:length(Momento_Llegada_UCI1$P_UCI)) {
  if(Momento_Llegada_UCI1$Nodo_anterior_UCI[z]=="Exp"){
    if(Momento_Llegada_UCI1$SPLIT_UCI1[z]!="CONSULTA"){
      contador_Exp_UCI<- contador_Exp_UCI + 1
      nodo_Exp_UCI[contador_Exp_UCI]<- "UCI"
      numero_paciente_Exp_UCI[contador_Exp_UCI]<- Momento_Llegada_UCI1
$P_UCI[z]
      Momento_terminacion_Exp_UCI[contador_Exp_UCI]<- Pacientes_UCI2$M
omento_Terminación_UCI2[which(Pacientes_UCI2$P_UCI2==numero_paciente_E
xp_UCI[contador_Exp_UCI])]
    }
  }
}
Porcentaje_Exp_UCI_Paso2<- (contador_Exp_UCI/length(Pacientes_Explorac
ion$P_Exploracion))*100

Exp_UCI<- data.frame(numero_paciente_Exp_UCI=numero_paciente_Exp_UCI,
                     Momento_terminacion_Exp_UCI=Momento_terminacion_E
xp_UCI,
                     nodo_Exp_UCI=nodo_Exp_UCI)
Exp_UCI<- Exp_UCI[with(Exp_UCI, order(Exp_UCI$numero_paciente_Exp_UCI)
), ]

#DE EXPLORACIÓN A CRÍTICOS
contador_Exp_CRI<- 0
nodo_Exp_CRI<- 0
numero_paciente_Exp_CRI<- 0
Momento_terminacion_Exp_CRI<- 0
for (x in 1:length(Momento_Llegada_CRI1$P_CRI)) {
  if(Momento_Llegada_CRI1$Nodo_anterior_CRI[x]=="Exp"){
    if(Momento_Llegada_CRI1$SPLIT_CRI1[x]!="CONSULTA"){
      contador_Exp_CRI<- contador_Exp_CRI + 1
      nodo_Exp_CRI[contador_Exp_CRI]<- "CRÍTICOS"
      numero_paciente_Exp_CRI[contador_Exp_CRI]<- Momento_Llegada_CRI1
$P_CRI[x]
      Momento_terminacion_Exp_CRI[contador_Exp_CRI]<- Pacientes_CRI2$M
omento_Terminación_CRI2[which(Pacientes_CRI2$P_CRI2==numero_paciente_E
xp_CRI[contador_Exp_CRI])]
    }
  }
}
Porcentaje_Exp_CRI_Paso2<- (contador_Exp_CRI/length(Pacientes_Explorac
ion$P_Exploracion))*100

Exp_CRI<- data.frame(numero_paciente_Exp_CRI=numero_paciente_Exp_CRI,
                     Momento_terminacion_Exp_CRI=Momento_terminacion_E
xp_CRI,
                     nodo_Exp_CRI=nodo_Exp_CRI)
Exp_CRI<- Exp_CRI[with(Exp_CRI, order(Exp_CRI$numero_paciente_Exp_CRI)
), ]

#DE EXPLORACIÓN A QUIROFANO
contador_Exp_QUI<- 0
nodo_Exp_QUI<- 0
```

```
numero_paciente_Exp_QUI<- 0
Momento_terminacion_Exp_QUI<- 0
for (c in 1:length(Momento_Llegada_QUI$P_QUI)) {
  if(Momento_Llegada_QUI$Nodo_anterior_QUI[c]=="Exp"){
    if(Momento_Llegada_QUI$Obs_QUI_SPLIT[c]!="CONSULTA"){
      contador_Exp_QUI<- contador_Exp_QUI + 1
      nodo_Exp_QUI[contador_Exp_QUI]<- "QUIROFANO"
      numero_paciente_Exp_QUI[contador_Exp_QUI]<- Momento_Llegada_QUI$
P_QUI[c]
      Momento_terminacion_Exp_QUI[contador_Exp_QUI]<- Pacientes_QUI2$M
omento_Terminación_QUI2[which(Pacientes_QUI2$P_QUI2==numero_paciente_E
xp_QUI[contador_Exp_QUI])]
    }
  }
}
Porcentaje_Exp_QUI_Paso2<- (contador_Exp_QUI/length(Pacientes_Explorac
ion$P_Exploracion))*100

Exp_QUI<- data.frame(numero_paciente_Exp_QUI=numero_paciente_Exp_QUI,
                     Momento_terminacion_Exp_QUI=Momento_terminacion_E
xp_QUI,
                     nodo_Exp_QUI=nodo_Exp_QUI)
Exp_QUI<- Exp_QUI[with(Exp_QUI, order(Exp_QUI$numero_paciente_Exp_QUI)
), ]

#DE RCP A UCI
contador_RCP_UCI<- 0
nodo_RCP_UCI<- 0
numero_paciente_RCP_UCI<- 0
Momento_terminacion_RCP_UCI<- 0
for (v in 1:length(Momento_Llegada_UCI1$P_UCI)) {
  if(Momento_Llegada_UCI1$Nodo_anterior_UCI[v]=="RCP"){
    if(Momento_Llegada_UCI1$SPLIT_UCI1[v]!="CONSULTA"){
      contador_RCP_UCI<- contador_RCP_UCI + 1
      nodo_RCP_UCI[contador_RCP_UCI]<- "UCI"
      numero_paciente_RCP_UCI[contador_RCP_UCI]<- Momento_Llegada_UCI1
$P_UCI[v]
      Momento_terminacion_RCP_UCI[contador_RCP_UCI]<- Pacientes_UCI2$M
omento_Terminación_UCI2[which(Pacientes_UCI2$P_UCI2==numero_paciente_R
CP_UCI[contador_RCP_UCI])]
    }
  }
}
Porcentaje_RCP_UCI_Paso2<- (contador_RCP_UCI/length(Pacientes_RCP$P_RC
P))*100

RCP_UCI<- data.frame(numero_paciente_RCP_UCI=numero_paciente_RCP_UCI,
                     Momento_terminacion_RCP_UCI=Momento_terminacion_R
CP_UCI,
                     nodo_RCP_UCI=nodo_RCP_UCI)
RCP_UCI<- RCP_UCI[with(RCP_UCI, order(RCP_UCI$numero_paciente_RCP_UCI)
), ]

#DE RCP A CRÍTICOS
```

```
contador_RCP_CRI<- 0
nodo_RCP_CRI<- 0
numero_paciente_RCP_CRI<- 0
Momento_terminacion_RCP_CRI<- 0
for (b in 1:length(Momento_Llegada_CRI1$P_CRI)) {
  if(Momento_Llegada_CRI1$Nodo_anterior_CRI[b]=="RCP"){
    if(Momento_Llegada_CRI1$SPLIT_CRI1[b]!="CONSULTA"){
      contador_RCP_CRI<- contador_RCP_CRI + 1
      nodo_RCP_CRI[contador_RCP_CRI]<- "CRÍTICOS"
      numero_paciente_RCP_CRI[contador_RCP_CRI]<- Momento_Llegada_CRI1
$P_CRI[b]
      Momento_terminacion_RCP_CRI[contador_RCP_CRI]<- Pacientes_CRI2$M
omento_Terminación_CRI2[which(Pacientes_CRI2$P_CRI2==numero_paciente_R
CP_CRI[contador_RCP_CRI])]
    }
  }
}
Porcentaje_RCP_CRI_Paso2<- (contador_RCP_CRI/length(Pacientes_RCP$P_RC
P))*100

RCP_CRI<- data.frame(numero_paciente_RCP_CRI=numero_paciente_RCP_CRI,
                     Momento_terminacion_RCP_CRI=Momento_terminacion_R
CP_CRI,
                     nodo_RCP_CRI=nodo_RCP_CRI)
RCP_CRI<- RCP_CRI[with(RCP_CRI, order(RCP_CRI$numero_paciente_RCP_CRI)
), ]

#DE RCP A QUIROFANO
contador_RCP_QUI<- 0
nodo_RCP_QUI<- 0
numero_paciente_RCP_QUI<- 0
Momento_terminacion_RCP_QUI<- 0
for (n in 1:length(Momento_Llegada_QUI$P_QUI)) {
  if(Momento_Llegada_QUI$Nodo_anterior_QUI[n]=="RCP"){
    if(Momento_Llegada_QUI$Obs_QUI_SPLIT[n]!="CONSULTA"){
      contador_RCP_QUI<- contador_RCP_QUI + 1
      nodo_RCP_QUI[contador_RCP_QUI]<- "QUIROFANO"
      numero_paciente_RCP_QUI[contador_RCP_QUI]<- Momento_Llegada_QUI$
P_QUI[n]
      Momento_terminacion_RCP_QUI[contador_RCP_QUI]<- Pacientes_QUI2$M
omento_Terminación_QUI2[which(Pacientes_QUI2$P_QUI2==numero_paciente_R
CP_QUI[contador_RCP_QUI])]
    }
  }
}
Porcentaje_RCP_QUI_Paso2<- (contador_RCP_QUI/length(Pacientes_RCP$P_RC
P))*100

RCP_QUI<- data.frame(numero_paciente_RCP_QUI=numero_paciente_RCP_QUI,
                     Momento_terminacion_RCP_QUI=Momento_terminacion_R
CP_QUI,
                     nodo_RCP_QUI=nodo_RCP_QUI)
RCP_QUI<- RCP_QUI[with(RCP_QUI, order(RCP_QUI$numero_paciente_RCP_QUI)
), ]
```

```
#DE OBSERVACIÓN A UCI
contador_Obs_UCI<- 0
nodo_Obs_UCI<- 0
numero_paciente_Obs_UCI<- 0
Momento_terminacion_Obs_UCI<- 0
for (q in 1:length(Momento_Llegada_UCI1$P_UCI)) {
  if(Momento_Llegada_UCI1$Nodo_anterior_UCI[q]=="Obs"){
    if(Momento_Llegada_UCI1$SPLIT_UCI1[q]=="NINGUNO"){
      contador_Obs_UCI<- contador_Obs_UCI + 1
      nodo_Obs_UCI[contador_Obs_UCI]<- "UCI"
      numero_paciente_Obs_UCI[contador_Obs_UCI]<- Momento_Llegada_UCI1
$P_UCI[q]
      Momento_terminacion_Obs_UCI[contador_Obs_UCI]<- Pacientes_UCI2$M
omento_Terminación_UCI2[which(Pacientes_UCI2$P_UCI2==numero_paciente_O
bs_UCI[contador_Obs_UCI])]
    }
  }
}
Porcentaje_Obs_UCI_Paso2<- (contador_Obs_UCI/length(which(Pacientes_Ob
servacion$Nodo_anterior_Obs1=="NINGUNO")))*100

Obs_UCI<- data.frame(numero_paciente_Obs_UCI=numero_paciente_Obs_UCI,
                     Momento_terminacion_Obs_UCI=Momento_terminacion_O
bs_UCI,
                     nodo_Obs_UCI=nodo_Obs_UCI)
Obs_UCI<- Obs_UCI[with(Obs_UCI, order(Obs_UCI$numero_paciente_Obs_UCI)
), ]

#DE OBSERVACIÓN A CRÍTICOS
contador_Obs_CRI<- 0
nodo_Obs_CRI<- 0
numero_paciente_Obs_CRI<- 0
Momento_terminacion_Obs_CRI<- 0
for (n in 1:length(Momento_Llegada_CRI1$P_CRI)) {
  if(Momento_Llegada_CRI1$Nodo_anterior_CRI[n]=="Obs"){
    if(Momento_Llegada_CRI1$SPLIT_CRI1[n]=="NINGUNO"){
      contador_Obs_CRI<- contador_Obs_CRI + 1
      nodo_Obs_CRI[contador_Obs_CRI]<- "CRÍTICOS"
      numero_paciente_Obs_CRI[contador_Obs_CRI]<- Momento_Llegada_CRI1
$P_CRI[n]
      Momento_terminacion_Obs_CRI[contador_Obs_CRI]<- Pacientes_CRI2$M
omento_Terminación_CRI2[which(Pacientes_CRI2$P_CRI2==numero_paciente_O
bs_CRI[contador_Obs_CRI])]
    }
  }
}
Porcentaje_Obs_CRI_Paso2<- (contador_Obs_CRI/length(which(Pacientes_Ob
servacion$Nodo_anterior_Obs1=="NINGUNO")))*100

Obs_CRI<- data.frame(numero_paciente_Obs_CRI=numero_paciente_Obs_CRI,
                     Momento_terminacion_Obs_CRI=Momento_terminacion_O
bs_CRI,
                     nodo_Obs_CRI=nodo_Obs_CRI)
```

```
Obs_CRI<- Obs_CRI[with(Obs_CRI, order(Obs_CRI$numero_paciente_Obs_CRI)
), ]

#DE OBSERVACIÓN A QUIROFANO
contador_Obs_QUI<- 0
nodo_Obs_QUI<- 0
numero_paciente_Obs_QUI<- 0
Momento_terminacion_Obs_QUI<- 0
for (w in 1:length(Momento_Llegada_QUI$P_QUI)) {
  if(Momento_Llegada_QUI$Nodo_anterior_QUI[w]=="Obs"){
    if(Momento_Llegada_QUI$Obs_QUI_SPLIT[w]=="NINGUNO"){
      contador_Obs_QUI<- contador_Obs_QUI + 1
      nodo_Obs_QUI[contador_Obs_QUI]<- "QUIROFANO"
      numero_paciente_Obs_QUI[contador_Obs_QUI]<- Momento_Llegada_QUI$
P_QUI[w]
      Momento_terminacion_Obs_QUI[contador_Obs_QUI]<- Pacientes_QUI2$M
omento_Terminación_QUI2[which(Pacientes_QUI2$P_QUI2==numero_paciente_O
bs_QUI[contador_Obs_QUI])]
    }
  }
}
Porcentaje_Obs_QUI_Paso2<- (contador_Obs_QUI/length(which(Pacientes_Ob
servacion$Nodo_anterior_Obs1=="NINGUNO")))*100

Obs_QUI<- data.frame(numero_paciente_Obs_QUI=numero_paciente_Obs_QUI,
                     Momento_terminacion_Obs_QUI=Momento_terminacion_O
bs_QUI,
                     nodo_Obs_QUI=nodo_Obs_QUI)
Obs_QUI<- Obs_QUI[with(Obs_QUI, order(Obs_QUI$numero_paciente_Obs_QUI)
), ]

