

**Experimentación con Sistemas Distribuidos  
utilizando protocolo TCP**

Guillermo Pizarro

Guayaquil, Ecuador

11 de noviembre de 2024

# Contenido

- Listado de Gráficos ..... 3
- Introducción ..... 4
- Objetivos ..... 4
  - Objetivo general ..... 4
  - Objetivos específicos..... 4
- Diseño del Experimento..... 4
  - Diseño del Almacén Rectangular ..... 4
  - Generación de pedidos ..... 6
  - Algoritmo de Agrupación de pedidos ..... 6
  - Algoritmo de Recogida de pedidos ..... 7
- Experimentación ..... 9
- Referencias Bibliográficas..... 9

## Listado de Gráficos

|   |   |
|---|---|
| Gráfico 1. Diseño del almacén rectangular.....                                | 5 |
| Gráfico 2. Ubicaciones en el almacén rectangular .....                        | 5 |
| Gráfico 3. Medidas en el pasillo frontal del almacén rectangular.....         | 5 |
| Gráfico 4. Medidas en el pasillo posterior del almacén rectangular. ....      | 6 |
| Gráfico 5. Enlace entre la ubicación y el id del ítem.....                    | 6 |
| Gráfico 6. Lista de pedidos.....  | 7 |
| Gráfico 7. Generación de los lotes con pedidos aleatorios. ....               | 7 |
| Gráfico 8. Camino por donde deben pasar los recolectores de los pedidos. .... | 8 |
| Gráfico 9. Algoritmo de recogida de pedidos S-SHAPE (caso 1).....             | 8 |
| Gráfico 10. Algoritmo de recogida de pedidos S-SHAPE (caso 2). ....           | 8 |
| Gráfico 11. Algoritmo de recogida de pedidos S-SHAPE (caso 3). ....           | 8 |

# Introducción

En el siguiente trabajo se debe implementar una simulación en un ambiente en programación secuencial, paralela y distribuida para comparar los tiempos de ejecución y verificar si es factible el uso de los sistemas distribuidos en un ambiente de procesamiento masivo de datos es pertinente o no.

## Objetivos

### Objetivo general

Implementar una simulación de la búsqueda de las distancias recorridas mediante la agrupación y recogida de pedidos en un almacén rectangular en un ambiente secuencial, paralelo y distribuido.

### Objetivos específicos

- Implementar un generador de pedido.
- Diseñar e implementar un simulador para un almacén rectangular para el cálculo de las distancias recorridas mediante la agrupación y recogida de pedidos en un almacén rectangular.
- Implementar en programación secuencial, paralela y distribuida utilizando contenedores.

## Diseño del Experimento

### Diseño del Almacén Rectangular

Implementar un almacén rectangular (ver **Gráfico 1**) con las siguientes características:

- Debe tener un diseño rectangular.
- Debe contar con 10 pasillos.
- Debe tener un pasillo frontal y un pasillo posterior.
- Con 200 ubicaciones (ver **Gráfico 2**) para almacenar los ítems a recoger (en cada ubicación es un ítem diferente, y en cada ubicación para la simulación siempre habrá un ítem a recoger; es decir, que siempre habrá el item004 en la ubicación 4 cuando pase el recolector).
- Cada ubicación tiene asignado un identificador como se muestra en el **Gráfico 2**.
- Con un depot en el pasillo frontal y frente al pasillo 1.

