

Engenharia Informática

Unidade Curricular: Estrutura de Dados

Docentes: Ricardo Santos

João Ramos

Fábio Silva

João Magalhães

Estudantes: 8170134 – Daniel Babo, Turma 1

8170573 – Mário Mendes, Turma 2

Licenciatura em Engenharia Informática

2ºAno

Ano letivo: 2018/2019

Felgueiras, 29 de Janeiro de 2019

Índice

Índice de figuras	iii
Introdução.....	4
Diagramas de classes	5
Estruturas utilizadas	10
Funcionalidades implementadas	11
Algoritmos implementados.....	12
Conclusão	13

Índice de figuras

Figura 1 - Diagrama de classes para o package ArrayList.....	5
Figura 2 - Diagrama de classes para o package Graphs	6
Figura 3 - Diagrama de classes para o package Heap	7
Figura 4 - Diagrama de classes para o package LinkedQueue	8
Figura 5 - Diagrama de classes para o package LinkedStack.....	8
Figura 6 - Diagrama de classes para o package User	9

Introdução

No âmbito da unidade curricular de Estrutura de Dados foi-nos solicitado pelos docentes da disciplina que elaborássemos um programa em linguagem JAVA que, através de uma plataforma social para a gestão de recursos humanos, fosse capaz de facilitar o encontro entre recrutadores/empresas e utilizadores da plataforma com as competências procuradas, por exemplo, competências profissionais, aspeto social, entre outras. Para isso, o programa deve ser capaz de carregar e guardar ficheiros JSON com a informação relativa aos utilizadores e conexões para processamento offline, bem como algoritmos adequados para o encontro entre empresas e utilizadores, com base nas suas características sociais e profissionais.

