

Trails4health

|  |  |
| --- | --- |
| **Curso(s):** | Engenharia Informática |
| **Unidade(s) Curricular(es):** | Inteligência Artificial |
| **Ano Letivo:** | 2017/2018 |
| **Docente:** | Maria Clara Silveira  Noel Lopes |
| **Data:** | 24-11-2017 |

Índice

[1. Introdução 3](#_Toc498518330)

[2. Estado da arte 4](#_Toc498518331)

[2.1. Lista de casos de uso 4](#_Toc498518332)

[2.2. Tabela com comparação de funcionalidades 7](#_Toc498518333)

[SpiralDevelopment – Andreia Ernesto 8](#_Toc498518334)

[TwoTierView – Janilta Pires 9](#_Toc498518335)

[Quitting Time – Pedro Sanches 10](#_Toc498518336)

[3. Análise de Requisitos 14](#_Toc498518337)

[3.1. Descrição do tema do projeto 14](#_Toc498518338)

[3.2. Diagrama de Contexto 14](#_Toc498518339)

[3.3. Tabela de Atores, objetivos e respetivos Casos de Uso 14](#_Toc498518340)

[3.4. Diagrama de Casos de Uso 14](#_Toc498518341)

[3.5. Descrição de os Casos de Uso 15](#_Toc498518342)

[3.6. Diagramas de Sequência 17](#_Toc498518343)

[3.7. Diagramas de Classes completo 18](#_Toc498518344)

[3.8. Diagrama de Estados 18](#_Toc498518345)

[3.9. Semântica de Classes 19](#_Toc498518346)

[Classe Utilizador 19](#_Toc498518347)

Índice de Figuras

[Figura 1 - Diagrama de casos de uso, alguns fora da fronteira 14](file:///C:\Users\Acer\Documents\GitHub\Trails4Health\Doc\Eng.%20Software%20II\1.%20RelatórioEngSoftwareII%20v1.docx#_Toc498518244)

[Figura 2 - Diagrama de casos de uso, apenas casos de uso dentro da fronteira 15](file:///C:\Users\Acer\Documents\GitHub\Trails4Health\Doc\Eng.%20Software%20II\1.%20RelatórioEngSoftwareII%20v1.docx#_Toc498518245)

[Figura 3 - Diagrama do estado Agendar Trilho 18](file:///C:\Users\Acer\Documents\GitHub\Trails4Health\Doc\Eng.%20Software%20II\1.%20RelatórioEngSoftwareII%20v1.docx#_Toc498518246)

1. Introdução
2. Estado da arte

# Lista de casos de uso

**Registar utilizador**

Para puder adaptar os trilhos e poder frequentá-los o utilizador terá de se registar pudendo ser através de uma nova conta, pelo facebook, gmail…

**Iniciar sessão utilizador**

Para puder adaptar os trilhos e poder frequentá-los o utilizador terá de iniciar sessão pudendo ser através de uma nova conta, pelo facebook, gmail…

**Questionar aptidão física**

O objetivo deste caso de uso é de adaptar os trilhos a cada pessoa. Será questionada a idade, a frequência de atividade física, o estado de saúde…

**Configurar próprio trilho**

Tenho em conta o questionário da aptidão física, podendo assim adaptar os trilhos a cada pessoa sabendo a dificuldade, a distância, entre outras variantes que se devem atribuir.

**Consultar todos os trilhos**

Podendo ordená-los por preço, distância, dificuldade, entre outras variantes.

**Guardar trilho com interesse**

Tendo em conta os trilhos vistos, o utilizador poderá guardar os que tiver interesse para fazer mais tarde.

**Agendar percurso**

Sabendo que quer fazer o trilho em questão, o utilizador poderá agendar de imediato o percurso. Neste caso de uso irá aparecer se está disponível ou não (pois poderá estar em manutenção por exemplo), o preço,

**Pagar percurso**

O utilizador poderá escolher o tipo de pagamento que quer fazer, podendo ser por referência multibanco, cartão de crédito…

**Consultar trilhos já percorridas**

O utilizador, no seu perfil, poderá ver o total das rotas percorridas, os minutos gastos, os quilómetros que já fez…

**Consultar rotas agendadas**

Num tipo de agenda o utilizador, no seu perfil, poderá ver as rotas que tem para fazer, tal como saber as rotas que já pagou e as que ainda tem para pagar, entre outras coisas.

**Consultar rotas guardadas**

O utilizador, também no seu perfil, poderá ver as rotas que guardou pois tinha interesse em percorrer.

**Redefinir aptidão física**

O utilizador, ainda no seu perfil, poderá ver a sua aptidão física e modifica-la pois já melhorou ou piorou. Isto para que a aplicação continue a ser eficiente adaptando os trilhos a cada um.

**Criar trilhos para privilégio donos projeto (Professores Desporto)**

Os donos do projeto poderão adicionar os trilhos para que a aplicação continue atualizada.

**Criar proposta de trilho**

Os utilizadores, podendo conhecer a serra e os trilhos, poderão adicionar trilhos à aplicação. Esses trilhos terão de ser validados pelos donos do projeto.

**Validar trilhos para privilégio donos projeto (Professores Desporto)**

Este caso de uso serve para os donos do projeto validem os trilhos que foram criados pelos utilizadores.

**Consultar utilizadores que fizeram os trilhos**

Este caso de uso serve para os donos do projeto terem uma perceção e saberem as estatísticas dos trilhos, se tiveram mais ou menos afluência por exemplo.

**Consultar trilhos pagos e não pagos**

Este caso de uso serve para os donos do projeto saberem quem já pagou ou não os trilhos, para relembrar as pessoas que têm trilhos para pagar, para saberem quanto dinheiro já foi feito…

**Enviar opinião**

Este caso de uso serve para os utilizadores poderem dar a sua opinião, dizer se o trilho está ou não bem feito, se continua transitável, se há algo a obstruir o caminho, o que poderá ser melhorado…

**Responder a questionário**

Este caso de uso serve para a avaliação do projeto perante os seus visitantes, assim os donos do mesmo sabem a satisfação e os pontos a melhorar. A resposta ao questionário serve também para a realização de estudos estatísticos do projeto.

**Designar interesses**

Expõe os interesses perante os nossos visitantes, de modo, a criar neles curiosidade para nos visitar, alcançando assim um maior sucesso. Assim, mostramos o melhor que temos para oferecer a quem nos vista.

**Recomendar**

Ao recomendarmos aos nossos visitantes conseguimos um maior sucesso da atividade realizada. Assim os nossos clientes estão preparados para a escolha das atividades a desenvolver e assim conseguirem adaptar-se a nós e nós a eles.

**Expor fotos e respetivas informações**

Ao apresentarmos fotos do trilho conseguimos criar um maior impacto perante os nossos visitantes, visto que uma imagem vale mais que mil palavras. As informações respetivas a cada trilho terão informação adicional a cada imagem. Deste modo, os visitantes terão uma noção real da aventura que poderão usufruir.

**Apresentar normas e condutas do protejo**

Este caso de uso resulta como se fosse o regulamento de utilização do projeto. Assim conseguimos uma melhor utilização dos trilhos perante o nosso público.

**Publicar horário**

Publicar o horário de funcionamento do projeto a fim de que os nossos clientes saibam o horário para nos visitarem.

**Aconselhar época do ano**

Este caso de uso resulta em informar os nossos visitantes a melhor época do ano para nos visitar e quais as vantagens das diferentes estações.

**Apresentar condições atmosféricas**

Como a base do nosso projeto é um trilho ao ar livre é importante aconselharmos os nossos clientes a época melhor para nos visitar.

**Indicar forma de como chegar e contactos**

Com este caso de uso conseguimos auxiliar os nossos visitantes de como chegar ás nossas instalações pelo melhor caminho. Com a informação dos nossos contactos tudo será mais simples, e seremos mais próximos dos nossos visitantes.

# Tabela com comparação de funcionalidades

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Trails4Health** | **Passadiços do Paiva** | **Arribas do Douro** |
| Registo/Login Utilizador | ✔ | x | x |
| Questionar Aptidão Física | ✔ | x | x |
| Configurar trilho | ✔ | x | ✔ |
| Consultar trilhos | ✔ | ✔ | ✔ |
| Guardar trilho com interesse | ✔ | x | ✔ |
| Agendar percurso | ✔ | ✔ | ✔ |
| Pagar Percurso | ✔ | ✔ | x |
| Consultar trilhos já percorridos | ✔ | x | x |
| Consultar rotas agendadas | ✔ | x | x |
| Consultar rotas guardadas | ✔ | x | x |
| Redefinir aptidão física | ✔ | x | x |
| Criar trilhos para privilégio donos do projeto | ✔ | x | x |
| Criar proposta de trilho | ✔ | x | x |
| Validar trilhos para privilégio donos do projeto | ✔ | x | x |
| Consultar utilizadores que fizeram os trilhos | ✔ | x | x |
| Consultar trilhos pagos e não pagos | ✔ | x | x |
| Enviar opinião | ✔ | ✔ | ✔ |
| Responder a questionário | ✔ | x | x |
| Designar interesses | x | ✔ | ✔ |
| Recomendar | x | ✔ | ✔ |
| Expor fotos e respetivas informações | x | ✔ | ✔ |
| Apresentar normas e condutas do projeto | x | ✔ | ✔ |
| Publicar horário | x | ✔ | ✔ |
| Aconselhar época do ano | x | ✔ | ✔ |
| Apresentar condições atmosféricas | x | x | ✔ |
| Indicar forma de como chegar e contatos | x | ✔ | ✔ |

* 1. **Descrição de Padrões de Desenvolvimento**

## SpiralDevelopment – Andreia Ernesto

O **problema** do padrão é o facto do desenvolvimento dos casos de uso serem difíceis de desenvolver num único passo, o que impede a incorporação de novos dados. Isto atrasa a descoberta de erros e fatores de risco.

