

# Implementación de Servidor de Cola Única en Sanautos S.A.S

Caleb Villalba<sup>1</sup> and Ralph Sliger<sup>2</sup>

**Abstract**—La implementación de un servidor de cola única dentro del taller automotriz Sanautos es una herramienta útil para anticipar el flujo de trabajos que se recibirán en un periodo de tiempo. Para realizar esta implementación se extrajeron los datos del software de control de presencia de la empresa para obtener los tiempos de llegada y de servicio de 5 empleados en un periodo de 6 meses. A partir de los resultados obtenidos por el algoritmo es posible determinar si los trabajadores trabajan de manera eficiente.

## I. INTRODUCCIÓN

Los software de control de presencia son utilizados de manera amplia en toda clase de empresas, a partir de este es posible mantener la cuenta de la cantidad de tiempo que permanece un empleado dentro de la planta. Esta información es útil para realizar análisis a los procesos llevados a cabo dentro de la empresa. Para este trabajo se implementó un algoritmo de servidor de cola única en la empresa "Sanautos S.A.S" ubicada en la ciudad de Cartagena De Indias - Colombia. Para la realización de este trabajo se obtuvieron los datos necesarios del software de gestión de Sanautos "Spiga+" de 5 empleados que mejor utilizaron el software para contabilizar sus horas.

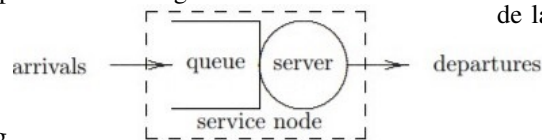
Con este trabajo se busca brindar información sobre la eficiencia de los trabajadores en un periodo de 6 meses desde 01/05/2018 hasta 31/12/2018 y brindar la capacidad de realizar este procedimiento de manera autónoma en la empresa.

## II. MARCO TEÓRICO

### A. Modelo conceptual de Servidor de Cola Única

Los automóviles llegan de manera aleatoria al taller buscando servicio, cuando el servicio es prestado se demora un tiempo aleatorio también. Al momento de terminar el trabajo el automóvil deja el taller.

Al momento de llegar el auto, si el trabajador está libre, este empezará a trabajar de manera inmediata en el auto que llegó. Por otra parte si el trabajador está ocupado en otro auto, el recién llegado entrará en la cola del trabajador. La disciplina de cola es FIFO (First In, First Out), es decir que el trabajador atenderá los autos por orden de llegada.



server.queue.jpg

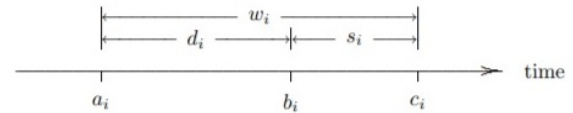
\*

C.Villalba y R.Sliger son estudiantes de la facultad de ingeniería de sistemas. Universidad Tecnológica de Bolívar, Cartagena de Indias, Colombia

### B. Modelo específico de Servidor de Cola Única

Para pasar del modelo conceptual a un modelo específico se asocian 6 variables a cada trabajo indexado con  $i = 0, 1, 2, 3, \dots$  automóvil que llega al taller:

- El tiempo de llegada del trabajo  $i$  es  $A_i$ .
- La demora del trabajo  $i$  en la cola es  $D_i \geq 0$ .
- El tiempo en donde el trabajo  $i$  empieza el servicio es  $B_i = A_i + D_i$ .
- El tiempo de servicio de un trabajo  $i$  es  $S_i > 0$ .
- Tiempo que dura el trabajo  $i$  en el taller (cola y servicio) es  $W_i = D_i + S_i$ .
- Tiempo en que el trabajo  $i$  completa el servicio (tiempo de salida) es  $C_i = A_i + W_i$



### C. Regresión lineal por mínimos cuadrados

El diagrama de dispersión es utilizado para determinar el tipo de relación entre dos variables, en donde este caso por la forma de los datos se eligió la regresión lineal para predecir el comportamiento de la relación entre variables.

Esta es una técnica estadística para estudiar la relación entre dos variables por medio del ajuste de una recta que conecte los puntos en donde estas se relacionan. Este método es útil para predecir fenómenos en un punto en el tiempo.

## III. DESARROLLO

### A. Completación de tablas de tiempos

Para la implementación del algoritmo de servidor de cola única se utilizó el lenguaje de programación Python.

El algoritmo recibe como parámetro las listas de tiempos de llegada y tiempos de servicio respectivamente de los trabajos para calcular sus tiempos de demora y los tiempos en que se completan estos. Se escogieron solamente 5 trabajadores debido a que los empleados restantes utilizaron de manera indebida el software de control de presencia.

