

Relatorio

Mapa Digital

Miguel Dias nº1070344 Nuno Ferreira nº1100647

Introdução

No âmbito da disciplina de estruturas de Informação, realizamos um projecto denominado Mapa Digital, que consiste na realização de um mapa turístico.

Este projecto foi desenvolvido na plataforma Visual Studio 2012 da Microsoft, utilizando a linguagem de programação c++, com recurso a uma biblioteca STL.

As características deste mapa digital são referidas no enunciado apresentado a seguir, pelo que o desenvolvimento do projecto segue essas mesmas orientações. Das quais se destacam os dois tipos de locais turístico: naturais e culturais. Todos eles possuem uma descrição que, no caso dos locais turísticos naturais inclui o tempo médio de visita, o horário de abertura e o de encerramento e que, no caso dos locais turísticos culturais inclui a área em quilómetros.

O mapa digital te nele representadas também as vias de ligação que unem os locais turísticos, estas vias de ligação permitem criar um grafo que liga os diferentes locais. As vias de ligação contêm informação como código, quilómetros, tempo médio de percurso, local origem e local destino. As vias de ligação que são, neste caso, estradas nacionais e auto-estradas, e que podem possuir diferentes tipos de pavimento.

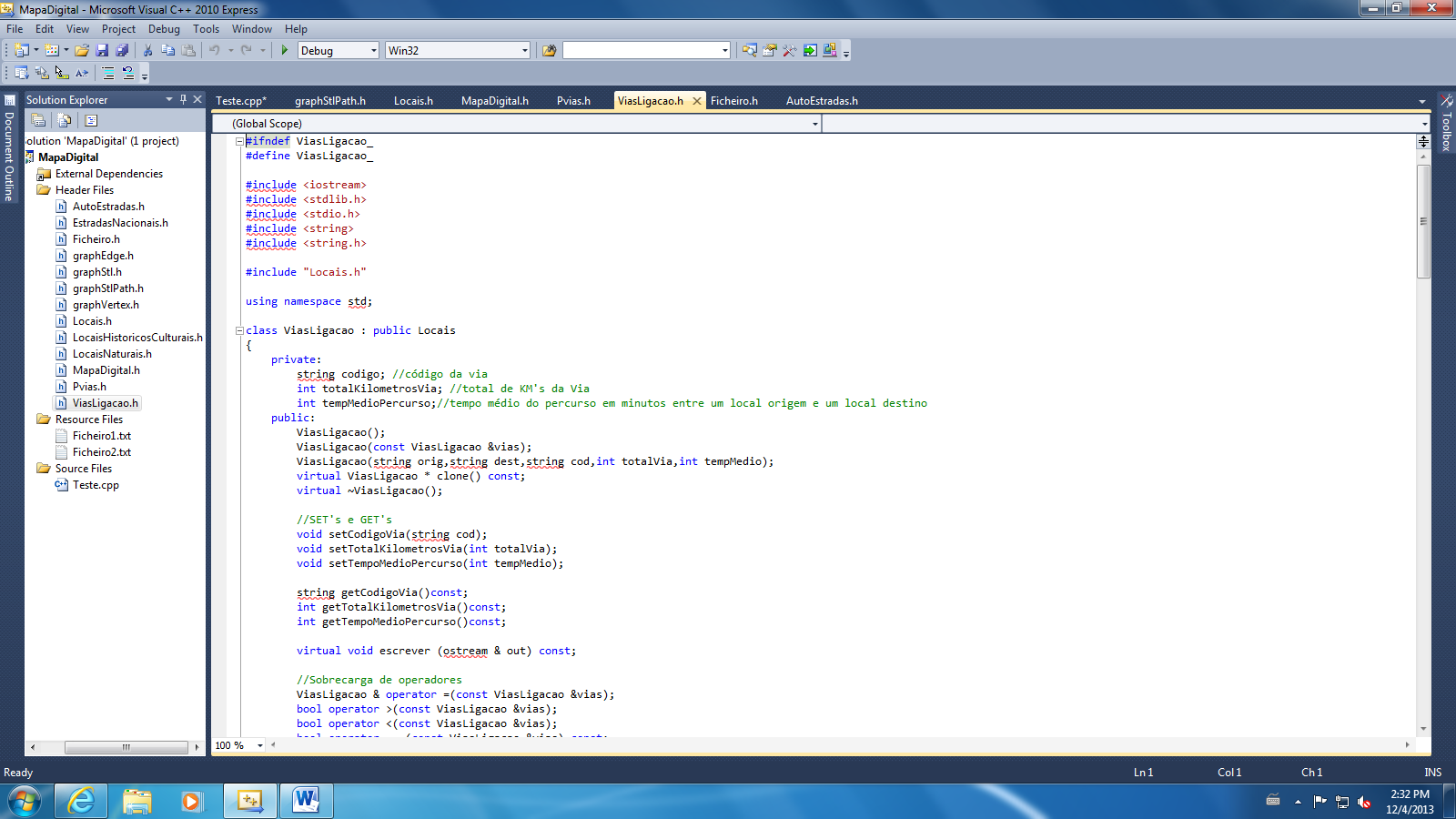
Os locais podem ser acedidos de um ficheiro de texto, inserido previamente pelo utilizador. O programa conta ainda os locais turísticos e ordena-os por ordem alfabética. Apresenta ainda outras funcionalidades, como, por exemplo, determinar o caminho mais curto entre dois locais ou determinar a complexidade temporal.