#DE OBSERVACIÓN A ALTA
contador_Obs_ALT<- 0
nodo_Obs_ALT<- 0
numero_paciente_Obs_ALT<- 0
Momento_terminacion_Obs_ALT<- 0
for (e in 1:length(Momento_Llegada_ALT$P_ALT)) {
  if(Momento_Llegada_ALT$Nodo_anterior_ALT[e]=="Obs"){
    if(Momento_Llegada_ALT$Obs_ALT_SPLIT[e]=="NINGUNO"){
      contador_Obs_ALT<- contador_Obs_ALT + 1
      nodo_Obs_ALT[contador_Obs_ALT]<- "ALTA"
      numero_paciente_Obs_ALT[contador_Obs_ALT]<- Momento_Llegada_ALT$
P_ALT[e]
      Momento_terminacion_Obs_ALT[contador_Obs_ALT]<- Momento_Llegada_
ALT$Llegada_ALT[e]
    }
  }
}
Porcentaje_Obs_ALT_Paso2<- (contador_Obs_ALT/length(which(Pacientes_Ob
servacion$Nodo_anterior_Obs1=="NINGUNO")))*100

Obs_ALT<- data.frame(numero_paciente_Obs_ALT=numero_paciente_Obs_ALT,
                     Momento_terminacion_Obs_ALT=Momento_terminacion_O
bs_ALT,
```

```
                    nodo_Obs_ALT=nodo_Obs_ALT)
Obs_ALT<- Obs_ALT[with(Obs_ALT, order(Obs_ALT$numero_paciente_Obs_ALT)
), ]

#DE OBSERVACIÓN A PLANTA
contador_Obs_PLA<- 0
nodo_Obs_PLA<- 0
numero_paciente_Obs_PLA<- 0
Momento_terminacion_Obs_PLA<- 0
for (r in 1:length(Momento_Llegada_PLA$P_PLA)) {
  if(Momento_Llegada_PLA$Nodo_anterior_PLA[r]=="OBS"){
    if(Momento_Llegada_PLA$SPLIT[r]=="NINGUNO"){
      contador_Obs_PLA<- contador_Obs_PLA + 1
      nodo_Obs_PLA[contador_Obs_PLA]<- "PLANTA"
      numero_paciente_Obs_PLA[contador_Obs_PLA]<- Momento_Llegada_PLA$
P_PLA[r]
      Momento_terminacion_Obs_PLA[contador_Obs_PLA]<- Pacientes_PLA2$M
omento_Terminación_PLA2[which(Pacientes_PLA2$P_PLA2==numero_paciente_O
bs_PLA[contador_Obs_PLA])]
    }
  }
}
Porcentaje_Obs_PLA_Paso2<- (contador_Obs_PLA/length(which(Pacientes_Ob
servacion$Nodo_anterior_Obs1=="NINGUNO")))*100

Obs_PLA<- data.frame(numero_paciente_Obs_PLA=numero_paciente_Obs_PLA,
                    Momento_terminacion_Obs_PLA=Momento_terminacion_O
bs_PLA,
                    nodo_Obs_PLA=nodo_Obs_PLA)
Obs_PLA<- Obs_PLA[with(Obs_PLA, order(Obs_PLA$numero_paciente_Obs_PLA)
), ]

Origen_Paso2<- c("CONSULTA","CONSULTA","CONSULTA","CONSULTA","CONSULTA
",
                 "EXPLORACIÓN","EXPLORACIÓN","EXPLORACIÓN",
                 "RCP","RCP","RCP","OBSERVACIÓN","OBSERVACIÓN",
                 "OBSERVACIÓN","OBSERVACIÓN","OBSERVACIÓN")
Destino_Paso2<- c("ALTA","OBSERVACIÓN","UCI","CRÍTICOS","QUIROFANO",
                  "UCI","CRÍTICOS","QUIROFANO","UCI",
                  "CRÍTICOS","QUIROFANO","UCI","CRÍTICOS",
                  "QUIROFANO","ALTA","PLANTA")
Porcentaje_Paso2<- c(Porcentaje_Con_Alta_Paso2,Porcentaje_Con_Observac
ion_Paso2,
                    Porcentaje_Con_UCI_Paso2,Porcentaje_Con_CRI_Paso2
,
                    Porcentaje_Con_QUI_Paso2,Porcentaje_Exp_UCI_Paso2
,
                    Porcentaje_Exp_CRI_Paso2,Porcentaje_Exp_QUI_Paso2
,
                    Porcentaje_RCP_UCI_Paso2,Porcentaje_RCP_CRI_Paso2
,
                    Porcentaje_RCP_QUI_Paso2,Porcentaje_Obs_UCI_Paso2
,
                    Porcentaje_Obs_CRI_Paso2,Porcentaje_Obs_QUI_Paso2
```

```
,
                      Porcentaje_Obs_ALT_Paso2,Porcentaje_Obs_PLA_Paso2
)
Numero_Paciente_Segundo_Paso<- c(Con_ALT$numero_paciente_Con_ALT,
                                 Con_OBS$numero_paciente_Con_OBS,
                                 Con_UCI$numero_paciente_Con_UCI,
                                 Con_CRI$numero_paciente_Con_CRI,
                                 Con_QUI$numero_paciente_Con_QUI,
                                 Exp_UCI$numero_paciente_Exp_UCI,
                                 Exp_CRI$numero_paciente_Exp_CRI,
                                 Exp_QUI$numero_paciente_Exp_QUI,
                                 RCP_UCI$numero_paciente_RCP_UCI,
                                 RCP_CRI$numero_paciente_RCP_CRI,
                                 RCP_QUI$numero_paciente_RCP_QUI,
                                 Obs_UCI$numero_paciente_Obs_UCI,
                                 Obs_CRI$numero_paciente_Obs_CRI,
                                 Obs_QUI$numero_paciente_Obs_QUI,
                                 Obs_ALT$numero_paciente_Obs_ALT,
                                 Obs_PLA$numero_paciente_Obs_PLA)

Numero_Paciente_Segundo_Paso[Numero_Paciente_Segundo_Paso==0]<- NA
Numero_Paciente_Segundo_Paso<- na.omit(Numero_Paciente_Segundo_Paso)

Donde_Llega_Segundo_Paso<- c(Con_ALT$nodo_Con_ALT,
                             Con_OBS$nodo_Con_OBS,
                             Con_UCI$nodo_Con_UCI,
                             Con_CRI$nodo_Con_CRI,
                             Con_QUI$nodo_Con_QUI,
                             Exp_UCI$nodo_Exp_UCI,
                             Exp_CRI$nodo_Exp_CRI,
                             Exp_QUI$nodo_Exp_QUI,
                             RCP_UCI$nodo_RCP_UCI,
                             RCP_CRI$nodo_RCP_CRI,
                             RCP_QUI$nodo_RCP_QUI,
                             Obs_UCI$nodo_Obs_UCI,
                             Obs_CRI$nodo_Obs_CRI,
                             Obs_QUI$nodo_Obs_QUI,
                             Obs_ALT$nodo_Obs_ALT,
                             Obs_PLA$nodo_Obs_PLA)

Donde_Llega_Segundo_Paso[Donde_Llega_Segundo_Paso==0]<- NA
Donde_Llega_Segundo_Paso<- na.omit(Donde_Llega_Segundo_Paso)

Cuando_Termina_Segundo_Paso<- c(Con_ALT$Momento_terminacion_Con_ALT,
                                Con_OBS$Momento_terminacion_Con_OBS,
                                Con_UCI$Momento_terminacion_Con_UCI,
                                Con_CRI$Momento_terminacion_Con_CRI,
                                Con_QUI$Momento_terminacion_Con_QUI,
                                Exp_UCI$Momento_terminacion_Exp_UCI,
                                Exp_CRI$Momento_terminacion_Exp_CRI,
                                Exp_QUI$Momento_terminacion_Exp_QUI,
                                RCP_UCI$Momento_terminacion_RCP_UCI,
                                RCP_CRI$Momento_terminacion_RCP_CRI,
                                RCP_QUI$Momento_terminacion_RCP_QUI,
```

```
                              Obs_UCI$Momento_terminacion_Obs_UCI,
                              Obs_CRI$Momento_terminacion_Obs_CRI,
                              Obs_QUI$Momento_terminacion_Obs_QUI,
                              Obs_ALT$Momento_terminacion_Obs_ALT,
                              Obs_PLA$Momento_terminacion_Obs_PLA)

Cuando_Termina_Segundo_Paso[Cuando_Termina_Segundo_Paso==0]<- NA
Cuando_Termina_Segundo_Paso<- na.omit(Cuando_Termina_Segundo_Paso)

Segundo_Paso<- data.frame(Numero_Paciente_Segundo_Paso=Numero_Paciente
_Segundo_Paso,
                          Donde_Llega_Segundo_Paso=Donde_Llega_Segundo
_Paso,
                          Cuando_Termina_Segundo_Paso=Cuando_Termina_S
egundo_Paso)

Segundo_Paso<- Segundo_Paso[with(Segundo_Paso, order(Segundo_Paso$Nume
ro_Paciente_Segundo_Paso)), ]

datos<- data.frame(Personas=Pacientes,Momento_Llegada_al_Sistema=Momen
to_Llegada_al_Sistema,Cuando_Termina_el_Triaje=Cuando_Termina_el_Triaj
e,Donde_Llega_Despues_Triaje=Donde_Llega_Despues_Triaje,Cuando_Llega=C
uando_Llega,Cuando_Termina=Cuando_Termina,Donde_Llega_Segundo_Paso=Seg
undo_Paso$Donde_Llega_Segundo_Paso,Cuando_Termina_Segundo_Paso=Segundo
_Paso$Cuando_Termina_Segundo_Paso)
}
############################### TERCER PASO ##################
{
#DE OBSERVACIÓN A UCI
contador_Obs_UCI_P3<- 0
nodo_Obs_UCI_P3<- 0
numero_paciente_Obs_UCI_P3<- 0
Momento_terminacion_Obs_UCI_P3<- 0
for (q in 1:length(Momento_Llegada_UCI1$P_UCI)) {
  if(Momento_Llegada_UCI1$Nodo_anterior_UCI[q]=="Obs"){
    if(Momento_Llegada_UCI1$SPLIT_UCI1[q]=="CONSULTA"){
      contador_Obs_UCI_P3<- contador_Obs_UCI_P3 + 1
      nodo_Obs_UCI_P3[contador_Obs_UCI_P3]<- "UCI"
      numero_paciente_Obs_UCI_P3[contador_Obs_UCI_P3]<- Momento_Llegad
a_UCI1$P_UCI[q]
      Momento_terminacion_Obs_UCI_P3[contador_Obs_UCI_P3]<- Pacientes_
UCI2$Momento_Terminación_UCI2[which(Pacientes_UCI2$P_UCI2==numero_paci
ente_Obs_UCI_P3[contador_Obs_UCI_P3])]
    }
  }
}
Porcentaje_Obs_UCI_Paso3<- (contador_Obs_UCI_P3/length(which(Pacientes
_Observacion$Nodo_anterior_Obs1=="CONSULTA")))*100
nodo_Obs_UCI_P3[nodo_Obs_UCI_P3==0]<- NA
nodo_Obs_UCI_P3<- na.omit(nodo_Obs_UCI_P3)
Porcentaje_Obs_UCI_Paso3[Porcentaje_Obs_UCI_Paso3==0]<- NA
Porcentaje_Obs_UCI_Paso3<- na.omit(Porcentaje_Obs_UCI_Paso3)
numero_paciente_Obs_UCI_P3[numero_paciente_Obs_UCI_P3==0]<- NA
numero_paciente_Obs_UCI_P3<- na.omit(numero_paciente_Obs_UCI_P3)
```

```
Momento_terminacion_Obs_UCI_P3[Momento_terminacion_Obs_UCI_P3==0]<- NA
Momento_terminacion_Obs_UCI_P3<- na.omit(Momento_terminacion_Obs_UCI_P
3)

Obs_UCI_P3<- data.frame(numero_paciente_Obs_UCI_P3=numero_paciente_Obs
_UCI_P3,
                        Momento_terminacion_Obs_UCI_P3=Momento_terminacio
n_Obs_UCI_P3,
                        nodo_Obs_UCI_P3=nodo_Obs_UCI_P3)
Obs_UCI_P3<- Obs_UCI_P3[with(Obs_UCI_P3, order(Obs_UCI_P3$numero_pacie
nte_Obs_UCI_P3)), ]

#DE OBSERVACIÓN A CRÍTICOS
contador_Obs_CRI_P3<- 0
nodo_Obs_CRI_P3<- 0
numero_paciente_Obs_CRI_P3<- 0
Momento_terminacion_Obs_CRI_P3<- 0
for (n in 1:length(Momento_Llegada_CRI1$P_CRI)) {
  if(Momento_Llegada_CRI1$Nodo_anterior_CRI[n]=="Obs"){
    if(Momento_Llegada_CRI1$SPLIT_CRI1[n]=="CONSULTA"){
      contador_Obs_CRI_P3<- contador_Obs_CRI_P3 + 1
      nodo_Obs_CRI_P3[contador_Obs_CRI_P3]<- "CRÍTICOS"
      numero_paciente_Obs_CRI_P3[contador_Obs_CRI_P3]<- Momento_Llegad
a_CRI1$P_CRI[n]
      Momento_terminacion_Obs_CRI_P3[contador_Obs_CRI_P3]<- Pacientes_
CRI2$Momento_Terminación_CRI2[which(Pacientes_CRI2$P_CRI2==numero_paci
ente_Obs_CRI_P3[contador_Obs_CRI_P3])]
    }
  }
}
Porcentaje_Obs_CRI_Paso3<- (contador_Obs_CRI_P3/length(which(Pacientes
_Observacion$Nodo_anterior_Obs1=="CONSULTA")))*100

nodo_Obs_CRI_P3[nodo_Obs_CRI_P3==0]<- NA
nodo_Obs_CRI_P3<- na.omit(nodo_Obs_CRI_P3)
Porcentaje_Obs_CRI_Paso3[Porcentaje_Obs_CRI_Paso3==0]<- NA
Porcentaje_Obs_CRI_Paso3<- na.omit(Porcentaje_Obs_CRI_Paso3)
numero_paciente_Obs_CRI_P3[numero_paciente_Obs_CRI_P3==0]<- NA
numero_paciente_Obs_CRI_P3<- na.omit(numero_paciente_Obs_CRI_P3)
Momento_terminacion_Obs_CRI_P3[Momento_terminacion_Obs_CRI_P3==0]<- NA
Momento_terminacion_Obs_CRI_P3<- na.omit(Momento_terminacion_Obs_CRI_P
3)

