

Programação Orientada aos Objetos - Projeto 2018/19

RELATÓRIO

Viagem de alunos do DEI



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D
COIMBRA



Bárbara Vieira – 2017246949

INDICE:

- Guia do Utilizador.....	2
- Estrutura utilizada.....	6
- Métodos mais importantes.....	8
- Conclusão.....	9
- Diagramas (Vão em anexo)	

Guia do Utilizador



FIGURA 1 - TELA DE LOGIN

Quando o utilizador inicia a aplicação, aparece no ecrã uma tela de login. Esta tela pede-lhe para preencher o nome, idade, número de aluno e ainda o seu orçamento para a viagem. Para além disso ainda pede para escolher se é aluno de licenciatura ou de mestrado, dado que o que aparecerá posteriormente será dependente disso. Se o utilizador inserir os seus dados incorretamente (se por exemplo escrever letras onde é suposto estar números e vice versa) ou faltar preencher campos, será mostrada no ecrã uma mensagem de erro.

Caso contrário, se os dados forem inseridos corretamente, será mostrada uma mensagem de sucesso e aparecerá uma nova frame, onde se poderá proceder ao planeamento da viagem.

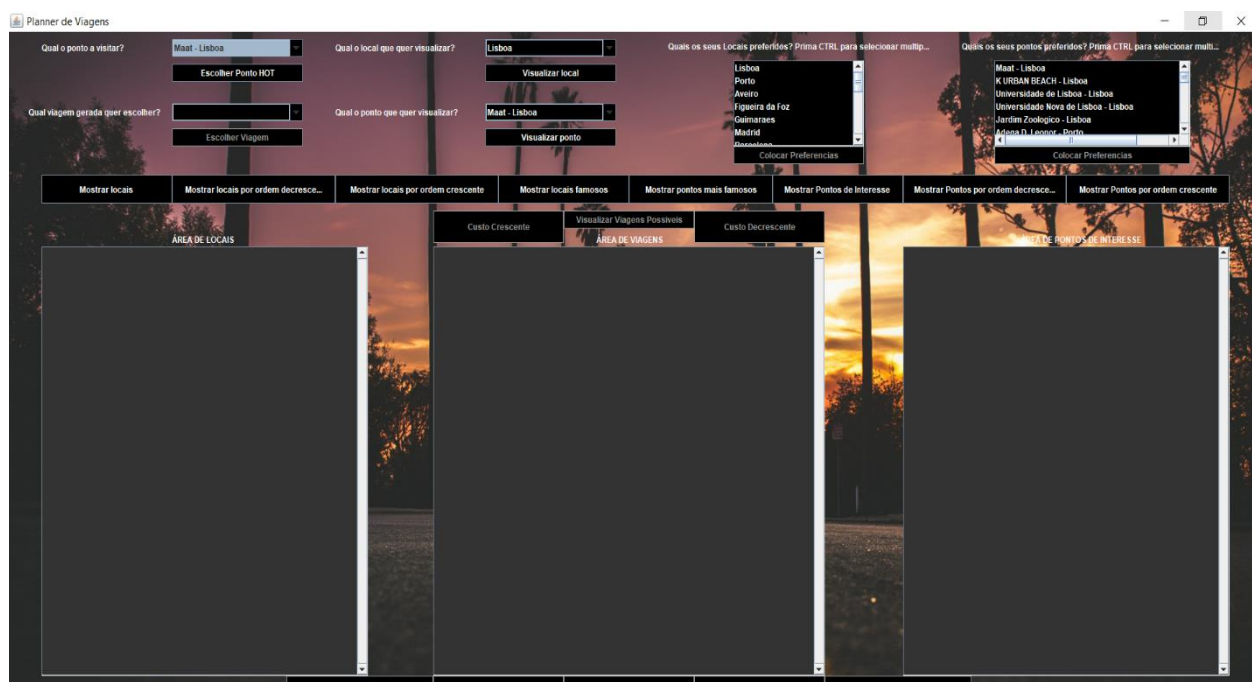


FIGURA 2 - PLANEADOR DE VIAGEM PARA ALUNO DE LICENCIATURA