Descrição

Complexidade Temporal:

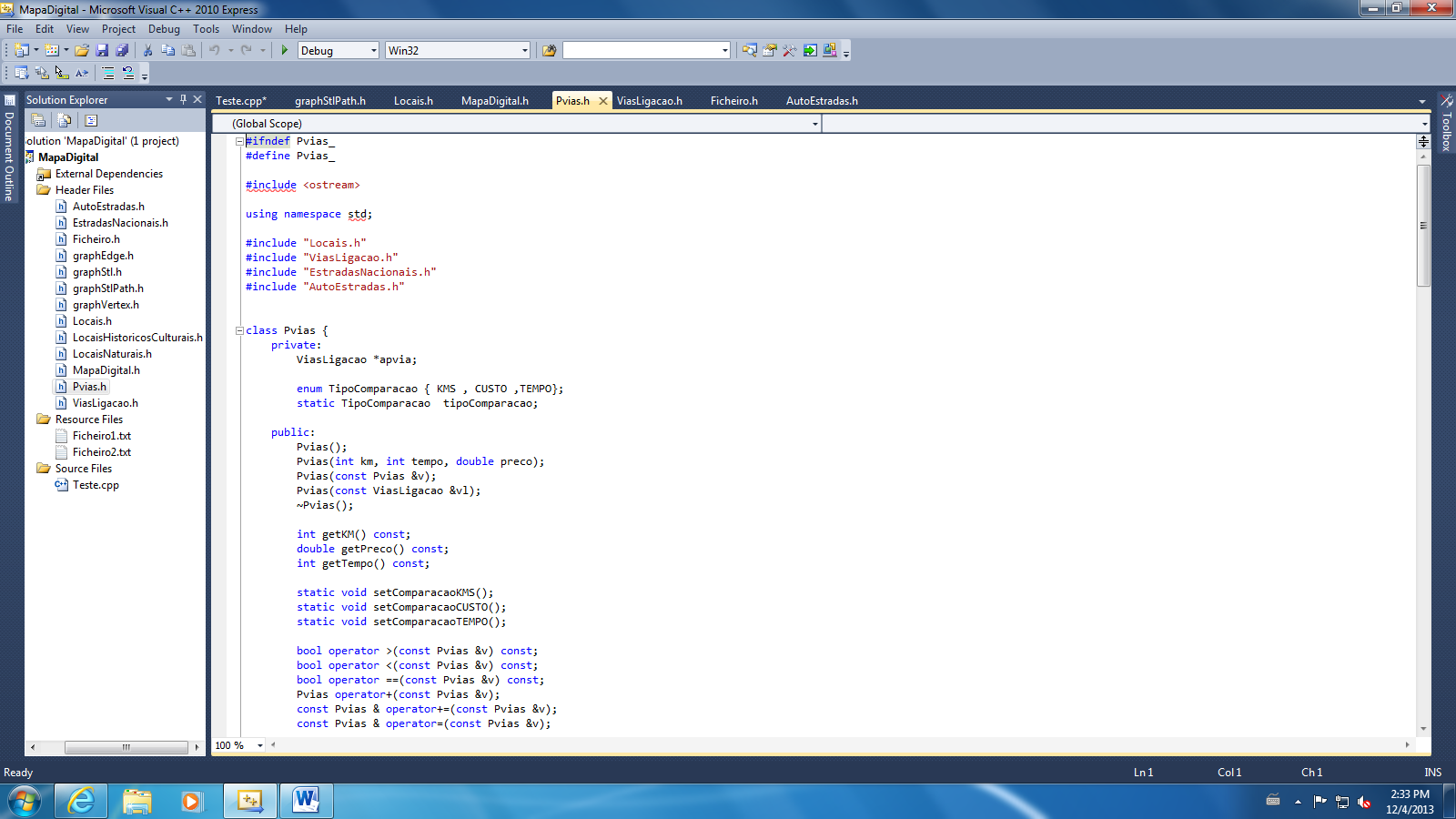
-De acordo com os métodos de criação do grafo, concluímos que se trata de uma O(n2).