A **solução** é que se faça o desenvolvimento através de um processo iterativo em

que cada iteração aumenta progressivamente a precisão e objetividade dos casos de uso.

Escrevendo os casos de uso de modo iterativo serve para reorganizar o próprio para que este esteja funcionalmente correto ou para saber se não está a fazer falta no projeto. Com Irá perder-se algum trabalho mas nunca aquele que iria ser preciso para que os casos de uso fossem escritos numa só fase. Porém no modo iterativo será necessário saber quando parar, isto é, deve-se parar quando se têm casos de uso que consigam satisfazer todas as necessidades das partes interessadas.

Os requisitos vão sofrendo alterações à medida que se vão analisando e a criação desses mesmo para, porém só ao fim de muita análise e trabalho.

Existe também um custo alto nos erros das funcionalidades. Se esse erro for detetado numa fase inicial de análise terá um custo insignificante, mas se a sua deteção acontecer depois de ser implementado o *software* terá custos enormes.

A equipa de desenvolvimento do *software* só compreende o funcionamento do mesmo depois da descrição detalhada de cada caso de uso. Deve-se então fazer um desenvolvimento iterativo para aumentar a precisão e objetividade das funcionalidades e do *software*.

A lista de atores torna-se precisa à medida que se conhece os requisitos do sistema. O nome do caso deve demonstrar o que ele representa.

A criação da lista de casos de usos apesar de levar algum trabalho, torna mais fáceis a alteração ou remoção dos próprios. Essa lista ajuda na compreensão de cada caso de uso sabendo quais se relacionam entre si e com os atores.

Podemos ver compreender estes conceitos através do exemplo do trabalho desta UC Trails4Life, temos atores como os Professores e os Turistas que têm objetivos diferentes dentro da aplicação.

**Considerações finais:**

É extremamente importante o desenvolvimento iterativo dos casos de uso, não sendo necessário o desenvolvimento de todos os modelos de uma só vez.

Como se melhora iterativamente o conjunto dos casos de uso candidatos também se deve melhorar os seus diagramas.

Não existe a exigência de que todos os casos de uso devam estar no diagrama, porém os diagramas devem ser compreensíveis.

Há medida que vamos aumentando a eficiência do conjunto dos casos de uso, poderá surgir a necessidade de modelos diferentes que se concentram em coisas diferentes, tal como:

* Diagramas fundamentados em atores
* Diagramas fundamentados em funcionalidades similares
* Diagramas fundamentados ao nível de abstração

## TwoTierView – Janilta Pires

E processo para resolver casos de uso atraves da revisão de casos de usos extensos, permite que toda equipe interessada contribua com a sua opinao, conhecimentos, exigindo assim um tempo excessivo. E necessário comentários para verificar e validar a escrita e o conteudo de maneira que seja validada a exatidão e exaustividade do desenvolvimento do projecto desde o inicio. As pessoas tendem acreditar que o seu trabalho é o melhor do que realmente é, e não conseguem enchergar os seus erros porque ja estão familiarizado com o seu trabalho. Por outro lado, uma pessoa lendo o trabalho pela primeira vez é mais provável que encontram erros.

Para que uma revisão seja efetiva as equipas precisam investir muito tempo e energia de forma a aumentar a mão de obra significativa, por isso precisamos ser judicioso ao realizar o projecto, agendando-as apenas quando necessário e exigindo o minimo de cada pessoa, de forma que o trablho possa ficar conforme foi proposto.

Assim sendo temos dois tipos de revisao: A primeira feito por uma equipe interna menor, possivelmente muitas vezes, revendo os casos de uso internamente para verificar sua legibilidade, implementabilidade, precisão, Essas avaliações "internas" podem ser críticas informais, reuniões formais ou ambas as coisas. Essas avaliações iniciais são para eliminar o "ruído" causado pela ortografia, gramatical e formatação e erros técnicos, que, quando deixados sem correção, são distrativos. No final destas analises internas as equipas afirmam que é QuittingTime e que os casos de uso estão completos, corretos e estão implementáveis de acordo com o que foi proposto.

A segunda feito pelo grupo completo, talvez apenas uma vez. Essa definição de "grupo completo", pode ser desenvolvedores mais um executivo, às vezes são os analistas de negócios e os programadores principais, às vezes são usuários, executivos e toda a equipe de programação. Esta revisao Concentra na analise de produtos finais.;

## Quitting Time – Pedro Sanches

O padrão de desenvolvimento Quitting Time tem por base contrabalançar o risco de atrasar o projeto com o risco de requisitos incompletos.

Os programadores, antes de avançarem com o projeto, recolhem todos os objetivos e requisitos do sistema, pensam em todos os detalhes, de modo, a no futuro não haver riscos na falta de alguma funcionalidade. Quando estão a desenvolver os casos de uso, suspendem quando identificam e determinam todos os atores e objetivos do projeto. Após o cliente aprovar, os programadores, implementam no sistema esses mesmos casos. Mas para que isto seja possível, por vezes, é necessário prolongar a recolha de requisitos, o que leva ao atraso do projeto. Este atraso, caso ocorra, terá um aumento dos custos à medida que o tempo vai avançando.

Para ajudar na conclusão do projeto, os programadores têm três perguntas essenciais que necessitam de ser respondidas positivamente:

* **1º** Todos os atores e objetivos estão devidamente identificados e documentados?
* **2º** O cliente reconhece que a lista de casos de uso está completa e cada caso de uso é legível e correto?
* **3º** Os programadores podem implementar os casos de uso?

Caso alguma destas questões seja negativa, é necessário corrigir de forma a que se tornem positivas, só assim os programadores terão 100% de certeza para avançar para a etapa seguinte.

Este padrão de desenvolvimento está ligado diretamente ao Desenvolvimento em Espiral.

* 1. **Descrição de alguns casos de uso não utilizados**

**Pagar percurso**

|  |  |
| --- | --- |
| **Nome:** | Pagar percurso |
| **Descrição:** | O turista pagar o percurso supondo que tem guia turístico |
| **Pré-condição:** | Registo e login válido |
| **Caminho principal:** | 1. O sistema apresenta o menu principal 2. O ator carrega na opção “percursos” 3. O sistema mostra os percursos disponíveis 4. O ator escolhe o percurso pretendido 5. O sistema pergunta se pretende guia 6. O ator seleciona a opção “sim” 7. O sistema exibe o montante a pagar, as opções de pagamento e dados para pagamento 8. O ator escolhe a forma de pagamento e executa o pagamento (introduz os dados: nome, nº cartão de crédito…) 9. O sistema emite o recibo e confirma o pagamento através do envio de um e-mail |
| **Caminho alternativo:** | 1. a) Sistema indisponível 2. a) Percurso lotado 3. a) Não haver guia disponível 4. a) Pagamento recusado |
| **Suplementos ou adornos:** | 1. Se existe guia disponível 2. Se o pagamento foi efetuado com sucesso 3. Garantir que o sistema funciona mesmo quando existe excesso de procura |
| **Pós-Condição:** | O sistema envia um email a confirmar pagamento. |

**Criar proposta de trilho**

|  |  |
| --- | --- |
| **Nome:** | Criar proposta de trilho |
| **Descrição:** | Os utilizadores podem adicionar trilhos à aplicação. Esses trilhos terão de ser validados pelos donos do projeto. |
| **Pré-condição:** | Registo e login válido |
| **Caminho principal:** | 1. O sistema apresenta o menu principal 2. O ator carrega na opção “ver trilhos” 3. O sistema apresenta os trilhos existentes 4. O ator carrega na opção “criar proposta de trilho” 5. O sistema interroga as características e detalhes do percurso (coordenadas, km’s, dificuldade do percurso) 6. O ator insere os dados pedidos e valida os dados 7. O sistema regista proposta de trilho |
| **Caminho alternativo:** | 1. a) Página indisponível 2. a) Dados incompletos ou inválidos 3. a) Trilho existente |
| **Suplementos ou adornos:** | 1. Obrigatoriamente, o trilho ainda não pode existir 2. O trilho tem de ser registado com sucesso 3. Garantir que o sistema funciona mesmo quando existe excesso de procura |
| **Pós-Condição:** |  |

