

# LISBON GUIDE

---

## Métodos de Desenvolvimento de Software 2010-2011

### **Realizado por:**

Catarina Albino, Nº 31794

Luís Silva, Nº 31698

Sara Gonçalves, Nº 31551

Turno Prático 1

**Professor: Vasco Amaral**

**27-05-2011**

## Índice

Introdução.....	2
Diagrama de Casos de Uso.....	4
Caso de Uso Efectuar Visita.....	5
Caso de Uso Interromper Visita.....	6
Caso de Uso Marcar Visita.....	7
Caso de Uso Personalizar Visita.....	8
Caso de Uso Obter Informação Local.....	10
Diagrama de Actividades.....	12
Diagrama de Actividades Efectuar Visita.....	12
Diagrama de Actividades Interromper Visita.....	12
Diagrama de Actividades Marcar Visita.....	13
Diagrama de Actividades Personalizar Visita.....	14
Diagrama de Actividades Obter Informação Local.....	15
Diagrama de Sequência.....	17
Diagrama de Sequência Efectuar Visita.....	17
Diagrama de Sequência Interromper Visita.....	17
Diagrama de Sequência Marcar Visita.....	18
Diagrama de Sequência Personalizar Visita.....	19
Diagrama de Sequência Obter Informação Local.....	20
Diagrama de Classes.....	21
OCL.....	22
Diagrama de Estados.....	23
Diagrama de Estados Efectuar Visita.....	23
Diagrama de Estados Marcar Visita.....	24
Diagrama de Estados Obter Informação Local.....	25
Diagrama de Pacotes Global.....	26
Lista de Requisitos Não Funcionais.....	27
Diagrama de instalação.....	28
Diagrama de componentes.....	29
Impacto dos Requisitos Não Funcionais.....	30
Conclusão.....	31
Anexos.....	32

## Introdução

Este trabalho consiste no desenho do sistema Lisbon Guide para o Turismo de Lisboa, que permite aos turistas explorar a cidade através de visitas guiadas aos pontos mais interessantes, ou do seu agrado, da cidade. Para tal o sistema oferece visitas guiadas pré-definidas bem como visitas guiadas moldadas aos gostos dos utilizadores.

Existem alguns aspectos que não estão claramente definidos no enunciado e como tal, permitem uma interpretação ambígua. Assim, apresentamos abaixo uma lista de ambiguidades e as respectivas opções tomadas:

- **Utilizador não registado pode ou não efectuar visitas:** optámos por disponibilizar a um utilizador não registado a hipótese de efectuar uma visita, tendo como restrição o facto de a mesma ter de ser pré-definida.
- **Um utilizador não registado ser capaz de personalizar uma visita:** esta opção não é disponibilizada pelo nosso sistema. Apenas um utilizador registado tem hipótese de personalizar, guardar e aceder a visitas previamente personalizadas, que estão disponíveis através da sua área pessoal, acessível usando um mecanismo de autenticação (*username* e *password*).
- **Possibilidade de remover visitas:** o utilizador registado pode remover visitas da sua “área pessoal” sendo estas por exemplo, visitas já efectuadas ou visitas previamente personalizadas por ele.
- **Possibilidade de um utilizador guardar a sua visita, e interromper esta com o fim de posteriormente a continuar:** disponibilizamos esta opção com o intuito de conferir uma maior flexibilidade ao utilizador no decorrer da visita, já que, por motivos diversos, o utilizador pode pretender interromper a visita e retomá-la mais tarde. Nesta situação o utilizador grava-a e ficando a mesma disponível para posterior continuação.
- **Uma visita ficar aberta mais de 24 horas, em que a respectiva sessão é limpa do sistema:** por motivos de optimização de espaço, definimos um prazo máximo durante o qual uma visita interrompida pode ficar aberta, neste caso 24 horas. Desta forma evitam-se
- **Obter informação de linhas de transporte:** é obtido o itinerário actual, e será a partir desse conjunto de pontos que são gerados os transportes precisos para a deslocação de um ponto para outro. Desta forma optou-se por não dar a oportunidade ao utilizador de escolher o destino, mas sim informá-lo dos transportes necessários para acompanhar o itinerário planeado.

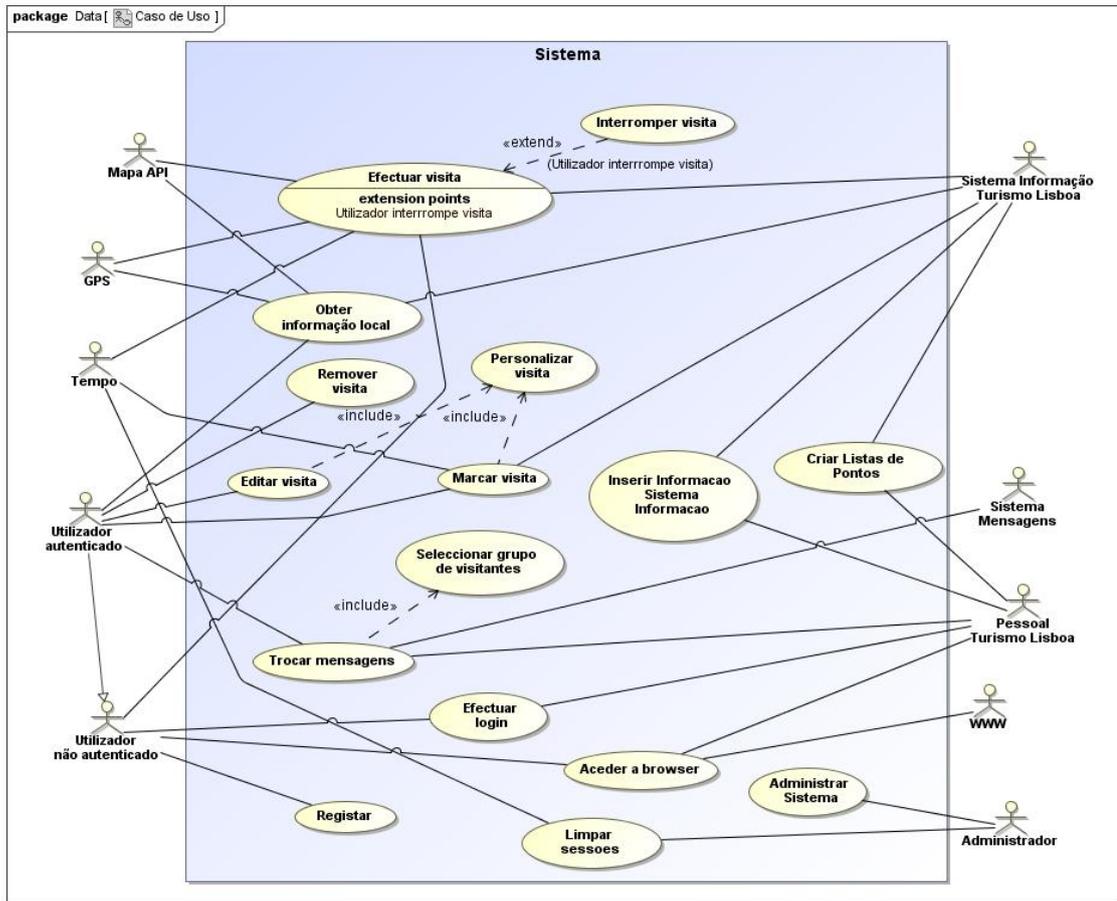
- **Que informação mostrar no caso de um ponto actual não estar previsto na visita:** Numa situação como esta, é apresentada informação contextual tal como o nome da rua.
- **Carregar informação no início de efectuar uma visita:** como o sistema teria de estar sempre conectado ao *wireless*, optou-se por agir desta forma, com vista a assegurar uma maior eficiência.

As três funcionalidades que consideramos ter mais relevância para o funcionamento do sistema são as que permitem ao utilizador marcar uma visita, efectuar uma visita e obter informação local no decorrer da visita. Como tal, serão as abordadas com detalhe neste trabalho. A cargo de Catarina Albino ficou o caso de uso “Marcar visita”, a cargo de Luís Silva o caso de uso “Efectuar visita” e a cargo de Sara Gonçalves o caso de uso “Obter Informação Local”.

