

Licenciatura em Engenharia Informática

# Índice

[Índice 2](#_Toc93280263)

[Introdução 3](#_Toc93280264)

[Implementação 4](#_Toc93280265)

[Estruturas de Dados 4](#_Toc93280266)

[Vetor de Zonas 4](#_Toc93280267)

[Vetor de Ilhas 5](#_Toc93280268)

[Vetor de Trabalhadores 5](#_Toc93280269)

[Classes 6](#_Toc93280270)

[Ilha 6](#_Toc93280271)

[Zona 14](#_Toc93280272)

[Edificio 21](#_Toc93280273)

[Trabalhador 24](#_Toc93280274)

[Comandos 27](#_Toc93280275)

[cons 27](#_Toc93280276)

[cont 27](#_Toc93280277)

[liga/des 27](#_Toc93280278)

[list 28](#_Toc93280279)

[config 28](#_Toc93280280)

[move 28](#_Toc93280281)

[vende 29](#_Toc93280282)

[debcash 29](#_Toc93280283)

[debed 29](#_Toc93280284)

[debkill 30](#_Toc93280285)

[save 30](#_Toc93280286)

[apaga 30](#_Toc93280287)

[load 31](#_Toc93280288)

[next 31](#_Toc93280289)

[upgrade 31](#_Toc93280290)

[Funções de Suporte 32](#_Toc93280291)

[isNumber 32](#_Toc93280292)

[validaPosicao 32](#_Toc93280293)

[mostraASCII 33](#_Toc93280294)

[menu 33](#_Toc93280295)

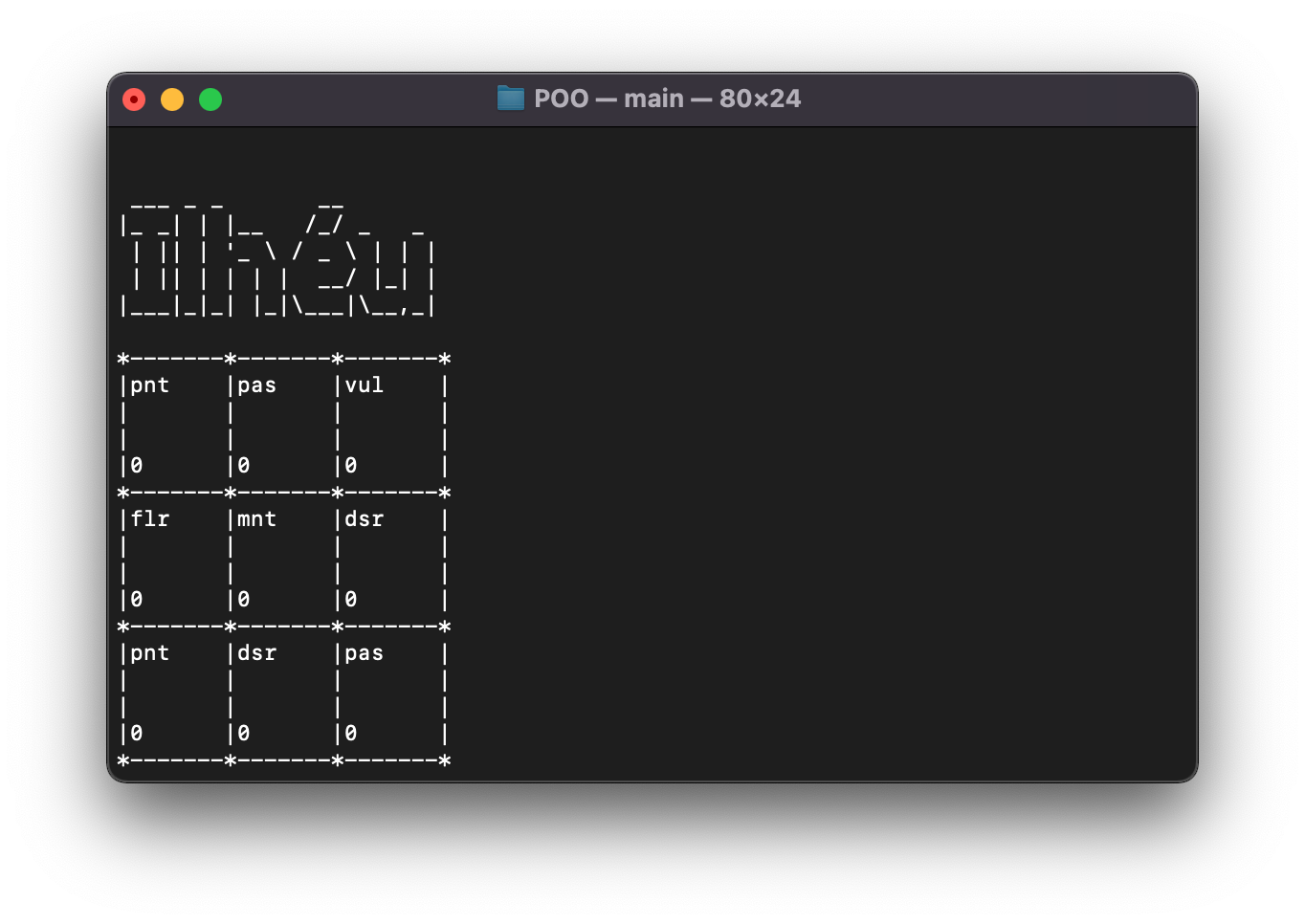
[Conclusão 34](#_Toc93280296)