Obs_CRI_P3<- data.frame(numero_paciente_Obs_CRI_P3=numero_paciente_Obs
_CRI_P3,
                        Momento_terminacion_Obs_CRI_P3=Momento_terminacio
n_Obs_CRI_P3,
                        nodo_Obs_CRI_P3=nodo_Obs_CRI_P3)
Obs_CRI_P3<- Obs_CRI_P3[with(Obs_CRI_P3, order(Obs_CRI_P3$numero_pacie
nte_Obs_CRI_P3)), ]

#DE OBSERVACIÓN A QUIROFANO
contador_Obs_QUI_P3<- 0
nodo_Obs_QUI_P3<- 0
```

```
numero_paciente_Obs_QUI_P3<- 0
Momento_terminacion_Obs_QUI_P3<- 0
for (w in 1:length(Momento_Llegada_QUI$P_QUI)) {
  if(Momento_Llegada_QUI$Nodo_anterior_QUI[w]=="Obs"){
    if(Momento_Llegada_QUI$Obs_QUI_SPLIT[w]=="CONSULTA"){
      contador_Obs_QUI_P3<- contador_Obs_QUI_P3 + 1
      nodo_Obs_QUI_P3[contador_Obs_QUI_P3]<- "QUIROFANO"
      numero_paciente_Obs_QUI_P3[contador_Obs_QUI_P3]<- Momento_Llegad
a_QUI$P_QUI[w]
      Momento_terminacion_Obs_QUI_P3[contador_Obs_QUI_P3]<- Pacientes_
QUI2$Momento_Terminación_QUI2[which(Pacientes_QUI2$P_QUI2==numero_paci
ente_Obs_QUI_P3[contador_Obs_QUI_P3])]
    }
  }
}
Porcentaje_Obs_QUI_Paso3<- (contador_Obs_QUI_P3/length(which(Pacientes
_Observacion$Nodo_anterior_Obs1=="CONSULTA")))*100

Obs_QUI_P3<- data.frame(numero_paciente_Obs_QUI_P3=numero_paciente_Obs
_QUI_P3,
                        Momento_terminacion_Obs_QUI_P3=Momento_termina
cion_Obs_QUI_P3,
                      nodo_Obs_QUI_P3=nodo_Obs_QUI_P3)
Obs_QUI_P3<- Obs_QUI_P3[with(Obs_QUI_P3, order(Obs_QUI_P3$numero_pacie
nte_Obs_QUI_P3)), ]

#DE OBSERVACIÓN A ALTA
contador_Obs_ALT_P3<- 0
nodo_Obs_ALT_P3<- 0
numero_paciente_Obs_ALT_P3<- 0
Momento_terminacion_Obs_ALT_P3<- 0
for (e in 1:length(Momento_Llegada_ALT$P_ALT)) {
  if(Momento_Llegada_ALT$Nodo_anterior_ALT[e]=="Obs"){
    if(Momento_Llegada_ALT$Obs_ALT_SPLIT[e]=="CONSULTA"){
      contador_Obs_ALT_P3<- contador_Obs_ALT_P3 + 1
      nodo_Obs_ALT_P3[contador_Obs_ALT_P3]<- "ALTA"
      numero_paciente_Obs_ALT_P3[contador_Obs_ALT_P3]<- Momento_Llegad
a_ALT$P_ALT[e]
      Momento_terminacion_Obs_ALT_P3[contador_Obs_ALT_P3]<- Momento_Ll
egada_ALT$Llegada_ALT[e]
    }
  }
}
Porcentaje_Obs_ALT_Paso3<- (contador_Obs_ALT_P3/length(which(Pacientes
_Observacion$Nodo_anterior_Obs1=="CONSULTA")))*100

Obs_ALT_P3<- data.frame(numero_paciente_Obs_ALT_P3=numero_paciente_Obs
_ALT_P3,
                        Momento_terminacion_Obs_ALT_P3=Momento_termina
cion_Obs_ALT_P3,
                        nodo_Obs_ALT_P3=nodo_Obs_ALT_P3)
Obs_ALT_P3<- Obs_ALT_P3[with(Obs_ALT_P3, order(Obs_ALT_P3$numero_pacie
nte_Obs_ALT_P3)), ]
```

```
#DE OBSERVACIÓN A PLANTA
contador_Obs_PLA_P3<- 0
nodo_Obs_PLA_P3<- 0
numero_paciente_Obs_PLA_P3<- 0
Momento_terminacion_Obs_PLA_P3<- 0
for (r in 1:length(Momento_Llegada_PLA$P_PLA)) {
  if(Momento_Llegada_PLA$Nodo_anterior_PLA[r]=="OBS"){
    if(Momento_Llegada_PLA$SPLIT[r]=="CONSULTA"){
      contador_Obs_PLA_P3<- contador_Obs_PLA_P3 + 1
      nodo_Obs_PLA_P3[contador_Obs_PLA_P3]<- "PLANTA"
      numero_paciente_Obs_PLA_P3[contador_Obs_PLA_P3]<- Momento_Llegad
a_PLA$P_PLA[r]
      Momento_terminacion_Obs_PLA_P3[contador_Obs_PLA_P3]<- Pacientes_
PLA2$Momento_Terminación_PLA2[which(Pacientes_PLA2$P_PLA2==numero_paci
ente_Obs_PLA_P3[contador_Obs_PLA_P3])]
    }
  }
}
Porcentaje_Obs_PLA_Paso3<- (contador_Obs_PLA_P3/length(which(Pacientes
_Observacion$Nodo_anterior_Obs1=="CONSULTA")))*100

Obs_PLA_P3<- data.frame(numero_paciente_Obs_PLA_P3=numero_paciente_Obs
_PLA_P3,
                        Momento_terminacion_Obs_PLA_P3=Momento_termina
cion_Obs_PLA_P3,
                      nodo_Obs_PLA_P3=nodo_Obs_PLA_P3)
Obs_PLA_P3<- Obs_PLA_P3[with(Obs_PLA_P3, order(Obs_PLA_P3$numero_pacie
nte_Obs_PLA_P3)), ]

#DE UCI A PLANTA
contador_UCI_PLA_P3<- 0
nodo_UCI_PLA_P3<- 0
numero_paciente_UCI_PLA_P3<- 0
Momento_terminacion_UCI_PLA_P3<- 0
for (r in 1:length(Momento_Llegada_PLA$P_PLA)) {
  if(Momento_Llegada_PLA$Nodo_anterior_PLA[r]=="UCI"){
    if(Momento_Llegada_PLA$ANTERIOR_P4[r]!="QUI_P3"){
      if(Momento_Llegada_PLA$SPLIT[r]=="NADA" | Momento_Llegada_PLA$SP
LIT[r]=="NINGUNO"){
        contador_UCI_PLA_P3<- contador_UCI_PLA_P3 + 1
        nodo_UCI_PLA_P3[contador_UCI_PLA_P3]<- "PLANTA"
        numero_paciente_UCI_PLA_P3[contador_UCI_PLA_P3]<- Momento_Lleg
ada_PLA$P_PLA[r]
        Momento_terminacion_UCI_PLA_P3[contador_UCI_PLA_P3]<- Paciente
s_PLA2$Momento_Terminación_PLA2[which(Pacientes_PLA2$P_PLA2==numero_pa
ciente_UCI_PLA_P3[contador_UCI_PLA_P3])]
      }
    }
  }
}
Porcentaje_UCI_PLA_Paso3<- 100

UCI_PLA_P3<- data.frame(numero_paciente_UCI_PLA_P3=numero_paciente_UCI
_PLA_P3,
```

```
                          Momento_terminacion_UCI_PLA_P3=Momento_termina
cion_UCI_PLA_P3,
                          nodo_UCI_PLA_P3=nodo_UCI_PLA_P3)
UCI_PLA_P3<- UCI_PLA_P3[with(UCI_PLA_P3, order(UCI_PLA_P3$numero_pacie
nte_UCI_PLA_P3)), ]

#DE CRÍTICOS A PLANTA
contador_CRI_PLA_P3<- 0
nodo_CRI_PLA_P3<- 0
numero_paciente_CRI_PLA_P3<- 0
Momento_terminacion_CRI_PLA_P3<- 0
for (r in 1:length(Momento_Llegada_PLA$P_PLA)) {
  if(Momento_Llegada_PLA$Nodo_anterior_PLA[r]=="CRI"){
    if(Momento_Llegada_PLA$ANTERIOR_P4[r]!="QUI_P3"){
      if(Momento_Llegada_PLA$SPLIT[r]=="NADA" | Momento_Llegada_PLA$SP
LIT[r]=="NINGUNO"){
        contador_CRI_PLA_P3<- contador_CRI_PLA_P3 + 1
        nodo_CRI_PLA_P3[contador_CRI_PLA_P3]<- "PLANTA"
        numero_paciente_CRI_PLA_P3[contador_CRI_PLA_P3]<- Momento_Lleg
ada_PLA$P_PLA[r]
        Momento_terminacion_CRI_PLA_P3[contador_CRI_PLA_P3]<- Paciente
s_PLA2$Momento_Terminación_PLA2[which(Pacientes_PLA2$P_PLA2==numero_pa
ciente_CRI_PLA_P3[contador_CRI_PLA_P3])]
    }
    }
  }
}
Porcentaje_CRI_PLA_Paso3<- 100

CRI_PLA_P3<- data.frame(numero_paciente_CRI_PLA_P3=numero_paciente_CRI
_PLA_P3,
                          Momento_terminacion_CRI_PLA_P3=Momento_termina
cion_CRI_PLA_P3,
                          nodo_CRI_PLA_P3=nodo_CRI_PLA_P3)
CRI_PLA_P3<- CRI_PLA_P3[with(CRI_PLA_P3, order(CRI_PLA_P3$numero_pacie
nte_CRI_PLA_P3)), ]

#DE QUIROFANO A PLANTA
contador_QUI_PLA_P3<- 0
nodo_QUI_PLA_P3<- 0
numero_paciente_QUI_PLA_P3<- 0
Momento_terminacion_QUI_PLA_P3<- 0
for (r in 1:length(Momento_Llegada_PLA$P_PLA)) {
  if(Momento_Llegada_PLA$Nodo_anterior_PLA[r]=="QUI_P3"){
    if(Momento_Llegada_PLA$SPLIT[r]=="NADA" | Momento_Llegada_PLA$SPLI
T[r]=="NINGUNO"){
      contador_QUI_PLA_P3<- contador_QUI_PLA_P3 + 1
      nodo_QUI_PLA_P3[contador_QUI_PLA_P3]<- "PLANTA"
      numero_paciente_QUI_PLA_P3[contador_QUI_PLA_P3]<- Momento_Llegad
a_PLA$P_PLA[r]
      Momento_terminacion_QUI_PLA_P3[contador_QUI_PLA_P3]<- Pacientes_
PLA2$Momento_Terminación_PLA2[which(Pacientes_PLA2$P_PLA2==numero_paci
ente_QUI_PLA_P3[contador_QUI_PLA_P3])]
    }
```

```
  }
}
Porcentaje_QUI_PLA_Paso3<- (contador_QUI_PLA_P3/length(which(Momento_L
legada_QUI$Obs_QUI_SPLIT=="NADA" | Momento_Llegada_QUI$Obs_QUI_SPLIT==
"NINGUNO")))*100

QUI_PLA_P3<- data.frame(numero_paciente_QUI_PLA_P3=numero_paciente_QUI
_PLA_P3,
                        Momento_terminacion_QUI_PLA_P3=Momento_termina
cion_QUI_PLA_P3,
                        nodo_QUI_PLA_P3=nodo_QUI_PLA_P3)
QUI_PLA_P3<- QUI_PLA_P3[with(QUI_PLA_P3, order(QUI_PLA_P3$numero_pacie
nte_QUI_PLA_P3)), ]

#DE QUIROFANO A CRÍTICOS
contador_QUI_CRI_P3<- 0
nodo_QUI_CRI_P3<- 0
numero_paciente_QUI_CRI_P3<- 0
Momento_terminacion_QUI_CRI_P3<- 0
for (n in 1:length(Momento_Llegada_CRI1$P_CRI)) {
  if(Momento_Llegada_CRI1$Nodo_anterior_CRI[n]=="QUI_P3"){
    if(Momento_Llegada_CRI1$SPLIT_CRI1[n]=="NADA" | Momento_Llegada_CR
I1$SPLIT_CRI1[n]=="NINGUNO"){
      contador_QUI_CRI_P3<- contador_QUI_CRI_P3 + 1
      nodo_QUI_CRI_P3[contador_QUI_CRI_P3]<- "CRÍTICOS"
      numero_paciente_QUI_CRI_P3[contador_QUI_CRI_P3]<- Momento_Llegad
a_CRI1$P_CRI[n]
      Momento_terminacion_QUI_CRI_P3[contador_QUI_CRI_P3]<- Pacientes_
CRI2$Momento_Terminación_CRI2[which(Pacientes_CRI2$P_CRI2==numero_paci
ente_QUI_CRI_P3[contador_QUI_CRI_P3])]
  }
}
}
Porcentaje_QUI_CRI_Paso3<- (contador_QUI_CRI_P3/length(which(Momento_L
legada_QUI$Obs_QUI_SPLIT=="NADA" | Momento_Llegada_QUI$Obs_QUI_SPLIT==
"NINGUNO")))*100

QUI_CRI_P3<- data.frame(numero_paciente_QUI_CRI_P3=numero_paciente_QUI
_CRI_P3,
                        Momento_terminacion_QUI_CRI_P3=Momento_termina
cion_QUI_CRI_P3,
                      nodo_QUI_CRI_P3=nodo_QUI_CRI_P3)
QUI_CRI_P3<- QUI_CRI_P3[with(QUI_CRI_P3, order(QUI_CRI_P3$numero_pacie
nte_QUI_CRI_P3)), ]

