[PROYECTO]: Optimización de almacén

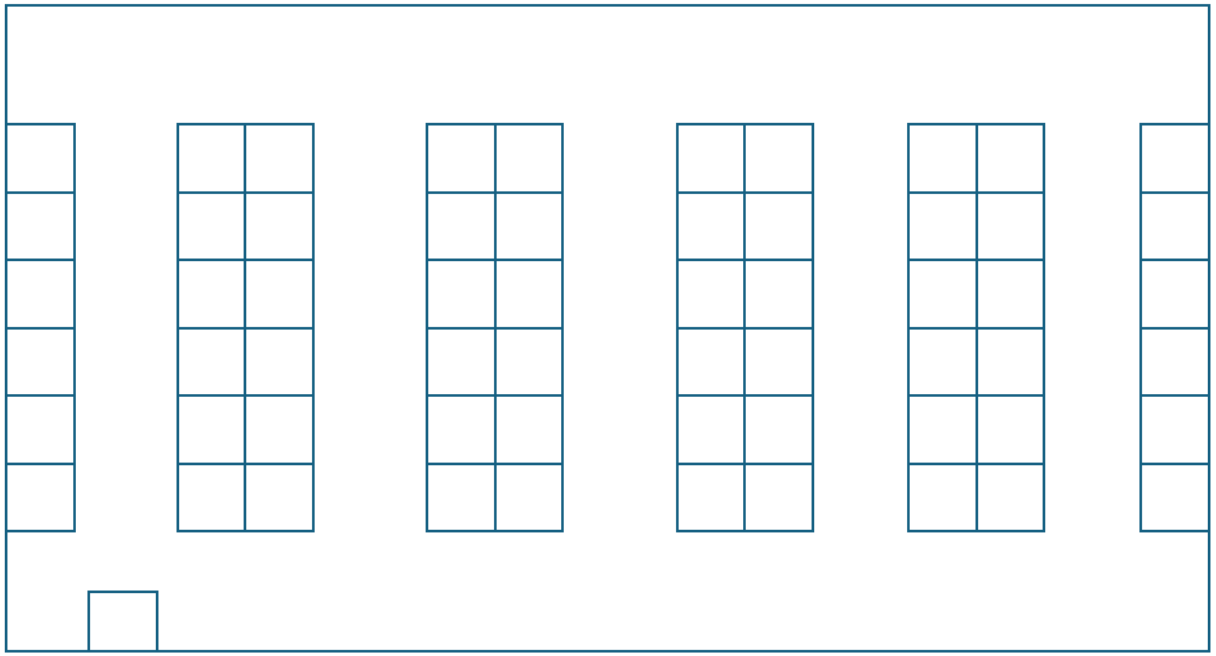
Guillermo Pizarro, guillermopv@uees.edu.ec

Descripción

Proyecto del Curso

Descripción del almacén rectangular

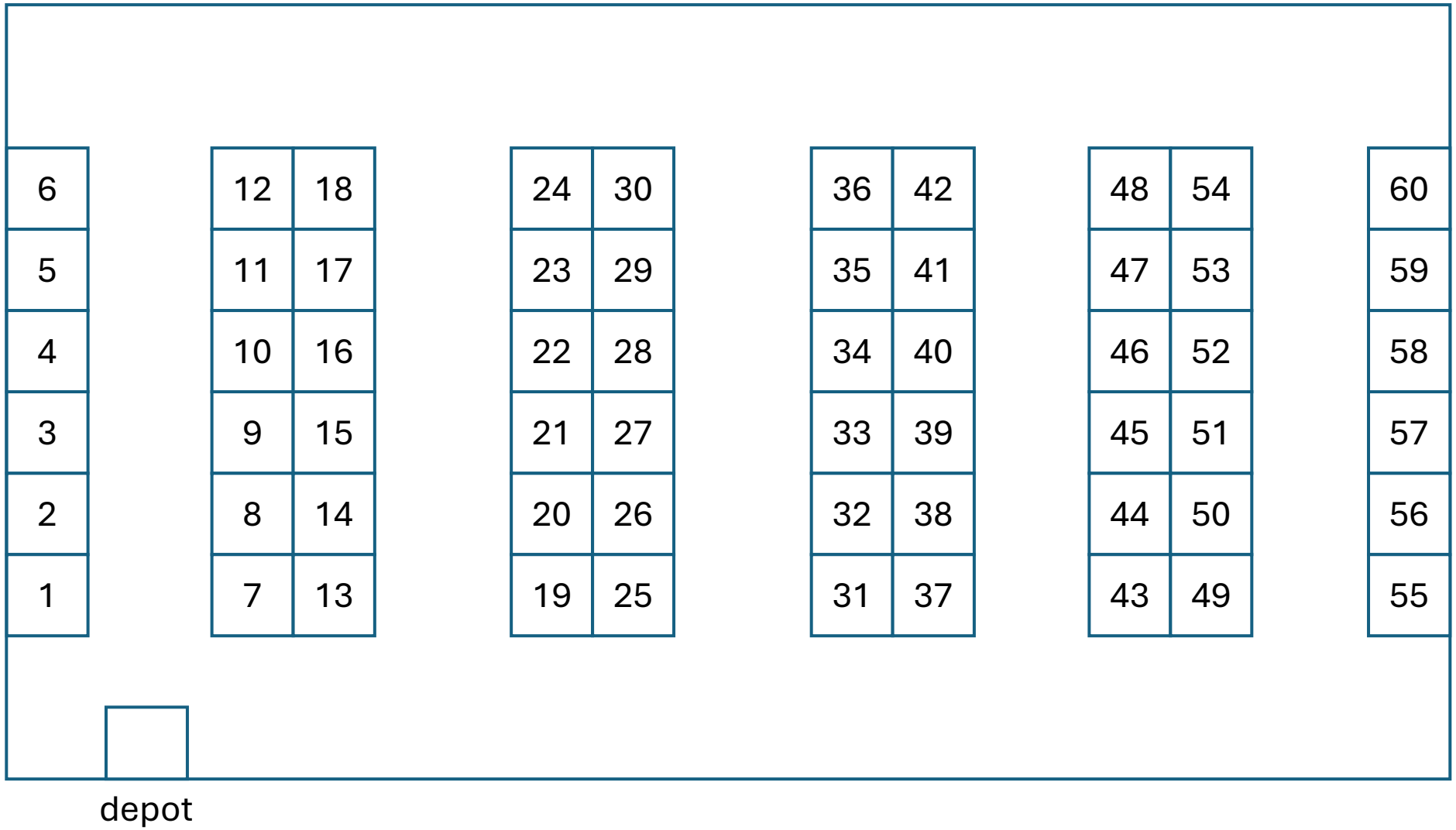
- Implementar el simulador de un almacén rectangular que tiene 5 pasillos y 60 ubicaciones (12 ubicaciones por pasillo) donde van alojados los ítems, en cada ubicación se aloja un ítem diferente (no se lleva el registro de la cantidad de ítems en cada ubicación).



