

**Trabalho Prático**  
**Programação Orientada por Objetos**

**NAUTILUS**  
sistema de gestão de frota náutica

# Índice

<b>1</b>	<b>Introdução .....</b>	<b>3</b>
<b>2</b>	<b>O NAUTILUS.....</b>	<b>3</b>
2.1	Marinheiros .....	3
2.2	Embarcações.....	3
2.3	Porto .....	4
<b>3</b>	<b>Requisitos funcionais .....</b>	<b>4</b>
3.1	Marinheiros .....	4
3.2	Embarcações.....	4
3.2.1	Barcos de Patrulha .....	4
3.2.2	Lanchas Rápidas .....	5
3.2.3	Navios de Suporte .....	5
3.2.4	Outras Embarcações (Opcional) .....	5
3.3	Porto .....	5
3.4	Outras funcionalidades e restrições .....	5
3.5	Modos de Funcionamento: .....	5
<b>4</b>	<b>Regras de Implementação e Avaliação .....</b>	<b>6</b>
4.1	Constituição de grupos .....	6
4.2	Regras de entrega do projeto .....	6
4.3	Regras gerais de implementação.....	6
4.4	Fases do projeto e documentos a entregar.....	6
4.5	Regras de avaliação do Projeto.....	6
4.5.1	Cálculo da Nota .....	7

## 1 Introdução

O objetivo deste projeto é desenvolver o sistema NAUTILUS, para permitir ao utilizador a gestão de uma frota náutica, para um porto operacional marítimo. Na figura 1 pode observar um esquema representativo das zonas de atuação da frota do porto.

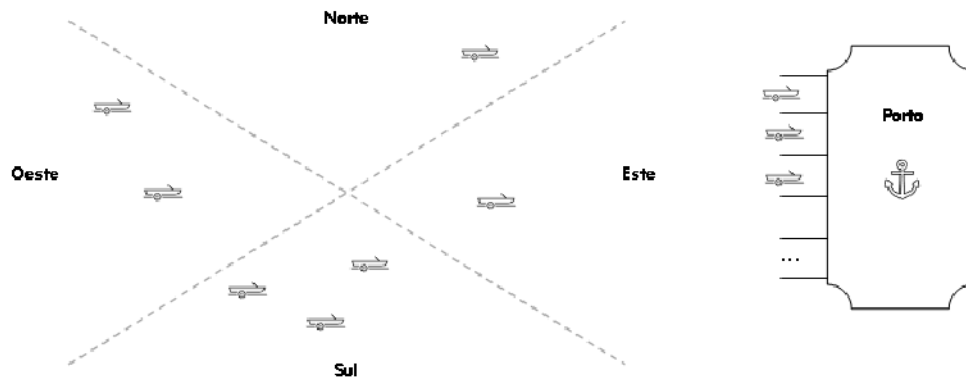


Figura 1 Representação ilustrativa do porto, zonas de atuação e embarcações.

Deve implementar o projeto seguindo os conceitos de Programação Orientada por Objetos (POO) e utilizando a linguagem JAVA.

## 2 O NAUTILUS

O NAUTILUS é um sistema com uma interface textual (i.e., um conjunto de menus/submenus em consola), através da qual será possível realizar todas as suas funções. O sistema deverá permitir a gestão de marinheiros e embarcações de um determinado porto, tendo em conta as seguintes características:

### 2.1 Marinheiros

Cada marinheiro tem um id, nome, data de nascimento e patente (Oficial, Sargento ou Praça).

Os marinheiros podem ser alocados numa dada embarcação.

### 2.2 Embarcações

Todas as embarcações devem ter um id, nome (trata-se da denominação oficial da embarcação, por exemplo: Titanic; Bismarck; Black Pearl; Santa Maria etc.), marca, modelo, data de fabrico e motor(es) – com determinado combustível, cilindrada [cm3] e potência [cv]. Cada embarcação deve ainda saber quantas embarcações existem no total.

Cada embarcação será associada a uma zona (zona norte, sul, este ou oeste).

Existem 3 tipos de embarcação:

- Barcos de Patrulha: equipadas com 1 motor, 2 a 4 tripulantes (obrigatório 1 e 1 só oficial), holofote e radar.