#DE QUIROFANO A UCI
contador_QUI_UCI_P3<- 0
nodo_QUI_UCI_P3<- 0
numero_paciente_QUI_UCI_P3<- 0
Momento_terminacion_QUI_UCI_P3<- 0
for (q in 1:length(Momento_Llegada_UCI1$P_UCI)) {
  if(Momento_Llegada_UCI1$Nodo_anterior_UCI[q]=="QUI_P3"){
    if(Momento_Llegada_UCI1$SPLIT_UCI1[q]=="NADA" | Momento_Llegada_UC
I1$SPLIT_UCI1[q]=="NINGUNO"){
```

```
      contador_QUI_UCI_P3<- contador_QUI_UCI_P3 + 1
      nodo_QUI_UCI_P3[contador_QUI_UCI_P3]<- "UCI"
      numero_paciente_QUI_UCI_P3[contador_QUI_UCI_P3]<- Momento_Llegad
a_UCI1$P_UCI[q]
      Momento_terminacion_QUI_UCI_P3[contador_QUI_UCI_P3]<- Pacientes_
UCI2$Momento_Terminación_UCI2[which(Pacientes_UCI2$P_UCI2==numero_paci
ente_QUI_UCI_P3[contador_QUI_UCI_P3])]
    }
  }
}
Porcentaje_QUI_UCI_Paso3<- (contador_QUI_UCI_P3/length(which(Momento_L
legada_QUI$Obs_QUI_SPLIT=="NADA" | Momento_Llegada_QUI$Obs_QUI_SPLIT==
"NINGUNO")))*100

nodo_QUI_UCI_P3[nodo_QUI_UCI_P3==0]<- NA
nodo_QUI_UCI_P3<- na.omit(nodo_QUI_UCI_P3)
Porcentaje_QUI_UCI_Paso3[Porcentaje_QUI_UCI_Paso3==0]<- NA
Porcentaje_QUI_UCI_Paso3<- na.omit(Porcentaje_QUI_UCI_Paso3)
numero_paciente_QUI_UCI_P3[numero_paciente_QUI_UCI_P3==0]<- NA
numero_paciente_QUI_UCI_P3<- na.omit(numero_paciente_QUI_UCI_P3)
Momento_terminacion_QUI_UCI_P3[Momento_terminacion_QUI_UCI_P3==0]<- NA
Momento_terminacion_QUI_UCI_P3<- na.omit(Momento_terminacion_QUI_UCI_P
3)

QUI_UCI_P3<- data.frame(numero_paciente_QUI_UCI_P3=numero_paciente_QUI
_UCI_P3,
                        Momento_terminacion_QUI_UCI_P3=Momento_termina
cion_QUI_UCI_P3,
                        nodo_QUI_UCI_P3=nodo_QUI_UCI_P3)
QUI_UCI_P3<- QUI_UCI_P3[with(QUI_UCI_P3, order(QUI_UCI_P3$numero_pacie
nte_QUI_UCI_P3)), ]
Origen_Paso3<- c("OBSERVACIÓN","OBSERVACIÓN","OBSERVACIÓN","OBSERVACIÓ
N","OBSERVACIÓN","UCI","CRÍTICOS","QUIROFANO","QUIROFANO","QUIROFANO")
Destino_Paso3<- c("UCI","CRÍTICOS","QUIROFANO","ALTA","PLANTA","PLANTA
",
                  "PLANTA","PLANTA","CRÍTICOS","UCI")

Porcentaje_Paso3<- c(Porcentaje_Obs_UCI_Paso3,Porcentaje_Obs_CRI_Paso3
,
                     Porcentaje_Obs_QUI_Paso3,Porcentaje_Obs_ALT_Paso3
,
                     Porcentaje_Obs_PLA_Paso3,Porcentaje_UCI_PLA_Paso3
,
                     Porcentaje_CRI_PLA_Paso3,Porcentaje_QUI_PLA_Paso3
,
                     Porcentaje_QUI_CRI_Paso3,Porcentaje_QUI_UCI_Paso3
)

Numero_Paciente_Tercer_Paso<- c(Obs_UCI_P3$numero_paciente_Obs_UCI_P3,
                                Obs_CRI_P3$numero_paciente_Obs_CRI_P3,
                                Obs_QUI_P3$numero_paciente_Obs_QUI_P3,
                                Obs_ALT_P3$numero_paciente_Obs_ALT_P3,
                                Obs_PLA_P3$numero_paciente_Obs_PLA_P3,
                                UCI_PLA_P3$numero_paciente_UCI_PLA_P3,
```

```
                                CRI_PLA_P3$numero_paciente_CRI_PLA_P3,
                                QUI_PLA_P3$numero_paciente_QUI_PLA_P3,
                                QUI_CRI_P3$numero_paciente_QUI_CRI_P3,
                                QUI_UCI_P3$numero_paciente_QUI_UCI_P3)

Donde_Llega_Tercer_Paso<-c(Obs_UCI_P3$nodo_Obs_UCI_P3,
                           Obs_CRI_P3$nodo_Obs_CRI_P3,
                           Obs_QUI_P3$nodo_Obs_QUI_P3,
                           Obs_ALT_P3$nodo_Obs_ALT_P3,
                           Obs_PLA_P3$nodo_Obs_PLA_P3,
                           UCI_PLA_P3$nodo_UCI_PLA_P3,
                           CRI_PLA_P3$nodo_CRI_PLA_P3,
                           QUI_PLA_P3$nodo_QUI_PLA_P3,
                           QUI_CRI_P3$nodo_QUI_CRI_P3,
                           QUI_UCI_P3$nodo_QUI_UCI_P3)

Cuando_Termina_Tercer_Paso<- c(Obs_UCI_P3$Momento_terminacion_Obs_UCI_
P3,
                               Obs_CRI_P3$Momento_terminacion_Obs_CRI_
P3,
                               Obs_QUI_P3$Momento_terminacion_Obs_QUI_
P3,
                               Obs_ALT_P3$Momento_terminacion_Obs_ALT_
P3,
                               Obs_PLA_P3$Momento_terminacion_Obs_PLA_
P3,
                               UCI_PLA_P3$Momento_terminacion_UCI_PLA_
P3,
                               CRI_PLA_P3$Momento_terminacion_CRI_PLA_
P3,
                               QUI_PLA_P3$Momento_terminacion_QUI_PLA_
P3,
                               QUI_CRI_P3$Momento_terminacion_QUI_CRI_
P3,
                               QUI_UCI_P3$Momento_terminacion_QUI_UCI_
P3)

Tercer_Paso<- data.frame(Numero_Paciente_Tercer_Paso=Numero_Paciente_T
ercer_Paso,Donde_Llega_Tercer_Paso=Donde_Llega_Tercer_Paso,Cuando_Term
ina_Tercer_Paso=Cuando_Termina_Tercer_Paso)

Tercer_Paso<- Tercer_Paso[with(Tercer_Paso, order(Tercer_Paso$Numero_P
aciente_Tercer_Paso)), ]

ki<- nrow(Tercer_Paso)+1

Tercer_Paso[ki:nrow(datos),1]<- 0

contar_1<- 0; contar_2<- 0; contar1_1<-0; contar1_2<- 0; personis_1<-
0
personis_2<- 0; personis_3<-0; personis_4<-0; diferencia_1<-0; diferen
cia_2<- 0
print<- 0; rango_inicio<- 0; rango_fin<- 0; Donde1<-0; Donde2<- 0; Don
de3<-0
```

```
Donde4<-0; Cuando1<-0; Cuando2<-0; Cuando3<- 0; Cuando4<-0; longitud5<
- 0
longitud6<-0; longitud7<-0; longitud8<- 0; longitud9<- 0

for (i in 2:sum(nclientes)) {
  if(Tercer_Paso$Numero_Paciente_Tercer_Paso[i]-Tercer_Paso$Numero_Pac
iente_Tercer_Paso[i-1]!=1){
    contar_1<-contar_1 +1
    diferencia_1[contar_1]<- Tercer_Paso$Numero_Paciente_Tercer_Paso[i
] - Tercer_Paso$Numero_Paciente_Tercer_Paso[i-1] - 1
    if(diferencia_1[contar_1]==1){
      contar1_1<- contar1_1+1
      personis_1[contar1_1]<- Tercer_Paso$Numero_Paciente_Tercer_Paso[
i-1]+1
      Donde1[contar1_1]<- NA
      Cuando1[contar1_1]<- NA
    }
    personis_1[personis_1==0]<- NA
    personis_1<- na.omit(personis_1)
  }
}
for (i in 2:sum(nclientes)) {
  if(Tercer_Paso$Numero_Paciente_Tercer_Paso[i]!=0){
  if(Tercer_Paso$Numero_Paciente_Tercer_Paso[i]-Tercer_Paso$Numero_Pac
iente_Tercer_Paso[i-1]!=1){
    contar_2<-contar_2 +1
    diferencia_2[contar_2]<- Tercer_Paso$Numero_Paciente_Tercer_Paso[i
] - Tercer_Paso$Numero_Paciente_Tercer_Paso[i-1] - 1
    if(diferencia_2[contar_2]!=1){
      contar1_2<- contar1_2+1
      añadir4<- contar1_2  + longitud5
      longitud5<- añadir4 + diferencia_2[contar_2] -1
      rango_inicio<- Tercer_Paso$Numero_Paciente_Tercer_Paso[i-1] + 1
      rango_fin<- Tercer_Paso$Numero_Paciente_Tercer_Paso[i] - 1
      personis_2[añadir4:longitud5]<- rango_inicio:rango_fin
      personis_2[personis_2==0]<- NA
      Donde2[añadir4:longitud5]<- 1
      Cuando2[añadir4:longitud5]<- 1
      Donde2[Donde2==0]<- NA
      Cuando2[Cuando2==0]<- NA
    }
  }
    if(Tercer_Paso$Numero_Paciente_Tercer_Paso[i]-Tercer_Paso$Numero_P
aciente_Tercer_Paso[i-1]==0){
      print<- Tercer_Paso$Numero_Paciente_Tercer_Paso[i]
    }
    }
}
personis_2<- na.omit(personis_2)
Donde2<- na.omit(Donde2)
Cuando2<- na.omit(Cuando2)
Donde2[Donde2==1]<- NA
Cuando2[Cuando2==1]<- NA
```

```
Tercer_Paso<- data.frame(Numero_Paciente_Tercer_Paso=Numero_Paciente_T
ercer_Paso,Donde_Llega_Tercer_Paso=Donde_Llega_Tercer_Paso,Cuando_Term
ina_Tercer_Paso=Cuando_Termina_Tercer_Paso)

Tercer_Paso<- Tercer_Paso[with(Tercer_Paso, order(Tercer_Paso$Numero_P
aciente_Tercer_Paso)), ]
if(min(Tercer_Paso$Numero_Paciente_Tercer_Paso)>1){
  longitud6<- min(Tercer_Paso$Numero_Paciente_Tercer_Paso)-1
  personis_3[1:longitud6]<- 1:longitud6
  Donde3[1:longitud6]<- NA
  Cuando3[1:longitud6]<- NA
}
if(max(Tercer_Paso$Numero_Paciente_Tercer_Paso)<sum(nclientes)){
  longitud7<- max(Tercer_Paso$Numero_Paciente_Tercer_Paso)+1
  longitud9<- max(Tercer_Paso$Numero_Paciente_Tercer_Paso)
  longitud8<- sum(nclientes) - longitud9
  personis_4[1:longitud8]<- longitud7:sum(nclientes)
  Donde4[1:longitud8]<- NA
  Cuando4[1:longitud8]<- NA
}
if(length(Donde4)==1 & personis_4==0){
   Donde4<- NA
   Donde4<- na.omit(Donde4)
}
if(length(Cuando4)==1 & personis_4==0){
   Cuando4<- NA
   Cuando4<- na.omit(Cuando4)
}
if(length(Donde3)==1 & personis_3==0){
   Donde3<- NA
   Donde3<- na.omit(Donde3)
}
if(length(Cuando3)==1 & personis_3==0){
   Cuando3<- NA
   Cuando3<- na.omit(Cuando3)
}
if(length(personis_3)==1 & personis_3==0){
  personis_3<- NA
  personis_3<- na.omit(personis_3)
}
if(length(personis_4)==1 & personis_4==0){
  personis_4<- NA
  personis_4<- na.omit(personis_4)
}
unir_personis<- c(personis_1,personis_2,personis_3,personis_4)
unir_Donde<- c(Donde1,Donde2,Donde3,Donde4)
unir_Cuando<- c(Cuando1,Cuando2,Cuando3,Cuando4)

Tercer_Paso<- data.frame(Numero_Paciente_Tercer_Paso=c(Numero_Paciente
_Tercer_Paso,unir_personis),Donde_Llega_Tercer_Paso=c(Donde_Llega_Terc
er_Paso,unir_Donde),Cuando_Termina_Tercer_Paso=c(Cuando_Termina_Tercer
_Paso,unir_Cuando))