Descripción del almacén rectangular

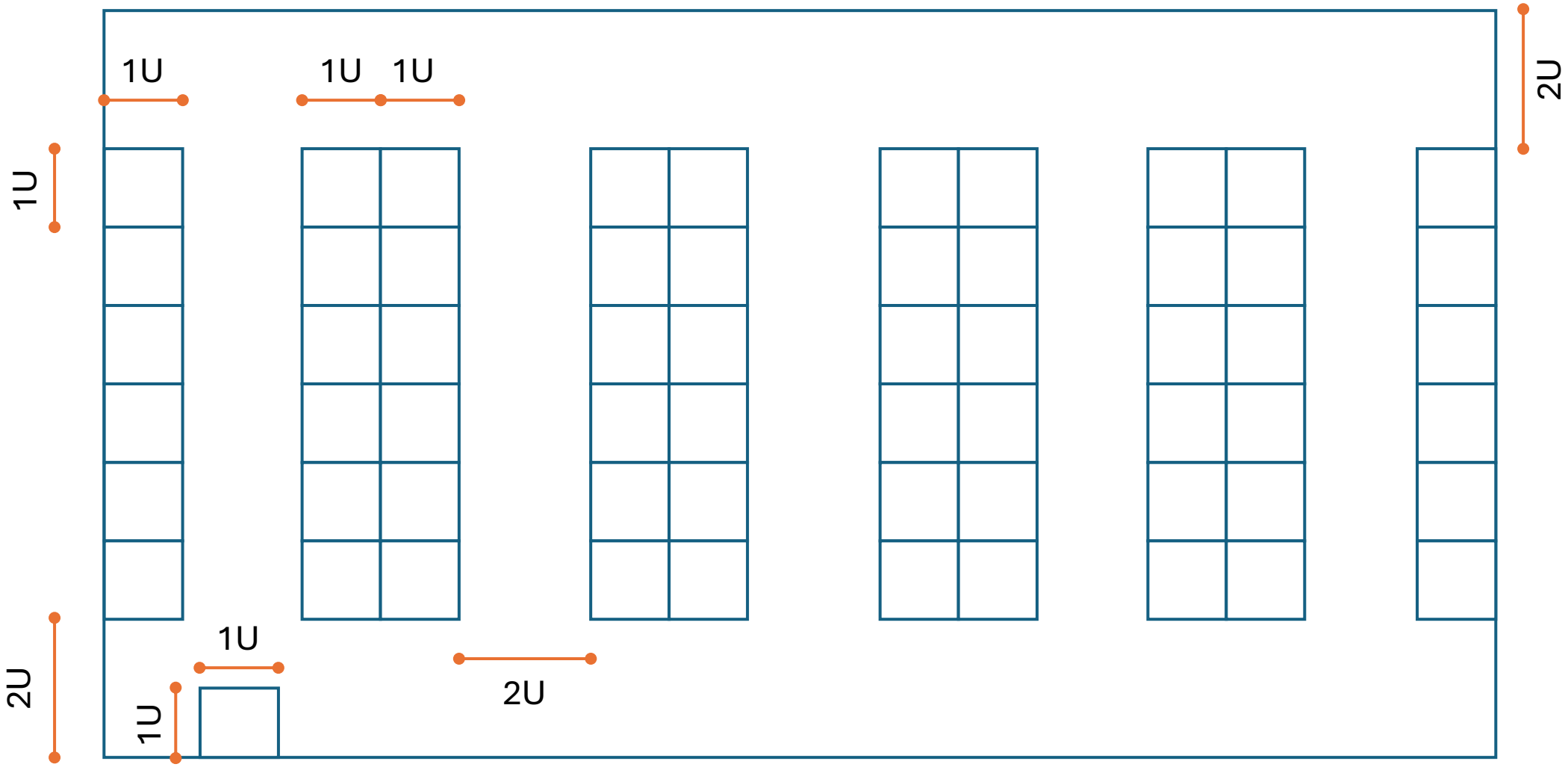
- Los ítems van en cada ubicación de la siguiente manera:
  - El ítem 4, va en la ubicación 4.
  - El ítem 5, va en la ubicación 5, y así sucesivamente.