# Introdução

O trabalho prático de Programação Orientada a Objetos consiste na criação de um jogo/simulação na linguagem de programação C++ em modo de consola.

O jogador escolhe o tamanho da ilha e vai industrializando a mesma colocando edifícios e trabalhadores nas várias zonas da ilha.

Existem várias maneiras do jogador interagir com o jogo de forma a ganhar dinheiro para mais tarde investir noutras zonas e existem também várias jogadas que este pode fazer, tais como: mover um trabalhador, vender edifícios, listar a informação atual do jogo de forma detalhada, etc...



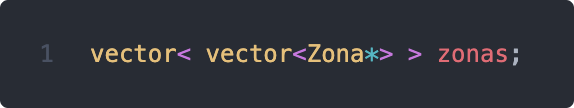
# Implementação

## Estruturas de Dados

### Vetor de Zonas

A estrutura de dados que foi utilizada para guardar os objetos da classe Zona foi um vetor bidimensional. Este vetor pertence à classe Ilha.

Para criar um vetor bidimensional utilizámos um vetor dentro de um vetor para podermos aceder aos objetos utilizando a notação vetor[i][j].



Foi necessário criar um vetor temporário para armazenar objetos da mesma coluna nesse vetor para posteriormente adicionarmos o vetor temporário ao vetor principal.

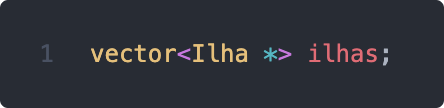
Uma imagem com texto

Descrição gerada automaticamente

A função responsável por criar este vetor é a função initIlha, pertencente à classe Ilha e esta é executada quando um objeto da classe Ilha é criado.

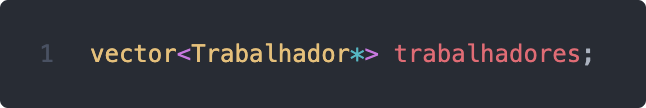
### Vetor de Ilhas

O vetor das ilhas é um vetor que pertence à classe Ilha e que serve para guardar os vários estados da ilha sempre que o jogador efetua o comando save.



### Vetor de Trabalhadores

Para guardar os trabalhadores de uma zona é necessário termos um vetor do tipo Trabalhador que possa armazenar os vários objetos correspondentes aos trabalhadores.



## Classes

As várias classes representam um tipo de “entidade” no jogo. Os objetos das classes contêm múltiplas informações sobre essa “entidade” e funções que nos permitem alterar ou obter várias informações à cerca de um objeto.

### Ilha

A classe Ilha é a classe principal do jogo. Representa uma ilha com várias zonas e também as informações e recursos de um jogador.