**Responder a questionário da Aptidão Física**

|  |  |
| --- | --- |
| **Nome:** | Responder a questionário da Aptidão Física |
| **Descrição:** | O objetivo deste caso de uso é de adaptar os trilhos a cada pessoa. Será questionada a idade, a frequência de atividade física, o estado de saúde… |
| **Pré-condição:** | Registo e login válido |
| **Caminho principal:** | 1. O ator carrega na opção “Definir aptidão física” 2. O sistema mostra o questionário com perguntas relacionadas à aptidão física do turista (idade, peso, doenças…) 3. O ator responde às perguntas efetuadas 4. O sistema confirma se o questionário foi devidamente preenchido 5. O ator confirma os dados 6. O sistema regista aptidão física |
| **Caminho alternativo:** | 1. a) Servidor indisponível 2. a) O ator não respondeu às perguntas obrigatórias então o sistema não deixou registar a aptidão física. |
| **Suplementos ou adornos:** | 1. O utilizador introduziu a idade superior a 120, não será possível 2. O utilizador introduziu a peso superior a 200kg, não será possível 3. Garantir que o sistema funciona mesmo quando existe excesso de procura |
| **Pós-Condição:** |  |

1. Análise de Requisitos

# Descrição do tema do projeto

# Diagrama de Contexto

# Tabela de Atores, objetivos e respetivos Casos de Uso

# Diagrama de Casos de Uso

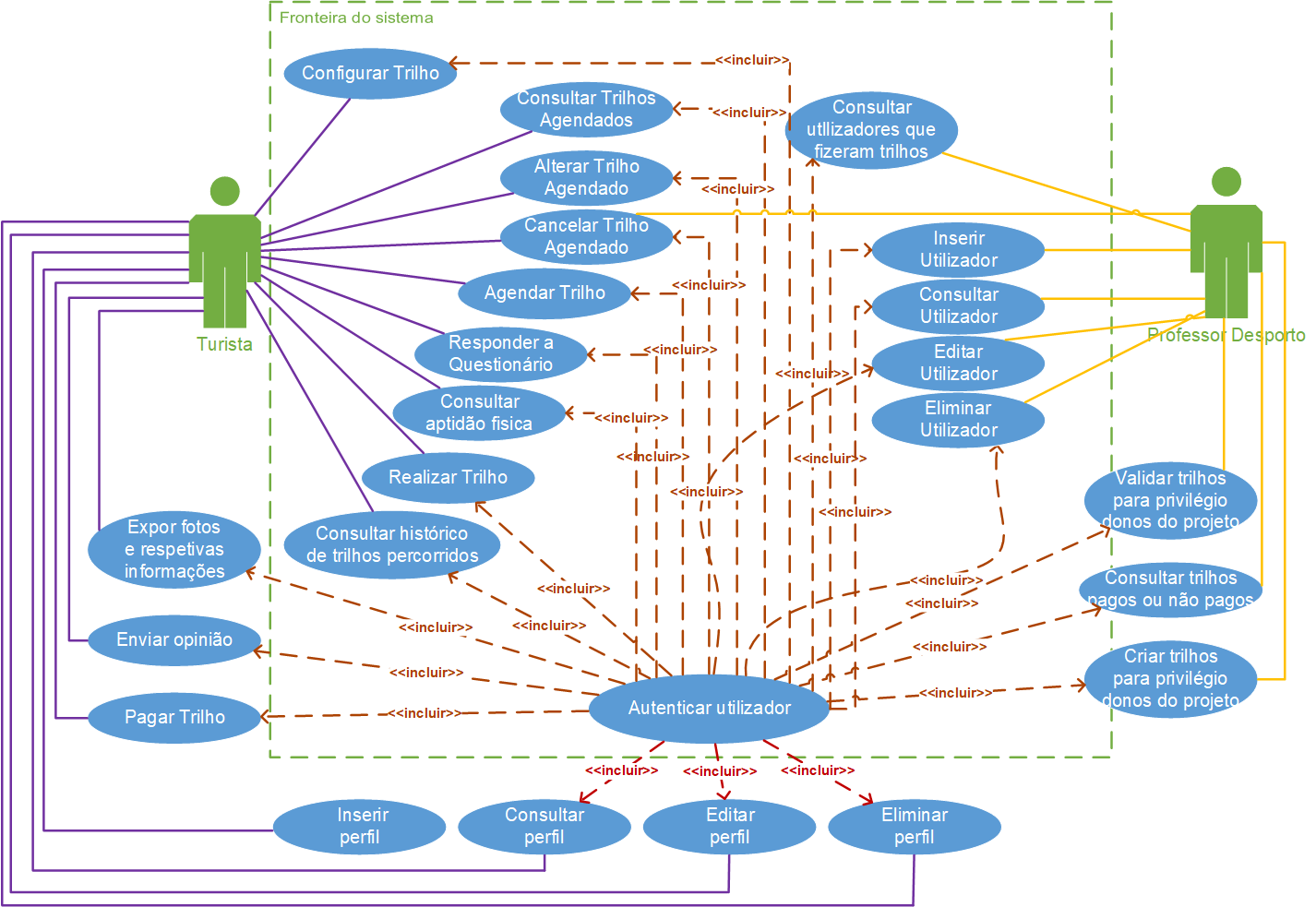
**Com casos de uso fora da fronteira**

Figura 1 - Diagrama de casos de uso, alguns fora da fronteira

**Apenas casos de uso dentro da fronteira**



Figura 2 - Diagrama de casos de uso, apenas casos de uso dentro da fronteira

# Descrição dos Casos de Uso

**Inserir utilizador**

|  |  |
| --- | --- |
| **Nome:** | Inserir utilizador |
| **Descrição:** | O objetivo deste caso de uso é o ator aceder ao sistema com o intuito de criar um utilizadores |
| **Pré-condição:** | Não têm |
| **Caminho principal:** | 1. O utilizador seleciona a opção de “Inserir utilizador” 2. O sistema disponibiliza o formulário de inserção de utilizador com todos os campos a preencher 3. O ator preenche o formulário de inserção de utilizador (introduz o Nome, Password, Morada...) 4. O sistema pede para confirmar 5. O utilizador confirma num botão guardar 6. O sistema Guarda o registo |
| **Caminho alternativo:** | O sistema cancela o registo se o utilizador carregar no botão cancelar a qualquer momento.  5.a. O sistema cancela porque o utilizador não confirmou |
| **Suplementos ou adornos:** | Testar se os campos obrigatórios estão preenchidos.  Testar o campo do NIF através do algoritmo do dígito de controlo.  Garantir que o sistema funciona mesmo quando existe excesso de procura |
| **Pós-Condição:** | Não têm |

**Consultar utilizador**

|  |  |
| --- | --- |
| **Nome:** | Consultar utilizador |
| **Descrição:** | O objetivo é o professor de desporto puder ver todos os turistas e o turista puder consultar o seu próprio perfil. |
| **Pré-condição:** | Registo e login válido |
| **Caminho principal:** | 1. O professor de desporto seleciona a opção “Consultar utilizadores” 2. O sistema apresenta uma interface “Gerir turistas” com um formulário para pesquisa e uma tabela com todos os turistas. 3. O ator vai pesquisar um Turista por Nome ou NIF. 4. O sistema mostra o resultado da pesquisa feita. |
| **Caminho alternativo:** | 1.a. O utilizador pode ser o turista.  4.a. O utilizador não existe. |
| **Suplementos ou adornos:** | Testar se o resultado coincide com a pesquisa.  Garantir que o sistema funciona mesmo quando existe excesso de procura |
| **Pós-Condição:** | Não têm**.** |

**Editar utilizador**

|  |  |
| --- | --- |
| **Nome:** | Editar utilizador |
| **Descrição:** | O objetivo é o professor de desporto puder ver editar os turistas e o turista puder editar o seu próprio perfil. |
| **Pré-condição:** | Registo e login válido |
| **Caminho principal:** | 1. O professor de desporto seleciona a opção “Editar utilizador” 2. O sistema apresenta uma interface “Editar turistas” com um formulário para pesquisa e uma tabela com todos os turistas. 3. O ator vai pesquisar um Turista por Nome ou NIF que pretende editar. 4. O sistema mostra o resultado da pesquisa feita com todos os campos prontos a ser editados. 5. O ator altera os campos pretendidos 6. O sistema pede para confirmar 7. O ator confirma num botão guardar 8. O sistema Guarda o registo |
| **Caminho alternativo:** | O sistema cancela o registo se o utilizador carregar no botão cancelar a qualquer momento.  1.a. O utilizador pode ser o turista.  4.a. O utilizador não existe.  7.a. O sistema cancela porque o utilizador não confirmou |
| **Suplementos ou adornos:** | Testar se os campos obrigatórios estão preenchidos.  Testar o campo do NIF através do algoritmo do dígito de controlo.  Garantir que o sistema funciona mesmo quando existe excesso de procura |
| **Pós-Condição:** | Não têm. |