Tercer_Paso<- Tercer_Paso[with(Tercer_Paso, order(Tercer_Paso$Numero_P
```

```
aciente_Tercer_Paso)), ]

datos<- data.frame(Personas=Pacientes,Momento_Llegada_al_Sistema=Momen
to_Llegada_al_Sistema,Cuando_Termina_el_Triaje=Cuando_Termina_el_Triaj
e,Donde_Llega_Despues_Triaje=Donde_Llega_Despues_Triaje,Cuando_Llega=C
uando_Llega,Cuando_Termina=Cuando_Termina,Donde_Llega_Segundo_Paso=Seg
undo_Paso$Donde_Llega_Segundo_Paso,Cuando_Termina_Segundo_Paso=Segundo
_Paso$Cuando_Termina_Segundo_Paso,Donde_Llega_Tercer_Paso=Tercer_Paso$
Donde_Llega_Tercer_Paso,Cuando_Termina_Tercer_Paso=Tercer_Paso$Cuando_
Termina_Tercer_Paso)
if(is.logical(Porcentaje_Obs_UCI_Paso3)==FALSE){
  Porcentaje_Paso3[1]<- 0
}
if(is.logical(Porcentaje_Obs_CRI_Paso3)==FALSE){
  Porcentaje_Paso3[2]<- 0
}
if(is.logical(Porcentaje_Obs_QUI_Paso3)==FALSE){
  Porcentaje_Paso3[3]<- 0
}
if(is.logical(Porcentaje_Obs_ALT_Paso3)==FALSE){
  Porcentaje_Paso3[4]<- 0
}
if(is.logical(Porcentaje_Obs_PLA_Paso3)==FALSE){
  Porcentaje_Paso3[5]<- 0
}
if(is.logical(Porcentaje_UCI_PLA_Paso3)==FALSE){
  Porcentaje_Paso3[6]<- 0
}
if(is.logical(Porcentaje_CRI_PLA_Paso3)==FALSE){
  Porcentaje_Paso3[7]<- 0
}
if(is.logical(Porcentaje_QUI_PLA_Paso3)==FALSE){
  Porcentaje_Paso3[8]<- 0
}
if(is.logical(Porcentaje_QUI_CRI_Paso3)==FALSE){
  Porcentaje_Paso3[9]<- 0
}
if(is.logical(Porcentaje_QUI_UCI_Paso3)==FALSE){
  Porcentaje_Paso3[10]<- 0
}
}
############################### CUARTO PASO ################
{
#DE UCI A PLANTA
contador_UCI_PLA_P4<- 0
nodo_UCI_PLA_P4<- 0
numero_paciente_UCI_PLA_P4<- 0
Momento_terminacion_UCI_PLA_P4<- 0
contador_UCI_PLA_P4_2<-0
nodo_UCI_PLA_P4_2<-0
numero_paciente_UCI_PLA_P4_2<- 0
Momento_terminacion_UCI_PLA_P4_2<-0
for (r in 1:length(Momento_Llegada_PLA$P_PLA)) {
  if(Momento_Llegada_PLA$Nodo_anterior_PLA[r]=="UCI"){
```

```
    if(Momento_Llegada_PLA$ANTERIOR_P4[r]=="QUI_P3"){
      if(Momento_Llegada_PLA$SPLIT[r]=="NADA" | Momento_Llegada_PLA$SP
LIT[r]=="NINGUNO"){
        contador_UCI_PLA_P4<- contador_UCI_PLA_P4 + 1
        nodo_UCI_PLA_P4[contador_UCI_PLA_P4]<- "PLANTA"
        numero_paciente_UCI_PLA_P4[contador_UCI_PLA_P4]<- Momento_Lleg
ada_PLA$P_PLA[r]
        Momento_terminacion_UCI_PLA_P4[contador_UCI_PLA_P4]<- Paciente
s_PLA2$Momento_Terminación_PLA2[which(Pacientes_PLA2$P_PLA2==numero_pa
ciente_UCI_PLA_P4[contador_UCI_PLA_P4])]
      }
    }
    if(Momento_Llegada_PLA$ANTERIOR_P4[r]=="Obs"){
      if(Momento_Llegada_PLA$SPLIT[r]=="CONSULTA"){
        contador_UCI_PLA_P4_2<- contador_UCI_PLA_P4_2 + 1
        nodo_UCI_PLA_P4_2[contador_UCI_PLA_P4_2]<- "PLANTA"
        numero_paciente_UCI_PLA_P4_2[contador_UCI_PLA_P4_2]<- Momento_
Llegada_PLA$P_PLA[r]
        Momento_terminacion_UCI_PLA_P4_2[contador_UCI_PLA_P4_2]<- Paci
entes_PLA2$Momento_Terminación_PLA2[which(Pacientes_PLA2$P_PLA2==numer
o_paciente_UCI_PLA_P4_2[contador_UCI_PLA_P4_2])]
      }
    }
  }
}
Porcentaje_UCI_PLA_Paso4<- 100

nodo_UCI_PLA_P4[nodo_UCI_PLA_P4==0]<- NA
nodo_UCI_PLA_P4<- na.omit(nodo_UCI_PLA_P4)
numero_paciente_UCI_PLA_P4[numero_paciente_UCI_PLA_P4==0]<- NA
numero_paciente_UCI_PLA_P4<- na.omit(numero_paciente_UCI_PLA_P4)
Momento_terminacion_UCI_PLA_P4[Momento_terminacion_UCI_PLA_P4==0]<- NA
Momento_terminacion_UCI_PLA_P4<- na.omit(Momento_terminacion_UCI_PLA_P
4)

nodo_UCI_PLA_P4_2[nodo_UCI_PLA_P4_2==0]<- NA
nodo_UCI_PLA_P4_2<- na.omit(nodo_UCI_PLA_P4_2)
Porcentaje_UCI_PLA_Paso4[Porcentaje_UCI_PLA_Paso4==0]<- NA
Porcentaje_UCI_PLA_Paso4<- na.omit(Porcentaje_UCI_PLA_Paso4)
numero_paciente_UCI_PLA_P4_2[numero_paciente_UCI_PLA_P4_2==0]<- NA
numero_paciente_UCI_PLA_P4_2<- na.omit(numero_paciente_UCI_PLA_P4_2)
Momento_terminacion_UCI_PLA_P4_2[Momento_terminacion_UCI_PLA_P4_2==0]<
- NA
Momento_terminacion_UCI_PLA_P4_2<- na.omit(Momento_terminacion_UCI_PLA
_P4_2)

UCI_PLA_P4<- data.frame(numero_paciente_UCI_PLA_P4=c(numero_paciente_U
CI_PLA_P4,numero_paciente_UCI_PLA_P4_2),
                        Momento_terminacion_UCI_PLA_P4=c(Momento_termi
nacion_UCI_PLA_P4,Momento_terminacion_UCI_PLA_P4_2),
                        nodo_UCI_PLA_P4=c(nodo_UCI_PLA_P4,nodo_UCI_PLA
_P4_2))
UCI_PLA_P4<- UCI_PLA_P4[with(UCI_PLA_P4, order(UCI_PLA_P4$numero_pacie
nte_UCI_PLA_P4)), ]
```

```
#DE CRÍTICOS A PLANTA
contador_CRI_PLA_P4<- 0
nodo_CRI_PLA_P4<- 0
numero_paciente_CRI_PLA_P4<- 0
Momento_terminacion_CRI_PLA_P4<- 0
contador_CRI_PLA_P4_2<-0
nodo_CRI_PLA_P4_2<-0
numero_paciente_CRI_PLA_P4_2<- 0
Momento_terminacion_CRI_PLA_P4_2<-0
for (r in 1:length(Momento_Llegada_PLA$P_PLA)) {
  if(Momento_Llegada_PLA$Nodo_anterior_PLA[r]=="CRI"){
    if(Momento_Llegada_PLA$ANTERIOR_P4[r]=="QUI_P3"){
      if(Momento_Llegada_PLA$SPLIT[r]=="NADA" | Momento_Llegada_PLA$SP
LIT[r]=="NINGUNO"){
        contador_CRI_PLA_P4<- contador_CRI_PLA_P4 + 1
        nodo_CRI_PLA_P4[contador_CRI_PLA_P4]<- "PLANTA"
        numero_paciente_CRI_PLA_P4[contador_CRI_PLA_P4]<- Momento_Lleg
ada_PLA$P_PLA[r]
        Momento_terminacion_CRI_PLA_P4[contador_CRI_PLA_P4]<- Paciente
s_PLA2$Momento_Terminación_PLA2[which(Pacientes_PLA2$P_PLA2==numero_pa
ciente_CRI_PLA_P4[contador_CRI_PLA_P4])]
      }
    }
    if(Momento_Llegada_PLA$ANTERIOR_P4[r]=="Obs"){
      if(Momento_Llegada_PLA$SPLIT[r]=="CONSULTA"){
        contador_CRI_PLA_P4_2<- contador_CRI_PLA_P4_2 + 1
        nodo_CRI_PLA_P4_2[contador_CRI_PLA_P4_2]<- "PLANTA"
        numero_paciente_CRI_PLA_P4_2[contador_CRI_PLA_P4_2]<- Momento_
Llegada_PLA$P_PLA[r]
        Momento_terminacion_CRI_PLA_P4_2[contador_CRI_PLA_P4_2]<- Paci
entes_PLA2$Momento_Terminación_PLA2[which(Pacientes_PLA2$P_PLA2==numer
o_paciente_CRI_PLA_P4_2[contador_CRI_PLA_P4_2])]
      }
    }
  }
}
Porcentaje_CRI_PLA_Paso4<- 100

nodo_CRI_PLA_P4_2[nodo_CRI_PLA_P4_2==0]<- NA
nodo_CRI_PLA_P4_2<- na.omit(nodo_CRI_PLA_P4_2)
Porcentaje_CRI_PLA_Paso4[Porcentaje_CRI_PLA_Paso4==0]<- NA
Porcentaje_CRI_PLA_Paso4<- na.omit(Porcentaje_CRI_PLA_Paso4)
numero_paciente_CRI_PLA_P4_2[numero_paciente_CRI_PLA_P4_2==0]<- NA
numero_paciente_CRI_PLA_P4_2<- na.omit(numero_paciente_CRI_PLA_P4_2)
Momento_terminacion_CRI_PLA_P4_2[Momento_terminacion_CRI_PLA_P4_2==0]<
- NA
Momento_terminacion_CRI_PLA_P4_2<- na.omit(Momento_terminacion_CRI_PLA
_P4_2)

CRI_PLA_P4<- data.frame(numero_paciente_CRI_PLA_P4=c(numero_paciente_C
RI_PLA_P4,numero_paciente_CRI_PLA_P4_2),
                        Momento_terminacion_CRI_PLA_P4=c(Momento_termi
nacion_CRI_PLA_P4,Momento_terminacion_CRI_PLA_P4_2),
```

```
                        nodo_CRI_PLA_P4=c(nodo_CRI_PLA_P4,nodo_CRI_PLA
_P4_2))
CRI_PLA_P4<- CRI_PLA_P4[with(CRI_PLA_P4, order(CRI_PLA_P4$numero_pacie
nte_CRI_PLA_P4)), ]

#DE QUIROFANO A PLANTA
contador_QUI_PLA_P4<- 0
nodo_QUI_PLA_P4<- 0
numero_paciente_QUI_PLA_P4<- 0
Momento_terminacion_QUI_PLA_P4<- 0
for (r in 1:length(Momento_Llegada_PLA$P_PLA)) {
  if(Momento_Llegada_PLA$Nodo_anterior_PLA[r]=="QUI_P3"){
    if(Momento_Llegada_PLA$SPLIT[r]=="CONSULTA"){
      contador_QUI_PLA_P4<- contador_QUI_PLA_P4 + 1
      nodo_QUI_PLA_P4[contador_QUI_PLA_P4]<- "PLANTA"
      numero_paciente_QUI_PLA_P4[contador_QUI_PLA_P4]<- Momento_Llegad
a_PLA$P_PLA[r]
      Momento_terminacion_QUI_PLA_P4[contador_QUI_PLA_P4]<- Pacientes_
PLA2$Momento_Terminación_PLA2[which(Pacientes_PLA2$P_PLA2==numero_paci
ente_QUI_PLA_P4[contador_QUI_PLA_P4])]
    }
  }
}
Porcentaje_QUI_PLA_Paso4<- (contador_QUI_PLA_P4/length(which(Momento_L
legada_QUI$Obs_QUI_SPLIT=="CONSULTA")))*100

nodo_QUI_PLA_P4[nodo_QUI_PLA_P4==0]<- NA
nodo_QUI_PLA_P4<- na.omit(nodo_QUI_PLA_P4)
Porcentaje_QUI_PLA_Paso4[Porcentaje_QUI_PLA_Paso4==0]<- NA
Porcentaje_QUI_PLA_Paso4<- na.omit(Porcentaje_QUI_PLA_Paso4)
numero_paciente_QUI_PLA_P4[numero_paciente_QUI_PLA_P4==0]<- NA
numero_paciente_QUI_PLA_P4<- na.omit(numero_paciente_QUI_PLA_P4)
Momento_terminacion_QUI_PLA_P4[Momento_terminacion_QUI_PLA_P4==0]<- NA
Momento_terminacion_QUI_PLA_P4<- na.omit(Momento_terminacion_QUI_PLA_P
4)

QUI_PLA_P4<- data.frame(numero_paciente_QUI_PLA_P4=numero_paciente_QUI
_PLA_P4,
                        Momento_terminacion_QUI_PLA_P4=Momento_termina
cion_QUI_PLA_P4,
                        nodo_QUI_PLA_P4=nodo_QUI_PLA_P4)
QUI_PLA_P4<- QUI_PLA_P4[with(QUI_PLA_P4, order(QUI_PLA_P4$numero_pacie
nte_QUI_PLA_P4)), ]

#DE QUIROFANO A CRÍTICOS
contador_QUI_CRI_P4<- 0
nodo_QUI_CRI_P4<- 0
numero_paciente_QUI_CRI_P4<- 0
Momento_terminacion_QUI_CRI_P4<- 0
for (n in 1:length(Momento_Llegada_CRI1$P_CRI)) {
  if(Momento_Llegada_CRI1$Nodo_anterior_CRI[n]=="QUI_P3"){
    if(Momento_Llegada_CRI1$SPLIT_CRI1[n]=="CONSULTA"){
      contador_QUI_CRI_P4<- contador_QUI_CRI_P4 + 1
      nodo_QUI_CRI_P4[contador_QUI_CRI_P4]<- "CRÍTICOS"
```

```
      numero_paciente_QUI_CRI_P4[contador_QUI_CRI_P4]<- Momento_Llegad
a_CRI1$P_CRI[n]
      Momento_terminacion_QUI_CRI_P4[contador_QUI_CRI_P4]<- Pacientes_
CRI2$Momento_Terminación_CRI2[which(Pacientes_CRI2$P_CRI2==numero_paci
ente_QUI_CRI_P4[contador_QUI_CRI_P4])]
    }
  }
}
Porcentaje_QUI_CRI_Paso4<- (contador_QUI_CRI_P4/length(which(Momento_L
legada_QUI$Obs_QUI_SPLIT=="CONSULTA")))*100

nodo_QUI_CRI_P4[nodo_QUI_CRI_P4==0]<- NA
nodo_QUI_CRI_P4<- na.omit(nodo_QUI_CRI_P4)
Porcentaje_QUI_CRI_Paso4[Porcentaje_QUI_CRI_Paso4==0]<- NA
Porcentaje_QUI_CRI_Paso4<- na.omit(Porcentaje_QUI_CRI_Paso4)
numero_paciente_QUI_CRI_P4[numero_paciente_QUI_CRI_P4==0]<- NA
numero_paciente_QUI_CRI_P4<- na.omit(numero_paciente_QUI_CRI_P4)
Momento_terminacion_QUI_CRI_P4[Momento_terminacion_QUI_CRI_P4==0]<- NA
Momento_terminacion_QUI_CRI_P4<- na.omit(Momento_terminacion_QUI_CRI_P
4)