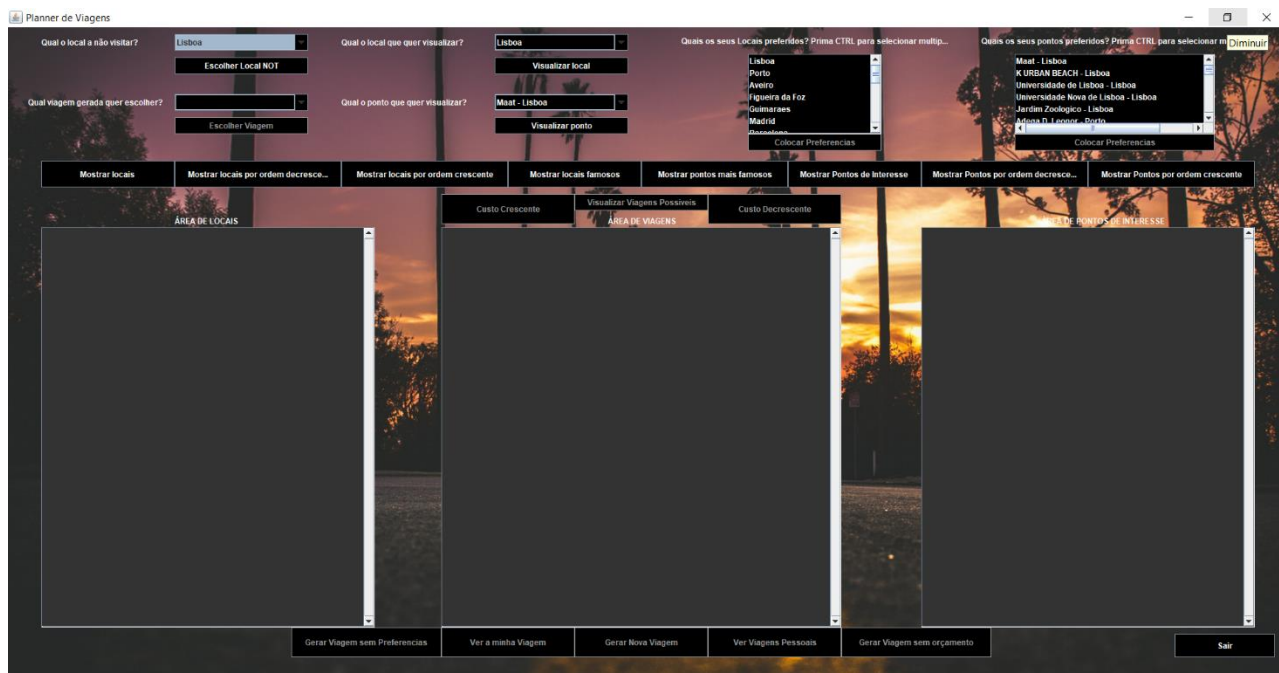


FIGURA 3 - PLANEADOR DE VIAGEM PARA ALUNO DE MESTRADO

Se o utilizador for aluno de Mestrado, na frame aparecerá uma zona para escolher o local a não visitar. Caso contrário, aparecerá uma zona para escolher o ponto de interesse a visitar.

Para além disto, cada utilizador poderá visualizar um local ou um ponto individualmente, ver os locais / pontos mais famosos (tendo em conta as preferências de utilizadores anteriores) e ver todos os locais/pontos por ordem crescente e decrescente de custo.



FIGURA 4 - VER TODOS OS LOCAIS/PONTOS POR ORDEM CRESCENTE E DECRESCENTE DE CUSTO



FIGURA 6 - VISUALIZAR LOCAIS E PONTOS DE INTERESSE

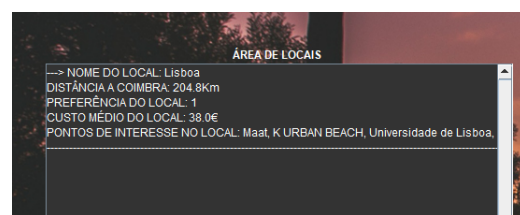


FIGURA 5 - A VISUALIZAR UM LOCAL



FIGURA 7 - A VISUALIZAR UM PONTO DE INTERESSE

Ainda poderá escolher pontos/loais preferidos, gerar viagens segundo preferências e orçamento, gerar viagens ignorando orçamento e gerar viagens ignorando as preferências. Porém, estas opções não estão disponíveis inicialmente, pois dependem das escolhas quanto ao local/ponto a não visitar/a visitar.

Para dar início ao planeamento, o aluno tem de escolher o local a não / ponto a visitar (conforme a sua graduação).