Apresentamos de seguida uma lista com uma descrição sumária de alguns casos de uso cuja funcionalidade achamos ser necessário explicitar:

- **Limpar sessões:** este caso de uso é responsável pela destruição de sessões de utilizadores em sistema. Esta operação pode ser realizada pelo administrador manualmente (que tem privilégios de manutenção do sistema) mas está igualmente automatizada: de certo em certo tempo o próprio sistema procede à eliminação das sessões.
- **Seleccionar grupo de visitantes:** este caso é incluído no caso de uso “Trocar mensagens” e permite aos actores intervenientes, neste caso o Pessoal de Turismo e os utilizadores autenticados, definirem um grupo de pessoas com quem trocar mensagens. Esta funcionalidade foi assegurado de modo a facilitar a comunicação entre visitantes para, por exemplo, marcação de encontros.
- **Remover visita:** o utilizador registado pode remover visitas previamente personalizadas por ele da sua área pessoal, sendo estas eliminadas do sistema.
- **Editar visita:** é semelhante ao personalizar visita, mas permite, por exemplo, que depois de uma visita estar terminada o utilizador possa redefinir a data / hora e marcá-la novamente.

# Diagrama de Casos de Uso



## Caso de Uso Efectuar Visita

**Descrição:** Ajuda o utilizador durante a visita, oferecendo-lhe a informação necessária para a sua movimentação de um local para outro, mostrando o ponto mais próximo a visitar e o caminho para lá chegar. Primeiramente o utilizador tem de escolher a visita que pretende efectuar, onde pode escolher uma visita predefinida ou uma editada.

Se o utilizador se afastar do percurso estabelecido, o sistema adapta-se e, se for caso, altera inclusive o percurso, pois o utilizador pode-se ter aproximado de outro ponto a visitar.

A geração dos próximos pontos turísticos tem sempre em conta a localização actual do utilizador, bem como a criação dos percursos tem em conta percursos “com sentido”, ou seja, percursos onde o utilizador não ande para trás e para a frente entre pontos turísticos. Os percursos entre pontos turísticos são sempre os mais curtos.

**Pré-condição:** Inicialmente encontrar-se ligado ao sistema de informação do Turismo Lisboa e a um sistema de Localização.

**Pós-condições:** Sessão terminada.

**Actores:** Utilizador não autenticado, GPS, Mapa API, Sistema de Informação Turismo Lisboa e Tempo.

### Cenário principal:

1. O caso de uso começa quando o utilizar inicia a visita
2. Se o utilizador pretender escolher uma visita editada
  - a. <<Include>> Efectuar Login
  - b. Mostrar visitas editadas pelo utilizador
3. Senão
  - a. Mostrar visitas predefinidas
4. O utilizador escolhe uma visita a efectuar
5. O sistema carrega toda a informação referente aos pontos turísticos da visita
6. Em cada ti segundos, o GPS obtêm as coordenadas da presente localização
7. O sistema define os próximos pontos turísticos
8. O sistema pede ao Mapa API caminhos para chegar aos próximos pontos turísticos
9. O sistema gera os próximos percursos a fazer entre pontos turísticos
10. O sistema indica o caminho para o próximo local e mostra a sua descrição
11. O caso de uso termina

### **Cenários secundários:**

#### Comunicação com Mapa API não estabelecida

- a. O sistema detecta impossibilidade de comunicação
- b. Mostra mensagem a alertar o utilizador
- c. Em cada ti segundos tenta restabelecer conexão
- d. Se não conseguir restabelecer a conexão
  1. O sistema mostra os últimos percursos entre pontos turísticos e suas descrições
  2. Continua ponto c.
- e. Continua no passo 10

**Extension Point:** Passo 8

#### Percursos desconhecido

- a. O sistema detecta que o utilizador não está a seguir o percurso definido
- b. O sistema alerta o utilizador
- c. Continua no passo 6

**Extension Point:** Passo 6

## **Caso de Uso Interromper Visita**

**Descrição:** Durante uma visita o utilizador pode decidir salvar a visita para posteriormente continuar. Este caso de uso permite ao utilizador este objectivo, ou simplesmente terminar a visita sem a guardar.

**Pré-descrição:** Utilizador registado se pretender guardar a visita.

**Atores:** Utilizador não autenticado.

### **Cenário principal:**

1. O caso de uso começa quando o utilizar pretende interromper uma visita
2. Se o utilizador pretende guardar a visita
  - a. <<Include>> Efectuar Login
  - b. Guardar estado da visita actual
3. O caso de uso termina

## Caso de Uso Marcar Visita

**Descrição:** Este caso de uso permite a um utilizador autenticado marcar uma visita pré-definida escolhendo-a de entre um conjunto de visitas pré-definidas já existentes ou personalizar uma visita consoante as suas preferências. Essa visita pode ser uma já editada previamente ou apenas a edição de uma visita pré-definida. É disponibilizada como visita pré-definida uma visita “vazia”, sem pontos turísticos associados, para dar a oportunidade ao utilizador de poder construir uma visita “de raiz”.

Quando a marcação da visita é terminada, caso seja não vazia, o sistema guarda-a para que possa ser posteriormente acedida (com o objectivo de ser efectuada).

**Actores:** Utilizador autenticado, Sistema Informação Turismo Lisboa

### Pré-condições:

- O utilizador é um utilizador registado e efectuou login

### Cenário Principal:

1. O caso de uso começa quando o utilizador escolhe marcar visita
2. O utilizador escolhe o tipo de visita que pretende marcar
3. Se o utilizador escolhe uma visita já editada
  - 3.1. <include> Personalizar visita
4. Se o utilizador escolher uma visita pré-definida
  - 4.1. O sistema conecta-se com o Sistema de Informação do Turismo de Lisboa e obtém as visitas pré-definidas
5. Enquanto o utilizador não terminar a marcação da visita pretendida:
  - 5.1. O utilizador selecciona uma das visitas pré-definidas
  - 5.2. O sistema conecta-se com o Sistema de Informação do Turismo de Lisboa e obtém todos os pontos turísticos (e respectivas descrições) que compõem a visita escolhida
  - 5.3. O sistema mostra os pontos turísticos e as respectivas descrições
  - 5.4. Se o utilizador pretender personalizar a visita:
    - 5.4.1.<include> Personalizar visita
6. O sistema grava a visita.
7. O caso de uso termina.

### Cenários Secundários:

#### “Marcação de uma visita sem pontos turísticos”

1. O utilizador selecciona uma visita vazia e termina marcação

2. O sistema envia uma mensagem de erro ao utilizador alertando-o que a visita não contém pontos turísticos
3. Retorna ao ponto 5.

**Extension Point:** Depois do passo 5.

#### “Falha na ligação ao Sistema de Turismo de Lisboa”

1. O sistema não consegue efectuar ligação com o Sistema de Informação do Turismo de Lisboa
2. O sistema envia uma mensagem alertando o utilizador
3. A cada ti segundos o sistema reconecta-se novamente com o Sistema de Turismo de Lisboa, até que a conexão seja bem sucedida
4. Retorna ao ponto 5.2.

**Extension Point:** Após o passo 5.2.

## Caso de Uso Personalizar Visita

**Descrição:** permite a um utilizador autenticado personalizar uma visita introduzindo um conjunto de preferências (culturais, arquitectónicas, lazer, etc.) e conjunto de restrições (temporais, por exemplo). O sistema apresenta, com base nessas informações, os pontos turísticos que mais se adaptam a elas. O utilizador pode adicionar / remover esses pontos sugeridos da visita.

O utilizador pode introduzir restrições temporais, mas apesar disso a visita não é restringida em função do tempo. É dada uma estimativa da duração da visita com o objectivo de o utilizador ter uma melhor noção do tempo que o percurso gerado pelo sistema (com base nos pontos seleccionados) vai demorar a ser percorrido.