QUI_CRI_P4<- data.frame(numero_paciente_QUI_CRI_P4=numero_paciente_QUI
_CRI_P4,
                        Momento_terminacion_QUI_CRI_P4=Momento_termina
cion_QUI_CRI_P4,
                        nodo_QUI_CRI_P4=nodo_QUI_CRI_P4)
QUI_CRI_P4<- QUI_CRI_P4[with(QUI_CRI_P4, order(QUI_CRI_P4$numero_pacie
nte_QUI_CRI_P4)), ]

#DE QUIROFANO A UCI
contador_QUI_UCI_P4<- 0
nodo_QUI_UCI_P4<- 0
numero_paciente_QUI_UCI_P4<- 0
Momento_terminacion_QUI_UCI_P4<- 0
for (q in 1:length(Momento_Llegada_UCI1$P_UCI)) {
  if(Momento_Llegada_UCI1$Nodo_anterior_UCI[q]=="QUI_P3"){
    if(Momento_Llegada_UCI1$SPLIT_UCI1[q]=="CONSULTA"){
      contador_QUI_UCI_P4<- contador_QUI_UCI_P4 + 1
      nodo_QUI_UCI_P4[contador_QUI_UCI_P4]<- "UCI"
      numero_paciente_QUI_UCI_P4[contador_QUI_UCI_P4]<- Momento_Llegad
a_UCI1$P_UCI[q]
      Momento_terminacion_QUI_UCI_P4[contador_QUI_UCI_P4]<- Pacientes_
UCI2$Momento_Terminación_UCI2[which(Pacientes_UCI2$P_UCI2==numero_paci
ente_QUI_UCI_P4[contador_QUI_UCI_P4])]
    }
  }
}
Porcentaje_QUI_UCI_Paso4<- (contador_QUI_UCI_P4/length(which(Momento_L
legada_QUI$Obs_QUI_SPLIT=="CONSULTA")))*100

nodo_QUI_UCI_P4[nodo_QUI_UCI_P4==0]<- NA
nodo_QUI_UCI_P4<- na.omit(nodo_QUI_UCI_P4)
Porcentaje_QUI_UCI_Paso4[Porcentaje_QUI_UCI_Paso4==0]<- NA
Porcentaje_QUI_UCI_Paso4<- na.omit(Porcentaje_QUI_UCI_Paso4)
```

```
numero_paciente_QUI_UCI_P4[numero_paciente_QUI_UCI_P4==0]<- NA
numero_paciente_QUI_UCI_P4<- na.omit(numero_paciente_QUI_UCI_P4)
Momento_terminacion_QUI_UCI_P4[Momento_terminacion_QUI_UCI_P4==0]<- NA
Momento_terminacion_QUI_UCI_P4<- na.omit(Momento_terminacion_QUI_UCI_P
4)

QUI_UCI_P4<- data.frame(numero_paciente_QUI_UCI_P4=numero_paciente_QUI
_UCI_P4,
                        Momento_terminacion_QUI_UCI_P4=Momento_termina
cion_QUI_UCI_P4,
                        nodo_QUI_UCI_P4=nodo_QUI_UCI_P4)
QUI_UCI_P4<- QUI_UCI_P4[with(QUI_UCI_P4, order(QUI_UCI_P4$numero_pacie
nte_QUI_UCI_P4)), ]

Origen_Paso4<- c("UCI","CRÍTICOS","QUIROFANO","QUIROFANO","QUIROFANO")
Destino_Paso4<- c("PLANTA","PLANTA","PLANTA","CRÍTICOS","UCI")
Porcentaje_Paso4<- c(Porcentaje_UCI_PLA_Paso4,Porcentaje_CRI_PLA_Paso4
,
                     Porcentaje_QUI_PLA_Paso4,Porcentaje_QUI_CRI_Paso4
,
                     Porcentaje_QUI_UCI_Paso4)

Numero_Paciente_Cuarto_Paso<- c(UCI_PLA_P4$numero_paciente_UCI_PLA_P4,
                                CRI_PLA_P4$numero_paciente_CRI_PLA_P4,
                                QUI_PLA_P4$numero_paciente_QUI_PLA_P4,
                                QUI_CRI_P4$numero_paciente_QUI_CRI_P4,
                                QUI_UCI_P4$numero_paciente_QUI_UCI_P4)

Donde_Llega_Cuarto_Paso<- c(UCI_PLA_P4$nodo_UCI_PLA_P4,
                            CRI_PLA_P4$nodo_CRI_PLA_P4,
                            QUI_PLA_P4$nodo_QUI_PLA_P4,
                            QUI_CRI_P4$nodo_QUI_CRI_P4,
                            QUI_UCI_P4$nodo_QUI_UCI_P4)

Cuando_Termina_Cuarto_Paso<- c(UCI_PLA_P4$Momento_terminacion_UCI_PLA_
P4,
                               CRI_PLA_P4$Momento_terminacion_CRI_PLA_
P4,
                               QUI_PLA_P4$Momento_terminacion_QUI_PLA_
P4,
                               QUI_CRI_P4$Momento_terminacion_QUI_CRI_
P4,
                               QUI_UCI_P4$Momento_terminacion_QUI_UCI_
P4)

Cuarto_Paso<- data.frame(Numero_Paciente_Cuarto_Paso=Numero_Paciente_C
uarto_Paso,Donde_Llega_Cuarto_Paso=Donde_Llega_Cuarto_Paso,Cuando_Term
ina_Cuarto_Paso=Cuando_Termina_Cuarto_Paso)

Cuarto_Paso<- Cuarto_Paso[with(Cuarto_Paso, order(Cuarto_Paso$Numero_P
aciente_Cuarto_Paso)), ]

ki<- nrow(Cuarto_Paso)+1
```

```
Cuarto_Paso[ki:nrow(datos),1]<- 0

contar_1<- 0; contar_2<- 0; contar1_1<-0; contar1_2<- 0; personis_1<-
0
personis_2<- 0; personis_3<-0; personis_4<-0; diferencia_1<-0; diferen
cia_2<- 0
print<- 0; rango_inicio<- 0; rango_fin<- 0; Donde1<-0; Donde2<- 0; Don
de3<-0
Donde4<-0; Cuando1<-0; Cuando2<-0; Cuando3<- 0; Cuando4<-0; longitud5<
- 0
longitud6<-0; longitud7<-0; longitud8<- 0; longitud9<- 0
for (i in 2:sum(nclientes)) {
  if(Cuarto_Paso$Numero_Paciente_Cuarto_Paso[i]-Cuarto_Paso$Numero_Pac
iente_Cuarto_Paso[i-1]!=1){
    contar_1<-contar_1 +1
    diferencia_1[contar_1]<- Cuarto_Paso$Numero_Paciente_Cuarto_Paso[i
] - Cuarto_Paso$Numero_Paciente_Cuarto_Paso[i-1] - 1
    if(diferencia_1[contar_1]==1){
      contar1_1<- contar1_1+1
      personis_1[contar1_1]<- Cuarto_Paso$Numero_Paciente_Cuarto_Paso[
i-1]+1
      Donde1[contar1_1]<- NA
      Cuando1[contar1_1]<- NA
    }
    personis_1[personis_1==0]<- NA
    personis_1<- na.omit(personis_1)
  }
}
for (i in 2:sum(nclientes)) {
  if(Cuarto_Paso$Numero_Paciente_Cuarto_Paso[i]!=0){
    if(Cuarto_Paso$Numero_Paciente_Cuarto_Paso[i]-Cuarto_Paso$Numero_P
aciente_Cuarto_Paso[i-1]!=1){
      contar_2<-contar_2 +1
      diferencia_2[contar_2]<- Cuarto_Paso$Numero_Paciente_Cuarto_Paso
[i] - Cuarto_Paso$Numero_Paciente_Cuarto_Paso[i-1] - 1
      if(diferencia_2[contar_2]!=1){
        contar1_2<- contar1_2+1
        añadir4<- contar1_2  + longitud5
        longitud5<- añadir4 + diferencia_2[contar_2] -1
        rango_inicio<- Cuarto_Paso$Numero_Paciente_Cuarto_Paso[i-1] +
1
        rango_fin<- Cuarto_Paso$Numero_Paciente_Cuarto_Paso[i] - 1
        personis_2[añadir4:longitud5]<- rango_inicio:rango_fin
        personis_2[personis_2==0]<- NA
        Donde2[añadir4:longitud5]<- 1
        Cuando2[añadir4:longitud5]<- 1
        Donde2[Donde2==0]<- NA
        Cuando2[Cuando2==0]<- NA
      }
    }
    if(Cuarto_Paso$Numero_Paciente_Cuarto_Paso[i]-Cuarto_Paso$Numero_P
aciente_Cuarto_Paso[i-1]==0){
      print<- Cuarto_Paso$Numero_Paciente_Cuarto_Paso[i]
    }
```

```
  }
}
personis_2<- na.omit(personis_2)
Donde2<- na.omit(Donde2)
Cuando2<- na.omit(Cuando2)
Donde2[Donde2==1]<- NA
Cuando2[Cuando2==1]<- NA

Cuarto_Paso<- data.frame(Numero_Paciente_Cuarto_Paso=Numero_Paciente_C
uarto_Paso,Donde_Llega_Cuarto_Paso=Donde_Llega_Cuarto_Paso,Cuando_Term
ina_Cuarto_Paso=Cuando_Termina_Cuarto_Paso)

Cuarto_Paso<- Cuarto_Paso[with(Cuarto_Paso, order(Cuarto_Paso$Numero_P
aciente_Cuarto_Paso)), ]

if(min(Cuarto_Paso$Numero_Paciente_Cuarto_Paso)>1){
  longitud6<- min(Cuarto_Paso$Numero_Paciente_Cuarto_Paso)-1
  personis_3[1:longitud6]<- 1:longitud6
  Donde3[1:longitud6]<- NA
  Cuando3[1:longitud6]<- NA
}
if(max(Cuarto_Paso$Numero_Paciente_Cuarto_Paso)<sum(nclientes)){
  longitud7<- max(Cuarto_Paso$Numero_Paciente_Cuarto_Paso)+1
  longitud9<- max(Cuarto_Paso$Numero_Paciente_Cuarto_Paso)
  longitud8<- sum(nclientes) - longitud9
  personis_4[1:longitud8]<- longitud7:sum(nclientes)
  Donde4[1:longitud8]<- NA
  Cuando4[1:longitud8]<- NA
}
if(length(Donde4)==1 & personis_4==0){
   Donde4<- NA
   Donde4<- na.omit(Donde4)
}
if(length(Cuando4)==1 & personis_4==0){
   Cuando4<- NA
   Cuando4<- na.omit(Cuando4)
}
if(Escenario==3){
if(length(Donde1)==1 & personis_1==0){
   Donde1<- NA
   Donde1<- na.omit(Donde1)
}
if(length(Cuando1)==1 & personis_1==0){
   Cuando1<- NA
   Cuando1<- na.omit(Cuando1)
}
}
if(Escenario==1){
if(length(Donde1)==1){
   Donde1<- NA
   Donde1<- na.omit(Donde1)
}
if(length(Cuando1)==1){
   Cuando1<- NA
```

```
   Cuando1<- na.omit(Cuando1)
}
}

if(Escenario==2){
if(length(Donde1)==1){
   Donde1<- NA
   Donde1<- na.omit(Donde1)
}
if(length(Cuando1)==1){
   Cuando1<- NA
   Cuando1<- na.omit(Cuando1)
}
}
if(length(Donde3)==1 & personis_3==0){
   Donde3<- NA
   Donde3<- na.omit(Donde3)
}
if(length(Cuando3)==1 & personis_3==0){
   Cuando3<- NA
   Cuando3<- na.omit(Cuando3)
}
if(length(personis_3)==1 & personis_3==0){
  personis_3<- NA
  personis_3<- na.omit(personis_3)
}
if(length(personis_4)==1 & personis_4==0){
  personis_4<- NA
  personis_4<- na.omit(personis_4)
}
unir_personis<- c(personis_1,personis_2,personis_3,personis_4)
unir_Donde<- c(Donde1,Donde2,Donde3,Donde4)
unir_Cuando<- c(Cuando1,Cuando2,Cuando3,Cuando4)

Cuarto_Paso<- data.frame(Numero_Paciente_Cuarto_Paso=c(Numero_Paciente
_Cuarto_Paso,unir_personis),Donde_Llega_Cuarto_Paso=c(Donde_Llega_Cuar
to_Paso,unir_Donde),Cuando_Termina_Cuarto_Paso=c(Cuando_Termina_Cuarto
_Paso,unir_Cuando))

Cuarto_Paso<- Cuarto_Paso[with(Cuarto_Paso, order(Cuarto_Paso$Numero_P
aciente_Cuarto_Paso)), ]

datos<- data.frame(Personas=Pacientes,Momento_Llegada_al_Sistema=Momen
to_Llegada_al_Sistema,Cuando_Termina_el_Triaje=Cuando_Termina_el_Triaj
e,Donde_Llega_Despues_Triaje=Donde_Llega_Despues_Triaje,Cuando_Llega=C
uando_Llega,Cuando_Termina=Cuando_Termina,Donde_Llega_Segundo_Paso=Seg
undo_Paso$Donde_Llega_Segundo_Paso,Cuando_Termina_Segundo_Paso=Segundo
_Paso$Cuando_Termina_Segundo_Paso,Donde_Llega_Tercer_Paso=Tercer_Paso$
Donde_Llega_Tercer_Paso,Cuando_Termina_Tercer_Paso=Tercer_Paso$Cuando_
Termina_Tercer_Paso,Donde_Llega_Cuarto_Paso=Cuarto_Paso$Donde_Llega_Cu
arto_Paso,Cuando_Termina_Cuarto_Paso=Cuarto_Paso$Cuando_Termina_Cuarto
_Paso)
if(is.logical(Porcentaje_UCI_PLA_Paso4)==FALSE){
  Porcentaje_Paso4[1]<- 0
```