- Lanchas Rápidas: equipadas com 2 a 4 motores, 2 a 4 tripulantes (obrigatório 1 sargento) e holofote.
- Navios de Suporte: 2 motores de potência superior a 25000cv, capacidade de carga [kg], 4 a 10 tripulantes (obrigatório 1 oficial e 1 sargento), número de camas hospitalares para receber pacientes a necessitar de cuidados médicos, holofote, botes salva-vidas e radar.

## 2.3 Porto

O porto tem um nome, um conjunto de embarcações associadas e deve manter registo do número total de missões que já ordenou.

## 3 Requisitos funcionais

### 3.1 Marinheiros

O id é um número sequencial e atribuído automaticamente pelo sistema. Um marinheiro é igual a outro se tiver o mesmo id.

Não pode haver Oficiais com menos de 35 anos.

Em todas as apresentações de informação, os marinheiros deverão ser mostrados por ordem crescente de nome.

Deve ser possível listar os marinheiros por ordem crescente de id, bem como por ordem decrescente de data de nascimento (mostrando a respetiva idade).

### 3.2 Embarcações

O id é um número sequencial e atribuído automaticamente pelo sistema. Uma embarcação é considerada igual a outra se tiver o mesmo id.

Em todas as apresentações de informação, as embarcações de uma determinada zona deverão ser mostradas por ordem crescente do seu id.

Deve ser possível listar as embarcações por ordem crescente de marca, bem como por ordem decrescente de ano de fabrico, com indicação do número de anos e meses da embarcação desde o seu fabrico.

O sistema deverá garantir que não são criadas embarcações com o mesmo nome (se isso acontecer, o programa deverá produzir uma exceção). Deverá também garantir que não existe uma embarcação com um nome nulo (se tal acontecer, o programa deverá produzir uma exceção).

As embarcações são associadas a 1 zona. É fundamental garantir que não se criam associações entre uma mesma embarcação e duas zonas diferentes.

As embarcações podem ser ativadas para operações em determinada zona, como descrito nos próximos pontos. Para serem ativadas têm que ter a tripulação mínima alocada à respetiva embarcação.

#### 3.2.1 Barcos de Patrulha

Os barcos de patrulha são ativados para procura e salvamento. Podem ser ativados vários barcos de patrulha para a mesma zona. Enquanto estão ativos ligam o holofote e radar. Quando 1 dos barcos de

patrulha termina o salvamento, volta ao porto e fica atracado. O radar apenas consegue detetar e ver os detalhes das embarcações que estão na mesma zona (para detetar, o radar tem de estar ligado)

### **3.2.2 Lanchas Rápidas**

As lanchas rápidas são ativadas para perseguição e captura. Podem ser ativadas várias lanchas rápidas para a mesma zona. Enquanto estão ativas ligam o holofote. Quando 1 das lanchas termina a captura, todas as lanchas rápidas na mesma zona deixam de estar em missão, voltam ao porto e ficam atracadas.

### **3.2.3 Navios de Suporte**

Os navios de suporte são ativados para dar apoio a determinada zona. Enquanto estão ativos ligam o holofote e radar. O radar apenas consegue detetar e ver os detalhes das embarcações que estão na mesma zona (para detetar, o radar tem de estar ligado).

### **3.2.4 Outras Embarcações (Opcional)**

(Opcional) Se pretender, poderá considerar e implementar outras embarcações.

## **3.3 Porto**

O porto pode dar ordens de ativação de embarcações para missões nas diferentes zonas (norte, sul, este e oeste). As embarcações que não estão em missão, estão atracadas no porto.

O porto tem um radar que consegue detetar as embarcações que estão atracadas no porto, bem como aquelas que estão na zona adjacente, conseguindo aceder a todos os detalhes das mesmas (para o radar poder detetar tem de estar ligado).

## **3.4 Outras funcionalidades e restrições**

Não deve ser possível registar duas vezes uma mesma embarcação ou marinheiro.

Uma mesma embarcação não pode ser registada em duas zonas diferentes.

