 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Época Especial	Ano letivo 2019/2020	Data 09/09/2020
	Curso LEI/LSIRC	Hora 14:30	
	Unidade Curricular Paradigmas de Programação	Duração	

## Objetivos

Com a realização do trabalho prático, pretende-se que os alunos ponham em prática todos os conhecimentos adquiridos na utilização do paradigma de programação orientado a objetos (POO) e a sua implementação na linguagem de programação Java, demonstrando as suas apetências em:

- Conhecer e compreender os conceitos fundamentais associados ao paradigma da programação orientada a objetos;
- Conceber e implementar, para problemas concretos, soluções que tenham por base o paradigma da programação orientada a objetos.
- Reconhecer e compreender a semântica e a sintaxe da linguagem Java.
- Reutilizar, alterar e desenvolver código recorrendo à linguagem Java tendo em vista um determinado problema com regras semânticas específicas.

Considere ainda que:

- Não é permitida a utilização de API's/conceitos Java que não tenham sido alvo de lecionação no ano letivo corrente da unidade curricular Paradigmas de Programação. Os alunos que pretendam utilizar API's adicionais devem atempadamente pedir autorização a um dos docentes da unidade curricular.
- Não é permitida a utilização de coleções Java predefinidas ([Java Collections Framework](#)).
- Os recursos de suporte ao trabalho referenciados no enunciado, são de utilização obrigatória.

## Enunciado

A empresa *TransportationSolutions* tem como principal atividade a disponibilização de soluções dedicadas ao transporte de encomendas das mais diversas empresas da região.

Como resultado de um processo de modernização tecnológica, a empresa pretende desenvolver uma API em linguagem Java capaz de suportar os requisitos de uma ferramenta de suporte à gestão de entrega de encomendas. Em particular, a empresa pretende suportar a gestão de entregas realizada pelos seus veículos.

Desta forma, os colaboradores da *TransportationSolutions* podem mais facilmente gerir e visualizar as entregas de encomendas.


## Requisitos gerais

A solução a desenvolver deverá ser capaz de gerir a acomodação dos vários itens (**Item**) para entrega num plano de entregas (**Delivery**). O plano de entregas (**Delivery**) será carregado num veículo (**Vehicle**) de uma frota de veículos. Cada veículo terá obrigatoriamente um condutor (**Driver**), que irá entregar os itens (**Item**) num plano de entregas (**Delivery**) para o endereço (**Destination**) dos seus clientes (**Customer**).

O condutor deverá possuir uma licença de condução (**LicenseType**) compatível com os requisitos do veículo (**Vehicle**) que irá utilizar para a entrega. Este requisito garante que a licença necessária para operar o veículo (**Vehicle**) definida no programa é cumprida pelo condutor (**Driver**).

Cada item (**Item**) deve ter na sua definição uma referência, um estado e requisitos de transporte (**TransportationTypes**). Deve também ser garantido que os requisitos de transporte de cada item (**Item**) (perigoso e/ou frágil) são cumpridos pelos tipos de transporte (**TransportationTypes**) do veículo (**Vehicle**) utilizado.

Na especificação da entrega (**Delivery**), os itens (**Item**) são acomodados no veículo (**Vehicle**) que representa uma “caixa” (**Box**) de grandes dimensões especialmente orientada para acomodar um conjunto de (**Item**) (ver Figura 1).

 <small>ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO</small>	Tipo de Prova Época Especial	Ano letivo 2019/2020	Data 09/09/2020
	Curso LEI/LSIRC	Hora 14:30	
	Unidade Curricular Paradigmas de Programação	Duração	

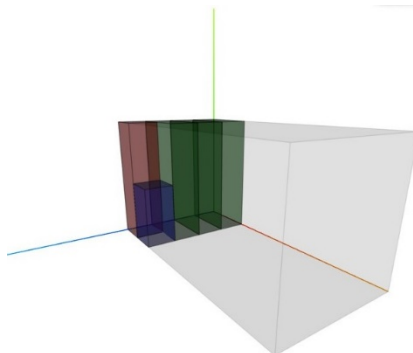


Figura 1. Exemplo de acomodação de itens num veículo

O objetivo passa por facilitar o processo de entrega (**Delivery**) de itens (**Item**), facilitando a gestão da frota e recursos humanos da empresa. Note que o posicionamento de itens (**Item**) nos veículos é realizado de forma manual (com coordenadas cartesianas especificadas pelo utilizador, ver Figura 2).