```
}
if(is.logical(Porcentaje_CRI_PLA_Paso4)==FALSE){
  Porcentaje_Paso4[2]<- 0
}
if(is.logical(Porcentaje_QUI_PLA_Paso4)==FALSE){
  Porcentaje_Paso4[3]<- 0
}
if(is.logical(Porcentaje_QUI_CRI_Paso4)==FALSE){
  Porcentaje_Paso4[5]<- 0
}
if(is.logical(Porcentaje_QUI_UCI_Paso4)==FALSE){
  Porcentaje_Paso4[5]<- 0
}
}
############################## QUINTO PASO ###############
{
#UCI A PLANTA
contador_UCI_PLA_P5<- 0
nodo_UCI_PLA_P5<- 0
numero_paciente_UCI_PLA_P5<- 0
Momento_terminacion_UCI_PLA_P5<- 0
for (r in 1:length(Momento_Llegada_PLA$P_PLA)) {
  if(Momento_Llegada_PLA$Nodo_anterior_PLA[r]=="UCI"){
    if(Momento_Llegada_PLA$ANTERIOR_P4[r]=="QUI_P3"){
      if(Momento_Llegada_PLA$SPLIT[r]=="CONSULTA"){
        contador_UCI_PLA_P5<- contador_UCI_PLA_P5 + 1
        nodo_UCI_PLA_P5[contador_UCI_PLA_P5]<- "PLANTA"
        numero_paciente_UCI_PLA_P5[contador_UCI_PLA_P5]<- Momento_Lleg
ada_PLA$P_PLA[r]
        Momento_terminacion_UCI_PLA_P5[contador_UCI_PLA_P5]<- Paciente
s_PLA2$Momento_Terminación_PLA2[which(Pacientes_PLA2$P_PLA2==numero_pa
ciente_UCI_PLA_P5[contador_UCI_PLA_P5])]
      }
    }
  }
}
Porcentaje_UCI_PLA_Paso5<- 100

nodo_UCI_PLA_P5[nodo_UCI_PLA_P5==0]<- NA
nodo_UCI_PLA_P5<- na.omit(nodo_UCI_PLA_P5)
Porcentaje_UCI_PLA_Paso5[Porcentaje_UCI_PLA_Paso5==0]<- NA
Porcentaje_UCI_PLA_Paso5<- na.omit(Porcentaje_UCI_PLA_Paso5)
numero_paciente_UCI_PLA_P5[numero_paciente_UCI_PLA_P5==0]<- NA
numero_paciente_UCI_PLA_P5<- na.omit(numero_paciente_UCI_PLA_P5)
Momento_terminacion_UCI_PLA_P5[Momento_terminacion_UCI_PLA_P5==0]<- NA
Momento_terminacion_UCI_PLA_P5<- na.omit(Momento_terminacion_UCI_PLA_P
5)

UCI_PLA_P5<- data.frame(numero_paciente_UCI_PLA_P5=c(numero_paciente_U
CI_PLA_P5),Momento_terminacion_UCI_PLA_P5=c(Momento_terminacion_UCI_PL
A_P5),nodo_UCI_PLA_P5=c(nodo_UCI_PLA_P5))
UCI_PLA_P5<- UCI_PLA_P5[with(UCI_PLA_P5, order(UCI_PLA_P5$numero_pacie
nte_UCI_PLA_P5)), ]
```

```
#DE CRÍTICOS A PLANTA
contador_CRI_PLA_P5<- 0
nodo_CRI_PLA_P5<- 0
numero_paciente_CRI_PLA_P5<- 0
Momento_terminacion_CRI_PLA_P5<- 0
for (r in 1:length(Momento_Llegada_PLA$P_PLA)) {
  if(Momento_Llegada_PLA$Nodo_anterior_PLA[r]=="CRI"){
    if(Momento_Llegada_PLA$ANTERIOR_P4[r]=="QUI_P3"){
      if(Momento_Llegada_PLA$SPLIT[r]=="CONSULTA"){
        contador_CRI_PLA_P5<- contador_CRI_PLA_P5 + 1
        nodo_CRI_PLA_P5[contador_CRI_PLA_P5]<- "PLANTA"
        numero_paciente_CRI_PLA_P5[contador_CRI_PLA_P5]<- Momento_Lleg
ada_PLA$P_PLA[r]
        Momento_terminacion_CRI_PLA_P5[contador_CRI_PLA_P5]<- Paciente
s_PLA2$Momento_Terminación_PLA2[which(Pacientes_PLA2$P_PLA2==numero_pa
ciente_CRI_PLA_P5[contador_CRI_PLA_P5])]
      }
    }
  }
}
Porcentaje_CRI_PLA_Paso5<- 100

nodo_CRI_PLA_P5[nodo_CRI_PLA_P5==0]<- NA
nodo_CRI_PLA_P5<- na.omit(nodo_CRI_PLA_P5)
Porcentaje_CRI_PLA_Paso5[Porcentaje_CRI_PLA_Paso5==0]<- NA
Porcentaje_CRI_PLA_Paso5<- na.omit(Porcentaje_CRI_PLA_Paso5)
numero_paciente_CRI_PLA_P5[numero_paciente_CRI_PLA_P5==0]<- NA
numero_paciente_CRI_PLA_P5<- na.omit(numero_paciente_CRI_PLA_P5)
Momento_terminacion_CRI_PLA_P5[Momento_terminacion_CRI_PLA_P5==0]<- NA
Momento_terminacion_CRI_PLA_P5<- na.omit(Momento_terminacion_CRI_PLA_P
5)

CRI_PLA_P5<- data.frame(numero_paciente_CRI_PLA_P5=c(numero_paciente_C
RI_PLA_P5),Momento_terminacion_CRI_PLA_P5=c(Momento_terminacion_CRI_PL
A_P5),nodo_CRI_PLA_P5=c(nodo_CRI_PLA_P5))
CRI_PLA_P5<- CRI_PLA_P5[with(CRI_PLA_P5, order(CRI_PLA_P5$numero_pacie
nte_CRI_PLA_P5)), ]

Origen_Paso5<- c("UCI","CRÍTICOS")
Destino_Paso5<- c("PLANTA","PLANTA")
Porcentaje_Paso5<- c(Porcentaje_UCI_PLA_Paso5,Porcentaje_CRI_PLA_Paso5
)

Numero_Paciente_Quinto_Paso<- c(UCI_PLA_P5$numero_paciente_UCI_PLA_P5,
                                CRI_PLA_P5$numero_paciente_CRI_PLA_P5)

Donde_Llega_Quinto_Paso<- c(UCI_PLA_P5$nodo_UCI_PLA_P5,
                            CRI_PLA_P5$nodo_CRI_PLA_P5)

Cuando_Termina_Quinto_Paso<- c(UCI_PLA_P5$Momento_terminacion_UCI_PLA_
P5,
                               CRI_PLA_P5$Momento_terminacion_CRI_PLA_
P5)
```

```
Quinto_Paso<- data.frame(Numero_Paciente_Quinto_Paso=Numero_Paciente_Q
uinto_Paso,Donde_Llega_Quinto_Paso=Donde_Llega_Quinto_Paso,Cuando_Term
ina_Quinto_Paso=Cuando_Termina_Quinto_Paso)

Quinto_Paso<- Quinto_Paso[with(Quinto_Paso, order(Quinto_Paso$Numero_P
aciente_Quinto_Paso)), ]
{
if(is.null(nrow(Quinto_Paso$Numero_Paciente_Quinto_Paso))==TRUE){
  Quinto_Paso[1:nrow(datos),1]<-0
}
else{
  ki<- nrow(Quinto_Paso)+1
  Quinto_Paso[ki:nrow(datos),1]<- 0
}
}
contar_1<- 0; contar_2<- 0; contar1_1<-0; contar1_2<- 0; personis_1<-
0
personis_2<- 0; personis_3<-0; personis_4<-0; diferencia_1<-0; diferen
cia_2<- 0
print<- 0; rango_inicio<- 0; rango_fin<- 0; Donde1<-0; Donde2<- 0; Don
de3<-0
Donde4<-0; Cuando1<-0; Cuando2<-0; Cuando3<- 0; Cuando4<-0; longitud5<
- 0
longitud6<-0; longitud7<-0; longitud8<- 0; longitud9<- 0
for (i in 2:sum(nclientes)) {
  if(Quinto_Paso$Numero_Paciente_Quinto_Paso[i]-Quinto_Paso$Numero_Pac
iente_Quinto_Paso[i-1]!=1){
    contar_1<-contar_1 +1
    diferencia_1[contar_1]<- Quinto_Paso$Numero_Paciente_Quinto_Paso[i
] - Quinto_Paso$Numero_Paciente_Quinto_Paso[i-1] - 1
    if(diferencia_1[contar_1]==1){
      contar1_1<- contar1_1+1
      personis_1[contar1_1]<- Quinto_Paso$Numero_Paciente_Quinto_Paso[
i-1]+1
      Donde1[contar1_1]<- NA
      Cuando1[contar1_1]<- NA
    }
    personis_1[personis_1==0]<- NA
    personis_1<- na.omit(personis_1)
  }
}
if(length(Donde1)==1){
  Donde1<- NA
  Donde1<- na.omit(Donde1)
}
if(length(Cuando1)==1){
  Cuando1<- NA
  Cuando1<- na.omit(Cuando1)
}
for (i in 2:sum(nclientes)) {
  if(Quinto_Paso$Numero_Paciente_Quinto_Paso[i]!=0){
    if(Quinto_Paso$Numero_Paciente_Quinto_Paso[i]-Quinto_Paso$Numero_P
aciente_Quinto_Paso[i-1]!=1){
      contar_2<-contar_2 +1
```

```
      diferencia_2[contar_2]<- Quinto_Paso$Numero_Paciente_Quinto_Paso
[i] - Quinto_Paso$Numero_Paciente_Quinto_Paso[i-1] - 1
      if(diferencia_2[contar_2]!=1){
        contar1_2<- contar1_2+1
        añadir4<- contar1_2  + longitud5
        longitud5<- añadir4 + diferencia_2[contar_2] -1
        rango_inicio<- Quinto_Paso$Numero_Paciente_Quinto_Paso[i-1] +
1
        rango_fin<- Quinto_Paso$Numero_Paciente_Quinto_Paso[i] - 1
        personis_2[añadir4:longitud5]<- rango_inicio:rango_fin
        personis_2[personis_2==0]<- NA
        Donde2[añadir4:longitud5]<- 1
        Cuando2[añadir4:longitud5]<- 1
        Donde2[Donde2==0]<- NA
        Cuando2[Cuando2==0]<- NA
      }
    }
    if(Quinto_Paso$Numero_Paciente_Quinto_Paso[i]-Quinto_Paso$Numero_P
aciente_Quinto_Paso[i-1]==0){
      print<- Quinto_Paso$Numero_Paciente_Quinto_Paso[i]
    }
  }
}
personis_2<- na.omit(personis_2)
Donde2<- na.omit(Donde2)
Cuando2<- na.omit(Cuando2)
Donde2[Donde2==1]<- NA
Cuando2[Cuando2==1]<- NA

Quinto_Paso<- data.frame(Numero_Paciente_Quinto_Paso=Numero_Paciente_Q
uinto_Paso,Donde_Llega_Quinto_Paso=Donde_Llega_Quinto_Paso,Cuando_Term
ina_Quinto_Paso=Cuando_Termina_Quinto_Paso)

Quinto_Paso<- Quinto_Paso[with(Quinto_Paso, order(Quinto_Paso$Numero_P
aciente_Quinto_Paso)), ]
Quinto_Paso1<- matrix(ncol = 3,nrow = nrow(datos))
if(is.null(nrow(Quinto_Paso$Numero_Paciente_Quinto_Paso))==TRUE){
  Quinto_Paso1[1:nrow(datos),1]<-1:nrow(datos)
  Quinto_Paso1[1:nrow(datos),2]<- NA
  Quinto_Paso1[1:nrow(datos),3]<- NA
  datos<- data.frame(Personas=Pacientes,Momento_Llegada_al_Sistema=Mom
ento_Llegada_al_Sistema,Cuando_Termina_el_Triaje=Cuando_Termina_el_Tri
aje,Donde_Llega_Despues_Triaje=Donde_Llega_Despues_Triaje,Cuando_Llega
=Cuando_Llega,Cuando_Termina=Cuando_Termina,Donde_Llega_Segundo_Paso=S
egundo_Paso$Donde_Llega_Segundo_Paso,Cuando_Termina_Segundo_Paso=Segun
do_Paso$Cuando_Termina_Segundo_Paso,Donde_Llega_Tercer_Paso=Tercer_Pas
o$Donde_Llega_Tercer_Paso,Cuando_Termina_Tercer_Paso=Tercer_Paso$Cuand
o_Termina_Tercer_Paso,Donde_Llega_Cuarto_Paso=Cuarto_Paso$Donde_Llega_
Cuarto_Paso,Cuando_Termina_Cuarto_Paso=Cuarto_Paso$Cuando_Termina_Cuar
to_Paso,Donde_Llega_Quinto_Paso=Quinto_Paso1[,2],Cuando_Termina_Quinto
_Paso=Quinto_Paso1[,3])
}
else{
if(length(Quinto_Paso$Numero_Paciente_Quinto_Paso)!=0){
```

```
if(min(Quinto_Paso$Numero_Paciente_Quinto_Paso)>1){
  longitud6<- min(Quinto_Paso$Numero_Paciente_Quinto_Paso)-1
  personis_3[1:longitud6]<- 1:longitud6
  Donde3[1:longitud6]<- NA
  Cuando3[1:longitud6]<- NA
}
else{
  personis_3<- NA
  Donde3<- NA
  Cuando3<- NA
}
}
if(length(Quinto_Paso$Numero_Paciente_Quinto_Paso)!=0){
if(max(Quinto_Paso$Numero_Paciente_Quinto_Paso)<sum(nclientes)){
  longitud7<- max(Quinto_Paso$Numero_Paciente_Quinto_Paso)+1
  longitud9<- max(Quinto_Paso$Numero_Paciente_Quinto_Paso)
  longitud8<- sum(nclientes) - longitud9
  personis_4[1:longitud8]<- longitud7:sum(nclientes)
  Donde4[1:longitud8]<- NA
  Cuando4[1:longitud8]<- NA
}
else{
  personis_4<- NA
  Donde4<- NA
  Cuando4<- NA
}
}
if(length(Donde4)==1){
  Donde4<- NA
  Donde4<- na.omit(Donde4)
}
if(length(Cuando4)==1){
  Cuando4<- NA
  Cuando4<- na.omit(Cuando4)
}
if(length(Donde2)==1){
  Donde2<- NA
  Donde2<- na.omit(Donde2)
}
if(length(Cuando2)==1){
  Cuando2<- NA
  Cuando2<- na.omit(Cuando2)
}
if(length(personis_2)==1){
  personis_2<- NA
  personis_2<- na.omit(personis_2)
}
if(length(Donde3)==1){
  Donde3<- NA
  Donde3<- na.omit(Donde3)
}
if(length(Cuando3)==1){
  Cuando3<- NA
  Cuando3<- na.omit(Cuando3)
```