Classe ViasLigacao.h:



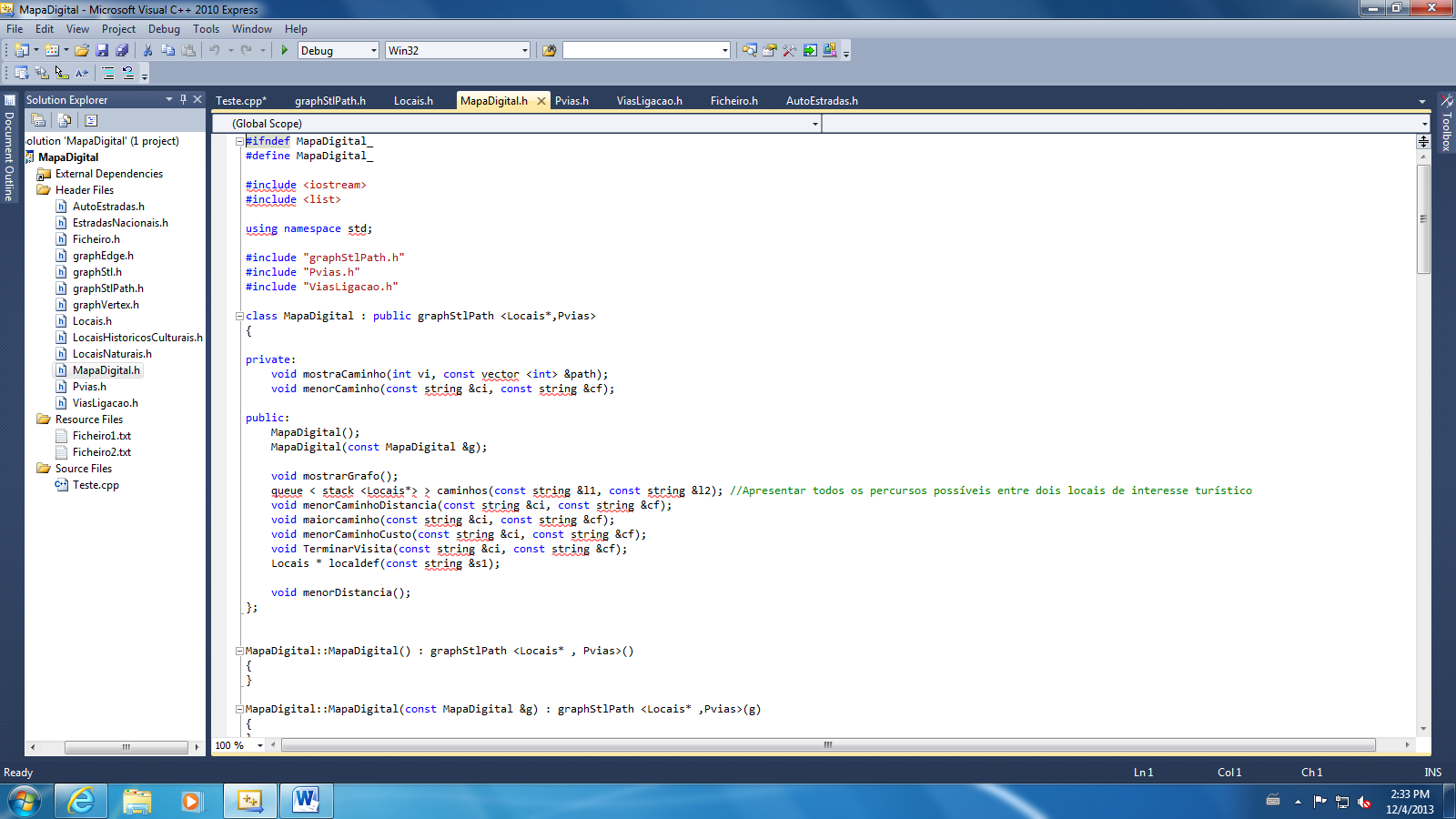
Nesta classe, o método escrever() permite fazer o output de um texto com as caracteristicas das vias de ligação, nomeadamente código, total de quilómetros e tempo médio de percurso, que por sua vez são determinadas e retornadas através dos métodos set e get respectivos.

Classe Pvias.h:



A classe PVias.h permite criar um apontador de vias de ligação de modo a poder fazer sets e gets com a sobrecarga de apontadores de locais.

Classe MapaDigital.h:

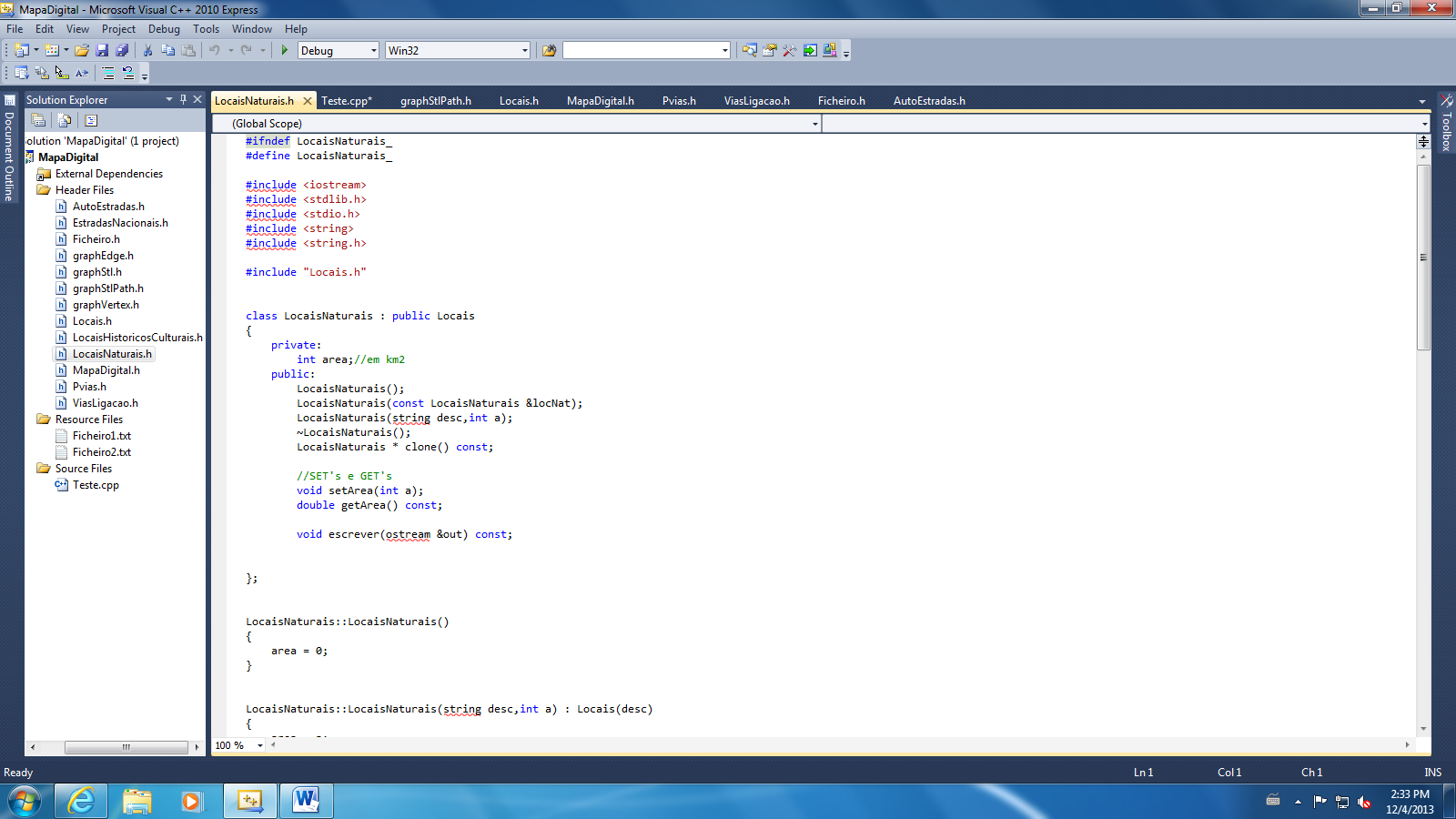


Na classe MapaDigital, o método criarGrafo permite criar um grafo recebendo os valores do método percorreListaVias e enviando-os para o método addGraphEdge, da classe GraphEdge.

O método percorreListaVias vai buscar os valores recebe os valores da lista das vias presentes na classe Pvias.h e envia-os para o método criarGrafo, como é descrito anteriormente.

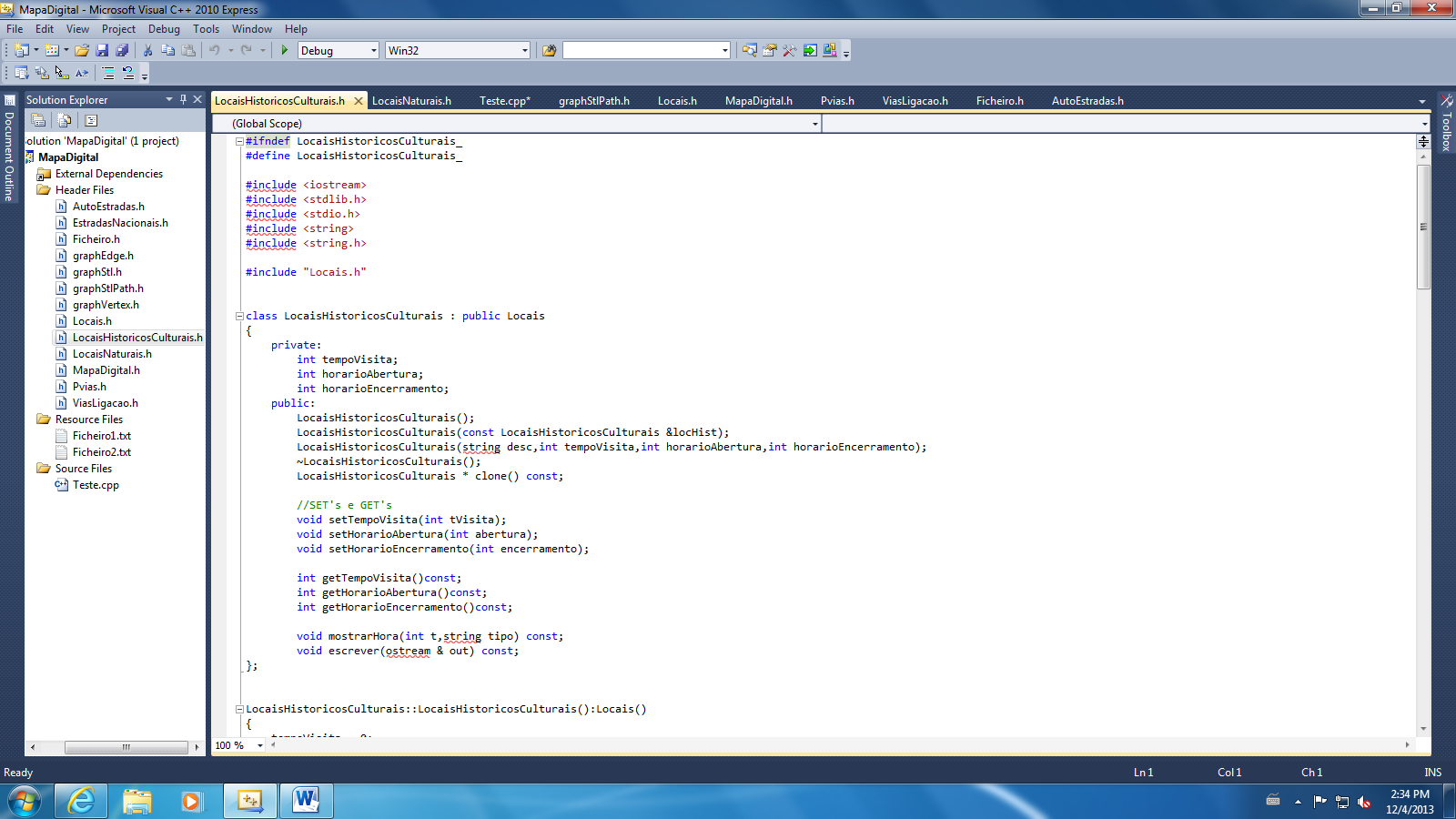
O método caminhoMaisCurto invoca o método do algoritmo de Dijsktra, utilizando a chave presente no método getDisjktraPath da classe GraphStlPath.

