

Ficha de trabalho Nº 3	
Disciplina	Modelação de Dados em Engenharia
Ano Lectivo	2020/2021
Objectivo	Modelar conhecimento usando Frames e linguagem gráfica UML
Aulas	2 aulas x 3 horas + 6 horas extra
Data de Entrega	2021/06/21
<p>Objetivos concretos:</p> <ol style="list-style-type: none">1. Modelação baseada em frames<ol style="list-style-type: none">a. Entender o conceito de modelação com frames e relevância num contexto de engenharia Electrotécnica.b. Representação de modelos com a linguagem Golog/Prolog.c. Modelar a informação do problema proposto usando conceito de frames, slots, relações e mecanismos de herança, métodos, demons.2. Modelação UML<ol style="list-style-type: none">a. Modelar os tipos de entidades que existem no sistema e como estes se relacionam entre si através de um Diagrama de Classes.	

Objetivos do Trabalho

Este trabalho destina-se a proporcionar aos alunos um contacto com os conceitos básicos de programação estruturada em *Frames*, tendo como suporte a linguagem Prolog e a biblioteca GOLOG, e também modelação UML tendo em vista a sua posterior utilização em atividades enquadradas no seio da Engenharia Eletrotécnica e de Computadores.

Deste modo o trabalho é dividido em 2 partes:

- I - Modelação baseada em frames,
- II - Modelação baseada em UML.

Parte I - Modelação baseada em frames

O estudo e a implementação do problema proposto nesta parte do trabalho servirão para fazer uma abordagem aos seguintes conceitos:

- Estudo do problema apresentado.
- Estruturação de conhecimento em *Frames*.
- Relações e mecanismos de herança.
- Utilização de programação lógica.
- Implementação de métodos e demónios de forma a implementar as funcionalidades pretendidas no problema.

O Problema

No seguimento do trabalho desenvolvido na Ficha de Trabalho 2, pretende-se desenvolver o sistema *FactoryManagement* para ajudar na gestão de produção e vendas de uma fábrica.

Neste contexto, consideramos que uma **fábrica produz vários produtos** (por exemplo, uma fábrica de instrumentos musicais produz pianos, órgãos, guitarras, etc.). Podem ainda existir **produtos de diferentes tipos, por exemplo, podemos ter guitarras clássicas ou eléctricas, pianos de cauda ou verticais, etc.** Sabe-se também que os **produtos são compostos por diferentes peças** (por exemplo, uma guitarra é composta por uma cabeça, corpo, cordas, etc.) e que **poderão haver peças que são partes constituintes de vários produtos** (por exemplo, cordas são partes constituintes de violoncelos, violinos, etc.). Assume-se que toda a logística associada à chegada das peças à fábrica é tratada pelo

sistema desenvolvido no trabalho 2, e que é possível saber a quantidade de peças existentes pois estão guardadas num local específico do armazém da fábrica.

A fábrica é caracterizada pelo seu nome, cidade, capacidade máxima de armazenamento de produtos, número total de produtos em stock, etc. Um produto, por sua vez, é representado pelo nome, referência, data de fabrico, preço unitário, quantidade em stock, etc. Uma peça é caracterizada pelo nome, referência, quantidade existente, etc.

Imagine então que foi contratado para implementar o sistema *FactoryManagement* efectuando uma modelação baseada em *Frames*. O sistema deverá permitir que o gestor da fábrica crie, visualize, altere e apague produtos e peças. Deverá também permitir a atribuição de peças a um produto tendo em consideração o número de peças que são necessárias para esse produto (por exemplo, um piano necessita de 88 teclas, 2 pedais, etc.) e também visualizar as peças de um determinado produto. O sistema deverá também permitir que um vendedor possa visualizar todos os produtos produzidos pela fábrica e suas características, por ordem de data de fabrico (crescente e/ou decrescente) e efectuar encomendas de produtos. Assim que uma encomenda é efectuada terá de ser verificada a existência do(s) produto(s) em stock e, caso seja possível avançar com a encomenda, assumir esses produtos como “vendidos” actualizando os respectivos stocks. Caso contrário o sistema terá de avisar o vendedor que não é possível efectuar a encomenda e despoletar um alerta onde se regista a data da ocorrência e os produtos que estão em ruptura de stock. O sistema deve também estabelecer um valor mínimo para a quantidade em stock dos produtos produzidos pela fábrica e cada vez que esse stock baixar esse valor, deverá accionar o processo de fabrico de mais produtos. Por exemplo, o sistema estabelece que o stock de produtos não pode ser inferior a 5, e neste momento existem em stock 12 pianos de cauda. Imaginando que é efectuada uma encomenda de 10 pianos de cauda, após a “venda” o stock passa a 2, logo o sistema deverá accionar o processo de fabrico/montagem de mais pianos de cauda (valor à consignação do implementador do sistema). O processo de fabrico passa por inicialmente identificar quais as peças que compõem o produto, e se existem as quantidades necessárias para o seu fabrico. Caso existam, o(s) produto(s) são então produzidos e actualizados os respectivos stocks, caso contrário deverá ser enviada uma mensagem de aviso.