#### **Pré-condições:**

- O utilizador é um utilizador registado e efectuou login

**Actores:** Utilizador Autenticado, Sistema Informação Turismo Lisboa

#### **Cenário Principal:**

1. O caso de uso inicia quando o utilizador escolhe personalizar uma visita
2. Enquanto o utilizador não terminar a personalização da visita:
  - 2.1. O utilizador introduz no sistema as preferências e restrições para a visita

- 2.2. O sistema conecta-se com o Sistema de Informação do Turismo de Lisboa e obtém todos os pontos turísticos (e respectivas descrições) que correspondam às preferências do utilizador
- 2.3. O sistema mostra os pontos turísticos e as respectivas descrições
- 2.4. O utilizador selecciona um ou mais pontos turísticos disponíveis
- 2.5. Se o utilizador pretender adicionar esses pontos:
  - 2.5.1. O sistema adiciona esses pontos à visita
- 2.6. Se o utilizador pretender remover esses pontos turísticos:
  - 2.6.1. O sistema remove esses pontos da visita
- 2.7. Se o utilizador pretender gravar visita:
  - 2.7.1. O sistema grava a visita.
3. O caso de uso termina.

### **Cenários Secundários:**

#### “Falha na ligação ao Sistema de Turismo de Lisboa”

1. O sistema não consegue efectuar ligação com o Sistema de Informação do Turismo de Lisboa
2. O sistema envia uma mensagem alertando o utilizador
3. A cada ti segundos o sistema reconecta-se novamente com o Sistema de Turismo de Lisboa, até que a conexão seja bem sucedida
4. Retorna ao ponto 2.2.

**Extension Point:** Após o passo 2.2.

## Caso de Uso Obter Informação Local

**Descrição:** Este caso de uso permite ao utilizador autenticado obter informação geral da cidade, incluindo informação sobre o local onde se encontra actualmente, horários de funcionamento dos pontos turísticos, linhas de transporte de e para um local.

**Actores:** Utilizador não autenticado, GPS, Mapa API, Sistema Informação Turismo Lisboa

### Cenário Principal:

- 1) O UC inicia quando o utilizador autenticado deseja obter informação local.
- 2) O sistema obtém as coordenadas da posição actual do utilizador a partir do GPS.
- 3) É verificado se o utilizador se encontra num ponto previsto da visita.
  - 3.1) Caso não se verifique o sistema conecta-se com o Sistema de Informação Turismo Lisboa para ser feito o load da informação do ponto actual.
- 4) Enquanto o utilizador não tomar a decisão de terminar a obtenção de informação.
  - 4.1) Se optar por se informar sobre as coordenadas da posição actual
    - 4.1.1) O Mapa API gera o mapa consoante as coordenadas actuais, indicando onde se encontra o utilizador.
    - 4.1.2) O mapa é divulgado ao utilizador.
  - 4.2) Se optar por se informar a nível contextual sobre o local onde se encontra
    - 4.2.1) É divulgado ao utilizador a informação contextual relacionada com o local.
  - 4.3) Se optar por se informar sobre as linhas de transporte
    - 4.3.1) O sistema obtém o percurso previsto (itinerário).
    - 4.3.2) O Mapa API gera o mapa consoante esse itinerário.
    - 4.3.3) O mapa é divulgado ao utilizador.
  - 4.5) Se optar por se informar sobre horários de funcionamento dos pontos turísticos.
    - 4.5.1) O sistema acede à informação carregada no início da visita.
    - 4.5.2) A informação é divulgada ao utilizador.
  - 4.6) Se optar por não obter mais informações o UC termina.

### Cenários Secundários:

#### “Ausência sinal GPS”

- 1) O Sistema não obtém as coordenadas via GPS.
- 2) A mensagem de ausência de sinal é enviada ao utilizador.
- 3) O UC termina.

**Extension Points:** Ponto 2.

#### “Falha na conexão ao wireless ”

- 1) O Sistema não consegue conectar-se ao Sistema de Informação Turismo Lisboa ou ao Mapa API.
- 2) É enviada a mensagem ao utilizador informando-o que nesse momento não é possível obter a desejada informação local.
- 3) O UC termina.

**Extension Points:** Ponto 3.

## Diagrama de Actividades

### Diagrama de Actividades Efectuar Visita

(Imagem em anexo)

Sendo que um utilizador não registado pode efectuar uma visita, inicialmente ele tem de explicitar se pretende carregar uma visita previamente editada. Após o utilizador escolher uma visita, por motivos de eficiência, toda a informação referente a esta visita é carregada para o TabletPC. No final deste processo, a visita é finalmente iniciada, sendo que em cada *ti* segundos o sistema actualiza a sua localização e adapta-se de acordo com esta.

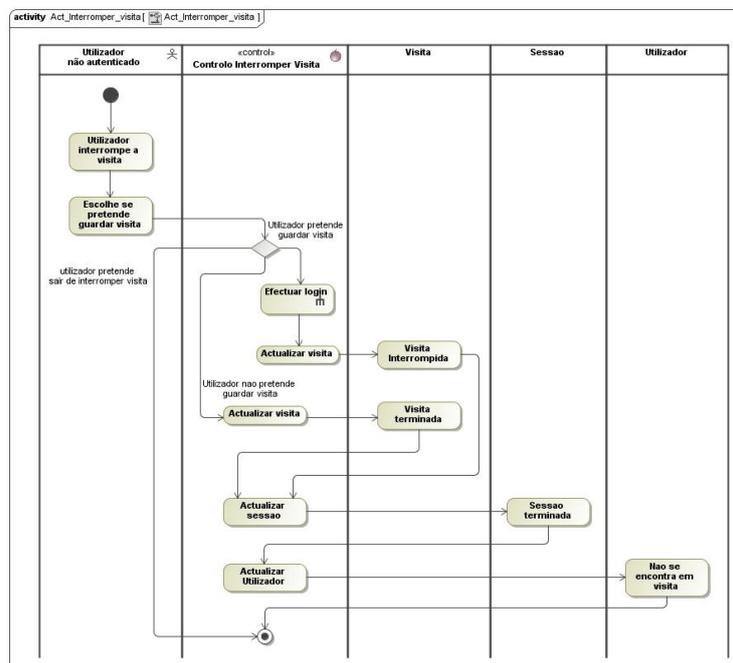
Todo o processo de *context sensitive*, está envolvido por uma região crítica para o sistema conseguir contornar facilmente problemas como por exemplo, o utilizador se desviar do percurso disponibilizado ou a falha de comunicação com os sistemas externos.

Por fim, também ela como *exception handler*, o utilizador pode interromper a visita, para ou termina-la ou simplesmente para interromper, com o objectivo de a continuar.

Foi considerado que os resultados devolvidos pelo Mapa API, estão de acordo com as necessidades do sistema e prontos a ser disponibilizado ao utilizador.

### Diagrama de Actividades Interromper Visita

Devido a estrutura que utilizamos para modelar este problema e ao que o utilizador pretende efectuar no interromper visita, o sistema tem de actualizar, de forma diferente, varias classes. Mantendo assim a boa consistência do sistema e por consequente um log de todos as visitas efectuadas pelos utilizadores e seus atributos.