Classe LocaisNaturais.h:



Na classe LocaisNaturais, para é de destacar os métodos setArea e getArea que determinam e retornam a área dos locais naturais, respectivamente. O método escrever, faz o output com o tipo de local e a respectiva área.

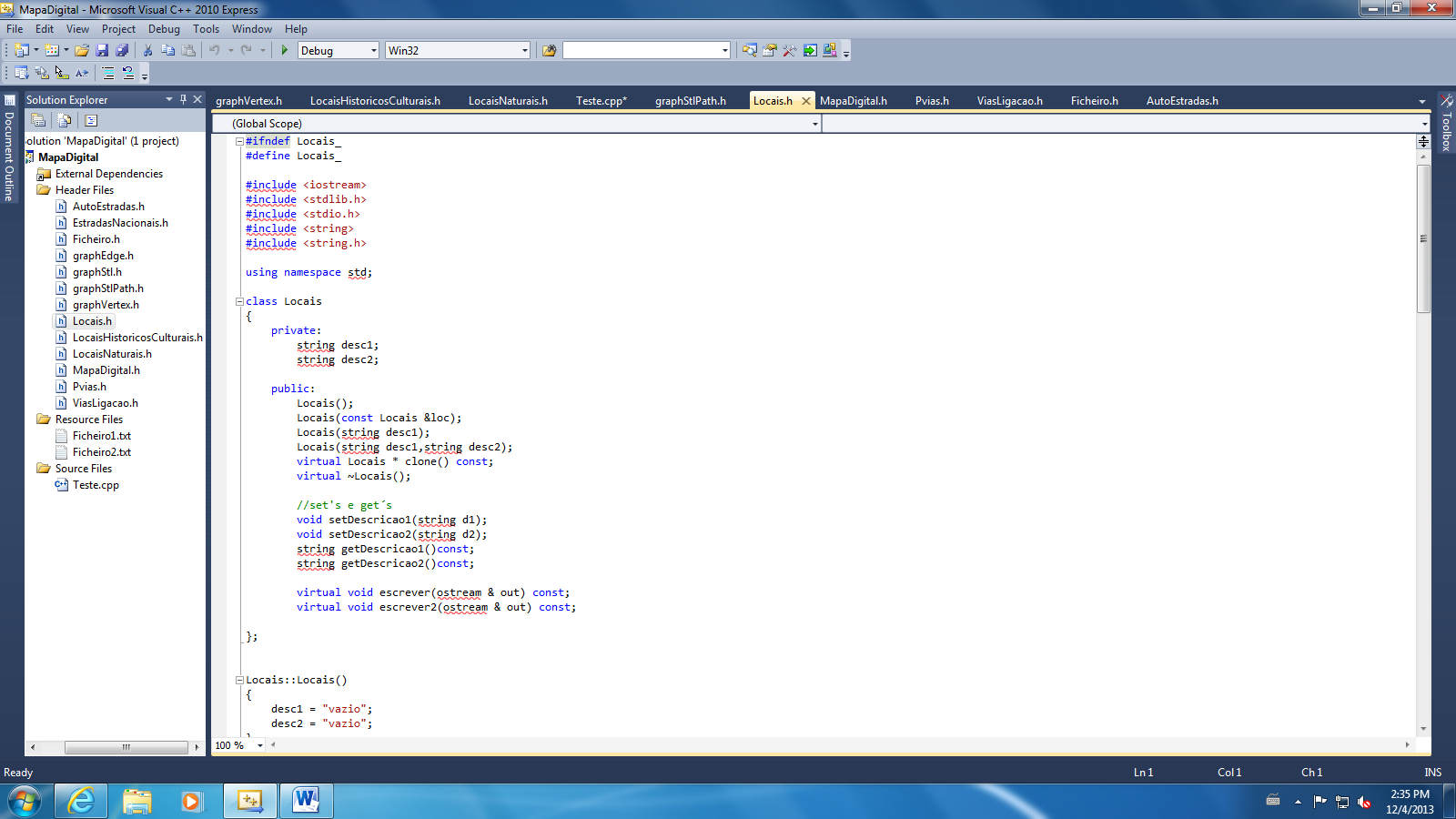
Classe LocaisHistoricosCulturais.h:



Na classe LocaisHistoricosCulturais os métodos mais importantes são os métodos setTempoVisita, setHorarioAbertura, setHorarioEncerramento, getTempoVisita, getHorarioAbertura e getHorarioEncerramento que determinam e retornam, respectivamente, os respectivos valores que as variáveis da classe LocaisHistoricos tomam.

São ainda criados métodos mostraHora que define o formato dos horários anteriormente designados e o método escrever que faz o output de toda a informação relativa aos locais históricos e culturais.

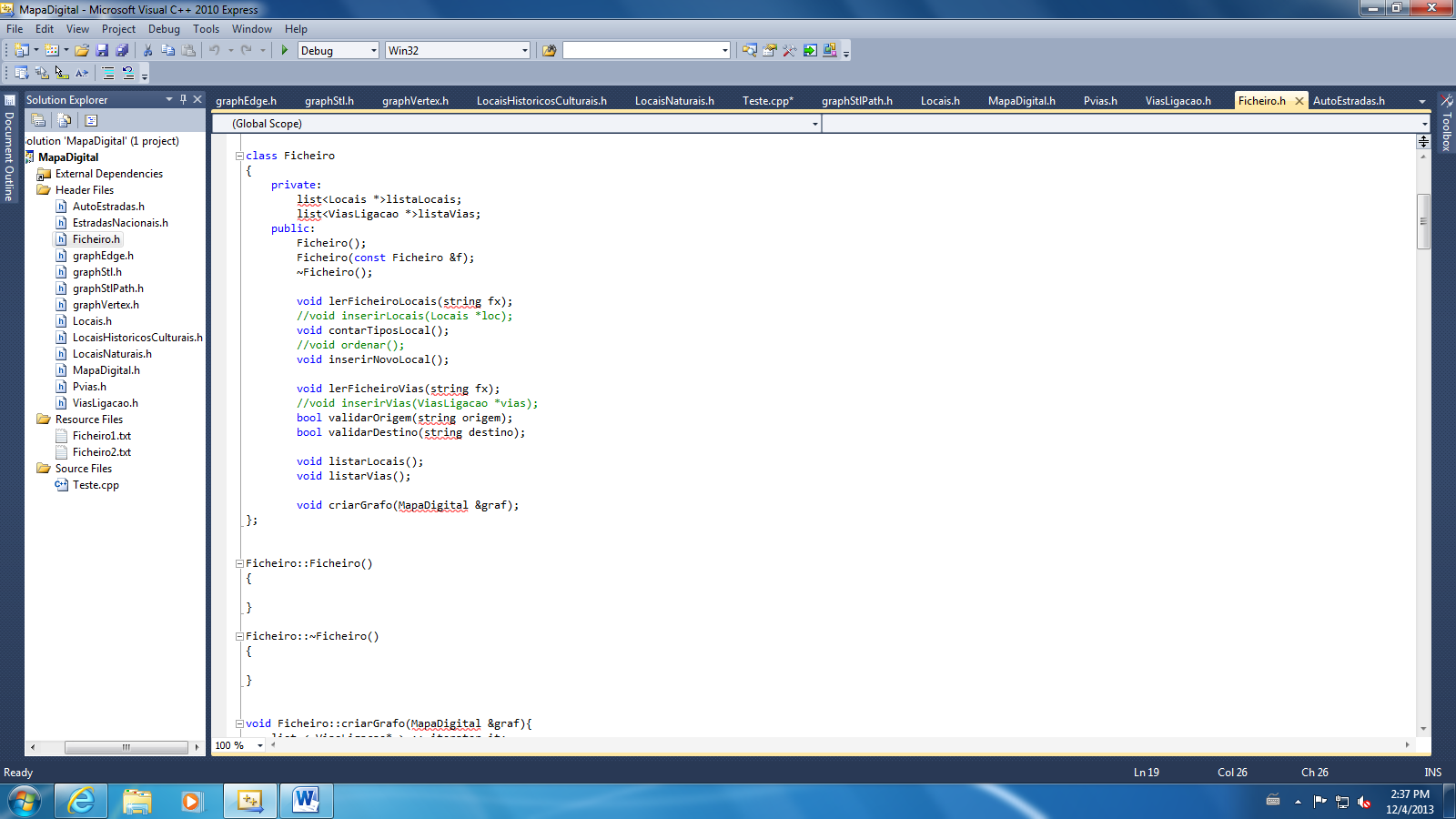
Classe Locais.h:



A classe Locais possui dois métodos virtuais, que lhe permitem fazer o output da descrição ou da origem e destino dos locais, respectivamente através do método escrever().