- Tal como se muestra en la siguiente figura.


6
5
4
3
2
1
12
11
10
9
8
7
18
17
16
15
14
13
24
23
22
21
20
19
30
29
28
27
26
25
36
35
34
33
32
31
42
41
40
39
38
37
48
47
46
45
44
43
54
53
52
51
50
49
60
59
58
57
56
55
depot

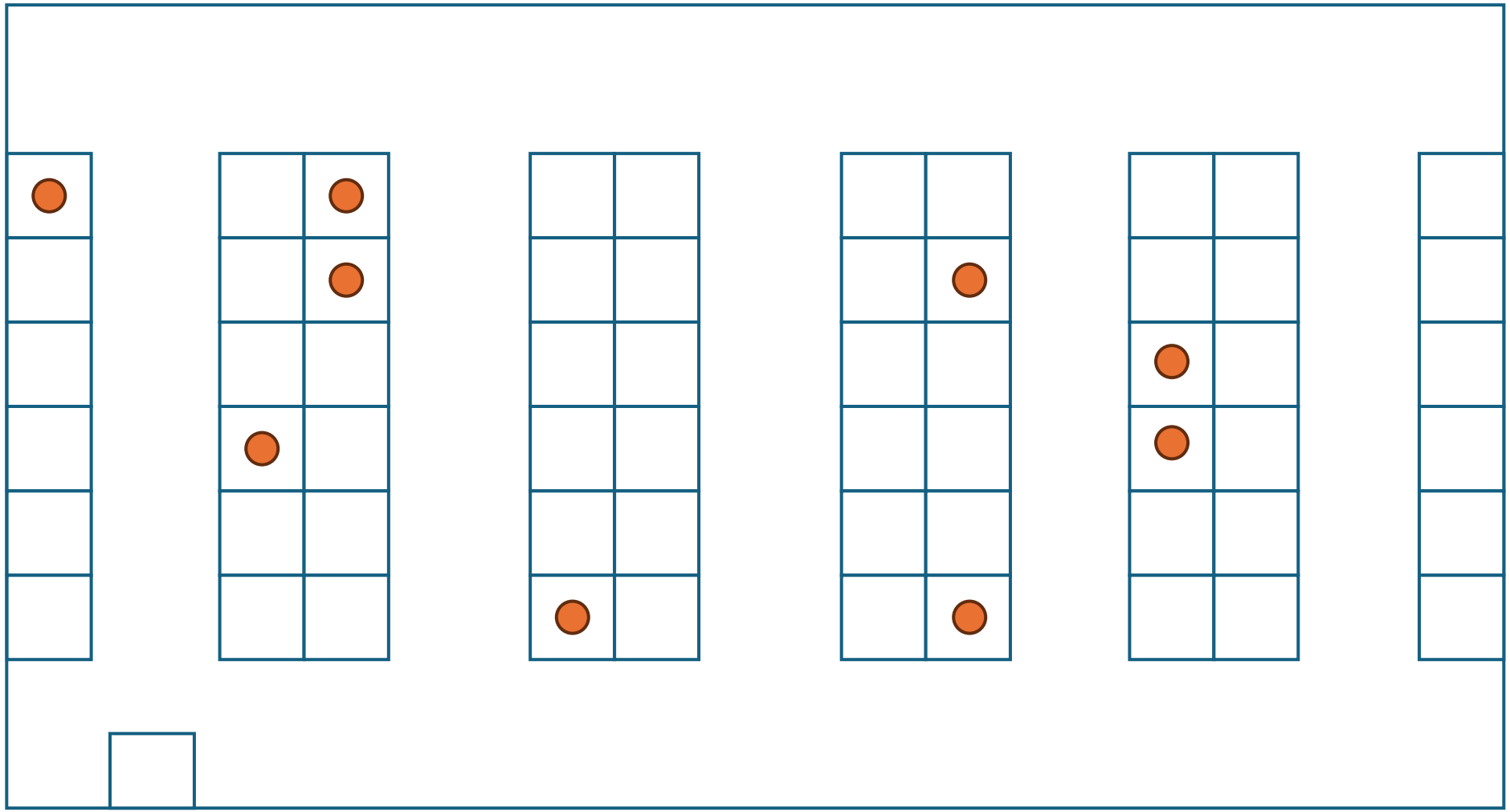
Descripción del almacén rectangular

- La velocidad de recorrido en el almacén es de 1 m/s y la capacidad del