

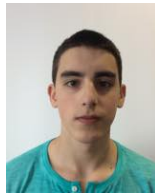
# Desenvolvimento de Sistemas de Software

## Mestrado Integrado em Engenharia Informática

Luís Capa  
A81960



Moisés Antunes  
A82263



Pedro Capa  
A83170



10 de Novembro de 2018

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Objetivos</b>	<b>3</b>
<b>3</b>	<b>Trabalho realizado</b>	<b>4</b>
<b>4</b>	<b>Análise crítica do trabalho realizado</b>	<b>5</b>

# Capítulo 1

## Introdução

Hoje em dia, os clientes das grandes marcas de automóveis têm a possibilidade de personalizar o carro da maneira como quiserem. Para tal, os clientes podem escolher peça a peça, escolher um pacote ou escolher configuração ótima em relação ao orçamento. O cliente não pode escolher todas as peças que quer, visto que há peças que são incompatíveis e outras que são obrigatórias. Os componentes na fábrica têm um stock, no caso de chegarem novas peças é necessário informar o sistema que chegaram novas peças e os carros que estão à espera dessas peças voltam para a fila de produção.

## Capítulo 2

# Objetivos

Esta unidade curricular tem o objetivo de mudar a forma como nós programamos, pois até agora sempre que recebíamos um projeto começava-se a escrever código. Para isso usou-se a linguagem UML para preparar a escrita do código de forma a cometer menos erros quer na criação de classes quer no objetivo do projeto em si. Este relatório tem o objetivo de clarificar o trabalho realizado no geral, bem como as suas partes.

## Capítulo 3

# Trabalho realizado

Para este trabalho foram criados alguns modelos UML como o modelo de domínio, UseCase e os diagramas de estado. No "Visual Paradigm" foi criado um modelo de domínio que mostra as principais relações entre entidades criadas para este problema. No mesmo programa foi criado o modelo de use cases. Este modelo mostra as principais interações entre os atores do sistema com o próprio sistema. Nesta aplicação foram consideradas duas entidades que teriam diferentes estados, o carro e a peça. Para descrever as diferentes fases destas entidades foi criado para cada uma um modelo de estado. O modelo de interface foi realizado no "NetBeans", uma vez que será essa a ferramenta usada para a implementação do projeto e é mais fácil de implementar a interface gráfica, porque tem um visualizador de classes interfaces que gera automaticamente o código dos componentes do swing, api de interfaces usada. A interface foi modelada de acordo com os Use Cases do projeto, uma vez que este indica que opções cada ator realiza no sistema.

## Capítulo 4

# Análise crítica do trabalho realizado

Vários “Use Cases” foram criados para garantir acesso a um sistema que permita tanto aos clientes como funcionários proceder à compra, venda e outras operações relativas aos automóveis da empresa. O nosso objetivo principal durante a criação destes “Use Cases” foi garantir que haveria várias opções de customização de carros e que os utilizadores do sistema teriam acesso aos dados que lhes diziam respeito sem, no entanto, exagerar na quantidade de opções disponíveis, que deixaria o sistema demasiado complicado de usar, especialmente para clientes não acostumados com algo do género. Nos modelos de estado consideramos que os principais métodos e interações com as entidades foram considerados, apesar de eventualmente haver algum em falta. Para o modelo de domínio estamos convictos que os principais conceitos e relações entre entidades foram inseridas neste modelo. No final, acreditamos que o trabalho que desenvolvemos até agora demonstra bem esse equilíbrio entre personalização e facilidade de uso que procuramos, que são apenas detalhes, mas é neles que se destaca uns projetos dos outros. Também achamos que o essencial foi feito apropriadamente e que não há falta de alguma funcionalidade que seja absolutamente necessário, sendo que consideramos o trabalho até agora tanto funcional como cómodo para o utilizador.