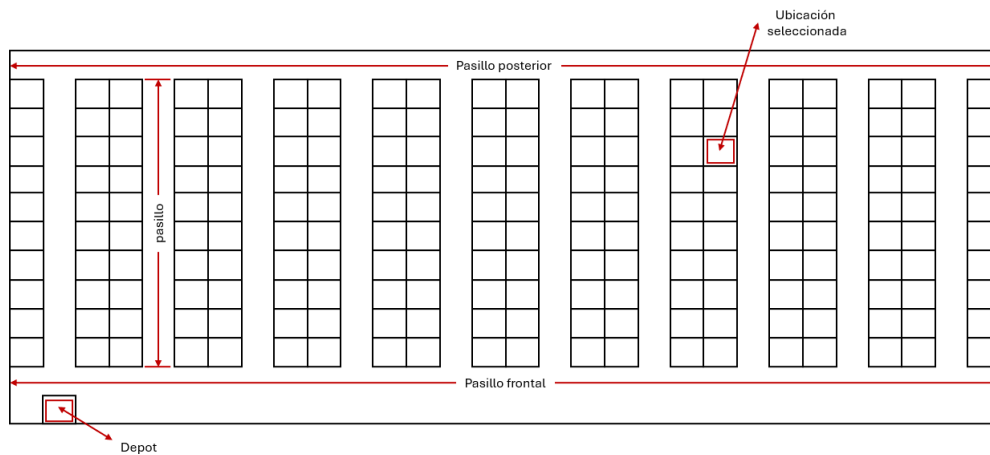


Gráfico 1. Diseño del almacén rectangular

|   |    |    |    |    |    |    |    |    |    |     |     |     |     |     |     |     |     |     |     |
|---|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 0 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 | 110 | 120 | 130 | 140 | 150 | 160 | 170 | 180 | 190 |
| 1 | 11 | 21 | 31 | 41 | 51 | 61 | 71 | 81 | 91 | 101 | 111 | 121 | 131 | 141 | 151 | 161 | 171 | 181 | 191 |
| 2 | 12 | 22 | 32 | 42 | 52 | 62 | 72 | 82 | 92 | 102 | 112 | 122 | 132 | 142 | 152 | 162 | 172 | 182 | 192 |
| 3 | 13 | 23 | 33 | 43 | 53 | 63 | 73 | 83 | 93 | 103 | 113 | 123 | 133 | 143 | 153 | 163 | 173 | 183 | 193 |
| 4 | 14 | 24 | 34 | 44 | 54 | 64 | 74 | 84 | 94 | 104 | 114 | 124 | 134 | 144 | 154 | 164 | 174 | 184 | 194 |
| 5 | 15 | 25 | 35 | 45 | 55 | 65 | 75 | 85 | 95 | 105 | 115 | 125 | 135 | 145 | 155 | 165 | 175 | 185 | 195 |
| 6 | 16 | 26 | 36 | 46 | 56 | 66 | 76 | 86 | 96 | 106 | 116 | 126 | 136 | 146 | 156 | 166 | 176 | 186 | 196 |
| 7 | 17 | 27 | 37 | 47 | 57 | 67 | 77 | 87 | 97 | 107 | 117 | 127 | 137 | 147 | 157 | 167 | 177 | 187 | 197 |
| 8 | 18 | 28 | 38 | 48 | 58 | 68 | 78 | 88 | 98 | 108 | 118 | 128 | 138 | 148 | 158 | 168 | 178 | 188 | 198 |
| 9 | 19 | 29 | 39 | 49 | 59 | 69 | 79 | 89 | 99 | 109 | 119 | 129 | 139 | 149 | 159 | 169 | 179 | 189 | 199 |

Gráfico 2. Ubicaciones en el almacén rectangular

Las dimensiones del almacén son las siguientes:

- La ubicación donde se almacena el ítem tiene las dimensiones de 1U de largo y 1U de ancho, lo mismo con el depot.
- El ancho del pasillo frontal es de 2U y el largo es de 30U (**Gráfico 3**).
- El ancho del pasillo posterior es de 1U y el largo es de 30U (**Gráfico 4**).
- Un pasillo tiene 1U de ancho y 10U de largo.

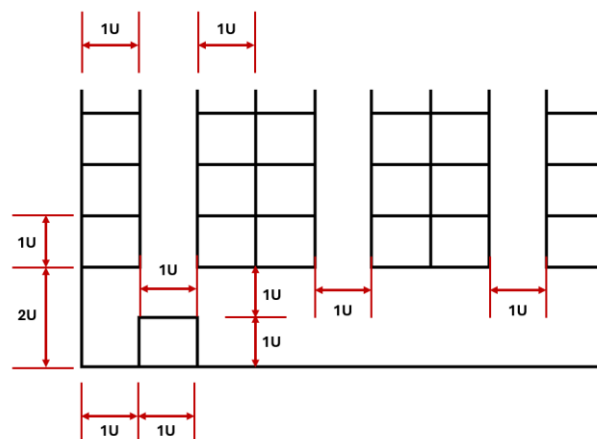


Gráfico 3. Medidas en el pasillo frontal del almacén rectangular.