* **Variáveis:**
* linhas, colunas: guarda o tamanho dessa ilha
* dia: guarda o dia atual
* saveName: guarda o nome do save da ilha
* saldo: guarda o saldo do jogador
* nrVigasMadeira, nrFerro, nrBarrasDeAco, nrCarvao, nrMadeira, nrEletricidade: guarda o número de recursos do jogador
* vigasMadeiraUsar, flag: variáveis de suporte
* custoMinaf, custoMinac, ...: custo dos vários edifícios
* validaAuxMinaf, validaAuxMinac, validaAuxCentral: mais variáveis auxiliares
* zonas: guarda as zonas da ilha
* ilhas: guarda as várias ilhas sempre que o jogador faz save
* **Funções:**
* Ilha: construtor da classe Ilha que trata de inicializar a ilha ou no caso de ser passado uma ilha como parâmetro substitui todos os dados da ilha com informação do save
* initIlha: pede ao jogador o tamanho da ilha e adiciona objetos da classe Zona ao vetor zonas de maneira aleatória
* mostraIlha: mostra a ilha visualmente
* addSaldo: adiciona ou remove saldo à conta do jogador
* getSaldo: obtém o saldo do jogador
* getInfoZona: obtém a informação textual de uma zona ou de todas as zonas caso não seja passado nenhum argumento à função
* getSaveName: obtém o nome do save da ilha
* getNrTrabalhadores: obtém o número de trabalhadores
* jogar: inicia o ciclo de jogo
* validaComando: valida os comandos inseridos pelo utilizador
* amanhecer: acontecem os vários efeitos das zonas
* anoitecer: colheita de recursos
* ~Ilha: elimina todas as zonas da ilha, libertando memória

#### Ilha

Define os valores por default para os custos dos edifícios, gera um valor aleatório para o saldo entre 50 e 100, inicializa a ilha e executa o ciclo de jogo.

Uma imagem com texto

Descrição gerada automaticamente

Se for passado por parâmetro um objeto do tipo ilha (construtor por cópia) todas as informações da ilha atual são sobrescritas com a informação da ilha que é recebida. Isto serve para dar load a um save que esteja guardado no vetor das ilhas.

#### initIlha

A função initilha pergunta ao utilizador quantas zonas é que quer que a ilha tenha (linhas e colunas) e adiciona ao vetor das zonas os objetos das classes derivadas da classe Zona (como a classe Central ou MinaFerro, por exemplo).

Uma imagem com texto

Descrição gerada automaticamente

#### mostraIlha

A função mostraIlha trata de mostrar a representação visual da ilha e todas as informações pertinentes a esta associada.

Para manter a representação da ilha uniforme usamos um conjunto de funções, como o setfill (que preenche a célula com espaços) e a função setw (para definir quantos espaços é que queremos para dar pad).

Uma imagem com texto

Descrição gerada automaticamente

#### addSaldo

Adiciona ou remove saldo à conta do jogador. Se o valor introduzido for negativo é debitado saldo.

Uma imagem com texto

Descrição gerada automaticamente

#### getSaldo

Obtém o saldo do jogador

Uma imagem com texto

Descrição gerada automaticamente

#### getInfoZona

Obtém uma descrição textual de todas as zonas da ilha. Se a função receber como parâmetro uma linha e uma coluna retorna uma representação textual detalhada para a zona nessa posição.

Uma imagem com texto

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

#### getSaveName

Obtém o nome do save da ilha.

#### Uma imagem com texto Descrição gerada automaticamente

#### getNrTrabalhadores

Obtém o número de trabalhadores total.

Uma imagem com texto

Descrição gerada automaticamente

#### jogar

Ciclo de jogo que consiste num ciclo infinito em que é chamado o validaComando e se o comando tiver sido executado com sucesso é mostrada a ilha com a alteração efetuada.

Uma imagem com texto

Descrição gerada automaticamente

#### validaComando

A função validaComando é das funções mais importantes do jogo visto que esta verifica se o comando que o utilizador introduziu é válido. Se for esse o caso, é executada a ação relativa ao comando que o jogador introduziu.

Uma imagem com texto

Descrição gerada automaticamente

É recebido um comando e com uma técnica que utiliza o istringstream conseguimos dividir o comando por espaços de forma a adicionarmos cada palavra individualmente a um vetor de argumentos que podemos utilizar mais tarde para verificar se os argumentos estão corretos.