**Eliminar utilizador**

|  |  |
| --- | --- |
| **Nome:** | Eliminar utilizador |
| **Descrição:** | O objetivo é o professor de desporto puder ver eliminar os turistas e o turista puder eliminar o seu próprio perfil. |
| **Pré-condição:** | Registo e login válido |
| **Caminho principal:** | 1. O professor de desporto seleciona a opção “Eliminar utilizador” 2. O sistema apresenta uma interface “Eliminar turistas” com um formulário para pesquisa e uma tabela com todos os turistas. 3. O ator vai pesquisar um Turista por Nome ou NIF que pretende eliminar. 4. O sistema mostra o resultado da pesquisa feita com todos os campos prontos a ser eliminados. 5. O ator carrega em eliminar 6. O sistema pede para confirmar 7. O ator confirma num botão eliminar 8. O sistema elimina o utilizador |
| **Caminho alternativo:** | O sistema cancela o registo se o utilizador carregar no botão cancelar a qualquer momento.  1.a. O utilizador pode ser o turista.  4.a. O utilizador não existe.  7.a. O sistema cancela porque o utilizador não confirmou |
| **Suplementos ou adornos:** | Garantir que o sistema funciona mesmo quando existe excesso de procura |
| **Pós-Condição:** | **Não têm** |

**Autenticar utilizador**

|  |  |
| --- | --- |
| **Nome:** | Autenticar Utilizador |
| **Descrição:** | O turista/professor de desporto ao autenticarem-se, através da conta previamente registado, *gmail* ou *facebook*, conseguem aceder a informações privilegiadas através do seu perfil. |
| **Pré-condição:** | Registo válido |
| **Caminho principal:** | 1. O ator seleciona a opção “login”; 2. O sistema apresenta os campos para a autenticação do utilizador (Nome/E-mail e Password); 3. O ator preenche os campos e valida os dados; 4. O sistema verifica se os campos foram devidamente preenchidos (Nome/E-mail e Password) e executa o login com sucesso; |
| **Caminho alternativo:** | 3. a) O utilizador não preencheu devidamente os campos.  4. a) O e-mail/password não ser válido. |
| **Suplementos ou adornos:** | 1. Garantir que o sistema funciona mesmo quando existe excesso de procura |
| **Pós-Condição:** | Não existe. |

**Consultar aptidão física**

|  |  |
| --- | --- |
| **Nome:** | Consultar aptidão física |
| **Descrição:** | Este caso de uso tem como objetivo o turista consultar a sua aptidão física. |
| **Pré-condição:** | Registo e login válido |
| **Caminho principal:** | 1. O ator seleciona a opção “Consultar aptidão física”; 2. O sistema apresenta todos os campos referentes à aptidão física do utilizador como o peso, doenças, altura, entre outros… |
| **Caminho alternativo:** | 1. a) O sistema informa que não há dados do mesmo caso o turista nunca tenha respondido ao questionário da aptidão física, |
| **Suplementos ou adornos:** | Não existe. |
| **Pós-Condição:** | Não existe. |

**Consultar histórico de trilhos percorridos**

|  |  |
| --- | --- |
| **Nome:** | Consultar histórico de trilhos percorridos |
| **Descrição:** | Este caso de uso permitir ao professor de desporto consultar o histórico de trilhos percorridos por cada utilizador. |
| **Pré-condição:** | Registo e login válido |
| **Caminho principal:** | 1. O ator seleciona a opção “Consultar histórico de trilhos percorridos”; 2. O sistema apresenta, numa base de dados, a totalidade dos trilhos percorridos por cada utilizador, assim como o tempo gasto em cada percurso/etapa e a distância total percorrida; 3. O ator pesquisa pelo utilizador inserindo nome ou NIF; 4. O sistema exibe os resultados da pesquisa efetuada. |
| **Caminho alternativo:** | Não existe. |
| **Suplementos ou adornos:** | 1. O ator introduz NIF superior a 9 dígitos. |
| **Pós-Condição:** | Não existe. |

**Consultar utilizadores que percorreram determinado trilho**

|  |  |
| --- | --- |
| **Nome:** | Consultar utilizadores que percorreram determinado trilho |
| **Descrição:** | O objetivo deste caso de uso é permitir ao professor de desporto visualizar os utilizadores que já percorreram determinado trilho |
| **Pré-condição:** | Registo e login válido |
| **Caminho principal:** | 1. O ator seleciona a opção “Consultar utilizadores que percorreram determinado trilho”; 2. O sistema apresentar os todos os trilhos existentes; 3. O ator seleciona o trilho pretendido; 4. O sistema apresenta, numa base de dados, todos os turistas registados no *website* (nome, e-mail, NIF…) que percorreram aquele determinado trilho; 5. O ator pesquisa pelo utilizador inserindo nome ou NIF. 6. O sistema exibe os resultados da pesquisa efetuada. |
| **Caminho alternativo:** | Não existe. |
| **Suplementos ou adornos:** | 1. O ator introduz NIF superior a 9 dígitos. 2. O ator não está registado no *website.* |
| **Pós-Condição:** | Não existe. |

**Configurar trilho**

|  |  |
| --- | --- |
| **Nome:** | Configurar trilho |
| **Descrição:** | Este caso de uso tem como objetivo apresentar os melhores trilhos consoante a aptidão física de cada utilizador. |
| **Pré-condição:** | Registo e login válido |
| **Caminho principal:** | 1. O ator seleciona a opção “Configurar trilho”; 2. O sistema apresenta os trilhos existentes anteriormente inseridos; 3. O ator determina a distância que pretende percorrer, o local de inicio e fim do trilho, data de inicio e o tempo que pretende gastar em cada etapa/percurso; 4. O sistema apresenta a melhor solução, de maneira, a enquadrar-se aos filtros pré-definidos pelo utilizador e a condição física (respostas ao questionário); 5. O ator seleciona o trilho que pretende percorrer de acordo com as propostas apresentadas; 6. O sistema guarda o registo do trilho. |
| **Caminho alternativo:** | 3. a). Trilho já existe na base de dados.  b). Dados incompletos ou inválidos |
| **Suplementos ou adornos:** | 1. Garantir que o sistema funciona mesmo quando existe excesso de procura 2. O utilizador introduziu uma distância superior a 500 km 3. O utilizador introduziu uma data inferior à data do sistema. |
| **Pós-Condição:** | Não existe. |

**Agendar Trilho**

|  |  |
| --- | --- |
| **Nome:** | Agenda Trilho |
| **Descrição:** | Este caso de uso tem como objetivo apresentar os melhores trilhos agendados consoante a aptidão física de cada Turista. |
| **Pré-condição:** | Registo e login válido |
| **Caminho principal:** | 1. O ator seleciona a opção “Agendar Trilho”; 2. O sistema apresenta os trilhos Agendados anteriormente; 3. O ator determina a data prevista que pretende percorrer, o local de inicio fim, e o tempo que pretende gastar em cada percurso; 4. O sistema apresenta a melhor solução, de maneira, a enquadrar-se aos filtros pré-definidos pelo turista e as dificuldades e a condição física (respostas ao questionário); 5. O ator seleciona o trilho que pretende percorrer de acordo com as propostas apresentadas; 6. O sistema guarda o registo do trilho. |
| **Caminho alternativo:** | 3. a). Trilho já existe na base de dados.  b). Dados incompletos ou inválidos |
| **Suplementos ou adornos:** | 1. Garantir que o sistema funciona mesmo quando existe excesso de procura 2. O Turista introduziu uma data inferior à data do sistema. |
| **Pós-Condição:** |  |

**Alterar a data prevista de inicio do trilho**

|  |  |
| --- | --- |
| **Nome:** | Alterar Trilho |
| **Descrição:** | O Turista altera a data prevista do inicio do trilho |
| **Pré-Condições:** | Login Válido |
| **Caminho Principal:** | 1. O ator seleciona um trilho a alterar; 2. O sistema apresenta todos os campos a alterar(data\_prevista\_inicio\_trilho;tempo gasto pretendido…) de acordo com as resposta do questionário, 3. O ator altera os campos desejado e submete o questionário; 4. O sistema pede para confirmar a alteração; 5. O ator confirma; 6. O sistema altera; |
| **Caminho Alternativo** | 2 a)Não haver resposta no questionário  4 a) Erro na alteração dos dados no formulário  7 a) sistema não altera |
| **Suplemento ou Adorno** | 1. Garantir que o sistema funcione só quando existir trilhos agendados ou seja quando existe uma   .data inicio  .data inicio prevista  .tempo gasto=branco  .estado agendamento=agendado  Atualiza o estado |
| **Pós-Condição** | O sistema envia uma notificação confirmando a alteração |