Diagramas de classes

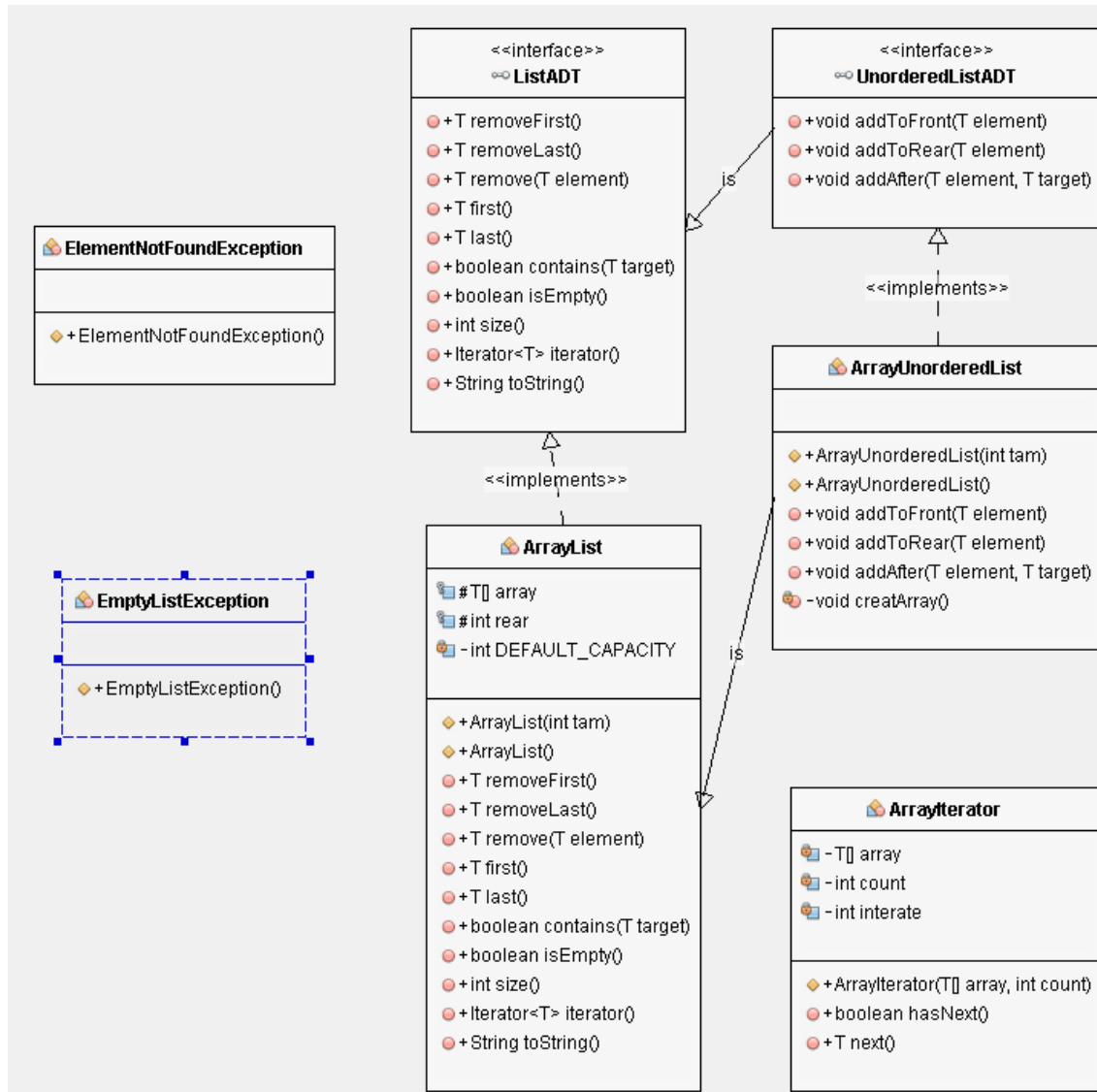


Figura 1-Diagrama de classes para o package ArrayList

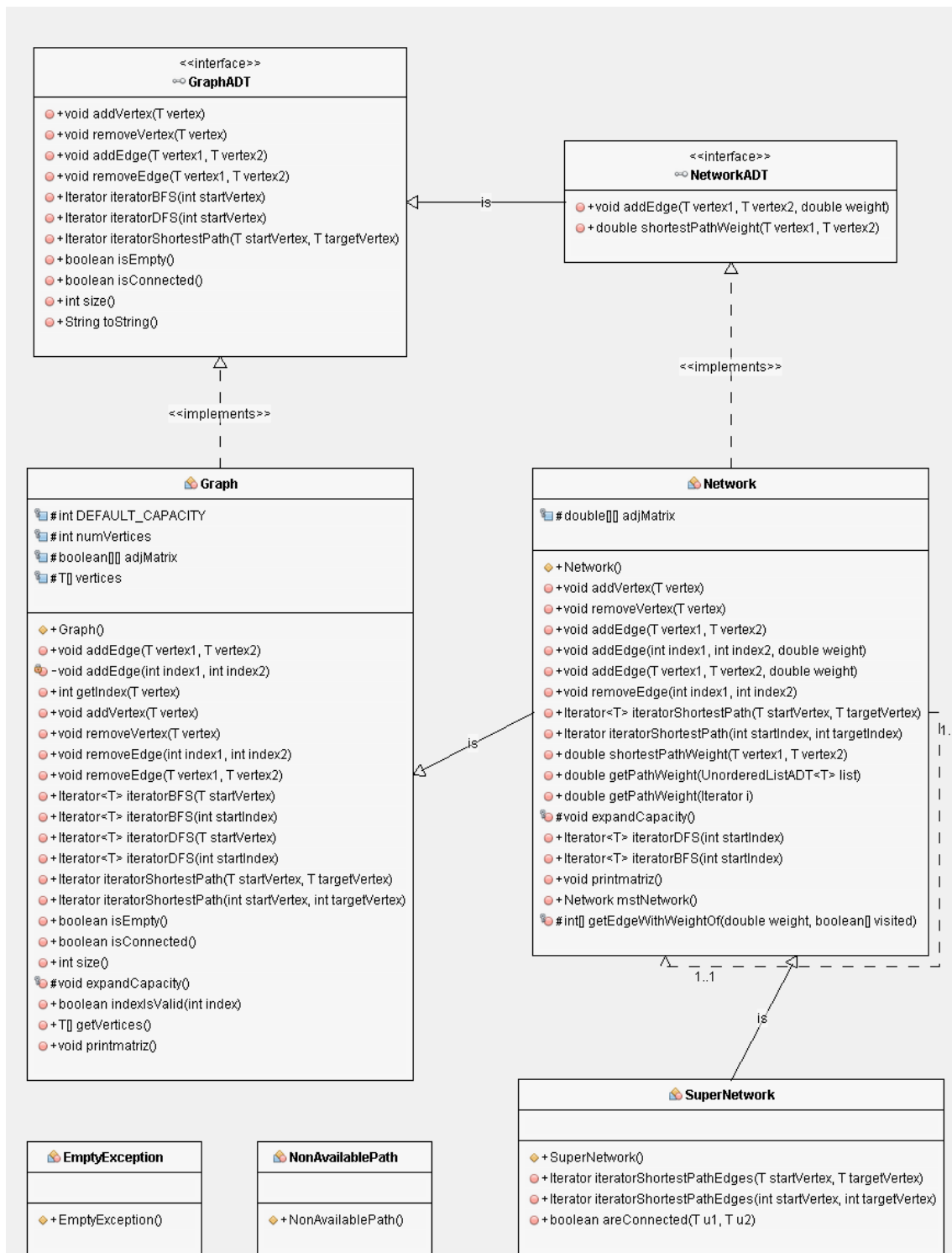


Figura 2-Diagrama de classes para o package Graphs

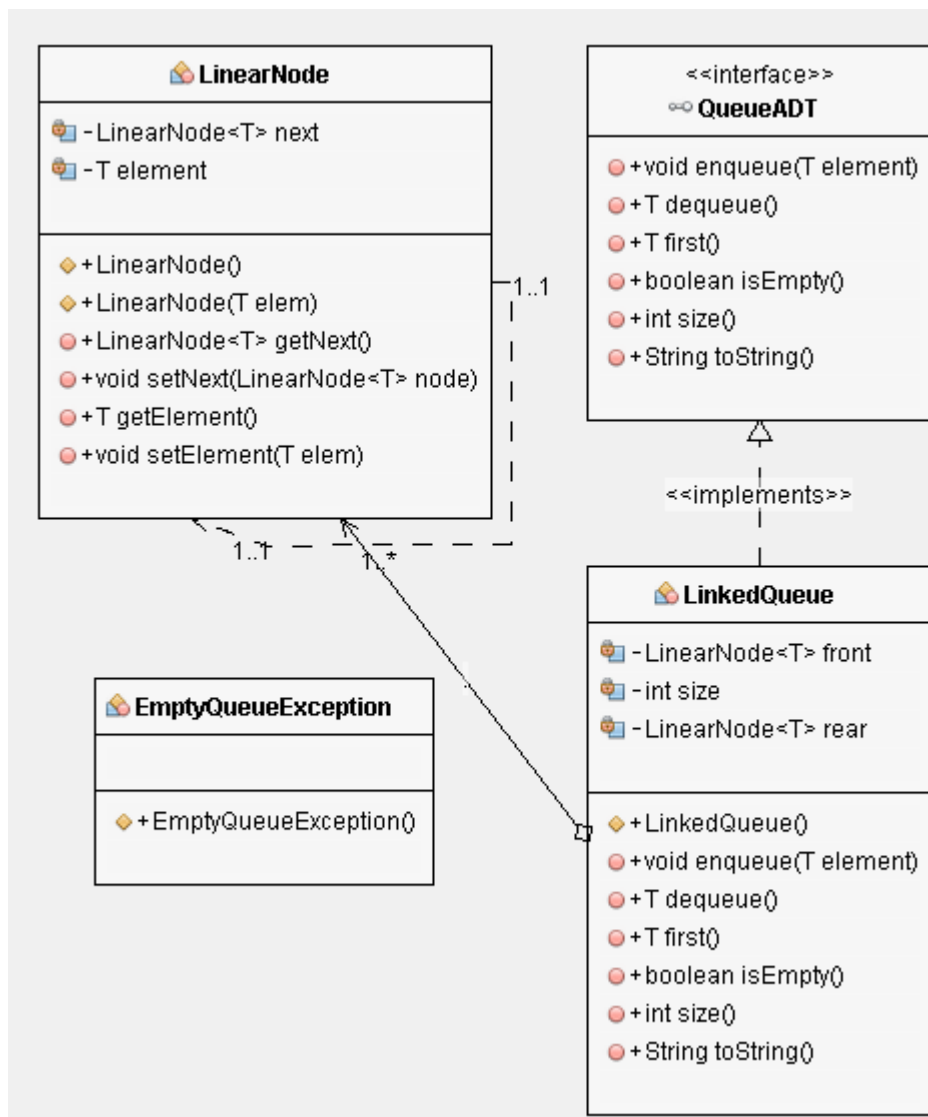


Figura 4-Diagrama de classes para o package `LinkedQueue`

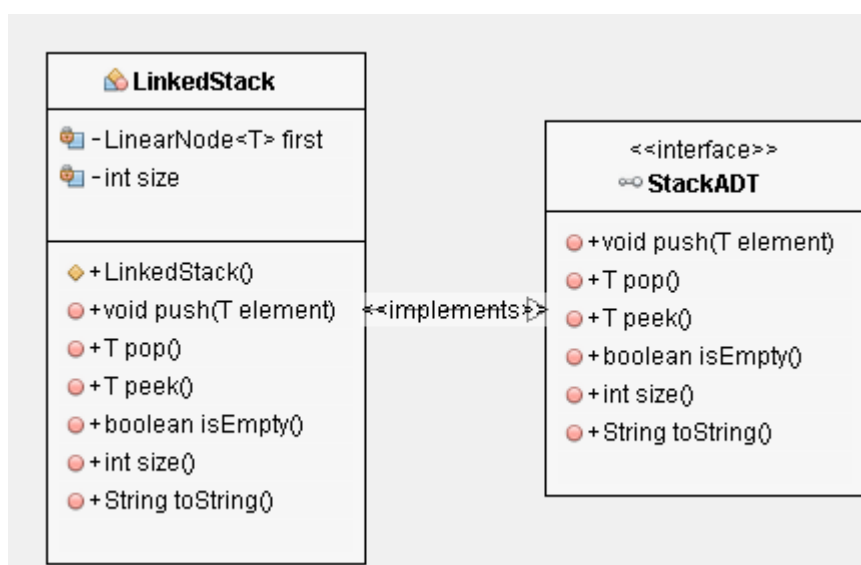


Figura 5-Diagrama de classes para o package `LinkedStack`

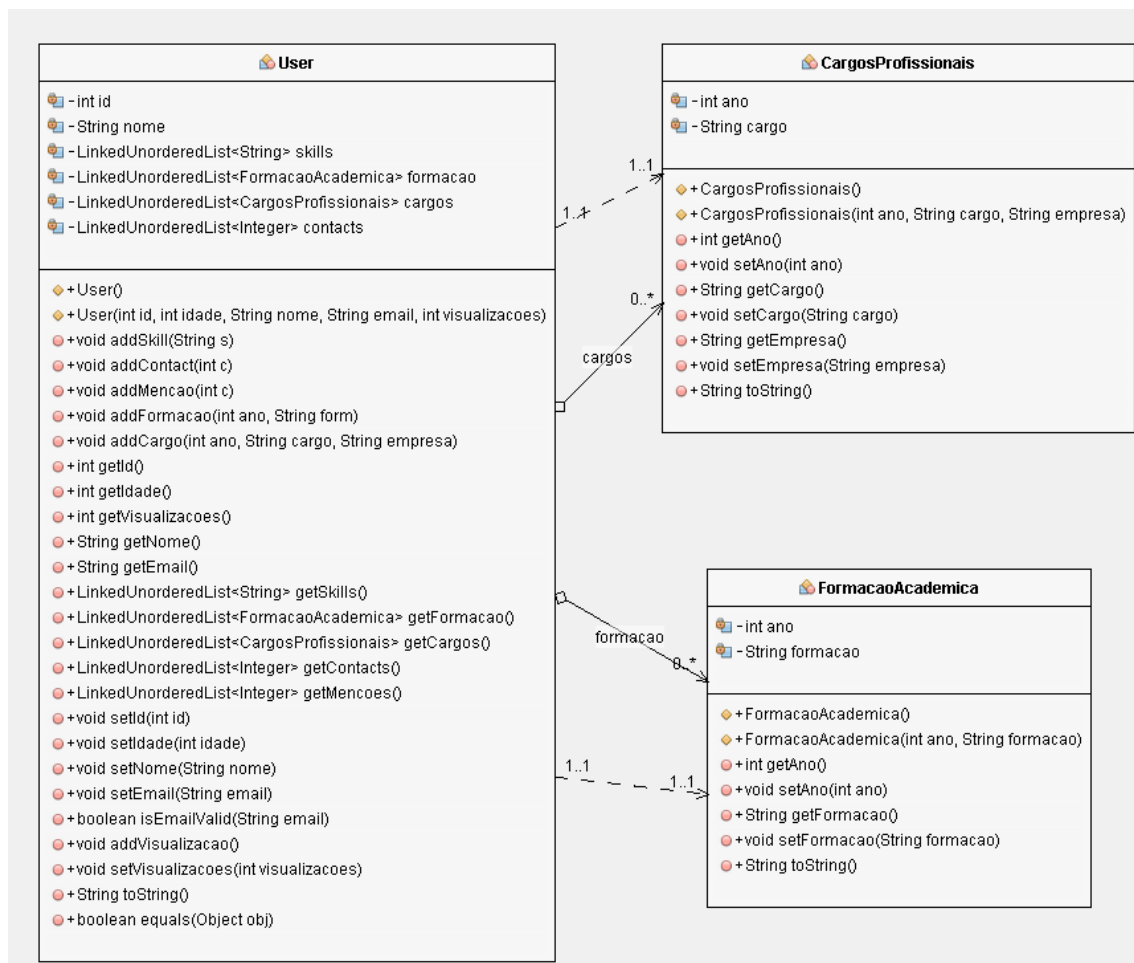


Figura 6-Diagrama de classes para o package User

Estruturas utilizadas

Para elaboração do trabalho prático usamos as estruturas desenvolvidas ao longo do semestre nas fichas práticas: LinkedList