#### amanhecer

A função amanhecer realiza os vários acontecimentos previstos para cada zona da ilha, como despedir um funcionário aleatoriamente, verificar se a ilha vai ser destruída por um vulcão, etc...

Uma imagem com texto

Descrição gerada automaticamente

#### anoitecer

A função anoitecer é a função mais completa do jogo, visto que efetua verificações relativas aos edifícios que se encontram adjacentes a um certo edifício de modo a produzir diferentes quantidades de recursos.

Uma imagem com texto

Descrição gerada automaticamente

#### ~Ilha

O destrutor da classe Ilha tem como função eliminar as zonas da ilha libertando assim memória armazenada previamente.

Uma imagem com texto

Descrição gerada automaticamente

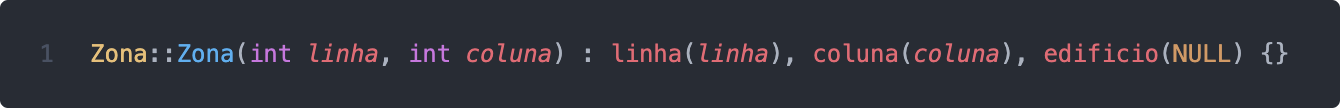
### Zona

A classe Zona representa uma zona existente na ilha. Pode ser povoada com trabalhadores e pode ser construído apenas um edifício na mesma.

* **Variáveis:**
* linha, coluna: guarda a linha e a coluna dessa zona
* sigla: guarda a sigla da zona
* trabalhadores: vetor que guarda os trabalhadores da zona
* edificio: objeto da classe Edifício colocado na zona
* **Funções:**
* Zona: construtor da classe Zona que define os valores das variáveis de um objeto
* getInfoZona: obtém a descrição textual da zona
* setLinha, setColuna: define a linha e a coluna dessa zona
* setSigla: define a sigla da zona
* getSigla: obtém a sigla da zona
* getLinha, getColuna: obtém a linha e a coluna dessa zona
* getSiglaTrabalhadores: obtém a sigla dos trabalhadores da zona
* getTrabalhadorById: obtém o objeto de um trabalhador dessa zona que tenha um certo ID
* getTrabalhadoresBySigla: obtém todos os trabalhadores que sejam de uma determinada profissão
* getTrabalhadores: obtém todos os trabalhadores da zona
* getNrTrabalhadores: obtém o número de trabalhadores na zona
* addTrabalhador, removeTrabalhador: adiciona ou remove um trabalhador da zona
* getSiglaEdificio: obtém a sigla do edifício na zona
* setEdificio: define o edifício nessa zona
* removeEdificio: remove o edifício dessa zona
* getArmazenamento: obtém o armazenamento da zona (caso se aplique)
* addArmazenamento: adiciona armazenamento à zona (caso se aplique)
* getArvores: obtém as árvores na zona (caso se aplique)
* addArvores: adiciona uma quantidade de árvores à zona (caso se aplique)
* getEdificio: obtém o objeto do edifício que se encontra na zona
* ~Zona: destrutor da zona que elimina os trabalhadores e o edifício libertando assim memória

#### Zona

Define as variáveis de uma zona quando esta é criada.



#### getInfoZona

Obtém a informação textual da zona, apresentando o nome da zona, o nome do edifício e se este está ligado ou não.

#### Uma imagem com texto Descrição gerada automaticamente

#### setLinha, setColuna

Define a linha e a coluna para essa zona

Uma imagem com texto

Descrição gerada automaticamente

getSigla

Obtém a sigla dessa zona

Uma imagem com texto

Descrição gerada automaticamente

#### getLinha, getColuna

Obtém a linha e a coluna dessa zona

Uma imagem com texto

Descrição gerada automaticamente

#### getSiglaTrabalhadores

Obtém a sigla dos trabalhores presentes na zona, por exemplo “OOMLMO”.

Uma imagem com texto

Descrição gerada automaticamente

#### getTrabalhadorById

Obtém o objeto de um trabalhador dessa zona com um determinado ID.

Uma imagem com texto

Descrição gerada automaticamente

#### getTrabalhadoresBySigla

Retorna um vetor que contém os objetos de todos os trabalhadores dessa zona, sendo estes de um determinado tipo.