**Consultar trilhos agendados**

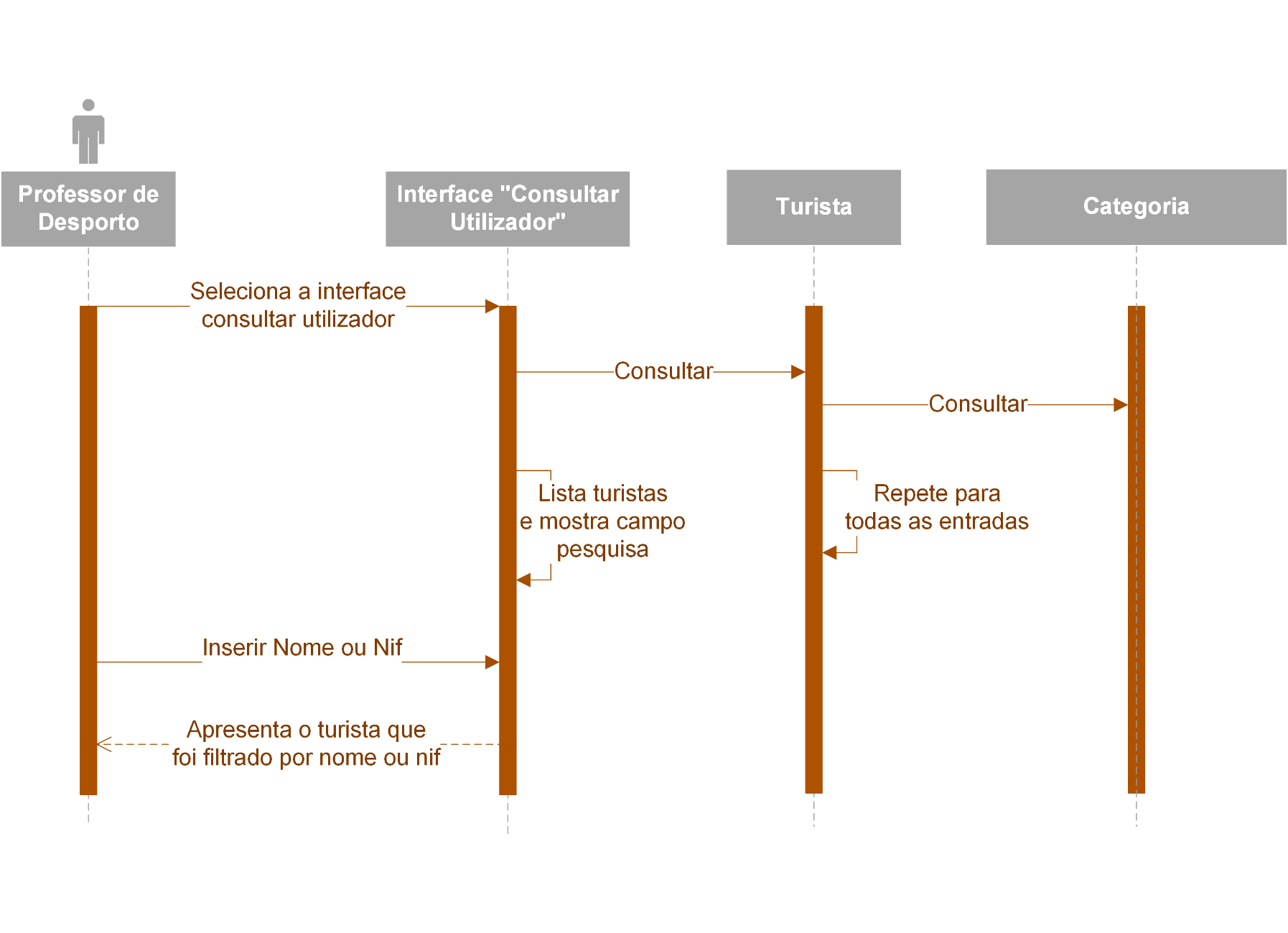
|  |  |
| --- | --- |
| **Nome:** | Consultar trilhos agendados |
| **Descrição:** | O Turista consulta o seu trilho agendado |
| **Pré-Condições:** | Login Válido |
| **Caminho Principal:** | 1. O ator seleciona a opção “Consultar trilho agendado”; 2. O sistema Apresenta na base de dados todos os trilhos agendados pelo turista; 3. O ator consulta os trilho agendado inserindo (Data Reserva, Data Previta do inicio do trilho,tempo pretendido a gastar); 4. O sistema exibe o resultado da pesquisa, |
| **Caminho Alternativo** | 1 a) Falha no sistema  1 b) Não haver trilho agendados  3 a) O sistema não mostrar os trilho já agendados |
| **Suplemento ou Adorno** | 1. Trilho a consultar não disponivel 2. Garantir que o trilho foi agendado |
| **Pós-Condição** |  |

**Cancelar Trilho Agendado**

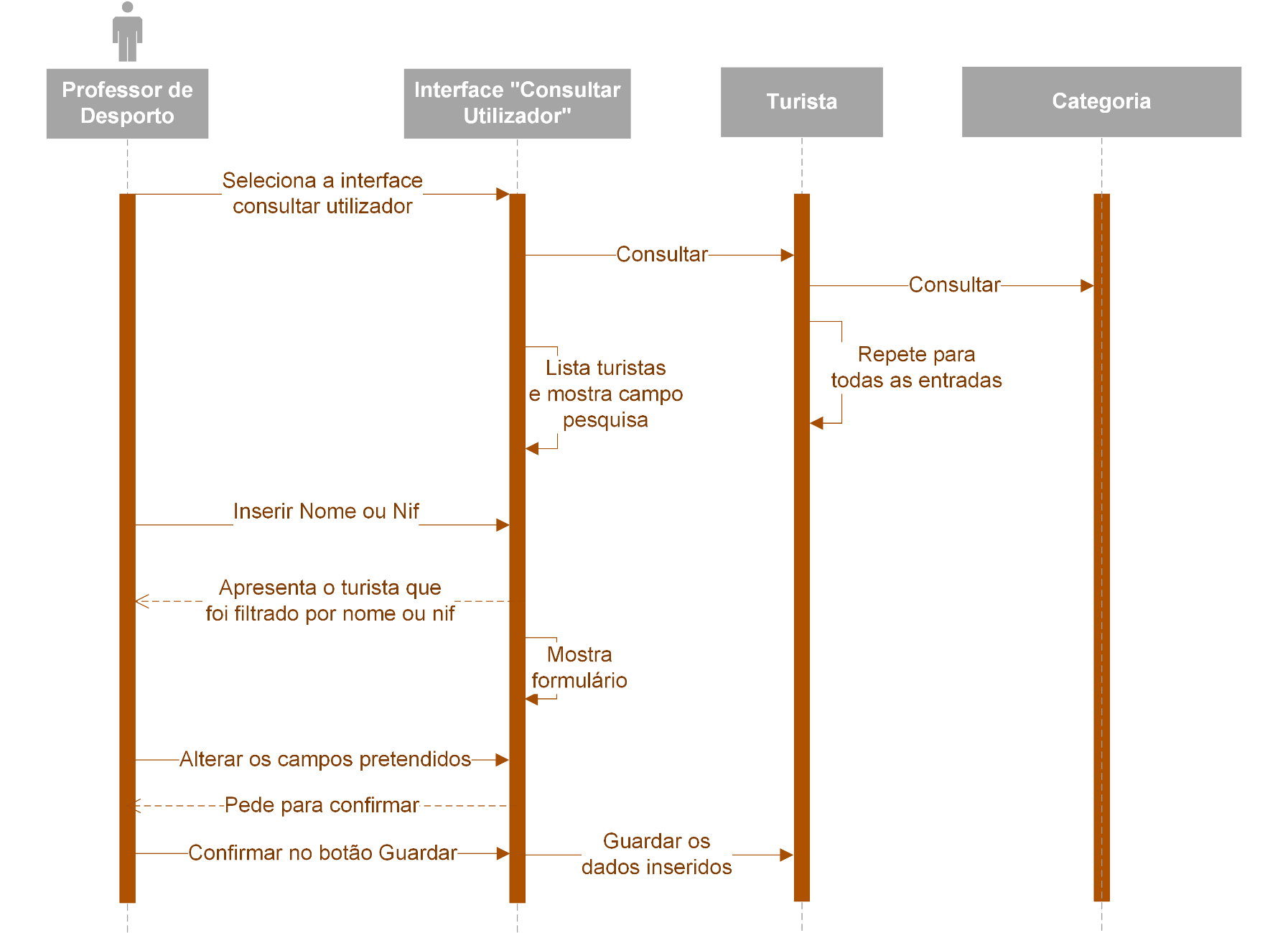
|  |  |
| --- | --- |
| **Nome:** | Cancelar trilho |
| **Descrição:** | O objectivo é o turista fazer o cancelamento do trilho agendado |
| **Pré-Condições:** | Login Válido |
| **Caminho Principal:** | 1. O ator seleciona a opção “Cancelar trilho” 2. O sistema apresenta um formulário com os dados de todos os trilhos agendados 3. O ator seleciona o trilho que pretende cancelar 4. O sistema apresenta a opção “Cancelar” 5. O ator clica no botão “Cancelar” 6. O sistema pede para confirmar 7. O ator confirma clicando no botão “cancelar trilho” 8. O sistema cancela |
| **Caminho Alternativo** | 2 a) Sistema não Apresenta o formulário  4 a) Erro na opção cancelar  6 a) Sistema não cancela |
| **Suplemento ou Adorno** | Se o estado do agendamento for igual cancelado |
| **Pós-Condição** | O sistema envia uma mensagem com sucesso |