Se almacenaron las listas de los tiempos de los trabajadores de la siguiente manera:

TABLE I  
LISTAS DE TIEMPOS DE EMPLEADOS

T. Llegada	Atable
T. Servicio	Stable
T. Demora	Dtable
T. Completa	Ctable

```

for i in range(len(Atable)):
    if (Atable[i] < Ctable[i-1]):
        Dtable.append(Ctable[i-1]-Atable[i])
    else:
        Dtable.append(0.0)
    Ctable[i]= Atable[i]+Stable[i]+Dtable[i]

```

## IV. CONCLUSIONES

### A. Presentación de resultados

Para presentar los datos al momento de ejecutar el programa se optó por imprimir las estadísticas de trabajo junto con la gráfica de la regresión lineal del trabajador escogido dentro del menú

### B. Implementación de regresión lineal

A través del método de mínimos cuadrados ordinarios se implementó un modelo de regresión lineal en python. Para la implementación del modelo se hace uso de la función de error cuadrático medio que permite hallar una matriz beta que permite hallar la línea de mejor ajuste para los valores trabajados. Para el desarrollo de la regresión en python se hizo uso de las librerías Numpy, seaborn que permitieron realizar las operaciones matriciales correspondientes para hallar y graficar la función.

Por cada empleado se tomó un valor de entrada 'x' que corresponde a los tiempos de llegada de cada empleado y la variable dependiente 'y' que corresponde al tiempo de demora de los empleados realizando la actividad. Con la regresión lineal nos permite predecir que tanto se demorará un empleado en desarrollar un trabajo de acuerdo a su hora de llegada.

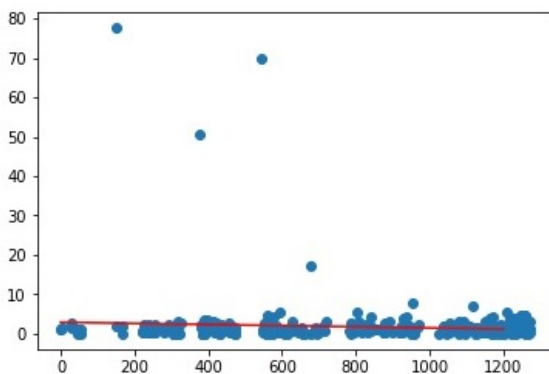
Para cada empleado en python se tomaron las listas donde se almacenaban su tiempo de llegada y la lista de la variable dependiente donde se almacenaba su tiempo de demora en la actividad.

Se definió una función beta igual a la inversa del valor de entrada 'x' por la matriz transpuesta de 'x' todo esa operación multiplicada por la matriz transpuesta de 'x' por la variable 'y' que halla una matriz con dos datos: el valor independiente y la pendiente. Con estos dos datos obtenidos por beta se puede hacer la regresión lineal para cada empleado a través de la librería Seaborn.

- Fórmula para hallar matriz de término independiente y pendiente:

$$\beta = (X^T X)^{-1} X^T Y$$

- Ejemplo de implementación de regresión lineal.



### B. conclusiones generales

#### Job-Average Statistics

Average Inter-Arrival Time: 3.7423446082751943  
 Average Service Time: 1.7948816197847353  
 Average Delay Time: 5.571876675275897

Arrival Rate: 0.26721216367642026  
 Service Rate: 0.5571398074263708  
 Average Wait: 7.366758295060633

#### Time-Averaged Statistics

Time-Averaged Number In The Node: 28.780104009179798  
 Time-Averaged Number In The Queue: 21.92298652299821  
 Time-Averaged Number In Service: 6.857117486181591

- Según los resultados arrojados, la tasa de servicio es aproximadamente 0.55 trabajos por hora, mientras que la tasa de llegada es 0.26 trabajos cada hora. Esto indica que los trabajos se completan más rápido de lo que ingresan al taller ms por un margen de 0.29. Es decir que cada hora que transcurre se irn acumulando los trabajos en la cola.

### C. Conclusiones específicas

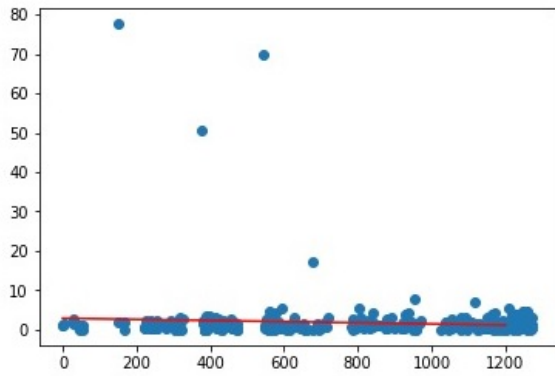
- Empleado 1: Alfonso. Tasa de llegada = 0.25 Tasa de servicio = 1.85
- Empleado 2: Efrain. Tasa de llegada = 0.25 Tasa de servicio = 0.65
- Empleado 3: Elio. Tasa de llegada = 0.25. Tasa de servicio = 0.52.
- Empleado 4: German. Tasa de llegada = 0.32. Tasa de servicio = 0.66.
- Empleado 5: Jaime. Tasa de llegada = 0.25. Tasa de servicio = 0.46.

