

# Manual Técnico

Elaborado pelo grupo AP-08-05

Rodrigo Vila Nova, nº202200196 turma 8 de LEI

David Santana, nº202200215 turma 8 de LEI

Este manual técnico tem como objetivo completar o manual técnico elaborado durante a primeira fase, relatando mudanças nas classes estabelecidas previamente ou, caso implementadas, novas classes do projeto e respectivos métodos/atributos principais

Nesta fase, nenhuma classe foi criada. No entanto, houve alterações na classe Simulator, Main, Direction, Rack, Grid.

A classe Simulator foi apagada, pois já não existe necessidade de existir, uma vez que a simulação passa a ter uma interface gráfica, e a classe Main foi alterada para conseguir mostrar essa interface gráfica. As classes Direction, Rack e Grid receberam alterações que facilitam as suas funções.

## Classe Main

A classe Main é agora uma subclasse da classe Application. A classe Main será usada para iniciar a interface gráfica e mostrar a simulação desenvolvida na primeira fase. Também é utilizada para ler o ficheiros com as informações pertinentes.

### Métodos mais importantes:

**public static void main(String[] args)** - Apenas executa o método launch(args), que por sua vez executa o método start().

**public static void setup()** - Lê o ficheiro de dados para a simulação, criando os locais de entrega e recolha, os produtos, prateleiras, carros, paredes e dimensões do armazém.

**public static void tick()** - Executa a função tick() de todos os elementos que a possuem.

**public static void runLine(String line)** – Lê a linha do ficheiro dada por parâmetro e executa ações diferentes dependendo da “palavra-chave” dessa linha.

**public static void updateGridUI()** –Atualiza a grelha usada na interface gráfica, atribuindo cores a cada telha consoante o elemento colocado. A criação da grelha é realizada por uma função privada.

**public void start(Stage primaryStage)** – Utiliza as outras funções descritas para inicializar e mostrar a interface gráfica com a grelha, e inicializa as listas observáveis e o gráfico que também fazem parte dessa interface. A atualização destes outros componentes é feita por outras funções privadas.

## **Enumerado Direction**

O enumerado Direction mantém-se igual, com apenas uma função nova:

**public Direction getDirection(String string)** – Devolve o valor Direction correspondente à String dada por parâmetro.

## Classe Rack

A classe Rack mantém-se igual, com apenas duas função alterada:

**public Vector getLoadPosition()** – Agora, em vez de definir a zona de carregamento das prateleiras sempre em baixo, define a zona de carregamento consoante a orientação das prateleiras.

**public Direction getLoadDirection()** – Devolve a direção de carga das prateleiras.

## Classe Grid

A classe Grid mantém-se igual, com apenas uma função nova:

**public ArrayList<Vector> getPositions(Positionable positionable)** – Retorna todas as posições ocupadas por um objeto.

Para além destas alterações nestas classes, o algoritmo responsável pelo movimento autónomo dos veículos sofreu alterações, visando corrigir um problema em que o carro travava, recalculando o seu caminho indefinidamente e também melhorar o comportamento dos veículos em geral, permitindo, entre outras funções, estacionar quando não tem nenhum pedido pendente, libertando o espaço de movimento dos outros veículos.