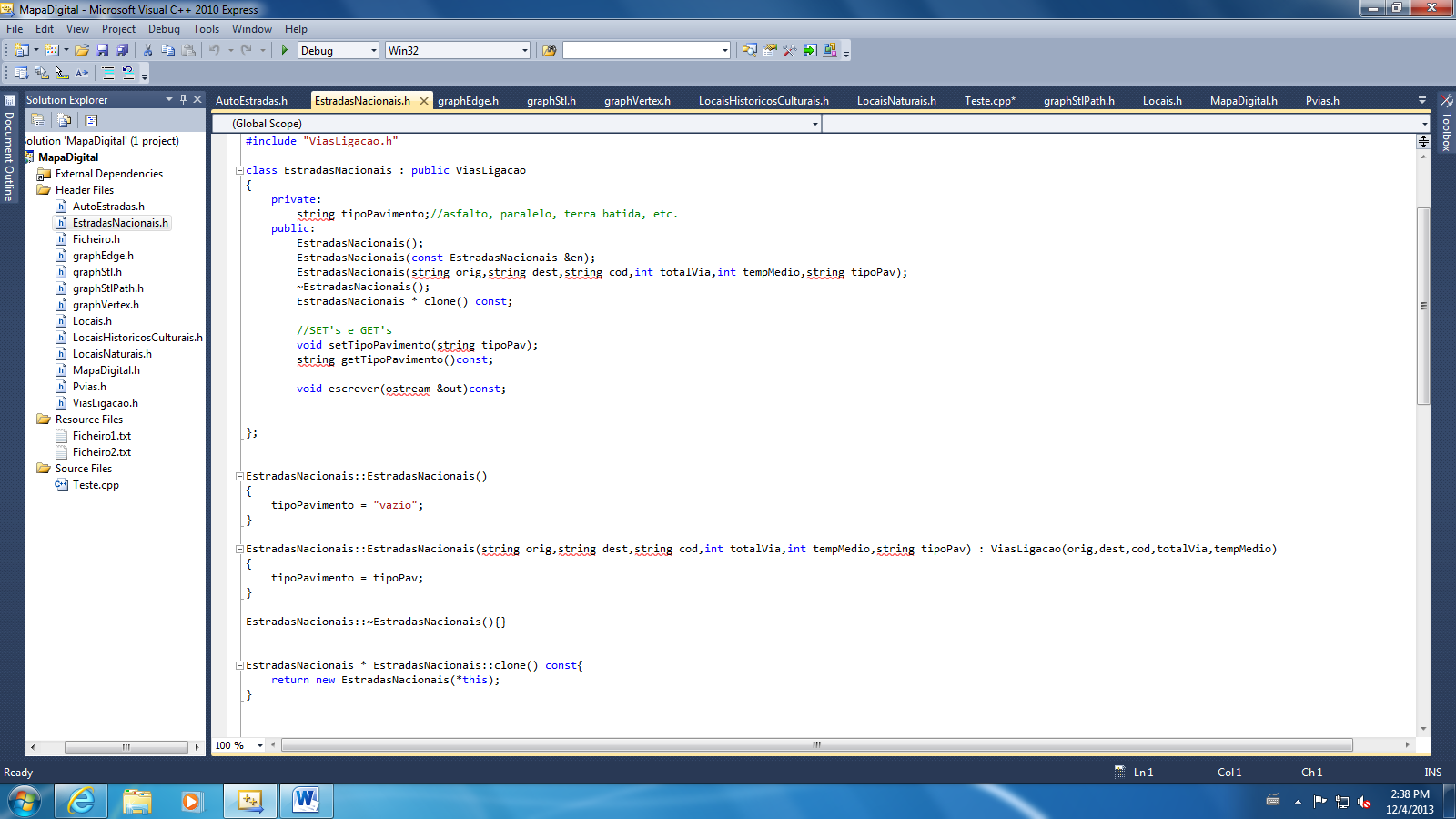
Possui ainda os métodos setDescrição1 e setDescrição2, que determinam a descrição dos locais naturais e dos locais históricos, respectivamente. Possui também os métodos getDescrição1 e getDescrição2 que retornam as descrições dos respectivos sets descritos anteriormente que lhes estão associados.

Classe Ficheiro.h:



A classe Ficheiro é constituída pelos métodos lerFicheirosLocais, que permite obter os locais constates do ficheiro com os locais; pelo método ContarTiposLocal, que conta quantos locais são naturais e quantos locais são culturais; pelo método inserirNovoLocal, que permite ao utilizador inserir um local no ficheiro; pelo método lerFicheiroVias, que abre o ficheiro com a informação sobre as vias de ligação; pelos métodos do tipo booleano ValidarOrigem e ValidarDestino que valida se existe destino e origem de uma via; e os métodos listarLocais e listarVias, que fazem o output das características destas. Foi também incorporado posteriormente o método criarGrafo.