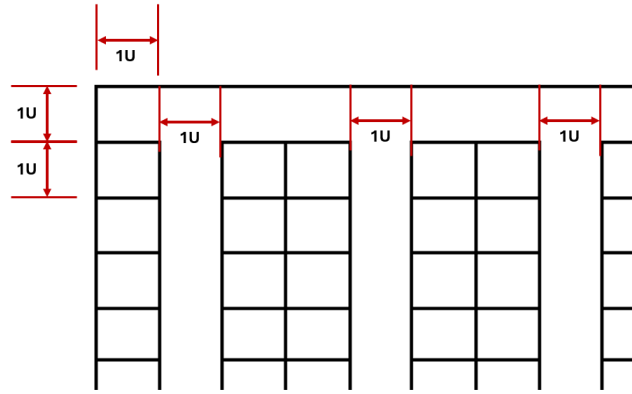


Gráfico 4. Medidas en el pasillo posterior del almacén rectangular.

## Generación de pedidos

Implementar un generador de pedidos, donde cada pedido contenga las siguientes especificaciones:

- Puede contener desde 5 ítems hasta máximo 30 ítems;
- No se puede repetir ítem en un pedido.

| Ubicación | ID ítem |
|-----------|---------|
| 0         | item000 |
| 1         | item001 |
| 2         | item002 |
| 3         | item003 |
| 4         | item004 |
| ...       |         |
| ...       |         |
| 197       | item197 |
| 198       | item198 |
| 199       | item199 |

Gráfico 5. Enlace entre la ubicación y el id del ítem.

## Algoritmo de Agrupación de pedidos

Implementar un algoritmo de agrupación de pedidos denominado como **RANDOM**; el cual, primero se tiene una lista de pedidos y luego generar lotes con pedidos aleatorios con un máximo de 45 ítems, cada pedido debe estar en un solo pedido.

Por ejemplo, se tiene una lista de pedidos (ver **Gráfico 6**); luego, se forman lotes con pedidos únicos hasta llegar a la capacidad máxima del lote (ver **Gráfico 7**) que es la capacidad del carrito donde se van colocando los ítems.

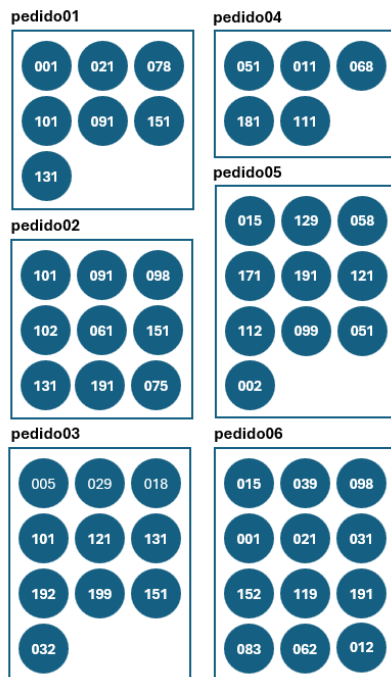


Gráfico 6. Lista de pedidos.



Gráfico 7. Generación de los lotes con pedidos aleatorios.

## Algoritmo de Recogida de pedidos

Por cada lote, se recogen los pedidos mediante el algoritmo S-SHAPE.

Al iniciar desde el depot, en el pasillo frontal, si hay ítem a recoger en el primer pasillo, se debe ingresar (caso 1); caso contrario, se debe continuar desde el pasillo frontal al siguiente pasillo hasta encontrar un ítem y cruzar el pasillo (caso 2).

Se debe buscar desde los pasillos frontal y posterior avanzando hacia la derecha (desde cualquiera de los dos), un pasillo donde existan ítems a recoger, realizando un movimiento S en los siguientes pasillos como se ve en los casos 1 y 2; es decir, cruzar el pasillo completo desde el pasillo frontal al posterior o desde el posterior al frontal, cuando existen ítems a recoger.

Para finalizar el recorrido, depende donde se encuentra el recolector antes de recorrer el último pasillo:

Si se encuentra en el pasillo posterior, y va a cruzar un pasillo (porque existe donde recoger) para ir al pasillo frontal, como en el caso 2.

Si se encuentra en el pasillo frontal, y llega al último pasillo a recorrer, debe ir hasta el elemento a recoger y regresar al pasillo frontal (donde estaba antes), este es el

único caso donde NO SE RECORRE TODO EL PASILLO, como en el caso 1; esto se debe hacer, aunque el ítem se encuentre al final del pasillo (caso 3), igual hay que regresar al pasillo frontal.

