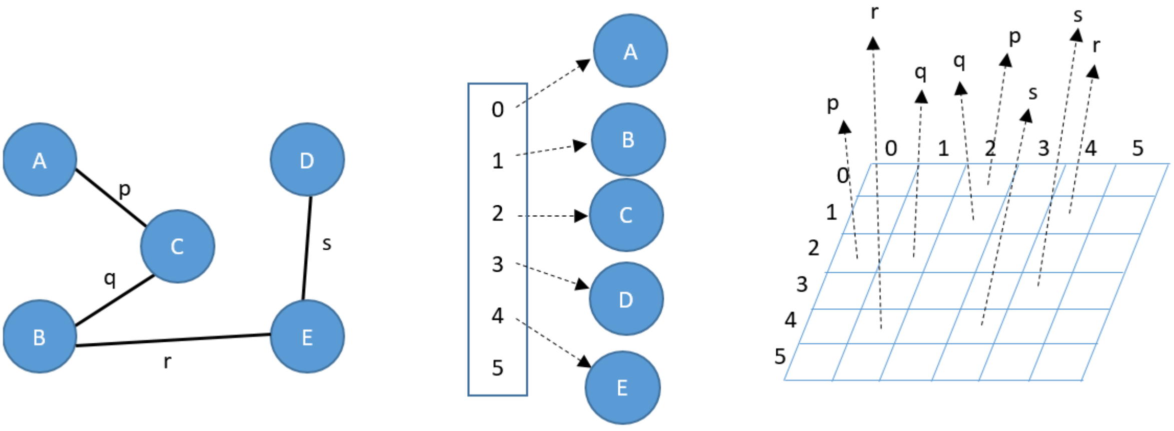
**Laboratory Class 5**

## Graphs: Adjacency Matrix Structure

**Estruturas de Informação**

A ***graph*** *G is* a set *V* of ***vertices*** and a collection *E* of pairs of vertices from *V*, called ***edges***. The aim of this worksheet is to complete and use an implementation of the Graph ADT based on the ***adjacency matrix r***epresentation.

As illustrated in figure 1, with this representation the set ***V* of *vertices*** are stored in an **ArrayList** and the set of edges are represented in a **matrix**.



a) b) c)

Figure1 - (a) An undirected graph G; (b) vertices' ArrayList; (c) edges' matrix

Download and extract the project **AdjacencyMatrixGraph_Initial** from moodle and analyse the classes.

All the classes use generic parameters V and E to designate the element type stored respectively at vertices and edges.

1. Complete the generic class **AdjacencyMatrixGraph<V, E>** to include the method to

   (i) return the set of **outgoing edges** of a particular vertex;

   (ii) return the set of **directed connected vertices** from a particular vertex.

2. Complete the generic class **GraphAlgorithms<V,E>** to include methods to

   (i) return a **DFS** visit of the graph from a particular vertex;

   (ii) return a new graph which represents the **transitive closure** of the current graph;

   (iii) complete the test classe to test the TransitiveClosure.

3. Complete the generic class **EdgeAsDoubleGraphAlgorithms<V,E>** to include the method to:

   (i) return the **shortest path** between a pair of vertices.