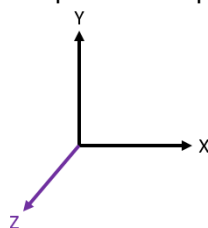


Figura 2. Eixos

Após a entrega de um item num plano de entregas (**Delivery**), condutor irá confirmar telefonicamente a entrega de cada item aos escritórios centrais que iram modificar o estado do item na aplicação.

Para gestão do negócio e todas as suas entidades, deverá existir uma plataforma de gestão (**Management**) onde deverá ser possível oferecer suporte à empresa TransportationSolutions para gerir:


- os seus veículos (**Vehicle**);
- os seus condutores (**Driver**);
- os seus itens (**Item**);
- os seus planos de entregas (**Delivery**);
- (outras operações que sejam úteis à empresa)

Para além de gerir as entidades de negócio, a plataforma de gestão deverá ser capaz de dar respostas sobre operações específicas necessárias como por exemplo ver a frota de veículos, confirmar a entrega de uma encomenda por parte do condutor, guardar um plano de encomendas ou restaurá-lo no programa.

O *workflow* básico para as encomendas e a sua entrega deve ter o seguinte racional:

1. Receber encomenda e registá-la;
2. Alocar a encomenda a um plano de entregas, com um veículo e condutor;
3. Iniciar as entregas;
4. Confirmar a entrega;
5. Na última encomenda, o plano de entregas deve ser fechado, o plano de entregas arquivado e o veículo libertado.

Deve ainda criar um menu textual para utilizar e testar a API.

 <div> ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO </div>	Tipo de Prova Época Especial	Ano letivo 2019/2020	Data 09/09/2020
	Curso LEI/LSIRC	Hora 14:30	
	Unidade Curricular Paradigmas de Programação	Duração	

## Descrição Técnica

Como suporte ao desenvolvimento da API, são disponibilizados um conjunto de recursos (*EE\_Resources*), de utilização obrigatória e que definem os contratos que permitem o desenvolvimento da API. Os conteúdos fornecidos são um complemento ao presente enunciado, contendo no JavaDoc informação mais específica sobre as particularidades de implementação de cada funcionalidade. A utilização dos contratos constitui um ponto de partida, cujos ficheiros não podem ser alterados. **Caso não utilize ou altere os recursos disponibilizados, todo o trabalho é invalidado.**

**No caso de existirem alterações à API, serão comunicadas por email institucional.**

Deverá realizar a implementação do código necessário para suportar cada uma das operações definidas nos contratos. Teste o mais exaustivamente possível o código que desenvolveu como resposta aos requisitos apresentados. Recorra a comentários JavaDoc e não só de modo a documentar, o mais exaustivamente possível, o código que desenvolveu.

A existência dos contratos **não deve ser impeditiva** para a implementação de novas funcionalidades e/ou novos métodos e/ou classes.

## Interface gráfica

Para complementar o processo de validação da API a desenvolver, é disponibilizado um componente que com base no código desenvolvido, apresenta uma *interface* gráfica para representar a acomodação de itens num veículo. Esta componente de visualização gráfica **já se encontra implementada**, devendo ser utilizada após a implementação e validação da API.


A visualização gráfica é exposta através da classe: `GUI` que disponibiliza um método `render` responsável por apresentar uma interface gráfica, utilizando o *browser* definido por defeito. No contexto da API, os ficheiros JSON<sup>1</sup> suportam o armazenamento dos dados de uma entrega. O JSON é um formato compacto para troca de dados entre sistemas, que suporta o armazenamento de dados no formato atributo-valor (ver <https://www.json.org/>).