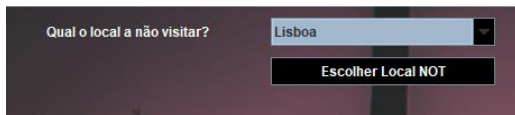


FIGURA 8 - ESCOLHER LOCAL NOT (ALUNO DE MESTRADO)

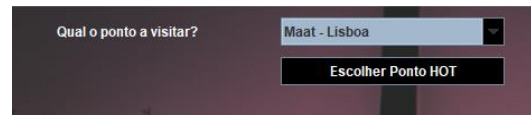


FIGURA 9 - ESCOLHER PONTO DE INTERESSE HOT (ALUNO DE LICENCIATURA)

Após isto ser feito, os botões Colocar Preferências (quanto a locais e pontos de interesse), Gerar Viagem Sem Preferências, Gerar Viagem Sem Orçamento e Gerar Nova Viagem ficam desbloqueados.



FIGURA 11 - BOTÕES PARA COLOCAR PREFERÊNCIAS

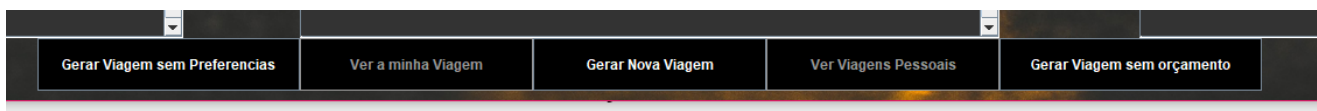


FIGURA 10 - BOTÕES PARA GERAR VIAGEM SEM PREFERÊNCIAS, GERAR NOVA VIAGEM E GERAR VIAGEM SEM ORÇAMENTO

Se o utilizador carregar no botão Gerar Nova Viagem antes de colocar preferências, será mostrada uma viagem dentro do orçamento e que obedeça à condição local/ponto a não visitar/ a visitar. Se o utilizador carregar no botão após colocar preferências, a viagem mostrada no ecrã seguirá então essas preferências, sendo que se o utilizador colocar 1 preferência, a viagem mostrada no ecrã irá a essa localização (se não for igual ao local not para o aluno de mestrado), e a outras duas aleatórias, sempre obedecendo ao orçamento, condição local/ponto a não visitar/ a visitar, e a visitar pelo menos um museu. Se o utilizador colocar 2 preferências, o raciocínio é análogo, mas apenas um local da viagem é aleatório. Se o utilizador selecionar 3 preferências, a aplicação fará todas as combinações de viagem com essas três preferências, sendo que se nenhum dos locais tiver um museu e/ou o orçamento não for obedecido será mostrada uma mensagem no ecrã a avisar o sucedido. Caso o utilizador selecione mais de 3 preferências, o que acontece é semelhante, mas entre essas preferências colocadas são escolhidas 3, sendo que se não existirem combinações possíveis dentro das exigências, será também mostrada uma mensagem no ecrã.

Quando o utilizador recorre ao botão Gerar Nova Viagem pela primeira vez, ou depois de ter seleccionado as preferências, é gerado também um array com todas as viagens possíveis dentro as preferências, orçamento, local/ponto a não visitar/ a visitar e que visitem pelo menos um museu. Esta lista de viagens pode ser visualizada ao carregar no botão Viagens Possíveis e também ao carregar nos botões Custo Crescente e Custo Decrescente, que apresentam cada viagem segundo o seu custo por ordem crescente e decrescente, respetivamente.

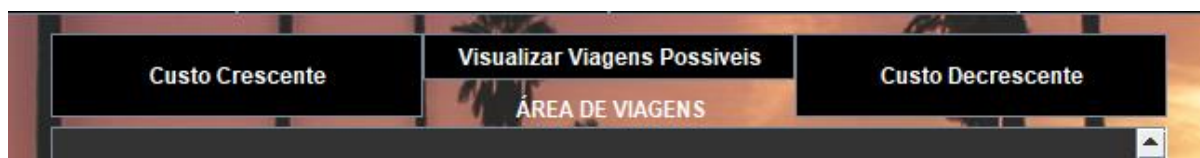


FIGURA 12 - VISUALIZAR VIAGENS POSSÍVEIS , CUSTO CRESCENTE E DECRESCENTE

Para além do botão Gerar Nova Viagem, temos também os botões Gerar Viagem Sem Preferências e Gerar Viagem Sem Orçamento mencionados anteriormente. O primeiro gera uma viagem ignorando as preferências escolhidas (mas incluindo as condições impostas, e tendo em conta o local/ponto a não visitar/ a visitar). O segundo já inclui as preferências, porém ignora o orçamento, gerando assim viagens que não eram possíveis com o orçamento imposto.