As funcionalidades do sistema *FactoryManagement* referidas requerem a construção de um menu/interface adequado para a interação com o utilizador.

Este problema deverá ser modelado através de uma estrutura em *Frames*, usando frames genéricas, concretas, relações e mecanismos de herança, as mais adequadas possível. Note-se que algumas das funcionalidades necessárias serão melhor solucionadas usando métodos Golog; outras são melhor solucionadas através de demónios.

Modelação/implementação

Durante a modelação deste problema é necessário considerar as seguintes fases do projecto:

1. Modelar este problema utilizando frames GOLOG.
2. Identificar e implementar os atributos/slots e regras/predicados necessários para resolver o problema pedido.
3. Implementar os métodos e demónios necessários.
4. Desenvolver um menu para simular o funcionamento do sistema.

Requisitos funcionais

A resposta a este problema implica a satisfação dos seguintes requisitos funcionais:

Req. Funcionais	Descrição
FR0	Diagrama de modelação em frames do problema (especificar frames, relações, instâncias, métodos, demónios, etc.)
FR1	Criar, visualizar, alterar e apagar produtos
FR2	Criar, visualizar, alterar e apagar peças
FR3	Atribuir peças (e respetivas quantidades) que constituem um produto
FR4	Listar peças de um dado produto
FR5	Listar produtos e suas características por ordem (crescente e/ou decrescente) de data de fabrico
FR6	Encomendar produto(s)
FR7	Visualizar mensagens de alarme
FR8	Proponha outro requisito relevante

Parte II - Modelação baseada em UML

FR 9 - Elabore o Diagrama de Classes que suporta o problema apresentado anteriormente.

Elementos de Reflexão

Esta parte consiste em considerações mais reflexivas que poderão servir como “inspiração” para o preenchimento do formulário Moodle. No final do trabalho, as seguintes questões deverão poder ser respondidas:

- Quais as vantagens/desvantagens da modelação por Frames, em relação à modelação efetuadas nos trabalhos práticos anteriores.
- A utilidade da modelação em Golog no contexto da Engenharia.
- Como funcionam os demónios?
- Quais as vantagens/desvantagens de modelação usando linguagem UML?

Este trabalho segue o seguinte plano:

1^a aula: Apresentação do trabalho e das ferramentas a utilizar. Revisão dos conceitos relacionados com a programação em Golog. Implementação das *Frames* com os respetivos atributos e métodos.

2^a aula: Resumo dos conceitos relacionados com programação reactiva. Criação dos demónios e ambiente de teste da estrutura resultante.

Entrega (Moodle)

Os alunos devem submeter, através do Moodle:

- ➔ O código fonte (todos os ficheiros relativos à solução do problema e implementação)
- ➔ Um vídeo de demonstração do trabalho com cerca de 5/8 minutos
- ➔ Questionário Moodle

Critérios de avaliação

É fornecida, a título de suporte, a tabela de cotações do projeto. A cotação de cada item será ponderada pela qualidade da solução/código obtida.

Itens		Cotação/ Penalização
Funcionalidades	FR0 - Diagrama de modelação em frames do problema (especificar frames, relações, instâncias, métodos, demónios, etc.)	2
	FR1 - Criar, visualizar, alterar e apagar produtos	1
	FR2 - Criar, visualizar, alterar e apagar peças	1
	FR3 - Atribuir peças e respetivas quantidades que constituem um produto	2
	FR4 - Listar peças de um dado produto	1,5
	FR5 - Listar produtos e suas características por ordem (crescente e/ou decrescente) de data de fabrico	2
	FR6 - Encomendar produto(s)	2
	FR7 - Visualizar mensagens de alarme	1,5
	FR8 - Proponha outro requisito relevante	2
	FR9 - Diagrama de classes UML	2
Vídeo		2
Questionário		1
TOTAL		20
Não entrega do vídeo e do questionário (* OBRIGATÓRIAS)		-20
Cada 5 dias de atraso na entrega (cumulativo)		-1

Docentes Teórica:

Luis Camarinha-Matos, cam@uninova.pt

Docentes Prática:

Ana Inês Oliveira, aio@uninova.pt

André Rocha, andre.rocha@uninova.pt

Filipa Ferrada, faf@uninova.pt

João Rosas, jrosas@uninova.pt