# Diagramas de Sequência

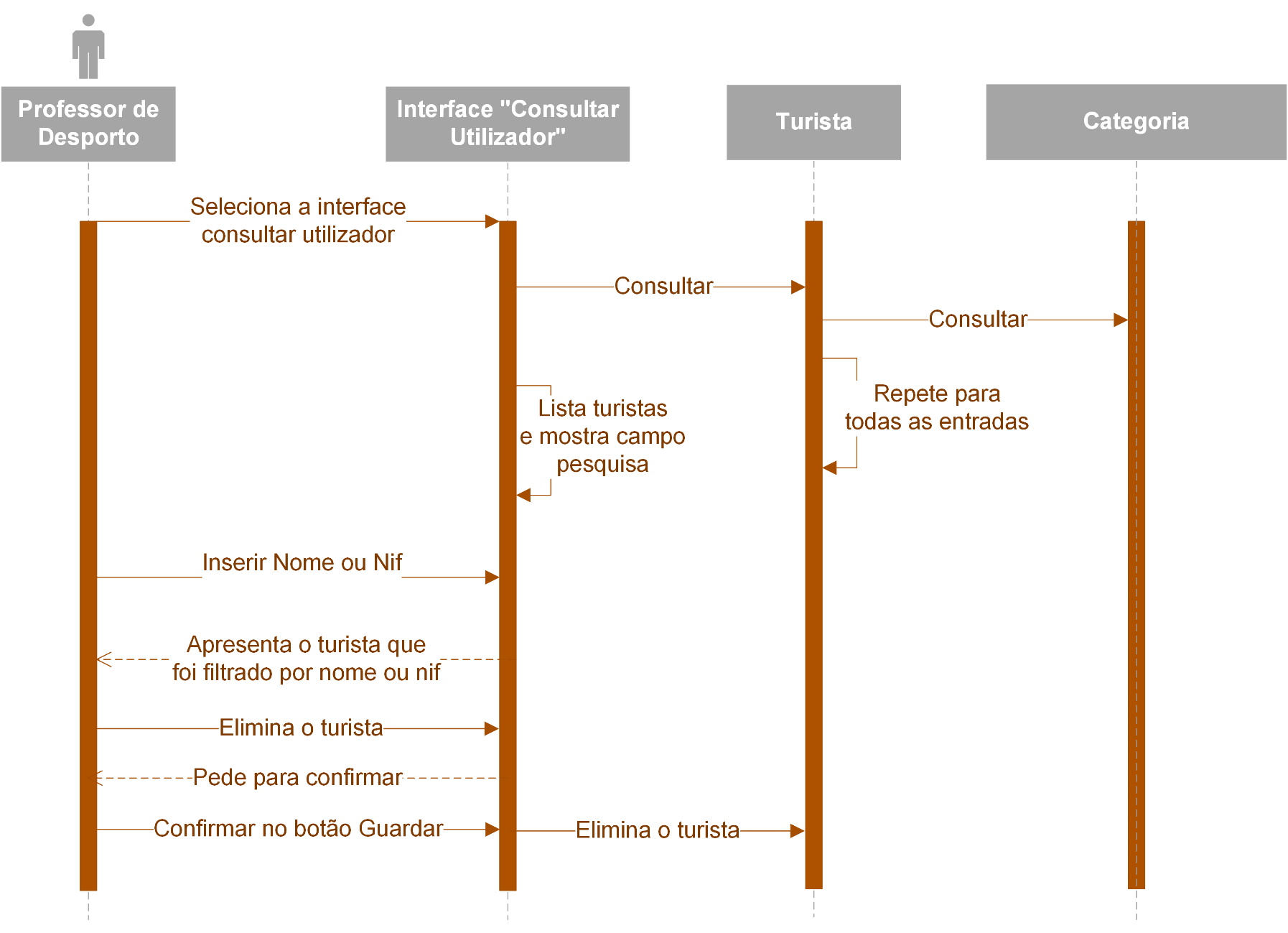
**Inserir Utilizador**

**Consultar Utilizador**

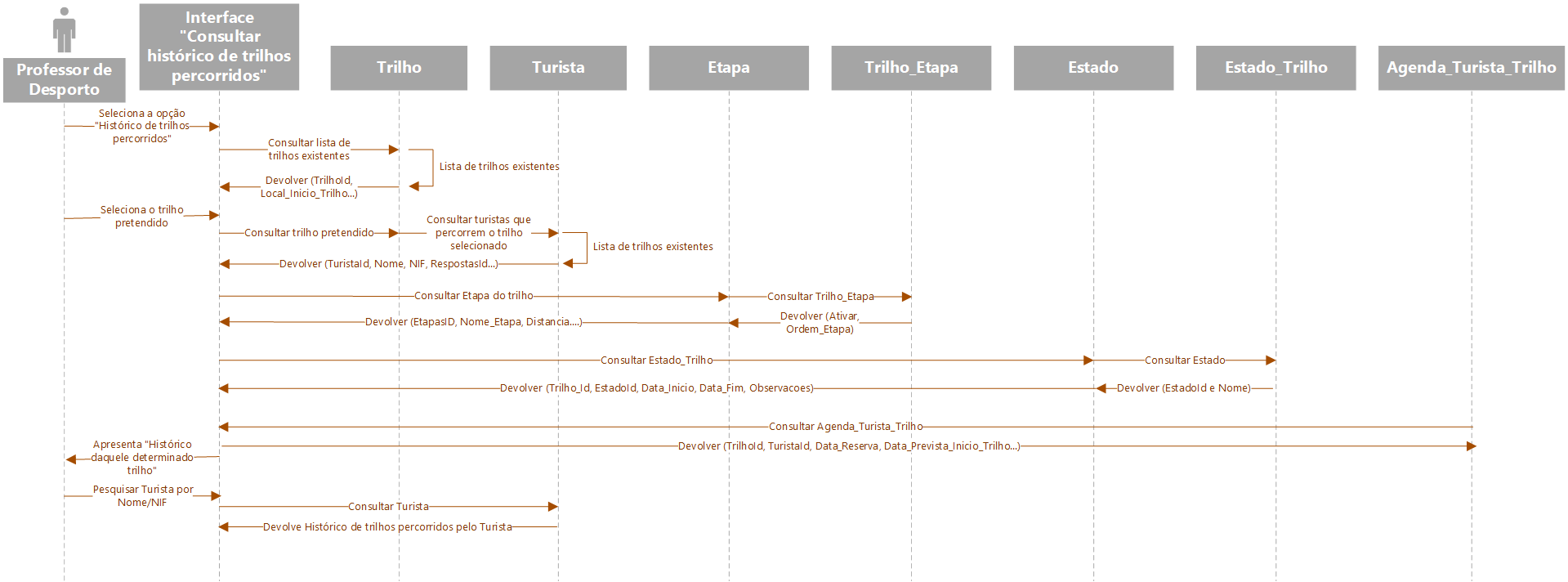
**Editar Utilizador**

****

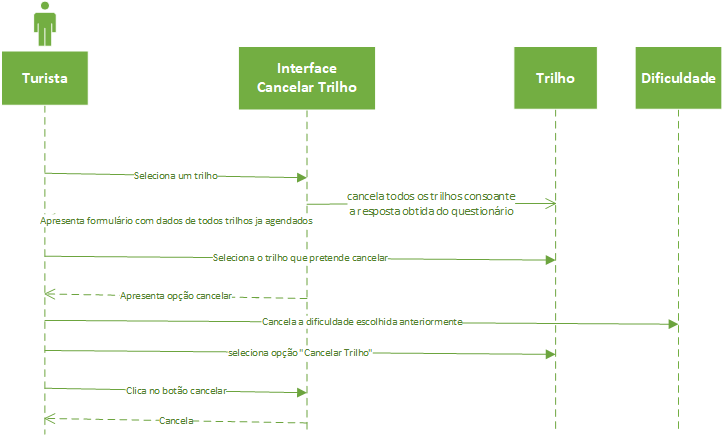
**Eliminar Utilizador**

****

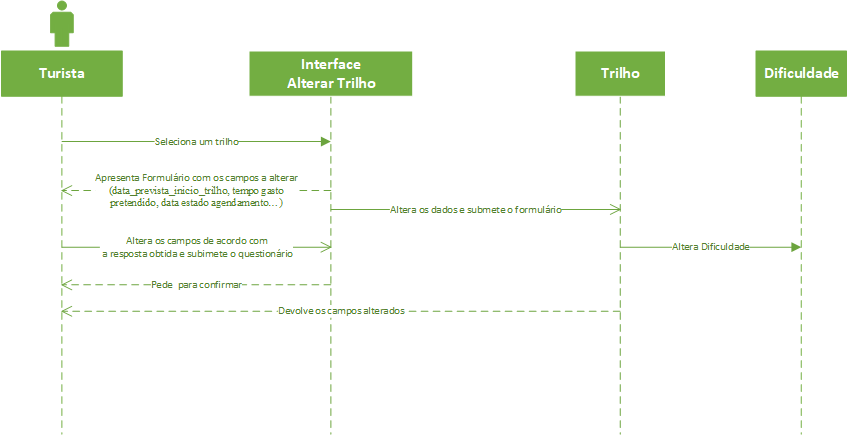
**Consultar histórico de trilhos percorridos**