Uma imagem com texto

Descrição gerada automaticamente

#### getTrabalhadores

Retorna o vetor dos trabalhadores dessa zona.

Uma imagem com texto

Descrição gerada automaticamente

#### getNrTrabalhadores

Obtém o número total de trabalhadores existentes nessa zona.

Uma imagem com texto

Descrição gerada automaticamente

#### addTrabalhador

Adiciona um trabalhador ao vetor de trabalhadores dessa zona.

Uma imagem com texto

Descrição gerada automaticamente

#### removeTrabalhador

Remove um trabalhador do vetor de trabalhadores da zona utilizando um iterador.

Uma imagem com texto

Descrição gerada automaticamente

#### getSiglaEdificio

Obtém a sigla do edifício construído nessa zona. Se não existir nenhum edifício nessa zona, retorna uma string vazia.

Uma imagem com texto

Descrição gerada automaticamente

#### setEdificio

Constrói o edifício nessa zona.

Uma imagem com texto

Descrição gerada automaticamente

#### removeEdificio

Remove o edifício dessa zona e liberta a memória por este alocada.

Uma imagem com texto

Descrição gerada automaticamente

#### getArmazenamento

Obtém o armazenamento de uma zona se essa zona tiver armazenamento. É uma função virtual portanto, caso uma zona não tenha armazenamento, é retornado o valor default que é 0.

Uma imagem com texto

Descrição gerada automaticamente

#### getEdificio

Retorna o objeto do edifício dessa zona.

Uma imagem com texto

Descrição gerada automaticamente

#### ~Zona

Elimina todos os trabalhadores da zona e remove o edifício libertando assim a memória por esses alocada.

Uma imagem com texto

Descrição gerada automaticamente

Na classe Zona existem seis classes derivadas:

* Deserto
* Montanha
* Floresta
* Pastagem
* Pantano
* Vulcao

Cada classe tem os seus próprios atributos para além dos da classe base.

Uma imagem com texto

Descrição gerada automaticamente

### Edificio

A classe Edificio representa um edifício que pode estar posicionado numa determinada zona da ilha.

* **Variáveis:**
* sigla: guarda a sigla desse edifício
* ligado: guarda o estado do edifício (ligado/desligado)
* diaConstrucao: guarda o dia em que o edifício foi construído
* **Funções:**
* Edificio: construtor da classe Edificio que trata de inicializar as variáveis desse objeto com informação pertinente passada por parâmetros
* ligar: liga o edifício
* desligar: desliga o edifício
* getLigado: obtém o estado do edifício (ligado/desligado)
* getDiaConstrucao: obtém o dia em que o edifício foi construído
* getSigla: obtém a sigla do edifício
* getCusto: obtém custo de construção do edifício (caso se aplique)
* getNivel: obtém nível do edifício (caso se aplique)
* addNivel: sobe o nível do edifício (caso se aplique)
* getArmazenamento: obtém o armazenamento de um edifício (caso se aplique)
* getArmazenamentoMax: obtém o armazenamento máximo de um edifício (caso se aplique)
* addArmazenamento: adiciona armazenamento à capacidade de armazenamento do edifício (caso se aplique)

#### Edificio

Define as variáveis do objeto com os valores passados por parâmetros.

Uma imagem com texto

Descrição gerada automaticamente

#### ligar, desligar

Liga ou desliga o edifício.

Uma imagem com texto

Descrição gerada automaticamente

#### getLigado

Obtém o estado do edifício (ligado/desligado).

Uma imagem com texto

Descrição gerada automaticamente

#### getDiaConstrucao

Obtém o dia de construção do edifício.

Uma imagem com texto

Descrição gerada automaticamente

#### getSigla

Obtém a sigla do edifício.

Uma imagem com texto

Descrição gerada automaticamente

#### getCusto

Obtém o custo de construção do edifício. É uma função virtual portanto, caso um edifício não possua um custo de construção é retornado 0 por default.

Uma imagem com texto

Descrição gerada automaticamente

Na classe Edificio existem seis classes derivadas:

* MinaFerro
* MinaCarvao
* Central
* Fundicao
* Bateria
* Purgatorio

Cada classe tem os seus próprios atributos para além dos da classe base.