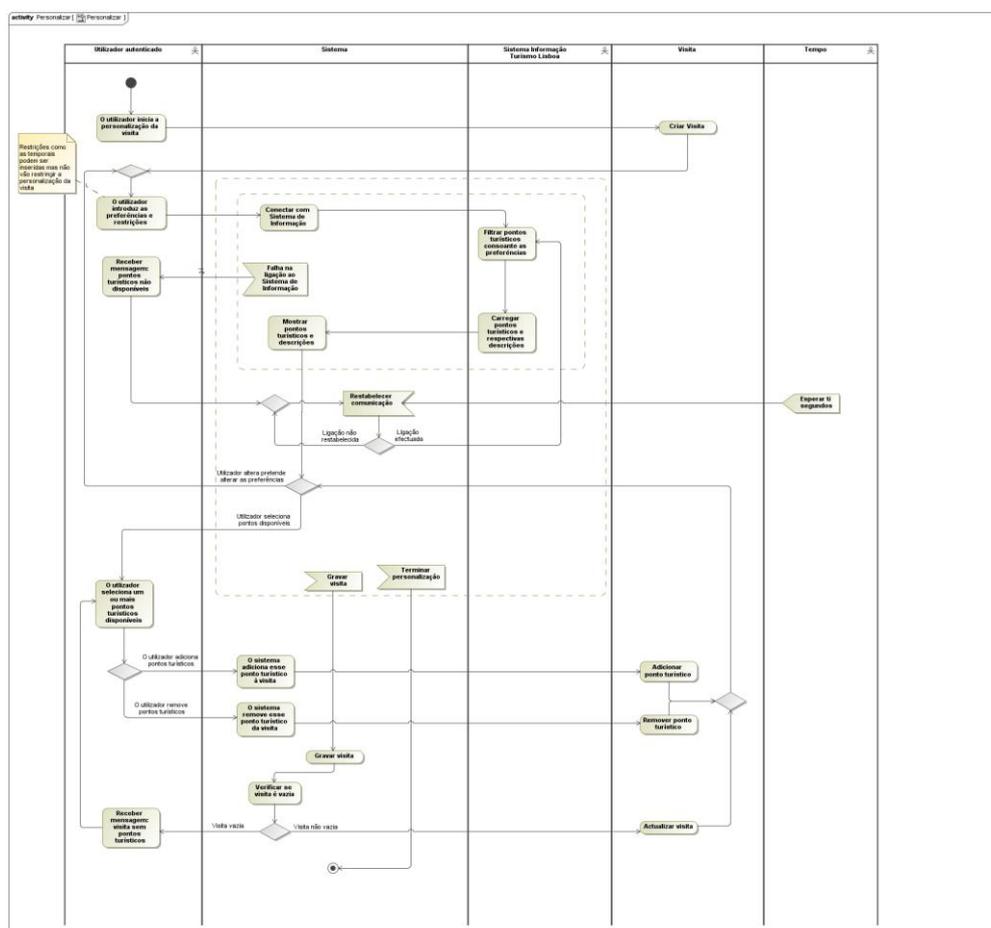




este escolha outra visita (ou escolha a visita vazia e a personalize). Caso contrário é criado um objecto Visita, contendo o conjunto de pontos correspondentes, e este é gravado.

O caso secundário “Falha na ligação ao Sistema de Turismo de Lisboa” foi descrito por meio de uma excepção já que a impossibilidade na conexão tanto pode acontecer inicialmente, ou no retorno da informação pedida. É então enviada uma mensagem ao utilizador informando-o da situação e de ti em ti segundos, a conexão tenta ser restabelecida. Quando o for, são obtidas novamente as informações pretendidas.

## Diagrama de Actividades Personalizar Visita

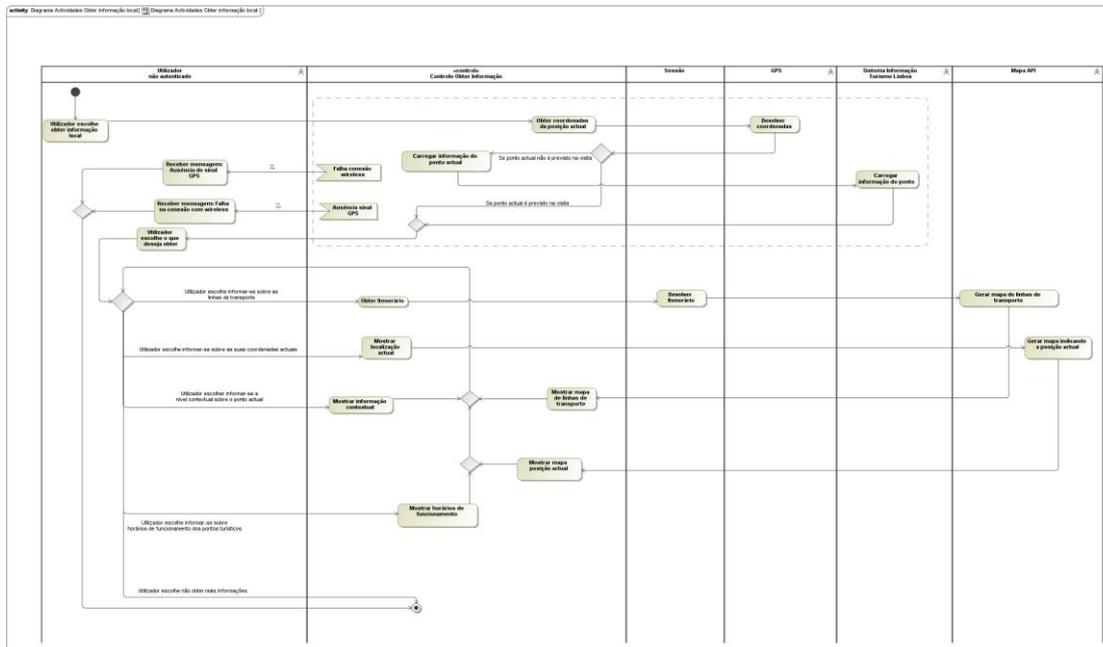


O utilizador introduz um conjunto de preferências, com base nas quais os pontos são filtrados e apresentados, e introduz igualmente restrições, por exemplo temporais. Relativamente a este tipo de restrição, tal não vai restringir o número de pontos turísticos que o utilizador vai poder ou não escolher, já que o tempo que um utilizador demora a visitar um determinado ponto turístico é completamente variável. Ainda assim, o sistema indica uma estimativa do tempo total que a visita irá demorar, estimativa essa que está guardada no sistema e que consiste, para cada ponto, no tempo médio que os utilizadores que já visitaram esse ponto demoraram a fazê-lo. Esta informação permite ao utilizador ter uma noção do tempo que irá necessitar para realizar a visita.

Ambas as opções gravar e terminar visita foram tratadas como excepções já que estas podem ocorrer mesmo que as informações acerca dos pontos turísticos que compõem a visita ainda não tenham sido apresentadas.

O caso secundário “Falha na ligação ao Sistema de Turismo de Lisboa” é descrito da mesma maneira que no diagrama anterior.

## Diagrama de Actividades Obter Informação Local



Neste diagrama, foi criada uma zona interrompível que abrange as acções de obter coordenadas e carregar informação, uma vez que é neste momento de execução que se corre o risco de falha na conexão ao *wireless* ou ausência do sinal GPS, na situação em que se origina a interrupção o utilizador recebe uma mensagem de alerta, e ocorre o final da actividade. A actividade correspondente a carregar informação do ponto actual, consiste em somente carregar a informação caso o ponto não esteja previsto na visita, pois toda a informação relativa aos pontos a visitar já foi carregada no início de uma visita.

As várias escolhas disponíveis ao utilizador relativamente ao tipo de informação que deseja obter, é representada por uma decisão, em que acaba por ser um ciclo enquanto o utilizador desejar obter mais informações.

Note-se que ao escolher obter informação das linhas de transporte, é necessário obter o itinerário (conjunto de pontos), e passando esse Set de pontos para o Mapa API, para que este gere o mapa indicando os transportes precisos para a deslocação de um local para o outro. Assumindo-se que o Mapa API é capaz de processar e calcular a informação relativa aos transportes precisos.

É de relevar que o mapa devolvido pelo Mapa API, é algo pronto a mostrar ao utilizador, em que o nosso sistema não necessita de fazer qualquer alteração sobre este.

Devido à estruturação que foi feita das classes a obtenção do itinerário é pedida à Sessão que guarda o diário de uma visita.