carrito donde van los elementos del pedido es de 10 ítems.
- Las dimensiones del almacén se detallan en la siguiente figura.


1U
1U
1U
2U
1U
1U
2U
1U
2U

Distribución de los ítems de un pedido en el almacén

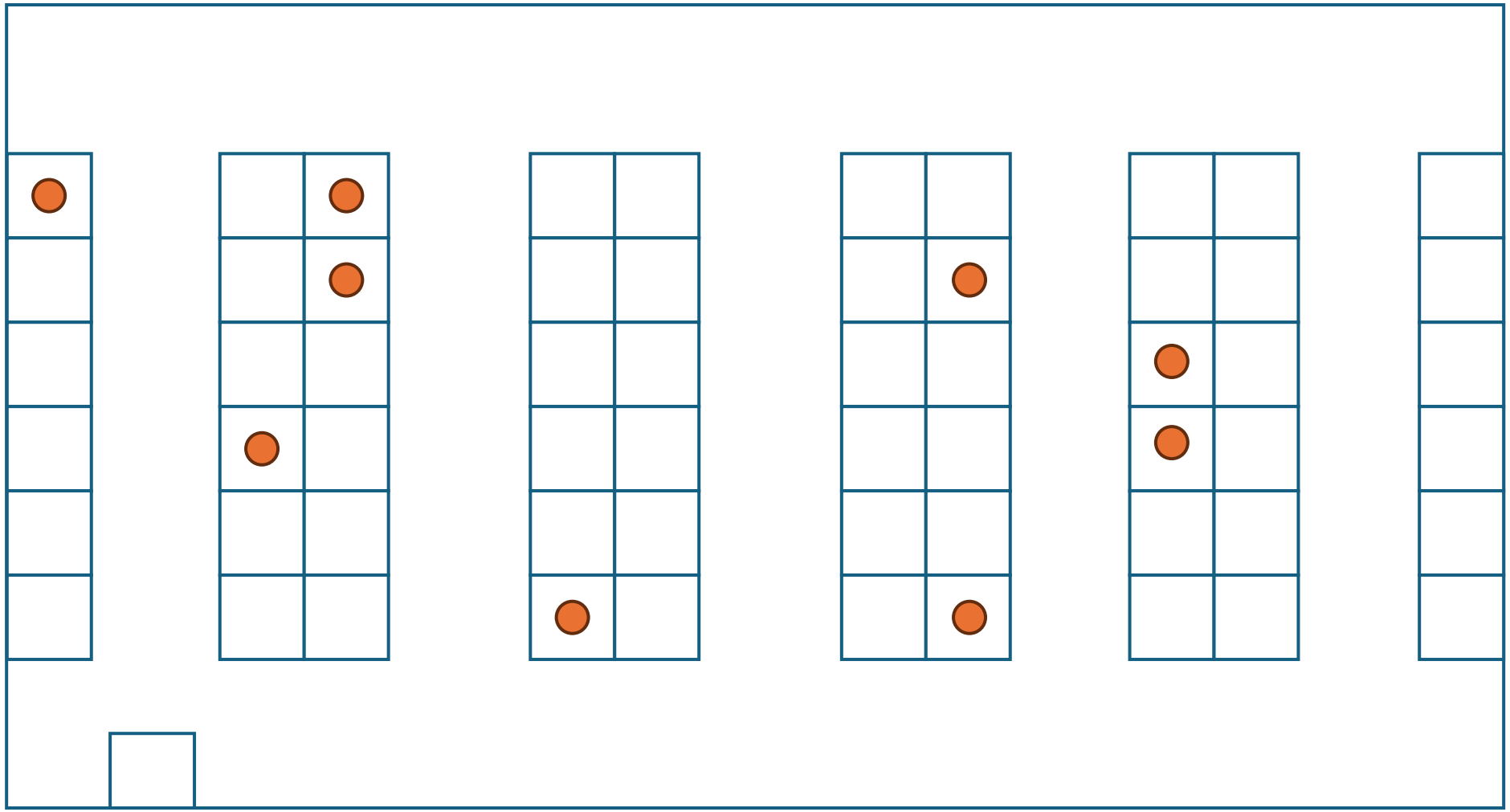
- A continuación, un ejemplo sobre cómo se distribuyen los elementos de un pedido en un almacén para ir a recoger los ítems.