Uma imagem com texto

Descrição gerada automaticamente

### Trabalhador

A classe Trabalhador representa um trabalhador que pode estar posicionado em qualquer zona da ilha.

* **Variáveis:**
* custo: guarda o custo de contratação do trabalhador
* sigla: guarda a sigla do trabalhador
* nrTrabalhadorGlobal: guarda o número global de trabalhadores independentemente da profissão
* nTrabalhador: guarda o número do trabalhador
* diaContrato: guarda o dia em que o trabalhador foi contratado
* idTrabalhador: guarda o ID do trabalhador
* **Funções:**
* Trabalhador: construtor da classe Trabalhador que tem como função definir as variáveis do objeto com valores pertinentes tal como o ID do trabalhador
* getIdTrabalhador: obtém o ID do trabalhador
* getCusto: obtém o custo de contratação do trabalhador
* getDiaContrato: obtém o dia em que o trabalhador foi contratado
* getSigla: obtém a sigla do trabalhador
* setSigla: define a sigla do trabalhador
* getDiasTrabalhados: obtém o número de dias trabalhados (caso se aplique)
* setDiasTrabalhados: define o número de dias trabalhados (caso se aplique)
* getOperacional: obtém o estado de um trabalhador (caso se aplique)
* setOperacional: define o estado de um trabalhador (caso se aplique)

#### Trabalhador

Construtor da classe que trata de definir as variáveis com valores recebidos como parâmetro. Para além disso gera também um ID para o trabalhador com base no dia em que foi contratado e no número de trabalhadores.

Uma imagem com texto

Descrição gerada automaticamente

#### getIdTrabalhador

Obtém o ID do trabalhador.

Uma imagem com texto

Descrição gerada automaticamente

#### getCusto

Obtém o custo de contratação do trabalhador.

Uma imagem com texto

Descrição gerada automaticamente

#### getDiaContrato

Obtém o dia em que o trabalhador foi contratado.

Uma imagem com texto

Descrição gerada automaticamente

#### getSigla

Obtém a sigla do trabalhador.

Uma imagem com texto

Descrição gerada automaticamente

#### setSigla

Define a sigla do trabalhador.

Uma imagem com texto

Descrição gerada automaticamente

#### getDiasTrabalhados

Obtém o número de dias trabalhados. É uma função virtual portanto se o trabalhador em que a função é chamada não tiver o atributo “diasTrabalhados” a função retorna o valor 0 por default.

Uma imagem com texto

Descrição gerada automaticamente

A classe Trabalhador tem três classes derivadas:

* Operario
* Lenhador
* Mineiro

Cada classe tem os seus próprios atributos para além dos da classe base.

Uma imagem com texto

Descrição gerada automaticamente

## Comandos

Os comandos são utilizados pelo jogador com o objetivo de interagir com o jogo, de modo a efetuar operações como construir edifícios ou contratar trabalhadores.

### cons

O comando cons permite ao jogador construir um edifício numa zona. São efetuadas várias verificações e se passar a todas é chamada a função setEdificio sobre a zona escolhida pelo utilizador.

Uma imagem com texto, monitor, captura de ecrã, ecrã

Descrição gerada automaticamente

### cont

O comando cont permite ao jogador contratar um trabalhador de um certo tipo. Este trabalhador é sempre colocado numa zona de pasto escolhida aleatoriamente.

Uma imagem com texto

Descrição gerada automaticamente

### liga/des

Os comandos liga e des permitem ao jogador ligar ou desligar um edifício de uma certa zona.

Uma imagem com texto

Descrição gerada automaticamente

### list

O comando list quando executado sem quaisquer argumentos adicionais mostra uma descrição textual de todas as zonas da ilha.

Quando acompanhado por um número de uma linha e de uma coluna mostra apenas informação mais detalhada sobre essa zona.

Uma imagem com texto

Descrição gerada automaticamente

### config

O comando config permite ao jogador redefinir o custo de compra ou contratação de um edifício ou trabalhador, respetivamente.

Uma imagem com texto

Descrição gerada automaticamente

### move

O comando move permite ao jogador mover um trabalhador para uma determinada zona da ilha.

Uma imagem com texto

Descrição gerada automaticamente

### vende

O comando vende serve dois propósitos: vender um determinado recurso e vender um edifício numa certa zona.