```
}
if(length(personis_3)==1){
  personis_3<- NA
  personis_3<- na.omit(personis_3)
}
if(length(personis_4)==1){
  personis_4<- NA
  personis_4<- na.omit(personis_4)
}
unir_personis<- c(personis_1,personis_2,personis_3,personis_4)
unir_Donde<- c(Donde1,Donde2,Donde3,Donde4)
unir_Cuando<- c(Cuando1,Cuando2,Cuando3,Cuando4)

Quinto_Paso<- data.frame(Numero_Paciente_Quinto_Paso=c(Numero_Paciente
_Quinto_Paso,unir_personis),Donde_Llega_Quinto_Paso=c(Donde_Llega_Quin
to_Paso,unir_Donde),Cuando_Termina_Quinto_Paso=c(Cuando_Termina_Quinto
_Paso,unir_Cuando))

Quinto_Paso<- Quinto_Paso[with(Quinto_Paso, order(Quinto_Paso$Numero_P
aciente_Quinto_Paso)), ]

datos<- data.frame(Personas=Pacientes,Momento_Llegada_al_Sistema=Momen
to_Llegada_al_Sistema,Cuando_Termina_el_Triaje=Cuando_Termina_el_Triaj
e,Donde_Llega_Despues_Triaje=Donde_Llega_Despues_Triaje,Cuando_Llega=C
uando_Llega,Cuando_Termina=Cuando_Termina,Donde_Llega_Segundo_Paso=Seg
undo_Paso$Donde_Llega_Segundo_Paso,Cuando_Termina_Segundo_Paso=Segundo
_Paso$Cuando_Termina_Segundo_Paso,Donde_Llega_Tercer_Paso=Tercer_Paso$
Donde_Llega_Tercer_Paso,Cuando_Termina_Tercer_Paso=Tercer_Paso$Cuando_
Termina_Tercer_Paso,Donde_Llega_Cuarto_Paso=Cuarto_Paso$Donde_Llega_Cu
arto_Paso,Cuando_Termina_Cuarto_Paso=Cuarto_Paso$Cuando_Termina_Cuarto
_Paso,Donde_Llega_Quinto_Paso=Quinto_Paso$Donde_Llega_Quinto_Paso,Cuan
do_Termina_Quinto_Paso=Quinto_Paso$Cuando_Termina_Quinto_Paso)
}
if(is.logical(Porcentaje_UCI_PLA_Paso5)==FALSE){
  Porcentaje_Paso5[1]<- 0
}
if(is.logical(Porcentaje_CRI_PLA_Paso5)==FALSE){
  Porcentaje_Paso5[2]<- 0
}
}

################## REALIZACIÓN DE LAS DIFERENTES TABLAS ##########

############################ Tipo de Escenario ###############
Tipo_Escenario %>%
  kbl(caption =  "<center><strong>Gravedad de Saturación en el sistema
</strong></center>") %>%
  kable_classic(full_width = F, html_font = "Cambria") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed",
"responsive"))

###### Medidas de eficacia del sistema de urgencias hospitalarias ####
Parametros %>%
  kbl(caption =  "<center><strong>Medidas de eficacia del sistema de u
rgencias hospitalarias </strong></center>") %>%
```

```
  kable_classic(full_width = F, html_font = "Cambria") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed",
"responsive")) %>%
  footnote(general_title = "Nota 1: Tiempo en minutos" ,general="") #%
>%
  #row_spec(row=1:11, background = "#FFFFF0")

########### Información general del sistema por cada pacientes #####
datatable(datos,
          caption = 'Información de cada uno de los pacientes en el Si
stema de urgencias  hospitalarias')

################## REALIZACIÓN DE LOS DIFERENTES GRÁFICOS ##########

######################### SATURACIÓN DE LA PLANTA #########
#Elegimos los datos que queremos plotear
Inicio_Grafico1<- Pacientes_PLA2 %>% count(Pacientes_PLA2$Llegada_Hora
_Paciente)
Fin_Grafico1<- Pacientes_PLA2 %>% count(Pacientes_PLA2$Salida_Hora_Pac
iente)
Medio1_Grafico1<- (Pacientes_PLA2$Llegada_Hora_Paciente)
Medio2_Grafico1<- as.factor(Pacientes_PLA2$Disponible_PLA2)
Medio3_Grafico1<- Pacientes_PLA2$Disponible_PLA2
Medio4_Grafico1<- data.frame(Medio1_Grafico1=Medio1_Grafico1,Medio1_Gr
afico1=Medio1_Grafico1,Medio3_Grafico1=Medio3_Grafico1)

Servidores_Grafico1<- Medio4_Grafico1 %>%  group_by(Medio1_Grafico1) %
>% summarise(Media = mean(Medio3_Grafico1))
N_servidores<-0
for (j in 1:nrow(Inicio_Grafico1)) {
  if(j==1){
  N_servidores[j]<- S_Disponible_PLA2-Inicio_Grafico1$n[j]
  }
  else{
    N_servidores[j]<- N_servidores[j-1]-Inicio_Grafico1$n[j]
  }
}
Inicio_Grafico1$servidores<- N_servidores
matriz<- matrix(0,ncol = 4, nrow = max(Inicio_Grafico1$`Pacientes_PLA2
$Llegada_Hora_Paciente`))
matriz[,1]<- 1:max(Inicio_Grafico1$`Pacientes_PLA2$Llegada_Hora_Pacien
te`)
matriz[,3]<- rep(40,max(Inicio_Grafico1$`Pacientes_PLA2$Llegada_Hora_P
aciente`))

for (j in 1:nrow(Inicio_Grafico1)) {
  matriz[which(matriz[,1]==Inicio_Grafico1[j,1]),2]<- Inicio_Grafico1[
j,2]
matriz[which(matriz[,1]==Inicio_Grafico1[j,1]):max(Inicio_Grafico1$`Pa
cientes_PLA2$Llegada_Hora_Paciente`),3]<- Inicio_Grafico1[j,3]
  matriz[which(matriz[,1]==Fin_Grafico1[j,1]),4]<- Fin_Grafico1[j,2]
}
formar<- 0
disp<- 0
gente<-0
```

```
minimo<-  min(Pacientes_PLA2$Llegada_Hora_Paciente)-1
if(min(Pacientes_PLA2$Llegada_Hora_Paciente)>=1){
    formar[1:minimo]<- 1:minimo
    disp[1:minimo]<- S_Disponible_PLA2
    gente[1:minimo]<- 1
}
Inicio_Grafico1_1<- data.frame(n=c(disp,Servidores_Grafico1$Media))
Inicio_Grafico11_1<- data.frame(n=c(gente,Inicio_Grafico1$n))
Inicio_Grafico11_2<- data.frame(n=c(gente,Fin_Grafico1$n))
#Creamos un data frame
Grafico1<-data.frame(Pacientes_entran=Inicio_Grafico11_1$n[1:24],
                     Pacientes_salen=Inicio_Grafico11_2$n[1:24],
                     Media_Servidores_Disponibles=Inicio_Grafico1_1$n[
1:24])
ggplot()+
  geom_line(aes(x=1:24, y=matriz[1:24,2], color="Pacientes que entran
a Planta"))+
  geom_line(aes(x=1:24, y=matriz[1:24,4], color="Pacientes que salen d
e Planta"))+
  geom_line(aes(x=1:24, y=matriz[1:24,3], color="Media de los servidor
es Disponibles"))+
  geom_point(aes(x=1:24, y=matriz[1:24,2]),color="black", size=.5)+
  geom_point(aes(x=1:24, y=matriz[1:24,3]),color="black", size=.5)+
  geom_point(aes(x=1:24, y=matriz[1:24,4]),color="black", size=.5)+
  ylab("Número de Pacientes que interactuan con Planta")+
  xlab("Horas del día")+
  guides(color = guide_legend(title = "Leyenda"))+
  scale_x_continuous(breaks = c(1:24))+
  theme(
    legend.position = c(0.4, 0.6),
    legend.justification = c(1, 1),
    legend.background = element_rect(fill = "white", colour = "black")
,
    plot.title = element_text(face = "bold", colour = "brown",size = r
el(2.2),
                              hjust = 0.5),
    panel.background = element_rect(fill = "white")) + #FFFFF0 #white
y ##F0FFFF
  ggtitle("Saturación de la Planta durante 24 horas")

########### TIEMPOS DE LOS PRIMEROS 10 PACIENTES EN UCI EN HORAS #####
###
Número_de_Pacientes<- 10
#Realizamos el gráfico
Grafico2<- plot_ly()
for(i in 1:Número_de_Pacientes){
  Grafico2 <- add_trace(Grafico2,
                x = c((Pacientes_UCI2$Llegada_UCI2/60)[i], (Pacientes
_UCI2$Llegada_UCI2/60)[i] + (Pacientes_UCI2$Tiempo_Espera_UCI2/60)[i]
+(Pacientes_UCI2$Tiempo_servicio_UCI2/60)[i]),
                y = c(i, i),
                mode = "lines",
                line = list(color = heat.colors(nrow(Pacientes_UCI2))
, width = 20),
```

```
                showlegend = F,
                hoverinfo = "text",
                text = paste("Paciente: ", Pacientes_UCI2$P_UCI2[i],
"<br>",
                             "Tiempo de Espera: ", (Pacientes_UCI2$Ti
empo_Espera_UCI2/60)[i], " horas <br>",
                             "Tiempo de servicio: ", (Pacientes_UCI2$
Tiempo_servicio_UCI2/60)[i]," horas <br>",
                             "Momento de Terminación: ", (Pacientes_U
CI2$Momento_Terminación_UCI2/60)[i]," horas <br>"),
                evaluate = T
  )
}
Grafico2<- layout(Grafico2,
             xaxis = list(showgrid = F, tickfont = list(color = "#bla
ck")),

             yaxis = list(showgrid = T, tickfont = list(color = "#bla
ck"),
                          tickmode = "array", tickvals = 1:Número_de_
Pacientes, ticktext = unique(Pacientes_UCI2$P_UCI2),
                          domain = c(0, 0.9))
              )
anotaciones<- list(xref = "paper",
          yref = "paper",
          x = 0.1,
          y = 1,
          xanchor = "left",
          text = paste0("Tiempos de los primeros 10 pacientes en UCI e
n horas"),
          font = list(color = '#264E86', size = 20, family = "Times Ne
w Roman"),
          ax = 0,
          ay = 0,
          align = "left",
          showarrow = FALSE)
Grafico2<- Grafico2 %>% layout(annotations = anotaciones)
Grafico2

########## Donde se dirigen las personas que llegan a la Consulta ####
###
Lugares<- c("UCI","Críticos","Quirofano","Alta","Observación")
Numero_de_Pacientes<- c(length(Momento_Llegada_UCI1$Nodo_anterior_UCI=
="Con"),
            length(Momento_Llegada_CRI1$Nodo_anterior_CRI=="Con"),
            length(Momento_Llegada_QUI$Nodo_anterior_QUI=="Con"),
            length(Momento_Llegada_ALT$Nodo_anterior_ALT=="Con"),
            length(Momento_Llegada_ALT$Nodo_anterior_ALT=="Con"))
color <- c("#A52A2A","#458B74","#FFE4C4","#F0E68C","#CD8162")
Grafico3<- data.frame(Lugares,Numero_de_Pacientes, color)
ggplot(Grafico3, aes(x = 1:5, y = 1:5, label = Lugares)) +
  geom_point(aes(size = Numero_de_Pacientes, colour = Lugares)) +
  geom_text(hjust = 1, size = 2) +
  theme_minimal()+
```

```
  ylab("Lugares")+
  xlab("Lugares")+
  ggtitle("Donde se dirigen las personas que llegan a la Consulta")

########################## GRAFO DEL SISTEMA ############
datos_grafo<- data.frame(Origen=c(Origen_Paso1,Origen_Paso2,Origen_Pas
o3,Origen_Paso4,Origen_Paso5),Destino=c(Destino_Paso1,Destino_Paso2,De
stino_Paso3,Destino_Paso4,Destino_Paso5),Porcentaje=round(c(Porcentaje
_Paso1,Porcentaje_Paso2,Porcentaje_Paso3,Porcentaje_Paso4,Porcentaje_P
aso5)))
gd=graph.data.frame(d = datos_grafo,directed = T)
E(gd)$Porcentaje <-datos_grafo$Porcentaje
new_gd=delete.vertices(graph = gd,v = V(graph = gd)[degree(graph = gd,
mode = "all")<=1])
colrs <- c("#FFE1FF","#7FFFD4","#CD8500","#A52A2A","#7FFF00",
           "#FF7256","#FFB90F","#9932CC","#B3EE3A","#FF1493","#BFEFFF"
)
new_gd$color<-colrs[as.factor(datos_grafo$Origen)]
plot.igraph(new_gd,
            vertex.shape="none",
            edge.arrow.size=0.6,
            edge.arrow.width=0.6,
            edge.curved = T,
            vertex.label.cex=1,
            main="Grafo con los porcentajes de translados de los pacie
ntes",
            vertex.label.color="black",
            vertex.color= "white",
            edge.color=new_gd$color,
            edge.label = paste(E(new_gd)$Porcentaje, sep = ""),
            edge.label.cex=1
)
legend(x=-2, y=1.2,levels(as.factor(datos_grafo$Origen)), pch=21,
       col="#777777", pt.bg=colrs, pt.cex=2, cex=.8, bty="n", ncol=1)

#Creamos un fichero excel con los datos generales de la simulación, es
decir,los pasos que da cada paciente dentro del sistema de urgencias h
ospitalarias
WriteXLS(datos, "datos_simulación.xlsx")

##########################################################################
###
################################## FIN  ##################################
###
##########################################################################
###
```