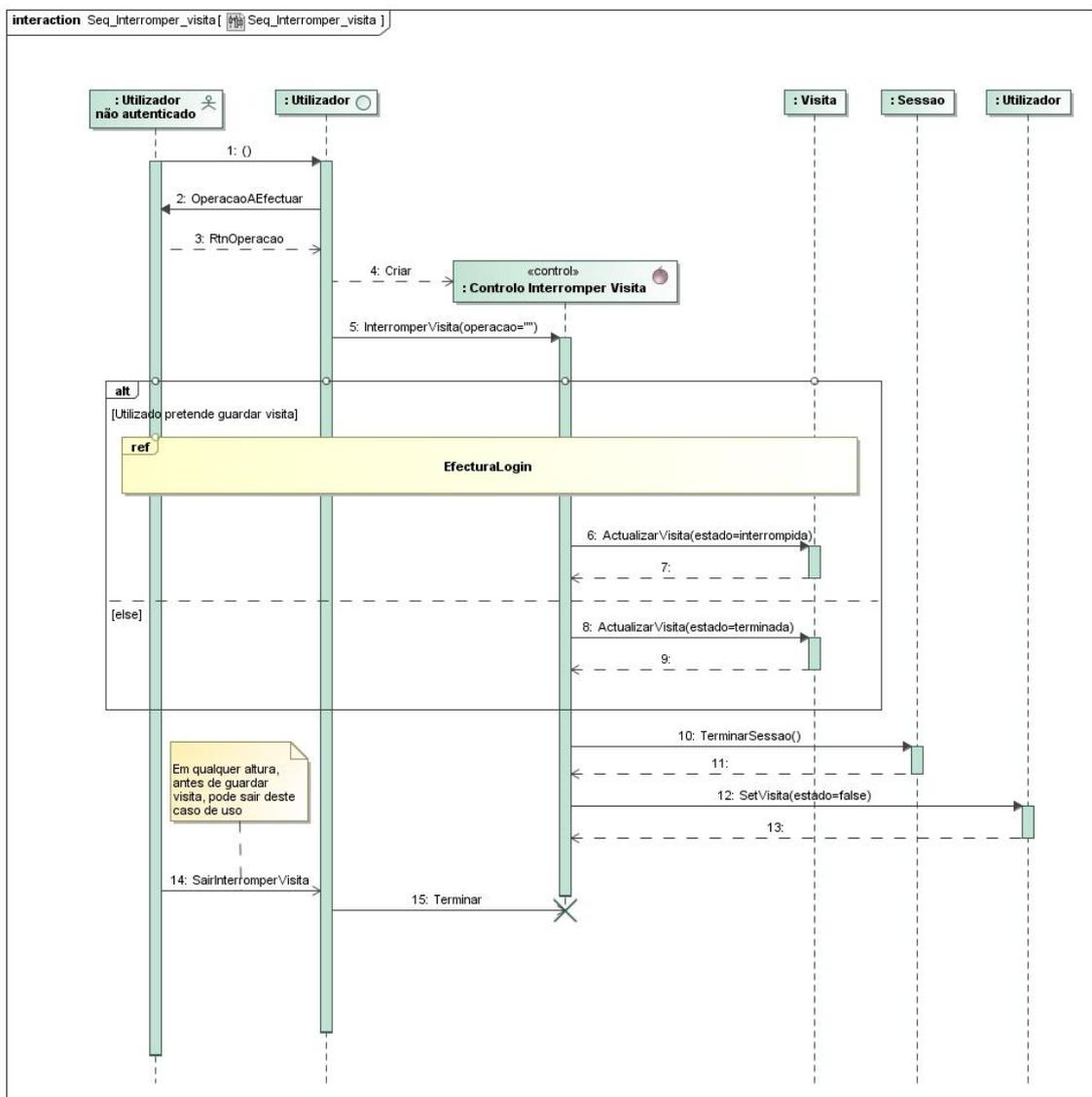
## Diagrama de Sequência

### Diagrama de Sequência Efectuar Visita

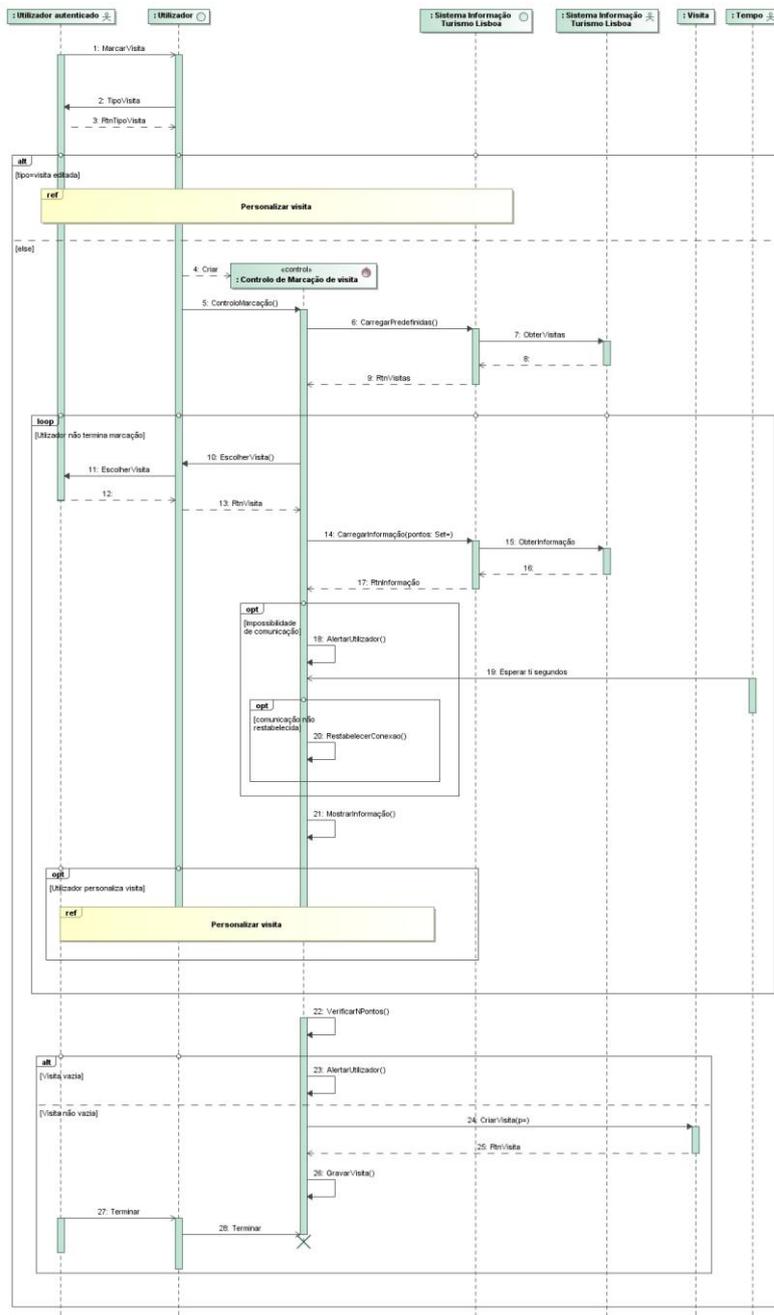
(Imagem em anexo)

Foi colocado em pontos que achamos adequados opt que representam as excepções do diagrama de actividades. Antes de criar o controlo é feito um diálogo entre o actor Utilizador e a sua interface de forma a adquirir dados necessários ao controlo, para depois disponibilizar ao controlo e este efectuar o seu processamento.

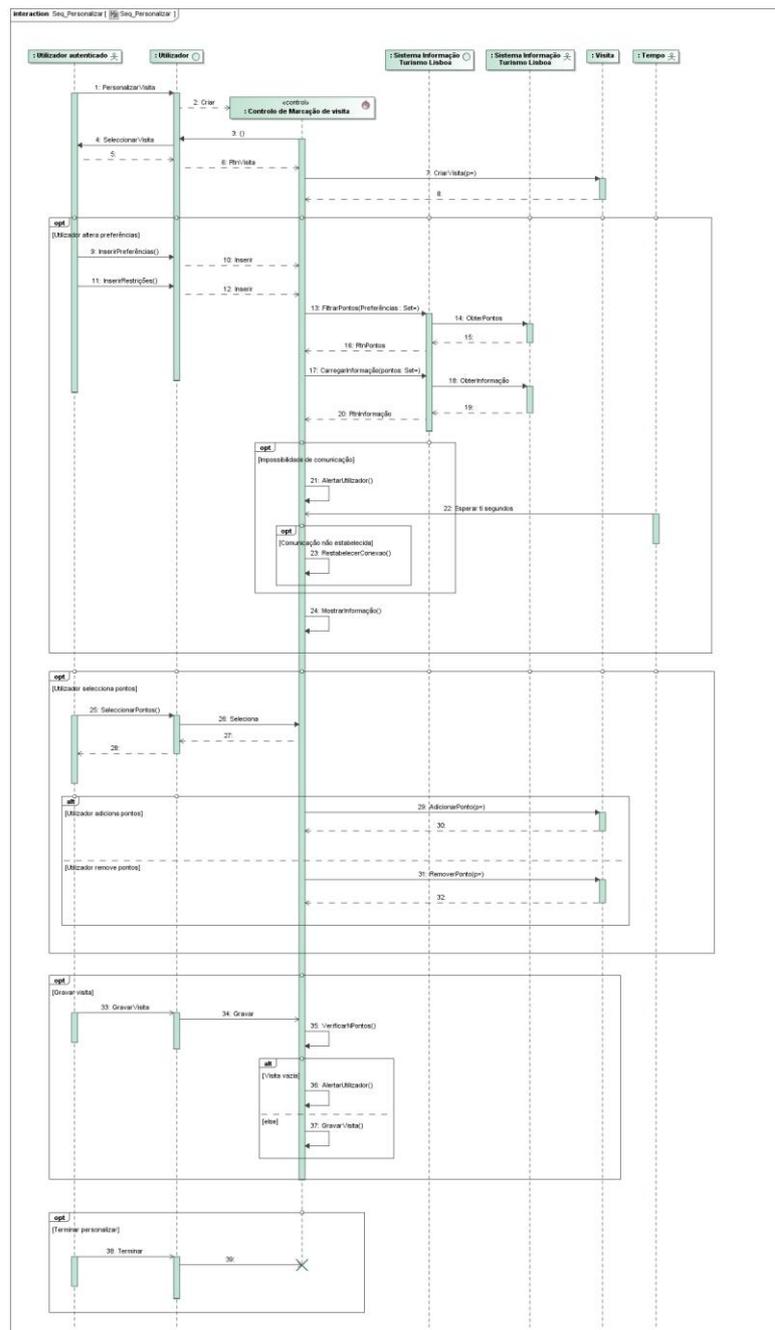
### Diagrama de Sequência Interromper Visita