Creación de la lista de pedidos

- Los pedidos deben tener como mínimo 5 elementos hasta 10 elementos.
- Los elementos se pueden repetir en el pedido.
- Implementar un generador de lista de pedidos aleatorio que permita ingresar el número de pedidos a generar. No hay un número máximo para los pedidos.
- A continuación, un ejemplo de un pedido, que tiene relación con el gráfico anterior.

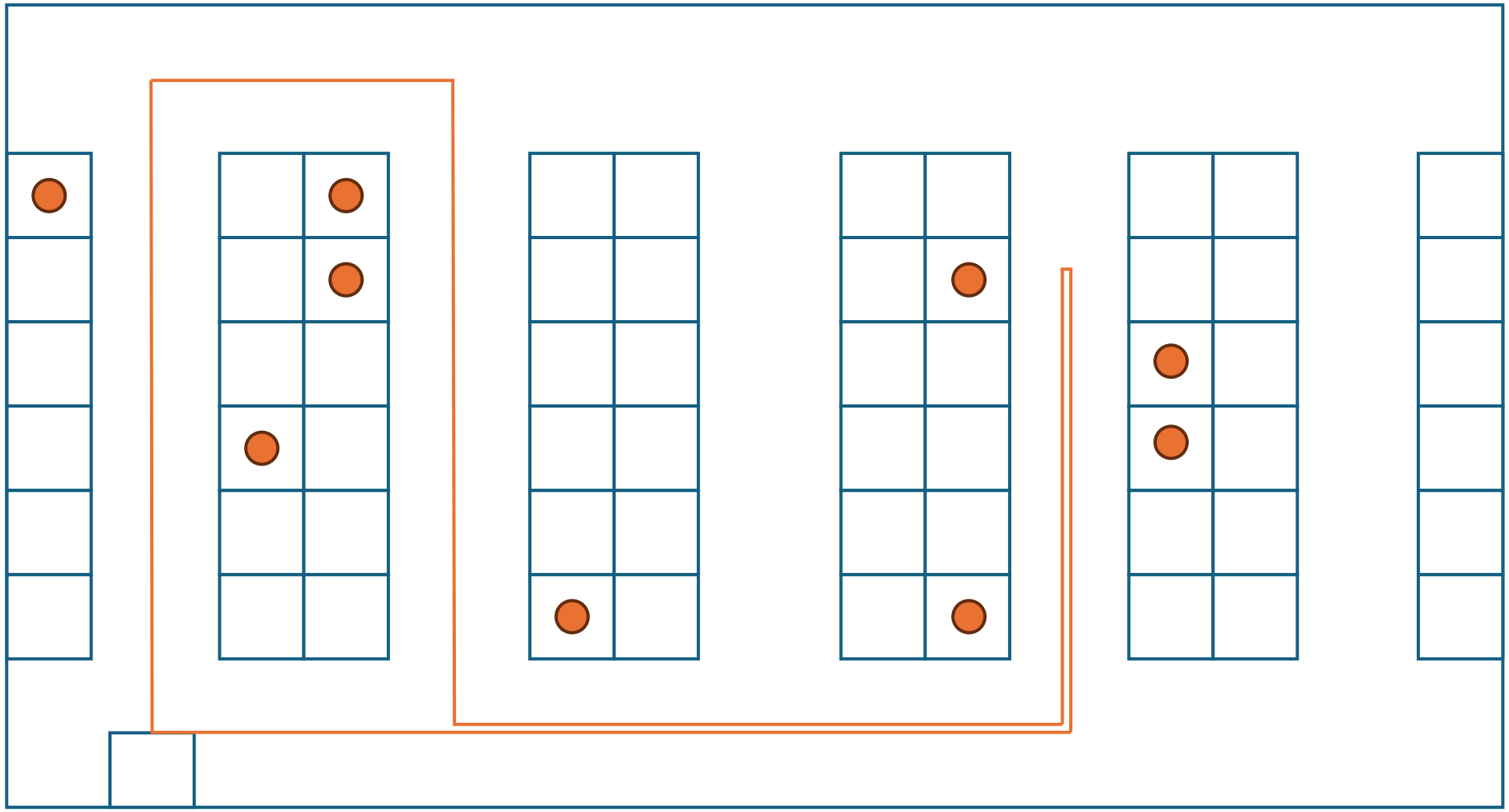
pedido p1
item 6
item 9
item 17
item 18
item 19
item 37
item 37
item 41
item 45
item 46