Sempre que o utilizador gera uma viagem a partir dos botões mencionados, esta é colocadas numa lista de viagens pessoais. Esta lista pode ser visualizada ao carregar no botão Ver Viagens Pessoais.

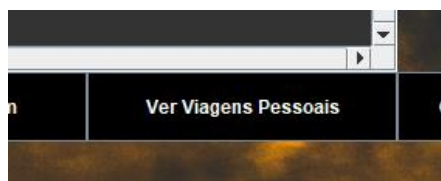


FIGURA 13 - BOTÃO VER VIAGENS PESSOAIS

Geralmente, a "viagem pessoal" do utilizador é a última viagem que este gerou, porém esta pode ser mudada. Para isso, pode ir à opção "Qual a viagem gerada quer escolher?", escolher uma viagem dentro das anteriormente geradas e carregar em Escolher Viagem. Após proceder a isto, pode verificar a mudança ao carregar no botão Ver a Minha Viagem, que apresenta a viagem atual do utilizador.



FIGURA 16 - BOTÃO ESCOLHER VIAGEM

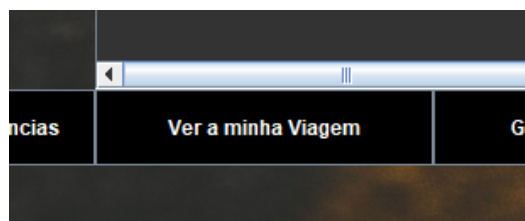


FIGURA 15 - BOTÃO VER A MINHA VIAGEM

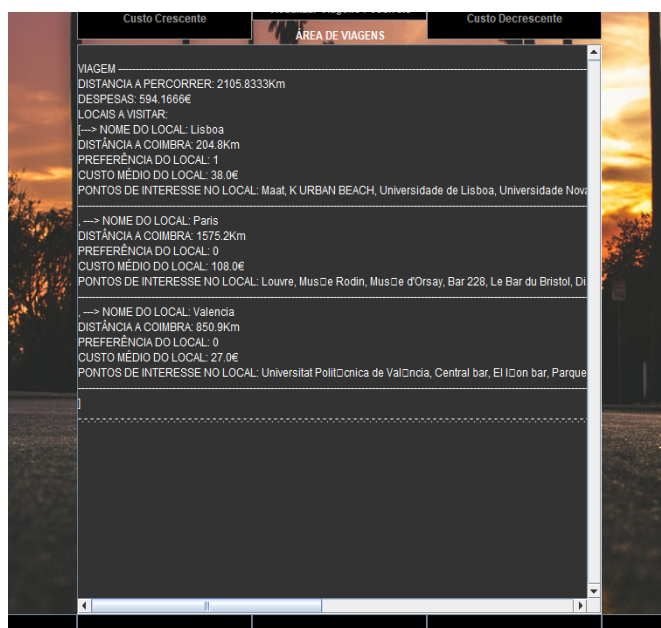


FIGURA 14 - UMA VIAGEM IMPRESSA NO ECRÃ

Se o utilizador quiser sair do planeamento da viagem, poderá carregar no botão Sair. Ao carregar, aparecerá uma janela a perguntar se quer efetivamente sair. Se escolher a opção Sim, aparecerá outra janela a perguntar se quer guardar a sua sessão, incluindo a sua informação, viagem e viagens pessoais.



FIGURA 19 - BOTÃO SAIR

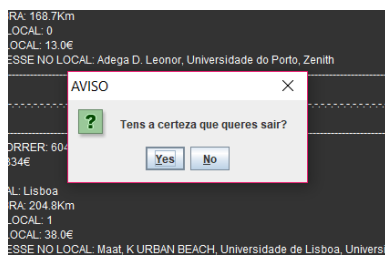


FIGURA 18 - MENSAGEM DE CONFIRMAÇÃO

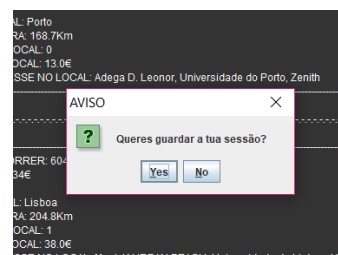


FIGURA 17 - MENSAGEM A PERGUNTAR SE QUER GUARDAR SESSÃO

Estrutura Utilizada

O código do projeto encontra-se dividido entre 20 classes:

- Aluno
- AlunoLicenciatura
- AlunoMestrado
- Bar
- Curso
- Ficheiros
- FrameLogin
- Horario
- JPanelWithBackground
- Local
- Museu
- Parque
- ParqueAquatico
- ParqueCultural
- ParqueDiversoes
- ParqueTematico
- PlanoViagem
- PontoInteresse
- Universidade
- Viagem

sendo que o main se encontra no ficheiro PlanoViagem.java.