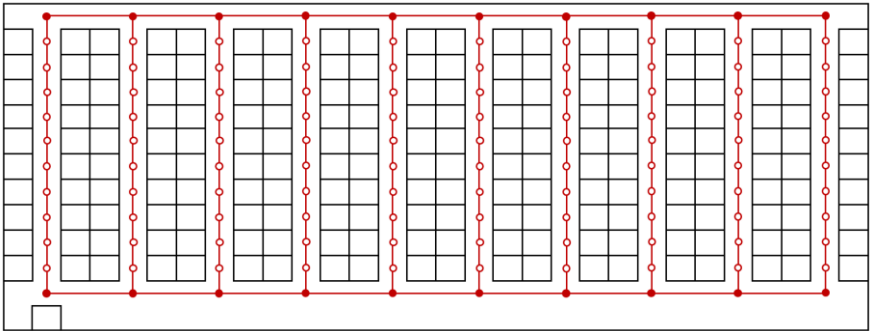


Gráfico 8. Camino por donde deben pasar los recolectores de los pedidos.

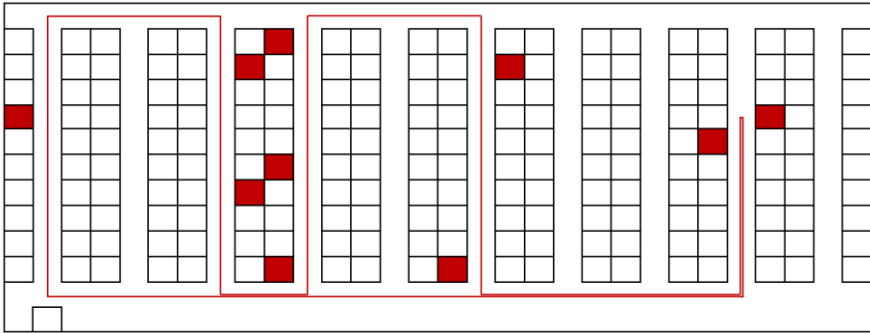


Gráfico 9. Algoritmo de recogida de pedidos S-SHAPE (caso 1).

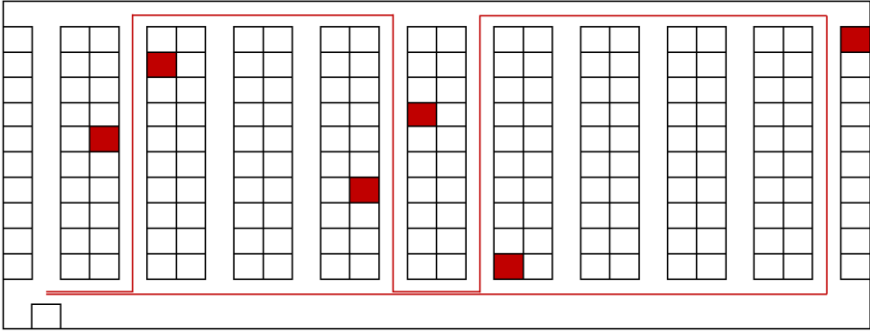


Gráfico 10. Algoritmo de recogida de pedidos S-SHAPE (caso 2).

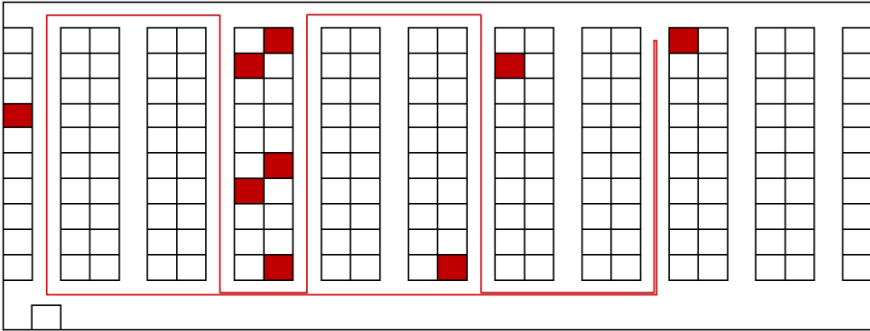


Gráfico 11. Algoritmo de recogida de pedidos S-SHAPE (caso 3).



Experimentación

Referencias Bibliográficas