

# Programação Orientada a Objetos

## Relatório Final



Trabalho elaborado por:

- ➔ Luís Lucas Prioste 2019142028 P3
- ➔ Tiago Conceição 2016015546 P5

# Organização do código apresentado

## 1. Quais foram as classes consideradas?

As classes consideradas e principais funcionalidades foram:

- Território
  - Ligar-se e desligar Império
  - Verificar se é Continente ou Ilha
  - Mudar o Ouro e os Produtos produzidos pelo Território
- TerritórioInicial
  - Herda funções do Território
- Continente
  - Verifica se é Continente ou Ilha
- Planicie, Montanha, Fortaleza, Mina, Duna, Castelo
  - Classes derivadas do Continente
- Ilhas
  - Herda funções do Território
- RefugioPiratas, Pescaria
  - Classes derivadas da Ilhas
- Mundo
  - Adicionar territórios
- Império
  - Conquista Território
  - Comprar Tecnologia
- Interface
  - Iniciar a Interface
- Construtor
  - Ler comandos
  - Gerir a sequência do jogo
  - Gerir turno e ano
- Eventos
  - Ação do Evento
- RecursoAbandonado, Invasao, AliancaDiplomatica, SemEvento
  - Herda funções do Eventos
- Tecnologia
  - Ação da tecnologia
- DronesMilitares, MisseisTeleguiados, DefesasTerritoriais, BolsaDeValores, BancoCentral
  - Herda funções da Tecnologia
- Loja
  - Verificar se é possível comprar Tecnologia
- Grava
  - Adicionar gravação

- Remover gravação
- Carregar gravação

Estas classes foram testadas na função interface.

- 2. Quais os conceitos/classe que identificou ao ler o enunciado?**
  - Vários tipos de Território
  - Mundo
  - Império
  - Nome do Território
  - Tipo do Território (Continente ou Ilha ou Território Inicial)
  - Resistência
  - Criação de Produtos
  - Criação de Ouro
  - Pontos de Vitória
  - Conquistado (saber se o Território está conquistado ou não)
  - Turno
  - Ano
  - Tecnologias
  - Eventos
  - Gravações de estados do jogo
  - Construtor que irá saber o que fazer com cada comando em diferentes fases do jogo
  - Interface
- 3. Relativamente a duas das principais classes da aplicação, identifique em que classes ou partes do programa são criados, armazenados e destruídos os seus objetos.**

Territórios: os objetos desta classe são criados, armazenados e destruídos na classe Mundo.

Império: os objetos desta classe são criados, armazenados e destruídos na classe Interface.
- 4. Indique um exemplo de uma responsabilidade atribuída a uma classe que esteja de acordo com a orientação dada acerca de Encapsulamento.**

A responsabilidade “comprar tecnologia” está atribuído à classe Loja porque tem o acesso a todas as tecnologias e tem acesso ao custo de cada tecnologia e depois verifica se o império tem ouro suficiente.
- 5. De entre as classes que fez, escolha duas e justifique por que considera que são classes com objetivos focado, coeso e sem dispersão.**

Classe Território: tem os dados e as características do território.

Classe Mundo: tem um vetor que armazena todos os territórios.
- 6. Relativamente à aplicação entregue, quais as classes que considera com responsabilidades de interface com o utilizador e quais as que representam a lógica?**

Responsabilidade de interface: Interface

Responsabilidade da lógica da aplicação: Construtor

- 7. Identifique o primeiro objeto para além da camada de interação com o utilizador que recebe e coordena uma funcionalidade de natureza lógica?**  
As ordens vindas da camada de interação com o utilizador são recebidas e processadas por um objeto da classe Construtor.
- 8. A classe que representa a envolvente de toda a lógica executa em pormenor muitas funcionalidades, ou delega noutras classes? Indique um exemplo em que esta classe delega uma funcionalidade noutra classe.**

A classe Construtor representa a envolvente de toda a lógica. Para executar os comandos enviados pelo utilizador, delega a função *acaoComando* que irá pedir a classe específica para devolver a resposta.

- 9. Dê um exemplo de uma funcionalidade que varia conforme o tipo do objeto que a invoca. Indique em que classes e métodos está implementada esta funcionalidade. (Não é necessário responder a esta pergunta na meta 1).**  
O território irá ter diferentes territórios, onde esses territórios iram herdar a classe território. Cada território poderá mudar os valores de produção tanto de ouro como de produto ao longo do jogo. Para isso temos uma função virtual.

```
virtual bool mudaProdOuro(int turno, int ano) = 0;
```

A função é implementada nas classes concretas derivadas do Territorio

```
virtual bool mudaProdOuro(int turno, int ano) override;
```

- 10. Apresente as principais classes da aplicação através da seguinte informação:**

**Classe:** Território

**Responsabilidades:**

- Obter o nome do Território
- Obter a Resistência do Território
- Obter Produtos do Território
- Obter Ouro do Território
- Obter Pontos de Vitória do Território
- Obter estado do Território (se conquistado ou não)

**Colaborações:** Mundo, Imperio

**Classe:** Império

**Responsabilidades:**

- Listar todos os territórios conquistados
- Conquistar um território
- Atualizar pontos de vitória
- Comprar tecnologia
- Recolher produtos e ouro

**Colaborações:** Território, Loja

**Classe:** Construtor

**Responsabilidades:**

- Ler comando
- Tratar as diferentes fases
- Verificar o turno e o ano
- Tratar a resposta para o Utilizador

**Colaborações:** Interface, Imperio, Mundo, Loja, Grava, Eventos

## Funcionalidades implementadas

Na tabela abaixo está assinalado o grau de realização dos requisitos pedidos para esta meta.

Componente do trabalho	Realizado	Realizado parcialmente	Não realizado
Configuração do Mundo	X		
Conquista de Territórios	X		
Visualização dos dados do jogo	X		
Funcionalidade/comandos disponibilizados ao jogador	X		
Dados do jogo	X		
Eventos	X		
Compra de unidades militares e tecnologia	X		
Recolha de produtos e ouro	X		
Conquistar/Passar	X		
Criação do Mundo	X		
Gravação	X		