e ArrayList, LinkedQueue, LinkedStack, Graph, Network e Heap, bem como todas as estruturas necessárias para a criação das mesmas. No caso das variáveis presentes na classe User foram utilizadas estruturas em lista ligada, evitando assim a criação de vários métodos `expandCapacity()`, pois cada variável era inicializada com um valor diferente. Noutros casos, como por exemplo no retorno de métodos, utilizamos o ArrayList. Ambas as implementações serviam para o objetivo do problema, então decidimos usar as duas.

Funcionalidades implementadas

Tendo em conta o enunciado, implementamos todos os requisitos obrigatórios, tendo deixado os de bonificação por falta de tempo. No entanto, parte de um dos requisitos encontra-se implementado, nomeadamente o de calcular o caminho mais curto com uma constante fixa para todas as ligações, apesar de não se encontrar disponível para uso na interface gráfica.

Algoritmos implementados

No trabalho foi implementado o algoritmo de Dijkstra tanto para encontrar o caminho mais curto pelo número de arestas como o caminho mais curto tendo em conta o peso da aresta. Depois de várias tentativas, foi implementado um algoritmo com o objetivo de calcular o caminho mais curto. A partir de um vértice inicial, calcula o caminho mais curto para todos os outros, guardando sempre o vértice anterior. No final vai ao último vértice e efetua o caminho inverso, guardando sempre o vértice ate chegar novamente ao inicial.

Conclusão

Com a realização deste trabalho prático ambos pudemos melhorar a capacidade de programação em java, principalmente no que concerne a algoritmos e interfaces gráficas, bem como o espírito de trabalho em equipa, dividindo tarefas e ajudando-nos um ao outro quando surgiram dificuldades. Apesar de ser um pouco demorado, conseguimos organizar o tempo de forma a conseguir implementar todas as funcionalidades pedidas. Para além da experiencia positiva também achamos que o trabalho realizado está bem conseguido.