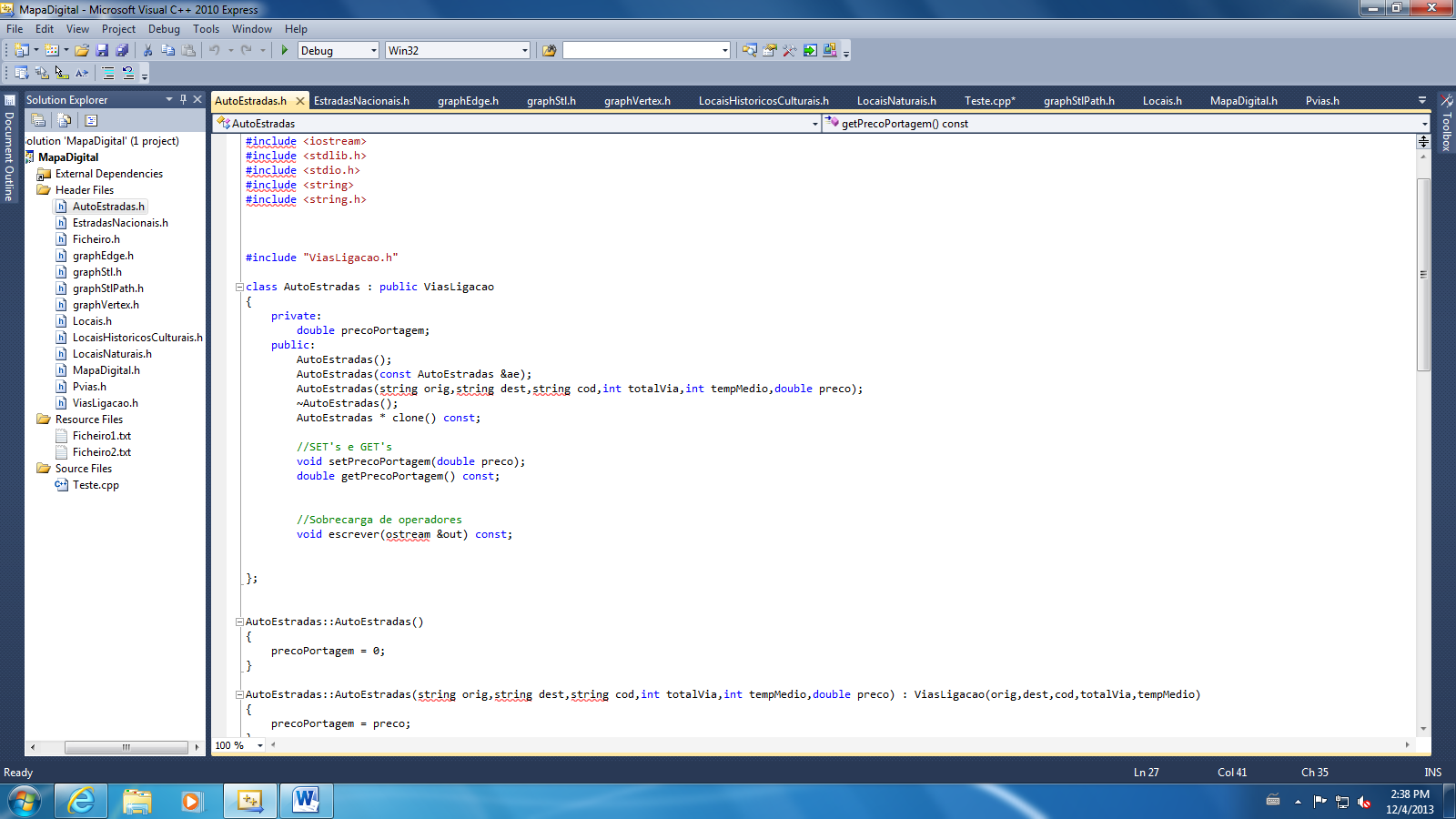
Classe EstradasNacionais.h:



A classe EstradasNacionais é uma subclasse da classe ViasDeLigacao.

Na classe EstradasNacionais o método setTipoPavimento que determina qual o tipo de pavimento para a estrada nacional em causa e um método getTipoPavimento que retorna o se o tipo de pavimento da estrada nacional.

Classe AutoEstradas.h:



A classe AutoEstradas é uma subclasse da superclasse ViasLigacao, nela foi implementado um método getPreçoPortagem que retorna o preço da portagem da autoestrada e um método setPreçoPortagem que deteremina um preço para a portagem da autoestrada

Conclusão

O desenvolvimento do projecto Mapa Digital, permitiu-nos desenvolver as nossas capacidades de trabalho em grupo e de desenvolvimento de projectos na linguagem de programação C++.

O que o resultado final do nosso projecto não foi o esperado, uma vez que alguma funcionalidades pedidas não cumprem os requisitos totais. Na elaboração deste projecto colocamos em prática as regras de programação orientada a objectos que nos foram transmitidas.