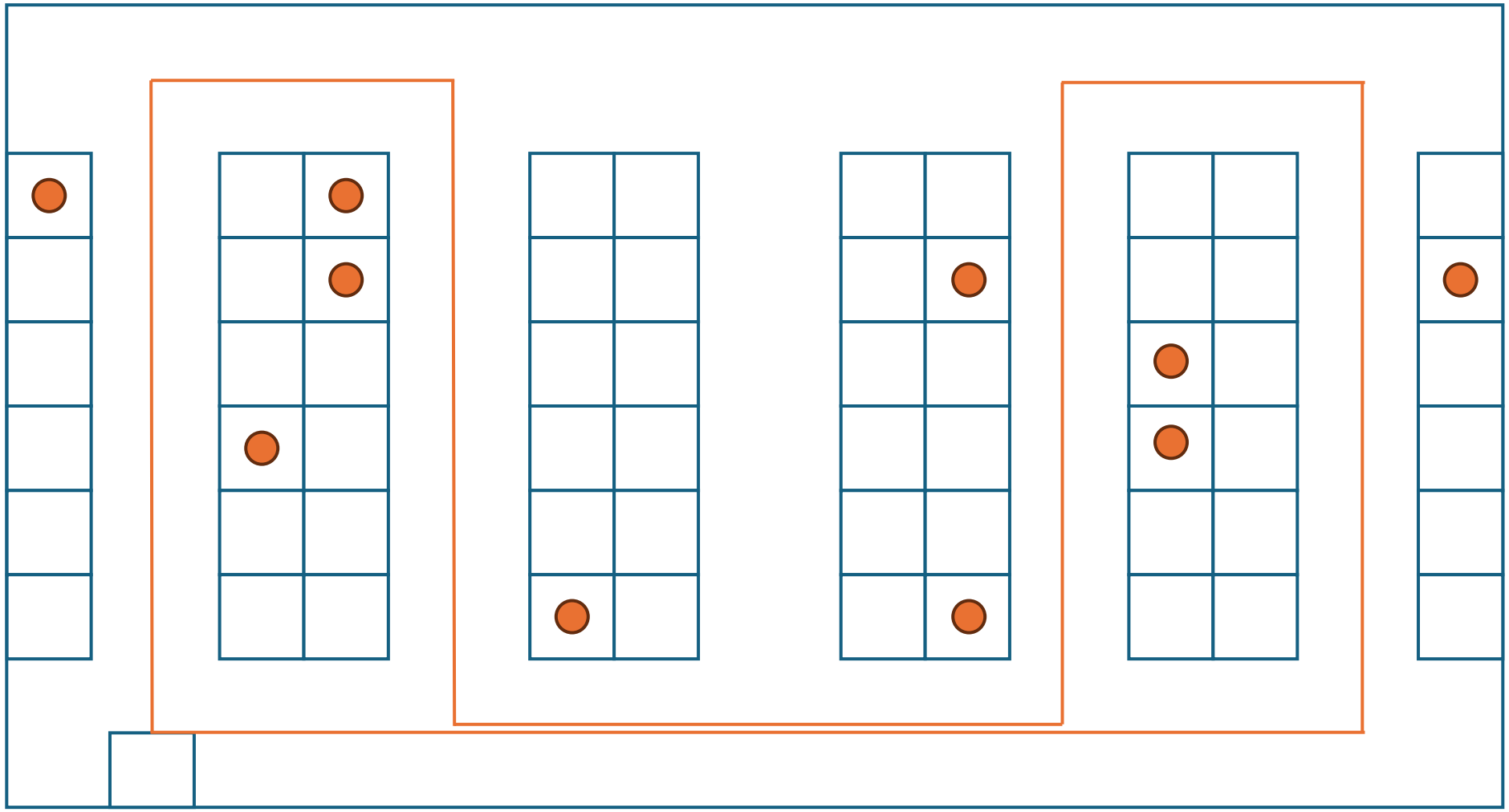
pedido p1
item 6
item 9
item 17
item 18
item 19
item 37
item 37
item 41
item 45
item 46