## Diagrama de Sequência Marcar Visita



## Diagrama de Sequência Personalizar Visita



Em ambos os diagramas, as exceções utilizadas nos diagramas de atividades correspondentes foram representadas através de blocos opcionais (“opt”).

Nos dois diagramas apenas foi utilizado um controlo (controlo de marcação de visita), visto que as operações executadas por este são iguais em ambos, e como tal, a criação de um outro controlo para o personalizar visita seria redundante.

## Diagrama de Sequência Obter Informação Local

(Imagem em anexo)

Neste diagrama de sequência, foram tomadas decisões relativas às escolhas do utilizador sobre o tipo de informações a obter, tal como a criação de um *loop* para que o controlo esteja sucessivamente a perguntar que tipo de informação o utilizador deseja obter, em que o utilizador responde com o respectivo tipo, o qual será usado para se decidir que opção irá ser executada pelo controlo. O controlo é criado cedo, pois é necessário obter inicialmente as coordenadas e informação que não esteja carregada sobre o ponto actual.

## Diagrama de Classes

(Imagem em anexo)

O nosso diagrama de classes só contém entidades, controlo e interfaces dos casos de uso que trabalhamos. Criamos uma classe Turismo que contém todos os valores por defeito e que permite a comunicação com todas as outras classes, como tal apesar de os controlos nos diagramas de actividades e sequencia comunicarem directamente com as outras classes, aqui navega até elas pela classe Turismo. Implementamos esta estrutura, pois pensamos que esta modela melhor o problema e constitui uma melhor resolução deste.

Na classe Utilizador existe um utilizador especial para o sistema guardar o estado actual da visita de utilizadores não registados. A classe EstadoVisita vai conter os pontos já visitados de uma visita e o tempo lá dispensado. A classe Sessao relaciona as visitas com utilizador, guardando a informação de quando começaram e de quando terminaram e os itinerários da visita.

Devido ao facto de uma visita ter três estados, terminado, a decorrer ou interrompida, o método ActualizarVisita(estado: Integer) recebe um valor inteiro que vai representar estes três estados e actualizar o estado da visita de forma correcta.

## OCL

### Invariante

- Os pontos turísticos visitados têm de pertencer à mesma:

```
context Visita
invariant pontosTuristicosVistados:
    contem->includesAll(tem.visitado)
```

- Username único:

```
context User
invariant usernameUnico:
    User.allInstances->ForAll(u1, u2: User | u1 <> u2 implies u1.username
    <> u2.username)
```

### Pré-condição

- Existir pontos a visitar quando obter pontos:

```
context Visita::ObterPontos()
pre: contem->size() > 0
```

- O ponto a adicionar não está contido no conjunto de pontos já adicionados:

```
context Visita::AdicionarPonto(ponto: PontoTuristico)
pre: contem->excludes(ponto)
```

### Pós-condição

- Adicionar ponto bem sucedido:

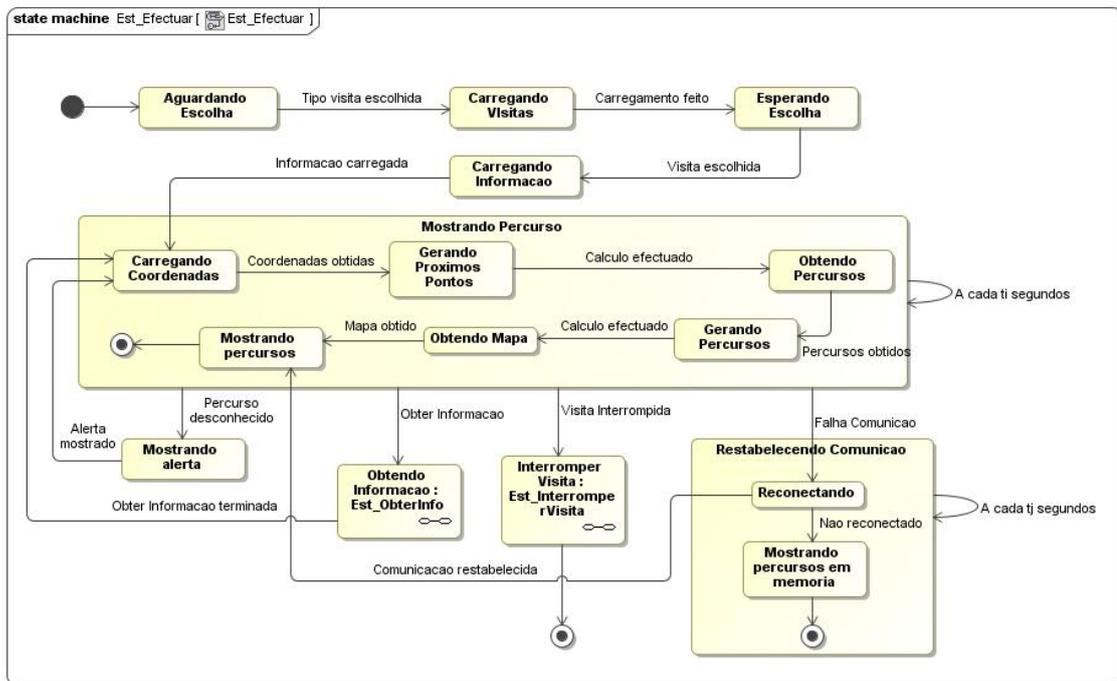
```
context Visita::AdicionarPonto(ponto: PontoTuristico)
post: contem->size() = contem@pre->size() + 1
```

- Nº de pontos na criação de uma visita menor que o por omissão:

```
context Visita::CriarVisita(ponto:Set)
post: contem->size <= visitaTurismo.maxPontosVisita
```