Este ficheiro é o que apresenta mais linhas de código, incluindo em si a interface gráfica para o planeamento de viagens e ainda outras funções necessárias para gerar viagens.

A interface gráfica para o login encontra-se no ficheiro FrameLogin.java.

Quanto à estruturação das classes em si, vide o diagrama em anexo.

Para além desta estrutura em classes, são utilizados ficheiros para escrita e leitura.

É utilizado um ficheiro de texto "places2.txt", que contem todas as informações sobre pontos de interesse e locais. Eis o ficheiro:

```
Lisboa
204.8
0
Maat - Museu - Arte e tecnologia - 9:00/20:00 - 0 - 2, K URBAN BEACH - Bar - 18:00/4:00 - 0 - 4.5 - 14, Universidade
de Lisboa - Universidade - Engenharia Informatica ; TI ; Gestão de informação, Universidade Nova de Lisboa -
Universidade - Engenharia Informatica ; Gestão, Jardim Zoológico - Tematico - 15 - 9 - Leoes;Girafas ; Hipotamos ;
Golfinhos;

Porto
168.7
0
Adega D. Leonor - Bar - 18:00/4:00 - 0 - 4.5 - 3, Universidade do Porto - Universidade - Engenharia Informatica ; TI ;
Multimedia ; Engenharia eletrotécnica, Zenith - Bar - 11:00/20:00 - 0 - 4.5 - 10

Aveiro
68.7
0
Costa Nova - Cultural - 0 - 0 - Praia ; Casas, Barba Azul - Bar - 11:00/4:00 - 0 - 4.5 - 49, Universidade de Aveiro -
Universidade - Engenharia Informatica ;Engenharia Telematica ;Engenharia de Telecomunicações
```

Como podem observar começa com o primeiro local, neste caso Lisboa, sendo que as 3 linhas seguintes contêm dados sobre o mesmo. O ficheiro segue a seguinte estruturação:

[NOME DO 1º LOCAL]

[DISTANCIA A COIMBRA]

[PREFERENCIA]

[PONTOS DE INTERESSE DO LOCAL E SUAS INFORMAÇÕES]

[NOME DO 2º LOCAL]

....

Sendo que cada ponto de interesse está estruturado deste modo:

[NOME] – [TIPO] – [OUTRAS INFORMAÇÕES RELATIVAS AO PONTO]

Para além deste ficheiro de texto, ainda existem mais dois ficheiros de objetos: um ficheiro de alunos, "alunos.txt", que guarda todas as informações dos alunos que utilizam a aplicação, e um ficheiro de locais, "locais.txt", que contem toda a informação sobre os locais, incluindo as suas preferências, pontos de interesse e preferências de ponto de interesse. Ambos os ficheiros são lidos ao utilizador fazer login, e escritos quando este decide sair.

Explicação de alguns métodos importantes

Dentro dos métodos mais importantes destacam-se: `geraViagens`, `calculaViagemMestrado`, `calculaViagemLicenciatura`, os métodos para visualizar viagens no ecrã e ainda o método `setPontoPreferido` para as classes `AlunoLicenciatura` e `AlunoMestrado`.

geraViagens

Quanto ao método `geraViagens`, este vai ser aquele que gerará os arrays de viagens possíveis dentro das opções que o utilizador escolheu.

Este método recebe um array, que será o nosso array de viagens. Ao entrar nele, entramos num ciclo condicionado pela variável `acum`, sendo que a cada iteração do ciclo é verificado se o aluno é de mestrado ou de licenciatura. Se for de Mestrado, entrará no método `calculaViagemMestrado`, que recebe também o array de viagens, caso contrário, entrará no método `calculaViagemLicenciatura`, que tal como o método anterior, recebe o array de viagens. Estes métodos têm como função gerar uma viagem (ou não) e adicioná-la ao array, tanto que, se não for gerada nenhuma (por não ter sido obedecida a condição do orçamento ou outra), nada será adicionado ao array. Se então o array de viagens não mudou o seu comprimento (nenhuma viagem foi gerada nessa iteração), é adicionado +1 à variável `acum`, que fará o ciclo parar ao atingir 2000 (o que quer dizer que muito provavelmente já não haverá mais nenhuma viagem a gerar). Se o array mudou o seu comprimento, a variável `acum` é colocada a 0.