Algoritmo de Recorrido

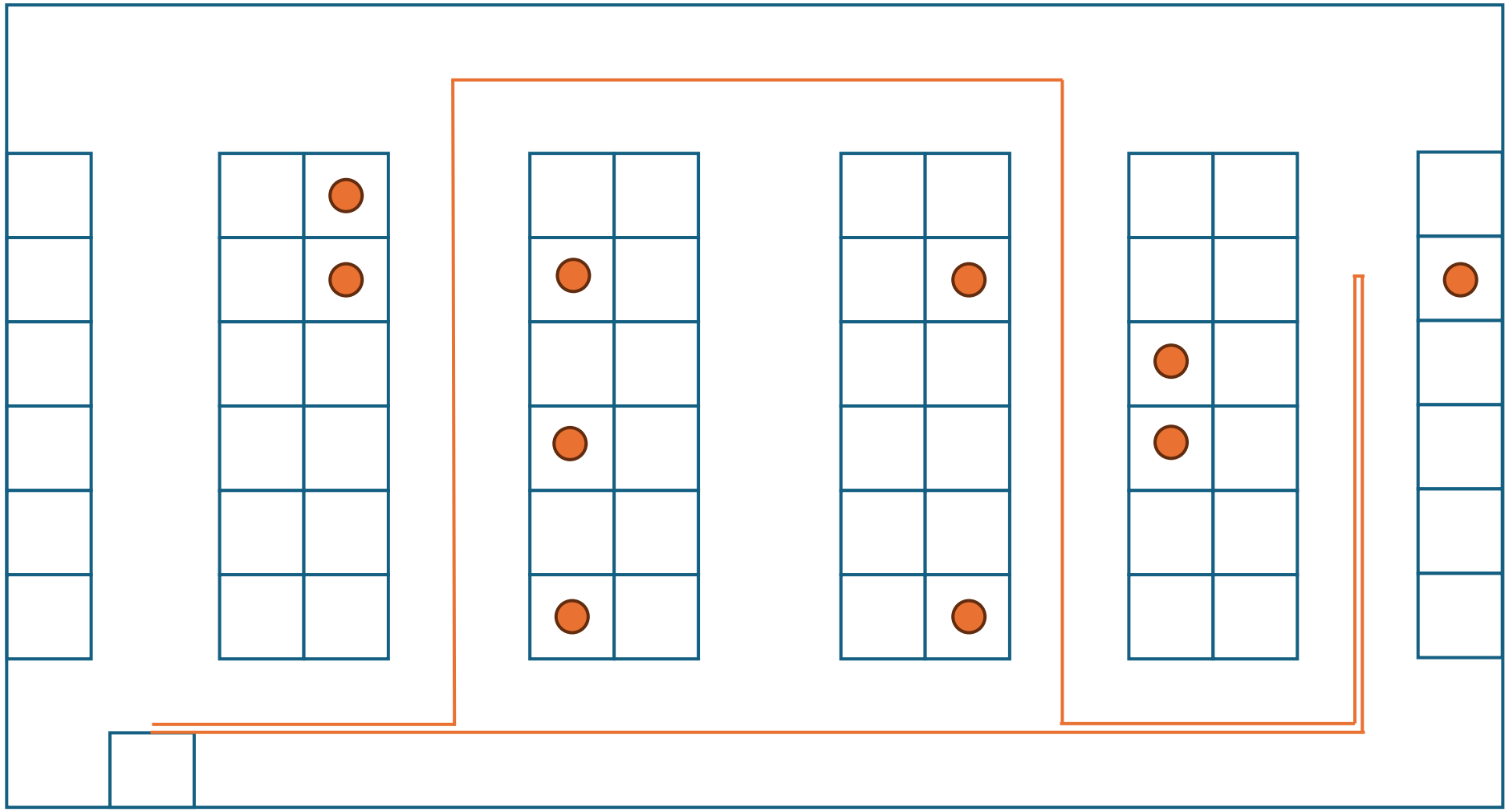
- El algoritmo a utilizar sería el S-Shape; el cual, realiza movimientos como la S.
- Al iniciar el recorrido se busca un pasillo donde se encuentren ítems a recoger; caso contrario, continúo al siguiente pasillo (**S-Shape:** Caso 3).
- Si llego al final del recorrido, si ingreso desde el frente del almacén y ya no hay más para recoger, me regreso por el mismo pasillo y voy hacia el depot (**S-Shape:** Caso 1).
- Si llego al final del recorrido, si ingreso desde el fondo del almacén y ya no hay más para recoger, recorro todo el pasillo y regreso al depot (**S-Shape:** Caso 2).



S-Shape: Caso 1



S-Shape: Caso 2



S-Shape: Caso 3

Resultados

- Al final de la animación se debe generar un archivo denominado como **resultados.txt** donde se almacene por cada pedido, la distancia total recorrida y el tiempo de recorrido.
- A continuación un ejemplo del archivo que contiene solo dos pedidos.

