

# Relatório

## Requisito Extra

### Grupo: 50

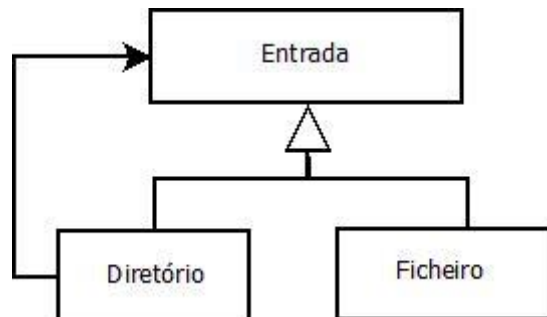
Diogo Mateus	Nº: 65889
Rodrigo Rosa	Nº: 70489
Ricardo Sequeira	Nº: 79750

### Objectivo

Com a implementação do requisito extra, a aplicação permite aceder a qualquer ao conteúdo total de um directório ou seja, a todas as entradas contidas neste, sejam ficheiros ou outros subdirectórios, neste caso este tipo de comandos é exemplificado com o cálculo do tamanho absoluto de um dado directório (soma do tamanho de todos os seus directórios filhos) e a procura de um ficheiro num directório e seus sub directórios.

### Implementação

A solução apresentada pelo nosso para a realização do requisito extra do projeto baseia-se sobretudo em dois padrões de desenho: o **Composite Pattern** e o **Template Method Pattern**. O primeiro padrão deve-se ao facto que um directório é um objeto que é constituído por entradas, ou seja, ficheiros ou mesmo outros directórios, formando uma hierarquia de classes com características similares.

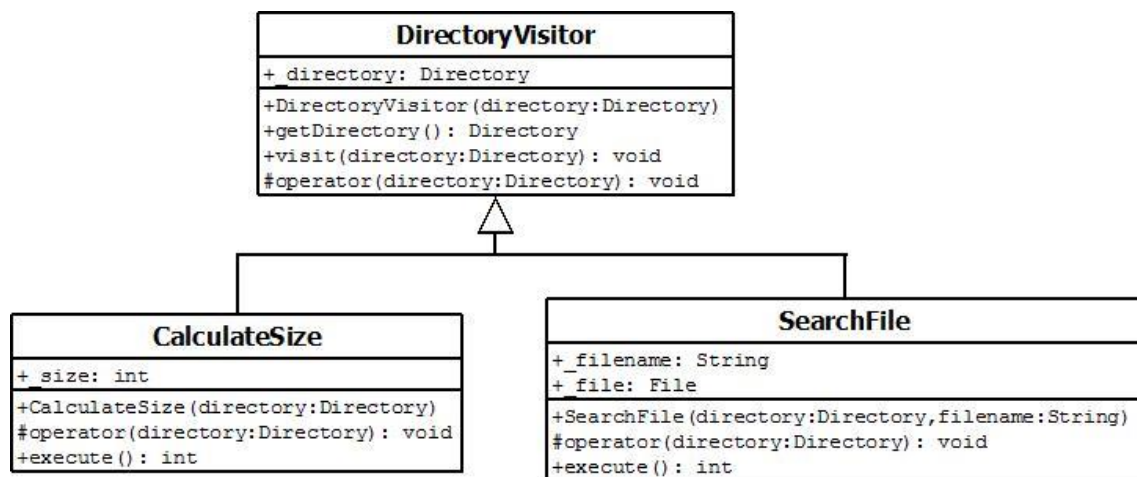


Analisando as duas funcionalidades propostas no requisito extra, começámos por estudar o que de comum havia entre estes dois comandos e mesmo de outros que poderíamos pensar em implementar. Concluímos que ambas exploravam todo o conteúdo de um directório, primeiro atuando sobre o próprio directório onde eram chamadas e depois atuando sobre todos os directórios que constituíam o directório inicial, até chegar ao final da hierarquia.

Assim sendo, procurávamos desenvolver um algoritmo que, apesar de ser maioritariamente semelhante entre as duas funcionalidades (ou outras que fossem adicionadas no futuro), haveriam certas frações desse mesmo algoritmo que seriam diferentes de acordo com o comando pretendido.

Para implementar esta ideia, o padrão de desenho que nos pareceu mais adequado foi o Template Method Pattern que nos permitiu escrever a parte fixa do algoritmo (*public void visit(Directory directory)*) numa superclasse abstrata genérica (*DirectoryVisitor*), que consiste em aplicar uma operação no diretório atual e depois aplicar a mesma operação em todos os subdiretórios, que não sejam o diretório pai nem o próprio diretório (estes já foram verificados – entraria em ciclo infinito).

Acontece que a solução para a facilidade na criação de comandos que sigam este algoritmo, sem a repetição de código, prende-se na chamada a um método abstracto (*operator(Directory directory)*) dentro da parte fixa do algoritmo presente no método *visit*. Desta maneira, para criar comandos como os dois do enunciado apenas temos de criar uma subclasse de *DirectoryVisitor*, implementar o método *operator* e retornar o resultado pretendido depois da chamada ao método *visit*.



- **Calcular o tamanho total do diretório (incluindo sub diretórios)**

Criámos a subclasse *CalculateSize* cujo método *operator* calcula o tamanho de um diretório enquanto o método *execute* chama o método fixo *visit* e retorna o tamanho.

- **Procurar um ficheiro no directório e nos seus sub-directórios:**

Criámos a subclasse *SearchFile* que recebe o nome do ficheiro a procurar aquando da sua criação cujo método *operator* procura o ficheiro no diretório onde se encontra enquanto o método *execute* chama o método fixo *visit* e retorna o caminho atual do ficheiro que se procurava.