De manera general se puede afirmar que los empleados completan los trabajos de forma eficiente puesto que sus tasas de servicio son mayores a sus tasas de llegada, por lo que pueden completar trabajos más rápido de lo que llegan estos al taller.

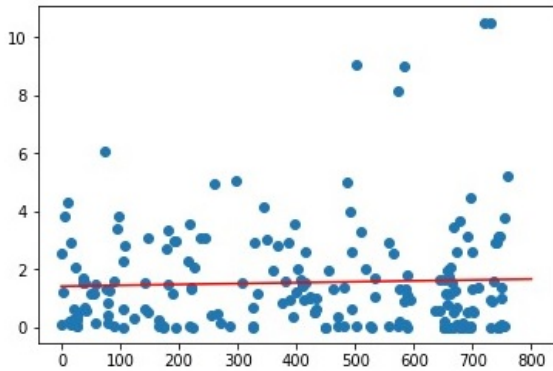
El empleado más eficiente es Efrain por tener el mayor margen entre sus tasas, siendo este de 0.41. Mientras que el empleado más ineficiente es Jaime por tener el menor margen de 0.21.

## V. ANEXOS

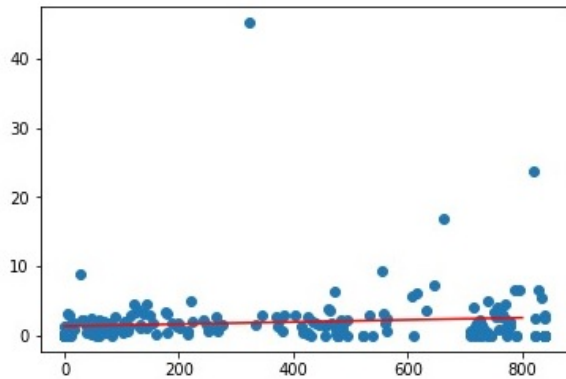
- Regresión lineal del empleado 1: Alfonso.



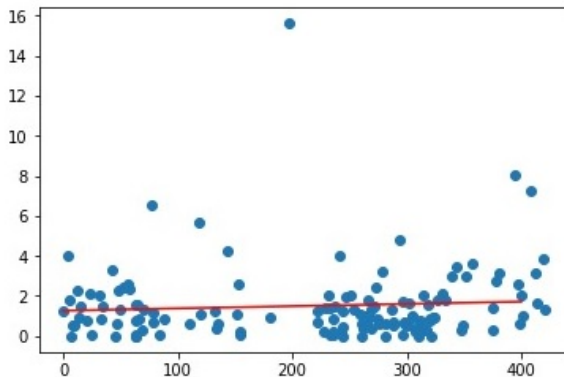
- Regresión lineal del empleado 2: Efrain.



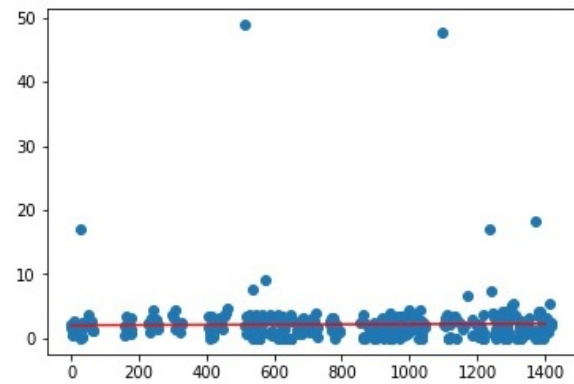
- Regresión lineal del empleado 3: Elio.



- Regresión lineal del empleado 4: German.



- Regresión lineal del empleado 5: Jaime.



## REFERENCES

- [1] Birta, L. and Arbez, G. (2007). Modelling and simulation. London: Springer.
- [2] Ahmad Koka, T., Badshah, V. and RAhmad Shah, R. (2017). Single and Multi Server Queuing Models: A Study. 5th ed.
- [3] Grus, J. (2015). Data Science From Scratch. O'Reilly Media, Inc.
- [4] Guido, S. and Miller, A. (2019). Introduction To Machine Learning With Python. O'Reilly Media, Inc.
- [5] Leemis, L. and Park, S. (2019). Discrete Event Simulation: A First Course. 6th ed. pp.12-25.
- [6] VanderPlas, J. (2016). Python Data Science Handbook. O'Reilly Media, Inc.