Ao sair do ciclo verificamos se o array de viagens está vazio ou não. Se não estiver, calculamos um índice aleatório entre 0 e o comprimento do array para escolhermos uma viagem aleatória dentro das geradas e recorremos à função `setViagem` (que pertence a classe `Aluno`) para a tornar a presente viagem do utilizador. Se esta viagem nunca tiver sido atribuída ao utilizador é também adicionada à combobox `escolher`, que servirá para o utilizador mudar de viagem.

calculaViagemMestrado e calculaViagemLicenciatura (e métodos chamados por eles)

Estes dois métodos são praticamente iguais, apenas diferindo nos métodos que chamam. Dado que os alunos de mestrado e alunos de licenciatura têm condições diferentes no que toca ao planeamento de viagens, foi necessário a criação de métodos diferentes. O método `calculaViagemMestrado`, chama os métodos `viagemMestradoTresOuMaisFavoritos`, `calculaViagemMestradoSemPreferencias` e `viagemMestradoMenosDeTresFavoritos`, tendo estes em conta o número de locais preferidos que o utilizador tem. Já o método `calculaViagemLicenciatura`, chama os métodos `viagemLicenciaturaTresOuMaisFavoritos`, `calculaViagemLicenciaturaSemPreferencias` e `viagemLicenciaturaMenosDeTresFavoritos`, que seguem um comportamento parecido aos de mestrado, diferindo apenas nas condições. Dentro destes métodos, é seguido um algoritmo para verificar as condições impostas, sendo que se estas forem cumpridas é gerada uma nova viagem que é adicionada ao array de viagens.

Métodos para visualizar viagens no ecrã

Outros métodos de considero importantes são os métodos e algoritmos associados à impressão de viagens do ecrã. Quando um utilizador carrega pela primeira vez nos botões de gerar viagens (Gerar Nova Viagem, Gerar Viagem Sem Orçamento e Gerar Viagem Sem Preferências), ou carrega nos dois primeiros botões pela primeira vez após colocar preferências, é chamado o método `geraViagens` mencionado anteriormente, e é impressa no ecrã a viagem que foi atribuída ao utilizador (caso tenha sido atribuída alguma). Caso o utilizador não esteja a carregar pela primeira vez é apenas escolhida aleatoriamente uma das viagens anteriormente geradas e colocadas no array respetivo (há um array de viagens tendo em conta preferências e orçamento, outro que ignora o orçamento, e outro que ignora as preferências), sendo essa viagem atribuída ao utilizador e mostrada do ecrã.

É de notar que o algoritmo para ignorar as preferências consiste na cópia do array de locais preferidos de um aluno para outro array temporário, seguido com a remoção de todos os elementos do array de locais preferidos do aluno. Após isto, é chamado o método `geraViagens`, que atualiza o array de viagens possíveis ignorando as preferências e atribui ao utilizador uma dessas viagens, imprimindo-a no ecrã. No final deste algoritmo, é copiado para o array de locais preferidos do aluno os locais preferidos iniciais, que se encontravam no array temporário.

Quanto ao algoritmo para ignorar o orçamento, este é semelhante ao anterior, no entanto é o orçamento inicial que é guardado numa variável temporária, colocado a 10000, e no final é novamente reposto.

setPontoPreferido para as classes `AlunoLicenciatura` e `AlunoMestrado`

Por fim, um outro método, que acho importante mencionar é o `setPontoPreferido`. Quando um utilizador coloca uma nova preferência no que toca a pontos de interesse este método é chamado. Porém, tendo em conta as condições impostas (o Aluno de Mestrado tem de escolher um local que não quer visitar), este difere entre os tipos de aluno. Para o aluno de Licenciatura é trivial. É aumentada a preferência do ponto, seguida de uma verificação se o local onde o ponto se encontra já é um local preferido. Se já se encontrar, o ponto é apenas adicionado aos pontos preferidos. Se não se encontrar, o local é adicionado aos preferidos, bem como o ponto de interesse.

Quanto ao aluno de mestrado, o raciocínio é semelhante , apenas com uma diferença. Para além da verificação do local se encontrar nos locais preferidos , também é verificado se o local é o local not. Se for, o local não é adicionado aos preferidos.

Conclusão

Para concluir, tenho a dizer que gostei bastante de realizar este projeto. Permitiu melhorar muito as minhas capacidades a programar java, bem como a ganhar o gosto a esta linguagem de programação.