## Diagrama de Estados

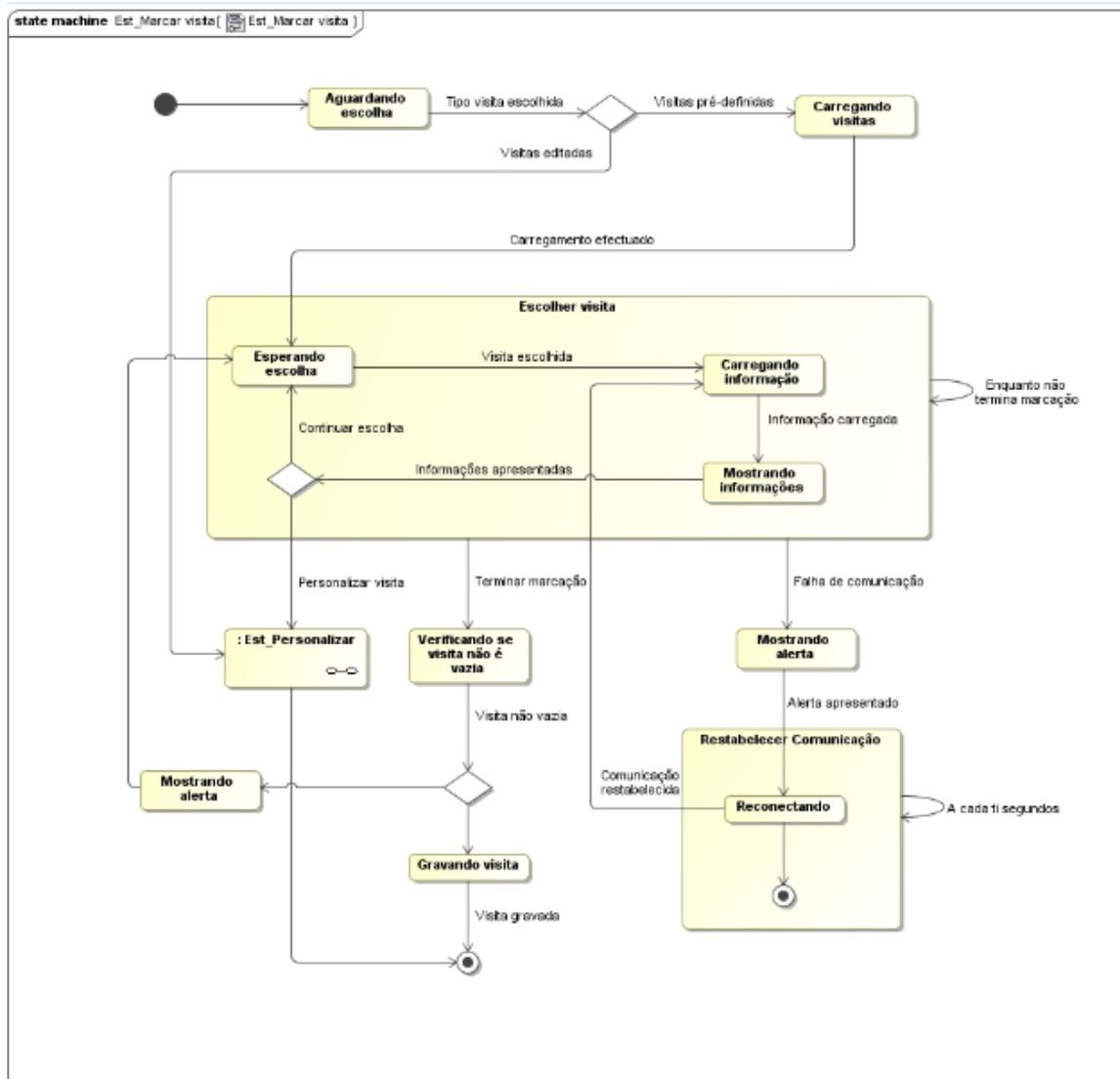
### Diagrama de Estados Efectuar Visita



Este diagrama de estados mostra os estados que a classe Visita irá passar durante a sua execução. O estado Mostrando Percurso é um estado composto, pois não pode ser representado num um único estado atómico. Este estado mostra todos os estados que o mostrar percurso passa.

As transições de Mostrando Percurso são na sua maioria acontecimentos que representam as excepções no diagrama de actividades.

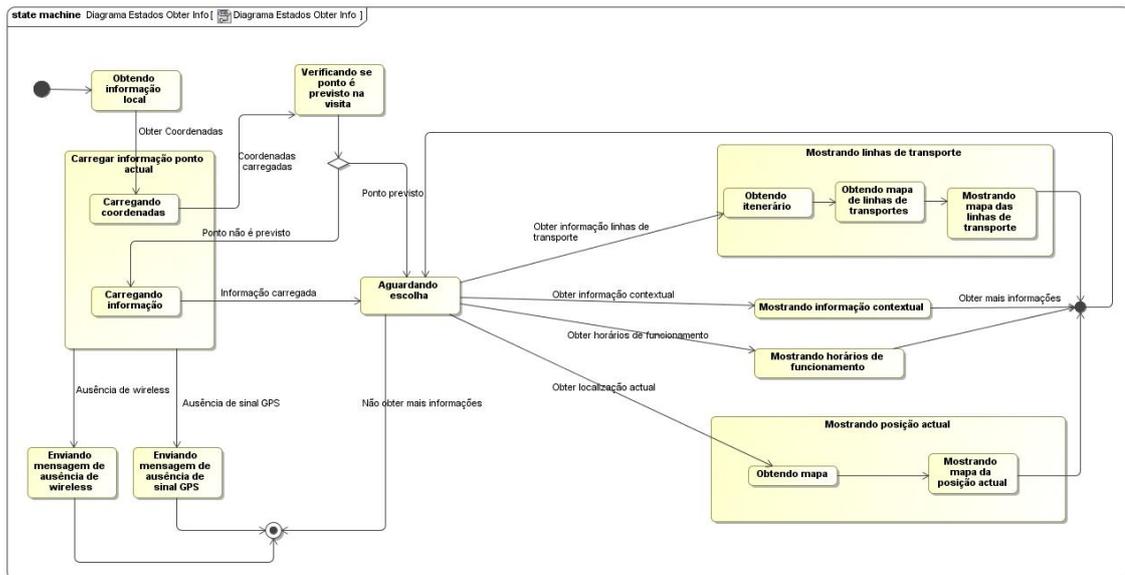
## Diagrama de Estados Marcar Visita



Neste diagrama é descrito o comportamento do caso de uso “Marcar visita”.

O estado composto “Escolher visita” representa a região interrompível pelas excepções de falha de ligação ao Sistema de Turismo e pelo terminação marcação por parte do utilizador.

## Diagrama de Estados Obter Informação Local



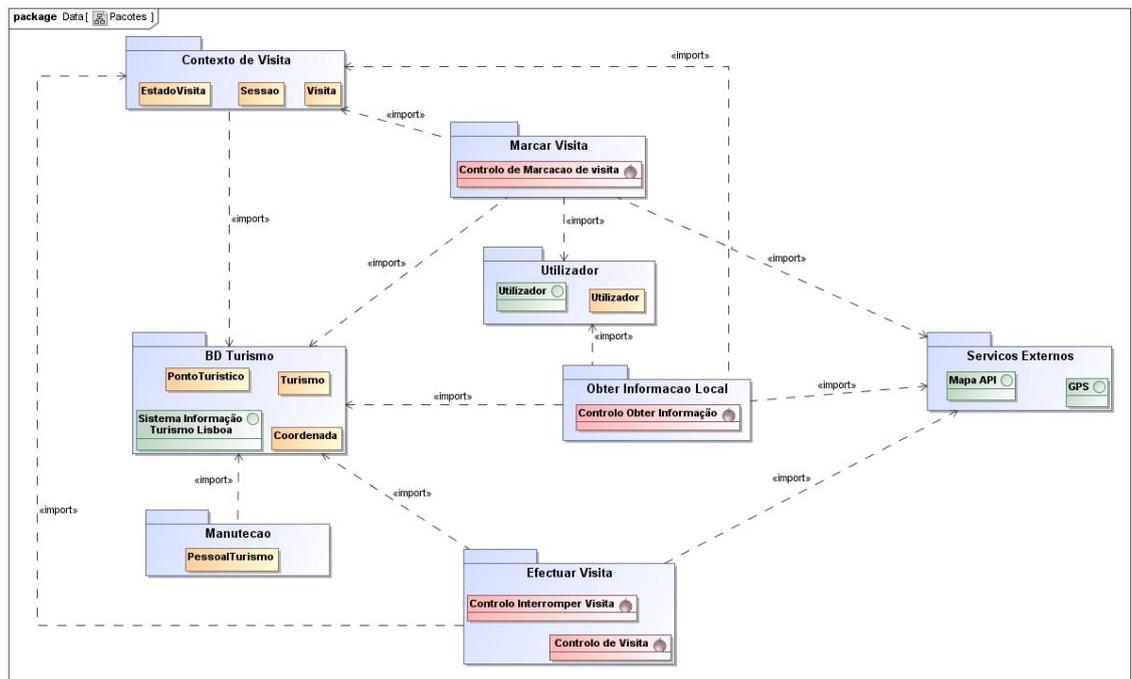
Neste diagrama é descrito o comportamento do caso de uso “Obter Informação Local”.