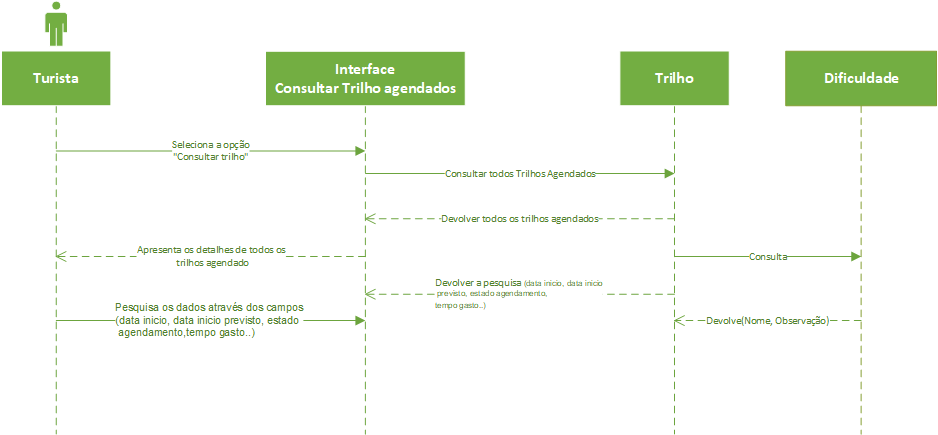
**Agendar Trilho**



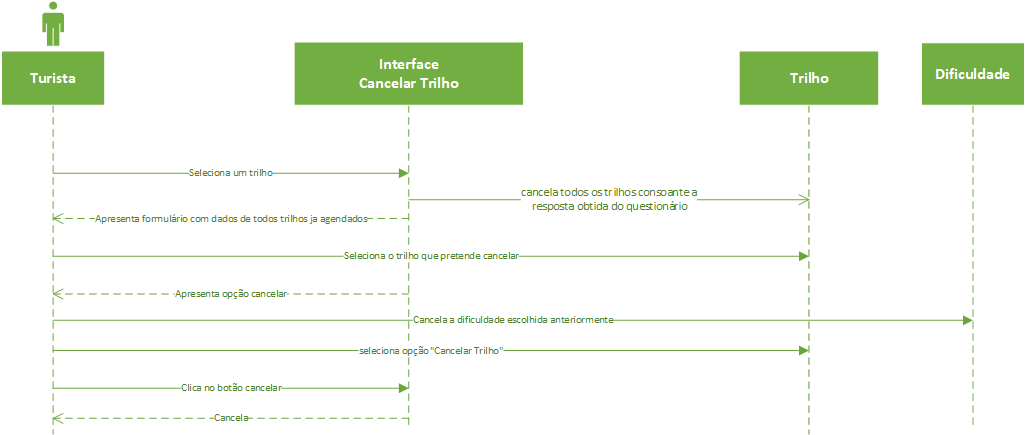
**Alterar Trilho Agendado**



**Consultar Trilho Agendado**



**Cancelar Trilho Agendado**



# Diagramas de Classes completo

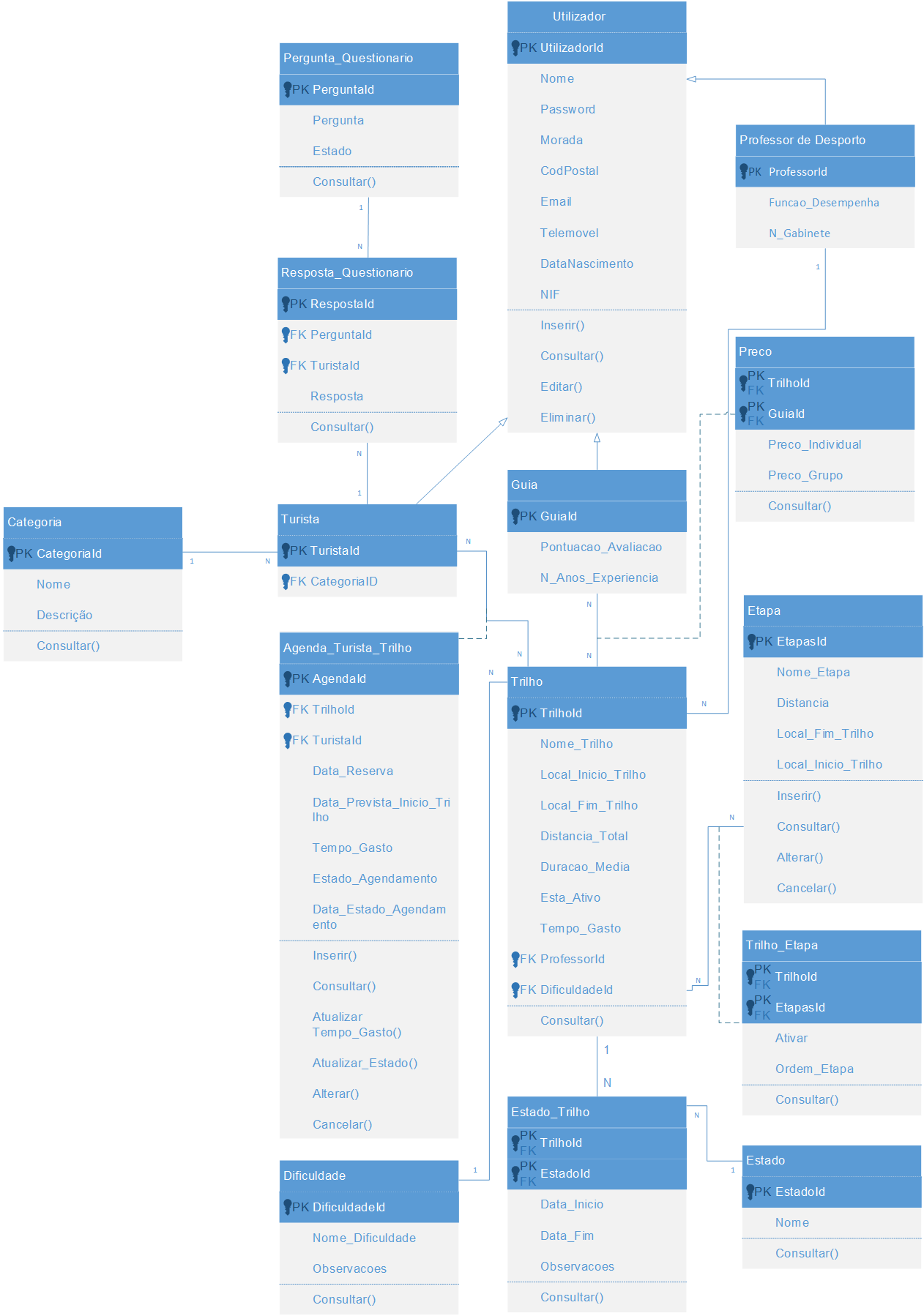


Figura 3 - Diagrama de classes

# Diagrama de Estados

**Agendar Trilho**

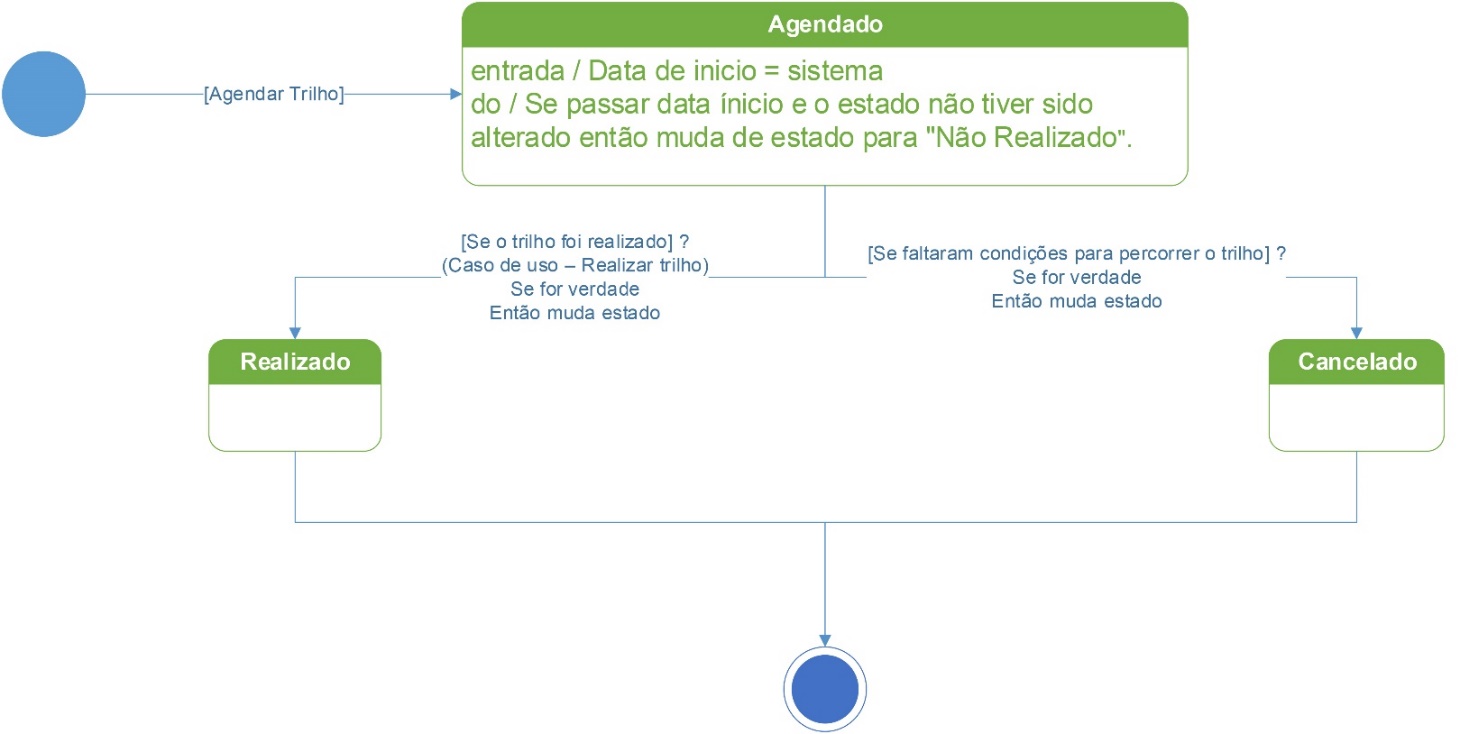


Figura 4 - Diagrama do estado Agendar Trilho

# Diagrama de atividades

# Diagrama de componentes

Este tipo de diagrama permite descrever os diversos “pedaços” de software que são os programas fonte, bibliotecas ou programas executáveis.

A figura abaixo apresentada ilustra o diagrama de componentes deste projeto.

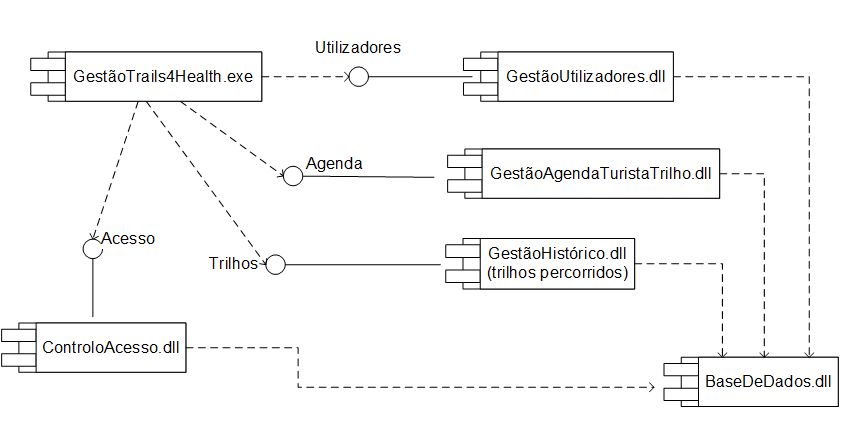


Figura 5 - Diagrama de componentes Trails4Health

# Diagrama de pacotes

# Diagrama de instalação

Este diagrama permite descrever a arquitetura do sistema em termos de hardware e a sua relação com os diferentes componentes (software).

A figura abaixo apresentada ilustra o diagrama de instalação deste projeto.

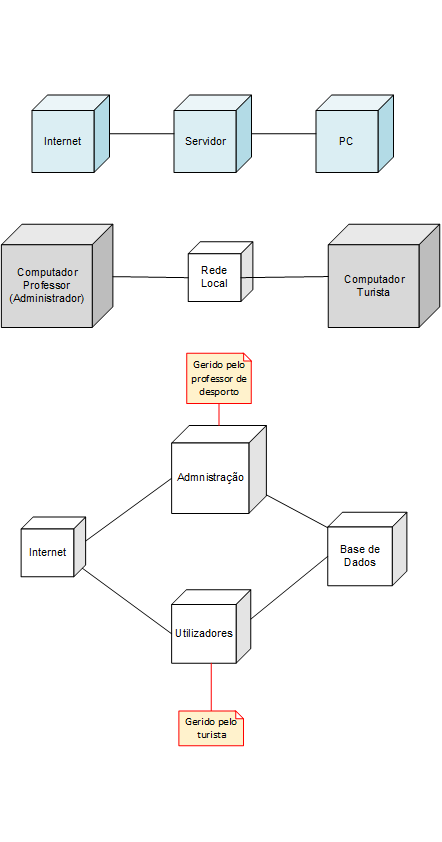


Figura 6 - Diagrama de instalação

# Semântica de Classes

## Classe Utilizador (Turista e Professor)

**Algoritmo do dígito de controlo do NIF**

1. Multiplicar:

8.º dígito por 2

7.º dígito por 3

6.º dígito por 4

5.º dígito por 5

4.º dígito por 6

3.º dígito por 7

2.º dígito por 8

1.º digito por 9

2. Somar todos os resultados.

3. Calcular o resto da divisão da soma por 11 (ou seja, o Módulo de 11)

4. Se resto for igual 0 ou a 1 o dígito de controlo será 0.

5. Senão o dígito de controlo é 11-resto.

## Classe Turista:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Classe: Professor, | | | | | |
| Nome do Campo | Tipo de dados | Descrição | Valores Válidos | Formato | Restrições |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Dicionário de dados:**

**Descrição das Operações:**

**Diagramas de sequência em que a classe participa:**

## Classe Agenda\_Turista\_Trilho:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Classe: Agenda\_Turista­\_Trilho, tem como objectivo o turista agendar o trilho que pretende percorrer | | | | | |
| Nome do Campo | Tipo de dados | Descrição | Valores Válidos | Formato | Restrições |
| TrilhoId (FK) | Numerico | Indica o ID de um trilho | --------- | Selecionado pelo utilizador | Obrigatório/não alterável |