Deverá ser possível gravar para ficheiro toda a informação do sistema, de forma a recuperar o estado entre execuções do mesmo.

Deverá ser possível exportar uma lista de embarcações e respetivos dados para um ficheiro de texto, devidamente formatado.

## **3.5 Modos de Funcionamento:**

O menu deverá permitir o funcionamento em dois modos: modo de manutenção e modo de utilização. No modo de manutenção serão disponibilizadas operações relacionadas com configurações do sistema: i.e., operações de criação, edição ou remoção. As restantes operações, relacionadas com o uso diário do sistema, serão disponibilizadas no modo de Utilização.

Deverá existir uma função RECOVER, que colocará todos as entidades do sistema com os últimos valores gravados em ficheiro.

## 4 Regras de Implementação e Avaliação

### 4.1 Constituição de grupos

Cada projeto deverá ser elaborado respeitando os grupos definidos em aula. Possíveis exceções devem ser discutidas com o docente.

### 4.2 Regras de entrega do projeto

- Cada fase do projeto deve ser entregue até à data estipulada, por **via exclusivamente eletrónica, utilizando o TEAMS**.
- Todos os ficheiros que compõem cada fase deverão ser compactados num único ficheiro em **formato ZIP**, denominado GrupoK\_XXXXX\_YYYYY\_ZZZZZ.zip, em que K é o número do grupo e X, Y e Z são os números dos alunos que constituem o grupo indicado.

### 4.3 Regras gerais de implementação

- O programa deve ser desenvolvido utilizando a linguagem Java em modo de consola e colocando em prática os conceitos fundamentais do paradigma de programação orientada por objetos, nomeadamente: encapsulamento de dados, classe e objetos, composição e herança de classes, classes abstratas, interfaces, polimorfismo e serialização.
- **Sugestão:** Todas as classes de coleção podem ser implementadas como heranças de classes de coleção parametrizáveis (e.g., `public class ColecaoEmbarcacoes extends xxxxx`).

Em relação às regras de codificação siga as convenções adotadas normalmente para a linguagem Java:

- A notação camelCase para o nome das variáveis locais e identificadores de atributos e métodos;
- A notação PascalCase para os nomes das classes e interfaces;
- Utilização de maiúsculas para os nomes das constantes e dos valores enumerados;
- Não utilize abreviaturas nem o símbolo '\_' nos identificadores (exceto nas constantes)
- O projeto deve cumprir tudo o que é pedido no seu enunciado, **sendo deixado ao critério do programador qualquer pormenor de implementação que não seja referido** no mesmo e que deve ser devidamente documentado.
- O desenvolvimento dos menus da interface com o utilizador fica ao critério do programador, sendo premiadas a facilidade de utilização.

### 4.4 Fases do projeto e documentos a entregar

O projeto será desenvolvido em fases, que serão detalhadas nas aulas, ao longo do semestre.

A **defesa/discussão obrigatória** do projeto terá lugar: no dia xx de xx.

### 4.5 Regras de avaliação do Projeto

- A classificação de cada fase do projeto terá em conta as especificidades dos materiais exigidos e focará a avaliação da qualidade da programação (fatores de qualidade do software), a estrutura do código criado segundo os princípios da programação orientada por objetos, tendo em conta conceitos como: a coesão de classes e métodos, o grau de acoplamento entre classes e o desenho de classes orientado pela responsabilidade, bem como a utilização/conhecimento da linguagem Java.
- Serão premiadas **a facilidade de utilização, a apresentação, a imaginação e a criatividade**.

- Na discussão final de projeto, será apurada a capacidade de o aluno produzir os materiais apresentados. Nos casos em que essa capacidade não for demonstrada, a nota atribuída será zero valores.
- Todos os projetos serão submetidos a um sistema de **detecção de cópias automático e de inteligência artificial**. Os projetos que forem assinalados como suspeitos, caso se confirme a fraude, **serão anulados**.

#### **4.5.1 Cálculo da Nota**

A nota final do projeto será o resultado da média das avaliações realizadas durante as aulas tendo em conta o trabalho individual desenvolvido por cada aluno, bem como a avaliação da discussão de defesa final dos projetos entregues.