A primeira transição de estado consiste em obter as coordenadas, para que a seguir seja verificado se o respectivo ponto já está previsto na visita, para que se saiba se é necessário carregar a informação relativa.

O estado composto “carregar informação ponto actual” demonstra de novo a região interrompível pelas possibilidades de ausência de *wireless* ou de sinal GPS, as quais conduzem ao final. É também feita uma junção para que seja mais perceptível a transição relativa ao utilizador desejar obter mais informações e voltar a escolher o tipo de informação a obter.

Optei por criar os dois estados “Mostrando linhas de transporte” e “Mostrando posição actual”, isto porque não se tratavam de acções atómicas.

## Diagrama de Pacotes Global



Todas as relações deste diagrama consistem em “import” pois são usadas por mais de uma classe, implicando que não possam ser privadas somente a uma.

Os pacotes foram criados com o intuito de respeitar uma lógica que diverge os conceitos de visita, controlos disponíveis, utilizador, serviços externos ao nosso sistema, manutenção e a base de dados, que tanto engloba a externa e a interna, que consiste em guardar todos os dados que é necessário guardar no nosso sistema.

## Lista de Requisitos Não Funcionais

- Eficiência
- Verificabilidade
- Estabilidade
- Consistência
- Flexibilidade
- Segurança

Destes 6 Requisitos Não Funcionais considerou-se como os 3 mais importantes:

### **1º Consistência**

Torna-se o requisito mais importante dado que a pedido do cliente o comportamento do sistema deve ser sensível ao contexto, adaptando-se durante o decorrer da visita consoante o local onde o utilizador se encontra. A medida tomada para que tal se concretizasse foi com a ajuda de um timer cujo dever é em cada ti segundos actualizar o estado da visita, ou seja, voltar a gerar próximos pontos a visitar, indicando simultaneamente os respectivos percursos.

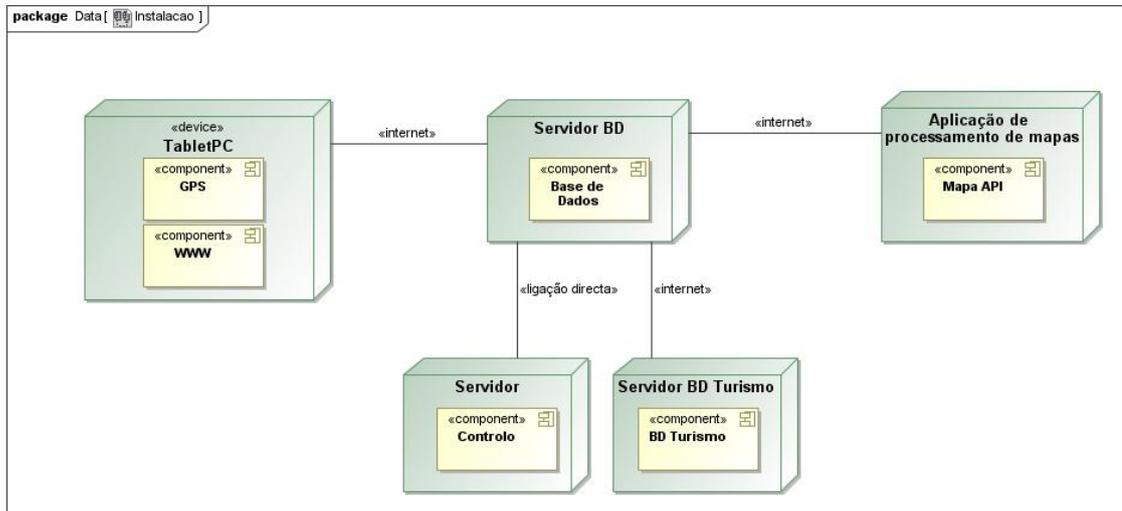
### **2º Eficiência**

O enunciado explicita que o sistema está sempre ligado a uma rede wireless a fim de obter informação. Uma consequência disto é o sistema estar sucessivamente a conectar-se ao wireless a fim de carregar informação. Decidiu-se então que seria preferível toda a informação ser carregada no início de uma visita ser efectuada, desta maneira reduz possíveis falhas ao nível de comunicação, e possível overload relacionado com os sucessivos carregamentos de informação.

### **3º Segurança**

O sistema permite ao utilizador visitar diversos pontos de interesse na cidade, quer seleccionando visitar turísticas pré-estabelecidas, quer gerando visitas à medida de acordo com as preferências e interesses do visitante, para além disto o utilizador pode ainda personalizar uma visita no momento em que a marca, e posteriormente a editar, para isto é feita uma restrição de que somente utilizadores autenticados podem concretizar estas duas últimas operações. Em suma, caso o utilizador não esteja autenticado é-lhe pedido o login. Tal restrição vai conduzir a um maior controlo sobre a informação, permitindo também ligações seguras.

## Diagrama de instalação



Um TabletPC é um dispositivo que irá conter o sistema de localização, que nos consideramos ser o GPS, e um dispositivo com ligação a internet para comunicar com o nosso sistema, o nosso WWW. O nosso sistema não irá conter a base de dados do Turismo de Lisboa, mas permite actualizá-la com os serviços que ela disponibiliza. Na nossa base de dados guardamos todos dados para o bom funcionamento do sistema. O servidor representa todo o processamento do sistema.



## Impacto dos Requisitos Não Funcionais

	Consistência	Eficiência	Segurança
Consistência		-	
Eficiência	-		-
Segurança		+	

Conforme os 3 requisitos não funcionais escolhidos existiram certos impactos tal como a necessidade de um utilizador efectuar o login, relativo ao requisito de segurança, que por sua vez prejudica a eficiência pois implica iniciar uma ligação segura por wireless e mantê-la enquanto necessário, para o acesso à área do utilizador. Como consequência disto advém o aumento do comportamento online.

No entanto, a eficiência ajuda a segurança pois implica menos trabalho online, uma vez que a informação é obtida totalmente no início, e uma ligação apenas volta a ser feita quando é necessário aceder à área de utilizador, desta forma os nossos dados permanecem mais protegidos, visto existirem menos ligações.

Relativamente à consistência, teve como impacto o facto de em cada ti segundos ser feita uma actualização do estado da visita, durante o decorrer desta. Existe uma relação mútua de prejuízo, por um lado é feito um acesso constante a sistemas externos que implica um processamento mais lento, esta situação foi resolvida diminuindo o acesso online colocando a informação disponível off-line, ajudando num processamento mais rápido. Por outro, caso hajam alterações na BD do Turismo Lisboa, não é feita qualquer actualização no nosso sistema, o que conduz ao acesso por parte do utilizador a informações desactualizadas, contudo a probabilidade de serem alteradas informações na BD é baixa, mantendo a consistência e eficiência.

## Conclusão

Durante a realização do trabalho tivemos alguma dificuldade em encontrar exemplos concretos de algumas situações que precisávamos de modelar nos nossos diagramas. Ainda assim tentámos ao máximo fazê-lo de forma correcta, usando como base os exemplos das aulas. No entanto o contexto destes não se enquadrava exactamente da mesma forma como o nosso.

Foram-nos surgindo diversas dúvidas relativas a ambiguidades de enunciado e a opções de modelação a tomar e para as esclarecer, mantivemos o contacto com o cliente. Esse contacto permitiu-nos estabelecer alguns pressupostos. Nem sempre as escolhas deste foram perceptíveis mas tentámos ao máximo efectuar o trabalho de forma a corresponder às mesmas, mantendo também uma perspectiva realista do projecto (tentar pensar no que fazia sentido numa situação real).

Ao longo do trabalho efectuámos diversas alterações nos diversos diagramas, com o objectivo de os melhorar. Contudo esperamos que todas as opções tomadas tenham sido as mais correctas para alcançar o objectivo do projecto.

## **Anexos**