| TuristaId(FK) | Numerico | Indica o ID de um turista | --------- | Botão opção equivalente | Obrigatório/não alterável |
| Data\_Reserva | Data Time | Data efectuado pelo turista | AAAA/MM/DD | Gerado pelo sistema | Obrigatório/Alterável/ Preenchido pelo utilizador |
| Data\_Prevista\_Inicio\_Trilho | Data Time | Dia previsto para percorrer o trilho | AAAA/MM/DD | Botão opção equivalente | Obrigatório/Alterável/  Preenchido pelo utilizador |
| Tempo\_Gasto | Numerico | Tempo que foi gasto durante o percurso | HH:MM:SS | ---- | ----- |
| Estado\_Agendamento | String | Indica o estado do agendamento do trilho se foi (realizado, cancelado…) | -------- | ----- | ----- |
| Data\_Estado\_Agendamento | Data Time | Indica o dia como está o agendamento do trilho | AAAA/MM/DD | ------ | ------ |

**Dicionário de dados:**

**Descrição das Operações:**

**Classe** Agenda\_Turista­\_Trilho

|  |  |
| --- | --- |
| **Nome** | **Descrição** |
| Criar()// **Get** |  |
| Criar()// **Post** | Operação que permite criar um novo Agendamento   1. Seleciona o Trilho 2. Sistema gera TuristaId   Turista seleciona a data prevista para realizar o trilho   1. Se todos os campos preenchido com execução 2. Cria nova agenda |

**Classe** Agenda\_Turista­\_Trilho +Alterar()

1. Sistema disponibliza todos os trilgos já agendados
2. O turista **altera** TrilhoId, TuristaId, Data\_Reserva, Data\_Prevista\_Agendamento, Tempo\_Gasto, Estado\_Agendamento, Data\_Estado\_Agendamento
3. Se os campos todos alterados
4. Atualizar Agenda\_Turista­\_Trilho
5. Fim

**Classe** Agenda\_Turista­\_Trilho +Consultar()

1. Introduzir o campo na qual deseja consultar por: TrilhoId, TuristaId, Data\_Reserva, Data\_Prevista\_Agendamento, Tempo\_Gasto, Estado\_Agendamento, Data\_Estado\_Agendamento
2. Atualiza o Agenda\_Turista­\_Trilho
3. fim

**Classe** Agenda\_Turista­\_Trilho +Cancelar()

1. Se a data prevista do agendamento do trilho corresponder a disponibilidade do turista
2. Devolver “Não pode cancelar o trilho”
3. Senão
4. Cancelar o Agenda\_Turista­\_Trilho
5. fim

**Diagramas de sequência em que a classe participa:**

## Classe Trilho:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Classe: Professor, | | | | | |
| Nome do Campo | Tipo de dados | Descrição | Valores Válidos | Formato | Restrições |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

**Dicionário de dados:**

**Descrição das Operações:**

**Diagramas de sequência em que a classe participa:**