Uma imagem com texto, captura de ecrã, ecrã

Descrição gerada automaticamente

Uma imagem com texto

Descrição gerada automaticamente

### debcash

O comando debcash é um comando de debugging que serve para adicionar ou remover dinheiro.

Uma imagem com texto

Descrição gerada automaticamente

### debed

O comando debed é um comando de debugging que serve para adicionar um edifício numa zona a custo zero.

Uma imagem com texto

Descrição gerada automaticamente

### debkill

O comando debkill é um comando de debugging que serve para remover (matar) um trabalhador de uma zona.

Uma imagem com texto

Descrição gerada automaticamente

### save

O comando save permite ao jogador efetuar uma cópia do estado da ilha para mais tarde poder fazer load do estado guardado com o save caso assim o deseje.

Uma imagem com texto, captura de ecrã, ecrã

Descrição gerada automaticamente

### apaga

O comando apaga permite ao jogador apagar um save previamente criado.

Uma imagem com texto, monitor, ecrã, captura de ecrã

Descrição gerada automaticamente

### load

O comando load permite ao jogador carregar um save substituindo assim a ilha atual com a ilha guardada nesse save.

Uma imagem com texto, ecrã, captura de ecrã

Descrição gerada automaticamente

### next

O comando next permite ao jogador avançar para o dia seguinte. Durante a noite é efetuada a colheita de recursos e de manhã são efetuados os vários efeitos previstos para os diferentes tipos de zona.



As funções anoitecer e amanhecer foram descritas em detalhe no tópico [amanhecer](#_amanhecer) e [anoitecer](#_anoitecer), respetivamente.

### upgrade

O comando upgrade permite ao jogador aumentar o nível do seu edifício caso esse edifício permita esse tipo de operação. São feitas várias verificações para certificarmo-nos que o jogador tem o que é necessário para aumentar o nível de um certo edifício.

Uma imagem com texto, monitor, captura de ecrã, televisão

Descrição gerada automaticamente

## Funções de Suporte

Foram criadas várias funções de suporte tais como: iniciar a ilha, mostrar a ilha, verificar se um número é realmente um número, validar os comandos, etc...

As funções de suporte desenvolvidas até ao momento são as seguintes:

* isNumber: verifica se uma string é um número válido
* validaPosicao: verifica se uma posição existe
* mostraASCII: mostra uma arte em forma de texto a dizer “Ilhéu”
* menu: apresenta o menu inicial

### isNumber

A função isNumber é uma simples função que utiliza expressões regulares (regex) para verificar se uma string é realmente um número válido. Esta função é utilizada em vários sítios para confirmar o input do utilizador.

Uma imagem com texto

Descrição gerada automaticamente

A expressão regular verifica se existe um menos ou não atrás de um caractere (feito com o -{0, 1}) que tem obrigatoriamente de ser um dígito positivo (graças ao *d+*). Se a string se enquadrar no formato da expressão regular a função regex\_match retorna true e esse é o output da função isNumber.

### validaPosicao

A função validaPosicao tem como objetivo verificar se a posição à qual queremos aceder existe. Recebe a posição à qual queremos aceder e o tamanho da ilha.

Retorna true se essa posição existir ou false caso contrário.

Uma imagem com texto

Descrição gerada automaticamente

### mostraASCII

Mostra uma arte em forma de texto a dizer “Ilhéu”.

Uma imagem com texto

Descrição gerada automaticamente

### menu

Menu inicial que pergunta ao utilizador se quer jogar

Uma imagem com texto

Descrição gerada automaticamente

# Conclusão

Uma imagem com noz, fruta

Descrição gerada automaticamenteEste trabalho prático permitiu-nos aplicar os conhecimentos adquiridos nas aulas de Programação Orientada a Objetos e colocou-nos à prova relativamente àquilo que sabíamos fazer. Sempre que não conseguíamos avançar com o desenvolvimento do trabalho éramos obrigados a fazer pesquisas de modo a conseguirmos ultrapassar certos obstáculos.

Por se tratar de um jogo, isso despertou em um grande interesse e motivação para realizar este trabalho.

Algo que também achámos interessante foi o paradigma de programação orientada a objetos que é bastante diferente do que estávamos habituados.