O ficheiro JSON relacionado com **a exportação de uma entrega** considera é apresentado na Figura 3.

```
{
  "depth": 100,
  "color": "black",
  "length": 100,
  "height": 100
  "items": [
    {
      "depth": 20,
      "x": 0,
      "length": 20,
      "y": 0,
      "z": 0,
      "height": 20,
      "color": "red",
    },
    {
      "depth": 20,
      "x": 20,
      "length": 20,
      "y": 0,
      "z": 0,
      "height": 20,
      "color": "green",
    },
    (...)
  ]
}
```

Figura 3. Excerto de um documento JSON com dados de uma entrega

<sup>1</sup> JavaScript Object Notation

 <div> ESCOLA SUPERIOR DE TECNOLOGIA E GESTÃO </div>	Tipo de Prova Época Especial	Ano letivo 2019/2020	Data 09/09/2020
	Curso LEI/LSIRC	Hora 14:30	
	Unidade Curricular Paradigmas de Programação	Duração	

**Nota:** a cor apenas é incluída para efeitos de visualização, mas deverá ser uma das fornecidas na enumeração: **Color**.

## Funcionalidade para extensão da API

Deverá adicionalmente implementar exportadores (**Exporter**) que considerando a informação do domínio apresentado, produza ficheiros JSON que:

- Represente uma folha de acompanhamento de um entrega (**Delivery**) com indicação dos itens a entregar em cada destinatário;
- Resumo dos itens a entregar por tipo de transporte (**TransportationTypes**);
- Resumo do estado da frota de veículos;
- Resumo do estado dos condutores.

Pode utilizar uma biblioteca Java para facilitar a leitura/escrita do documento. A escolha da biblioteca de manipulação de documentos JSON a utilizar fica ao critério de cada aluno. A título de exemplo, é disponibilizado na plataforma moodle um excerto de código demonstrativo de utilização da biblioteca: *json-simple*<sup>2</sup>).

## Elaboração e entrega do trabalho

Este trabalho é realizado **individualmente**. O trabalho deve ser entregue até às **23:55** horas do dia **4 de setembro de 2020**, devendo a entrega ser feita através da página da unidade curricular de Paradigmas de Programação em <http://moodle.estg.ipp.pt>.

A defesa será realizada por turnos no dia do exame de época especial (a partir das 14:30). O horário exato para cada aluno será comunicado na plataforma moodle após a entrega do trabalho.

Considera-se por defesa satisfatória, quando o aluno demonstra que realizou o trabalho submetido e que **domina todos os conceitos de programação orientada a objetos aplicados na resolução do trabalho**. Tentativas de fraude, resultarão na avaliação do trabalho como: **Fraude Académica**.

## Formato da entrega

Os trabalhos entregues deverão evitar (se possível) utilizar caminhos absolutos ou endereços específicos, de modo a que possam ser facilmente utilizados em qualquer máquina. Para além disso, e no sentido de facilitar a receção dos vários trabalhos recebidos, estes deverão observar as seguintes regras:

O trabalho desenvolvido deverá ser entregue através do moodle, através da submissão de um ficheiro com o nome PP\_EE\_<nr\_do\_aluno>.zip, contendo:

- Os ficheiros criados incluindo o(s) projeto(s) do IDE Netbeans e uma pasta com a distribuição (jar) da solução proposta.
- Recorra a comentários JavaDoc, e não só, de modo a documentar, o mais exaustivamente possível, o código desenvolvido.
- Cada ficheiro de código entregue terá de possuir no início do mesmo um comentário com pelo menos a seguinte informação (com as adaptações óbvias para cada aluno):

```

/*
 * Nome: <Nome completo do aluno>
 * Número: <Número mecanográfico do aluno>
 */

```

Os alunos que não realizem a entrega do trabalho até à data/hora definida serão sujeitos a **penalização** ou a **invalidação do trabalho**.

<sup>2</sup> <https://code.google.com/archive/p/json-simple/>