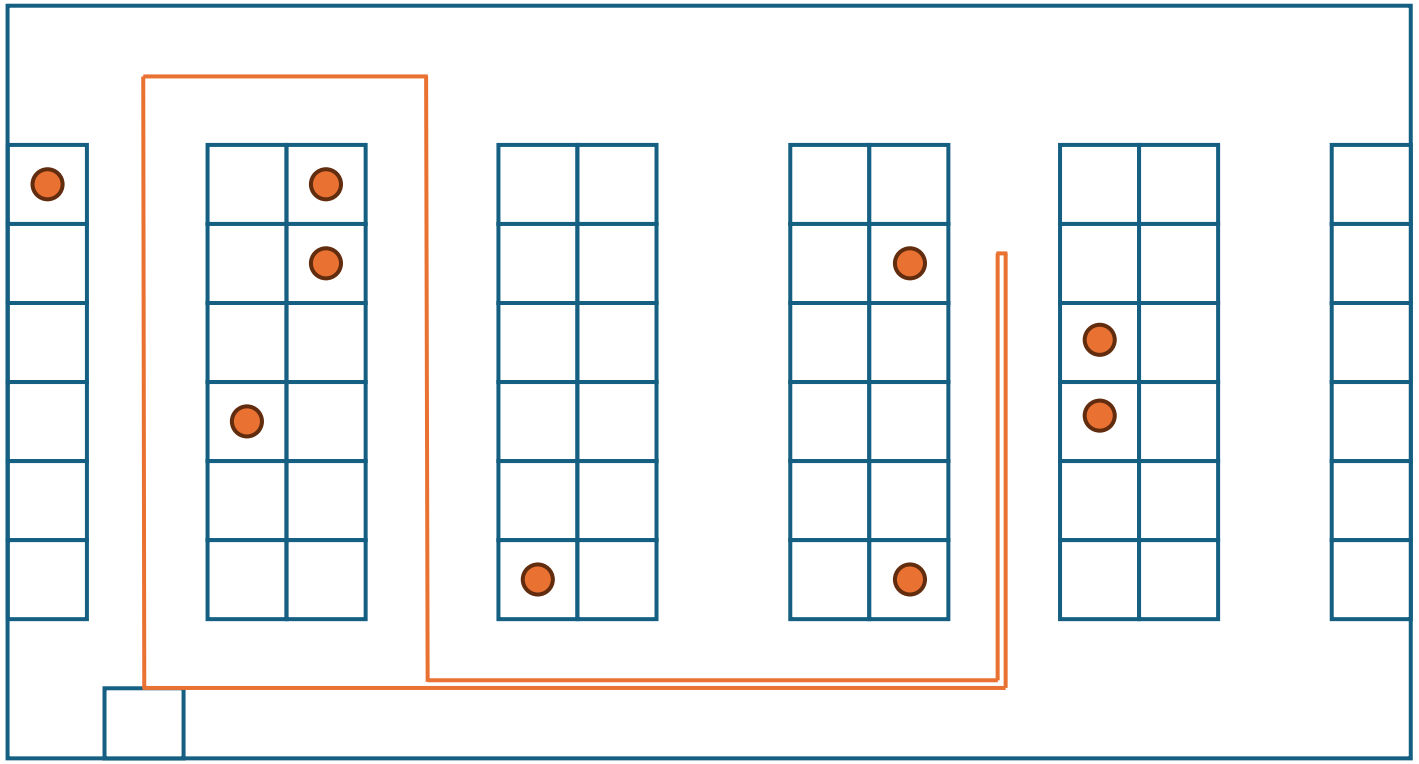
```
pedido p1,140,140
item 6
item 9
item 17
item 18
item 19
item 37
item 37
item 41
item 45
item 46
pedido p2,85,85
item 46
item 19
item 37
item 18
item 29
item 17
```

Especificaciones Técnicas

Proyecto del Curso

Número de Pedidos:

Ejecutar | Parar | Finalizar



Pedido: _____

Distancia total: _____

Tiempo: _____

Especificaciones Técnicas

- Interfaz gráfica
  - El proyecto debe estar implementado en Java con interfaz gráfica en Java Swing.

Especificaciones Técnicas

- Programación Concurrente
  - Usar un hilo para cada etapa del proceso: la generación de pedidos, el cálculo del recorrido, generación del archivo de **resultados.txt** y la recolección de ítems.
  - Esta estructura divide el proceso en etapas, donde cada etapa es manejada por un hilo separado. Aunque el recorrido en sí es secuencial (un solo carrito en un momento dado), puedes utilizar hilos para manejar la generación de pedidos, el cálculo del recorrido, generación de archivo y la recolección de ítems de manera concurrente, mejorando la eficiencia del programa en general.

Entregables

Proyecto del Curso

Entregables

- Código fuente en Java.
- Documentación del proyecto.

Rúbrica

Proyecto del Curso

Criterio	Puntos	Descripción
Funcionalidad General	20	El simulador cumple con los requisitos básicos del proyecto: generación de pedidos, cálculo del recorrido, y recolección de ítems.
Uso de Hilos	20	Implementación correcta y eficiente de hilos para manejar diferentes etapas del proceso (generación de pedidos, cálculo del recorrido, recolección de ítems).
Sincronización y Concurrencia	20	Uso adecuado de técnicas de sincronización (synchronized, wait/notify) y mecanismos de concurrencia (semaphores, locks) para gestionar el acceso a recursos compartidos.
Algoritmo de Recorrido (S-Shape)	10	Implementación del algoritmo S-Shape de manera correcta y eficiente, con manejo adecuado de los diferentes casos descritos.
Interfaz Gráfica (Java Swing)	10	Interfaz gráfica intuitiva y funcional para interactuar con el simulador, mostrando la información relevante y permitiendo controlar el proceso.
Generación de Archivo de Resultados	10	Generación correcta y detallada del archivo resultados.txt con la distancia total recorrida y el tiempo de recorrido por cada pedido.
Calidad del Código	10	Código bien estructurado, legible y comentado, siguiendo buenas prácticas de programación en Java.
Documentación	5	Documentación adecuada del proyecto, explicando el diseño, la implementación y el uso del simulador.
Innovación y Creatividad	5	Uso de técnicas innovadoras y soluciones creativas para mejorar la funcionalidad y eficiencia del simulador.